Fullstack Project - Samuel Deotti

This project aims to develop a fullstack application for a todo list platform. Users can login, add, view, edit, and remove a task, from their list. The application includes both a front-end and a back-end, with API for user and tasks management.

Technologies Used

For the development of this application, **Java** was used alongside the **Spring Boot** framework, providing a structure for building the backend services following the principles of RESTful architecture.

On the frontend, **React** with **TypeScript** was utilized to create a dynamic and type-safe user interface.

The database management was handled by **MySql**, with queries and data manipulations abstracted by the Spring Data JPA.

Additionally, **Docker** was used for containerization, making the deployment process more efficient and consistent.

What Was Developed

- A MySql database using Spring Data JPA, containing tables to store user information, post categories, and all relevant post data;
- Endpoints that interact with the MySQL database, supporting CRUD operations;
- Middlewares for validating request data and ensuring the user is authenticated and authorized to perform specific actions;
- A frontend interface built with React and TypeScript, offering a responsive and type-safe user experience;
- Containerization of the application with Docker, ensuring consistency across different environments.

PROF

Frontend

How to Build and Run the Application

1. Clone the project from GitHub:

   ```
   git clone https://github.com/samueldeotti/todo-list.git
   ```

   or

   ```
   git clone git@github.com:samueldeotti/todo-list.git
   ```

2. Navigate to the project directory:

```
cd todo-list
```

3. Build and Run the Docker Compose services:

```
docker-compose up --build
```

4. Wait for a new tab to open in your browser, or open it manually. The application usually runs on port 3000, but it may open on a different port. If you encounter issues, check the terminal for the specified port.

```
http://localhost:3000/
```

How to Stop the Application

To stop the application, press `Ctrl + C` in the terminal where Docker Compose is running, or run the following command in the project root directory:
`bash docker-compose down`

Frontend Routes

- /signin: `User signin page`
- /signup: `User signup page`
- /tasks: `Dashboard for adding, viewing, editing, and removing tasks`

Backend Ports

- User Management API: `8080`
- MySQL: `3306`

PROF

API Documentation

- For documentation on the User and Task Management API, navigate to Documentation in your browser.