Dhirubhai Ambani Institute of Information and Communication Technology

PC-641 M.Sc. (IT) WINTER INTERNSHIP PROJECT REPORT 2023

Student Details
Name: Vraj Patel
Batch : MSc IT 2022 – 2024
Student ID : 202212078

Internship Details
On – Campus
Mentor: Prof. Lavneet Singh

INDEX

Contents

INTRODUCTION

During my recent internship, I had the privilege of working under the guidance of Professor Lavneet Singh, where I was immersed in the development of a Stack Overflow clone using the MERN stack (MongoDB, Express.js, React.js, Node.js). This project provided me with invaluable experience in full-stack development practices.

The internship spanned four months, divided into two phases. In the initial two months, I focused on acquiring the technical skills and knowledge essential for executing the Stack Overflow clone project. This learning phase involved deep dives into various technologies, including MongoDB for database management, Express.js for backend development, React.js for frontend development, and Node.js for server-side scripting. Additionally, I gained expertise in Firebase for implementing authentication and authorization functionalities.

Throughout this phase, I familiarized myself with version control using Git and GitHub, utilized IDEs like Visual Studio Code and WebStorm for coding, and engaged in API testing using tools like Postman.

With a solid foundation, I transitioned into the development phase of the Stack Overflow clone project in the subsequent two months. During this period, I applied my acquired skills and knowledge to build a comprehensive platform resembling Stack Overflow.

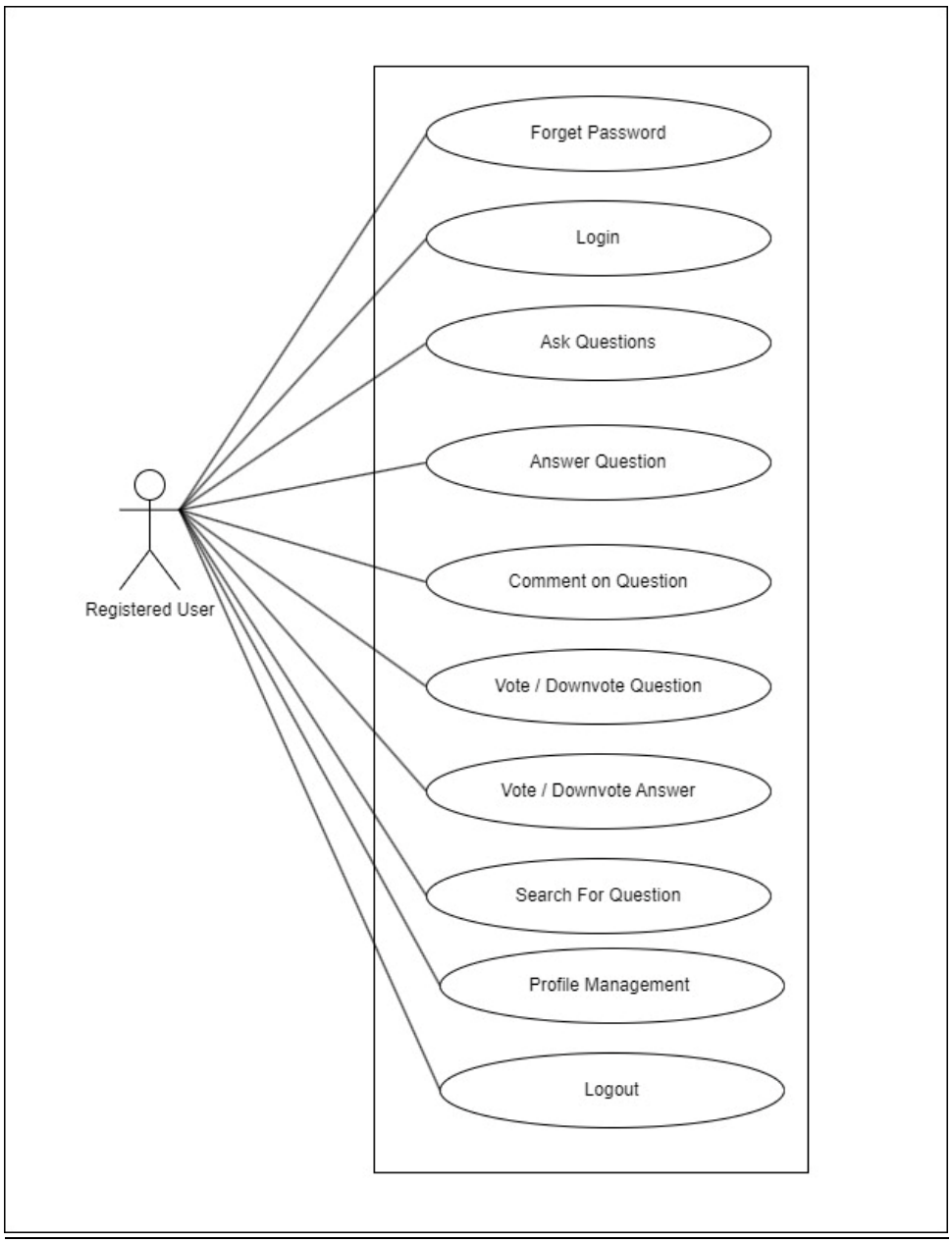
The Stack Overflow clone aimed to provide users with a centralized platform for asking and answering technical questions, sharing knowledge, and collaborating with fellow developers. It featured functionalities tailored to meet the diverse needs of developers, including user authentication, question posting, answer submission, voting, commenting, and searching.

Overall, my internship experience enhanced my technical proficiency and provided valuable insights into the real-world application of MERN stack technologies and Firebase authentication in developing scalable and user-friendly web applications.

SCOPE STATEMENT

The Stack Overflow clone project aims to replicate the core functionalities of the original platform, focusing on user authentication, question and answer functionality, tagging, voting, search, user profiles, and responsive design. Using Firebase for authentication, the system will provide a secure environment for users to ask and answer questions, tag topics, vote on content, search for information, view user profiles, and access the platform across different devices.

DETAILED USE CASE THAT HAVE BEEN IMPLEMENTED



PROJECT CONTRIBUTION

- Standard Functionality

  - Registration

    - Developed user registration features allowing new users to create accounts on the platform.
    - Implemented form validation and error handling to ensure the accuracy and completeness of registration details.

