**A**

**REPORT**

**On**

**“WEATHER APP”**

*Submitted in partial fulfillment of*

*the requirements for the 3rd semester sessional examination*

*of*

BACHELOR OF TECHNOLOGY

*In*

**COMPUTER SCIENCE & ENGINEERING**

By

Sushree Sarbani nanda(22CSE578)

Jyotirmayee Mishra(22CSE486)

***Under the supervision of***

**DR. SIBOPRASAD PATRO**

**(Assistant Professor CSE)**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**SCHOOL OF ENGINEERING AND TECHNOLOGY**

****

**GIET UNIVERSITY, GUNUPUR**

**2023-24**

** GIET UNIVERSITY**

***Gunupur-765022, At – Gobriguda, Po- Kharling, Dist. - Rayagada, Odisha***

***Contact:- +91 7735745535, 06857-250170,172,Visit us:-* www.giet.edu**

**Department of Computer Science & Engineering**

** CERTIFICATE**

## This is to certify that the project work entitled “WEATHER APP” is done by SUSHREE SARBANI NANDA & JYOTIRMAYEE MISHRA in partial fulfilment of the requirements for the 3RD Semester Sessional Examination of Bachelor of Technology in Computer Science and Engineering during the academic year 2023-24 This work is submitted to the department as a part of evaluation of 3rd Semester Learning Project.

***Dr. Sibo Prasad Patro Mr. SUDHEER PUNURI***

***Project Supervisor Class teacher***

***Mr. BHABANI SANKAR PRADHAN Dr .Sanjay Kumar Kuanar***

***Project Coordinator, 3rd Year HOD, CSE***

**ACKNOWLEDGEMENT**

We would like to thank my supervisor **Dr. Sibo Prasad Patro** for helping and providing the needful help whenever we needed on the making of this project.

Secondly, We would like to thank my class teacher Mr. Sudheer Punuri for guiding and providing an insight on the making of this project.

Their support and ideas helped us to design the working and Designing of this project. His constant support helped us to finalize the working of our project.

Thirdly, I would like to thanks our Project Coordinator Mr. Bhabani Sankar Pradhan for mentoring and guiding us throughout this project.

A special gratitude to our HOD **Dr. Sanjay Kumar Kuanar** for encouraging us throughout the project and rectifying our mistakes and led us in a perfect path to complete the project.

Thanks to friends and family members for being the backbone support during the entire project.

Sushree Sarbani Nanda

**TABLE OF CONTENTS**

**CAHPTER 1** 1-4

1.1 Abstract 1

1.2 Introduction 2

* Purpose 3
* Scope 3
* Features 4

**CHAPTER 2. SYSTEM ANALYSIS** 5-6

2.1 Hardware Requirements 5

2.2 Software Requirements 6

**CHAPTER 3** 7-12

3.1 Language Used 7

3.2 Library Used 8

**CHAPTER 4. SYSTEM DESIGN & SPECIFICATION** 13-21

4.1 FLOW CHART 13

4.2 ER DIAGRAM (LIBRARY) 14

4.3 SCREEN SHOTS 15-24

**CHAPTER 5. CODING** 25-40

**CHATER 6. TESTING** 41

**CONCLUSION** 42

**LIMITATION** 43

**REFERENCE** 44

**.**

**CHAPTER – 1**

**1.1 ABSTRACT**

College application is an cross-platform based application which is the new technical way to manage all department related jobs. It provides a simple interface for maintenance of student information. It can be used by educational institutes or colleges to maintain the records of students easily. The creation and management of accurate, up-to-date information regarding a students’ academic career is critically important in the university as well as colleges. Student information system deals with all kind of student details, academic related reports, college details, course details, curriculum and other resource related details too. It will also have faculty details, batch execution details, students’ details in all aspects, the various academic notifications to the staff and students updated by the college administration. It also facilitate us explore all the activities happening in the college, Different reports and Queries can be generated based on vast options related to students, batch, course, faculty, exams, semesters, certification and even for the entire college The placement officer is responsible for updating the placement related information like eligible criteria for a particular company, arriving date for the company which is coming for recruitment, the list of students who are eligible for attending the recruitment process. E-Library (also referred to as digital library) is a special library with a focused collection of digital object.

**1.2 INTRODUCTION**

The design and implementation of the system is to provide service in institute and colleges. The system is to provide comprehensive student information system and user interface is to replace the current paper records. College Staff uploads attendance, results, share subject notes and college notifications through a secure, online interface using android devices. All data will be stored on the collage server and validated on the server before actual record alteration occurs. The system plans for student user interface, allowing students to access tips and tricks as provided by their seniors. All data is stored securely on SQL servers managed by the college Administrator. This system will decrease the paperwork and time needed to access student records. Previously, college relied heavily on paper records for this initiative which had it’s own disadvantages. This system provides a simple interface for the maintenance of student information. It can be used by educational institutes or colleges to maintain the records of students easily. Achieving this objective is difficult using a manual system as the information are scattered, can be redundant and collecting relevant information may be very time consuming. Our proposed system ensures to overcome these limitations. . Online Attendance and Feedback System is software developed for daily student attendance in schools, colleges and institutes. If facilitates to access the information of a particular student in a particular class. There is another part which is feedback, the student can give the feedback at anytime from anywhere to faculty. This feedback can be reviewed by the admin or the management committee of the institute through which the confidentiality of the feedback of the faculty can be maintained. This application is developed for daily student attendance in colleges and institutes. The teachers can send the attendance summary and feedback about the students to their parents. It can also be useful in an organization or company at a certain limit not the whole application

**1.2.1 PURPOSE**

The system which is used nowadays has some drawbacks which need to be improved for better performance. The system through which the feedback is taken is not good enough. The views of each and every student are not expressed through these systems. As the technology is developed day by day we need to use this technology so we can get an efficient result in adequate time. For attendance management in the present system all work is done on paper. The whole session attendance is stored in register and at the end of the session the reports are generated. We are not interested in generating report in the middle of the session or as per the requirement because it takes more time in calculation. At the end of session the students who don’t have 75% attendance get a notice. This is a very time consuming process. In the present system the result is viewed on the notice board. It requires lot of paperwork and is time consuming. Moreover, there is no system still present through which students can take advice from senior students. College cannot even provide urgent notifications to students in case of emergency**.**

The system architecture has a smart phone with android OS, a database server and the user as its components. The android smart phone or tablet must use 3G,4G or Wi-Fi network for internet connectivity to ensure better performance The user will login to the application through an android smart phone. The user-type is verified with the database server and access is given to the appropriate user. The web application also can be used to login and perform certain operations such as ensures that internet is on. In this module, application generates PDF file dynamically using java program.

