

Previsió del temps per posicionament

Ride Sweetly - website

The logo of the Universitat Oberta de Catalunya (UOC) is displayed in a large, bold, dark blue font. The letters 'U', 'O', and 'C' are visible, with the 'U' and 'O' being significantly larger than the 'C'.

Josep Vallejo Duarri

Multimèdia

Desenvolupament d'aplicacions interactives

Tutor/a de TF

Lluís Beltrà Valenzuela

Professor/a responsable de l'assignatura

Carlos Casado Martínez

06/2023

Universitat Oberta
de Catalunya



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FITXA DEL TREBALL FINAL

Títol del treball:	Previsió del temps per posicionament
Nom de l'autor:	Josep Vallejo
Nom del consultor/a:	Lluís Beltrà Valenzuela
Nom del PRA:	Carlos Casado Martínez
Data de lliurament (mm/aaaa):	06/2023
Titulació o programa:	Grau de Multimèdia
Àrea del Treball Final:	Desenvolupament d'aplicacions interactives
Idioma del treball:	Català
Paraules clau	Geolocalització, <i>forecast</i> .
Resum del Treball	
<p>El ciclisme és una de les activitats a l'aire lliure que ha vist incrementada la seva popularitat en els darrers anys i, amb l'objectiu de seguir cobrint les necessitats que poden sorgir dins d'aquest col·lectiu, s'ha desenvolupat l'eina (amb forma de web) Ride Sweetly.</p> <p>Basant-nos en la predicció meteorològica en uns llocs determinats, obtinguda gràcies a les API de Google Maps i OpenWeather, accedim a un inventari que ha registrat prèviament l'usuari. Dins aquest inventari, trobem diferents tipus de peces de roba i complements per a ciclistes amb uns rangs de temperatures associats.</p> <p>Comparant la informació obtinguda de la previsió i la disponible a l'inventari de l'usuari, l'aplicació rastreja i selecciona les peces de roba i complements més adients, sempre segons la tipologia i rang de temperatures disponibles, amb l'objectiu d'oferir la millor proposta de vestimenta ciclista.</p> <p>Tot això a través d'una autenticació d'usuari, amb la finalitat de poder disposar d'un inventari personalitzat i completament editable per a cadascun d'aquests.</p>	

Abstract

Cycling is an outdoor activity that has increase its popularity during the last years and, with the purpose of continue covering the needs that could appear into this group of people, we launch the tool (as a website) Ride Sweetly.

Based on weather forecast of specific places that we can get using Google Maps and Open Weather APIs, we access an inventory that users have previously registered on their profile. In this inventory, we find all kind of cycling clothes and complements with a range of temperatures linked to each item.

Comparing the data that we get from weather forecast with the data in the user's inventory, the app scan and select, considering type of clothing, the available items matching with temperature's range. The unique purpose is to show the best cycling outfit to the user.

We are going to do all we mentioned before through user's authentication. This way, we are going to offer a personal and fully editable inventory to each user.

Índex

1.	Introducció	1
1.1.	Context i justificació del Treball	1
1.2.	Objectius del Treball	2
1.3.	Impacte en sostenibilitat, ètic-social i de diversitat.....	3
1.4.	Enfocament i mètode seguit.....	4
1.5.	Planificació del Treball.....	6
1.6.	Breu descripció dels altres capítols de la memòria.....	9
2.	Materials i mètodes	11
2.1.	Configuració i test inicial de l'API Maps JavaScript.....	11
2.2.	Configuració i test inicial de l'API Open Weather	15
2.3.	Integració APIs	18
2.4.	Arquitectura bàsica del lloc web	22
2.5.	Servidor local i base de dades	35
2.6.	Funcionament web i recollida de dades	38
2.7.	Autenticació d'usuari amb Firebase	45
2.8.	Branding.....	52
2.9.	<i>Mockups</i> i prototip	56
2.10.	Compliment de requeriments legals a la pàgina	68
2.11.	Desplegament web	68
3.	Resultats.....	71
4.	Conclusions i treballs futurs	72
5.	Glossari.....	77
6.	Bibliografia	80
7.	Annexos.....	81

Llista de figures

Figura 1. Diagrama del sistema	7
Figura 2. Creació d'un projecte a Google Cloud	11
Figura 3. Obtenció de la clau de Maps JavaScript API	12
Figura 4. Codi de crida a l'API de Maps JavaScript	12
Figura 5. Estructura pàgina web	23
Figura 6. Menú desplegable de Wampserver	35
Figura 7. Retall de la pantalla localhost de Wampserver	35
Figura 8. Retall de la pantalla Base de datos de phpMyAdmin	36
Figura 9. Retall de la pantalla Estructura de phpMyAdmin	36
Figura 10. Instrucció SQL d'introducció de dades per defecte	37
Figura 11. Flux pàgina inicial. Afegint localitzacions	38
Figura 12. Flux pàgina inicial. Previ a la petició	39
Figura 13. Flux de la petició i gestió de la informació obtinguda	42
Figura 14. Creació de projecte a Firebase	46
Figura 15. Vinculant <i>app</i> a Firebase	46
Figura 16. Obtenint <i>script</i> de Firebase	47
Figura 17. Habilitar proveïdor a Firebase Authentication	48
Figura 18. Referència CDN de Firebase Authentication	48
Figura 19. Logotips del sector	53
Figura 20. Paleta de colors de Ride Sweetly	53
Figura 21. Logotip de Ride Sweetly	54
Figura 22. Exemple d'aplicació de l'isotip	55
Figura 23. Tipografia Quicksand	55
Figura 24. Pantalla de creació de la base de dades	69

1. Introducció

1.1. Context i justificació del Treball

En els últims 3 anys, des de l'inici de la pandèmia provocada pel coronavirus [SARS-CoV-2 \(COVID-19\)](#), **l'afició a l'esport s'ha vist incrementada de manera exponencial** [1] amb el pas dels mesos. Amb aquest context, el ciclisme ha guanyat una popularitat mai vista dins de totes les capes de la societat.

Tot i que aquesta tendència sembla haver tocat sostre, la consciència sobre el benestar, la alimentació, i la necessitat de fer esport per gaudir d'una vida saludable, ha arrelat dins la població i, especialment, dins el jovent. Per tant, sembla un sector interessant i amb potencial per als propers anys.

Actualment, s'ha cobert amb multitud d'aplicacions, pàgines web i vídeos, moltes de les necessitats que poden tenir les persones que formen part del col·lectiu ciclista: enregistrar una activitat i compartir-la, programar un recorregut i obtenir una previsió de la durada d'aquest segons la teva forma física, manteniment bàsic i avançat de la bicicleta... i molta més informació. Ara bé, tot i que al sector **trobem aplicacions que mostren la temperatura, l'evolució d'aquesta (*forecast*¹) i més informació relativa** com, per exemple, la humitat (normalment es tracta d'una funcionalitat que requereix d'una subscripció de pagament) **cap aplicació analitza o gestiona la informació obtinguda per proposar-te un *outfit*² adaptat a les condicions.**

Amb l'aplicació [Ride Sweetly](#) es pretén cobrir aquesta necessitat que encara no ha cobert cap competidor potencial, ja que es tracta d'un dels dilemes més recurrents que té tot ciclista abans de sortir de casa: "Quina roba és la més adient per sortir avui?" L'aplicació disposarà de la roba i complements que haurà introduït prèviament l'usuari i li suggerirà la millor opció dins del que estigui disponible. En cas de no trobar els elements adequats, s'informarà a l'usuari d'aquest faltant.

Es tracta, a part de recollir aquesta informació sobre la previsió del temps i senzillament mostrar-la, de gestionar-la i personalitzar-la amb l'única intenció de donar valor afegit a la informació obtinguda i, així, millorar l'experiència del ciclista al llarg de qualsevol ruta que vulgui realitzar.

Competidors principals i oportunitat

Dins d'aquest sector trobem actors principals com **Strava, Komoot o Wikiloc** [2]. Aquestes aplicacions tan populars tenen una capacitat econòmica i de desenvolupament molt gran, que fan complicat el posicionament del nostre aplicatiu

dins del seu quadrant de mercat. Per tant, **la nostra oportunitat (a mig termini) passar per posicionar-nos com un complement d'aquestes aplicacions.** S'aprofitaria el treball de màrqueting d'aquestes grans empreses per associar el nostre producte com un valor afegit.

Després del desenvolupament web i la creació d'una estructura sòlida per al nostre projecte, es podria iniciar el desenvolupament d'una aplicació mòbil que es pogués vincular a alguna d'aquestes aplicacions anteriorment mencionades. Així, connectant Ride Sweetly a Komoot (per exemple) obtindríem la informació de la ruta planificada per l'usuari i podríem oferir la informació sobre la millor roba i complements a dur.

Finalment, aprofitant la integració amb altres aplicacions com Garmin Connect o Strava, que ofereixen una gran varietat d'esports a triar per l'usuari, podríem **obrir l'àmbit de Ride Sweetly i així abastar tots els esports possibles.**

1.2. Objectius del Treball

El resultat final del treball és la creació d'una pàgina web, adaptable a la pantalla de qualsevol dispositiu, amb la funció principal de mostrar el pronòstic del temps en uns llocs i moments determinats per tal de suggerir un *outfit* ciclista totalment personalitzat a l'usuari.

Per assolir l'objectiu principal, haurem de:

- Obtenir els paràmetres de latitud i longitud d'un punt en concret del mapa (tipus Google Maps).
- Obtenir la informació sobre la previsió del temps en un lloc (latitud i longitud) específic.
- Permetre l'enviament, en el format adequat, de la informació de posicionament obtinguda a l'API³ de previsió del temps.
- Permetre l'autenticació d'usuari i emmagatzemar informació personalitzada en una base de dades.
- Crear una pàgina de perfil on l'usuari pugui afegir, eliminar i editar l'equipament disponible i els rangs de temperatura associats a cada peça de roba, a part d'informació bàsica.

- Programar un algoritme que gestioni la informació del *forecast* obtinguda, comparant-la amb les dades de l'usuari i mostrant els resultats seleccionats.
- Dissenyar una arquitectura bàsica per la web.

1.3. Impacte en sostenibilitat, ètic-social i de diversitat

Sostenibilitat

La producció de qualsevol tipus de roba per part de les grans empreses d'aquest sector, a dia d'avui, comporta el malbaratament d'aigua i altres recursos naturals, a part de la contaminació associada als processos de producció. Tot i que cada vegada hi ha una conscienciació més forta sobre la sostenibilitat, i les marques comencen a incloure aquest aspecte per tal d'oferir valor afegit als seus productes [3], **l'usuari final té l'última paraula**, triant marques o botigues que garanteixin un procés el més sostenible possible.

El resultat final del treball, ofereix l'oportunitat a cada consumidor de detectar únicament aquelles peces de roba que necessita, evitant la compra de productes innecessaris fomentat pel màrqueting de les empreses i el consumisme exagerat que ja està arrelat a la societat actual.

També, el producte final d'aquest treball, fomenta i intenta donar una eina més al sector ciclista, per tal d'**afavorir i facilitar l'ús de la bicicleta**, ja sigui per la realització d'exercici com per fer *commuting*⁴ sostenible, evitant l'ús d'alternatives contaminants.

Per tant, de forma indirecta, Ride Sweetly col·labora amb els objectius de:

- ODS⁵ 11: Ciutats i comunitats sostenibles.
- ODS 12: Consum i producció responsables.
- ODS 13: Acció climàtica.

Comportament ètic i responsabilitat social

Aquest treball es podria fer girar entorn al consumisme i, mitjançant la inserció de publicitat (aprofitant la informació disponible dels usuaris), fomentar compres impulsives o crear falses necessitats als usuaris.

Però, amb Ride Sweetly, **es vol girar entorn l'usuari i no entorn als objectius de grans empreses**. Per això, és vital no inserir publicitat a la web que pugui fomentar un consumisme poc ètic i responsable en aquesta primera fase. Més endavant, si sorgís la necessitat de rendibilitzar el projecte, seria més honest convertir-la en una aplicació de pagament i, respectant la privacitat del client, crear una secció de recomanacions genèriques amb productes i marques que compleixin uns requisits fonamentals com la sostenibilitat i la responsabilitat social, creant sinergies entre marques amigues.

Tot plegat fomentaria un comportament ètic, responsable, i l'aparició d'oportunitats laborals:

- ODS 8: Treball decent i creixement econòmic.

Diversitat i drets humans

El resultat d'aquest treball va dirigit a totes les persones que utilitzin o vulguin utilitzar la bicicleta en el seu dia a dia, independentment de la raça, religió, ètnia, ideologia o orientació sexual, entre d'altres. Cap funcionalitat de Ride Sweetly requereix cap informació prèviament esmentada, oferint igualtat de resultats per a tothom.

El més rellevant d'aquest aspecte serà controlar l'ús de grafismes al lloc web i controlar el masclisme ocult que trobem en moltes situacions que donem per normalitzades:

- OD5: Igualtat de gènere.
- OD10: Desigualtats reduïdes.

1.4. Enfocament i mètode seguit

Aquest projecte es planteja en tres fases: una d'inicial on **es proposa la creació d'un lloc web amb les funcionalitats principals (objectiu d'aquest TFG ⁶)**, una segona on es desenvolupa una aplicació per als principals sistemes ([Android](#) ⁷ i [iOS](#) ⁸) i, finalment, una tercera fase on es busca la integració de l'eina amb les *apps* ⁹ més importants del sector, com Strava, Komoot, etc.

- Fase 1: Llençar i testar els resultats del producte web al mercat.

- Fase 2: Consolidar la proposta amb el desenvolupament de l'aplicació corresponent.
- Fase 3: Integar la solució amb les aplicacions més conegudes del mercat.

L'estratègia, més que desenvolupar un producte nou, tracta de crear un subproducte, ja que actualment existeixen moltes aplicacions que treballen amb la mateixa informació que la nostra proposta. **El punt diferenciador recau en la gestió d'aquesta informació (previsió del temps) i com se li dona un valor afegit que cobreix una necessitat.**

Centrant-nos en la fase 1, que és la que ens ocupa, s'ha decidit crear el producte en forma de web pels següents motius:

- Permet un desenvolupament versàtil, ja que amb llenguatges com HTML ¹⁰, CSS ¹¹, i JavaScript ¹², es pot programar tot el necessari i permeten tot tipus de comunicacions.
- Permet l'adaptació a tot tipus de dispositius.
- Permet un desenvolupament més ràpid, ja que per *timings* i coneixements, no és possible l'execució amb altres llenguatges o sistemes de programació.
- Permetrà testejar el producte en un entorn molt semblant al d'una aplicació, afavorint l'enfocament de les fases futures.
- Permet l'aprofitament, per a les següents fases, de part del codi i estructura del projecte.
- És un producte que pot conviure amb les futures aplicacions, oferint sempre una opció web.

És important entendre que el meu perfil no és el d'un programador amb una base sòlida de conceptes i coneixements. Per tant, haig de ser conseqüent tant amb les capacitats d'un mateix com amb la tria de tecnologies i llenguatges a utilitzar. Els *timings* són molt justos i d'una altra manera seria molt complicat assolir l'objectiu principal.

1.5. Planificació del Treball

Les tasques necessàries per assolir l'objectiu principal amb èxit són:

- **Configurar i testar [API de Google Maps](#):** Necessitarem obtenir els paràmetres de latitud i longitud que l'usuari vagi clicant sobre un mapa.
- **Configurar i testar [API Open Weather Map](#):** Comprovar el funcionament de l'aplicació i requeriments.
- **Integrar APIs de Google Maps i Open Weather Map:** Assegurar que al clicar sobre el mapa es transmet la informació necessària a l'API de previsió del temps i que aquesta mostra els resultats.
- **Dissenyar una arquitectura bàsica del lloc web:** Abans de seguir, definirem un estructura bàsica per seguir desenvolupant la pàgina web amb certes garanties.
- **Configurar servidor local:** Per a realitzar les primeres proves, utilitzarem [WAMP](#)¹³ com a servidor.
- **Crear i definir base de dades:** Creació d'una base de dades [MySQL](#)¹⁴ i definició d'una taula i els seus atributs.
- **Introduir valors per defecte a la base de dades:** Tot i que cada usuari s'haurà d'autenticar i disposarà del seu inventari personalitzat, en aquesta primera fase treballarem amb valors per defecte.
- **Desenvolupar algoritme de recol·lecció de dades:** Serà necessari per recol·lectar i mostrar per pantalla els ítems de la BBDD¹⁵ que compleixin els requisits de temperatura obtinguts de la previsió del temps.
- **Configurar autenticació amb Firebase:** Un cop assolit un funcionament estable de tot l'anterior, passarem a afegir l'autenticació d'usuari, per tal d'afegir la personalització que volem oferir a cada usuari. Per a aquesta finalitat ens recolzarem amb la plataforma [Firebase](#) de Google. En aquest cas farem ús de [Firebase Authentication](#).
- **Crear i definir base de dades definitiva:** A continuació, fent ús de MySQL, crearem la base de dades definitiva del projecte a un servidor que tinc contractat per ús personal a [Dondominio](#).

- **Desplegar i testar totes les funcionalitats de la pàgina web.**

Un cop finalitzades totes les tasques, la nostra pàgina web hauria de connectar-se a Firebase Authentication per iniciar la sessió de l'usuari corresponent. A continuació, faria utilització de les API de Google Maps i Open Weather per obtenir les temperatures de cada posicionament donat per, finalment, mitjançant la informació de la base de dades de l'usuari seleccionat, mostrar aquesta per pantalla:

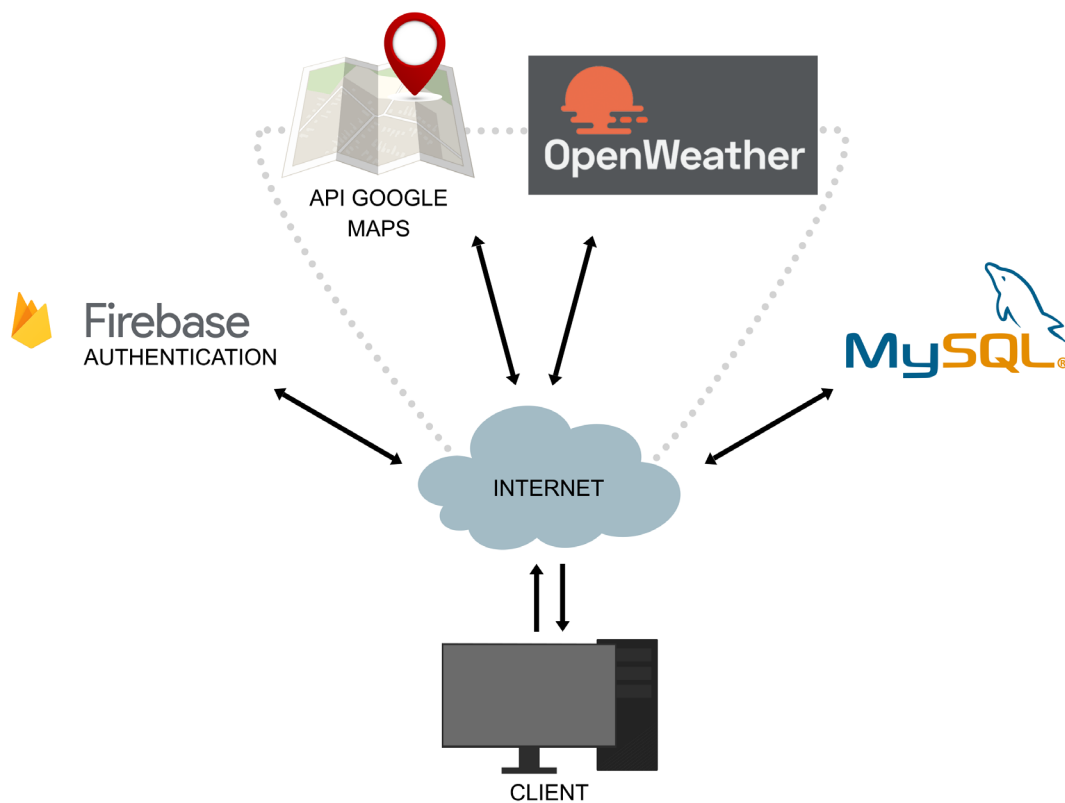


Figura 1. Diagrama del sistema.

A continuació mostrem un diagrama de Gantt ¹⁶ amb les tasques a realitzar:

1.6. Breu descripció dels altres capítols de la memòria

Capítol 2.1: Configuració i test inicial de l'API Maps JavaScript

Configuració de l'API que ens permetrà carregar Google Maps a la pàgina web i poder disposar dels paràmetres de latitud i longitud que l'usuari vagi seleccionant. També, es programa un petit codi per comprovar el seu funcionament.

Capítol 2.2: Configuració i test inicial de l'API Open Weather

Configuració de l'API que ens permetrà, un vegada li hem passat una localització (com, per exemple, els paràmetres de latitud i longitud d'un punt específic obtinguts de Google Maps), obtenir la previsió del temps d'aquesta. S'expliquen algunes de les seves característiques i també es prepara un petit codi per testejar el seu funcionament.

Capítol 2.3: Integració APIs

Configuració i metodologia utilitzada per integrar el funcionament de l'API de Google Maps i Open Weather Map per tal d'anar obtenint resultats dels punts seleccionats.

Capítol 2.4: Arquitectura bàsica del lloc web

Es descompon la pàgina web en el que seran les seves seccions i estructura bàsica, finalitzant en el disseny del *wireframes*¹⁷ necessaris.

Capítol 2.5: Servidor local i base de dades

Configuració del servidor local i creació de la base de dades que ens permetran anar desenvolupant la nostra pàgina web des del *localhost*¹⁸. En aquest capítol s'utilitza l'eina Wampserver, que ens proveeix del servidor i de [phpMyAdmin](#)¹⁹, mitjançant el qual crearem la base de dades amb MySQL.

Capítol 2.6: Funcionament web i recollida de dades

Durant aquest capítol es defineixen i es programen les comprovacions i la petició principals de la nostra pàgina web. S'expliquen les bases del funcionament, a partir del flux d'informació i la seva gestió, fins convertir-la en una petició vàlida que obtindrà la informació necessària de la base de dades.

Capítol 2.7: Autenticació d'usuari amb Firebase

Configuració de Firebase, sistema que ens permetrà crear i autenticar usuaris a la nostra pàgina web, des de la creació del projecte a la plataforma fins a la programació del codi necessari a la pàgina.

Capítol 2.8: Branding

Breu explicació sobre les línies visuals bàsiques sobre les que es construirà la imatge de marca de Ride Sweetly.

Capítol 2.9: Compliment de requeriments legals a la pàgina

Nota sobre punts legals, com l'ús de *cookies*²⁰ o la política de privacitat, que la llei ens obliga a informar a la pàgina. No es desenvolupen els punts, sinó que simplement s'adverteix de la necessitat d'elaborar-los correctament en un hipotètic llançament real del producte.

Capítol 2.10: Desplegament web

Explicació sobre el desplegament de la pàgina web amb l'objectiu de posar-la accessible a usuaris. Des de la creació final de la base de dades MySQL i l'allotjament de tota la configuració de la pàgina web a un servidor de DonDominio.

2. Materials i mètodes

2.1. Configuració i test inicial de l'API Maps JavaScript

Per gestionar tot el referent al posicionament s'ha triat la utilització de la plataforma [Google Cloud](#). Específicament, d'una de les APIs que formen part de la secció de Google Maps: **Maps JavaScript API**. Tot i que es tracta d'una plataforma de pagament, ofereix un període de prova de 90 dies amb un crèdit mensual de 300\$. Després d'aquest període, es pot mantenir el seu ús gratuït sempre que et mantinguis per sota de les 28.500 descàrregues per mes (que equival a uns 200 \$ de crèdit al mes).

Aquesta API ens permet crear el mapa que utilitzarem per a la nostra pàgina web i ens ofereix un ventall de mètodes i guies per crear multitud d'interaccions entre el mapa i l'usuari. En una primera instància, voldrem marcar sobre al mapa la geolocalització actual del client i, a partir d'aquí, que el propi usuari pugui anar actualitzant el marcador sobre el mapa, clicant directament sobre aquest. L'objectiu és obtenir les dades de latitud i longitud de tots els punts d'interès que l'usuari necessiti per, posteriorment, utilitzar-los per cercar el *forecast* del temps.

Configuració

Primer, necessitarem tenir un compte de Google per poder accedir a la consola de Google Cloud. Un cop dins de la consola, crearem el nostre projecte.

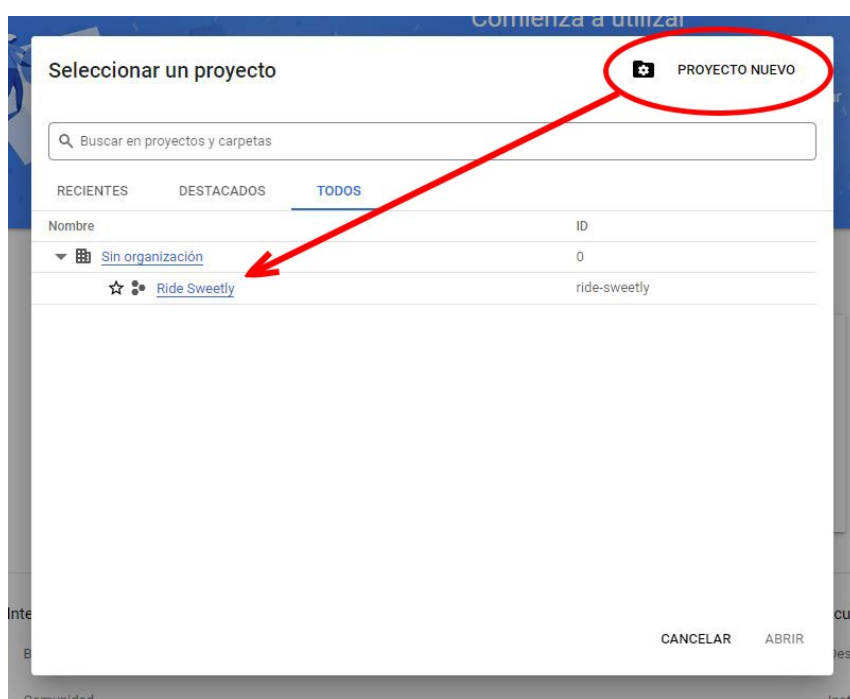


Figura 2. Creació d'un projecte a Google Cloud.

Després d'activar la prova gratuïta a Google Cloud, cercarem i habilitarem l'API de Maps JavaScript. És **imprescindible disposar d'una clau API** per poder-la cridar des del codi JavaScript de la nostra pàgina web. Per obtenir aquesta clau, només s'haurà d'accedir al menú lateral i entrar on indica: Credencials.



Figura 3. Obtenció de la clau de Maps JavaScript API.

Ara, només necessitarem inserir el codi que ens proporcionen dins de la documentació de l'API a la nostra pàgina web, substituint la part on posa "YOUR_API_KEY" per la clau obtinguda prèviament.

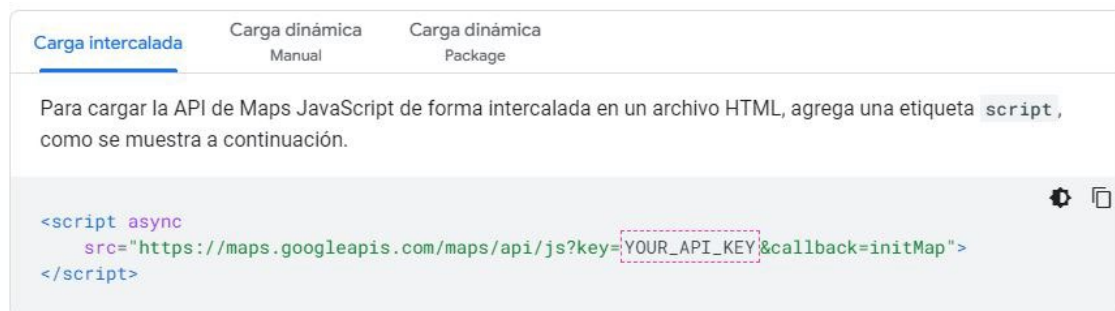


Figura 4. Codi de crida a l'API de Maps JavaScript.

Per testejar el funcionament de l'API, a la nostra pàgina executarem la funció `setPosition()` amb l'única intenció d'obtenir la geolocalització del navegador i, així, poder crear el mapa amb un marcador indicant el posicionament actual.

```
// Establim el posicionament inicial del dispositiu a través de la geolocalització del navegador:
setPosition();
function setPosition() {
    // Si el navegador permet la geolocalització,
    disparem el mètode getCurrentPosition():
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(success, error, options);
    }
    // Si el navegador no permet la geolocalització, disparem directament error():
```

```

    } else {
        error();
    }
};

```

Amb `navigator.geolocation` sabem si el navegador disposa de la geolocalització habilitada i amb `getCurrentPosition()` gestionem la informació obtinguda. En aquest últim pas, disposem de tres variables:

- `success`: Obté la informació de posicionament i la passa com arguments per executar la funció `initMap()`.
- `error`: En cas de tenir denegat l'accés a la informació de posicionament del navegador, senzillament inicialitza les variables amb uns valors per defecte (la latitud i la longitud de la ciutat de Nova York) per poder executar immediatament la funció `initMap()`;
- `options`: Definim un sèrie de requisits que s'indiquen a la figura 6.

