
**A report**
**on**

# "JUQuery"

**Submitted in partial fulfilment for the award of the degree of**

## BACHELOR OF TECHNOLOGY
## IN
## SOFTWARE ENGINEERING

**Submitted by**

**Souvik Kar**
**18BTRSE026**

**R.Srisarvesh**
**18BTRSE027**

**T Aswin Barath**
**18BTRSE031**

**Vyshnav Raj**
**18BTRSE033**

**Under the guidance of**
**Prof. Sujatha K**

**Faculty of Engineering and Technology**
**Jain (Deemed-To-Be) University**
**Department of Computer Science and Engineering**
Jain Global Campus, Kanakapura Taluk - 562112
Ramanagara District, Karnataka, India

**2020-2021**


Department of Computer Science & Engineering
Jain Global campus, Kanakapura Taluk
Ramanagara District - 562112 Karnataka, India

# CERTIFICATE

This is to certify that the project work titled **"JUQuery"** is carried out by **Souvik Kar (18BTRSE026), R.Srisarvesh (18BTRSE027), T Aswin Barath (18BTRSE031), Vyshnav Raj (18BTRSE033)**, a bonafide students of Bachelor of Technology at the Faculty of Engineering and Technology, Jain (Deemed-to-be) University, Bangalore in partial fulfilment for the award of degree Bachelor of Technology in **Software Engineering**, during the Academic year 2020-2021.

| | | |
|---|---|---|
| **Prof. Sujatha K**<br>**Internal Guide**<br>Assistant Professor<br>Dept. of CSE,<br>Faculty of Engineering and Technology,<br>Jain (Deemed-to-be)<br>University | **Dr. Kuldeep Sharma**<br><br>Head of the Department,<br>Dept. of CSE,<br>Faculty of Engineering and Technology, Jain<br>(Deemed-to-be)<br>University | **Dr. Hariprasad S A**<br><br>Director,<br>Faculty of Engineering<br>And<br>Technology,<br>Jain (Deemed-to-be)<br>University |

Name of the Examiner Signature of Examiner

1.

2.

# DECLARATION

We, **Souvik Kar (18BTRSE026), R.Srisarvesh (18BTRSE027), T Aswin Barath (18BTRSE031), Vyshnav Raj (18BTRSE033),** are students of sixth semester B. Tech in **Software Engineering,** at Faculty of Engineering and Technology, **Jain (Deemed-To-Be) University,** hereby declare that the project work titled **"JUQuery"** has been carried out by us and submitted in partial fulfilment for the award of degree in **Bachelor of Technology in Software Engineering** during the academic year **2020-2021.** Further, the matter presented in the project has not been submitted previously by anybody for the award of any degree or any diploma to any other University, to the best of our knowledge and faith.

Signature

Souvik Kar
USN : 18BTRSE026

R. Srisarvesh
USN : 18BTRSE027

T Aswin Barath
USN : 18BTRSE031

Vyshnav Raj
USN : 18BTRSE033

Place : Bangalore
Date :

# ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of a large number of individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to **Faculty of Engineering and Technology, Jain (Deemed-to-be) University,** for providing us with a great opportunity to pursue our Bachelor's Degree in this institution.

In particular we would like to thank **Dr. Hariprasad S A, Director, Faculty of Engineering and Technology, Jain (Deemed-to-be University),** for his constant encouragement and expert advice.

It is a matter of immense pleasure to express our sincere thanks to **Dr. Kuldeep Sharma, H.O.D., Department of Computer Science and Engineering, Jain (Deemed-to-be University),** for providing right academic guidance that made our task possible.

It is a matter of immense pleasure to express our sincere thanks to **Mr. Santosh S., Program Coordinator, Dept. of Computer Science and Engineering (SE), Jain (Deemed-to-be University),** for providing right academic guidance that made our task possible.

We would like to thank our internal guide **Ms. Sujatha K, Dept. of Computer Science and Engineering, Jain (Deemed-to-be University),** for sparing her valuable time to extend help in every step of our project work, which paved the way for smooth progress and fruitful culmination of the project.

We would like to thank our Project Coordinator **Prof. Mr. Karthikeyan. S** and all the staff members of Computer Science and Engineering for their support.

We are also grateful to our family and friends who provided us with every requirement throughout the course.

We would like to thank one and all who directly or indirectly helped us in completing the Project work successfully.

Signature of Students

# TABLE OF CONTENTS

# ABSTRACT

The web development has undergone many changes and updates since its inception and people are opting in more for the web development. After the development of 'web development of that application', students are extensively making project of web development. This leaves the conventional and traditional methods of software development in a state of void. That said, some beginners and start-ups face difficulties since with recent upgrades the web development field is becoming complex. The Internet provides multiple vague solutions for entirely undirected questions on the same. Developers are confused even more so.

To answer this need, 'JUQuery' introduces a set of facilities which facilitates infrastructure of college. This improves students' knowledge about campus and at the same time it inform about any events is going on in campus.

# LIST OF FIGURES

# NOMENCLATURE USED

| | |
|---|---|
| Firebase | • Google Firebase Authentication<br>• Google Firebase Firestore Database<br>• Google Firebase Hosting |
| AGSD | Agile based Global Software Development |
| GUI | Graphical User Interface |
| QoS | Quality of Service |
| SDLC | Software Development LifeCycle |
| ASD | Agile Software Development |
| JS | JavaScript<br>• ReactJs<br>• ReduxJs |

# 1. INTRODUCTION

Nowadays ‘Question answer’ website has a huge impact on society and people especially students. To gain more knowledge on specific category or subject, a portal for posting queries and getting answers will be a good choice. Our application provides a chance for others to read queries, analyze them, and choose the true answer based on highest voting.

“JUQuery” web application is a way for everyone to express their ideas and queries on various topics like “technology, technical events, innovation, placements, sports, cultural events, transportations, hostel life, canteen, and many other things” happening in Jain University. It is an exclusive question answering web application for our university students, teachers, alumni, and for those who want to know about it. Each question answer is saved in our application frequently and they will be arranged in about 10 categories as of now. This portal will allow the users to engage with the presenter and the content of the presentation. The more engagement, the better the presentation.

**“When handled well, Q&A–questions and answers–becomes the most important element of a presentation.”** - Christopher Witt

## 1.1 Overview

In JUQuery application it continuously provides the platform for information sharing and studying. The more you ask doubts the more you learn .it is the environment for everyone can showcase their knowledge assets and that will use full for everyone who wants to know.

JUQuery allows you to engage in meaningful conversations with your target audience and establish your brand as an expert in your field. The ultimate goal of this website is just to spread the knowledge and to help the people who are seeking the knowledge.

For example a student from a rural area wants to take an admission from our college, he wants to know about the food items available in canteen. officially he can't ask that ; In these type of situation we are understanding the importance of our JUQuery application by using it he can add query then seniors or others who knows will answer his query and clarify his doubts. Everyone has to first register to our website for accessing the platform. Coming to the business area also it will be a success full project.

Asking questions or expressing doubts and requesting information leads to knowing completely about any topic that we are interested to know about. And in every Educational Institute, there is an actually different perspective on things that are there on the institute:

- Management Perspective
- Student Perspective

But students who are in the institute always want to know both perspectives. And only through the virtual way we can able to connect to all the students of the Institute and make them get to know completely about what they wanted to know.

"JUQuery" is a platform to know about the academic and non-academic environment, campus life of a student and faculty and student relationship in the institute. Any student who is part of the institute can have an account on this application and can post any questions and can answer any of the questions posted by other users.

## 1.2. Problem Definition.

JU Query application mainly focuses on promoting virtual interactions and help each and every student to know what is exactly going on the college campus. So our application is defined as software that allows any student to ask/answer any query, post any content, share notes and share their important speeches. Our system will be consist of one front end components and a backend component. The front end component is for students to view the Q&A, upload Q&A, share notes and register events. The backend component is to store and maintain the data of every student profile.

## 1.3 Objective(s)

"JUQuery" aims to deliver a guide which answers to the demand of a general yet centralized idea or a suggestion which could address and possibly solve the query about the various field of Jain University, making the overall work flow easy for the entire management and functionality teams. So this project although addresses issues which could not only help the student of Jain University but the other who wants to know about Jain University infrastructure.

This guide is crafted carefully to bring out the best adopted standard development technique which is Agile development and used it to smoothen the end users experience in posting the questions and answers and also see about already posted questions and answers without any confusion.

Our next work is to add chat option for it as a student and talk with their seniors to come to know about some untold queries. Also going to add event creation features on it.

## 1.4 Methodology

This guide proposes a step-by-step guide with a model case following which provides a solution for issues related to query about Jain University infrastructures. Here is an overview of problems faced and how its model provides solution to those problems:

- First and foremost, there is no standard application to know about Jain University's non-academic things/ events. This guide simplifies it by breaking down the major steps and combining issue prone steps with it. The model is a step by step process which eases the pain of self-searching unknown troubleshooting issues.
- There is always a growing confusion on where to get the non-academic details and infrastructure about any college. The guide's presented model, has included a specific ask query panel as one of its initial steps which is used to post question directly to question feed and will make it visible and answerable every users of this application. For instance, consider user needs to know a certain question-A of section AA then it will show the section-AA at top of the question with user details and timestamp and below of it, it will display the question.
- It also have an answer button in the rightmost of the question. After clicking on it, it will open a modal to post the answer. After posting answer by so many users, the latest answer will be displayed on top of answer feed.
- Being a user of the application they have authority to post and answer queries in ethical manner. They can sort out the section by clicking on the sidebar's options.
- To prepare this guide, various models like waterfall, spiral, V model etc. were studied and some flaws were highlighted that have been traditionally passed down by those models. Next, this guide makes sure Agile principles are implemented. This enabled the members working on the project to track and manage issues and sprints in real time and become more aware of Agile Global Software Development (AGSD) during the phase of 1 year lockdown due to the recent COVID-19 widespread. Even though none of them could physically meet, Agile Scrum principles allowed them to stay in sync and track progress with each element that were distributed amongst the development team members. So with everything being stated this guide and the project supports agile principles is completely in sync with it.

## 1.5 Hardware and Software used

Since the project in itself is whole and sole based on use of firebase platform, it is an acknowledgement as well as a sense of stating the obvious, that the hardware requirements are bare minimum.

**Have a look at the Hardware requirements for the project:**

- The Hardware is required to test and run things locally before committing changes to the main team branch on Firebase.
- All though not compulsion but each developer needs to own a Laptop/PC where he/she can work on.
- A bare minimum configuration of 2 GB RAM with 10+ GB of free disk space would be more than sufficient.
- Operating Systems are user preference. Whichever OS the user is comfortable with can be used.
- The Laptop/PC configurations aren't much stressed. Any configuration machine whether old or now heavy or lite doesn't matter. One needs it run code and open websites that's all.
- A good internet connection is required. A low latency connection on development machine is always welcomed and appreciated. However, since the project deals with Firebase and web applications, the software requirements seem to be more. Have a look at the Software requirements for the project.
- GitHub Software and local Git repository should be present in the machines.
- Knowledge on application development.
- Knowledge on developing applications.
- Text/Code Editor: Atom or Visual Studio Code
- Web Browser with scripting enabled.