**1.2.2 SCOPE**

This project deals with the various functioning in College management process. The main idea is to implement a proper process to system. In our existing system contains a many operations registration, student search, fees, attendance, exam records, performance of the student etc. All these activity takeout manually by administrator.

In our model, it deals with the Operations in system. For example, when students fill the registration form the record is stored in the database. And display the details of student is perform by retrieving information from database table.

* + 1. **FEATURES**

1. **VIEW ATTENDANCE :**

Students can view attendance uploaded by the faculty or admin. They can check the attendance at any time but they cannot manipulate the data.

1. **FEE PAYMENT SYSTEM :**

Student can pay fee through this app which reduce the rush in admin block. It is based on Razorpay Payment gateway which provides secure transections.

1. **ONLINE NOTES :**

Student can download class notes in the form of pdf through this app. It contains all departments handwritten notes of all subjects.

1. **NOTICE BOARD :**

Notices are sent to the students by HOD or admin. Useful information, college notices, important announcements are received on students registered phone. They can view it anytime.

1. **LIBRARY :**

Student can add books to their bucket, student can update books without physically report in Library.

**CHAPTER – 2**

**SYSTEM ANALYSIS**

**2.1 HARDWARE REQUIREMENT**

**Mobile Phone :**

We need android phone or ios based phone to run android apps which we are building

there is no specific requirements of mobile phone to run android apps.

Min android version supported – 6

Min RAM - 512 MB

**LAPTOP/PC :**

**Operating System Version –**

Microsoft Windows 7/8/10 (32- or 64-bit)

The Android Emulator only supports 64-bit Windows**.**

**RAM –**

4 GB RAM minimum; 8 GB RAM recommended.

**Free digital storage –**

2 GB of available digital storage minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image).

**2.2 SOFTWARE REQUIREMENT**

1. **ANDROID STUDIO :**

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems or as a subscription-based service in 2020

1. **VS CODE :**

Visual Studio Code is an editor that favors simplicity over having an endless assortment of bells and whistles. Most of the functionality is exposed by typing in the thing you need into the top search bar. Everything you would expect to be there is there: debugging, breakpoints, etc.

1. **EMULATOR :**

The Android Emulator simulates Android devices on your computer so that you can test your application on a variety of devices and Android API levels without needing to have each physical device. The emulator provides almost all of the capabilities of a real Android device. You can simulate incoming phone calls and text messages, specify the location of the device, simulate different network speeds, simulate rotation and other hardware sensors, access the Google Play Store, and much more.

1. **AVD MANAGER :**

AVD is an emulator configuration that simulates a physical Android device. This makes it possible to run and test your app in a variety of configurations that simulate different physical Android devices

**CHAPTER – 3**

**3.1 LANGUAGE USED** / Emulator

Mob Mobile Phone-> The Android Emulator s

**DART :**

The Dart language is type safe; it uses static type checking to ensure that a variable’s value always matches the variable’s static type. Sometimes, this is referred to as sound typing. Although types are mandatory, type annotations are optional because of type inference. The Dart typing system is also flexible, allowing the use of a dynamic type combined with runtime checks, which can be useful during experimentation or for code that needs to be especially dynamic.

Dart offers sound null safety, meaning that values can’t be null unless you say they can be. With sound null safety, Dart can protect you from null exceptions at runtime through static code analysis. Unlike many other null-safe languages, when Dart determines that a variable is non-nullable, that variable is always non-nullable. If you inspect your running code in the debugger, you’ll see that non-nullability is retained at runtime (hence sound null safety).

The following code sample showcases several Dart language features, including libraries, async calls, nullable and non-nullable types, arrow syntax, generators, streams, and getters. To find examples of using additional Dart features, see the samples page. To learn more about the language, take the Dart language tour.

Dart’s compiler technology lets you run code in different ways:

- **Native platform**: For apps targeting mobile and desktop devices, Dart includes both a Dart VM with just-in-time (JIT) compilation and an ahead-of-time (AOT) compiler for producing machine code.

**- Web platform**: For apps targeting the web, Dart includes both a development time compiler (dartdevc) and a production time compiler (dart2js). Both compilers translate Dart into JavaScript

that you can test your application on a variety of devices an

**3.2 LIBRARIES USED**

* **cupertino\_icons : ^1.0.2**

1. Identifiers for the supported Cupertino icons.
2. Use with the Icon class to show specific icons.
3. Icons are identified by their name as listed below.
4. To use this class, make sure you add a dependency on Cupertino icons in your project's pubspec.yaml file. This ensures that the Cupertino Icons font is included in your application. This font is used to display the icons.

* **firebase\_core : ^1.7.0**

The firebase\_core plugin is responsible for connecting your Flutter app to your Firebase project. The plugin must be installed and initialized before the usage of any other FlutterFire plugins. It provides basic functionality such as:

* Initializing FlutterFire.
* Creating Secondary Firebase App Instances.

FlutterFire requires a default Firebase app to be present before initialization, otherwise an exception will be thrown. The steps for setting up a default app for your platform can be found in the Getting Started documentation. Some plugins such as Analytics & Performance Monitoring are only compatible with the default Firebase app, however, plugins such as Authentication can take advantage of Secondary Firebase Apps, allowing you to use multiple Firebase projects at once.

