Module	**Portfolio**	**Assessment Type**
Collaborative Development (5CS024)	1	Individual Report

EXPENSES MANAGEMENT SYSTEM DEVELOPER

Student Id	: 2227486
Student Name	: Nayan Raj Khanal
Section	: L5CG4
Group	: L5CG4 Group 5
Role	: Developer
Instructor	: Mr. Adarsha Khadka
Submitted on	: 1st April 2023
Word Count	: 2673

Acknowledgement

I am grateful of the chance to pursue higher education that Herald College Kathmandu and the University of Wolverhampton provided me with as well as their ongoing support during my academic career. My personal and professional development and objectives have both been greatly influenced by the education and experiences I have acquired from these institutions.

I also want to express my gratitude to my instructor, Mr. Adarsha Khadka, whose advice and mentoring were extremely helpful. His dedication to the classroom and enthusiasm for the material inspired me to pursue academic excellence.

I also like to thank the module team for their commitment to the course and for making the classroom a positive learning atmosphere. Their knowledge of the subject and their insights have improved my understanding of it.

I also want to thank the internet for giving me access to a variety of knowledge and material that I needed to finish this report.

Finally, I want to express my gratitude to my family and friends for their unwavering love, support, and inspiration throughout my academic career. They have been a constant source of inspiration for me, and I am appreciative of having them in my life.

I owe you all a debt of gratitude for helping me advance both academically and personally. I'm grateful.

Table of Contents

Table of Figure

1. Self-appraisal form

Student number	2227486	Name	Nayan Raj Khanal
Project	Expenses Management System	Date	1st April 2023
Role	Developer	Team	L5CG4 (Group 5)
Sprint (1 or 2)	1		

2. Personal objectives – performance measurement

Objectives	**Evidence provided**	**Evaluation** *Student / tutor*	
Choosing for the relevant technologies	The selection of appropriate technologies is critical for the success of any project. As we aimed for a user-friendly and readily accessible system, developing a mobile application appeared to be the optimal choice over web or desktop. After careful consideration, we concluded that Flutter, a mobile app development framework, is the ideal choice for our project. Furthermore, the combination of Golang and MongoDB for back-end development and database management, respectively, complemented each other well. A well-structured and concise report on the given objective has been provided. **Appendix A**	**9**	
Tutor feedback:			
Implementing functional requirement	I convinced my team to use Flutter, Golang, and MongoDB for our project, despite being an inexperienced developer. While it may have been safer to stick with familiar technologies, I took a risk and chose to learn something new. We just started our development process, so far it has been going well and I am determined to continue learning and complete the project with these new technologies. Screenshots as well as proper comments and explanation of the codes has been provided. **Appendix B**	**9**	
Tutor feedback:			
		/20	/20

3. Collaboration Document

3.1. Evidence of good communication and file sharing

Effective communication is essential for the success of any team-based project. To ensure smooth communication within my team, I frequently interacted and shared updates in the Campfire section, where I informed the team of the ongoing and completed tasks. I also utilized the message board to pin important files and links for easy accessibility. Furthermore, informal communication such as quick updates and instant messaging were carried out on Discord and Messenger.

Screenshots of conversations and effective use of Basecamp tools have been provided.

Appendix C

3.2. Continuing Personal Development (CPD)

Success requires constant self-improvement and learning. As a beginner developer, I started building an Expense Management System app with technologies that are completely new to me like Figma for UI designs, Flutter for front-end, Golang for back-end, and MongoDB for databases. I have completed the Sprint 1 assignment by taking online courses on Flutter, Golang, and MongoDB, and learning Figma for UI design. This experience has given me useful knowledge and skills to apply in future development projects.

A descriptive report and screenshots of ongoing courses have been provided.

Appendix D

3.3. Issue Tracking

Even while working correctly, challenges will inevitably arise in everyone's daily lives. No one is perfect. Issues are certain to occur while creating software. Client's satisfaction with their product is solely in the developers' hand hence it is essential as a developer I give my best so that the client gets what they paid for. Hence working together with the BA, we have tried and tested the code rigorously and completed it in the allocated time.

An informative report and evidences of GitHub issues collaboration between the BA and developer has been provided.

Appendix E

4. Appendix A

4.1. Technology Maze: A Justification Report

4.1.1. Introduction

The success of a mobile app development project relies heavily on selecting the appropriate technologies. This report will outline the technologies we have chosen to meet the client's requirements for mobile app development, along with the reasoning behind our decisions. We will also discuss any potential risks and limitations associated with the technologies we have selected.

4.1.2. Technologies For Application Development

Numerous technologies have evolved that are appropriate for mobile app development, such as programming languages, platforms, and database management systems (DBMS). Some popular and reliable technologies include Dart, Golang, Android Studio, Flutter, MongoDB, Firebase etc. (Technologies, 2023)

4.1.3. Justification Of The Technology:

After countless discussions and meticulous planning, we have narrowed down our technology choices to three.

For front-end development, we have chosen Flutter, an open-source framework developed by Google that uses Dart programming language. Flutter's unique feature of Hot Reload allows real-time viewing of code changes, making the development process quicker. The community provides extensive support and resources for novice developers. The reason for choosing Dart as the programming language is that Flutter is powered by Dart and also it is optimized for building mobile apps. (Thomas, 2019)

For back-end development, we have chosen Golang, an open-source, modern programming language developed by Google. Golang is easy to learn, strongly typed, and has a large community with several third-party tools and libraries that can simplify development. (Chris, 2021)

Finally, for DBMS, we have chosen MongoDB, a popular open-source nonrelational DBMS with extensive support in the developer community. After extensive research, we believe that Go and MongoDB are an excellent combination because of the lightweight, data-friendly syntax and compiled performance of Go. (MongoDB, 2023)

4.1.4. Risks/Limitations:

Although we are confident that the technologies, we have chosen are the most suitable for the project, there are potential risks and limitations to consider. For instance, Flutter may require a powerful machine to operate at peak performance, and its evolving library may not have all the features needed. Golang's type system, while beneficial in many cases, can be restrictive, and its garbage collector can cause performance problems when dealing with large datasets. MongoDB's dynamic schema can lead to data inconsistencies and errors if not managed properly, and its scalability and performance can be affected by the hardware and network infrastructure on which it is deployed. (Chris, 2021) (MongoDB, 2023) (Thomas, 2019)

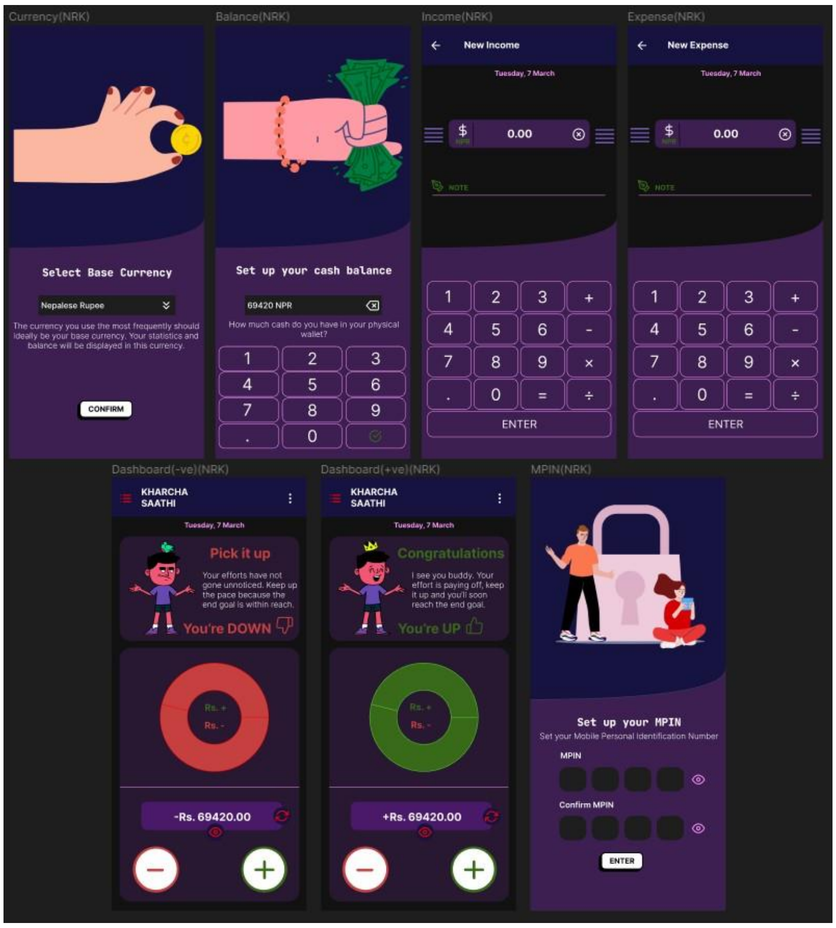
4.1.5. Conclusion:

To summarize, choosing the right technologies for mobile application development is vital for the project's success. We have conducted extensive research and evaluation and have decided to use Flutter for front-end development, Golang for back-end development, and MongoDB for database management. Although these technologies have their drawbacks, we believe they are the most appropriate for this project. We need to plan, monitor, and manage them carefully to ensure optimal performance and prevent any potential risks that may arise from using these technologies.

5. Appendix B

5.1. Implementing functional requirement

5.1.1. UI



The currency you use the most frequently should ideally be your base currency. Your statistics and balance will be displayed in this currency.

How much cash do you have in your physical wallet?

Your efforts have not gone unnoticed. Keep up the pace because the end goal is within reach.

I see you buddy. Your effort is paying off, keep it up and you'll soon reach the end goal.

Set your Mobile Personal Identification Number

Figure 1. 7 Keyframes (UI)

Using Figma, designed 7 key frames assigned to me. Moving from top to bottom, the **first frame** asks the user for their **default currency**, this currency will be used to manage the expenses of the user. The **second frame** asks users to set their **initial balance**, all their management is carried out from this starting point. The **third and fourth frame** allows users to carry out their **calculation** alongside with the ability to add **notes** for the specific transaction. The **fifth and sixth frame** are designed with the concept that, if the balance of the user drops below their initial balance, a **customized dashboard** urging the user to fix their habits is displayed and if the balance is steady/up then another dashboard is displayed acknowledging the users' efforts. Finally, the **seventh frame** asks user to set their **Mobile Personal Identification Number (MPIN)**, this 4-digit code will be used for almost everything in the app.