**Here's the learning curve:**

- Front-End Design languages
- Knowledge about Firebase
- Interaction between Firebase-client
- Firebase query handling
- Command-Line
- Deployment Testing

# 2. Literature Survey

## 2.1 References for Literature Survey

### 2.1.1 Reference 1 - Research paper on SocialQ&A: An Online Social Network Based Question and Answer System

**Insights from the reference**

The primary objective of this paper is to improve the performance of Q&A systems by actively forwarding questions to users who are capable and willing to answer the questions. The research paper describes the design, architecture, algorithms, implementation of SocialQ&A (an online social network based Q&A system), and comprehensive large-scale simulation to evaluate SocialQ&A in comparison with other methods. The paper also emphasises on the implementation of a real prototype of SocialQ&A, and detailed analysis of the Q&A behavior of real users and questions from a small-scale real-world SocialQ&A system.

SocialQ&A leverages the social network properties of common-interest and mutual-trust friend relationships to identify an asker through friendship who is most likely to answer the question, and enhance the user's security. SocialQ&A also improves with security and efficiency enhancements by protecting user privacy and identifying, and retrieving answers automatically for recurrent questions. The results suggest that social networks can be leveraged to improve the answer quality and asker’s waiting time.

**Limitations of the reference**

- The implementation of a social network based Q&A system is complex and demands more time and resources to be spent over design and development of the platform.

### 2.1.2 Reference 2 - Research paper on Facilitating Students' Collaboration and Learning in a Question and Answer System.

**Insights from the reference**

The research paper describes the design ideas embodied in Green Dolphin (GD), a novel educational question and answer board with a social web interface that crowd-sources the task of answering technical questions for students in introductory programming courses. While GD has some commonalities with existing systems such as Stack Overflow and Piazza, it also has several distinctive features that set it apart, such as automatic student expertise identification and code assessment, and delaying the publication of expert answers to encourage student participation.

These features are designed to increase student participation, collaboration and sense of ownership. Students gain new knowledge from the flow of questions and answers in the system. They develop communication skills by asking and answering questions as well as programming and debugging skills.

**Limitations of the reference**

- The Green Dolphin platform limits the user audience only to students who are either good at programming or learning how to program through courses.

### 2.1.3 Reference 3 - Research paper on Wisdom in the Social Crowd: an Analysis of Quora.

**Insights from the reference**

Efforts such as Wikipedia have shown the ability of user communities to collect, organize and curate information on the Internet. Recently, a number of question and answer (Q&A) sites have successfully built large growing knowledge repositories, each driven by a wide range of questions and answers from its users community. While sites like Yahoo Answers have stalled and begun to shrink, one site still going strong is Quora, a rapidly growing service that augments a regular Q&A system with social links between users. Despite its success, however, little is known about what drives Quora's growth, and how it continues to connect visitors and experts to the right questions as it grows.

This paper presents the results of a detailed analysis of Quora using measurements and shed light on the impact of three different connection networks (or graphs) inside Quora, a graph connecting topics to users, a social graph connecting users, and a graph connecting related questions. The results show that heterogeneity in the user and question graphs are significant contributors to the quality of Quora's knowledge base. One drives the attention and activity of users, and the other directs them to a small set of popular and interesting questions.

**Limitations of the reference**

- Quora uses popups and interstitials to force users to login or register before they can see more of the content, similar to a metered paywall.

## 2.1.4 Reference 4 - Stack Overflow web application

**Insights from the reference**

Stack Overflow is a question and answer website for professional and enthusiast programmers. It is the flagship site of the Stack Exchange Network, created in 2008 by Jeff Atwood and Joel Spolsky. It features questions and answers on a wide range of topics in computer programming. Stack Overflow is the largest, most trusted online community for developers to learn, share their programming knowledge, and build their careers.

The website serves as a platform for users to ask and answer questions, and, through membership and active participation, to vote questions and answers up or down similar to reedit and edit questions and answers in a fashion similar to a wiki. Users of Stack Overflow can earn reputation points and "badges"; for example, a person is awarded 10 reputation points for receiving an "up" vote on a question or an answer to a question, and can receive badges for their valued contributions, which represents a gamification of the traditional Q&A website. Users unlock new privileges with an increase in reputation like the ability to vote, comment, and even edit other people's posts.

As of March 2021 Stack Overflow has over 14 million registered users, and has received over 21 million questions and 31 million answers. Based on the type of tags assigned to questions, the top eight most discussed topics on the site are: JavaScript, Java, C#, PHP, Android, Python, jQuery, and HTML. Stack Overflow also has a Jobs section to assist developers in finding their next opportunity. For employers, Stack Overflow provides tools to brand their business, advertise their openings on the site, and source candidates from Stack Overflow's database of developers who are open to being contacted.

**Limitations of Stack Overflow**

- This platform lacks to provide a feature for students from college to form local communities and groups to connect with each other.

## 2.1.5 Reference 5 - Quora web application

**Insights from the reference**

Quora is an American question-and-answer website where questions are asked, answered, followed, and edited by Internet users, either factually or in the form of opinions. Its owner, Quora Inc., is based in Mountain View, California, United States. Quora is a place to gain and share knowledge. It's a platform to ask questions and connect with people who contribute unique insights and quality answers.

In September 2018, Quora reported that it was receiving 300 million unique visitors every month. Despite its large number of registered users, Quora did not possess the same level of mainstream cultural dominance as sites like Twitter, which, at the time, had roughly 326 million registered users. This may have been because a large number of registered users on the site did not use it regularly and many did not even know they had accounts since they had either created them unknowingly through other social media sites linked to Quora or created them years previously and forgotten about them.

**2.5.2 Limitations of Quora:**

- Quora uses popups and interstitials to force users to login or register before they can see more of the content, similar to a metered paywall.

### 2.1.6 Reference 6 - React documentation

React documentation provides the related resources to learn more about the React - JavaScript library. People come to React from different backgrounds and with different learning styles. Whether we prefer a more theoretical or a practical approach, React documentation provides us with practical tutorials [9] to provide the opportunity of learning by doing. Or even prefer to learn concepts step by step, we can start with the guide to main concepts.

React documentation goes at a faster pace and introduces the most important React concepts in a detailed, beginner-friendly way. The React documentation assumes some familiarity with programming in the JavaScript language. This documentation always reflects the latest stable version of React.

## 2.2 Related Work

A revised 2017 Research highlights the different problems defined to implement React in market growth and popularity. These problems are mainly about (“big Development Community”), (“React core Architecture”), (“React is fast and Agile”). They highlighted that using React makes it easier to create interactive user interfaces. It efficiently updates through rendering the exact components to the view of each state and makes the data changes in the application.

The next research on Firebase, It is a backend platform for building Web applications. It offers a real time database, different APIs, multiple authentication types, hosting platform and much more. Firebase frees developers to focus on crafting fantastic user experiences. They don’t need to manage servers. You don’t need to write APIs. Firebase is your server, your API and your data store, all written so generically that you can modify it to suit most needs. Overall, it has a great deal of information providing surface knowledge on business benefits of SaaS. It has a ton of useful features that can all be tuned and tweaked to power the back-end of a **SaaS** application.

## 2.3 Existing Work

Some existing work on one of the features which the model directly or indirectly provides to or rather highlights it to the viewers. Redux proprietary research topic in 2016 by Alex, a front-end web developer emphasized its concentration on Redux; it is one of the hottest libraries in front-end development these days. However, many people are confused about what it is and what its benefits are.

Majority of our works and discussions are raised on 2021.Community-based Question Answering destinations enable users to find solutions to complex, point by point and individual questions from different users. Be that as it may, since answering a question relies upon the capacity and readiness of users to address the asker's needs, a huge portion of the questions stay unanswered. A fresher just took admission in Jain University, does not know anything about college and does not have any connection to others. He/ She can directly take our application and can find out his/ her queries. If his / her query is not present then can post his/ her query here.

System will ensure the students do not provide any fake answers or events through our site. Students will be able to see the live events happening on campus. This research is also provided with Comprehensive details about SDLC. The Software Development Life Cycle (SDLC) refers to a methodology with clearly defined processes for creating high-quality software. In detail, the SDLC methodology focuses on the following phases of software development. In this we can assign every team member to contribute their work separately through respective phases.

## 2.4 Problems in Existing system

Here are the list of problems, some of them are already explained in related and existing work

- Risk of mismanagement and of data when the project is under development.
- It will be more complicated when new features associates with environment settings and will become cost-bulky.
- ReactJS standard framework reusing methods and services saves the programming effort and improves reliability of the application.
- Communication between customer and owner is not direct.
- The complete hierarchy is not implemented manually.
- A system can be developed to be precise and that will be used if install must still be a good speculation for the organization.
- Still will there be no hostility from the user that will undermine the promising application benefits.

## 2.5 Proposed System

This addresses the demand of our project's idea and those services help the participating students or teachers to find better answers from so many queries and also learn about our university in non-invasive ways.

"JUQuery" is a platform for the purpose of knowing about the academic and non-academic environment, campus life of a student, and faculty and student relationship of Jain University in virtual way. This application provides us with complete information about campus life in student point of view, academic information in college point of view.

As stated, it provides guidance in following aspects,

- It gives an entrancing view of the whole process and services of JUQuery web application.
- The structure of the measurement system is on the basis of the findings of an earlier stage of the research that had resulted in the development of a number of process-based KPIs and the key issues they encompass.
- It follows Agile Methodology so get the benefits like Regular adaptation to changing circumstances , Even late changes in requirements are welcomed,
- And also testing practices have worked wonders for numerous organizations with positive aspects.
- On initial stages fire base provides an extensive help of a wide range of services and features and it is well known for backend-as-a-service. It has a ton of useful features that can all be tuned and tweaked to power the back-end of a SaaS application.