A continuació s'adjunta el codi corresponent:

```

// Si tot és correcte i se'ns permet obtenir la geolocalització, es dispara
success():
function success(geolocationPosition) {
    // Obtenim els paràmetres de latitud i longitud de l'objecte geolocationPosition
    (coords.latitude i coords.longitude):
    let lat = geolocationPosition.coords.latitude;
    let lng = geolocationPosition.coords.longitude;

    initMap(lat, lng);
};

// Si no es pot obtenir la geolocalització del navegador, s'adverteix a l'usuari i
indiquem un posicionament per defecte (en aquest cas, el de la ciutat de Nova York):
function error(err) {
    let lat = 40.7142700;
    let lng = -74.0059700;

    initMap(lat, lng);

    alert("We can't get your position!");
};

var options = {

```

```

EnableHighAccuracy: true, // Habilitem un posicionament acurat.
Timeout: 3000, // Establim 3 segons com a temps d'espera màxim.
MaximunAge: 0 // Establim que volem el posicionament actual.
};

```

Com podem comprovar al codi anterior, la finalitat és executar la funció `initMap()` amb unes variables de posicionament com arguments. Aquesta funció la trobem a la documentació de l'API Maps JavaScript i és l'encarregada de crear el mapa. A més, s'ha afegit més codi a aquest mètode, per tal de carregar un marcador i poder-lo actualitzar clicant sobre el mapa.

```

function initMap(x, y) {
  let map;
  let location = { lat: x, lng: y };
  let marker;

  if (x !== undefined) { // Esperem a que arribi un valor vàlid abans d'executar la
    // funció.
    map = new google.maps.Map(document.getElementById("map"), {
      center: { lat: x, lng: y }, // Centrem el mapa en aquesta posició.
      zoom: 14, // Indiquem el zoom del mapa.
    });

    // Afegim el marcador:
    addMarker(location, map);

    // Cada vegada que fem click amb el ratolí sobre el mapa, afegirem un nou
    // marcador:
    google.maps.event.addListener(map, "click", (event) => {
      marker.setMap(null); // Primer, eliminem el marcador existent.
      addMarker(event.latLng, map);
      // Actualitzem la localització:
      location.lat = event.latLng.lat();
      location.lng = event.latLng.lng();
      console.log(location);
    });

    // Definim i situem un marcador:
    function addMarker(location, map) {
      marker = new google.maps.Marker({
        position: location,
        map: map
      });
    };
  };
}
};

```

A continuació ja podem crear una pàgina HTML bàsica amb una mica de CSS aplicat i comprovar que tot el codi funciona correctament.

2.2. Configuració i test inicial de l'API Open Weather

Per obtenir el **pronòstic del temps** utilitzarem una de les APIs més populars, com és **Open Weather Map**. Aquesta disposa de diferents subscripcions de pagament, però en el nostre cas serà suficient amb l'opció gratuïta, que ens permet, entre altres coses:

- 60 sol·licituds/minut.
- 1 milió de sol·licituds/mes.
- Temps actual.
- Previsió del temps per cada 3 hores durant 5 dies.
- Mapes bàsics del temps.
- Panell del temps.

A més, en cas de necessitat, es disposa de plans gratuïts per estudiants que permeten ampliar algunes de les opcions disponibles de l'API.

Configuració

Primer, necessitarem registrar-nos i automàticament ja disposarem d'una API Key que haurem d'utilitzar més endavant per a cada crida de l'API de Open Weather.

Per fer la petició utilitzarem el codi que trobem a la documentació amb un parell de consideracions:

```
https://api.openweathermap.org/data/2.5/forecast?lat=${lat}&lon=${lng}&cnt=2&appid=${apiWeatherKey}&units=metric
```

A `${lat}` y `${lng}` haurem d'indicar el valor de latitud i longitud respectivament. A més, afegim el tros de codi `&cnt=2`. Aquesta part de codi limita les respostes obtingudes de la petició a 2, que correspon a la informació de la previsió del temps de l'hora actual (el moment en que es realitza la petició) i d'aquí a tres hores. Per al

nostre objectiu no necessitarem més informació que aquesta, ja que ens interessa el pronòstic del temps just al sortir en bici. Finalment indiquem, afegint `&units=metric` al final de la petició, que volem que la unitat sigui en graus Celsius.

A continuació, s'ha de realitzar una petició [AJAX](#)²¹ a l'API de Open Weather Map amb tota la informació anterior. Utilitzarem una funció fletxa amb `async/await`. A partir d'aquí obtindrem una resposta en format [JSON](#)²² amb multitud de paràmetres, però passem a indicar els que ens interessarà gestionar:

- `list[0].main.feels_like`: Sensació tèrmica de temperatura actual.
- `list[0].weather[0].main`: Temps actual (Rain, Clear...).
- `list[0].wind.gust`: Velocitat actual de les ràfegues de vent.
- `list[1].main.feels_like`: Sensació tèrmica de temperatura a les 3 hores.
- `list[1].weather[0].main`: Temps al cap de 3 hores.
- `list[1].wind.gust`: Velocitat, al cap de 3 hores, de les ràfegues de vent.
- `city.name`: Zona on es calcula la previsió del temps.

Utilitzarem els paràmetres anteriors per obtenir la sensació tèrmica més alta i més baixa dels diferents punts que marqui l'usuari al mapa i, així, comparar aquesta informació amb la informació personalitzada de la base de dades. D'aquesta manera podrem mostrar el resultat que busquem: seleccionar els ítems de la base de dades que compleixen uns requisits de temperatura donats. A més, si les ràfegues de vent o el temps (el valor de `weather[0].main`) excedeix d'uns límits establerts, alertarem a l'usuari del possible perill.

Possibles valors de `weather[0].main`:

- Thunderstorm: Alerta
- Drizzle: Alerta
- Rain: Alerta
- Snow: Alerta
- Mist: Alerta
- Smoke: Alerta

- Haze: Alerta
- Fog: Alerta
- Sand: Alerta
- Dust: Alerta
- Ash: Alerta
- Squall: Alerta
- Tornado: Alerta
- Clear
- Clouds

Es pot consultar l'estructura de la resposta sencera JSON a <https://openweathermap.org/forecast5#JSON>

A continuació es pot veure el codi utilitzat per testejar l'API de Open Weather:

```
// URL per accedir a l'API. Conté els paràmetres de geolocalització (lat, lng) i la
clau API:
const url = `https://api.openweathermap.org/data/2.5/forecast?lat=${lat}&lon=${lng}&
nt=2&appid=${apiWeatherKey}&units=metric`;
// Funció fletxa per executar la petició AJAX i obtenir la informació de l'API
const getWeatherInfo = async () => {
  try {
    // Ens assegurem que no aprofiti res del cache.
    const response = await fetch(url, {cache: "no-cache"});
    // Si la resposta és correcte:
    if (response.ok) {
      // Obtenim la resposta en format json:
      const jsonResponse = await response.json();
      // A continuació obtenim la temperatura, el valor principal del paràmetre
      "weather", i el vent. Com es pot comprovar, agafem un valor inicial
      (valor actual) i un de final (valor al cap de 3 hores):
      let initial_temp = jsonResponse.list[0].main.feels_like;
      let final_temp = jsonResponse.list[1].main.feels_like;
      let initial_weather = jsonResponse.list[0].weather[0].main;
      let final_weather = jsonResponse.list[1].weather[0].main;
      let initial_wind = jsonResponse.list[0].wind.gust;
      let final_wind = jsonResponse.list[1].wind.gust;
      let city = jsonResponse.city.name; // Nom de la zona.

      console.log(initial_temp);
      console.log(final_temp);
      console.log(initial_weather);
      console.log(final_weather);
      console.log(initial_wind);
      console.log(final_wind);
    }
  }
}
```

```

        console.log(city);
    }
    // Si torna un error el mostrem a la consola:
    } catch (error) {
        console.log(error);
    }
};

```

2.3. Integració APIs

Un cop s'ha comprovat el correcte funcionament de les API Maps JavaScript i Open Weather Map, necessitem connectar-les de manera que, cada vegada que marquem un punt de posicionament sobre el mapa utilitzant Maps JavaScript, automàticament actualitzem les variables de latitud i longitud. D'aquesta forma, si fem la petició a l'API d'Open Weather Map per obtenir la previsió del temps, sempre disposarà del posicionament actualitzat.

Ho platejarem afegint un botó que, cada cop que el cliquem, dispararà la petició per obtenir la previsió del temps de les últimes dades de posicionament disponibles. També modificarem el *callback* ²³ del codi que carrega l'API de Maps JavaScript, canviant l'inicial `initMap` per `setPosition` (des d'aquesta última ja farem la crida a `initMap`).

El codi HTML bàsic quedaria així:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
  <meta name="description" content="Previsió del temps per posicionament | TGF - Desenvolupament d'aplicacions interactives">
  <meta name="keywords" content="forecast, geolocation">
  <meta name="author" content="Josep Vallejo Duarri">
  <meta name="application-name" content="ride sweetly">
  <title>ride sweetly</title>
  <link rel="shortcut icon" href="img/favicon.svg" />
  <link href='http://fonts.googleapis.com/css?family=Roboto:400,300,700' rel='stylesheet' type='text/css'>
  <link rel="stylesheet" type="text/css" href="main.css">
</head>

```



```

<body>
  <div id="map"></div>

  <!--Botó que dispara la crida a l'API de Open Weather Map-->
  <button id="forecastButton">Add point</button>

  <!--Carreguem l'API de Maps JavaScript:-->
  <script async
    src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBRBxqfhgLG9gJEiKpIVfAT
    5krU2dMg9Yc&callback=setPosition">
  </script>
  <script src="main.js"></script>
</body>

</html>

```

Tot i que s'ha explicat com s'han anat testejant les APIs corresponents, mostrem el **codi final de les dues API integrades:**

```

/*
  -----OBTENIR GEOLOCALITZACIÓ DEL NAVEGADOR-----
*/
// Inicialitzem variables per guardar més endavant les coordenades de posicionament:
var lat;
var lng;
// I més variables per emmagatzemar resultats de les crides a l'API de previsió del
temps:
var temp = new Array();
var wind = new Array();
var weather = new Array();
var place;

// Establim el posicionament inicial del dispositiu a través de la geolocalització
del navegador:
setPosition();
function setPosition() {
  // Si el navegador permet la geolocalització, disparem el mètode
  getCurrentPosition():
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(success, error, options);
  // Si el navegador no permet la geolocalització, disparem directament error():
  } else {
    error();
  }
};

var options = {

```

```

    EnableHighAccuracy: true, // Habilitem un posicionament acurat.
    Timeout: 3000, // Establim 3 segons com a temps d'espera màxim.
    MaximunAge: 0 // Establim que volem el posicionament actual.
};

// Si tot és correcte i se'ns permet obtenir la geolocalització, es dispara
success():
function success(geolocationPosition) {
    // Obtenim els paràmetres de latitud i longitud de l'objecte geolocationPosition
    (coords.latitude i coords.longitude):
    lat = geolocationPosition.coords.latitude;
    lng = geolocationPosition.coords.longitude;

    initMap(lat, lng);
    updateUrlWeatherCall(); // Actualitzem la URL de la petició a Open Weather Map
    amb les dades de posicionament donades.
};

// Si no es pot obtenir la geolocalització del navegador, s'adverteix a l'usuari i
indiquem un posicionament per defecte (en aquest cas, el de la ciutat de Nova York):
function error(err) {
    lat = 40.7142700;
    lng = -74.0059700;

    initMap(lat, lng);
    updateUrlWeatherCall(); // Actualitzem la URL de la petició a Open Weather Map
    amb les dades de posicionament donades.

    alert("We can't get your position!");
};
/*
-----
*/
/*
CARREGAR MAPA AMB LA GEOLOCALITZACIÓ DELS PUNTS (MARCADORS) SELECCIONATS
*/
function initMap(x, y) {
    let map;
    let location = { lat: x, lng: y };
    let marker;

    if (x !== undefined) { // Esperem a que arribi un valor vàlid abans d'executar la
        funció.
        map = new google.maps.Map(document.getElementById("map"), {
            center: { lat: x, lng: y }, // Centrem el mapa en aquesta posició.
            zoom: 14, // Indiquem el zoom del mapa.
        });
    }
};

```

```

// Afegim el marcador:
addMarker(location, map);

// Cada vegada que fem click amb el ratolí sobre el mapa, afegirem un nou
marcador:
google.maps.event.addListener(map, "click", (event) => {
    marker.setMap(null); // Primer, eliminem el marcador existent.
    addMarker(event.latLng, map);
    // Actualitzem la localització:
    lat = event.latLng.lat();
    lng = event.latLng.lng();
    location.lat = lat;
    location.lng = lng;
    updateUrlWeatherCall();
});

// Definim i situem un marcador:
function addMarker(location, map) {
    marker = new google.maps.Marker({
        position: location,
        map: map
    });
};

}
};
/*
-----
*/
/*
PREVISIÓ DEL TEMPS: OPEN WEATHER API
*/
const forecast_button = document.getElementById("forecast_button");
const apiWeatherKey = "21e97e7de27fce68c74d289d2869e067"; // Clau API necessària per
// executar cada petició.

var url;
// Funció per actualitzar el valor de la variable url:
function updateUrlWeatherCall() {
    url =
`https://api.openweathermap.org/data/2.5/forecast?lat=${lat}&lon=${lng}&cnt=2&appid=${
apiWeatherKey}&units=metric`; // URL per accedir a l'API. Conté els paràmetres de
// geolocalització (lat, lng) i la clau API.
}
// Funció fletxa per executar la petició AJAX i obtenir la informació de l'API
const getWeatherInfo = async () => {
    try {
        const response = await fetch(url, {cache: "no-cache"}); // Ens assegurem que
// no aprofiti res del
// cache.

        // Si la resposta és correcte:

```

```

    if (response.ok) {
        const jsonResponse = await response.json(); // Obtenim la resposta en
                                                    format json.

