Readme

This project was created as a pre-assignment for [Solita's Dev Academy 2023](https://github.com/solita/dev-academy-2023-exercise).

In this web app you can search for city bike stations in Helsinki and Espoo, view average journeys from each station, and view the location on map.

The app has a database of journeys made in May, June and July 2021. You can search for trips based on the departure station, return station and date.

The backend is done using Node, Express and Javascript and the frontend is done with React.js.

**Cloud hosted app**

Here is a link to the cloud hosted application: LINK

The app is hosted on Amazon Web Services. The backend is hosted on AWS Elastic Beanstalk. AWS Elastic Beanstalk is a service that makes it easy to deploy and manage web applications in the Amazon Web Services (AWS) cloud. It automates the details of capacity provisioning, load balancing, auto-scaling, and application deployment, creating an environment that runs a version of your application.

The frontend of the application is stored in AWS S3 bucket. S3 is good for hosting frontends because it offers scalable and secure storage for static websites.

**Running the app locally**

-Required: newest version of Node.js

-Clone the repository by running git clone <https://github.com/kaisadia/City-bike-app.git> in your terminal.

- Install all packages by running npm install in the root of the project.

-Building the frontend: run npm run build in the root of the frontend

-This project uses AWS Aurora PostgreSQL database. To connect your own database, create an .env file with the following environmental variables:

PGUSER=xxxx

PGHOST=xxxx

PGDATABASE=xxxx

PGPASSWORD=xxxx

PGPORT=5432

-To make the frontend stations maps available, set up your own REACT\_APP\_GOOGLE\_MAPS\_API\_KEY and store it in the .env.local file.

**Running the app with Docker**

The backend and frontend both consist of their own Docker-files.

To run the frontend in Docker:

docker run -p 80:3000 -e REACT\_APP\_GOOGLE\_MAPS\_API\_KEY=”your-api-key” kaisadia/citybikes-frontend:latest

To run the backend in Docker:

docker run -p 8080:8080 -e PGUSER=xxxx -e PGHOST=xxxx -e PGDATABASE=xxxx -e PGPASSWORD=xxxx -e PGPORT=5432 kaisadia/citybikes-backend:latest

**Database**

The database is cloud hosted in AWS Aurora. It’s a fully managed, PostgreSQL-compatible relational database engine that combines the speed, reliability, and manageability of Amazon Aurora with the simplicity and cost-effectiveness of open-source databases. I chose a relational database for this project since the data is consistent. I used PgAdmin as a GUI-tool.

I created two tables, stations and trips, with the following schemas:

CREATE TABLE trips (

id integer NOT NULL,

dep\_time character varying(50),

ret\_time character varying(50),

dep\_station\_id integer,

dep\_station\_name character varying(50),

ret\_station\_id integer,

ret\_station\_name character varying(50),

covered\_distance integer,

duration integer,

dep\_time\_new timestamp without time zone

);

From trips, I deleted duplicate rows and rows, where duration was less than 10 seconds or where distance was less than 10 meters.

CREATE TABLE stations (

id character varying(50) NOT NULL,

station\_id integer,

station\_name character varying(50),

address character varying(50),

city character varying(50),

capacity character varying(50),

x double precision,

y double precision

);

API

Tests

For running the tests locally, use npm test.

Backend API tests are done with Jest and Supertest, which are two libraries used for backend API testing. Jest is a JavaScript-based testing framework that lets you test both front-end and back-end applications, while SuperTest is a library for doing HTTP testing. Together, they allow for behavioral testing of the input and output of data without knowledge of the internal structure of the API.

Frontend tests are made with Jest and react-testing-library.