LONDON METROPOLITAN UNIVERSITY

Module Code & Module Title

CS6P05NT Final Year Project

Assessment Weightage & Type

40% Year-long Individual Model

Year and Semester

2021 - 2022

Student Name: Samuel Sherpa

London Met ID: 19031860

College ID: np05cp4a190148

Assignment Due Date: 27-04-2022

Assignment Submission Date: 27-04-2022

Word Count:

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Acknowledgment

First of all, I would like to open my heart and life in a fresh way where I surrender to acknowledge God the almighty in all my ways that his hand is upon me because of which I am able to study in this prestigious college and acquire to get more knowledge and skills.

I would like to express my special thanks of gratitude to my internal and external supervisors Mr.Ravi Rouniyar & Mr.Biwash Adhikari for their guidance and support in completing my coursework. Their calm nature and inspiring words helped me to work harder to finish the FYP.

I would like to acknowledge my parents and sister for their continuous support in my home by providing me with a peaceful environment where I could study very well. Last but not least I would like to thank my classmates who all helped me with my need and suggested making the project better.

Thank you all from the bottom of my heart. God bless you all.

Abstract

In Nepal, +5.5% of internet users have increased after the Covid19 pandemic since all of the education systems and work from home went online. But for the students, there aren't any good educational online platforms where students could ask their academic subject questions and get answers. This simple problem lightens a bulb in my head to build the android app 'Prashnottar'(PT). The whole project was built using the Flutter framework, dart programming language, and Laravel framework for backend with MySql database.

Different UML diagrams like class diagrams, activity diagrams, communication diagrams, and sequence diagrams were designed which helped to give a basic outline for feature development. System testing and unit testing were done for debugging after proper test planning to be confident that the application project works. The legal, social, and ethical issues associated with the 'Prashnottar' were properly understood for a peaceful and positive impact on end-users.

In this final report, you would be able to see my half-year journey of building a simple app that solves complex students' problems. I would be introducing your project in detail, then we understand the users, then have a ride in development bike finishing to the testing and analysis of features and stability of the app with a conclusion. For the more wholeness of the project, a lot is going on in the appendix part as well. The whole above sentences were PT in a nutshell, to know in detail let's go through the whole documentation.

Table of Contents

Table of figure

Table of tables

Table of Abbreviation

PT	Prasnottar - (name of my project)
OS	Operating system
DB	Database

Chapter 1. Introduction

1.1. Project Description

The name of the FYP is "Prashnottar" (PT), which is a Nepali word, in English, it means the question and answer. As the name suggests, the main theme of 'Prashnottar' is to let users search for the answers to their questions, which would be answered by other users. In brief: 'Prashnottar' is an android app built using Flutter framework, Laravel framework for backend, and Mysql for the database: for students studying in between classes 4-12 where they will be choosing their current studying class, then they will start to view answers to the questions asked about different topics of a particular subject, comment it, like it and if the answer is too good, then upvote it for it to come up in search result. In addition to these features, they would also be able to answer others' questions by either writing or sharing images of their written answer.

With the detection of a new variant of COVID, 'Omicron' found in SouthAfrica, the world could again go into lockdown pushing education online. If not 'Omicron', other variants may arise, but with or without lockdown, it's obvious that students don't always rely on school and tuition, but they tend to search for answers and solutions online to have a better understanding of the topic which makes this project a chance to have a real value impact in coming days.

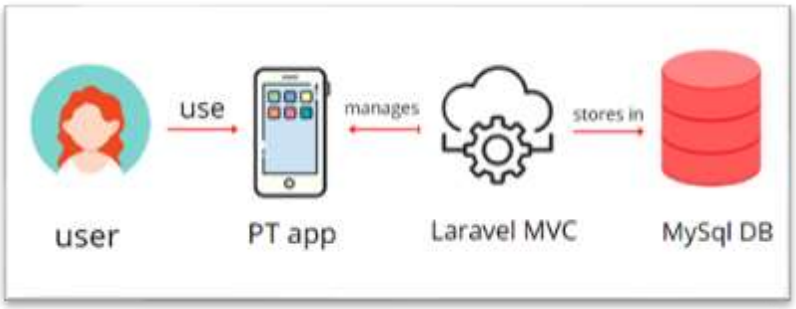


Figure 1 Project description in the picture

A user uses an android app that was developed with the help of the Larvel MVC pattern. The project is created in such a way that the front end shows data, powered by the backend managed by the API of the application.

1.2. Current Scenario

In Nepal, there were about 35,222 elementary and secondary schools and 10 universities with more than 1400 colleges and campuses until 2016 (WENR, 2021). The number of students in Nepal increased by 407 percent between 2000 and 2013, from 94,041 to 4,77,077 students 2013. In 2016, there were about 3,61,077 students in Nepal. (WENR, 2021). At present 2021, the correct number of students studying at the school level is not published, but we can estimate from the above data and growth tendencies that there could be more than one million students in Nepal with a population of almost 29.40 million in January 2021 (Kemp, 2021).

There were 10.78 million internet users in Nepal in Jan 2021 also it's noticeable fact that 567 thousand (+5.5%) internet users increased between 2020 -2021, and it's no surprise that this increment number is of those students who got a new smartphone and other digital gadgets from their parents for their online classes (Kemp, 2021).

1.3. Problem Domain And Project As A Solution

1.3.1. Problem Domain

In 1.2, I've mentioned statistics to show the potential in the education field where technology if invented for students to help in the study could be a great idea. It's sad to know that for the Nepali education system, there are no proper, thorough-focused, and well-defined specific education materials present on the internet where students could go and search for solutions to confusion regarding any particular subject module's topic just like a tuition teacher. For e.g: why does stem or root modification occur in plants? Why is a convex mirror used as a side mirror in vehicles? What is the converse of the Pythagoras theorem? Etc.

The below bar graph shows us a result of a survey participated by 77 BIT & BBA students, where they clarified the problems they used to face during their tuition and coaching times which is 3-4 years back.

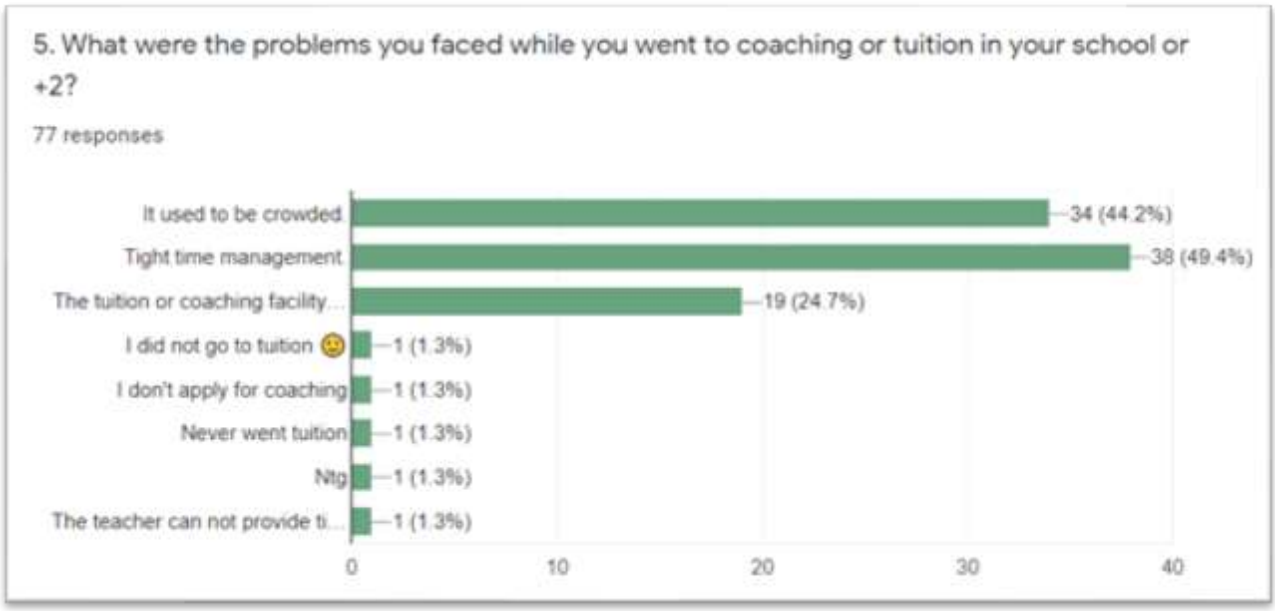


Figure 2 Problem faced in coching and tuition

Again I asked if they had any brothers or sisters going to tuition. The below is a survey result. In the below result, 57% of responders' siblings go to tuition or coaching.

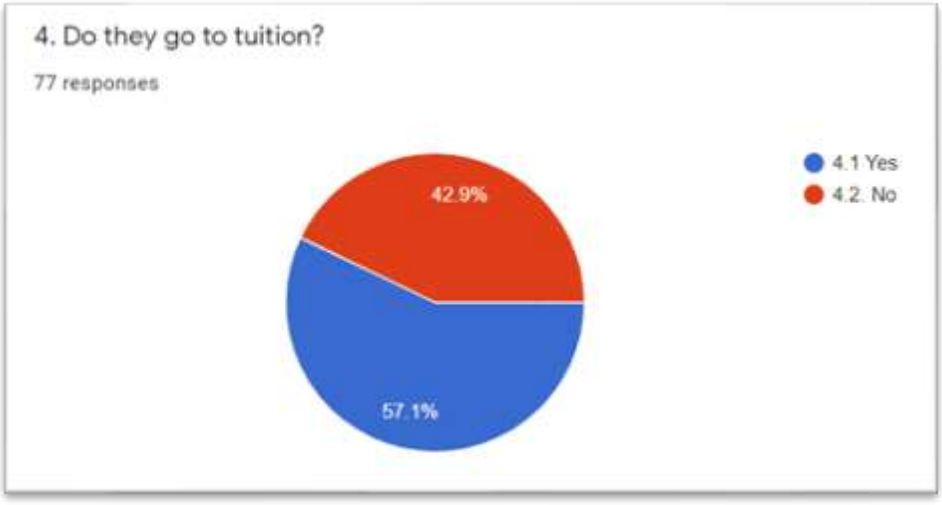


Figure 3 Survey asking if their siblings go for tuition.

The below survey results show us that, 76% of them don't have time to teach their siblings like a teacher. While others either are not able enough to teach or become irritated while teaching.

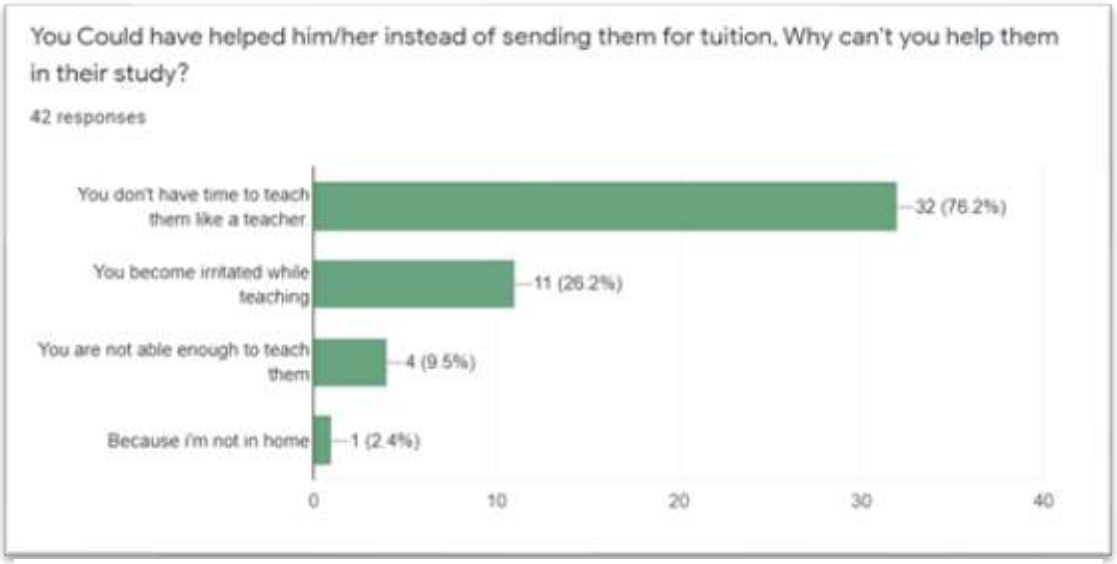


Figure 4 Survey result of why not helping siblings in their studies.

Below is a survey result of those responders, who's siblings don't go to tuition or any coaching centers. The most voted answer as the main reason for not going for help was, they searched solutions and helped out themselves from online educators, which gives us a green signal that students do use the internet to learn their unlearned topics still in 2021 by students

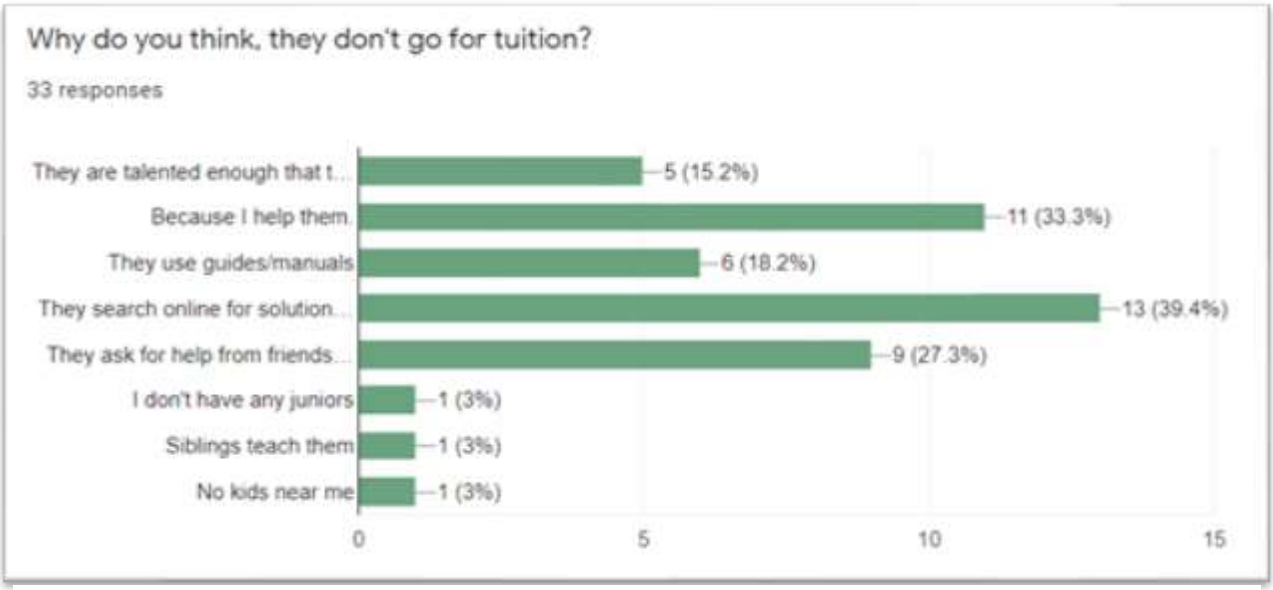


Figure 6 Survey result asking whey their siblings didn't go for tuition.

.

I asked participants if they went online on the internet to find solutions and understand book topics when they were studying in school and +2 level. In the below

Figure 5 Survey asking people if they used help online for study.

survey result, we see 80% did take help online.

In the below figure 5, we can see 41 percent of students never got their query answers and 28 percent of students were never happy with the quality of solution.

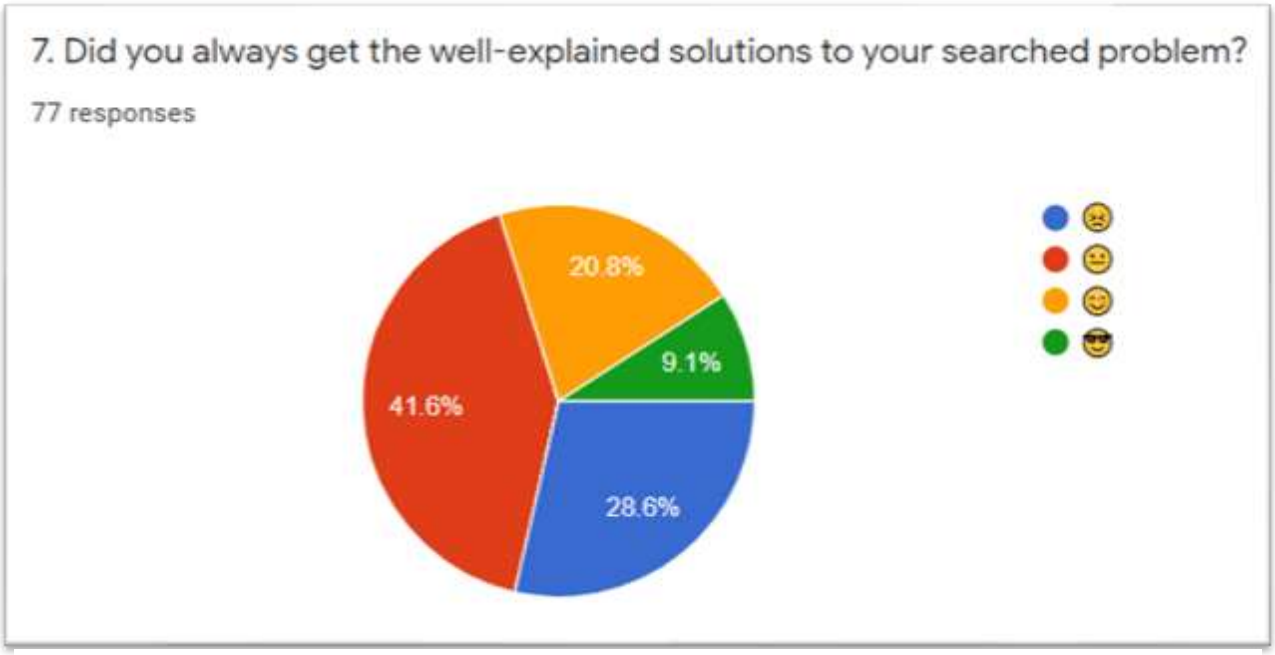


Figure 7 Survey result of whether good solutions are available or not in online platforms.

From the above survey, it was clear that our generation of students faced different problems while studying intuition and coaching centers, did search online for solutions but the satisfactory result was never found.From the survey of 77 participants, It was concluded that in Nepal there is no proper platform for students to rely on for study materials. Since 1993, the year internet was introduced in Nepal, till 2021 still after 28 years, students have to rely on tuition centers to clear their doubt on the study. There is no smart and easy way to explain different concepts of science and math by which students could understand and remember for a longer period.

1.3.2. The project as a Solution

As we have seen, students of Nepal during 2020-2021, in general, all have got their smartphones and laptops for education purposes. As in the survey, we have seen that most of the students studying at the school level, search for online help for their subject study:

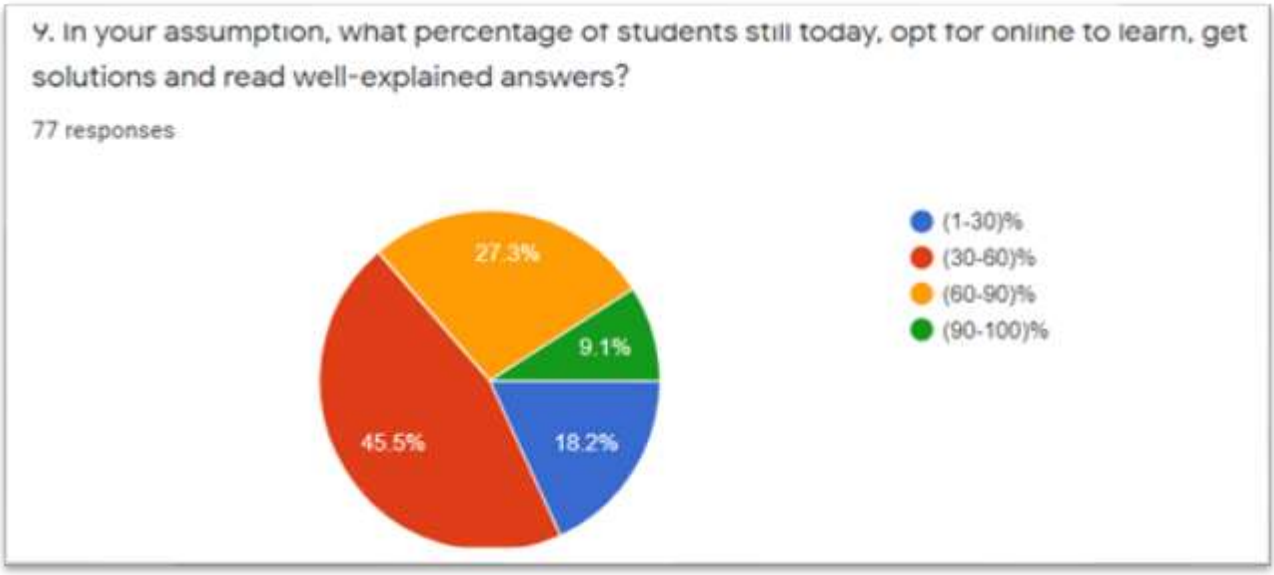


Figure 8 Survey result of assumption of what percentage of students still search online for help.

PT would be for the android platform but if possible, I would love to provide it for IOS platforms as well. Students from class 8 to 12 would be able to choose their class, then subject, then chapter of which they are having confusion and not being able to solve a problem. There they can find answers, solutions, and explained notes given by another fellow student. For the best answers, it would be voted by thumbs up by other fellow students who have already so that it can come at the top for any other researching student. If the searched question is not there, then they can simply post a question there.

By this, any student can contribute their knowledge with other fellow students, make friends and help or take help online. 'Project as a Solution' below:

i) PT will help students to revise any subject chapter-wise even if they forget after being taught in school.

ii) PT will also help parents to understand the concept of their un-understood topics to teach their small children while teaching. E.g. I had one uncle who was not able to understand geometry to teach his 12-year daughter. With the help of PT, he will be able to understand himself and teach his daughter.

iii) As PT is my FYP, so no one has to pay like tuition fee.

iv) As everyone in the 21st century is internet access, they can use PT. So no need to travel far like for tuition centers.

v) PT won't be crowded and noisy like most of the tuition and coaching centers. It will be ad-free.

vi) PT is like a school where friends are teachers because they will be learning from their friends.

vii) With PT, the students of Nepal would probably be competitive in study fields (I wish), and continuously engaged in study matters rather than being more engaged in social networking apps and online games.

1.4. Aim And Objectives

1.4.1. Aims

The main is to help every student of Nepal by building them an app, 'Prashnottar' where they can take help, share knowledge and take notes of academic subjects like science, math, Optional maths, English, Nepali, etc in case they are not able to understand in school or not able to get tuition classes for any reasons.

1.4.2. Objectives

The ways PT will be fulfilling aims are as follows:

i) PT will be built using the flutter framework and dart programming language available for both android and ios users.

ii) PT will be free of cost, it won't charge students to use.

iii) Pt will create an online platform for Nepali students and ways to get 'Prashnottar' (PT)- coins as rewards and push their ranking in PT.

iv) PT will increase awareness on taking help online using PT where they can also help other fellow students.

v) Students can ask questions, answer any asked questions, bookmark their favorite answers in respective folders, and like the favorite post of question and anwers, this'll help students to use internet also for study matters not just for entertainment.

1.5. Structure of the Report

1.5.1. Background

In this chapter, we will be getting to know in detail about the end users of PT. Then, we will see in detail about the technical terms related to the project PT. The technical terms includes detail about the IDEs, programming languages, frameworks and hardwares I used to make the project. Also we will see similar projects and comapre their features with the PT.

1.5.2. Development

In this chapter we will be introduced to different software development methodologies and the used one. The used methodology will be explained with brief explanation of each phase with the work done in those phases. Then pre survey and post survey questions will be written where as the result to the survey will be present in the appendix portion.

1.5.3. Testing And Analysis

The most important chapter where it will start with a test plan for unit testing and system testing. The tests will be done with proper screeshot evidence. Here I will be including white box testing, black-box testing, integration testing, and user evaluation testing.

1.5.4. Conclusion

The final chapter includes summary of the whole project and its journey till finishing documentation.The chapter will also explain advantages and limitations of the project. Then finally, the future works for making this application much more effective and reliable will also be included.

Chapter 2. Background

2.1. About The End Users

As 'Prashnottar' (PT) is for educational purposes, it's mainly targeted for students and teaching professionals who want to contribute their knowledge online to other students. PT will be accessible to android users. Anyone can signup and login to the app using G-mail, then set up a profile and dashboard (can start to use PT as student or instructor), then make him choose a class which he/she studies (if student option is chosen), then give options to choose any subject, then a particular chapter and start reading answers of questions asked and answered by other students all over the country. Students studying between classes 8-12 would be able to use the app experiencing an ad-free and well-designed interface.

We can estimate the age of the end-user, students could be between (13-21) and users entering as an instructor could be between (18-90). I was able to recall my school days when I used to find a lot of problems in understanding science concepts, maths, social, geography, and many more and I used to try to understand those solutions using different websites and youtube videos. But, rarely I used to find good explained papers and videos. I faced a lot of problems in school and tuition centers as well. Remembering all those ironic and golden days, I decided to make an app targeted especially at students.

2.2. Understanding The Solution (Technical Terms related to the project)

2.2.1. Overview of the system

2.2.2. Technical terms and definition

2.2.2.1. IDE (Visual Studio Code)

I used Visual Studio Code for making PT becauase it is a lightweight but powerful source code editor which makes coding flutter easy with available extensions. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity) (Visual Studio Code Getting started, 2022). It ate less memory

which was a great spec for me when I compared it with android studio, therefore I chose this IDE.

2.2.2.2. Operating System (Android)

Worldwide, android OS is the most used in mobile market with 71.7 coverage (Statcounter, 2022). Google developed android OS to be used for touchscreen, cell phones, and tablets. Also now the Google has employed android software in televisions, cars, and wristwatches which all are fitted with a unique user interface (Android Operating System, 2021).

2.2.2.3. Programming Languages

2.2.2.3.1. Dart

Dart is primarily used with Flutter to build mobile apps which could be run in both anroid and ios devices. It was launced back in 2011 at a conference in Denmark. At that time it's market and usage was very less compared to C#, JavaScript, or Java. Dart is a type safe language which uses static type checking to ensure that variable's value always matches the variables' static type, also the best part is it offers sound null safety (Dart Overview, 2022).

For me, it was great coding with Dart.

2.2.2.3.2. PHP

Php which stands for 'Hypertext Preprocessor) is a widely used general purpose scripting language. Php code is executed on the server. The best part of using Php is that it is simple for a newcomer, but has advance features for a professional programmer (PHP, 2022). By the way I used php 8.0.8 version.

2.2.2.4. Frameworks

2.2.2.4.1. Flutter

Flutter was used to make PT. Flutter is an open source framework developed by Google for building beautiful, natively compiled, multi-platform applications by just a single codebase. The first version of Flutter known as 'sky' ran only on Android Os which was 2015 back at Dart Developer summit on December 4,2018. Flutter is UI library of different toolkit for developers to build any applications beautiful (codemagic, 2019).

2.2.2.4.2. Laravel

Laravel is an open-supply PHP framework, which is strong and smooth to understand. It follows a model-view-controller layout pattern. Laravel reuses the prevailing additives of various frameworks which enables in growing an internet software. The net software hence designed is extra based and pragmatic. Laravel gives a wealthy set of functionalities which includes the simple functions of PHP frameworks like CodeIgniter, Yii and different programming languages like Ruby on Rails. Laravel has a completely wealthy set of functions with the intention to enhance the velocity of net development (Laravel - Oview Tutorialspoint, 2022).

2.2.2.5. Database (Mysql)

MySql is a relational database management system (RDBMS) developed by Oracle which is based on structured query language (SQL). MySql is the most popular recognized technologies in the modern big data ecosystem enjoyed by all IT professionals and companies (Talend, 2022).

2.2.2.6. Packages used in the project

Ten dart packages were used in the project which were and installed from pub.dev. The packages are listed below:

1. animated_splash_screen → Used to create splash screen effect.
2. google_fonts → Used to change the font when required.
3. flutter_local_notifications → Used to notify used that your post has been liked.
4. cupertino_icons → To use default set of icon used by Flutter's Cupertino widgets.
5. flutter_svg → Used to use svg format images.
6. flutter_inset_box_shadow → Use
7. font_awesome_flutter → It was used to put neumorphism effect on the app.
8. http: → to make HTTP requests.
9. shared_preferences: → Used to get access to the storage of the device.
10. image_picker → Used to take the image from the image library.

2.2.2.7. API (REST API)

REST API stands for **RE**presentional **S**tate **T**ransfer and **A**pplication **P**rogramming **I**nterface. It is an architectural style that defines a set of rules to create web services. In

a Client-server communication, REST instructs to create an object of the data requested by the client and send the values of the object in response to the user.

Reasons to choose Rest API:

i) It is easy to learn and understand.
ii) We can organize complicated applications and make easy use of resources.
iii) The high load can be managed by HTTP proxy and cache.
iv) HTTP procedure call-outs are carried to retrieve data and requests.
v) With the use of Oauth protocols verifying REST requests, the security of a network is strong.

2.3. Similar Projects

Reviewing similar systems or apps which would be resonating with my solution was a great way to learn many things. Such as designs, technologies,

approaches to users, etc. Below were some similar apps which I studied for further clarification:

2.3.1. Answers

'Answers' is the homework helping app and contributes to finishing students' faculty assignments from math and chemistry to records and biology. The app is devoted to assisting college students with fixing phrase troubles, solving math equations, entire technological know-how homework, and lots more! User-pleasant Q&A webpage is the ad-free, fast, and limitless homework helper that scholars can matter on. Answers have over a hundred million questions and solutions and near 10,000 observe subjects like math, algebra, chemistry, calculus, technological know-how, physics, and biology (Google Play, 2021).

I have been using this app to know its potential and usage among students, I find it very helpful. The ad-free user interface and reward system made me a happy user. We can search for answers by writing a question in the search box, surprisingly it brings good answers. But, it seems that the app is still in the development phase. It's not completed yet. When we click particular subjects, then non of the subject is chosen.

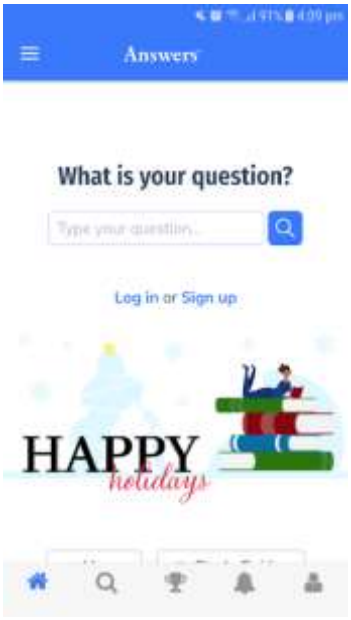


Figure 9 Answers welcome pagte

Figure 12 Subject Dasbboard

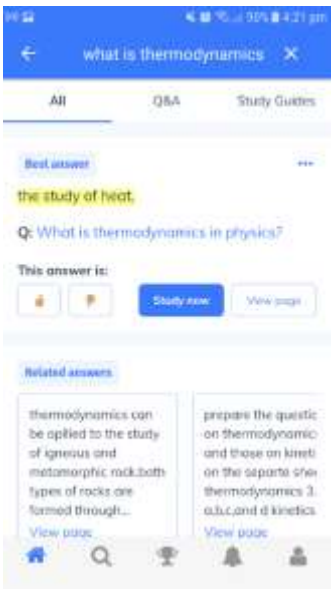


Figure 11 Question asking

Figure 10 writting answer to the question

2.3.2. Bartleby

Bartleby is a homework helper app for students. It has a math answer scanner which will help to get a solution faster and easier. If we ask in the search bar, we will get good answers. We can choose a particular school textbook to answer to, it includes math, science, history, geometry, chemistry, engineering, business, and many more. We can snap a photo or send queries to our particular subject matter, we would be getting a proper answer within 30 minutes.

The only thing which I didn't like is that most of the basics and valuable features for students are available only for subscribed students (Google PlayStore, 2021).

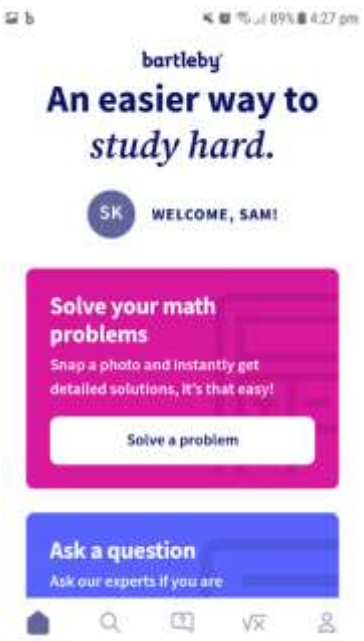


Figure 13 Bartleby Welocome Page

Figure 14 Bartleby questioning section

Figure 15 Bartleby math sover section

2.3.3. Mimo

Mimo is an online platform to learn to code, especially web designing and python language. It teaches users from basics to advance. It has a topic-wise explanation and exercises with the best animation and reward system. It allows us to make friends online, with coins and the streak we achieve, it creates competitiveness among each other (Google Play_Mimo, 2021). I liked everything about Mimo, the only thing they can improve is, they could add more topics to learn not just only web development and python.

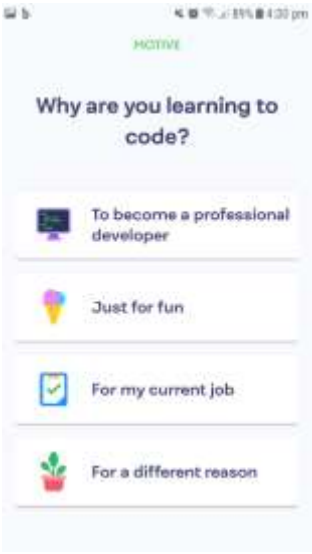


Figure 19 Mimo welcome section

Figure 18 Mimo asking experience in coding

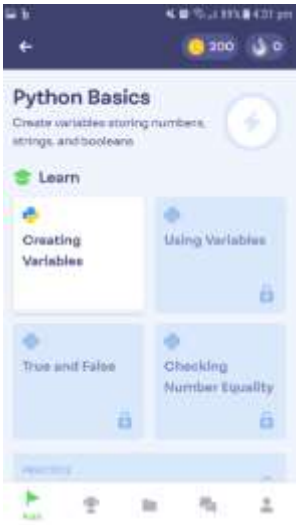


Figure 16 Python Dashboard

Figure 17 Mimo exercise section

2.3.4. Saathi

Saathi is a free MCQ app for Maths and Science with over 250000 questions which all are created by IITians for students of classes 6,7,8,9,10. It can be used as a revision app to help them score more in the exam. There are Personalized Question recommendations powered by Saathi AI. It provides us a Real-Time Statistics and analysis of reports of our study. It provides a beautiful animated user interface that boosts students studying spirit (Google PlayStore, 2021).

Figure 23 Saathi splash screen

Figure 22 Welcome page

Figure 21 Choosing class in Saathi app.

Figure 20 Math section in Saathi

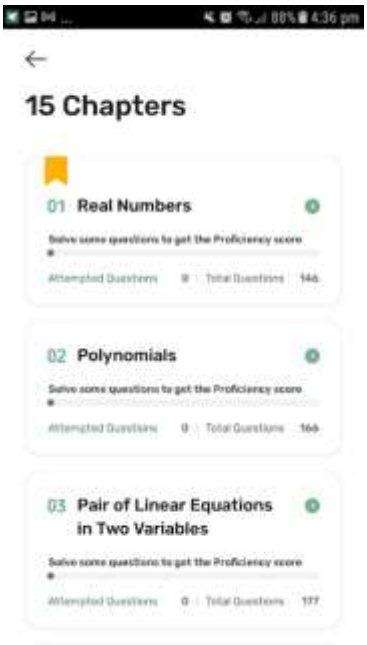


Figure 25 Chapter section in saathi

Figure 24 Quiz section in Saathi

2.3.5. Khan Academy

For students, in Khan Academy, there are thousands of interactive exercises to do, videos to watch and learn, and lots of articles to read. We can study math, science, economics, history, and many more. Practicing exercises, quizzes, and attending tests can push students' academic performance. We can download videos and bookmark favorite topics which we could watch and study offline. There's nothing I didn't like about Khan Academy.

Figure 29 Welcome page in Khan Academy app.

Figure 28 Subject section in Khan Academy app.

Figure 27 Class choosing section in KA

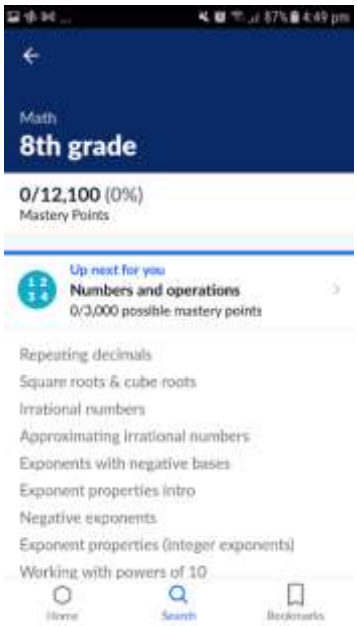


Figure 26 Math Chapters in 8th grade class of Khan Academy

2.3.6. MeroStudy

MeroStudy is a Nepali educational mobile app and web portal which brings contents entire educational sector. It continuously provides information regarding the educational resources We can get notification and update of reslut visited educatioinal website in Nepal (Google play store, 2021).

The only things I liked is, it not only want to deliver educational topics to students but also it brings news on different vacancies, events, scholarship, admissions, universities, courses and many more. But, I don't know why, it's been two months, I am trying to use this app but it's not working, only tab bar and user settings are opened and all contents are empty.

Figure 32 MeroStudy welcome Page

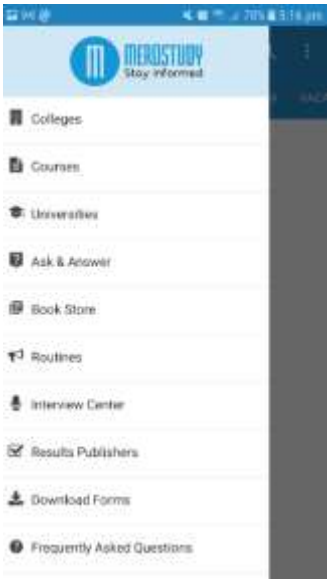


Figure 30 MeroStudy settigns

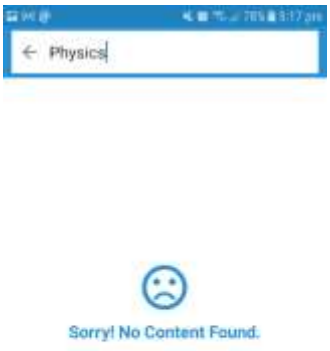


Figure 31 MeroStudy question asking section

2.3.7. Neema Academy

Neema Academy is a Nepali ed-tech startup formed in 2018. It offers on-demand videos, 3D animations, game-based teachings, textual explanations, and many more study topics. It aims to bridge the gap between in-class learning and visual-knowledge-based learning.

In the app, we cannot search for answers by posting questions but search only for subjects or chapters. It doesn't contain bookmark features, but it does have a progress page where we could view our progress over the week.

Figure 34 Neema Academy User Dashboard

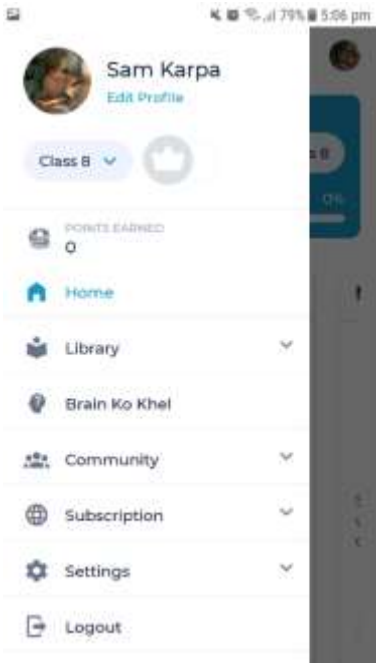


Figure 35 Neema Academy settings section

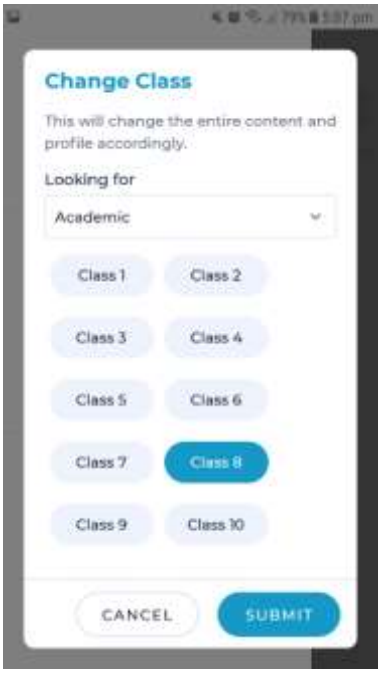


Figure 33 Choosing class in Neema

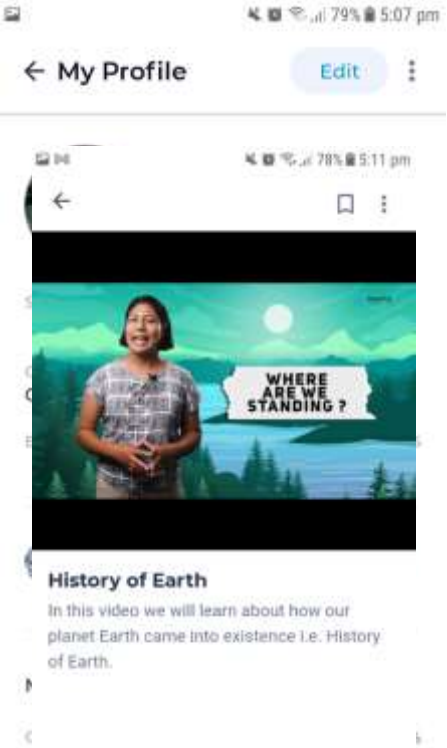


Figure 38 User profile in Neema Academy

Figure 39 Video explanation section

Figure 37 Subject Section in Neema Academy

2.3.8. Kullabs

Kullabs is an app developed by Kul Techno Lab and Research Centre. Their free product kullabs.com has been providing service to more than 70000 users. There are notes and related videos from youtube for each listed topic. There are solved exercises from each topic. We can select class, then subject, and then can choose any topic to study or revise.

Figure 36 Chapter section

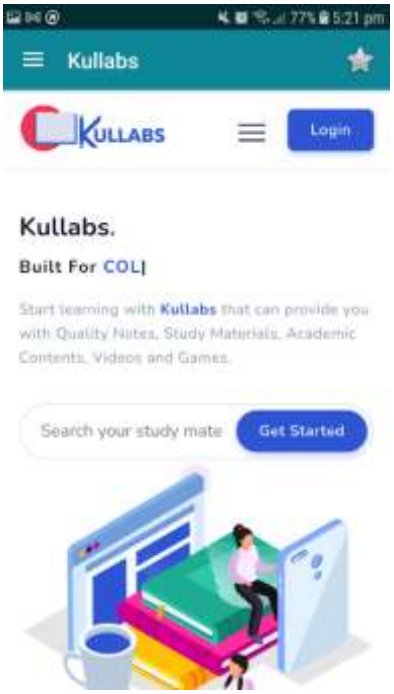


Figure 42 Welcome Page Khullabs

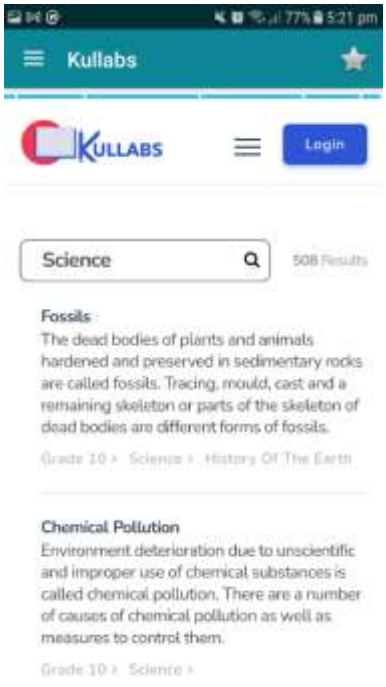


Figure 40 Khullabs question answer section

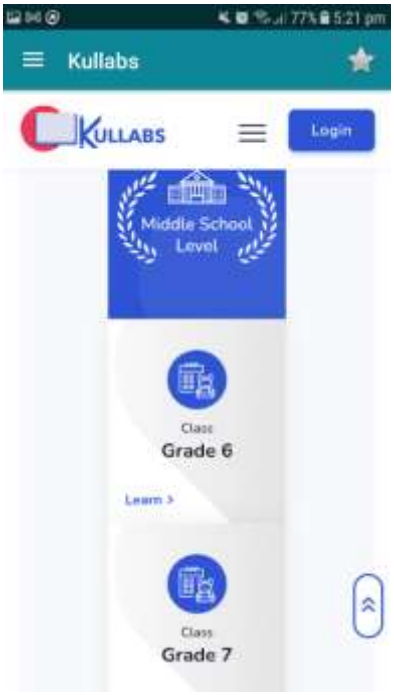


Figure 41 Choosing class section in Khullabs

Figure 45 Subject Section

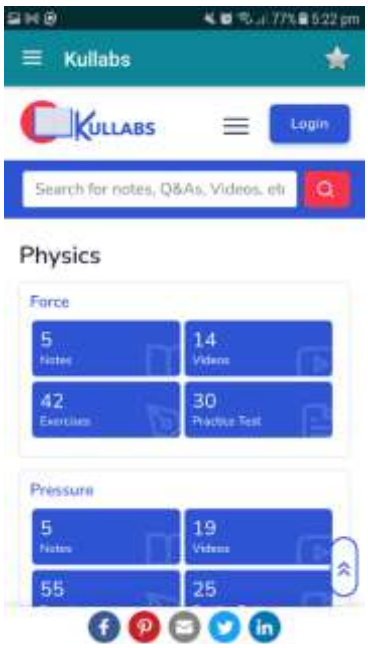


Figure 43 Khullabs subject section

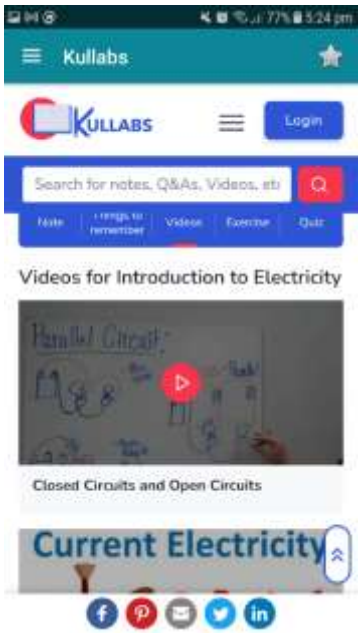


Figure 44 Khullabs video section

2.4. Comparisons

2.4.1. Comparison based on feature.

Features	Answers	Bartleby	Mimo Saathi	Saathi	Khan Academy	MeroStudy	Neema Academy	Kullabs
Log in/Logout	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Register	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Choose Class	No	No	No	Yes	Yes	No	Yes	Yes
Choose Subject	No	No	No	Yes	Yes	No	Yes	Yes
Choose Chapter	Yes	Yes	No	Yes	Yes	No	Yes	Yes
Ask Question Post	No	Yes	No	No	No	No	No	No
Share picutre also in Question post	No	No	No	No	No	No	No	No
Edit or Delete Post	No	No	No	No	No	No	No	No
Like/Dislike	No	No	No	No	No	No	No	No
Notification on Like	No	No	No	No	No	No	No	No
Answer to question	No	No	No	No	No	No	No	No
Update/Delete answer	No	No	No	No	No	No	No	No
Update User Profile	Yes	Yes	Yes	Yes		No	Yes	No

Table 1 Comparision based on features

2.4.2. Comparison based on feature.

Name of Similar Application.	Positive Analysis.	Negative Analysis.	Features liked or taken.
1. Answers	Beautiful and simple UI. Users can ask a question and get an answer.	Subjects mentioned in the app don't function. The Reward system is not working.	Learned to make UI as simple and consistent much.
2. Bartleby	Simple and consistent UI. Only three dashboards are present on the home page but the whole app is functional. Math question scanner feature is also present.	Like, comment and upvote features are not present. It cannot help users to judge which answer is better.	Question scanner feature if implemented, the app could be better.
3. Mimo	Beautiful course and user dashboard, the reward system is best, able to make friends in this platform. Beautiful animated designs.	Only a few courses are available(web design and Python).	A better user dashboard and use of animation would be better for student apps.
4. Saathi	Beautiful logo buffering while loading, simple and consistent design. Bookmark and report display features are best.	Only two subjects, Math and Science are present.	Bookmark and statistics report display of user would be great.

5. Khan Academy	Ad-free, clean, and pleasant User experience.	Everything is best.	An ad-free and systematic approach is needed.
6. MeroStudy	It doesn't work.	There's no any content to read, study or watch.	There was nothing to be liked. I think the software company is not maintaining the app.
7. Neema Academy	I like their all of the features, especially that progress page over the week.	Most of the features are only for subscribed members.	It has a very clean and systematic UI, which I liked a lot.
8. Kullabs		Sometime there comes an ad coming when scrolling	It has prepared contents according to class and subjects like I would.

Table 2 Critical analysis of the feautres.

Chapter 3. Development

3.1. Considered Methodologies

Software development life cycles (SDLC), is a process used by software engineers to ensure the quality and correctness of the built software. It makes sure, the development is completed in the pre-defined time frame and cost. It tracks and controls different development phases of software development. With speed, it reduces project risks and enhances project management overall with a result of a happy client relationship. (Guru, 2021) Different phases of SDLC are as follows:

- Requirement Analysis
  Feasibility Study
- Design
- Coding
- Testing
- Install Deploy
- Maintainance

Popular SDLC models which I considered using, but was not familiar with some of them are:

- Waterfall Model -
- Prototype Model -
- Incremental Model
- V-Model -
- Spiral Model -
- Big Bang Model
- Rational Unified Process (RUP)

Let me explain a few of the above-mentioned methodologies.

- **Waterfall Model.**

  The waterfall model is the earliest Process Model to be introduced. The development process is divided into separate phases. Each phase must be

completed before the next phase can begin so there is no overlapping. The output of one phase will be acting as the input for the next phase sequentially. (Point, 2020)

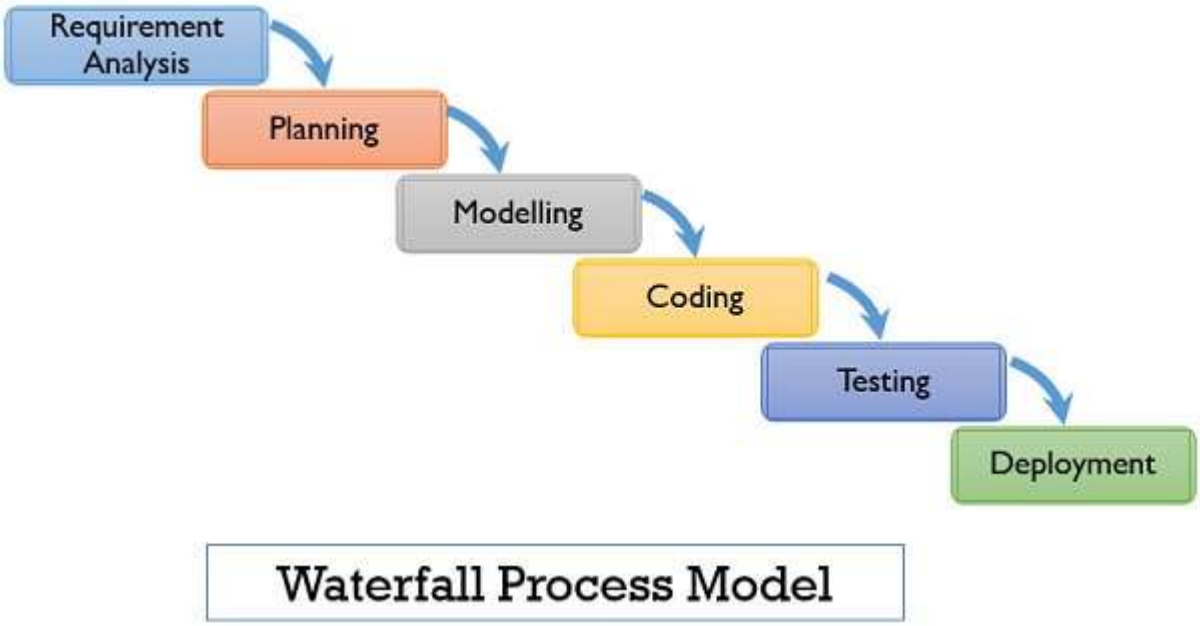


Figure 46 Waterfall model

Advantages of Waterfall Model

- It is quite simple and smooth to recognize and use.
- The requirement is thoroughly documented, clean, and fixed.
- Technology is thought and isn't always dynamic.
- There aren't any ambiguous requirements.
- Ample sources with required information are to be had to guide the product.
- The task is short. (Point, 2020)

Disadvantages of Waterfall Model

- Until late during the life cycle, no working software is produced.

