A Project Report

on

VIRTUAL EXPO

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

by

Divya Sai N	**(174G1A0517)**
Jyoshna E	**(174G1A0526)**
Sathyamshubhava V	**(174G1A0560)**
Rajesh J S	**(174G1A0555)**

Under the guidance of

Mr. C. Sudheer Kumar M.Tech., (Ph.D)

Assistant Professor

Department of Computer Science & Engineering

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

ANANTHAPURAMU

(Affiliated to JNTUA, Accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi & Accredited by NBA (EEE, ECE & CSE))

2020-2021

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY
ANANTHAPURAMU

(Affiliated to JNTUA, Accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi & Accredited by NBA (EEE, ECE & CSE))

Certificate

This is to certify that the project report entitled Virtual Expo is the bonafide work carried out by **Divya Sai N** bearing Roll Number 174G1A0517, **Jyoshna E** bearing Roll Number 174G1A0526, **Sathyamshubhava V** bearing Roll Number 174G1A0560, **Rajesh J S** bearing Roll Number 174G1A0555 in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year of 2020 - 2021.

Guide

Mr. C. Sudheer Kumar M.Tech., (Ph.D)
Assistant Professor

Head of the Department

Dr. G. K. V. Narasimha Reddy M.Tech., Ph.D
Professor & HOD

Date:

EXTERNAL EXAMINER

Place: Ananthapuram

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we now have the opportunity to express our gratitude for all of them.

It is with immense pleasure that we would like to express our indebted gratitude to our Guide **Mr. C. Sudheer Kumar, M.Tech., (Ph.D), Computer Science and Engineering,** who has guided us a lot and encouraged us in every step of the project work. We thank him for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We express our sincere gratitude to our project coordinators **Dr. P. Chitralingappa, M.Tech., Ph.D, Associate Professor, Computer Science & Engineering** and **Mrs. M. Sowmya, M.Tech, Assistant Professor, Computer Science & Engineering** for their valuable suggestions and unstinting encouragement which enabled us to complete our project successfully in time.

We are very much thankful to **Dr. G. K. V. Narasimha Reddy, M.Tech., Ph.D, Professor & Head of the Department, Computer Science & Engineering,** for his kind support and for providing necessary facilities to carry out the project.

We wish to convey our special thanks to **Dr. G. Balakrishna, M.Tech., Ph.D, Principal** of **Srinivasa Ramanujan Institute of Technology** for giving the required information in doing our project work. Not to forget, we thank all other faculty and non-teaching staff, and my friends who had directly or indirectly helped and supported us in completing our project in time.

We also express our sincere thanks to the Management for providing excellent facilities.

Finally, we wish to convey our gratitude to our family who fostered all the requirements and facilities that we need.

Project Associates

DECLARATION

We, Ms. N. Divya Sai bearing reg. no. : 174G1A0517, Ms. E. Jyoshna bearing reg. no. : 174G1A0526, Mr. V. Sathyamshubhava bearing reg. no. : 174G1A0560, Mr. J. S. Rajesh bearing reg. no. : 174G1A0555, students of SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, Rotarypuram, hereby declare that the dissertation entitled "VIRTUAL EXPO" embodies the report of our project work carried out by us during IV Year Bachelor of Technology under the guidance of C. Sudheer Kumar, M.Tech, Department of CSE and this work has been submitted for the partial fulfillment of the requirements for the award of the Bachelor of Technology degree.

The results embodied in this project report have not been submitted to any other Universities of Institute for the award of Degree.

Divya Sai N	(174G1A0517)
Jyoshna E	(174G1A0526)
Sathyamshubhava V	(174G1A0560)
Rajesh J S	(174G1A0555)

Contents

List of Figures

List of Tables

LIST OF ABBREVIATIONS

PR	Pull Request
SDK	Software Development Kit
UML	Unified Model Language
UI	User Interface
VSCODE	Visual Studio Code
IDE	Integrated Development Environment
YAML	YAML Ain’t Markup Language
FCM	Firebase Cloud Messaging
SRS	Software Requirement Specifications
UAT	User Acceptance Testing

ABSTRACT

An era of Virtuality (talks, meets, chats, exams, interviews, etc.) has evolved and is in the need of developing the necessities, tools and utilities with a good security has a lot of demand for the current scenario in order to provide the best experience for the users. A small atom cannot be viewed by many people physically at once, but it is possible to see the same atom virtually by millions of people and give their opinions, reviews, statements and conclusions at once.

The main objective of our project is to provide a platform to showcase the student's projects in a mobile application which can be viewed by the public (here, restricted to SRIT members). The Mobile Application will be projected in the form of a contest and an individual user can express their interest and comments on the projects in the form of points/likes/any parameter. The project is developed with the FLUTTER toolkit which uses the DART language with the help of FIREBASE backend functionality.

Keywords: Contest, Flutter, Firebase, Project, Expo, Review, Mobile Application.

CHAPTER - 1
INTRODUCTION

An era of Virtuality came into existence and this will be continued for a longer stretch of time. In order to tackle our problems with ease, a lot many tools, platforms, utilities and applications are to be developed. Our Project is one of the applications (Mobile App) where the students can post their projects and every individual (SRIT member) can show their interest and comments on the project.

It can also be conducted as a Contest with any parameter like LIKES/POINTS in order to get the best among the best.

1.1 Problem Definition

For an organization like SRIT, it is hard for students to reach out to each and every project on a single day in the EXPO. A special channel/platform is not present for the individual organization to have a virtual expo and to get their members opinion.

1.2 Solution

A Mobile application to post the student projects (such that the end user is able to understand) and can conduct like a contest by considering the responses of the reviewers (Review estimation will be done based on the Rubrics of Student Evaluation designed by JNTUA).

1.3 Objective

The objective of this project is to provide information about projects done by the final year students and thereby reduce the amount of effort that the user has to go through in searching for information about projects that have been done by the final year students.

It also helps students to popularize the projects that have been done in the college by enabling the user to provide information about their project, just by conducting a Virtual Expo. such that every single student had the access to check over the projects done by their seniors whenever they wanted to.

1.4 Flutter

Flutter is an open-source UI software development kit created by Google. It is used to develop applications for Android, iOS, Linux, Mac, Windows, and the web from a single codebase by using DART language.

1.4.1 Characteristics

- Fast Development
- Expressive and Flexible UI
- Native Performance

1.5 Dart

Dart is a programming language developed by Google that is used to build mobile, desktop, server, and web applications. Dart is an object-oriented, class-based, garbage-collected language with C-style syntax. Dart can compile to either native code or JavaScript. It supports interfaces, abstract classes, and type inference.

1.6 Firebase

Firebase is a platform developed by Google which is capable of handling the backend functionality of an Application (Mobile and Web).

1.6.1 Features/Products we used:

- Authentication
- Cloud Firestore - NoSQL Database
- Cloud Messaging

1.7 Overview Of Development

The following figure (Fig.1.1) describes the overview of development of the complete project.

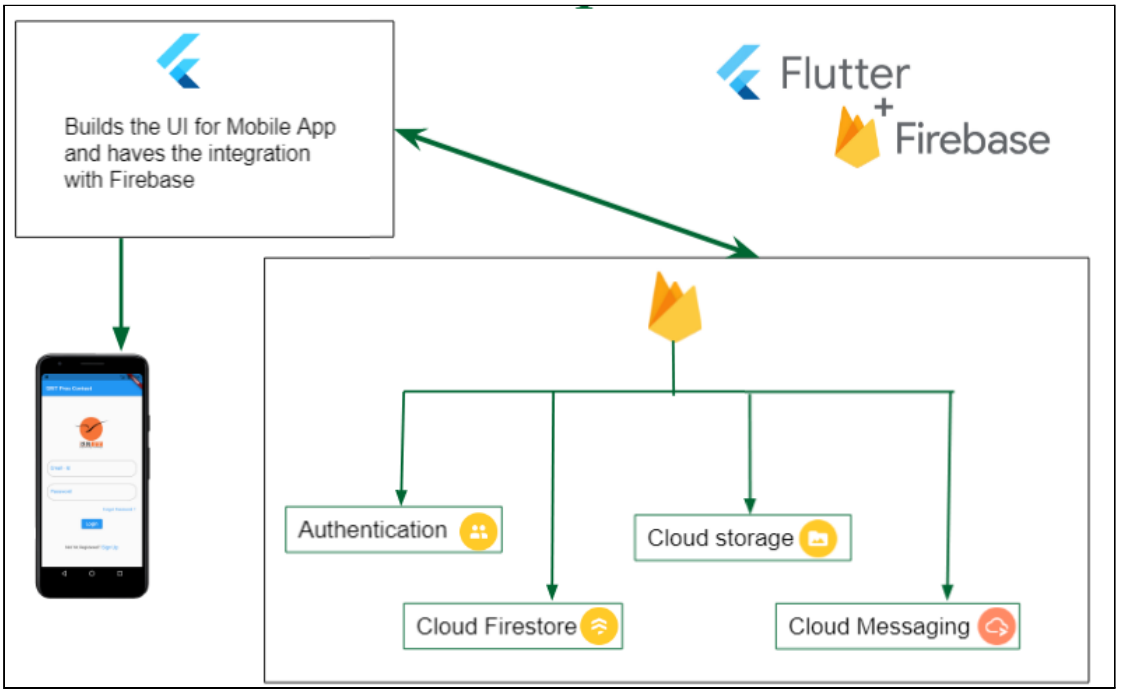


Fig.1.1. Overview of Development.

CHAPTER - 2
LITERATURE SURVEY

2.1 Existing System

For an Organization like SRIT, it is hard for students to reach out to each and every project on a single day in the EXPO. A special channel/ platform is not present for the individual organizations to have a virtual expo and to get their members' opinion .

2.2 Disadvantages of Existing System:

- It is tough to expose the projects physically because it consumes more time.
- Students do not get the information about the projects unless they have a chance.
- Students even get a chance to go through the project once, they cannot have an opportunity to go through them again whenever they want to.
- There is no specific chance for students to popularize their projects, as they do not have a specified platform.
- Not everyone is able to be free at the time when there is a physical project expo conducted so they would miss the chance of knowing the projects.

2.3 Proposed System:

A Mobile application to post the student projects (such that the end user is able to understand) and can conduct like a contest by considering the likes of the public (here, restricted to SRIT members).

2.4 Advantages of Proposed System:

The advantages of the proposed system are as follows:

- The main advantage of this app is it provides students as well as staff a friendly platform to view the projects.
- Proposed system helps students in reducing the need to conduct multiple physical expos in college in order to explain and to describe about the project

that has been done in their final year or other projects that they can organize by enabling the user to upload the information about the project through "VIRTUAL EXPO" option.

