

Contents

Goals.....	4
Targeted Users	4
Data Flow Diagram	4
Sequence Diagram.....	5
Software Design Decisions	6
Justification of Software Design Decisions	6
Libraries Used and Code	7
Social, Legal, Ethical and Security (SLES) Issues	14
Social Issues.....	14
Legal Issues	15
Ethical Issues	15
Security Issues	16
Application's Usage	Error! Bookmark not defined.
Development Strategy.....	16
Phases	17
Sprint App Objective	17
Phase-1.....	17
Phase-2.....	20
Phase-3.....	20
Phase-4.....	22
Burn-down Chart.....	23
Burn-down for Sprint-1.....	23
Burn-down for Sprint-2.....	23
Burn-down for Sprint-3.....	23
Burn-down for Sprint-4.....	24
Firebase DB at backend and regarding API authentication and searching.....	25
Trip Planner Application (UI).....	28
GIT	35
1 st commit	35
2 nd commit	38
3 rd commit.....	39
Reference.....	41

Table of Figures

Figure 1. Data Flow Diagram I 5

Figure 2. Sequence Diagram 5

Libraries Used and Code

Figure 3. Navigation routing with Libraries and developed Code 7

Figure 4. Screen routing with Libraries and developed Code 8

Figure 5. Book Screen UI with Libraries and developed Code 8

Figure 6. Contact Us Screen UI with Libraries and developed Code 9

Figure 7. Detail Screen with Libraries and developed Code 10

Figure 8. NavDrawer with Libraries and developed Code 11

Figure 9. Custom Search View utils with Libraries and developed Code 12

Figure 10. Extension utils with Libraries and developed Code 13

Figure 11. Trip Field 14

Development Strategy

Figure 12. Sprint App Objectives and development procedure 17

Figure 13. Phase I (i) 18

Figure 14. Phase I (ii) 18

Figure 15. Phase I Sprint I and II 19

Figure 16. Phase I Sprint III and IV 19

Figure 17. Phase II 20

Figure 18. Phase III 21

Figure 19. Phase IV 22

Figure 20. Phase IV (ignored tasks and unable to complete tasks) 22

Burn-down Chart

Figure 21. Burn Down Chart Sprint I 23

Figure 22. Burn Down Chart Sprint II 23

Figure 23. Burn Down Chart Sprint III 24

Figure 24. Burn Down Chart Sprint IV	24
Firebase DB at backend and regarding API authentication and searching	
Figure 25. Firebase DB	25
Figure 26. Firebase Voting app overview	25
Figure 27. Firebase authentication	25
Figure 28. Sign in methods in firebase.....	26
Figure 29. Gmail log in method in firebase	26
Figure 30. Cloud firestore.....	26
Figure 31. Collection list in firebase	27
Figure 32. Collection starting in DB.....	27
Trip Planner Application (UI)	
Figure 33. Welcome Page.....	28
Figure 34. Registration Page.....	28
Figure 35. Login Page	29
Figure 36. Home Page I.....	29
Figure 37. Home Page (with dropdown I).....	30
Figure 38. Home Page (with dropdown II)	31
Figure 39. Details Page I.....	31
Figure 40. Details Page II.....	32
Figure 41. Details Page III	32
Figure 42. Details Page IV	33
Figure 43. Details Page V.....	33
Figure 44. Transportation Booking Page	34
Figure 45. Contact Us Page	34
Figure 46. Log Out Page	35
GIT	
Figure 47. Ist Commit (overview)	35
Figure 48. Ist Commit (Master).....	36
Figure 49. Ist Commit (Master II)	37

Figure 50. Ist Commit (Master III)	37
Figure 51. IInd Commit (Overview)	38
Figure 52. IInd Commit (Master)	38
Figure 53. IInd Commit (Code)	39
Figure 54. IIIRD Commit (Overview)	39
Figure 55. IIIRD Commit (Master)	40
Figure 56. IIIRD Commit (Code)	40

Trip Planner Application

Goals

The targeted goal behind creating this application is to simplify or streamline the process of planning a trip for travellers. It provides tools for looking up and reserving trip locations and modes of transportation, and it may even let users compare costs from other travel agencies. The application's traveller region has features for managing bookings, logging in, and searching for trips. Finally, we can say that this application provides a user-friendly platform to the travellers with the help of which users can plan their travel journey online at any time.

Targeted Users

- Travellers

Data Flow Diagram

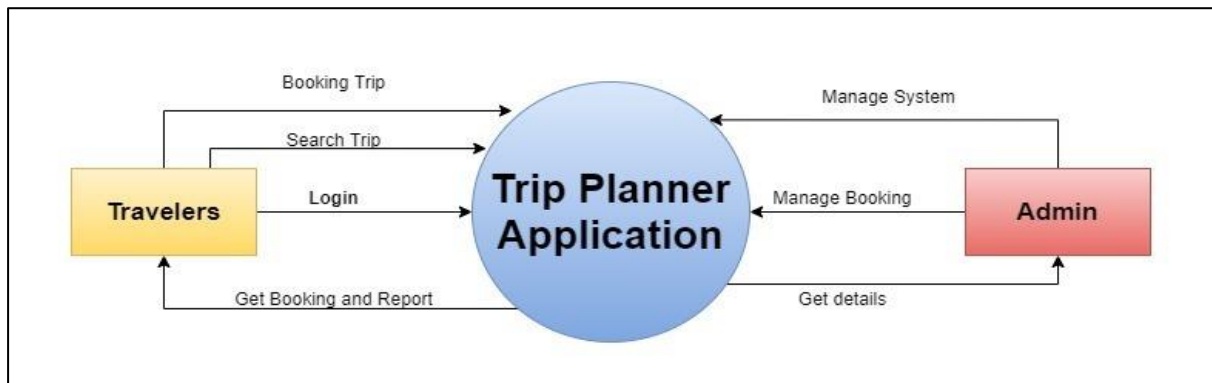


Figure 1. Data Flow Diagram I

Sequence Diagram

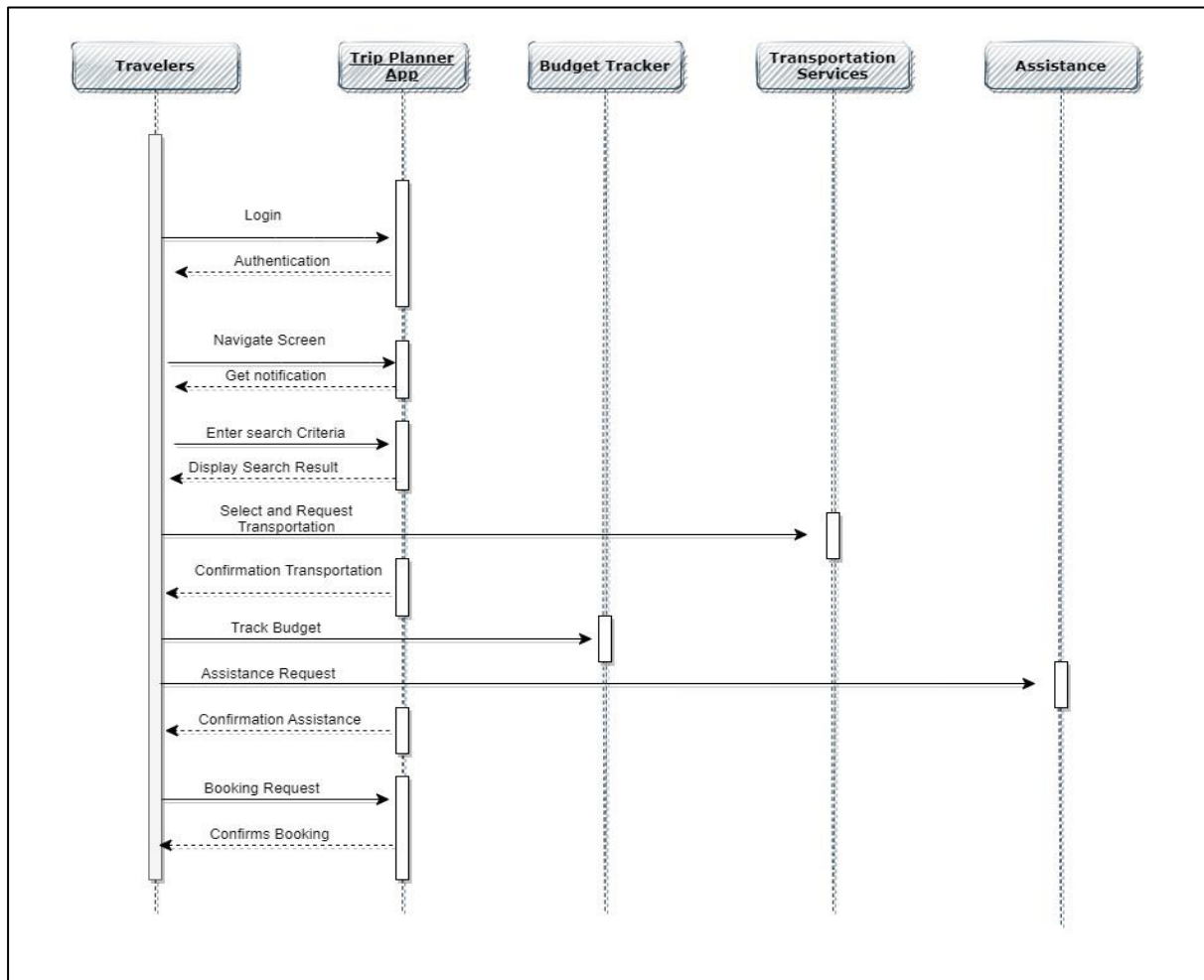


Figure 2. Sequence Diagram

Software Design Decisions

Android Studio was used to construct the application, which was then tested on an emulator. The Java programming language is used to create the application.

Justification of Software Design Decisions

Design & Layout:

The majority of app activities employ a linear structure and scroll view to support various screen sizes.

Bottom navigation bar and top app bar are consistent throughout the application.

User Experience:

user development for tutors was given priority over that for students.

The student user section should come before the student session features are added.