5.1.2. Front-end Development

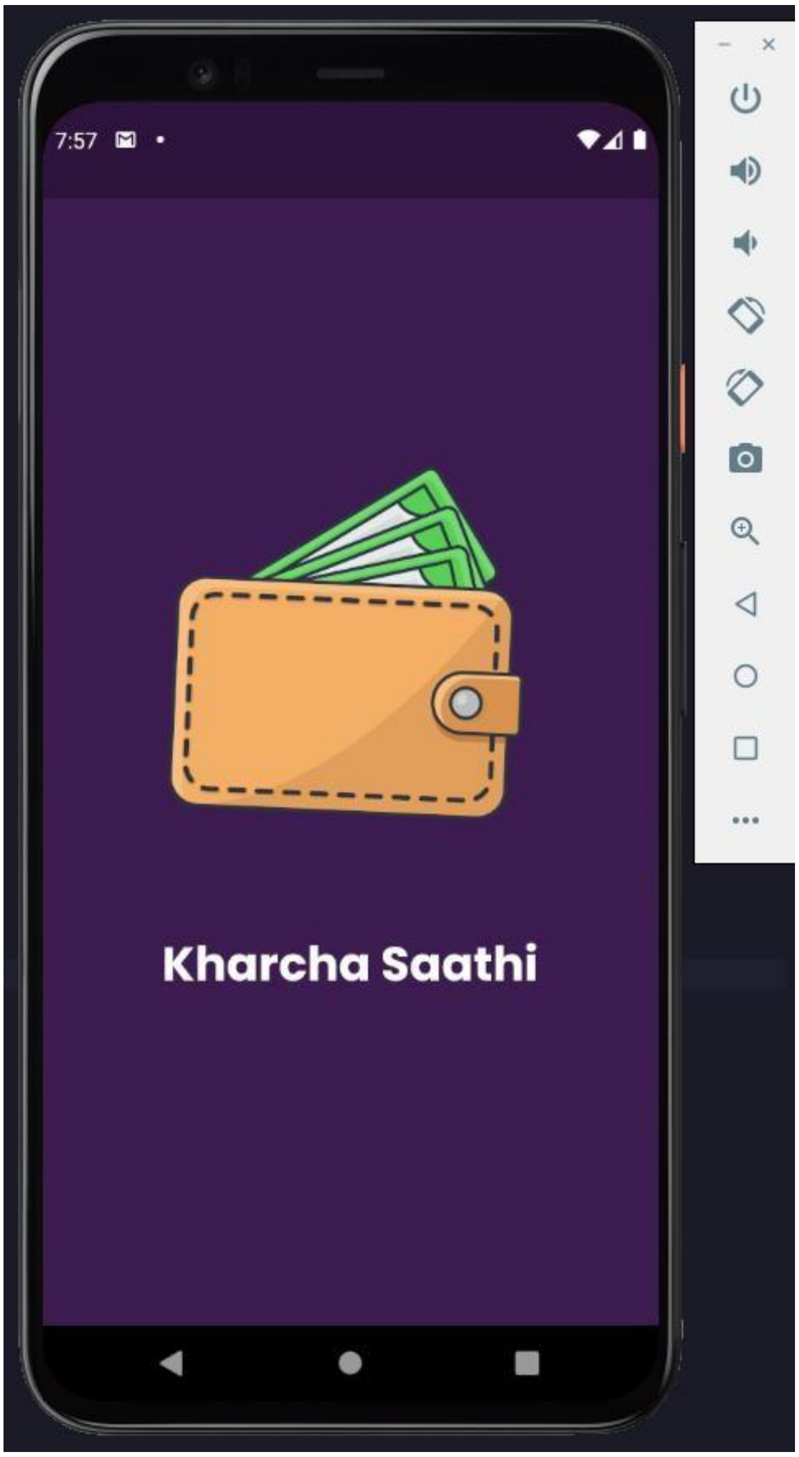


Figure 2. Splash screen Page

Designed and developed an animated splash screen for the app using flutter. It runs for 4 seconds and then the log in page is displayed.

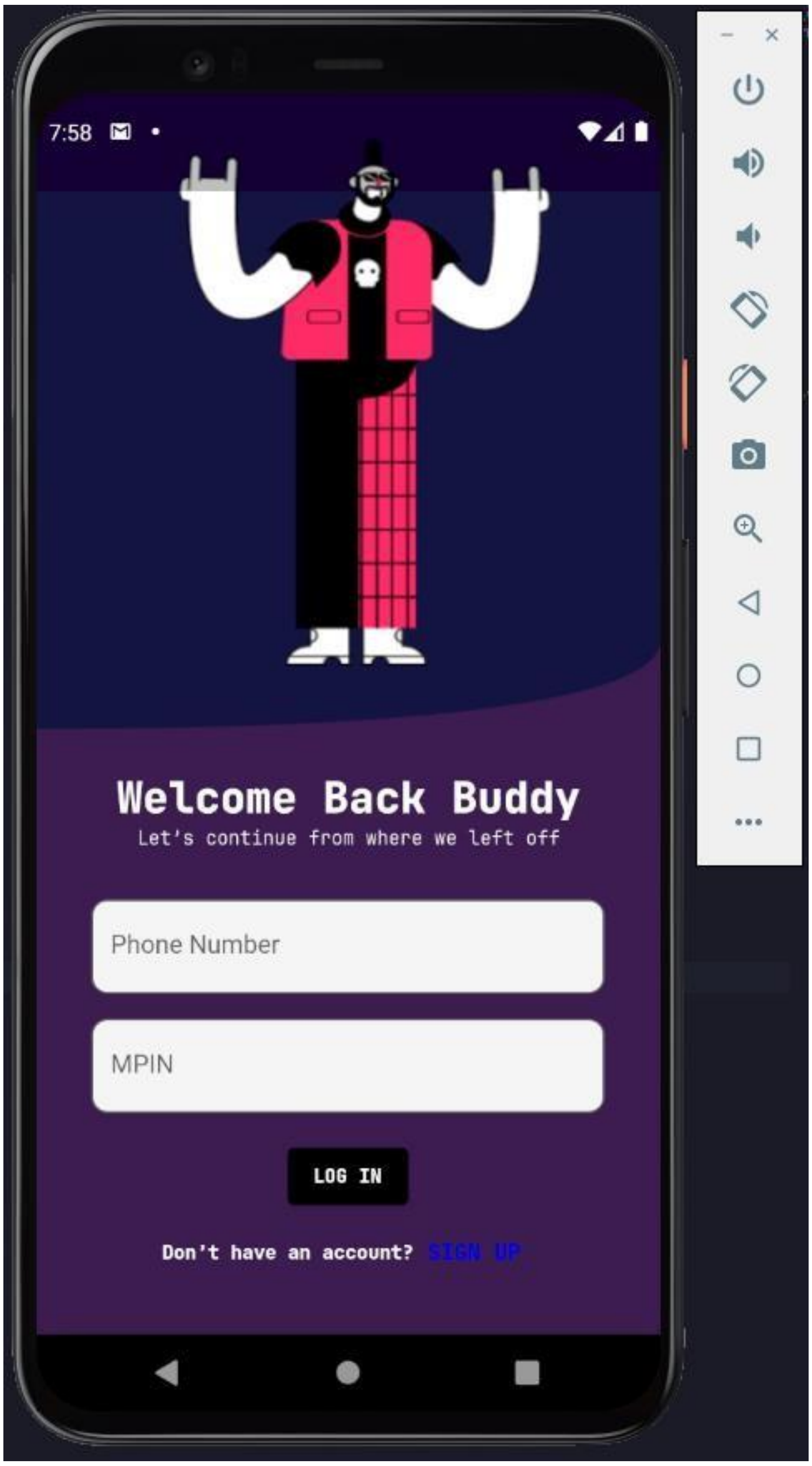
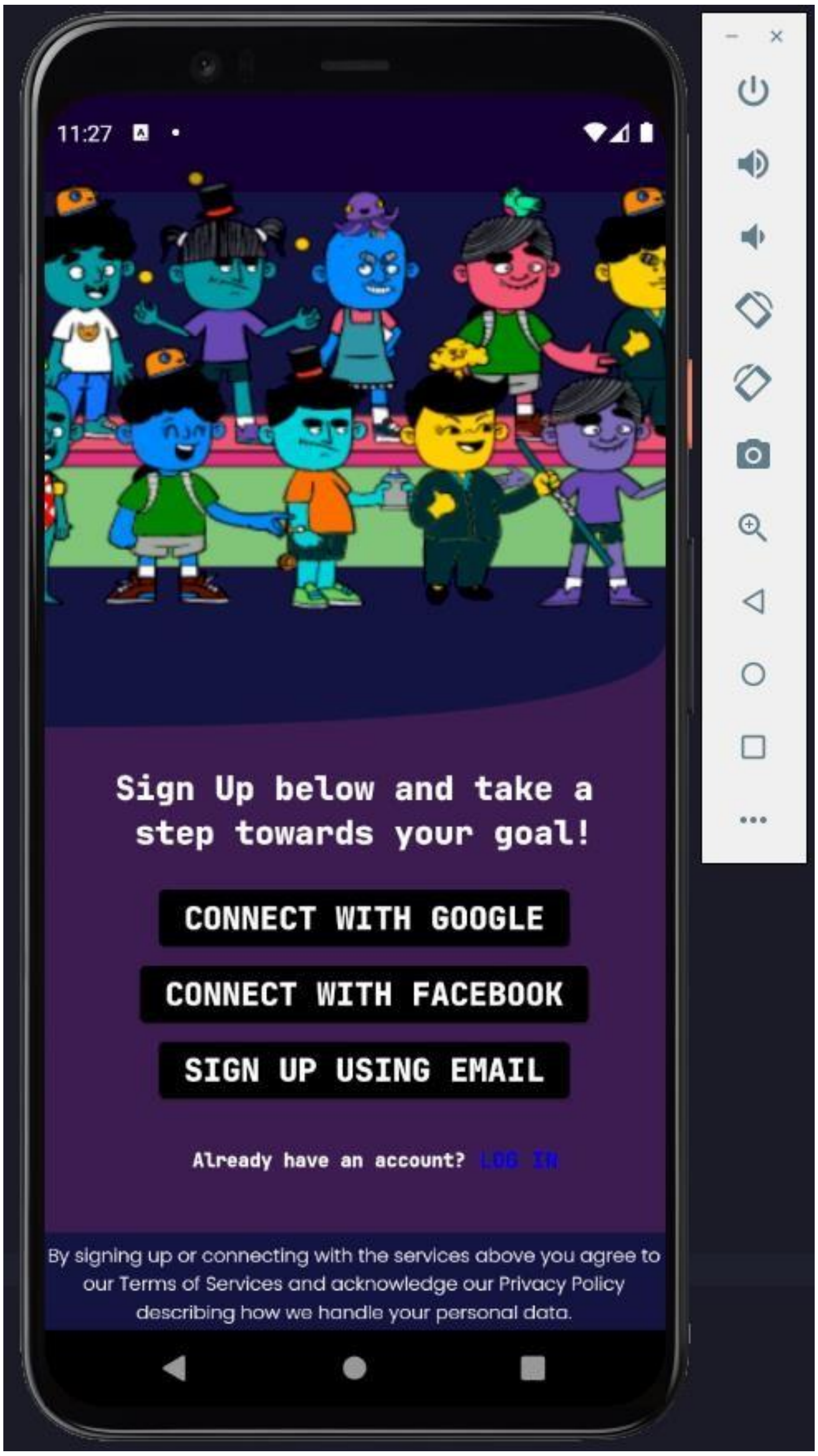

7:58
Welcome Back Buddy
Let's continue from where we left off
Phone Number
MPIN
LOG IN
Don't have an account? SIGN UP

Figure 3. Login Page

Designed and developed a log in screen for the app using flutter. The components are responsive i.e., they're scaled automatically for any device. The password field is made to hide the characters and the button and link are functional.


11:27

Sign Up below and take a step towards your goal!

By signing up or connecting with the services above you agree to our Terms of Services and acknowledge our Privacy Policy describing how we handle your personal data.

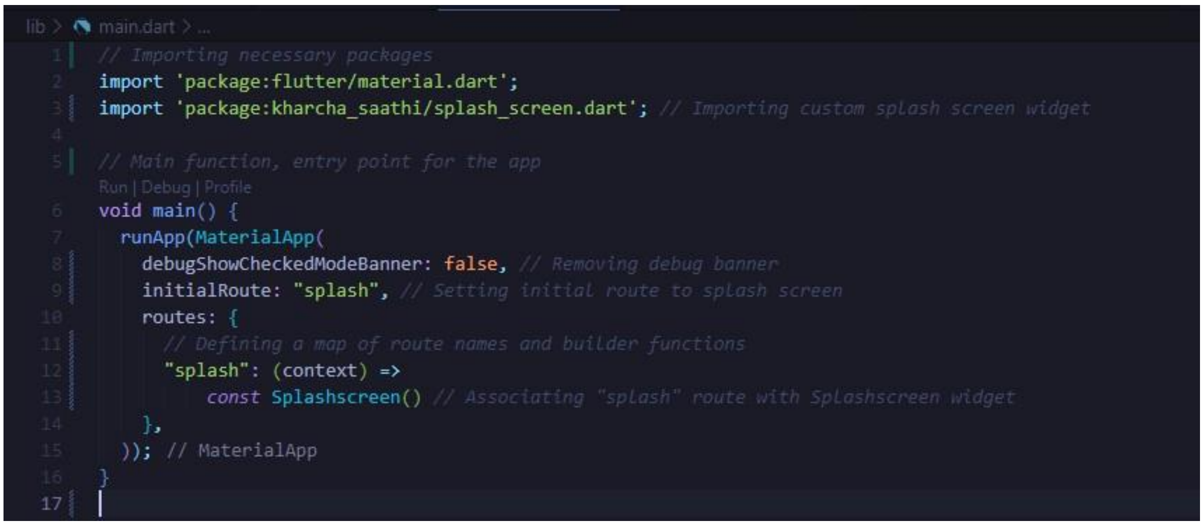
Figure 4. Signup Page

Designed and developed a signup screen for the app using flutter. Three elevated buttons and one text button is used.