**To access the default app, call the initializeApp or app method on the Firebase class:**

**FirebaseApp defaultApp = await Firebase.initializeApp();**

// or

**FirebaseApp defaultApp = Firebase.app();**

* **firebase\_database :**

Firestore is a flexible, scalable NoSQL cloud database to store and sync data. It keeps your data in sync across client apps through realtime listeners and offers offline support so you can build responsive apps that work regardless of network latency or Internet connectivity.

To start using the Cloud Firestore package within your project, import it at the top of your project files:

**import 'package:cloud\_firestore/cloud\_firestore.dart';**

Before using Firestore, you must first have ensured you have initialized FlutterFire.

To create a new Firestore instance, call the instance getter on FirebaseFirestore:

**FirebaseFirestore firestore = FirebaseFirestore.instance;**

By default, this allows you to interact with Firestore using the default Firebase App used whilst installing FlutterFire on your platform. If however you'd like to use Firestore with a secondary Firebase App, use the instanceFor method:

**FirebaseApp secondaryApp = Firebase.app('SecondaryApp');**

**FirebaseFirestore firestore = FirebaseFirestore.instanceFor(app: secondaryApp);**

Firestore stores data within "documents", which are contained within "collections". Documents can also contain nested collections. For example, our users would each have their own "document" stored inside the "Users" collection. The collection method allows us to reference a collection within our code.

* **firebase\_auth: ^3.1.3**

Firebase Authentication provides backend services & easy-to-use SDKs to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.

Firebase Auth provides many methods and utilities for enabling you to integrate secure authentication into your new or existing Flutter application. In many cases, you will need to know about the authentication state of your user, such as whether they're logged in or logged out.

Firebase Auth enables you to subscribe in realtime to this state via a Stream. Once called, the stream provides an immediate event of the user's current authentication state, and then provides subsequent events whenever the authentication state changes.

* **flutter\_secure\_storage: ^4.2.1**

A Flutter plugin to store data in secure storage:

Keychain is used for iOS

AES encryption is used for Android. AES secret key is encrypted with RSA and RSA key is stored in KeyStore

lib secret is used for Linux.

Note KeyStore was introduced in Android 4.3 (API level 18). The plugin wouldn't work for earlier versions.

* **flutter\_svg: ^0.23.0+1**

An SVG rendering and widget library for Flutter, which allows painting and displaying Scalable Vector Graphics 1.1 files.

Basic usage (to create an SVG rendering widget from an asset):

final String assetName = 'assets/image.svg';

final Widget svg = SvgPicture.asset(

assetName,

semanticsLabel: 'Acme Logo'

* **webview\_flutter: ^2.1.2**

A Flutter plugin that provides a WebView widget.

On iOS the WebView widget is backed by a [WKWebView](https://developer.apple.com/documentation/webkit/wkwebview); On Android the WebView widget is backed by a web view .

* **google\_fonts :**

The google\_fonts package for Flutter allows you to easily use any of the 977 fonts (and their variants) from font.google.com in your Flutter app.

With the google\_fonts package, .ttf or .otf files do not need to be stored in your assets folder and mapped in the pubspec. Instead, they can be fetched once via http at runtime, and cached in the app's file system. This is ideal for development, and can be the preferred behavior for production apps that are looking to reduce the app bundle size. Still, you may at any time choose to include the font file in the assets, and the Google Fonts package will prioritize pre-bundled files over http fetching. Because of this, the Google Fonts package allows developers to choose between pre-bundling the fonts and loading them over http, while using the same API

* **http:**

A composable, Future-based library for making HTTP requests. This package contains a set of high-level functions and classes that make it easy to consume HTTP resources. It's multi-platform, and supports mobile, desktop, and the browser

* **shared\_preferences: ^2.0.8 :**

Wraps platform-specific persistent storage for simple data (NSUserDefaults on iOS and macOS, Shared Preferences on Android, etc.). Data may be persisted to disk asynchronously, and there is no guarantee that writes will be persisted to disk after returning, so this plugin must not be used for storing critical data.

* **cloud\_firestore: ^2.5.4 :**

Cloud Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud. Like Firebase Realtime Database, it keeps your data in sync across client apps through realtime listeners and offers offline support for mobile and web so you can build responsive apps that work regardless of network latency or Internet connectivity. Cloud Firestore also offers seamless integration with other Firebase and Google Cloud products, including Cloud Functions.

* **ionicons: ^0.1.2 :**

This package includes 1332 icons of [Ionicons](https://ionicons.com/) [v5.4.0](https://github.com/ionic-team/ionicons/releases/tag/v5.3.0). The naming convention is the same as the CSS names, all dashes replaced with underscores.

* **animations: ^2.0.2 :**

This package contains pre-canned animations for commonly-desired effects. The animations can be customized with your content and dropped into your application to delight your users.

* **razorpay\_flutter: ^1.2.7 :**

Razor Pay is one of the best payment gateways which provided plugin in a flutter. It was the most straightforward to integrate in-product payment widget that kicked us off with Razorpay yet it was the quick and dependable help for both issue settlement and on-boarding new highlights that have kept us cooperating with them! .

Sign up for a Razorpay account.

Generate API Keys in Test Mode. Once you are done with the integration, you can generate Live Mode API keys and replace them in the integration.

Read our Payment Flow document before proceeding with the integration.