# 3. System Design

## 3.1 Architecture

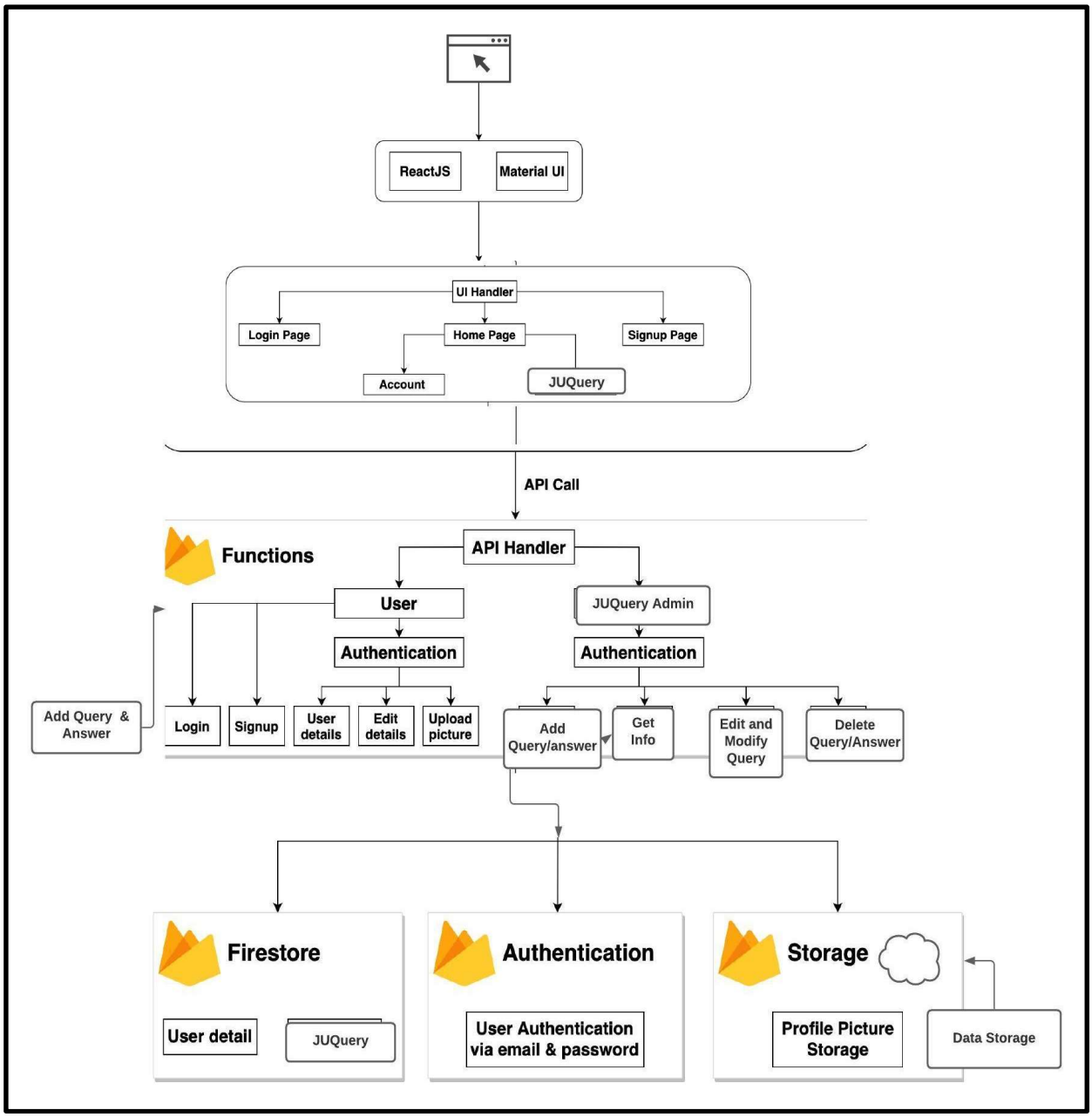


Figure 3.1.1: Architecture and Flow Diagram

After clicking any icon as given in the web page, through ReactJs it sends user to API call. There API handler will do the authentication part combining with Firebase Authentication. Also it uses Firestore Database to keep our all the data.

Through our application the following actions to be performed which leads to a hassle free and easy environment for **"JUQuery"** web application development,

1. It all starts with the decision on "what type of application needs to be developed for our university and also based on the current business needs". This is essential and acts as a base for deciding what do agile users expect as an outcome and what kind of technology needs to be included to achieve it. It also helps in further evaluation of cloud services that can be used for achieving business logic.
2. Requirements for software are laid, which encompasses all types of requirements such as mission, goal, scalability, audience, feature set, etc. This helps in gathering information about users and market and how they react with each other creating demand for the application. This is a part of MVP development scheme that is used in combination with agile in this guide.
3. Then a thorough analysis is done on what kind of cloud provider will be best suited and the decision for a provider is mainly based on cost, legal, and compliance.
4. Once the provider is decided, its platform is evaluated, and its services can help us to achieve the solution. The main agenda is to look at what kind of web application is being developed and which kind of target audience would be its market. This can bring clarity to the services which needs to be put into action to achieve the development.
5. Agile development methodology is much more beneficial as compared to other traditional software development methodologies.
6. As a part of Agile development, the team gets to choose the ReactJs and ReduxJS libraries. The guide doesn't specify the user to pick a specific one as depending on the type of application and adaptation of the developers, any of the two can be picked and followed to manage modules.
7. MVP or Minimum Viable Product is the correct balance of Minimum and Viability factor which ensures proper interests of end users and also takes in the minimum amount of effort that needs to be put in the development. Efforts here can be a replacement of cost, scalability, mobility and sometimes even features. A product of MVP development method is a subset of Minimum and Viable.

8. The mixture of Agile values and MVP development gives a much better, clear and clean solution as it
    - Ensures delivery of working software in weeks rather than months.
    - Instead of customer satisfaction, customer interest is prioritized.
    - Sustainable development with a constantly evolving pace depending on evaluating customer wants and delivering a MVP in process.
    - Simplicity in workflow ensures maximizing the amount of work undone.
    - Welcoming changing requirements
    - Regularly the team reflects how to cash in customers' interest and become more effective.
9. In the implementation phase, implementation is laid where business logic, backend, front end is implemented using suitable technologies.
10. Every version produced through the development proceedings is tested both locally and on the cloud in a pseudo production environment keeping all the edge cases into consideration.
11. In the final stage, it deployed in the fire base since it is a cloud-ready architecture there are no major lifts and shifts during deployment which leads to the final stage of maintenance.
12. In the final stage of maintenance, continuous improvement takes place based on the changing in customer or user needs. This guide truly promotes team growth over process and tools and always is adaptable to change, the model can endure change and keep up with it.

**Firebase** is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps.

- The Firebase API is secured via tokens—in order to generate such a token, we need to call Firebase's Client SDK and log in with a valid user/password credential.
- When successful, Firebase will send a token back in the response which we can then add to the header of any following request we want to perform.
- Firebase gives us complete control over authentication by allowing you to authenticate users or devices using secure JSON Web Tokens (JWTs).

- we generate these tokens on your server, pass them back to a client device, and then use them to authenticate via the signInWithCustomToken() method To reduce problems promoting code changes from development to staging to production, instead of including API keys in the code itself, either set them as environment variables or include them in a configuration file.
- Firebase data is retrieved by either a one-time call to GetValueAsync() or attaching to an event on a Firebase Database reference.

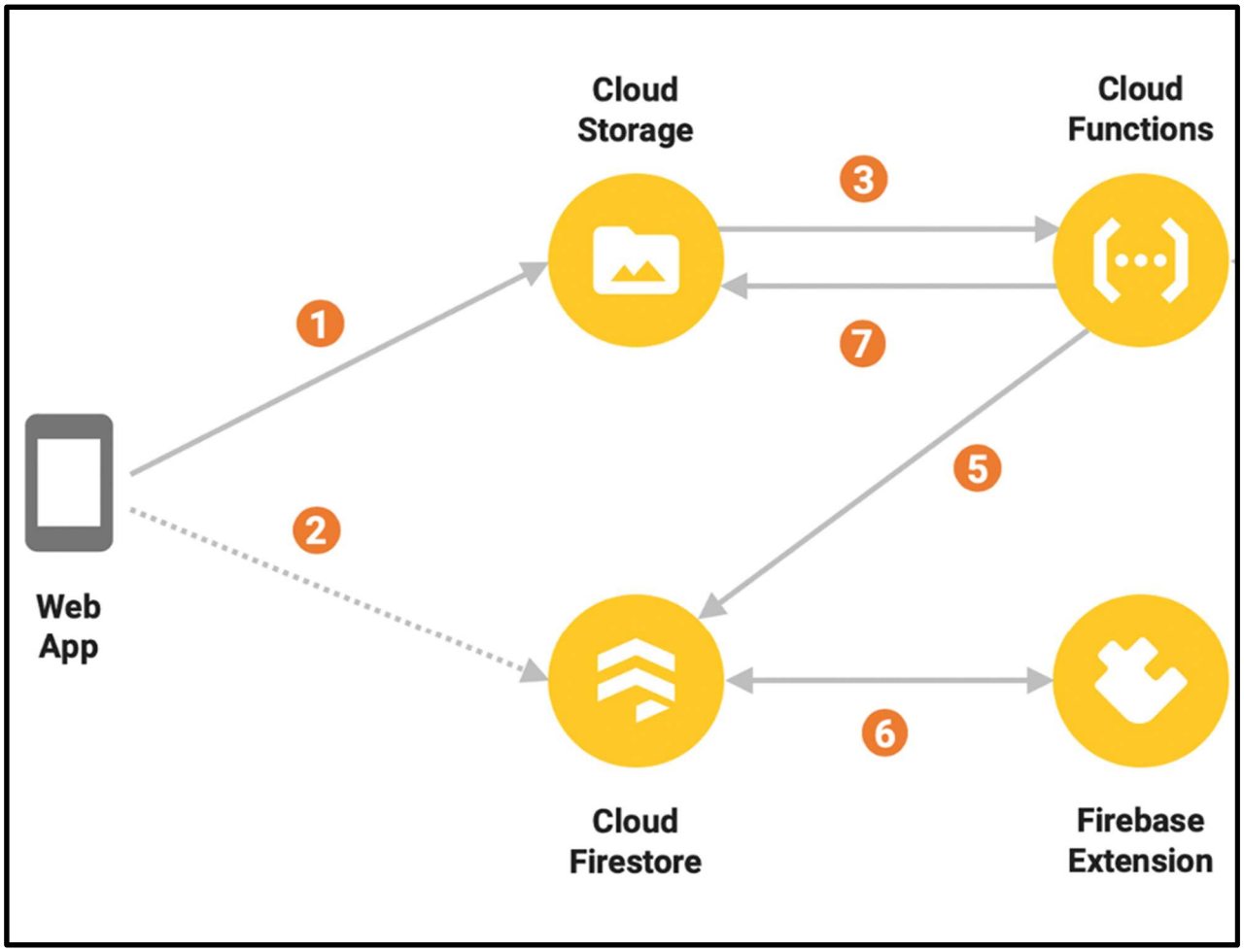


Figure 3.1.2: Application's connection with Cloud Firestore

The diagram shows the application's connection with Firestore for store and fetch data. Firebase extensions are added to implementation part to connect with it. Then we used firebase functions to store and fetch data and performing some sorting operations.