- Proposed system helps students in reducing the efforts that they have to go through in searching for the information about projects in college websites that take place in the college every year by providing all information about projects that take place in every year.
- This proposed system can also provide users an opportunity to check out the project irrespective of time and place.
- students can vote on the projects, staff can both comment and vote on the project, which is a good opportunity for participants to know their work.

CHAPTER - 3
REQUIREMENT ANALYSIS

3.1 Hardware Requirements

The following table (Table 3.1) shows the hardware requirements of this project in order to develop.

Table 3.1 Hardware Requirements

Operating Systems	For Windows OS-Windows 7 SP1 or later (64-bit), x86-64 based
Disk Space	*4.5 GB* (including disk space for IDE/tools)
RAM	8 GB

3.2 Software Requirement Specifications

SRS is a document created by a system analyst after the requirements are collected. SRS defines how the intended software will interact with hardware, external interfaces, speed of operation, response time of system, portability of software across various platforms, maintainability, speed of recovery after crashing, Security, Quality, Limitations etc.

The requirements received from clients are written in natural language. It is the responsibility of system analysts to document the requirements in technical language so that they can be comprehended and useful by the software development team.

The SRS is a specification for a specific software product, program, or set of applications that perform particular functions in a specific environment. It serves several goals depending on who is writing it. First, the SRS could be written by the client of a system. Second, the SRS could be written by a developer of the system. The two methods create entirely various situations and establish different purposes for the document altogether. The first case, SRS, is used to define the needs and expectations of the users. The second case, SRS, is written for various purposes and serves as a contract document between customer and developer.

3.3 Software Requirements

The Software requirements required for developing this project are shown in the following table (Table 3.2).

Table 3.2 Software Requirements

Programming language	Dart
SDK	Flutter
Backend Functionality	Firebase
Supportive software	Android Studio
IDE	Visual Studio Code
Tools	Git 2.x

3.3.1 VSCODE Installation

1. Download the Visual Studio Code installer for Windows.
2. Once it is downloaded, run the installer (VSCodeUserSetup-{version}.exe). This will only take a minute.
3. By default, VS Code is installed under

   C:\users\{username}\AppData\Local\Programs\Microsoft VS Code.

3.3.2 Git Installation

There are also a few ways to install Git on Windows. The most official build is available for download on the Git website. Visit https://git-scm.com/download/win and the download will start automatically. Note that this is a project called Git for Windows, which is separate from Git itself; for more information on it, go to https://gitforwindows.org.

To get an automated installation you can use the Git Chocolatey package. Note that the Chocolatey package is community maintained.

3.3.3 Flutter Installation

Get the Flutter SDK

1. Download the following installation bundle to get the latest stable release of the Flutter SDK: https://flutter.dev/docs/get-started/install/windows

   For other release channels, and older builds, see the SDK releases page.

2. Extract the zip file and place the contained flutter in the desired installation location for the Flutter SDK.

   (for example, C:\Users\<your-user-name>\Documents).

3. If you do not want to install a fixed version of the installation bundle, you can skip steps 1 and 2. Instead, get the source code from the Flutter repo on GitHub, and change branches or tags as needed. For example:

Update your path

If you wish to run Flutter commands in the regular Windows console, take these steps to add Flutter to the PATH environment variable:

- From the Start search bar, enter ‘env’ and select Edit environment variables for your account.
- Under User variables check if there is an entry called Path:
    - If the entry exists, append the full path to flutter\bin using ; as a separator from existing values.
    - If the entry doesn’t exist, create a new user variable named Path with the full path to flutter\bin as its value.

You have to close and reopen any existing console windows for these changes to take effect.

Run flutter doctor

From a console window that has the Flutter directory in the path (see above), run the following command to see if there are any platform dependencies you need to complete the setup:

C:\src\flutter>flutter doctor

This command checks your environment and displays a report of the status of your Flutter installation. Check the output carefully shown in the figure (Fig.2.1), For other software you might need to install or further tasks to perform (shown in bold text).

For example:

```
[-] Android toolchain - develop for Android devices
    • Android SDK at D:\Android\sdk
    X Android SDK is missing command line tools; download from https://goo.gl/XxQghQ
    • Try re-installing or updating your Android SDK,
      visit https://flutter.dev/setup/#android-setup for detailed instructions.
```

Fig.2.1. Report of flutter doctor.

The following sections describe how to perform these tasks and finish the setup process. Once you have installed any missing dependencies, you can run the flutter doctor command again to verify that you've set everything up correctly.

Android Setup

Install Android Studio

1. Download and install Android Studio.
2. Start Android Studio, and go through the 'Android Studio Setup Wizard'. This installs the latest Android SDK, Android SDK Command-line Tools, and Android SDK Build-Tools, which are required by Flutter when developing for Android.
3. Run flutter doctor to confirm that Flutter has located your installation of Android Studio. If Flutter cannot locate it, run flutter config --android-studio-dir <directory> to set the directory that Android Studio is installed to.

Set up your Android device

To prepare to run and test your Flutter app on an Android device, you need an Android device running Android 4.1 (API level 16) or higher.

1. Enable Developer options and USB debugging on your device. Detailed instructions are available in the Android documentation.
2. Windows-only: Install the Google USB Driver.
3. Using a USB cable, plug your phone into your computer. If prompted on your device, authorize your computer to access your device.
4. In the terminal, run the flutter devices command to verify that Flutter recognizes your connected Android device. By default, Flutter uses the version of the Android SDK where your adb tool is based. If you want Flutter to use a different installation of the Android SDK, you must set the ANDROID_SDK_ROOT environment variable to that installation directory.

Set up the Android Emulator

To prepare to run and test your Flutter app on the Android emulator, follow these steps:

1. Enable VM acceleration on your machine.
2. Launch Android Studio, click the AVD Manager icon, and select Create Virtual Device…
    - In older versions of Android Studio, you should instead launch Android Studio > Tools > Android > AVD Manager and select Create Virtual Device…. (The Android submenu is only present when inside an Android project.)
    - If you do not have a project open, you can choose Configure > AVD Manager and select Create Virtual Device…
3. Choose a device definition and select Next.
4. Select one or more system images for the Android versions you want to emulate, and select Next. An *x86* or *x86_64* image is recommended.
5. Under Emulated Performance, select Hardware - GLES 2.0 to enable hardware acceleration.

6. Verify the AVD configuration is correct, and select Finish.
   For details on the above steps, see Managing AVDs.
7. In Android Virtual Device Manager, click Run in the toolbar. The emulator starts up and displays the default canvas for your selected OS version and device.

Agree to Android Licenses

Before you can use Flutter, you must agree to the licenses of the Android SDK platform. This step should be done after you have installed the tools listed above.

1. Make sure that you have a version of Java 8 installed and that your JAVA_HOME environment variable is set to the JDK's folder.
   Android Studio versions 2.2 and higher come with a JDK, so this should already be done.
2. Open an elevated console window and run the following command to begin signing licenses

   flutter doctor --android-licenses
3. Review the terms of each license carefully before agreeing to them.
4. Once you are done agreeing with licenses, run flutter doctor again to confirm that you are ready to use Flutter.

Enable Desktop Support

At the command line, perform the following command to enable Win32 desktop support:

flutter config --enable-windows-desktop

For Windows UWP desktop support perform the following commands to switch to the dev channel, upgrade Flutter, and enable UWP.

flutter channel dev

$ flutter upgrade

$ flutter config --enable-windows-uwp-desktop

3.3.4 Adding Firebase to Flutter

STEP 1: Register your App with Firebase

1. In the center of the Firebase console's project overview page, click the **iOS** icon (plat_ios) to launch the setup workflow.
   If you've already added an app to your Firebase project, click **Add app** to display the platform options.
2. Enter your app's bundle ID in the **iOS bundle ID** field.
   Find this bundle ID from your open project in XCode. Select the top-level app in the project navigator, then access the **General** tab. The **Bundle Identifier** value is the iOS bundle ID (for example, com.yourcompany.ios-app-name).
3. (Optional) Enter other app information as prompted by the setup workflow.
   - **App nickname**: An internal, convenience identifier that is only visible to you in the Firebase console
   - **App Store ID**: Used by Firebase Dynamic Links to redirect users to your App Store page and by Google Analytics to import conversion events into Google Ads. If your app doesn't yet have an App Store ID, you can add the ID later in your Project Settings.
4. Click **Register app**.

STEP 2: Add a Firebase Configuration File

1. Click **Download GoogleService-Info.plist** to obtain your Firebase iOS config file (GoogleService-Info.plist).
   - You can download your Firebase iOS config file again at any time.
   - Make sure the config file is not appended with additional characters, like (2).
2. Using Xcode, move the file into the Runner/Runner directory of your Flutter app.
3. Back in the Firebase console setup workflow, click **Next** to skip the remaining steps.
4. Continue to Add FlutterFire plugins.

STEP 3: Add Flutter Fire Plugins

Flutter uses plugins to provide access to a wide range of platform-specific services, such as Firebase APIs. Plugins include platform-specific code to access services and APIs on each platform.

Firebase is accessed through a number of different libraries, one for each Firebase product (for example: Realtime Database, Authentication, Analytics, or Cloud Storage). Flutter provides a set of Firebase plugins, which are collectively called **FlutterFire**.

Since Flutter is a multi-platform SDK, each FlutterFire plugin is applicable for both iOS and Android. So, if you add any FlutterFire plugin to your Flutter app, it will be used by both the iOS and Android versions of your Firebase app.

1. Ensure that your app is not currently running in your emulator or on your device.
2. From the root directory of your Flutter app, open your pubspec.yaml file.
3. Add the FlutterFire plugin for the Firebase Core Flutter SDK.

```
dependencies:
  flutter:
    sdk: flutter
```

Add the dependency for the Firebase Core Flutter SDK

firebase_core: ^0.4.0+9

Add the FlutterFire plugins for the Firebase products that you want to use.

```
dependencies:
  flutter:
    sdk: flutter
```

Check that you have this dependency (added in the previous step)

```
firebase_core: ^0.4.0+9
```

Add the dependency for the FlutterFire plugin for Google Analytics

firebase_analytics: ^5.0.2

Add the dependencies for any other Firebase products you want to use in app

For example, to use Firebase Authentication and Cloud Firestore

firebase_auth: ^0.14.0+5

cloud_firestore: ^0.12.9+5

4. Run flutter **packages get**.
   For more information about managing packages and plugins, refer to Using Packages.
5. If you added Analytics, run your app to send verification to Firebase that you've successfully integrated Firebase. Otherwise, you can skip the verification step.

You're all set! Your Flutter app is registered and configured to use Firebase.