- Too much risk and uncertainty.
- Bad for complex and object-oriented projects.
- Terrible for long and ongoing projects.
- Cannot assist change requirements. (Point, 2020)

❖ Incremental Model

Here, requirements are divided into multiple standalone modules of the development cycle. Each module must go through the requirements, design, implementation, and testing phases. Succeeding the release of the module adds function to the previous release. The process is continued until the system is completed. (JavaTPoint, 2020)

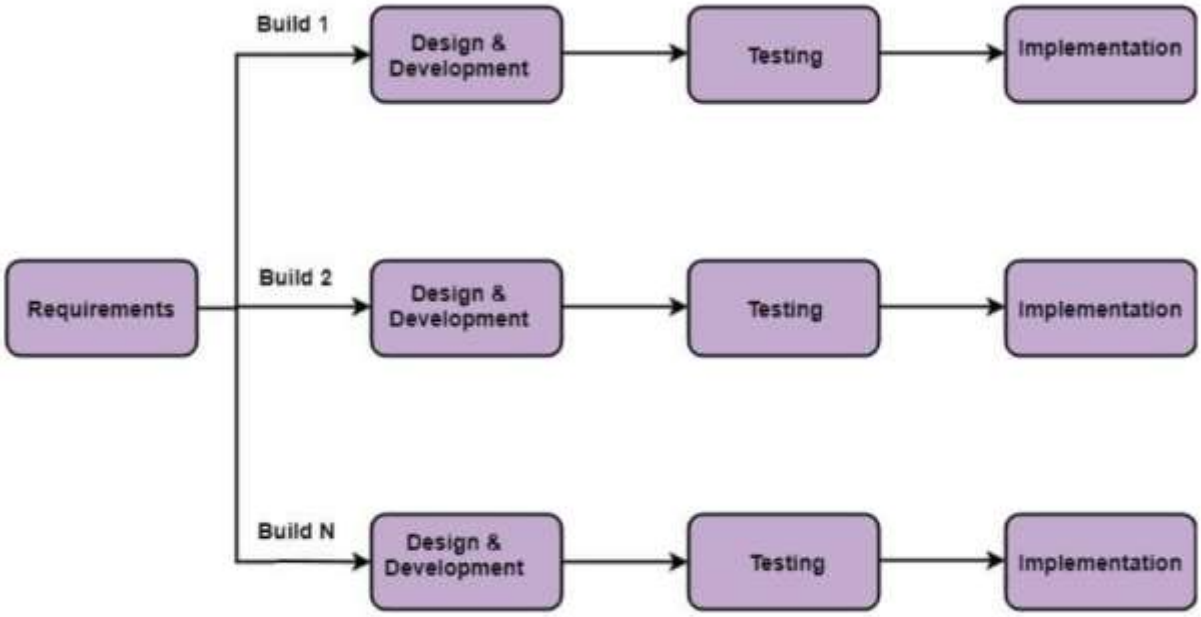


Figure 47 Incemental model

Advantages of Incremental Model.

- Errors are easy to recognize.
- Testing and debugging are easy.
- It is more flexible.

- Risk can be managed simply.
- The client is important. (JavaTPoint, 2020)

Disadvantages of Incremental Model.

- Good planning is needed.
- The total cost is high.
- Good module interfaces are needed. (Point, 2020)

❖ **Spiral Model**

This model combines the idea of iterative development with systematic, controlled aspects of the waterfall model. It allows incremental releases of the product through each iteration around the spiral. (tutorialspoint, tutorialspoint, 2021)

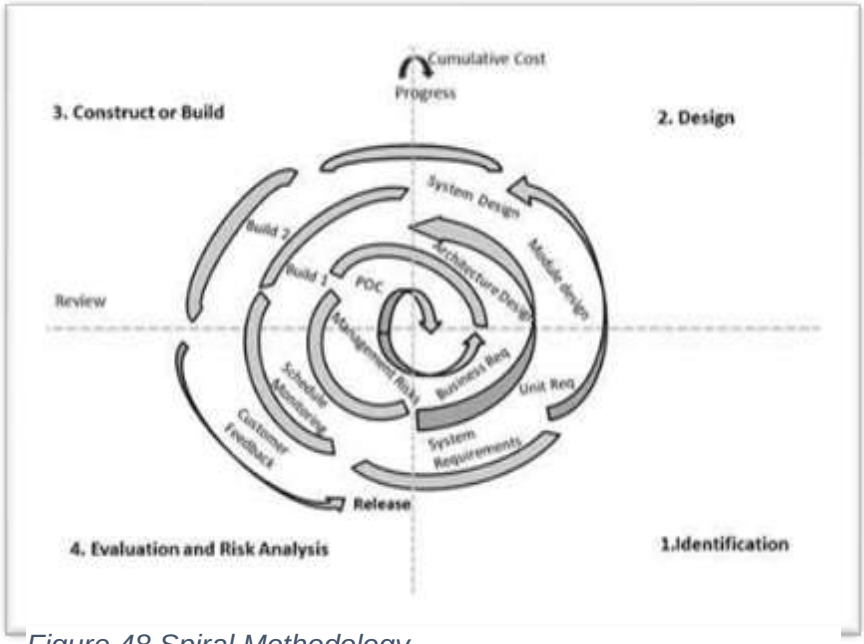


Figure 48 Spiral Methodology

Advantages of the Spiral model:

- Requirements can be added as it needs a change.
- The explosive use of prototypes is allowed.
- Requirements are recorded precisely.
- Users can view the system early.
- The whole development can be divided into smaller parts.
- Risky parts can be developed earlier which helps in risk management. (tutorialspoint, tutorialspoint, 2021)

Disadvantages of Spiral-Model

- Management becomes more complex.
- The project end date cannot be known earlier.
- Spiral might not go as defined.
- Intermediate stages require excessive documentation. (tutorialspoint, tutorialspoint, 2021)

3.1.1. Justification for Selecting or Rejecting the above Methodology

3.1.1.1 Justification for rejecting Waterfall Model

Project Scenario	Justification
The FYP must be completed within six months and must have successful development stages and a clear deadline.	This model is too much risky and is uncertain.
The FYP is complex and works around some classes and objects.	It is very bad for complex and object-oriented projects.
With time, the project might include different features and designs.	It cannot assist change requirements.

Table 3 justification for rejecting waterfall model

3.1.1.2 Justification for rejecting Incremental Model

Project Scenario	Justification
Need for good planning designing.	A very good planning design is not always to be expected by a student.
Well-defined module interfaces are needed.	The not very module can be finished on time according to Gantt and if not, then further work cannot begin.
Rectifying a problem in one unit requires correction in all the units.	It consumes a lot of time

Table 4 Justification for rejecting Incremental Model

3.1.1.4 Justification for rejecting Spiral Model

Project Scenario	Justification
It has complex processes.	High experts and professionals are required to run the model.
No end goal.	At the early stages of the project, the end date of FYP cannot be evaluated.

Table 5 Justification for rejecting Spiral Model

3.2. Selected Methodology

RUP(Chosen Methodology):

I have chosen Rational Unified Process (RUP) for the methodological roadmap to build 'Prashnottar'. It is based on Agile methodology which splits the project life cycle into four phases where on each phase, all six core development disciplines take place which are: business, modeling, requirements, analysis and design, implementation, testing, and deployment. (Study, 2021)

I have chosen RUP to create high-quality software with a predictable budget and time frame. The most beautiful thing about RUP is, each of the life cycle phases can be repeated, if needed.

3.3. Phases Of Methodology

. Let us discuss the 4 phases of RUP in detail:

Inception

I. Scheduling Resources
II. Cost and Time Estimation
III. Planning
IV. Risk Management
V. Prototypes and Development

In the Inception phase, we will have a general vision for the project initiative with multiple parameters. We will get the project scope. (Master, 2020)

Elaboration

I. Analysis of problem domain
II. Use Case Diagram Development
III. System Architecture Development

In the Elaboration phase, we will get functional and non-functional parameters. We will understand the full Software Architecture Description. We might be able to justify whether to prove the project plan or not. We will fully have the result of actual resource cost versus planned resource. (Master, 2020)

Construction

I. System Build
II. System Operational Manual
III. User Manual
IV. Test Cases

In the construction phase, we will be ready to develop all components and features and integrate them into the product. We will fully focus on managing resources to optimize

costs, schedules, and quality. The software will be designed, written, and tested successfully. (Master, 2020)

Transition

I. Training
II. Beta Testing
III. Analysis of User's Review
IV. Supporting & maintaining product

In the Transition phase, the last phase is the phase where the product is finally finished, released, and delivered to the customer. In this phase, bugs will be fixed, correct the problems and finish the features which were postponed. This is the phase of deployment after successful beta testing. (Master, 2020)

Strong reasons why I chose RUP:

Project Scenario	Justification
It follows Agile principle number 2.	It welcomes required changes, even late in development.
Documentation is power for any project.	It provides proper documentation of the software.

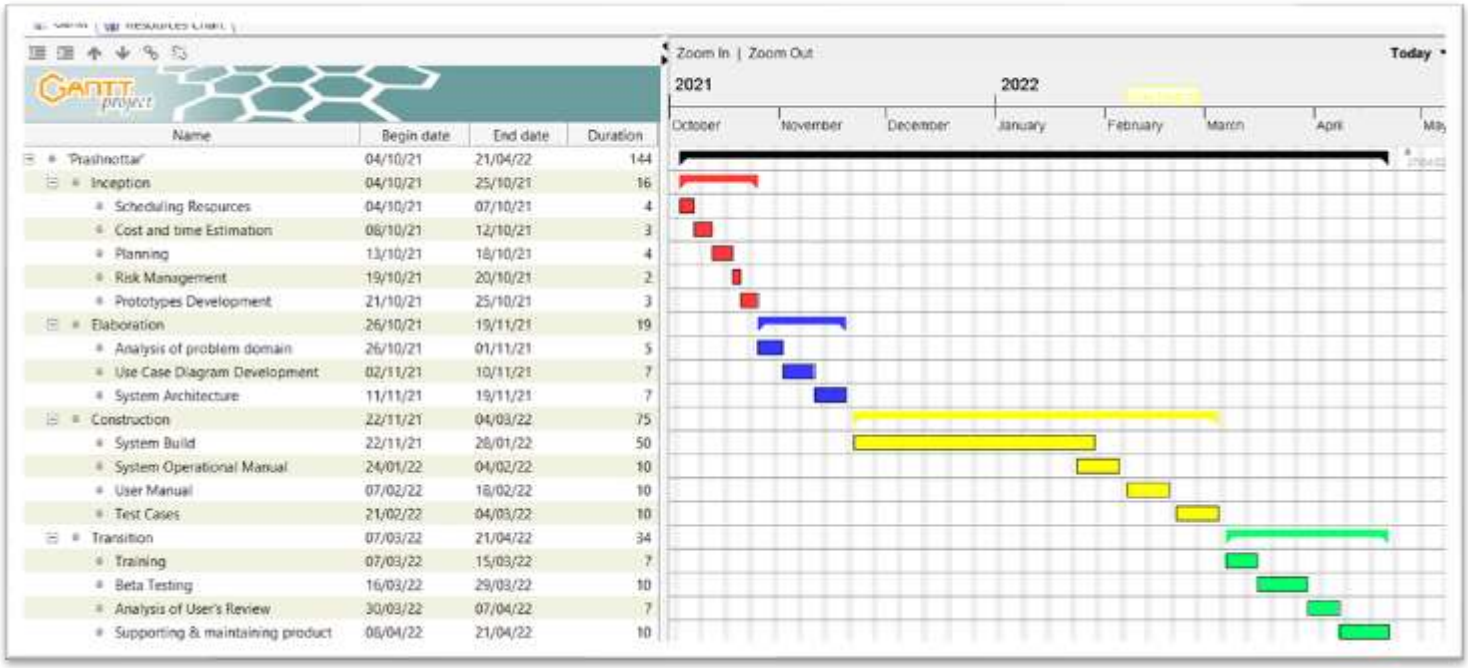
Table 6 Reasons for selecting RUP methodology

- It helps to change requirements in the project whether they are coming from customers or from the project itself.
- It welcomes change at any time.
- It provides proper documentation of the software product.
- It helps to find issues early in the process life cycle.
- It reduces development costs and improves process control and risk management.

Cons of RUP

- Individuals must be experts and professionals in their respective fields to develop.

- The integration in the development process might hurt some more fundamental activities such as testing.
- Multiple stages of the workflow might be complex.
- It is challenging for organizations to implement which has small teams or projects. (Master, 2020)



Name	Begin date	End date	Duration
Prashnottar	04/10/21	21/04/22	144
Inception	04/10/21	25/10/21	16
Scheduling Resources	04/10/21	07/10/21	4
Cost and time Estimation	08/10/21	12/10/21	3
Planning	13/10/21	18/10/21	4
Risk Management	19/10/21	20/10/21	2
Prototypes Development	21/10/21	25/10/21	3
Elaboration	26/10/21	19/11/21	19
Analysis of problem domain	26/10/21	01/11/21	5
Use Case Diagram Development	02/11/21	10/11/21	7
System Architecture	11/11/21	19/11/21	7
Construction	22/11/21	04/03/22	75
System Build	22/11/21	28/01/22	50
System Operational Manual	24/01/22	04/02/22	10
User Manual	07/02/22	18/02/22	10
Test Cases	21/02/22	04/03/22	10
Transition	07/03/22	21/04/22	34
Training	07/03/22	15/03/22	7
Beta Testing	16/03/22	29/03/22	10
Analysis of User's Review	30/03/22	07/04/22	7
Supporting & maintaining product	08/04/22	21/04/22	10

Table 7 Gantt Chart after analyzing the whole project through RUP phases

3.4. Survey Results

3.4.1. Pre-Survey Questions are below:

Questions which were asked in the Pre-Survey quesitons are as follows:

1. Your Name

   _____

2. Your email address:

   _____

3. Do you have any brothers or sisters, or neighboring children studying in class from 4-12?
   - Yes
   - No
4. Have you ever helped them or any of your juniors with studying? (E.g. Helping in Math homework.)
   - Yes
   - No

5. Do they go to tuition?

   3.1 Yes

   5.1. 1. You could have helped him/her instead of sending them for tuition, Why can't you help them in their study?
   - You don't have time to teach them like a teacher.
   - You become irritated while teaching.
   - You are not able enough to teach them.
   - Other reasons: _____

   5.2. No
   - They are talented enough that they don't need to go to tuition.
   - Because I help them.
   - They use guides/manuals.
   - Search online for solutions or to understand the concept of their subject matter.
   - They ask for help from friends or teachers.

6. When you were studying till senior high school(+2), which platforms, websites, or apps did you use to take help for study? (Optional to answer)

   _____

7. Which all of below do you use most of the time, at least once a day?
   - Quora
   - Stack Overflow
   - Medium
   - Qapop
   - Reddit

- Twitter
- Youtube

8. Which other platform do you use to post a question to get answers or read others' answers for any other questions? (For e.g Quora)(Optional to answer)

   _____

9. Which platform do you specifically use for professional use? e.g. I usually use stack overflow for different errors and fix bugs.

   _____

10. Would you recommend this app to your brothers and sisters?
    - Recommend
    - Strongly recommend
    - Neutral
    - Don't recommend
    - Strongly Don't recommend

11. Would students use this app, if it becomes helpful as it was said above in the objectives?

Don't know. Doubt it. Neutral Sure Would love it.

12. If you were to go back in time and be again in school life, would you use 'Prashnottar'?
    - 👍
    - 👎

13. Would you think, 'Prashnottar' would be helpful for students?

No Might be Hope so Yes Would be helpful.

14. Rate the FYP project in terms of its usefulness and uniqueness.

Useless Boring Nice Brilliant Mind-Blowing

15. It would be great if you could leave your wonderful insights and helpful comments for more edges to improve. (No word count restrictions. I would be happy to read all of those.)

_____

16) What were the problems you faced while you went to coaching or tuition in your school or +2?

- It used to be crowded.
- Time management.
- The tuition or coaching facility was too far.
- Others.

17) Did you ever search for online help to deal with your study confusion when you were studying in school or +2?

18) What percentage of students still today, opt for online to get solutions and well-explained answers?

1-30%

30-60%

60-90%

90-100%

18) Did you always get the well-explained solutions to your searched problem?

19) What features below would be good for everyone's convenience?

i) easy search implication

ii) Correct and well-explained answers, pictures from individuals.

iii) No ads, clean and clear UI/UX.

iv) Others.

The picure of the responses and their explanation would be given in the appendix portion.

3.4.2. Post-Survey Results

Questions of Post Survey Results are below:

1. Your name
2. Do you think students search for answers and solutions online to solve their confusion and problem?
3. Do you also prefer online help more than physical help for your problems related to study?
4. Do you think like & comment would be nice to put on answers to the post in Prashnottar?
5. Do you think Prashnottar would be useful and effective?
6. Rate the project in terms of its uniqueness.
7. Would you recommend Prashnottar to juniors ?
8. Could you have used Prashnottar if you were in school?
9. Is there any suggestions for the project?

The picture of the responses and their explanation would be given in the appendix portion.

3.5. Requirement Analysis

3.5.1. Feature Requirements

As we know, the PT is an app for students' productivity and welfare hence, it has its respective features which were required. The features required in the apps would be login, register, choose class, choose the subject, choose a chapter, choose to make a post for asking the question by sharing the picture, and on that post, other users must be able to like and answer the comments. Every user must have the privilege to update their answers or post. They must also be able to update their profile and name.

By saving this much information, another user would be able to read and study different concepts of science and math. All of the user's data are stored in mysql with the help of

laravel. These were some features that were not currently in this application. Features like bookmarking a post and following a person were not present.

3.5.2. Hardware Requirements

Hardware requirements while making the Prashnottar were as follow:

i) Android phone with Android version more than 9.
ii) Laptop
iii) Internet

3.6. Design

3.6.1. Logo of the Application

Figure 49 Logo of prashnottar.

3.5.2. System Overflow Diagram

3.5.2.1. From context of Flutter application

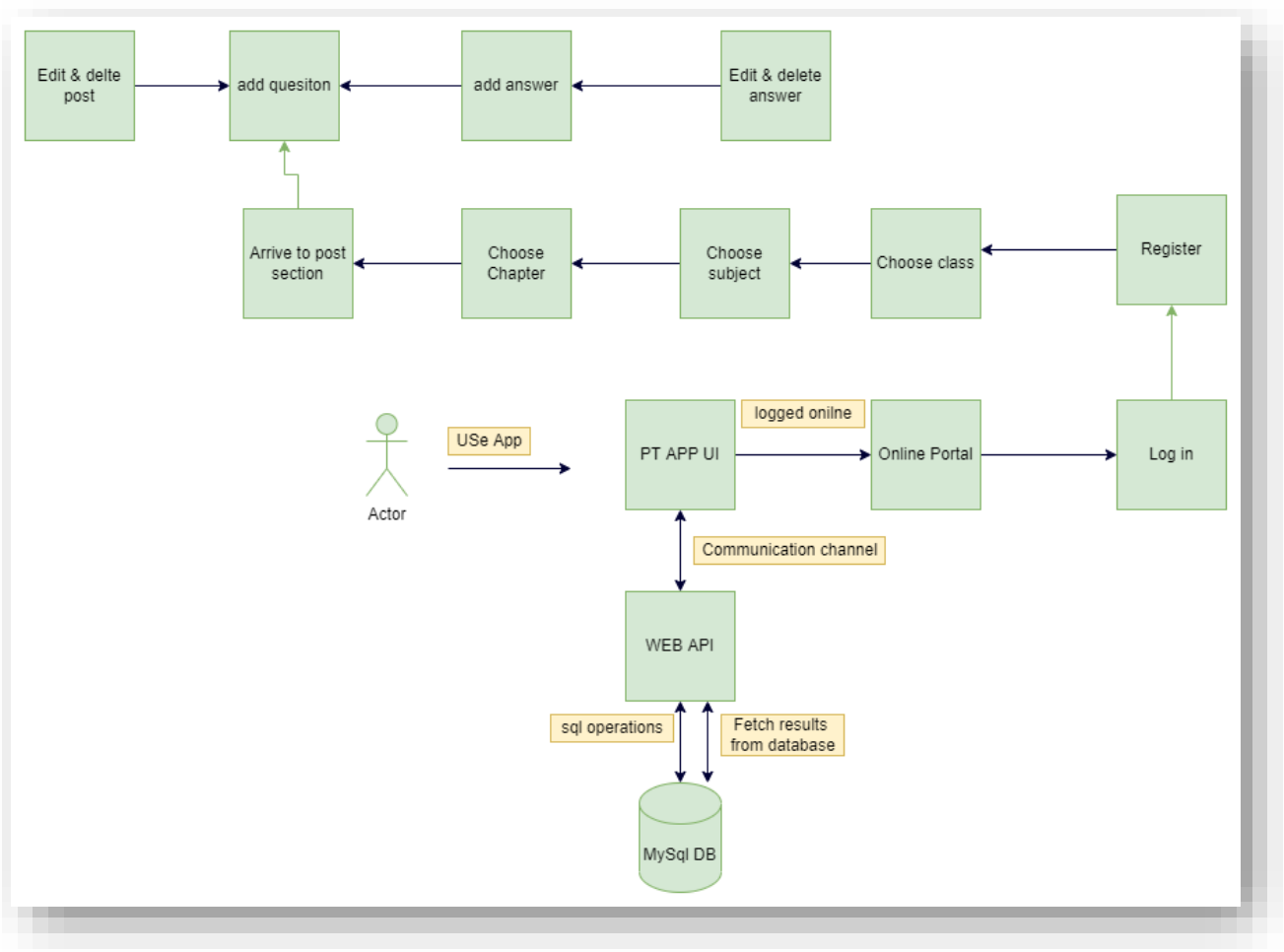


Figure 50 System overflow Diagram from Context of Flutter Application

3.5.2.2. System over flow from the context of Web API (Laravel)

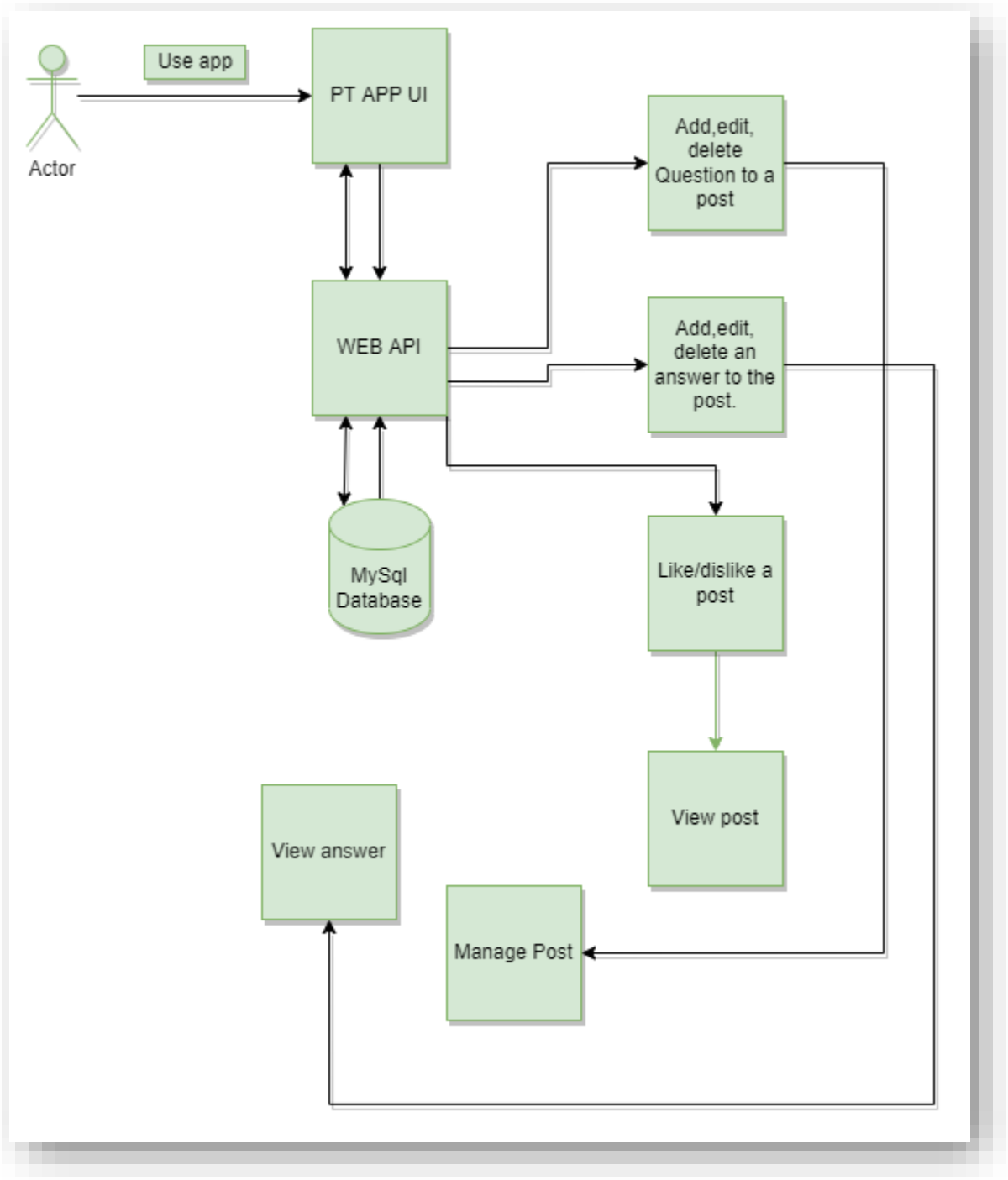


Figure 51 System Overflow Diagram from Context of Laravel API.

3.5.2.3. Entity Relationship Diagram of Overall System

3.5.2.5. Final Use Case Diagram.

Final use Case Diagram is placed in appendix D.

3.5.2.5. Wireframes

Wireframes are included in appendix D.

3.5.3. UML Diagrams of different features.

3.5.3.1. Log in feature UML diagrams

High level use case

Use case name	Login user
Actor(s)	User
Description	A user can log in to the system if he/she has already registered otherwise he/she should register first to log in to the system.

Figure 52 High level use case of Log in User

Use case diagram

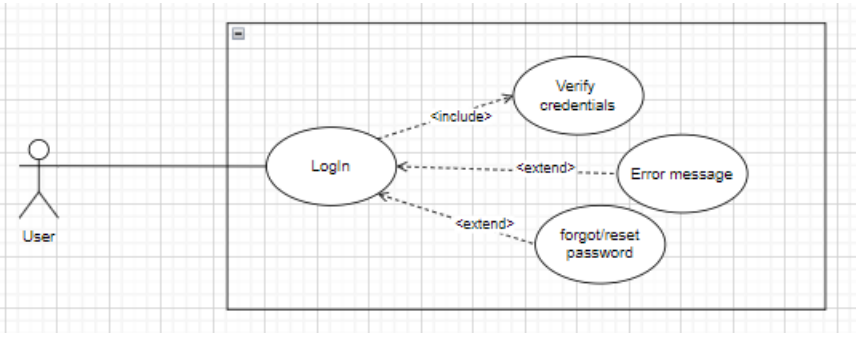


Figure 53 Use case diagram of Log in

Sequence diagram

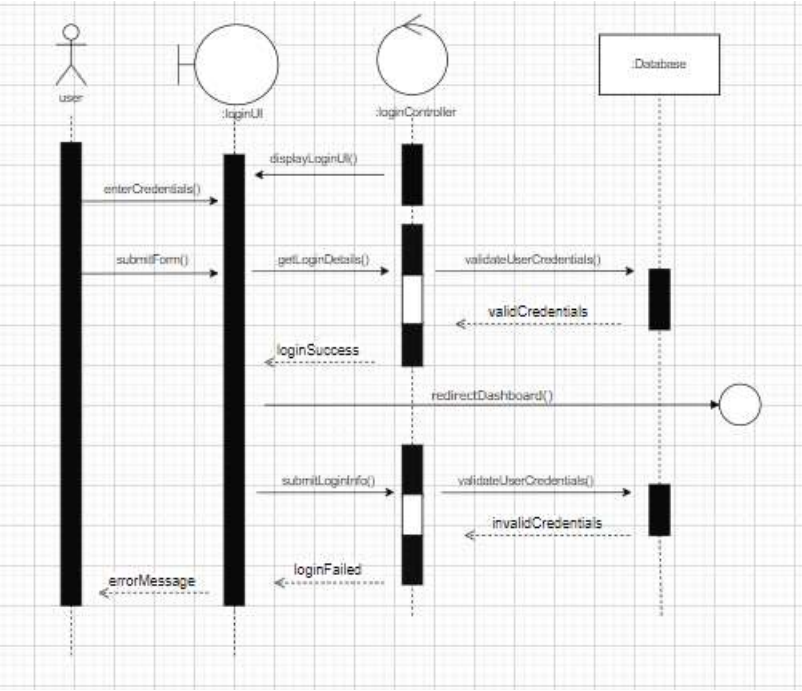


Figure 54 Sequence diagram of log in

Collaboration diagram

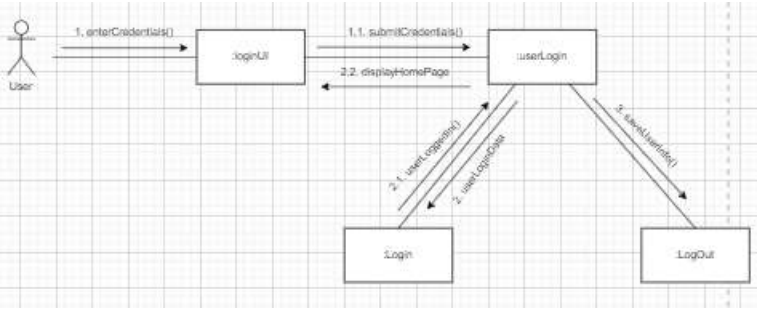


Figure 55 Collaboration diagram of log in

3.5.3.1. Register feature UML diagrams

High level use case

Use case name	Register user
Actor(s)	User
Description	A user should register first to login into the system. A user should fill the necessary details in registration to register his/her profile in the database.

Figure 56 High level use case of Register user.

Use case diagram

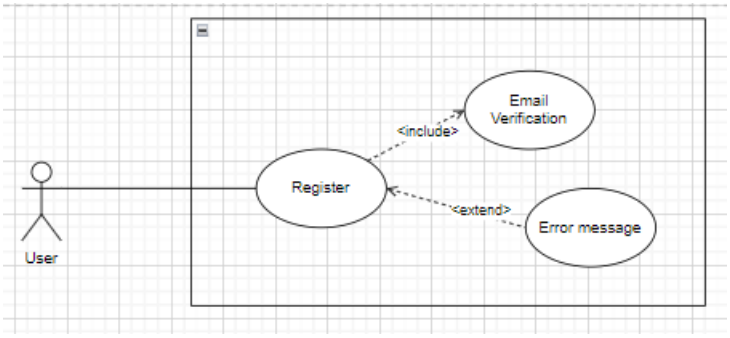


Figure 57 Use case diagram of register

Sequence diagram

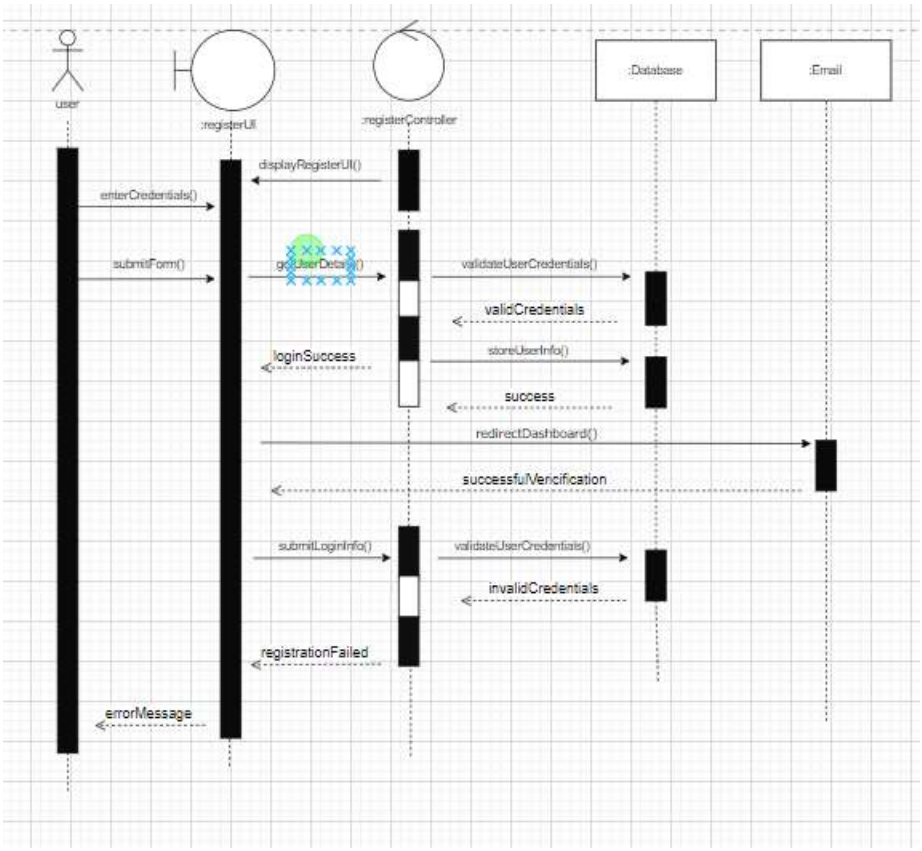


Figure 58 Sequence diagram of register.

Collaboration diagram

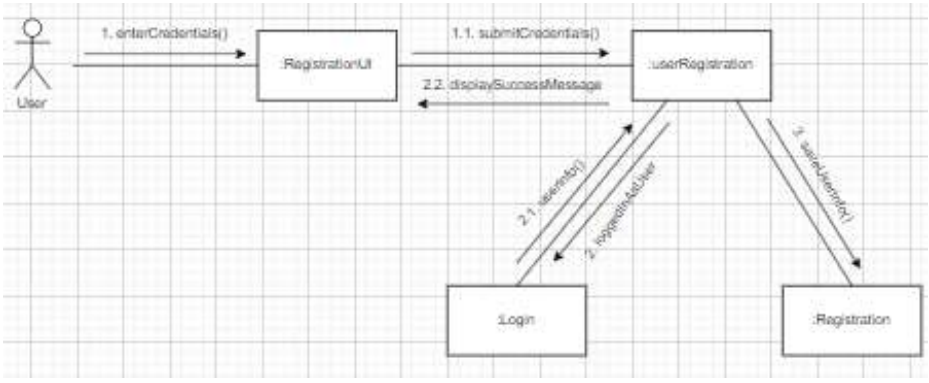


Figure 59 colaboration diagram of register

3.5.3.1. Post question & Answer feature UML diagrams

High level use case

Use case name	Ask Question
Actor(s)	User
Description	A user can edit or delete his/her post and also can answer to others posts.

Figure 60 High level use case of post question and answer feature UML diagrams.

Use case diagram

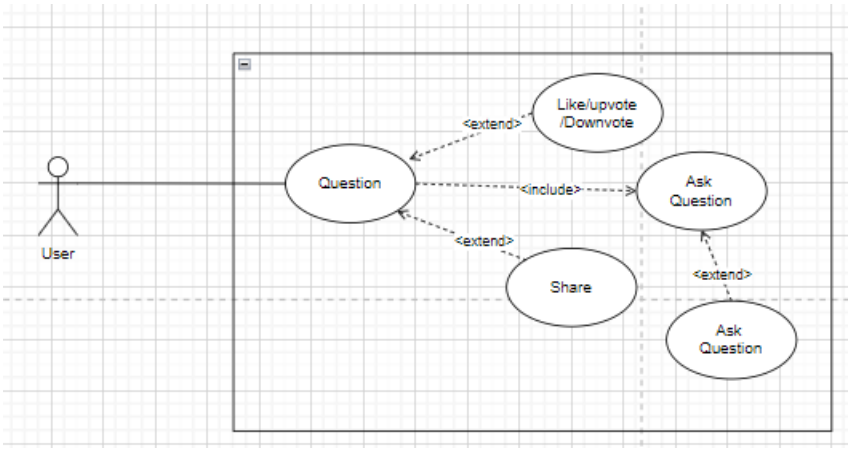


Figure 61 Use case diagram of edit, delete post and answer.

Sequence diagram

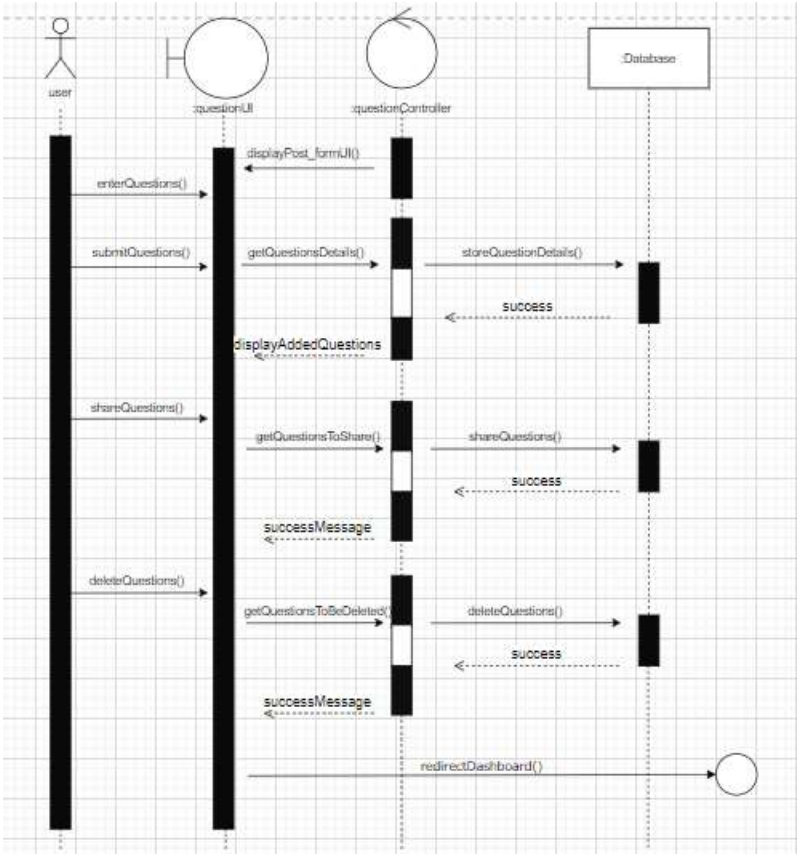


Figure 62 sequence diagram edit, delete, and answer post.

Collaboration diagram

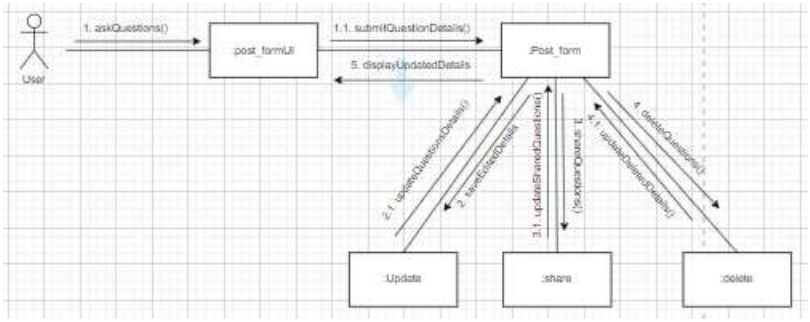


Figure 63 Collaboration diagram of edit, delete question and answer post.

3.5.3.1. Edit User Profile UML diagrams

High level use case

Use case name	Edit Profile
Actor(s)	User
Description	User can edit their personal information here. Except their e-mail address they can change their every information like username, password, contact number, etc.

Use case diagram

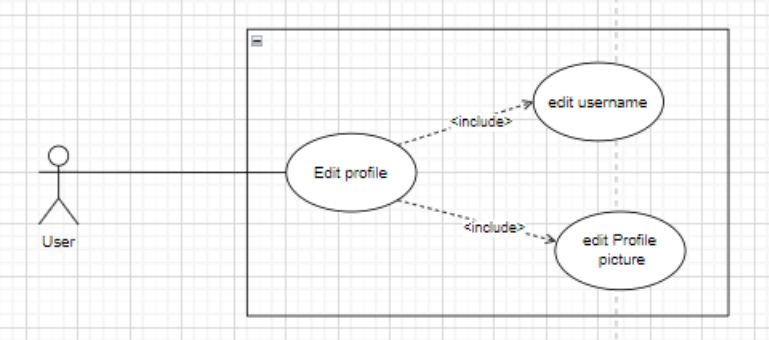


Figure 64 High level use case of edit profile.

Sequence diagram

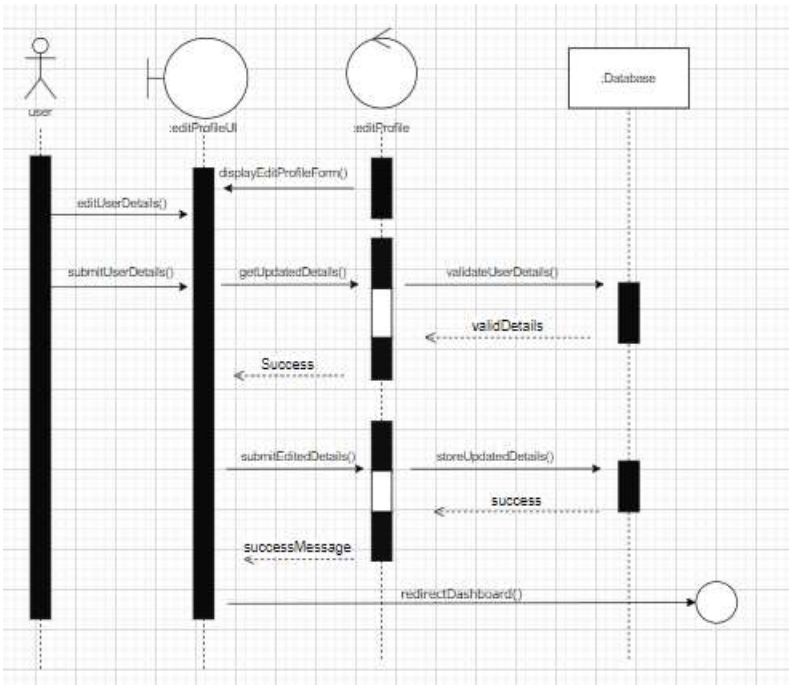


Figure 65 sequence diagram of edit profile.

Collaboration diagram

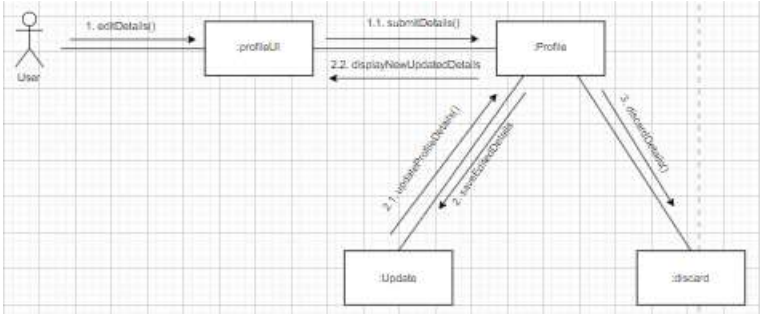


Figure 66 Collaboration diagram of edit profile.

3.7. Implementation (Screenshots of different features.)

3.7.1. App icon in mobile phone

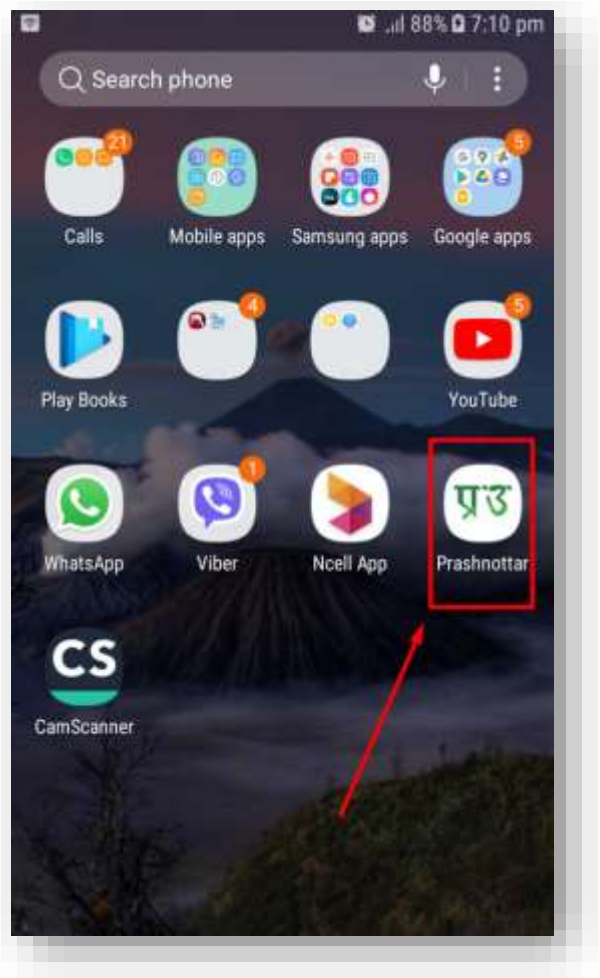


Figure 67 Prashnottar icon in mobile phone

3.7.2. Registering in the PT

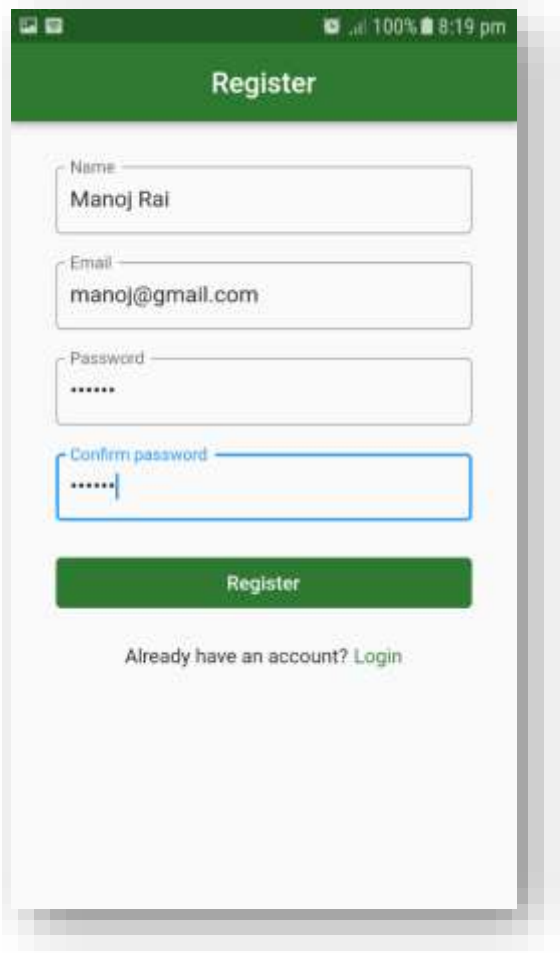


Figure 68 New register in the PT

3.7.3. Log in to PT

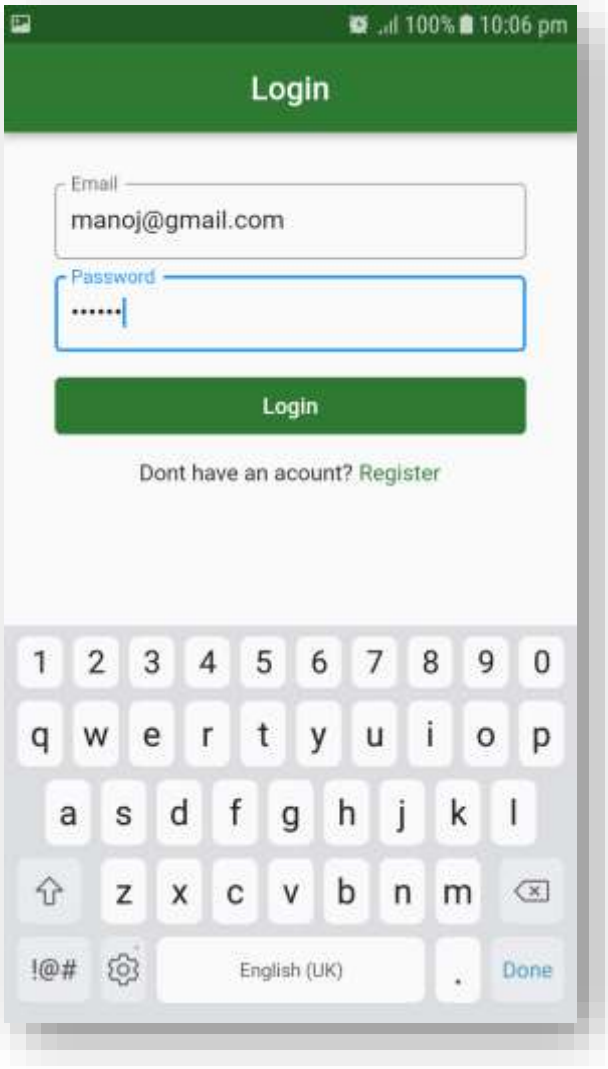


Figure 70 After succussful register, logging to the system.

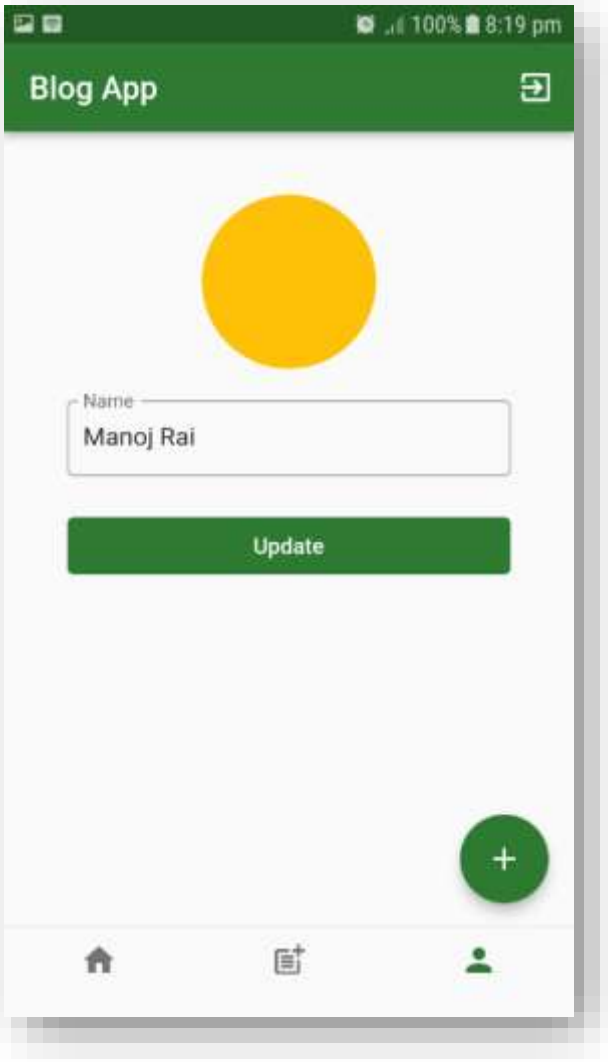


Figure 69 Successfully logged ot the prashnottar app

3.7.4. Choose Class

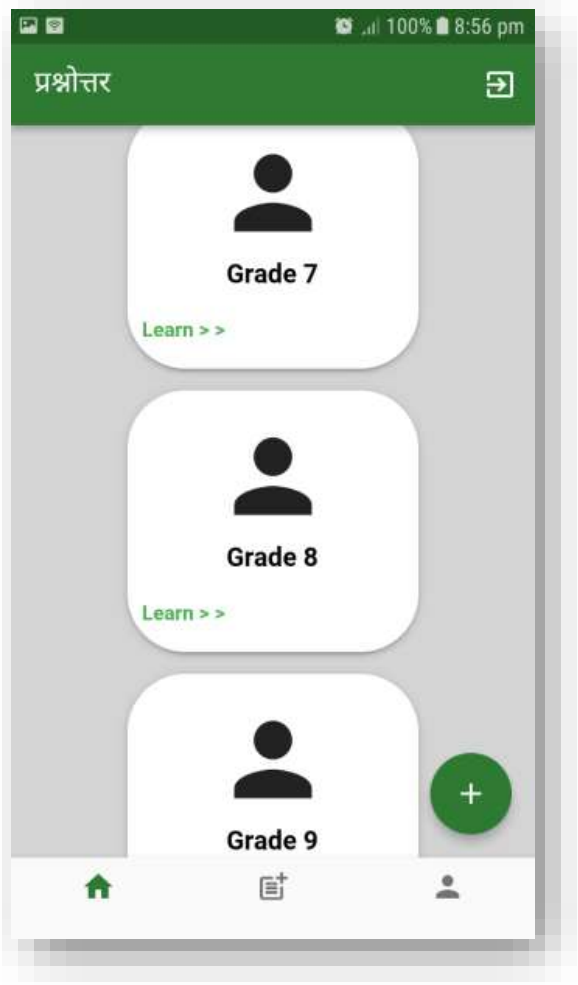


Figure 71 Choose class of the PT.