Libraries Used and Code

```
package com.tripplanner.routing

import androidx.compose.runtime.Composable
import androidx.navigation.compose.NavHost
import androidx.navigation.compose.composable
import androidx.navigation.compose.rememberNavController
import com.tripplanner.ui.bookScreen.BookScreen
import com.tripplanner.ui.contact_us.ContactUsScreen
import com.tripplanner.ui.detail.DetailScreen
import com.tripplanner.ui.login.LoginScreen
import com.tripplanner.ui.main.MainScreen
import com.tripplanner.ui.profileScreen.ProfileScreen
import com.tripplanner.ui.register.RegisterScreen
import com.tripplanner.ui.searchListing.SearchScreen
import com.tripplanner.ui.splash.SplashScreen

@Composable
fun Navigation() {
    val navController = rememberNavController()

    NavHost(
        navController = navController,
        startDestination = Screen.SplashScreen.route
    ) {
        composable(route = Screen.SplashScreen.route) {
            SplashScreen(navController = navController)
        }
        composable(route = Screen.LoginScreen.route) {
            LoginScreen(navController = navController)
        }
        composable(route = Screen.RegisterScreen.route) {
            RegisterScreen(navController = navController)
        }
        composable(route = Screen.MainScreen.route) {
            MainScreen(navController = navController)
        }
        composable(route = Screen.Detail.route+ "{name}/{index}") {
            val name = it.arguments?.getString("name")
            val index = it.arguments?.getString("index")
            if(name!=null && index!=null) {
                DetailScreen(navController = navController,name = name,index=index)
            }
        }
        composable(route = Screen.SearchList.route) {
            SearchScreen(navController = navController)
        }
        composable(route = Screen.ProfileScreen.route) {
            ProfileScreen(navController = navController)
        }
        composable(route = Screen.ContactUsScreen.route) {
            ContactUsScreen(navController = navController)
        }
    }
}
```

Figure 3. Navigation routing with Libraries and developed Code

```

package com.tripplanner.routing

sealed class Screen(val route: String) {

    object SplashScreen: Screen("splash_screen")
    object LoginScreen: Screen("login_screen")
    object RegisterScreen: Screen("register_screen")
    object MainScreen: Screen("main_screen")
    object SearchList: Screen("search_list")
    object Detail: Screen("detail_screen")
    object ProfileScreen: Screen("profile_screen")
    object ContactUsScreen: Screen("contact_us_screen")
    object BookScreen: Screen("book_screen")

}

```

Figure 4.Screen routing with Libraries and developed Code

```

package com.tripplanner.ui.bookScreen

import android.annotation.SuppressLint
import android.widget.Toast
import androidx.compose.foundation.*
import androidx.compose.foundation.layout.*
import androidx.compose.material.icons.Icons
import androidx.compose.material.icons.rounded.ArrowBack
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.ui.ExperimentalComposeUiApi
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.TextStyle
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.input.KeyboardType
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.navigation.NavController
import com.tripplanner.R
import com.tripplanner.routing.Screen
import com.tripplanner.ui.theme.TripPlannerAppTheme
import com.tripplanner.ui.theme.black
import com.tripplanner.ui.theme.green
import com.tripplanner.ui.theme.white
import com.tripplanner.utils.TripField
import com.tripplanner.utils.RoundedButton
import com.tripplanner.utils.isValidEmail

@OptIn(ExperimentalMaterial3Api::class, ExperimentalComposeUiApi::class)
@SuppressLint("UnusedMaterial3ScaffoldPaddingParameter")
@Composable
fun BookScreen(navController: NavController) {
    val context = LocalContext.current
    val scrollState = rememberScrollState()
    var name by remember { mutableStateOf("") }
    var email by remember { mutableStateOf("") }
    var mobile by remember { mutableStateOf("") }
    var address by remember { mutableStateOf("") }
    var isSubmit by remember { mutableStateOf(false) }
}

```

Figure 5. Book Screen UI with Libraries and developed Code


```

package com.tripplanner.ui.contact_us

import android.annotation.SuppressLint
import android.widget.Toast
import androidx.compose.foundation.*
import androidx.compose.foundation.layout.*
import androidx.compose.material.icons.Icons
import androidx.compose.material.icons.rounded.ArrowBack
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.ui.ExperimentalComposeUiApi
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.text.TextStyle
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.input.KeyboardType
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.navigation.NavController
import com.tripplanner.ui.theme.TripPlannerAppTheme
import com.tripplanner.ui.theme.black
import com.tripplanner.ui.theme.green
import com.tripplanner.ui.theme.white
import com.tripplanner.utils.TripField
import com.tripplanner.utils.RoundedButton
import com.tripplanner.utils.isValidEmail

@OptIn(ExperimentalMaterial3Api::class, ExperimentalComposeUiApi::class)
@SuppressLint("UnusedMaterial3ScaffoldPaddingParameter")
@Composable
fun ContactUsScreen(navController: NavController) {
    val context = LocalContext.current
    val scrollState = rememberScrollState()
    var name by remember { mutableStateOf("") }
    var email by remember { mutableStateOf("") }
    var message by remember { mutableStateOf("") }

```

Figure 6. Contact Us Screen UI with Libraries and developed Code

```

package com.tripplanner.ui.detail

import android.annotation.SuppressLint
import android.widget.Toast
import androidx.compose.foundation.*
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material.icons.Icons
import androidx.compose.material.icons.filled.KeyboardArrowDown
import androidx.compose.material.icons.filled.KeyboardArrowUp
import androidx.compose.material.icons.filled.MoreVert
import androidx.compose.material.icons.rounded.ArrowBack
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.runtime.saveable.rememberSaveable
import androidx.compose.ui.Alignment
import androidx.compose.ui.ExperimentalComposeUiApi
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.graphics.ColorFilter
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.TextStyle
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.compose.ui.window.Popup
import androidx.compose.ui.window.PopupProperties
import androidx.navigation.NavController
import com.tripplanner.R
import com.tripplanner.routing.Screen
import com.tripplanner.ui.theme.TripPlannerAppTheme
import com.tripplanner.ui.theme.black
import com.tripplanner.ui.theme.green
import com.tripplanner.ui.theme.white
import com.tripplanner.utils.RoundedButton
import com.tripplanner.utils.getData

@OptIn(ExperimentalMaterial3Api::class, ExperimentalComposeUiApi::class)
@SuppressLint("UnusedMaterial3ScaffoldPaddingParameter")
@Composable
fun DetailScreen(navController: NavController, name:String,index:String) {
    val context = LocalContext.current
    val scrollState = rememberScrollState()
    val data = getData(index)

    TripPlannerAppTheme {
        Scaffold {

```

Figure 7. Detail Screen with Libraries and developed Code

```

package com.tripplanner.ui.drawer

import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.shape.CircleShape
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.clip
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.unit.dp
import com.tripplanner.R
import com.tripplanner.ui.theme.black
import com.tripplanner.ui.theme.green
import com.tripplanner.ui.theme.white

@Composable
fun DrawerHeader() {
    Spacer(modifier = Modifier.background(green).padding(top = 45.dp))
    Column(
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally,
        modifier = Modifier.background(color = white).padding(top = 45.dp)
            .fillMaxWidth()
    ) {
        Image(
            painterResource(id = R.drawable.ic_trip_planner),
            contentDescription = null,
            contentScale = ContentScale.Crop,
            modifier = Modifier
                .clip(CircleShape).size(150.dp)
        )
        Spacer(modifier = Modifier)

        androidx.compose.material3.Text(
            text = stringResource(id = R.string.app_name),
            textAlign = TextAlign.Center,
            style = MaterialTheme.typography.bodyLarge,
            color = black,
        )
    }
}

```

Figure 8. NavDrawer with Libraries and developed Code

```

package com.tripplanner.utils

import androidx.compose.foundation.background
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.Box
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.shape.CircleShape
import androidx.compose.material.TextField
import androidx.compose.material.TextFieldDefaults
import androidx.compose.material.icons.Icons
import androidx.compose.material.icons.filled.Search
import androidx.compose.material3.Icon
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.clip
import androidx.compose.ui.unit.dp
import com.tripplanner.ui.theme.white

@Composable
fun CustomSearchView(
    search: String,
    modifier: Modifier = Modifier,
    onValueChange: (String) -> Unit,
    onClick: () -> Unit
) {
    Box(
        modifier = modifier
            .fillMaxWidth()
            .padding(10.dp)
            .clip(CircleShape)
            .background(white)
    ) {
        TextField(value = search,
            onValueChange = onValueChange,
            modifier = Modifier.fillMaxWidth(),
            colors = TextFieldDefaults.outlinedTextFieldColors(
                focusedBorderColor = white,
                unfocusedBorderColor = white,
                disabledBorderColor = white,
            ),
            trailingIcon = {
                Icon(
                    imageVector = Icons.Default.Search,
                    contentDescription = "",
                    modifier = Modifier.clickable {
                        onClick()
                    }
                )
            },
        )
    }
}

