

# REPORT

## A short 2-3 paragraph summary of how you built the application

To build the website, we used a combination of front-end and back-end technologies along with a database to store and manage the user data. For the front-end, we used HTML, CSS, and React JS to create the user interface of the website. This included designing and building the login page, create account page, home page, view profile page, and question details page. We also used React JS to handle user interactions and state management on the client-side.

For the back-end, we used Node JS and Express to build the server-side logic of the website. This included creating APIs to handle user authentication, account creation, data retrieval, and data storage. We also implemented server-side validation to ensure the security and integrity of the user data. To manage the user data, we used a Postgres database, which allowed us to store and retrieve user information, questions, and answers.

Overall, the website was built using a modern and scalable technology stack, which ensured a high level of performance and security. The use of front-end and back-end technologies enabled us to create a seamless user experience and handle complex interactions between the client and server.

## Overall System Architecture—

### Programming Languages used

**FRONTEND:** HTML, CSS, React JS

**BACKEND:** Node JS, Express, Postman, Cors

**DATABASE:** Postgres

## Webpages --

**Login page:** The login page is the first page a user sees when they visit a website or application

**Create Account page:** The create account page is where users can register for a new account on a website or application.

**Home page:** The home page is the main page of a website. It has navigation bar with links to view profile page and to logout. Here the user can post his question and search for answer for a given question.

**View Profile page:** The view profile page is where users can see their personal information and their id

**Question Details page:** The Question Details page is where user can see the answers to his question and post his answer to a given question.

## Frontend Components

**Login.js:** For the user to login to the website.

**CreateAccount.js:** For the new users to create accounts.

**Navbar.js:** From here user can navigate to his profile and logout of the website.

**DropDown.js:** Here the user can select tags for his website.

**ViewProfile.js:** Here user can see his profile picture, name, email, badges, the questions he contributed to.

**PostQuestion.js:** The user can post his question here.

**PostAnswer.js:** The user can post his answer for a given question here.

**QuestionDetail.js:** Here the user can see the answers to his question and delete and edit their questions and answers.

## Contributions

### Aakash

I am responsible for integrating the back-end APIs with the front-end components, ensuring that the web application was fully functional and responsive.

To achieve this, I collaborated closely with my team mates to understand the API endpoints and data structures, and then used a client-side HTTP library such to connect the front-end components to the APIs.

I implemented error handling and validation checks to ensure that the user input was properly validated and processed by the back-end APIs.

I designed and implemented the CSS for the website. Throughout the development process, I tested the CSS and components to ensure that they functioned correctly, were accessible to all users, and were compatible with website.

I am responsible for implementing an auto-search feature on the website, which allowed users to quickly find information based on partial or full display names.

## **Mukunda**

I am responsible for creating React JS for UI of the website, ensuring that each page was modular, maintainable, and scalable for achieving this, I collaborated closely with my team mates to ensure that the React JS accurately reflected the design and functionality of each page.

I followed best practices and conventions for React development, including creating reusable components, using states to manage data and behaviour, and ensuring that the components were properly encapsulated.

I am also responsible for designing data schema in the Postgres SQL database for the website to ensure that the database accurately reflected the user needs and requirements with UI.

I am responsible for implementing tag feature and search of the tags on the website, which allowed users to categorize and organize questions based on related topics or keywords.

## **Naresh**

I am responsible for designing and implementing the backend APIs for the website, working closely with the front-end developers to ensure that the APIs accurately reflected the user needs and requirements.

I made the login and create-account APIs to check if login data matches with database or insert data into it if a new account is created. For the autocomplete and search-questions APIs, I utilized indexing techniques to ensure fast and accurate results for the user.

The fetch-answers and view-profile APIs provided comprehensive information about a particular question or user, allowing the user to view relevant details and make informed decisions.

I implemented the post-question, post-answer, delete-post, edit-question, and edit-answer APIs to allow users to create, modify, and delete posts and answers as necessary.

Throughout the development process, I rigorously tested each API to ensure that it functioned correctly and was secure against attacks or errors through postman application.

I was responsible for designing and implementing the Postgres SQL database for the website. I created the necessary database tables, columns, and constraints to store user information, post data, and other relevant information.

As the project progressed, I updated the database schema to accommodate new features, changes in user requirements, and performance improvements.

## **Jaya Chandra**

I am involved in the decision-making process for selecting the programming languages and tools to be used for the website from my earlier course project experience.

I have created the starter code for backend and suggested suitable dependencies to use.

I collaborated with the development team to ensure that the chosen technologies were compatible for the website's requirements and helped in coordinating between frontend and backend.

I contributed in backend API templates and setting up cookies. I tested each API to ensure that it functioned correctly and was secure against attacks or errors through postman application.