1. **Project Title** : VoteWise

**Project Description :**

## **Case Study Overview**

**1. Introduction:**

The project involves the development of a web application designed for voting purposes. The application encompasses functionalities catering to both voters and candidates, facilitating user authentication, candidate manifesto viewing, and voting procedures.

**2. User Authentication:**

Implemented a robust user authentication system comprising login and signup functionalities for both voters and candidates.

Employed secure authentication techniques such as password hashing to ensure user data integrity and confidentiality**.**

**3. Voter Functionality:**

Upon successful login, voters gain access to view the manifestos of all candidates.

Developed a dedicated page allowing voters to peruse candidate profiles and their respective manifestos.

Integrated a seamless voting mechanism enabling voters to cast their ballots for their preferred candidate.

**4. Candidate Functionality:**

Candidates, upon logging in, are granted access to their individual profiles.

Provided candidates with the capability to modify their profiles, encompassing personal information and manifesto details.

**5. User Interface:**

Developed an intuitive and user-friendly interface catering to the needs of both voters and candidates.

Ensured responsiveness across various devices to enhance user accessibility and experience.

**Technology Stack Used**:

**1. MongoDB**:

MongoDB serves as the primary database management system for storing user information, candidate profiles, and voting data. Its flexible document-based structure aligns well with the dynamic nature of the application.

**2. Express.js:**

Express.js is utilized as the backend web application framework to handle server-side logic and routing. It provides a robust set of features for building RESTful APIs and handling HTTP requests.

**3. React.js:**

React.js forms the frontend user interface library, offering a component-based architecture for building interactive and dynamic user interfaces. Its declarative approach enhances code maintainability and reusability.

**4. Node.js:**

Node.js serves as the server-side runtime environment, facilitating the execution of JavaScript code outside the browser. It enables efficient handling of concurrent requests and asynchronous I/O operations.

**5. JSON Web Token (JWT):**

JWT is employed for user authentication and authorization purposes. It enables the generation of secure tokens containing user claims, which are validated to grant access to protected resources within the application.