3.7.5. Choose Subject

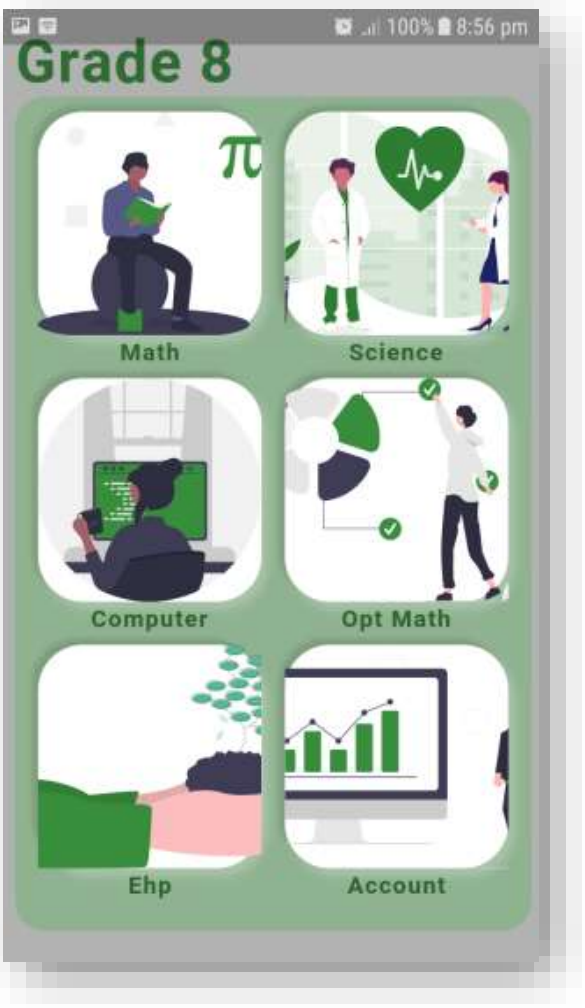


Figure 72 Choose subject of grade 8

3.7.6. Choose Chapter of the Subject

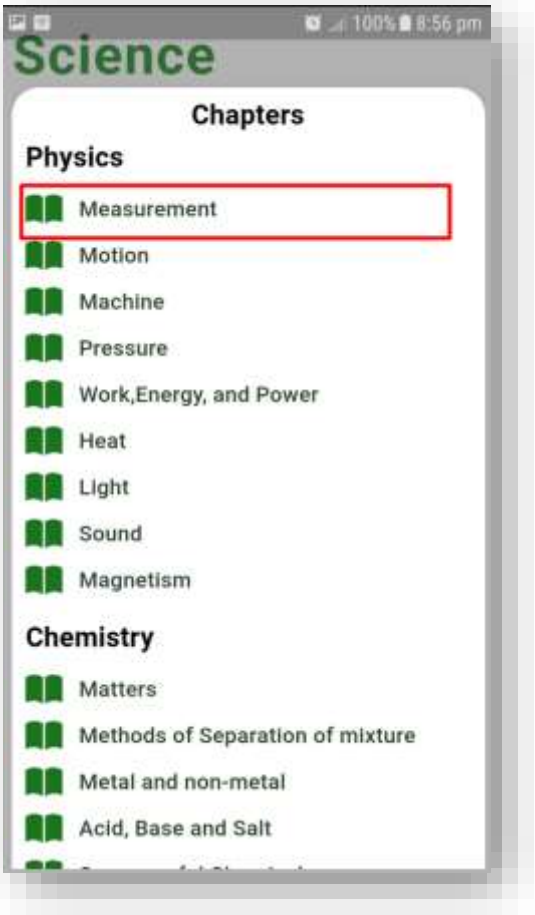


Figure 73 Choose chapter from Science Subject.

3.7.7. Make question post with Sharing picture

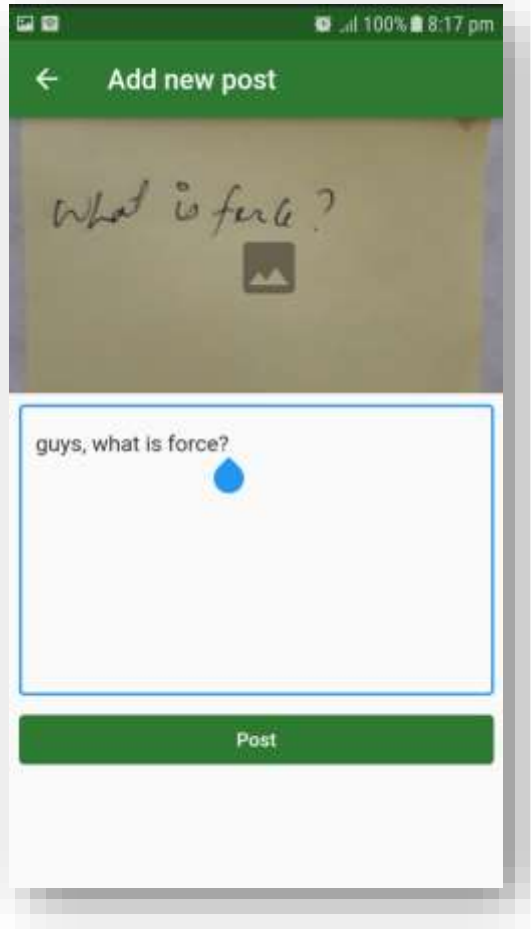


Figure 75 Ask a question in a post sharing a picture.

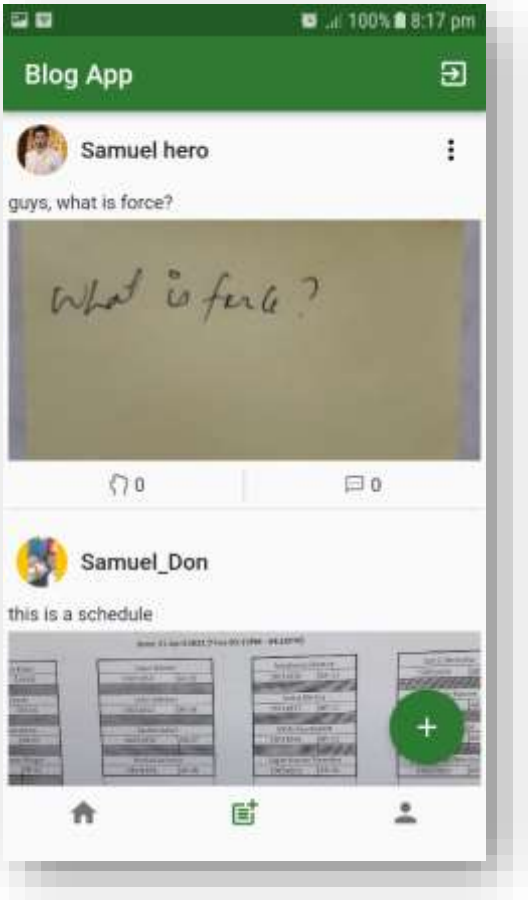


Figure 74 Question post successfull.

3.7.8. Edit post

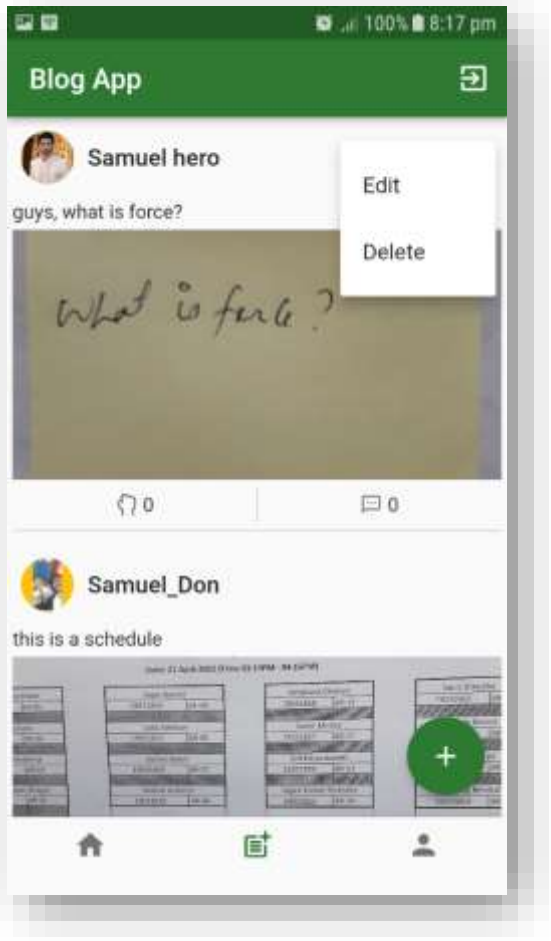


Figure 77 Editing a post

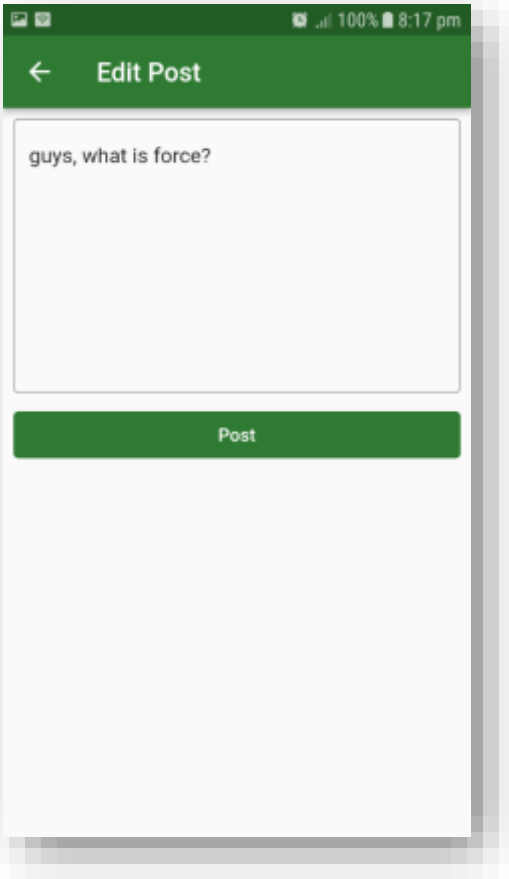


Figure 76 Editing the question post.

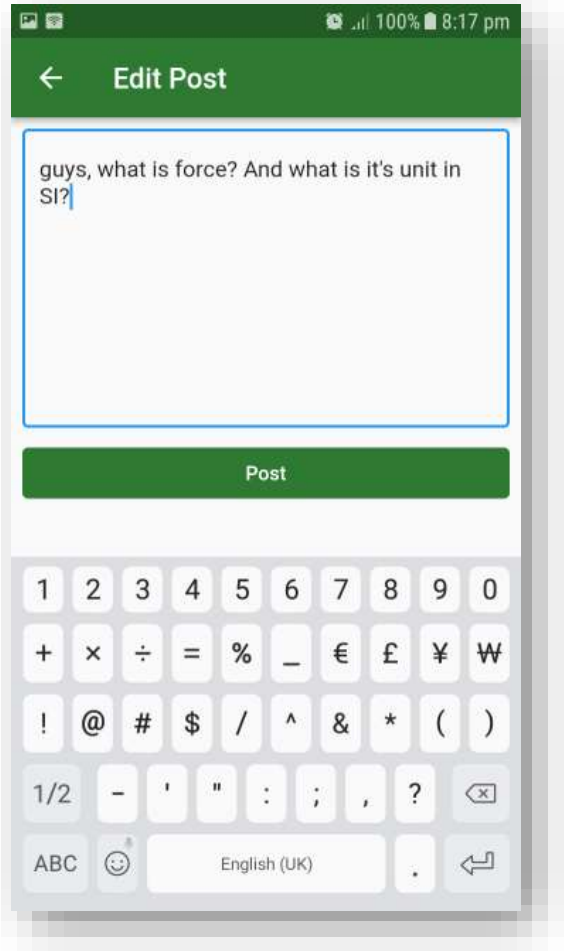


Figure 79 Editing the post

Figure 78 Successfully post edited.

3.7.10. Delete Post

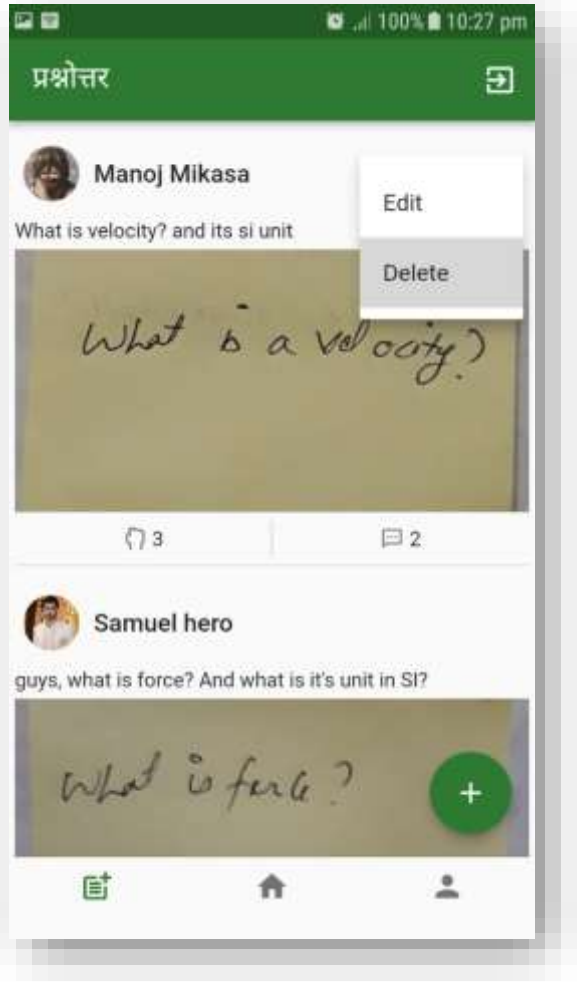


Figure 81 Deleting a post

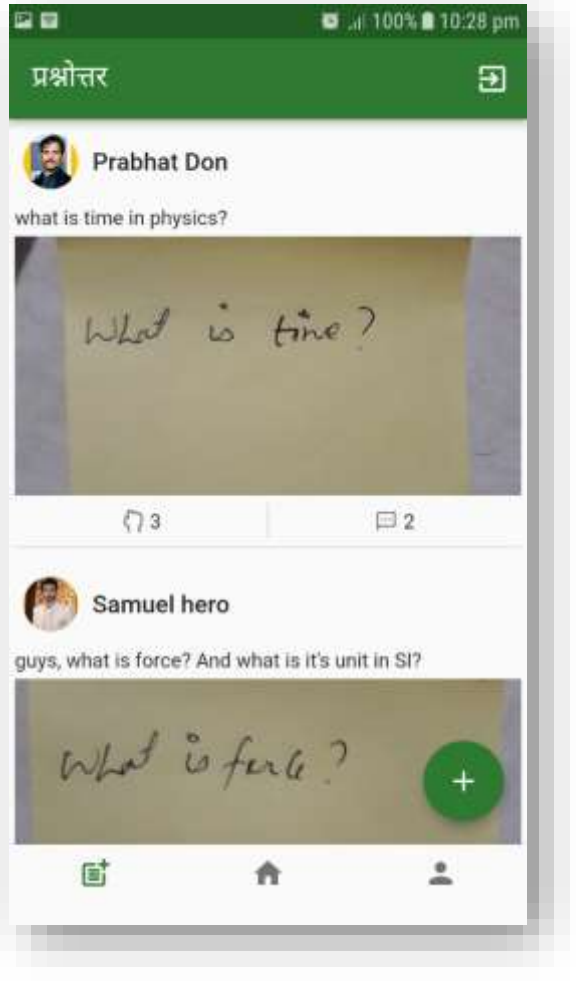


Figure 80 Post deleted succesfully.

3.7.11. Answer the post

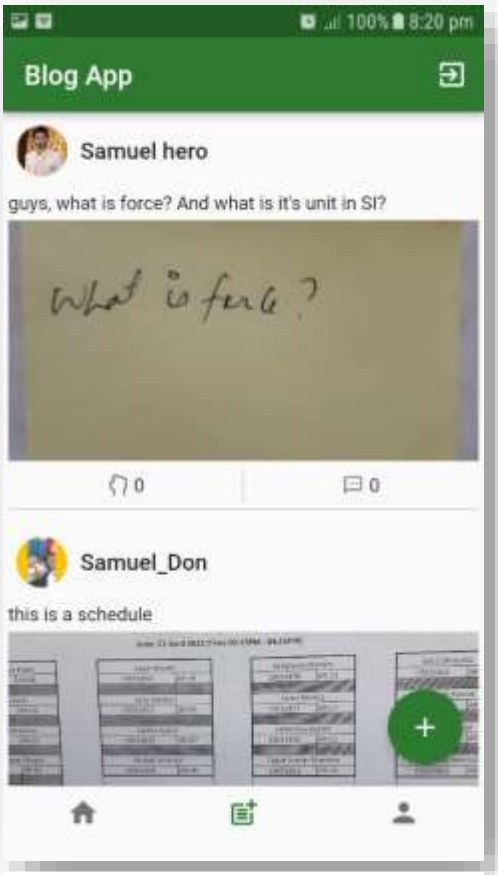


Figure 84 The post doesn't has any of the answer

Figure 83 Writting an answer to the post.

Figure 82 Answer given successfully to the post.

3.7.12. Edit answer

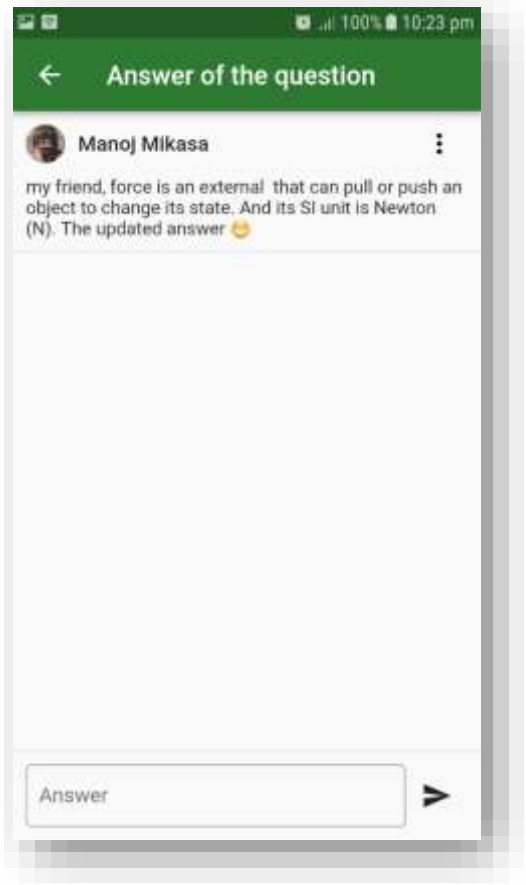


Figure 85 Answer updated successfully.

3.7.13. Delete Answer

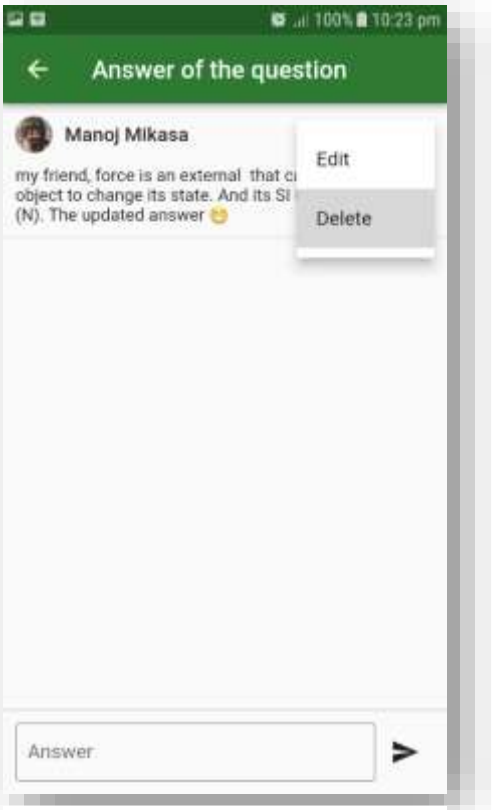


Figure 88 Deleting an answer.

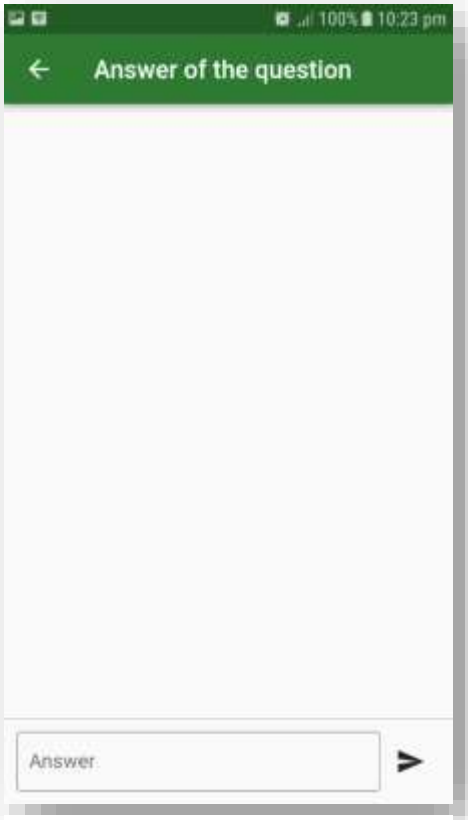


Figure 87 Answer deleted successfully.

Figure 86 No answer in the post..

3.7.14. Like the post

Figure 89 Post is liked.

3.7.15. Dislike the post

Figure 90 Post is disliked.

3.7.16. Edit Update profile

Figure 93 User Manoj Rai to be updated.

Figure 92 Selecting image from the gallery.

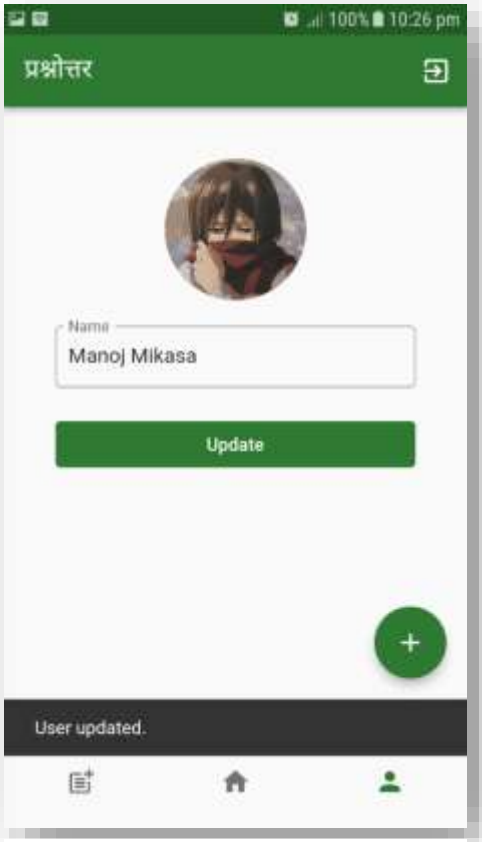


Figure 91 User profile and name updated successfully.

Chapter 4: Testing And Analysis

4.1. Test Plan

We will be examining our sofware with the help of two type of test, which are: unit testing and system testing . The aim of testing is to find bugs and explain their significance . With the help of testing we will decrease the risk by proactively identifying and assisting with the resolution of issues.

4.1.1. Test plan of Unit Testing

In the Unit testing, we will go through the testing of API's using post man. If we want to check the unit testing of flutter alone, then it is of no use, since the required datas will be backed up by backend. So those features as a whole will be tested only in system testing.

4.1.2. Test plan of System Testing

In System testing, we will use and test all the necessary features. Altogether we will be testing 15 features.

4.2. Unit Testing

4.2.1. Test case 1: Unit test of registering validation API.

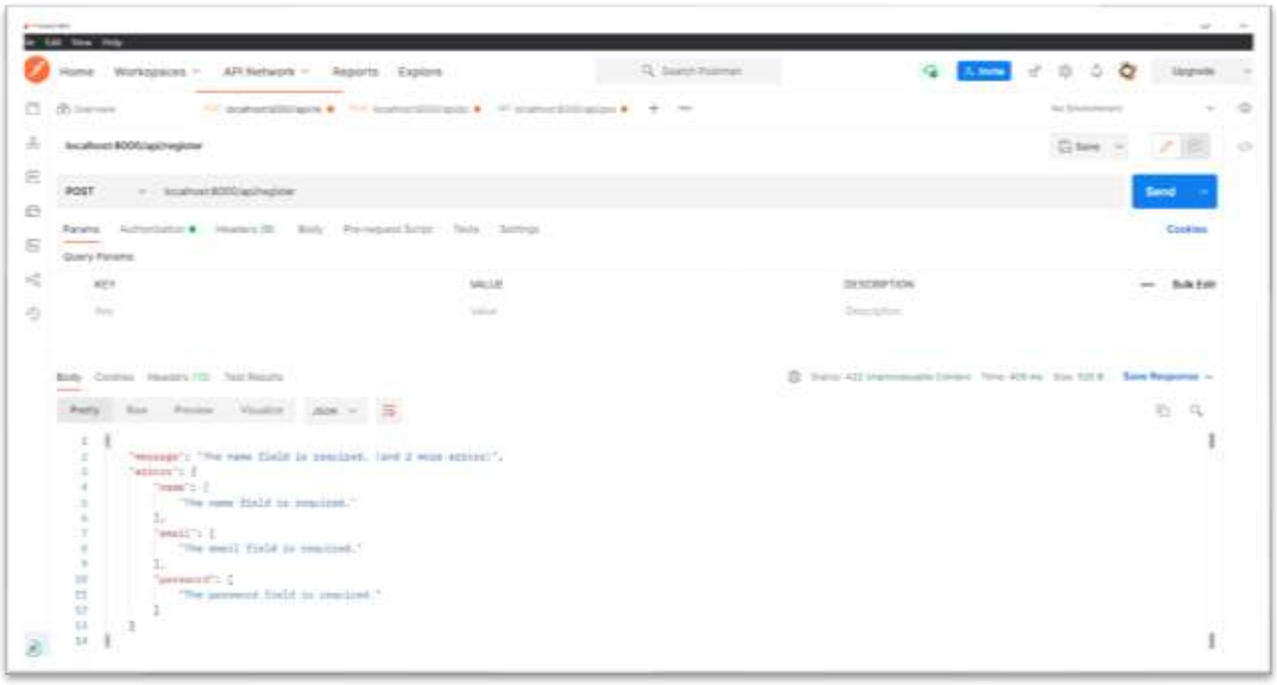


Figure 94 Registering validation

Objective	To validate user credentials before registering.
Action	In the Laravel file; name, email, and password was made compulsory, which helped when API code was written in postman.
Expected Result	The user must not be registered.
Actual Result	The error message came.
Conclusion	The test was Successful.

Table 8 Result of unit test case 1

4.2.2. Test case 2: API unit test of registering new user.

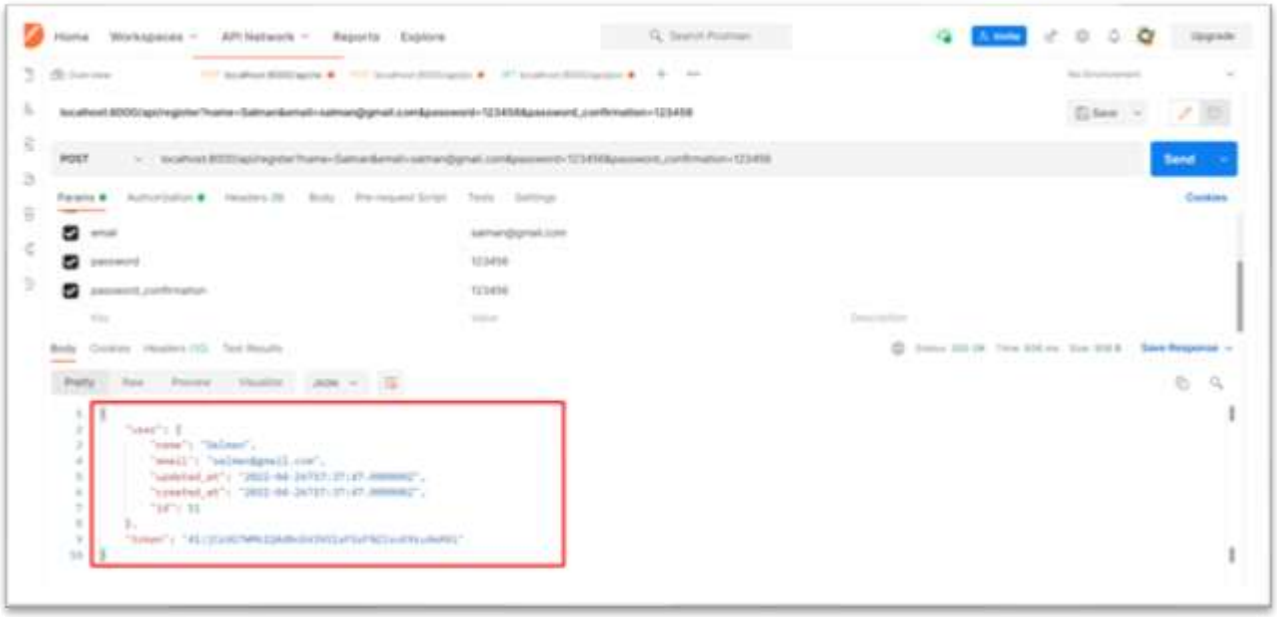


Figure 95 New user, Salman added to the system

Objective	To register new user in the database.
Action	Registering new user API code was used in the postman.
Expected Result	A new user must be made.
Actual Result	A new user named Salman with salman@gmail.com was made.
Conclusion	The test was successful.

Table 9 Result of unit test case 2.

4.2.3. Test case 3: Unit test of registering same user validation.

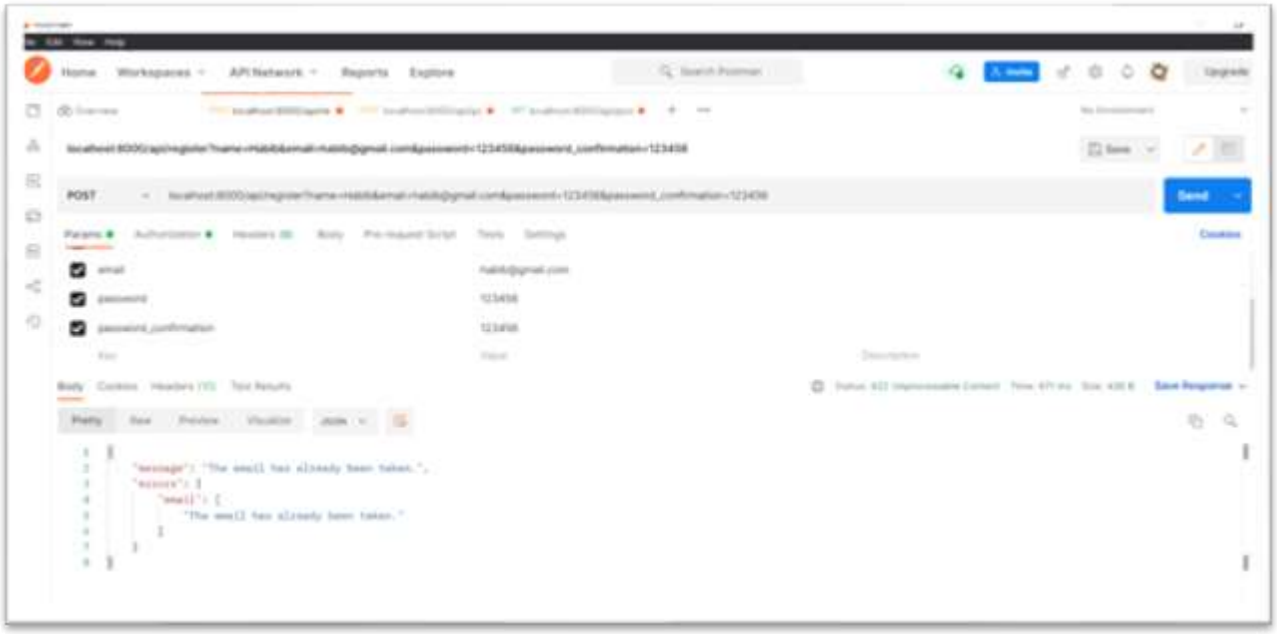


Figure 96 trying to register a registered user.

Objective	To check if already registered user are again registered or not.
Action	To put registered user credentials api in the postman.
Expected Result	Error message must appear.
Actual Result	No registered user was allowed to register again.
Conclusion	The test was successful.

Table 10 Result of unit test case 3.

4.2.4. Test case 4: Log in API unit tests.

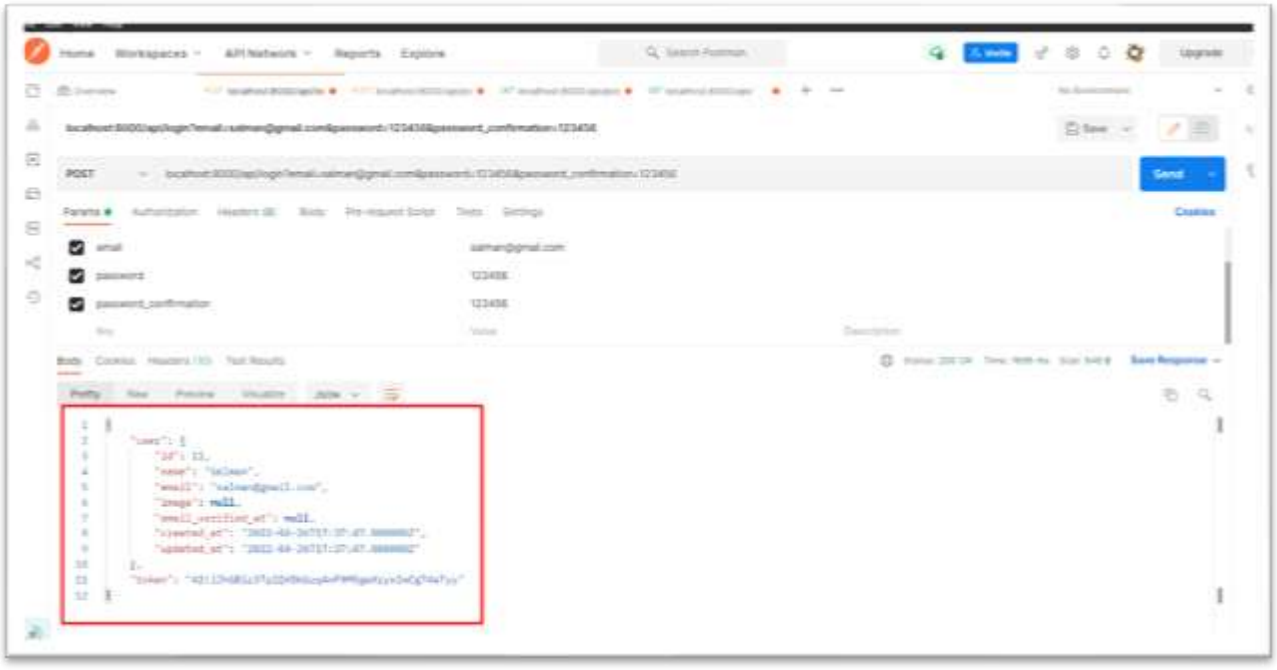


Table 11 Login api test

Objective	To log in to the database system.
Action	Log in api was used in the postman.
Expected Result	Registered user must be able to log in to the system.
Actual Result	Log in to the system was successful.
Conclusion	The test was succesful.

Table 12 Result of unit test case 4

4.2.5. Test case 5: Logout feature API unit tests.

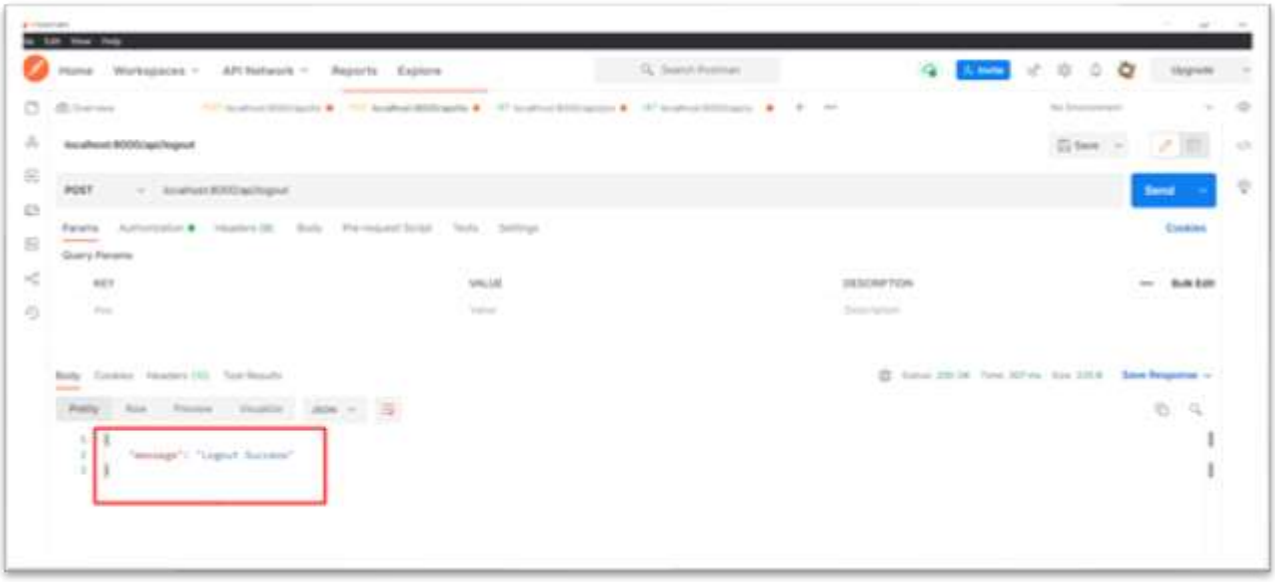


Figure 97 Logout successfully form the system

Objective	To logout user from the system.
Action	User logout api code in the postman.
Expected Result	User must be able to logout from the system.
Actual Result	User was able to be logged out.
Conclusion	The test was successful.

Table 13 Result of unit test case 5

4.2.6. Test case 6: Get user data

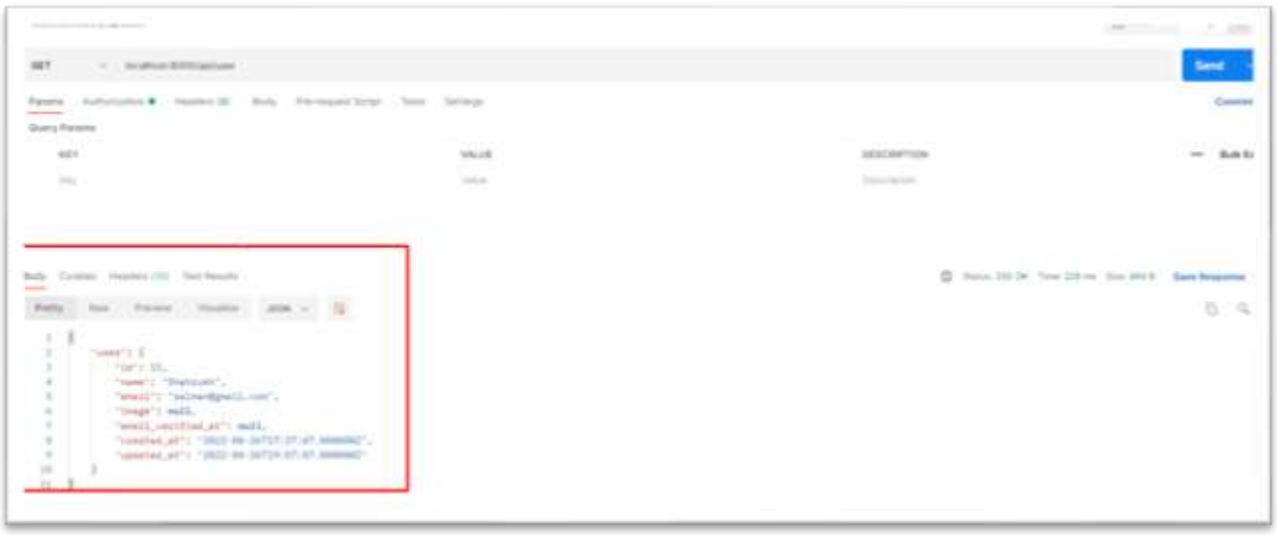


Table 14 Get user data using user API

Objective	To know detail about a user.
Action	User API code was used in the postman.
Expected Result	Detail of user must be returned.
Actual Result	Detail of user was printed.
Conclusion	The test was successful.

Table 15 Result of unit test case 6.

4.2.7. Test case 7: Update user data

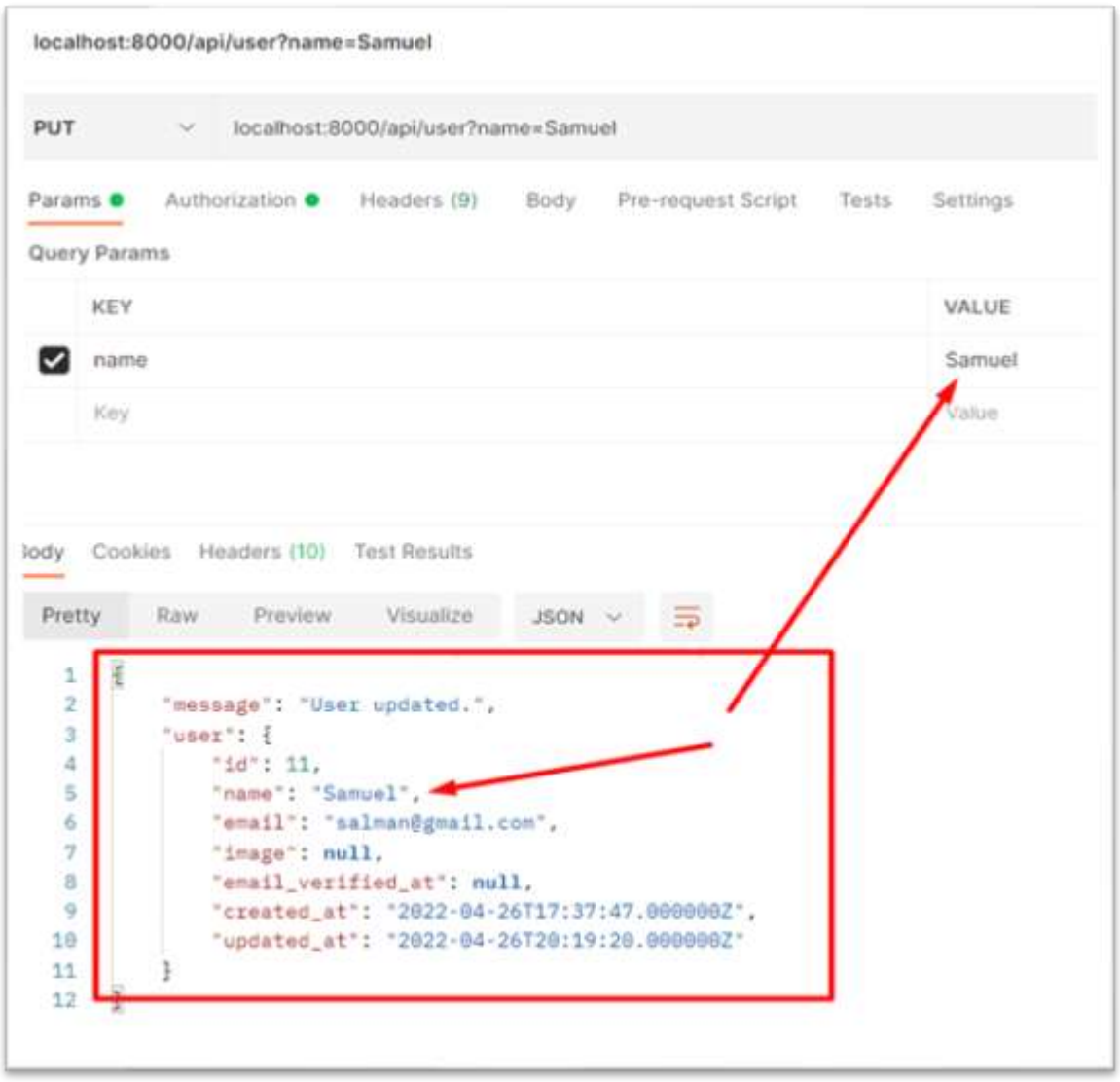


Table 16 User name was updated.

Objective	To update user credentials.
Action	User update API was used in the postman.
Expected Result	User name must be changed.
Actual Result	The name was changed.
Conclusion	The test was successful.

Table 17 Result of unit test case 7.

4.2.8. Test case 8: Getting to view question post detail

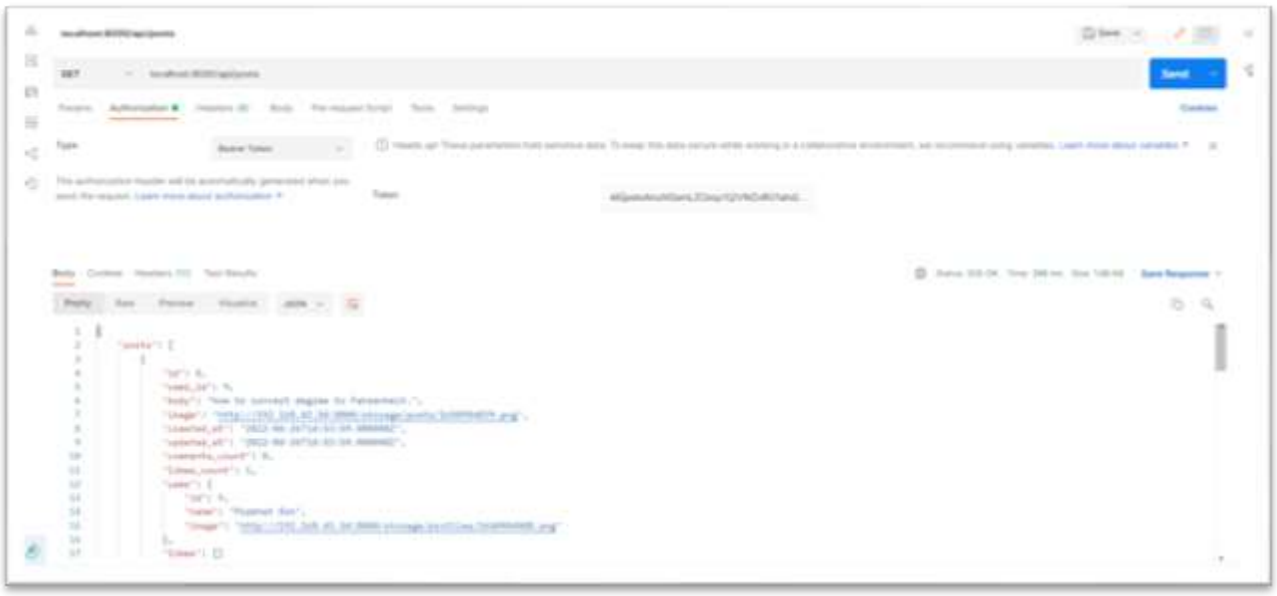


Figure 98 Question answer post

Objective	To view detail of a post.
Action	Use post get API in the postman.
Expected Result	A detail of post must be returned.
Actual Result	The detail of a post was printed.
Conclusion	The test was successful.

Table 18 Result of unit test case 8.

4.2.9. Test case 9: Ask a question or post a question in a post.

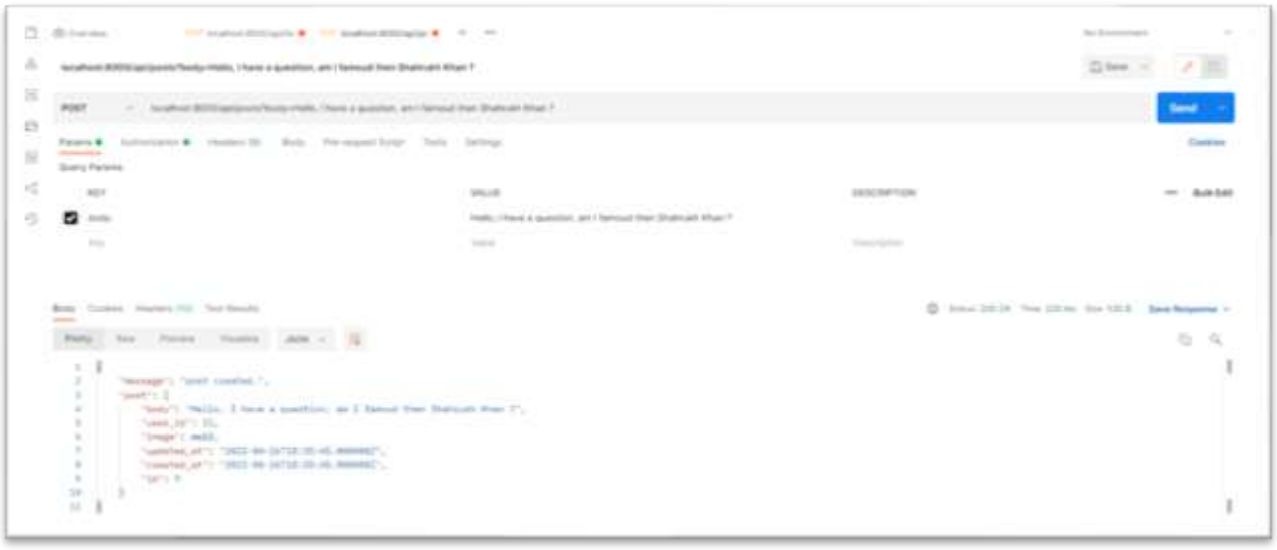


Figure 99 Making a question post.

Objective	To ask a question in a post by the user.
Action	API of a Body of a post was used in the postman.
Expected Result	User must be able to post a question.
Actual Result	User was able to include body part of the post where qeustion was asked.
Conclusion	The test was successful.

Table 19 Result of unit test case 9

4.2.10. Test case 10: Unit test of answering the posted question.

Figure 100 Giving answers on the post id 9

Objective	To view a answer of a post.
Action	View answer API code was used in the postman.
Expected Result	User must be able to view answers written by other users.
Actual Result	User was able to read answer of the post.
Conclusion	The test was successful.

Table 20 Result of unit test case 10.

4.2.11. Test Case 11: Getting to view question and answer posts in api unit tests.

Figure 101 writting an answer on the post.

Objective	To let user write an anwer to questions asked in a post.
Action	API of writing answer was applied in the postman.
Expected Result	User must be able to anwer to a certain question in the post.
Actual Result	User was able to answer to a question of the post.
Conclusion	The test was successful.

