

Truck Tour App

The goal of this project is to create a web portal for users to take tracking on the truck's trip and their expenses. The task is given by CHI Deutschland Cargo Handling GmbH as a test project for the Full Stack Software Developer position.

Table of Contents

- About The Project
- Built With
- Getting Started
- Prerequisites
- Installation
- Usage
- Approaches

About The Project

The project is to Create a demo platform for stakeholder Jeff as a way to track tour fuel costs for the trucking fleet. The frontend is developed using React js and the backend is developed using Nest js.

Criteria:

- There should be a page to enter a new truck which is identified by the plate number. It must have a property to enter its fuel price per liter.
- An overview page "Truck fleet" should exist to list all available trucks which are entered into the system
- A truck can be deleted from the overview page and also selected from the overview page to be edited.
- A page should exist to enter a truck tour which is identified by a unique name. It must have a property for distance in kilometers, selected truck from the truck fleet and date of tour start and end.
- An overview page "Truck tours" should exist to list all available tours which are entered into the system. This table must contain a column for the cost of every tour which is calculated by fuel price per liter * distance.

- A tour can be deleted from the overview page and selected from the overview page to be edited.

Built With

This section has the list of frameworks/libraries used to bootstrap the project.

- [Nest.js](#)
- [React.js](#)
- [Bootstrap](#)
- [MongoDB](#)

(back to top)

Getting Started

The project is developed in javascript environment. So, to start the project nodejs environment need to be in the local machine.

Prerequisites and Run

The app is running in two different the frontend on port 3000 and for backend port on 4000. It's possible to run the two systems separately or combinedly. To run the system first need to go to the backend folder and the run dev.

* /backend location

```
cd backend
```

* Run the app

```
npm run dev
```

Or to run the backend and frontend separately the following command need to use.

* /backend location

```
cd backend
```

* Run the Backend

```
npm run server
```

* Run the Frontend

```
npm run client
```

Usage

In the frontend has two major features an overview page "Truck fleet" and an overview page "Truck tours".

- Truck fleet

Truck Fleet
List of all available Trucks information

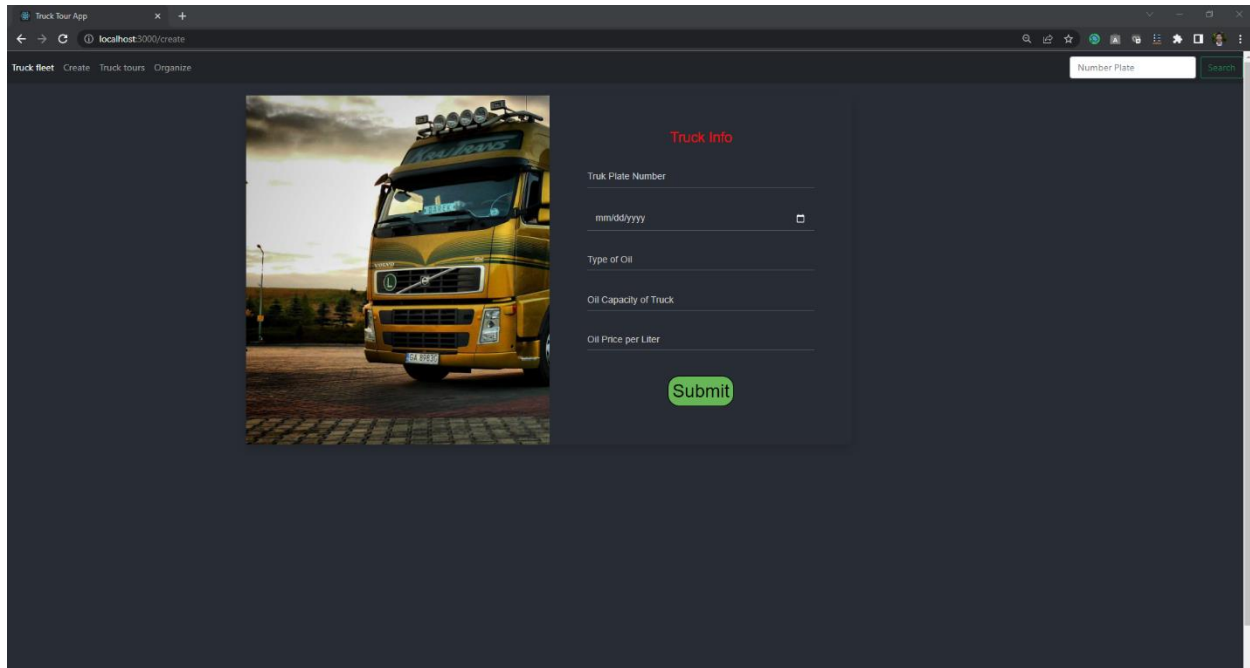
| Truck Number Plate | Entry Date | Fuel Type | Fuel Tank Capacity | Fuel Price | Action |
|--------------------|------------|-----------|--------------------|------------|---|
| CB121643 | 2022-05-16 | Patrol | 56 | 2 | Edit Remove |
| PS1122345 | 2022-05-15 | Patrol | 23 | 2 | Edit Remove |
| PS1122345 | 2022-05-15 | Patrol | 23 | 2 | Edit Remove |
| DA01PS043 | 10.05.2022 | diesel | 25 | 1.96 | Edit Remove |
| PS1122345 | 2022-05-15 | Patrol | 21 | 1 | Edit Remove |