        // A continuació obtenim la temperatura, el valor principal del paràmetre
        "weather", i el vent.
        // Com es pot comprovar, agafem un valor inicial (valor actual) i un de
        final (valor al cap de 3 hores):
        let initial_temp = jsonResponse.list[0].main.feels_like;
        let final_temp = jsonResponse.list[1].main.feels_like;
        let initial_weather = jsonResponse.list[0].weather[0].main;
        let final_weather = jsonResponse.list[1].weather[0].main;
        let initial_wind = jsonResponse.list[0].wind.gust;
        let final_wind = jsonResponse.list[1].wind.gust;
        let city = jsonResponse.city.name;
        // Insertem les dades dins de variables per poder-les utilitzar
        posteriorment:
        temp.push(initial_temp, final_temp);
        wind.push(initial_wind, final_wind);
        weather.push(initial_weather, final_weather);
        place = city;
    }
    // Si torna un error el mostrem a la consola:
    } catch (error) {
        console.log(error);
    }
};
forecast_button.addEventListener("click", getWeatherInfo);
/*
-----
*/

```

2.4. Arquitectura bàsica del lloc web

Estructura pàgina web

Es tractarà d'una aplicació amb:

- INICI: Aquesta pàgina contindrà la funció principal de la web. Des d'aquí es podrà cercar la previsió del temps de diferents localitzacions i obtenir els resultats.
- SIGN IN: Pàgina per iniciar sessió. En cas de que no s'estigui registrat a Ride Sweetly, t'enviarà a la pàgina REGISTER. Aquesta pàgina es farà amb forma de modal que apareixerà al clicar sobre el botó corresponent.

- SIGN UP: Pàgina per registrar-se a Ride Sweetly.
- PROFILE: Pàgina només disponible si has iniciat sessió. Es mostrarà la informació de l'usuari.
 - o ADD CLOTHES: Funcionalitat dins de PROFILE, que torna a carregar la pàgina cada cop que s'afegeix una nova peça a l'inventari de roba.
- ABOUT US: Pàgina amb informació bàsica sobre la pàgina web i el seu funcionament.

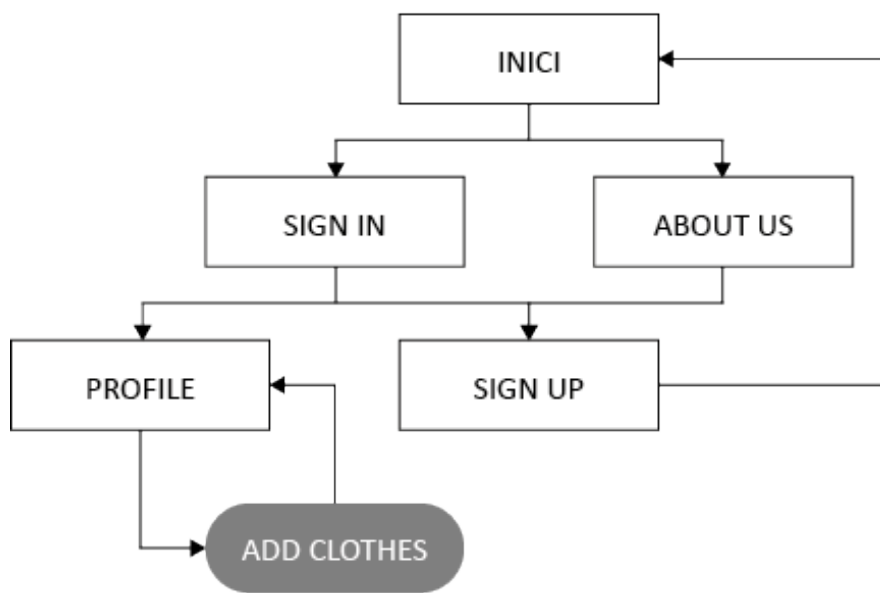
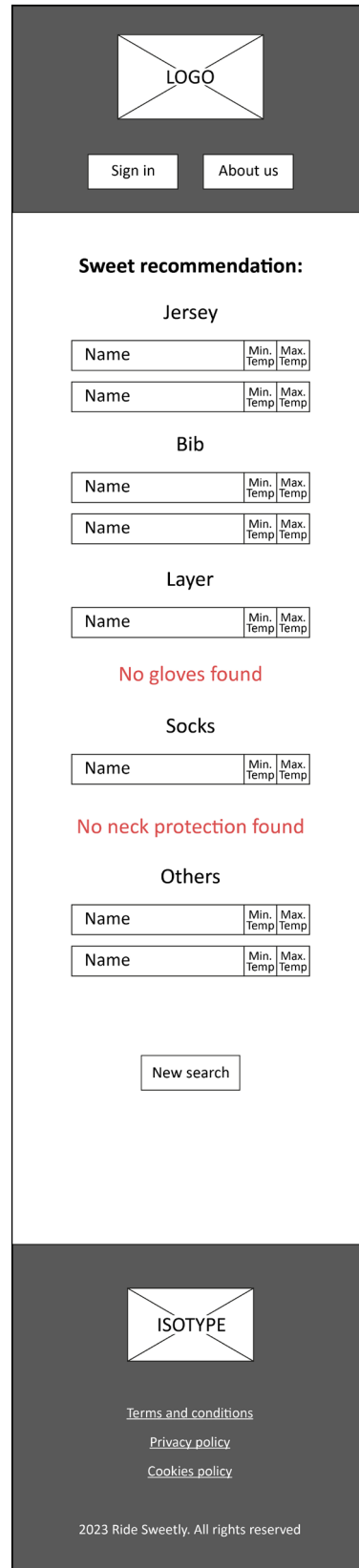
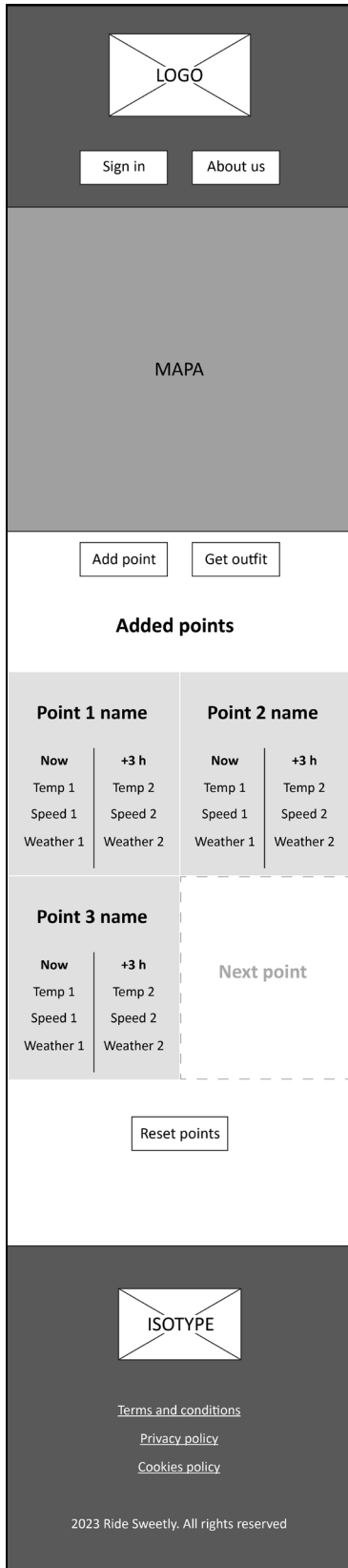


Figura 5. Estructura pàgina web.

Wireframes

Mobile wireframe 390px: INICI (2 wireframes: un per quan s'estan afegint localitzacions i l'altre per quan es mostren els resultats de la cerca).



Mobile wireframe 390px: SIGN IN i SIGN UP

Sign in to Ride Sweetly X

Email

Password

Log in

Don't have an account?
[Create account](#)

LOGO

Sign UP to Ride Sweetly

Name

Last name

Sig up

Cancel

ISOTYPE

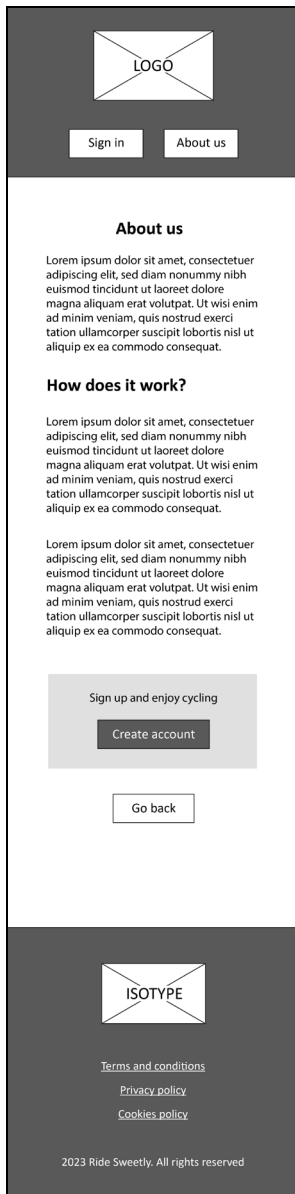
[Terms and conditions](#)

[Privacy policy](#)

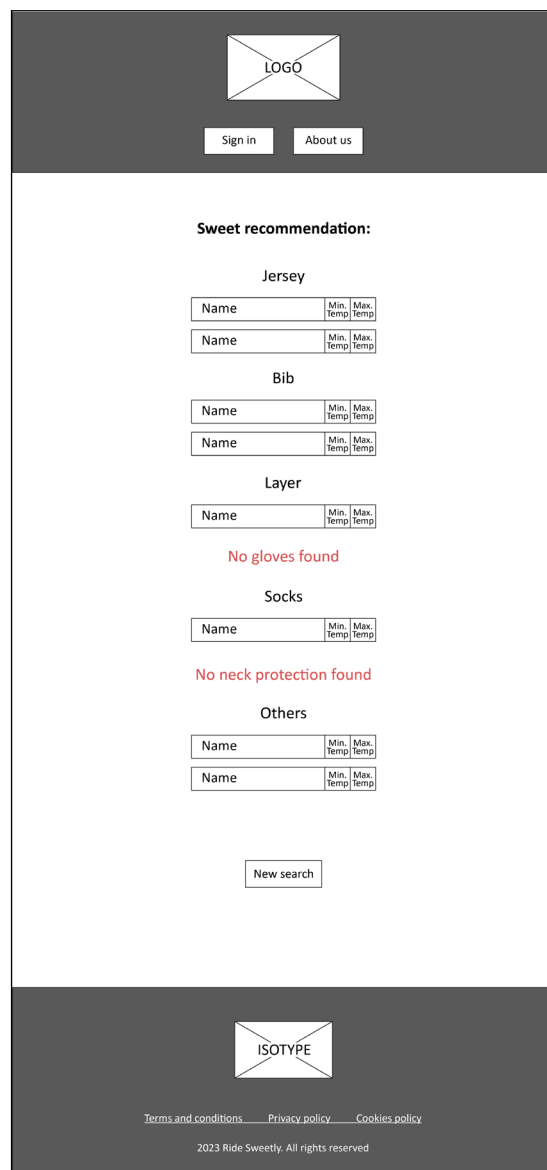
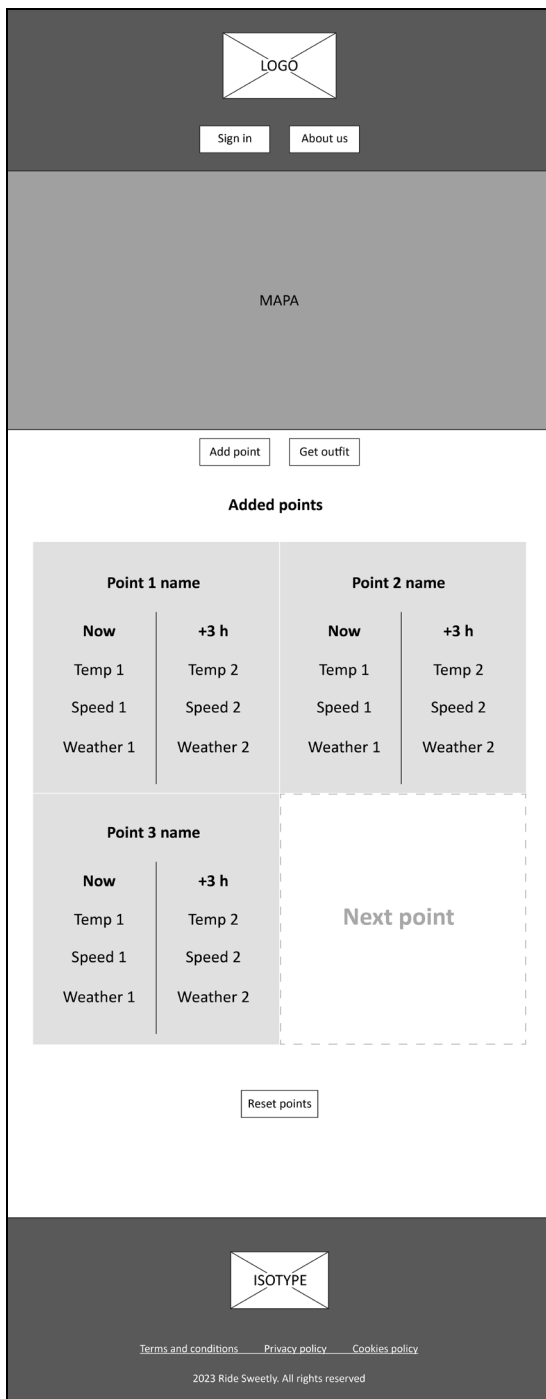
[Cookies policy](#)

2023 Ride Sweetly. All rights reserved

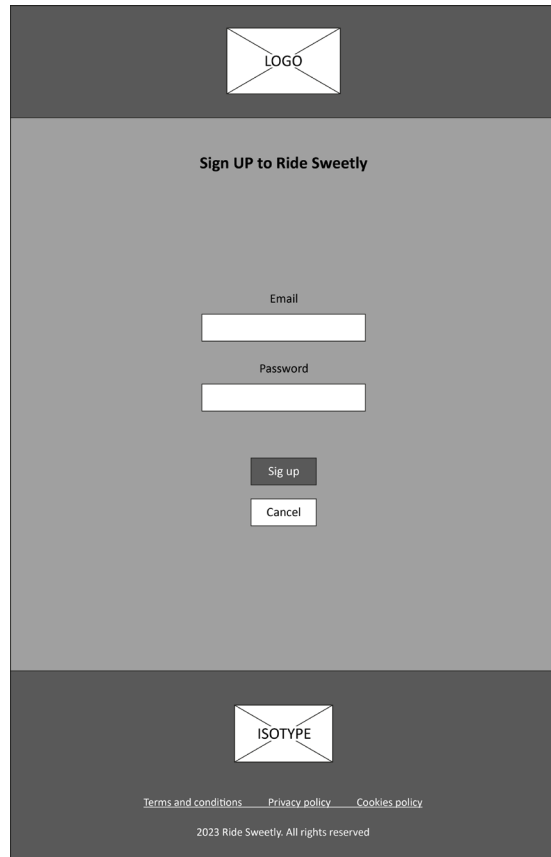
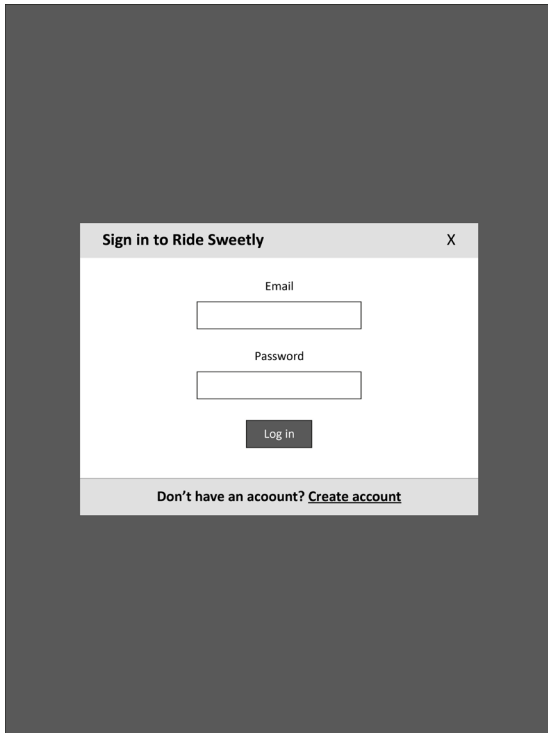
Mobile wireframe 390px: ABOUT US i PROFILE



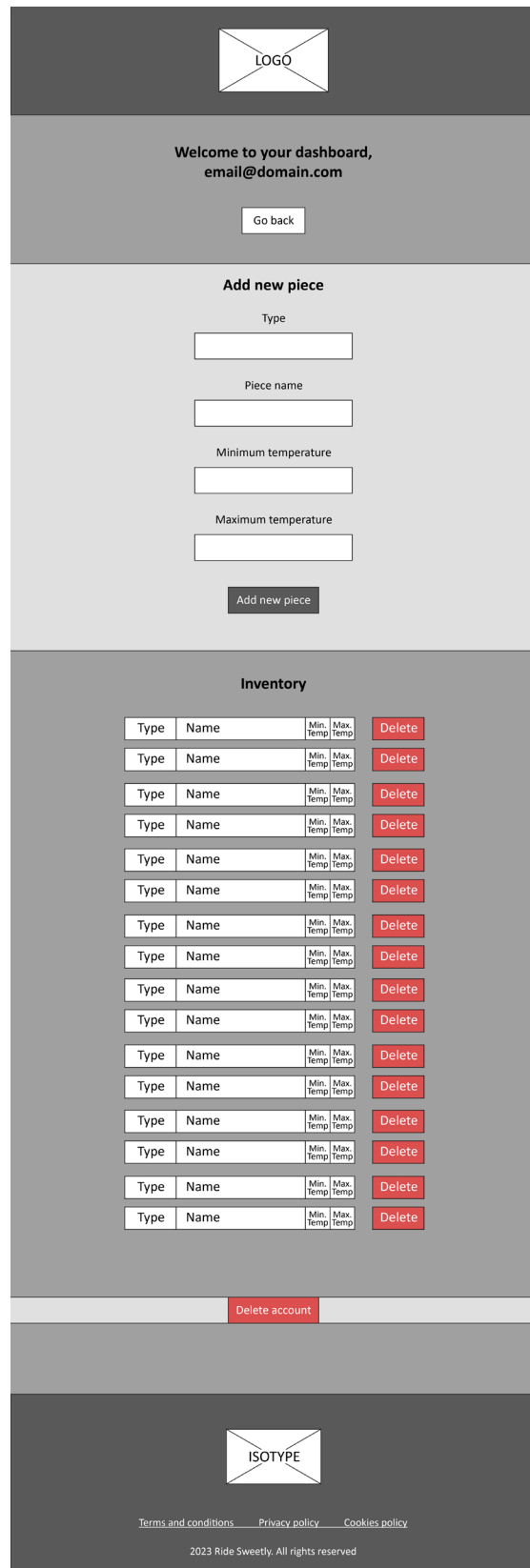
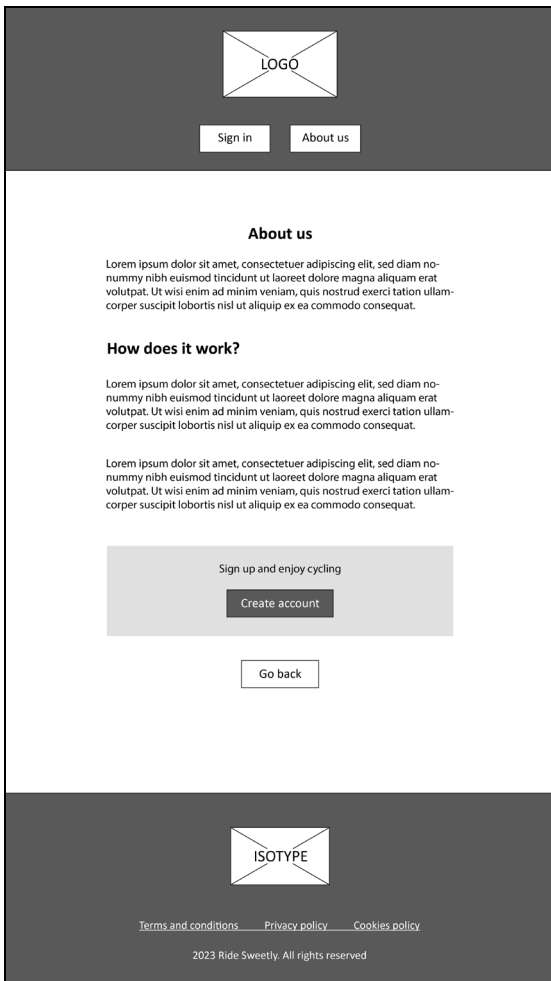
Tablet wireframe 768px: INICI (2 wireframes: un per quan s'estan afegint localitzacions i l'altre per quan es mostren els resultats de la cerca).



Tablet wireframe 768px: SIGN IN i SIGN UP

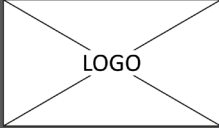


Tablet wireframe 768px: ABOUT US i PROFILE



Desktop wireframe 1024px: INICI





[Sign in](#)

[About us](#)

Sweet recommendation:

Jersey

Name	Min. Temp	Max. Temp
------	-----------	-----------

Name	Min. Temp	Max. Temp
------	-----------	-----------

Bib

Name	Min. Temp	Max. Temp
------	-----------	-----------

Name	Min. Temp	Max. Temp
------	-----------	-----------

Layer

Name	Min. Temp	Max. Temp
------	-----------	-----------

No gloves found

Socks

Name	Min. Temp	Max. Temp
------	-----------	-----------

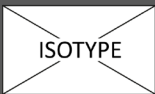
No neck protection found

Others

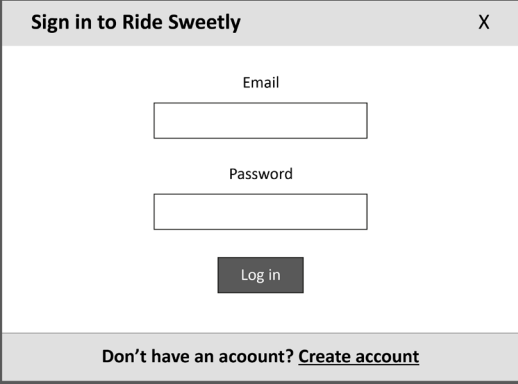
Name	Min. Temp	Max. Temp
------	-----------	-----------

Name	Min. Temp	Max. Temp
------	-----------	-----------

[New Search](#)

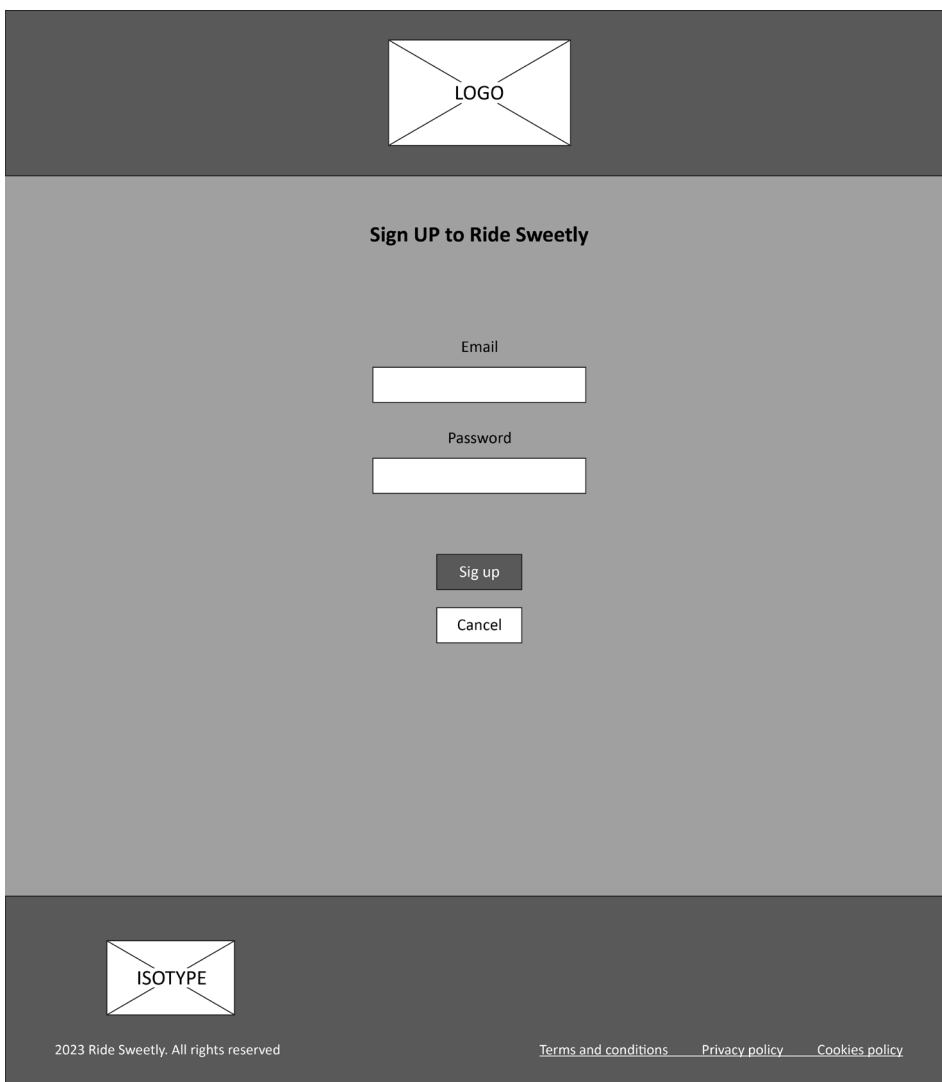


Desktop wireframe 1024px: SIGN IN



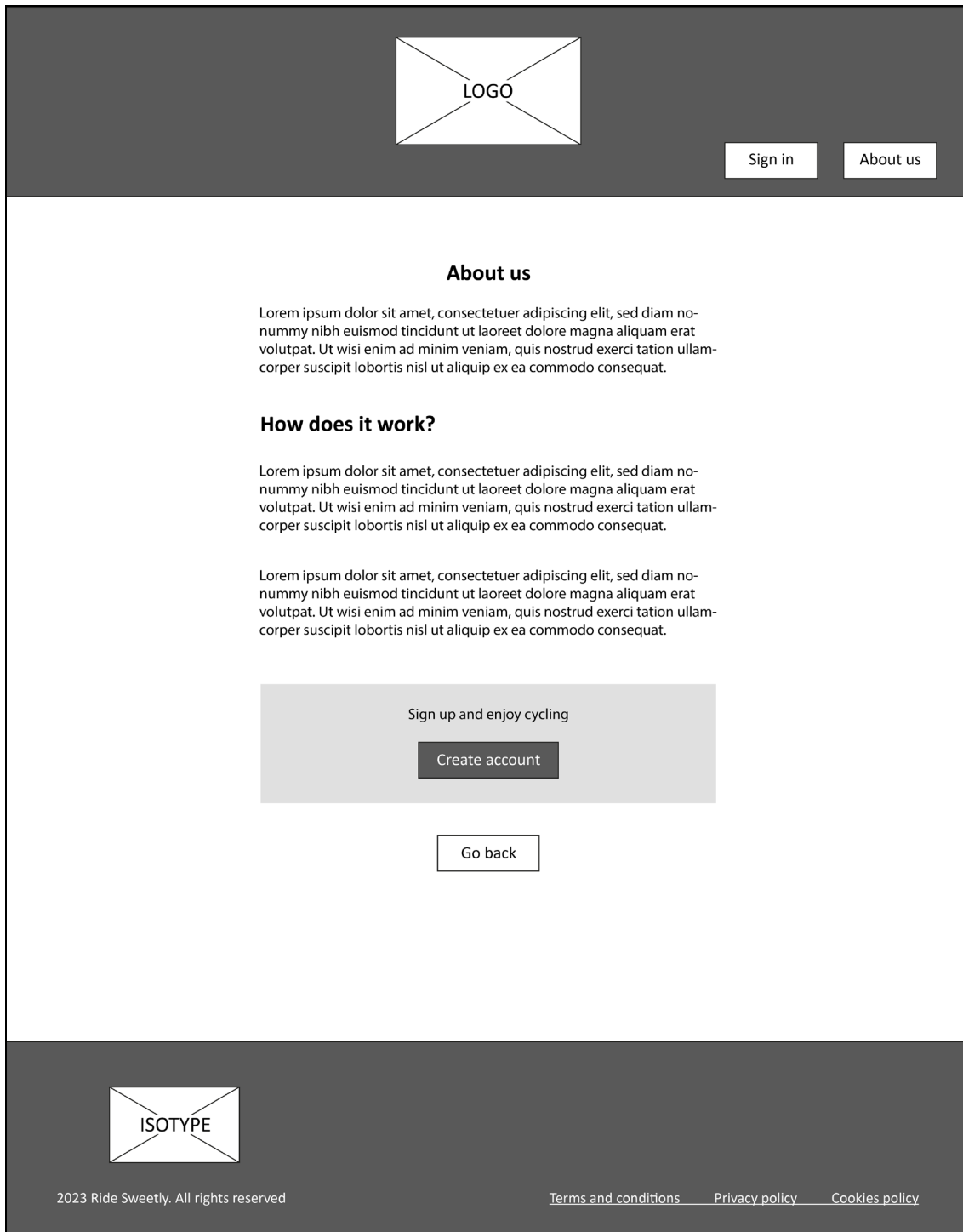
A wireframe of a sign-in dialog box. The dialog has a title bar with the text "Sign in to Ride Sweetly" and a close button "X". Below the title bar, there are two input fields: "Email" and "Password". Below the "Password" field is a "Log in" button. At the bottom of the dialog, there is a link: "Don't have an account? [Create account](#)".

Desktop wireframe 1024px: SIGN UP

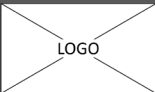


A wireframe of a sign-up page. At the top center is a logo placeholder labeled "LOGO". Below the logo is the heading "Sign UP to Ride Sweetly". Underneath the heading are two input fields: "Email" and "Password". Below the "Password" field are two buttons: "Sig up" and "Cancel". At the bottom left of the page is another logo placeholder labeled "ISOTYPE". At the bottom right, there is a footer with the text "2023 Ride Sweetly. All rights reserved" and three links: "Terms and conditions", "Privacy policy", and "Cookies policy".

Desktop wireframe 1024px: ABOUT US



Desktop wireframe 1024px: PROFILE



Welcome to your dashboard,
email@domain.com

Go back

Add new piece

Type

Piece name

Minimum temperature


Maximum temperature

Add new piece

Inventory

Type	Name	Min. Temp	Max. Temp	Delete
Type	Name	Min. Temp	Max. Temp	Delete
Type	Name	Min. Temp	Max. Temp	Delete
Type	Name	Min. Temp	Max. Temp	Delete
Type	Name	Min. Temp	Max. Temp	Delete
Type	Name	Min. Temp	Max. Temp	Delete
Type	Name	Min. Temp	Max. Temp	Delete
Type	Name	Min. Temp	Max. Temp	Delete
Type	Name	Min. Temp	Max. Temp	Delete
Type	Name	Min. Temp	Max. Temp	Delete
Type	Name	Min. Temp	Max. Temp	Delete
Type	Name	Min. Temp	Max. Temp	Delete
Type	Name	Min. Temp	Max. Temp	Delete
Type	Name	Min. Temp	Max. Temp	Delete
Type	Name	Min. Temp	Max. Temp	Delete
Type	Name	Min. Temp	Max. Temp	Delete

Delete account




2023 Ride Sweetly. All rights reserved [Terms and conditions](#) [Privacy policy](#) [Cookies policy](#)

2.5. Servidor local i base de dades

Per realitzar les proves inicials necessitarem configurar un servidor local i crear una base de dades on puguem introduir les primeres dades per defecte.

Configuració Wampserver

Per al “servidor” **utilitzarem WAMP**, una programari gratuït per a la plataforma [Windows](#) que inclou tot el que necessitem: [Apache](#) com a servidor web i MySQL per a la base de dades, entre d’altres.

Per a descarregar-lo haurem d’anar a la pàgina web oficial www.wampserver.com i seguir les indicacions. Si tot ha anat bé, podrem accedir a ell clicant sobre la icona  que apareix a la barra de tasques.

Durant la configuració haurem establert una carpeta de nom “wamp64” que a la vegada contindrà la carpeta “www”. Dins d’aquesta última haurem de crear el nostre projecte web: ridesweetly. A continuació, accedim al *localhost*, ja sigui introduint la direcció directament al navegador web o mitjançant el menú que ens ofereix Wampserver, i podrem comprovar que ja tenint el nostre projecte creat (al que podrem accedir des de la direcció **localhost/ridesweetly**).

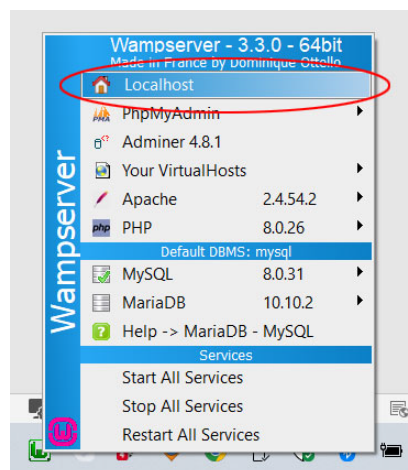


Figura 6. Menú desplegable de Wampserver.

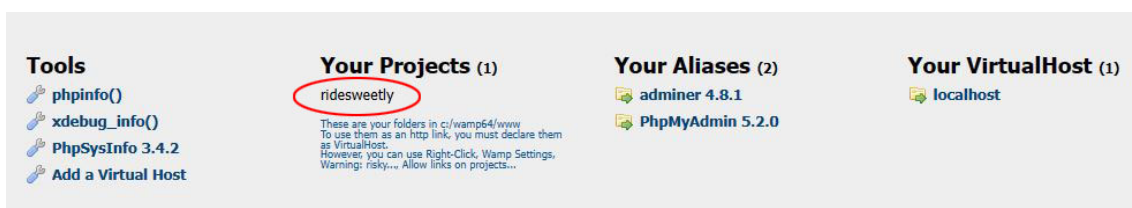


Figura 7. Retall de la pantalla *localhost* de Wampserver.

Configuració base de dades

Per a la base de dades necessitarem accedir al gestor mitjançant l'opció phpMyAdmin que trobem al menú de Wampserver i, un cop dins, anem a la pestanya "Base de datos", li donem un nom i assignem una col·lació, i cliquem a "Crear":



Figura 8. Retall de la pantalla Base de datos de phpMyAdmin.

Ara tocarà crear la taula on anirem guardant les dades. Per això, anirem a la pestanya "Estructura" i crearem una taula de nom *inventory* i 6 columnes, que donarà com a resultat el següent:

	#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/>	1	id	int			No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/>	2	email	varchar(75)	utf8mb4_spanish_ci		No	Ninguna		
<input type="checkbox"/>	3	type	varchar(15)	utf8mb4_spanish_ci		No	Ninguna		
<input type="checkbox"/>	4	name	varchar(30)	utf8mb4_spanish_ci		No	Ninguna		
<input type="checkbox"/>	5	mintemp	int			No	Ninguna		
<input type="checkbox"/>	6	maxtemp	int			No	Ninguna		

Figura 9. Retall de la pantalla Estructura de phpMyAdmin.

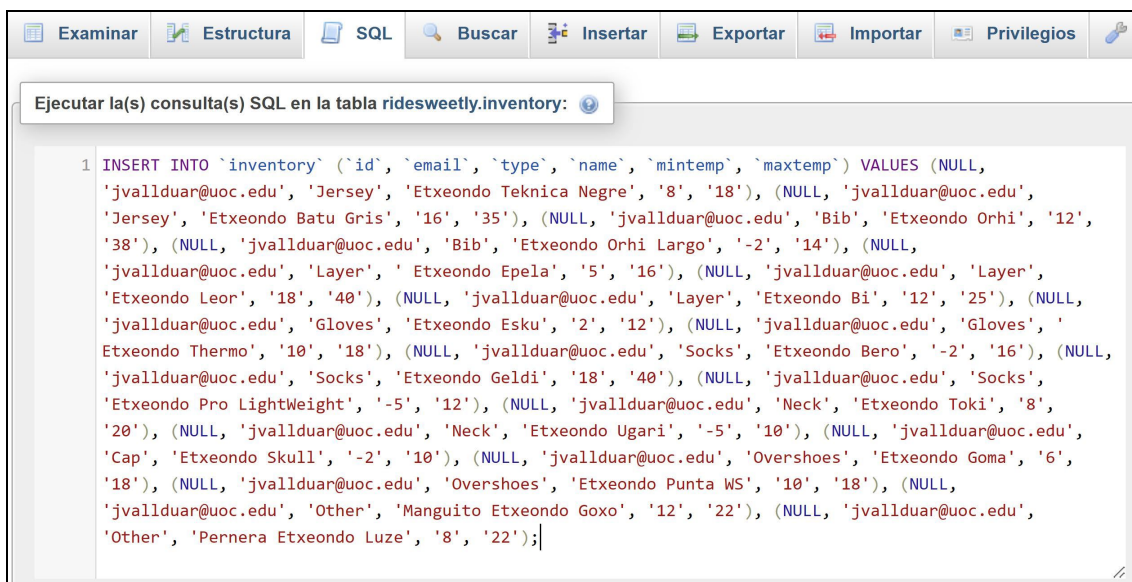
Les columnes establertes per a cada fila d'informació que s'afegirà són:

- **id**: Clau única numèrica que identificarà cada fila. S'incrementarà automàticament per a cada inserció.

- **email:** Camp de text per emmagatzemar el email que utilitzarem per identificar l'usuari.
- **type:** Camp de text per emmagatzemar el tipus de peça que s'està afegint a l'inventari (Jersey, Bib, Gloves...).
- **name:** Camp de text on l'usuari indica un nom per la peça de roba.
- **mintemp:** Camp numèric que indica la temperatura mínima associada a la peça de roba.
- **maxtemp:** Camp numèric que indica la temperatura màxima associada a la peça de roba.

Introducció valors a la base de dades

Per seguir amb el desenvolupament introduïrem unes dades provisionals per defecte. Pe fer-ho, es pot utilitzar la interfície de camps que ens ofereix phpMyAdmin (un cop dins de la taula inventory, clicant a la pestanya "Insertar") o directament des de la pestanya "[SQL](#)"²⁴, escrivint la instrucció corresponent en llenguatge SQL. En el següent exemple, utilitzem aquest últim mètode:



The screenshot shows the phpMyAdmin interface with the 'SQL' tab selected. The title bar reads 'Ejecutar la(s) consulta(s) SQL en la tabla ridesweetly.inventory:'. The main area contains the following SQL statement:

```

1 INSERT INTO `inventory` (`id`, `email`, `type`, `name`, `mintemp`, `maxtemp`) VALUES (NULL,
'jvallduar@uoc.edu', 'Jersey', 'Etxeondo Teknica Negre', '8', '18'), (NULL, 'jvallduar@uoc.edu',
'Jersey', 'Etxeondo Batu Gris', '16', '35'), (NULL, 'jvallduar@uoc.edu', 'Bib', 'Etxeondo Orhi', '12',
'38'), (NULL, 'jvallduar@uoc.edu', 'Bib', 'Etxeondo Orhi Largo', '-2', '14'), (NULL,
'jvallduar@uoc.edu', 'Layer', 'Etxeondo Epela', '5', '16'), (NULL, 'jvallduar@uoc.edu', 'Layer',
'Etxeondo Leor', '18', '40'), (NULL, 'jvallduar@uoc.edu', 'Layer', 'Etxeondo Bi', '12', '25'), (NULL,
'jvallduar@uoc.edu', 'Gloves', 'Etxeondo Esku', '2', '12'), (NULL, 'jvallduar@uoc.edu', 'Gloves',
'Etxeondo Thermo', '10', '18'), (NULL, 'jvallduar@uoc.edu', 'Socks', 'Etxeondo Bero', '-2', '16'), (NULL,
'jvallduar@uoc.edu', 'Socks', 'Etxeondo Geldi', '18', '40'), (NULL, 'jvallduar@uoc.edu', 'Socks',
'Etxeondo Pro LightWeight', '-5', '12'), (NULL, 'jvallduar@uoc.edu', 'Neck', 'Etxeondo Toki', '8',
'20'), (NULL, 'jvallduar@uoc.edu', 'Neck', 'Etxeondo Ugari', '-5', '10'), (NULL, 'jvallduar@uoc.edu',
'Cap', 'Etxeondo Skull', '-2', '10'), (NULL, 'jvallduar@uoc.edu', 'Overshoes', 'Etxeondo Goma', '6',
'18'), (NULL, 'jvallduar@uoc.edu', 'Overshoes', 'Etxeondo Punta WS', '10', '18'), (NULL,
'jvallduar@uoc.edu', 'Other', 'Manguito Etxeondo Goxo', '12', '22'), (NULL, 'jvallduar@uoc.edu',
'Other', 'Pernera Etxeondo Luze', '8', '22');|

```

Figura 10. Instrucció SQL d'introducció de dades per defecte.

2.6. Funcionament web i recollida de dades

Un cop finalitzats els tests amb les APIs de Maps Javascript i Open Weather Map, i la configuració del servidor local juntament amb la base de dades, ens centrarem en gestionar la informació de *forecast* obtinguda per tal de llençar una petició al servidor i, d'aquesta manera, aconseguir identificar els ítems de la base de dades que s'ajusten a les temperatures presentades. A la vegada, anirem explicant els fluxos principals de la pàgina. Això últim ens ajudarà a entendre millor el seu funcionament.

A la pàgina d'inici disposarem del mapa que ens servirà per afegir les localitzacions gràcies al botó "Add point". Un cop seleccionat el punt desitjat al mapa i clicant el botó, s'envien les coordenades de la localització a l'API Open Weather Map per obtenir el pronòstic del temps. En aquest moment, la pàgina afegeix la informació obtinguda per pantalla i seguirà afegint tots els punts que anem indicant, un rere l'altre. Un cop tinguem tots els punts, es pot clicar el botó "Get outfit" per llençar la petició a la base de dades.

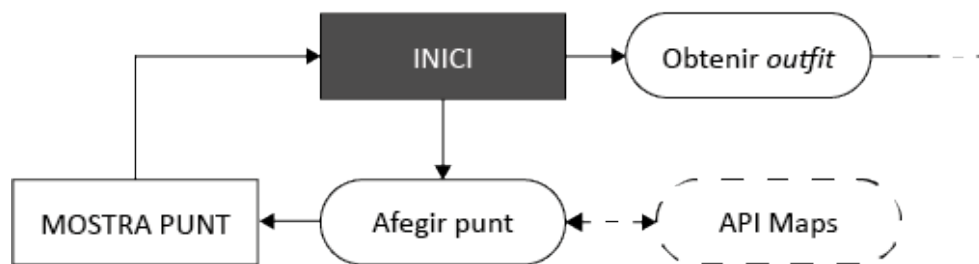


Figura 11. Flux pàgina inicial. Afegint localitzacions.

La figura 11 mostra com, cada vegada que s'afegeix un punt, s'obté el pronòstic de l'API i s'afegeix la informació.

Un cop es vol llençar la petició de consulta, primer es revisen una sèrie de punts, com es mostra a la figura 12:

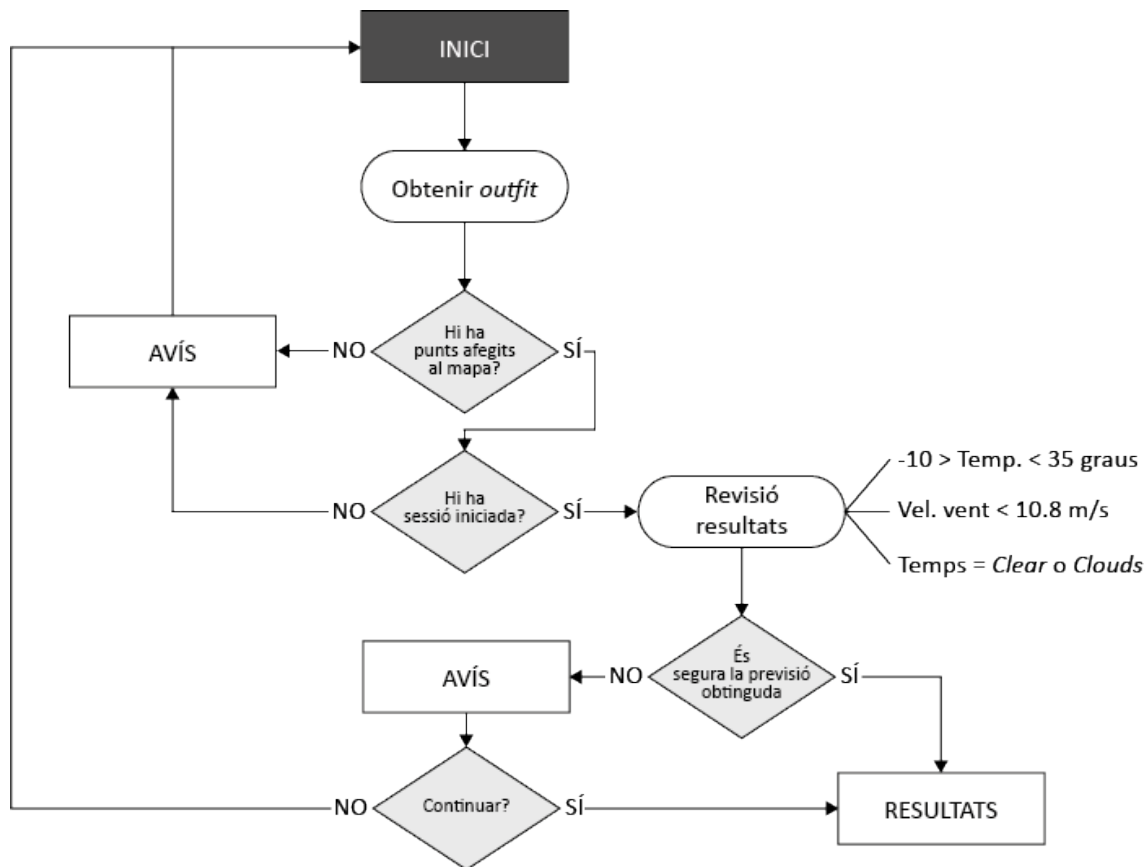


Figura 12. Flux pàgina inicial. Previ a la petició.

Com es mostra a la figura anterior, un cop es clica “Get outfit” (“Obtenir outfit” a la figura), es revisa si l’usuari ha afegit punts amb el mapa. En cas contrari, se li demanarà que, com a mínim, afegeixi una localització per poder cercar el pronòstic d’aquesta. En segon lloc, es comprovarà que l’usuari hagi iniciat sessió. Aquest últim punt encara no s’ha treballat, però serà una necessitat afegir-ho quan implementem l’autenticació d’usuari, ja que haurem de cercar els ítems que té aquest vinculats al seu compte personal.

Si tot l’anterior és correcte, es revisarà si, de totes les previsions obtingudes, s’ha detectat algun perill. Això serà si, en qualsevol de les localitzacions, s’ha detectat:

- La temperatura és inferior a -10 graus o superior a 35 graus.
- Si la velocitat del vent és superior a 10,8 metres/segon.
- Si el paràmetre `weather.main` mostra un dels següents valors: Thunderstorm, Drizzle, Rain, Snow, Mist, Smoke, Haze, Fog, Sand, Dust, Ash, Squall o Tornado (deixant com únics valors segurs: Clear o Clouds).

Realment, aquestes comprovacions les farem a la vegada que anem afegint els punts per pantalla. Així, si afegint el punt ja detectem un o més d'aquests perills, es deixarà marcat en vermell, tot i que l'avís saltarà al clicar "Get outfit".

Com hem vist anteriorment, de les localitzacions marcades guardem la informació inicial i final (al cap de 3 hores) de la temperatura, el temps i el vent, utilitzant les variables tipus [Array](#): temp, weather i wind, respectivament:

```
// Insertem les dades dins de variables per poder-les utilitzar posteriorment:
temp.push(initial_temp, final_temp);
wind.push(initial_wind, final_wind);
weather.push(initial_weather, final_weather);
```

Doncs, abans d'aquest punt, farem les comprovacions:

```
// Funcions per determinar si és segur sortir en bici:
function checkTemp(x) {
    // Si la temperatura excedeix els 35 graus:
    if (x > 35) {
        be_safe = false;
        add_danger = true;
    }
    // Si la temperatura baixa dels -10 graus:
    if (x < -10) {
        be_safe = false;
        add_danger = true;
    }
}
function checkWeather(x) {
    // Tot el que no sigui ennuvolat o clar tindrà risc per l'usuari:
    if (x != "Clouds" && x != "Clear") {
        be_safe = false;
        add_danger = true;
    }
}
function checkWind(x) {
    // Si la velocitat és superior a 10,8 m/s:
    if (x > 10.8) {
        be_safe = false;
        add_danger = true;
    }
}
```

Aquestes funcions es dispararan cada vegada que l'usuari afegixi una localització, on la variable `x` és el valor que s'està passant i les variables `be_safe` i `add_danger` serveixen per:

- `be_safe`: Serà la variable que es revisarà abans de llençar la petició a la base de dades. Si el valor es `false`, s'avisarà a l'usuari de si està segur que vol continuar.