Table 21 Result of unit test case 11

4.2.12. Test case 12: Getting to know the detail about answer of the post.

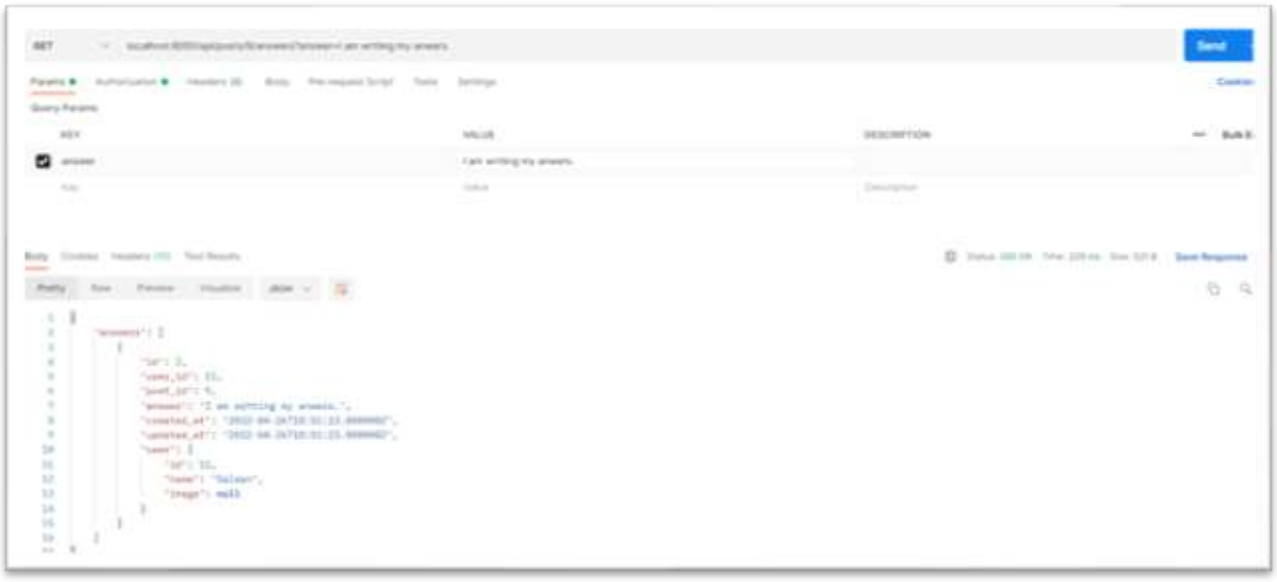


Figure 102 API test to view answer written.

Objective	To view detail of the anwer of a post.
Action	Use API of view anwer of a particular post in a postman.
Expected Result	User must be able to view detail of answer.
Actual Result	User was able to view the detail of answer, like who created and to which question does it belong.
Conclusion	The test was successful.

Table 22 Result of unit test case 12

4.2.13. Test case 13: Getting to know the detail about anwer of the post.

Figure 103 Liking on post 14.

Objective	To let users like a post.
Action	Insert like API in the postman.
Expected Result	User must be able like or dislike a post.
Actual Result	User was able to like and dislike a post.
Conclusion	The test was successful.

Table 23 Result of unit test case 13

4.3. System Testing

4.3.1. Starting all the application

System testing is all about testing the whole live project after integrating API's with flutter and store data in the database. Especially there were three different phases done to complete system testing:

i) Start mysql database using Xampp.

ii) Visual Studio Code to start and it's terminal to host the Laravel API

iii) Build the whole app in my android phone using VS code.

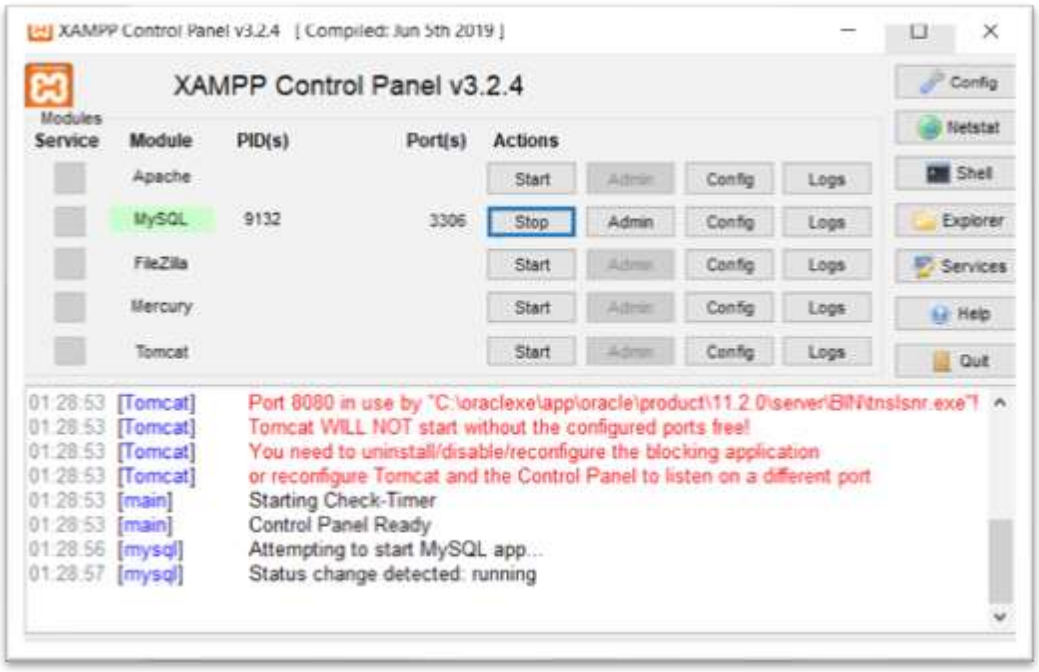


Figure 104 Starting mysql database using xampp.

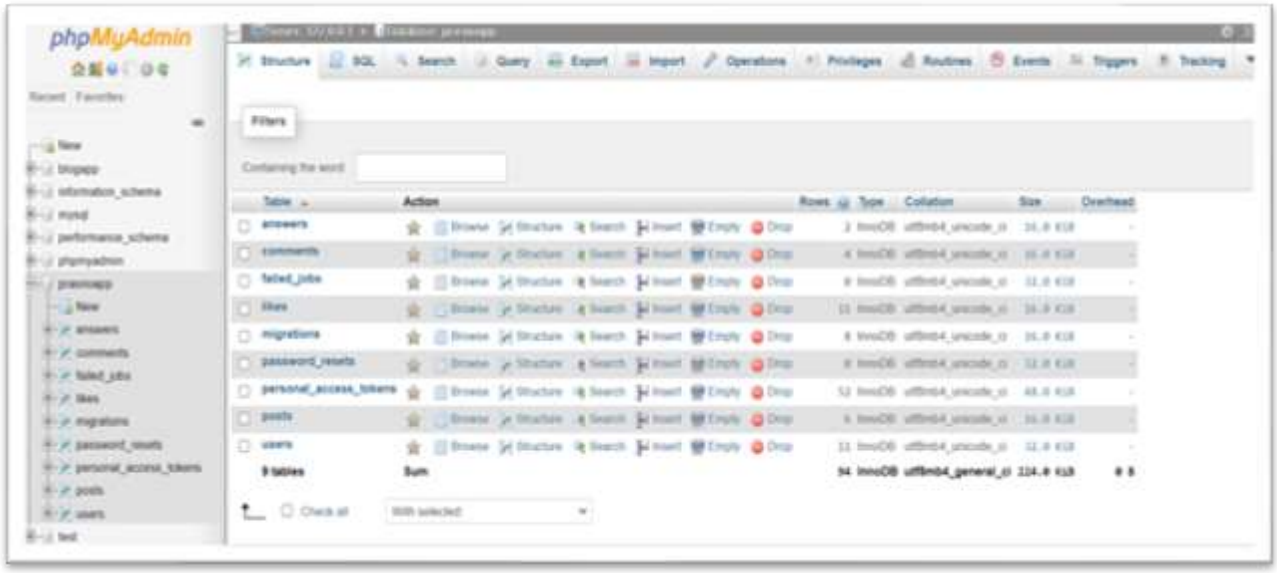


Figure 105 Database prasnoapp of Prashnottar

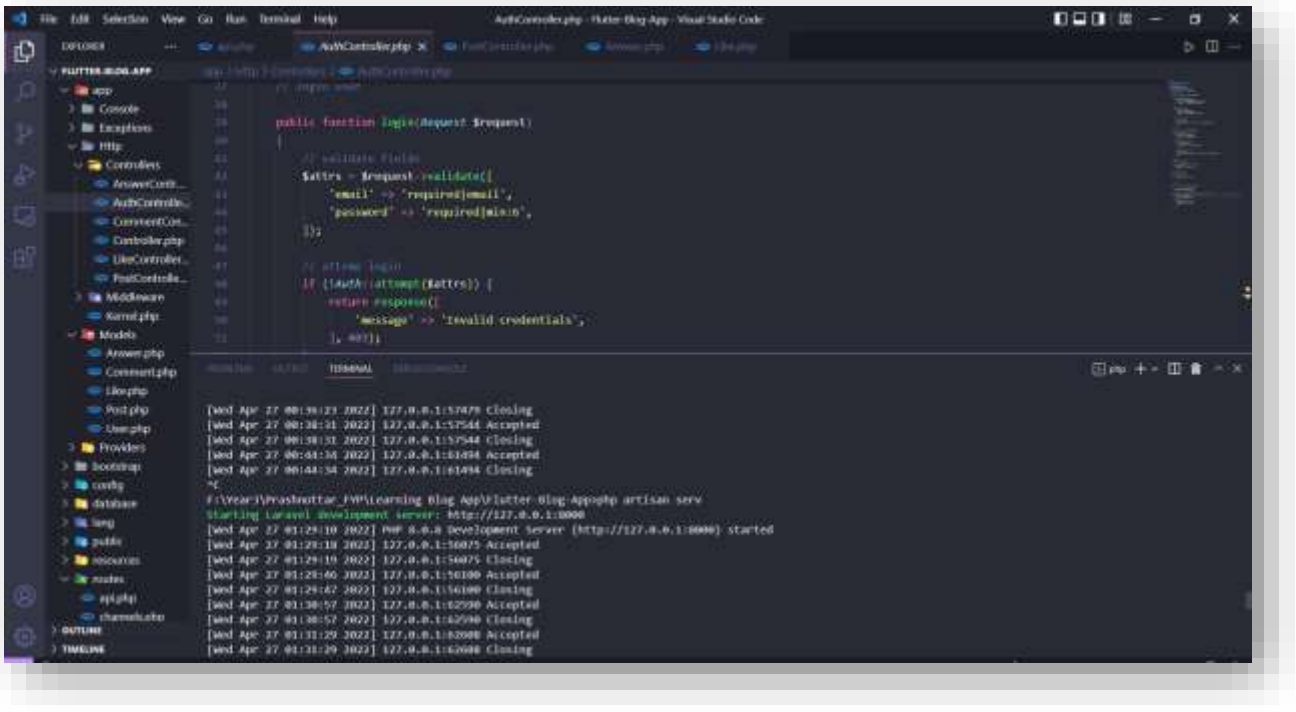


Figure 106 Starting laravel project in VScode terminal.

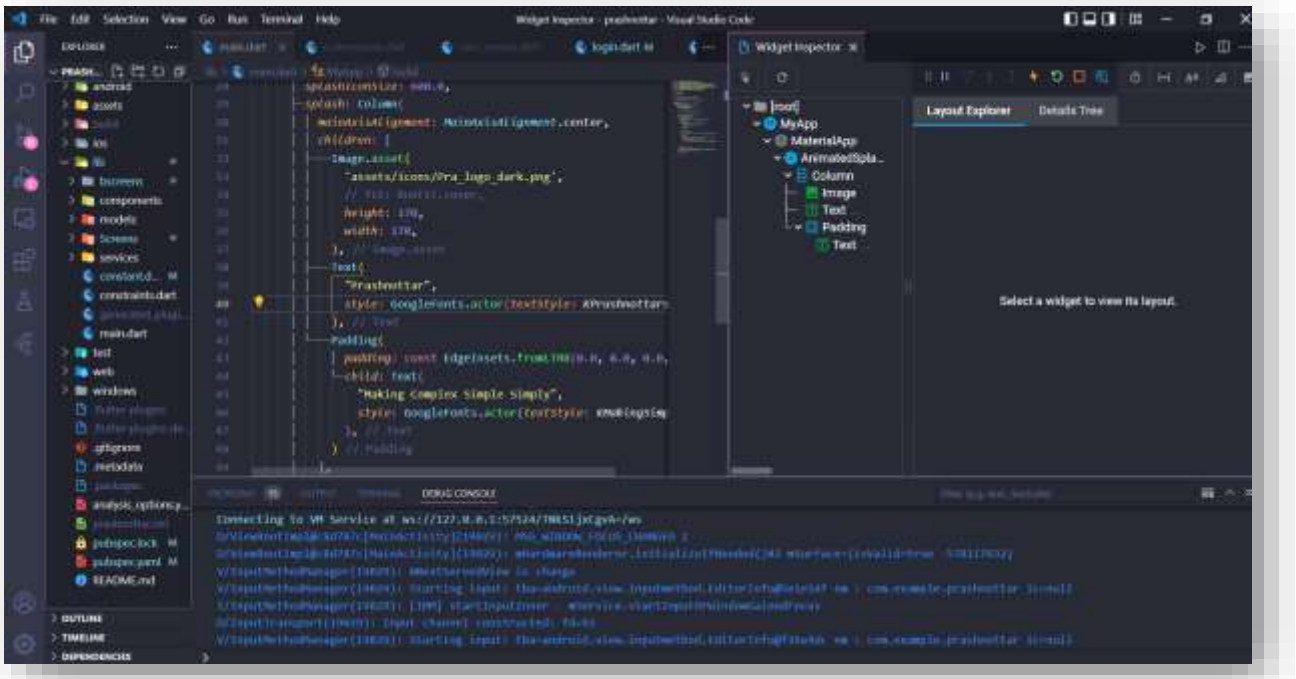


Figure 107 Starting android app in android phone using VS Code.

4.3.2. Test caes 1: App icon in mobile phone

Figure 108 To test if app icon is shown in the android phone.

Objective	To see if app logo is shown in android phone.
Action	Run the app in phone using USB cable and run it using VS Code.
Expected Result	The logo must be shown.
Actual Result	The logo was shown.
Conclusion	The test was successful.

Figure 109 Result of system test case 1.

4.3.3. Test case 2: Registering in the PT

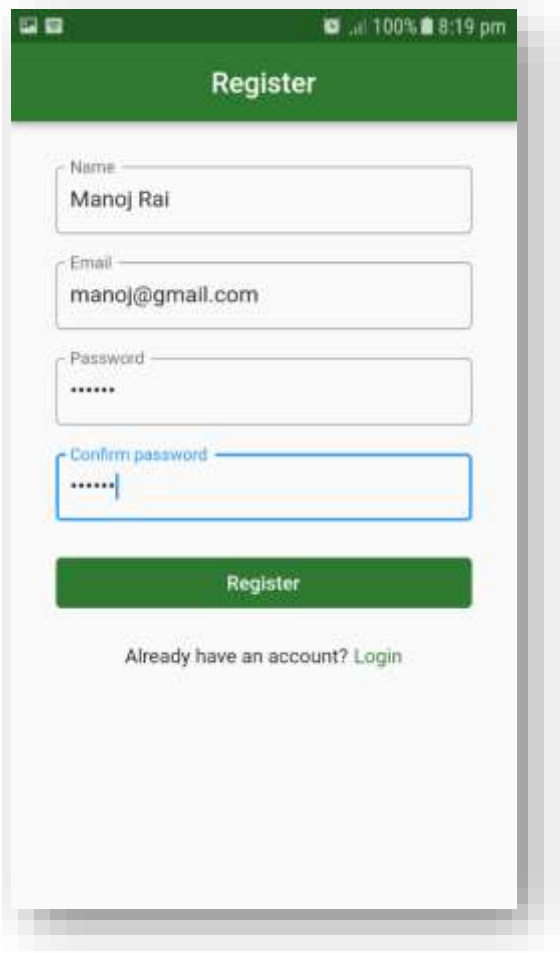


Figure 110 New registration in PT.

Objective	To see if a new user is registered in the app or not.
Action	Run the program and try to fill in the form with non registered email.
Expected Result	The registration must be done.
Actual Result	The registration was successful.
Conclusion	The test was successful.

Figure 111 Result of system test case 2.

4.3.4. Test case 3: Log in to PT

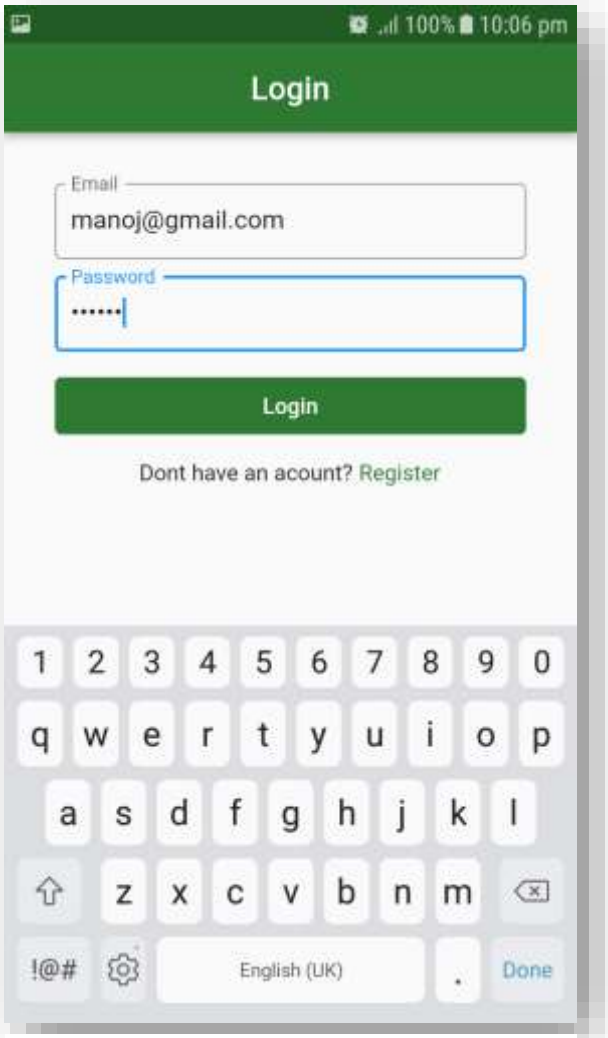


Figure 113 Log in to the PT with previous registered account.

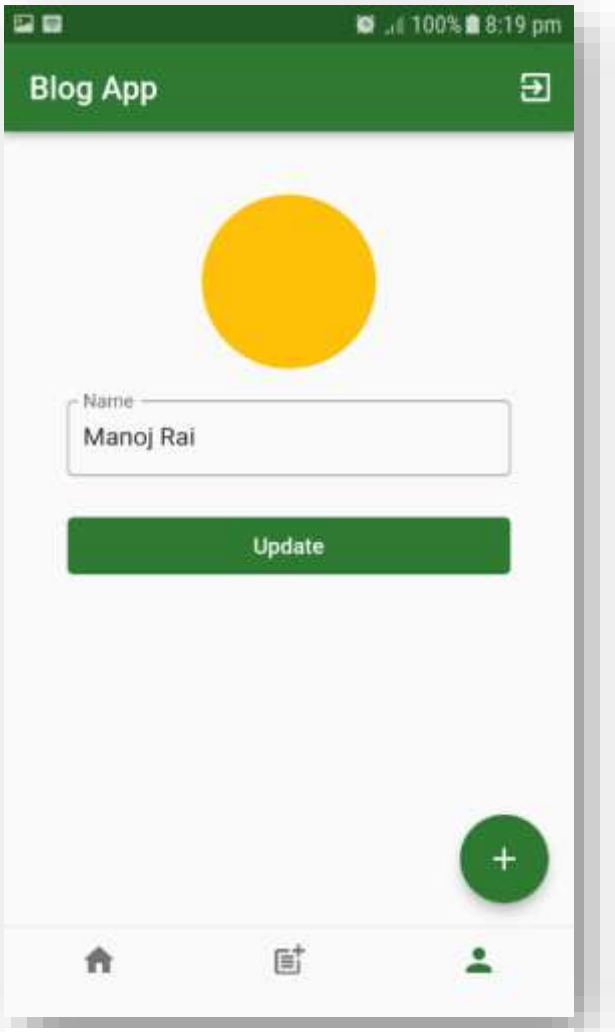


Figure 112 Successfully logged in

Objective	To log in with registered account.
Action	Fill the form with valid email and password.
Expected Result	Afer log in, user must be taken to the homepage.
Actual Result	User was successfully logged in.
Conclusion	The test was successful.

Figure 114 Result of system test case 3.

4.3.5. Test case 4: Choose Class

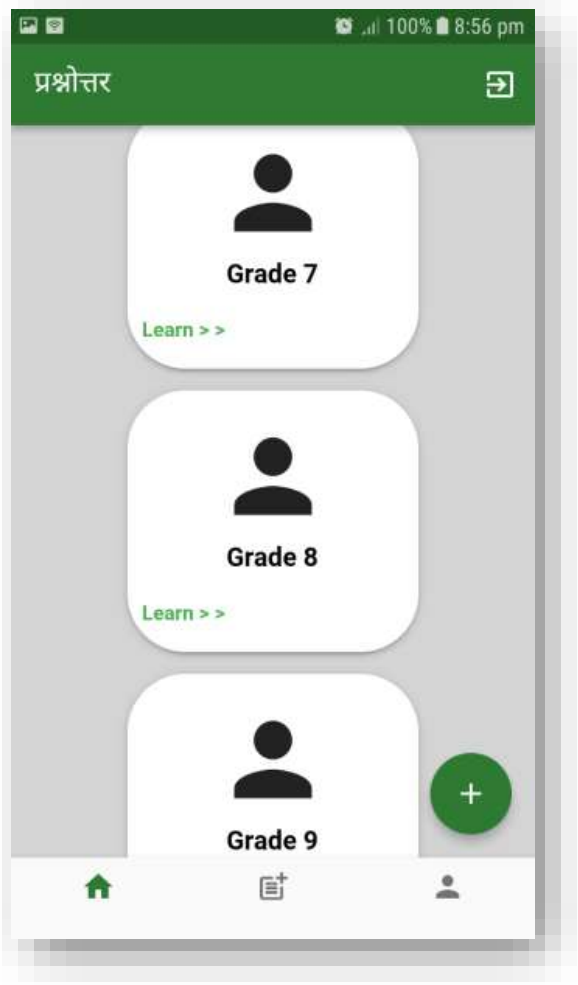


Figure 115 Choose class of the PT.

Objective	To allow user to choose between class 8-10.
Action	To click on a respective class tile.
Expected Result	User must be taken to subject screen.
Actual Result	User was taken to the subject screen.
Conclusion	The test was successful.

Figure 116 System testing to class selection.

4.3.6. Test case 5: Choose Subject

Figure 117 Choose subject of grade 8

Objective	To let user select a particular subject.
Action	To touch a tile of a particular subject.
Expected Result	User must be taken to subject's chapter screen.
Actual Result	User was successfully taken to chapter screen.
Conclusion	The test was successful.

Figure 118 Select a partifular subject.

4.3.7. Test case 6: Choose Chapter of the Subject

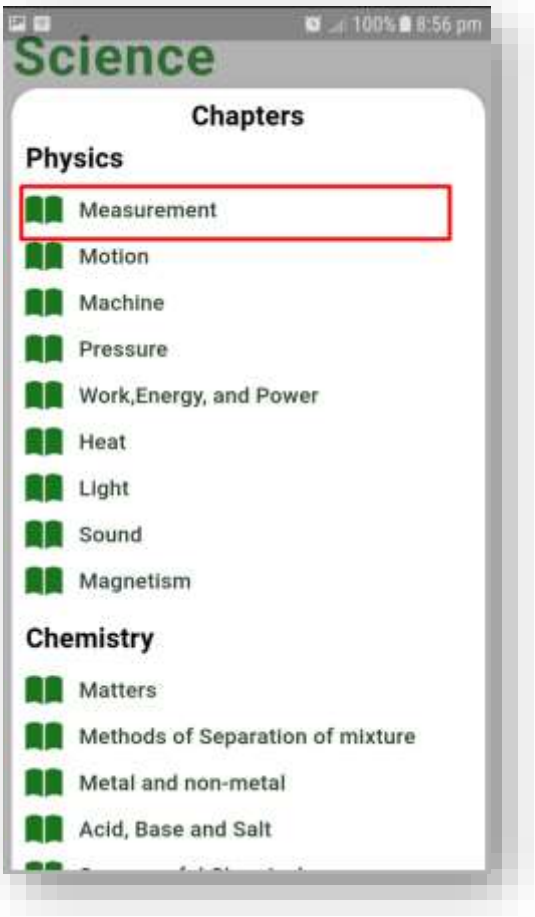


Figure 119 Choose chapter from Science Subject.

Objective	To let user select a particular subject among other subjects.
Action	Touch a name of the chapter.
Expected Result	Take it to the blog section where different questions related to the chapter would be asked.
Actual Result	It took to the blog section.
Conclusion	The test was successful.

Figure 120 System testing of selecting a chapter.

4.3.8. Test case 7: Make question post with Sharing picture

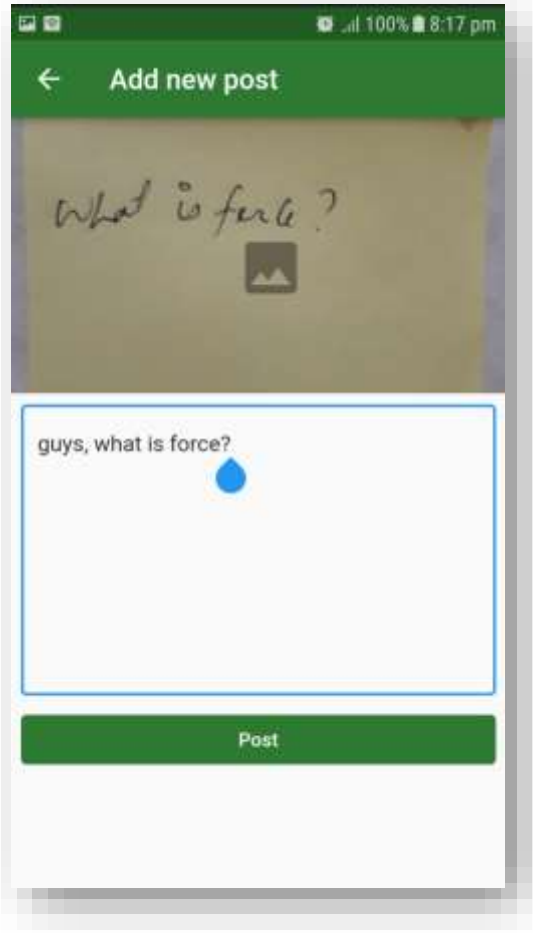


Figure 122 Ask a question in a post sharing a picture.

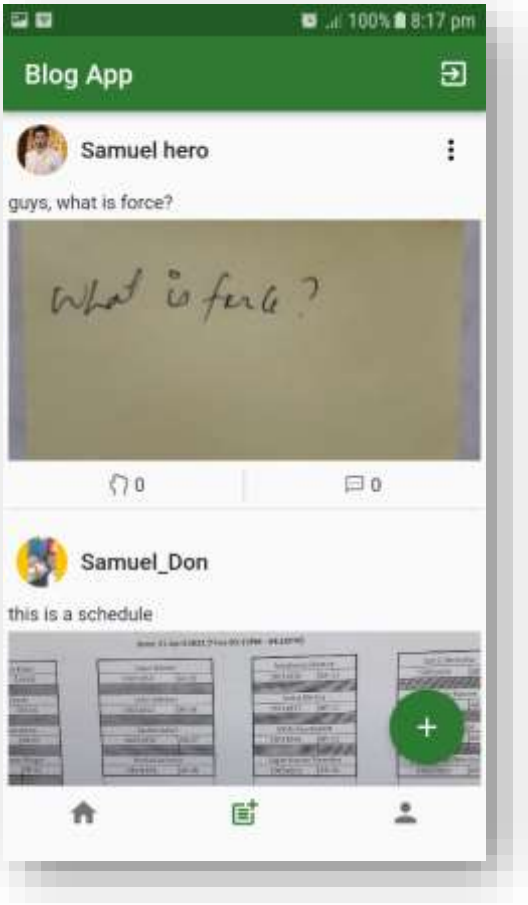


Figure 121 Question post successfull.

Objective	To be able to ask a question asking a question.
Action	Click that floating plus button and just ask a question with addition of a picture.
Expected Result	User must be able to ask a question in a blog section.
Actual Result	User was able to make a question post in blog section.
Conclusion	The test was successful.

Figure 123 System test of asking a question.

4.3.9. Test case 8: Edit post

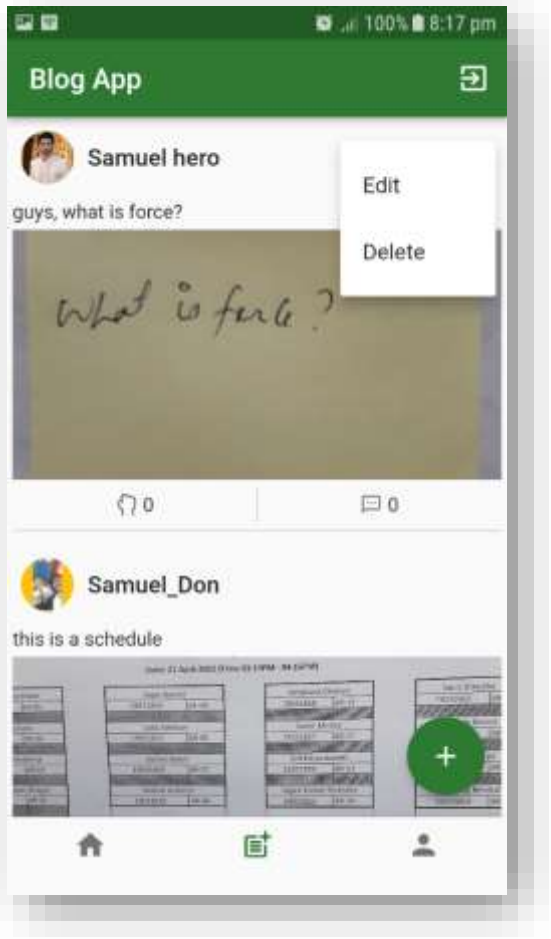


Figure 125 Editing a post

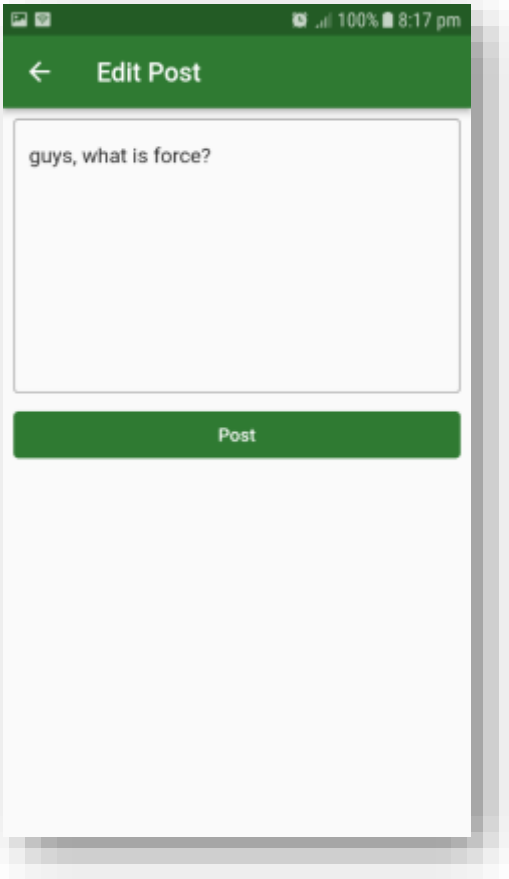


Figure 124 Editing the question post.

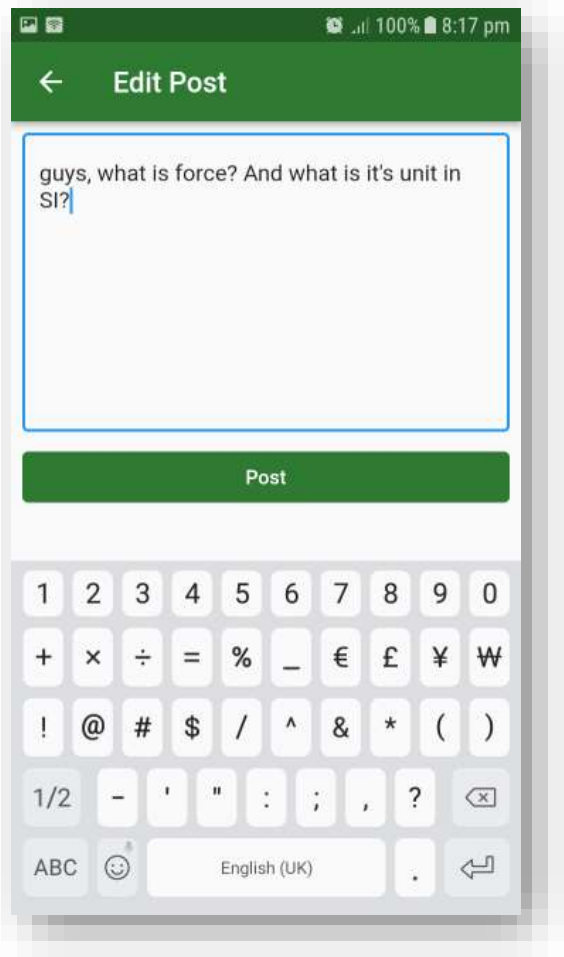


Figure 126 Successfully post edited.

Figure 127 Editing the post

Objective	Just let user to edit their question post.
Action	Click on those three dots, and click edit and edit the post.
Expected Result	To be able to edit and save post.
Actual Result	Blog was able to be edited.
Conclusion	The test was successful.

Figure 128 System test of edit post.

4.3.10. Test case 9: Delete Post

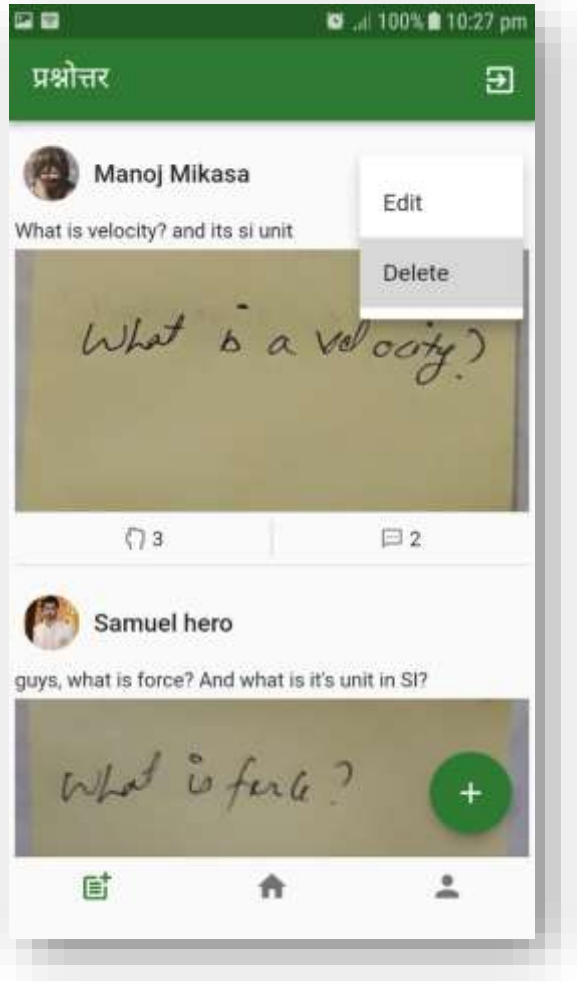


Figure 130 Deleting a post

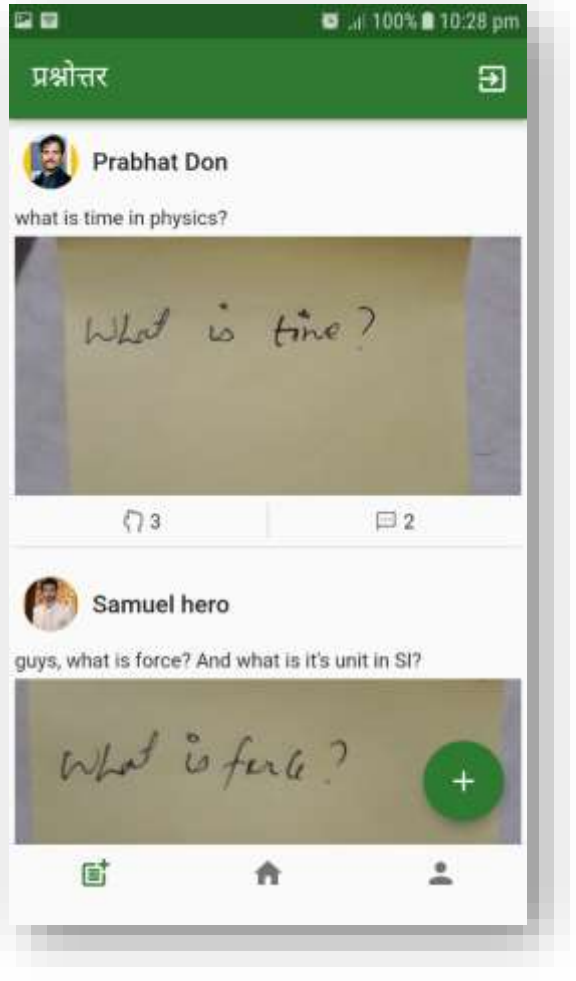


Figure 129 Post deleted succesfully.

Objective	Let user delete their post.
Action	Click on three dots and click delete.
Expected Result	The post must be deleted along with its likes and answers.
Actual Result	The question post was deleted successfully.
Conclusion	The test was successful.

Figure 131 System test of delete question post.

4.3.11. Test case 10: Answer the post

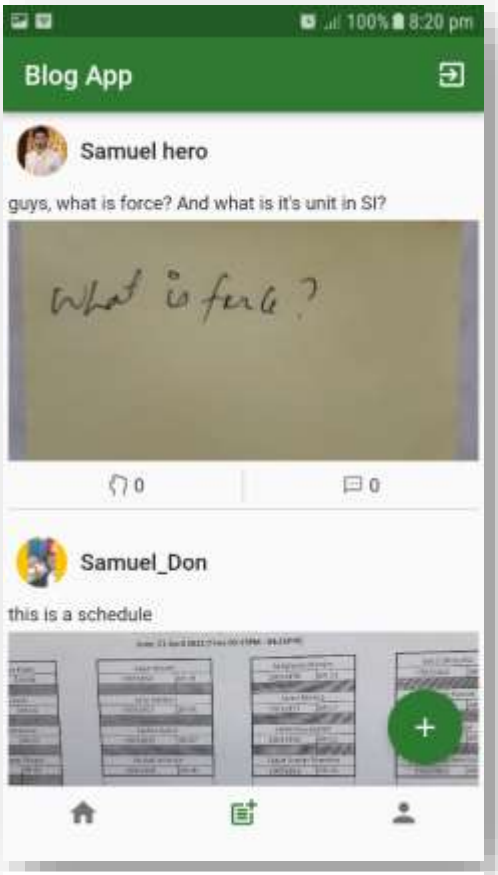


Figure 134 The post doesn't has any of the answer

Figure 133 Writting an answer to the post.

Figure 132 Answer given successfully to the post.

Objective	Let user answer to the question of a post.
Action	Just click that message icon and add answer to the post.
Expected Result	The answer must be saved with a post.
Actual Result	The answer were successfully saved and shown to others.
Conclusion	The test was successfully.

Figure 135 System testing of anwering to a question post.

4.3.12. Test case 11: Edit answer

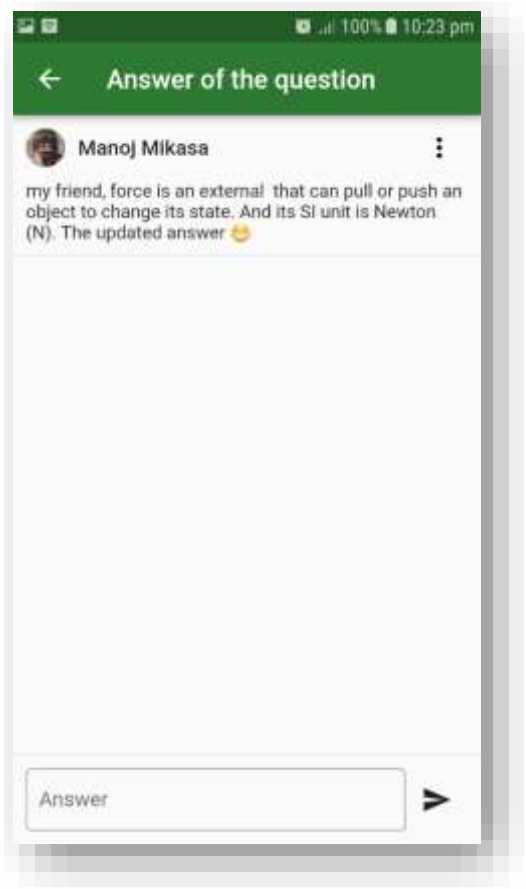


Figure 136 Answer updated successfully.

Objective	To let users edit their answer.
Action	Click to the three dots of answer and edit the answer.
Expected Result	Answer must be edited successfully.
Actual Result	Answer was edited.
Conclusion	The test was successfully.

Figure 137 System test of editing an answer.

4.3.13. Test case 12: Delete Answer

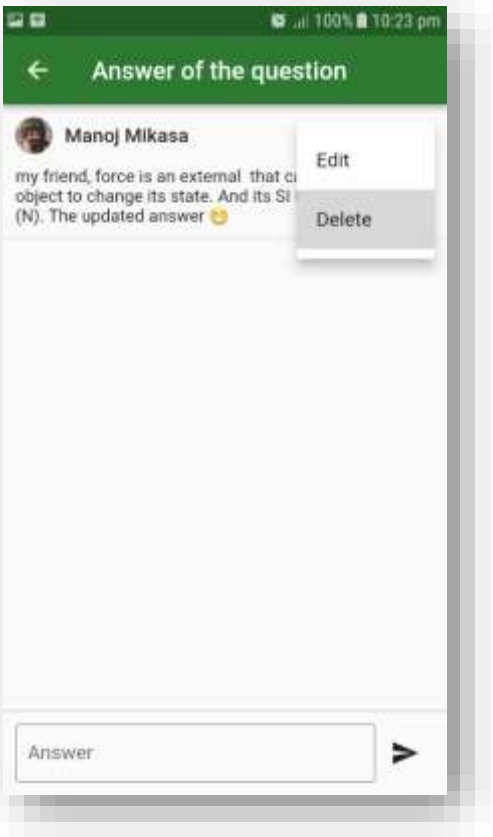


Figure 140 Deleting an answer.

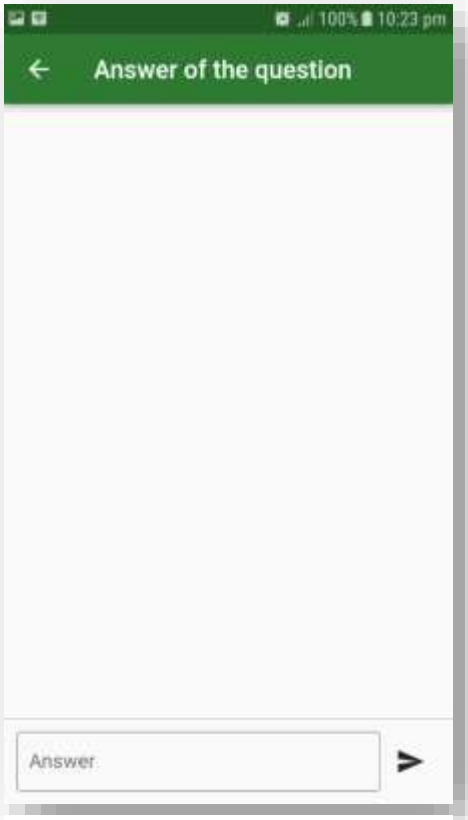


Figure 139 Answer deleted successfully.

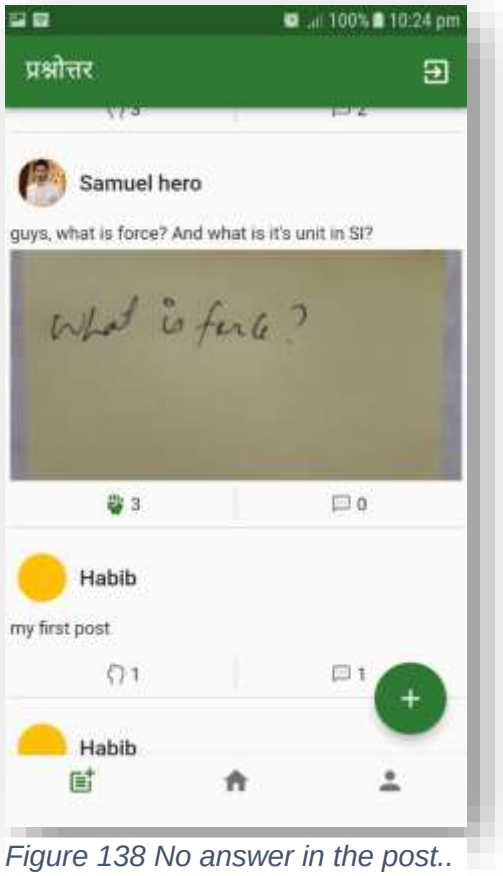


Figure 138 No answer in the post..

Objective	To let user delete their answer.
Action	Just click the three dots, and delte the post.
Expected Result	Answer must be deleted.
Actual Result	The answer was deleted successfully.
Conclusion	The test was successful.

Figure 141 System testing of deleting an answer.

4.3.14. Test case 13: Like the post

Figure 142 Post is liked.

Objective	To let user like a post for it's question and answer popularity.
Action	Click the fist icon.
Expected Result	The icon must turn green and increase the count by one.
Actual Result	As expected.
Conclusion	The test was successful.

Table 24 Result of system test case 14.

4.3.15. Test case 14: Dislike the post

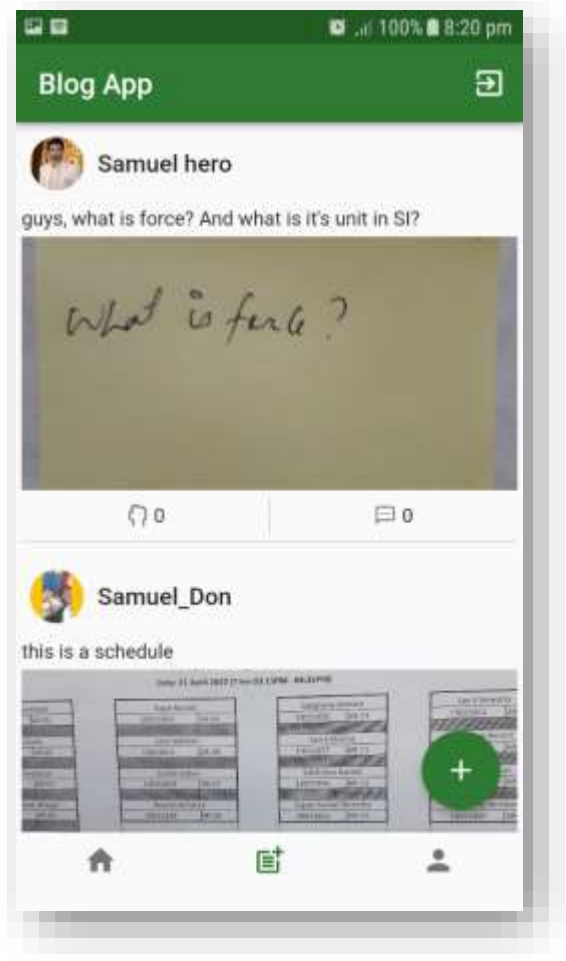


Figure 143 Post is disliked.

Objective	To dislike a post.
Action	Click the liked fist icon.
Expected Result	It must turn black and the count must decrease.
Actual Result	As expected, the icon was turned black and it's count decreased.
Conclusion	The test was successful.

Table 25 Result of system test case 15

4.3.16. Test case 15: Edit Update profile

Figure 146 User Manoj Rai to be updated.

Figure 145 Selecting image from the gallery.

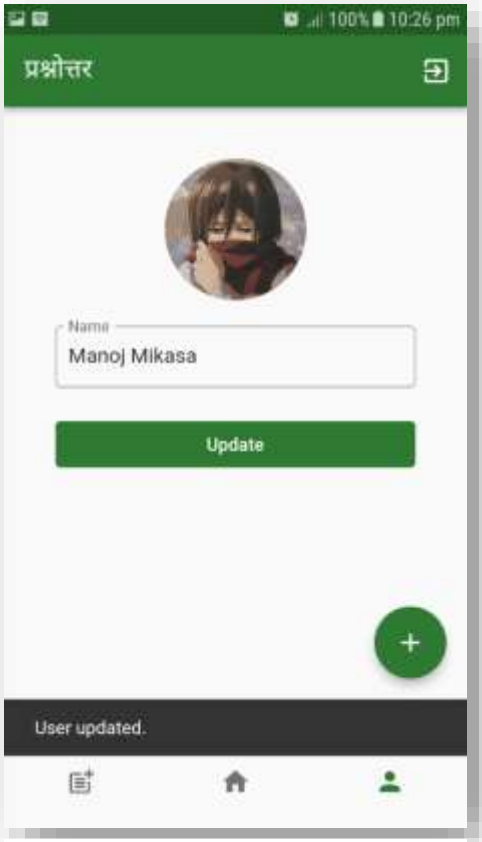


Figure 144 User profile and name updated successfully.

Objective	User must be able to edit their name and profile picture.
Action	User must click to the user icon in the nav bar and click on the picture and name, edit those and click update button.
Expected Result	Profile picture and name must be changed.
Actual Result	The name and picture were changed.
Conclusion	The test was successful.

Figure 147 System test of editing user profile.

4.4. Critical Analysis

The two tests: Unit testing and System testing showed all functionality of features present in the project that they worked as expected with no bugs. The results were as follows:

i) All the API's worked fine.
ii) All the API's were properly integrated with flutter app.
iii) The performance of the system was nice.
iv) The UI was very simple and user friendly.
v) Bookmark feature would be very helpful for students to refer it in exam time, so it would be better to include them.
vi) The whole project made from scratch was supported by an android phone, laptop, laravel, flutter and mysql databases.

The app seem to be very simple so it has to include features like follow person, and bookmark posts. In coming days hopefully those features would also be inlcuded.

Chapter 5: Conclusion

In life, we see, that even long roads have an end to them, and the same goes for this FYP. Finally, to include, I've learned and revised many IT skills like Flutter, Laravel, MySql database management, dart, Php, Figma designing, pressure handling, critical thinking, and many other life lessons during the journey of this project. Their real use in the IT world made my hard work overwhelming in my career as an IT professional. After finishing the FYP I had a blast of happiness and little sadness at the same time. As the twilight of my 3rd year has arrived in my eyes and flowered pink blossom flowers of gratitude in the garden of my heart for my college and friends. I have learned so much, especially from research and mistakes I made on the way here. The project was very difficult for me as I was very new to the backend and API.

A lot of time I needed to ask the module teacher about my FYP progress, but many times I was not able to show necessary progress on time. But my basketful thanks go to Mr. Ravi Rouniyar sir for always being there and helping me when I was in trouble.

Learning new things is never easy, if it was easy then everyone would have studied IT and done it. I also suffered pain and depression when I was not able to pull out a good design for PT and code it in VS Code. Sometimes I was not able to find out good research material. But after all, all that we can do is remain cool, patient, and keep hustling, and that’s what I did. Due to this writing, this conclusion today is possible. Everything considered, in the end, I would like to say that building a working android app for the FYP is in some way fun and rewarding. I now have learned to sit in front of a laptop screen for several hours without distraction and complaining. Hope to study at this same pace to be a good software engineer someday and make an impact in other's life.

5.1. Legal, Social And Ethical Issues

5.1.1. Legal Issues

Legal issues are concerns or a situation of breakage of law made by the government which may sometime need the help of a lawyer to sort out is know as legal issues (Solicitors Regulation Authority, 2021). The project, PT does not violate any laws that would result in legal action. All the resources, tools, frameworks, and essential concepts to build the project are available online. The information provided by the user is only their

name and email address which would be safe and secure because it's in my hands (just kidding).

5.1.2. Social Issues

The social issue is an implication that arises when there's something that has negative consequences for other human beings, which might be based on their caste, culture, living habitat, educational differences, own's perspective on matters, and many more.

The PT has nothing to do with any of the social issues instead it's a students app that helps them to find the solution to their problems. PT could be used by any human being irrespective of their age or background.

5.1.3. Ethical Issues

Ethical issues arise when the thought of someone or a given decision or activity creates a conflict with a society's moral principles (MyAccountingCourse, 2022). The project doesn't hurt anyone's sentiments or beliefs. PT is an open free application that could be used by anyone. The codes and techniques used are free and available online.

5.1. Advantages

The advantages of using PT in our lives are as follows:

i) It is free of cost.
ii) It is a productive app since it helps in your studies.
iii) Users can choose a class, subject, and then a chapter to study or post a question.
iv) To the question, the user can answer or like.
v) It is helpful and useful for Nepali community students.
vi) It is ad-free.
vii) The app aware children make a good use of study apps and not just play games or watch movies online.
viii) It inspires people to share their knowledge and experience by bringing people together.

5.1. Limitations

i) The PT is available for only android.

ii) The PT is still not ready to be launched on market.

iii) It doesn't contain a bookmark and follows feature which could have enhanced the usage and effectiveness of the app.

iv) The app works for only grade 8 students.