  - Login

    - Created login functionalities enabling users to authenticate themselves and access their accounts securely.
    - Utilized Firebase authentication flows for seamless login experiences across devices.

  - Password Recovery

    - Implemented a password recovery process allowing users to reset their passwords in case of forgotten credentials.
      Utilized email verification and sent a link to reset password. I have used Firebase to implement this functionality.

- User Module

1. Question Posting

    - Users can initiate discussions and seek assistance by posting questions related to various topics.
    - Created intuitive interfaces enabling users to input question titles and descriptions.
    - Integrated rich text editing functionalities using Quill.js to facilitate detailed and well-formatted question submissions.
    - Implemented validation mechanisms to ensure that questions are accurately and completely filled before submission.

2. Answer Submission

    - Users can contribute their insights and knowledge by providing answers to posted questions.
    - Designed user-friendly interfaces allowing users to compose and submit answers efficiently.
    - Utilized Quill.js for rich text editing capabilities, enabling users to format their answers effectively.
    - Implemented backend logic to handle answer submissions securely and efficiently, ensuring that answers are stored and displayed accurately.

3. Commenting System

    - Users can engage in discussions and provide feedback by adding comments to questions and answers.
    - Developed intuitive interfaces for users to input and submit comments seamlessly.
    - Implemented real-time comment updates using WebSocket technology, allowing users to view and respond to comments without refreshing the page.
    - Utilized MongoDB to store comment data securely, ensuring that comments are associated with the respective questions and answers.

4. Voting Mechanism

    - Users can express their opinions and vote on the helpfulness of questions and answers.
    - Designed interactive voting buttons allowing users to upvote or downvote questions and answers based on their relevance and usefulness.
    - Utilized MongoDB to store and manage voting data securely, ensuring that user votes are accurately recorded and reflected in the content ranking.

- Implemented algorithms to calculate and display the overall score of questions and answers based on user votes, facilitating community-driven content curation.

5. Search Functionality

   - Users can easily discover relevant content by searching for questions based on keywords or tags.
   - Integrated a robust search functionality enabling users to enter search queries and retrieve relevant results in real-time.
   - Utilized MongoDB text indexes and search algorithms to analyse and match search queries with question titles, descriptions, and tags.
   - Implemented autocomplete and suggestion features to assist users in refining their search queries and discovering related topics more efficiently.

6. Profile Management

   - Users can manage their personal information and preferences within the platform.
   - Created user-friendly interfaces allowing users to update their profile details, including profile pictures, display names, and contact information.
   - Implemented CRUD operations to handle profile data securely, ensuring that user updates are accurately reflected in the system.
   - Integrated authentication and authorization mechanisms to protect user profiles and ensure that only authorized users can access and modify their information.

TOOLS, TECHNOLOGY, API'S AND LIBRARY USED

1. MongoDB : MongoDB is a NoSQL database used to store and manage project data in a flexible and scalable manner.

2. Express.js : Express.js is a web application framework for Node.js, providing a robust wset of features for building APIs and web applications.

3. React.js : React.js is a JavaScript library for building user interfaces, allowing for the creation of interactive and dynamic front-end components.

4. Node.js : Node.js is a runtime environment that enables the execution of JavaScript code on the server-side, facilitating the development of scalable and high-performance web applications.

5. Firebase Authentication : Firebase Authentication is a service provided by Google that allows for easy integration of user authentication and authorization functionalities into web applications.

6. Firebase Storage : Firebase Storage is a service provided by Google that allows for easy integration to upload image and get NetworkURL.

7. Firebase Firestore : Firebase firestore is a service provided by Google that allows for easy integration to add data to firebase and manage them.

8. Quill.js : Quill.js is a rich text editor library used to enhance the user experience when creating and formatting textual content within the application.

9. Axios : Axios is a promise-based HTTP client for making AJAX requests to the server, facilitating seamless communication between the front-end and back-end of the application.

10. React Router : React Router is a routing library for React.js applications, enabling the creation of dynamic and navigable user interfaces with multiple views and pages.

11. React-Quill : React-Quill is a React.js component wrapper for Quill.js, allowing for easy integration of rich text editing capabilities into React.js applications.

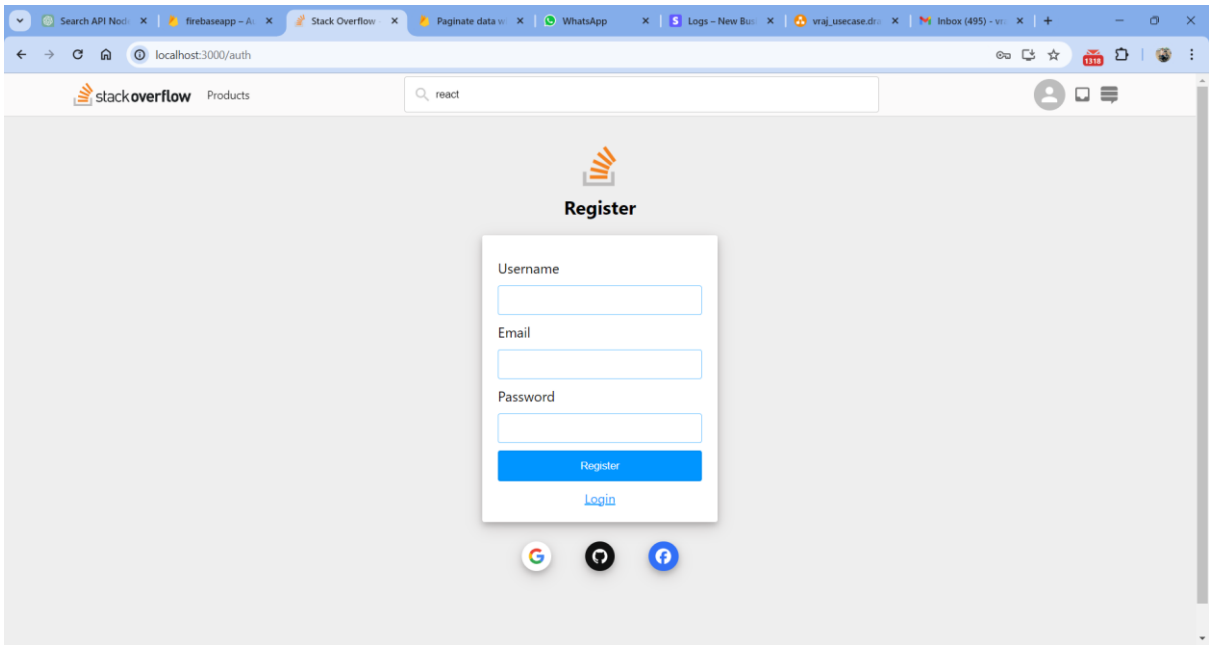
12. React Tag Input Component : React Tag Input Component is a customizable component used to create and manage tags within the application, enhancing the organization and categorization of content.

