

# Travel Blog

---

A hamburgcodingschool project.

## Features

---

### View of anonymous users

Anonymous users or users that aren't logged in yet shouldn't be able to access our blog. You can decide between two options:

1. Show nothing but a login button with a friendly message to anonymous users.
2. Show a map and a login button, this shall give the user an idea of how the blog looks like. They can either click on the login button or anywhere on the map to trigger the login flow.

Using Firebase Auth UI signing up and signing in are handled in the same flow so we don't need to worry about it.

### Once signed in

- Display some sort of greeting with the user (maybe inside a header)
- Add a logout button for the user to end the session

The app consists of a couple of main parts you can arrange and display to your liking:

### The Map

The Google Map can be clicked to show the form with the location of the click. To visually show the user where they clicked show a marker on the click location.

Additionally, you should load all blog posts and display their locations as markers on the map. The markers should be connected with a line and the map should be automatically zoomed to fit all markers.

When hovering over a marker we want to show the title of the blog post in an info window.

### The Form

The form is shown after the map is clicked. It receives the latitude and longitude. Technically, this is enough to create a new blog post. Solely for the purpose of a better user experience we want to use reverse geocoding to look up the address of the given location.

The form should have three input elements:

- one for the title of the blog post
- one for the content of the post

- one to upload one an image

Additionally it should have a *Submit* and a *Cancel* button. The *Submit* button reads the data from the input elements and creates a new blog post in Firebase. The *Cancel* button discards all entered values and hides the form. Remember to hide the marker on the map as well.

Make sure the form can only be submitted when at least the title and content is filled out.

## The Posts

The posts are displayed as markers on the map. To make it more appealing we also want to display them in a list on the blog.

Each post should show the title, content and if available the image.

Also it should the address of where the post happened (use reverse geocoding to look it up).

## Bonus Features

---

If you implemented everything above: Congratulations, you've build your first large app. 🍷

Here are some bonus features you can add to flex your developer muscles:

- When a marker is clicked we want to display the corresponding blog post (either by scrolling it into view or in a modal dialog)
- Each post should have a button to delete it
- Each post should have a button to edit it (when clicking the button, open the form and pass the current blog post as initial data)
- Add a local search which can be used to filter blog posts (by their title and content)
- Build a filter (checkbox) that only displays blog posts with images (in the list and on the map)
- Each post should have a button to like (or bookmark) it (when clicked it should toggle a field in the Firebase database)
  - Build a filter (checkbox) that only displays liked / bookmarked blog posts (in the list and on the map)
- Add an option to hide the map
- Add an option to hide the list of blog posts

## Interesting Links

---

- Everything you need to know about JavaScript: <https://developer.mozilla.org/en-US/docs/Web>
- Firebase Auth with Firebase UI: <https://firebase.google.com/docs/auth/web/firebaseui> and <https://github.com/firebase/firebaseui-web>
- Code Examples for Google Maps: <https://developers-dot-devsite-v2-prod.appspot.com/maps/documentation/javascript/examples/>
- Firebase Storage documentation: <https://firebase.google.com/docs/storage/web/start>

- Firebase Firestore documentation: <https://firebase.google.com/docs/firestore/manage-data/add-data>