CHAPTER - 4
DESIGN

4.1 Introduction to UML

The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic, semantic and pragmatic rules. A UML system is represented using five different views that describe the system from a distinctly different perspective.

UML is specifically constructed through two different domains, they are:

- UML Analysis modeling, this focuses on the user model and structural model views of the systems.

- UML Design modeling, this focuses on the behavioral modeling, implementation modeling and environmental model views.

Usage of UML in Project

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time to the market. These techniques include component technology, visual programming, patterns and frameworks. Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems. The UML was designed to respond to these needs. Simply, systems design refers to the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements which can be done easily through UML diagrams.

4.2 Data Flow Diagram

A data-flow diagram is a way of representing the flow of data of a process or a system (usually an information system). This also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

The data-flow diagram is part of the structured-analysis modeling tools. When using UML, the activity diagram typically takes over the role of the data-flow diagram.

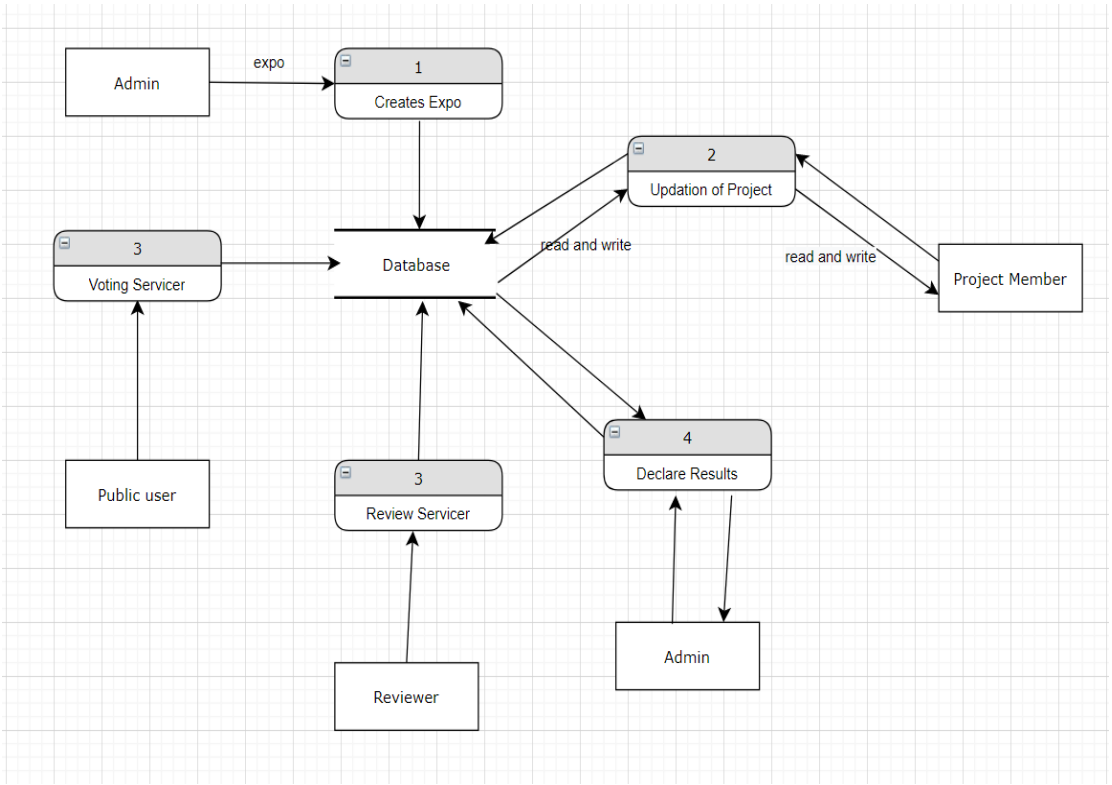


Fig.4.1. Data flow diagram.

The above data flow diagram (Fig.4.1) describes how this project has done. Admin plays a major role as he is the one having the control over all the actions performed. Admin Creates an Expo that to be stored through Database. Then participants upload their projects in the contest .If there will be any update regarding

the project a project member can make changes to one's projects .They can update until the live contest starts. Then Admin provides the voting service to the users. Regarding the voting that will be Different for the Reviewers and the Normal users. Reviewers had the opportunity to view, vote as well as comment .When it comes to the Normal users, they cannot comment but they can view and vote .Based on the votes and comments admin declares the results.

4.3 Activity Diagram

An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. We can depict both sequential processing and concurrent processing of activities using an activity diagram. They are used in business and process modeling where their primary use is to depict the dynamic aspects of a system.

4.3.1 Activity Diagram - Mobile Application

Here the diagram explains the flow of actions performed. At first a login page with login details would appear. Then there would be authentication which allows the users only with the given specific login details. Once the entry details would be correct then the user could successfully login, if not it shows invalid credentials. Once the user logs in there appears a home screen showing Expos, Profile, past contests and live contests. Regarding the Expos and live contests only Admin had the accessibility to view the results. Regarding the profile it contains details of all the users. Past contests had the details of all the conducted expos. Live contests contain the present Expos to be viewed and uploaded. The app is about contesting the actions that have been varied with the Reviewers and Normal students. Normal users can only view and vote for the contest. But reviewers had an opportunity to both vote as well as comment. At last considering all these could produce the winner with highest likes

The following figure (Fig.4.2) shows the activity diagram for the mobile application.

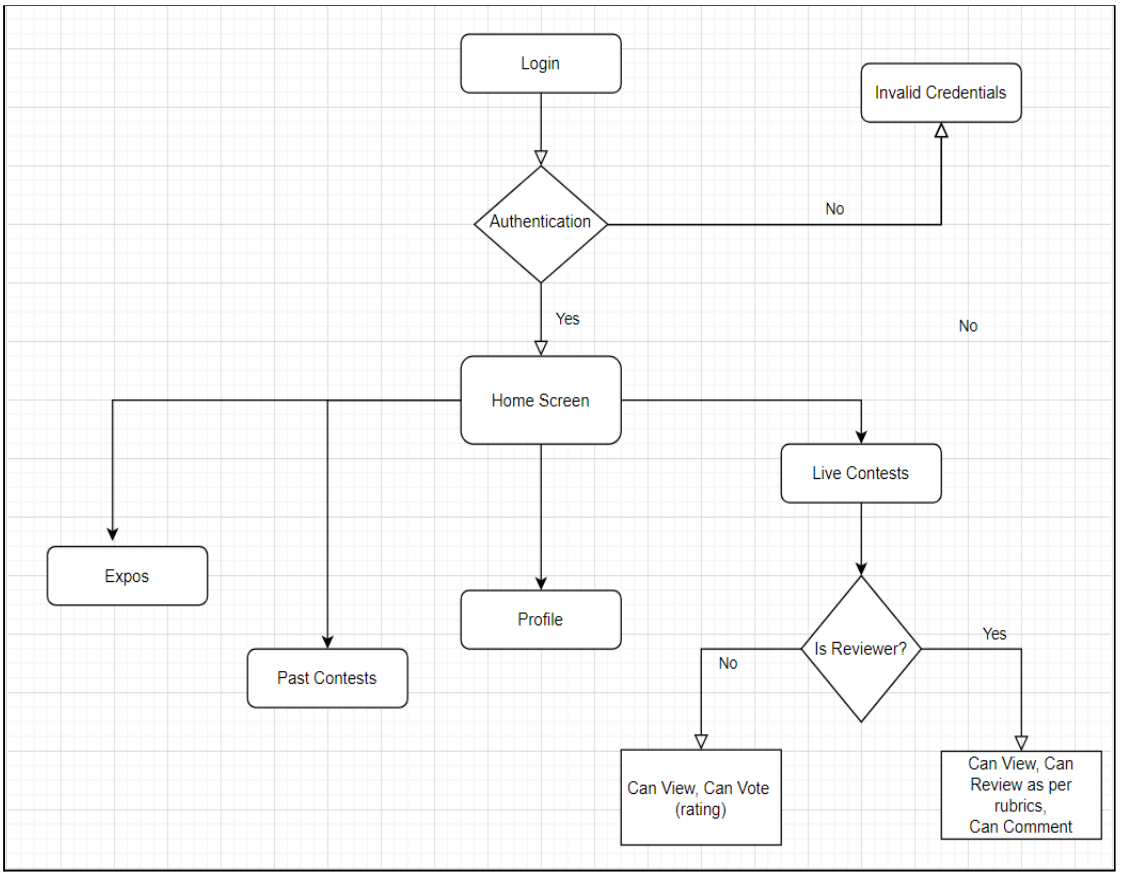


Fig.4.2. Activity Diagram for Mobile Application.

4.3.2 Activity Diagram - Web Application

Here the diagram explains the flow of actions performed. At first a login page with login details would appear. Then there would be authentication which allows the users only with the given specific login details. Once the entry details would be correct then the user could successfully login, if not it shows invalid credentials. Once the user logged in there appears a home screen showing Expos, Profile, past contests and Live contests. It runs on two cases that is if the user was an Admin then he can able to perform different actions such as managing the Expos, Creating the Expos, Hosting the contest and publishing the results based on votes.But if a project is needed to be edited then only the one who uploads the project can edit but Normal users do not, they only should view and vote. The same action is allowed for reviewers to vote and comment.

The following figure (Fig.4.3) shows the activity diagram for the mobile application.