```
let answer = true;
// Si la previsió del temps no és segura, fem una alerta esperant confirmació
o no de l'usuari:
if (!be_safe) {
    answer = confirm("Some of the places you marked have bad weather
    forecasting. We recommend you stay safe.");
}
```

- `add_danger`: Afegim la classe `danger`, en cas de ser `true`, a l'element HTML que contindrà la informació del pronòstic, per tal de ressaltar-lo en vermell:

```
// Si alguna dada no és segura afegim la classe .danger:
if (add_danger) {
    point.className = "danger";
}
```

*La variable `point` conté l'element HTML `article`, on introduïm la informació referent a la previsió del temps d'un punt localitzat.

Finalment, es fa la petició a la base de dades i es mostren els resultats. Aquesta petició es farà utilitzant la informació de la temperatura mitja entre el valor màxim i mínim obtinguts d'entre totes les localitzacions. Es fa així per arribar a un punt de compromís, ja que un ciclista ha de sortir amb les característiques de roba més polivalents possible. Aleshores, de la variable de tipus `Array` `temp`, extraurem les dades corresponents:

```
let max_temp = Math.max(...temp); // Obtenim el valor màxim de l'array temp.
let min_temp = Math.min(...temp); // Obtenim el valor mínim de l'array temp.
let avg_temp = (max_temp + min_temp)/2; // Obtenim el valor mig.
```

A la figura 13, podem veure el flux de la petició i la gestió de la informació:

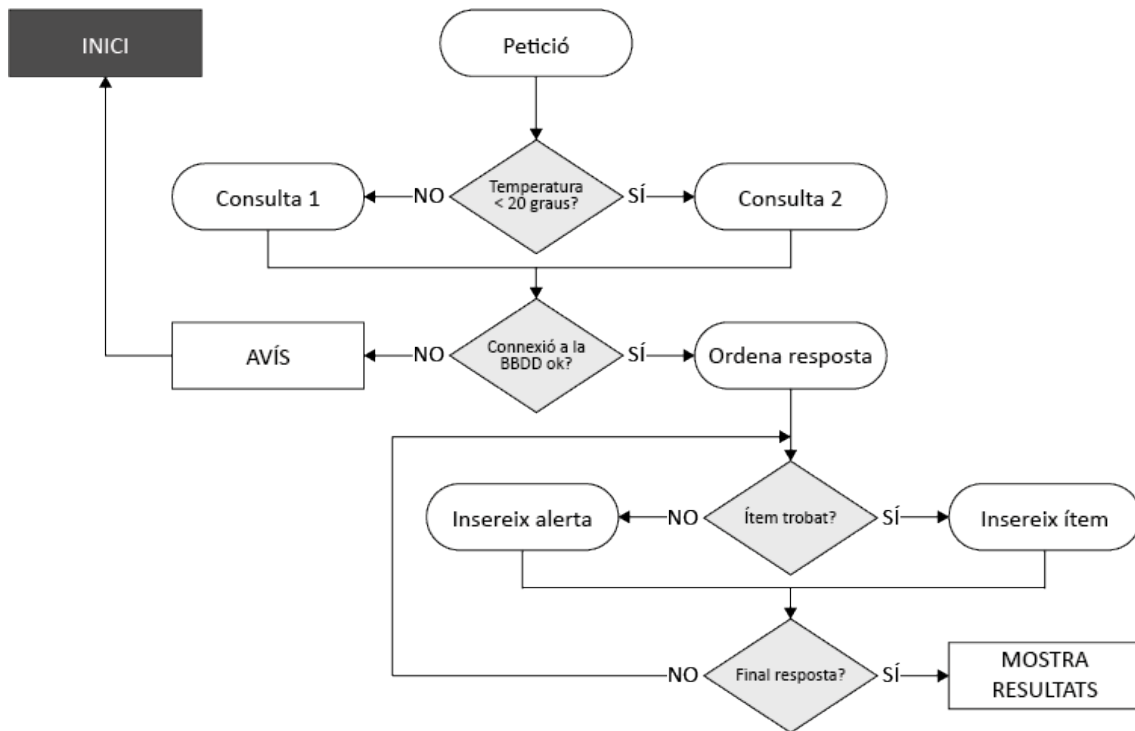


Figura 13. Flux de la petició i gestió de la informació obtinguda.

Com podem observar, abans de llençar la petició es comprova si la temperatura mitja que s'utilitzarà per llençar la consulta a la BBDD, és inferior a 20 graus. Això és així ja que utilitzarem aquesta temperatura com a llindar per triar si utilitzem la Consulta 1 o la Consulta 2. En la primera consulta es cercaran tots els tipus d'ítems existents a la base de dades de l'usuari. En canvi, a la Consulta 2, es descarten els ítems de tipus *Neck*, *Overshoes* i *Cap*, ja que es tracten d'accessoris necessaris quan fa més fred.

La Consulta 1 seria:

```

(SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 -
${avg_temp}) dif FROM inventory WHERE type = "Jersey" LIMIT 2)
UNION
(SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 -
${avg_temp}) dif FROM inventory WHERE type = "Bib" LIMIT 2)
UNION
(SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 -
${avg_temp}) dif FROM inventory WHERE type = "Layer" LIMIT 2)
UNION
(SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 -
${avg_temp}) dif FROM inventory WHERE type = "Gloves" LIMIT 2)
UNION
(SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 -
${avg_temp}) dif FROM inventory WHERE type = "Socks" LIMIT 2)
UNION
(SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 -
${avg_temp}) dif FROM inventory WHERE type = "Neck" LIMIT 2)
UNION

```



```
(SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 -
${avg_temp}) dif FROM inventory WHERE type = "Overshoes" LIMIT 2)
UNION
(SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 -
${avg_temp}) dif FROM inventory WHERE type = "Cap" LIMIT 2)
UNION
(SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 -
${avg_temp}) dif FROM inventory WHERE type = "Other")
ORDER BY dif;
```

Bàsicament, estem unint 9 consultes (una per cada tipus de peça de roba) en una de sola gràcies a la clàusula UNION i, finalment, les ordenem (ORDER BY) utilitzant el valor de la columna dif. A cada consulta, amb SELECT, obtenim els valors de les columnes type, name, mintemp i maxtemp d'un tipus específic de roba utilitzant la clàusula WHERE i, a continuació, limitem el nombre de resultats a 2 utilitzant LIMIT 2. Per obtenir els millors resultats possibles, utilitzem ABS((mintemp + maxtemp) / 2 - \${avg_temp}) dif. Amb això, obtenim el valor de la columna dif per a cada peça, que no deixa de ser la diferència entre la mitja de la temperatura mínima i màxima associada a la peça de roba ((mintemp + maxtemp) / 2) i la temperatura passada amb la consulta (avg_temp). Amb ORDER BY és quan obtenim els valors amb menys diferència de cada tipus.

La Consulta 2 és igual a la Consulta 1, però eliminant les consultes que fan referència als tipus *Neck*, *Overshoes* i *Cap*. També cal destacar que, per a la categoria de tipus *Other*, no es limita el nombre de resultats possibles a 2, ja que aquesta categoria la deixarem perquè l'usuari pugui registrar tots els accessoris que no encaixin a les categories anteriors.

Un cop realitzada la petició i obtinguda una resposta, ordenem els resultats de manera que cada tipus de roba quedi emmagatzemada en una variable diferent. Per a fer-ho, utilitzarem la iteració [for](#) per recórrer la resposta i mitjançant un condicional [switch](#) anirem identificant cada tipus de roba i inserint-la a la variable corresponent. També, cal mencionar que abans del condicional switch, afegirem un condicional [if](#) per descartar els ítems obtinguts amb un valor superior a 15 a la columna dif. Això serà així per evitar que l'aplicació recomani ítems amb una temperatura associada molt dispar a l'obtinguda del *forecast* del temps.

```
// Recorrem la resposta de la consulta i anem guardant la informació de cada ítem a
l'Array corresponent:
for (let item of jsonRes) {
  if (item.dif < 15) {
    switch (item.type) {
      case "Jersey":
        jerseyRes.push(`<ul class="search-results"><li
```

```

        class="piece-name">${item.name}</li><li class="piece-min-
temp">${item.mintemp}&deg</li><li class="piece-max-
temp">${item.maxtemp}&deg</li></ul>`;
        break;
    case "Bib":
        bibRes.push(`<ul class="search-results"><li class="piece-
name">${item.name}</li><li class="piece-min-
temp">${item.mintemp}&deg</li><li class="piece-max-
temp">${item.maxtemp}&deg</li></ul>`);
        break;
    case "Layer":
        layerRes.push(`<ul class="search-results"><li
class="piece-name">${item.name}</li><li class="piece-min-
temp">${item.mintemp}&deg</li><li class="piece-max-
temp">${item.maxtemp}&deg</li></ul>`);
        break;
    case "Gloves":
        glovesRes.push(`<ul class="search-results"><li
class="piece-name">${item.name}</li><li class="piece-min-
temp">${item.mintemp}&deg</li><li class="piece-max-
temp">${item.maxtemp}&deg</li></ul>`);
        break;
    case "Socks":
        socksRes.push(`<ul class="search-results"><li
class="piece-name">${item.name}</li><li class="piece-min-
temp">${item.mintemp}&deg</li><li class="piece-max-
temp">${item.maxtemp}&deg</li></ul>`);
        break;
    case "Neck":
        if (avg_temp < 20) {
            neckRes.push(`<ul class="search-results"><li
class="piece-name">${item.name}</li><li
class="piece-min-temp">${item.mintemp}&deg</li><li
class="piece-max-
temp">${item.maxtemp}&deg</li></ul>`);
        }
        break;
    case "Overshoes":
        if (avg_temp < 20) {
            shoesRes.push(`<ul class="search-results"><li
class="piece-name">${item.name}</li><li
class="piece-min-temp">${item.mintemp}&deg</li><li
class="piece-max-
temp">${item.maxtemp}&deg</li></ul>`);
        }
        break;
    case "Cap":
        if (avg_temp < 20) {
            capRes.push(`<ul class="search-results"><li

```

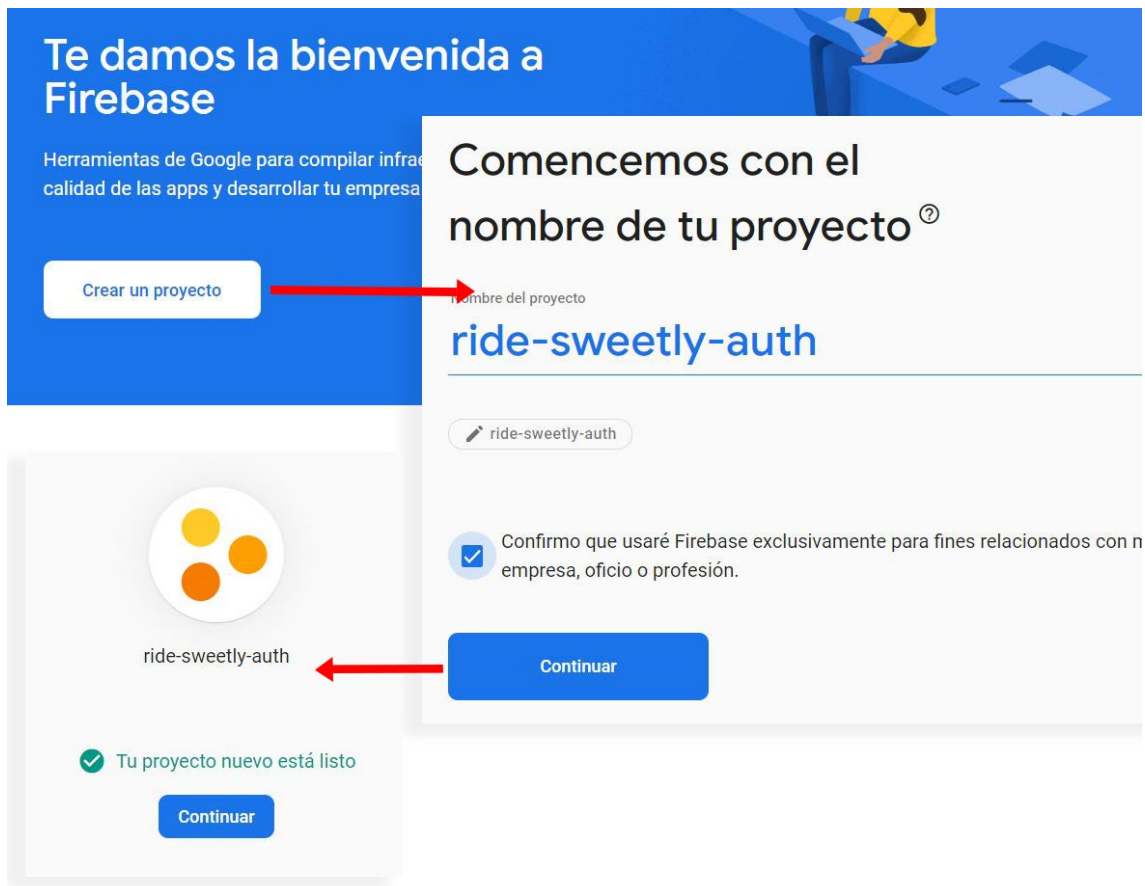



Figura 14. Creació de projecte a Firebase.

A continuació, per poder enllaçar i utilitzar Firebase a la nostra pàgina web, necessitarem “instal·lar-lo” dins d’aquesta. Com ja hem comentat en capítols anteriors, nosaltres estem construint l’aplicació a base de JavaScript, CSS i HTML i, per tant, ahora d’instal·lar Firebase ho farem copiant els *scripts* corresponents als nostres arxius.

Des de la configuració general del projecte, accedirem a les *apps* vinculades al nostre projecte (haurien de ser zero):

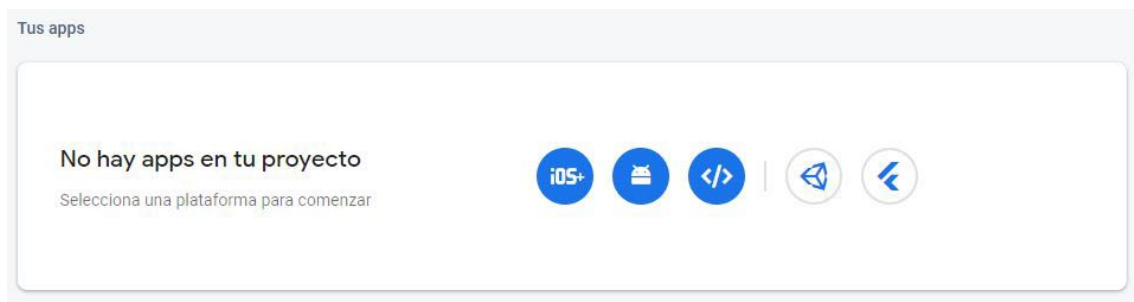



Figura 15. Vinculant *app* a Firebase.

Com que ens basarem en la utilització de JavaScript, clicarem sobre la icona  per accedir a les pantalles que ens proporcionaran el codi.

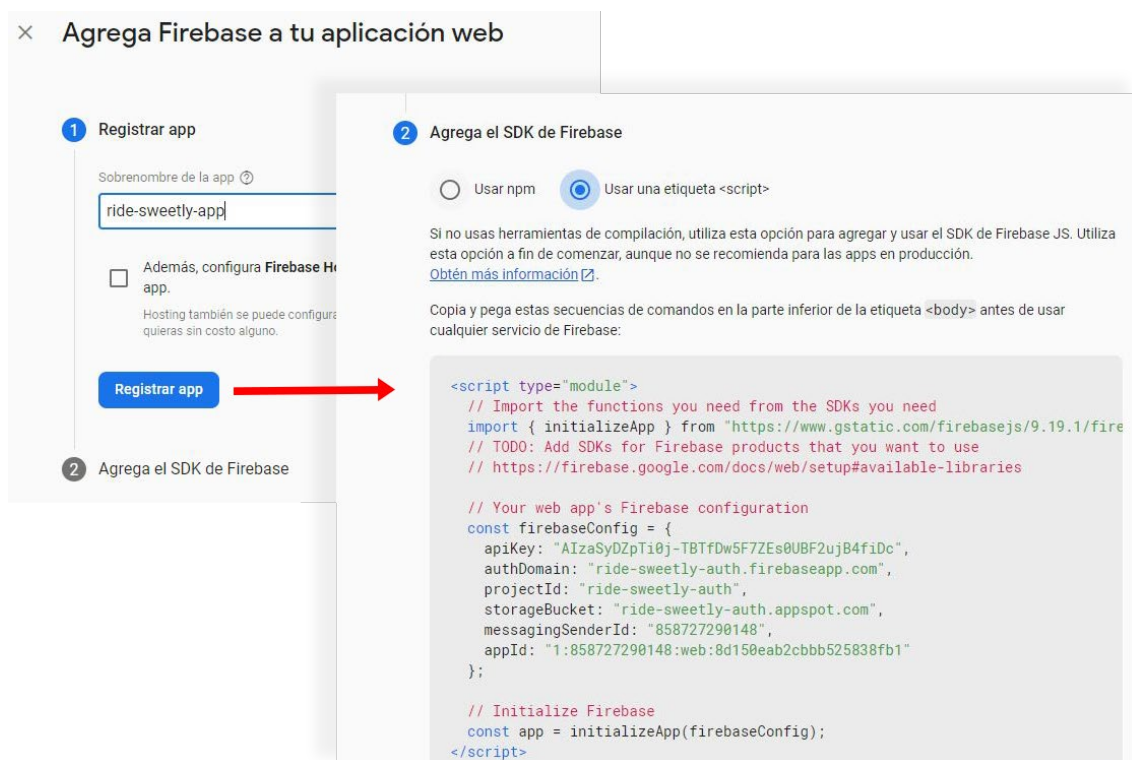


Figura 16. Obtenint *script* de Firebase.

Copiarem el codi de la Figura 16 dins d'un arxiu JavaScript. L'única variació que farem serà que exportarem la constant `app` que inicialitza Firebase, de manera que puguem importar-lo després d'una forma més ordenada a altres arxius JavaScript, que utilitzarem per treballar.

Per tant, creem un arxiu `firebase.js` que conté el següent codi:

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "https://www.gstatic.com/firebasejs/9.18.0/firebase-app.js";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyDZpTi0j-TBTfDw5F7ZE50UBF2ujB4fiDc",
  authDomain: "ride-sweetly-auth.firebaseio.com",
  projectId: "ride-sweetly-auth",
  storageBucket: "ride-sweetly-auth.appspot.com",
  messagingSenderId: "858727290148",
  appId: "1:858727290148:web:8d150eab2cbbb525838fb1"
```

```
};

// Initialize Firebase
export const app = initializeApp(firebaseConfig);
```

Finalment, dins la consola de Firebase, anirem a “Compilación -> Authentication”, i habilitarem com a proveïdor l’opció “Correo electrónico/contraseña”.

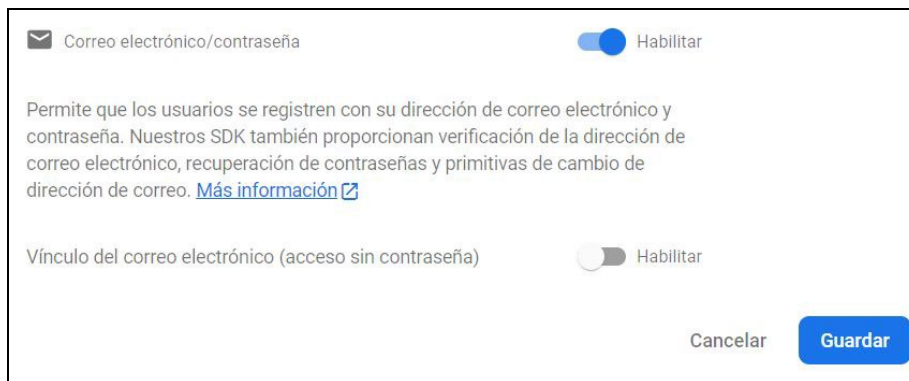


Figura 17. Habilitar proveedor a Firebase Authentication.

Configuració de creació d’usuaris

Dins la documentació de Firebase Authentication per Web podem accedir a totes les biblioteques disponibles: <https://firebase.google.com/docs/web/learn-more?hl=es-419#available-libraries>. Aquí, necessitem copiar la referència a la biblioteca Authentication, dins del desplegable “SDK de Firebase JS disponibles (de la CDN)”, al nostre arxiu firebase.js.

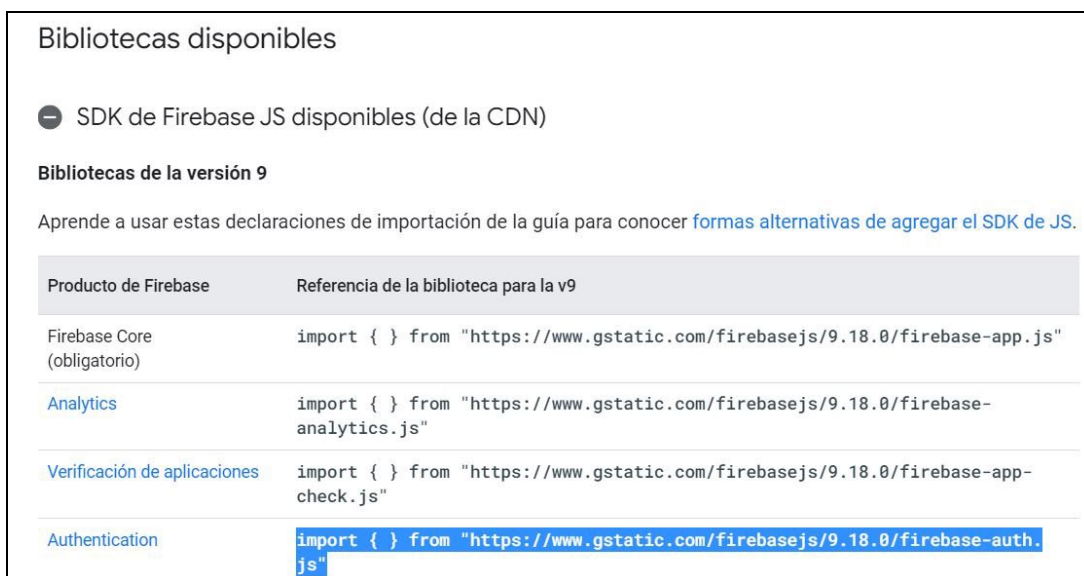


Figura 18. Referència CDN de Firebase Authentication.

Seguint la documentació de la pàgina

<https://firebase.google.com/docs/auth/web/start?hl=es-419>, haurem de configurar `getAuth` (mètode per poder autenticar amb Firebase) i `createUserWithEmailAndPassword` (mètode per crear usuaris a Firebase).

Crearem un nou arxiu (en el nostre cas, `auth-signup.js`), on anirem afegint el codi:

```
// Importem la funció necessària per crear i autenticar usuaris:
import { createUserWithEmailAndPassword } from "https://www.gstatic.com/firebasejs/9.18.0/firebase-auth.js";
import { auth } from "./firebase.js"; // Importem l'objecte auth de firebase.js

const signupForm = document.getElementById("signup-form"); // Referenciem el formulari de registre d'usuari

signupForm.addEventListener("submit", async (e) => {
  e.preventDefault(); // Evitem el funcionament per defecte de l'esdeveniment
  // Obtenim els valors dels camps del formulari:
  const email = signupForm["email"].value;
  const pass = signupForm["pass"].value;

  try {
    await createUserWithEmailAndPassword(auth, email, pass);
    alert("New user added successfully");
  } catch (error) {
    console.log(error);
  }
});
```

Bàsicament, amb la constant `signupForm` estem referenciant al formulari de registre. Amb l'[addEventListener](#) agafem la informació dels camps del formulari i executem la funció `createUserWithEmailAndPassword` de Firebase, per crear el nou usuari. Tot això es fa de forma asíncrona, i necessitem utilitzar [async/await](#) i [try/catch](#) per gestionar la connexió i els resultats d'aquesta. Finalment, per obtenir l'objecte `auth` de `firebase.js`, hem hagut d'afegir el següent codi a aquest últim arxiu:

```
import { getAuth } from "https://www.gstatic.com/firebasejs/9.18.0/firebase-auth.js";
export const auth = getAuth(app);
```

El que s'aconsegueix amb això és habilitar l'autenticació (`getAuth()`) a la nostra aplicació (`app`), que després exportem com la constant `auth` perquè pugui ser utilitzada per altres arxius.

A partir d'aquí ja podem afegir una petita gestió dels errors i afegir la desconnexió de la sessió de l'usuari creat, ja que un cop s'executa `createUserWithEmailAndPassword` es queda la sessió d'aquest iniciada, i el que farem serà redireccionar-lo a l'inici i que des d'allà, si és el cas, iniciï sessió de nou.

Per afegir la desconnexió, haurem d'importar la funció `signOut` des de Firebase amb:

```
import { signOut } from "https://www.gstatic.com/firebasejs/9.18.0/firebase-auth.js";  
// Importem la funció per tamcar sessió
```

Després, l'executarem passant l'objecte d'autenticació `auth` creat anteriorment, que conté tota la informació de la sessió.

Configuració de l'inici de sessió

Per poder iniciar sessió i tenir informació sobre aquesta necessitarem importar dues funcions de Firebase:

- `onAuthStateChanged`: Amb aquesta funció sabrem si hi ha una sessió iniciada o no. Això ens serà d'especial utilitat per poder canviar i personalitzar parts de la pàgina web (si hi ha sessió iniciada, no mostrarem el botó "Sign in" del menú, sinó que ho canviarem per la direcció de correu electrònic de l'usuari connectat, per exemple).

```
// Si tenim una sessió iniciada, mostrarem el botó de Log out i el email de la  
sessió:  
onAuthStateChanged(auth, async (user) => {  
  let session_name = document.getElementById("session-owner");  
  let show = document.getElementsByClassName("login-display");  
  let hide = document.getElementsByClassName("logout-display");  
  
  if (user) {  
    for (let item of show) {  
      item.style.display = "block";  
    }  
    for (let item of hide) {  
      item.style.display = "none";  
    }  
    session_name.innerText = user.email;  
  } else {  
    for (let item of show) {  
      item.style.display = "none";  
    }  
  }  
}
```



```
    for (let item of hide) {  
        item.style.display = "block";  
    }  
}  
});
```

Al codi anterior, senzillament referenciem al que volem mostrar/ocultar de la nostra pàgina segons si existeix o no un usuari amb sessió iniciada. A partir d'aquí fem les iteracions necessàries per canviar la visibilitat dels elements i inserir, si és el cas, la direcció de correu electrònic al botó corresponent.

- `signInWithEmailAndPassword`: Aquesta funció ens permet iniciar la sessió d'un usuari que s'hagi registrat prèviament utilitzant un correu electrònic i contrasenya. El codi de configuració és molt semblant al de registre d'usuaris amb `createUserWithEmailAndPassword`, però canviant la funció que s'executa.

Adaptacions a les consultes SQL

Ara que ja podem crear usuaris i distingir si tenim una sessió iniciada o no, és necessari adaptar les consultes SQL prèviament fetes, ja que necessitarem obtenir els productes d'un usuari específic i aquest només haurà de poder afegir/eliminar productes del seu inventari. Per tant, i amb l'objectiu d'identificar cada usuari, a les consultes afegirem una clàusula condicional extra `WHERE email=user.email, on user.email` fa referència a la direcció de correu electrònic d'aquest.