5.2. Codes and its explanation

Sticking to the schedule, I started the development of front-end assigned to me. The code was written in VScode code editor with Dart as the programming language and using Flutter framework for running the application.

5.2.1. main.dart:

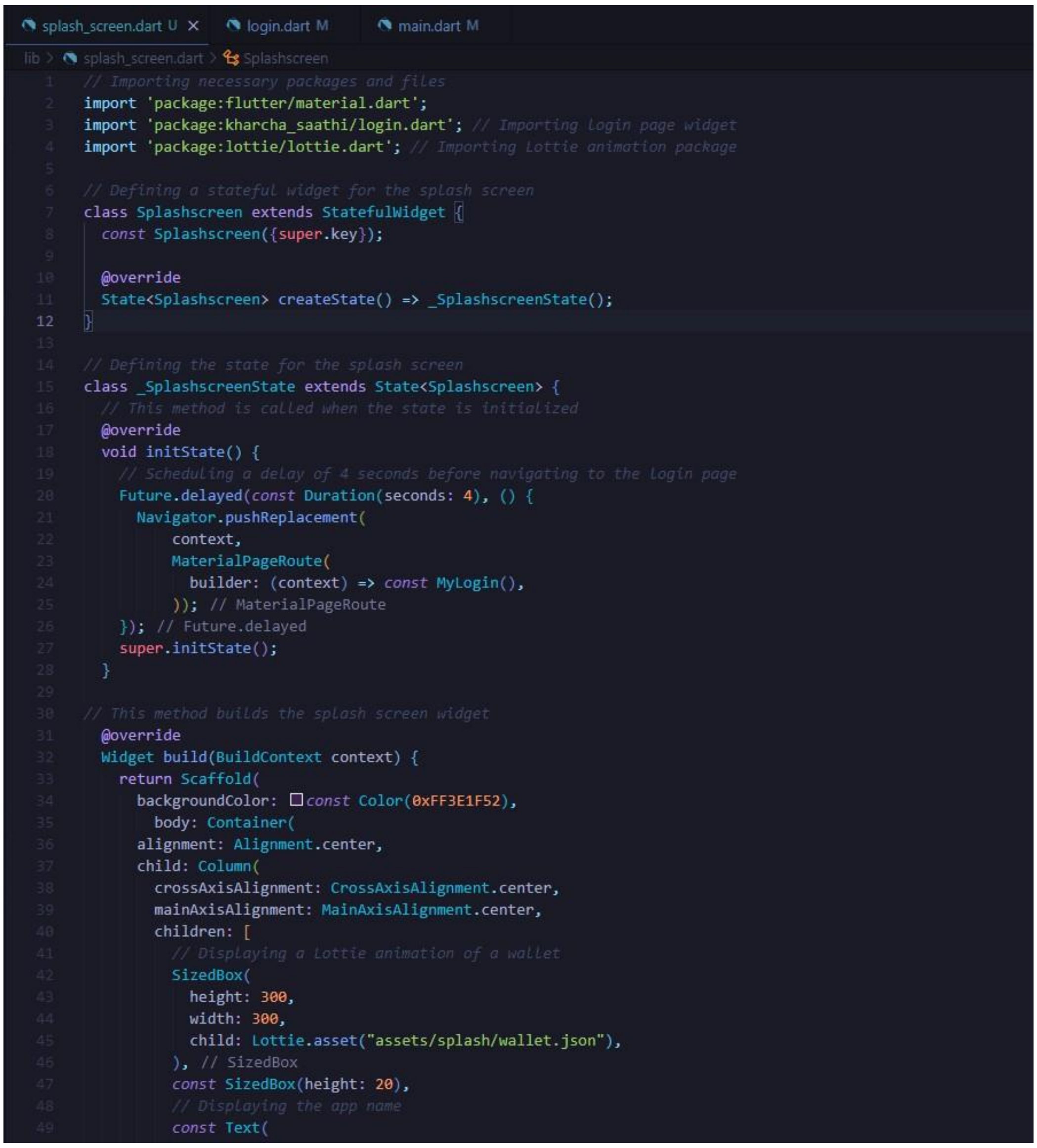


```
lib > main.dart > ...
// Importing necessary packages
import 'package:flutter/material.dart';
import 'package:kharcha_saathi/splash_screen.dart'; // Importing custom splash screen widget

// Main function, entry point for the app
Run | Debug | Profile
void main() {
  runApp(MaterialApp(
    debugShowCheckedModeBanner: false, // Removing debug banner
    initialRoute: "splash", // Setting initial route to splash screen
    routes: {
      // Defining a map of route names and builder functions
      "splash": (context) =>
          const Splashscreen() // Associating "splash" route with Splashscreen widget
    },
  )); // MaterialApp
}

```

Figure 5. main.dart

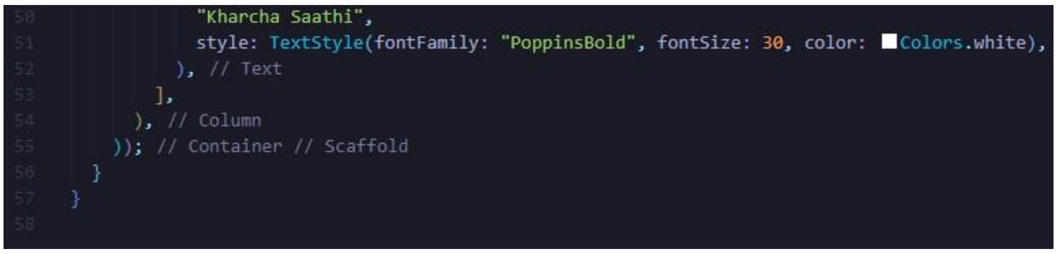
This code sets up the entry point for the Flutter app, defining a map of route names and builder functions to associate each route with a widget. It imports necessary packages and a custom splash screen widget. The **runApp()** function creates a new **MaterialApp** with a **debugShowCheckedModeBanner** property to remove the debug banner, an **initialRoute** property to set the starting route to the splash screen, and a **routes** property to define the **Splashscreen** widget as the target of the "splash" route.

5.2.2. splash_screen.dart



```dart
// Importing necessary packages and files
import 'package:flutter/material.dart';
import 'package:kharcha_saathi/login.dart'; // Importing login page widget
import 'package:lottie/lottie.dart'; // Importing Lottie animation package

// Defining a stateful widget for the splash screen
class Splashscreen extends StatefulWidget {
  const Splashscreen({super.key});

  @override
  State<Splashscreen> createState() => _SplashscreenState();
}

// Defining the state for the splash screen
class _SplashscreenState extends State<Splashscreen> {
  // This method is called when the state is initialized
  @override
  void initState() {
    // Scheduling a delay of 4 seconds before navigating to the login page
    Future.delayed(const Duration(seconds: 4), () {
      Navigator.pushReplacement(
          context,
          MaterialPageRoute(
            builder: (context) => const MyLogin(),
          )); // MaterialPageRoute
    }); // Future.delayed
    super.initState();
  }

// This method builds the splash screen widget
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: const Color(0xFF3E1F52),
        body: Container(
      alignment: Alignment.center,
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.center,
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          // Displaying a Lottie animation of a wallet
          SizedBox(
            height: 300,
            width: 300,
            child: Lottie.asset("assets/splash/wallet.json"),
          ), // SizedBox
          const SizedBox(height: 20),
          // Displaying the app name
          const Text(
```

Figure 6. splash_screen.dart (i)

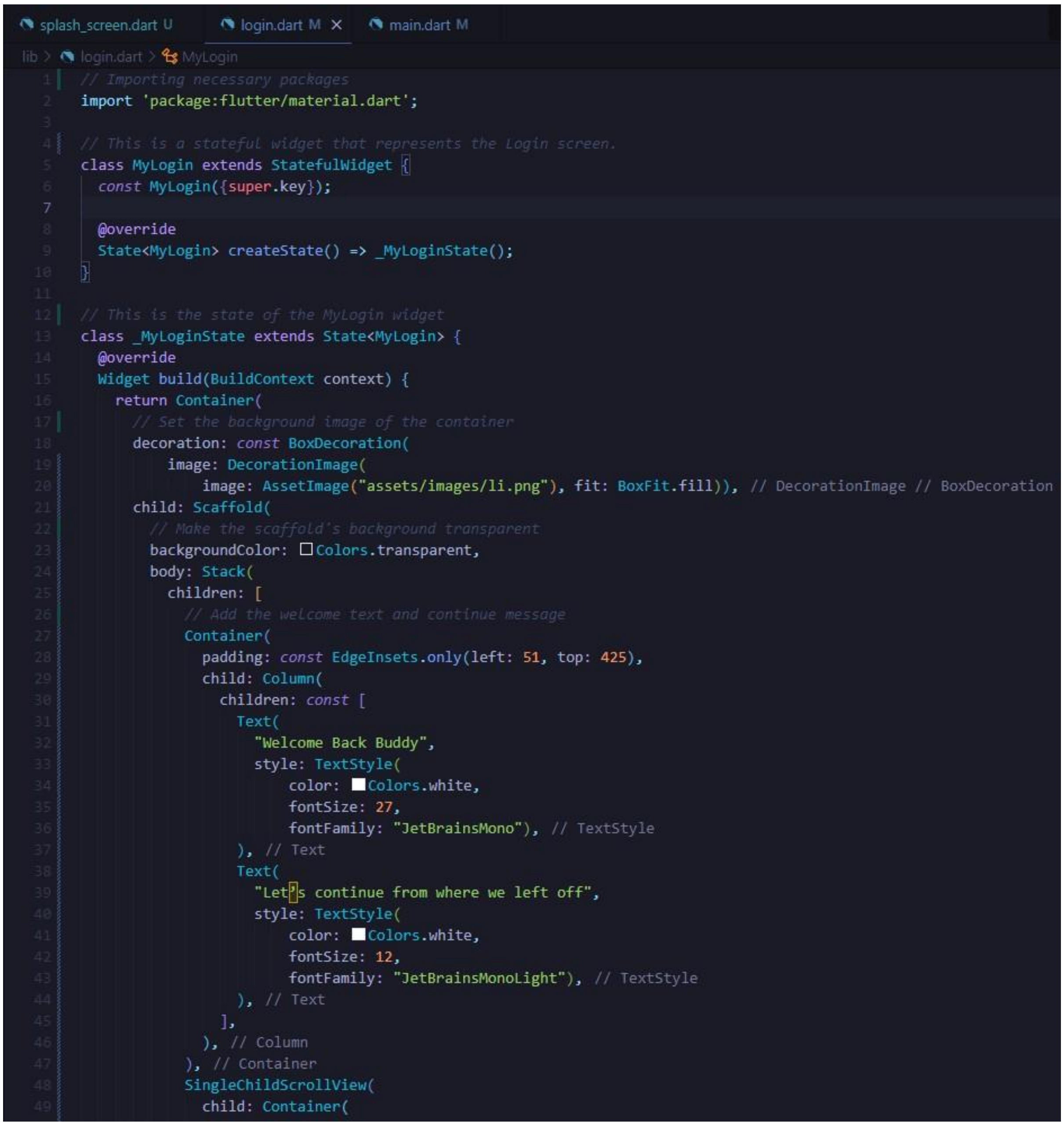


```dart
            "Kharcha Saathi",
            style: TextStyle(fontFamily: "PoppinsBold", fontSize: 30, color: Colors.white),
          ), // Text
        ],
      ), // Column
    )); // Container // Scaffold
  }
}
```

