Back-end Documentation

The backend part of the project is currently handling the registered users and the services that they provide or accept, we have implemented some admin functionalities like getting all the users registered on our app, get details about a specific user, update user or create new user and the same functionality also goes with the users' services.

The backend structure consists of four parts:

1) Server.js

Which is the entry point to the backend and contains:

1. Initialization of express framework.

2. Connection to MongoDB which is a NOSQL database.

3. Routes for the users API and the service API.

4. Some dependencies that help facilitates the backend functionality.

2) Models

Models handle the interaction process with the database We have user model and service model.





3) Controllers

Controllers are working with the data given from or to the model

1.For the users controller we implemented

getAllUsers()

getUser()

createUser()

updateUser()

2.For the service controller we implemented

getAllServices()

getService()

createService()

updateService()

4) Routes

Routes connect controllers, models and requests together.

1.Users routes Connect user model with its controller.

- get all users using GET request => router.get('/', UsersController.getAllUsers);

- create user using POST request => router.post('/', UsersController.setUser);

- get one user using GET request => router.get('/:id', UsersController.getUser);

- update user using PATCH request => router.patch('/:id', UsersController.UpdateUser);

2. Services routes connect service model with its controller

- get all services using GET request => router.get('/', ServicesController.getAllServices);

- create service using POST request => router.post('/', ServicesController.setService);

- get one service using GET request => router.get('/:id', ServicesController.getService);

- update service using PATCH request =>

router.patch('/:id', ServicesController.UpdateService);