Un problema que tenim degut a com hem anat configurant la pàgina web és que, segons la pàgina on ens trobem d'aquesta, no podem accedir directament la informació de la sessió que tenim iniciada i, en conseqüència, no tenim la capacitat d'obtenir la direcció de correu electrònic de l'usuari. Per solucionar-ho, un cop iniciem sessió, guardarem automàticament la informació del correu electrònic de l'usuari en una *cookie*:

```
document.cookie = "user= " + user.email;
```

Amb el codi anterior, creem la *cookie* "user" i li afegim el valor corresponent. Quan tanquem sessió, eliminarem la *cookie*. D'aquesta manera sempre tindrem accessible, des de la *cookie* "user", la direcció de correu.

Configuració de l'eliminació d'usuari

Des de la pàgina de perfil de cada usuari, aquest podrà eliminar el seu compte permanentment sempre que així ho vulgui.

Per dur a terme l'anterior, utilitzarem dues funcions que tornarem a importar de Firebase:

- `reauthenticateWithCredential`: Aquesta funció la utilitzarem per tornar a validar l'usuari, ja que l'eliminació del compte es tracta d'una acció que requereix un pas previ de seguretat.
- `deleteUser`: Aquesta funció eliminarà el compte de l'usuari a Firebase. També executarem una petició SQL per eliminar tots els productes de l'inventari associats a la direcció de correu electrònic de l'usuari.

Per poder executar l'acció d'eliminació d'usuari, ho farem des de la pàgina d'inici, ja que allà disposarem de la configuració de Firebase (a la pàgina de perfil, per motius de com hem anat configurant la pàgina, no disposem d'aquesta configuració).

Crearem una *cookie* de nom "deleteAccount" i li donarem un valor de "true". Des de la pàgina d'inici detectarem aquesta *cookie* i procedirem a executar les funcions comentades d'eliminació d'usuari (sempre que el seu valor de la *cookie* sigui "true").

Finalment, durant l'execució de l'eliminació d'usuari, també hauré d'executar una petició SQL a la base de dades per eliminar tots els ítems d'aquest a la base de dades.

2.8. Branding

Tot i que no es pretén desenvolupar un *branding*²⁵ com a tal en aquest treball (això podria donar per un TFG a part), sí que és important marcar unes línies bàsiques que dotin al producte final de personalitat i consistència, a la vegada que ens ofereix un punt d'inici per desenvolupar la marca més endavant.

El propòsit de Ride Sweetly és **fomentar, des d'un punt de vista "friendly"²⁶, l'ús de la bicicleta d'una forma segura, responsable, i sostenible**, sense afavorir el consumisme deliberat. Tot i que en aquest punt embrionari del projecte l'aplicació es centra en la seva funcionalitat, més endavant s'haurien de crear estratègies d'imatge de marca, ja sigui a partir d'accions específiques o bé establint sinergies amb empreses afins als

valors principals de Ride Sweetly. A partir d'aquest propòsit basarem les nostres decisions.

Colors

Les aplicacions principals del sector utilitzen un gran ventall de colors, des del vermell fins al blau, passant pels verdosos elèctrics i més. En el nostre cas, ens centrem en un verd poc saturat agradable a la vista, que ens permet diferenciar-nos de la competència i, a la vegada, transmet tranquil·litat i natura.



Figura 19. Logotips del sector. D'esquerra a dreta: Garmin Connect, Komoot, Ride Sweetly, Wikiloc, Trailforks i Strava.

MAIN COLOR

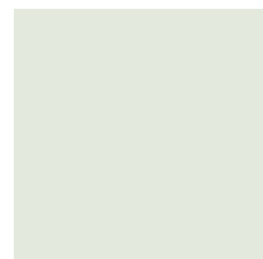


#B9C7A9

SECONDARY COLORS



#42473D



#E4E9DE

FUNCTIONAL COLORS



#DA3434



#D1A147

Figura 20. Paleta de colors de Ride Sweetly, donats en valor hexadecimal.

- Color principal #B9C7A9: Aquest color ens permetrà crear la base de la marca i serà el més **representatiu**. A la nostra pàgina tindrà presència a la capçalera i altres elements com botons i algun fons.
- Color secundari #42473D: Aquest color, una variant fosca del principal, ens permetrà **ressaltar** elements com títols, botons importants de la pàgina web o el *footer* d'aquesta.
- Color secundari #E4E9DE: Aquesta variant més clara del color principal només s'utilitzarà com a **fons d'altres elements** de la pàgina web, ja sigui per descarregar-la del color principal o perquè ens ajuda a ordenar la informació mostrada.
- Colors funcionals #DA3434 i #D1A147: Aquests colors, que es troben en un tercer nivell d'importància dins la marca, només tenen la funció de potenciar el missatge d'alerta/atenció d'alguns elements que doten d'una funcionalitat específica a la pàgina web, com **botons d'eliminació o cancel·lació**.

Logotip

El logotip de Ride Sweetly està format per text i, per una altra part, per una imatge que simula una bicicleta que funciona a la vegada de isotip (quan la separem del logotip) i de doble "e", acabant de completar la paraula *sweetly*:



Figura 21. Logotip de Ride Sweetly.

La intenció és crear un **logotip simpàtic i proper, lluny de pretensions** innecessàries. Respecte al color, depenent de si s'utilitza sobre fons blanc o de color (veure figura 21), són el verd principal #B9C7A9 juntament amb el verd fosc secundari #42473D, que utilitzarem per ressaltar la bicicleta (o doble "e").

És important remarcar l'element que forma la bicicleta dins del logotip, ja que serà un recurs a utilitzar fora d'aquest i molt representatiu de la marca. Exemples d'aplicació serien a la icona d'una futura aplicació (veure l'anterior figura 19) o, fent girar les

“rodes” a mode de GIF ²⁷, com icona que es mostra mentre es carrega contingut a l'app o pàgina web:



Figura 22. Exemple d'aplicació de l'isotip: Icona de càrrega.

Tipografia

Un altre aspecte molt important és la tipografia triada per al logotip. Aquesta, ens permetrà seguir definint la personalitat de la marca i transmetre una sèrie de sensacions específiques.

La tipografia escollida ha estat [Quicksand](#). Es tracta d'una tipografia dissenyada per Andrew Paglinawan amb llicència *open source* ²⁸ ([Open Font License](#)), **arrodonida**, i **sense rematar** ni amb gràcies decoratives (*sans-serif*). Aquestes últimes característiques donen com a resultat una tipografia agradable a la vista i gens agressiva, ideal per al text del logotip de Ride Sweetly i complementar-la amb un isotip arrodonit, com la bicicleta característica de la nostra marca:

QUICKSAND

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

1234567890

Styles:

Quicksand Light 300

Quicksand Regular 400

Quicksand Medium 500

Quicksand SemiBold 600

Quicksand Bold 700

Figura 23. Tipografia Quicksand.

A part, **per als textos de la pàgina web s'ha utilitzat [Roboto](#)**, dissenyada per Christian Robertson (Google). Igual que Quicksand, es tracta d'una tipografia *sans-serif*, però en

aquest cas de línies més rectes i grotesques per facilitar la llegibilitat. La seva llicència també és *open source*: [Apache License \(Versió 2.0.\)](#).

2.9. Mockups i prototip

A continuació, s'ha treballat un disseny que, a partir de l'arquitectura creada anteriorment per a la pàgina web, compleixi els **requeriments visuals de marca** i afavoreixi una **experiència d'usuari** agradable i efectiva.

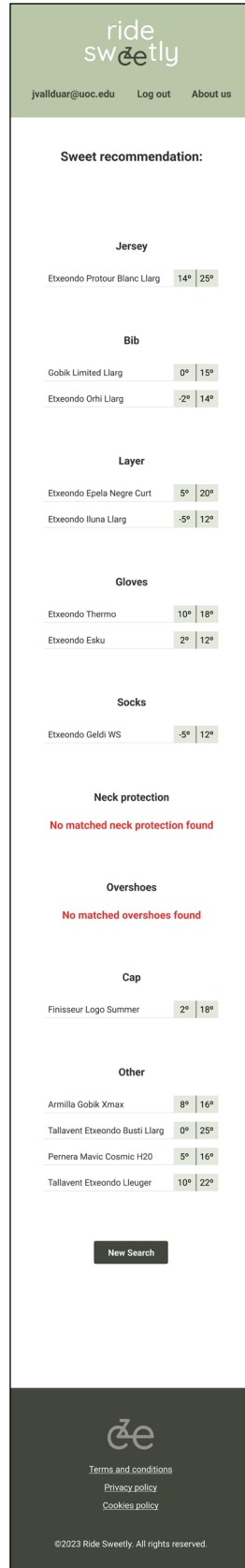
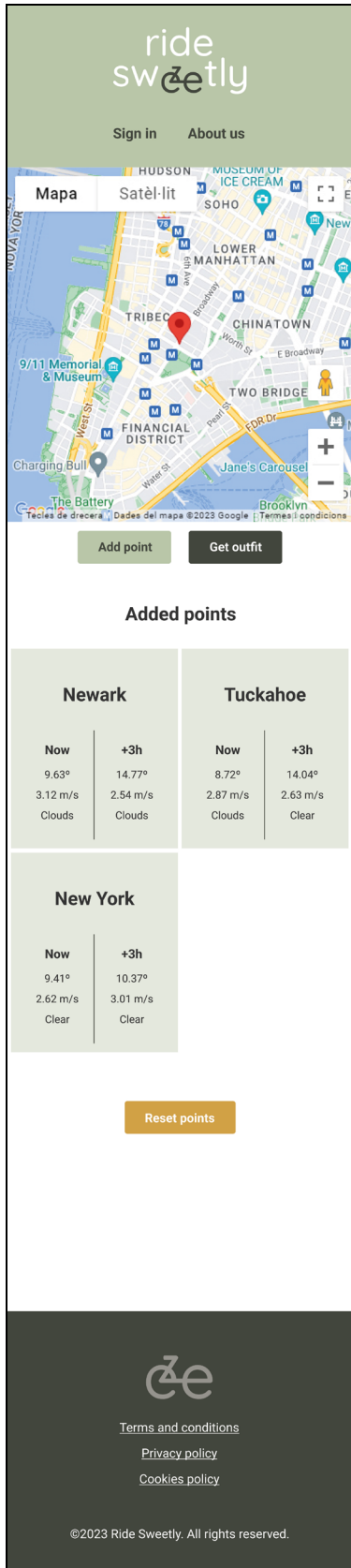
També es pot navegar per la versió d'escriptori (1024x600px) a partir del prototip bàsic al que es pot accedir a partir del següent enllaç:

<https://www.figma.com/proto/y1sEtRrp4icle66OlBETBe/Mockups?type=design&node-id=8-1452&scaling=scale-down&page-id=8%3A1451&starting-point-node-id=8%3A1452>

Tots els dissenys han estat realitzats amb **Figma**, una de les eines més conegudes per al disseny i creació de prototips per a pàgines web.

Mockups

Mobile mockups 390px: INICI



Mobile mockups 390px: SIGN IN i SIGN UP


Sign in to Ride Sweetly X

Email

Password

[Log in](#)

Don't have an account? [Create account](#)



Sign up to Ride Sweetly


Account information

Email

Password

[Sign up](#)

[Cancel](#)



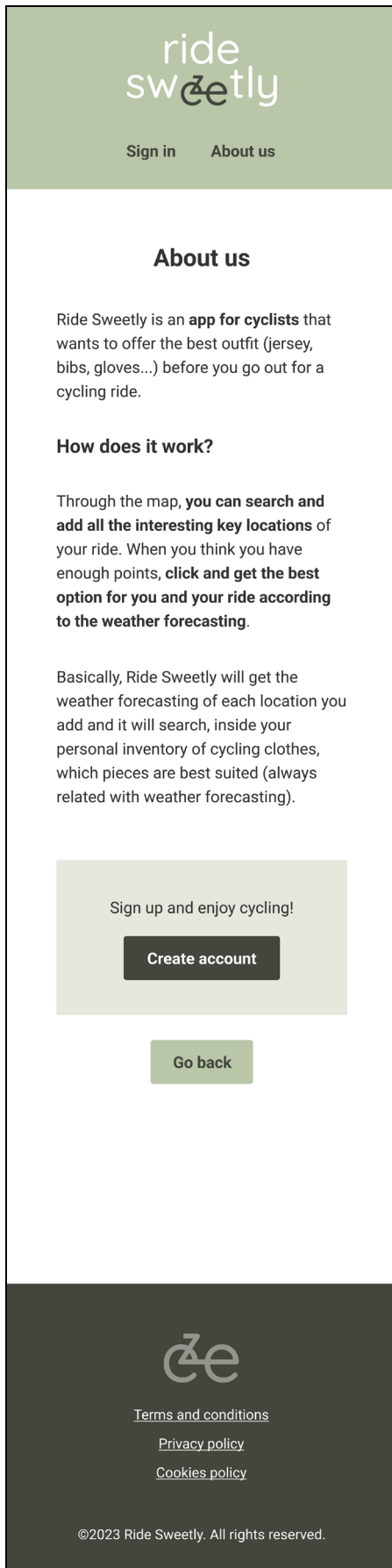
[Terms and conditions](#)

[Privacy policy](#)

[Cookies policy](#)

©2023 Ride Sweetly. All rights reserved.

Mobile mockups 390px: ABOUT US i PROFILE



Tablet mockups 768px: INICI

ride
swæetly

Sign in About us

Mapa Satèl·lit

©2023 Ride Sweetly. All rights reserved.

Add point
Get outfit

Added points

Newark

Now	+3h
9.63°	14.77°
3.12 m/s	2.54 m/s
Clouds	Clouds

Tuckhoe

Now	+3h
8.72°	14.04°
2.87 m/s	2.63 m/s
Clouds	Clear

New York

Now	+3h
9.41°	10.37°
2.62 m/s	3.01 m/s
Clear	Clear

Reset points

æ

[Terms and conditions](#)
[Privacy policy](#)
[Cookies policy](#)

©2023 Ride Sweetly. All rights reserved.

ride
swæetly

jvallduar@uoc.edu Log out About us

Sweet recommendation:

Jersey

Etxeondo Protour Blanc Lliarg	14°	25°
-------------------------------	-----	-----

Bib

Gobik Limited Lliarg	0°	15°
Etxeondo Orhli Lliarg	-2°	14°

Layer

Etxeondo Epela Negre Curt	5°	20°
Etxeondo Iluna Lliarg	-5°	12°

Gloves

Etxeondo Thermo	10°	18°
Etxeondo Esku	2°	12°

Socks

Etxeondo Geldi WS	-5°	12°
-------------------	-----	-----

Neck protection

No matched neck protection found

Overshoes

No matched overshoes found

Cap

Finnisseur Logo Summer	2°	18°
------------------------	----	-----

Other

Armilla Gobik Xmax	8°	16°
Tallavent Etxeondo Busti Lliarg	0°	25°
Pernera Mavic Cosmic H20	5°	16°
Tallavent Etxeondo Lleuger	10°	22°

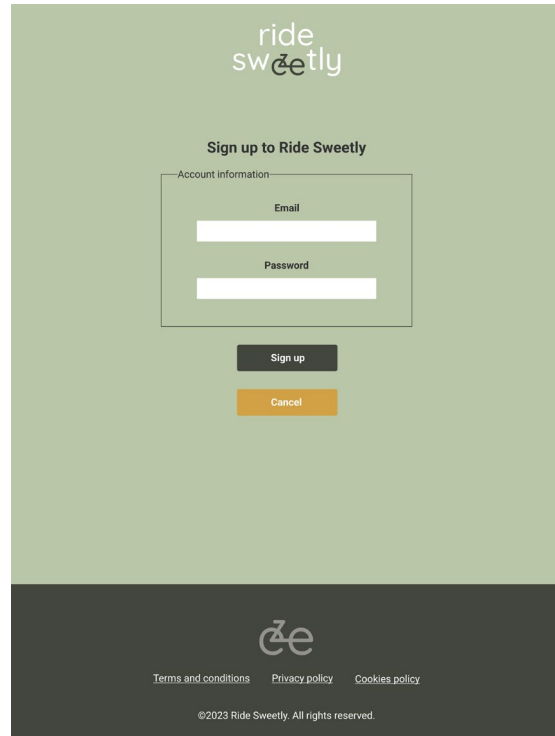
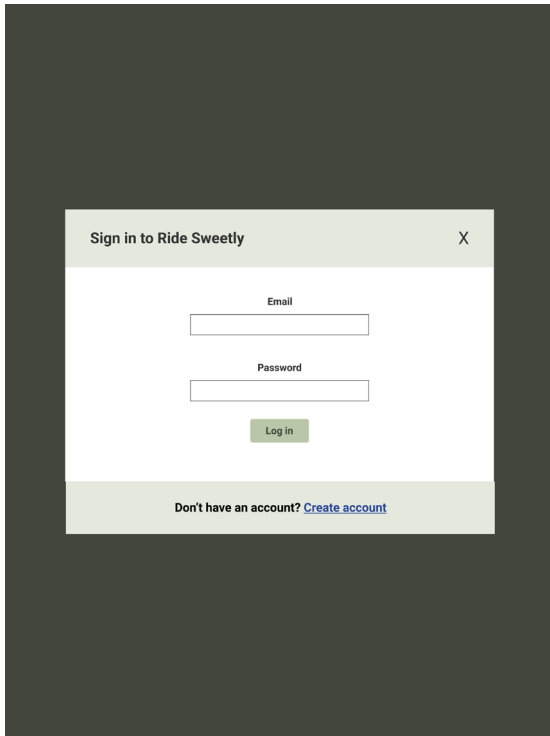
New Search

æ

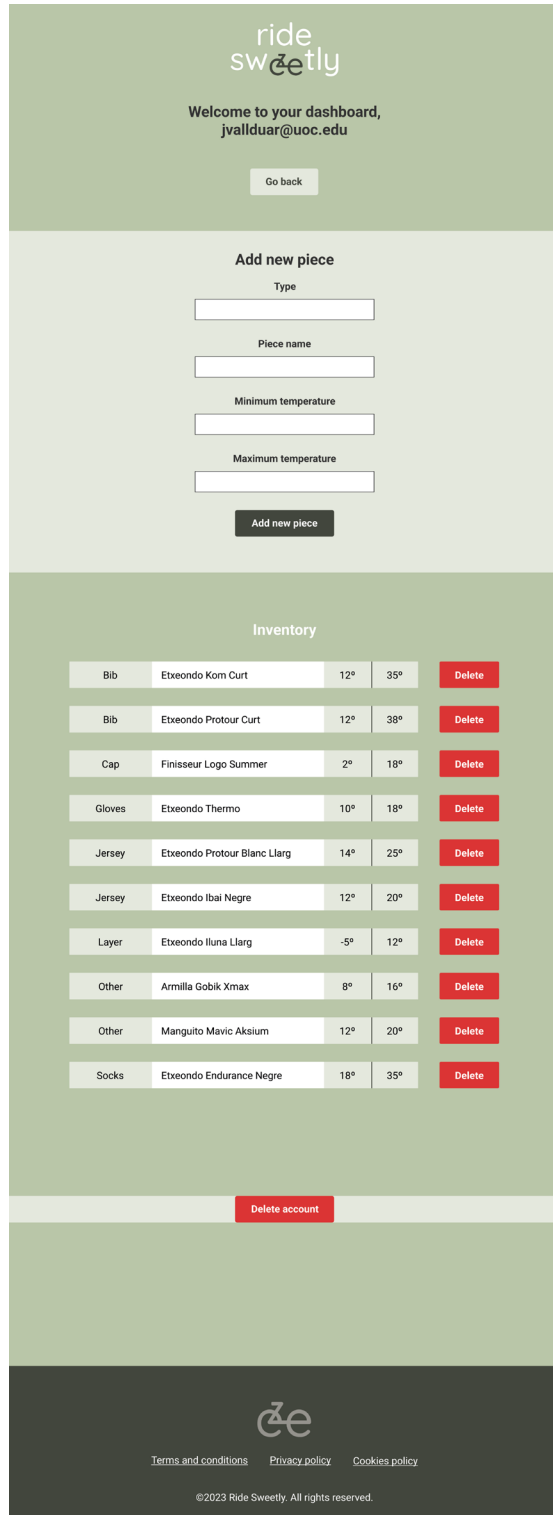
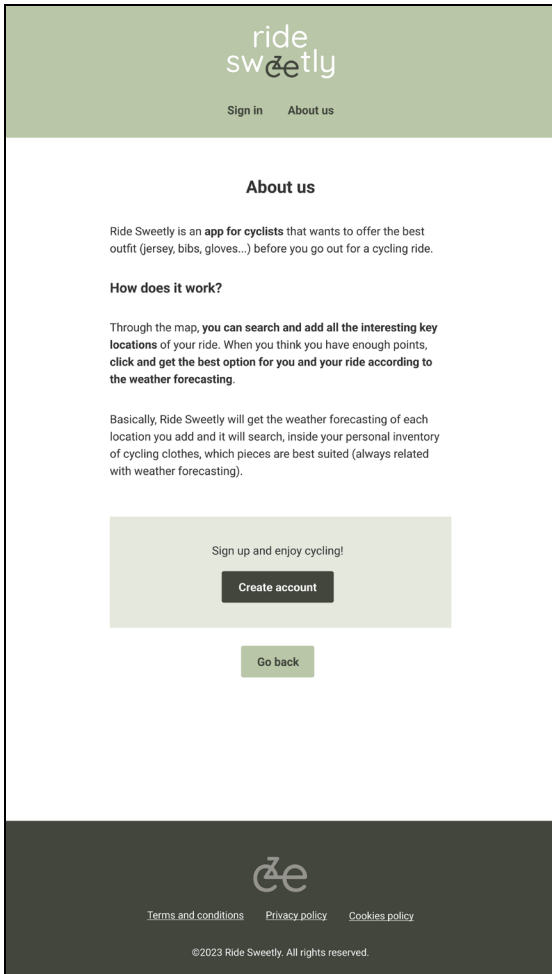
[Terms and conditions](#)
[Privacy policy](#)
[Cookies policy](#)

©2023 Ride Sweetly. All rights reserved.

Tablet mockups 768px: SIGN IN i SIGN UP



Tablet mockups 768px: ABOUT US i PROFILE



Desktop mockups 1024px: INICI

ride
sweetly

[Sign in](#) [About us](#)

Mapa Satèlit

Jersey City

Liberty House restaurant & Events

Liberty State Park

NOVA JERSEY NOVA YORK

HUDSON SQUARE

MUSEUM OF ICE CREAM

SOHO

BOWERY

EAST VILLAGE

LOWER MANHATTAN

CHINATOWN

LOWER EAST SIDE

TRIBECA

FINANCIAL DISTRICT

TWO BRIDGES

DUMBO

9/11 Memorial & Museum

Charging Bull

The Battery

Janc's Carousel

Manhattan Bridge

East River

Williamsburg Bridge

Peter Luger Steak H

L'Indi

Google

Dades del mapa ©2023 Google

Termes i condicions

Add point Get outfit

Added points

Newark		Tuckhoe	
Now	+3h	Now	+3h
9.63°	14.77°	8.72°	14.04°
3.12 m/s	2.54 m/s	2.87 m/s	2.63 m/s
Clouds	Clouds	Clouds	Clear

New York	
Now	+3h
9.41°	10.37°
2.62 m/s	3.01 m/s
Clear	Clear

Reset points

©2023 Ride Sweetly. All rights reserved. [Terms and conditions](#) [Privacy policy](#) [Cookies policy](#)

Sweet recommendation:

Jersey

Etxeondo Protour Blanc Llarg	14°	25°
------------------------------	-----	-----

Bib

Gobik Limited Llarg	0°	15°
Etxeondo Orhi Llarg	-2°	14°

Layer

Etxeondo Epela Negre Curt	5°	20°
Etxeondo Iluna Llarg	-5°	12°

Gloves

Etxeondo Thermo	10°	18°
Etxeondo Esku	2°	12°

Socks

Etxeondo Geldi WS	-5°	12°
-------------------	-----	-----

Neck protection

No matched neck protection found

Overshoes

No matched overshoes found

Cap

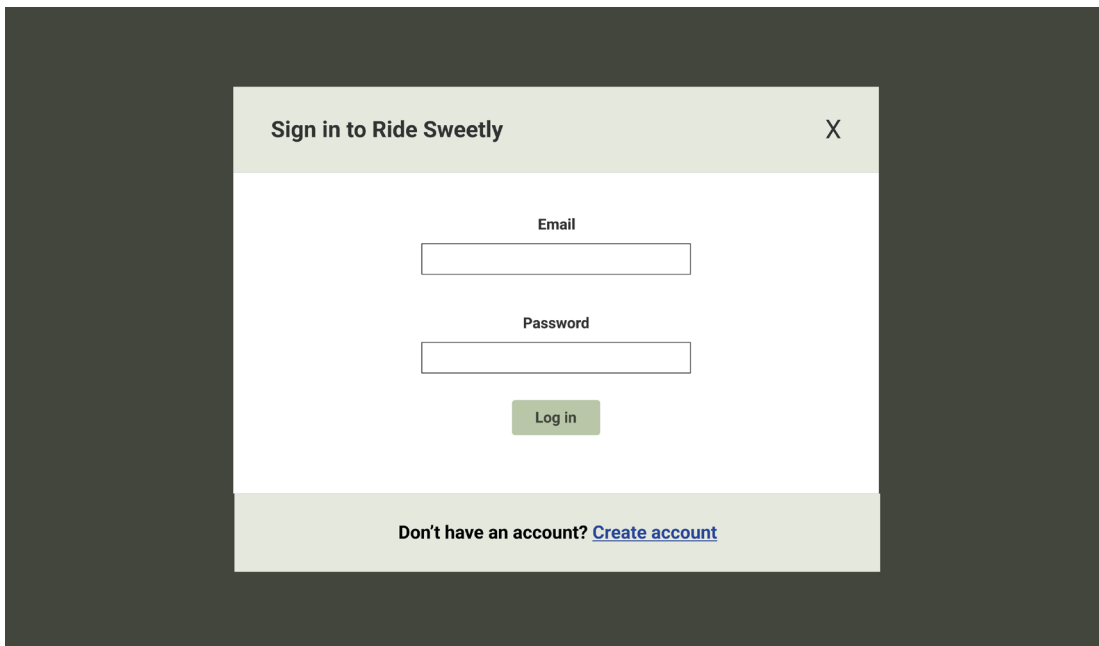
Finisseur Logo Summer	2°	18°
-----------------------	----	-----

Other

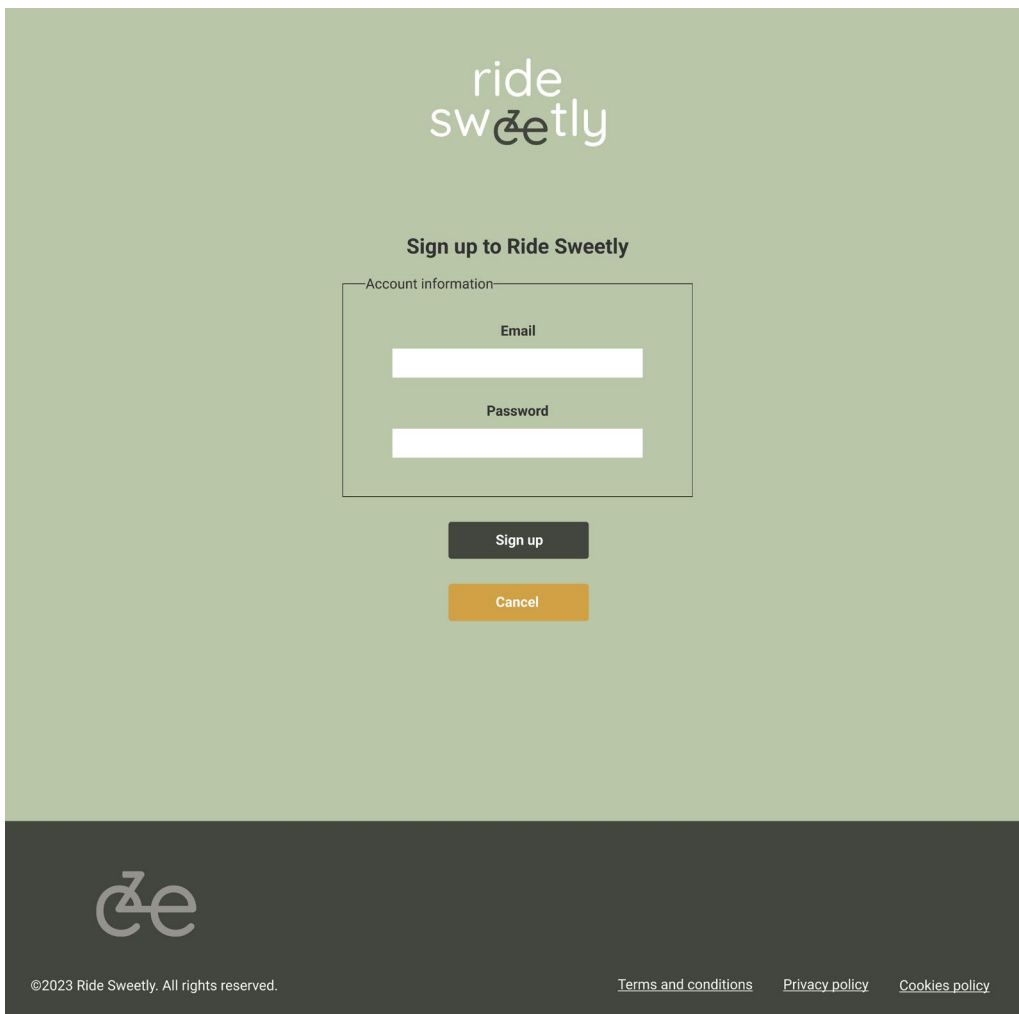
Armillia Gobik Xmax	8°	16°
Tallavent Etxeondo Busti Llarg	0°	25°
Pernera Mavic Cosmic H20	5°	16°
Tallavent Etxeondo Lleuger	10°	22°

