**MERN Social Media Application Documentation**

**Overview**

The **MERN Social Media Application** is a full-stack web application that enables users to connect, share, and engage with posts. Built using the MERN stack (MongoDB, Express.js, React.js, Node.js), the app provides a seamless user experience with responsive design and a robust backend for secure data management.

**Key Features**

* **User Authentication**: Secure registration and login system using JWT.
* **User Profiles**: View and edit personal profiles, including updating profile pictures and bios.
* **Posts and Feeds**: Create, update, and delete posts. View a feed of user posts.
* **Follow System**: Follow/unfollow users to see their posts in your feed.
* **Responsive Design**: Works on desktops, tablets, and mobile devices.

**Technologies Used**

**Frontend**

* **React.js**: For building the user interface.
* **Redux**: For state management.
* **Axios**: For handling API requests.
* **CSS**: For styling and responsive design.

**Backend**

* **Node.js**: Runtime environment for the backend.
* **Express.js**: For creating RESTful APIs.
* **MongoDB**: NoSQL database for storing user and post data.
* **Mongoose**: For database modeling and interaction.
* **JWT**: For secure authentication.
* **bcrypt.js**: For hashing user passwords.

**System Requirements**

* **Node.js**: v14 or later
* **npm**: v6 or later
* **MongoDB**: Local instance or MongoDB Atlas

**Application Architecture**

**Frontend**

1. **Components**: React components handle user interactions and display data from the backend.
2. **State Management**: Redux manages global state, such as user authentication and posts.
3. **API Integration**: Axios communicates with backend APIs to fetch and manipulate data.

**Backend**

1. **Routes**: RESTful API routes for user authentication, posts, and profile management.
2. **Controllers**: Handle business logic for each route.
3. **Database Models**: Define data schemas using Mongoose.
4. **Authentication**: Middleware ensures secure access using JWT.

**Setup Instructions**

**Step 1: Clone the Repository**

git clone https://github.com/your-username/mern-socialmedia.git

cd mern-socialmedia

**Step 2: Install Dependencies**

For the backend:

cd backend

npm install

For the frontend:

cd frontend

npm install

**Step 3: Configure Environment Variables**

Create a .env file in the backend directory with the following values:

PORT=5000

MONGODB\_CONNECTION=your-mongodb-connection-string

JWTKEY=your-jwt-secret-key

**Step 4: Run the Application**

Start the backend:

cd backend

npm start

Start the frontend:

cd frontend

npm start

**API Documentation**

**Base URL**

* Backend: https://mern-socialmedia-master-backend.onrender.com
* Live App: [https://mern-socialmedia-master.onrender.com](https://mern-socialmedia-master.onrender.com/)

**Endpoints**

**Authentication**

1. **Register User**
   * **POST** /auth/register
   * **Request Body**:
   * {
   * "firstname": "John",
   * "lastname": "Doe",
   * "username": "john\_doe",
   * "password": "password123"
   * }
   * **Response**:
   * {
   * "user": { /\* user object \*/ },
   * "token": "jwt-token"
   * }
2. **Login User**
   * **POST** /auth/login
   * **Request Body**:
   * {
   * "username": "john\_doe",
   * "password": "password123"
   * }
   * **Response**:
   * {
   * "user": { /\* user object \*/ },
   * "token": "jwt-token"
   * }

**User Management**

1. **Get User Profile**
   * **GET** /user/:id
   * **Response**:
   * {
   * "id": "user-id",
   * "username": "john\_doe",
   * "firstname": "John",
   * "lastname": "Doe",
   * "followers": [],
   * "following": []
   * }
2. **Update User Profile**
   * **PUT** /user/:id
   * **Request Body** (optional fields):
   * {
   * "firstname": "John",
   * "lastname": "Doe",
   * "profilePicture": "image-url"
   * }

**Posts**

1. **Create Post**
   * **POST** /posts
   * **Request Body**:
   * {
   * "userId": "user-id",
   * "desc": "This is my post!",
   * "image": "image-url"
   * }
2. **Get All Posts**
   * **GET** /posts
   * **Response**:
   * [
   * {
   * "id": "post-id",
   * "userId": "user-id",
   * "desc": "This is my post!",
   * "image": "image-url",
   * "likes": []
   * }
   * ]
3. **Delete Post**
   * **DELETE** /posts/:id

**Usage Instructions**

1. **Register** for an account or log in with an existing account.
2. **Create posts** from the homepage or your profile.
3. **Follow other users** to see their posts on your feed.
4. **Edit your profile** to update your bio and pictures.

**Future Enhancements**

* Real-time chat using WebSockets.
* Notifications for likes, comments, and follows.
* Enhanced explore section for discovering users and posts.

**Screenshots**

[Add screenshots of key features such as login, profile page, and post feed.]

**Contributing**

Contributions are welcome! Follow these steps:

1. Fork the repository.
2. Create a feature branch: git checkout -b feature-name.
3. Commit your changes: git commit -m "Added feature-name".
4. Push the changes: git push origin feature-name.
5. Open a pull request.

**License**

This project is licensed under the [MIT License](https://chatgpt.com/c/LICENSE).