Knowledge Transfer Document for CampusThreads

Project Overview

CampusThreads is a web-based collaborative platform for campus communities. It is built using React.js for the front end, Node.js for the back end, Tailwind CSS for styling, and MongoDB for the database.

Technologies Used

- Frontend: React.js, Tailwind CSS

- Backend: Node.js, Express.js

- Database: MongoDB

- Version Control: Git, GitHub

Project Structure

- client/: Contains the React.js front-end code.

- server/: Contains the Node.js back-end code.

- .env.example: Sample environment variables.

- README.md: Project documentation and setup instructions.

Prerequisites

- Node.js (version 14.x or higher)

- npm (version 6.x or higher) or yarn (version 1.x or higher)

- MongoDB (version 4.x or higher)

```sh{command}```

Wherever you see the above line it implies you have to write the “command” in terminal.

Setting Up the Project Locally

1. Download the Project Folder

Download the project folder from the provided link and unzip it to your desired location.

2. Set Up Environment Variables

Create a `.env` file in the `server` directory and configure the necessary environment variables. Refer to `.env.example` for the required variables.

3. Install Dependencies

Frontend

Navigate to the `client` directory and install the dependencies.

```sh
cd client
npm install
```

Backend

Navigate to the `server` directory and install the dependencies.

```sh
cd ../server
npm install
```

4. Start the Development Server

Backend

Start the Node.js server.

```sh
cd server
node index.js
```

Frontend

Open a new terminal window and start the React development server.

```sh
cd client
npm run dev
```

The application will be accessible at `http://localhost:3000`.

Key Features

- My Questions: A section for users to view and manage their questions.

- Answer a Question: A section where users can browse and answer questions posted by others.

- User Authentication: Secure login and registration system.

- Real-time Updates: Updates are reflected in real-time using websockets.

- Question Tags: Users can tag questions with relevant topics to improve searchability.

- Like : Users can like questions and answers.

- Commenting: Users can add comments to questions and answers for further clarification.

- Profile Management: Users can create and manage their profiles, including updating personal information and profile pictures.

- Notification System: Alerts users of new activity, such as new answers to their questions or comments on their posts.

- Search Functionality: Advanced search to filter questions and answers by keywords, tags, and other criteria.

Future Enhancements

- Private Messaging: Implement a feature for users to send private messages to each other.

- Reputation System: Introduce a reputation system where users earn points for their contributions.

- Mobile App: Develop a mobile application to complement the web platform.

- Gamification: Add gamification elements such as badges and achievements to increase user engagement.

Contact

For any questions or further assistance, feel free to contact to 202101113@daiict.ac.in.