```

Figure 9. Custom Search View utils with Libraries and developed Code

```

package com.triplanner.utils

import java.util.regex.Pattern

fun isValidEmail(s: String): Boolean {
    val emailPattern = ("^([\\w-]+\\.)+[\\w-]+([a-zA-Z]{1}|[\\w-]{2,}))@"
        + "((((([0-1]?[0-9]|25[0-5])2[0-4][0-9])\\.|([0-1]?
        + "[0-9]{1,2}|25[0-5]|2[0-4][0-9])\\.|
        + "([0-1]?[0-9]|25[0-5]|2[0-4][0-9])\\.|([0-1]?
        + "[0-9]{1,2}|25[0-5]|2[0-4][0-9])){1})|
        + "([a-zA-Z]+[\\w-]+\\.)+[a-zA-Z]{2,4})$")
    val pattern = Pattern.compile(emailPattern)
    val matcher = pattern.matcher(s)

    return !matcher.matches()
}

fun getData(index:String) : String {
    var data = ""
    when(index) {
        "0" -> {
            data = "Delhi>>Paris>>Amsterdam>>switzerland(multiple cities)>>Venice>>Pisa>>Rome>>delhi .\n" +
                "\n" +
                "Itinerary:\n" +
                "•\tParis-3 days\n" +
                "•\tAmsterdam-2 days\n" +
                "•\tSwitzerland-3 days\n" +
                "•\tVenice-2 days\n" +
                "•\tPisa-half day\n" +
                "•\tRome-3 and a half days\n" +
                "•\t24 hours flight -to and fro\n" +
                "\n" +
                "\n" +
                "Expenses: 1700 Gbp for each person\n" +
                "\n" +
                "You can calculate based on number of persons and can display the total expenses approximately.\n"
        }
        "1" -> {
            data = "Day 1: Departure from Delhi\n" +
                "•\tFly from Delhi to Paris, arrive in Paris is Charles de Gaulle Airport (CDG).\n" +
                "Day 2-4: Exploring Paris\n" +
                "•\tmust-visit places include the Eiffel Tower, Louvre Museum, Notre-Dame Cathedral, Champs-Élysées, Montmartre, and the Seine River.\n" +
                "•\tTake a cruise along the Seine River to enjoy the scenic beauty of Paris.\n" +
                "\n" +
                "Day 5: Day Trip to Versailles\n" +
                "•\tTake a day trip to the Palace of Versailles, located just outside of Paris. Explore the stunning palace, its gardens, and the Hall of Mirrors.\n" +
                "\n" +
                "•\t.\n" +
                "Day 6: Cultural Immersion\n" +
                "•\tVisit art galleries, charming boutiques, and local markets.\n" +
                "•\tIndulge in French pastries and macarons at renowned patisseries.\n" +
                "•\tEnjoy a traditional French dinner at a local bistro.\n" +
                "•\t\n"
        }
    }
}

```

Figure 10. Extension utils with Libraries and developed Code

```

package com.tripplanner.utils

import androidx.compose.foundation.clickable
import androidx.compose.foundation.focusable
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.foundation.text.KeyboardOptions
import androidx.compose.material3.*
import androidx.compose.runtime.Composable
import androidx.compose.ui.ExperimentalComposeUiApi
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.input.KeyboardCapitalization
import androidx.compose.ui.text.input.KeyboardType
import androidx.compose.ui.text.input.VisualTransformation
import androidx.compose.ui.unit.Dp
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp

@OptIn(ExperimentalMaterial3Api::class, ExperimentalComposeUiApi::class)
@Composable
fun TripField(
    value: String,
    backgroundColor : Color,
    placeholder: String,
    onChange: (String) -> Unit,
    onClick: () -> Unit = {},
    isEnabled: Boolean = true,
    readOnly: Boolean = false,
    keyboardType: KeyboardType = KeyboardType.Text,
    keyboardCapitalization: KeyboardCapitalization = KeyboardCapitalization.Sentences,
    visualTransformation: VisualTransformation = VisualTransformation.None,
    leadingIcon: @Composable (() -> Unit)? = null,
    trailingIcon: @Composable (() -> Unit)? = null,
    horizontalPadding : Dp = 0.dp,
    verticalPadding : Dp = 3.dp,
    colors : TextFieldColors? = null,
    label : @Composable (() -> Unit)? = null,
) {

    OutlinedTextField(
        value = value,
        shape = RoundedCornerShape(10.dp),
        keyboardOptions = KeyboardOptions(keyboardType = keyboardType, capitalization = keyboardCapitalization),
        visualTransformation = visualTransformation,
        enabled = isEnabled,
        readOnly = readOnly,

```

Figure 11. Trip Field

Social, Legal, Ethical and Security (SLES) Issues

Social Issues		
S. No.	Issues	Description
1.	Inclusivity and Accessibility	The primary tenet of the Trip Planer mobile app is that anyone may use it and feel involved, regardless of economic background, language barrier, or disability. The Trip Planer app should ensure that users from a various of backgrounds may use it.
2.	Awareness of Cultural Differences	The Trip Planer app ought to make an effort to respect cultural diversity and be sensitive when making recommendations and ideas; this includes honouring local traditions and customs. To prevent

		misrepresentations, the app needs to work with local specialists.
3.	Impact on Community	The Trip Planer app should make an effort to mitigate any potential harm that it may cause to small businesses and local populations in popular tourist areas, such as overcrowding due to traffic jams.
4.	Environment Effectiveness	The Trip Planer app should promote the sustainable and responsible trip planning practices, which helps to minimize carbon footprint and support eco-friendly accommodations and transportation options.

Legal Issues

1.	Intellectual Property Rights Policy	The Trip Planer mobile app attempts to obtain the necessary licenses and permits from the relevant authorities in order to use images, videos, and other content related to the tourist destination according to the Copyright and Trademark Policy of UK
2.	Content Accuracy and Liability	The accuracy of the information should be confirmed and explained using the Trip Planer mobile application, especially in regards to operation hours, entry fees, and recommendations for tourist attractions. A disclaimer on the app's responsibility for content inaccuracies and any accidents that might occur when users visit places that are suggested to them should also be included. For this the following laws are need to be obeyed as per governmental authorities- ▪ Consumer Rights Act 2015 (CRA 2015) ▪ Misrepresentation Act 1968 ▪ Product Liability Act 1994
3.	Service Conditions	The Trip Planer app needs to make it clear what the developer's, and user's and travel agencies legal obligations are in the event of cancellations, disagreements, or accidents that occur while it is being planned.

Ethical Issues

1.	Transparency	The Trip Planer Mobile Application should try to provide clear and accurate information about trip options, terms and conditions and prices to empower user to make informed decisions.
----	---------------------	---

2.	Entanglements in Interest	It should be doable for the Trip Planer mobile application to steer clear of conflicts of interest while making recommendations or making reservations, such as endorsing particular travel agencies or locations in order to make money.
3.	Sophisticated Planning	The Trip Planer app needs to consider how it can guarantee current and reliable travel planning information for the benefit of its users.
Security Issues		
1.	Data Security of User	To prevent unauthorized access or breaches of user's personal information, including travel preferences and booking details, the Trip Planer app must incorporate a strong and robust data security system.
2.	Payment Security	The security of online transactions made via the Trip Planer mobile app should be guaranteed. This includes encrypted payment processing and safe card information storage.
3.	Authentication	The Trip Planer Mobile Application should implement strong authentication mechanisms to verify the identities of users and travel agents accessing the app via email safety and notification.
4.	Travel Safety	The Trip Planer app need to be used to offer trustworthy data and tools, such as risk evaluations for destinations, emergency contacts, and travel advisories, to guarantee user safety and security while they are on the road.

Development Strategy

For the development of this Trip Planer Mobile Application, the one of the best method i.e., Agile methodology is used. The Agile method ensures continuous development of the application.

For the development of this Trip Planer Mobile Application, the Development Strategy is divided into one 'Sprint App Objective' and four 'sprints'. Each sprint is used for the guidance of the tasks as cards.

Phases

Sprint App Objective

The Sprint App Objective consist of the task that need to consider and we want to develop for this application. The cards I have used in the Sprint App Objective are-

- * Fundamental Requirement of the Online Voting System Application
- * Development of Online Voting System Application
- * Code Development and Database schema
- * GIT Deployment
- * Testing and UAT
- * Documentations

The following images describes the Sprint App Objective along with the development procedure.

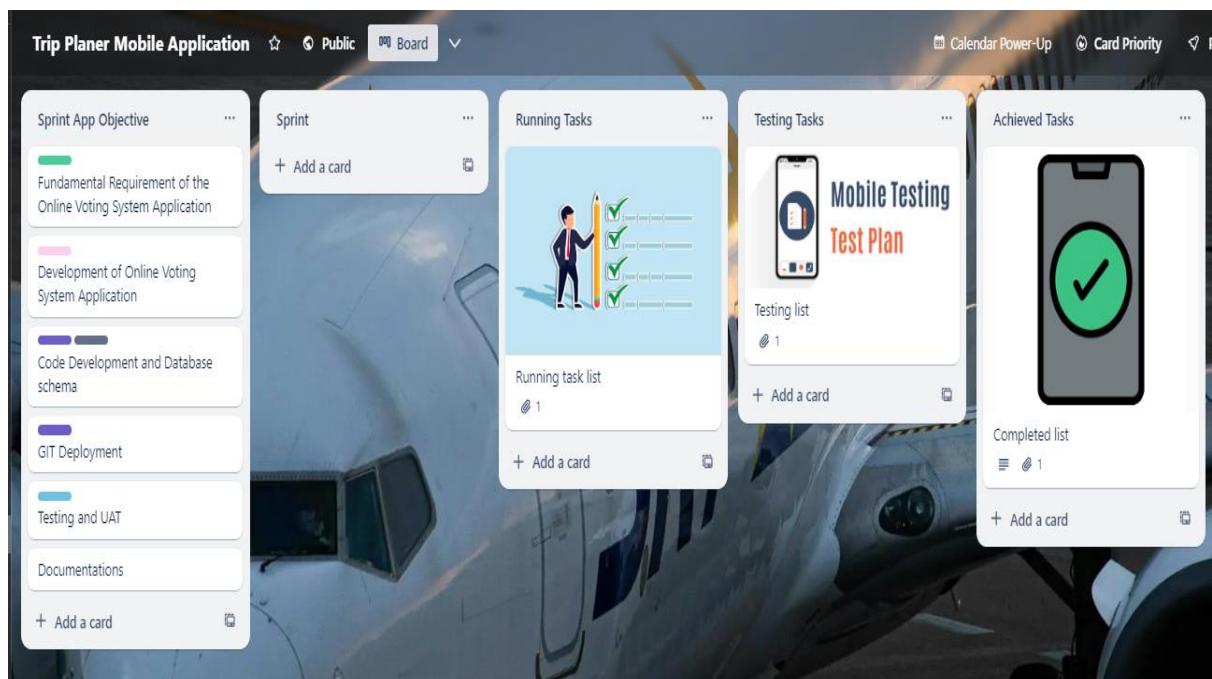


Figure 12. Sprint App Objectives and development procedure

Phase-1

The enablement of the all four sprints are consists in this phase. The titles for each need to do sprints are as-