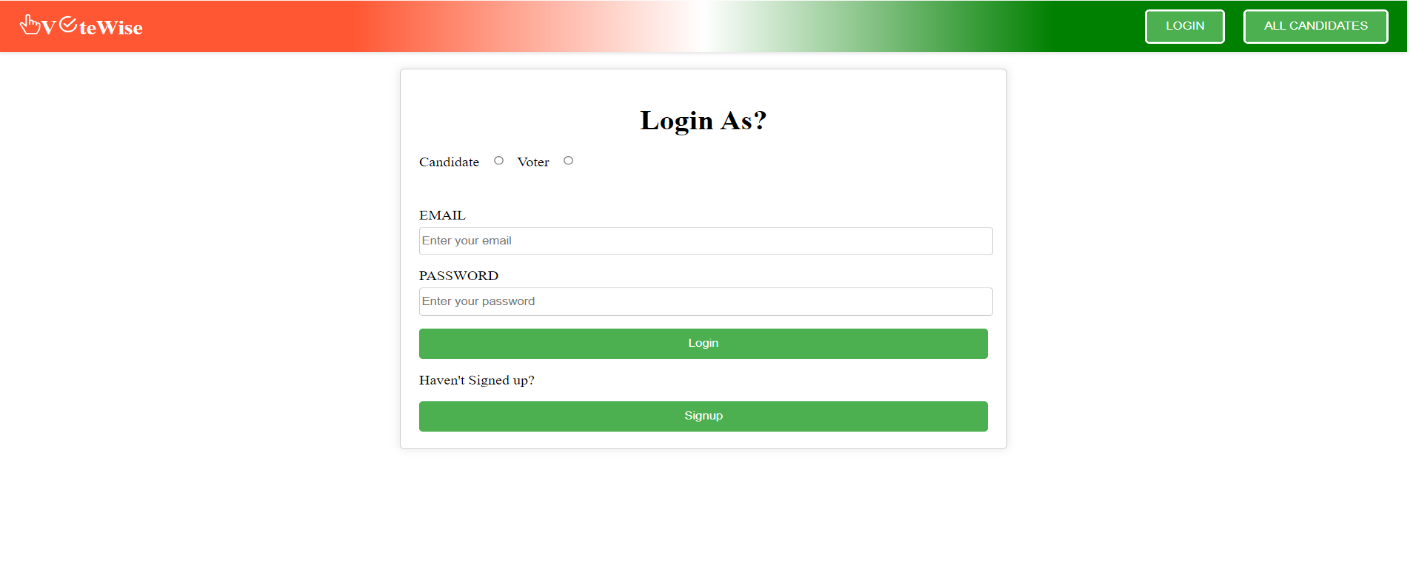
**6. Mongoose:**

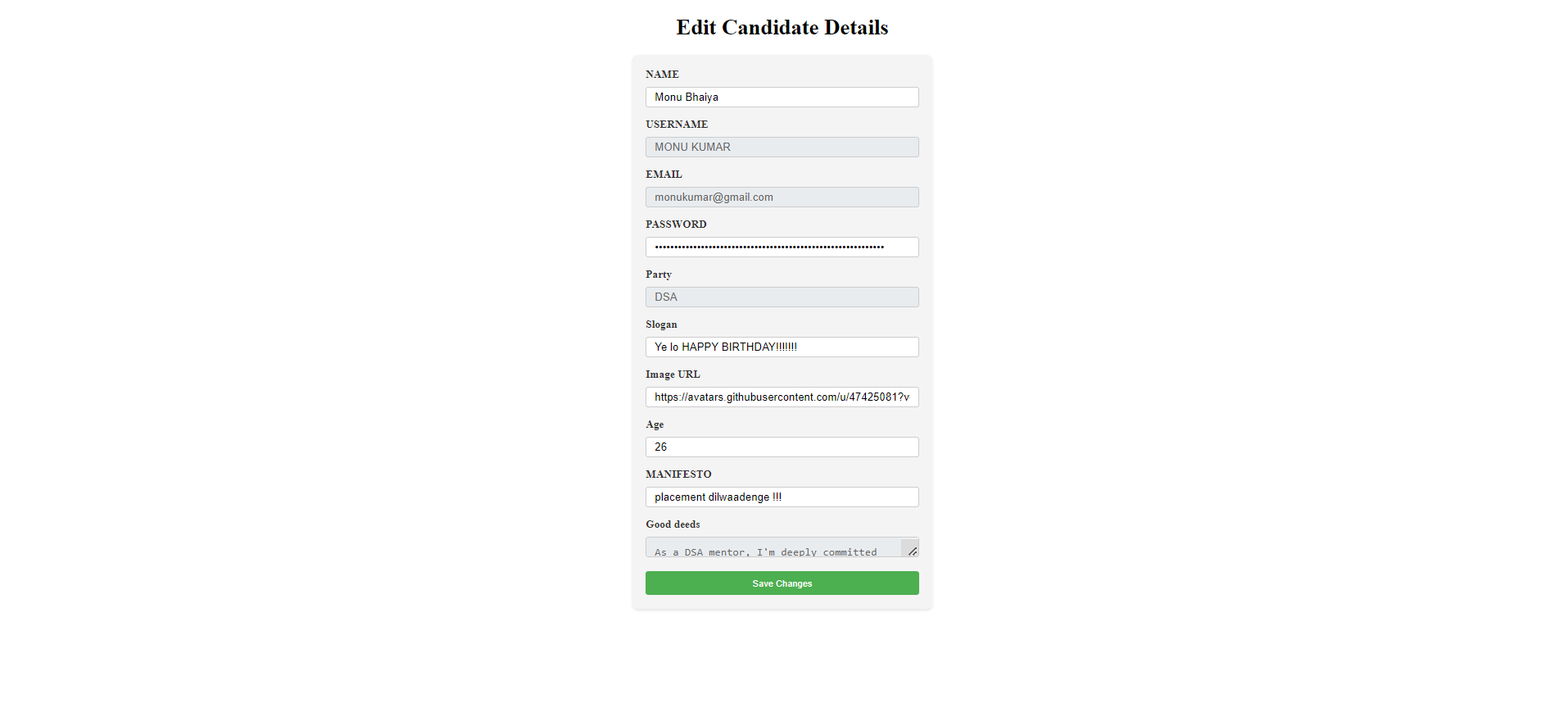
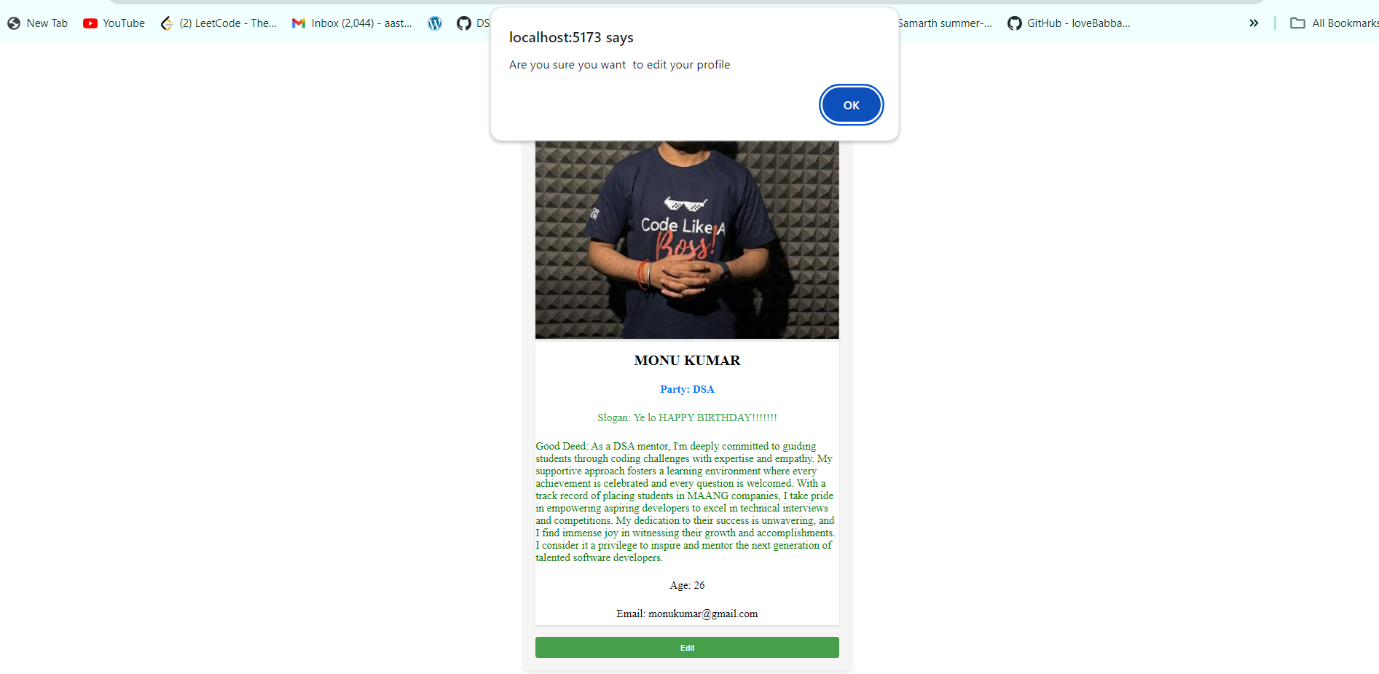
Mongoose is utilized as an Object Data Modeling (ODM) library for MongoDB, providing a straightforward mechanism for interacting with the database through JavaScript objects. It simplifies data validation, schema definition, and query execution.

**Key Challenges**

* Consumer Segmentation: Utilize sales data to categorize consumers effectively.
* Financing Solutions: Develop tailored financing options based on data-driven insights.
* Decision Support: Provide analytical support to determine the likelihood of purchase based on car attributes.

**Authentication:**





**Particular Voter**



