

Project: Geotrack Application

Vaibhav Singh

November 8, 2023

1 Goal of the Project

The goal of this project is to create an application that provides users with a precise location tracking feature and helps them keep track of their running and cycling goals within a specified location. The application will allow users to mark specific locations where they want to engage in these activities and will log all user-provided data in local storage. This way, even when the user closes the application and returns at a later time, their previously saved goals will be readily accessible.

2 Technologies Used

The following technologies will be used to develop the Geotrack Application:

- HTML
- CSS
- JavaScript (including Object-Oriented Programming concepts)
- Geolocation API
- Leaflet.js
- Leaflet.css
- Local Storage Browser API

3 How the Project Works

The Geotrack Application is designed to provide users with a seamless experience of tracking their running and cycling goals at precise locations. Here's a step-by-step explanation of how the project functions:

- **User Location Access:** When the user visits the webpage, a pop-up will appear asking for access to their location.

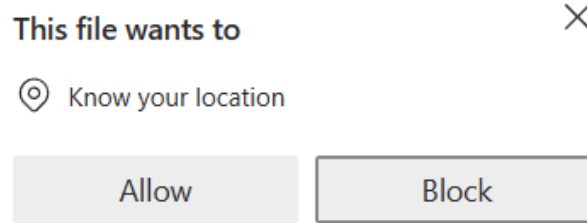


Figure 1: Location Display

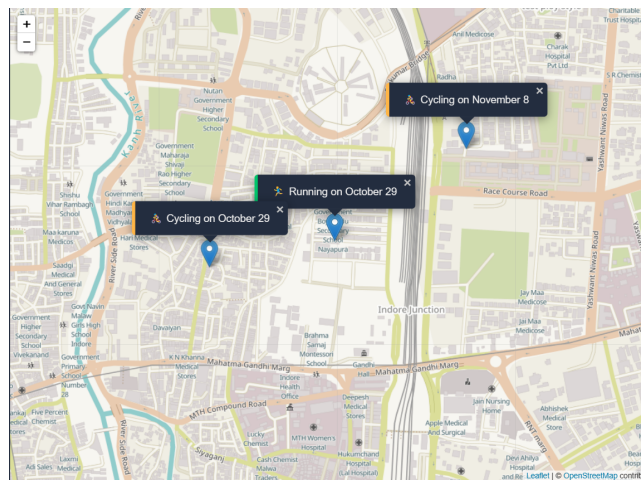


Figure 2: Setting Goals

- **Location Display:** After the user has allowed access to their location, it will show their exact location and their previous activities that the user has marked on the web page.
- **Setting Goals:** Whenever the user clicks on the map, a form will appear with two main options - Running or cycling. Now the user can set their goal for either running or cycling.
- **Form Submission:** When the user submits the form, their location with their goal will be marked on the map.
- **Local Storage:** Additionally, the data provided by the user will be saved in the local storage of the browser. This way, even if the user exits the browser and visits the application next time, it will show their goals, and they can keep track of their goals on the map.

Type	Runni <input type="button" value="v"/>	Distance	<input type="text" value="km"/>
Duration	<input type="text" value="min"/>	Steps per minute	<input type="text" value="step/min"/>

Figure 3: Running Goal

Type	Cyclin <input type="button" value="v"/>	Distance	<input type="text" value="km"/>
Duration	<input type="text" value="min"/>	Gain	<input type="text" value="meters"/>

Figure 4: Cycling Goal

Cycling on November 8

11 KM
 12 MIN
 55.0 KM/H
 12 M

Cycling on November 8

11 KM
 12 MIN
 55.0 KM/H
 10 M

Cycling on October 29

11 KM
 11 MIN
 60.0 KM/H
 13 M

Running on October 29

13 KM
 12 MIN
 0.9 MIN/KM
 11 SPM

Figure 5: Form Submission

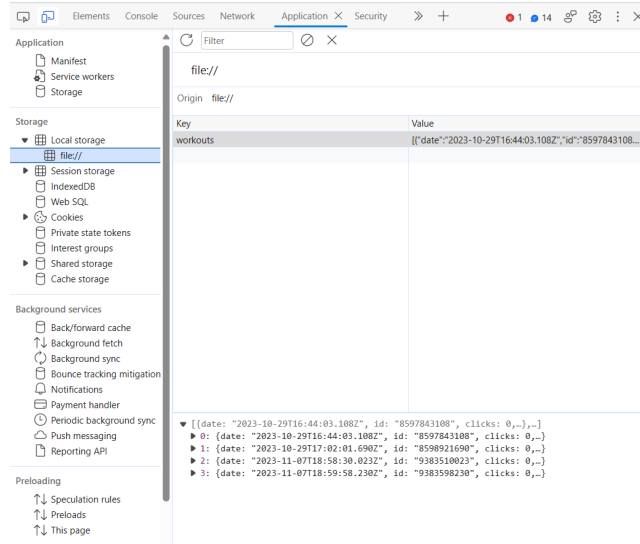


Figure 6: Local Storage

4 How the Project is Made

The Geotrack Application is constructed using the following steps:

- **HTML and CSS:** HTML and CSS are used to design the structure of the webpage.
- **Geolocation API:** When the user opens the application and allows access to their location using the Geolocation API in the browser, the exact longitude and latitude are received. With the help of the OpenStreetMap API, the user's precise location is displayed on the map.
- **Marker Placement:** Whenever the user clicks on a particular location, an event handler function is triggered to obtain the exact coordinates of that location. Using this data, a marker is displayed at the exact location.
- **Goal Form:** When the user clicks on a location, an HTML form pops up, presenting two options: running or cycling, each with its own suboptions.
- **Form Submission:** Upon form submission, a marker is set on the chosen location with a tag indicating "Running/Cycling on Date." This information also appears in the log on the left side of the page.
- **Local Storage:** All data is stored in the local storage of the browser in the form of key-value pairs.

- **Object-Oriented Code:** The code is organized using Object-Oriented Programming (OOP) principles. The main class, App, contains two subclasses, Workouts and Map, and various methods, including the constructor, getPosition, loadMap, showForm, toggleElevationField, and newWorkout. The Running and Cycling classes inherit from the Workout class, providing specialized functionality.

5 Sources Preferred during the Making of Project

Throughout the development of the project, we will refer to the following sources for guidance and documentation:

- Mozilla Developer Network (MDN) Web Docs: <https://developer.mozilla.org/>
- Leaflet Documentation