Integration Steps :

Follow these steps to integrate your Flutter application with the Razorpay Payment Gateway:

* Install the Razorpay Flutter Plugin
* Add the Dependencies
* Import Package
* Create Razorpay instance
* Attach Event Listeners
* Create an Order on Server-side
* Add Checkout Options
* Open Checkout
* Store Fields in Server
* Verify Payment Signature
* **hexcolor: ^2.0.5 :**

hex color plugin allows you to add hex color codes to your flutter projects

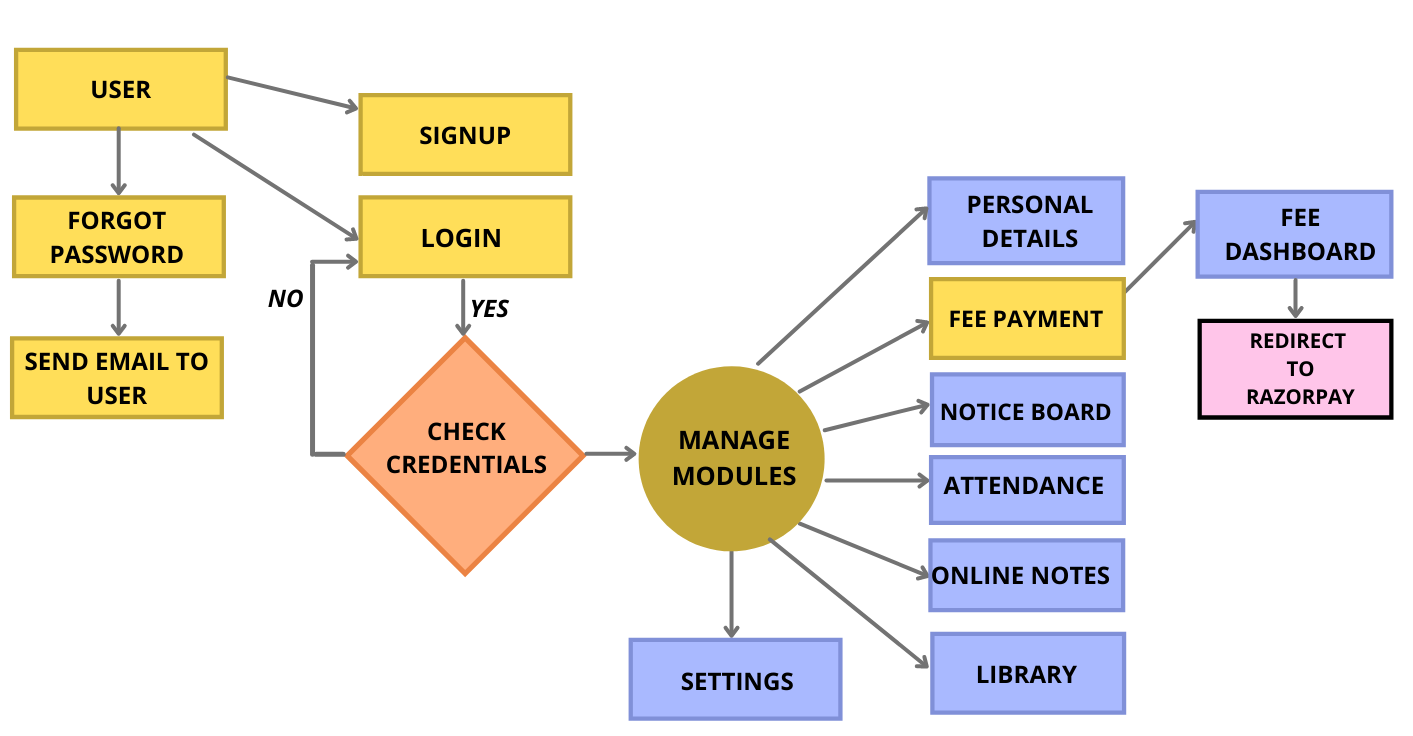
* **bottom\_navy\_bar: ^6.0.0 :**

A beautiful and animated bottom navigation. The navigation bar use your current theme, but you are free to customize it.