Figure 7. splash_screen.dart (ii)

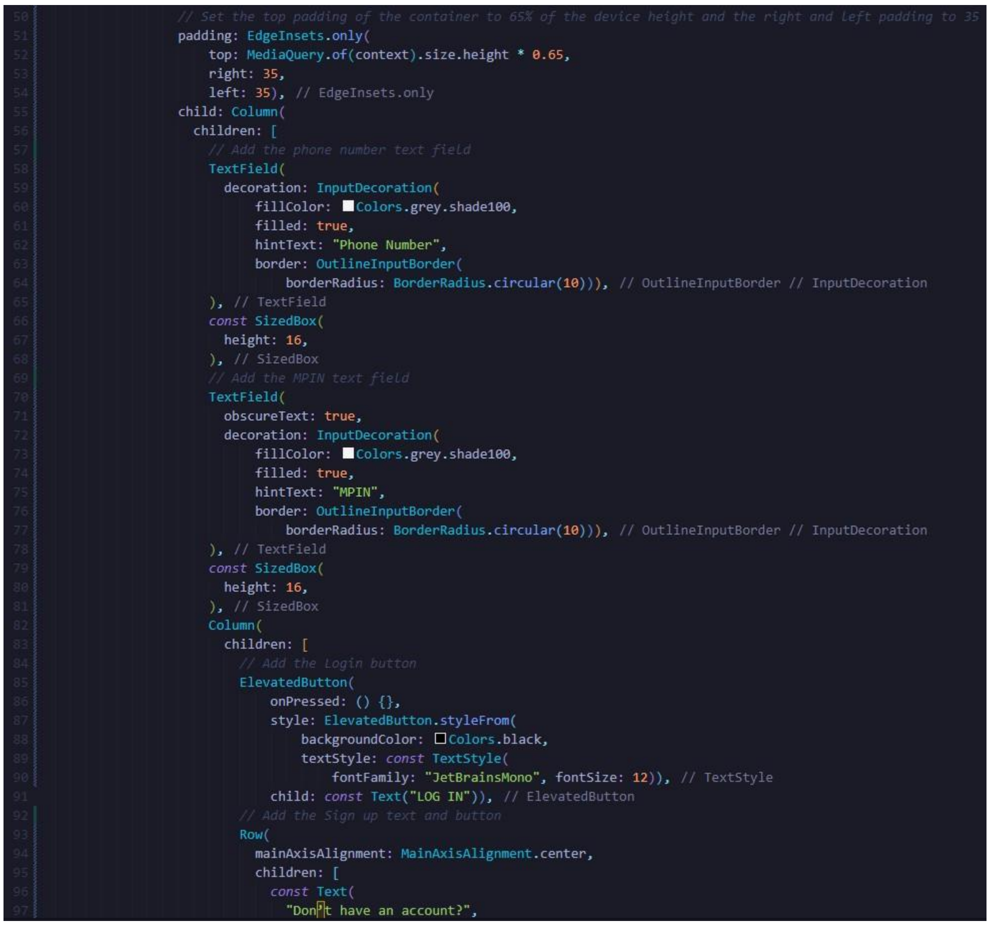
This Flutter code imports necessary packages and files, including a login page widget and a Lottie animation package. It defines a stateful widget for the splash screen, which displays an animated wallet and the app name "Kharcha Saathi" on a colored background. The **_SplashscreenState** class sets a 4-second delay using the **Future.delayed** method, then navigates to the login page using the **Navigator.pushReplacement** method. This widget is called by the main function and is associated with the "splash" route.

5.2.3. login.dart:



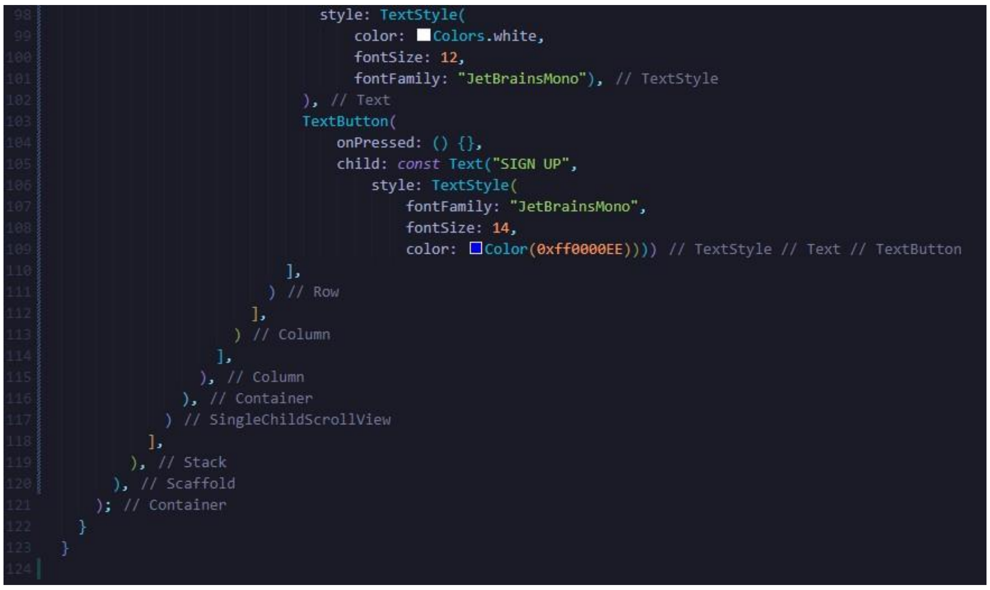
```dart
// Importing necessary packages
import 'package:flutter/material.dart';

// This is a stateful widget that represents the Login screen.
class MyLogin extends StatefulWidget {
  const MyLogin({super.key});

  @override
  State<MyLogin> createState() => _MyLoginState();
}

// This is the state of the MyLogin widget
class _MyLoginState extends State<MyLogin> {
  @override
  Widget build(BuildContext context) {
    return Container(
      // Set the background image of the container
      decoration: const BoxDecoration(
          image: DecorationImage(
              image: AssetImage("assets/images/li.png"), fit: BoxFit.fill)), // DecorationImage // BoxDecoration
      child: Scaffold(
        // Make the scaffold's background transparent
        backgroundColor: Colors.transparent,
        body: Stack(
          children: [
            // Add the welcome text and continue message
            Container(
              padding: const EdgeInsets.only(left: 51, top: 425),
              child: Column(
                children: const [
                  Text(
                    "Welcome Back Buddy",
                    style: TextStyle(
                        color: Colors.white,
                        fontSize: 27,
                        fontFamily: "JetBrainsMono"), // TextStyle
                  ), // Text
                  Text(
                    "Let's continue from where we left off",
                    style: TextStyle(
                        color: Colors.white,
                        fontSize: 12,
                        fontFamily: "JetBrainsMonoLight"), // TextStyle
                  ), // Text
                ],
              ), // Column
            ), // Container
            SingleChildScrollView(
              child: Container(
```

Figure 8. login.dart (i)



```
                  // Set the top padding of the container to 65% of the device height and the right and left padding to 35
                  padding: EdgeInsets.only(
                      top: MediaQuery.of(context).size.height * 0.65,
                      right: 35,
                      left: 35), // EdgeInsets.only
                  child: Column(
                    children: [
                      // Add the phone number text field
                      TextField(
                        decoration: InputDecoration(
                            fillColor: Colors.grey.shade100,
                            filled: true,
                            hintText: "Phone Number",
                            border: OutlineInputBorder(
                                borderRadius: BorderRadius.circular(10))), // OutlineInputBorder // InputDecoration
                      ), // TextField
                      const SizedBox(
                        height: 16,
                      ), // SizedBox
                      // Add the MPIN text field
                      TextField(
                        obscureText: true,
                        decoration: InputDecoration(
                            fillColor: Colors.grey.shade100,
                            filled: true,
                            hintText: "MPIN",
                            border: OutlineInputBorder(
                                borderRadius: BorderRadius.circular(10))), // OutlineInputBorder // InputDecoration
                      ), // TextField
                      const SizedBox(
                        height: 16,
                      ), // SizedBox
                      Column(
                        children: [
                          // Add the Login button
                          ElevatedButton(
                              onPressed: () {},
                              style: ElevatedButton.styleFrom(
                                  backgroundColor: Colors.black,
                                  textStyle: const TextStyle(
                                      fontFamily: "JetBrainsMono", fontSize: 12)), // TextStyle
                              child: const Text("LOG IN")), // ElevatedButton
                          // Add the Sign up text and button
                          Row(
                            mainAxisAlignment: MainAxisAlignment.center,
                            children: [
                              const Text(
                                "Don't have an account?",
```

Figure 9. login.dart (ii)

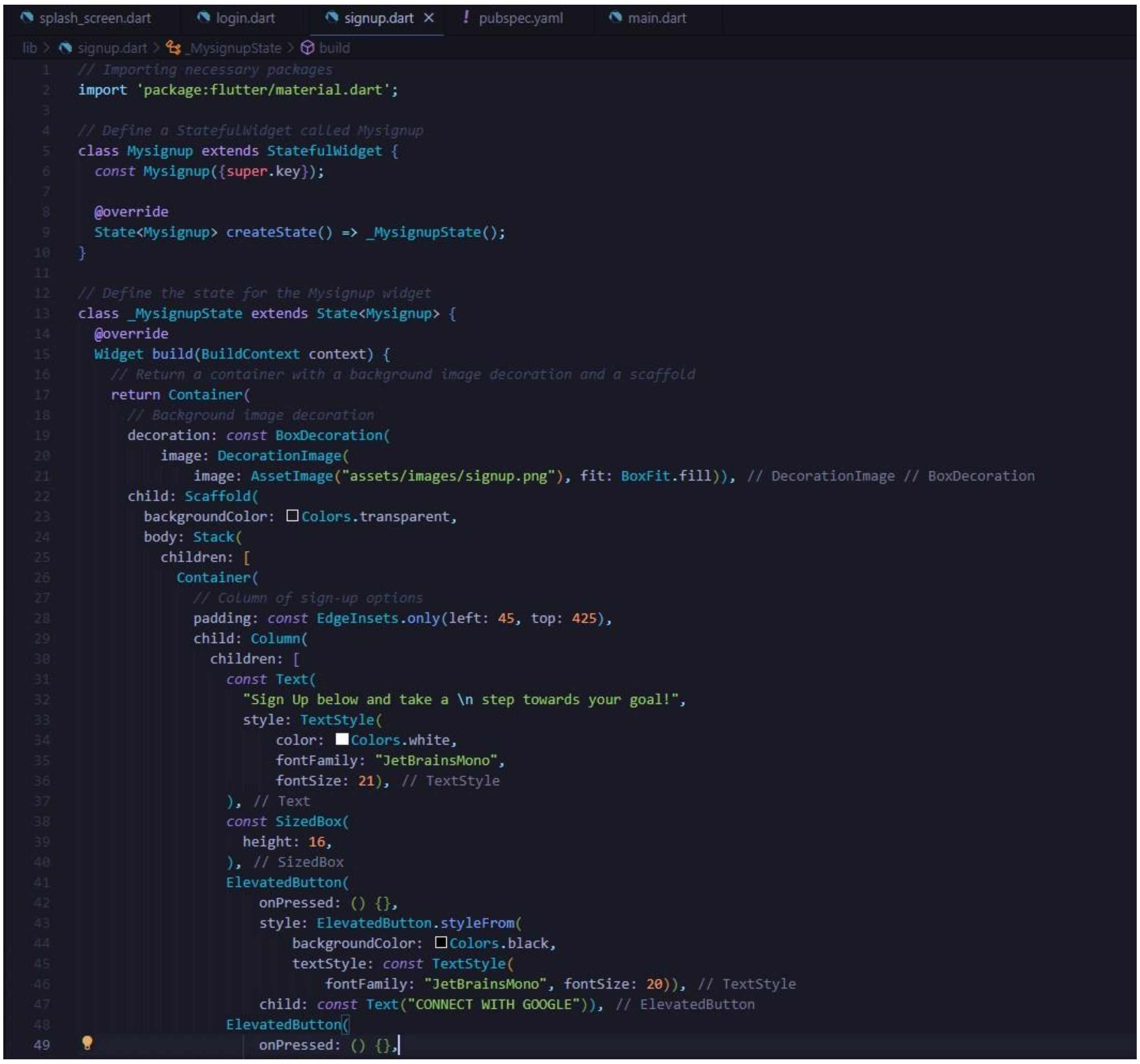


```
                                style: TextStyle(
                                    color: Colors.white,
                                    fontSize: 12,
                                    fontFamily: "JetBrainsMono"), // TextStyle
                              ), // Text
                              TextButton(
                                  onPressed: () {},
                                  child: const Text("SIGN UP",
                                      style: TextStyle(
                                          fontFamily: "JetBrainsMono",
                                          fontSize: 14,
                                          color: Color(0xff0000EE)))) // TextStyle // Text // TextButton
                            ],
                          ) // Row
                        ],
                      ) // Column
                    ],
                  ), // Column
                ), // Container
              ) // SingleChildScrollView
            ],
          ), // Stack
        ), // Scaffold
      ); // Container
    }
  }
```