## 3.2 Use case Diagram

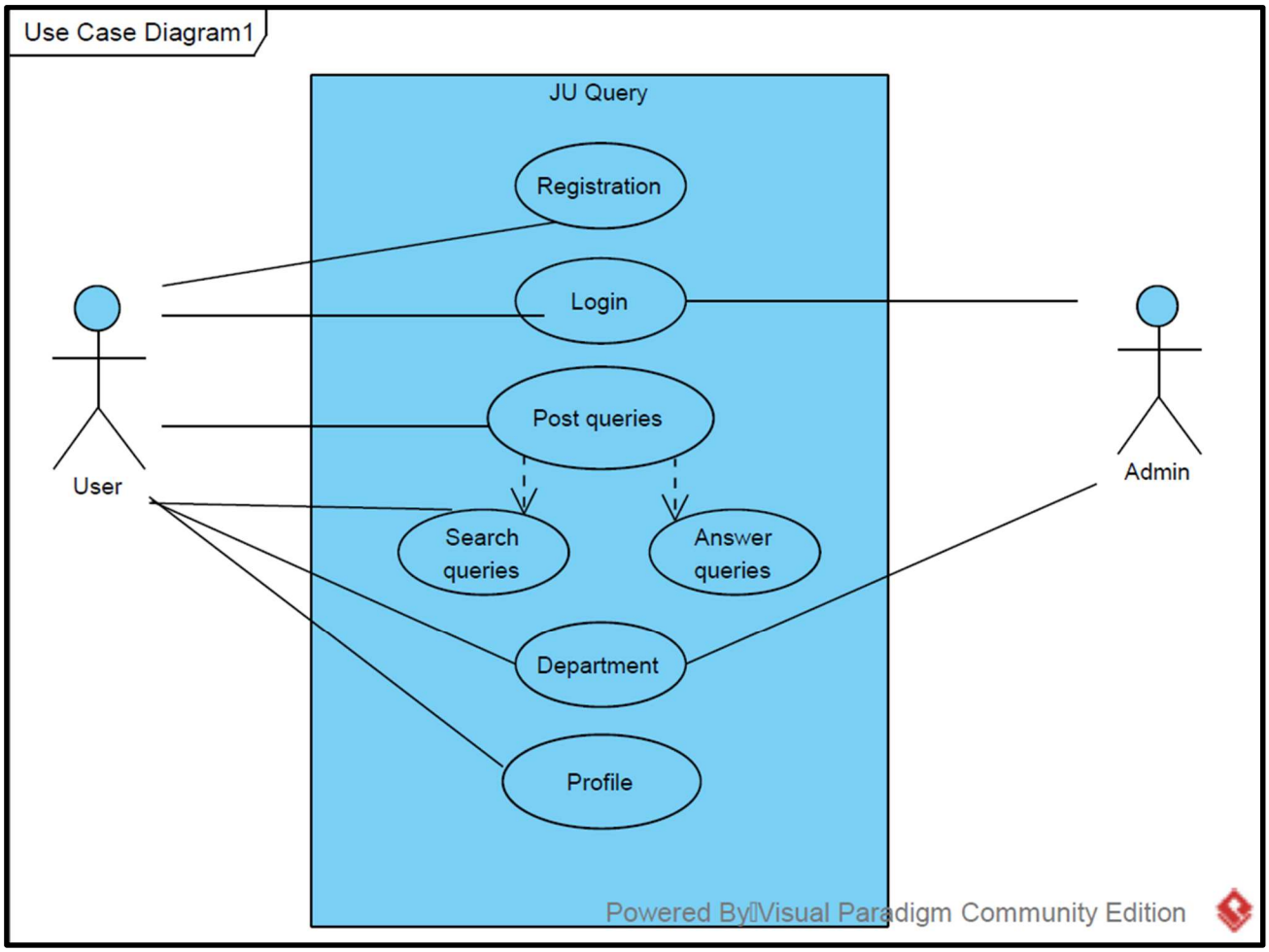


Figure 3.2: A general use case used in the architecture

In it there are two actors called User and Admin. And the use cases are Registration, Login, Post queries, Search queries, Answer queries, Section, Profile.

## 3.3 Activity Diagram

### 3.3.1 User Portion

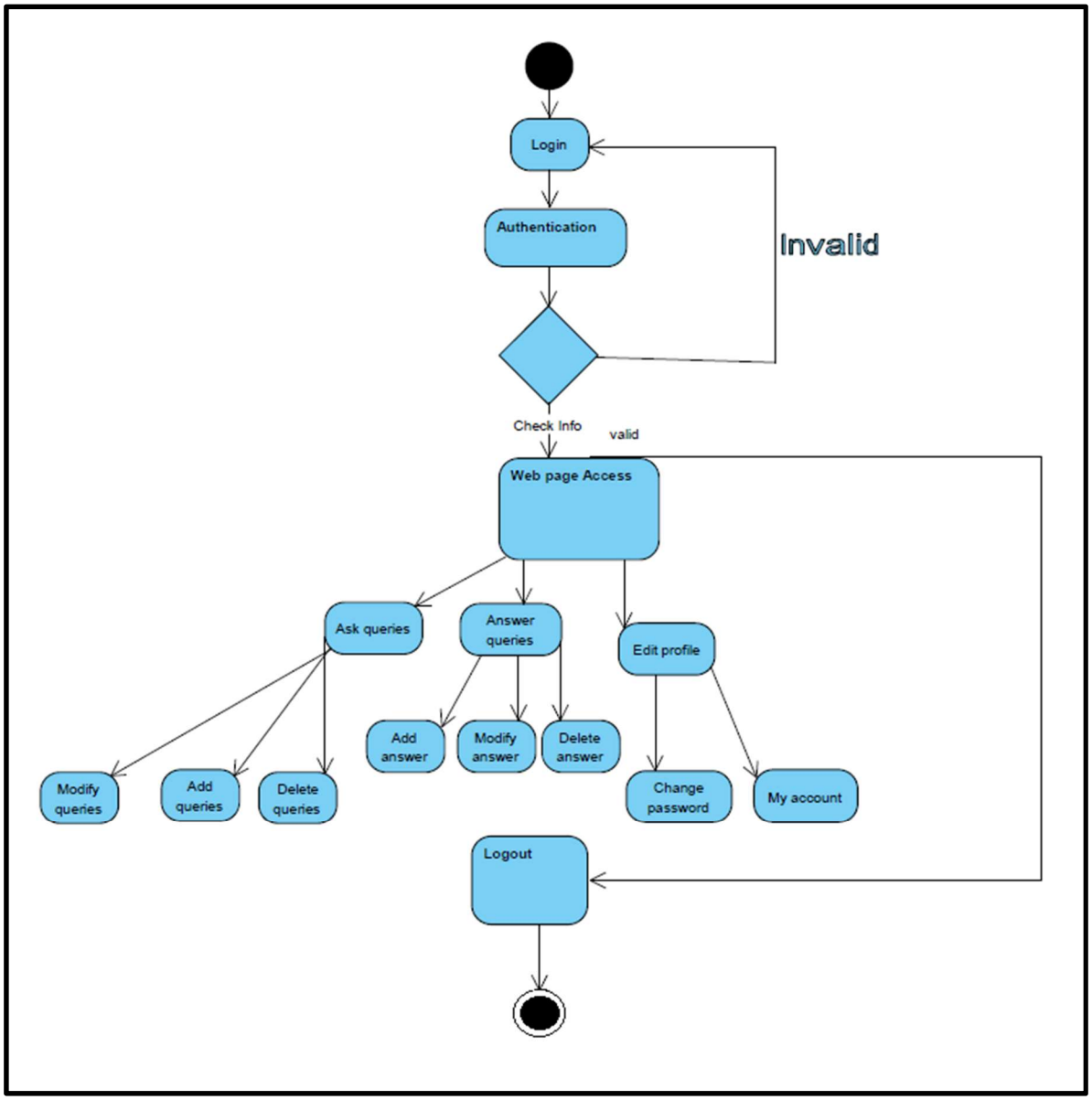


Figure 3.3.1: An activity diagram depicting the user side

Here through activity diagram we are disclosing are application's work flow. First of all user need to login or register, then it will take you for accessing the web page. Then user can perform any operation shown above.

### 3.3.2 Admin Portion

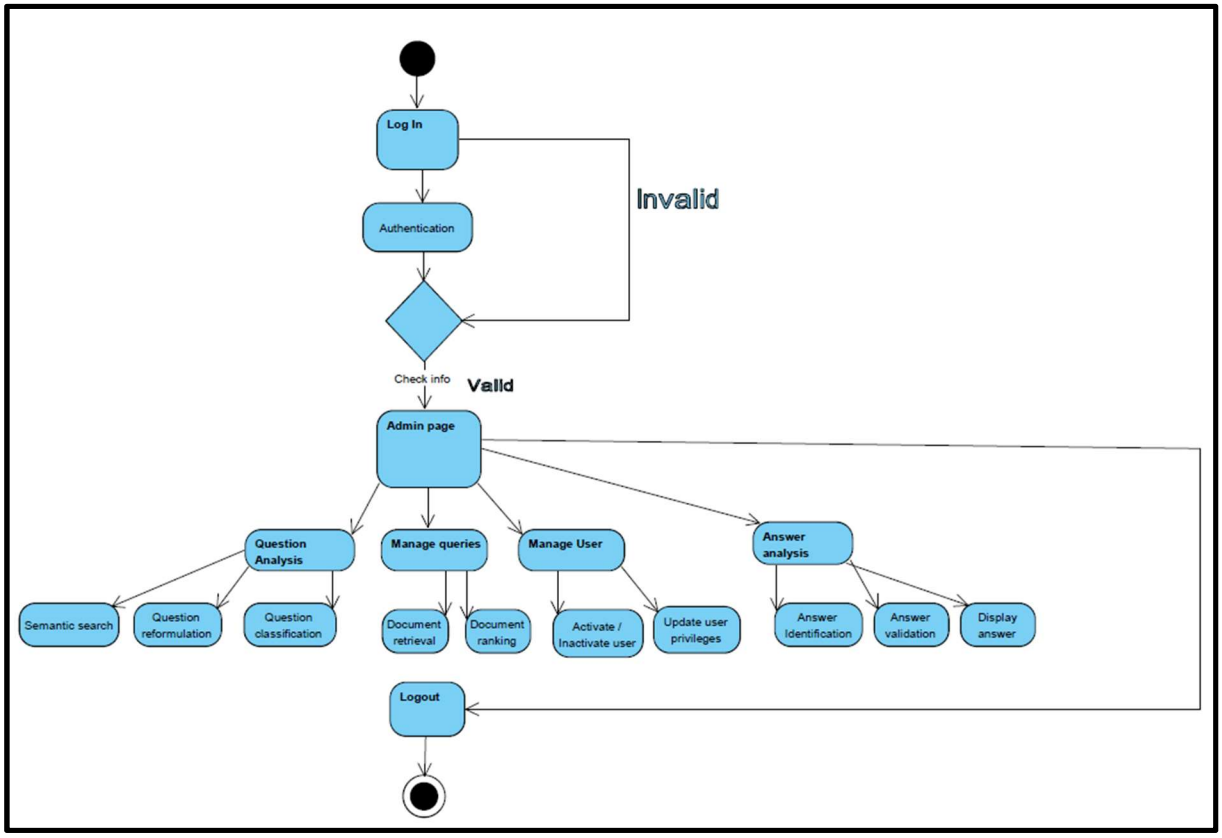


Figure 3.3.2: An activity diagram depicting the admin side

Here admin have to login in Firebase Firestore. After putting valid credentials he is supposed to be logged in admin page. His work is to notice on the users and delete some unethical question and answers and posts.

