# Social I Project Documentation

## Table of Contents

1. Introduction
2. Project Goals
3. Technologies Used
4. Features
5. Project Structure
6. Installation Guide
7. Usage
8. API Endpoints
9. Application flow
10. Contribution Guidlines
11. Contact Information

## Introduction

The Social I project is a web application designed to mimic the core functionalities of Instagram, a popular social media platform. This project aims to provide a learning experience for developers by replicating key features such as user authentication, photo sharing, liking, commenting, and following users.

## Project Goals

* To create a fully functional clone of Instagram using modern web technologies.
* To provide a hands-on experience with front-end and back-end development.
* To implement user authentication and authorization.
* To offer an interactive user interface with real-time updates.
* To practice building and integrating RESTful APIs.

## Technologies Used

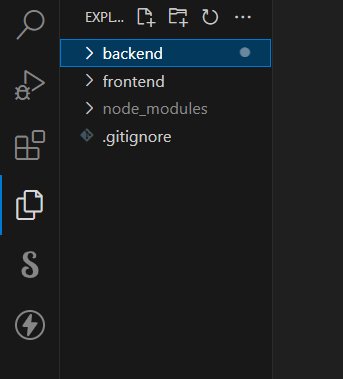
* **Frontend**: React.js, Redux, HTML, CSS, JavaScript, Tailwind
* **Backend**: Node.js, Express.js
* **Database**: MongoDB
* **Authentication**: JWT (JSON Web Tokens), Passport.js
* **File Storage**: Firebase Storage
* **Version Control**: Git, GitHub
* **Deployment**:Render, AWS EC2 or Vercel

## Features

1. **User Authentication**: Sign up, login, and logout functionalities using JWT.
2. **User Profiles**: Create and manage user profiles, including the ability to upload a profile picture and bio.
3. **Photo Upload**: Users can upload photos with captions.
4. **Feed**: Display a feed of photos from followed users, sorted by date.
5. **Like and Comment**: Users can like and comment on photos.
6. **Follow/Unfollow**: Users can follow or unfollow other users.
7. **Messaging and Chat**: Engage in private conversations and group chats with friends and followers.
8. **Notifications**: Users receive notifications for likes, comments, and follows (optional feature).
9. **Real-time Updates:** Stay up to date with the latest activities and posts from your connections. Receive instant notifications for likes, comments, and mentions, ensuring you never miss out on important interactions.