- Truck tours

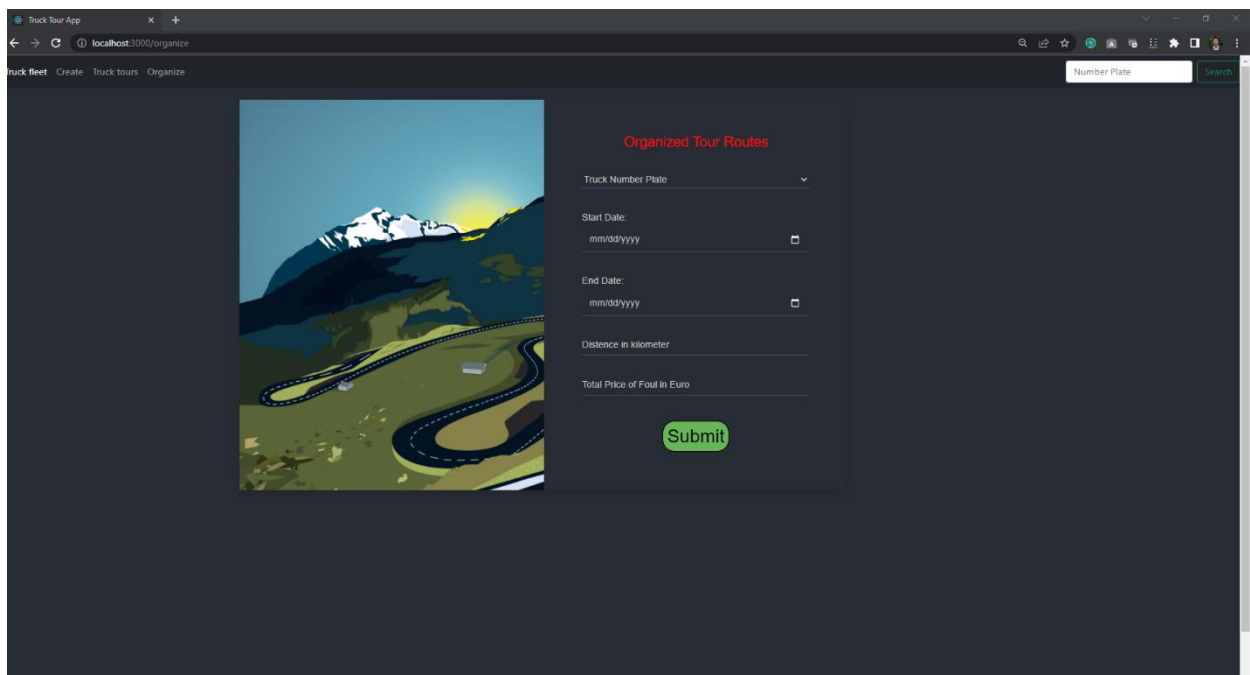
Truck Tours
List of all available Truck Tours

| Index | Truck Number Plate | Start Date | End Date | Distance (km) | Total Cost (EURO) | Action |
|-------------------|--------------------|------------|------------|---------------|-------------------|---|
| 20220517DA01PS043 | DA01PS043 | 2022-05-17 | 2022-05-22 | 25 | 110 | Edit Remove |

For “Truck fleet” page user can input information like the Truck number plate, date of input, type of fuel, fuel capacity , and fuel price.



Depending on the “Truck fleet” the user can generate the “Truck tours” with the information about the start and end date, Truck number plate, and distance and get the cost according to the trip information. The cost is calculated by fuel price per liter * distance and divide by average fuel per liter.



Approaches

To develop the project the following approaches that I follow.

1. Creating the backend and frontend for the initial project using nest js and react js.
2. Developing the backend for CRUD methods.
3. Connecting the backend with database MongoDB.
4. Designing the frontend in react JS.
5. Connecting the frontend and backend.
6. Fixing the bugs and issues.
7. Add required documentations.