5.1. Future Work

For the sake of FYP, the project that I submitted is enough. But if we have to talk about real life, then a lot of our things are supposed to be added.

i) In the login and registering, a real Gmail is not used, Which has to be changed.

ii) Students have to find the blog which they created or liked or commented on, therefore, if there was a feature called bookmark then, a lot easy would have been. Therefore bookmark feature is my next priority.

iii) Users are now not getting a notification when their posts and answer is been liked.

iv) Also, I need to add a comment feature on the answer, where people would leave their remarks on answers written by other fellow users.

Chapter 6: References

References

Android Operating System. (2021, 02 03). Retrieved from Investopedia: https://www.investopedia.com/terms/a/android-operating-system.asp

codemagic. (2019, 01 18). Retrieved from blog.codemagic.io: https://blog.codemagic.io/what-is-flutter-benefits-and-limitations/

Dart Overview. (2022, 04 26). Retrieved from dart.dev: https://dart.dev/overview

Google Play. (2021, 12 11). Retrieved from Google Playstore: https://play.google.com/work/apps/details?id=com.answers.answandroid

Google play store. (2021, 11 11). Retrieved from Playsotre_MeroStudy: https://play.google.com/work/apps/details?id=com.merostudy

Google Play_Mimo. (2021, 11 12). Retrieved from PlayStore: https://play.google.com/work/apps/details?id=com.getmimo

Google PlayStore. (2021, 11 12). Retrieved from PlayStore: https://play.google.com/work/apps/details?id=com.bartleby.learn

Google PlayStore. (2021, 11 12). Retrieved from PlayStore: https://play.google.com/work/apps/details?id=com.tag_hive.saathi.saathi

Guru. (2021). *Guru99*. Retrieved 10 1, 2021, from https://www.guru99.com/software-development-life-cycle-tutorial.html

JavaTPoint. (2020). *javaTpoint*. Retrieved Nov 11, 2021, from https://www.javatpoint.com/software-engineering-incremental-model

Kemp, S. (2021, Feb 12). *Digital 2021: Nepal*. Retrieved from Datareportal: https://datareportal.com/reports/digital-2021-nepal

Khabar, O. (2021, July 29). *OnlineKhabar*. Retrieved from OnlineKhabar: https://english.onlinekhabar.com/nepals-91-of-the-population-have-access-to-the-internet-regulator.html

Laravel - Oview Tutorialspoint. (2022, 04 26). Retrieved from tutorialspoint.com: https://www.tutorialspoint.com/laravel/laravel_overview.htm

Master. (2020). *Master2Teach*. Retrieved 10 01, 2021, from https://master2teach.com/software-engineering/rational-unified-process-rup/

MyAccountingCourse. (2022, 04 12). Retrieved from myaccountingcourse: https://www.myaccountingcourse.com/accounting-dictionary/ethical-issues#:~:text=What%20Does%20Ethical%20Issues%20Mean,question%20from%20an%20ethical%20standpoint.

PHP. (2022, 04 26). Retrieved from php.net: https://www.php.net/manual/en/intro-whatis.php

Point, T. (2020). *TutorialsPoint*. Retrieved Nov 11, 2021, from https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm

Solicitors Regulation Authority. (2021, 08 12). Retrieved from sra.org.uk: https://www.sra.org.uk/consumers/choosing/legal-issue/#:~:text=A%20legal%20issue%20is%20something,or%20resolved%20by%20the%20law.

Statcounter. (2022, 3 15). Retrieved from gs.statcounter.com: https://gs.statcounter.com/os-market-share/mobile/worldwide

Study. (2021). *Study*. Retrieved 10 01, 2021, from https://study.com/academy/lesson/what-is-the-rational-unified-process-methodology-tools-examples.html

Talend. (2022, 04 26). Retrieved from talend.com: https://www.talend.com/resources/what-is-mysql/

tutorialspoint. (2021). *tutorialspoint*. Retrieved Nov 11, 2021, from https://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm

Visual Studio Code Getting started. (2022, 04 27). Retrieved from code.visualstudio.com: https://code.visualstudio.com/docs

WENR, W. (2021). *WENR*. Retrieved 10 10, 2021, from https://wenr.wes.org/2018/04/education-in-nepal

Chapter 8: Appendix

8.1. Appendix A: Pre-Survey

8.1.1. Pre-Survey Form

Prashnottar

What is Prashnottar?

Prashnottar is an android app, for students from classes 4-12, where students can choose their class, then choose their subject and then choose a chapter and ask questions for the topics which they don't understand. In addition to this main feature, they can also answer others' questions and get likes, Upvotes, and rewards for the best answers to get themselves on the leader board.

Objectives of Prasnottar:

- Students would take help online with a specific platform developed for them.
- They don't have to always rely on school and tuition teachers to get answers and solutions to their problem.
- Bookmarks and revision notes will help them during exams.
- Students all over Nepal will be questioning and answering, so better results and answers will be delivered.
- A friendly and competitive platform will be built for Nepali students.

Why Prashnottar?

Nepal with a students population of more than 1 million (10 Lakh), still in 2021 doesn't have a specific robust platform for students to deal with their confusions, understand different concepts of science, math, and many more, ask questions, or give answers at free of cost.

np05cp4a190148@iic.edu.np Switch account

Table 26 Pre-survey form part 1

Email *

Your email

1. Your Name *

Your answer

2. Do you have any brothers or sisters, or neighboring children studying in class from 4-12? *

- I do
- I don't

3. Have you ever helped them or any of your juniors with studying?(E.g. Helping in Math homework) *

- I have
- I haven't
- Never

Table 27 Pre-survey form part 2

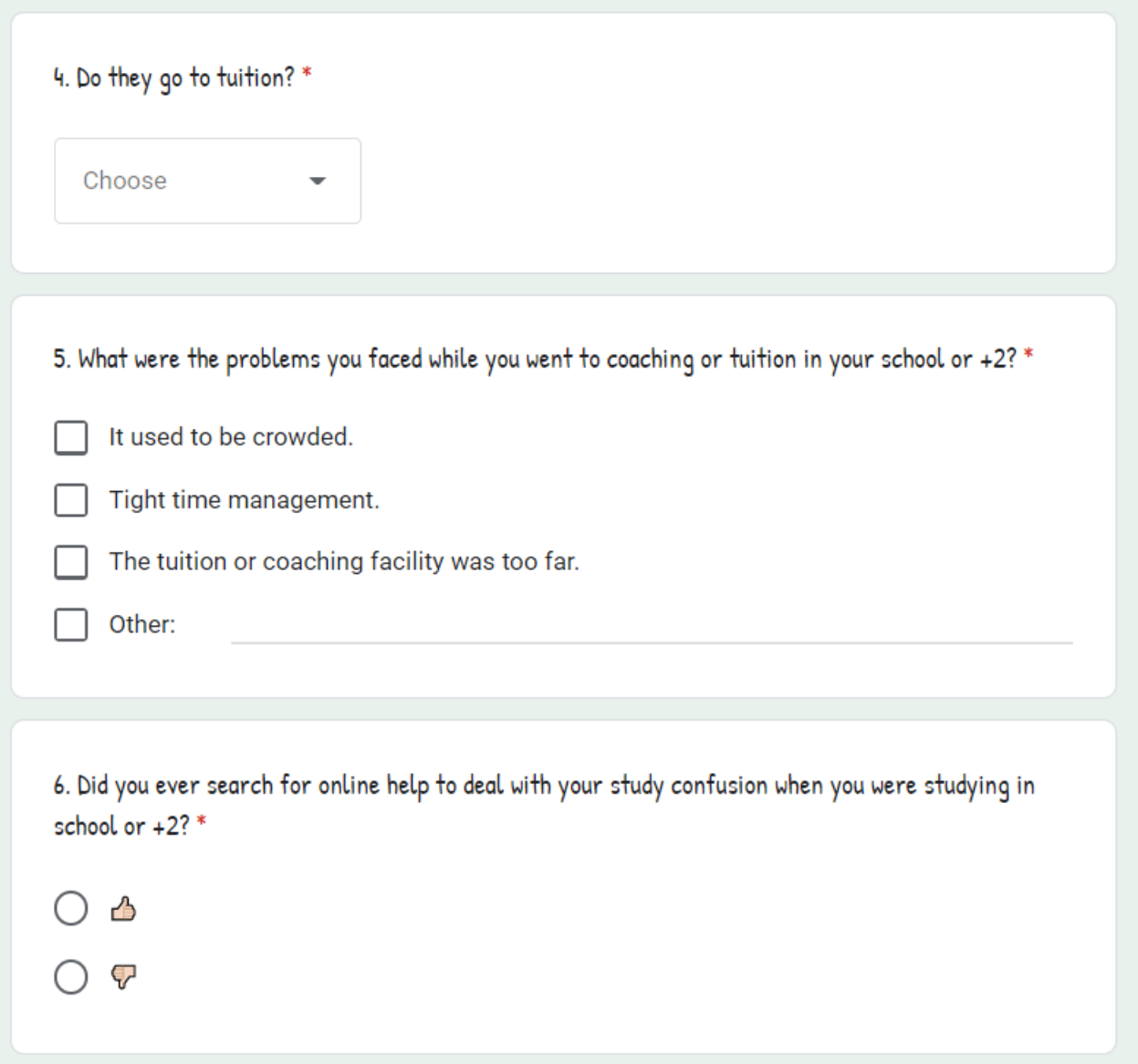


Table 28 Pre-survey form part 3

7. Did you always get the well-explained solutions to your searched problem? *

- ○ 😣
- ○ 😐
- ○ 😊
- ○ 😎

8. What features below would be good for everyone's convenience? *

- ☐ Easy search implication
- ☐ Correct and well-explained answers, pictures from individuals
- ☐ Scoring and reward to good answers
- ☐ Other: _____

Table 29 Pre-survey form part 4

9. In your assumption, what percentage of students still today, opt for online to learn, get solutions and read well-explained answers?

- ◯ (1-30)%
- ◯ (30-60)%
- ◯ (60-90)%
- ◯ (90-100)%

10. Would you recommend this app to your brothers, sisters or any of your juniors? *

- ◯ Don't recommend
- ◯ Strongly don't recommend
- ◯ Neutral
- ◯ Recommend
- ◯ Strongly recommend

Table 30 Pre-survey form part 5

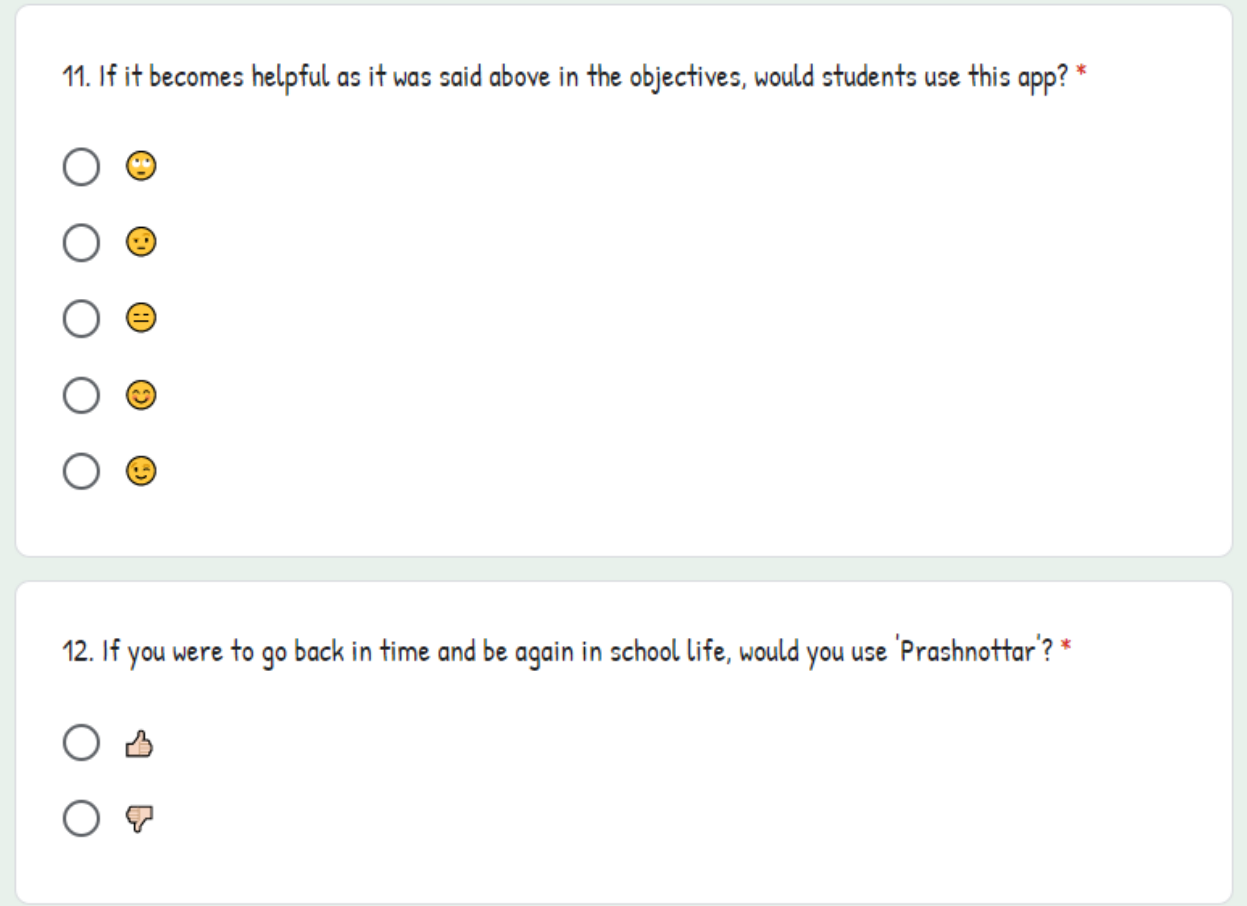


Table 31 Pre-survey form part 6

13. Would you think,'Prashnottar' would be helpful for students? *

- ○ 😖
- ○ 😕
- ○ 😑
- ○ 😊
- ○ 😊

14. Rate the FYP project interms of its usefulness and uniqueness. *

1	2	3	4	5
○	○	○	○	○

15. It would be great if you could leave your wonderful insights and helpful comments for more edges to improve.(No words count restrictions. I would be happy to read all of those.) *