## 3.4 Class Diagram

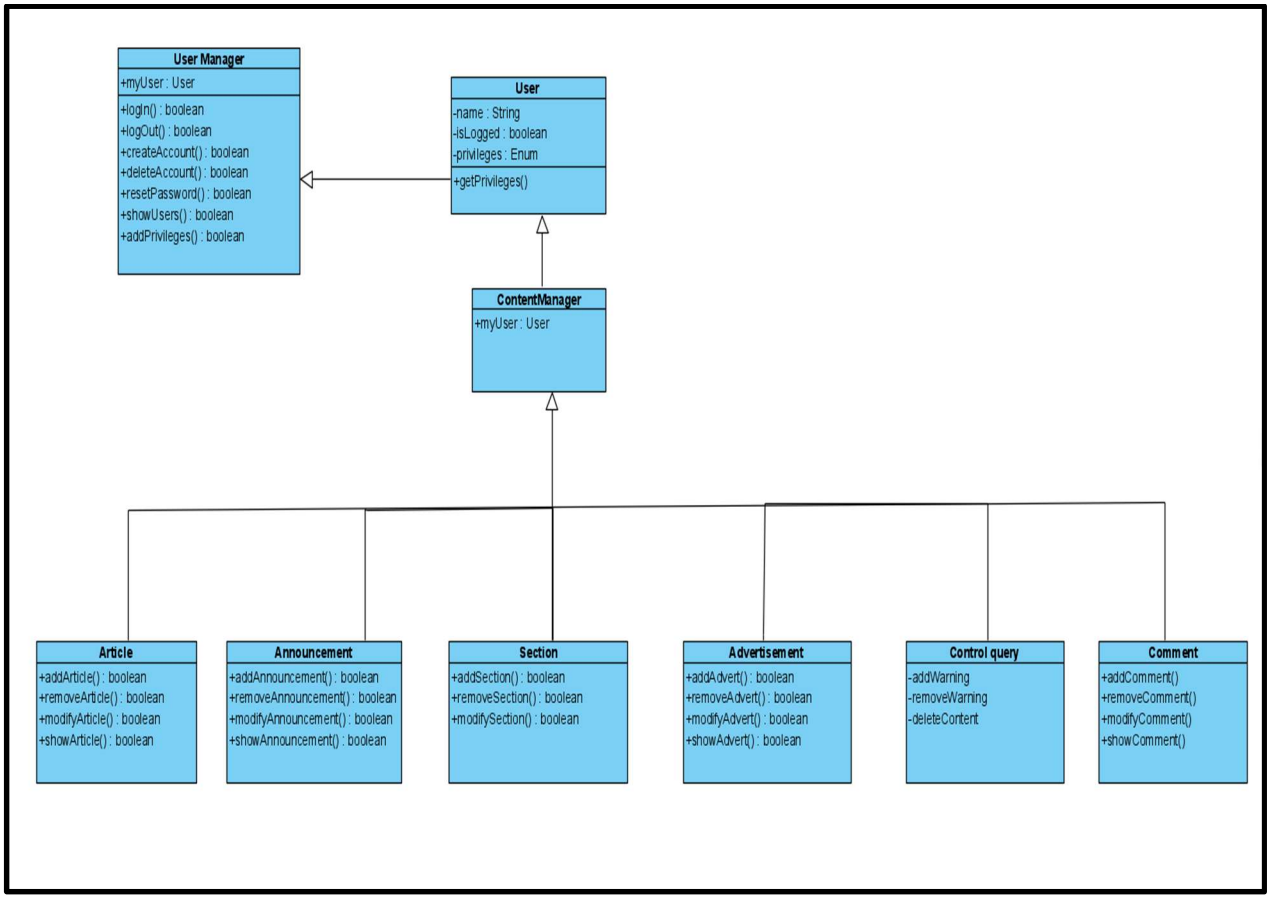


Figure 3.4: A class diagram depicting the coding architecture

In class diagram we explained about all the function in a proper way. Here we also mentioned the data types and the objects of any class. Each class has methods with their specific data types.

## 3.5 Sequence Diagram

### 3.5.1 User Portion:

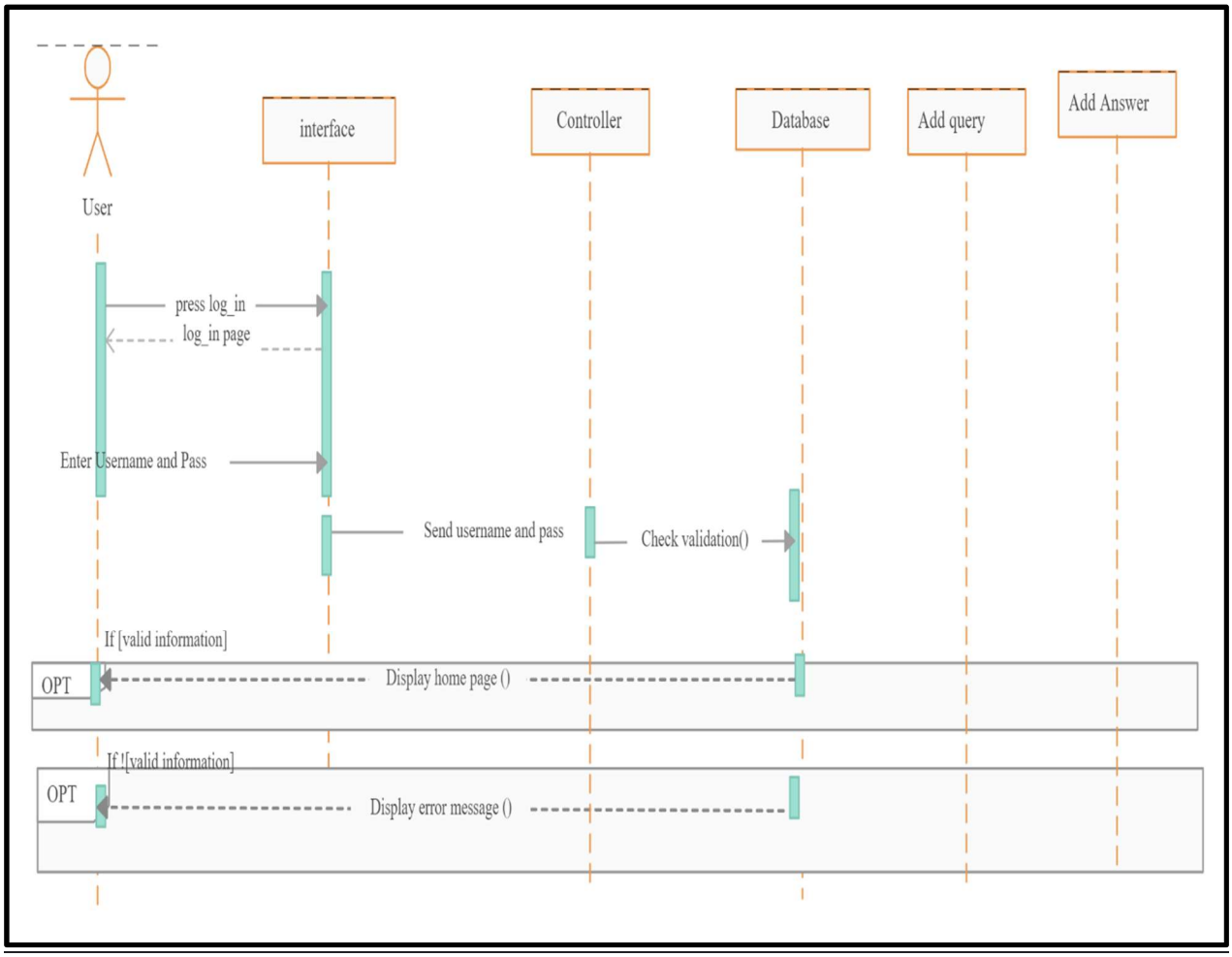


Figure 3.5.1: A general sequence flow used in the architecture

Above diagram shows how user's working flow will continue by sequence. First it will take you in interface. Then controller come in sequence, means all the work of front end will be done. Then connection with database will happen for all the operation.

### 3.5.2 Admin Portion:

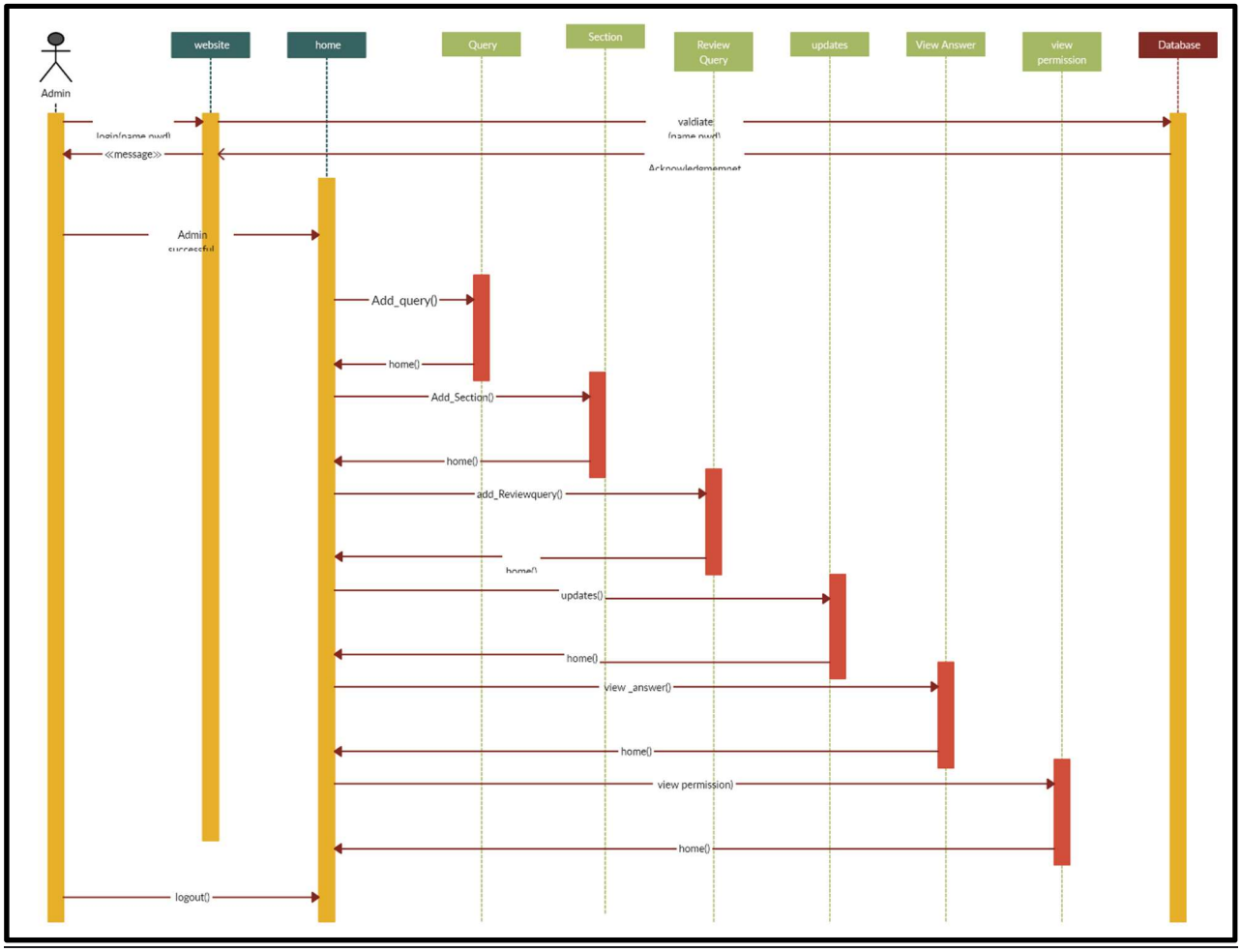


Figure 3.5.2: A general sequence flow used in the architecture

Above diagram shows how admin's working flow will continue by sequence. First it will take you in admin interface. Then all query and answers and user will be displayed to them. If they got any suspicious incident or post they can filter it out and the remaining will be send to database.