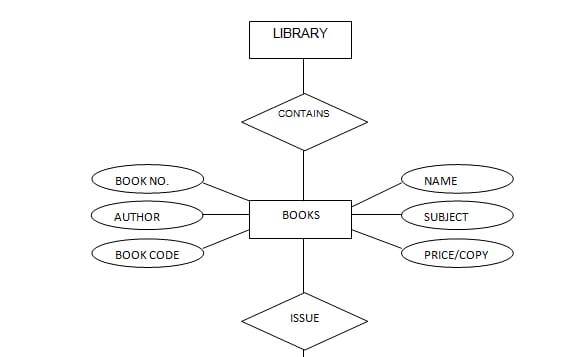
**CHAPTER – 4**

## **SYSTEM DESIGN & SPECIFICATION**

**4.1 FLOW CHART**



**4.2** **ER DIAGRAM (LIBRARY)**



* 1. **SCREEN SHOTS**

|  |  |
| --- | --- |
|  |  |

**( 1st SCREEN WITH ANIMATION ) (ASKING FOR SIGN IN / SIGN UP)**

**SHOT -1 SHOT -2**

**(SIGN UP PAGE) (LOGIN PAGE)**

**SHOT - 3 SHOT - 4**

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

**(RESET PASSWORD) (ACCOUNT RECOVERY E-MAIL)**

**SHOT -5 SHOT -6**

|  |  |
| --- | --- |
|  |  |

**(MAIN DASHBOARD) (PERSONAL INFORMATION)**

**SHOT – 7 SHOT - 8**

|  |  |
| --- | --- |
|  |  |

**(FEE PAYMENT PAGE) (MODE OF PAYMENTS)**

**SHOT – 9 SHOT – 10**

|  |  |
| --- | --- |
|  |  |

**(ONLINE NOTES SIGN UP/ SIGN IN PAGE) (NOTICE BOARD)**

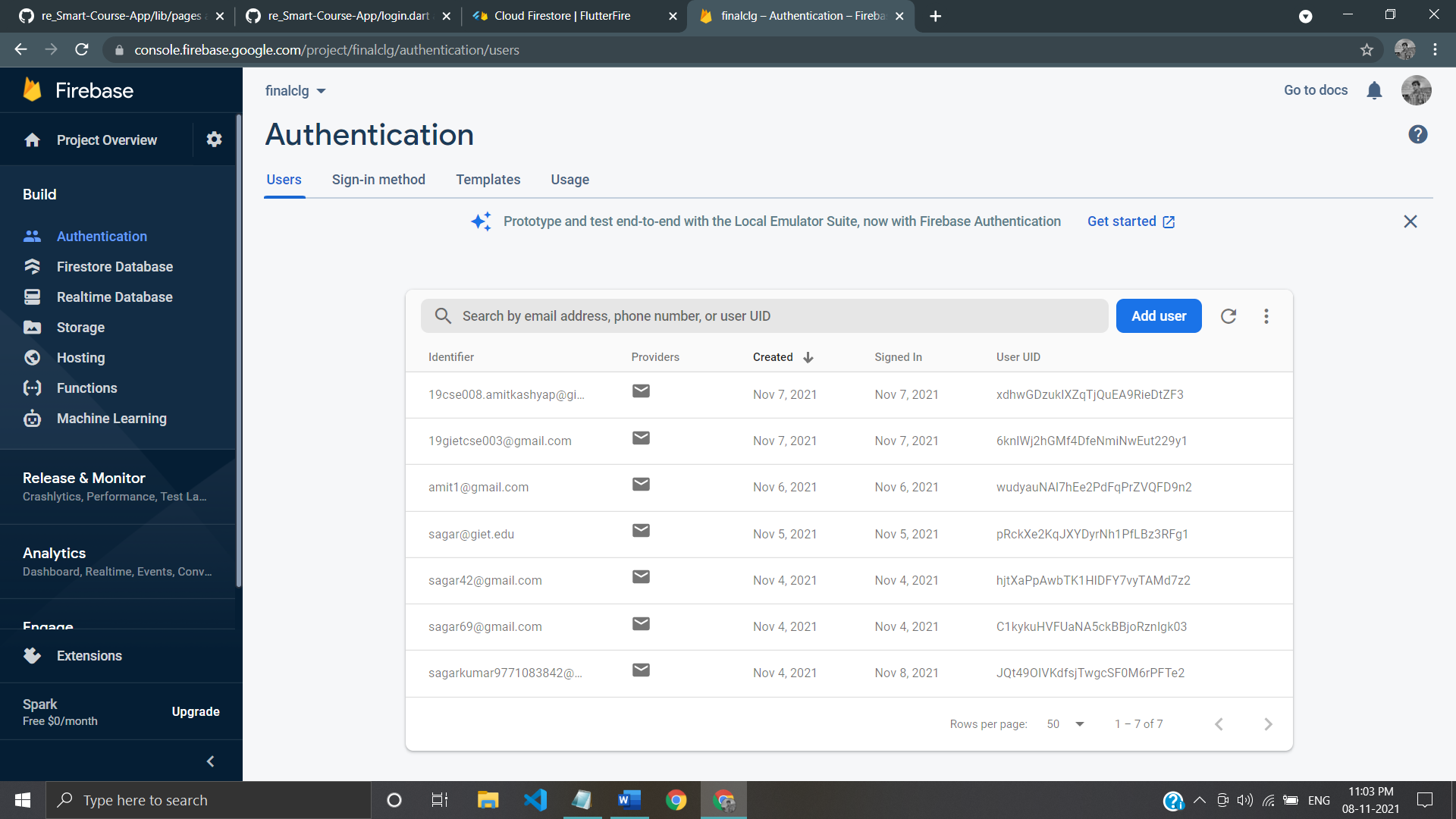
**SHOT – 11 SHOT – 12**

|  |  |
| --- | --- |
|  |  |

**(ATTENDANCE LOGIN) (DEVELOPERS PROFILE)**

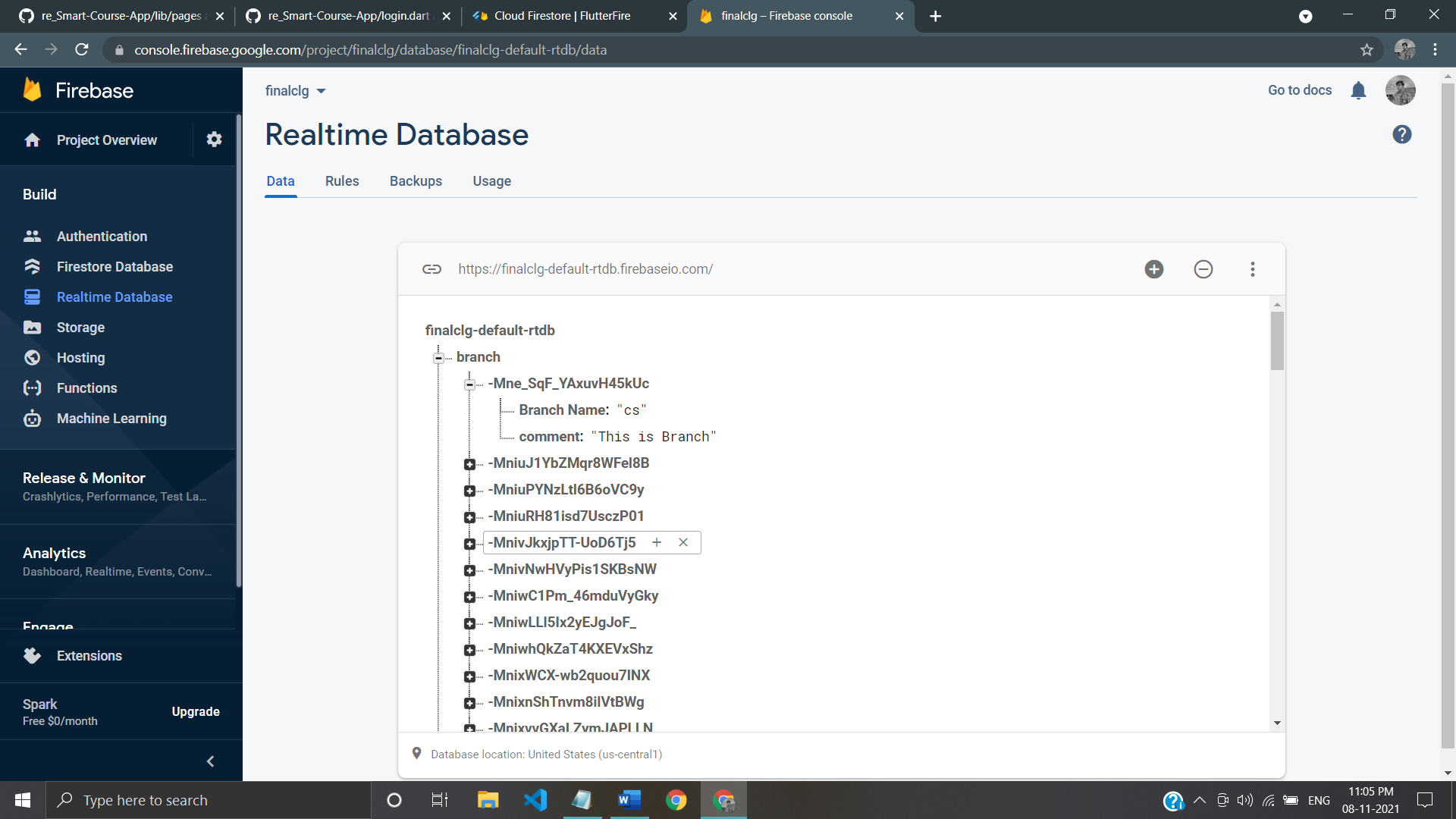
**SHOT – 13 SHOT - 14**

* 1. **DATABASE SCREEN SHOTS**



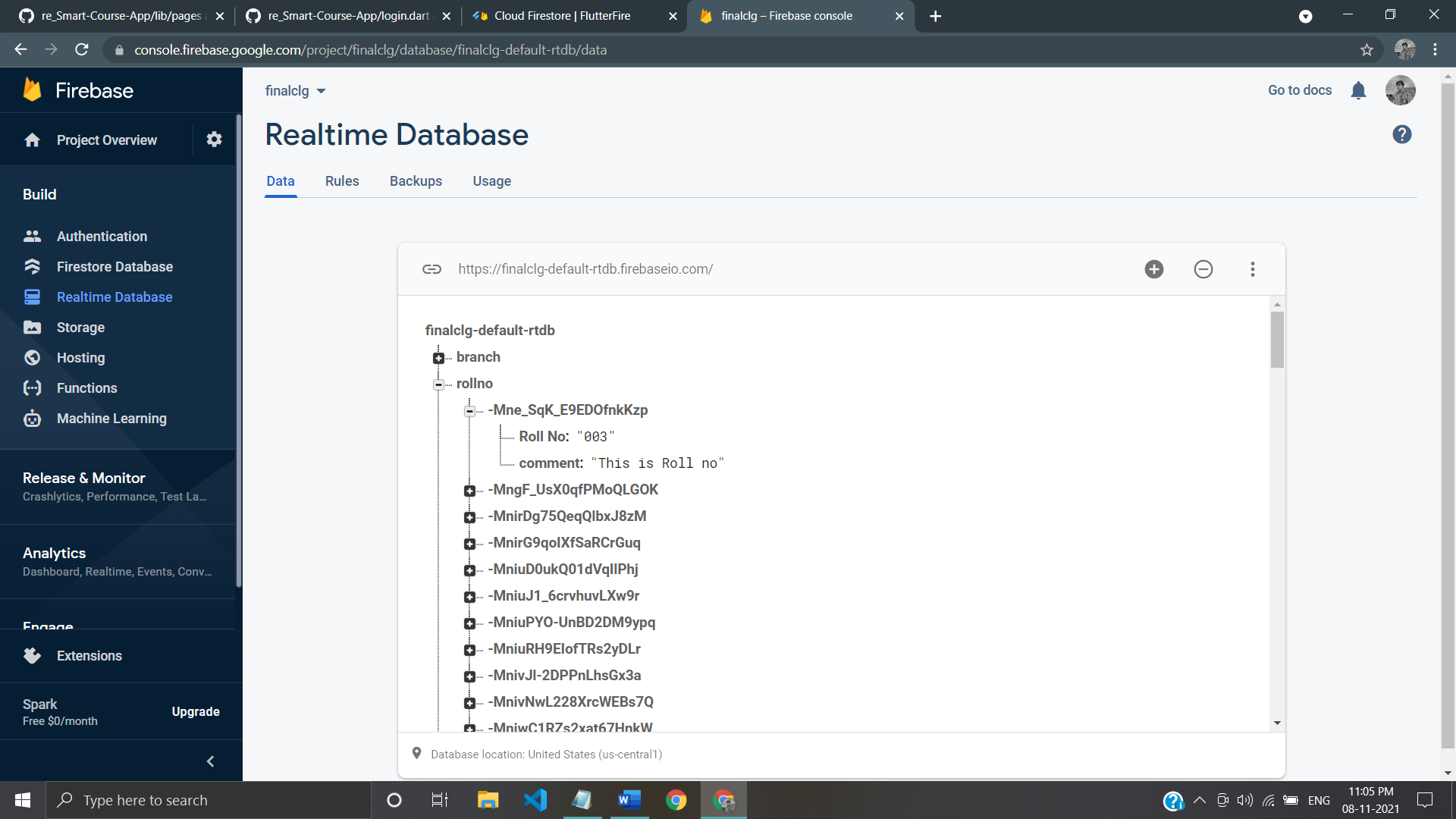
**(USER AUTHENTICATION)**

**SHOT -15**



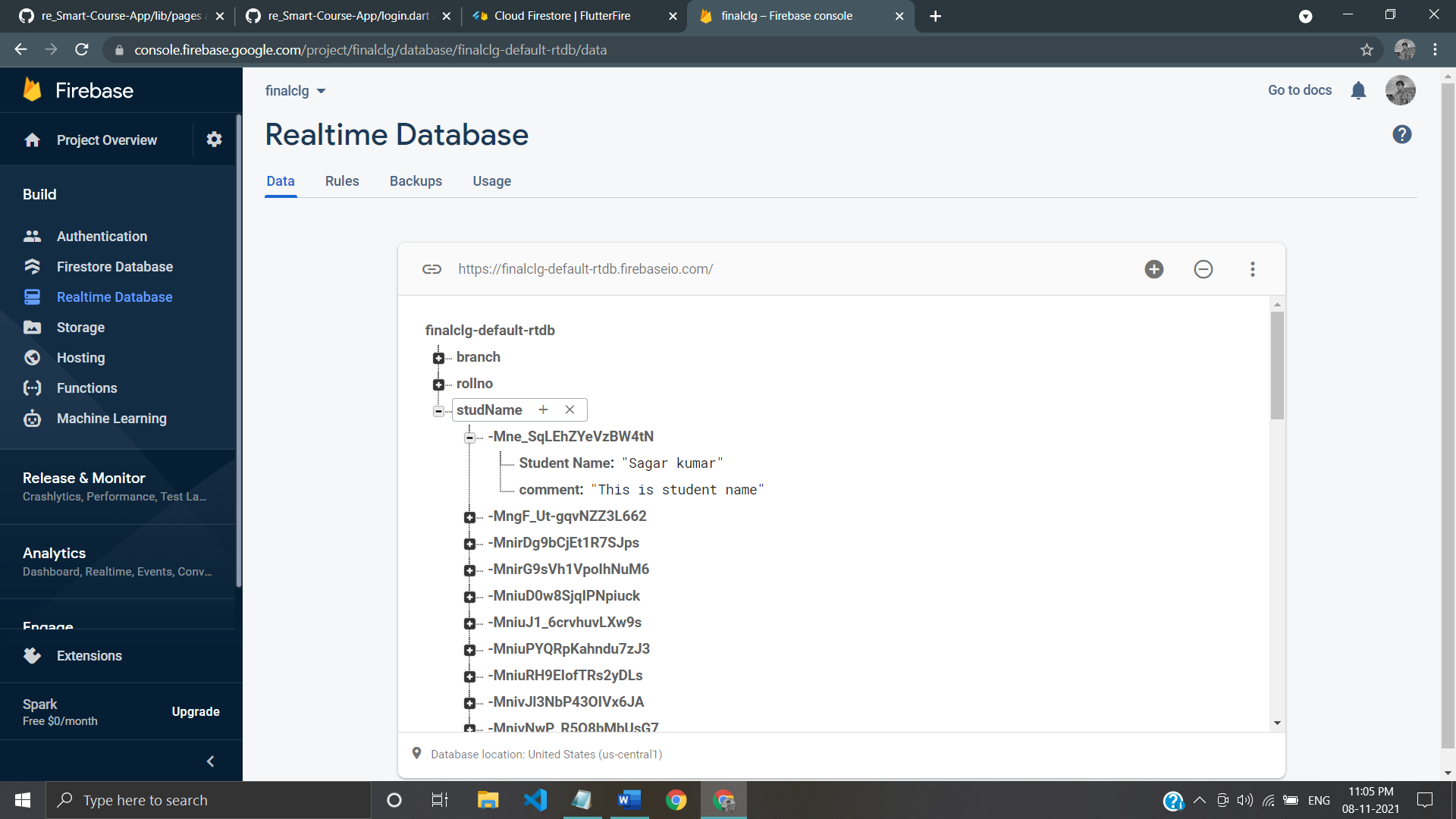
**(REALTIME DATABASE FOR BRANCH)**

**SHOT – 16**



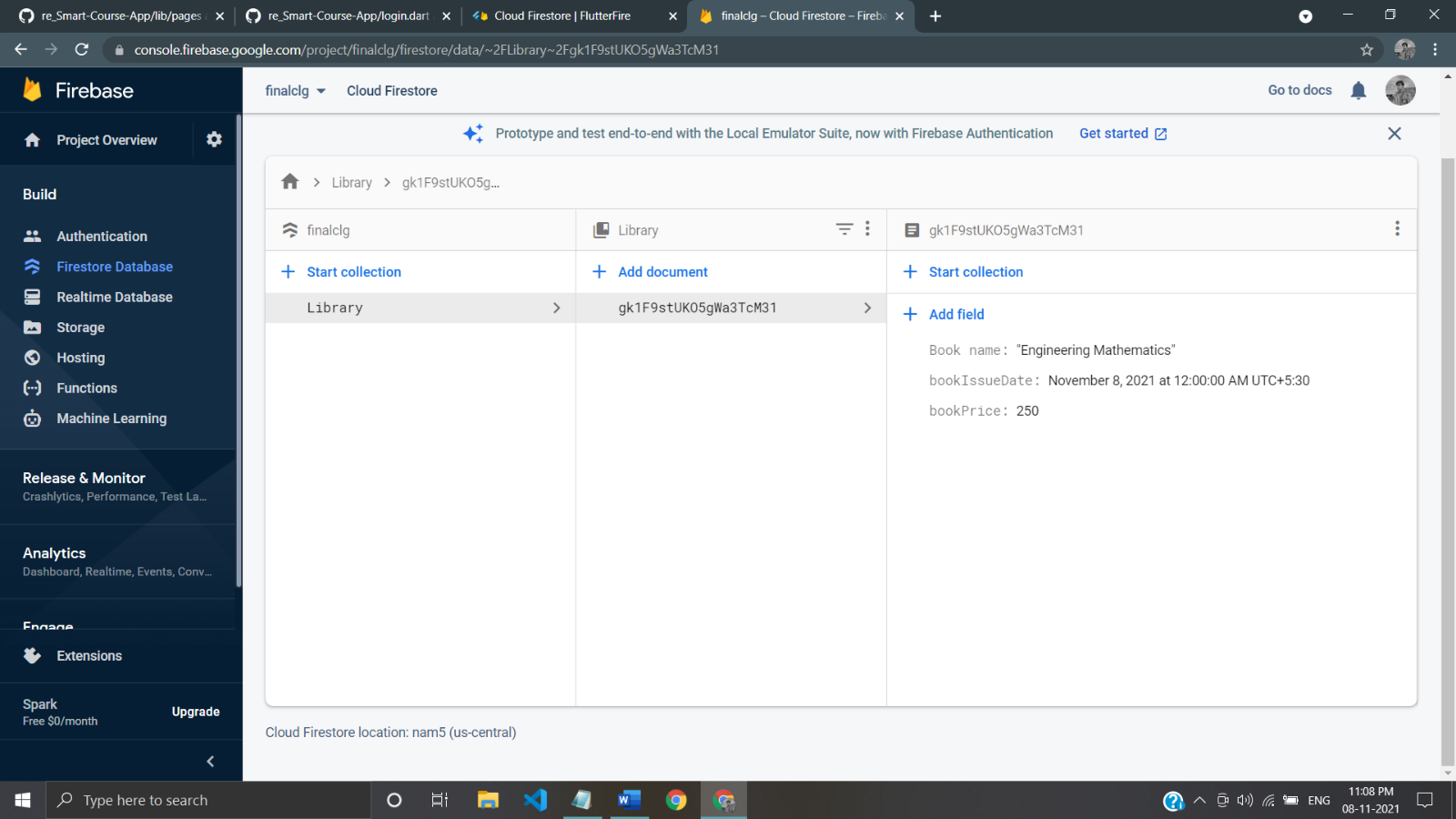
**(REALTIME DATABASE FOR ROLL NUMBER)**

**SHOT – 17**



**(REALTIME DATABASE FOR STUDENT NAME)**

**SHOT - 18**



**(FIREBASE DATABASE FOR LIBRARY)**

**SHOT -19**

**CHAPTER - 5**

**PROJECT CODING**

**5.1 User Login**

final \_formkey = GlobalKey<FormState>();

var email = "";

var password = "";

final emailController = TextEditingController();

final passwordController = TextEditingController();

final storage = new FlutterSecureStorage();

userLogin() async {

try {

UserCredential userCredential = await FirebaseAuth.instance

