

Project documentation

for

Developing a Progressive Web App with JavaScript/Express.js

https://letseat-heroku.herokuapp.com

Within the training programme

for

Certified JavaScript Software Developer

October – April 2020/21

WIFI – Vienna

Name: Claudia Reut

Trainer/Coach: Alexander Vincenz

APP Description

**letSeat.** is a progressive web app, deployed with the Cloud Application Platform **Heroku** where a user can register and create an account in order to search restaurants, bars, gelaterias, bakeries etc. in Vienna.

The server part of the **letSeat.** application is **a Node-/Express-Backend** and the communication between front- & backend is handled via **REST-API**.

The application includes the **CRUD** functions so that admin is able to create, read, update and delete data. The data is stored in a JSON file named **places.json.**

A user can register on the platform with a username, password and e-mail address. The password is stored in encrypted form and the whole data is added to a JSON file named **user.json.** User can log in using the e-mail and the password, in order to search, save and remove favorite places in the account.

The admin side allows the user to add, edit and delete places to and from the database.

Server side programming

API - Endpoints

The following endpoints are to be find in server.js

F01 – Endpoint for user login

app.post("/login", loginUser);

F02 – Endpoint for user registration

app.post("/user", registerUser);

F03 – Endpoint for getting all places & conditionally also user´s saved favorite places

app.get("/places/:userid", getAllPlaces);

F04 – Endpoint for adding a place

app.post("/places/add", addPlace);

F05 – Endpoint for deleting a place

app.delete("/places/:id", deletePlace);

F06 – Endpoint for getting the place by id (to be edited)

app.get("/admin/edit/:id", getPlaceToEdit);

F07 – Endpoint for editing a place by id (saving changes)

app.put("/places/:id", editPlace);

F08 – Endpoint for saving a favorite place id in user

app.post("/places/:placeid/:userid", savePlace);

F09 – Endpoint for removing a favorite place id from user

app.delete("/places/:placeid/:userid", unsavePlace);

MoSCoW Prioritisation

**Must Have**

* Full Stack App (Front-&Backend)
* Progressive Web App
* Database – JSON file
* User can register & login into the frontend application
* Encrypted password in JSON file

**Should Have**

* CRUD functions (create/read/update/delete)
* Implementing unique id with nanoid
* SessionStorage for logged in user

**Could Have**

* Authentication via JWT- Web Token to secure API-transactions & private routes
* Uploading pictures
* Geolocation including geocoding for transforming the address in coordinates.
* MongoDB/ MySQL Database instead of JSON File

**Would Have**

* Frontend implemented with React framework

Timetable (Milestones)

**1.Coaching**

11.02.

* Concept creating using draw.io
* Creating Gitlab-Repository including Milestones/Issues

**2.Coaching**

23.02.

* Setting up Express.js
* Endpoint F01 for Log in
* Endpoint F02 for Registration
* Style with Bootstrap
* Frontend validation for log in/registration

**3.Coaching**

09.03.

* Admin Interface
* Endpoints F03 for adding a new place
* Checkboxes for adding places
* Validation to avoid double registration
* Adding Bcrypt for log in for comparing passwords
* Adding Bcrypt for register for hashing the passwords and save it in the database
* Rendering the places including Bootstrap & CSS

**4.Coaching**

23.03.

* Backend validation for wrong/not existing e-mail/password
* Endpoints F05 & F06 for deleting & editing a place
* Endpoint F07 for editing a place by id

**5.Coaching**

08.04.

* User Interface
* Rendering the places including Bootstrap & CSS
* SessionStorage for logged user
* Endpoint F08 for saving favorite place id in user
* Endpoint F09 for removing favorite place id from user
* F03 for getting all places and getting all favorite places from the user
* PWA with deployed with the Cloud Application Platform Heroku