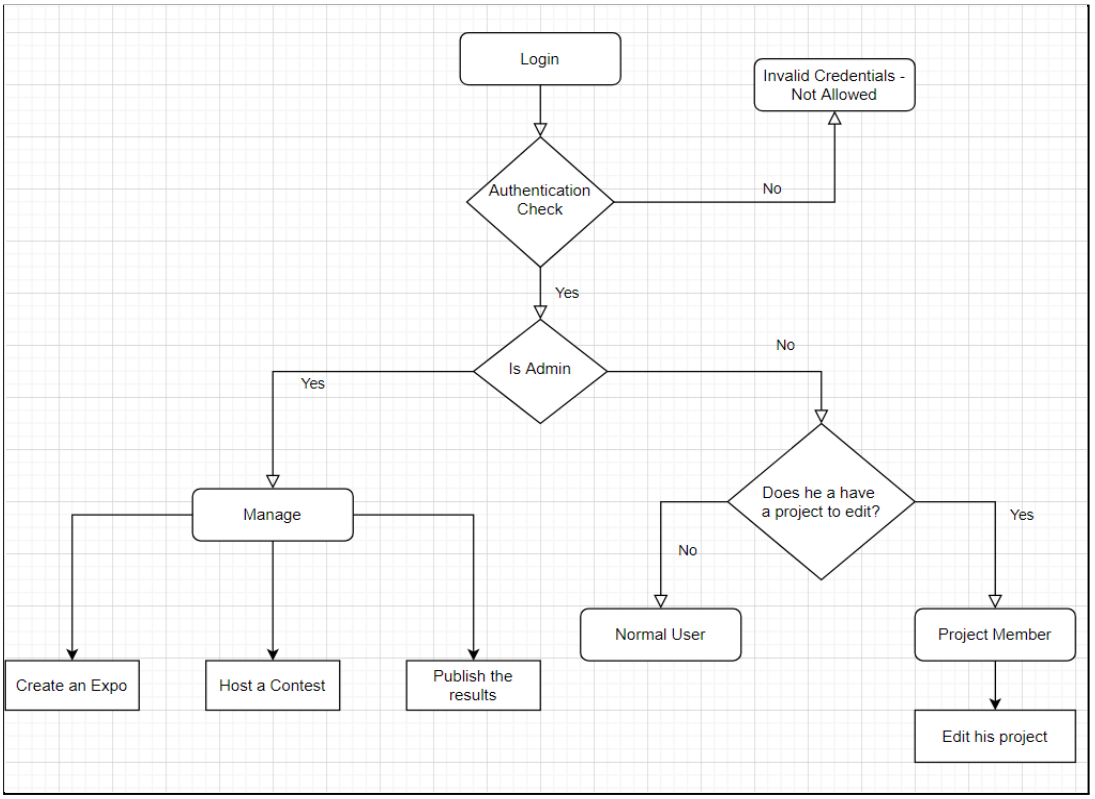


Fig.4.3. Activity Diagram for Web Application.

CHAPTER - 5
IMPLEMENTATION

We developed the complete project using the software development tools like Git, VSCode with the help of Github and Firebase platforms.

5.1 PIPELINE SETUP

Firstly, we set up a flow from the level of writing code to the level of publishing/distributing it (for our testers). So, whatever the addition/modification of code is made, It should reflect to the tester.

We maintain two repositories in Github where one is SRIT_VIRTUAL_EXPO for the mobile application and the other is SRIT_VIRTUAL_EXPO_WEB_APP for the website application.

Code in VS > Test > make a PR in github > Test > Merge > Publish

5.2 FLUTTER CODE

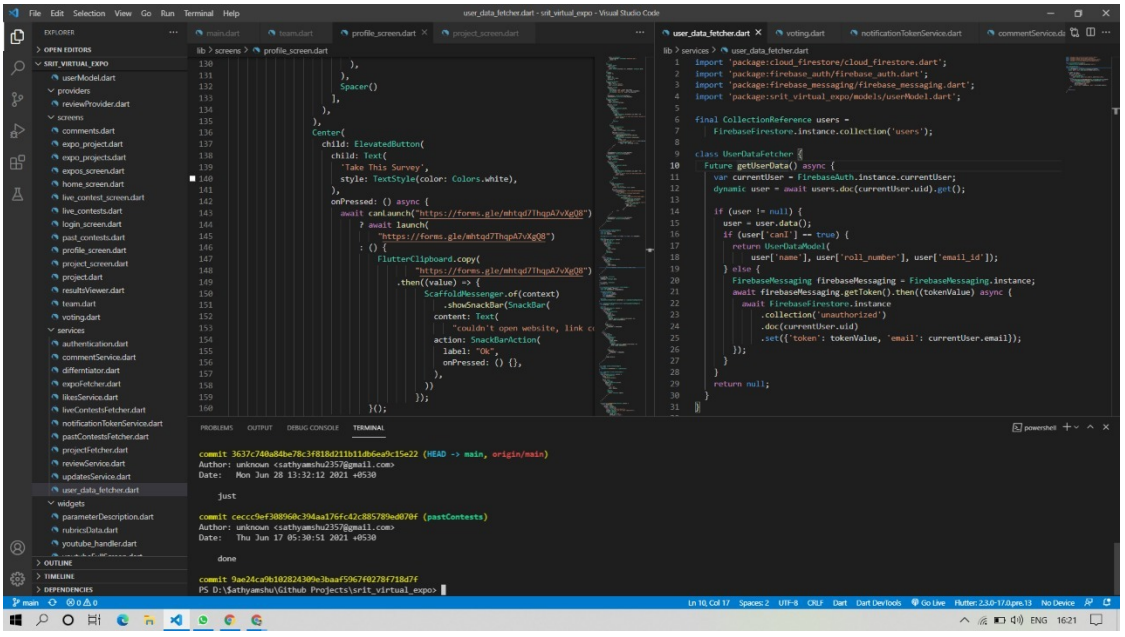


Fig.5.1. Flutter Code for Mobile Application.

The above figure (Fig.5.1) is a screenshot of the view of the Flutter code for the Mobile Application.

The below figure (Fig.5.2) is a screenshot of the view of the Flutter code for the Web Application.

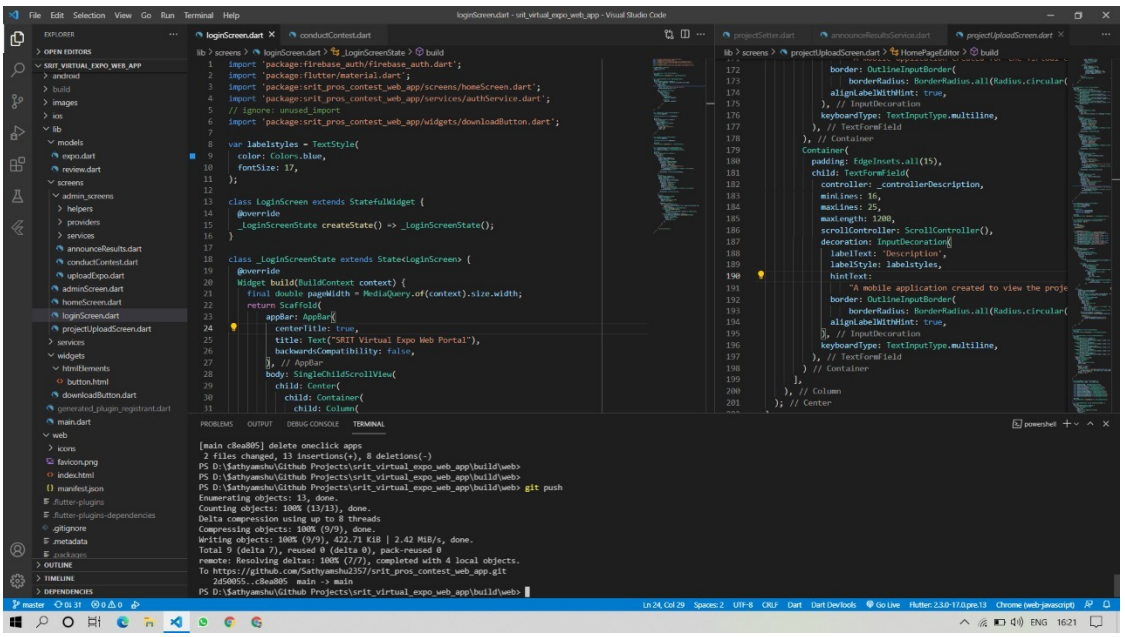


Fig.5.2. Flutter Code for Web Application.

5.3 Firebase

5.3.1 Authentication

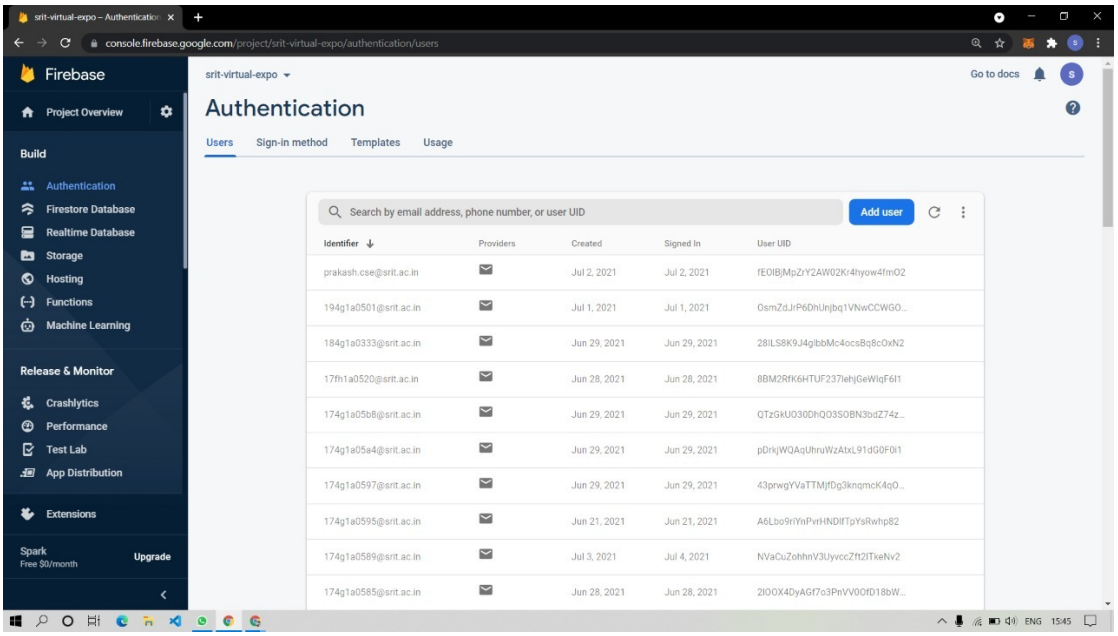


Fig.5.3. Firebase Authentication.

All the users authentication details will be stored in this section and It is possible to reset their password, disable them and delete their accounts and a screenshot of those are shown in the above figure (Fig.5.3).

5.3.2 Cloud Firestore - NoSQL Database

Firebase supports both SQL and NoSQL databases. We opted for the NoSql database for our project in order to implement the application with optimum cost to deploy and maintain for the complete organization like SRIT.