## 3.6 ER Diagram

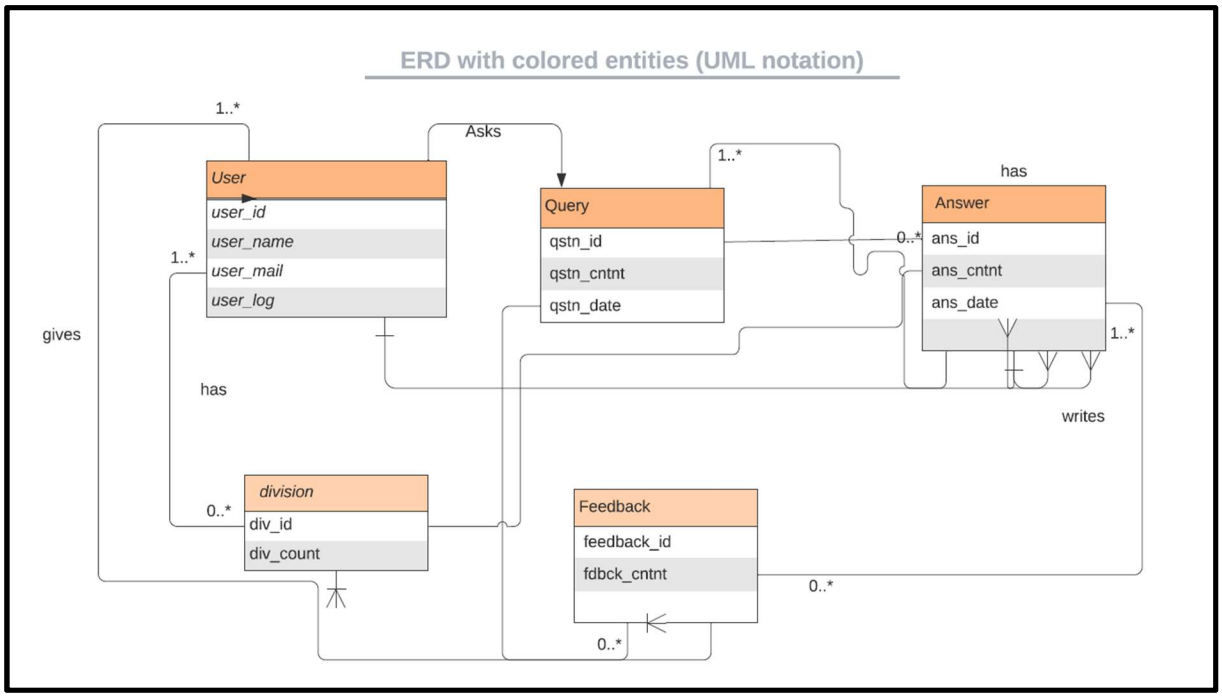


Figure 3.6: An ER diagram depicting the user side

By this we are disclosing about what data will be stored in data base for every user and every question and every answer. For user it will take user id, email, name, and photo. For Query it will take question id, section, question, timestamp. For answer it will collect answer id, answer, and timestamp. All are connected using the primary key of each entity.

# 4. Tool Description

## 4.1 Hardware Requirements: Description

The hardware requirements for Developers, Users and Admins at JU Query web application are as follows:

- An Intel Pentium 4 processor or later that's SSE3 capable
- 1 GB of RAM
- 1.6 GHz or faster processor
- Hard disk space: 10 GB or above
- Keyboard and mouse or other pointing device
- Network adapters for connection to the Internet
- Graphics display resolution

## 4.2 Software Requirements: Description

The Software requirements for Developers, Users and Admins at JU Query web application are divided into subsections as follows:

### 4.2.1 Operating System Requirements

- OS X El Capitán (10.11+)
- Windows 7 (with .NET Framework 4.5.2), 8.0, 8.1 and 10 (32-bit and 64-bit)
- Linux (Debian): Ubuntu Desktop 16.04, Debian 9
- Linux (Red Hat): Red Hat Enterprise Linux 7, CentOS 8, Fedora 24

### 4.2.2 Client-side Software Requirements

- Google Chrome
- Visual Studio Code
- Git Bash
- Command Prompt
- Windows Powershell

### 4.2.3 Developer libraries

- React.js
- React-modal
- Redux.js
- Material-ui
- Firebase
- React-testing-library

These requirements satisfy the Developers, Users and Admins to access the JU Query Web Application for seamless user experience, easy access to the database of Queries and answers through API-requests and secure authentication using Firebase Auth functions.

# 5. Implementation

“JUQuery " is a platform that provides us with complete information about campus life from a student point of view, and academic information from a college point of view. It features a familiar and well-thought-out and attractive user interface, much like posting a query, giving a response to a query, helping a student through a personal chat if required.

A user has complete freedom to post any query, and can also make a funny post in a legal way. Also students can register/ host any campus related events and also can take the feedback from other students through this platform. So, the following techniques have been taken into the implementation of the web application:

## 5.1. Description Part

### 5.1.1. Functional programming using functional components

- Functional programming is a concept of creating pure functions for software logic.
- It avoids concepts of mutable data and a shared state as used in Object-oriented programming. The functional programming is based more on expressions and declarations rather than the statements.
- The functional programming depends only on the arguments passed to the function.
- In React, a functional component is a plain JavaScript function that returns JSX.
- In functional programming, data cannot be stored in objects and it can only be transformed by creating functions.
- Functional components lack a significant amount of features as compared to class-based components. The gap is made up with the help of a special ReactJS concept called “hooks”. Hooks are special functions that allow ReactJS features to be used in functional components.

So, in our project we have incorporated functional programming using functional components where we pass data as shared state through functional arguments.

### 5.1.2. State management using Redux.js library

- Redux is a predictable state container designed to help us write JavaScript apps that behave consistently across client, server, and native environments and are easy to test.
- With Redux, the state of your application is kept in a store, and each component can access any state that it needs from this store.
- State management is essentially a way to facilitate communication and sharing of data across components.
- It creates a tangible data structure to represent the state of your app that you can read from and write to.
- As it's mostly used as a state management tool with React, we have taken advantage of it to manage states of our platform event triggers such as user login/logout, show/hide popup, user inputs to provide a user friendly experience.

### 5.1.3. Specific re-rendering using ReactDOM in React.js

- In React every UI piece is a component, and each component has a state.
- React follows the observable pattern and listens for state changes.
- When the state of a component changes, React updates the virtual DOM tree.
- Once the virtual DOM has been updated, React then compares the current version of the virtual DOM with the previous version of the virtual DOM. This process is called “diffing”.
- Once React knows which virtual DOM objects have changed, then React updates only those objects, in the real DOM.
- This makes the performance far better when compared to manipulating the real DOM directly.
- This makes React standout as a high performance JavaScript library.
- ReactDOM.render() controls the contents of the container node you pass in. Any existing DOM elements inside are replaced when first called. Later calls use React’s DOM diffing algorithm for efficient updates.
- We have taken advantage of this React.js feature to re-render only those parts of the website which require a refresh, saving a lot of data usage from the user' experience with our platform.

### 5.1.4. Secure API authentication using Firebase

- Firebase Authentication is an extensible token-based auth system and provides out-of-the-box integrations with the most common providers such as Google, Facebook, and Twitter, among others.
- It enables us to use custom claims which we'll leverage to build a flexible role-based API.
- The Firebase API is secured via tokens—in order to generate such a token, we need to call Firebase's Client SDK and log in with a valid user/password credential.
- When successful, Firebase will send a token back in the response which we can then add to the header of any following request we want to perform.
- We take leverage over this secure features of Firebase Authentication API to keep our users secure and gain their trust over our platform's user experience
- Firebase gives us complete control over authentication by allowing you to authenticate users or devices using secure JSON Web Tokens (JWTs).
- We generate these tokens on your server, pass them back to a client device, and then use them to authenticate via the signInWithCustomToken() method To reduce problems promoting code changes from development to staging to production, instead of including API keys in the code itself, either set them as environment variables or include them in a configuration file.

### 5.1.5. Real time fetching data using Firebase Firestore database

- Firebase data is retrieved by either a one-time call to GetValueAsync() or attaching to an event on a Firebase Database reference.
- The event listener is called once for the initial state of the data and again anytime the data changes.
- Firebase is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit.
- It is built on Google's infrastructure.
- It is categorized as a NoSQL database program, which stores data in JSON-like documents.
- Stores data as collections of documents.

- Simple data is easy to store in documents, which are very similar to JSON. Complex, hierarchical data is easier to organize at scale, using sub collections within documents.
- Requires less denormalization and data flattening.
- Firestore also features richer, faster queries and scales further which is very helpful with Real time fetching of data from backend and rendering data on front-end with a very low latency which makes the user experience very effective.

This is how we have created the “JU Query” Web application, by using these technologies and techniques along with testing each component of the codebase locally and in production using in-built react-testing-library.

For each feature, a React component is implemented to follow the separation of concern principle which is connected together using Dependency injection and component re-rendering to make coding implementation effective along with ease to extend and maintain in the future making it scalable and flexible.

This component structure provides the user with on-time delivery of update releases and upgrades.

## 5.2. Coding Part (some parts)

**Query.jsx**

```
import React from "react";
import Navbar from "./Navbar";
import Sidebar from "./Sidebar";
import Feed from "./Feed";
import Widget from "./Widget";
import '../css/Juquery.css';
const Juquery=()=>{
    return(
        <div className='juquery'>
            <Navbar/>
            <div className='juquery__content'>
                {/* <Sidebar/> */}
                <Feed/>
                {/* <Widget/> */}
            </div>
        </div>
    );
};
export default Juquery;
```