- * Sprint-1: Requirement Gathering
- * Sprint-2: Wire Frame with UI & UX
- * Sprint-3: Coding, Git Repository for Version Control of UI design, Database schema
- * Sprint-4: Testing of mobile App, Deployment, UAT & Documentation

The 'Need to Do' Tasks are mentioned in the following image-

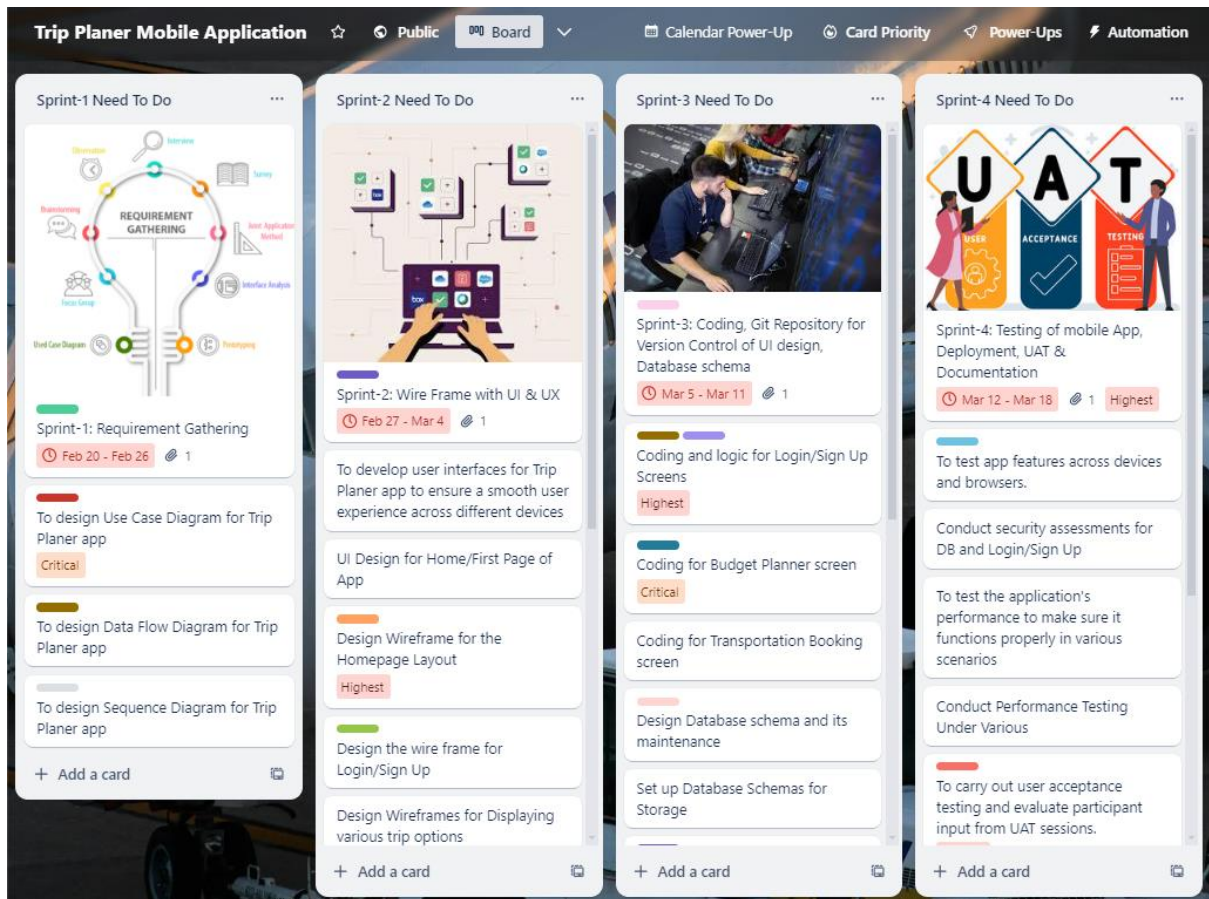


Figure 13. Phase I (i)

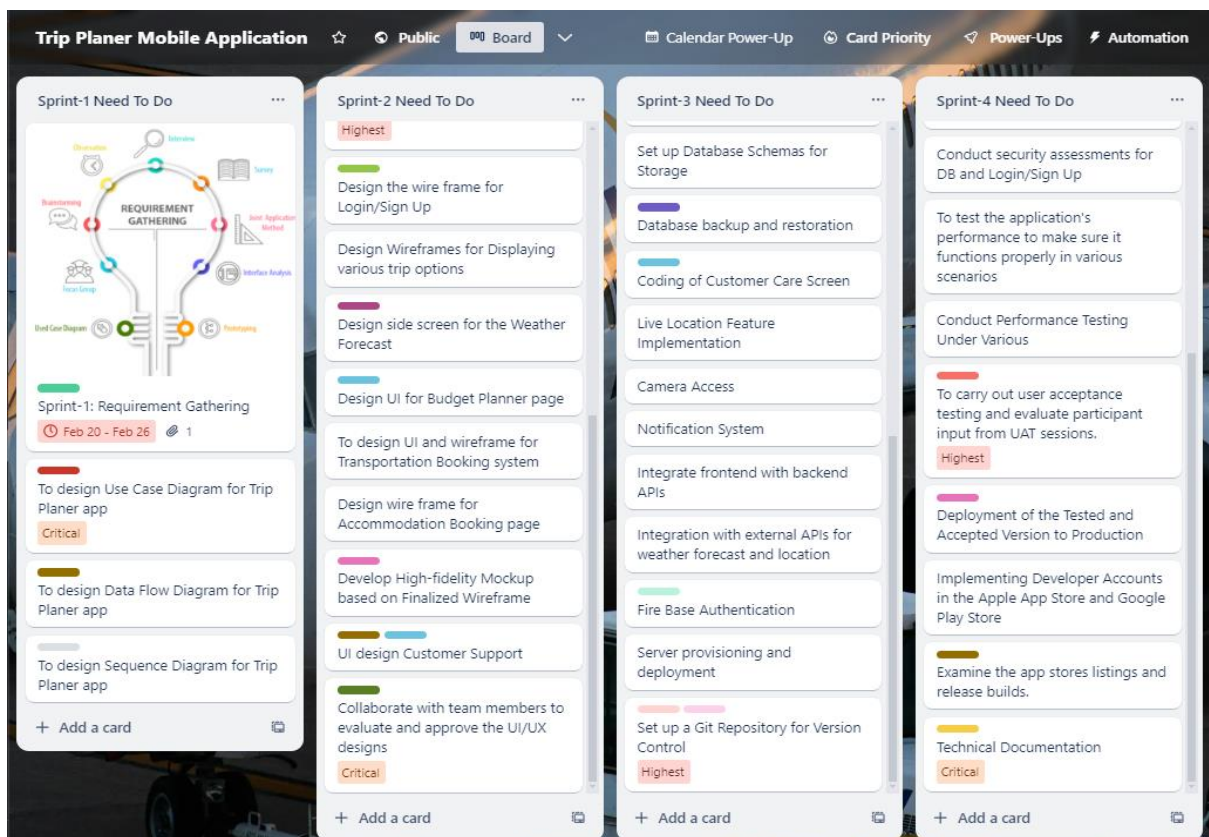


Figure 14. Phase I (ii)

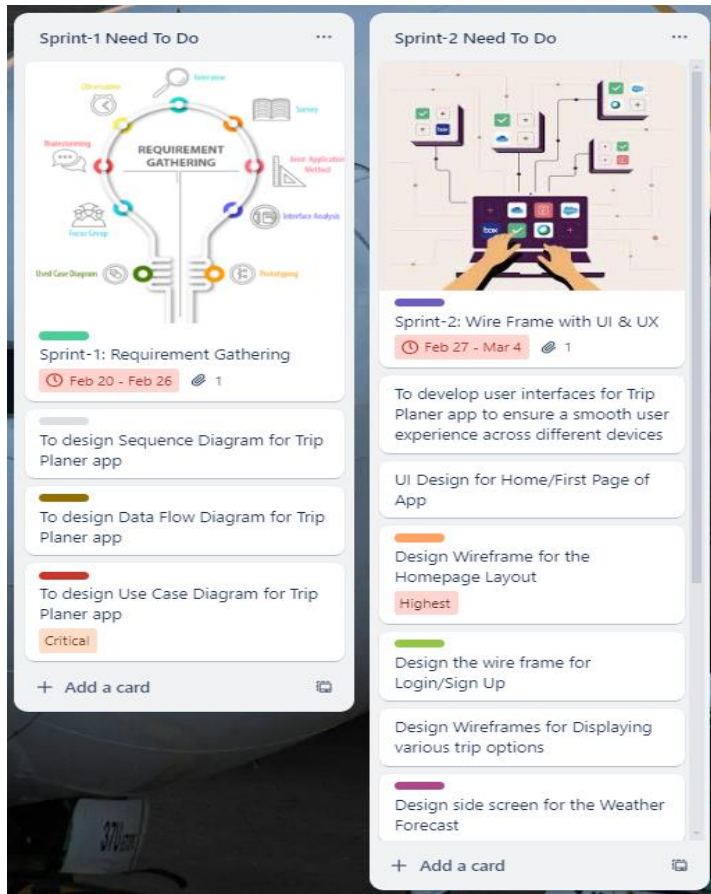


Figure 15. Phase I Sprint I and II

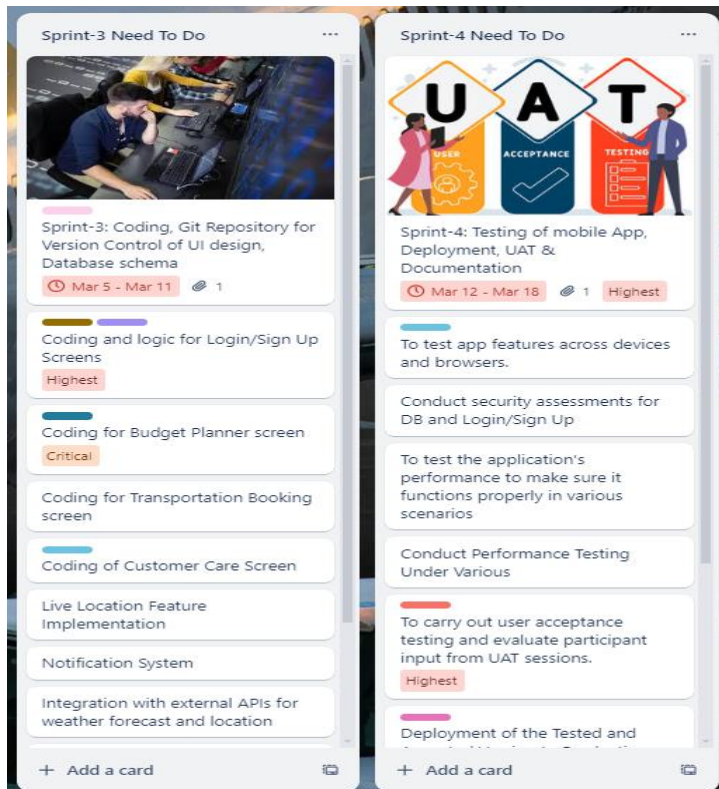


Figure 16. Phase I Sprint III and IV

Phase-2

In this phase-2, all the sprint tasks that are on progress are moved in the 'Running Tasks' list. The following image represent the 'Running Tasks' list-