Your answer

Table 32 Pre-survey form part 7

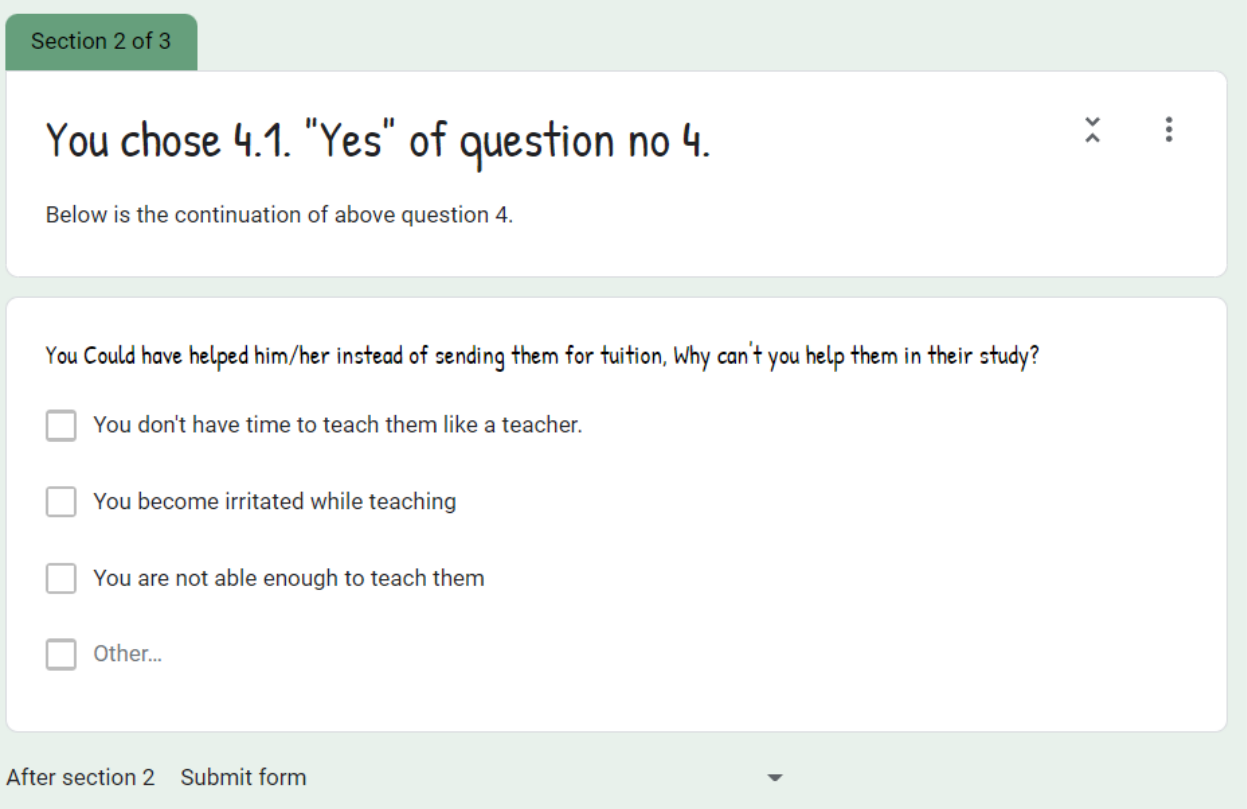


Figure 148 Pre survey form part 8

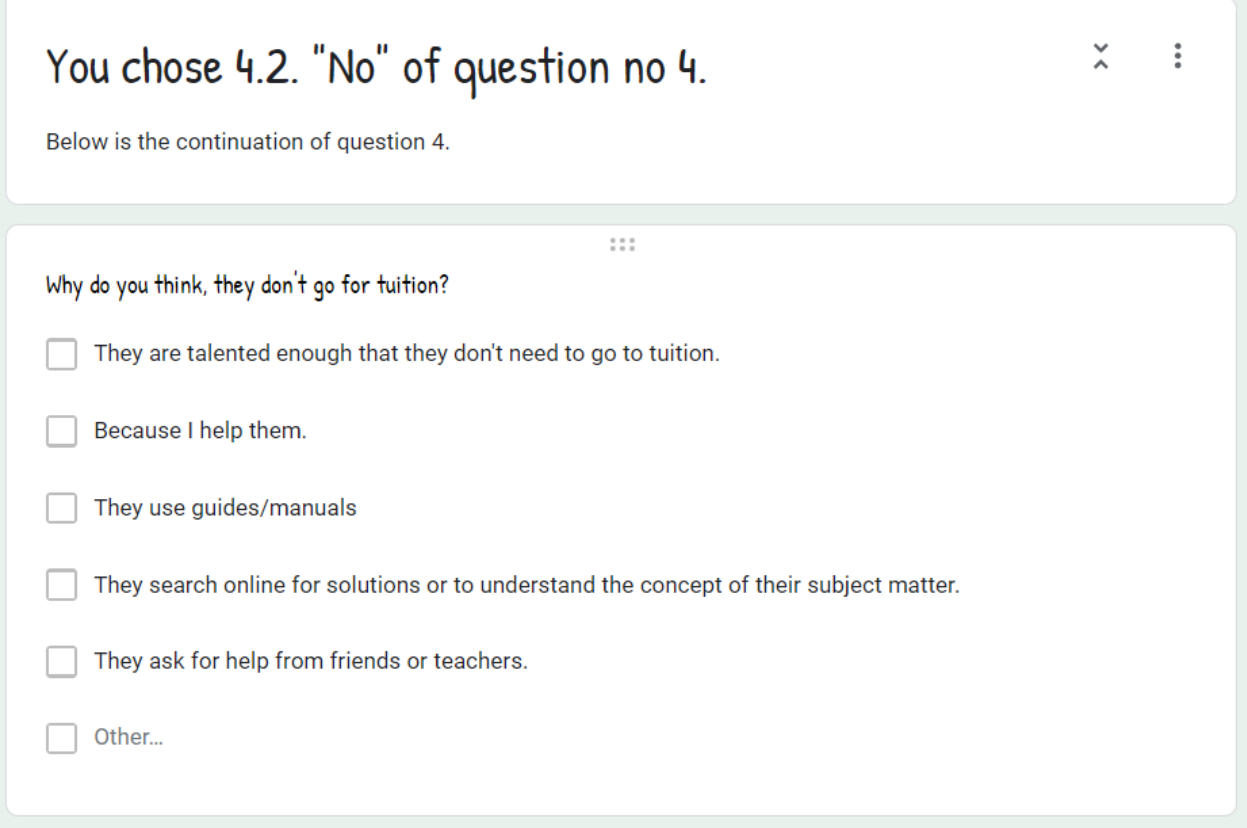


Figure 149 pre survey form part 9

8.1.2. Sample Of Filled Pre-Survey Forms

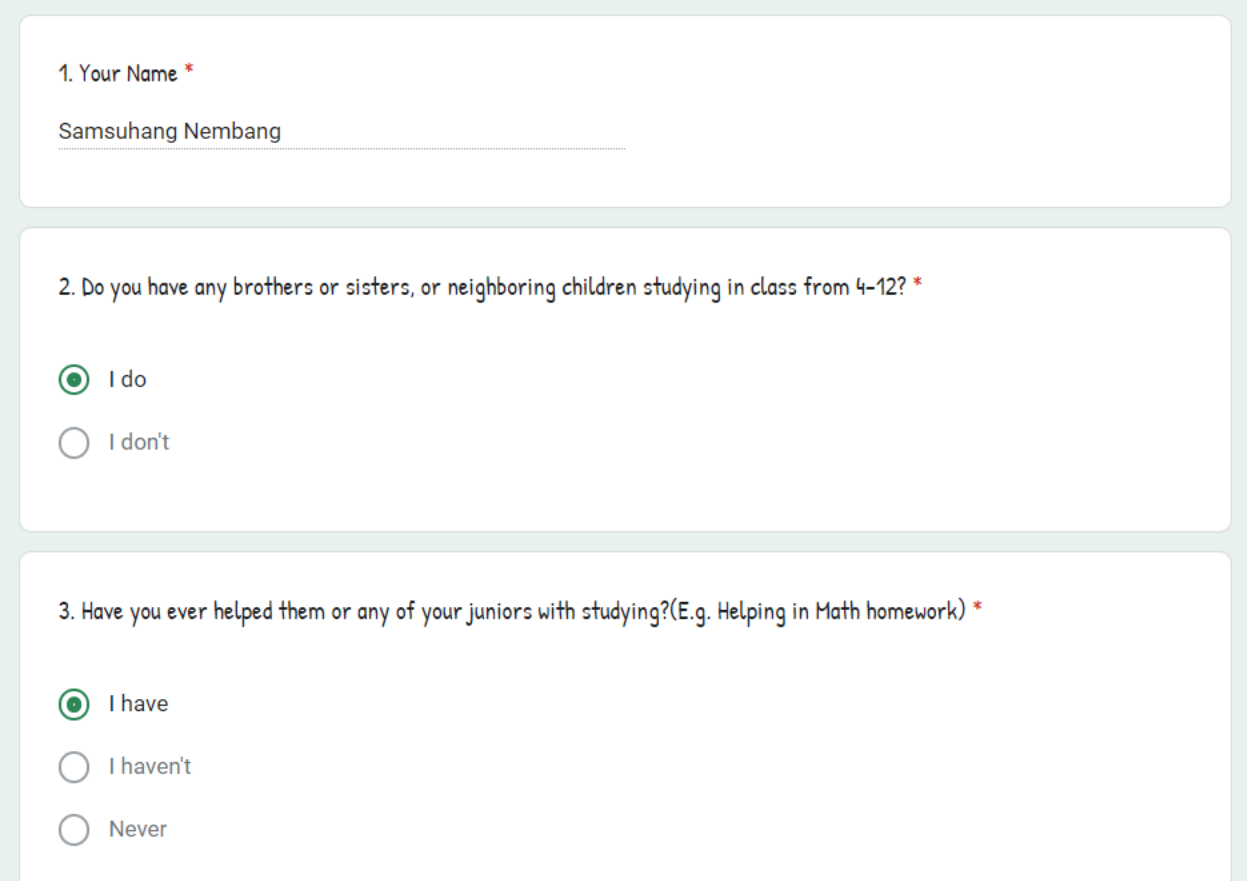


Figure 150 Pre survey form part 10

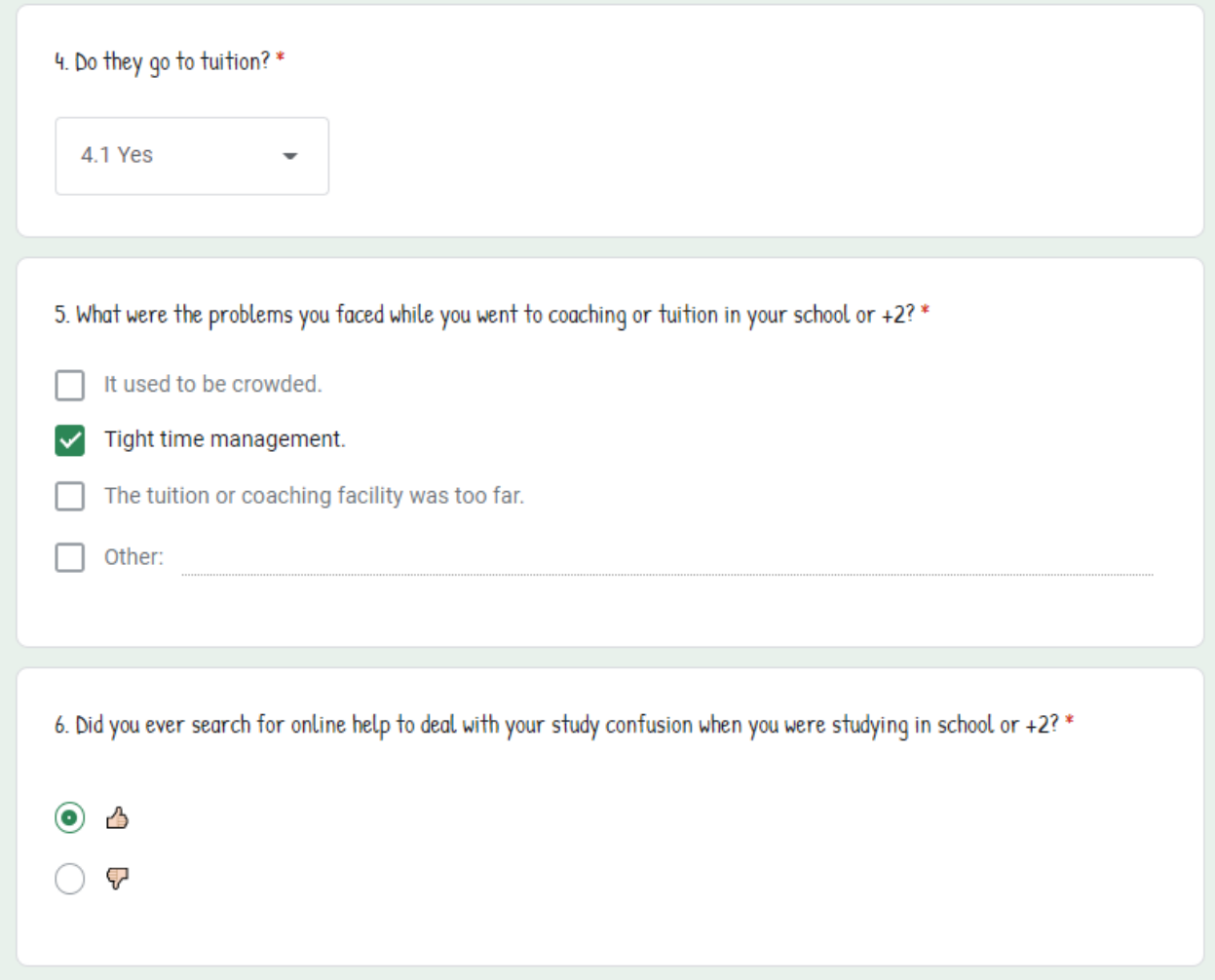


Figure 151 Pre survey form part 11

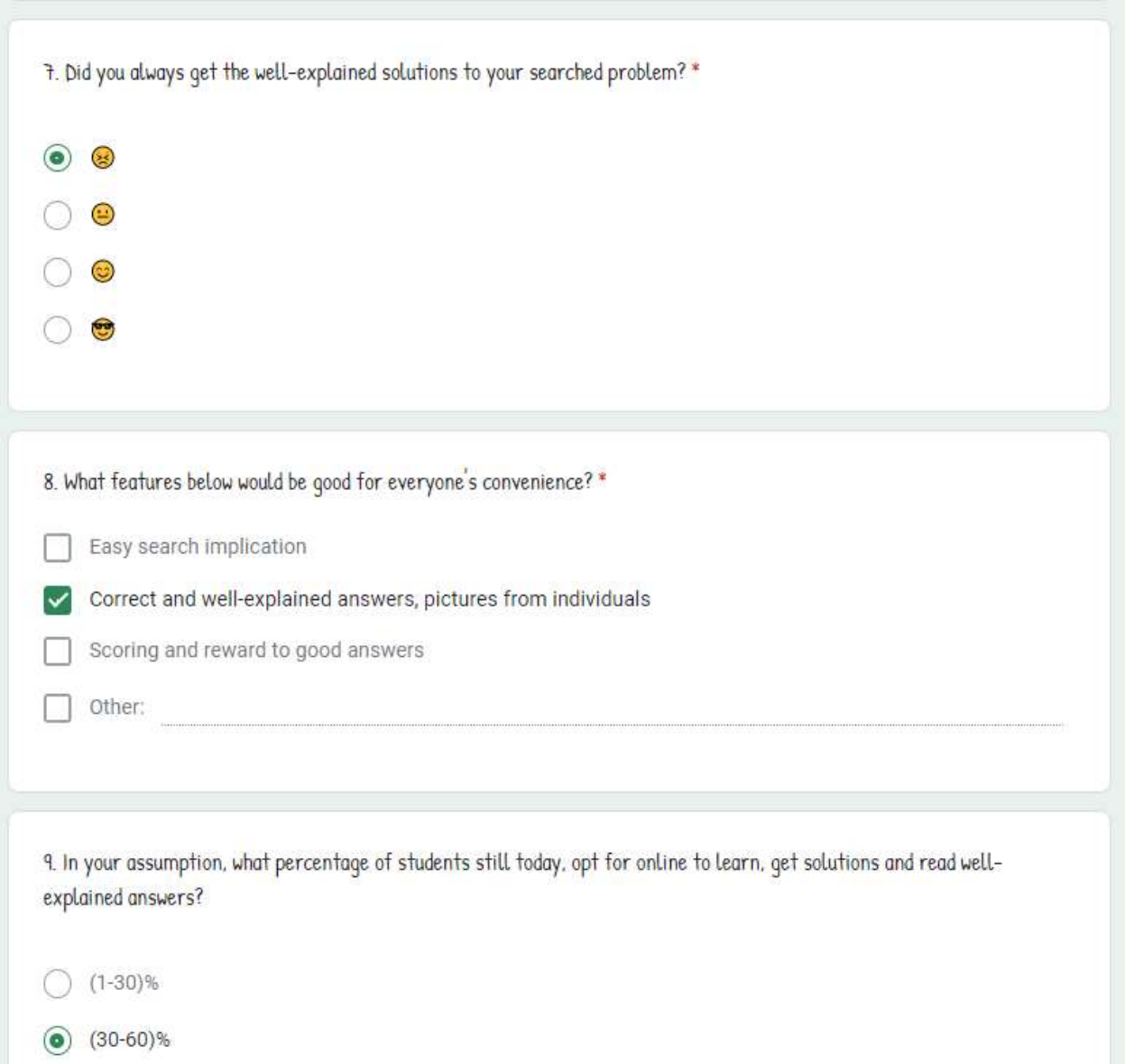


Figure 152 Pre survey form part 12

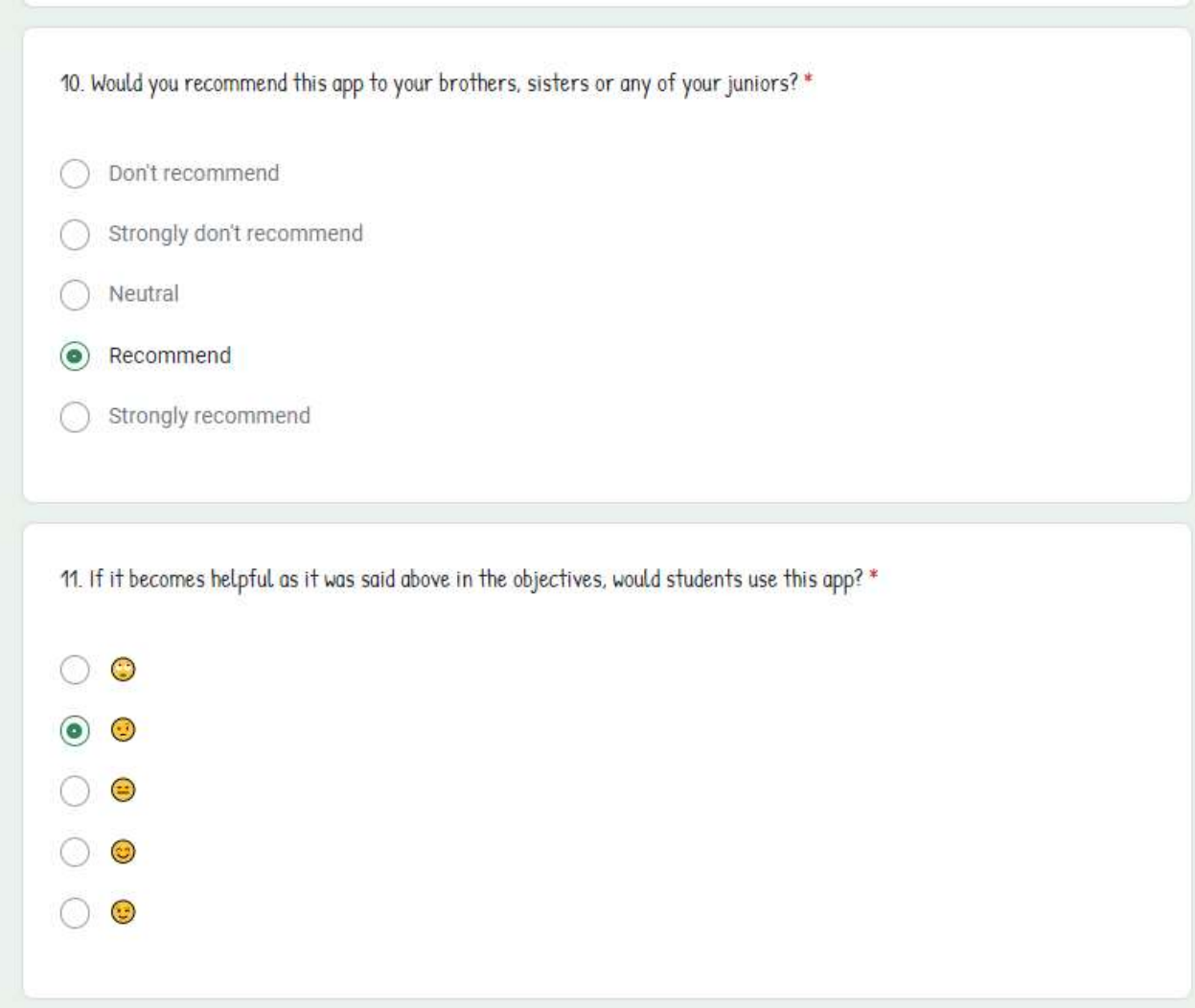


Figure 153 Pre survey form part 13

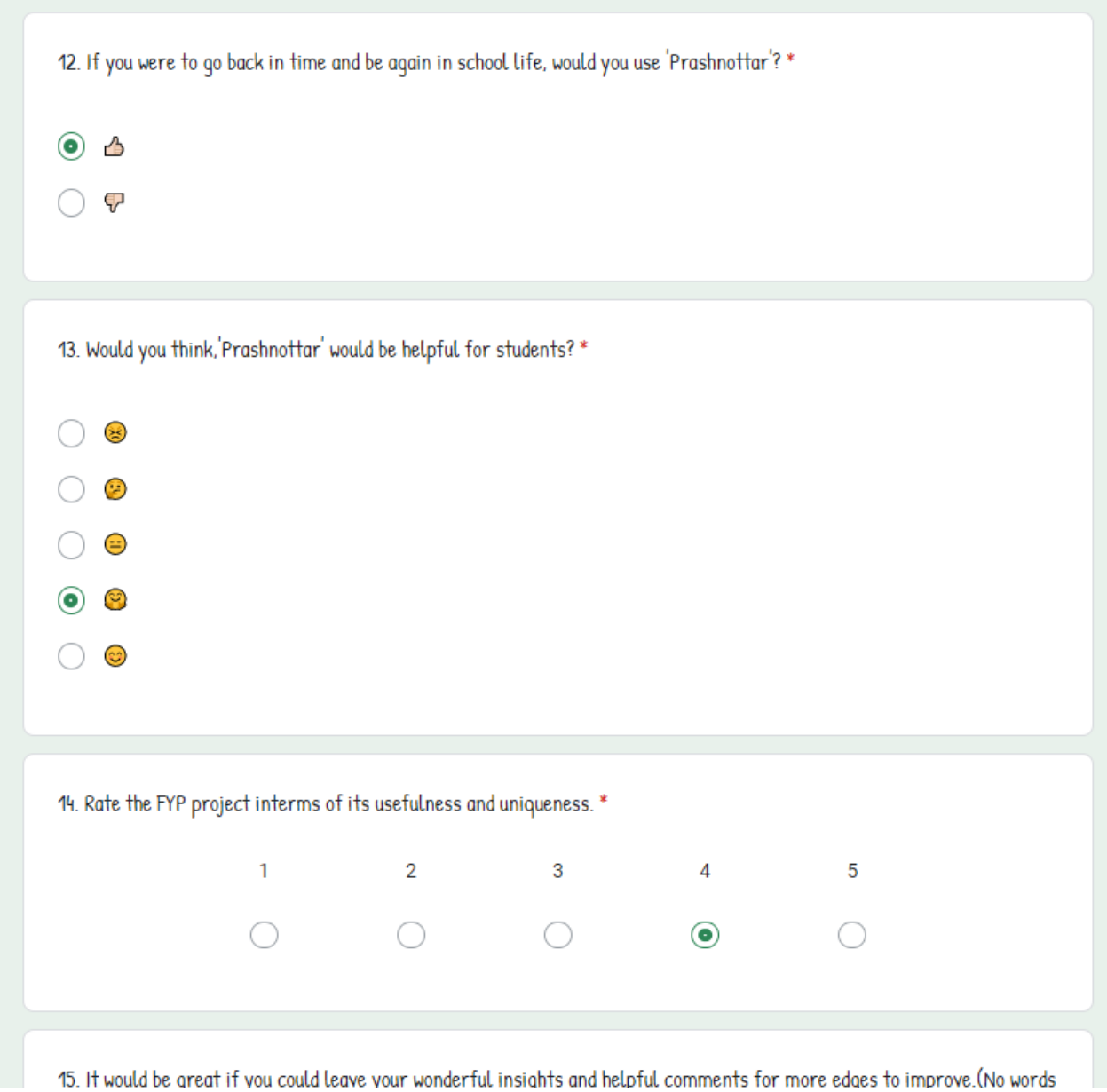


Figure 154 Pre survey form part 14

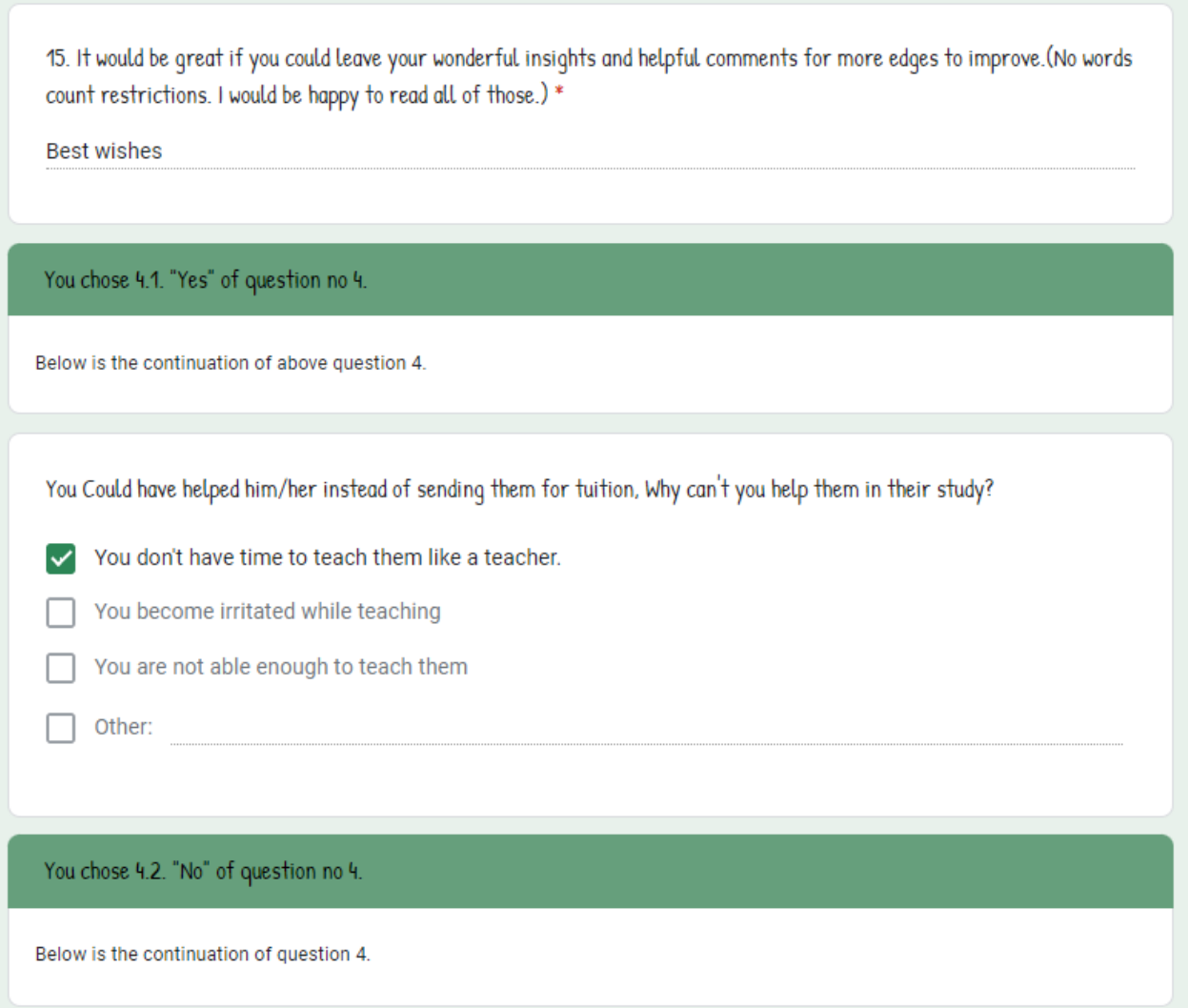


Figure 155 Pre survey form part 15

8.1.3. Pre-Survey Result

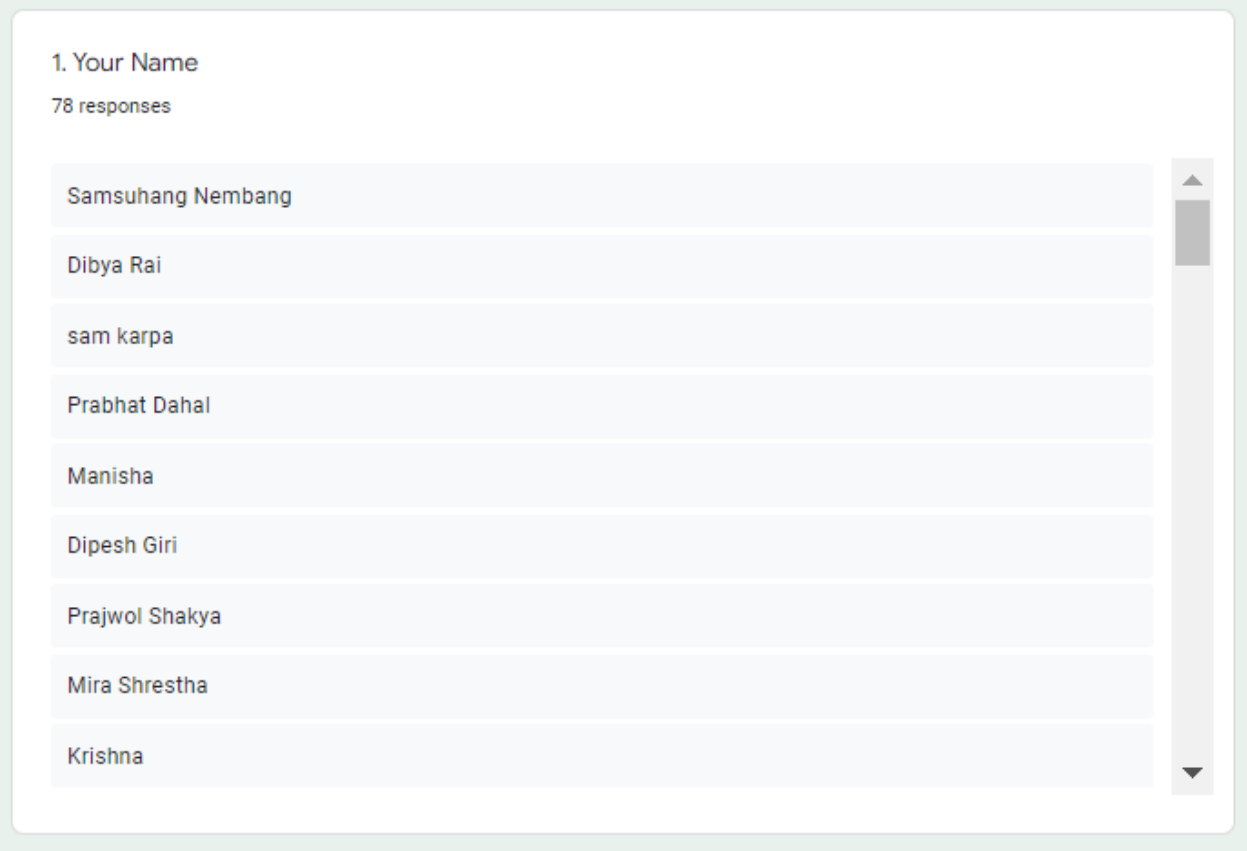


Figure 156 Pre survey form result part 1

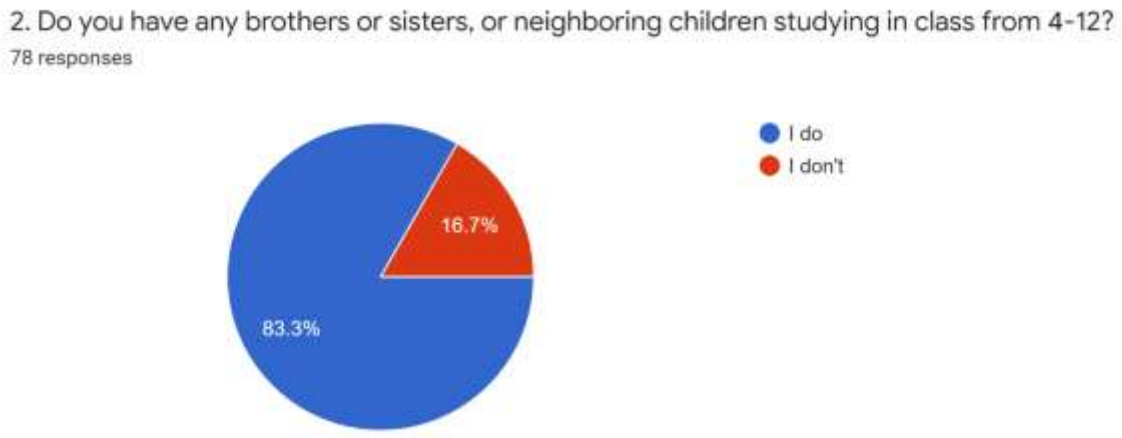


Figure 157 Pre survey form result part 2

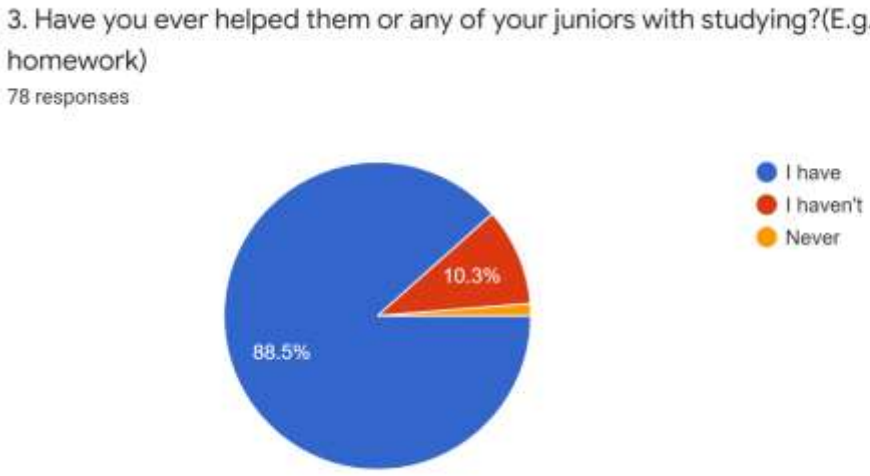


Figure 158 Pre survey form result part 3

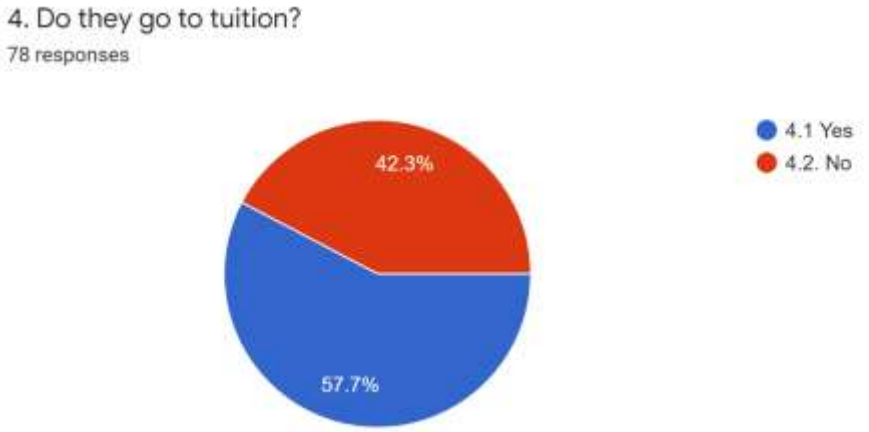


Figure 159 Pre survey form result part 4

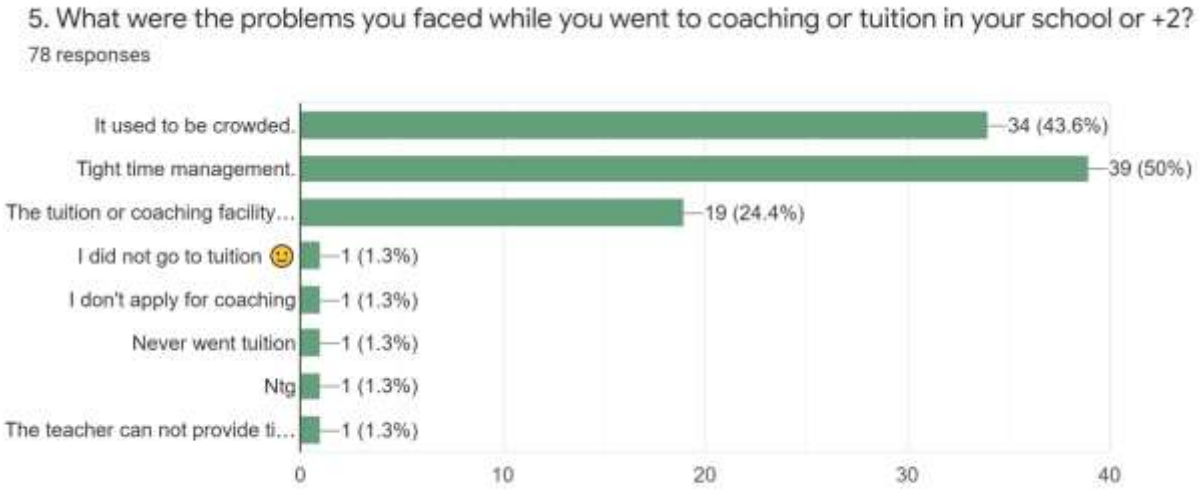


Figure 160 Pre survey form result part 5

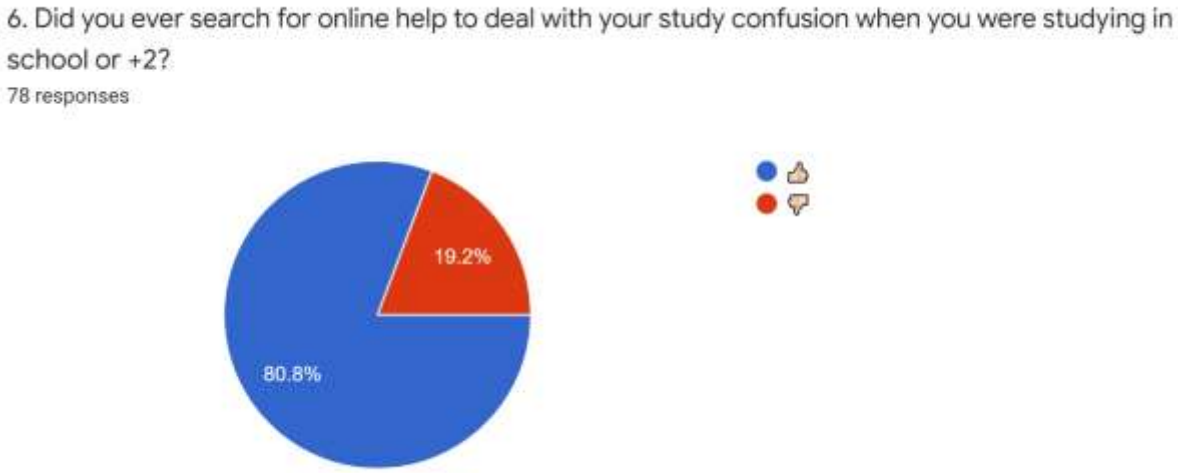


Figure 161 Pre survey form result part 6

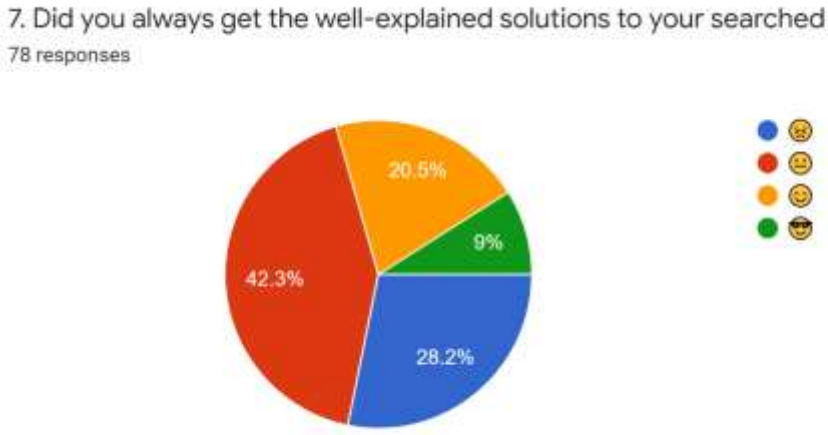


Figure 162 Pre survey form result part 7

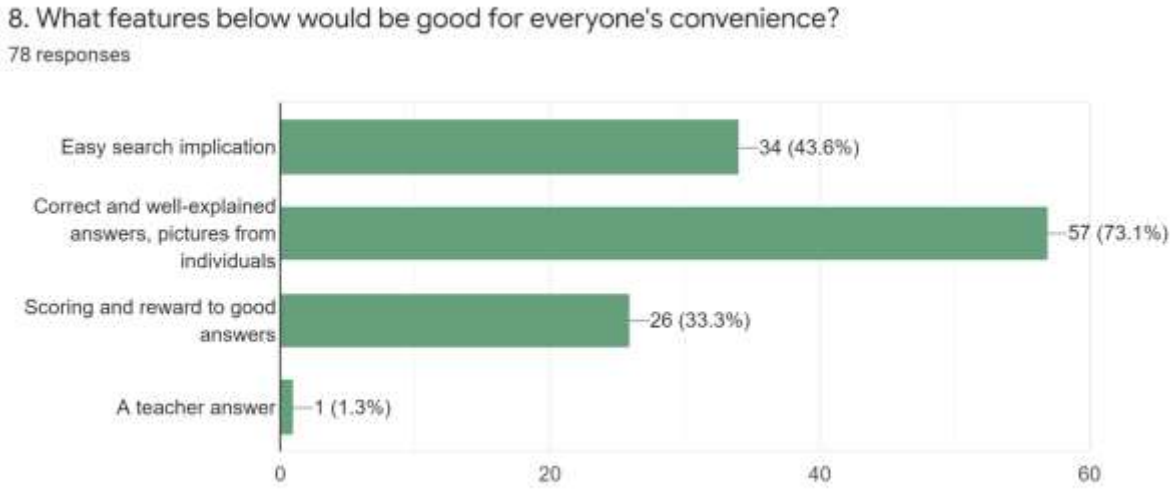


Figure 163 Pre survey form result part 8

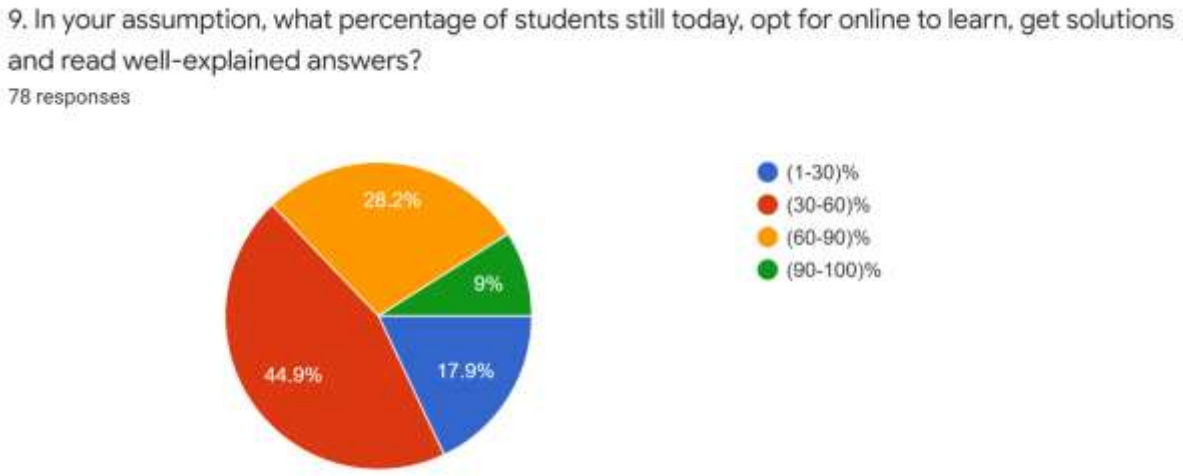


Figure 164 Pre survey form result part 9

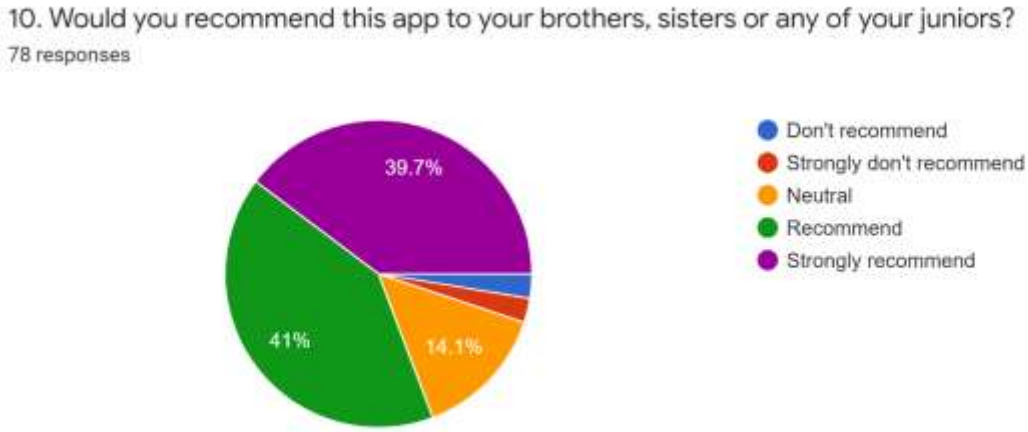


Figure 165 Pre survey form result part 10

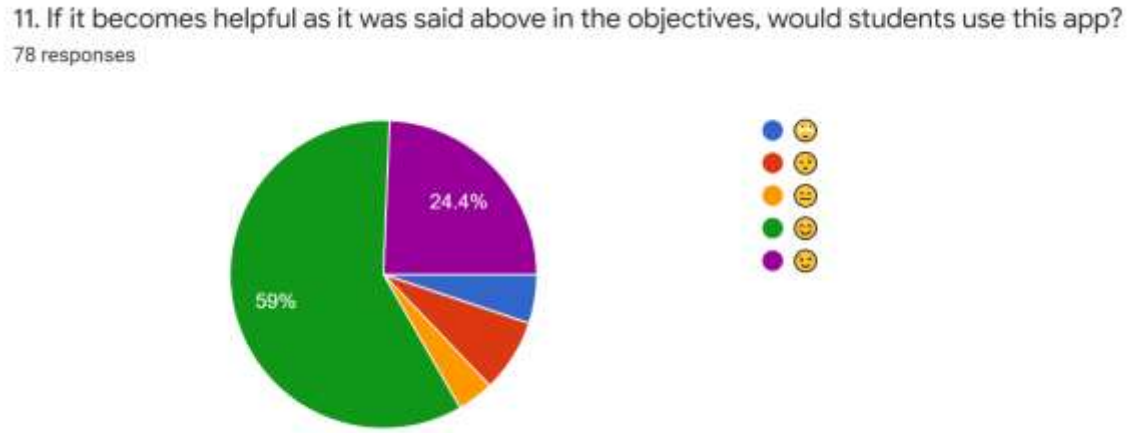


Figure 166 Pre survey form result part 11

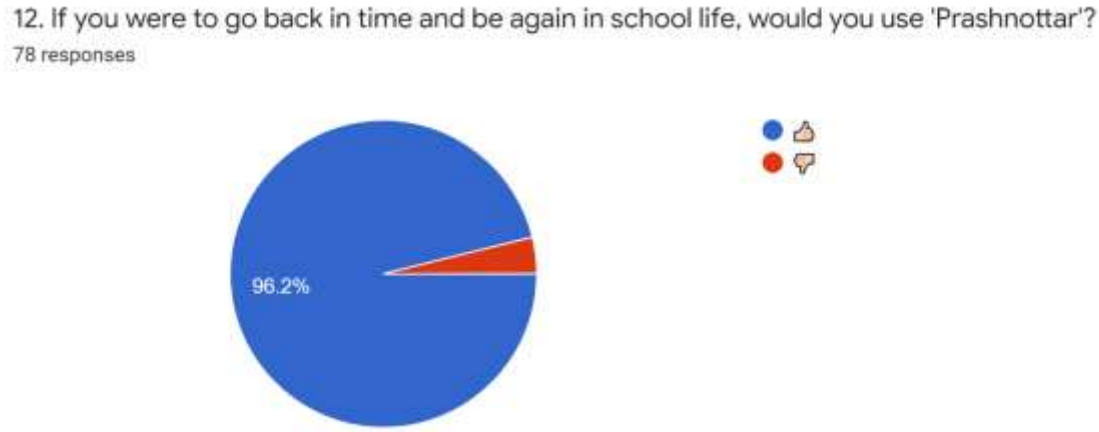


Figure 167 Pre survey form result part 12

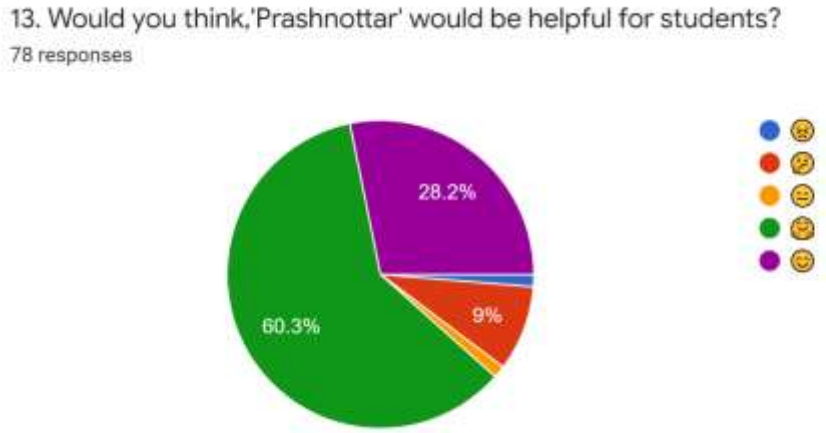


Figure 168 Pre survey form result part 13

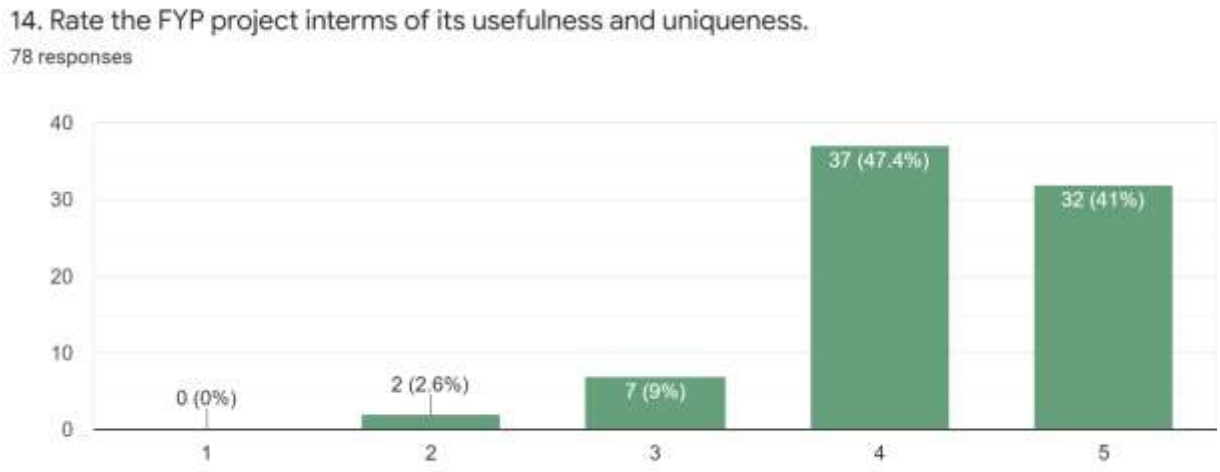


Figure 169 Pre survey form result part 14

Figure 170 Pre survey form result part 15

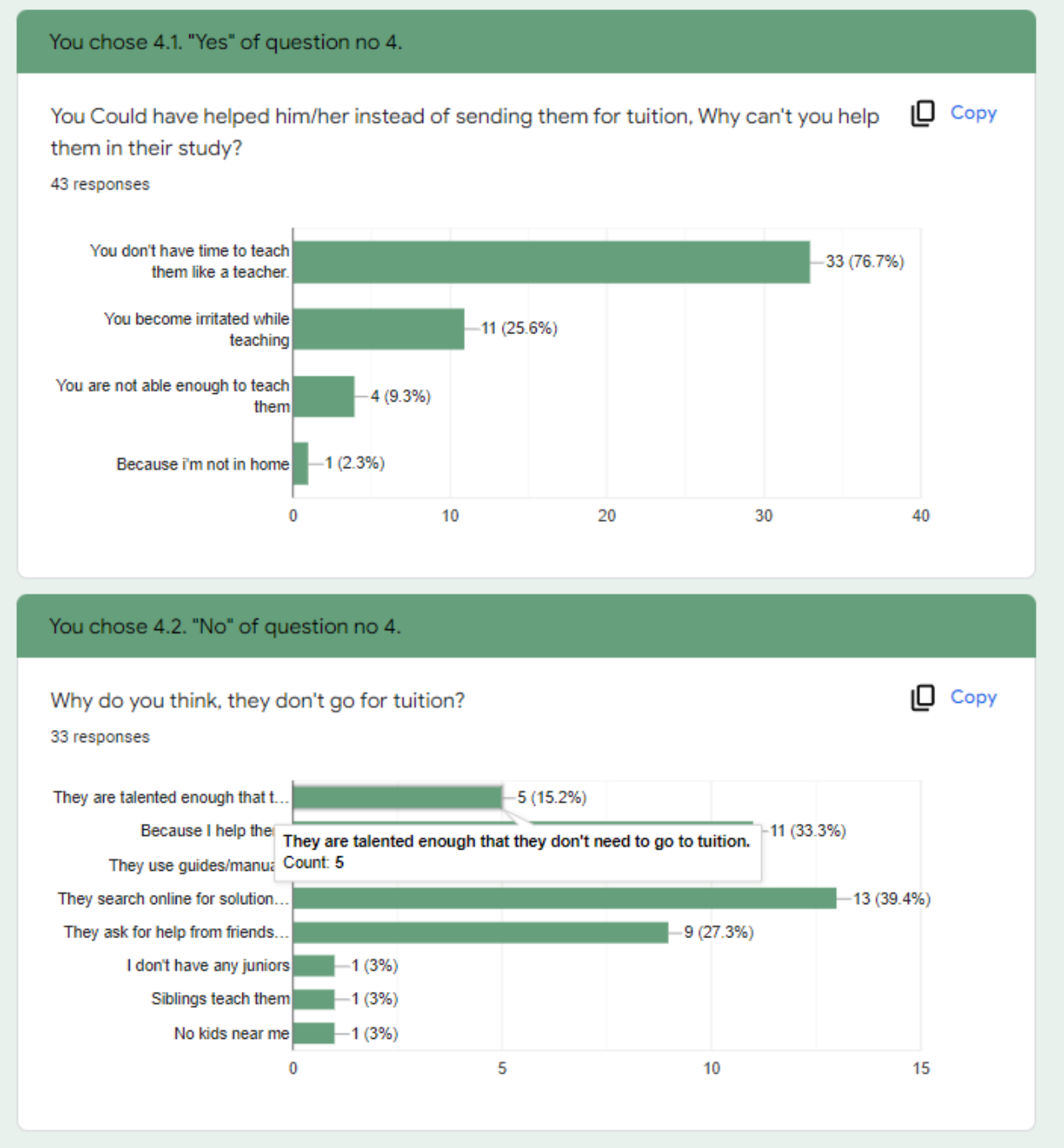


Figure 171 Pre survey form result part 16

8.2. Appendix

B: Post-Survey

8.2.1. Post-Survey Form

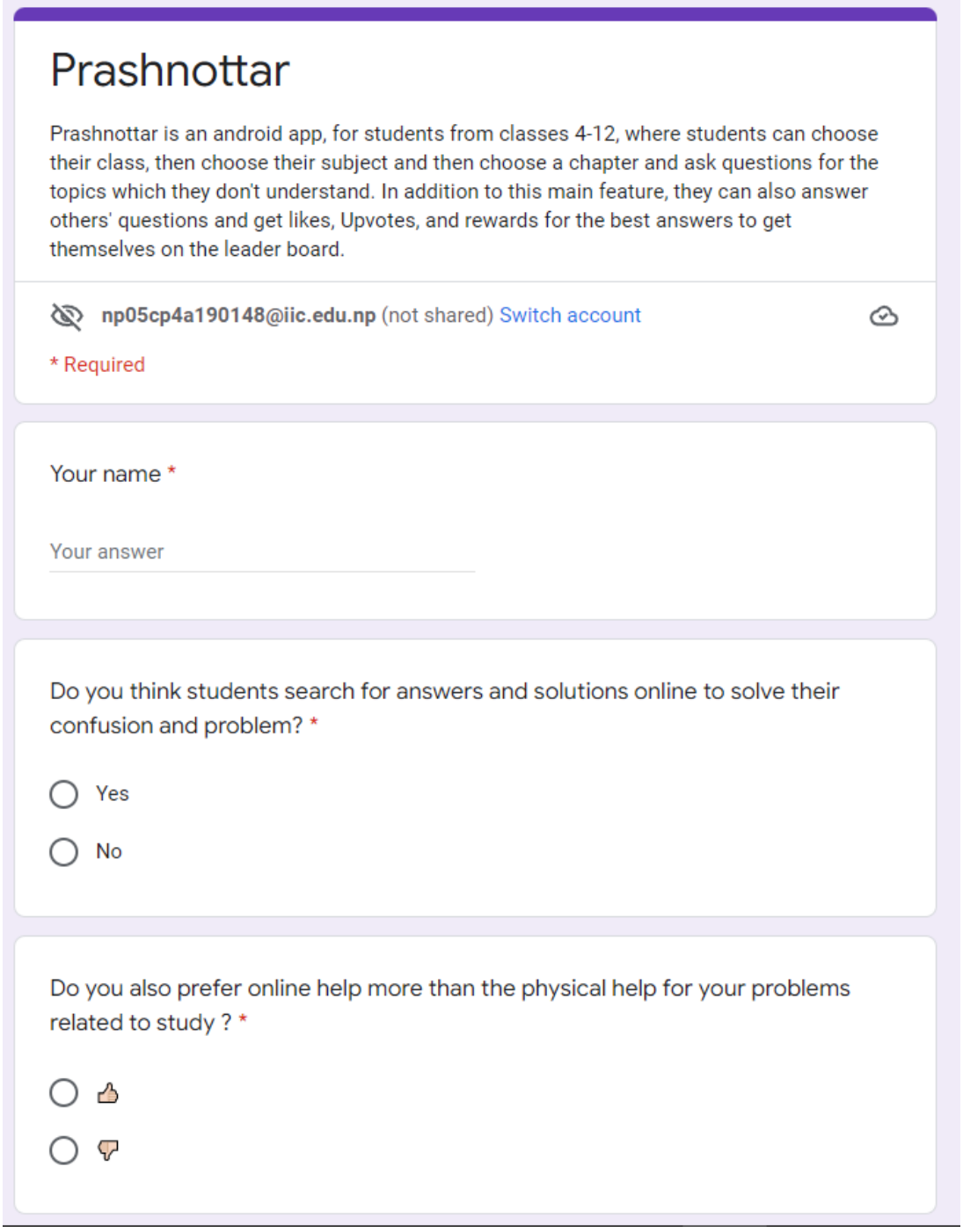


Figure 172 Post Survey question form part 1

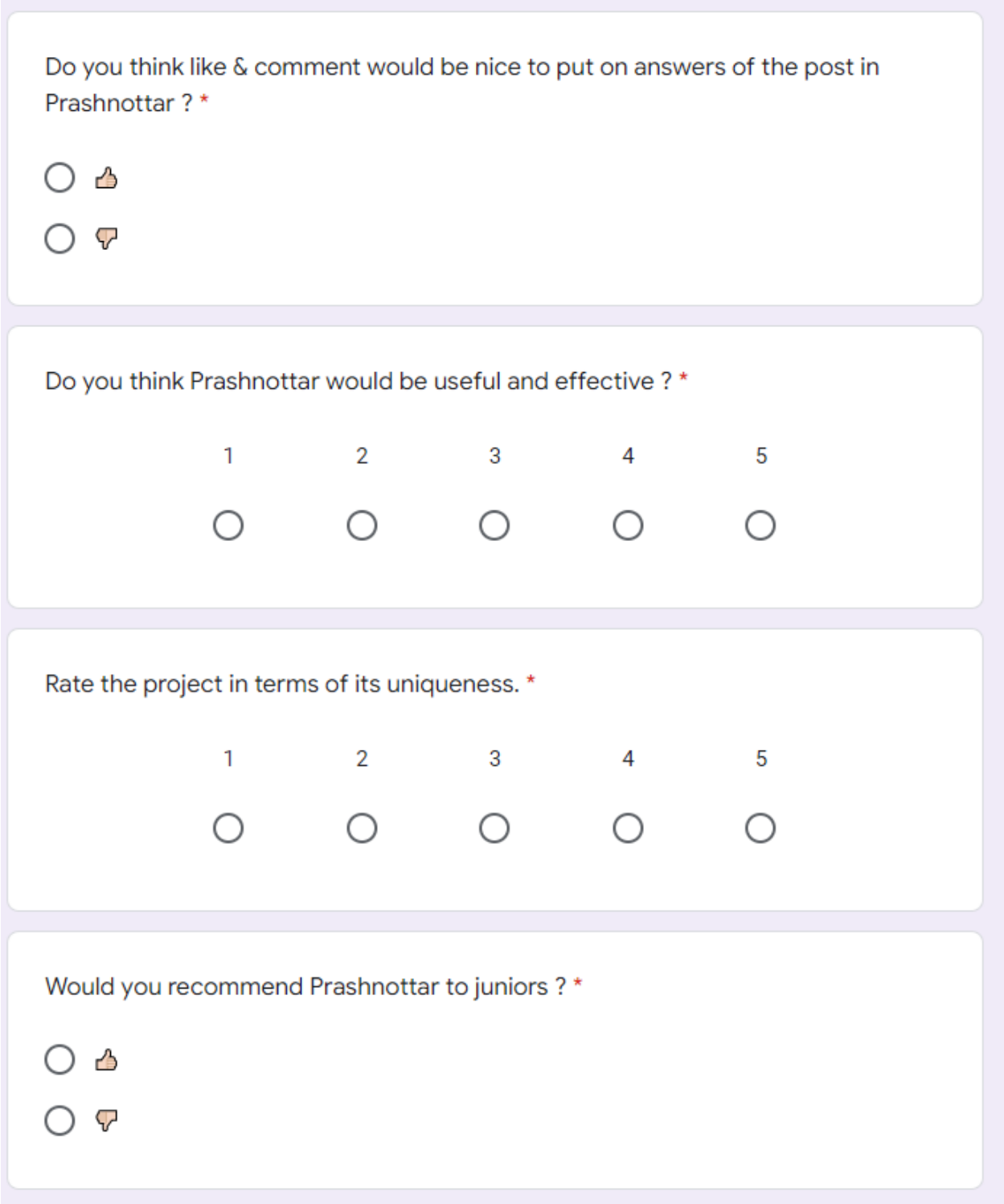


Figure 173 Pre survey form result part 2

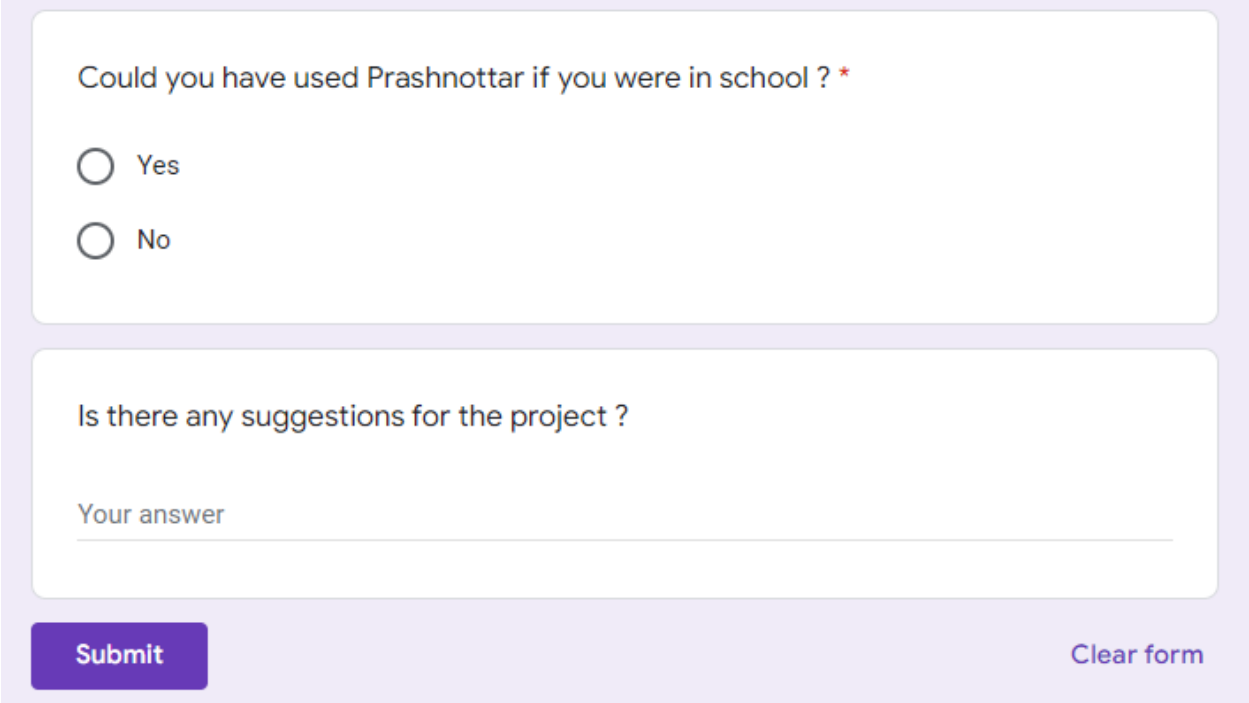


Figure 174 Pre survey form result part 3

8.2.2. Sample Of Filled Pre-Survey Forms

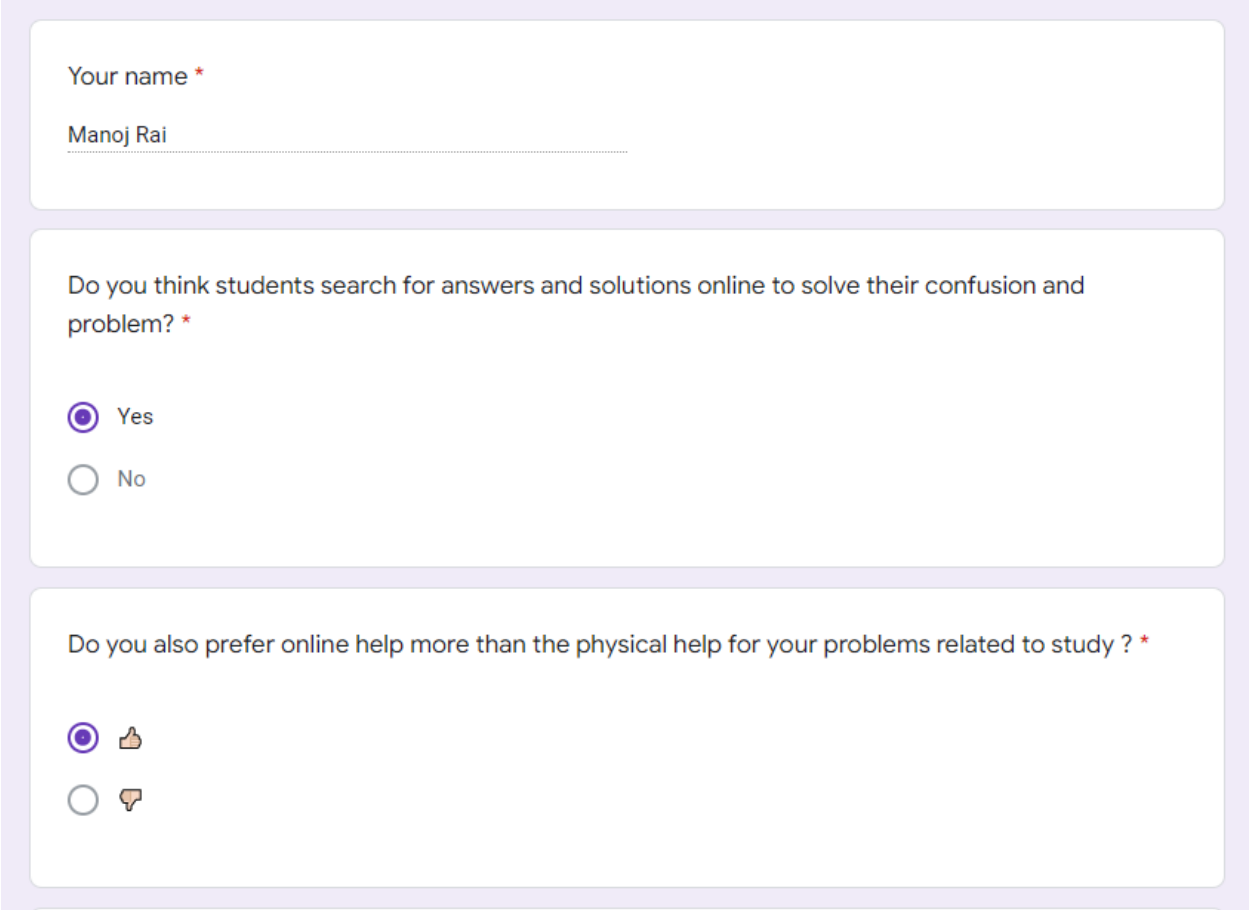


Figure 175 Pre survey filled form part 1

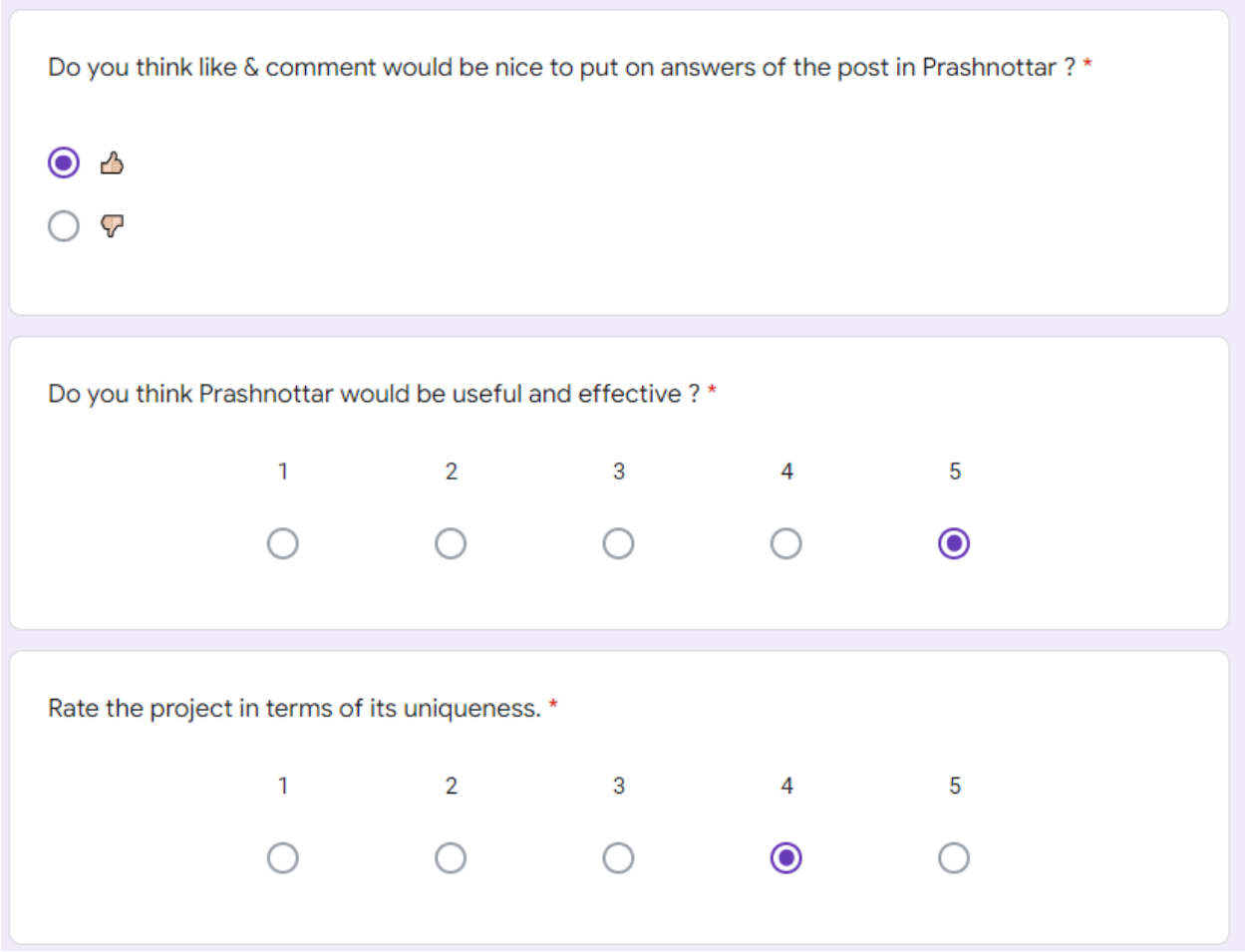


Figure 176 Pre survey filled form part 2

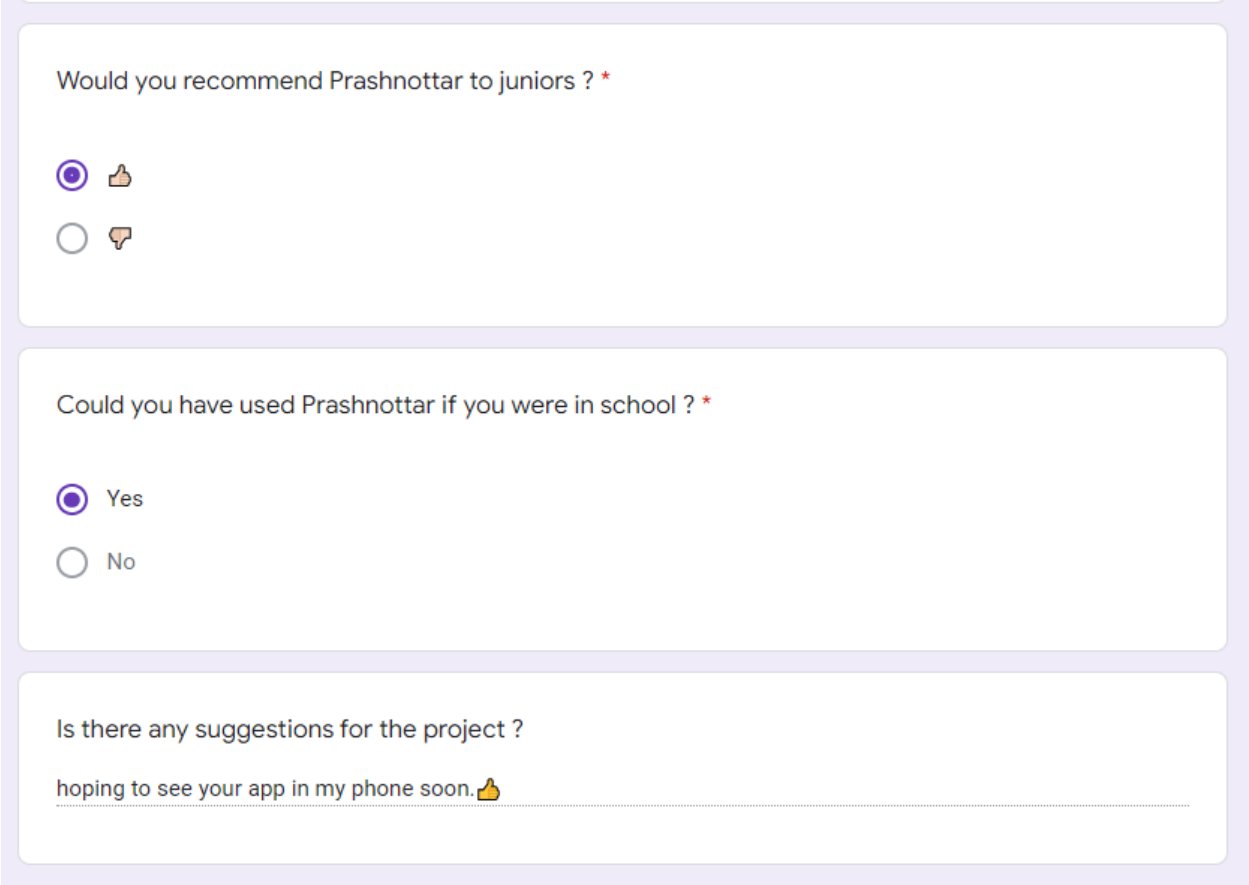


Figure 177 Pre survey filled form part 3

8.2.3. Post-Survey Result

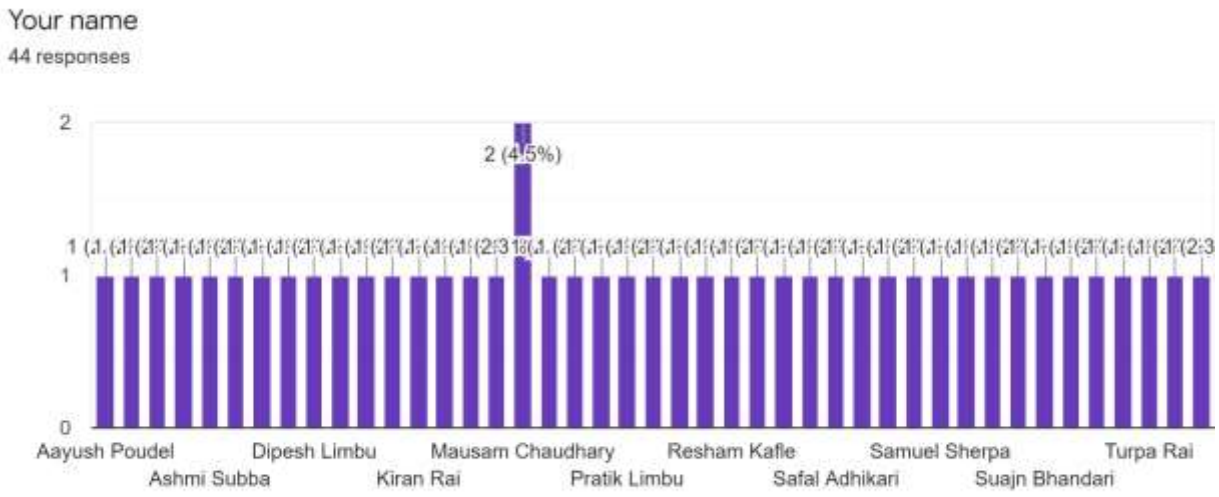


Figure 178 Pre survey result part 1

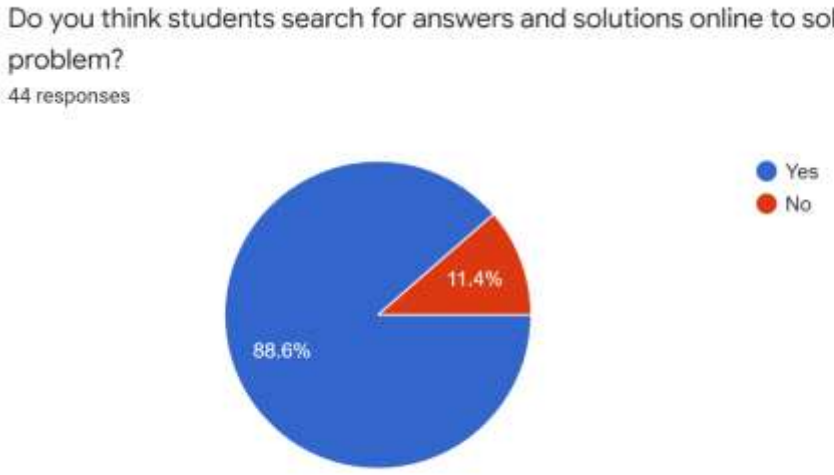


Figure 179 Pre survey result part 2

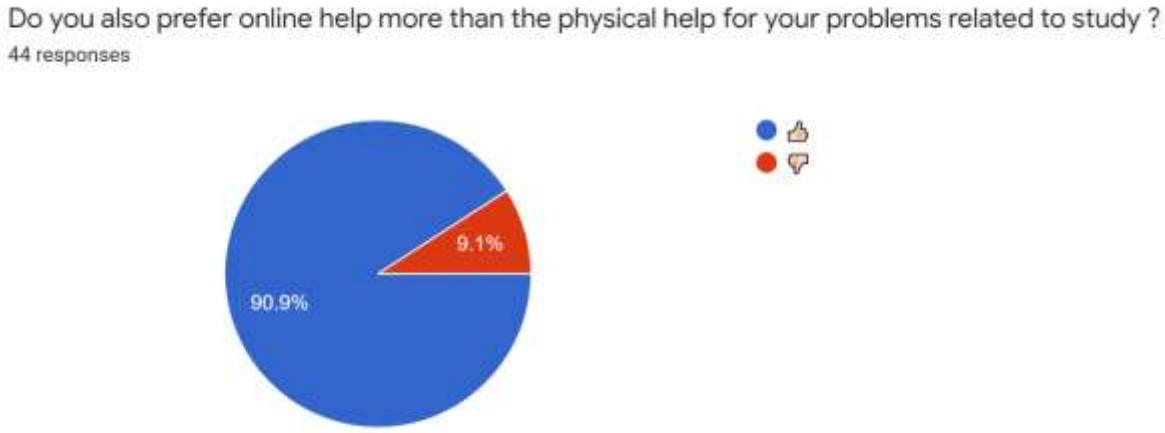


Figure 180 Pre survey result part 3

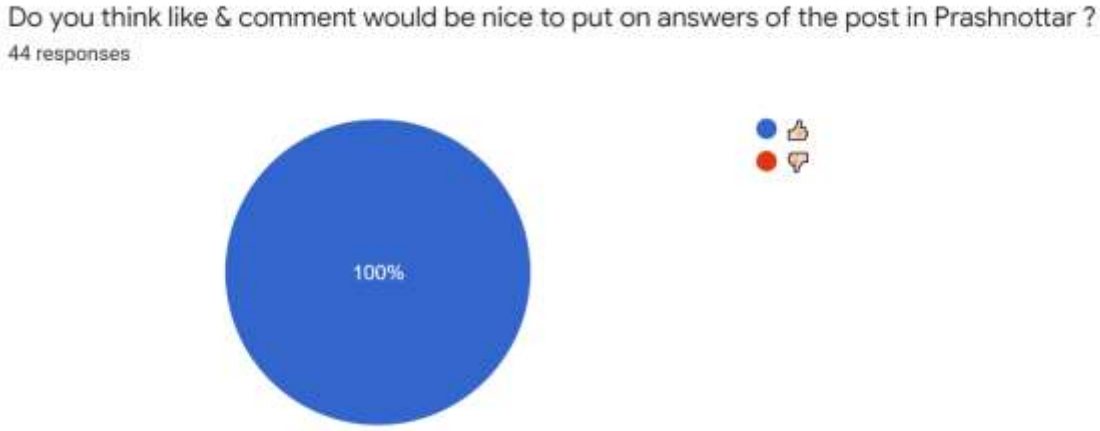


Figure 181 Pre survey result part 4

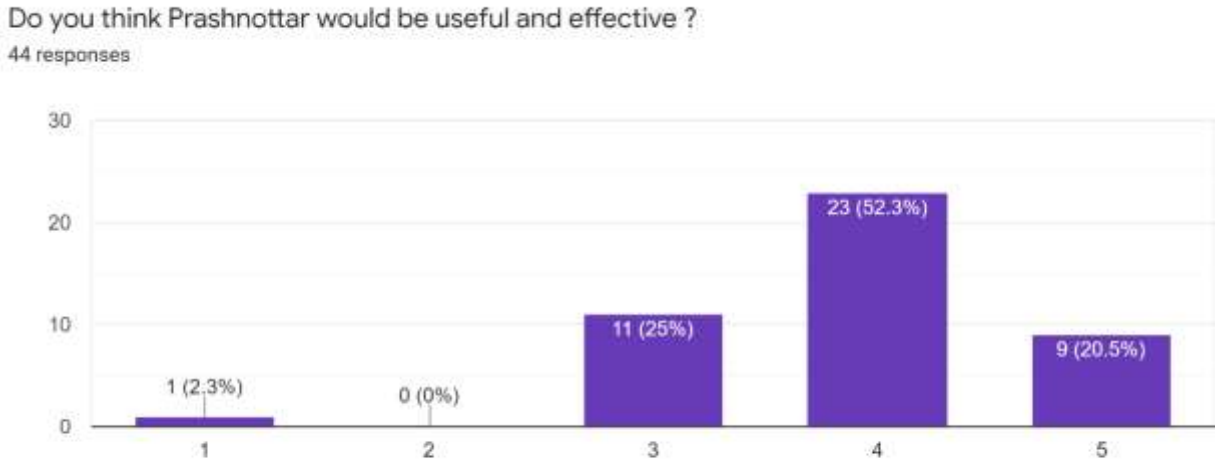


Figure 182 Pre survey result part 5

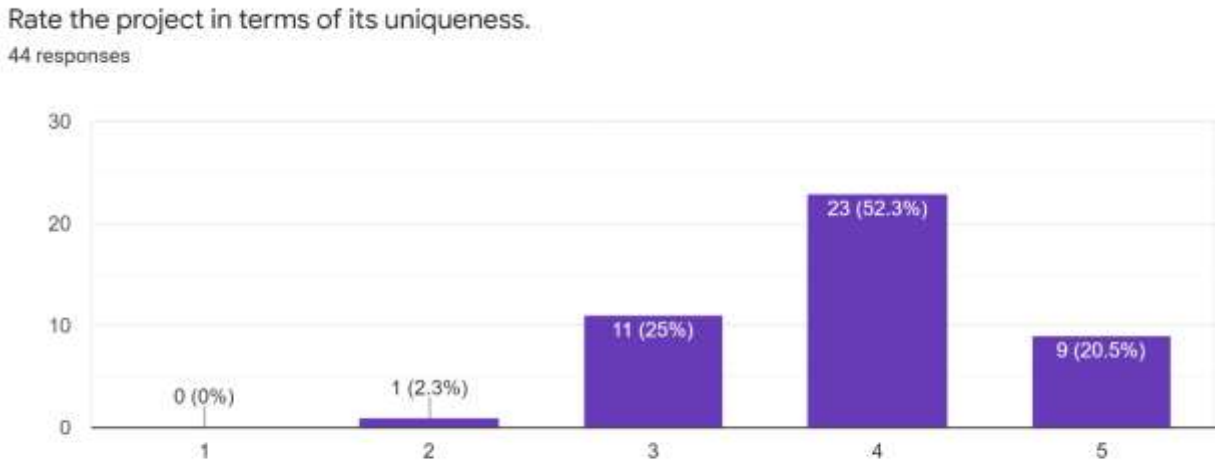


Figure 183 Pre survey result part 6

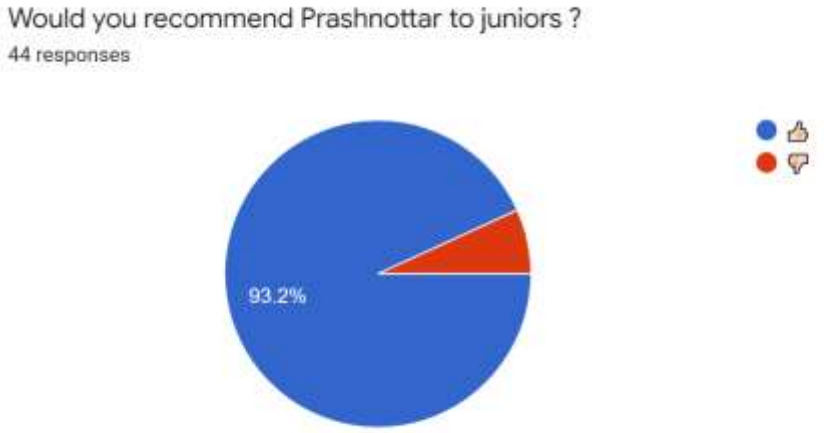


Figure 184 Pre survey result part 7

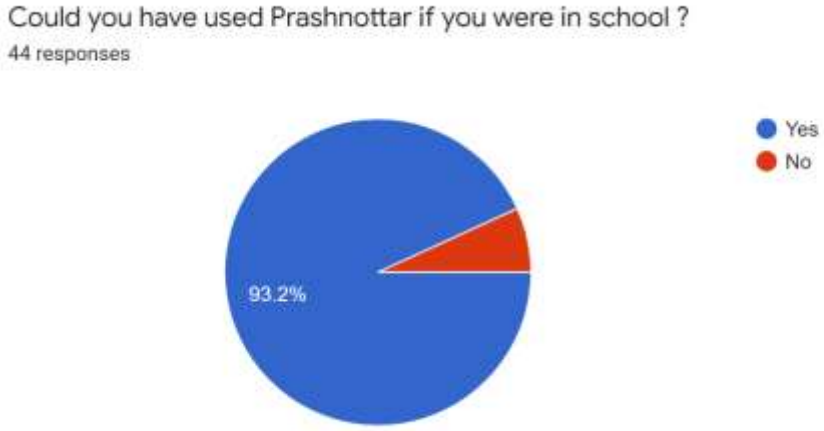


Figure 185 Pre survey result part 8

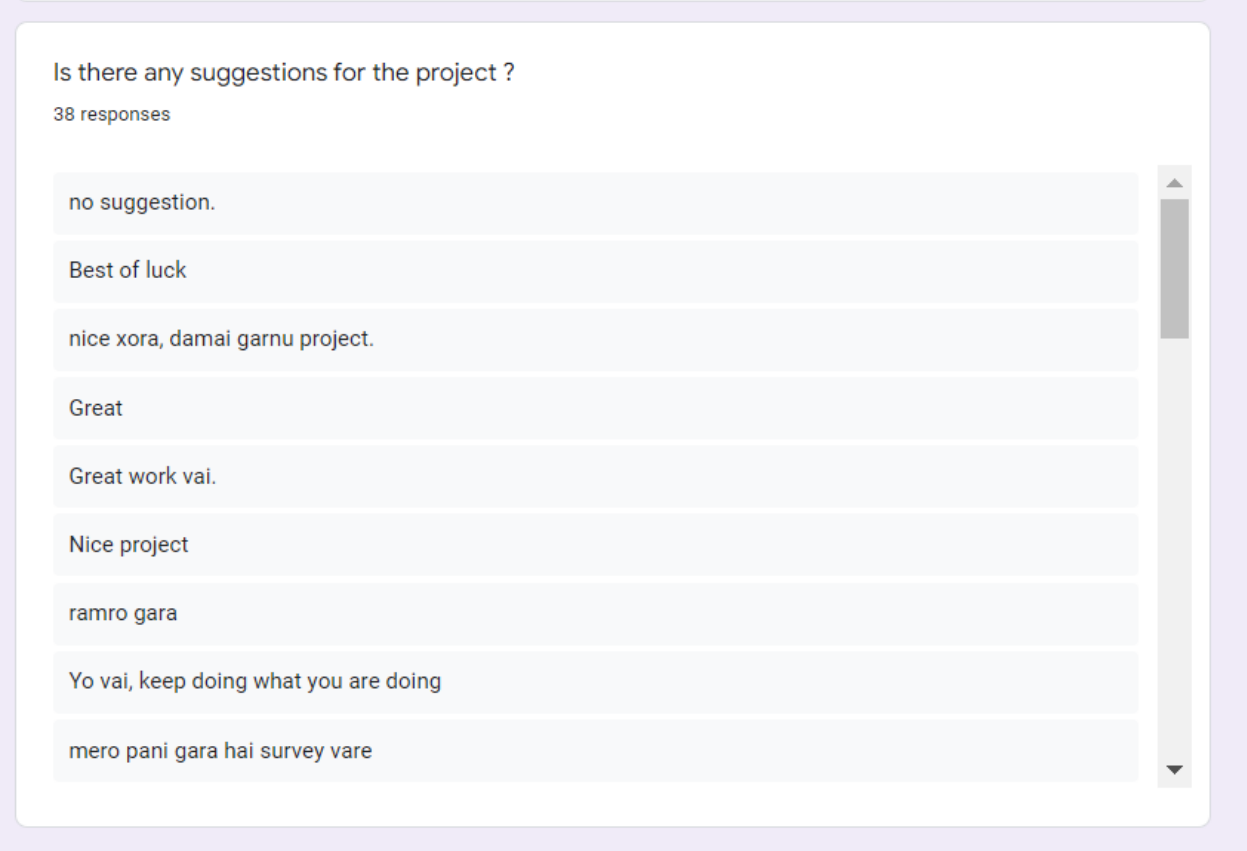


Figure 186 Pre survey result part 9

8.3. Appendix C: Sample Codes

8.3.1. Sample Code Of The UI

8.3.1.1. Sample Code of Home Screen UI

lib > bscreens > home.dart > _HomeState

```dart
import 'package:flutter/material.dart';
import 'package:prashnottar/Screens/class_home/chooseclass.dart';
import 'package:prashnottar/bscreens/post_screen.dart';
import 'package:prashnottar/bscreens/profile.dart';

import '../services/user_service.dart';
import 'login.dart';
import 'post_form.dart';

class Home extends StatefulWidget {
  @override
  _HomeState createState() => _HomeState();
}

class _HomeState extends State<Home> {
  //int currentIndex = 0;
  int _currentIndex = 0;
  final List<Widget> _tablist = [
    PostScreen(),
    ChooseClassPage(),
    Profile(),
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: _tablist.elementAt(_currentIndex),
      appBar: AppBar(
        title: Text('प्रश्नोत्तर'),
        backgroundColor: Color.fromARGB(255, 47, 122, 50),
        actions: [
          IconButton(
```