13. Material-UI : Material-UI is a popular React UI framework that provides pre-designed components and themes for building visually appealing and responsive user interfaces.

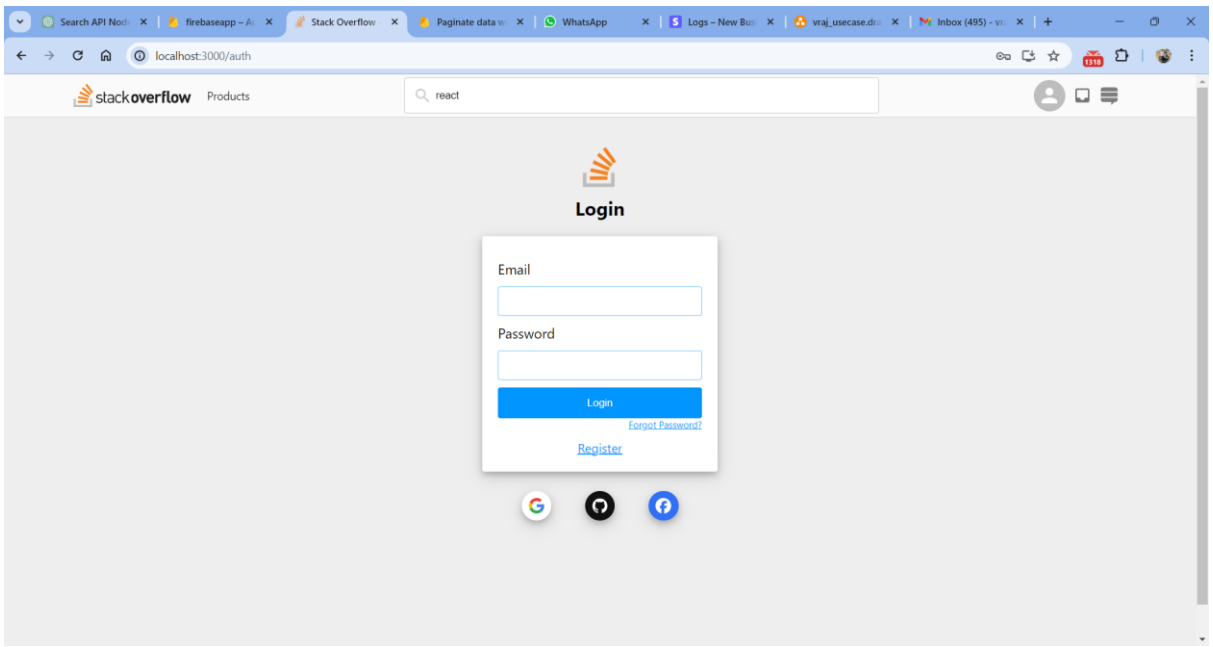
14. React Redux : React Redux is a state management library for React.js applications, enabling the centralized management of application state and data flow.