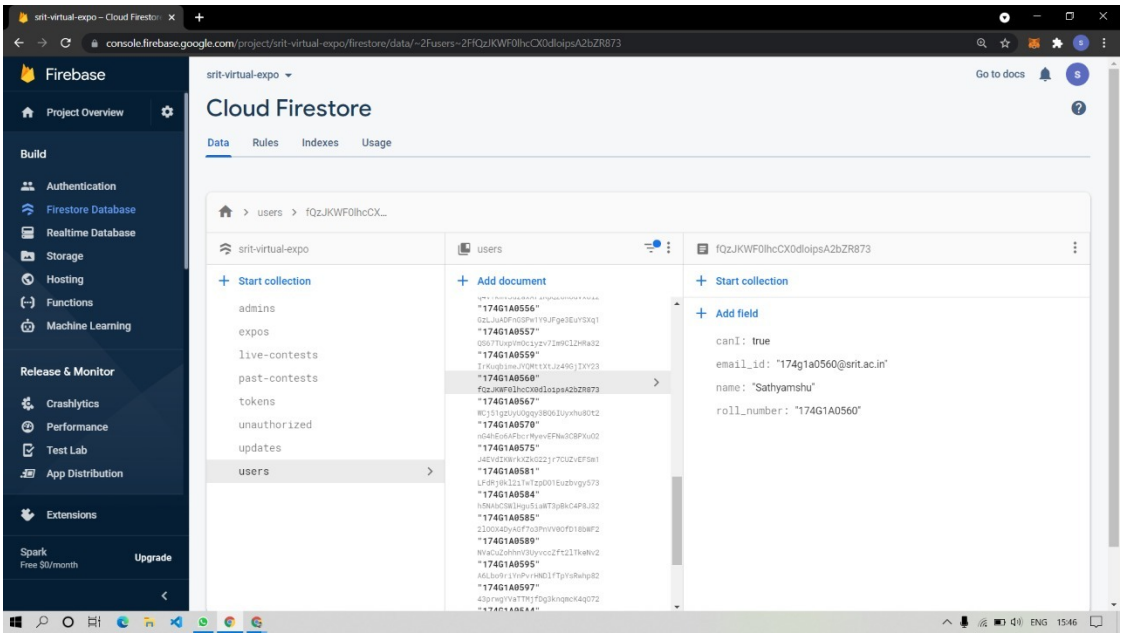


Fig.5.4. Firestore - NoSQL Database.

5.3.3 Firestore Usage

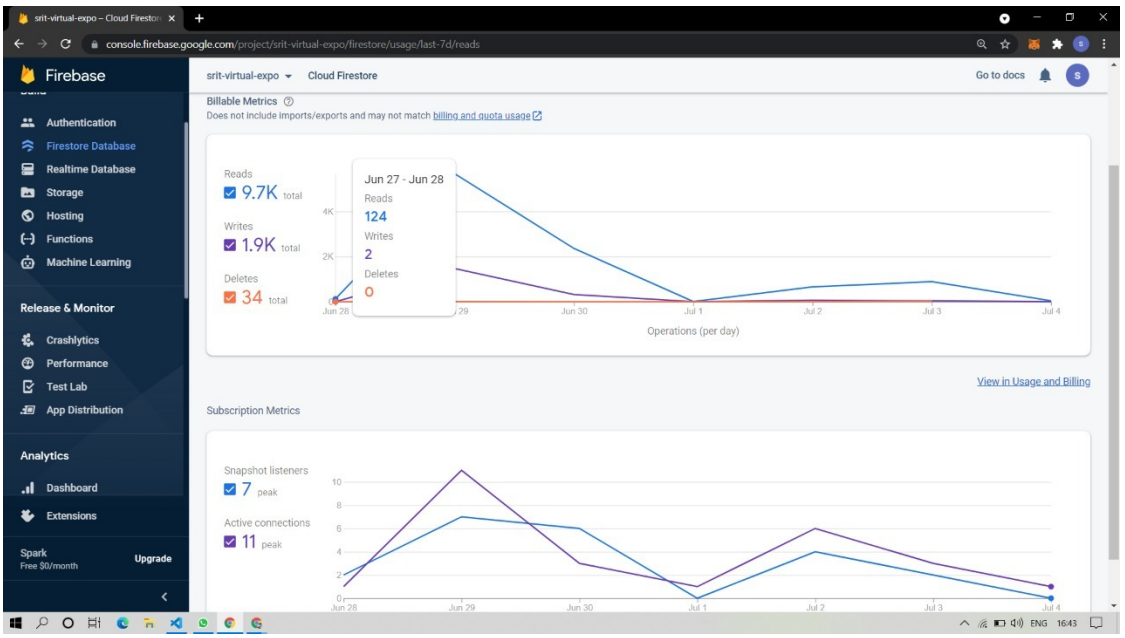


Fig.5.5. Firestore - Usage Statistics.

5.3.4 Analytics:

We also enabled and maintained the Analytics for both mobile and web apps which makes it easier to understand the traffic and usage of the applications.

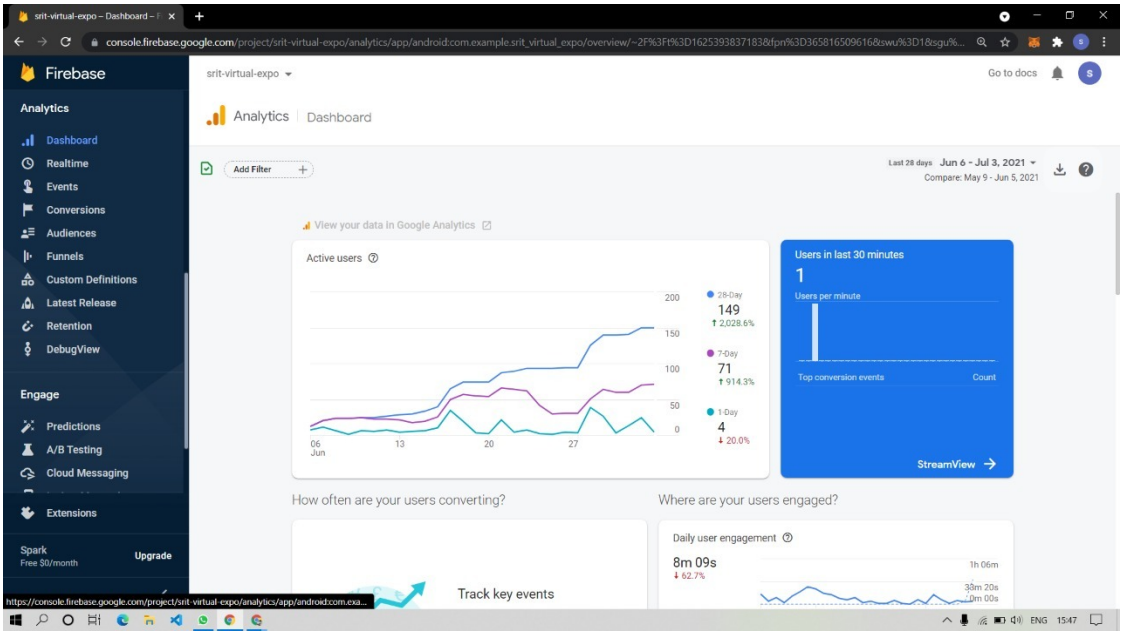


Fig.5.6. Firebase Analytics.

5.3.5 Cloud Messaging

In order to send the notifications and updates for the events in an Organization, we enabled the Firebase Cloud Messaging which helps our users to receive the updates and notifications.

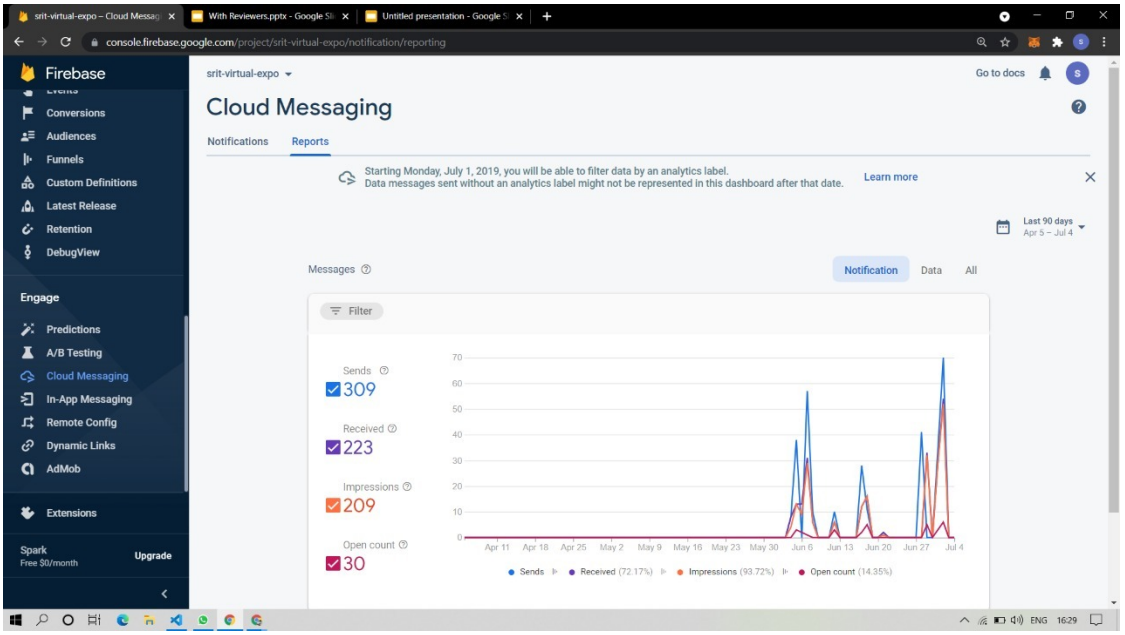


Fig.5.7. Firebase Cloud Messaging.

CHAPTER - 6
TESTING

6.1 Unit Testing

Unit testing involves the testing of each unit or an individual component of the software application. It is the first level of functional testing. The aim behind unit testing is to validate unit components with its performance.

A unit is a single testable part of a software system and tested during the development phase of the application software.

The purpose of unit testing is to test the correctness of isolated code. Aunit component is an individual function or code of the application. White box testing approach used for unit testing and usually done by the developers.

Whenever the application is ready and given to the Test engineer, he/she will start checking every component of the module or module of the application independently or one by one, and this process is known as Unit testing or components testing.

Some crucial reasons.

- Unit testing helps testers and developers to understand the base of code that makes them able to change defect causing code quickly.
- Unit testing helps in the documentation.
- Unit testing fixes defects very early in the development phase that's why there's a possibility to occur a smaller number of defects in upcoming testing levels.
- It helps with code reusability by migrating code and test cases .

6.2 Integration Testing

Integration testing is the second level of the software testing process after unit testing. In this testing, units or individual components of the software are testedina group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.

Unit testing uses modules for testing purposes, and these modules are combined and tested in integration testing. The Software is developed with a number of software modules that are coded by different coders or programmers. The goal of integration testing is to check the correctness of communication among all the modules.

Guidelines for Integration Testing

- ❖ We go for the integration testing only after the functional testing is completed for each module of the application.
- ❖ We always do integration testing by picking module by module so that a proper sequence is followed, and also we do not miss out on any integration scenarios.
- ❖ First, determine the test case strategy through which executable test cases can be prepared according to test data.
- ❖ Examine the structure and architecture of the application and identify the crucial modules to test them first and also identify all possible scenarios.
- ❖ Design test cases to verify each interface in detail.
- ❖ Choose input data for test case execution. Input data plays a significant role in testing.
- ❖ If we find any bugs then communicate the bug reports to developers and fix defects and retest.
- ❖ Perform positive and negative integration testing.
- ❖ Here positive testing implies that if a query is sent to the admin through contact form and if it reaches the admin it is positive testing or else it is negative testing.