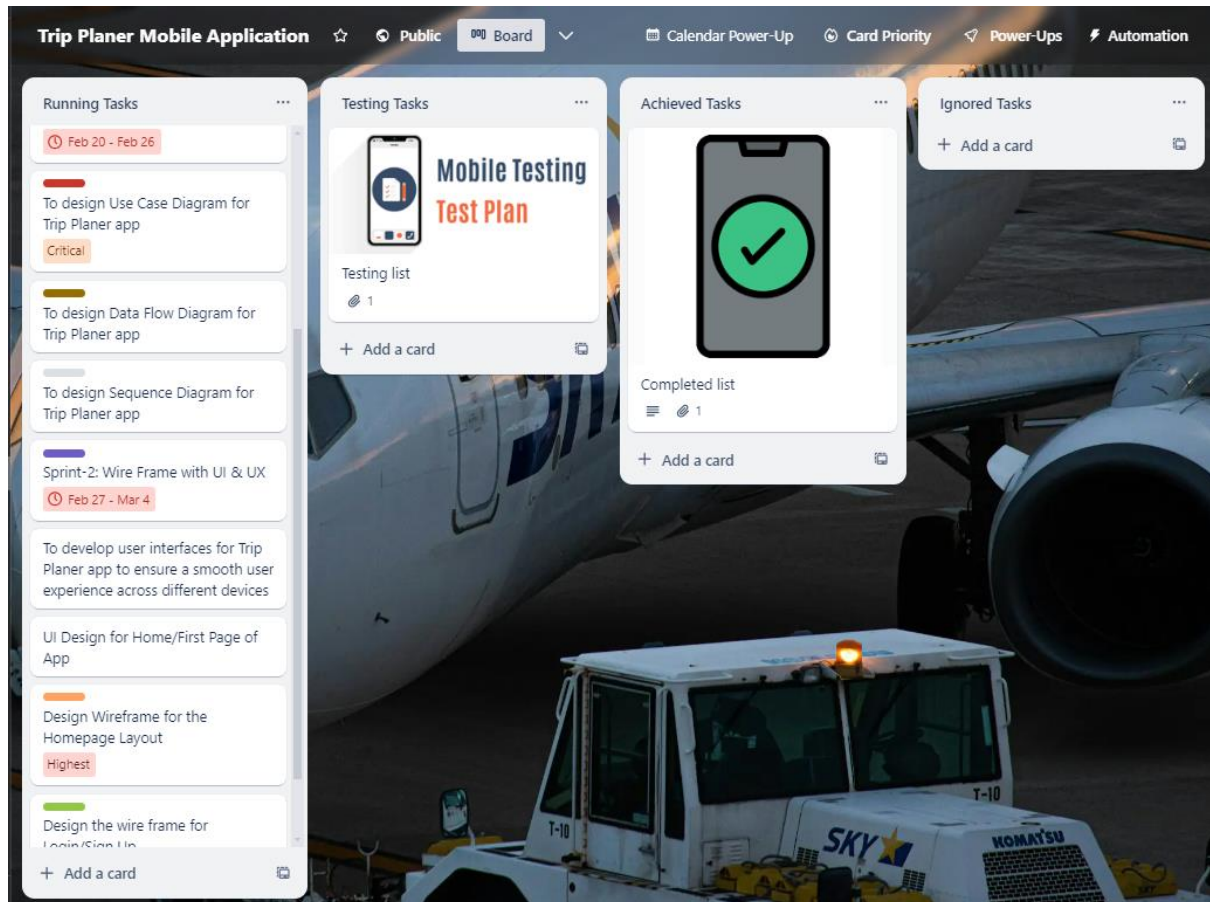


Figure 17. Phase II

Phase-3

The phase-3 describes the implementation of the testing of the developed application in order to verify that this app can perform at saviour load conditions or not. For the testing purpose the sprint, 'Testing Tasks' is created by me.