**Navbar.jsx**

```
import React from 'react';
import HomeIcon from '@material-ui/icons/Home';
import '../css/Navbar.css';
import FeaturedPlayListOutlinedIcon from '@material-
ui/icons/FeaturedPlayListOutlined';
import AssignmentTurnedInOutlinedIcon from '@material-
ui/icons/AssignmentTurnedInOutlined';
import PeopleAltOutlinedIcon from '@material-ui/icons/PeopleAltOutlined';
import NotificationsOutlinedIcon from '@material-ui/icons/NotificationsOutlined';
import SearchIcon from '@material-ui/icons/Search';
import {Avatar, Input} from '@material-ui/core';
import LanguageIcon from '@material-ui/icons/Language';
import {Button} from '@material-ui/core';
import { useSelector } from 'react-redux';
import { useState } from 'react';
import {selectUser} from '../features/userSlice';
import db, {auth} from '../firebase';
import Modal from 'react-modal';
import { Link } from '@material-ui/icons';
```

```
import firebase from 'firebase';
Modal.setAppElement("#root");
const Navbar=()=>{
    const user=useSelector(selectUser)
    const [openModal,setOpenModal]=useState(false);
    const [selectOption,setSelectOption]=useState();
    const [input,setInput]=useState();
    const [inputUrl,setInputUrl]=useState();
    const handleChange=(e)=>{
        setSelectOption(e.target.value);
    }
    const handleQuestion=(e)=>{
        e.preventDefault()
        setOpenModal(false);
        db.collection('queries').add({
            section:selectOption,
            question:input,
            imageUrl:inputUrl,
            user:user,
            timestamp:firebase.firestore.FieldValue.serverTimestamp(),
        });
        setInput("");
        setInputUrl("");
        setSelectOption("");

    }
    console.log(selectOption)
    return(
        <div className='qHeader'>
            <div className='qHeader__logo'>
                <img src="../jgi_logo.jpg"
                alt="JGI icon" />
            </div>
            <div className='qHeader__icons'>
                <div className='qHeader__icon'>
                    <HomeIcon/>
                </div>
                <div className='qHeader__icon'>
                    <FeaturedPlayListOutlinedIcon/>
                </div>
                <div className='qHeader__icon'>
                    <AssignmentTurnedInOutlinedIcon/>
                </div>
                <div className='qHeader__icon'>
```

```
                    <PeopleAltOutlinedIcon/>
                </div>
                <div className='qHeader__icon'>
                    <NotificationsOutlinedIcon/>
                </div>
            </div>
            <div className='qHeader__input'>
                <SearchIcon/>
                <input type="text" placeholder='Search JU Query' />
            </div>
            <div className='qHeader__Rem'>
                <div className='qHeader__avatar'>
                    <Avatar onClick={()=>auth.signOut()} src={user.photo}/>
                </div>
                <LanguageIcon/>
                <Button onClick={()=>setOpenModal(true)}>Add Query</Button>
                <Modal isOpen={openModal}
                onRequestClose={()=>setOpenModal(false)}
                shouldCloseOnOverlayClick={false}
                style={{
                    overlay:{
                        width:700,
                        height:600,
                        backgroundColor:"rgba(0,0,0,0.8)",
                        zIndex:"1000",
                        top:"50%",
                        left:"50%",
                        marginTop:"-300px",
                        marginLeft:"-350px"
                    }
                }}>
                    <div className='modal__title'>
                        <h5>Add Question</h5>
                        <h5>Choose Section</h5>
                        <h5>Share Link</h5>
                    </div>
                    <div className='modal__info'>
                        <Avatar
                        className='avatar'
                        src={user.photo}/>
                        <p>{user.display? user.display:user.email}</p>
                        <div className='modal__select'>
                            <select name="Choose Section" id="" onChange={handleC
hange} required>
```

```
                                <option value="">Select..</option>
                                <option value="Technology">Technology</option>
                                <option value="Technical Events">Technical Events
</option>
                                <option value="Innovation">Innovation</option>
                                <option value="Placement">Placement</option>
                                <option value="Sports">Sports</option>
                                <option value="Cultural Events">Cultural Events</
option>
                                <option value="Transportations">Transportations</
option>
                                <option value="Hostel Life">Hostel Life</option>
                                <option value="Canteen">Canteen</option>
                                <option value="Others">Others</option>
                            </select>
                        </div>
                    </div>

                    <div className='modal__field'>
                        <Input
                        value={input}
                        required
                        onChange={(e)=>setInput(e.target.value)}
                        type='text'
                        placeholder="Select an appropriate option and post a ques
tion"/>

                        <div className='modal__fieldLink'>
                            <Link/>
                            <Input
                            value={inputUrl}
                            onChange={(e)=>setInputUrl(e.target.value)}
                            className='modal__link'
                            type='text'
                            placeholder="Optional: include a link that gives cont
ext"/>
                        </div>
                        <div className='modal__buttons'>
                            <button
                            className='cancle'
                            onClick={()=>setOpenModal(false)}>
                                Close
                            </button>
```

```
                            <button onClick={handleQuestion} type='submit' classN
ame='add'>
                                Add Question
                            </button>
                        </div>
                    </div>
                </Modal>
            </div>
        </div>
    );
};
export default Navbar;
```

**Feed.jsx**

```
import React, { useEffect, useState } from 'react';
import '../css/Feed.css';
import db from '../firebase';
import Post from './Post';
import QueryBox from './QueryBox';
// import Sidebar from './Sidebar';

function Feed() {

    const [posts, setPosts] = useState([]);
    const [selection,setSelection]=useState()
    console.log(selection)
    useEffect(() => {
        selection?db.collection('queries')
        .orderBy('timestamp', 'desc')
        .where('section','==',selection)
        .onSnapshot(snapshot => setPosts(snapshot.docs.map((doc)=> ({
        id: doc.id,
        query: doc.data(),
    }
    )))):
    db.collection('queries')
            .orderBy('timestamp', 'desc')
            // .where('section','==',selection)
            .onSnapshot(snapshot => setPosts(snapshot.docs.map((doc)=> ({
            id: doc.id,
            query: doc.data()       }
        ))))
```

```
    })
    return (
        <div className='main'>
            <div className='sidebar'>
                {/* <Sidebar/> */}
                <div className='sidebar'>
                    {/* <SidebarOptions/> */}
                <div className='sidebaroptions'>
                <div className='sidebaroption' onClick={()=>setSelection("Technol
ogy")}>
                <img src="https://qphs.fs.quoracdn.net/main-thumb-t-2177-100-
JiR07D1TQSfeQzRvWXomVaY4Poj2f8Yb.jpeg"
                alt="Technology" />
                <p
                // value={Technology}
                // onClick={(e)=>setSelection("Technology")}
                >Technology</p>
            </div>
            <div className='sidebaroption' onClick={()=>setSelection("Technical E
vents")}>
                <img src="https://qphs.fs.quoracdn.net/main-thumb-t-18797-100-
B7cd2Zkrhke1Bd3yn2Vq0HIwXIVg0JaW.jpeg"
                alt="Technical Events" />
                <p
                // onClick={()=>setSelection("Technical Events")}
                >Technical Events</p>
            </div>
            <div className='sidebaroption' onClick={()=>setSelection("Innovation"
)}>
                <img src="https://qphs.fs.quoracdn.net/main-thumb-t-3906-100-
3WJwKe2alb83spIH7rfECXY49noETA9x.jpeg"
                alt="Innovation" />
                <p>Innovation</p>
            </div>
            <div className='sidebaroption' onClick={()=>setSelection("Placement")
}>
                <img src="  https://qphs.fs.quoracdn.net/main-thumb-t-996-100-
bfZBQjeEenKKl8fcNY4tVv0FyArtB0Mb.jpeg"
                alt="Placement" />
                <p>Placement</p>
            </div>
            <div className='sidebaroption' onClick={()=>setSelection("Sports")}>
                <img src="https://qphs.fs.quoracdn.net/main-thumb-t-795-100-
fujtazxcynewvtxarymxenpcldxwczqq.jpeg"
```

```
                alt="Sports" />
                <p>Sports</p>
            </div>
            <div className='sidebaroption' onClick={()=>setSelection("Cultural Ev
ents")}>
                <img src="https://qphs.fs.quoracdn.net/main-thumb-t-801-100-
Sf8h894FXbQZQit0TeqDrrqS6xw6dwCQ.jpeg"
                alt="Cultural Events" />
                <p>Cultural Events</p>
            </div>
            <div className='sidebaroption' onClick={()=>setSelection("Transportat
ions")}>
                <img src="https://qphs.fs.quoracdn.net/main-thumb-t-2587-100-
Cg7htHNXLNzX0Pf3PGbKJ9RFT5y54Geb.jpeg"
                alt="Transportations" />
                <p>Transportations</p>
            </div>
            <div className='sidebaroption' onClick={()=>setSelection("Hostel Life
")}>
                <img src="  https://qphs.fs.quoracdn.net/main-thumb-t-4600-100-
Tu9r6tIH3ysCZM71TLxbSRhOeejLecHy.jpeg"
                alt="Hostel Life" />
                <p>Hostel Life</p>
            </div>
            <div className='sidebaroption' onClick={()=>setSelection("Canteen")}>
                <img src="https://qphs.fs.quoracdn.net/main-thumb-t-1027-100-
tgvinqyfjlnnyjglbiicbisovkptvjqn.jpeg"
                alt="Canteen" />
                <p>Canteen</p>
            </div>
            <div className='sidebaroption' onClick={()=>setSelection("Others")}>
                <img src="https://qphs.fs.quoracdn.net/main-thumb-t-3229-100-
uazulrnpoeacbeuytiyqovquifuuurer.jpeg"
                alt="Others" />
                <p>Others</p>
            </div>
        </div>
        </div>
            </div>

            <div className="feed">
                <QueryBox />
                {
                    posts.map(({id, query}) => (
```

```
                    <Post
                        key={id}
                        Id={id}
                        image={query.imageUrl}
                        section={query.section}
                        query= {query.question}
                        timestamp = {query.timestamp}
                        juQueryUser = {query.user}
                    />
                ))
            }
        </div>
        <div className="widget">
        <div className="widget__header">
            <h5>Spaces to follow</h5>
        </div>
        <div className="widget__contents">
        <div className="widget__content">
            <img
                src="
MAAAAvQTlLAAAAflBMVEX///8UHDv19fUQGTkABTEAACLd3uCtrbMAAC8LFTcAACpwc4MrMkx6fIfV1tm
1trwAACcAAB0ACTEAABrIyc4bIkFqbXu+v8UAAADP0NTn6OkAAA1iZnaLjZh1d4Wkpa09Qlk2OU2bnaZb
XW5FSF1AQlMfITwAABUiKkdSVmrWFoGWAAACe0lEQVR4nO2afZOaMBCHhSTyJhhA4fAMWHu9tt//C1YlI
njQIV7PzXR+z38O2ckzkF2ThcUCAAAAAAAAAAAAAAAAAZuHW8UNEuy/2EqH/CK/LL/bizHkEAS94/QdeTM
wke6oXWy2TWSz32TO9gtXcgMiHF7xmAy8z4GUGvMyAlxnwMgNeZsDLDHiZ8Tcv17XPK/FUXRS1ahKLvHb
Ngfs8O8H97Ntx0FIi9Fqv0sDpCMTKs8IrCntWl4thcVtpZF6xcD4gb2JUXrH8qHUWI/aqUt3fYTwVIteP
lElF6+V+1/2abLsvk6RSbyE7L7BjF0DjtfG11g9dt1yVBUHQS0gar5q3GXi4TVsGzqYXQOPlt6srXfeGb
AYln8Rr99reru10AIlXmV4m5bVlXk1bvMR6OoDES4WXSf3SMq9rOm6mA0i8oktVZU5lmVehve43g+Re3E
6viNv5HON20sy2da/rBLetTnjtXlV40wE0/0N5W1eL6QASL/flMinzLfNabNt9s2h6Q7xBFtB46YXP3m8
ux5+D3RiNVxW0G8PgXafkbp8xxm/be6rzUFtZz++VV8pbN3v//Jvlivr8uLye006HICnl9a18GBN7Lby7
JoD2Ij4/nlD5hy8XmIi6y3R9k6PDh1oZo1/3Z5ZvKe+eZsDTQ397QdovrOKDL8ITUh7q4eaCuI+5S8qjU
sqr7r8/s6DvOwq84GUAvMww9qqf6sV+letZlL/vvkurpkb+Ey+HhTPRf1CdV5GODpNi7KWXuZcpnVeUjV
5nnNqLj16n98L9ssIrlXMzcZBuL10+5nKU/HNeC+8xmuu05dSIz2kBAAAAAAAAAAAAAAAAAOX8Ad608luA
68LAAAAAASUVORK5CYII="
                    alt=""
                />
                <div className="widget__contentTitle">
                    <h5>Jain University (JU) Chapter</h5>
                    <p>Initiative to provide guidance and awareness</p>
                </div>
            </div>
            <div className="widget__content">
                <img
```

```
                    src="https://cdn-images-
1.medium.com/fit/c/64/64/1*pys6Xb0bV4Ld9KcI8np8QQ.png"
                    alt=""
                />
                <div className="widget__contentTitle">
                    <h5>TechSoftware</h5>
                    <p>Technology and Software blends into singularity</p>
                </div>
            </div>
        </div>
        </div>
        </div>
    )
}

export default Feed;
```

# 6. Results and Discussion

"JUQuery" web application is a deliverable for students and faculties of the college incorporated with production-grade coding standards. This application is quite composite in nature and requires certain technical prerequisites to understand the implementation used in the background.

