**Final Evaluation Summary:**

Our social media app provides a dynamic platform for users to connect and interact. Utilizing modern technologies like React, Node.js, and MongoDB, the app ensures a seamless and engaging user experience. Users can easily register, customize their profiles, and communicate in real-time within a vibrant online community.

**Final URL of web application:**

https://final-assignment-project.netlify.app/

**Features and functions of web application:**

**Registration with Profile Image:** Users can easily register for the app, personalize their profiles with images, and set up their initial account details.

**Login Page:** Secure login functionality allows users to access their accounts quickly and securely.

**Post Management:** Users can create, edit, and delete posts, enabling them to share content seamlessly with others.

**Chat and Messaging:** A comprehensive chat list facilitates real-time messaging between users, promoting instant communication.

**Channel/Group Feature:** Users can create and participate in channels or groups, fostering community engagement and interaction.

**Profile Management:** Robust profile management tools enable users to update their information, manage privacy settings, and customize their profiles.

**Password Change:** Users have the option to change their passwords securely within the app, ensuring account security and user control.

**Languages and Technologies Implemented:**

**Front-end Technologies:**

**React:** A popular JavaScript library for building user interfaces, enabling dynamic and responsive web applications.

**Material UI (MUI):** A comprehensive UI component library for React, providing pre-designed components that enhance the app's visual appeal and user experience.

**Socket.io-client:** A client-side library for real-time, bidirectional communication between web clients and servers, enabling features like live chat and notifications.

**Redux:** A state management library for JavaScript applications, ensuring predictable state updates and efficient data handling.

**Axios:** A promise-based HTTP client for making API requests, simplifying communication with the backend server.

**Yup:** A JavaScript schema builder for value parsing and validation, used for form validation to ensure data integrity.

**Back-end Technologies:**

**Socket.io:** A library for real-time web applications, enabling real-time, bidirectional communication between web clients and servers.

**Node.js:** A JavaScript runtime built on Chrome's V8 engine, used for building scalable and efficient server-side applications.

**MongoDB:** A NoSQL database, providing flexible and scalable data storage for application data.

**Mongoose:** An Object Data Modeling (ODM) library for MongoDB and Node.js, offering schema-based solutions to model application data.

**Express:** A minimal and flexible Node.js web application framework, providing robust features for building web and mobile applications.

**Bcrypt:** A library for hashing passwords, ensuring secure storage and authentication of user credentials.

**Dotenv:** A module that loads environment variables from a .env file into process.env, simplifying configuration management.

**CORS:** A middleware to enable Cross-Origin Resource Sharing, allowing controlled access to resources from different domains.

**Multer:** A middleware for handling multipart/form-data, used for uploading files.

**Morgan:** A HTTP request logger middleware for Node.js, used for logging requests to the server.