Figure 187 Sample code of home screen UI

8.3.1.2. Sample Code of Comment Screen UI

```dart
import 'package:flutter/material.dart';
import 'package:prashnottar/models/api_response.dart';
import 'package:prashnottar/models/comment.dart';
import 'package:prashnottar/services/comment_service.dart';
import 'package:prashnottar/services/user_service.dart';

import '../constant.dart';
import 'login.dart';

class CommentScreen extends StatefulWidget {
  final int? postId;

  CommentScreen({this.postId});

  @override
  _CommentScreenState createState() => _CommentScreenState();
}

class _CommentScreenState extends State<CommentScreen> {
  List<dynamic> _commentsList = [];
  bool _loading = true;
  int userId = 0;
  int _editCommentId = 0;
  TextEditingController _txtCommentController = TextEditingController();

  // Get comments
  Future<void> _getComments() async {
    userId = await getUserId();
    ApiResponse response = await getComments(widget.postId ?? 0);

    if (response.error == null) {
      setState(() {
```

Figure 188 Sample code of home comment screen

8.3.1.3. Sample Code of Loading Screen UI

```dart
import 'package:flutter/material.dart';
import 'package:prashnottar/bscreens/home.dart';
import 'package:prashnottar/constant.dart';

import '../models/api_response.dart';
import '../services/user_service.dart';
import 'login.dart';

class Loading extends StatefulWidget {
  @override
  _LoadingState createState() => _LoadingState();
}

class _LoadingState extends State<Loading> {
  void _loadUserInfo() async {
    String token = await getToken();
    if (token == '') {
      Navigator.of(context).pushAndRemoveUntil(
          MaterialPageRoute(builder: (context) => Login()), (route) => f
    } else {
      ApiResponse response = await getUserDetail();
      if (response.error == null) {
        Navigator.of(context).pushAndRemoveUntil(
            MaterialPageRoute(builder: (context) => Home()), (route) =>
      } else if (response.error == unauthorized) {
        Navigator.of(context).pushAndRemoveUntil(
            MaterialPageRoute(builder: (context) => Login()), (route) =>
      } else {
        ScaffoldMessenger.of(context).showSnackBar(SnackBar(
        content: Text('${response.error}'),
        )); // SnackBar
      }
```

Figure 189 Sample code of loading screen

8.3.1.4. Sample Code of Login Page Screen UI

```dart
import 'package:flutter/material.dart';
import 'package:prashnottar/models/api_response.dart';
import 'package:prashnottar/models/user.dart';
import 'package:prashnottar/services/user_service.dart';
import 'package:shared_preferences/shared_preferences.dart';

import '../constant.dart';
import 'home.dart';
import 'register.dart';

class Login extends StatefulWidget {
  @override
  _LoginState createState() => _LoginState();
}

class _LoginState extends State<Login> {
  final GlobalKey<FormState> formkey = GlobalKey<FormState>();
  TextEditingController txtEmail = TextEditingController();
  TextEditingController txtPassword = TextEditingController();
  bool loading = false;

  void _loginUser() async {
    ApiResponse response = await login(txtEmail.text, txtPassword.text);
    if (response.error == null) {
      _saveAndRedirectToHome(response.data as User);
    } else {
      setState(() {
        loading = false;
      });
      ScaffoldMessenger.of(context)
          .showSnackBar(SnackBar(content: Text('${response.error}')));
    }
  }
```

Figure 190 Sample code of login page screen

8.3.1.5. Sample Code of Post_Form Screen UI

```dart
import 'dart:io';

import 'package:flutter/material.dart';
import 'package:image_picker/image_picker.dart';

import '../constant.dart';
import '../models/api_response.dart';
import '../models/post.dart';
import '../services/post_service.dart';
import '../services/user_service.dart';
import 'login.dart';

class PostForm extends StatefulWidget {
  final Post? post;
  final String? title;

  PostForm({this.post, this.title});

  @override
  _PostFormState createState() => _PostFormState();
}

class _PostFormState extends State<PostForm> {
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
  final TextEditingController _txtControllerBody = TextEditingController
  bool _loading = false;
  File? _imageFile;
  final _picker = ImagePicker();

  Future getImage() async {
    final pickedFile = await _picker.pickImage(source: ImageSource.galle
    if (pickedFile != null) {
      setState(() {
```

Figure 191 Sample code of post_form screen UI

8.3.1.6. Sample Code of Post_Screen UI

```dart
import 'package:flutter/material.dart';
import 'package:prashnottar/bscreens/comment_screen.dart';
import 'package:prashnottar/constant.dart';
import 'package:prashnottar/models/api_response.dart';

import '../models/post.dart';
import '../services/post_service.dart';
import '../services/user_service.dart';
import 'login.dart';
import 'post_form.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';

class PostScreen extends StatefulWidget {
  @override
  _PostScreenState createState() => _PostScreenState();
}

class _PostScreenState extends State<PostScreen> {
  List<dynamic> _postList = [];
  int userId = 0;
  bool _loading = true;

  // get all posts
  Future<void> retrievePosts() async {
    userId = await getUserId();
    ApiResponse response = await getPosts();

    if (response.error == null) {
      setState(() {
        _postList = response.data as List<dynamic>;
        _loading = _loading ? !_loading : _loading;
      });
    } else if (response.error == unauthorized) {
```

Figure 192 Sample code of post Screen UI

8.3.1.7. Sample Code of Profile Screen UI

```dart
import 'dart:io';

import 'package:flutter/material.dart';
import 'package:image_picker/image_picker.dart';
import 'package:prashnottar/models/api_response.dart';
import 'package:prashnottar/models/user.dart';

import '../constant.dart';
import '../services/user_service.dart';
import 'login.dart';

class Profile extends StatefulWidget {
  @override
  _ProfileState createState() => _ProfileState();
}

class _ProfileState extends State<Profile> {
  User? user;
  bool loading = true;
  GlobalKey<FormState> formKey = GlobalKey<FormState>();
  File? _imageFile;
  final _picker = ImagePicker();
  TextEditingController txtNameController = TextEditingController();

  Future getImage() async {
    final pickedFile = await _picker.getImage(source: ImageSource.galler
    if (pickedFile != null) {
      setState(() {
        _imageFile = File(pickedFile.path);
      });
    }
  }
```

Figure 193 Sample code of Profile Screen UI

8.3.1.8. Sample Code of Register User UI

lib > bscreens > register.dart > ...

```dart
import 'package:flutter/material.dart';
import 'package:prashnottar/models/api_response.dart';
import 'package:prashnottar/models/user.dart';
import 'package:shared_preferences/shared_preferences.dart';

import '../constant.dart';
import '../services/user_service.dart';
import 'home.dart';
import 'login.dart';

class Register extends StatefulWidget {
  @override
  _RegisterState createState() => _RegisterState();
}

class _RegisterState extends State<Register> {
  GlobalKey<FormState> formKey = GlobalKey<FormState>();
  bool loading = false;
  TextEditingController nameController = TextEditingController(),
      emailController = TextEditingController(),
      passwordController = TextEditingController(),
      passwordConfirmController = TextEditingController();

  void _registerUser() async {
    ApiResponse response = await register(
        nameController.text, emailController.text, passwordController.t
    if (response.error == null) {
      _saveAndRedirectToHome(response.data as User);
    } else {
      setState(() {
        loading = !loading;
      });
```

Figure 194 Sample code of Register User UI

8.3.1.9. Sample Code of Choose Class UI

```dart
import 'package:flutter/material.dart';
import 'package:prashnottar/Screens/IconManagement/iconcontent.dart';
import 'package:prashnottar/Screens/class_home/choosesubject.dart';
import 'package:prashnottar/Screens/class_home/makenavbar.dart';
import '                                                    rade.dart';

class Ch
  const ChooseClassPage({Key? key}) : super(key: key);
  @override
  State<ChooseClassPage> createState() => _ChooseClassState();
}

class _ChooseClassState extends State<ChooseClassPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Color.fromARGB(220, 206, 206, 206),
      body: SingleChildScrollView(
        child: Center(
          child: Padding(
            padding: const EdgeInsets.fromLTRB(0.0, 40, 0.0, 0.0),
            child:
              Column(mainAxisAlignment: MainAxisAlignment.start, child
                ReusableClass(
                  onPress: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(
                        builder: (context) {
                          return ChooseSubject();
                        },
                      ), // MaterialPageRoute
```

ChooseClassPage ChooseClassPage({Key? key})

package:prashnottar/Screens/class_home/chooseclass.dart

Figure 195 Sample code of Choose Class UI

8.3.1.10. Sample Code of Choose Subject UI

```dart
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:prashnottar/Screens/Chapters/account_8.dart';
import 'package:prashnottar/Screens/Chapters/computer_8.dart';
import 'package:prashnottar/Screens/Chapters/ehp_8.dart';
import 'package:prashnottar/Screens/Chapters/math_8.dart';
import 'package:prashnottar/Screens/Chapters/optionalmath_8.dart';
import 'package:prashnottar/Screens/Chapters/science_8.dart';
import 'package:prashnottar/Screens/class_home/reusablesubject.dart';
import 'package:prashnottar/constraints.dart';

class ChooseSubject extends StatefulWidget {
  const ChooseSubject({Key? key}) : super(key: key);

  @override
  State<ChooseSubject> createState() => _ChooseSubjectState();
}

class _ChooseSubjectState extends State<ChooseSubject> {
  @override
  Widget build(BuildContext context) {
    // Size size = MediaQuery.of(context).size;
    return Scaffold(
      backgroundColor: Color.fromARGB(220, 206, 206, 206),
      body: Padding(
        padding: const EdgeInsets.fromLTRB(8.0, 12, 5.0, 8.0),
        child: SingleChildScrollView(
          child: Column(
            // mainAxisAlignment: MainAxisAlignment.start,
            children: [
              Align(
                alignment: Alignment.topLeft,
                child: Text(
```

Figure 196 Sample code of Choose Subject UI

8.3.1.11. Sample Code of Choose Chapter(Science) UI

```dart
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:prashnottar/Screens/Social_APP/Class_8_Science/measureme
import 'package:prashnottar/Screens/class_home/reusablechapterrow.dart';
import 'package:prashnottar/bscreens/home.dart';
import 'package:prashnottar/bscreens/loading.dart';
import 'package:prashnottar/bscreens/post_screen.dart';
import 'package:prashnottar/constraints.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';

class Science_8 extends StatefulWidget {
  const Science_8({Key? key}) : super(key: key);

  @override
  State<Science_8> createState() => _Science_8State();
}

class _Science_8State extends State<Science_8> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Color.fromARGB(220, 206, 206, 206),
      body: Padding(
        padding: const EdgeInsets.fromLTRB(8.0, 12, 5.0, 8.0),
        child: SingleChildScrollView(
          child: Column(children: [
            Align(
              alignment: Alignment.topLeft,
              child: Text(
                "Science",
                style: GoogleFonts.poppins(textStyle: KHeadingTextstyle
              ), // Text
```

Figure 197 Sample code of choose chapter

8.3.2. Sample Code For The Automation Script

8.3.2.1. Sample Code of AnswerController

app > Http > Controllers > AuthController.php

```php
<?php

namespace App\Http\Controllers;

use App\Models\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class AuthController extends Controller
{
    //Register user
    public function register(Request $request)
    {
        // validate fields
        $attrs = $request->validate([
            'name' => 'required|string',
            'email' => 'required|email|unique:users,email',
            'password' => 'required|min:6|confirmed'
        ]);

        //create user

        $user = User::create([
            'name' => $attrs['name'],
            'email' => $attrs['email'],
            'password' => bcrypt($attrs['password'])
        ]);

        // return user & token in response

        return response([
            'user' => $user,
            'token' => $user->createToken('secret')->plainTextT
```

Figure 198 Sample code of AnswerController.php

8.3.2.1. Sample Code of AuthController

app > Http > Controllers > AuthController.php

```php
<?php

namespace App\Http\Controllers;

use App\Models\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class AuthController extends Controller
{
    //Register user
    public function register(Request $request)
    {
        // validate fields
        $attrs = $request->validate([
            'name' => 'required|string',
            'email' => 'required|email|unique:users,email',
            'password' => 'required|min:6|confirmed'
        ]);

        //create user

        $user = User::create([
            'name' => $attrs['name'],
            'email' => $attrs['email'],
            'password' => bcrypt($attrs['password'])
        ]);

        // return user & token in response

        return response([
            'user' => $user,
            'token' => $user->createToken('secret')->plainTextT
```

Figure 199 Sample code of Auth Controller

8.3.2.1. Sample Code of CommentController

```php
<?php

namespace App\Http\Controllers;

use App\Models\Comment;
use App\Models\Post;
use Illuminate\Http\Request;

class CommentController extends Controller
{
    // get all comments of a post
    public function index($id)
    {
        $post = Post::find($id);
        if (!$post) {
            return response([
                'message' => 'Post not found.',
            ], 403);
        }

        return response([

            'comments' => $post->comments()->with('user:id,name
        ], 200);
    }

    //create a comment
    public function store(Request $request, $id)
    {
        $post = Post::find($id);
        if (!$post) {
            return response([
                'message' => 'Post not found.',
```

Figure 200 Sample code of CommentController

8.3.2.1. Sample Code of Controller

app > Http > Controllers > Controller.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
use Illuminate\Foundation\Bus\DispatchesJobs;
use Illuminate\Foundation\Validation\ValidatesRequests;
use Illuminate\Routing\Controller as BaseController;
use Illuminate\Support\Facades\URL;

class Controller extends BaseController
{
    use AuthorizesRequests, DispatchesJobs, ValidatesRequests;

    public function saveImage($image, $path = 'public')
    {
        if (!$image) {
            return null;

        }

        $filename = time() . '.png';
        //save image
        \Storage::disk($path)->put($filename, base64_decode($im

        // return the path
        //URL is the base url exp: localhost: 8000
        return URL::to('/') . '/storage/' . $path . '/' . $file
    }
}

```

Figure 201 Sample code of Controllers

8.3.2.1. Sample Code of LikeController

app > Http > Controllers > LikeController.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Post;
use App\Models\Like;

class LikeController extends Controller
{
    //like or unlike

    public function likeOrUnlike($id)
    {
        $post = Post::find($id);
        if (!$post) {
            return response([
                'message' => 'Post not found.',
            ], 403);
        }

        $like = $post->likes()->where('user_id',auth()->user()-

        // if not liked then like
        if(!$like)
        {
            Like::create([
                'post_id' => $id,
                'user_id' => auth()->user()->id
            ]);

            return response([
                'message' => 'Liked.',
```

Figure 202 Sample code of Like Controller

8.3.2.1. Sample Code of PostController

app > Http > Controllers > PostController.php

```php
<?php

namespace App\Http\Controllers;

use App\Models\Post;
use App\Models\User;
use Illuminate\Http\Request;

class PostController extends Controller
{
    //get all posts

    public function index()
    {
        return response([
            'posts' => Post::orderBy('created_at', 'desc')->wit
                ->with('likes', function ($like) {
                    return $like->where('user_id', auth()->user
                        ->select('id', 'user_id', 'post_id')->g
                })
                ->get(),
        ], 200);
    }
// get single post

    public function show($id)
    {
        return response([
            'post' => Post::where('id', $id)->withCount('commen
        ], 200);
    }

// create a post
```

Figure 203 Sample code of Post controller

8.3.2.1. Sample Code of Answer.php

app > Models > Answer.php

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Answer extends Model
{
    use HasFactory;

    protected $fillable = [
        'answer',
        'user_id',
        'post_id'
    ];

    public function user(){
        return $this->belongsTo(User::class);
    }
}

```

Figure 204 Sample code of Answer.php

8.3.2.1. Sample Code of Comment.php

app > Models > Comment.php

```php
<?php

namespace App\Models;

use App\Models\User;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Comment extends Model
{
    use HasFactory;

    protected $fillable = [
        'comment',
        'user_id',
        'post_id',
    ];

    public function user()
    {
        return $this->belongsTo(User::class);
    }
}
```

Figure 205 Sample code of comment.php

8.3.2.1. Sample Code of Like.php

PostController.php | Answer.php | Comment.php | Like.php

app > Models > Like.php

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Like extends Model
{
    use HasFactory;

    protected $fillable = [
        'user_id',
        'post_id'
    ];
}
```

Figure 206 Sample code of like.php

8.3.2.1. Sample Code of Post.php

app > Models > Post.php

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use App\Models\User;
use App\Models\Comment;
use App\Models\Answer;
use App\Models\Like;


class Post extends Model
{
    use HasFactory;

    protected $fillable = [
        'body',
        'user_id',
        'image'
    ];


    public function user(){
        return $this->belongsTo(User::class);
    }

    public function comments()
    {
        return $this->hasMany(Comment::class);
    }

    public function answers()
```

Figure 207 Sample code of Post.php

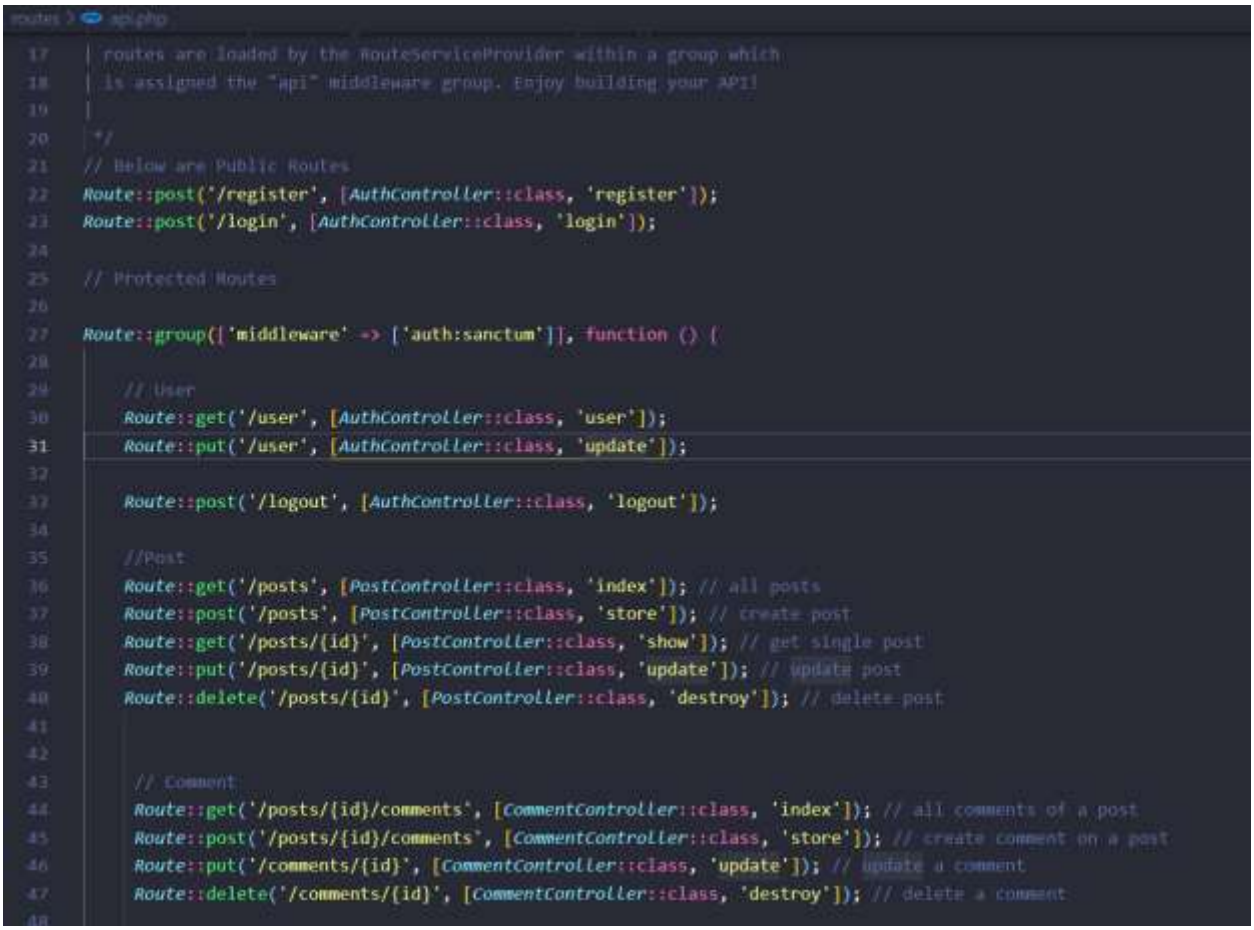
8.3.2.1. Sample Code of User.php

app > Models > User.php

```php
<?php

namespace App\Models;

use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'email',
        'image',
        'password',
    ];

    /**
     * The attributes that should be hidden for serialization
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password'
```

Figure 208 Sample code of user.php

8.3.2.1. Sample Code of api.php



```
routes > api.php

 |  routes are loaded by the RouteServiceProvider within a group which
 |  is assigned the "api" middleware group. Enjoy building your API!
 |
 */
// Below are Public Routes
Route::post('/register', [AuthController::class, 'register']);
Route::post('/login', [AuthController::class, 'login']);

// Protected Routes

Route::group(['middleware' => ['auth:sanctum']], function () {

    // User
    Route::get('/user', [AuthController::class, 'user']);
    Route::put('/user', [AuthController::class, 'update']);

    Route::post('/logout', [AuthController::class, 'logout']);

    //Post
    Route::get('/posts', [PostController::class, 'index']); // all posts
    Route::post('/posts', [PostController::class, 'store']); // create post
    Route::get('/posts/{id}', [PostController::class, 'show']); // get single post
    Route::put('/posts/{id}', [PostController::class, 'update']); // update post
    Route::delete('/posts/{id}', [PostController::class, 'destroy']); // delete post