New Search

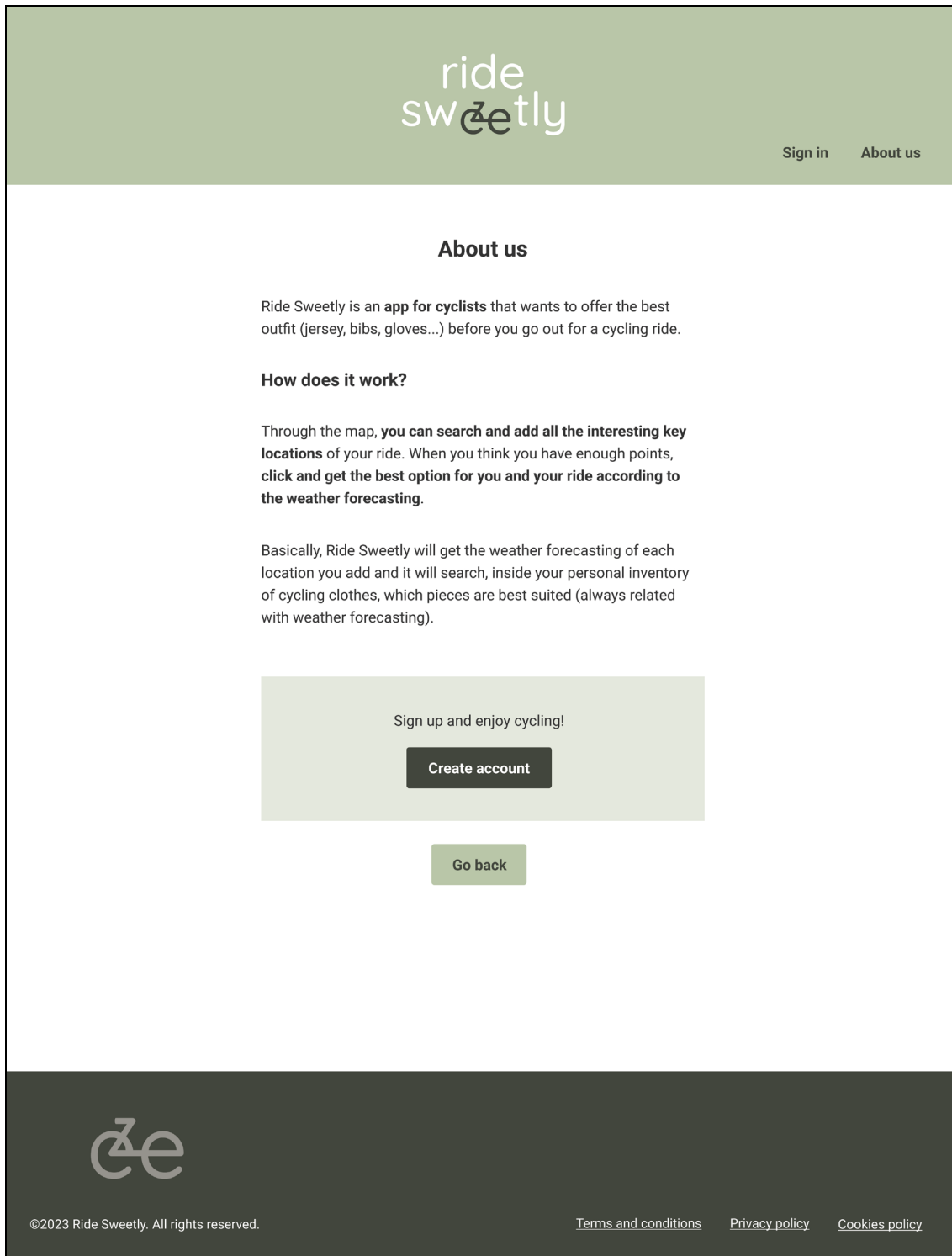
Desktop mockups 1024px: SIGN IN



Desktop mockups 1024px: SIGN UP



Desktop mockups 1024px: ABOUT US



Desktop mockups 1024px: PROFILE

ride
sweetly

Welcome to your dashboard,
jvallduar@uoc.edu

[Go back](#)

Add new piece

Type

Piece name

Minimum temperature


Maximum temperature

[Add new piece](#)

Inventory

Bib	Etxeondo Kom Curt	12°	35°	Delete
Bib	Etxeondo Protour Curt	12°	38°	Delete
Cap	Finisseur Logo Summer	2°	18°	Delete
Gloves	Etxeondo Thermo	10°	18°	Delete
Jersey	Etxeondo Protour Blanc Llarg	14°	25°	Delete
Jersey	Etxeondo Ibai Negre	12°	20°	Delete
Layer	Etxeondo Iluna Llarg	-5°	12°	Delete
Other	Armilla Gobik Xmax	8°	16°	Delete
Other	Manguito Mavic Aksium	12°	20°	Delete
Socks	Etxeondo Endurance Negre	18°	35°	Delete

[Delete account](#)



©2023 Ride Sweetly. All rights reserved. [Terms and conditions](#) [Privacy policy](#) [Cookies policy](#)

2.10. Compliment de requeriments legals a la pàgina

A la pàgina web deixarem preparats tres enllaços: *terms and conditions*, *privacy policy*, i *cookies policy*.

En aquests enllaços, abans del llançament de la pàgina, s'hauran de detallar els propòsits i finalitat d'aquesta, l'ús que se li donarà a la informació obtinguda de l'usuari, i tota la informació relativa a les galetes utilitzades. També s'inclourà informació sobre la titularitat del lloc i informació de contacte per possibles gestions de protecció de dades (GDPR ²⁹) i més.

Com que, ara per ara, es tracta d'un prototip que forma part d'un treball final de grau, es deixaran aquests enllaços sense contingut. Únicament funcionen com a recordatori d'uns requeriments legals necessaris que qualsevol pàgina que obté informació personal d'usuaris i que presta algun tipus de servei a aquests ha de complir.

2.11. Desplegament web

Fins ara hem estat utilitzant un servidor local. Ara, un cop finalitzat el desenvolupament, necessitarem desplegar tot plegat en un servidor accessible. En aquest cas utilitzarem el servidor de DonDominio, ja que dispo d'un domini comprat juntament d'un allotjament bàsic, que inclou la creació de fins a 10 bases de dades.

Per crear la base de dades, accedirem al tauler personal de la plataforma DonDominio i, dins dels dominis disponibles, seleccionarem l'allotjament de josepvallejo.com. Un cop dins, a la pestanya "Bases de datos" clicarem a "Crear Base de Datos":

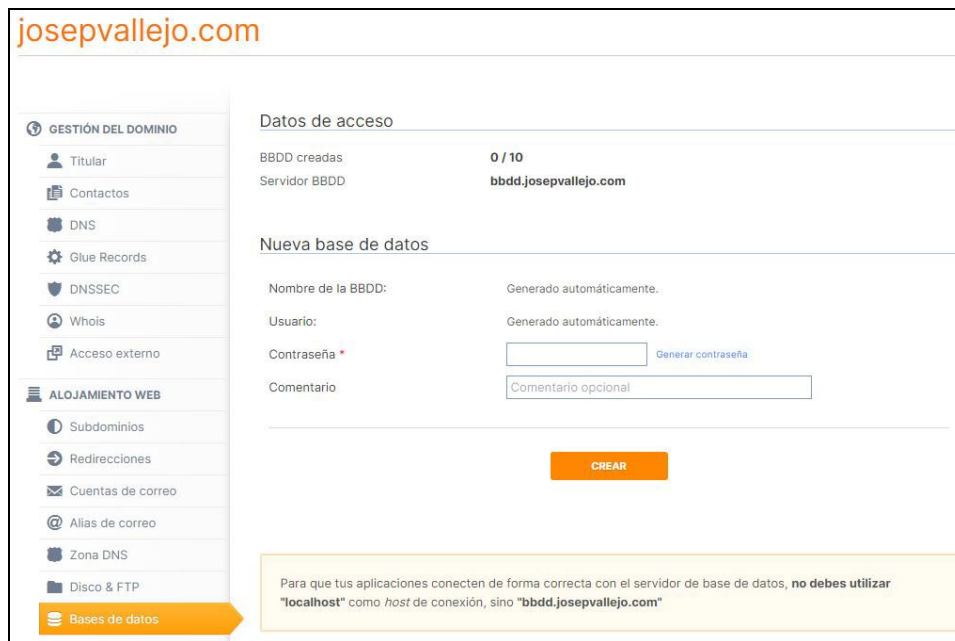


Figura 24. Pantalla de creació de la base de dades.

Un cop generada la contrasenya li donarem a “crear” i ja disposarem de la informació:

Servidor BBDD: bbdd.josepvallejo.com

Usuari: ddb202034

Contrasenya: *****

Base dades: ddb202034

Per evitar haver de tornar a crear la taula de la base de dades, el que farem és exportar-la de la base de dades del servidor local i importar-la a la nova base de dades. El procés és molt senzill, ja que quasi només s’ha de clicar sobre “exportar” i “importar”.

Finalment, haurem de modificar les configuracions dels arxius PHP ³⁰ que executen les connexions a la BBDD, per incloure la informació de la nova:

```
<?php header("Content-Type: text/html;charset=utf-8");

function connectToBBDD() {
    //Paràmetres de connexió
    $server = "bbdd.josepvallejo.com";
    $user = "ddb202034";
    $pass = "*****";
    $bd = "ddb202034";
    //Realització de la connexió
    $con = mysqli_connect ($server, $user, $pass, $bd);
```

```
return $con;  
}  
?>
```

A continuació, faltirà penjar tots els arxius que formen la nostra pàgina web. Per a fer-ho, accedirem a la carpeta “public” del domini josepvallejo.com, a través de la pestanya “Disco & FTP” del tauler personal de DonDominio. Dins d’aquesta crearem la carpeta “ridesweetly” i arrossegarem tots els arxius.

Així, accedint a www.josepvallejo.com/ridesweetly hauríem de ser capaços d’accedir a la pàgina web i que aquesta funcioni correctament.

Problema

Alhora de fer les proves des del servidor, aquest ens ha denegat l’obtenció de dades quan intentàvem accedir a aquestes (**error 403 Forbidden**). Després d’estar fent proves i revisant el codi, s’ha descobert que això és degut a que, fins el moment, hem estat construint les peticions SLQ directament des dels arxius JavaScript, i després passàvem aquesta petició ja construïda directament a l’arxiu PHP que executa la petició cap a la base de dades.

El servidor de DonDominio actua així ja que detecta aquest comportament i, per protegir-se de possibles atacs denominats [SQL Injection](#) (els hackers aprofiten aquesta debilitat per **inserir codi i manipular la consulta SQL original**), no permet l’execució de la consulta.

Solució

Veient la situació, s’ha decidit actualitzar part del codi, de manera que **des dels arxius JavaScript només es passin cap als arxius PHP els paràmetres necessaris per personalitzar les consultes**. Un cop que l’arxiu PHP que ha d’executar la consulta disposa dels paràmetres, es crea la petició directament des d’aquest últim arxiu.

3. Resultats

El producte final obtingut, després d'aquesta fase inicial, ha estat **una pàgina web totalment operativa** on un usuari es pot donar d'alta a la plataforma, iniciar sessió gràcies a Firebase, i crear el seu perfil de productes.

La pàgina web utilitza correctament l'API de Google Maps JavaScript per mostrar per pantalla el mapa a partir del qual l'usuari anirà seleccionant els punts (localitzacions) que posteriorment s'enviaran a l'API de Open Weather Map. Aquesta última, anirà retornant, per a cada punt, el pronòstic del temps.

Un cop seleccionats tots els punts que es creguin necessaris, ja es pot executar la consulta per cercar, sempre segons el pronòstic del temps obtingut, els ítems dels que disposa l'usuari que millor s'adaptin a les condicions meteorològiques.

Per fer la consulta dels ítems, es calcula una mitja sobre les temperatures obtingudes i es cerca, dins de l'inventari personal de l'usuari que està amb la sessió iniciada, quins elements estan dins del rang de temperatures previst.

Perquè l'usuari pugui anar afegint/eliminat ítems del seu inventari, i així personalitzar-lo adequadament, s'ha creat un pàgina de perfil on aquest disposa del formulari d'inserció. En aquest formulari podrà indicar el tipus de la peça, un nom descriptiu, i el rang de temperatures que li vol associar (temperatura mínima i màxima d'ús recomanades per peça).

A partir d'aquest producte, s'ha assolit una base funcional i operativa que s'utilitzarà per evolucionar el projecte a les properes fases futures.

URL: <http://www.josepvallejo.com/ridesweetly>

4. Conclusions i treballs futurs

El projecte ha assolit l'objectiu principal: l'execució d'una pàgina web que cobreix una necessitat existent dins del col·lectiu ciclista. A més, planta les bases i l'inici del full de ruta d'un projecte amb gran potencial de creixement i diversificació.

Durant el treball, s'ha desenvolupat correctament la comunicació entre la pàgina web i les APIs de Google Maps JavaScript i Open Weather Map. Aquestes APIs ens han proporcionat les funcionalitats esperades.

També s'ha pogut treballar l'autenticació d'usuari utilitzant Firebase i gràcies a la base de dades creada amb MySQL (mitjançant phpMyAdmin), existeix la possibilitat de disposar d'inventaris personalitzats per a cada usuari, que poden actualitzar al seu gust a través de cada perfil.

Finalment, s'han pogut lligar els resultats donats per les APIs amb l'inventari personal de cada usuari, definint les sentències SQL necessàries creades a partir de funcions JavaScript, que ens han facilitat la gestió de la informació de la pàgina web.

Una planificació conservadora i amb bastant folgança per a cada tasca ens ha permès poder complir-la sense gaires entrebancs. Ha estat clau dedicar el temps necessari a identificar clarament les tasques principals i indispensables per dur a terme el projecte. Amb aquesta planificació s'ha intentat no abastar més del que els *timings* i coneixements permetien, i sent molts conscients de les limitacions personals. Amb això, s'ha aconseguit una execució correcta i amb suficient temps de reacció als problemes sorgits.

La metodologia prevista, els sistemes previstos, i l'estructura del projecte prevista ha estat com s'havia plantejat des d'un inici. Sobre aquest punt, ha estat de gran rellevància la cerca d'informació un cop s'han tingut clares les necessitats principals que s'havien de cobrir:

- Necessitat d'obtenció de dades de pronòstic del temps segons una localització. Solucionat amb l'API d'Open Weather Map.
- Necessitat de marcatge sobre un mapa per obtenir dades de latitud i longitud de localitzacions específiques. Solucionat amb l'API de Maps JavaScript de Google.
- Necessitat de crear una base de dades. Solucionat a través de MySQL/phpMyAdmin.

- Necessitat d'autenticar usuaris. Solucionat amb Firebase.

Partint de l'anterior, també ha estat important definir l'HTML, el CSS, i la utilització de JavaScript, sense cap *framework*³¹/entorn/aplicació com [Vue.js](#) o [React](#) pel mig, com eines per construir i lligar tota la aplicació. S'ha volgut prioritzar l'assoliment de l'objectiu fins al final, i vistos els coneixements bàsics adquirits de Vue.js per un mateix durant el grau de Multimèdia i els *timings* disponibles per al desenvolupament, era un risc utilitzar una d'aquestes aplicacions.

Sobre els impactes de sostenibilitat, ètic-socials, i de diversitat previstos, l'aplicació aconsegueix afavorir l'ús de la bicicleta i de la forma més sostenible possible. En les línies de futur exposades més endavant, el desenvolupament de Ride Sweetly gira entorn l'acció climàtica i consum responsables, sense distincions de gènere, sexe o culturals.

Línies de treball futur

A l'inici del projecte s'han marcat 3 fases que descriuen a curt i mig termini el pla de ruta que hauria de seguir Ride Sweetly:

- Fase 1: Desenvolupament i llançament de la pàgina web (punt de partida d'aquest treball).
- Fase 2: Desenvolupament de l'aplicació mòbil.
- Fase 3: Integració de Ride Sweetly amb les aplicacions més conegudes del sector.

A continuació, passem a comentar el que es podria dur a terme durant la fase 2 i 3 del projecte.

Fase 2:

Durant aquesta fase, s'haurà de triar sobre quin entorn es vol **desenvolupar l'aplicació**. Per tal de minimitzar despeses i recursos, es podria triar un llenguatge multiplataforma con React Native.

També, s'haurien de redefinir els sistemes utilitzats fins el moment, per tal d'intentar unificar plataformes (com per exemple, portar-ho tot a entorns com [Microsoft-Azure](#) o plataformes de Google) i, d'aquesta manera, potenciar més fàcilment la estabilitat i la seguretat del producte.

Referent a l'experiència d'usuari, es podrien millorar:

- Procés d'alta d'usuari: Que l'usuari pugui introduir més informació, com el seu nom i cognoms o la data de naixement, per tal de personalitzar encara més l'experiència. També s'hauria d'afegir la validació de direcció de correu electrònic amb l'enviament d'un enllaç de confirmació. Per acabar, es podria obligar a la introducció de contrasenyes amb un mínim de condicions de seguretat.
- Perfil d'usuari: Que l'usuari pugui editar un ítem ja introduït prèviament a l'inventari, ja sigui per canviar el rang de temperatures o el nom d'aquest. També s'han d'incloure opcions d'idioma disponibles i altres com, per exemple, si es vol visualitzar la informació en graus Celsius o Fahrenheit, o la velocitat del vent en metres, quilòmetres, milles... per segon.
- Procés d'obtenció del *forecasting* per localització: L'usuari va cercant les localitzacions a través de la navegació pel mapa principal. Seria bo afegir una segona opció, com una barra de cerca que et porti directament al punt d'interès introduït. També s'ha de considerar si mostrar més informació sobre el pronòstic del temps pot donar més valor a l'aplicació. A més, seria interessant que l'usuari pogués eliminar, de forma individual, les localitzacions que va afegint (ja sigui perquè s'ha equivocat introduint un punt o perquè ha canviat d'opinió). Ara mateix s'han d'eliminar totes i tornar a començar.
- Procés d'obtenció d'ítems de l'inventari: Aquí, una de les grans millores que es podrien introduir, i que ens permet posicionar Ride Sweetly com una aplicació sostenible i compromesa amb el comerç just, responsable i de proximitat, és la inserció de recomanacions de compra quan es detecta que a l'usuari li falta alguna peça de roba. Es tracta d'establir acords amb marques o botigues de tots els països que respectin una sèrie de compromisos amb el planeta i la sostenibilitat sempre afavorint el comerç de proximitat. Així, quan al mostrar els resultats l'aplicació detectés que a l'usuari li falta una samarreta d'hivern, es podria inserir una recomanació de botiga on comprar-la segons la seva geolocalització (comerç de proximitat).

- Secció de marques justes amb el planeta: Seguint amb l'últim punt, es podria afegir una secció on es recomanin botigues i marques de tot arreu. Òbviament, totes aquestes haurien de complir el criteris necessaris per formar part d'aquesta secció, que s'hauria d'anar actualitzant periòdicament.
- Donació: Per intentar fer tot plegat més sostenible, es podria afegir l'opció de, ja sigui de forma puntual o mensual, que l'usuari pugui fer una donació d'un import de diners a determinar a Ride Sweetly.
- Obtenció de resultats de l'inventari: Un tema molt interessant (i que donaria per un segon TFG) seria esbrinar quins serien els criteris adients a tenir en compte alhora de cercar la roba adequada des de l'inventari de l'usuari segons el *forecast* del temps obtingut. En aquest treball, s'ha utilitzat la diferència entre les mitges obtingudes dels rangs de temperatura disponibles per a cada ítem de l'inventari, de manera que s'ordenen i seleccionen els ítems amb la menor diferència amb el *forecast* obtingut. Quin és el problema? Doncs que moltes peces de roba, en el món del ciclisme, presenten rangs de temperatura "òptims" massa amplis. Per tant, depenent de la percepció personal que tingui cada usuari sobre la calor o el fred, els resultats que presenti l'aplicació, tot i ser correctes segons la informació dels fabricants, pot ser inadequada per a l'usuari.

La solució al problema anterior, podria ser:

- 1- Que al perfil d'usuari, aquest pugui indicar la sensibilitat que té al fred o a la calor a partir d'una escala com:

Et consideres una persona...

- a) Amb molta sensibilitat al fred
- b) Amb bastant sensibilitat al fred
- c) Amb sensibilitat al fred
- d) Normal
- e) Amb sensibilitat a la calor
- f) Amb bastanta sensibilitat a la calor
- g) Amb molta sensibilitat a la calor

Segons la resposta que indiqui l'usuari, es pot personalitzar la cerca per obtenir uns resultats més acurats segons les preferències d'aquest.

- 2- Que al introduir un ítem a l'inventari (a través del formulari), es defineixi una comprovació que detecti si el rang d'aquest ítem és massa ampli i, si és així, s'avisí a l'usuari perquè el redueixi segons el seu criteri.

La solució número 1 seria molt personalitzada i menys invasiva, mentre que la solució 2 podria perjudicar l'experiència d'usuari i crear cert estrès a aquest, al donar-li la responsabilitat de decidir.

La finalització d'aquesta segona fase ajudarà a consolidar els valors de Ride Sweetly i hauria de ser la punta de llança d'aquesta per posicionar-se correctament al mercat.

Fase 3:

Aquesta fase s'intentaria executar en una termini mig-llarg, ja que serà necessari haver consolidat mínimament la marca Ride Sweetly al mercat.

Es tracta de crear sinergies amb els actors principals del sector, com Wikiloc o Strava, establir aliances amb aquests, i estendre els valors de Ride Sweetly cap a altres marques en la mesura del possible. Aquesta fase tracta d'**impulsar dos aspectes** molt importants:

- 1- La marca Ride Sweetly al mercat, millorant la seva visibilitat.
- 2- Els valors de Ride Sweetly, entre d'altres, cap a marques molt importants del sector i, així, afavorir el compromís d'aquestes amb la sostenibilitat i el comerç just.

5. Glossari

- ¹ *Forecast* : Terme anglès que significa “pronòstic” o “previsió”. Terme molt utilitzat, individualment, per fer referència a la previsió del temps.
- ² *Outfit* : Terme anglès que significa “vestit” o que fa referència a un conjunt de peces de vestir.
- ³ API : Acrònim en anglès de «*Application Programming Interface*».
- ⁴ *Commuting* : Terme anglès que fa referència als desplaçaments recurrents de, per exemple, casa fins la feina.
- ⁵ ODS : Acrònim de «Objectius de Desenvolupament Sostenible».
- ⁶ TFG : Acrònim de «Treball Final de Grau».
- ⁷ Android : Sistema operatiu, de codi obert, per a dispositius mòbils. Actualment, propietat de Google LLC.
- ⁸ iOS : Sistema operatiu, de codi tancat, per a dispositius mòbils. Actualment, propietat d’Apple Inc.
- ⁹ *App* : Abreviatura del terme anglès *application*.
- ¹⁰ HTML : Llenguatge de marcatge per programar pàgines web. Acrònim en anglès de «*HyperText Markup Language*».
- ¹¹ CSS : Llenguatge de disseny gràfic, sobretot per aplicar estil visual a pàgines web. Acrònim en anglès de «*Cascading Style Sheets*».
- ¹² JavaScript : Llenguatge de programació del costat del client, basat en objectes.
- ¹³ WAMP : Sistema, basat en Windows, que utilitza un conjunt de programes per gestionar pàgines web. Acrònim de «Windows, Apache, MySQL, PHP».
- ¹⁴ MySQL : Programa de gestió de bases de dades de codi obert.

- ¹⁵ BBDD : Abreviatura de “Bases de dades”.
- ¹⁶ Gantt : Referent al diagrama que serveix per temporitzar i assignar recursos a tasques d’un projecte.
- ¹⁷ *Wireframe* : Representació visual sobre l’estructura d’una pàgina web.
- ¹⁸ *Localhost* : Terme que fa referència a la direcció IP utilitzada per instal·lar servidors web en un ordinador.
- ¹⁹ phpMyAdmin : Programa per gestionar bases de dades MySQL.
- ²⁰ *Cookies* : Informació emmagatzemada de forma temporal pel navegador web.
- ²¹ AJAX : Sistema utilitzat per carregar dades des d’un servidor en segon pla, sense necessitat de recarregar la pàgina o bloquejar el seu funcionament. Acrònim en anglès de «*Asynchronous JavaScript And XML*».
- ²² JSON : Format de text utilitzat per l’intercanvi de dades. Acrònim en anglès de «*JavaScript Object Notation*».
- ²³ *callback* : Referent a la funció que es passa com argument en una altra perquè sigui executada en algun moment.
- ²⁴ SQL : Llenguatge utilitzat per gestionar informació de bases de dades relacionals. Acrònim en anglès de «*Structured Query Language*».
- ²⁵ *Branding* : Terme anglès que significa “marca”. S’utilitza per referenciar al conjunt d’accions que ajuden a construir i posicionar una marca.
- ²⁶ *Friendly* : Terme anglès que significa “amigable”.
- ²⁷ GIF : Format de compressió d’imatges, molt utilitzat a pàgines web. Acrònim en anglès de «*Graphics Interchange Format*».
- ²⁸ *Open source* : Paraules en anglès que signifiquen “codi obert”.
- ²⁹ GDPR : Reglament europeu sobre la protecció de dades de persones.

Acrònim en anglès de «*General Data Protection Regulation*».

³⁰ PHP : Llenguatge de programació del costat del servidor. Originalment, acrònim en anglès de «*Personal Home Page*».

³¹ *Framework* : Terme anglès que significa “marc” o “entorn de treball”.