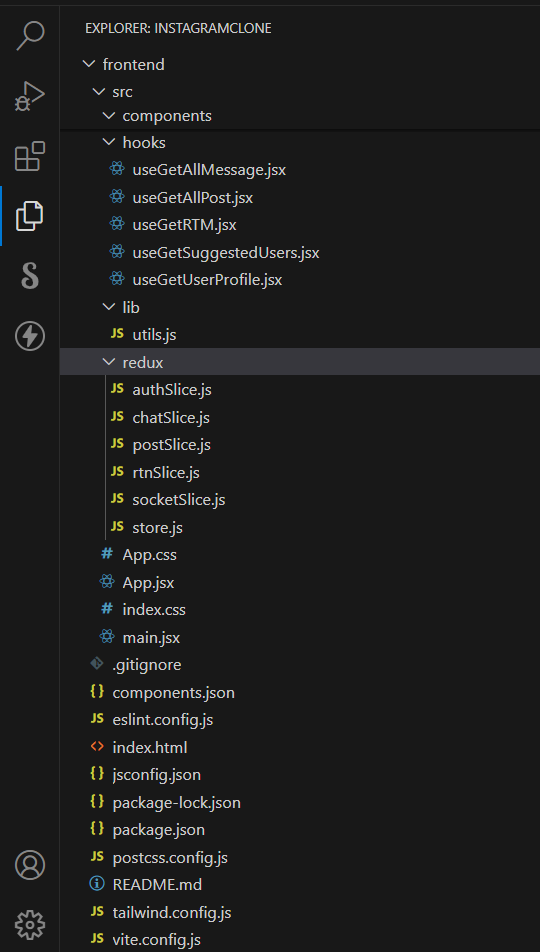
## Project Structure

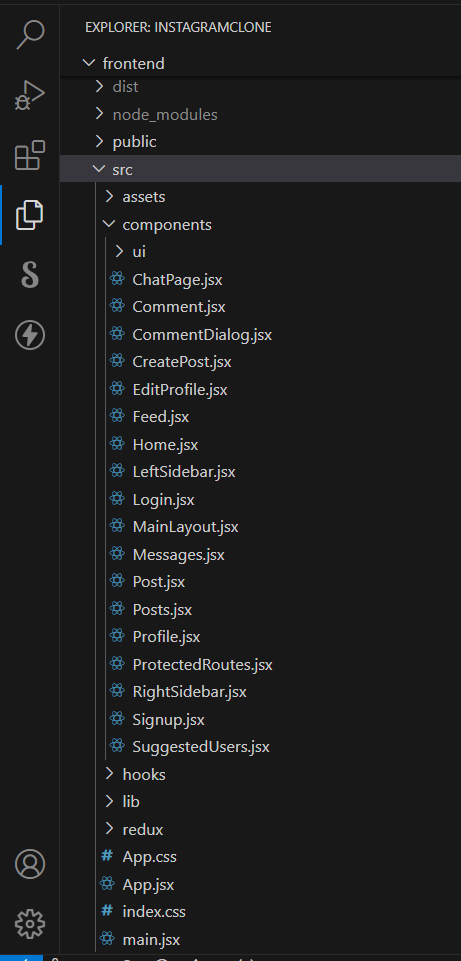
* Inside the SocialeX (social media app) directory, we have the following folders



* **Client directory:**

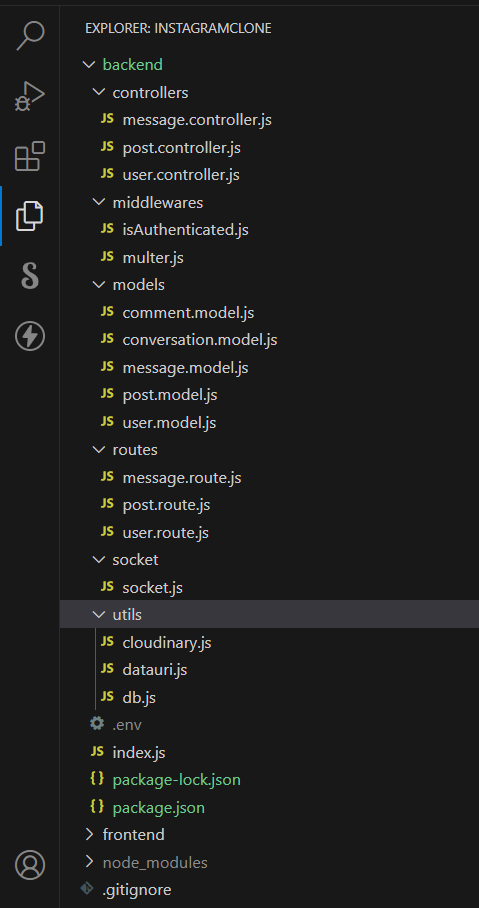
The below directory structure represents the directories and files in the client folder (front end) where, react Js is used along with Api’s such as socket.io.





* **Server directory:**

The below directory structure represents the directories and files in the server folder (back end) where, node js, express js and mongodb are used along with socket.io.



### PRE-REQUISITES

Here are the key prerequisites for developing a full-stack application using Node.js, Express.js, MongoDB, React.js, Socket.io:

* **Node.js and npm**:

Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the server-side. It provides a scalable and efficient platform for building network applications.

Install Node.js and npm on your development machine, as they are required to run JavaScript on the server-side.

* Download:<https://nodejs.org/en/download/>
* Installation instructions:<https://nodejs.org/en/download/package-manager/>
* **Express.js**:

Express.js is a fast and minimalist web application framework for Node.js. It simplifies the process of creating robust APIs and web applications, offering features like routing, middleware support, and modular architecture. Install Express.js, a web application framework for Node.js, which handles server-side routing, middleware, and API development.

