Advanced Web Applications Project Documentation

---

1. Technology Choices

My web application uses MERN-Stack (MongoDB, Express.js, React.js, Node.js), which leverages a modern and robust technology stack to ensure optimal performance, scalability, and a seamless user experience. Below are the key technologies used in the development of my application. The technologies I chose are the ones showcased and used in the course and I have become familiar with them thus I decided to use them in this project.

MongoDB

MongoDB serves as my backend database, providing a flexible and scalable NoSQL solution. It efficiently stores and retrieves data, making it well-suited for my application's dynamic content.

Express.js

Express.js, a minimal and flexible Node.js web application framework, is employed to build the backend server. It simplifies the creation of robust APIs, handles HTTP requests, and streamlines middleware integration.

React.js

React.js, a powerful frontend library developed by Facebook, is the cornerstone of my user interface. Its component-based architecture facilitates the creation of interactive and dynamic user interfaces, ensuring a smooth and engaging experience.

Node.js

Node.js powers my backend server, allowing for asynchronous and event-driven execution. Its lightweight and efficient design makes it an ideal choice for building scalable network applications, ensuring optimal performance for my web application.

In addition, to secure my application and protect user data, I have implemented JWT (JSON Web Tokens) authentication. JWT provides a secure and efficient way to authenticate users and transmit information between parties. Furthermore, I utilize Local web storage to store authentication tokens, ensuring seamless and secure access to authorized resources while maintaining user privacy and security.

2. Installation Guidelines

To set up the development environment and run the application locally, follow these installation guidelines. (node v18.18.2)

Download the Repository

- Clone the repository using “git clone https://github.com/EricLUT20/MERN-Project”
    - Alternatively, you can download the project repository as a ZIP file from https://github.com/EricLUT20/MERN-Project.
    - Extract the ZIP file to a location of your choice.

Install Dependencies

- Navigate to the MERN-Project or MERN-Project-main (depending on how you downloaded the project)
- Run npm install in the “MERN-Project” or “MERN-Project-main” (root) directory to install the required dependencies for both the front and back end.

Configure MongoDB

- Install and configure MongoDB on your machine if you haven't already.
- Ensure that the server can connect to MongoDB at mongodb://localhost:27017/mern-project.

Start the Application

- Navigate to the MERN-Project or MERN-Project-main (depending on how you downloaded the project)
- In the “MERN-Project” or “MERN-Project-main” (root) directory run “npm start” to start both the client for the front end and the server for the back end.
    - Alternatively, you can open two terminals, one for the client and one for the server.
    - In the client terminal, navigate to the client directory and run “npm start” to start the frontend development server.
    - In the server terminal, navigate to the server directory and run npm start to start the backend server.

Access the Application

- Open a web browser and enter localhost:3000 in the address bar to access the application's front end.
- The application's front end should now be visible, allowing you to interact with its features.

Server Access

- Ensure the backend server is running on localhost:5000.

- The backend server provides the necessary API endpoints for the frontend.

---

3. User Manual

The user manual provides detailed instructions on using the MERN Dating App. Whether you're a new user looking to register or an existing user exploring the features, this guide will help you navigate through the application seamlessly.

Registration and Login

- Register a New Account:
    - Open up localhost:3000 in your preferred web browser.
        - Unauthorized users will be automatically redirected to the login page.
    - Navigate to the application's registration page.
    - Provide the required information, including a valid email address and password.
    - Click the "Register" button to create a new account.
- Login to Your Account:
    - Access the login page of the application.
    - Enter your registered email and password.
    - Click the "Login" button to sign in.
    - Now you can access all of the private/authorized pages (home, messages, profile, and user profiles)

Application Features

- Header Navigation:
    - At the top of every page (except for registration and login), you'll find a navigation header.
    - Use the header to move between different sections of the application.
    - The available navigation options include:
        - Home: Directs you to the main home page.
        - Messages: This takes you to the messages section to interact with your matches.
        - Profile: Allows you to view and edit your profile information.
        - Logout: Logs you out of the application and redirects you back to the login page.
- User Interaction (Home page):
    - Upon logging in, users are directed to the home page.
    - Authenticated users can view other users on the home page and express preferences by liking or disliking them.
    - You can also view other users' profiles by clicking on their names on the match card.

        - If you like a user and they have already liked you back you will be prompted to start chatting immediately and redirected to the “messages” page with the liked user’s chat opened.
- Chat Functionality (Messages page):
    - Authenticated users who have mutually liked each other can engage in two-way chats on the “messages” page.
    - Your matches will be displayed on the left from which you can select a match you want to chat with.
    - You can send and receive messages as well as view your previous messages between you and your matches with the timestamps.
    - You can filter messages by a keyword to find specific messages you want.
- Profile Management (Profile page):
    - Authenticated users can update their profiles such as name, title, and bio.
    - Optionally, upload images to enhance their profiles.
- Others pages:
    - Other user’s pages, which can be accessed by clicking on their name on the home page.
    - Registration page and Login page.

Responsive Design

- Cross-Device Compatibility:
    - The application is designed to be accessible on both mobile devices and desktop browsers.
- Materialize Integration:
    - The responsive design leverages the capabilities of Materialize for an optimal user experience.

---

4. Features Implemented

I have implemented all of the required mandatory features as well as multiple optional features. Considering all of the features I have implemented in this project, I kindly request a total of 41 points for this project. Below is a list of my implemented features throughout the project with a reasonable justification and explanation for each. Thank you for your consideration.

Feature & Justification	**Points**
Feature: Basic features (as stated in the previous chapter) with well-written documentation **Justification:** I implemented all mandatory features with well-written documentation.	25
Feature: Utilization of a frontside framework, such as React, but you can also use Angular, Vue, or some other	5

Justification: I implemented the client in the React frontside framework.	
Feature: One can swipe left or right to dislike or like the profile **Justification:** I implemented a swiping feature similar to Tinder using React Swipeable with a swiping animation as well using CSS. (react-swipeable - npm (npmjs.com)). You can swipe the card to the right to “Like” the user and swipe to the left to “Pass/Dislike” the user.	2
Feature: Use of a pager when there are more than 10 chats available **Justification:** I implemented a feature where there is a pager in messages meaning that when there are more than 10 chats and new pages are added and you click on different pages to see other users and start chatting with them when you want.	2
Feature: If match is being found the UI gives option to start chat immediately **Justification:** I implemented the UI option to start chatting when a match is found. When you like another user and the liked user has already liked you it will prompt you to a UI option to start chatting now by pressing the green round “arrow” button on the right.	2
Feature: User can click username and see user profile page where name, register date, (user picture), and user bio is listed **Justification:** I implemented a feature, where you can click the username of a user on the home page or messages to see the user profile page with name, registration date, title, and bio.	2
Feature: Provide a search that can filter out only those messages that have the searched keyword **Justification:** I implemented a search filter feature in messages so that the user can type “Hello” for example and click the “Filter” button next to send and it will find all messages between users where there is “Hello” included somewhere in the message.	3