

The My-World-Story Project

Introduction

Welcome to the "My World Story" project! In this assignment, you will complete a web application that serves as an interactive, map-based journal. The application allows users to write stories, associate each story with a specific geographical location, and save their stories to a server.

You will be working with a pre-existing codebase that has some key features missing. Your task is to implement the missing functionality and then use the completed application to create your own collection of five stories from around the world.

This project will give you hands-on experience with modern web development concepts, including asynchronous JavaScript, DOM manipulation, working with third-party libraries, creating a dynamic user interface using signals, and interacting with a REST API.

The goal is to give you a real-world web application experience. The app demonstrates concepts commonly found in modern applications.

The user interface is built using signals—a modern concept increasingly adopted by frontend frameworks like Angular, React, and Vue.js. See our [Lecture 12](#).

The app also uses the [Leaflet](#) library, although you won't need to interact with it directly. A custom wrapper around Leaflet has been created for easier use. See our [Lecture 14](#) about web components.

Project Structure

This project consists of two parts:

1. Complete the app to bring it into working condition.
2. Write five short (personal) stories about five locations and mark them on the world map.

Part 1: The My-World-Story App

Tasks

1. Understand the folder/project structure and the roles of the files. Refer to the table below:

File/Folder	Description
android-chrome-192x192.png	Icon file, see favicon.io
android-chrome-512x512.png	Icon file
apple-touch-icon.png	Icon file
favicon-16x16.png	Icon file
favicon-32x32.png	Icon file
favicon.ico	Icon file

File/Folder	Description
images/	Images for the Leaflet component
index.html	The main entry point of the web application
leaflet.css	Styling for the Leaflet component
leaflet.js	The Leaflet JavaScript code
main.css	Application-specific styling
main.js	JavaScript code for interacting with API endpoints
osm-map.js	Web component that wraps the Leaflet library for easier usage
signal.js	Signal implementation from Lecture 12
site.webmanifest	JSON file providing metadata about the web application

2. Your first task is to set the `ownerName` constant (line 87) to your own name. This serves two purposes:

1. Identifying the student for evaluation
2. Filtering the student's records in the backend database

3. Understand the provided functions in `main.js`.

4. Complete the app's functionality by improving `index.html`:

- At startup: read the first record from the backend. If there is no record, create a new one.
- Analyze the returned data. It contains all the necessary information to drive the user interface.
- Update the signals (which drive the user interface) with the returned data.
- Keep the state of the buttons in sync with the signals.
- Understand the data flow and complete the remaining functionality.

The app is considered fully functional when:

- New stories can be created.
- The save button is only enabled if the current story data has been modified. Clicking it should update the backend correctly.
- The delete button is only enabled if a story with a valid `id` exists.
- The new button is only enabled if there is room for a new story in the database (limited to five stories for this project). Note: the database can technically store billions of stories.
- The previous button is only enabled if a previous record exists.
- The next button is only enabled if a next record exists.
- Story navigation works seamlessly.
- The map displays the correct location.

Test the app to ensure all functionality works as expected. You should be able to create, update, and delete stories. Navigation buttons should function correctly, the interface should update accordingly, the map should "fly" to the correct location, and the buttons should be enabled or disabled appropriately.

Tip: We were already experimenting with the backend api in our last lecture, see [Lecture 16](#). The downloadable exercise on this page demonstrates the full set of crud operations. This can be a starting point to better understand the api, or as a debugging aid, if you want to analyze the data that was actually stored in the data base while you were coding this app. You will also find a detailed description of the backend api on the slide before (161).

Part 2: Write Your World Stories

Use the app to write five short, personal stories and associate them with five different locations on the map. The stories need to be saved in the backend database.

To submit the assignment, zip the entire project folder and upload it to MS Teams.