     // Comment
     Route::get('/posts/{id}/comments', [CommentController::class, 'index']); // all comments of a post
     Route::post('/posts/{id}/comments', [CommentController::class, 'store']); // create comment on a post
     Route::put('/comments/{id}', [CommentController::class, 'update']); // update a comment
     Route::delete('/comments/{id}', [CommentController::class, 'destroy']); // delete a comment
```

Figure 209 Sample code of api.php

8.4. Appendix D: Designs

8.4.1. Gantt Chart

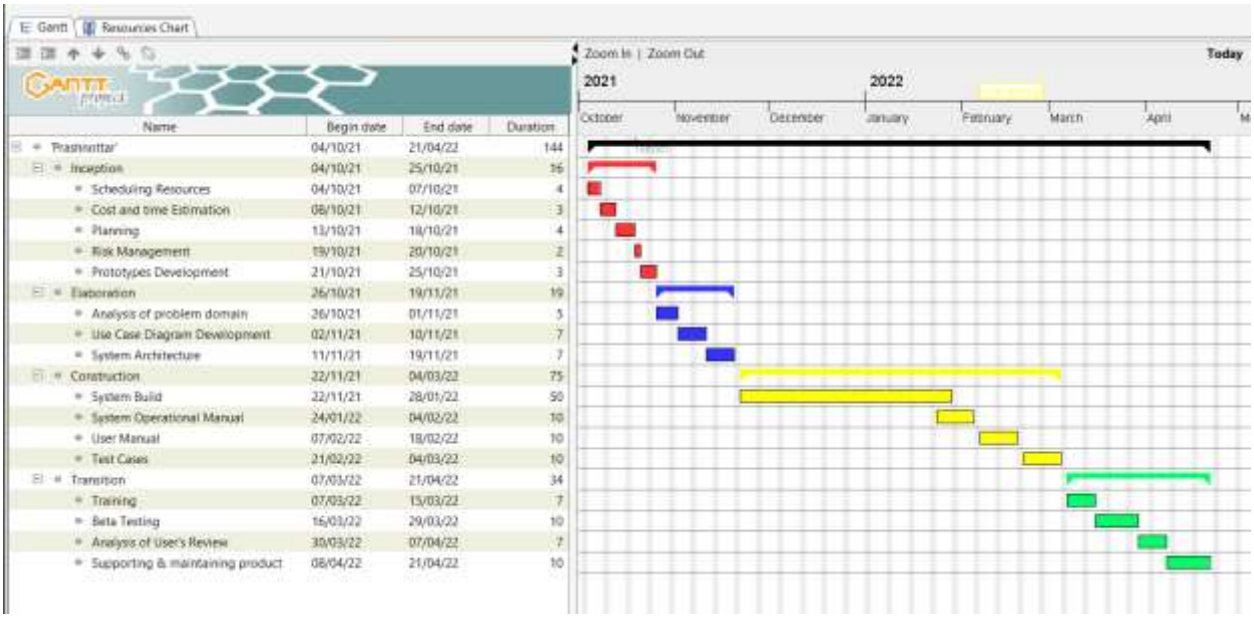


Figure 210 Gant chart

8.4.2. Work BreakDown Structure

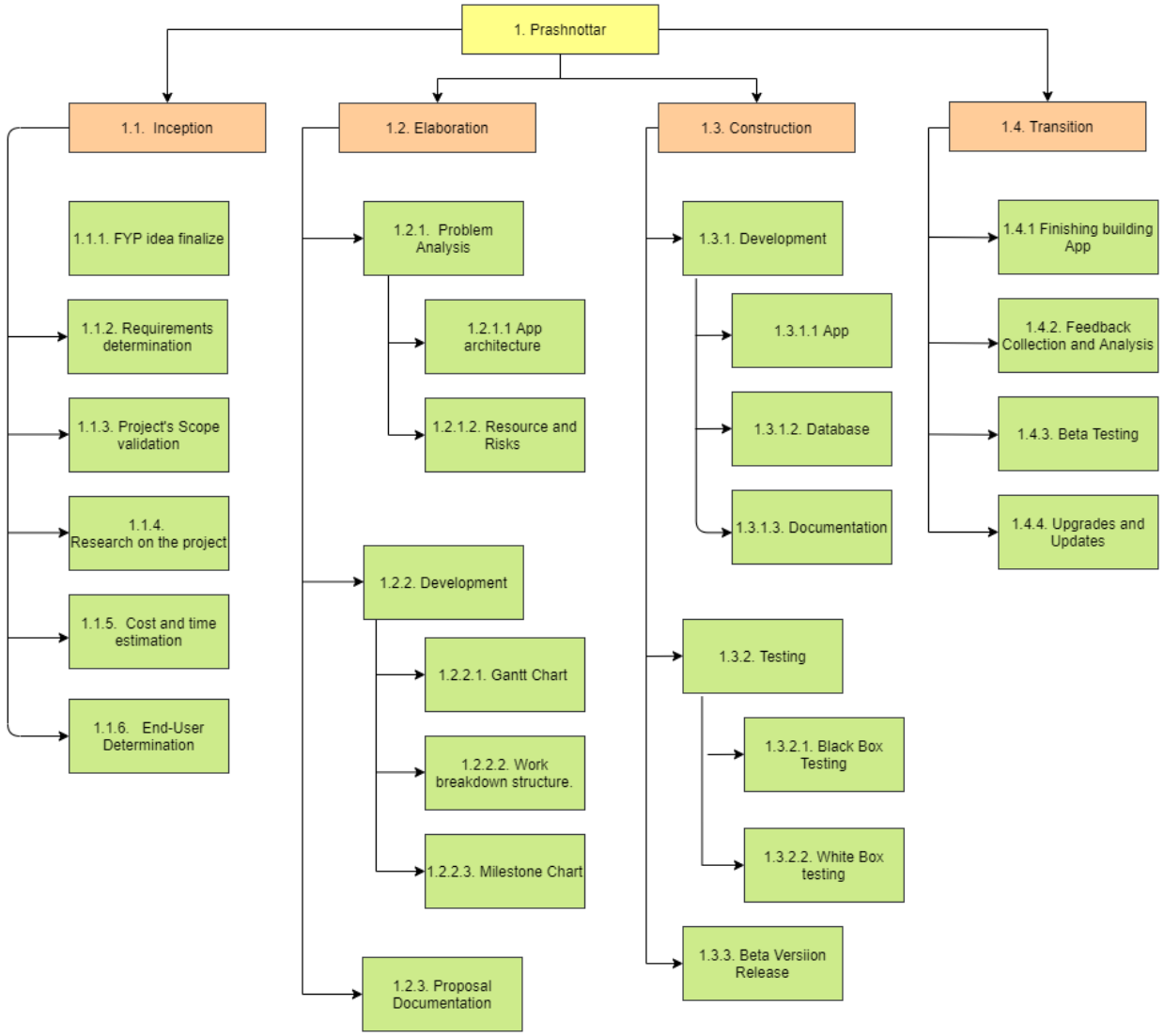


Figure 211 Work break down structure.

8.4.3. MileStone Chart

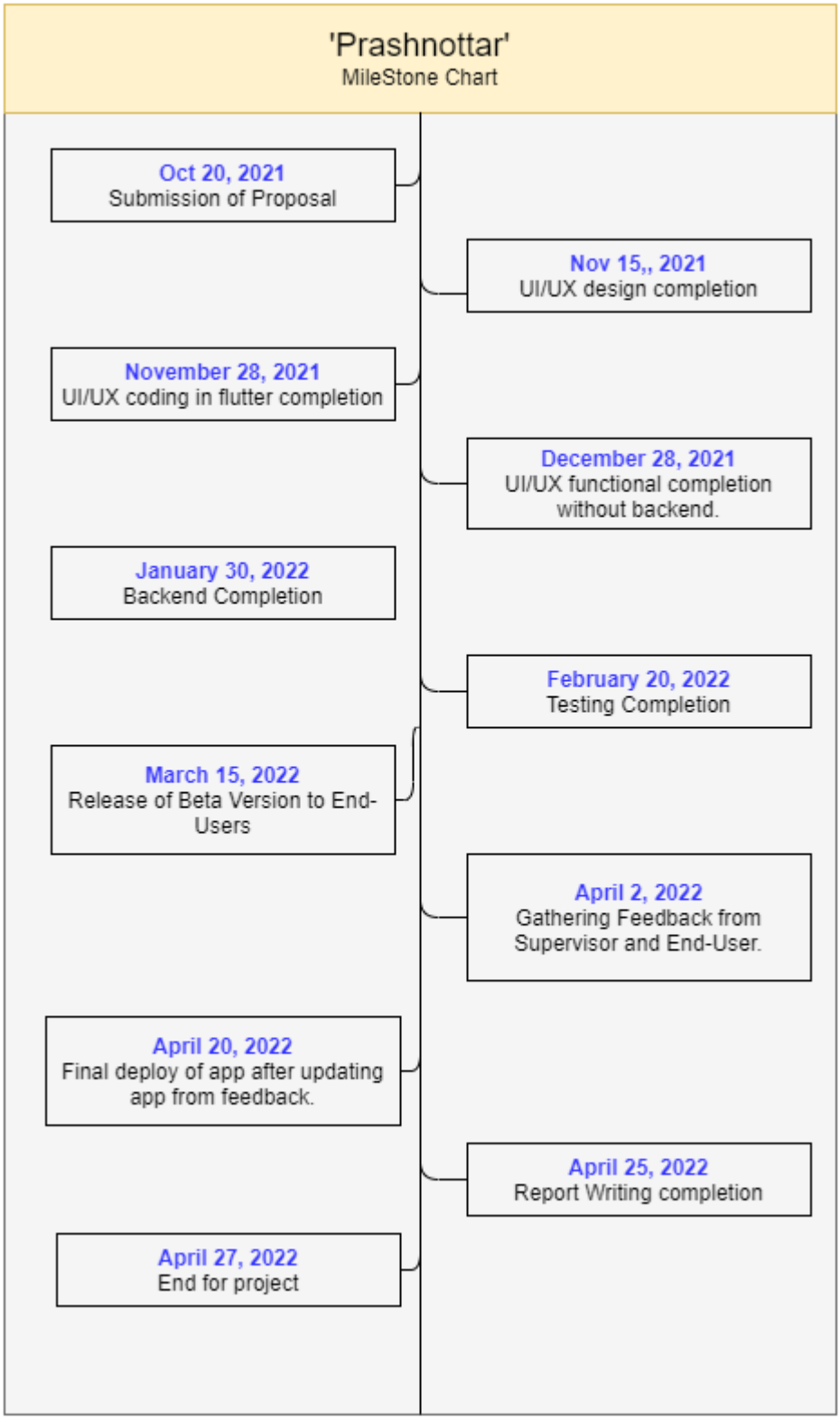


Figure 212 Milestone chart

8.3. Use Case

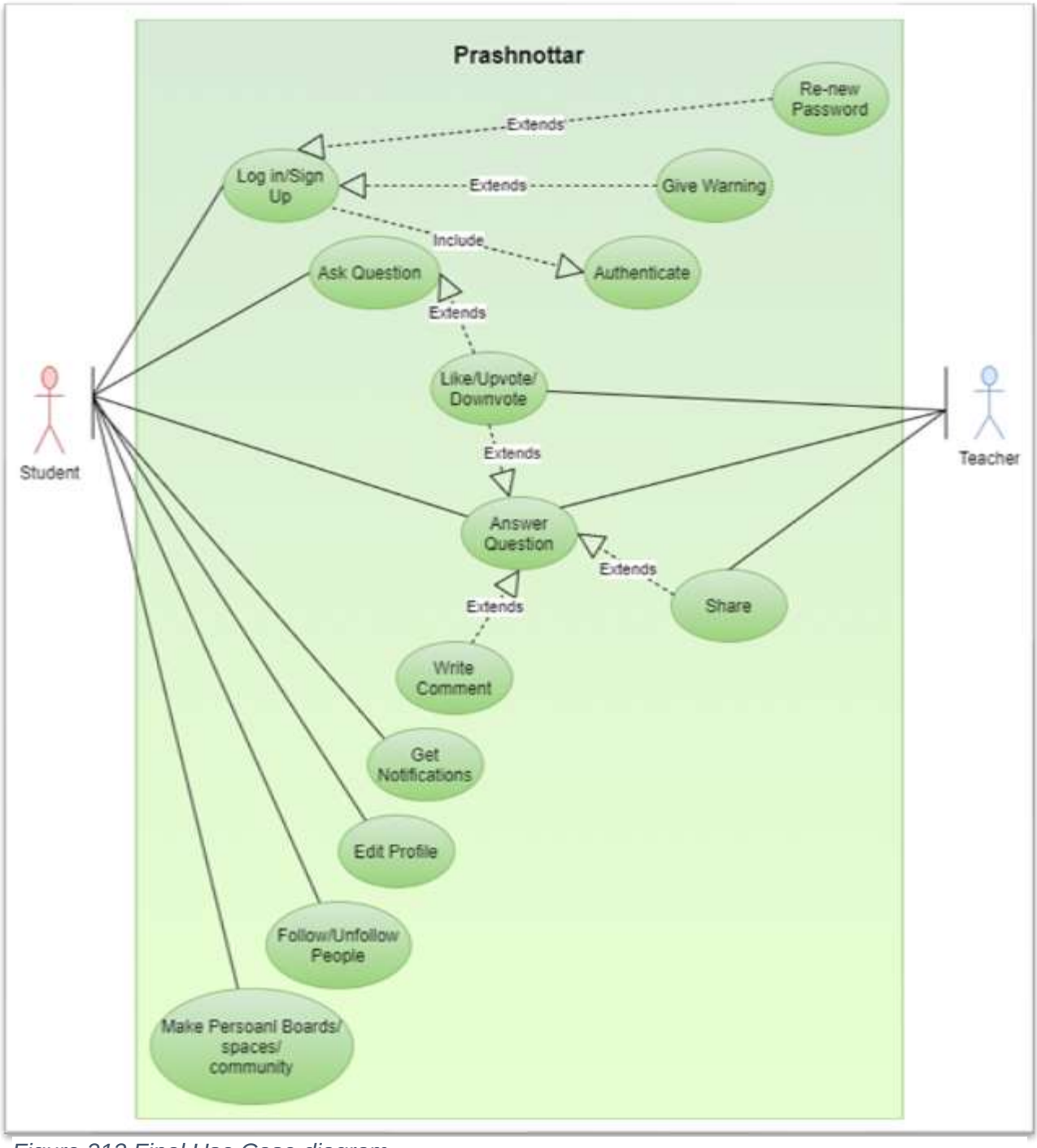


Figure 213 Final Use Case diagram

8.4. Wireframe

Figure 214 Log in wireframe

Figure 215 welcome screen wireframe

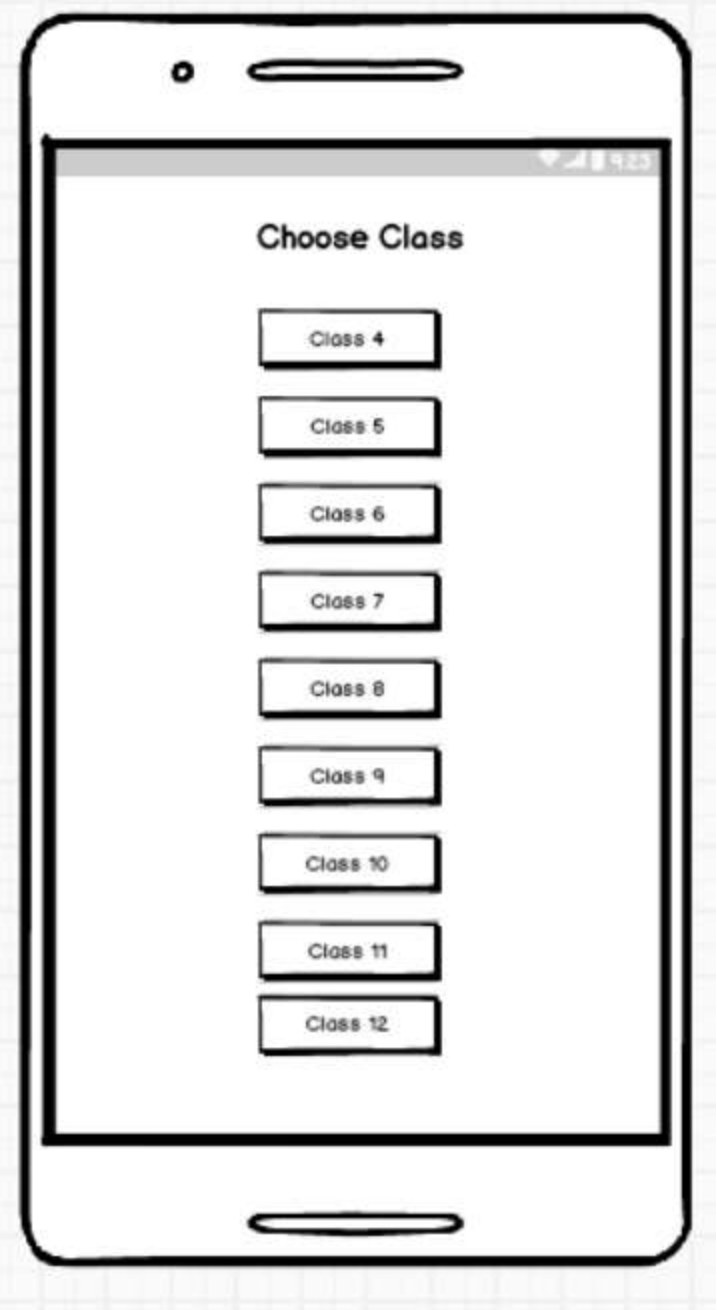


Figure 216 choose class wireframe

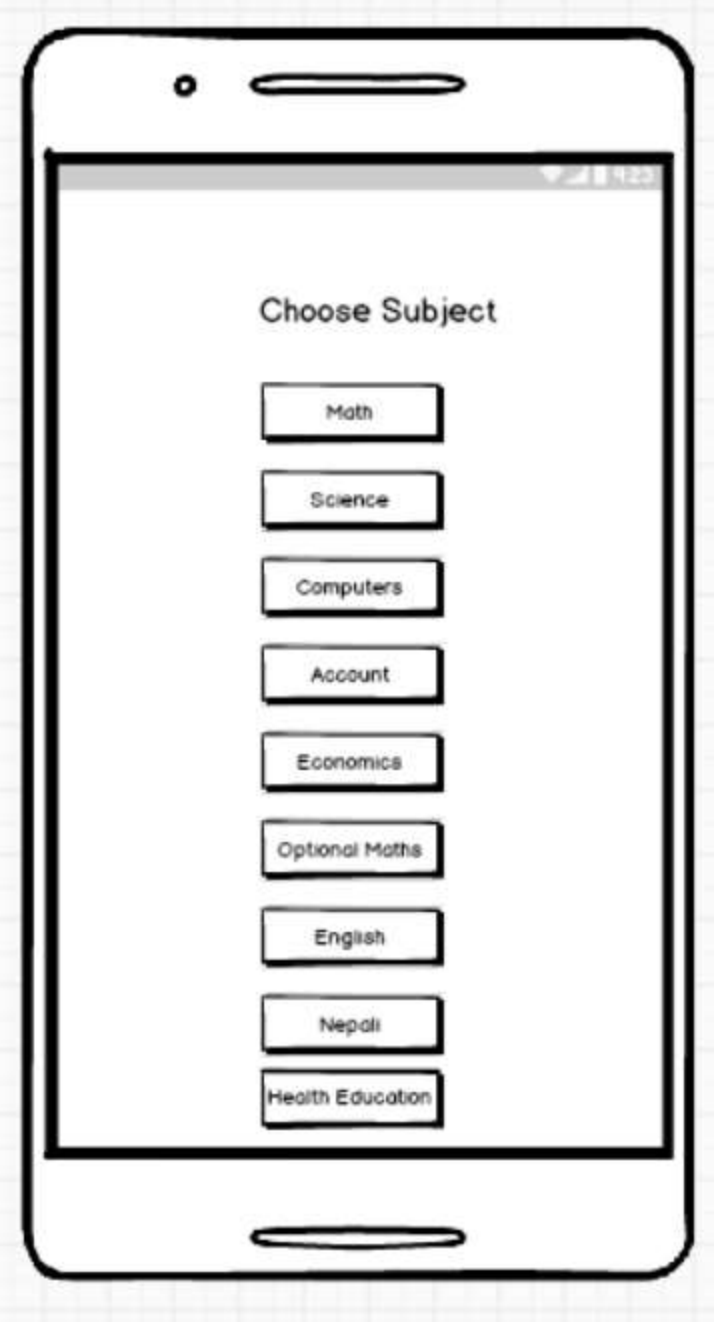


Figure 217 choose subject wireframe

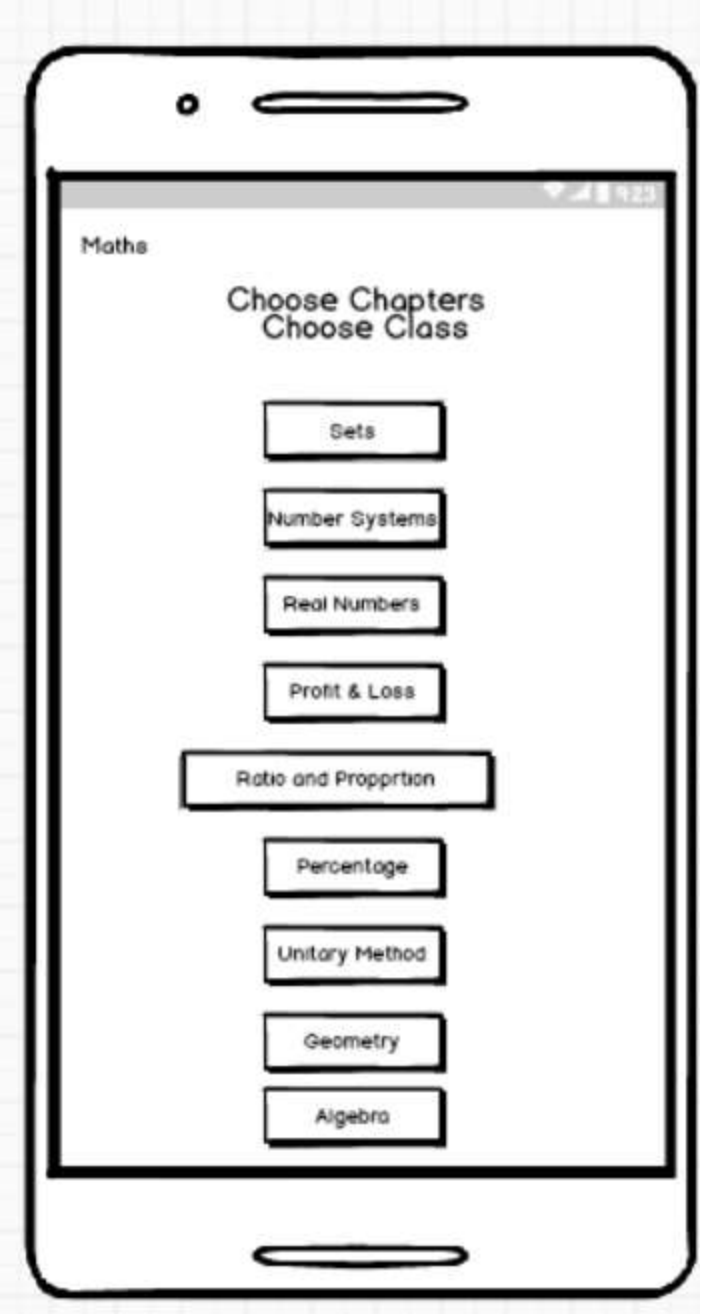


Figure 218 Choose chapter wireframe

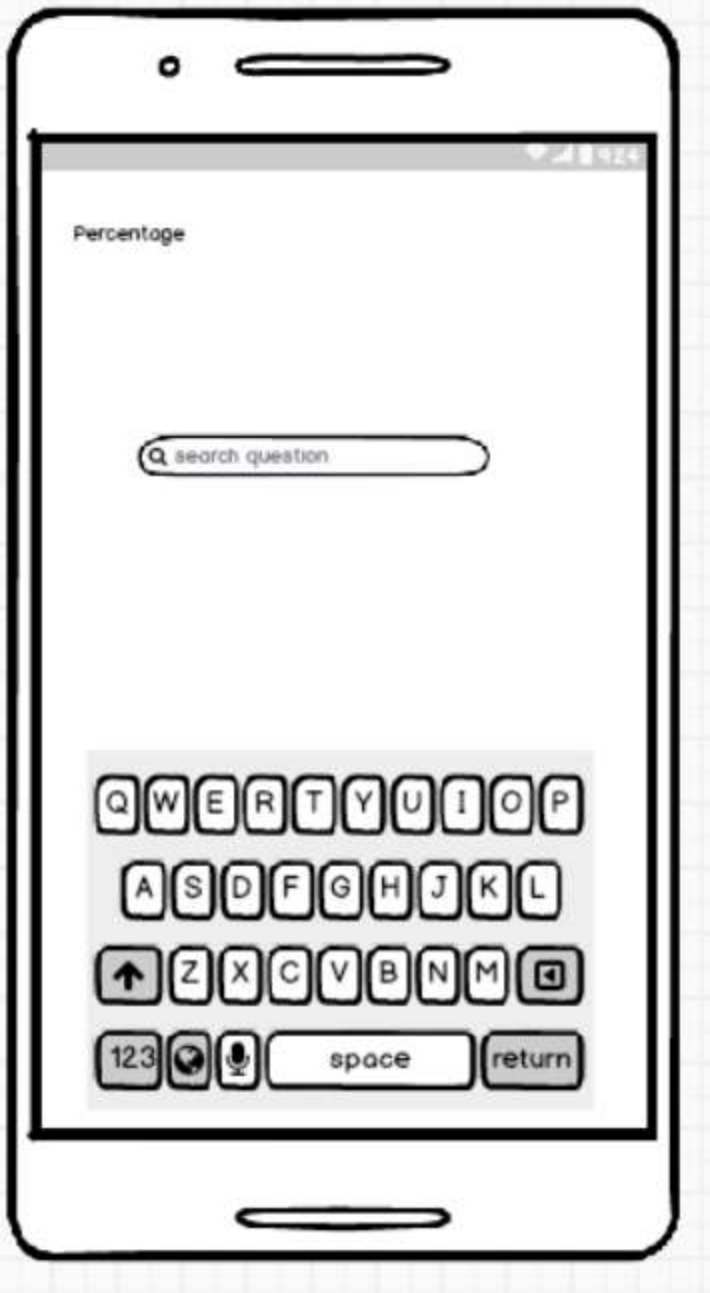


Figure 219 Search wireframe

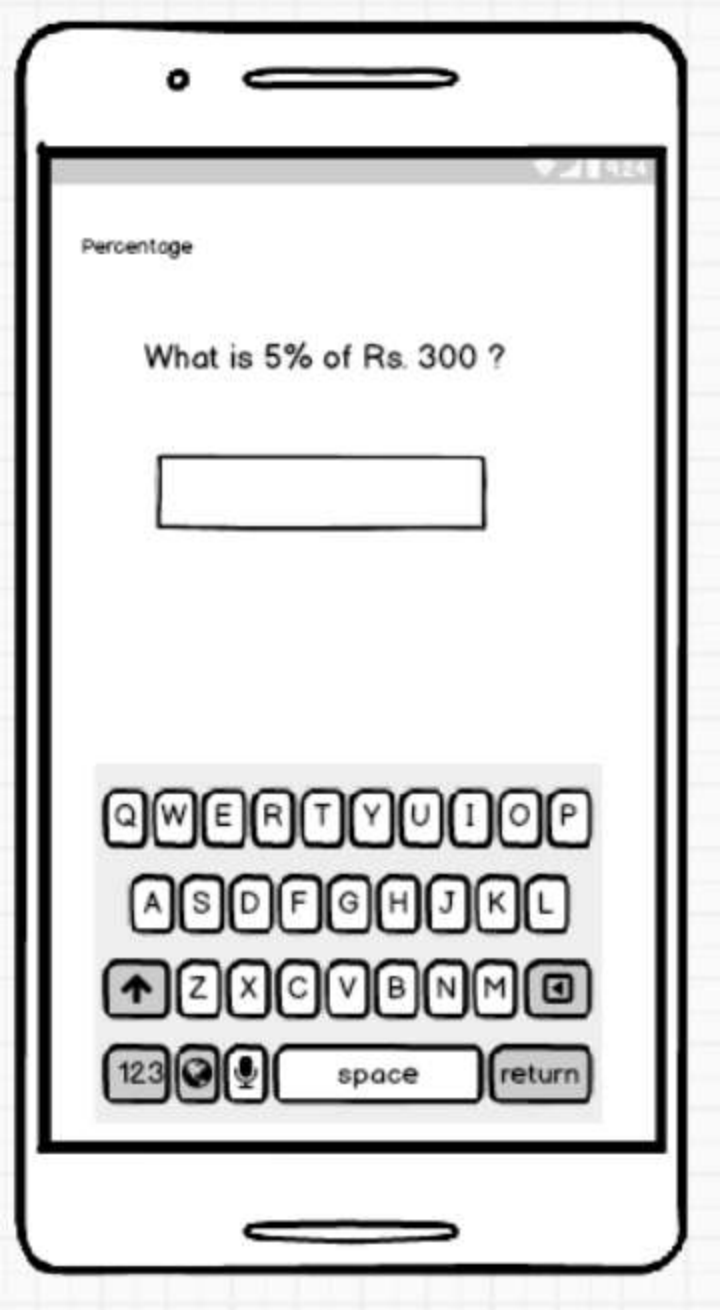


Figure 220 ask question wireframe

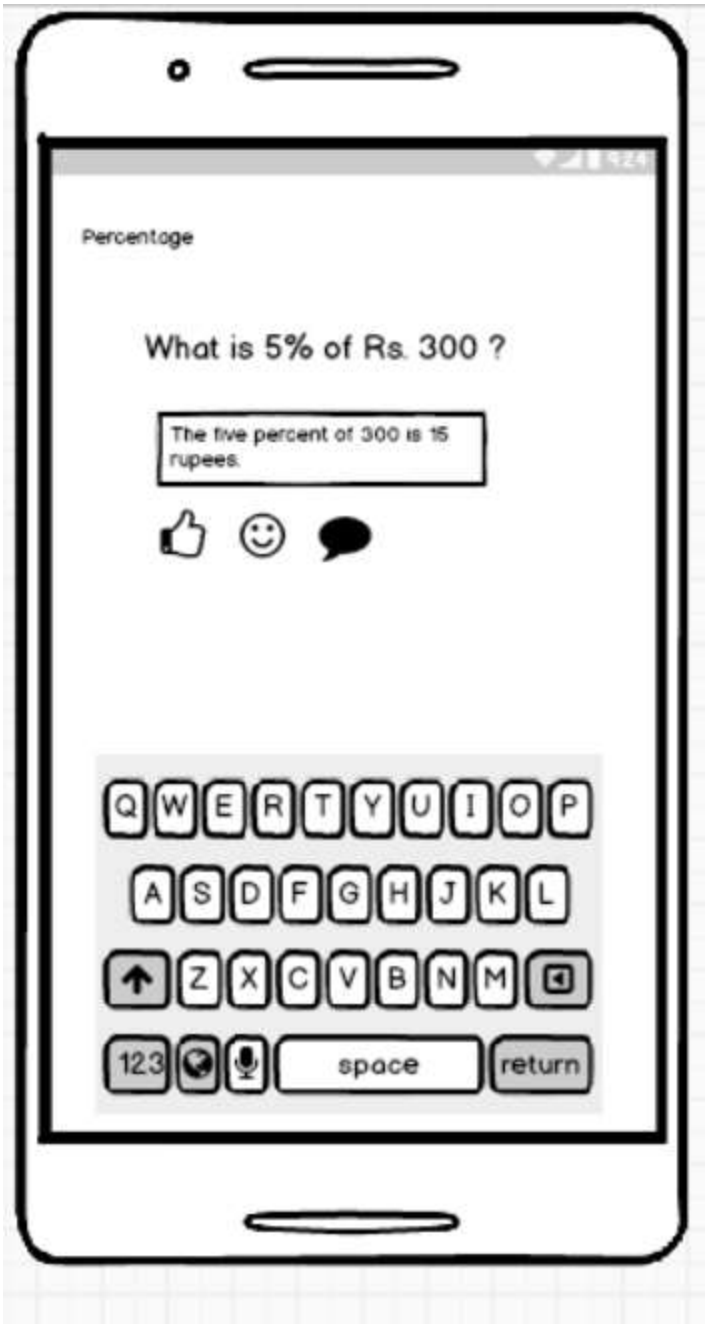


Figure 221 Answer wireframe

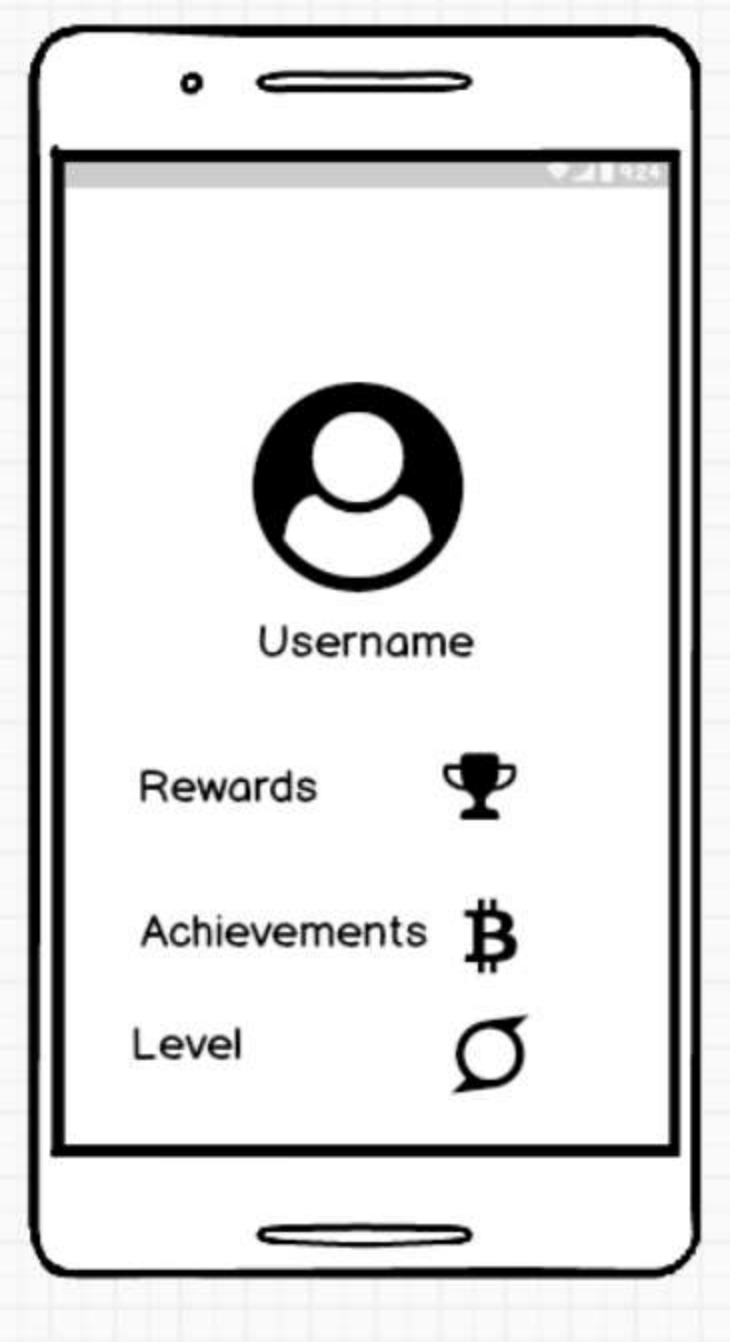


Figure 222 User profile wireframe

8.5. Appendix E: ScreenShots Of the System

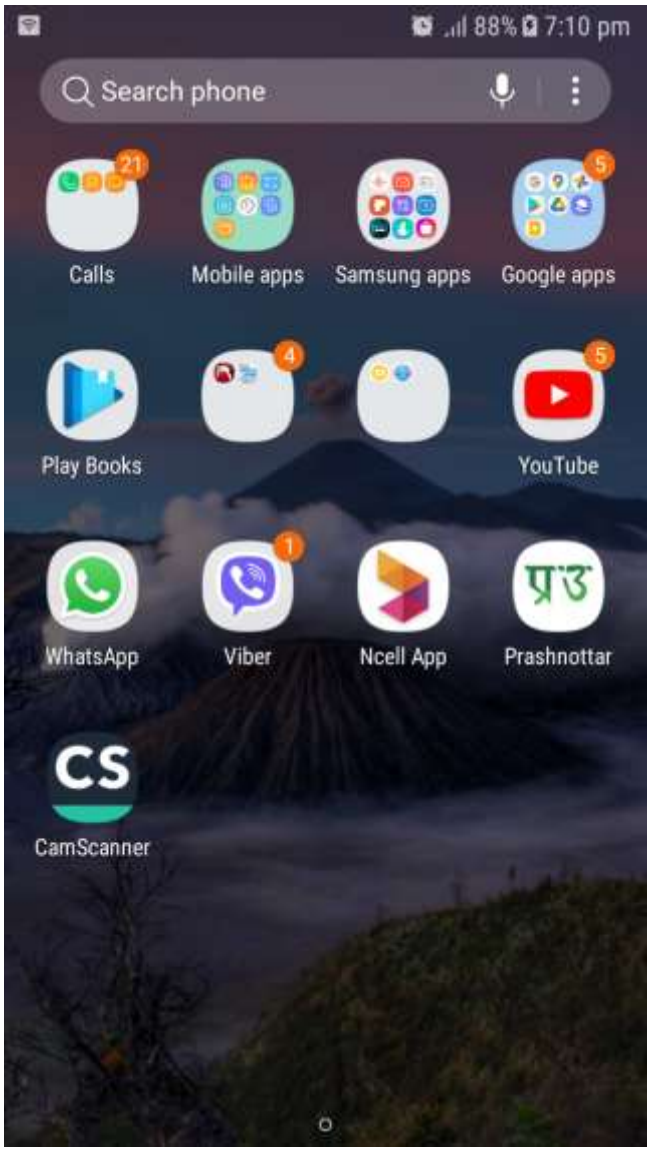


Figure 223 Prashnottar app in the device.

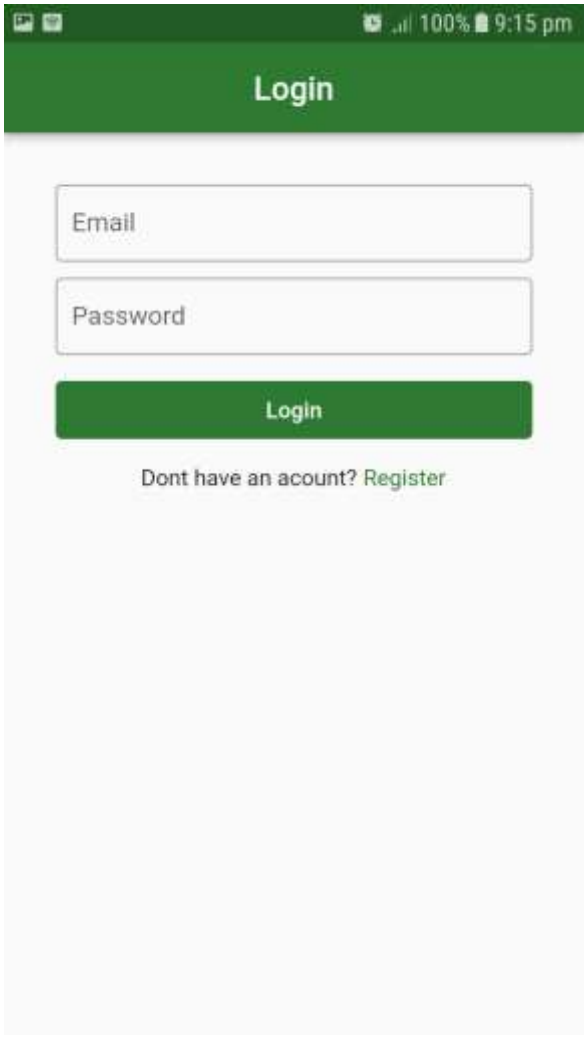


Figure 224 login UI

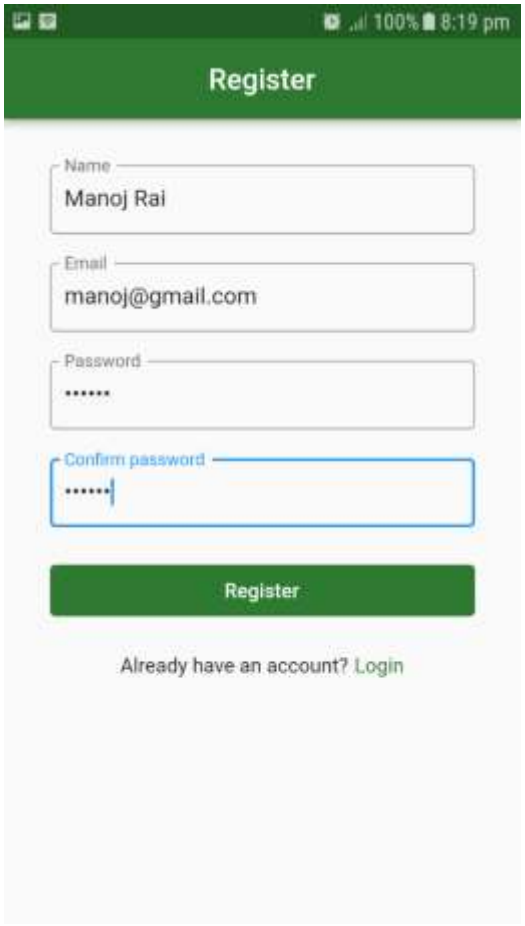


Figure 225 SS of register

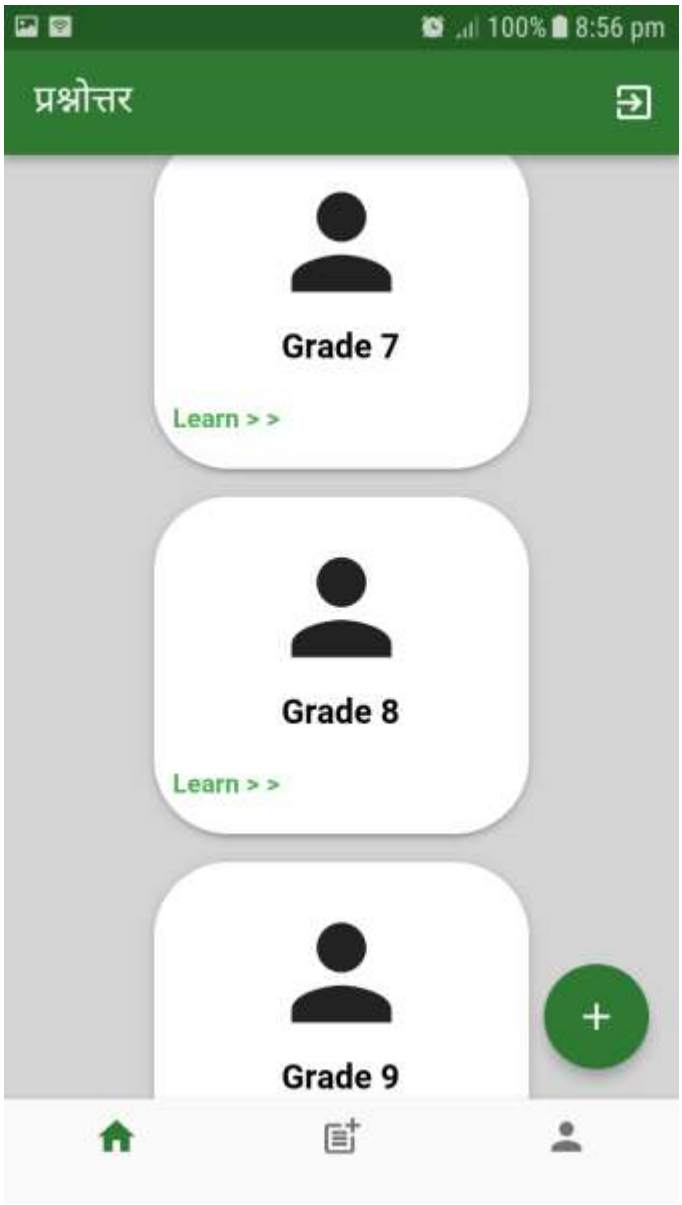


Figure 226 Choose class SS fo the system

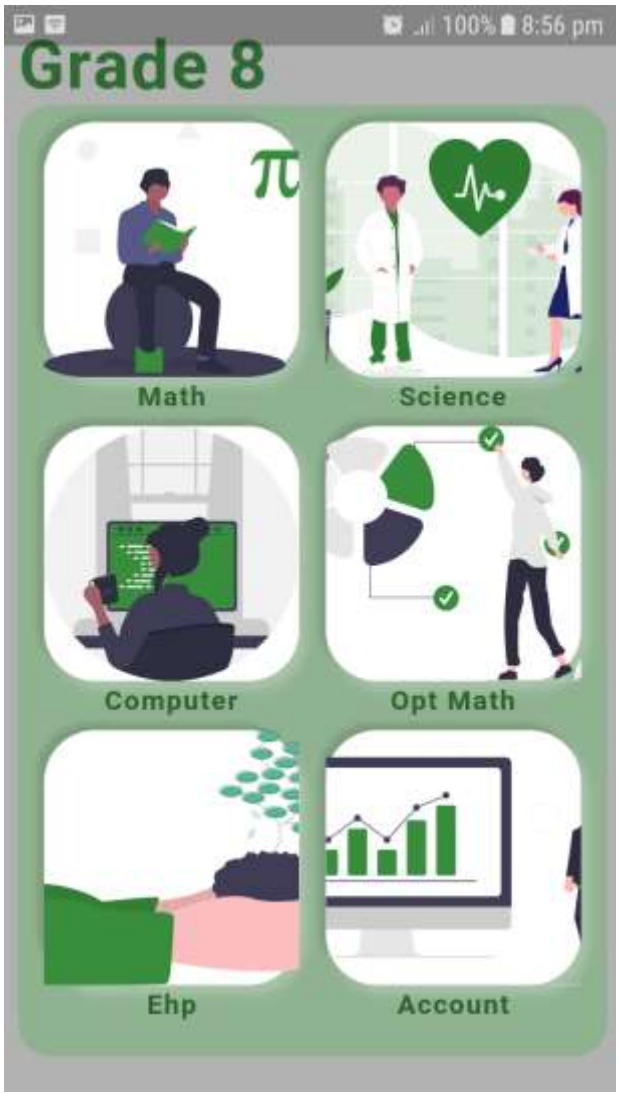


Figure 227 choose subject ss of the system.

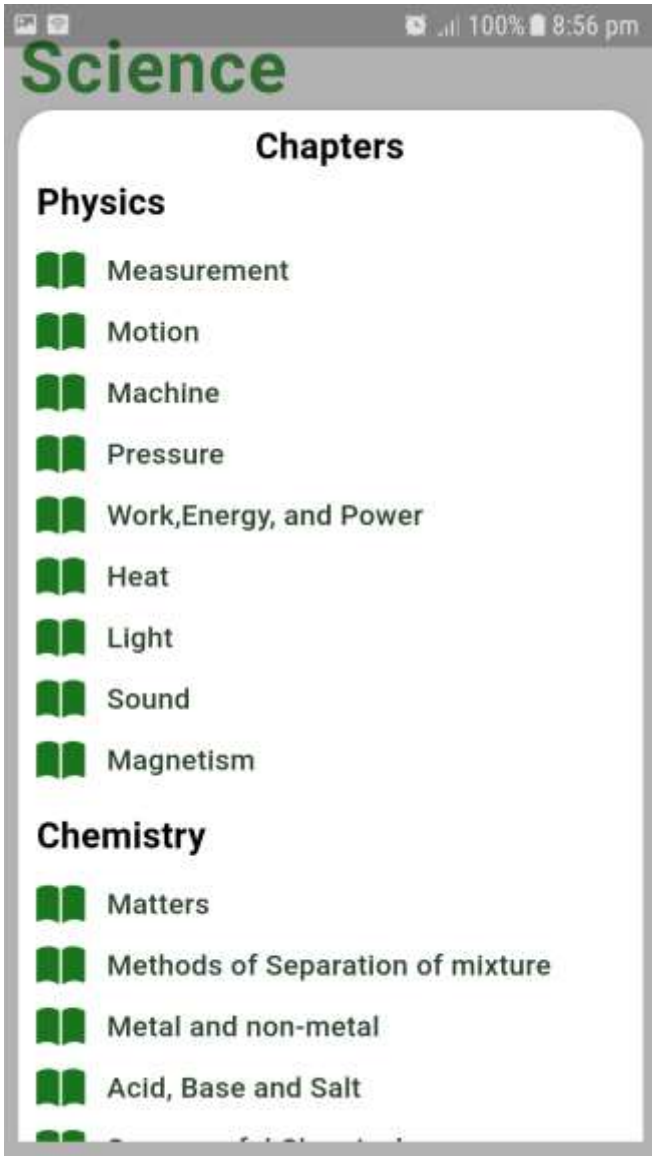
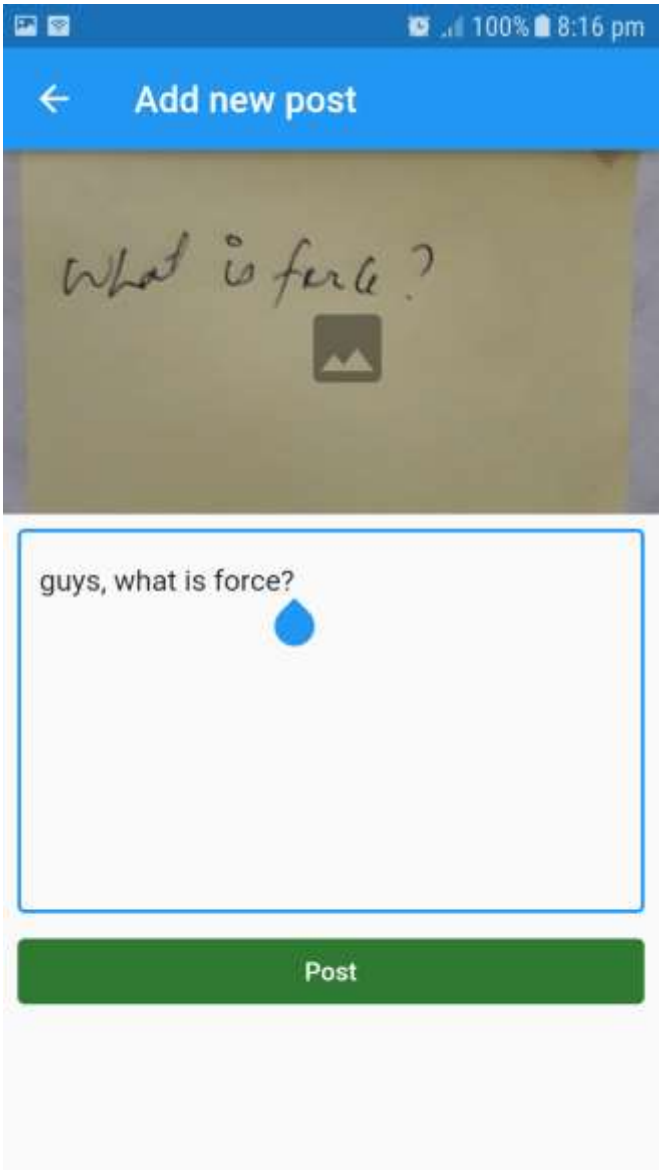


Figure 228 choose chapter system ss


100% 8:16 pm
Add new post
guys, what is force?
Post


100% 8:17 pm
Blog App
Samuel hero
guys, what is force?
What is force?
0
0
Samuel_Don
this is a schedule


100% 8:17 pm
Edit Post
guys, what is force? And what is it's unit in SI?
Post
1 2 3 4 5 6 7 8 9 0
+ × ÷ = % _ € £ ¥ ₩
! @ # $ / ^ & * ( )
1/2 - ' " : ; , ?
ABC
English (UK)
.

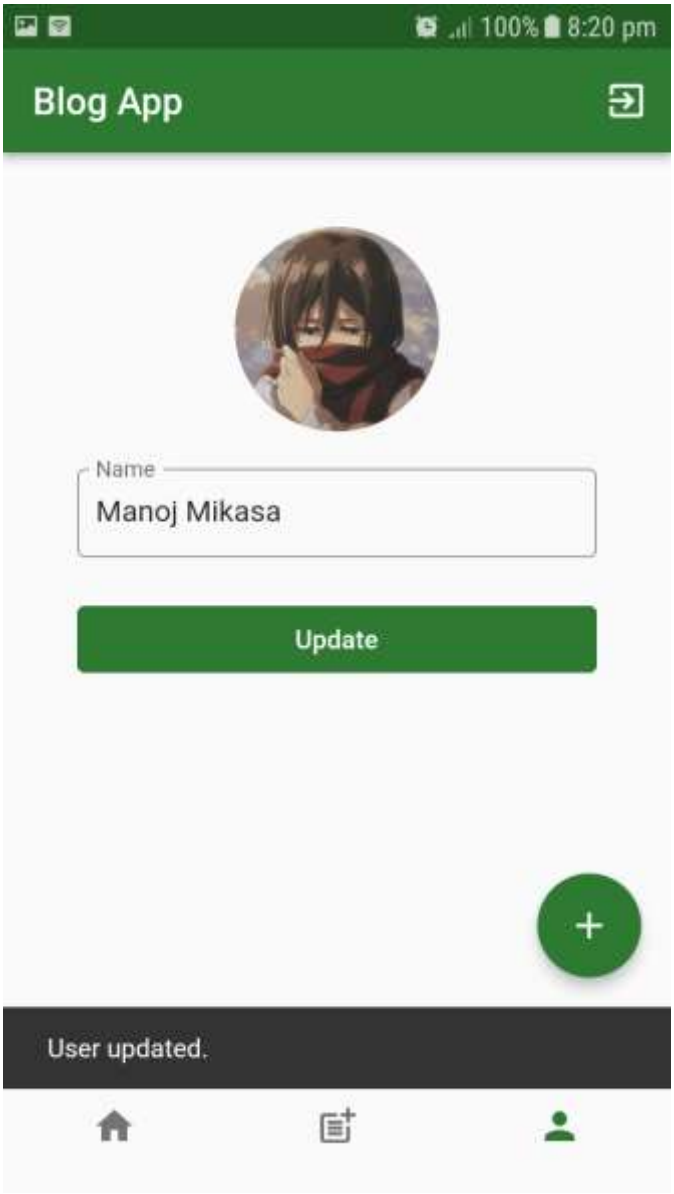
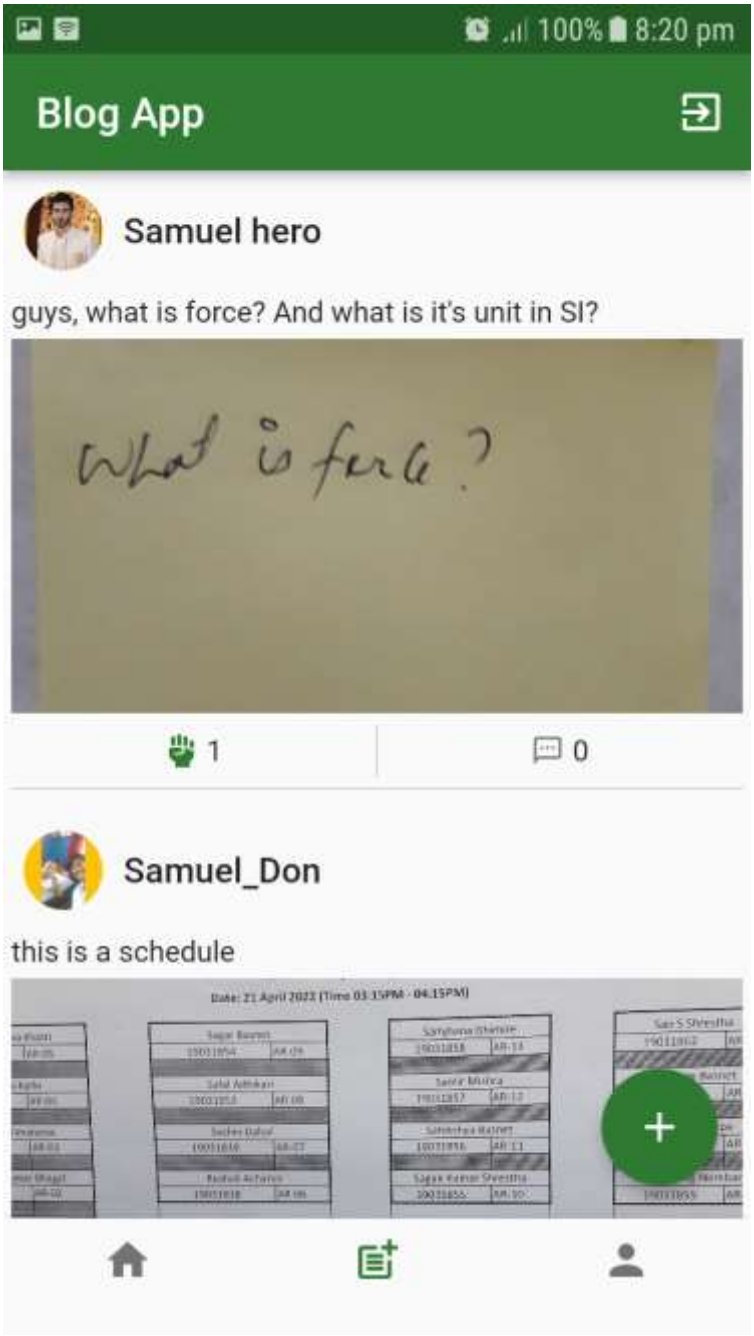


Figure 229 Update user profile


100% 8:20 pm
Blog App
Samuel hero
guys, what is force? And what is it's unit in SI?
What is force?
1
0
Samuel_Don
this is a schedule

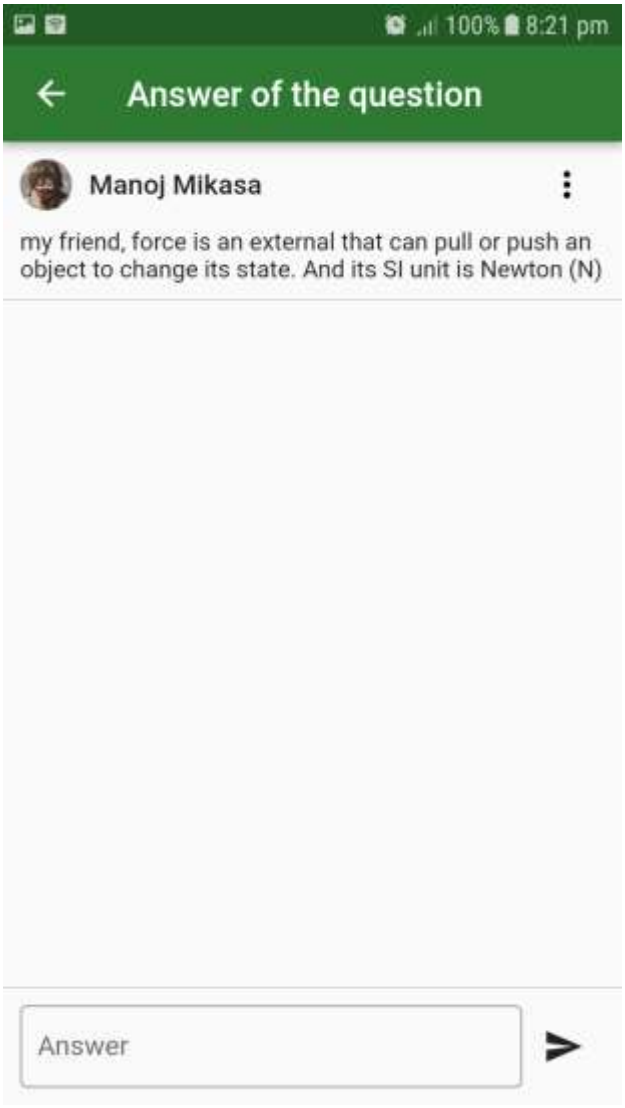


Figure 230 answer ss of the system.

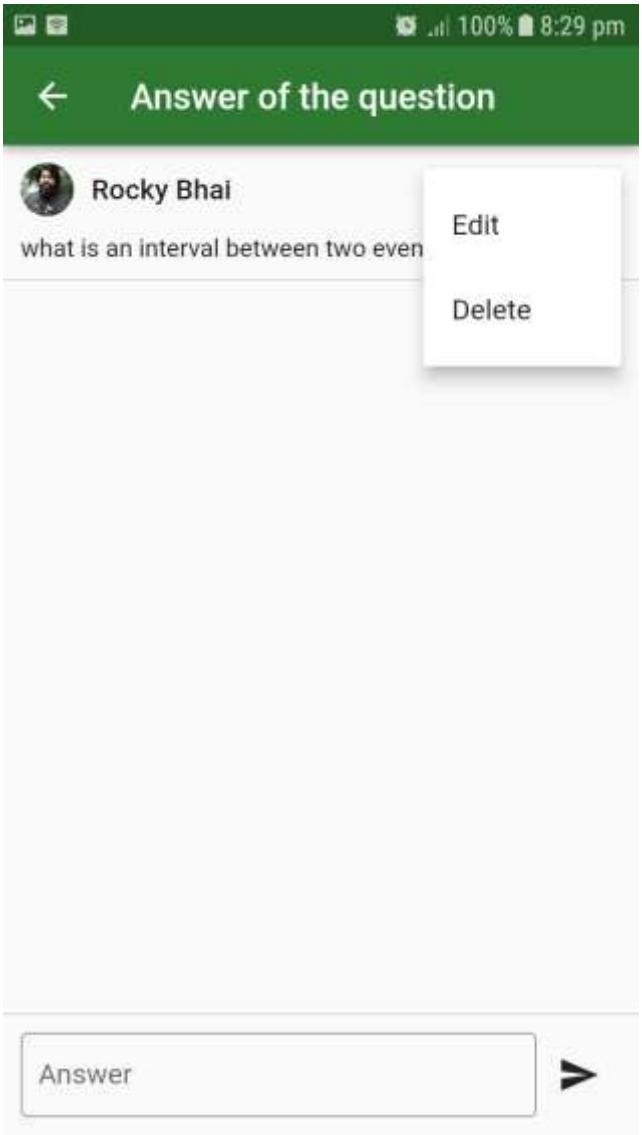


Figure 231 Edit answer by a user.

8.6. Appendix F: User FeedBack

8.6.1. User Feedback form

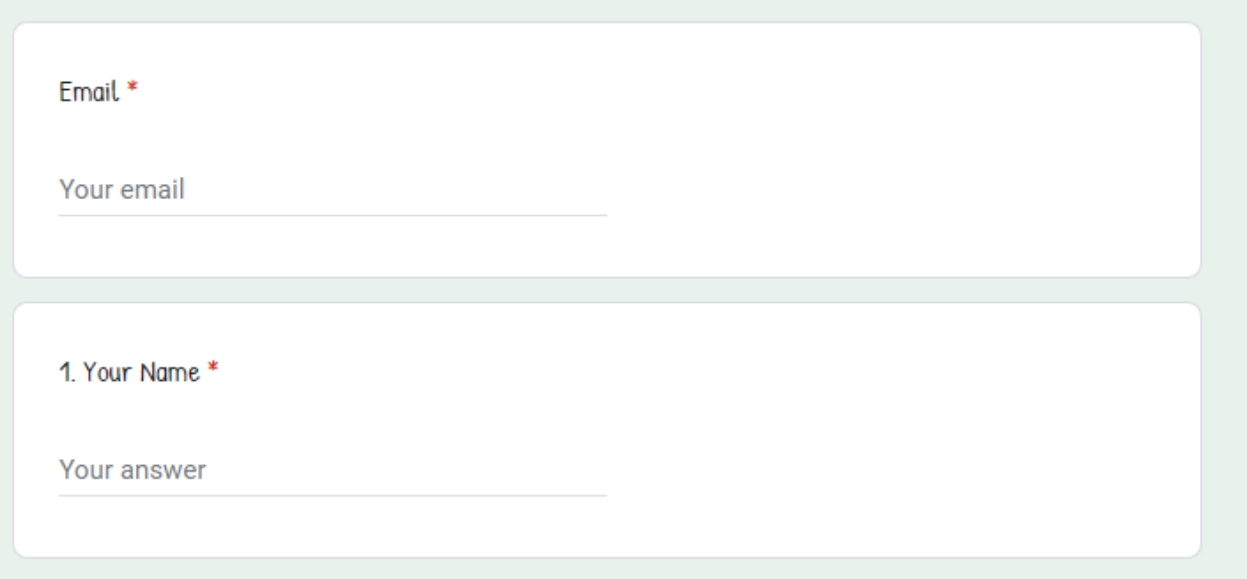


Figure 232 User feedback form part 1

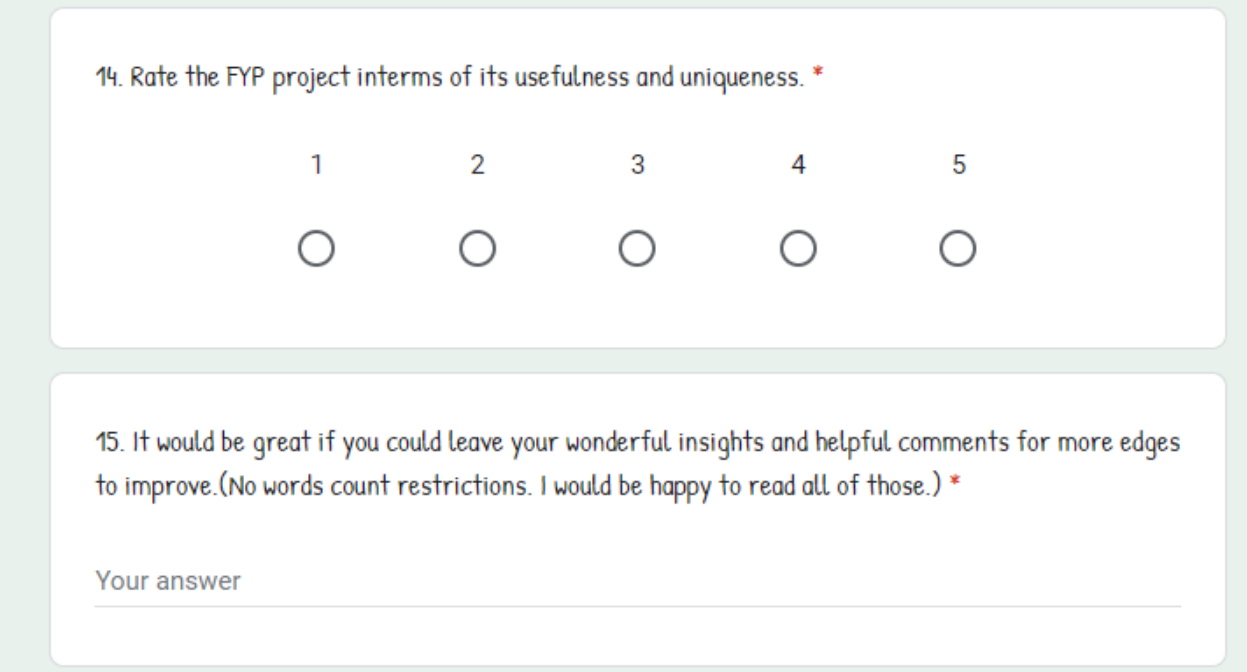


Figure 233 User feedback form part 2

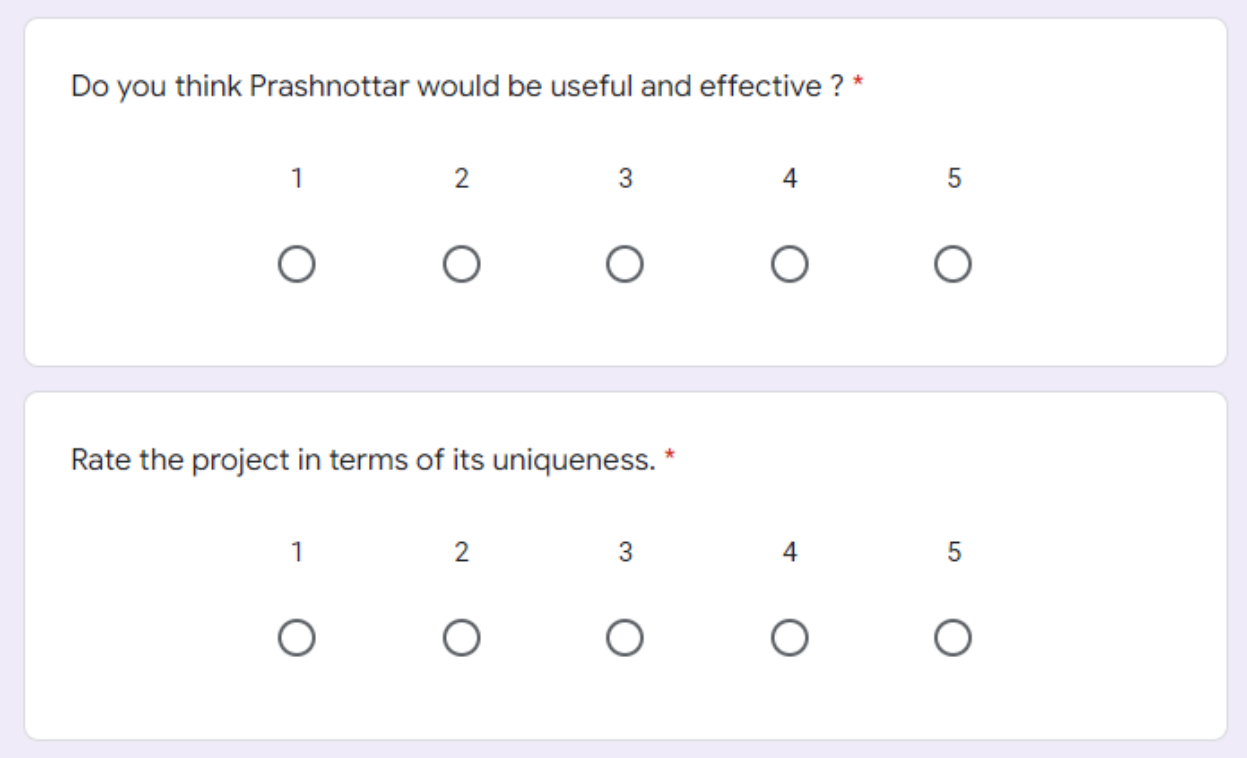


Figure 234 User feedback form part 2

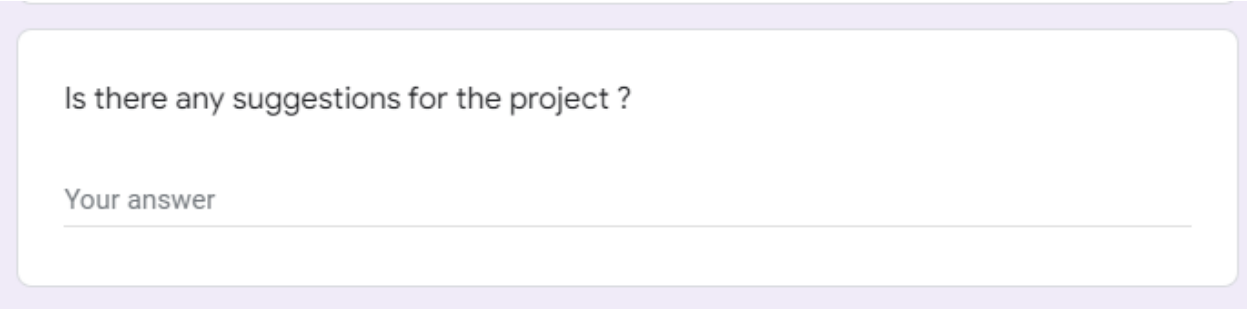


Figure 235 User feedback form part 3

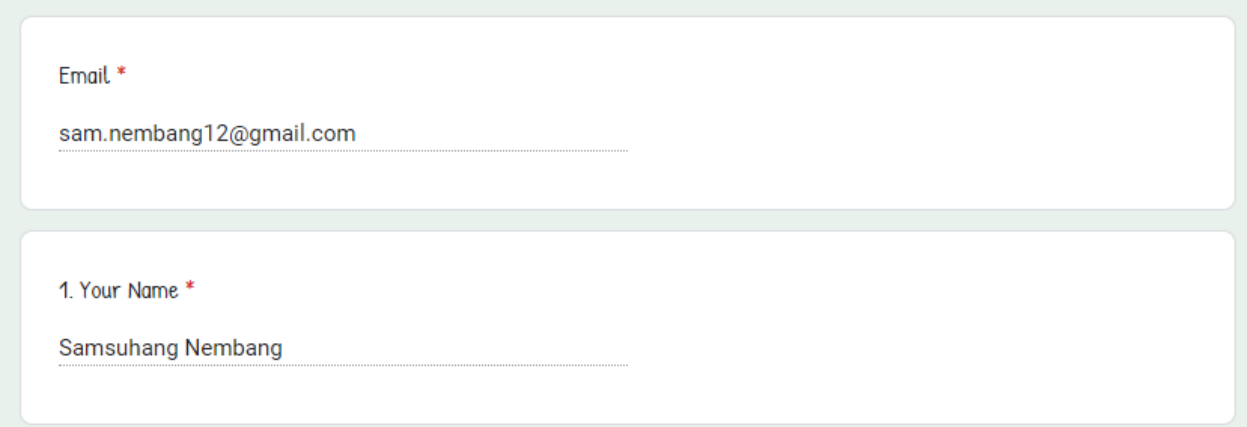


Figure 236 User feedback form part 3

8.6.2. Sample of Filled User Feedback Forms

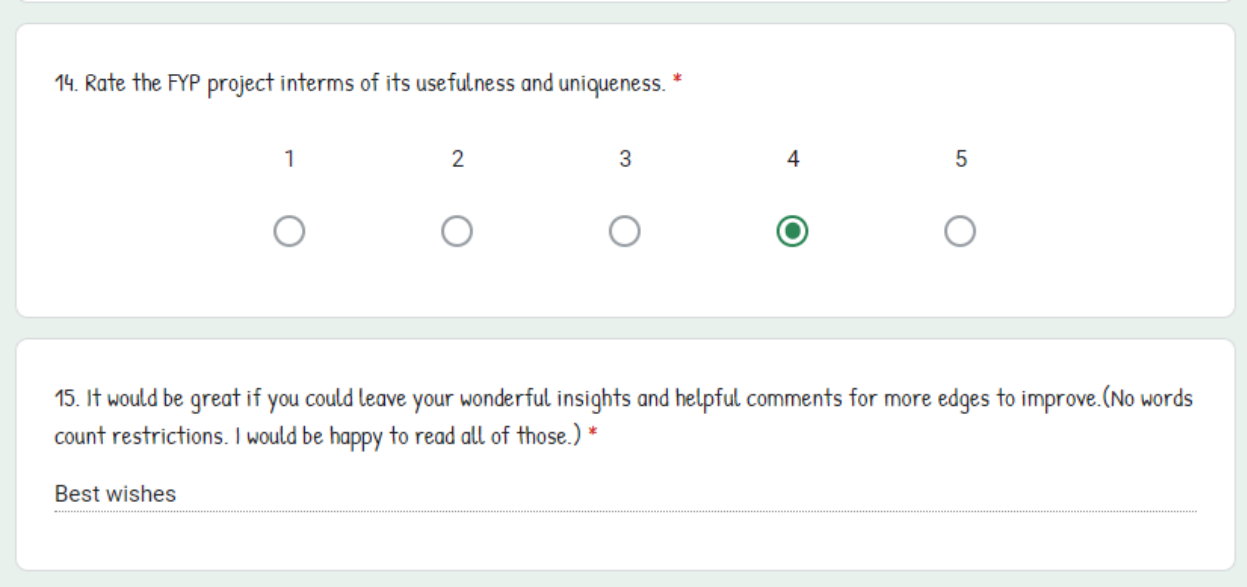


Figure 237 User filled feedback form part 1

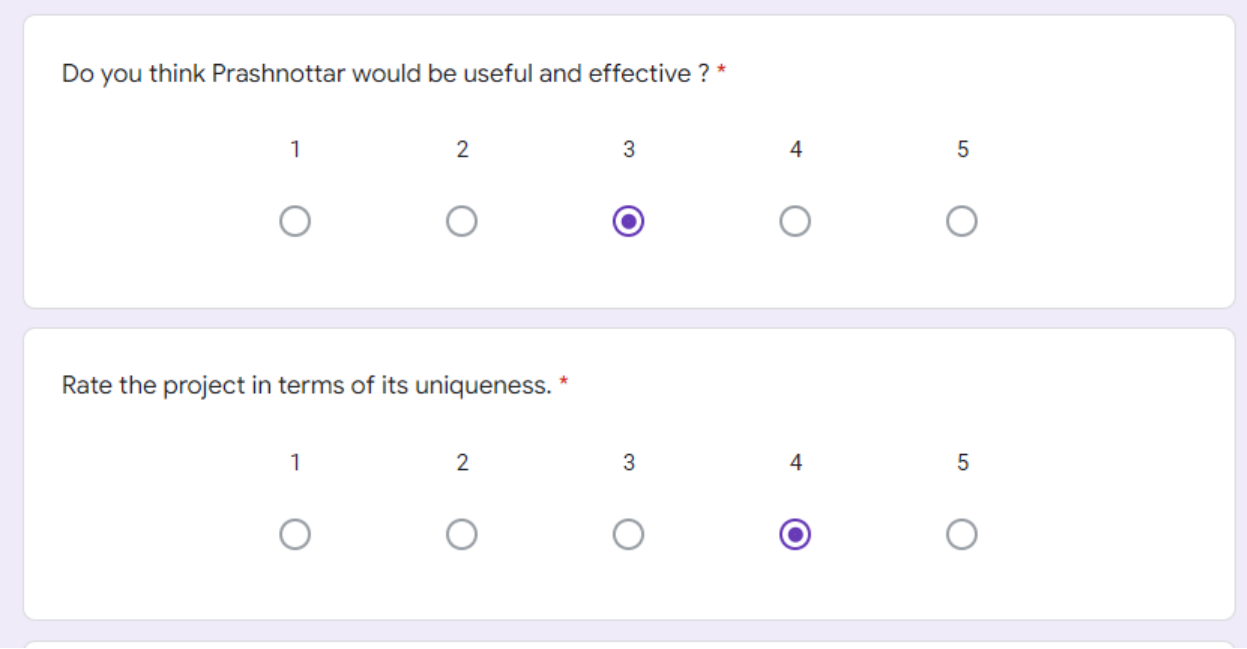


Figure 238 User filled feedback form part 1

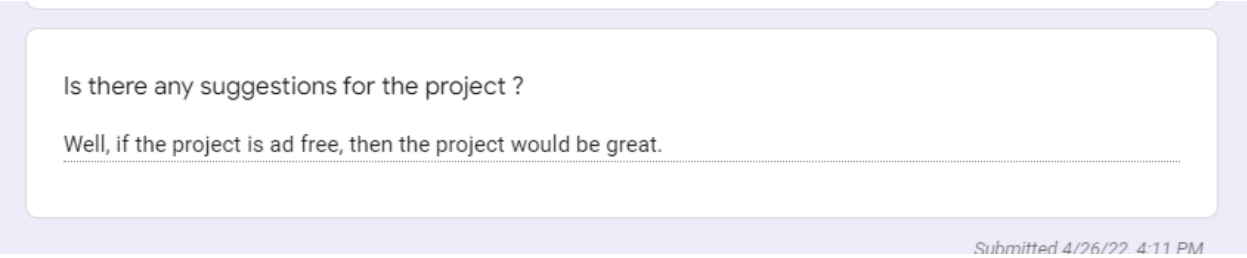


Figure 239 User filled feedback form part 2