The web application is intended for college students and faculties making them the target audience for this platform and mainly focuses on promoting virtual interactions and helps each and every student to know what is exactly going on the college campus.

The technology stack used in the process of preparation of this:

- HTML5
- CSS3
- JavaScript with ES6 features
- NPM
- React JS
- Redux JS
- React-testing-library
- Material-UI
- Git & GitHub
- Firebase Firestore
- Firebase Authentication
- Firebase Deploy

## 6.1 Registration page


*Figure 6.1: A screenshot of registration page of JU Query*

## 6.2 Home page

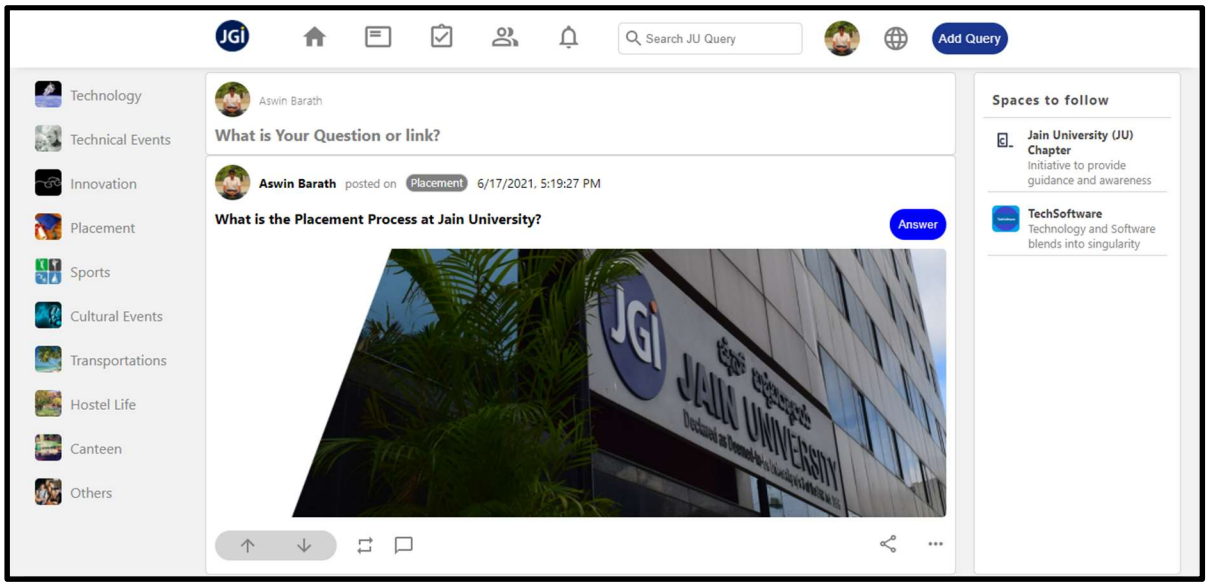


*Figure 6.2: A screenshot of Live Question & Answer feature on JU Query web app*

## 6.3 Add Query feature

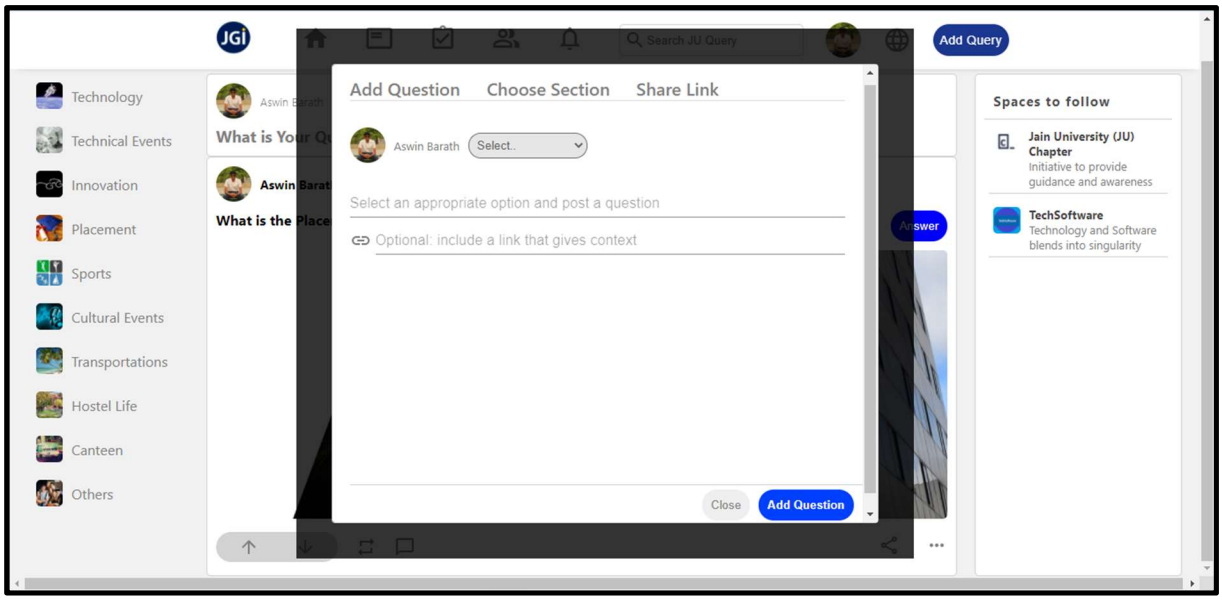


*Figure 6.3: A screenshot of Live Add query on JU Query web app*

## 6.4 Add Answer feature

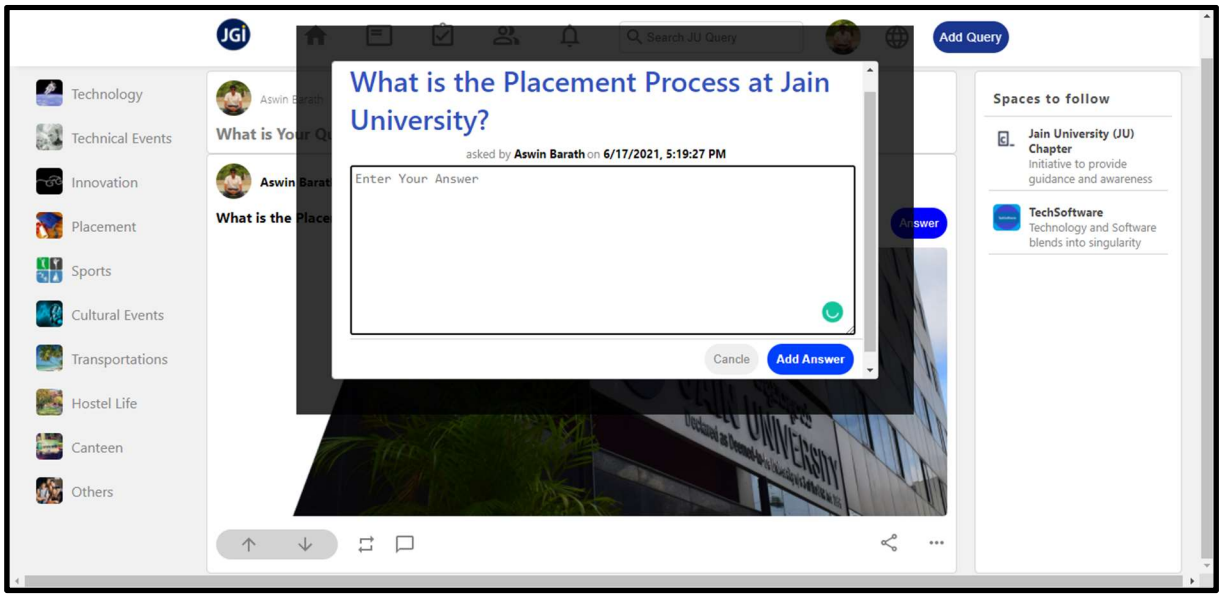


*Figure 6.4: A screenshot of Live Add Answer feature on JU Query web app*

# CONCLUSIONS AND FUTURE SCOPE

In this project, we have done an application that helps the students of a college to get to know about their academic and non-academic information from the student's perspective.

And every student of the college can post any query they want to post and anyone who wants to answer the post can do that. This process enables students across the campus even from different departments to interact with each other and get collaborative knowledge.

And we have 9 different kinds of categories where we can have questions about and if the question the student wants to ask is not listed in the type of the category he can select others and can still post.

And in the future, there will be features like a chat option where students in the college can connect to a fellow student and can have a conversation with the student. And also students can organize an event through this application and students can register for the event they wanted to through this application.

# References

[1] SocialQ&A: An Online Social Network Based Question and Answer System, Published on August 2015, Conference: 2015 24th International Conference on Computer Communication and Networks (ICCCN), Authors: Haiying Shen; Guoxin Liu; Haoyu Wang; Nikhil Vithlani, Link: https://ieeexplore.ieee.org/abstract/document/7748490

[2] Facilitating students' collaboration and learning in a question and answer system, Published on February 2013, Conference: Proceedings of the 2013 conference on Computer supported cooperative work companion, Authors: Chulakorn AritajatiN; Hari Narayanan, Link: https://dl.acm.org/doi/abs/10.1145/2441955.2441983

[3] Wisdom in the Social Crowd: an Analysis of Quora, Published on May 2013, Conference: The 22nd International World Wide Web Conference (WWW), Authors: Gang Wang; Konark Gill; Manish Mohanlal; Haitao Zheng; Ben Y Zhao, Link: https://www.researchgate.net/publication/281376797_Wisdom_in_the_Social_Crowd_an_Analysis_of_Quora

[4] Stack Overflow, Link: https://stackoverflow.com/

[5] Wikipedia article on Stack Overflow, Link: https://en.wikipedia.org/wiki/Stack_Overflow

[6] Quora, Link: https://www.quora.com/

[7] Wikipedia article on Quora, Link: https://en.wikipedia.org/wiki/Quora

[8] React Documentation, Link: https://reactjs.org/docs/getting-started.html

[9] Practical tutorials from React documentation, Link: https://reactjs.org/tutorial/tutorial.html

[10] Guide to main React concepts, Link: https://reactjs.org/docs/hello-world.html

# APPENDIX-I

## PHOTOGRAPHS

| STUDENT NAME | EMAIL ID | PERMANENT ADDRESS | PHONE NUMBER | PHOTOGRAPH |
|---|---|---|---|---|
| Souvik Kar | karsouvik7872115553@gmail.com | Debra, Paschim Medinipure, West Bengal, India | +91 7872115553 |  |
| R. Srisarvesh | r.srisarvesh@gmail.com | Coimbatore, Tamil Nadu, India | +91 7094322569 |  |
| T Aswin Barath | aswin2001barath@gmail.com | Aruvankadu, Ooty, Tamil Nadu, India | +91 8754105798 |  |
| D Vyshnav Raj | 18btrse033@jainuniversity.ac.in | Kannapuram, Kannur, Kerala , India | +91 9847029428 |  |

# APPENDIX – II

## SOURCE CODE

GitHub: https://github.com/souvik-21/ju-query/tree/master

Note: This web application is still in development phase and shall be proprietary and may be commercialized of the creators wish to do it.