Installation: Open your command prompt or terminal and run the following command:

**npm install express**

* **MongoDB**:

MongoDB is a flexible and scalable NoSQL database that stores data in a JSON-like format. It provides high performance, horizontal scalability, and seamless integration with Node.js, making it ideal for handling large amounts of structured and unstructured data.

Set up a MongoDB database to store your application's data.

* Download:<https://www.mongodb.com/try/download/community>
* Installation instructions:<https://docs.mongodb.com/manual/installation/>
* **React.js**:

React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications.

Install React.js, a JavaScript library for building user interfaces.

Follow the installation guide:<https://reactjs.org/docs/create-a-new-react-app.html>

* **Socket.io**:

Socket.io is a real-time bidirectional communication library that enables seamless communication between the server and clients. It allows for real-time data exchange, event-based messaging, and facilitates the development of real-time applications such as chat, collaboration, and gaming platforms.

Install Socket.io, a real-time bidirectional communication library for web applications.

Installation:

* Open your command prompt or terminal of server and run the following command: **npm install socket.io**
* Open your command prompt or terminal of client and run the following command: **npm install socket.io-client**
* **HTML, CSS, and JavaScript**: Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.
* **Database Connectivity**: Use a MongoDB driver or an Object-Document Mapping (ODM) library like Mongoose to connect your Node.js server with the MongoDB database and perform CRUD (Create, Read, Update, Delete) operations. To Connect the Database with Node JS go through the below provided link:

• [https://www.section.io/engineering-education/nodejs- mongoosejs-mongodb/](https://www.section.io/engineering-education/nodejs-%20mongoosejs-mongodb/)

* **Version Control**: Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.

• Git: Download and installation instructions can be found at:<https://git-scm.com/downloads>

* **Development Environment**: Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code.

• Visual Studio Code: Download from<https://code.visualstudio.com/download>

**Install Dependencies:**

• Navigate into the cloned repository directory:

**cd SocialeX**

• Install the required dependencies by running the following commands:

**cd client**

**npm install**

**cd ../server**

**npm install**

* **Start the Development Server**:

• To start the development server, execute the following command:

**npm start**

• The video conference app will be accessible at http://localhost:3000

* **Access the App:**

• Open your web browser and navigate to [http://localhost:3000](http://localhost:3000/).

## Usage

* **Signup**: Create a new account by signing up.
* **Login**: Use your credentials to log in.
* **Upload Photo**: Click the upload button to share a new photo.
* **Interact**: Like, comment, and follow other users.

## API Endpoints

|  |  |  |
| --- | --- | --- |
| **HTTP Method** | **Endpoint** | **Description** |
| GET | /api/users | Get all users |
| POST | /api/auth/signup | Register a new user |
| POST | /api/auth/login | Login user |
| GET | /api/posts | Get all posts |
| POST | /api/posts | Create a new post |
| GET | /api/posts/ | Get a specific post by ID |
| POST | /api/posts/  /like | Like a specific post |
| POST | /api/posts/comment | Comment on a specific post |

### Application flow

## User:

## Create and manage a personal profile.

## Share posts, photos, videos, and stories with their network.

## Engage in conversations through comments, likes, and shares.

## Follow other users and discover new accounts, topics, and trends.

## Explore and discover new content, communities, and opportunities.

## Interact with notifications and stay updated with the activities of their connections.

## Utilize messaging and chat features to communicate with friends and followers.

## Contribution Guidelines

## Fork the repository and create a new branch for your feature.

## Commit your changes with clear and concise commit messages.

## Create a pull request and describe the changes you've made.

## Ensure all tests pass and the application runs without errors before submitting a pull request.

## Contact Information

For questions or collaboration, please contact:

* **Name**: kanika singh
* **Email**: kanikac363l@gmail.com
* **GitHub**:kanikasingh4