Figure 10. login.dart (iii)

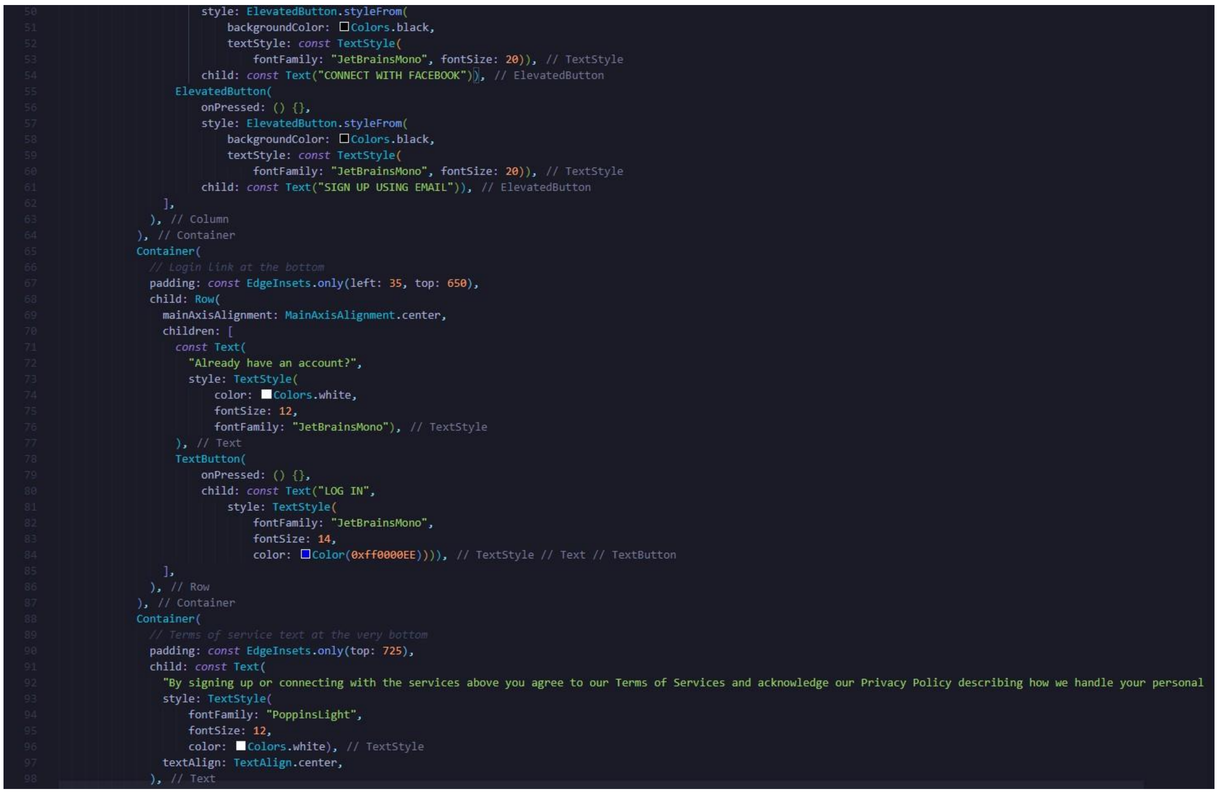
This code is a Flutter widget that represents a login screen. It contains two text fields for phone number and MPIN, a login button, and a sign-up button. It also has a background image and a welcome text with a continue message. The widget is implemented as a stateful widget, and its state has a build method that returns a Scaffold with a Stack containing the login interface elements. The text fields, buttons, and text are styled using various properties such as font size, font family, and color.

5.2.4. signup.dart



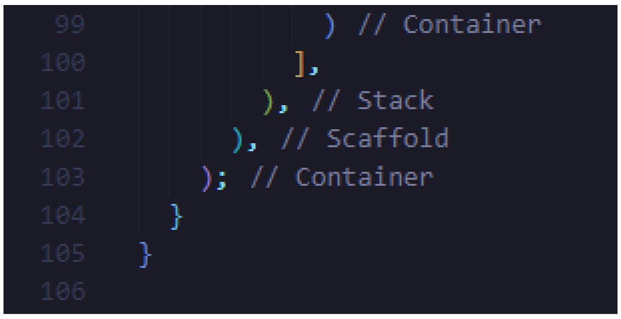
```dart
// Importing necessary packages
import 'package:flutter/material.dart';

// Define a StatefulWidget called Mysignup
class Mysignup extends StatefulWidget {
  const Mysignup({super.key});

  @override
  State<Mysignup> createState() => _MysignupState();
}

// Define the state for the Mysignup widget
class _MysignupState extends State<Mysignup> {
  @override
  Widget build(BuildContext context) {
    // Return a container with a background image decoration and a scaffold
    return Container(
      // Background image decoration
      decoration: const BoxDecoration(
          image: DecorationImage(
              image: AssetImage("assets/images/signup.png"), fit: BoxFit.fill)), // DecorationImage // BoxDecoration
      child: Scaffold(
        backgroundColor: Colors.transparent,
        body: Stack(
          children: [
            Container(
              // Column of sign-up options
              padding: const EdgeInsets.only(left: 45, top: 425),
              child: Column(
                children: [
                  const Text(
                    "Sign Up below and take a \n step towards your goal!",
                    style: TextStyle(
                        color: Colors.white,
                        fontFamily: "JetBrainsMono",
                        fontSize: 21), // TextStyle
                  ), // Text
                  const SizedBox(
                    height: 16,
                  ), // SizedBox
                  ElevatedButton(
                      onPressed: () {},
                      style: ElevatedButton.styleFrom(
                          backgroundColor: Colors.black,
                          textStyle: const TextStyle(
                              fontFamily: "JetBrainsMono", fontSize: 20)), // TextStyle
                      child: const Text("CONNECT WITH GOOGLE")), // ElevatedButton
                  ElevatedButton(
                      onPressed: () {},
```

Figure 11. signup.dart (i)



```
                  style: ElevatedButton.styleFrom(
                      backgroundColor: Colors.black,
                      textStyle: const TextStyle(
                          fontFamily: "JetBrainsMono", fontSize: 20)), // TextStyle
                  child: const Text("CONNECT WITH FACEBOOK")), // ElevatedButton
              ElevatedButton(
                  onPressed: () {},
                  style: ElevatedButton.styleFrom(
                      backgroundColor: Colors.black,
                      textStyle: const TextStyle(
                          fontFamily: "JetBrainsMono", fontSize: 20)), // TextStyle
                  child: const Text("SIGN UP USING EMAIL")), // ElevatedButton
            ],
          ), // Column
        ), // Container
        Container(
          // Login link at the bottom
          padding: const EdgeInsets.only(left: 35, top: 650),
          child: Row(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              const Text(
                "Already have an account?",
                style: TextStyle(
                    color: Colors.white,
                    fontSize: 12,
                    fontFamily: "JetBrainsMono"), // TextStyle
              ), // Text
              TextButton(
                  onPressed: () {},
                  child: const Text("LOG IN",
                      style: TextStyle(
                          fontFamily: "JetBrainsMono",
                          fontSize: 14,
                          color: Color(0xff0000EE)))), // TextStyle // Text // TextButton
            ],
          ), // Row
        ), // Container
        Container(
          // Terms of service text at the very bottom
          padding: const EdgeInsets.only(top: 725),
          child: const Text(
            "By signing up or connecting with the services above you agree to our Terms of Services and acknowledge our Privacy Policy describing how we handle your personal
            style: TextStyle(
                fontFamily: "PoppinsLight",
                fontSize: 12,
                color: Colors.white), // TextStyle
            textAlign: TextAlign.center,
          ), // Text
```

Figure 12. signup.dart (ii)

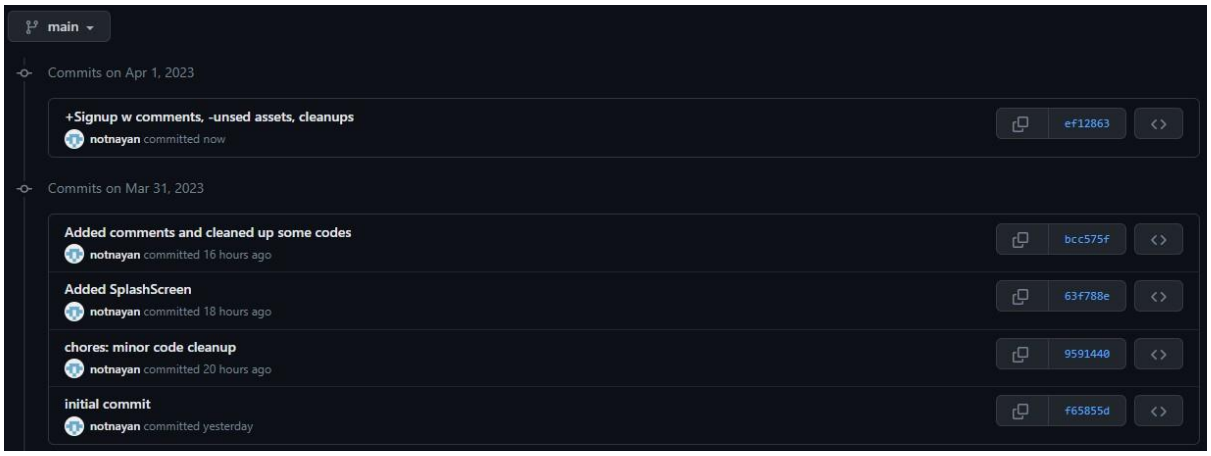


```
          ) // Container
        ],
      ), // Stack
    ), // Scaffold
  ); // Container
}
}
```

Figure 13. signup.dart (iii)

This Flutter code defines a Mysignup widget that displays a sign-up page with options to connect using Google, Facebook, or email. The page also includes a login link at the bottom and a terms of service statement at the very bottom. The design utilizes a background image imported from Figma.