.signInWithEmailAndPassword(email: email, password: password);

await storage.write(key: 'uid', value: userCredential.user?.uid);

Navigator.pushReplacement(

context,

MaterialPageRoute(

builder: (context) => UserMain(),

),

);

} on FirebaseAuthException catch (e) {

if (e.code == 'user-not-found') {

print('No User Found for that Email');

ScaffoldMessenger.of(context).showSnackBar(

const SnackBar(

backgroundColor: Colors.redAccent,

content: Text(

"No User Found for that Email",

style: TextStyle(fontSize: 18.0, color: Colors.white),

),

),

);

} else if (e.code == 'wrong-password') {

print('Wrong Password Provided by the User');

ScaffoldMessenger.of(context).showSnackBar(

const SnackBar(

backgroundColor: Colors.redAccent,

content: Text(

"Wrong Password Provided by the User",

style: TextStyle(fontSize: 18.0, color: Colors.white),

),

),

);

}

}

}

**CHAPTER – 6**

**TESTING**

The more features your app has, the harder it is to test manually. Automated tests help ensure that your app performs correctly before you publish it, while retaining your feature and bug fix velocity.

Automated testing falls into a few categories:

1. A unit test tests a single function, method, or class.
2. A widget test (in other UI frameworks referred to as component test) tests a single widget.
3. An integration test tests a complete app or a large part of an app.

**Unit tests**

A unit test tests a single function, method, or class. The goal of a unit test is to verify the correctness of a unit of logic under a variety of conditions. External dependencies of the unit under test are generally mocked out. Unit tests generally don’t read from or write to disk, render to screen, or receive user actions from outside the process running the test.

