Summary of the Application:

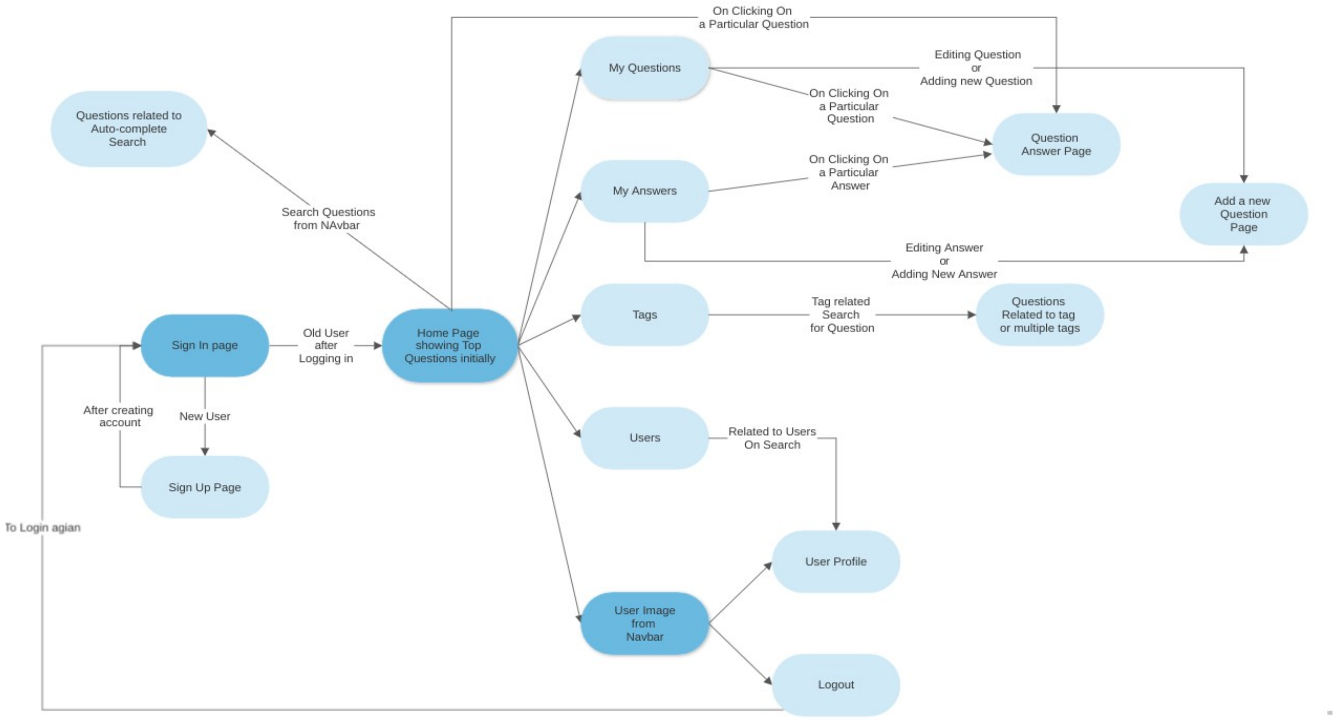
A database named “cqadb” is created with schema given in “backend/our_schema.sql” and sample data is populated. Two servers are created one server running backend part (ExpressJS) and another server running frontend part (ReactJS). Starting part of whole application is Login page (“Login.js for frontend adn login.js for backend”). There are 3 components of login page Sign-up,Sign-in,Forgot Password. For Sign-up part the username and password entered to create new user is taken to backend and inserted into database accordingly. For Sign in part when a user enters his username and password it is taken to backend and verfied accordingly. For forgot password component is user forgots his password then the prompt is made asking for username and the passsword will be reset to his username. If a user checks on remember me then cookies are made to remeber the user login session so he need not login again untill the user presses on logout button else he is redirected to login page when he logs out or the page is closed.

When a user is logged into the homepage 5 tabs along with profile picture can be seen. In Every page in frontend the user can sort both ascending and descending based on up_votes and time.When a user wishes to edit his profile he can press “Edit Profile” and modify the details accordingly (“Profile.js for frontend and “profile.js” for backend) . The remaining 5 components are “Top Questions”,“My Questions”,“Users”,“Tags”,”My_answers”. For the Top Questions (“Top_Questions.js” for frontend and “top_questions.js” for backend) sql query is written to rank all questions in ascending order based on question reputation and displayed accordingly. Same thing applies for My Questions and My Answers except that sql queries is varied accordingly. The user can view his/her posted questions and answers. When ever an user clicks on any question he wants to view he is redirected to question page (“Quest_Ans.js” for frontend and “quest_ans.js” for backend).

For each Question the user wants a view the requested question and the answer posts can be seen below. The user can up_vote, down_vote,comment and edit the question or answer. And also user can see when the posts are posted and who posted them. User can add his answer also. All the front end for this is built on “Quest_Ans.js” reactJS and frontend and the back end can be seen on “quest_ans.js” and the several queries are written to keep the data consistent like last accessed date,last edited date,number of views,Accepting answers. A user can ask any question to whole CQA website by clicking button “Ask Question” which is visible on every page. A user can search questions based on multple tags (“Tags.js” for frontend and “tags.js” for backend) queries are written to return questions which contain atleast one selected tag. User can see different users present in CQA website in Users page (“Users.js” for front end and “users.js” for backend) they can view their profile and their details. Auto Searching is enabled every where on whole website to make easier searching. User can also search different posts on search bar kept in Navbar above and view the post accordingly.

Overall system architecture :

This flow chart will show



Programming languages used :

(1) For Frontend :

Html
Css
Bootstrap 5
React Js – Axios

(2) For Backend :

Node Js
Express Js
My Sql - DataBase

Contribution of each group member :

(1) Y Govinda Rohith :

Integration and linking of all pages
Login Page Backend
Question-Answer Page both Frontend and Backend
Sort by time, name, Upvotes in all Pages

(2) Urla Smm Teja :

Add a new Question Page Backend
Tags Page Backend
Users Page Backend
Autocomplete Search and cookies

(3) Pettugadi Pranav :

User Profile Page Backend
Question-Answer Page Backend
Top Questions/My Questions / My Answers

(4) Rajulapati Bhargava Ram :

Login Page Frontend
User Profile Page Frontend
Add a new Question Page Frontend
Tags / Users Page Frontend
Navbar / Side Navbar