6.3 Validation Testing

- The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements.
- Validation Testing ensures that the product actually meets the client's needs. It can also be defined as to demonstrate that the product fulfills its intended use

when deployed in an appropriate environment.

- It answers the question, Are we building the right product? **Validation Testing** .

Workflow

- Validation testing can be best demonstrated using V-Model. The Software/product under test is evaluated during this type of testing.
- Validation testing is testing where testers performed functional and functional testing. Here functional testing includes Unit Testing (UT), IntegrationTesting (IT) and System Testing (ST), and non-functional testing includes User acceptance testing (UAT).
- Validation testing is also known as dynamic testing, where we are ensuring that "we have developed the product right." And it also checks that the software meets the business needs of the client.

White box testing

White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing.

Testing

System testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer- based system. Ultimately, the software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.

User Acceptance Testing (UAT)

(UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done.

CHAPTER - 7
EXECUTION & RESULTS

Firstly, when the mobile application is initiated, If the user is not logged in, then he will be given two options either to Register or use the existing account and it is shown in the figure (Fig.7.1).

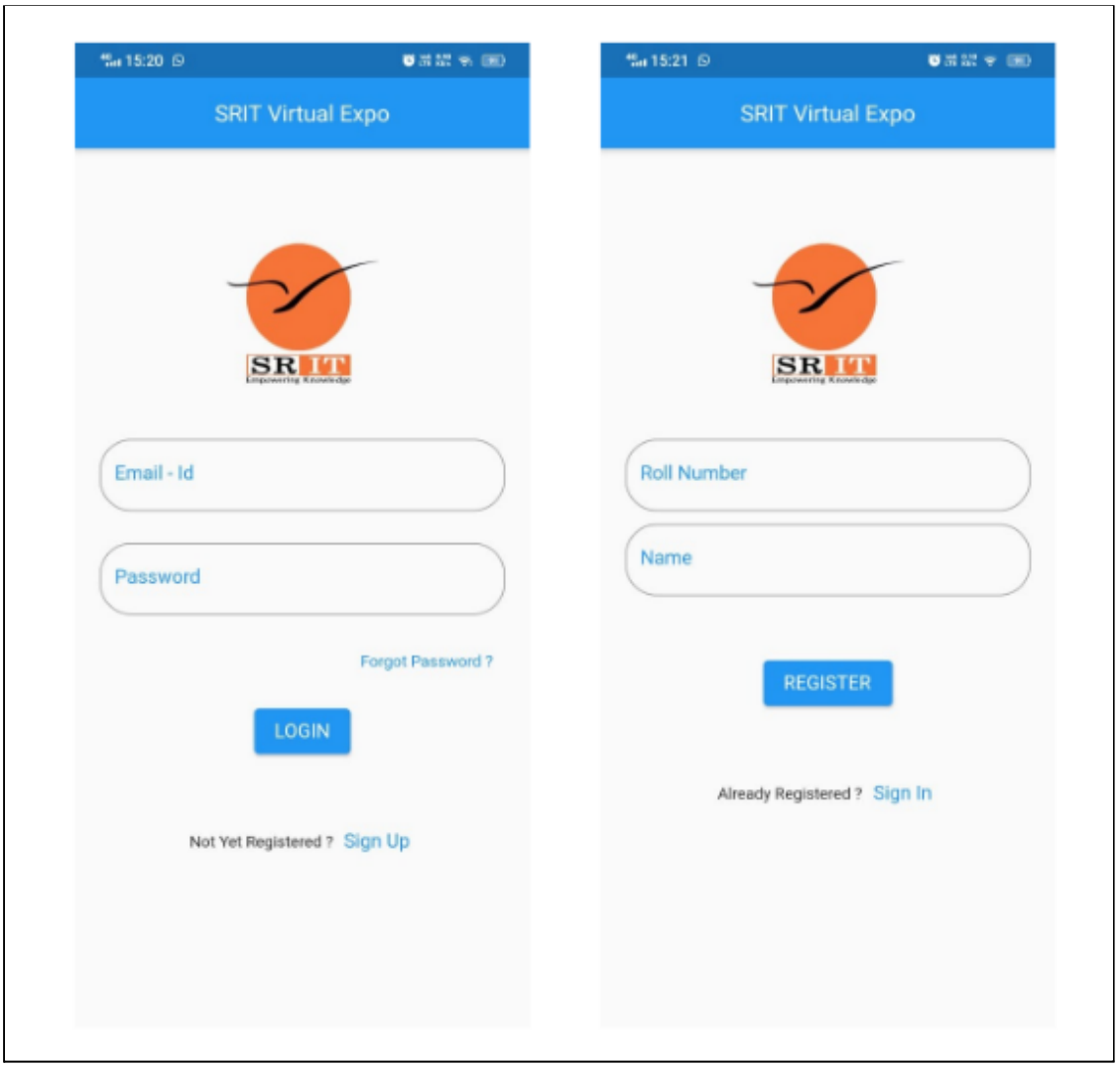


Fig.7.1. Mobile App's Signin & Signup Screen.

Right after the user logged in through the valid credentials the following figure appeared. This page shows the Expos, past contests, Live contests, Profile and Log out. Profile consists of the details of the users. It has the user Name, Roll number, Email id. It includes the settings option which allows the users to reset their passwords by getting a link to the registered mail.

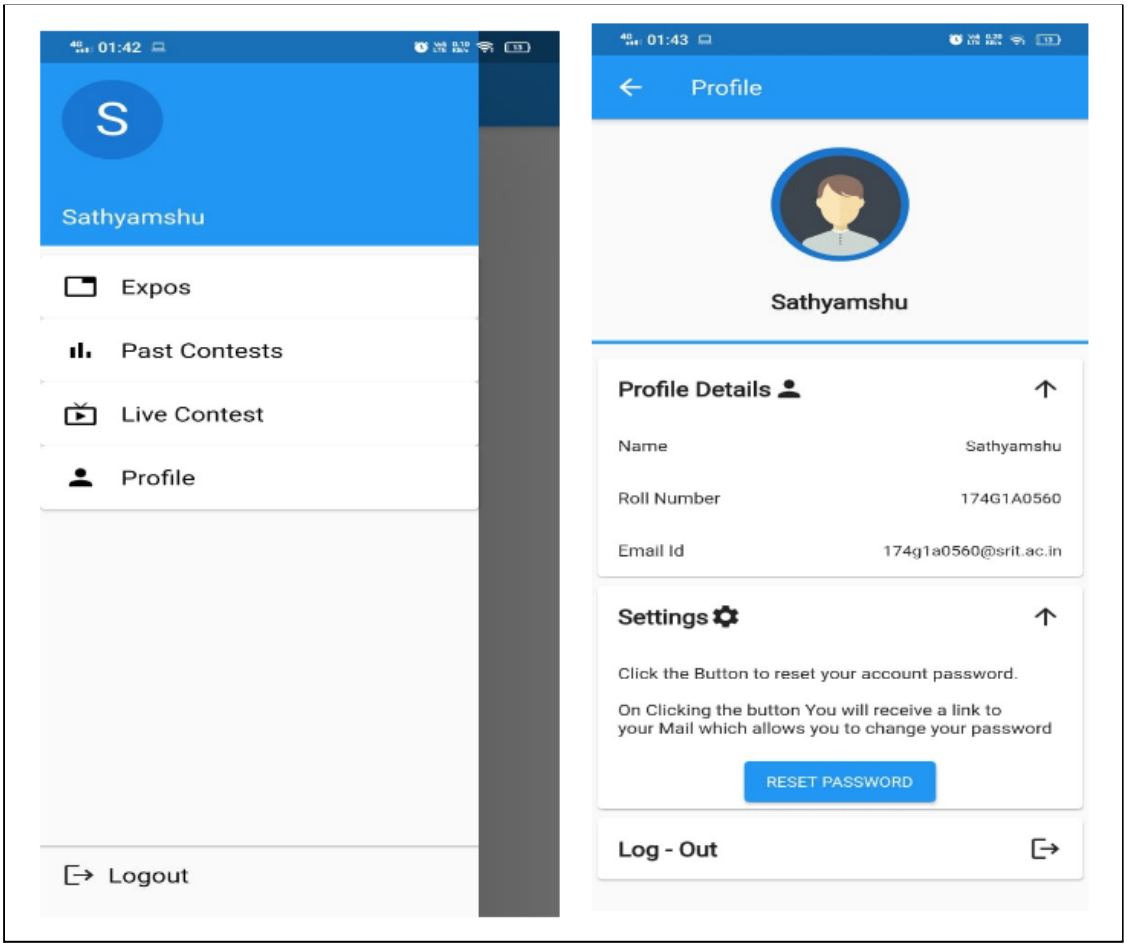


Fig.7.2. Mobile App's Menu bar & Profile Screen.

Coming to the Live contest Icon it consists of all the projects uploaded by the participants .It also provides the project members to upload the updated features respective with one's projects before the contest begins. It remembers the participants regarding the time of the contest through a timer.

Contest page consists of all the details regarding the projects uploaded by every single team with a like button beside it. By clicking on every single project it displays the project.

Fig.7.3. Live Contest Screen.

After clicking on the project it led us to the contest description page where all the details about the Projects were available such as Project, Team, Vote. The project can be displayed in the YouTube video link Format followed by the Description of the project. Every Team had an opportunity to describe their project in brief so that a public user can get an idea about the project.

Then we had the Team icon where it contains all the details of the individual project team members. It views the details of every single team member with the picture and their personal details such as Name, Roll number, Branch, Personal website/Blog, Github profile.