**Widget tests**

A widget test (in other UI frameworks referred to as component test) tests a single widget. The goal of a widget test is to verify that the widget’s UI looks and interacts as expected. Testing a widget involves multiple classes and requires a test environment that provides the appropriate widget lifecycle context.

**Integration tests**

An integration test tests a complete app or a large part of an app. The goal of an integration test is to verify that all the widgets and services being tested work together as expected. Furthermore, you can use integration tests to verify your app’s performance.

**CONCLUSION**

This android based project will do the task through the application. It will decrease the paper work. The admin, faculty or the student will perform all the task very easily and more convenience way. The application offers reliability, security, time savings and easy control. . It can be used as a base for creating and enhancing applications for viewing results, tracking attendance for colleges. Students and their parents will also view results, attendance and curriculum details using this application, also students can view details, notificationsanywhere and anytime. The application will greatly simplify and speed up the result preparation and management process. The proposed system will decrease the work time of the admin as well as the faculty. This will bring more perfection to the work.

**LIMITATIONS**

The drawbacks in College application can be counted on fingers; with mostly only benefits, these systems have a few countable downsides. Often, applications face minor technical glitches and these systems are no exception but, ratification is immediate. Only, people who are accustomed to regular use of smartphones or computers can operate this software. Extensive modules and features make it difficult for a user to utilise the application. With huge flow in traffic the application is prone to performance issues. Few companies market their products at extravagant price, which are not affordable by growing organizations. Absence of proper internet-network makes it difficult for a user to access information, which is a significant disadvantage. The risk of data mishandling might be bothersome; but all these drawbacks can be evaded by choosing proper, cost-efficient and best software that best benefits an organization.

**REFERENCES**

1. Flutter Packages Website - <https://pub.dev/>
2. Flutter official website - <https://flutter.dev/>
3. Stackoverflow - <https://stackoverflow.com/questions/tagged/android>
4. Flutter official github page - <https://github.com/flutter/flutter>
5. Flutter Plugins website - <https://github.com/flutter/plugins>
6. Firebase installation website - <https://firebase.flutter.dev/docs/installation/android>
7. Flutter and Firebase integration setup - <https://github.com/FirebaseExtended/flutterfire>
8. Firebase documentation - <https://firebase.flutter.dev/docs/auth/overview>
9. Error handling in firebase auth - <https://firebase.flutter.dev/docs/auth/error-handling>
10. Firebase database - <https://firebase.flutter.dev/docs/database/overview>
11. Payment gateway - <https://razorpay.com/docs/payment-gateway/flutter-integration/standard/>
12. Firestore docs - <https://firebase.google.com/docs/firestore>
13. Firebase flutter website - <https://firebase.flutter.dev/docs/firestore/overview/>