What is it?

Mile Tracker is a simple web application that helps me keep track of all my miles ridden on my bicycle or a stationary bike.

What will it do?

Its main aim is to collect information about the days ride. To do that the following information would be collected:

- Number of miles ridden
- Name for the ride
- Current Date (auto generated by db)
- GPX file to plot bike ride on a map (optional)
- Notes about the ride (optional)
- Images from the ride (optional)
- Authentication

Information Collected

Miles Ridden

The first type of information collected is the number of miles ridden. This information is collected on a day by day basis. The user would be able to enter a decimal number of what their current number of miles ridden for the day was. This information is required.

Title for the ride

The name of the ride will serve as a way for the user to associate a memory of whatever happened during the ride that could make it memorable. This input is not optional.

GPX file data

The application will also take in an optional GPX file that would show the route they have ridden. This field is optional. The ridden routes would only be shown after the upload is done and the user select a particular ride, a new page showing the information collected in a visually appealing manner. The information gotten from the GPX file if provided would be used to plot the routes rode by the user on a map like strava.

Date

The date and time when a user uploads data about the ride to the application, they do not have to worry about specifying what day and time they rode, all that would automatically generate the time and date by using the database date and time DATATYPES

Notes about the ride

The user can add notes of whatever to a particular ride if they wish to. This field is optional.

Images from the ride

The user can also upload images taken during the ride. This field is optional.

Authentication

Because the information being collected from the user can be used to identify them, password authentication is used to secure the data.

Technology Stack

To decide what the technology stack of the application would be, there is a need to evaluate what the application itself would be doing. "records " in this document refer to bike rides.

Functionality

The application will need to handle the following information to be functional and useful:

- Authentication
- CRUD operations on records
- Show all records in the database (preferably in a table)

- Show a single record in the database (with all the information)
- Display a map if a GPX file is provided

Technologies Used

The entire application would be powered by the Laravel framework because it is what I am most familiar with and this application is not entirely complex.

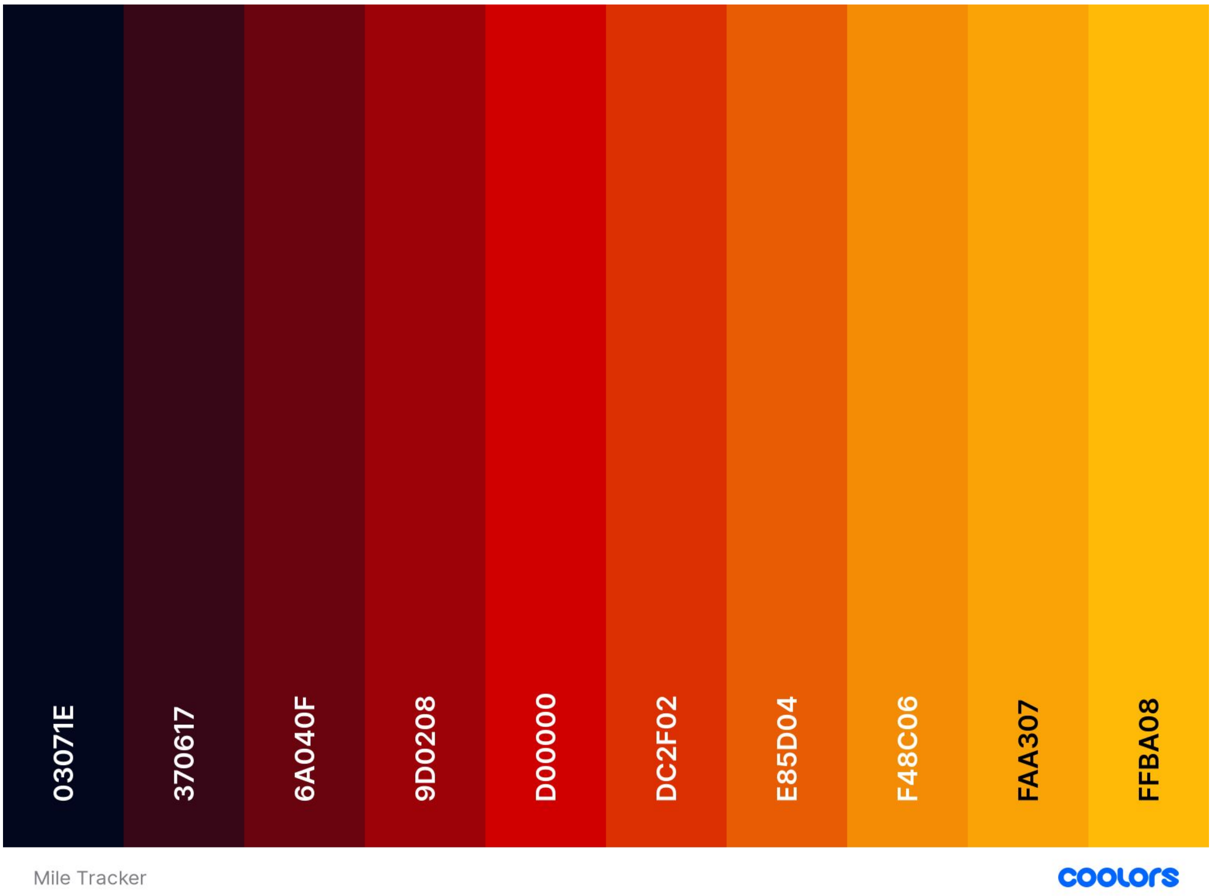
The datastore would be powered by SQLite because :

- this is not an entirely complex application
- it has to be cost effective to run on a server ($5 a month).
- A RDBMS is way outrageous because the application would be use almost exclusively by me.
- SQLite is easy to set up.

For the view, Tailwind CSS would be used because it is super easy and I don't have to write any dreaded CSS which is what persuades me away from web applications anyway.

For the map data information that would be gotten from the GPX file, a javascript library called leaflet.js would be used for the maps and a for processing GPX data with leaflet.js a plugin called leaflet-gpx would be used.

Color Palette



Deployment

For the web server, since the application would be built using the Laravel framework, NGinx would be used as the webserver.

Before the application is ready for deployment into a production environment, the following optimizations have to be done to ensure security of data, best performance.

Tailwindcss

Before a project that uses tailwindcss can be deployed into production, the documentation has information on things that need to be done to ensure the css is as lightweight as possible.

Laravel

To deploy Laravel to production, some optimization techniques have to be put in place, the documentation has detailed information on how to go about it.

Server

The project would be hosted on digitalocean $5 server, the documentation for getting the server into a secure state can be found here