SNAPSHOTS

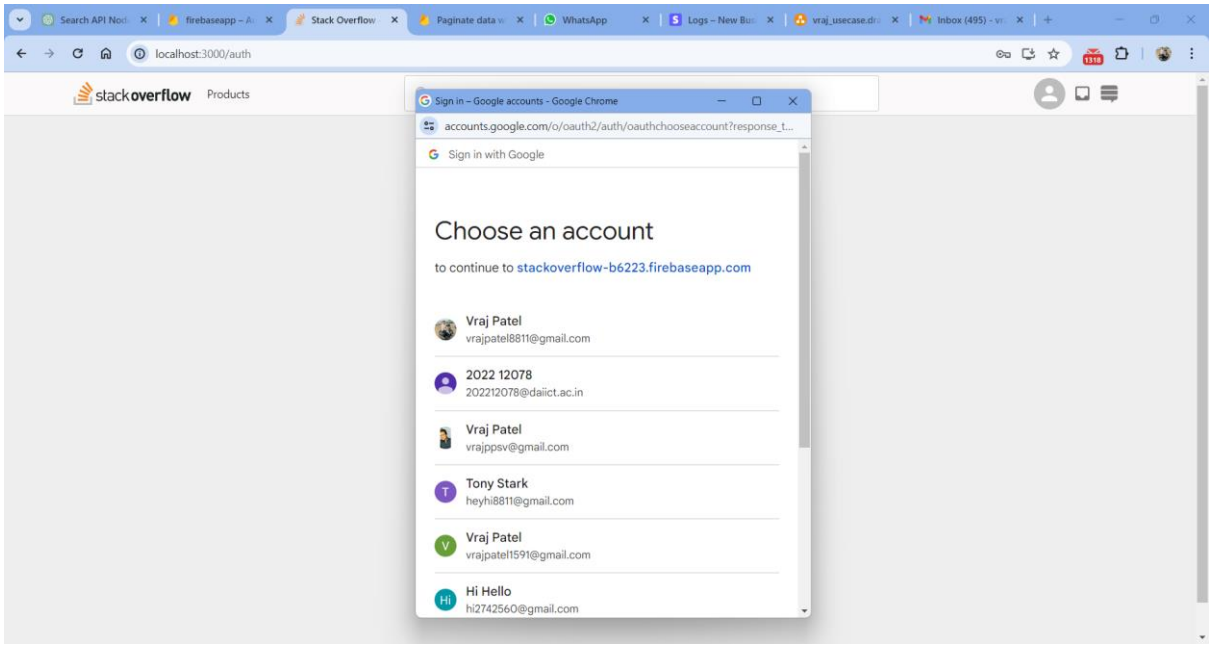
- Register



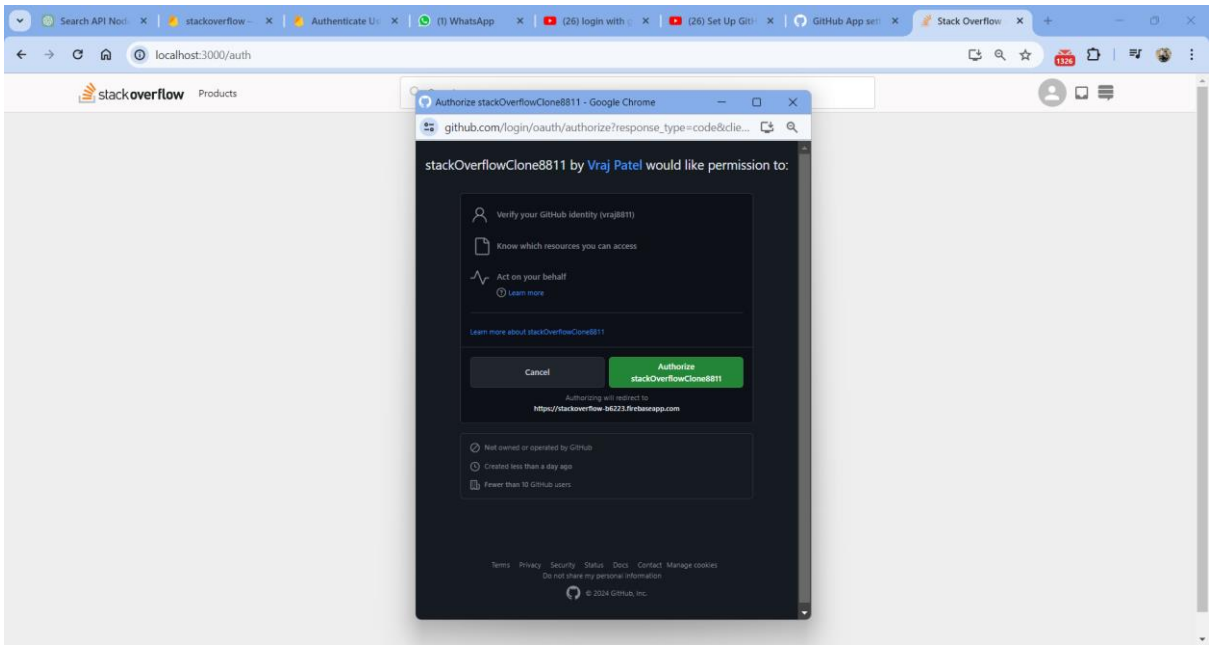
- Login



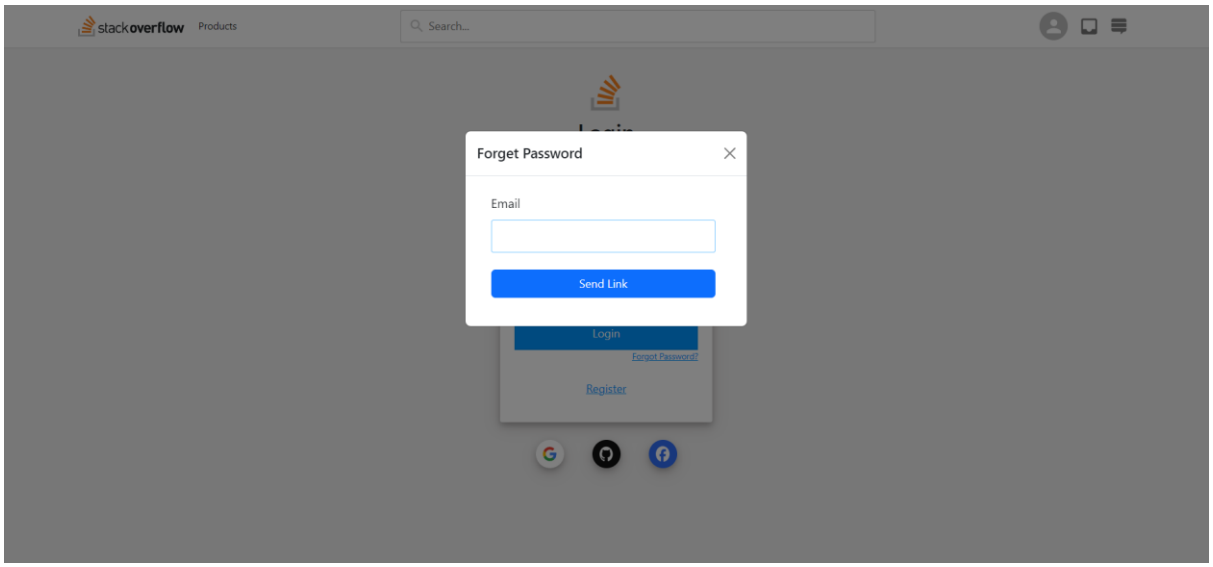
- Login With Google

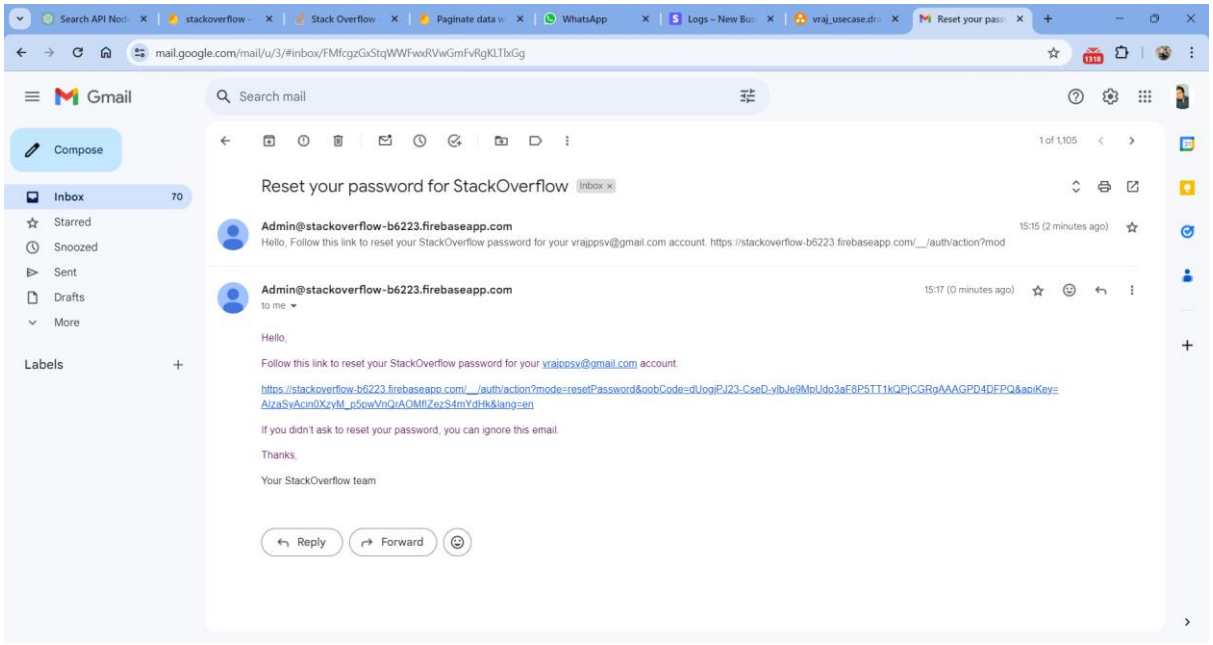


- Login with Github

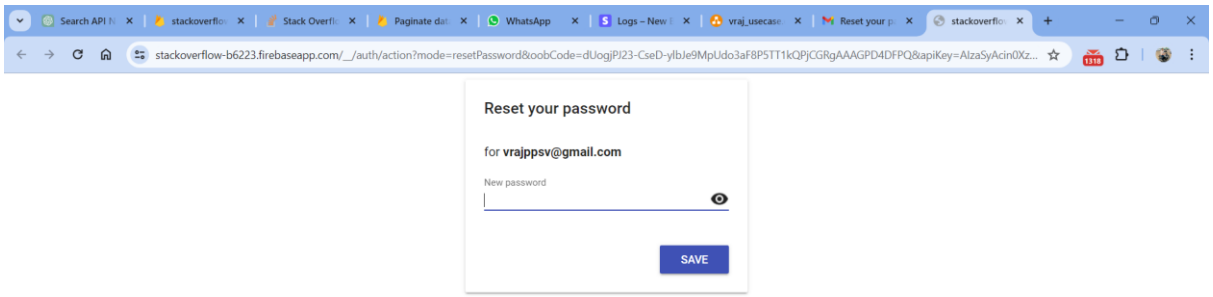


- Forget Password

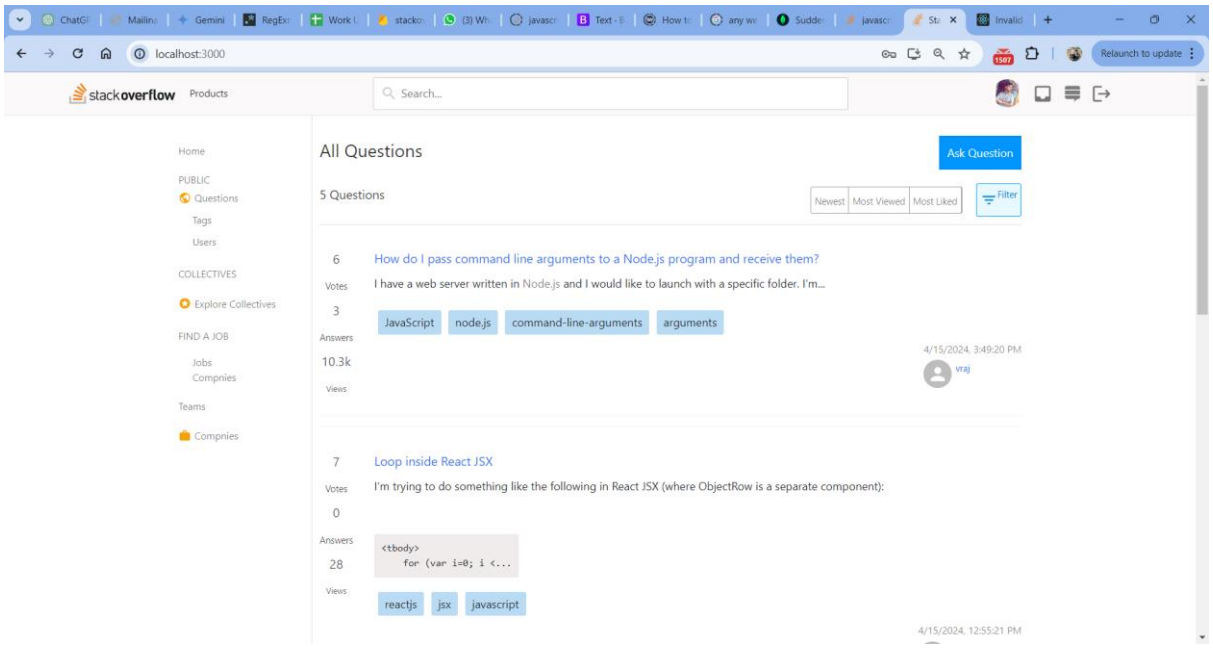




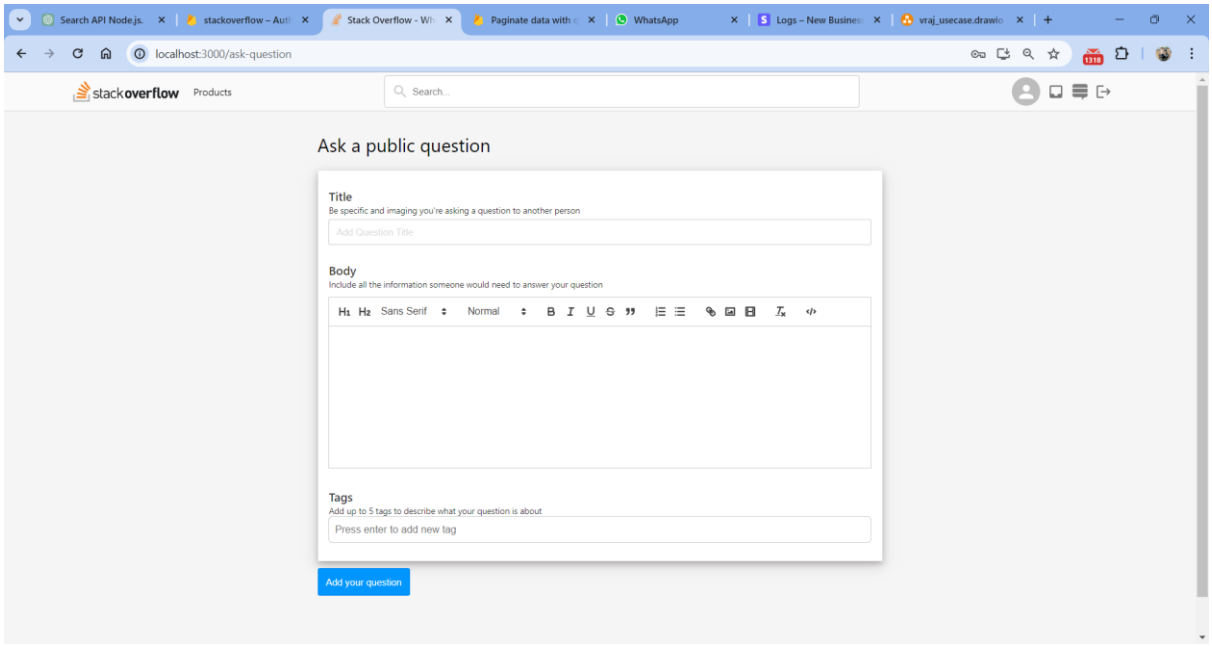
- Reset Password



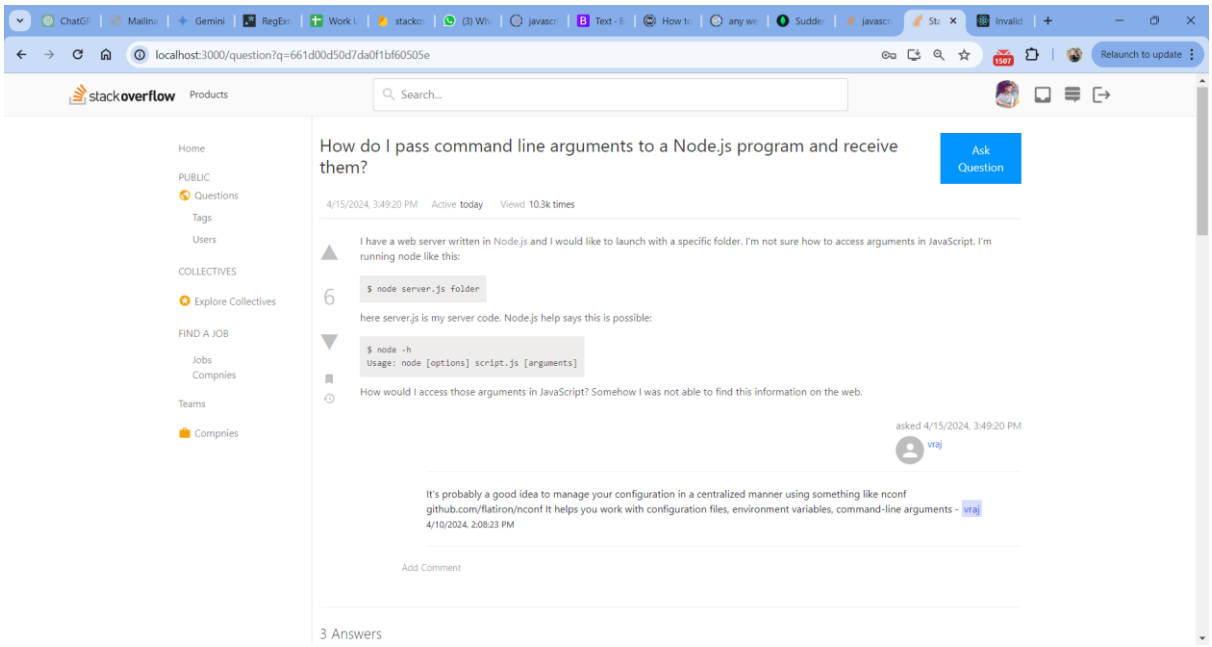
- Home Page



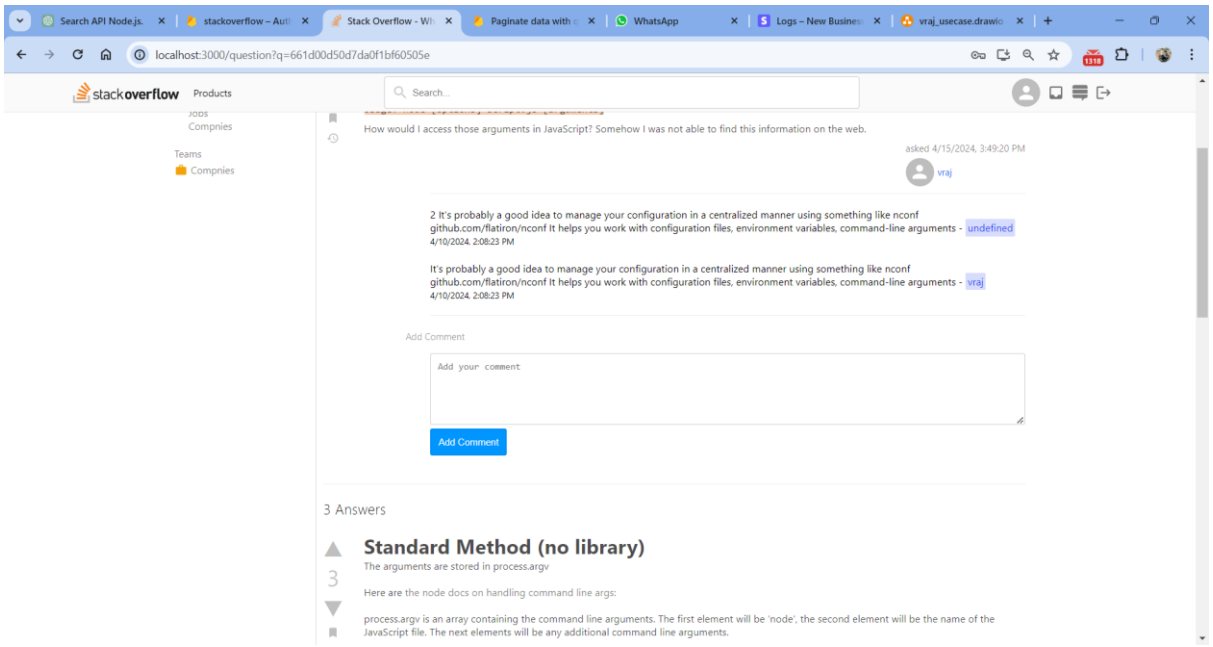
- Ask Question



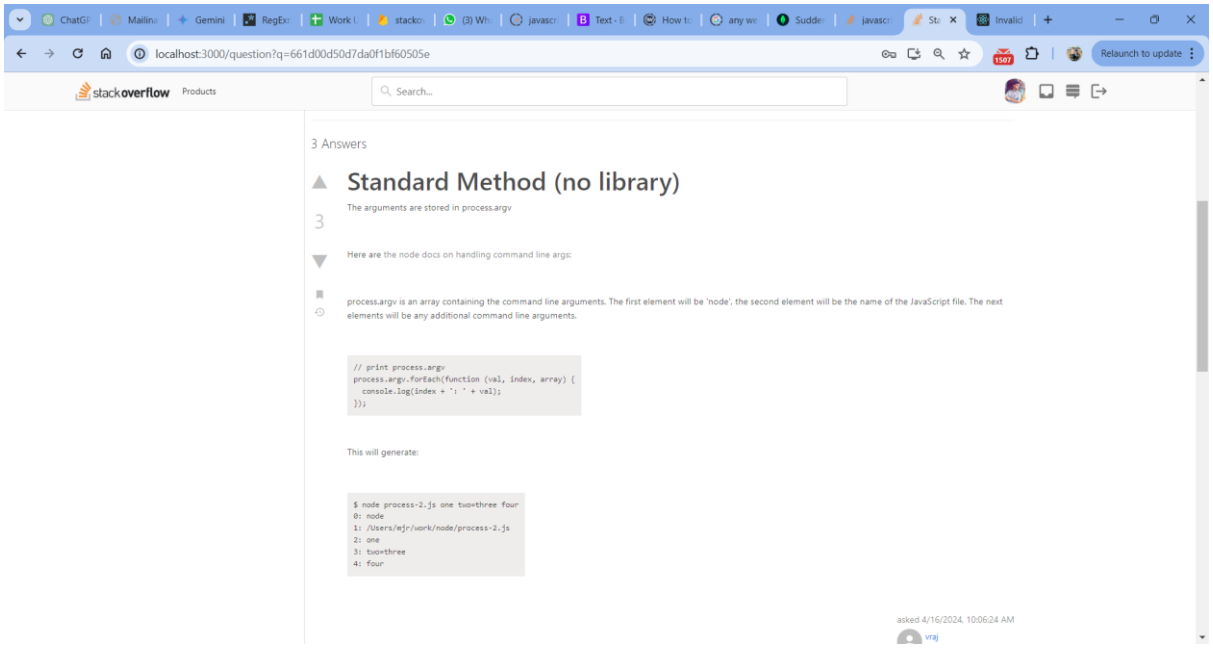
- Detailed View of Question