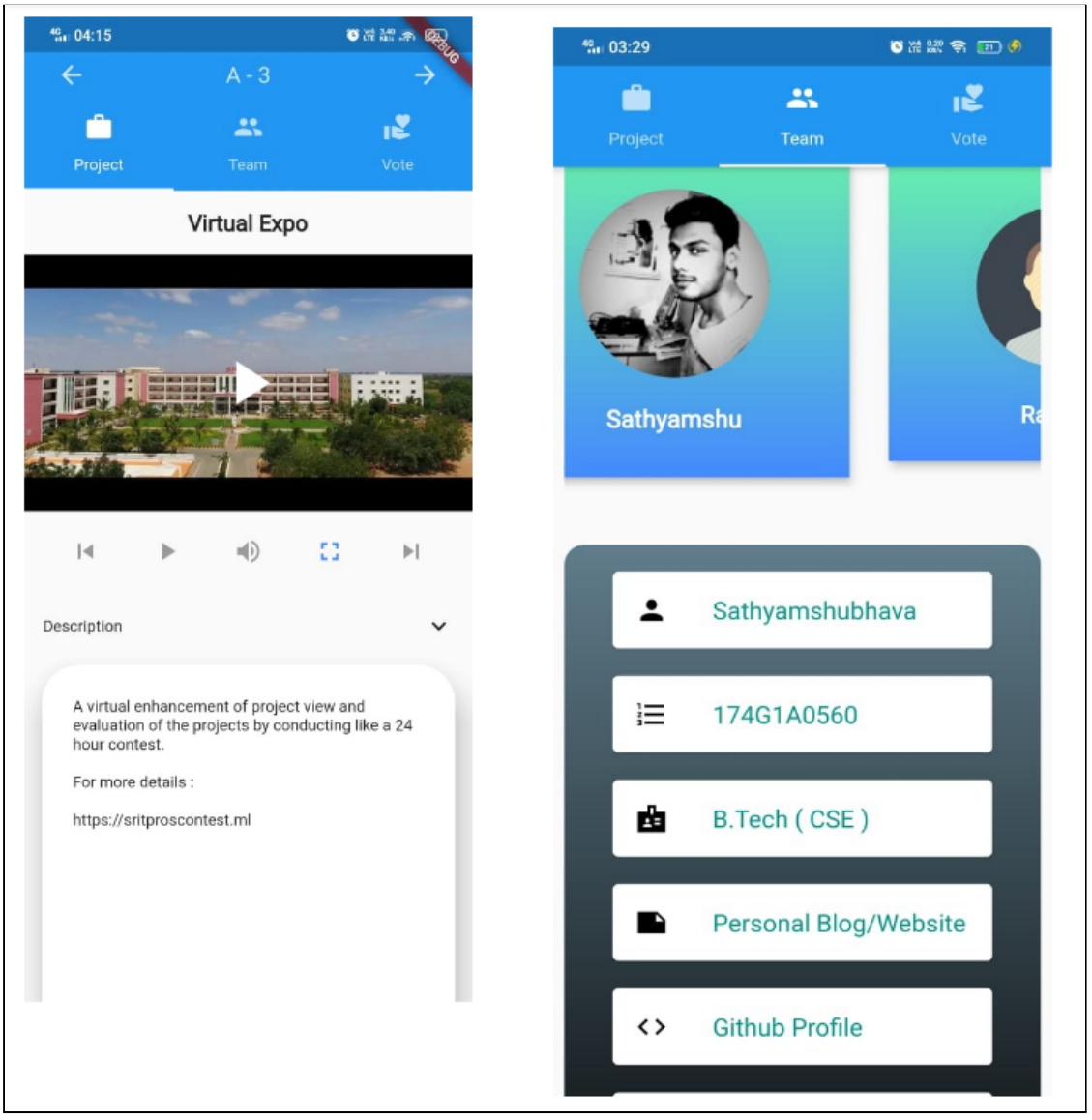


Fig.7.4. Project Details Screen.

As shown above, A project can be explained with a video and description there. Also It has the possibility to view the project teammates.

Right after viewing the projects users start to vote according to their own thoughts on the project. The voting session is allowed for all the users in the scale format. After that there would be a comment section which can be accessed only to the reviewers which allows them to write the views about the project.

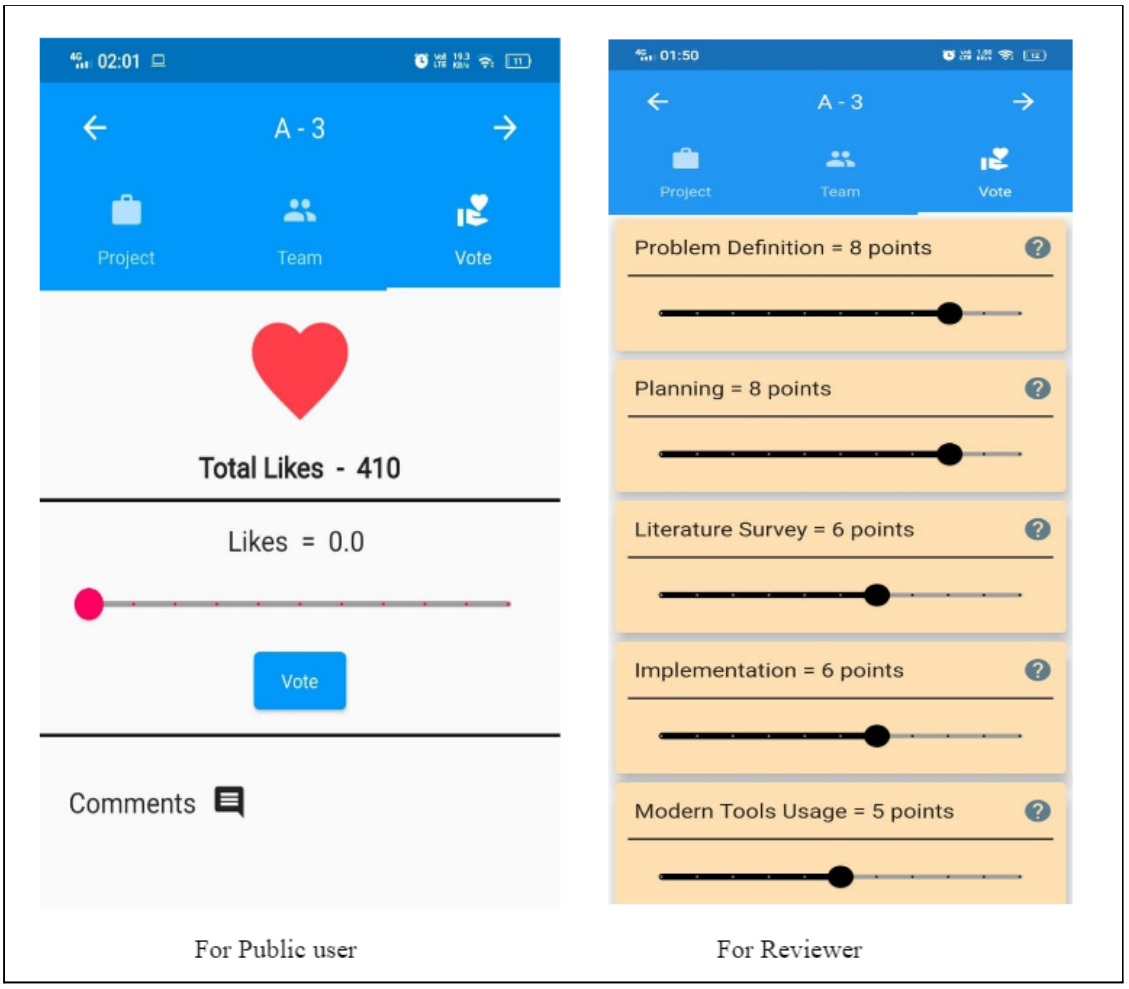


Fig.7.5. Public Vote V/s Reviewer Vote.

In this way, A Reviewer and a normal student (public user) can vote for a particular project in a contest within the stipulated time and also he has the possibility to change his points further but only before the contest ends.

In order to upload a project by a team, They need to log in through the web portal hosted at https://sritproscontest.ml. After logging into the app with valid credentials, the user will be on the homepage. Homepage consists of the options select

Expo name and Batch name. If the project member wants to change the project or edit it for updates then he can do that through Edit My project option below. After that they can logout through the logout option on the top.

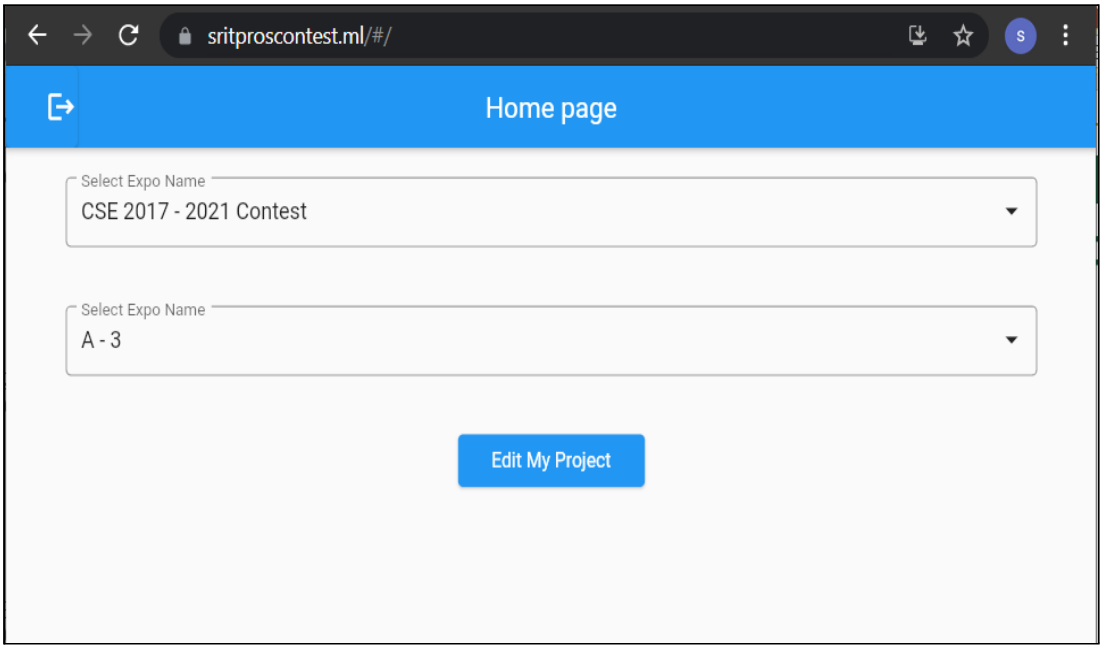


Fig.7.6. Project Upload Selection Screen.

After getting into the next page then every project member has to submit some of the details regarding the project as well as their personal details. This page enters details such as Project title, YouTube link to display their project video, Brief description about their project. Then coming to the personal details students should enter their id, roll no, profile link, Full name, personal blog, Github profile link, Linked in profile.