5.3. Use of Version control (Screenshots of logs)



Commits on Apr 1, 2023

+Signup w comments, -unsed assets, cleanups
notnayan committed now — ef12863

Commits on Mar 31, 2023

Added comments and cleaned up some codes
notnayan committed 16 hours ago — bcc575f

Added SplashScreen
notnayan committed 18 hours ago — 63f788e

chores: minor code cleanup
notnayan committed 20 hours ago — 9591440

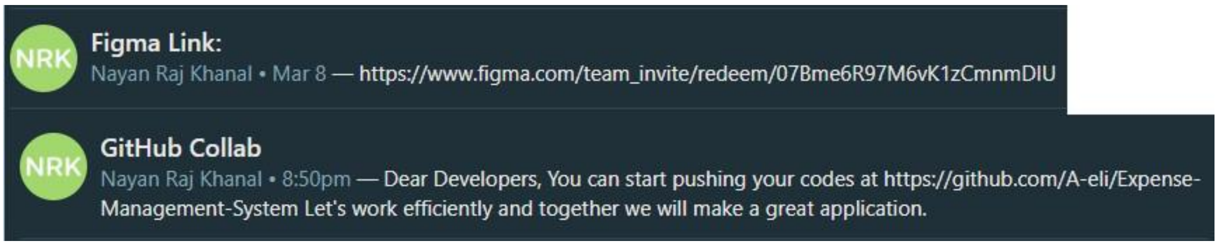
initial commit
notnayan committed yesterday — f65855d

Figure 14. Commit Logs

Daily commits of the codes in GitHub from the start of development phase i.e., March 31st 2023.

6. Appendix C

6.1. Evidences of Good communication and file sharing

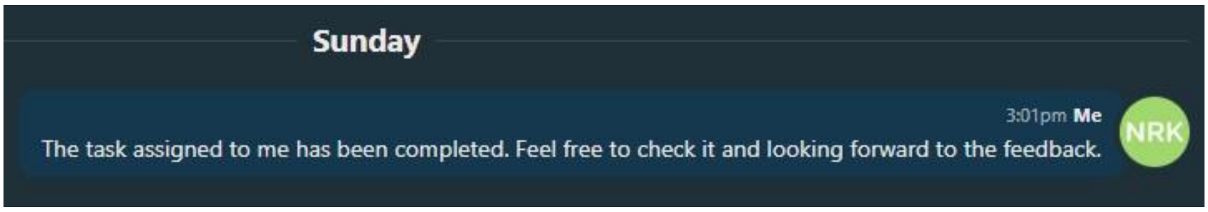


NRK **Figma Link:**
Nayan Raj Khanal • Mar 8 — https://www.figma.com/team_invite/redeem/07Bme6R97M6vK1zCmnmDIU

NRK **GitHub Collab**
Nayan Raj Khanal • 8:50pm — Dear Developers, You can start pushing your codes at https://github.com/A-eli/Expense-Management-System Let's work efficiently and together we will make a great application.

Figure 15. Message Board (Evidence)

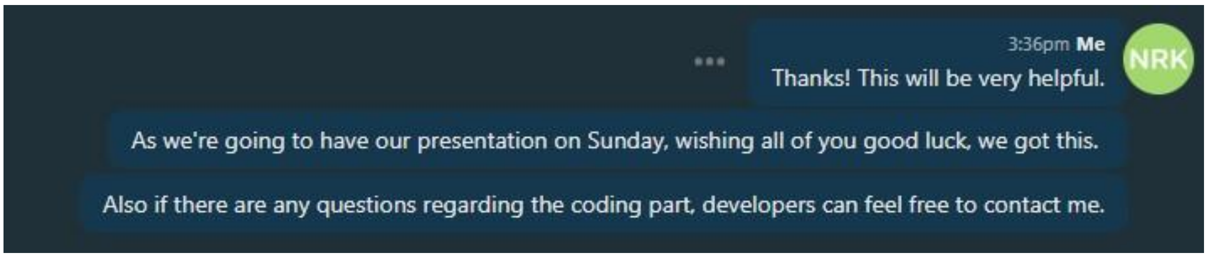
I used the message board to post links and inform fellow team members about Figma and GitHub.



Sunday

3:01pm **Me** NRK
The task assigned to me has been completed. Feel free to check it and looking forward to the feedback.

Figure 16. Chat (i) (Evidence)



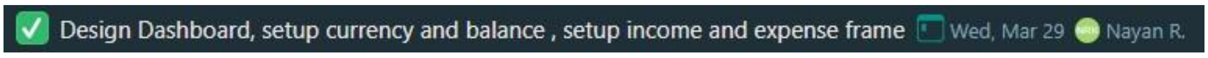
3:36pm **Me** NRK
Thanks! This will be very helpful.

As we're going to have our presentation on Sunday, wishing all of you good luck, we got this.

Also if there are any questions regarding the coding part, developers can feel free to contact me.

Figure 17. Chat (ii) (Evidence)

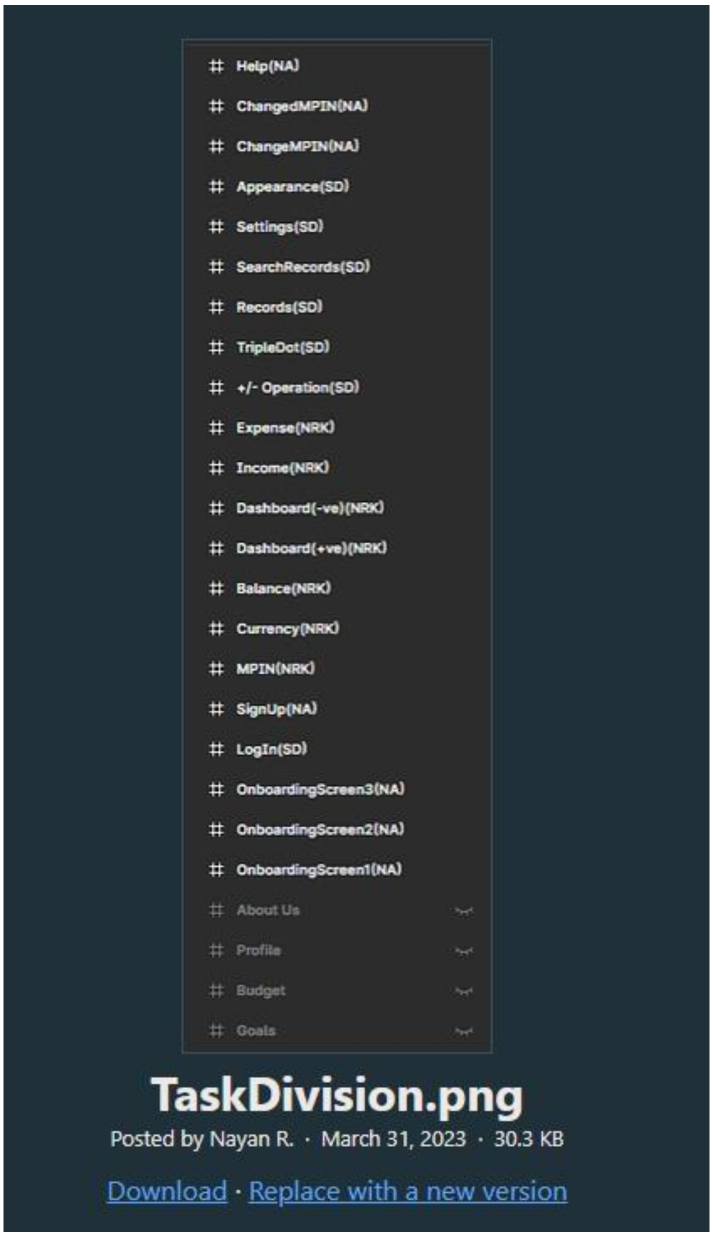
The above two pictures show me actively communicating with my team members. Responding to the tasks given to me and providing morale support.



✓ Design Dashboard, setup currency and balance , setup income and expense frame Wed, Mar 29 Nayan R.

Figure 18. TO DO List (Evidence)

I actively check the TO DO list and punctually complete the assigned tasks.



Help(NA)
ChangedMPIN(NA)
ChangeMPIN(NA)
Appearance(SD)
Settings(SD)
SearchRecords(SD)
Records(SD)
TripleDot(SD)
+/- Operation(SD)
Expense(NRK)
Income(NRK)
Dashboard(-ve)(NRK)
Dashboard(+ve)(NRK)
Balance(NRK)
Currency(NRK)
MPIN(NRK)
SignUp(NA)
LogIn(SD)
OnboardingScreen3(NA)
OnboardingScreen2(NA)
OnboardingScreen1(NA)
About Us
Profile
Budget
Goals

TaskDivision.png

Posted by Nayan R. · March 31, 2023 · 30.3 KB

Download · Replace with a new version

Figure 19. File Sharing (Evidence)

I share relevant files in the Doc & Files sections. Here, I have posted task division for fellow developers so that they can easily find the frames they need to develop.

7. Appendix D

7.1. Evidences of Continuing Personal Development (CPD)

7.1.1. INTRODUCTION

To be successful in anything one must work on oneself, strive to learn new things daily and be consistent. But now a days you'll find anything you want in the internet if you know where to look. It is a boon for every individual and I was no different.

Being a novice developer who has previously only studied Java, HTML, CSS, Javascript, and Python, but this project uses completely new technologies i.e., Figma for UI designs, Flutter for front-end development, Golang for Back-end development and finally MongoDB for databases. It is a steep mountain to overcome but I diligently dedicated my time to continue my personal development in order to effectively complete this project.

7.1.2. CPD

To begin, I have attended and I am still attending various online courses to gain a better understanding of Flutter, Dart, Golang, and MongoDB. Recently, I had attended a Flutter Development course on YouTube that provided me with an overview of the framework and its features. This course also taught me how to build splash screen and responsive log in page using Flutter, which has been incredibly useful in my current project. I am also currently undergoing a 37-hour long course from freeCodeCamp on Flutter. Additionally, I have attended a Golang course that provided me with a basic understanding of the language and its uses. I was able to test and write codes upon the end of the video. Furthermore, I have attended MongoDB courses that have given me a better understanding of database management systems and how to implement them within an app. Overall, these courses have helped me to better understand the technologies required for my project and have enabled me to apply my knowledge effectively. I have also enrolled for courses at freeCodeCamp and I am halfway through getting the certificates for it.

One of the primary requirements of my project was to design a user interface (UI) for the app. To achieve this, I had to learn how to use Figma, a popular UI design tool. I watched several YouTube videos on Figma, which taught me how to design interfaces, create assets, and collaborate with others on designs. I also consulted my peers to gain a better understanding of Figma and its features and capabilities. As a result of this, I was able to design the UI for my app using Figma effectively.

7.1.3. CONCLUSION

To sum it up, the development of an Expense Management System app using Flutter, Dart, Golang, and MongoDB has required a significant amount of continuing personal development. Throughout the project, I have surfed all the web and watch all the relevant YouTube videos and enrolled for all the available courses and learned the basics and how to use of these new tools and technologies. Through this project, I have gained a wealth of knowledge and experience that will serve me well in my future endeavours.