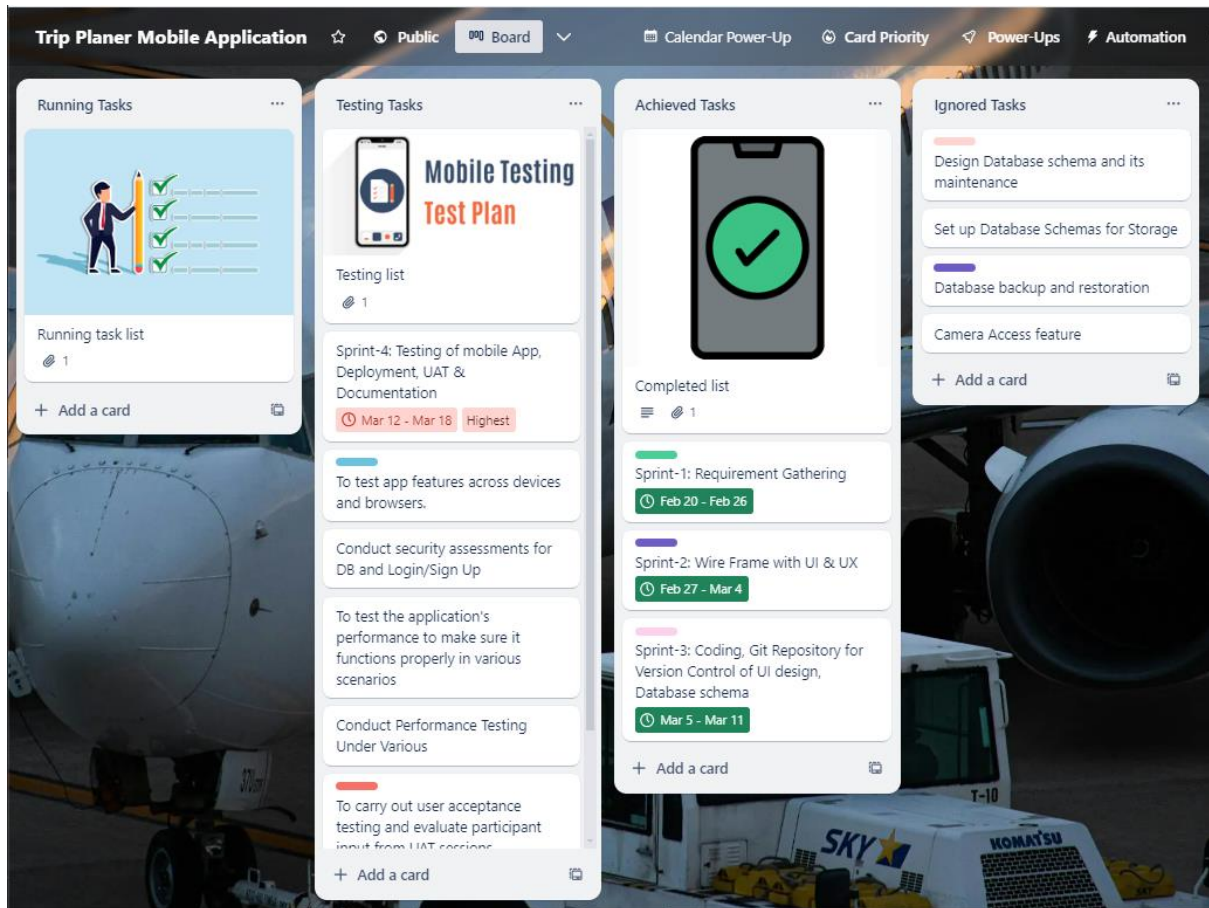


Figure 18. Phase III

Phase-4

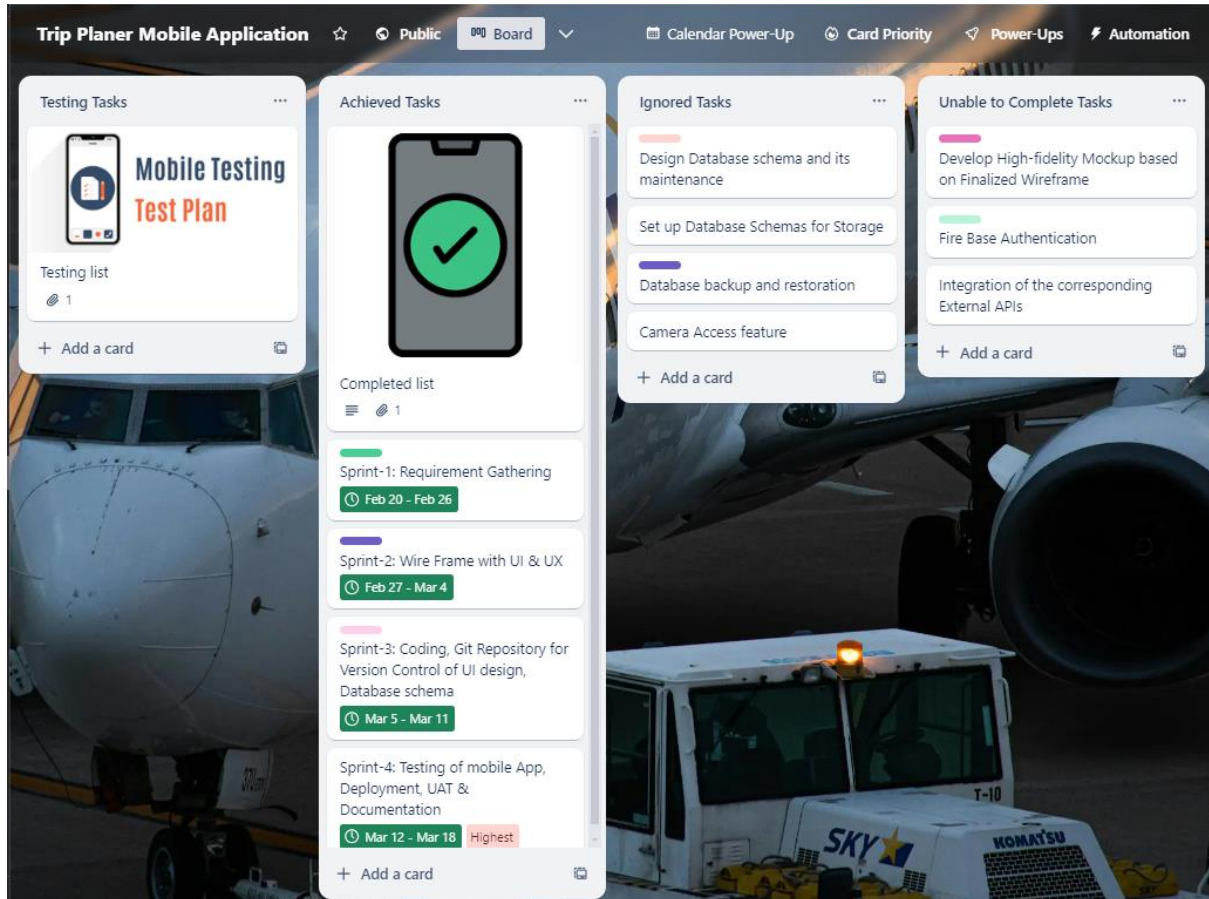


Figure 19. Phase IV

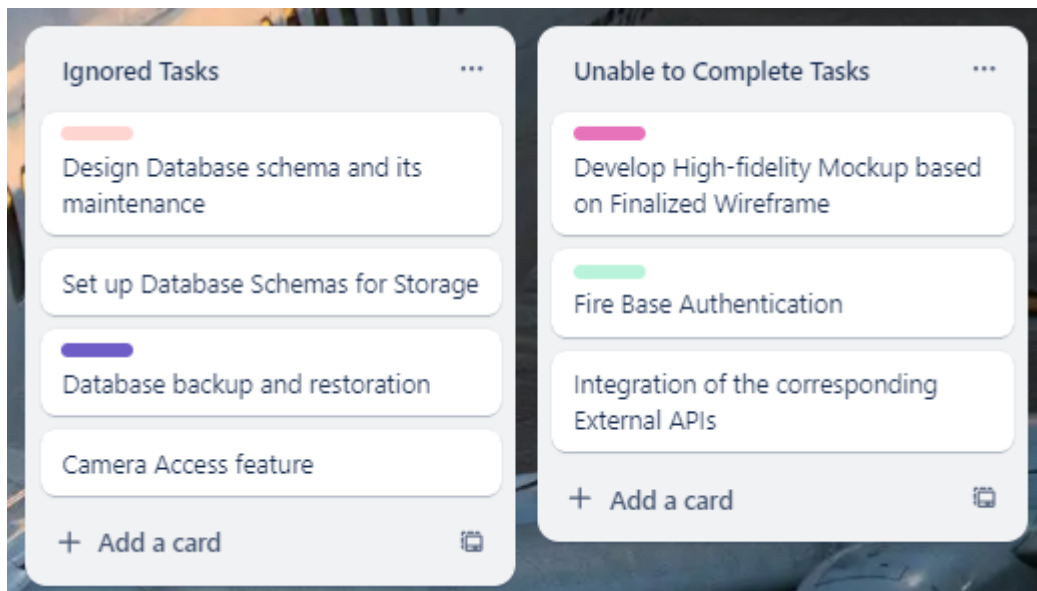


Figure 20. Phase IV (ignored tasks and unable to complete tasks)

Burn-down Chart

A burnt-down chart is a visual tool often used in project management, specially in agile methodologies like Scram. It traces the progress of work completed against the time or effort available in a project iteration, normally a sprint.