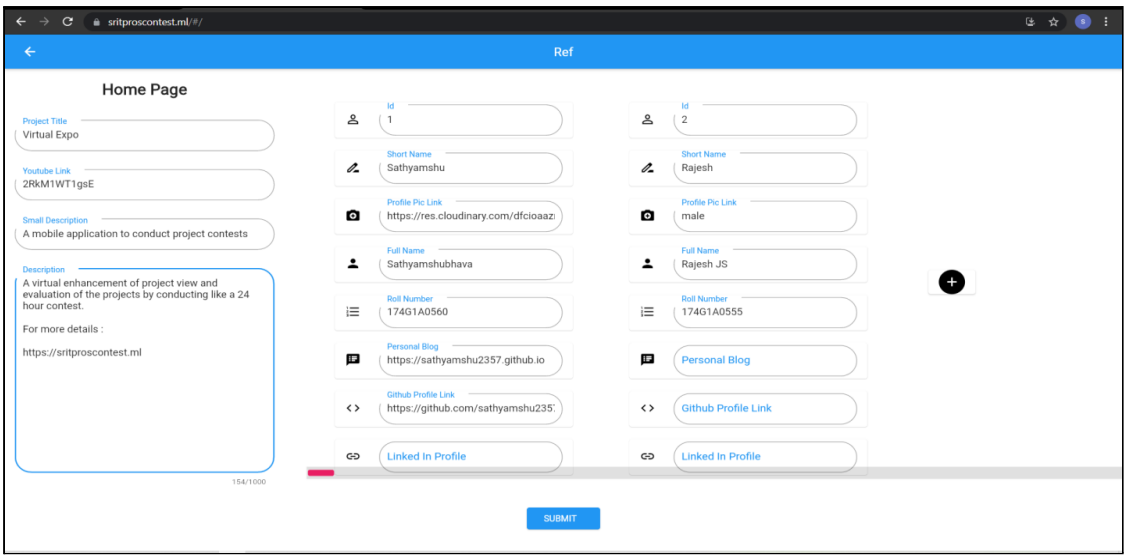


Fig.7.7. Project Upload Screen.

Right after submitting the details there will be an Admin portal which collects the details of the project with the following details. Name of the Expo, Description of the Expo, then batch details follow Name of the batch, Editor of the batch. Then the admin allows them to upload their project for the contest.

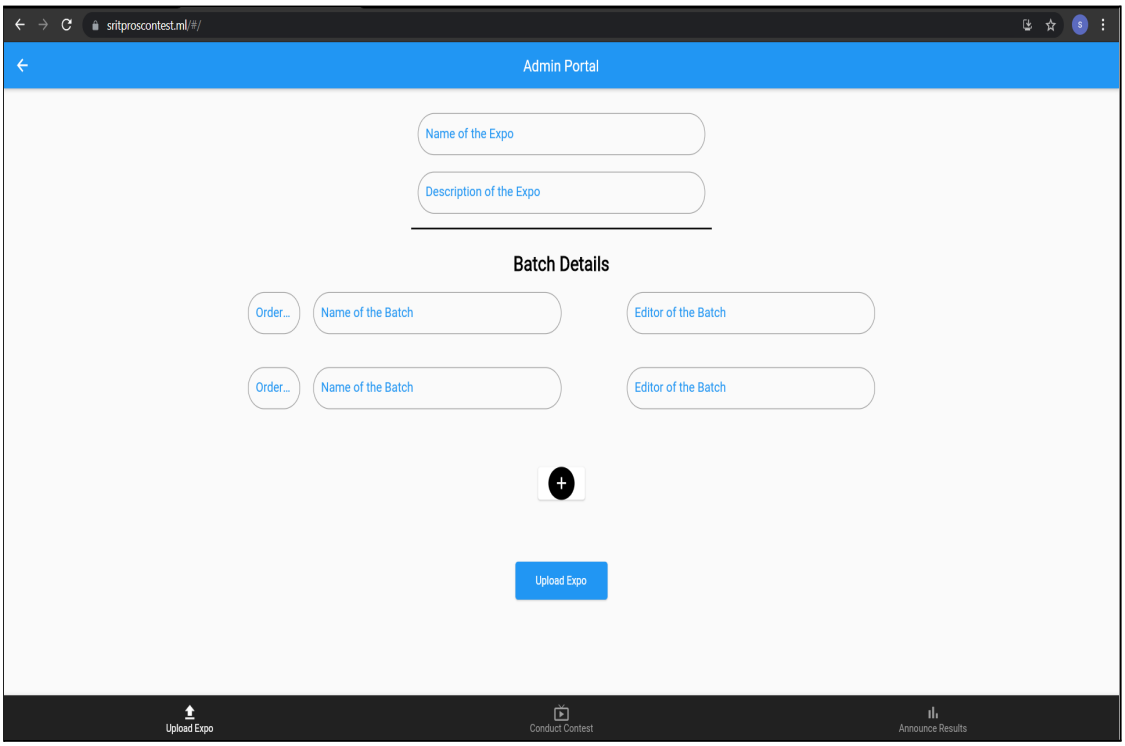


Fig.7.8. Expo Creation by Admin.

Admin conducts the contest and allows all the reviewers to watch out for the projects .Then the reviewers give their thoughts about the project through comments and vote for them. Once the contest is over Admin evaluates all the votes, Based on them release the final result. This contest is scheduled based on Considering all the aspects of the projects.

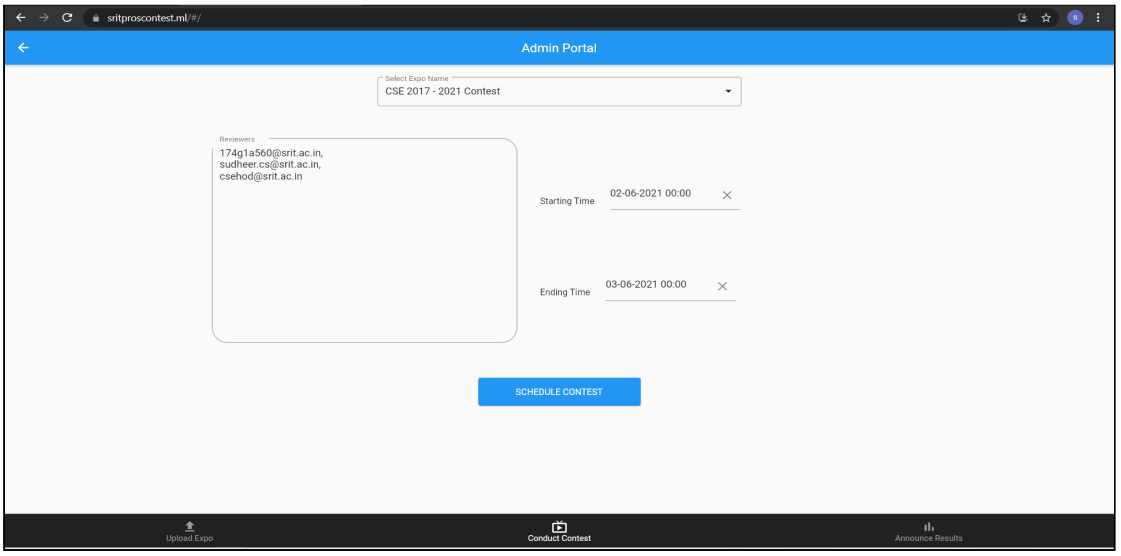
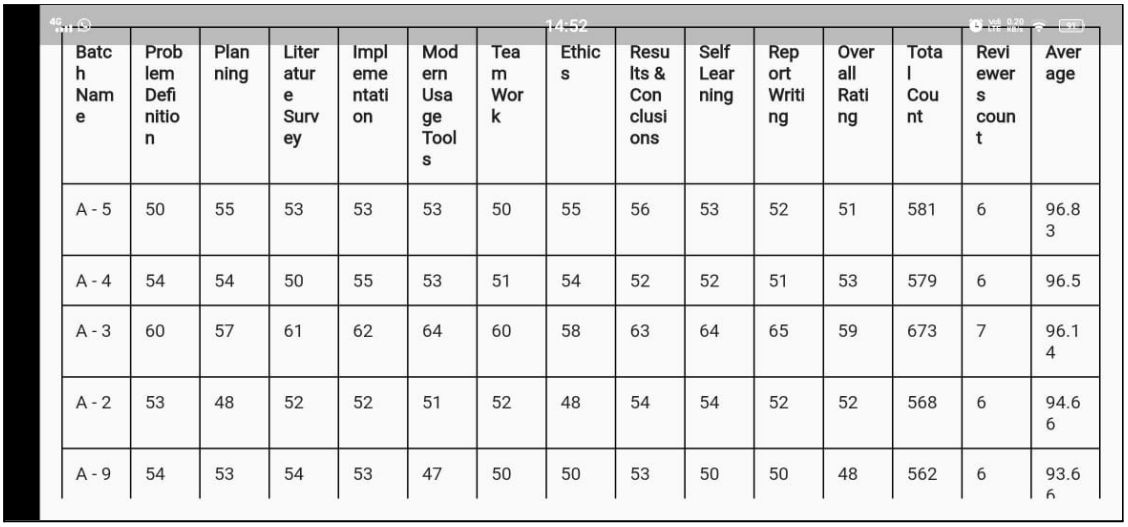


Fig.7.9. Conducting a Contest by Admin.

Once the contest is successfully completed, then the Admin will publish the results and these results can be viewed from the mobile app under the Past Contests section and it contains the winners of the contest.



Batch Name	Problem Definition	Planning	Literature Survey	Implementation	Modern Usage Tools	Team Work	Ethics	Results & Conclusions	Self Learning	Report Writing	Overall Rating	Total Count	Reviewers count	Average
A - 5	50	55	53	53	53	50	55	56	53	52	51	581	6	96.83
A - 4	54	54	50	55	53	51	54	52	52	51	53	579	6	96.5
A - 3	60	57	61	62	64	60	58	63	64	65	59	673	7	96.14
A - 2	53	48	52	52	51	52	48	54	54	52	52	568	6	94.66
A - 9	54	53	54	53	47	50	50	53	50	50	48	562	6	93.66

Fig.7.10. Results of the Contest.

Conclusion

An Individual Organization can showcase their students' projects which will be the examples for the future students of the Organization. Also they can conduct the project contest to evaluate the best among the best and the most useful project among the all. In this way a Student will learn how a project should be developed and knows the demand in the market for his knowledge. It lets even the other students to reach the students of their thoughts and interests and can work together.

There is also a possibility to collaborate with other Organizations and can share their projects with them and also can conduct a Inter - College - Contest. Collaboration with other Organizations increases the wide thinking of the students and lets them get knowledge on a lot more projects.

Bibliography

Books

[1] learn-google-flutter-fast-65-example-apps by Mark Clow.

Articles

[2] https://nikkitagandhi.medium.com/flutter-with-firebase-2bac25075683 .

[3] https://medium.com/47billion/how-to-use-firebase-with-flutter-e4a47a7470ce
.

Complete Documentation by Google

[4] https://flutter.dev .

[5] https://firebase.google.com/docs