6. Bibliografia

- [1] Haubold, H. (2020). ¿El aumento de la movilidad ciclista a raíz de la COVID-19 es real? ¡Los números dicen que sí! *Velo-City*. Recuperat el 2 de març de 2023 de: <https://www.velo-city2021.com/es/blog/el-aumento-de-la-movilidad-ciclista-a-raiz-de-la-covid-19-es-real-los-numeros-dicen-que-si/#>
- [2] BICIO. (2021). Las mejores aplicaciones para planificar rutas en bici *Sport*. Recuperat el 2 de març de 2023 de: <https://www.sport.es/bicio/las-mejores-aplicaciones-para-planificar-rutas-en-bici/>
- [3] Jiménez, B. (2023). Una buena reputación en sostenibilidad ayuda (y mucho) a las marcas a aumentar su valor económico. *Marketing Directo*. Recuperat el 3 de març de 2023 de: <https://www.marketingdirecto.com/marketing-general/tendencias/informe-valor-economico-sostenibilidad-marcas>

7. Annexos

En els annexes es troba el codi final utilitzat per a la realització de la pàgina web, de la següent forma:

1. Pàgines HTML
2. Pàgines CSS
3. Pàgines JavaScript
4. Pàgines PHP

Annexa 1 – Pàgines HTML

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="description" content="Previsió del temps per posicionament | TGF -
Desenvolupament d'aplicacions interactives">
  <meta name="keywords" content="forecast, geolocation">
  <meta name="author" content="Josep Vallejo Duarri">
  <meta name="application-name" content="ride sweetly">
  <title>ride sweetly</title>
  <link rel="shortcut icon" href="img/favicon.svg" />
  <link href='http://fonts.googleapis.com/css?family=Roboto:400,300,700'
rel='stylesheet' type='text/css'>
  <link rel="stylesheet" type="text/css" href="css/main.css">
  <link rel="stylesheet" type="text/css" href="css/modal.css">
  <script src="js/auth-signin.js" type="module"></script>
</head>
<body>
  <header>
    <a href="index.html" target="_self"></a>
  <nav>
    <ul id="menu">
```

```

        <li class="logout-display"><button type="button" data-open="signin-
modal">Sign in</button></li>
        <li class="login-display"><a href="profile.html"
target="_self"><button id="session-owner" type="button"></button></a></li>
        <li class="login-display"><button id="logout-session"
type="button">Log out</button></li>
        <li><button id="about-button" type="button">About us</button></li>
    </ul>
</nav>
</header>

<!--Modal per poder iniciar sessió-->
<div class="modal" id="signin-modal">
    <div class="modal-dialog">
        <header class="modal-header">
            <h2>Sign in to Ride Sweetly</h2>
            <button class="close-modal" aria-label="close modal" data-
close>X</button>
        </header>
        <section class="modal-content">
            <form id="signin-form">
                <label for="email">Email</label>
                <input type="email" id="email" name="email" required>
                <label for="password">Password</label>
                <input type="password" id="pass" name="password" required>
                <button class="button" type="submit">Log in</button>
            </form>
        </section>
        <footer class="modal-footer">
            <h3>Don't have an account? <a href="signup.html"
target="_self">Create account</a></h3>
        </footer>
    </div>
</div>

<!--Modal per confirmar l'eliminació d'usuari-->
<div class="modal" id="delete-modal">
    <div class="modal-dialog">
        <header class="modal-header">
            <h2>Confirm your account information...</h2>
            <button class="close-modal" aria-label="close modal" data-
close>X</button>
        </header>
        <section class="modal-content">
            <form id="delete-form">
                <label for="email">Email</label>
                <input type="email" id="email2" name="email" required>
                <label for="password">Password</label>
                <input type="password" id="pass2" name="password" required>

```



```

        <button class="button" type="submit">Delete</button>
    </form>
</section>
</div>
</div>

<section id="content">
    <!--Imatge GIF que només es mostra durant la càrrega de contingut-->
    <div class="loading">
        
    </div>

    <div id="map">
        <!--Imatge GIF que només es mostra durant la càrrega de contingut-->
        <div class="loading2">
            
        </div>
    </div>

    <div id="main-buttons">
        <!--Botó que dispara la crida a l'API de Open Weather Map-->
        <button id="forecast-button" class="button">Add point</button>
        <!--Botó que dispara la petició a la base de dades-->
        <button id="outfit-button" class="button">Get outfit</button>
    </div>

    <div id="forecast-results">
        <!--Imatge GIF que només es mostra durant la càrrega de contingut-->
        <div class="loading">
            
        </div>
    </div>
    <div id="reset-points-container">
        <button id="reset-points-button" class="button">Reset points</button>
    </div>
</section>

<footer id="footer"></footer>

<!--Carreguem l'API de Maps JavaScript!-->
<script async defer
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBRBxqfhgLG9gJEiKpIVfAT5krU2dMg
9Yc&callback=setPosition">
</script>
<script type="text/javascript" src="js/jquery-3.6.0.min.js"></script>
<script src="js/main.js"></script>
<script src="js/modal.js"></script>

```

```

<!--Carreguem el footer a la pàgina-->
<script>
  $(document).ready(function () {
    $("#footer").load("footer.html");
  });
</script>
</body>
</html>

```

aboutus.html

```

<div id="about-content">
  <h1>About us</h1>
  <p>Ride Sweetly is an <strong>app for cyclists</strong> that wants to offer the
  best outfit (jersey, bibs, gloves...) before you go out for a cycling ride.</p>
  <h2>How does it work?</h2>
  <p>Through the map, <strong>you can search and add all the interesting key
  locations</strong> of your ride. When you think you have enough points, <strong>click
  and get the best option for you and your ride according to the weather
  forecasting</strong>.</p>
  <p>Basically, Ride Sweetly will get the weather forecasting of each location you
  add and it will search, inside your personal inventory of cycling clothes, which
  pieces are best suited (always related with weather forecasting).</p>
  <div id="about-signup">
    <p>Sign up and enjoy cycling!</p>
    <a href="signup.html" target="_self"><button type="button"
  class="button">Create account</button></a>
  </div>
  <a href="index.html" target="_self"><button type="button" class="button">Go
  back</button></a>
</div>

```

profile.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="description" content="Previsió del temps per posicionament | TGF -
  Desenvolupament d'aplicacions interactives">
  <meta name="keywords" content="forecast, geolocation">
  <meta name="author" content="Josep Vallejo Duarri">
  <meta name="application-name" content="ride sweetly">
  <title>ride sweetly - profile</title>
  <link rel="shortcut icon" href="img/favicon.svg" />

```

```

    <link href='http://fonts.googleapis.com/css?family=Roboto:400,300,700'
rel='stylesheet' type='text/css'>
    <link rel="stylesheet" type="text/css" href="css/main.css">
    <link rel="stylesheet" type="text/css" href="css/profile.css">
</head>

<body>
  <header>
    <a href="index.html" target="_self"></a>
  </header>

  <section id="content" class="profile-content">
    <div id="goback">
      <a href="index.html" id="go-back-button" target="_self"><button
type="button" class="button">Go back</button></a>
    </div>

    <div id="add-inventory">
      <h2>Add new piece</h2>
      <form id="add-piece-form">
        <label for="type">Type</label>
        <select name="type" id="select-type" form="add-piece-form" required>
          <option value="Jersey">Jersey</option>
          <option value="Bib">Bib</option>
          <option value="Layer">Layer</option>
          <option value="Gloves">Gloves</option>
          <option value="Socks">Socks</option>
          <option value="Neck">Neck</option>
          <option value="Cap">Cap</option>
          <option value="Overshoes">Overshoes</option>
          <option value="Other">Other</option>
        </select>
        <label for="name">Piece name</label>
        <input type="text" name="name" id="item-name" maxlength="30"
required>
        <label for="mintemp">Minimum temperature</label>
        <input type="number" name="mintemp" id="item-mintemp" required>
        <label for="maxtemp">Maximum temperature</label>
        <input type="number" name="maxtemp" id="item-maxtemp" required>
        <button class="button" type="submit">Add new piece</button>
      </form>
    </div>

    <div id="inventory">
      <h2>Inventory</h2>
      <!--Imatge GIF que només es mostra durant la càrrega de contingut-->
      <div class="loading">
        

```

```

        </div>
    </div>

    <div id="delete-account">
        <button type="button" id="delete-account-button" class="button">Delete
account</button>
    </div>
</section>

<footer id="footer"></footer>

<script type="text/javascript" src="js/jquery-3.6.0.min.js"></script>
<script type="text/javascript" src="js/profile.js"></script>
<!--Carreguem el footer a la pàgina-->
<script>
    $(document).ready(function () {
        $("#footer").load("footer.html"); // Carreguem el footer
        getItems();
    });
</script>
</body>
</html>

```

signup.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Previsió del temps per posicionament | TGF -
Desenvolupament d'aplicacions interactives">
    <meta name="keywords" content="forecast, geolocation">
    <meta name="author" content="Josep Vallejo Duarri">
    <meta name="application-name" content="ride sweetly">
    <title>ride sweetly - new account</title>
    <link rel="shortcut icon" href="img/favicon.svg" />
    <link href='http://fonts.googleapis.com/css?family=Roboto:400,300,700'
rel='stylesheet' type='text/css'>
    <link rel="stylesheet" type="text/css" href="css/main.css">
    <link rel="stylesheet" type="text/css" href="css/signup.css">
    <script src="js/auth-signup.js" type="module"></script>
</head>

<body>
    <header>

```

```

    <a href="index.html" target="_self"></a>
  </header>

  <section id="content" class="signup-content">
    <h1>Sign up to Ride Sweetly</h1>
    <form id="signup-form">
      <fieldset>
        <!--Imatge GIF que només es mostra durant la càrrega de contingut-->
        <div class="loading">
          
        </div>
        <legend>Account information</legend>
        <label for="email">Email</label>
        <input type="email" id="email" name="email" required>
        <label for="password">Password</label>
        <input type="password" id="pass" name="password" required>
      </fieldset>
      <button class="button" type="submit">Sign up</button>
      <a href="index.html" id="cancel-button" target="_self"><button
class="button" type="button">Cancel</button></a>
    </form>
  </section>

  <footer id="footer"></footer>

  <script type="text/javascript" src="js/jquery-3.6.0.min.js"></script>
  <!--Carreguem el footer a la pàgina-->
  <script>
    $(document).ready(function () {
      $("#footer").load("footer.html");
    });
  </script>
</body>
</html>

```

Annexa 2 – Pàgines CSS

main.css

```

html,
body {
  position: relative;
  height: 100%;
  margin: 0;
  padding: 0;
  font-family: 'Roboto', sans-serif;

```

```

    color: #333;
    font-size: 14px;
    letter-spacing: 0.5px;
    word-wrap: break-word;
}

button,
ul,
li {
    all: unset;
}

* {
    -moz-box-sizing: border-box;
    -webkit-box-sizing: border-box;
    box-sizing: border-box;
}

:root {
    --main-color: #B9C7A9;
    --secondary-color: #42473D;
    --light-color: #e4e9de;
    --orange-color: #d1a147;
    --red-color: #da3434;
}

div.loading,
div.loading2 {
    position: absolute;
    width: 100%;
    height: 100%;
    top: 0;
    left: 0;
    display: flex;
    justify-content: center;
    align-items: center;
}

div.loading {
    visibility: hidden;
}

div.loading img,
div.loading2 img {
    position: relative;
    width: 75px;
    z-index: 999999;
}

```

```

header {
  position: relative;
  width: 100%;
  padding: 25px 5px 20px 5px;
  background-color: var(--main-color);
  display: flex;
  flex-direction: column;
  flex-wrap: wrap;
  align-items: center;
  justify-content: center;
}

header img#logo {
  position: relative;
  width: 150px;
}

nav {
  position: relative;
  width: 100%;
}

nav ul#menu {
  position: relative;
  width: 100%;
  margin-top: 30px;
  color: var(--secondary-color);
  font-size: 16px;
  font-weight: bold;
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  justify-content: center;
  align-items: flex-start;
}

nav ul#menu li {
  position: relative;
  padding: 5px 10px;
  margin-left: 10px;
}

nav ul#menu li a {
  text-decoration: none;
  color: var(--secondary-color);
}

nav ul#menu li:hover {
  cursor: pointer;
}

```

```

    opacity: 0.8;
  }

  nav ul#menu li:first-child {
    margin-left: 0;
  }

  nav ul#menu li.login-display {
    display: none;
  }

  section#content {
    position: relative;
    width: 100%;
    display: flex;
    flex-direction: column;
    flex-wrap: wrap;
    justify-content: flex-start;
    align-items: center;
  }

  #about-content {
    position: relative;
    width: 100%;
    max-width: 576PX;
    padding: 10px 50px;
    margin-top: 25px;
    font-size: 16px;
    line-height: 24px;
    display: flex;
    flex-direction: column;
    justify-content: flex-start;
    align-items: center;
  }

  #about-content h1,
  #about-content h2 {
    color: var(--secondary-color);
  }

  #about-content h1 {
    text-align: center;
  }

  #about-content h2 {
    width: 100%;
    text-align: left;
    font-size: 1.2em;
  }

```



```
#about-content strong {
  color: var(--secondary-color);
}

#about-content button {
  margin-top: 25px;
}

#about-content a {
  text-decoration: none;
}

#about-signup {
  position: relative;
  width: 100%;
  padding: 20px 20px 35px 20px;
  margin-top: 35px;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: flex-start;
  background-color: var(--light-color);
}

#about-signup button {
  margin-top: inherit;
  color: white;
  background-color: var(--secondary-color);
}

#map {
  position: relative;
  width: 100%;
  height: 400px;
}

#main-buttons {
  position: relative;
  width: 100%;
  text-align: center;
}

button:hover {
  cursor: pointer;
}

button.button {
  padding: 10px 20px;
```

```

border-radius: 5%;
font-weight: bold;
color: var(--secondary-color);
background-color: var(--main-color);
}

button.button:hover {
  opacity: 0.9;
}

button#forecast-button,
button#outfit-button {
  margin: 10px;
}

button#outfit-button {
  color: white;
  background-color: var(--secondary-color);
}

button#reset-points-button {
  color: white;
  background-color: var(--orange-color);
}

#reset-points-container {
  position: relative;
  width: 100%;
  margin-top: 50px;
  text-align: center;
  display: none;
}

#forecast-results {
  position: relative;
  width: 100%;
  padding: 5px;
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  justify-content: space-between;
  align-items: flex-start;
}

#forecast-results h2 {
  width: 100%;
  min-height: 50px;
  text-align: center;
  display: flex;

```

```

flex-direction: column;
flex-wrap: wrap;
align-items: center;
justify-content: center;
}

article.point,
article.danger {
  position: relative;
  width: calc(50% - 2px);
  min-height: 225px;
  padding: 0 10px;
  margin-top: 4px;
  background-color: var(--light-color);
  display: flex;
  flex-direction: column;
  flex-wrap: wrap;
  justify-content: center;
}

article.danger {
  background-color: var(--red-color);
  color: white;
}

.weather-info {
  position: relative;
  width: 100%;
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  justify-content: center;
  align-items: flex-start;
}

.weather-info div {
  position: relative;
  width: 50%;
  padding: 0 10px;
  border-left: 1px solid var(--secondary-color);
  font-size: 12px;
  text-align: center;
}

.weather-info div:first-child {
  border-left: 0;
}

```

```

h2#sweet-title {
  margin-top: 50px;
  margin-bottom: 50px;
}

h2.type-title,
h2.no-necessary-piece,
h2.missing-piece {
  margin-bottom: 10px;
  font-size: 1.2em;
}

h2.type-title {
  margin-top: 75px;
}

h2.missing-piece {
  color: var(--red-color);
}

h2.no-necessary-piece {
  color: var(--orange-color);
}

ul.search-results {
  position: relative;
  width: 100%;
  margin-top: 15px;
  text-align: center;
  display: flex;
  flex-direction: row;
  flex-wrap: wrap;
  justify-content: center;
  align-items: center;
}

ul.search-results li {
  padding: 5px;
  margin: 0;
  text-align: center;
  border-bottom: 1px solid var(--light-color);
  display: inline-block;
}

ul.search-results li.piece-name {
  width: 200px;
  text-align: left;
  background-color: white;
}

```

```

ul.search-results li.piece-min-temp,
ul.search-results li.piece-max-temp {
    width: 25px;
    border-right: 1px solid var(--secondary-color);
    background-color: var(--light-color);
}

ul.search-results li.piece-max-temp {
    border: none;
}

#content a#new-search {
    margin-top: 75px;
    text-decoration: none !important;
}

#content a#new-search button {
    color: white;
    background-color: var(--secondary-color);
}

footer#footer {
    position: relative;
    width: 100%;
    padding: 50px 25px 20px 25px;
    margin-top: 200px;
    color: white;
    font-size: 12px;
    text-align: center;
    background-color: var(--secondary-color);
    display: flex;
    flex-direction: column;
    flex-wrap: wrap;
    justify-content: flex-start;
    align-items: center;
}

footer#footer img {
    position: relative;
    width: 75px;
    opacity: 0.5;
}

#footer-conditions {
    position: relative;
    width: 100%;
    display: flex;
    flex-direction: column;
}

```

```

    flex-wrap: wrap;
    justify-content: flex-start;
    align-items: center;
}

#footer-conditions div {
    position: relative;
    width: 100%;
    margin-top: 25px;
    display: flex;
    flex-direction: column;
    flex-wrap: wrap;
    justify-content: flex-start;
    align-items: center;
}

#footer-conditions div span {
    margin-top: 15px;
    text-decoration: underline;
}

#footer-conditions div span:hover {
    cursor: pointer;
    opacity: 0.8;
}

#footer-conditions div span:first-child {
    margin-top: 0;
}

#footer-conditions p {
    margin-top: 50px;
}

@media (min-width: 576px) {

    #forecast-results {
        width: 500px;
        padding: 0;
    }
}

@media (min-width: 768px) {

    #forecast-results {
        width: 700px;
    }
}

```

```

article.point,
article.danger {
  min-height: 300px;
}

article.point h2,
article.danger h2 {
  margin-top: 0;
}

.weather-info div {
  font-size: 14px;
}

ul.search-results li {
  padding: 10px 20px;
}

footer#footer img {
  width: 100px;
}

#footer-conditions {
  flex-direction: row;
  justify-content: center;
  align-items: flex-start;
}

#footer-conditions div {
  flex-direction: row;
  justify-content: center;
}

#footer-conditions div span {
  margin: 15px;
}

#footer-conditions div span:first-child {
  margin-top: 15px;
}

#footer-conditions p {
  margin-top: 15px;
}
}

@media (min-width: 1024px) {

```

```

header {
  padding: 50px 20px 20px 20px;
}

header img#logo {
  width: 175px;
}

nav ul#menu {
  margin-top: 0;
  justify-content: flex-end;
}

footer#footer {
  flex-direction: row;
  justify-content: flex-start;
}

footer#footer img {
  margin-left: 66px;
}

#footer-conditions {
  margin-top: 25px;
  justify-content: space-between;
}

#footer-conditions div {
  width: 500px;
  margin-top: 0;
  justify-content: flex-end;
}

#footer-conditions p {
  order: -1;
}
}

@media (min-width: 1200px) {

  #map {
    height: 500px;
  }

  #forecast-results {
    width: 1096px;
    padding: 0 20px;
    justify-content: flex-start;
  }
}

```



```

#forecast-results > h2 {
  border-bottom: 4px solid var(--secondary-color);
  align-items: flex-start;
}

article.point,
article.danger {
  width: 324px;
  padding: 0 10px;
  margin: 14px;
}
}

@media (min-width: 1920px) {

  header,
  footer#footer {
    padding: 50px 100px 20px 100px;
  }

  #forecast-results {
    width: 1448px;
  }
}

```

modal.css

```

.modal {
  position: fixed;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  background-color: var(--secondary-color);
  display: flex;
  align-items: center;
  justify-content: center;
  visibility: hidden;
  opacity: 0;
  transition: all 0.35s ease-in;
  z-index: 100;
}

.modal-dialog {
  position: relative;
  width: 90%;
  max-width: 600px;
}

```

```

background-color: white;
display: flex;
flex-direction: column;
align-items: flex-start;
justify-content: center;
overflow: auto;
}

.modal-header,
.modal-footer {
  position: relative;
  width: 100%;
  padding: 10px 35px;
  background-color: var(--light-color);
  display: flex;
  align-items: center;
}

.modal-header {
  flex-direction: row;
  justify-content: space-between;
}

.close-modal {
  font-size: 24px;
}

.modal-content {
  position: relative;
  width: 100%;
  padding: 50px;
  display: flex;
  flex-direction: column;
  align-items: flex-start;
  justify-content: center;
}

#signin-form,
#delete-form {
  position: relative;
  width: 100%;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
}

#signin-form label,
#delete-form label {

```

```

    position: relative;
    font-weight: bold;
    margin-bottom: 10px;
}

#signin-form input,
#delete-form input {
    position: relative;
    width: 100%;
    max-width: 250px;
    padding: 5px 10px;
    margin-bottom: 25px;
}

#delete-form button.button {
    background-color: var(--red-color);
    color: white;
}

.modal-footer {
    flex-direction: column;
    justify-content: center;
    text-align: center;
}

.modal.is-visible {
    visibility: visible;
    opacity: 1;
}

```

profile.css

```

body {
    background-color: var(--main-color);
}

.profile-content {
    position: relative;
    width: 100%;
}

.profile-content h2 {
    text-align: center;
}

#goback {
    position: relative;
    width: 100%;
}

```

```
display: flex;
flex-direction: column;
justify-content: center;
align-items: center;
}

#goback button {
  color: white;
  background-color: var(--orange-color);
}

#goback a {
  text-decoration: none;
}

#goback a button {
  margin-top: 25px;
  color: var(--secondary-color);
  background-color: var(--light-color);
}

#add-inventory {
  padding: 10px 20px 50px 10px;
  background-color: var(--light-color);
}

#add-inventory,
#inventory {
  position: relative;
  width: 100%;
  margin-top: 50px;
  display: flex;
  flex-direction: column;
  flex-wrap: wrap;
  justify-content: flex-start;
  align-items: center;
}

#inventory {
  width: 350px;
  padding: 0 20px;
}

#add-piece-form {
  position: relative;
  width: 100%;
  display: flex;
  flex-direction: column;
```

```

    align-items: center;
    justify-content: center;
}

#add-piece-form label {
    position: relative;
    font-weight: bold;
    margin-bottom: 10px;
}

#add-piece-form input,
#add-piece-form select {
    position: relative;
    width: 100%;
    max-width: 250px;
    padding: 5px 10px;
    margin-bottom: 25px;
}

#add-piece-form button {
    color: white;
    background-color: var(--secondary-color);
}

#inventory h2 {
    color: white;
}

#inventory ul {
    position: relative;
    width: 100%;
    margin-top: 15px;
    text-align: center;
    display: flex;
    flex-direction: row;
    flex-wrap: wrap;
    justify-content: center;
    align-items: center;
}

#inventory ul li {
    padding: 5px;
    margin: 0;
    text-align: center;
    display: inline-block;
}

#inventory ul li.piece-type {
    width: 75px;
}

```

```

    background-color: var(--light-color);
}

#inventory ul li.piece-name {
    width: 200px;
    text-align: left;
    background-color: white;
}

#inventory ul li.piece-min-temp,
#inventory ul li.piece-max-temp {
    width: 25px;
    border-right: 1px solid var(--secondary-color);
    background-color: var(--light-color);
}

#inventory ul li.piece-max-temp {
    border: none;
}

#inventory ul li button {
    margin-left: 10px;
    padding: 5px;
    color: white;
    background-color: var(--red-color);
}

#delete-account {
    position: relative;
    width: 100%;
    margin-top: 150px;
    text-align: center;
    background-color: var(--light-color);
}

#delete-account button {
    color: white;
    background-color: var(--red-color);
}

@media (min-width: 576px) {

    #inventory {
        width: 100%;
    }

    #inventory ul {

```

```
        margin-top: 5px;
    }
}

@media (min-width: 768px) {

    #inventory ul li,
    #inventory ul li button {
        padding: 10px 20px;
    }
}
```

signup.css

```
body {
    background-color: var(--main-color);
}

.signup-content {
    position: relative;
    width: 100%;
    padding: 0 20px;
    margin-top: 50px;
}

#signup-form,
#signup-form fieldset {
    position: relative;
    width: 100%;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
}

#signup-form fieldset {
    max-width: 350px;
    padding: 20px;
    margin-bottom: 25px;
}

#signup-form label {
    position: relative;
    font-weight: bold;
    margin-bottom: 10px;
}
```

```

#signup-form input {
  position: relative;
  width: 100%;
  max-width: 250px;
  padding: 5px 10px;
  border: none;
  margin-bottom: 25px;
}

#signup-form button.button {
  min-width: 100px;
  text-align: center;
  color: white;
  background-color: var(--secondary-color);
}

#signup-form button.button:last-child {
  margin-top: 25px;
  background-color: var(--orange-color);
}

#signup-form a {
  text-decoration: none;
}

```

Annexa 3 – Pàgines JavaScript

main.js

```

/*
  -----MOSTRAR ICONA DE CÀRREGA-----
*/
const loading = (val) => {

  let el = document.getElementsByClassName("loading");

  if (val == true) {
    el[1].style.visibility = "visible";

  } else {
    el[1].style.visibility = "hidden";
  }
};
/*
  -----
*/

```



```

/*
  -----OBTENIR GEOLOCALITZACIÓ DEL NAVEGADOR-----
*/
// Inicialitzem variables per guardar més endavant les coordenades de posicionament:
var lat;
var lng;
// I més variables per emmagatzemar resultats de les crides a l'API de previsió del
temps:
var temp = new Array();
var wind = new Array();
var weather = new Array();
var place;

// Establim el posicionament inicial del dispositiu a través de la geolocalització
del navegador:
setPosition();
function setPosition() {
  // Si el navegador permet la geolocalització, disparem el mètode
getCurrentPosition():
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(success, error, options);
  // Si el navegador no permet la geolocalització, disparem directament error():
  } else {
    error();
  }
};

var options = {
  EnableHighAccuracy: true, // Habilitem un posicionament acurat.
  Timeout: 3000, // Establim 3 segons com a temps d'espera màxim.
  MaximumAge: 0 // Establim que volem el posicionament actual.
};

// Si tot és correcte i se'ns permet obtenir la geolocalització, es dispara
success():
function success(geolocationPosition) {
  // Obtenim els paràmetres de latitud i longitud de l'objecte geolocationPosition
(coords.latitude i coords.longitude):
  lat = geolocationPosition.coords.latitude;
  lng = geolocationPosition.coords.longitude;

  initMap(lat, lng);
  updateUrlWeatherCall(); // Actualitzem la URL de la petició a Open Weather Map
amb les dades de posicionament donades.
};

```

```

// Si no es pot obtenir la geolocalització del navegador, s'adverteix a l'usuari i
indiquem un posicionament per defecte (en aquest cas, el de la ciutat de Nova York):
function error(err) {
    lat = 40.7142700;
    lng = -74.0059700;

    initMap(lat, lng);
    updateUrlWeatherCall(); // Actualitzem la URL de la petició a Open Weather Map
amb les dades de posicionament donades.
};
/*
-----
*/

/*
CARREGAR MAPA AMB LA GEOLOCALITZACIÓ DELS PUNTS (MARCADORS) SELECCIONATS
*/
function initMap(x, y) {
    let map;
    let location = { lat: x, lng: y };
    let marker;

    if (x !== undefined) { // Esperem a que arribi un valor vàlid abans d'executar la
funció.
        map = new google.maps.Map(document.getElementById("map"), {
            center: { lat: x, lng: y }, // Centrem el mapa en aquesta posició.
            zoom: 14, // Indiquem el zoom del mapa.
        });

        // Afegim el marcador:
        addMarker(location, map);

        // Cada vegada que fem click amb el ratolí sobre el mapa, afegirem un nou
marcador:
        google.maps.event.addListener(map, "click", (event) => {
            marker.setMap(null); // Primer, eliminem el marcador existent.
            addMarker(event.latLng, map);
            // Actualitzem la localització:
            lat = event.latLng.lat();
            lng = event.latLng.lng();
            location.lat = lat;
            location.lng = lng;
            updateUrlWeatherCall();
        });

        // Definim i situem un marcador:

```

```

    function addMarker(location, map) {
        marker = new google.maps.Marker({
            position: location,
            map: map
        });
    };
};
/*
-----
*/

/*
PREVISIÓ DEL TEMPS: OPEN WEATHER API
*/
const forecast_button = document.getElementById("forecast-button");
const apiWeatherKey = "21e97e7de27fce68c74d289d2869e067"; // Clau API necessària per
executar cada petició.
var url;
// Funció per actualitzar el valor de la variable url:
function updateUrlWeatherCall() {
    url =
`https://api.openweathermap.org/data/2.5/forecast?lat=${lat}&lon=${lng}&cnt=2&appid=${
apiWeatherKey}&units=metric`; // URL per accedir a l'API. Conté els paràmetres de
geolocalització (lat, lng) i la clau API.
}
// Funció fletxa per executar la petició AJAX i obtenir la informació de l'API
var be_safe = true; // Aquesta variable la utilitzarem per determinar si la previsió
és segura.
var add_danger = false; // Aquesta variable la utilitzarem per marcar una previsió
insegura.
const getWeatherInfo = async () => {
    loading(true); // Executem la funció que mostra el GIF mentre es carrega la
informació
    try {
        const response = await fetch(url, {cache: "no-cache"}); // Ens assegurem que
no aprofiti res del cache.
        // Si la resposta és correcte:
        if (response.ok) {
            const jsonResponse = await response.json(); // Obtenim la resposta en
format json.
            // A continuació obtenim la temperatura, el valor principal del paràmetre
"weather", i el vent.
            // Com es pot comprovar, agafem un valor inicial (valor actual) i un de
final (valor al cap de 3 hores):
            let initial_temp = jsonResponse.list[0].main.feels_like;

```

```

let final_temp = jsonResponse.list[1].main.feels_like;
let initial_weather = jsonResponse.list[0].weather[0].main;
let final_weather = jsonResponse.list[1].weather[0].main;
let initial_wind = jsonResponse.list[0].wind.gust;
let final_wind = jsonResponse.list[1].wind.gust;
let city = jsonResponse.city.name;
add_danger = false;
// Comprovem si les dades obtingudes són segures per sortir en bici:
checkTemp(initial_temp);
checkTemp(final_temp);
checkWeather(initial_weather);
checkWeather(final_weather);
checkWind(initial_wind);
checkWind(final_wind);
// A continuació insertem les dades del punt marcat per mostrar-lo a la
pàgina:

let results = document.getElementById("forecast-results");
let point = document.createElement("article");
point.className = "point";
let content_point = `

## ${city}</h2><div class="weather- info"><div><h3>Now</h3><p>${initial_temp}&deg</p><p>${initial_wind} m/s</p><p>${initial_weather}</p></div><div><h3>+3h</h3><p>${final_temp}&deg</p><p>${f inal_wind} m/s</p><p>${final_weather}</p></div></div>`; point.insertAdjacentHTML("beforeend", content_point); // Si alguna dada no és segura afegim la classe .danger: if (add_danger) { point.className = "danger"; } // Si encara no hem introduït cap previsió, afegim un element <h2> i el botó per fer reset: if (results.children.length == 1) { results.insertAdjacentHTML("beforeend", "<h2>Added points</h2>"); let reset = document.getElementById("reset-points-container"); reset.style.display = "inline-block"; } // Insertem la informació a la pàgina: results.appendChild(point); // Funcions per determinar si és segur sortir en bici: function checkTemp(x) { // Si la temperatura excedeix els 35 graus: if (x > 35) { be_safe = false; add_danger = true; } // Si la temperatura baixa dels -10 graus: if (x < -10) { be_safe = false; add_danger = true; } }


```

```

    }
    function checkWeather(x) {
        // Tot el que no sigui ennuvolat o clar tindrà risc per l'usuari:
        if (x !== "Clouds" && x !== "Clear") {
            be_safe = false;
            add_danger = true;
        }
    }
    function checkWind(x) {
        // Si la velocitat és superior a 10,8 m/s:
        if (x > 10.8) {
            be_safe = false;
            add_danger = true;
        }
    }
    // Insertem les dades dins de variables per poder-les utilitzar
posteriorment:
    temp.push(initial_temp, final_temp);
    wind.push(initial_wind, final_wind);
    weather.push(initial_weather, final_weather);
    place = city;
}
// Si torna un error el mostrem a la consola:
} catch (error) {
    console.log(error);
}
loading(false);
});
forecast_button.addEventListener("click", getWeatherInfo);

// Funció per fer reset dels punts que s'han anat afegint per fer forecast:
const reset_points_button = document.getElementById("reset-points-button");
function resetPoints() {
    let results = document.getElementById("forecast-results");
    let reset = document.getElementById("reset-points-container");
    reset.style.display = "none";
    results.innerHTML = `<div class="loading"></div>`;
    temp = new Array();
    wind = new Array();
    weather = new Array();
    place = "";
    be_safe = true;
    add_danger = false;
}
reset_points_button.addEventListener("click", resetPoints);
/*
-----
*/

```

```

/*
  OBTENCIÓ DE DADES DE L'USUARI
*/
const outfit_button = document.getElementById("outfit-button");
// Funció per gestionar la informació obtinguda de la previsió del temps i accedir a
la base de dades:
const getOutfitInfo = () => {

  if (temp.length == 0) { // Si no s'ha marcat cap punt al mapa, no podem fer la
petició a la base de dades.
    alert("You need to set at least one point");

  } else {
    // Primer mirarem si hi ha un usuari connectat observant el valor de la
cookie "user":
    let cookie_list = document.cookie.split(";");

    for (let i in cookie_list) {
      let search = cookie_list[i].search("user");
      // Si trobem la cookie, consultem el seu valor":
      if (search > -1) {
        let mycookie = cookie_list[i];
        let index = mycookie.indexOf("=");
        var cookie_value = mycookie.substring(index+1);
      }
    }
    // Si la cookie té valor, executarem la resta d'accions. En cas contrari,
farem l'advertència:
    if (cookie_value == "") {
      alert("You need to be logged!");
    } else {

      let answer = true;
      // Si la previsió del temps no és segura, fem una alerta esperant
confirmació o no de l'usuari:
      if (!be_safe) {
        answer = confirm("Some of the places you marked have bad weather
forecasting. We recommend you stay safe.");
      }

      if (answer) {
        // Agafem els valors màxims i mínims de vent i temperatura, ja que
són els únics que utilitzarem:
        let max_temp = Math.max(...temp);

```