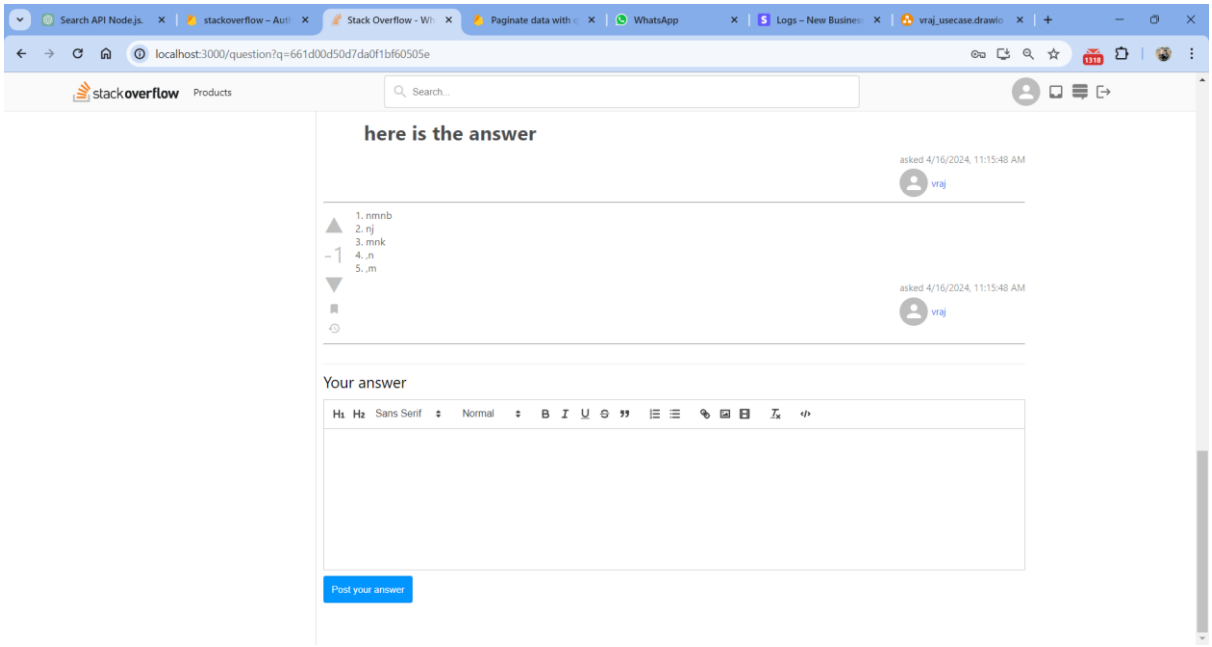
- Add Comments



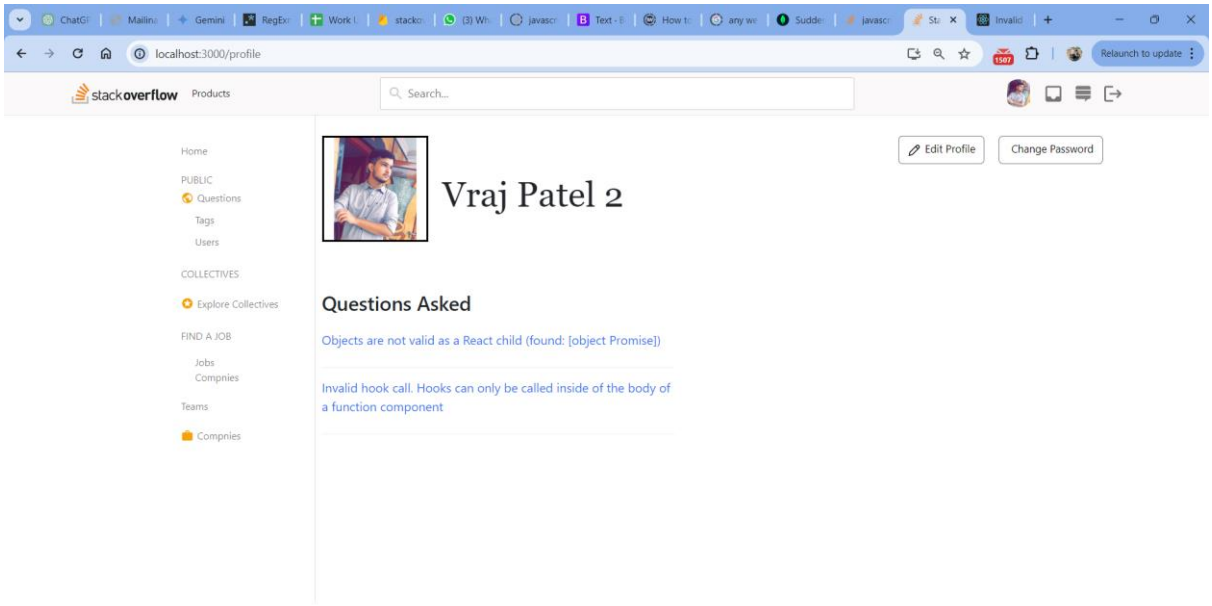
- Detailed View of answers

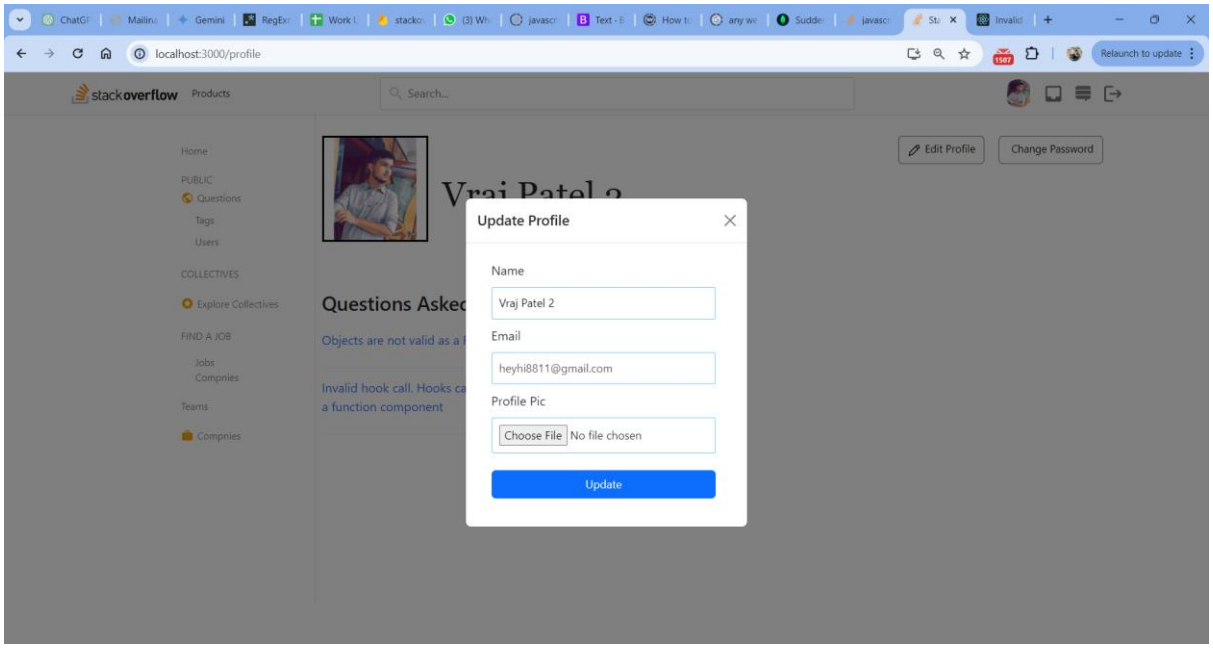


- Post Answer



- Edit profile




localhost:3000/profile
stackoverflow
Products
Search...
Home
PUBLIC
Questions
Tags
Users
COLLECTIVES
Explore Collectives
FIND A JOB
Jobs
Compnies
Teams
Compnies
Edit Profile
Change Password
Questions Asked
Update Profile
Name
Vraj Patel 2
Email
heyhi8811@gmail.com
Profile Pic
Choose File
No file chosen
Update

TESTING STRATEGY

- Unit Testing
    - Unit testing involves testing individual components or modules of the application in isolation to verify that they function as expected.
    - Test each function, method, or component independently to validate its behaviour and logic.

- Integration Testing
    - Integration testing focuses on testing the interaction between different modules or components to ensure they work together seamlessly.
    - Develop test cases to verify the integration points between frontend and backend systems, API endpoints, database connections, and third-party services.
    - Test scenarios such as data flow between frontend and backend, API request/response handling, and error handling during integration.

- End-to-End (E2E) Testing
    - End-to-End testing involves testing the entire application flow from start to finish to ensure all components function together as expected.
    - Test user journeys, critical workflows, and user interface interactions to validate the overall functionality and usability of the application.

LESSON LEARNED

During my internship, I acquired several key learnings that will shape my career as a software engineer:

- Enhanced understanding of development principles, encompassing server-side programming, database management, APIs, and system architecture, including rendering server-side pages. Additionally, gained insights into frontend development.
- Acquired practical experience in utilizing essential technologies such as MERN stack (MongoDB, Express.js, React.js, Node.js), Firebase for authentication, and other relevant tools and libraries.
- Developed strong problem-solving and debugging skills by addressing real-world challenges encountered during project development, thereby improving overall proficiency in software engineering.