7.1.4. EVIDENCES

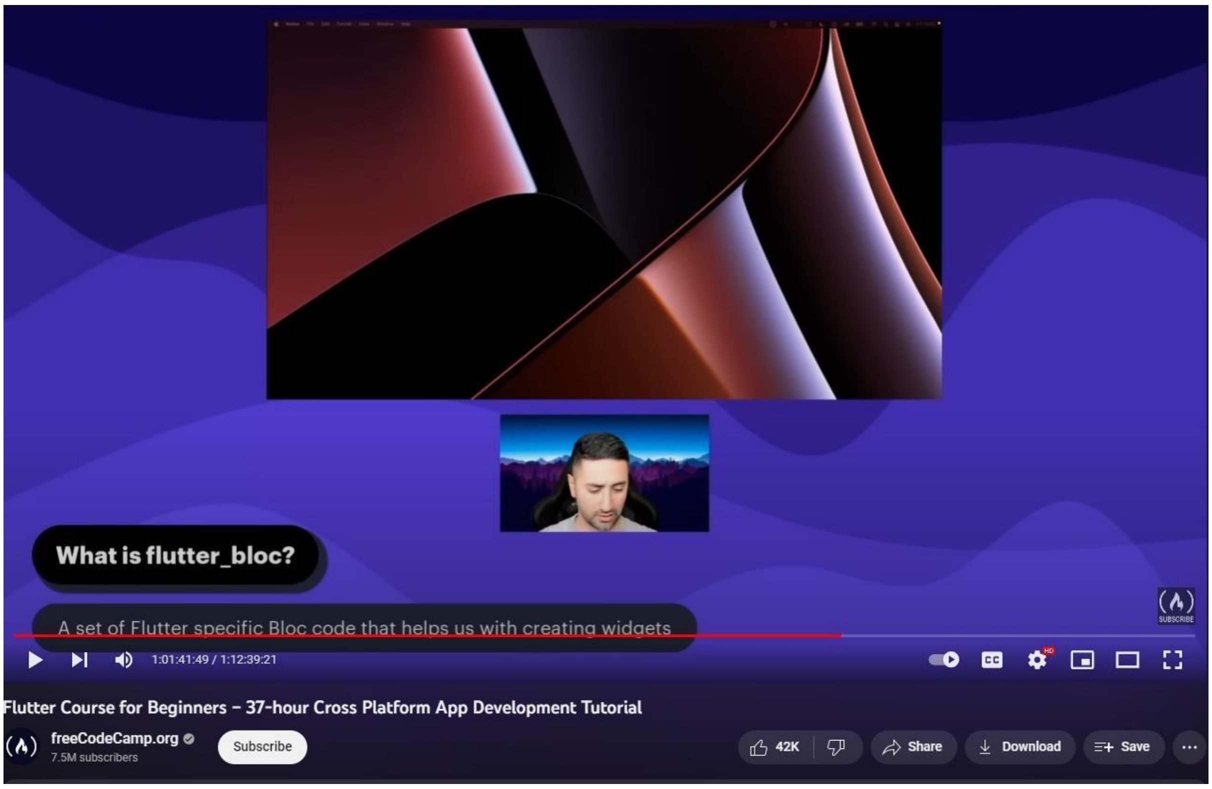
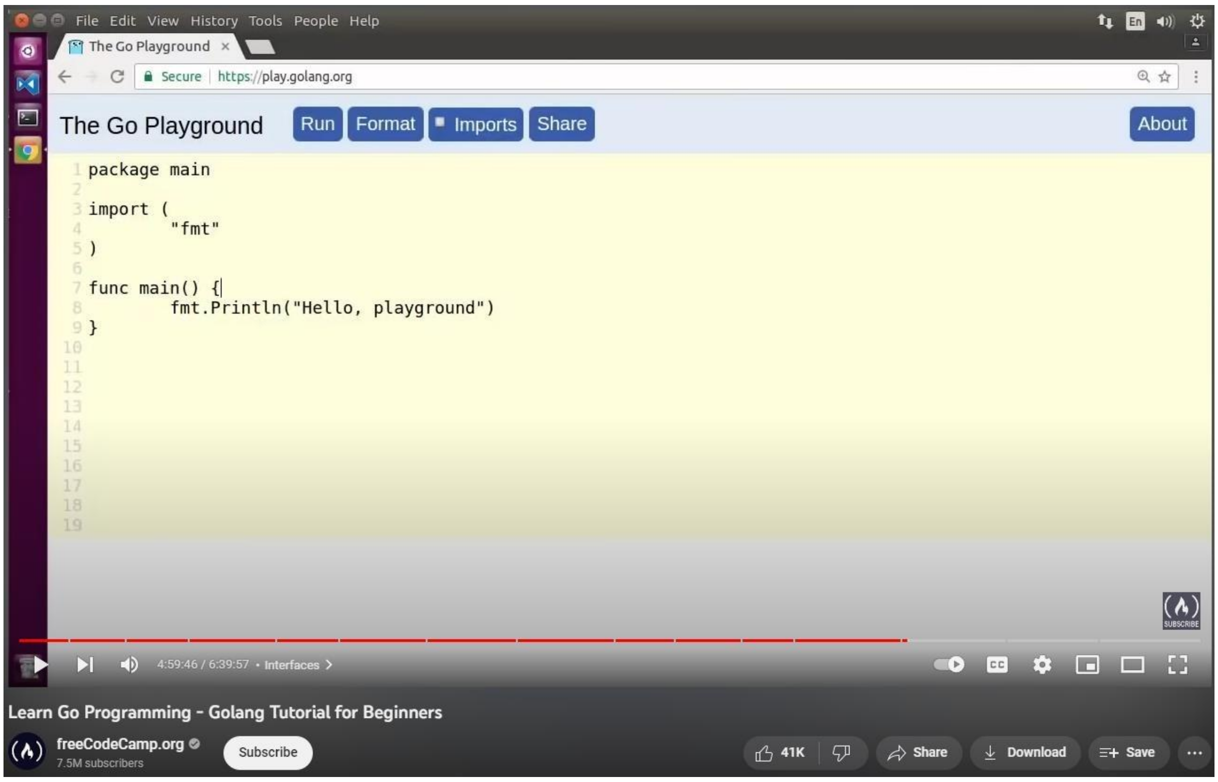
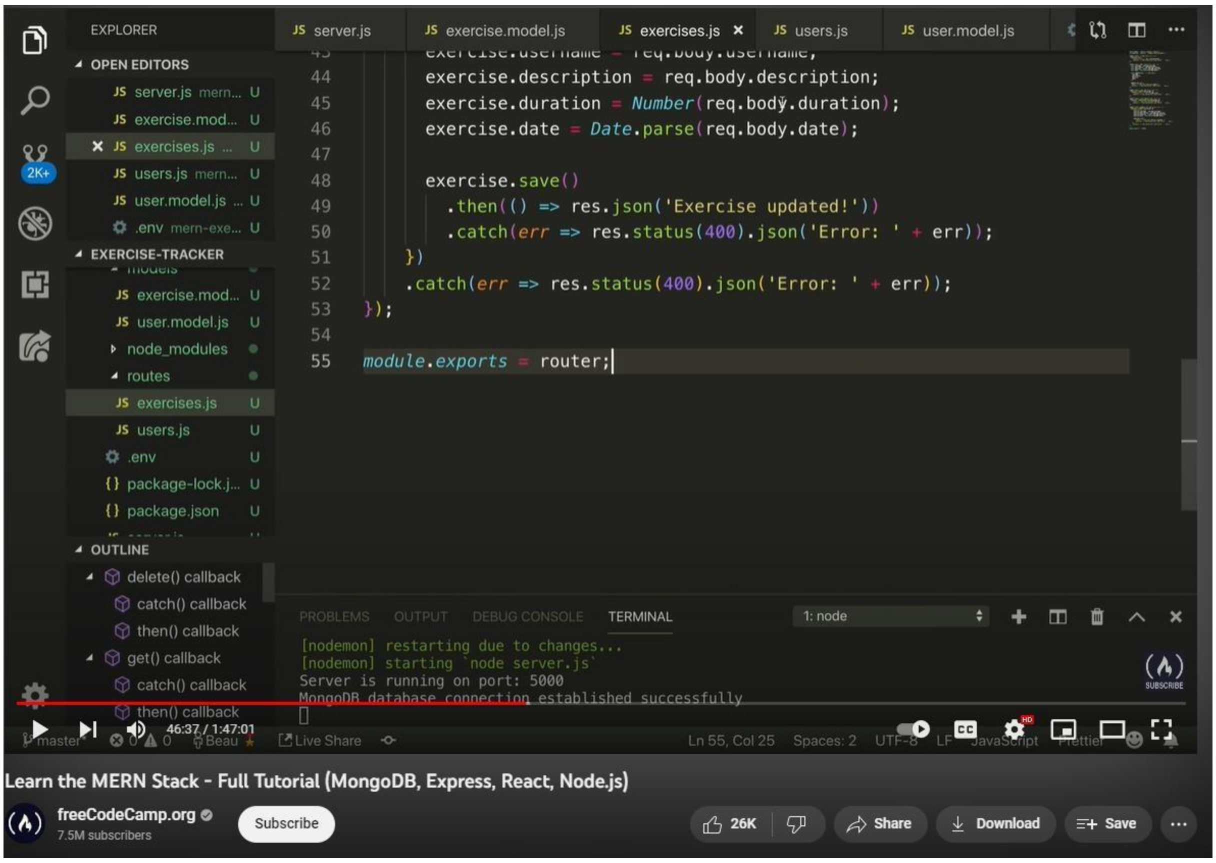

What is flutter_bloc?
A set of Flutter specific Bloc code that helps us with creating widgets
1:01:41:49 / 1:12:39:21
Flutter Course for Beginners – 37-hour Cross Platform App Development Tutorial
freeCodeCamp.org
7.5M subscribers
Subscribe
42K
Share
Download
Save

Figure 20. freeCodeCamp Flutter Course



```go
package main

import (
	"fmt"
)

func main() {
	fmt.Println("Hello, playground")
}
```

Figure 21. freeCodeCamp Golang Course



```javascript
    exercise.description = req.body.description;
    exercise.duration = Number(req.body.duration);
    exercise.date = Date.parse(req.body.date);

    exercise.save()
      .then(() => res.json('Exercise updated!'))
      .catch(err => res.status(400).json('Error: ' + err));
  })
  .catch(err => res.status(400).json('Error: ' + err));
});

module.exports = router;
```

```
[nodemon] restarting due to changes...
[nodemon] starting `node server.js`
Server is running on port: 5000
MongoDB database connection established successfully
```

Figure 22. freeCodeCamp MERN Stack Course

8. Appendix E

8.1. Evidences of Issue tracking

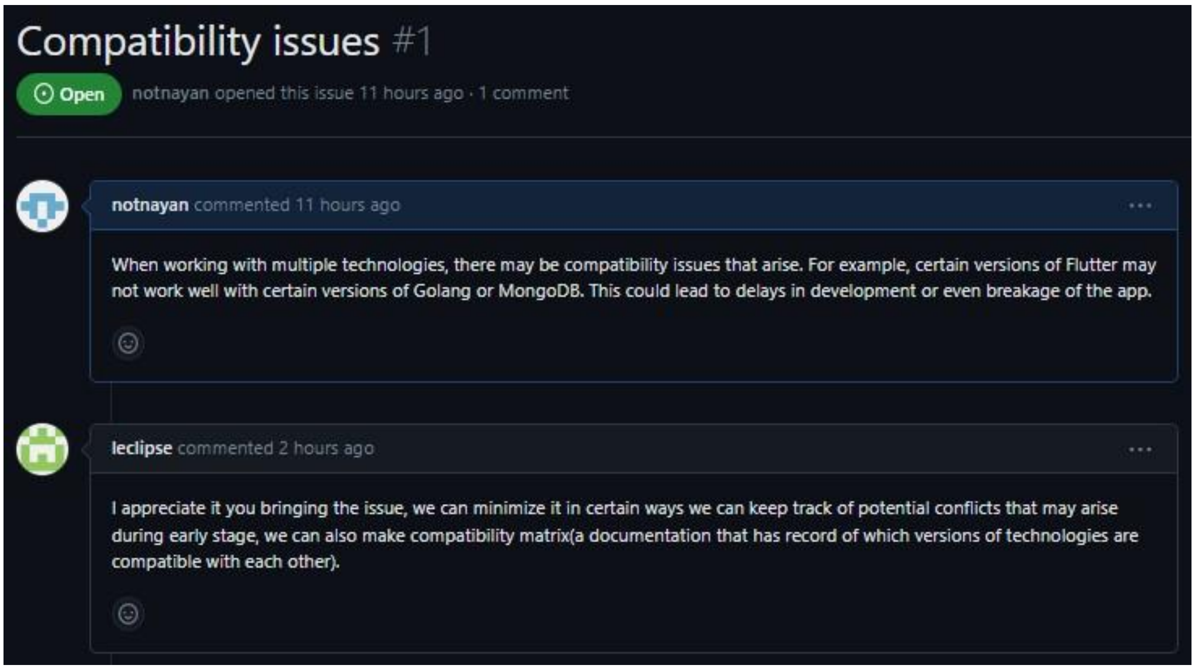
8.1.1. INTRODUCTION

As a developer, it is my responsibility to provide logical solutions to the problems in hand. There were a handful of issues brought up when we picked up the project. But due to the collaboration of the team we were able to overcome it. For every problem there is a solution.