```

    let min_temp = Math.min(...temp);
    let avg_temp = (max_temp + min_temp)/2; // Obtenim la mitja de les
temperatures, que serà la que utilitzarem per filtrar els ítems de la base de dades.
    let request = "";
    // Obtenim el email de l'usuari, que l'utilitzem per fer la cerca a
la BBDD:
    let email = cookie_value;

    // Si la temperatura és superior a 20 graus, es descartaran les
categories que són més per l'hivern: Neck, Overshoes i Cap.
    if (avg_temp < 20) {
        request = [avg_temp, email, 1];
    } else {
        request = [avg_temp, email, 2];
    }
    loading(true); // Executem la funció que mostra el GIF mentre es
carrega la informació
    // Es llença la petició a la base de dades:
$.get("getItems.php", { request }, function(res) {
    // Si la connexió no és possible, es llença un avís:
    if (!res) {
        alert("Ops! We got an error during the process. Try
again...");
    }
    // Si la connexió és correcte:
    } else {
        // Inicialitzem variables i els Arrays que utilitzarem per
anar guardant la informació de cada ítem per tipus.
        let content = document.getElementById("content");
        let finalRes = "<h2 id='sweet-title'>Sweet
recommendation:</h2>";
        let jerseyRes = new Array();
        let bibRes = new Array();
        let layerRes = new Array();
        let glovesRes = new Array();
        let socksRes = new Array();
        let neckRes = new Array();
        let capRes = new Array();
        let shoesRes = new Array();
        let otherRes = new Array();
        // La primera posició de cada Array serà el títol del tipus
de peça:
        jerseyRes.push("<h2 class='type-title'>Jersey</h2>");
        bibRes.push("<h2 class='type-title'>Bib</h2>");
        layerRes.push("<h2 class='type-title'>Layer</h2>");
        glovesRes.push("<h2 class='type-title'>Gloves</h2>");
        socksRes.push("<h2 class='type-title'>Socks</h2>");
        neckRes.push("<h2 class='type-title'>Neck protection</h2>");
        capRes.push("<h2 class='type-title'>Cap</h2>");
        shoesRes.push("<h2 class='type-title'>Overshoes</h2>");

```

```

        otherRes.push("<h2 class='type-title'>Other</h2>");
        jsonRes = JSON.parse(res); // Obtenim la resposta de la
consulta en format JSON.
        // Recorrem la resposta de la consulta i anem guardant la
informació de cada ítem a l'Array corresponent:
        for (let item of jsonRes) {
            if (item.dif < 15) {
                switch (item.type) {
                    case "Jersey":
                        jerseyRes.push(`<ul class="search-
results"><li class="piece-name">${item.name}</li><li class="piece-min-
temp">${item.mintemp}&deg</li><li class="piece-max-
temp">${item.maxtemp}&deg</li></ul>`);
                        break;
                    case "Bib":
                        bibRes.push(`<ul class="search-results"><li
class="piece-name">${item.name}</li><li class="piece-min-
temp">${item.mintemp}&deg</li><li class="piece-max-
temp">${item.maxtemp}&deg</li></ul>`);
                        break;
                    case "Layer":
                        layerRes.push(`<ul class="search-results"><li
class="piece-name">${item.name}</li><li class="piece-min-
temp">${item.mintemp}&deg</li><li class="piece-max-
temp">${item.maxtemp}&deg</li></ul>`);
                        break;
                    case "Gloves":
                        glovesRes.push(`<ul class="search-
results"><li class="piece-name">${item.name}</li><li class="piece-min-
temp">${item.mintemp}&deg</li><li class="piece-max-
temp">${item.maxtemp}&deg</li></ul>`);
                        break;
                    case "Socks":
                        socksRes.push(`<ul class="search-results"><li
class="piece-name">${item.name}</li><li class="piece-min-
temp">${item.mintemp}&deg</li><li class="piece-max-
temp">${item.maxtemp}&deg</li></ul>`);
                        break;
                    case "Neck":
                        if (avg_temp < 20) {
                            neckRes.push(`<ul class="search-
results"><li class="piece-name">${item.name}</li><li class="piece-min-
temp">${item.mintemp}&deg</li><li class="piece-max-
temp">${item.maxtemp}&deg</li></ul>`);
                        }
                        break;
                    case "Overshoes":
                        if (avg_temp < 20) {

```



```

        shoesRes.push(`<ul class="search-
results"><li class="piece-name">${item.name}</li><li class="piece-min-
temp">${item.mintemp}&deg</li><li class="piece-max-
temp">${item.maxtemp}&deg</li></ul>`);
    }
    break;
    case "Cap":
        if (avg_temp < 20) {
            capRes.push(`<ul class="search-
results"><li class="piece-name">${item.name}</li><li class="piece-min-
temp">${item.mintemp}&deg</li><li class="piece-max-
temp">${item.maxtemp}&deg</li></ul>`);
        }
        break;
    default:
        otherRes.push(`<ul class="search-results"><li
class="piece-name">${item.name}</li><li class="piece-min-
temp">${item.mintemp}&deg</li><li class="piece-max-
temp">${item.maxtemp}&deg</li></ul>`);
        break;
    }
}
}
}
// A continuació comprovem, per a cada tipus de peça, si hem
obtingut resultats.
// Si no hem obtingut cap resultat, afegim l'avís i a
continuació afegim aquests resultats a la variable finalRes.
// A finalRes anem guardant tots els resultats obtinguts.
if (jerseyRes.length == 1) {
    jerseyRes.push("<h2 class='missing-piece'>No matched
jersey found</h2>");
}
for (let piece of jerseyRes) {
    finalRes += piece;
}
if (bibRes.length == 1) {
    bibRes.push("<h2 class='missing-piece'>No matched bib
found</h2>");
}
for (let piece of bibRes) {
    finalRes += piece;
}
if (layerRes.length == 1) {
    layerRes.push("<h2 class='missing-piece'>No matched layer
found</h2>");
}
for (let piece of layerRes) {
    finalRes += piece;
}
}
}
}

```

```

        if (glovesRes.length == 1) {
            glovesRes.push("<h2 class='missing-piece'>No matched
gloves found</h2>");
        }
        for (let piece of glovesRes) {
            finalRes += piece;
        }
        if (socksRes.length == 1) {
            socksRes.push("<h2 class='missing-piece'>No matched socks
found</h2>");
        }
        for (let piece of socksRes) {
            finalRes += piece;
        }
        // En el cas de les categories Neck, Overshoes i Cap,
diferenciem si és necessari mostrar cap resultat segons
// si la temperatura és superior o no a 20 graus.
        if (avg_temp < 20) {
            if (neckRes.length == 1) {
                neckRes.push("<h2 class='missing-piece'>No matched
neck protection found</h2>");
            }
            if (shoesRes.length == 1) {
                shoesRes.push("<h2 class='missing-piece'>No matched
overshoes found</h2>");
            }
            if (capRes.length == 1) {
                capRes.push("<h2 class='missing-piece'>No matched cap
found</h2>");
            }
        } else {
            neckRes.push("<h2 class='no-necessary-piece'>The neck
protection is not necessary</h2>");
            shoesRes.push("<h2 class='no-necessary-piece'>The
overshoes are not necessary</h2>");
            capRes.push("<h2 class='no-necessary-piece'>The cap is
not necessary</h2>");
        }
        for (let piece of neckRes) {
            finalRes += piece;
        }
        for (let piece of shoesRes) {
            finalRes += piece;
        }
        for (let piece of capRes) {
            finalRes += piece;
        }
        if (otherRes.length == 1) {

```

```

        otherRes.push("<h2 class='no-necessary-piece'>No matched
pieces found</h2>");
    }
    for (let piece of otherRes) {
        finalRes += piece;
    }
    // Insertem els resultats a "content" i, per si de cas, fem
scroll per assegurar que tornem a estar a dalt de tot de la pàgina.
    finalRes += '<a href="index.html" id="new-search"
target="_self"><button type="button" class="button">New Search</button></a>';
    content.innerHTML = `<div class="loading"></div>` + finalRes;
    window.scrollTo(0, 0);
    }
    });
    loading(false);
    }
    }
    }
};
outfit_button.addEventListener("click", getOutfitInfo);
/*
-----
*/

/*
CÀRREGA DE LA PÀGINA ABOUT US
*/
const about_button = document.getElementById("about-button");
function insertAboutPage() {
    // En aquest cas utilitzarem la funció load() de la llibreria de jQuery, ja que
ens facilita molt la feina en aquest cas:
    $("#content").load('aboutus.html');
}
about_button.addEventListener("click", insertAboutPage);
/*
-----
*/

```

modal.js

```

const openEls = document.querySelectorAll("[data-open]");
const closeEls = document.querySelectorAll("[data-close]");
const isVisible = "is-visible";

```

```

// Obrim el modal de sign in:
for (const el of openEls) {
  el.addEventListener("click", function() {
    const modalId = this.dataset.open;
    document.getElementById(modalId).classList.add(isVisible);
  });
}

// Tanquem el modal si cliquem la X del header del modal:
for (const el of closeEls) {
  el.addEventListener("click", function() {
    this.parentElement.parentElement.parentElement.classList.remove(isVisible);
  });
}

// Si cliquem fora del modal de sign in, el tanquem:
document.addEventListener("click", e => {
  if (e.target == document.querySelector(".modal.is-visible")) {
    document.querySelector(".modal.is-visible").classList.remove(isVisible);
  }
});

// Si cliquem la tecla Escape, tanquem el modal de sign in:
document.addEventListener("keyup", e => {
  if (e.key == "Escape" && document.querySelector(".modal.is-visible")) {
    document.querySelector(".modal.is-visible").classList.remove(isVisible);
  }
});

```

profile.js

```

/*
  CÀRREGA, INSERCIÓ I ELIMINACIÓ D'ÍTEMS DE L'INVENTARI DE L'USUARI
*/

/*
  -----MOSTRAR ICONA DE CÀRREGA-----
*/
const loading = (val) => {

  let el = document.getElementsByClassName("loading");

  if (val == true) {
    el[0].style.visibility = "visible";
  } else {
    el[0].style.visibility = "hidden";
  }
}

```

```

    }
};
/*
-----
*/

// Funció per carregar l'inventari disponible de l'usuari:
function getItems() {
    // Primer, insertem el títol amb la direcció corresponent:
    let content = document.getElementById("content");
    let cookie_list = document.cookie.split(";");
    for (let i in cookie_list) {
        let search = cookie_list[i].search("user");
        // Si trobem la cookie, consultem el seu valor":
        if (search > -1) {
            let mycookie = cookie_list[i]
            let index = mycookie.indexOf("=");
            var cookie_value = mycookie.substring(index+1);
        }
    }
    let email = cookie_value;
    content.insertAdjacentHTML("afterbegin", "<h2>Welcome to your dashboard, </br>" +
email + "</h2>");

    let request = email;

    loading(true); // Executem la funció que mostra el GIF mentre es carrega la
informació

$.get("getItems2.php", { request }, function(res) {
    if (!res) {
        alert("Ops! We got an error during the process. Try again...");
    } else {
        let inventory = document.getElementById("inventory");
        let finalRes = "";
        jsonRes = JSON.parse(res);

        for (let item of jsonRes) {
            finalRes += `<ul><li data-id="${item.id}" class="piece-
type">${item.type}</li><li class="piece-name">${item.name}</li><li class="piece-min-
temp">${item.mintemp}&deg</li><li class="piece-max-temp">${item.maxtemp}&deg</li><li
class="delete-piece"><button type="button" class="button delete-
button">Delete</button></li></ul>`;
        }

        inventory.insertAdjacentHTML("beforeend", finalRes);
    }
}

```

```

    }
  });
  loading(false);
}

// Funcions per insertar un ítem a l'inventari de l'usuari:
const insertForm = document.getElementById("add-piece-form");
insertForm.addEventListener("submit", e => {
  e.preventDefault();
  // Aquí només cerquem la direcció email de l'usuari que s'ha guardat a les
  cookies:
  let cookie_list = document.cookie.split(";");

  for (let i in cookie_list) {
    let search = cookie_list[i].search("user");
    // Si trobem la cookie, consultem el seu valor:
    if (search > -1) {
      let mycookie = cookie_list[i]
      let index = mycookie.indexOf("=");
      var cookie_value = mycookie.substring(index+1);
    }
  }

  let email = cookie_value;
  let type = document.getElementById("select-type").value;
  let name = document.getElementById("item-name").value;
  let min_temp = document.getElementById("item-mintemp").value;
  let max_temp = document.getElementById("item-maxtemp").value;
  let request = [email, type, name, min_temp, max_temp];

  loading(true); // Executem la funció que mostra el GIF mentre es carrega la
  informació

  $.post("editItem.php", { request }, function(res) {

    if (!res) {
      alert("Ops! We got an error during the process. Try again...");
    } else {
      alert("Item added to your inventory!");
      location.reload();
    }
  });
  loading(false);
});

// Funcions per eliminar un ítem a l'inventari de l'usuari:
document.addEventListener("click", (e) => {
  let clickedItem = e.target;

```

```

if (!clickedItem.matches(".delete-button")) {
    return;
};

clickedItem = clickedItem.parentNode.parentNode.firstChild;
let item_id = clickedItem.getAttribute("data-id");
let item_name = clickedItem.nextSibling.outerText;
let request = [item_id, 2];
let answer = confirm(`Do you want to delete ${item_name} from your inventory?`);

if (answer) {
    loading(true); // Executem la funció que mostra el GIF mentre es carrega la
informació
    $.post("editItem.php", { request }, function(res) {

        if (!res) {
            alert("Ops! We got an error during the process. Try again..");

        } else {
            alert("Item deleted from your inventory!");
            location.reload();
        }
    });
    loading(false);
}
});
/*
-----
*/

/*
S'EXECUTA L'ELIMINACIÓ D'USUARI
*/
const deleteAccount = document.getElementById("delete-account-button");
deleteAccount.addEventListener("click", () => {
    // Creem la cookie que recollirem després per validar l'eliminació del compte:
    document.cookie = "deleteAccount=true";
    // Anem a la pàgina principal:
    let go_back = document.getElementById("go-back-button");
    go_back.click();
});
/*
-----
*/

```

firebase.js

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "https://www.gstatic.com/firebasejs/9.18.0/firebase-app.js";
// Obtenim de Firebase el mòdul getAuth, que necessitem per autenticar usuaris:
import { getAuth } from "https://www.gstatic.com/firebasejs/9.18.0/firebase-auth.js";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyDZpTi0j-TBTfDw5F7ZEs0UBF2ujB4fiDc",
  authDomain: "ride-sweetly-auth.firebaseio.com",
  projectId: "ride-sweetly-auth",
  storageBucket: "ride-sweetly-auth.appspot.com",
  messagingSenderId: "858727290148",
  appId: "1:858727290148:web:8d150eab2cbbb525838fb1"
};

// Initialize Firebase
export const app = initializeApp(firebaseConfig);
export const auth = getAuth(app);
```

auth-signup.js

```
// Importem la funció necessària per crear i autenticar usuaris:
import { createUserWithEmailAndPassword, signInWithEmailAndPassword, signOut } from
"https://www.gstatic.com/firebasejs/9.18.0/firebase-auth.js";
import { auth } from "./firebase.js"; // Importem l'objecte auth de firebase.js

/*
  -----MOSTRAR ICONA DE CÀRREGA-----
*/
const loading = (val) => {

  let el = document.getElementsByClassName("loading");

  if (val == true) {
    el[0].style.visibility = "visible";
  } else {
    el[0].style.visibility = "hidden";
  }
};
/*
```



```

-----
*/

/*
    REGISTRE DE NOUS USUARIS
*/
const signupForm = document.getElementById("signup-form"); // Referenciem el
formulari de registre d'usuari

signupForm.addEventListener("submit", async (e) => {
    e.preventDefault(); // Evitem el funcionament per defecte de l'esdeveniment
    // Obtenim els valors dels camps del formulari:
    const email = signupForm["email"].value;
    const pass = signupForm["pass"].value;
    loading(true); // Executem la funció que mostra el GIF mentre es carrega la
informació
    try {
        await createUserWithEmailAndPassword(auth, email, pass);
        alert("New user added successfully");
        // Tancarem la sessió iniciada i redireccionarem a la pàgina inicial:
        signOut(auth);
        let go_back = document.getElementById("cancel-button");
        go_back.click(); // Tornem a la pàgina inicial

    } catch (error) {
        console.log(error);
        // Si dona error, gestionarem els missatge segons el codi d'aquest:
        switch (error.code) {
            case "auth/email-already-in-use":
                alert("Email already in use");
                signupForm["email"].focus();
                break;
            case "auth/invalid-email":
                alert("Invalid email");
                signupForm["email"].focus();
                break;
            case "auth/weak-password":
                alert("Weak password");
                signupForm["pass"].focus();
                break;
            default:
                alert("Something went wrong: " + error.code);
        }
    }
    loading(false);
}

```

```
});  
/*  
-----  
*/
```

auth-signin.js

```
import { auth } from "../firebase.js"; // Importem l'objecte auth de firebase.js  
import { onAuthStateChanged, signInWithEmailAndPassword, signOut, deleteUser,  
reauthenticateWithCredential, EmailAuthProvider } from  
"https://www.gstatic.com/firebasejs/9.18.0/firebase-auth.js"; // Importem la funcions  
de Firebase  
  
/*  
-----MOSTRAR ICONA DE CÀRREGA-----  
*/  
const loading = (val) => {  
  
  let el = document.getElementsByClassName("loading");  
  
  if (val == true) {  
    el[0].style.visibility = "visible";  
  
  } else {  
    el[0].style.visibility = "hidden";  
  }  
};  
/*  
-----  
*/  
  
/*  
  INICI DE SESSIÓ  
*/  
const signinForm = document.getElementById("signin-form"); // Referenciem el  
formulari d'inici de sessió  
  
signinForm.addEventListener("submit", async (e) => {  
  e.preventDefault(); // Evitem el funcionament per defecte de l'esdeveniment  
  // Obtenim els valors dels camps del formulari:  
  const email = signinForm["email"].value;  
  const pass = signinForm["pass"].value;
```

```

try {
  loading(true); // Executem la funció que mostra el GIF mentre es carrega la
informació
  await signInWithEmailAndPassword(auth, email, pass);
  alert("Session successfully started");
  // Borrem els valors dels camps del formulari
  signInForm["email"].value = "";
  signInForm["pass"].value = "";
  // Tanquem la finestra:
  let go_back = document.querySelectorAll("[data-close]");
  go_back[0].click();

} catch (error) {
  console.log(error);
  // Si dona error, gestionarem els missatge segons el codi d'aquest:
  switch (error.code) {
    case "auth/wrong-password":
      alert("Wrong password");
      signInForm["pass"].focus();
      break;
    case "auth/user-not-found":
      alert("User not found");
      signInForm["email"].focus();
      break;
    default:
      alert("Something went wrong: " + error.code);
  }
}
loading(false);
});

/*
  TANCAMENT DE SESSIÓ
*/
const logout = document.getElementById("logout-session");
logout.addEventListener("click", async () => {
  loading(true); // Executem la funció que mostra el GIF mentre es carrega la
informació
  await signOut(auth);
  alert("Session closed");
  loading(false);
});
/*
  -----
*/

```

```

/*
    SESSIÓ INICIADA
*/
// Si tenim una sessió iniciada, mostrarem el botó de Log out i el email de la
sessió.
// També tenim una part del codi que només serveix per anar comprovant el valor de la
cookie "deleteAccount".
onAuthStateChanged(auth, (user) => {
    let session_name = document.getElementById("session-owner");
    let show = document.getElementsByClassName("login-display");
    let hide = document.getElementsByClassName("logout-display");

    if (user) {

        for (let item of show) {
            item.style.display = "block";
        }
        for (let item of hide) {
            item.style.display = "none";
        }
        session_name.innerText = user.email;
        // Guardem el email en una cookie per utilitzar-la després:
        document.cookie = "user= " + user.email;

        // CODI PER ACABAR D'ELIMINAR UN COMPTE D'USUARI
        // Fem un Array de l'String que guarda la informació de cookies:
        let cookie_list = document.cookie.split(";");
        // Cerquem el nom de la cookie que ens interessa dins l'Array.
        // En aquest cas ens interessa saber si la cookie de nom deleteAccount té
valor "true" (volem eliminar l'usuari):
        for (let i in cookie_list) {
            let search = cookie_list[i].search("deleteAccount");
            // Si trobem la cookie, consultem que el seu valor sigui "true":
            if (search > -1) {
                let mycookie = cookie_list[i];
                let index = mycookie.indexOf("=");
                let cookie_value = mycookie.substring(index+1);

                if (cookie_value == "true") {
                    // Ens assegurem que realment es vol eliminar el compte:
                    let answer = confirm(`Do you want to delete your account? This
action can not be undone...`);
                    // Si ha confirmat la petició:
                    if (answer) {
                        // Fem visible el formulari perquè confirmi l'eliminació del
compte:
                        document.getElementById("delete-
modal").classList.add(isVisible);

```

```

        } else {
            // Sempre que hi hagi un error, farem "reset" a la cookie
deleteAccount...
            document.cookie = "deleteAccount=; max-age:-1";
        }
    }
}

} else {

    for (let item of show) {
        item.style.display = "none";
    }
    for (let item of hide) {
        item.style.display = "block";
    }
    // Borrem la cookie:
    document.cookie = "user=; max-age:-1";
    document.cookie = "deleteAccount=; max-age:-1";
}
});
/*
-----
*/

/*
EXECUTEM L'ELIMINACIÓ D'USUARI
*/
const deleteForm = document.getElementById("delete-form"); // Aquí estem referenciant
al formulari per confirmar l'eliminació d'un usuari
// Amb "deleteForm" executem l'eliminació d'un usuari
deleteForm.addEventListener("submit", async (e) => {
    e.preventDefault(); // Evitem el funcionament per defecte de l'esdeveniment
    // Obtenim els valors dels camps del formulari:
    const email = deleteForm["email2"].value;
    const pass = deleteForm["pass2"].value;
    loading(true); // Executem la funció que mostra el GIF mentre es carrega la
informació
    try {
        await signInWithEmailAndPassword(auth, email, pass);
        // Borrem els valors dels camps del formulari
        deleteForm["email2"].value = "";
        deleteForm["pass2"].value = "";
        // Tanquem la finestra:
        let go_back = document.querySelectorAll("[data-close]");
        go_back[1].click();

```

```

// Preparem la petició SQL per als items de la base de dades:
let request = [email, 3];

onAuthStateChanged(auth, (user) => {

  try {
    // Validem l'usuari amb les funcions importades des de Firebase:
    const credential = EmailAuthProvider.credential(email, pass);

    reauthenticateWithCredential(user, credential).then(() => {

      deleteUser(user).then(() => {
        // Eliminem els items de la BBDD vinculats al email de
l'usuari eliminat:
        $.post("editItem.php", { request }, function(res) {

          if (!res) {
            alert("Ops! We got an error during the process. Try
again...");

          } else {
            alert("User deleted");
          }
        });

        // Fem reset al valor de les cookies:
        document.cookie = "deleteAccount=; max-age:-1";
        document.cookie = "user=; max-age:-1";

      }).catch((error) => {
        alert("An error occurred... " + error);
        // Sempre que hi hagi un error, farem "reset" a la cookie
deleteAccount...
        document.cookie = "deleteAccount=; max-age:-1";
      });
    });

  } catch (error) {
    console.log(error);
    alert("An error occurred... " + error);
    // Sempre que hi hagi un error, farem "reset" a la cookie
deleteAccount...
    document.cookie = "deleteAccount=; max-age:-1";
  }
});

} catch (error) {
  console.log(error);
}

```

```

// Si dona error, gestionarem els missatge segons el codi d'aquest:
switch (error.code) {
    case "auth/wrong-password":
        alert("Wrong password");
        deleteForm["pass"].focus();
        break;
    case "auth/user-not-found":
        alert("User not found");
        deleteForm["email"].focus();
        break;
    default:
        alert("Something went wrong: " + error.code);
}
}
loading(false);
});
/*
-----
*/

```

Annexa 4 – Pàgines PHP

functions.php

```

<?php header("Content-Type: text/html;charset=utf-8");

function connectToBBDD() {
    //Paràmetres de connexió
    $server = "bbdd.josepvallejo.com";
    $user = "ddb202062";
    $pass = "Qu7ZS,I87wf{";
    $bd = "ddb202062";
    //Realització de la connexió
    $con = mysqli_connect ($server, $user, $pass, $bd);

    return $con;
}

function disconnectToBBDD($res, $con) {

    mysqli_free_result($res); //Alliberem explícitament els recursos utilitzats
després de la consulta
    mysqli_close($con); //Tanquem explícitament la connexió a la base de dades
}
?>

```

getItems.php

```
<?php header("Content-Type: text/html;charset=utf-8");

include 'functions.php';

$con = connectToBBDD(); // Connexió a la BBDD

$request = $_GET["request"]; // Agafem la petició passada

if ($request[2] == 1) {
    $request = "(SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 - $request[0]) dif FROM inventory WHERE email='$request[1]' AND type = 'Jersey' LIMIT 2) UNION (SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 - $request[0]) dif FROM inventory WHERE email='$request[1]' AND type = 'Bib' LIMIT 2) UNION (SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 - $request[0]) dif FROM inventory WHERE email='$request[1]' AND type = 'Layer' LIMIT 2) UNION (SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 - $request[0]) dif FROM inventory WHERE email='$request[1]' AND type = 'Gloves' LIMIT 2) UNION (SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 - $request[0]) dif FROM inventory WHERE email='$request[1]' AND type = 'Socks' LIMIT 2) UNION (SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 - $request[0]) dif FROM inventory WHERE email='$request[1]' AND type = 'Neck' LIMIT 2) UNION (SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 - $request[0]) dif FROM inventory WHERE email='$request[1]' AND type = 'Overshoes' LIMIT 2) UNION (SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 - $request[0]) dif FROM inventory WHERE email='$request[1]' AND type = 'Cap' LIMIT 2) UNION (SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 - $request[0]) dif FROM inventory WHERE email='$request[1]' AND type = 'Other') ORDER BY dif;";
} else {
    $request = "(SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 - $request[0]) dif FROM inventory WHERE email='$request[1]' AND type = 'Jersey' LIMIT 2) UNION (SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 - $request[0]) dif FROM inventory WHERE email='$request[1]' AND type = 'Bib' LIMIT 2) UNION (SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 - $request[0]) dif FROM inventory WHERE email='$request[1]' AND type = 'Layer' LIMIT 2) UNION (SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 - $request[0]) dif FROM inventory WHERE email='$request[1]' AND type = 'Gloves' LIMIT 2) UNION (SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 - $request[0]) dif FROM inventory WHERE email='$request[1]' AND type = 'Socks' LIMIT 2) UNION (SELECT type, name, mintemp, maxtemp, ABS((mintemp + maxtemp) / 2 - $request[0]) dif FROM inventory WHERE email='$request[1]' AND type = 'Other') ORDER BY dif;";
}

if (!$con) {

    echo (false);
```



```

} else {

    mysqli_set_charset($con, "utf8"); //formato de datos utf8
    $res = mysqli_query($con, $request); // Es realitza la petició al servidor

    if ($res != null) {

        $result = array(); //Array on anirem guardant la informació recollida
        // Recopilem la informació:
        $i = 0;
        while ($row = mysqli_fetch_array($res)) {

            $result[$i] = $row;
            $i++;
        }

        echo json_encode($result); // Retornem l'array en format JSON

    } else {

        echo (false);
        die();
    }
}

disconnectToBBDD($res, $con); // Desconnexió de la BBDD

?>

```

getItems2.php

```

<?php header("Content-Type: text/html;charset=utf-8");

include 'functions.php';

$con = connectToBBDD(); // Connexió a la BBDD

$email = $_GET["request"]; // Agafem la petició passada
$request = "SELECT * FROM inventory WHERE email='$email' ORDER BY type";

if (!$con) {

    echo (false);

} else {

    mysqli_set_charset($con, "utf8"); //formato de datos utf8

```

```

$res = mysqli_query($con, $request); // Es realitza la petició al servidor

if ($res != null) {

    $result = array(); //Array on anirem guardant la informació recollida
    // Recopilem la informació:
    $i = 0;
    while ($row = mysqli_fetch_array($res)) {

        $result[$i] = $row;
        $i++;
    }

    echo json_encode($result); // Retornem l'array en format JSON

} else {

    echo (false);
    die();
}
}

disconnectToBBDD($res, $con); // Desconnexió de la BBDD

?>

```

editItem.php

```

<?php header("Content-Type: text/html;charset=utf-8");

include 'functions.php';

$con = connectToBBDD(); // Connexió a la BBDD

$request = $_POST["request"]; // Agafem la petició passada

if ($request[1] == 2) {
    // ELIMINACIÓ D'ITEM
    $request = "DELETE FROM inventory WHERE id = $request[0]";
} else if ($request[1] == 3) {
    // ELIMINACIÓ DE TOTS ELS ITEMS DE L'USUARI
    $request = "DELETE FROM inventory WHERE email = '$request[0]'";
} else {
    // INSERCIÓ D'ITEM

```

```
    $request = "INSERT INTO inventory (`id`, `email`, `type`, `name`, `min-temp`,
`max-temp`) VALUES (NULL, '$request[0]', '$request[1]', '$request[2]', '$request[3]',
'$request[4]');";

}

if (!$con) {

    alert("Connexió OK");

    echo (false);

} else {

    mysqli_set_charset($con, "utf8"); //formato de datos utf8
    $res = mysqli_query($con, $request); // Es realitza la petició al servidor

    if ($res != null) {

        echo (true);

    } else {

        echo (false);
        die();
    }
}

disconnectToBBDD($res, $con); // Desconnexió de la BBDD

?>
```