Burn-down for Sprint-1

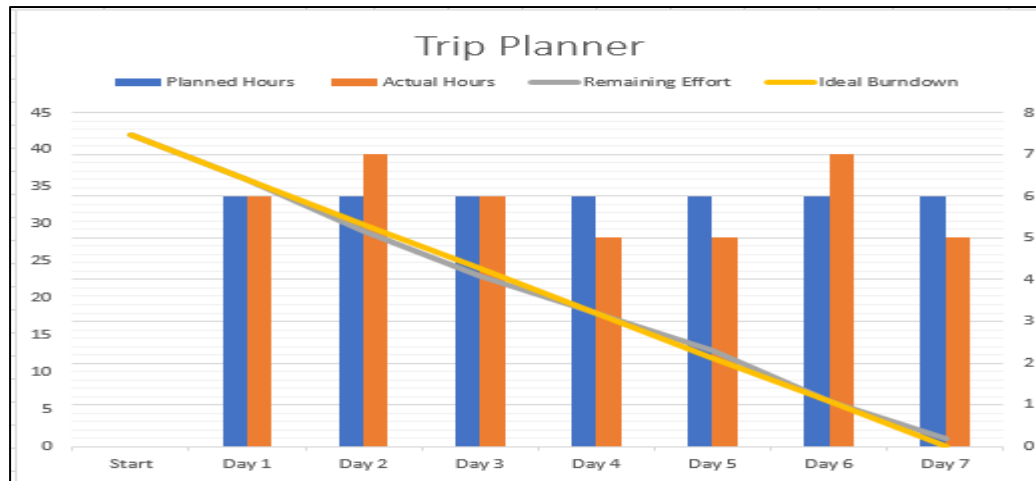


Figure 21. Burn Down Chart Sprint I

Burn-down for Sprint-2

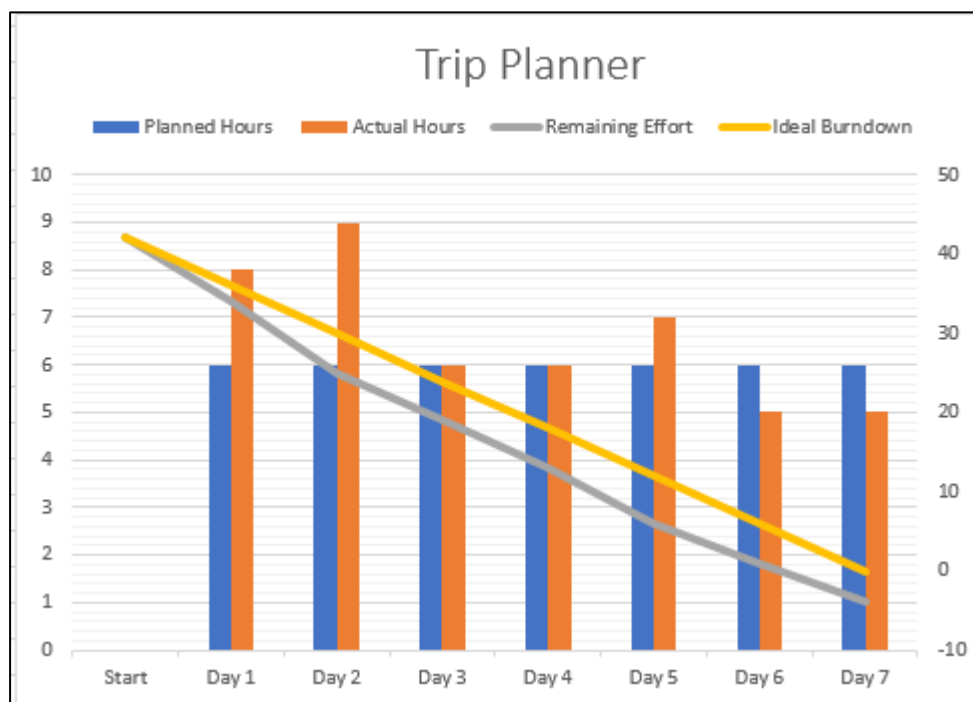


Figure 22. Burn Down Chart Sprint II

Burn-down for Sprint-3

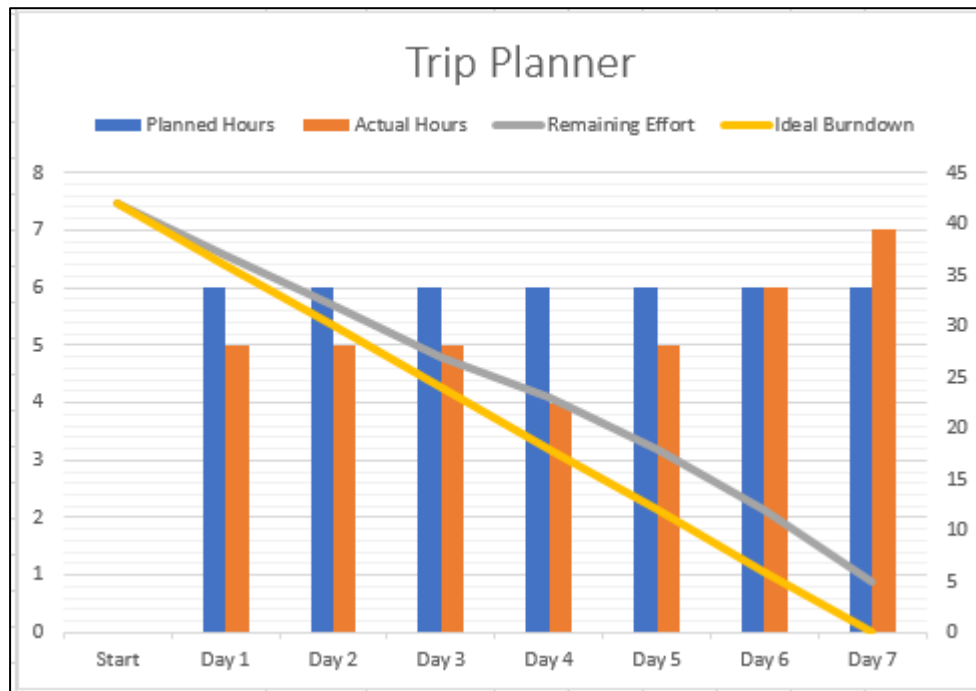


Figure 23. Burn Down Chart Sprint III

Burn-down for Sprint-4

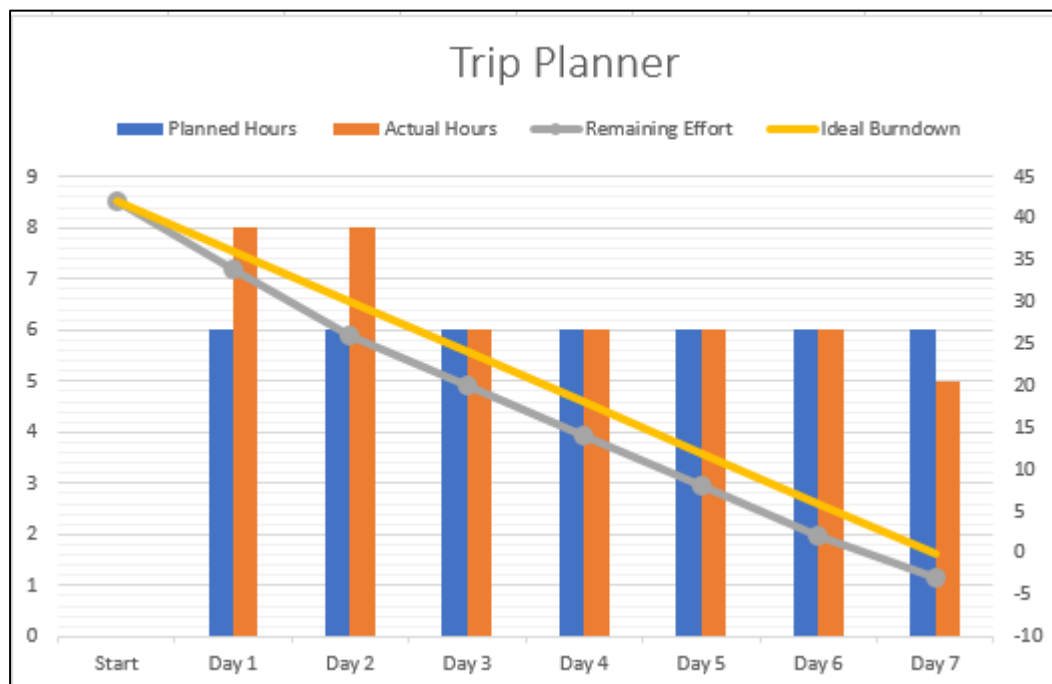


Figure 24. Burn Down Chart Sprint IV

Firebase DB at backend and regarding API authentication and searching

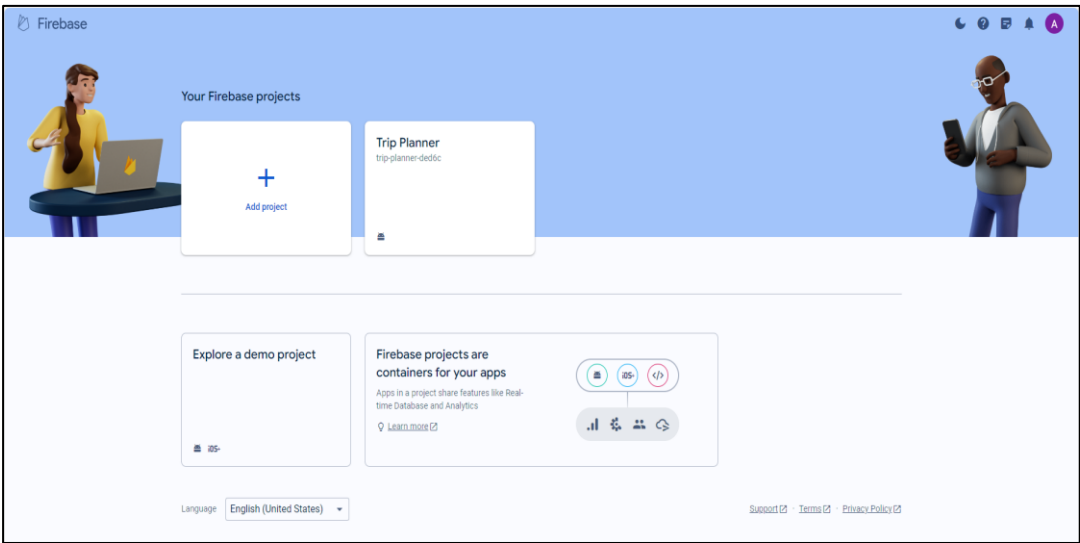


Figure 25. Firebase DB

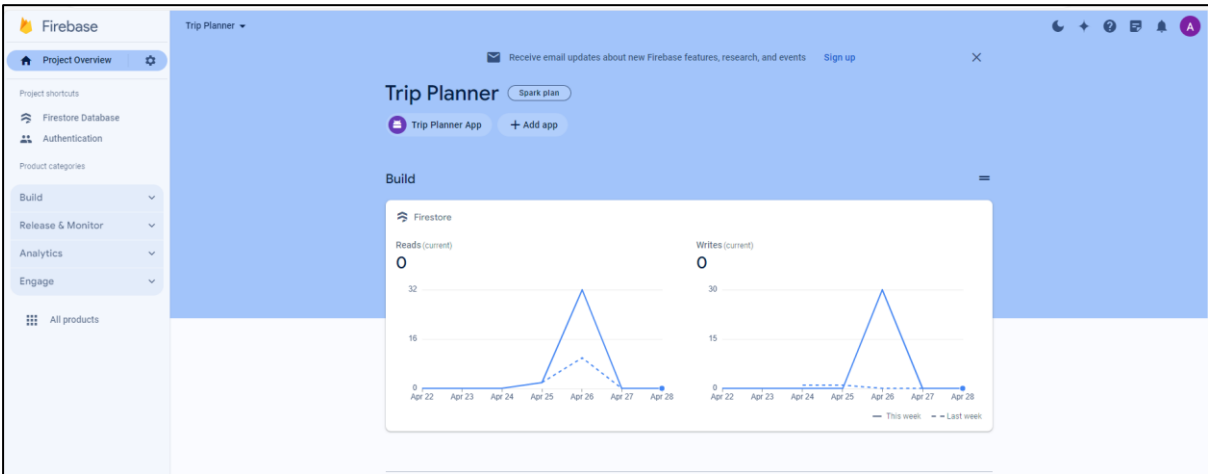


Figure 26. Firebase Voting app overview

 This screenshot shows the 'Authentication' page in the Firebase console. The left sidebar has 'Authentication' selected. The main area has tabs for 'Users', 'Sign-in method', 'Templates', 'Usage', 'Settings', and 'Extensions'. The 'Users' tab is active, displaying a table of users. At the top of the table is a search bar and an 'Add user' button. The table lists three users with their email addresses, providers, creation dates, sign-in dates, and user UIDs.

Identifier	Providers	Created ↓	Signed In	User UID
sushil@gmail.com	📧	Apr 29, 2024	Apr 29, 2024	OoiUW8spTEeC365c89PeMJ4...
anshu@gmail.com	📧	Apr 28, 2024	Apr 28, 2024	Yi9LrLITMPOZFhRjTf6tiOevC...
anuj@gmail.com	📧	Apr 27, 2024	Apr 27, 2024	wm4L7J4qPfdmfg2i4wVP9...