8.1.2. ISSUE TRACKING

We used GitHub Issues as our primary place for monitoring and handling issues concerning our project as it allows for easy bug tracking and also it can track feature requests, documentation updates, and other project-related tasks. The centralized location allowed the Project Manager and the Developers to collaborate and discuss protentional solutions. The tracking system of GitHub is easy to use and understand. To ensure effective issue tracking the user or here the Business Analysts could create a new issue ticket containing vital information, such as the title, description, and any relevant files. (Issues, 2023)

8.1.3. EVIDENCES


Compatibility issues #1

Open notnayan opened this issue 11 hours ago · 1 comment

notnayan commented 11 hours ago

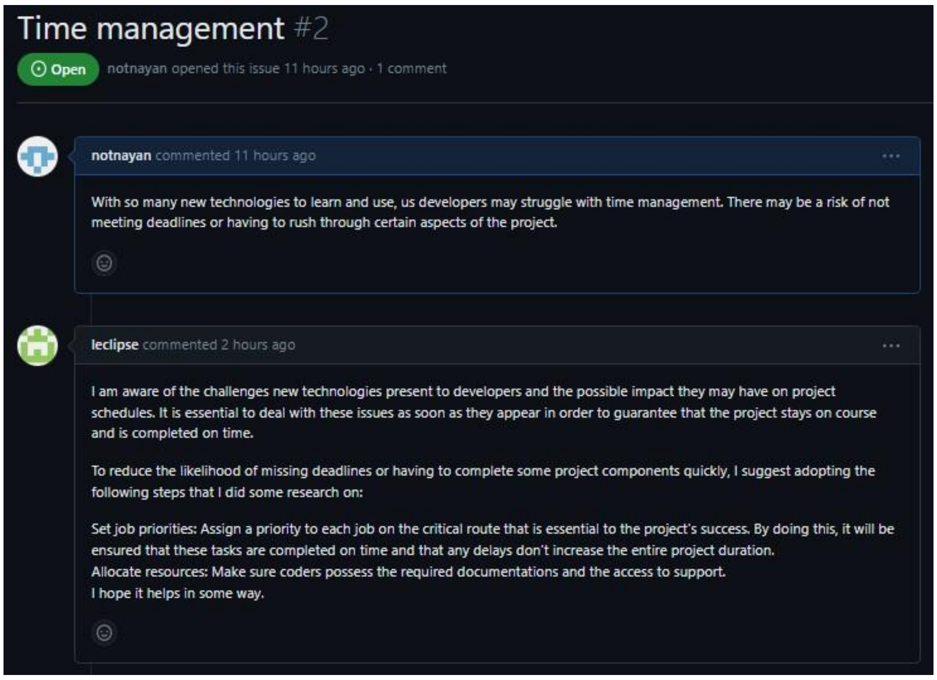
When working with multiple technologies, there may be compatibility issues that arise. For example, certain versions of Flutter may not work well with certain versions of Golang or MongoDB. This could lead to delays in development or even breakage of the app.

leclipse commented 2 hours ago

I appreciate it you bringing the issue, we can minimize it in certain ways we can keep track of potential conflicts that may arise during early stage, we can also make compatibility matrix(a documentation that has record of which versions of technologies are compatible with each other).

Figure 23. Issue (i)

I made an issue regarding the compatibilities issues we might face on our way as we'll be using multiple technologies new to us, posing a risk of delays or breakage of code.


Time management #2

Open notnayan opened this issue 11 hours ago · 1 comment

notnayan commented 11 hours ago

With so many new technologies to learn and use, us developers may struggle with time management. There may be a risk of not meeting deadlines or having to rush through certain aspects of the project.

leclipse commented 2 hours ago

I am aware of the challenges new technologies present to developers and the possible impact they may have on project schedules. It is essential to deal with these issues as soon as they appear in order to guarantee that the project stays on course and is completed on time.

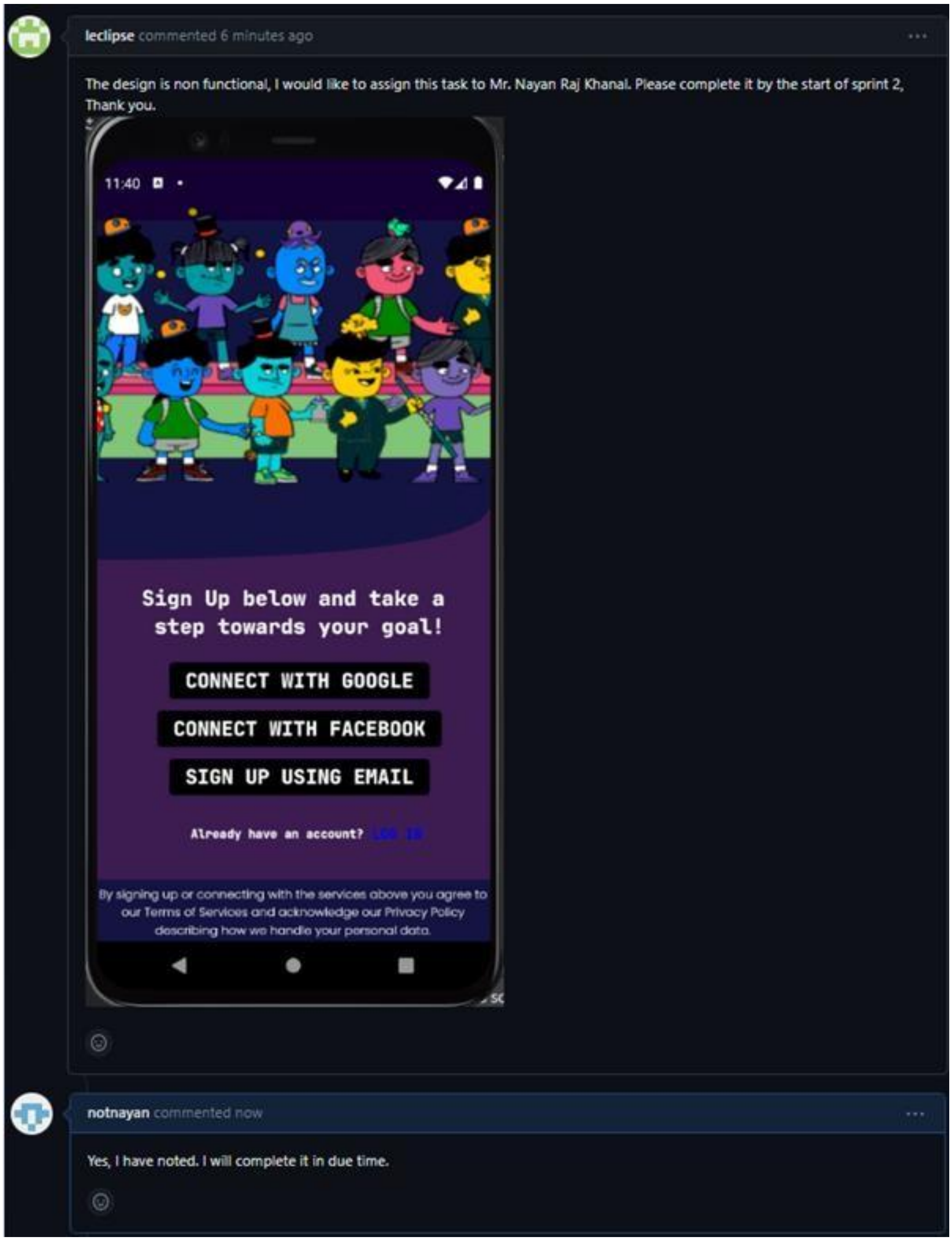
To reduce the likelihood of missing deadlines or having to complete some project components quickly, I suggest adopting the following steps that I did some research on:

Set job priorities: Assign a priority to each job on the critical route that is essential to the project's success. By doing this, it will be ensured that these tasks are completed on time and that any delays don't increase the entire project duration.
Allocate resources: Make sure coders possess the required documentations and the access to support.
I hope it helps in some way.

Figure 24. Issue (ii)

I made another issue relative to the first. Many new technologies mean new stuff to learn hence more time consuming and risks of not meeting the deadline.

Here, our BA tested and found an issue of the app being nonfunctional and tasked me to complete it by the start of sprint 2. I have already started the task.



leclipse commented 6 minutes ago

The design is non functional, I would like to assign this task to Mr. Nayan Raj Khanal. Please complete it by the start of sprint 2, Thank you.

11:40
Sign Up below and take a step towards your goal!
CONNECT WITH GOOGLE
CONNECT WITH FACEBOOK
SIGN UP USING EMAIL
Already have an account? LOG IN
By signing up or connecting with the services above you agree to our Terms of Services and acknowledge our Privacy Policy describing how we handle your personal data.

notnayan commented now

Yes, I have noted. I will complete it in due time.

Figure 25. Issue (iii)

Here, our BA tested and found an issue of the app being nonfunctional and tasked me to complete it by the start of sprint 2. I have already started the task.

8.1.4. CONCLUSION

In conclusion, issue tracking is an essential component of software development that makes it simple to monitor and manage issues that arise throughout the course of a project. With this approach, effective teamwork is promoted, resulting in the creation of effective solutions to problems. Thanks to GitHub Issues we were able to track and solve the underlying issues of our project. The evidence provided in this report emphasizes how crucial issue tracking is to a project's success. Compatibility, learning new technologies, and unusable app functionality were among the problems found; they were resolved in the allotted time.

9. References

Chris, K. (2021, October 7). *What is Go? Golang Programming Language Meaning Explained*. Retrieved from freeCodeCamp: https://www.freecodecamp.org/news/what-is-go-programming-language/

Issues, A. (2023, April 1). *About issues*. Retrieved from GitHub Docs: https://docs.github.com/en/issues/tracking-your-work-with-issues/about-issues

MongoDB, W. i. (2023, April 1). *What Is MongoDB?* Retrieved from MongoDB: https://www.mongodb.com/what-is-mongodb

Technologies, M. A. (2023, April 1). *Mobile App Development Technologies*. Retrieved from orient: https://www.orientsoftware.com/technologies/mobile-technologies/

Thomas, G. (2019, December 12). *What is Flutter and Why You Should Learn it in 2020*. Retrieved from freeCodeCamp: https://www.freecodecamp.org/news/what-is-flutter-and-why-you-should-learn-it-in-2020/

10. Appendix

GitHub Repository Link: https://github.com/A-eli/Expense-Management-System

Figma UI Link: https://www.figma.com/file/iqENJ4DBNfcxAM98x1ileN/DarkModeUI?node-id=28%3A65&t=3DlbCp1lykXWcQ5p-1

freeCodeCamp 1: https://www.youtube.com/watch?v=VPvVD8t02U8

freeCodeCamp 2: https://www.youtube.com/watch?v=YS4e4q9oBaU&t=22s

freeCodeCamp 3: https://www.youtube.com/watch?v=7CqJlxBYj-M