Figure 27. Firebase authentication

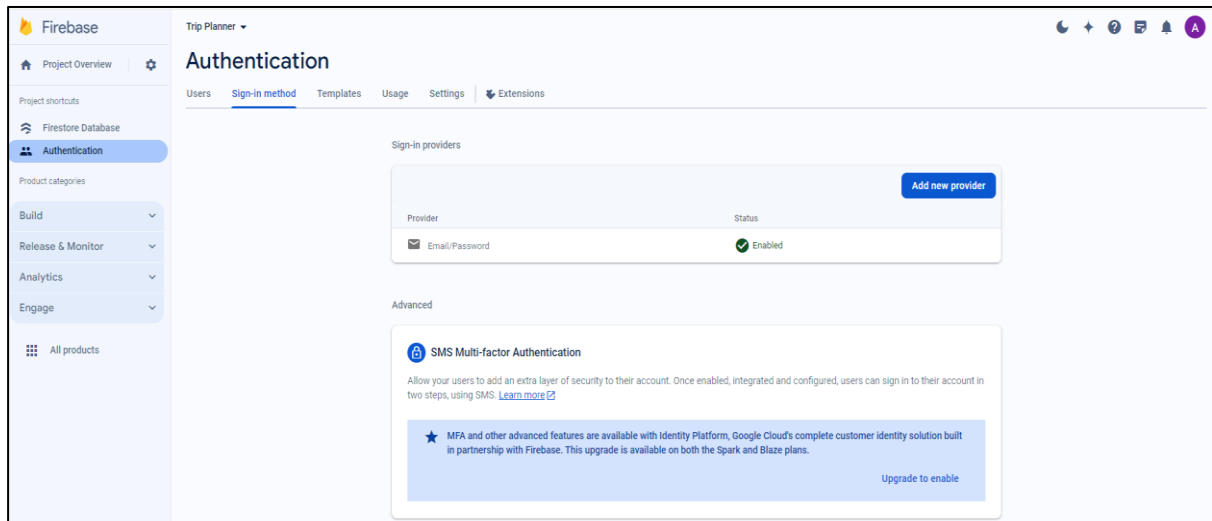


Figure 28. Sign in methods in firebase

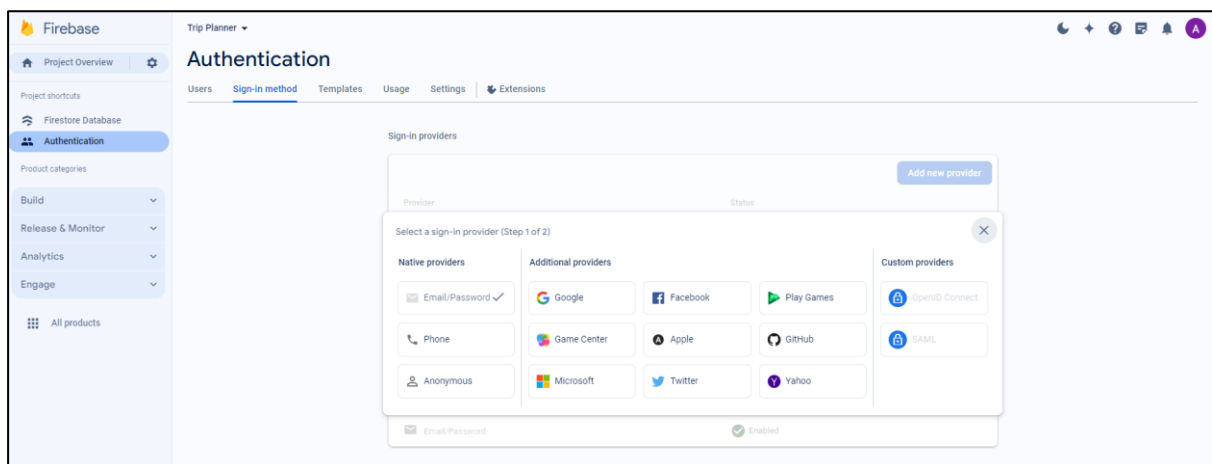


Figure 29. Gmail log in method in firebase

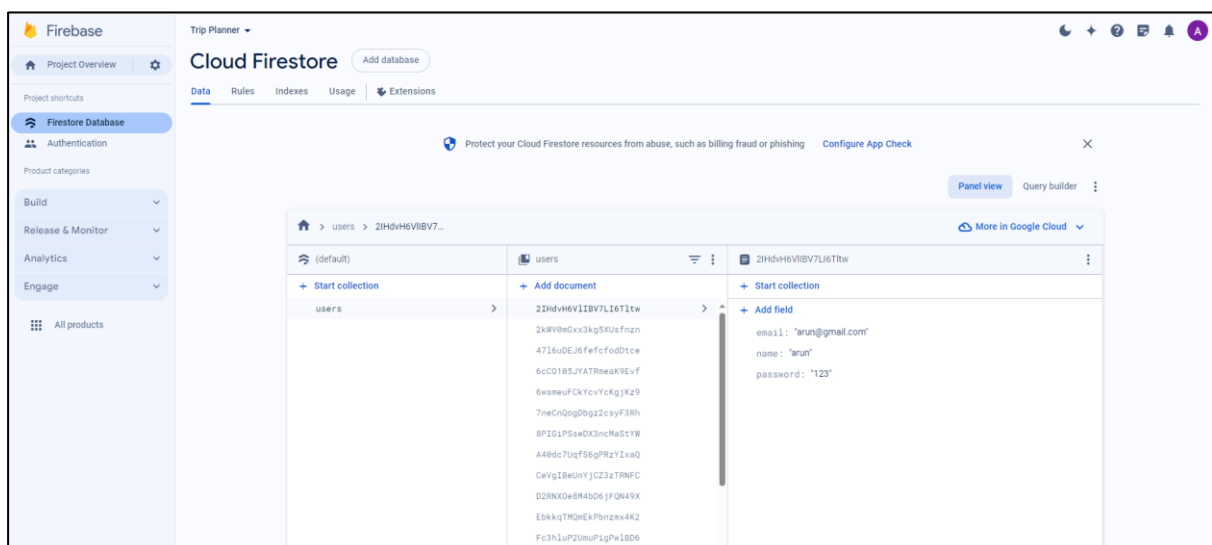


Figure 30. Cloud firestore

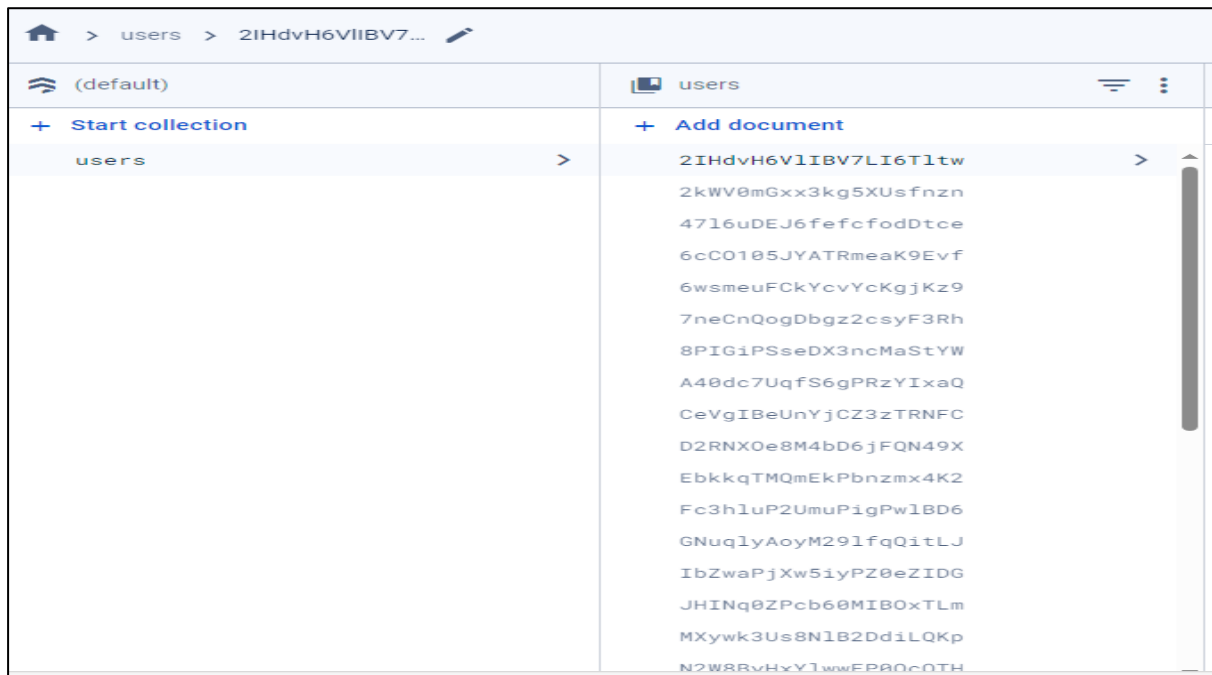


Figure 31. Collection list in firebase

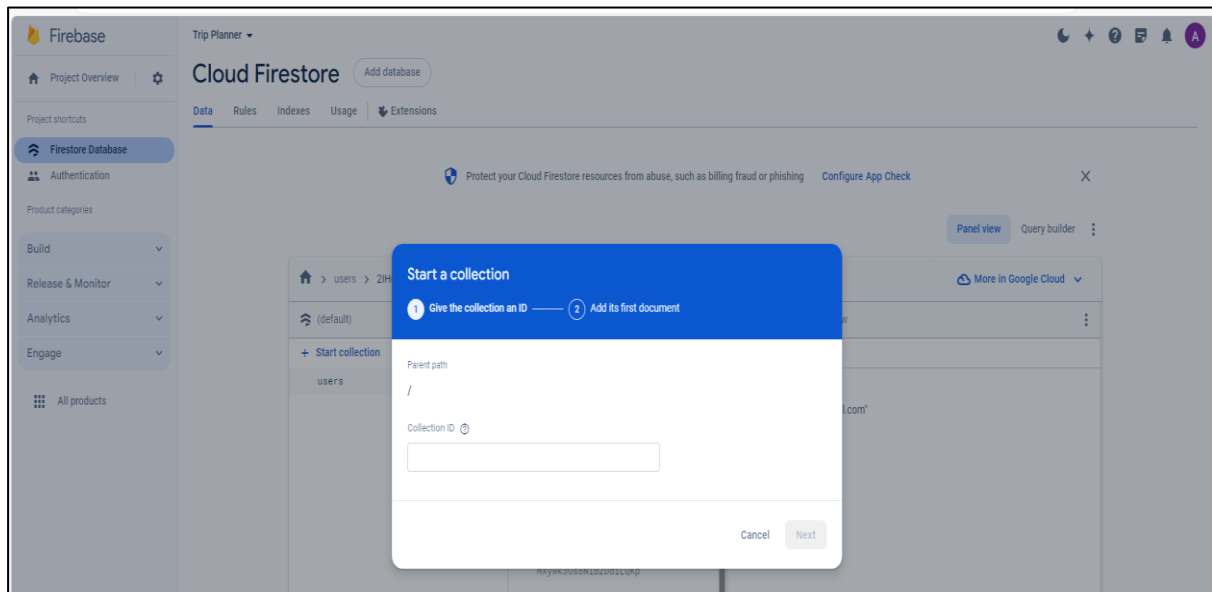


Figure 32. Collection starting in DB

Trip Planner Application (UI)

1. Welcome Page –



Figure 33. Welcome Page

2. Registration Page –

A screenshot of the registration page of the Trip Planner App. The page has a green background. At the top, there is a header with the text "Trip Planner App". Below the header, there are three input fields for "Name", "Email", and "Password". Each field has a placeholder text "Enter name", "Enter email", and "Enter Password" respectively. Below the input fields, there is a "Register" button. At the bottom, there is a link that says "Already have an account? Login". The page is framed by a green border.

Figure 34. Registration Page

3. Login Page –

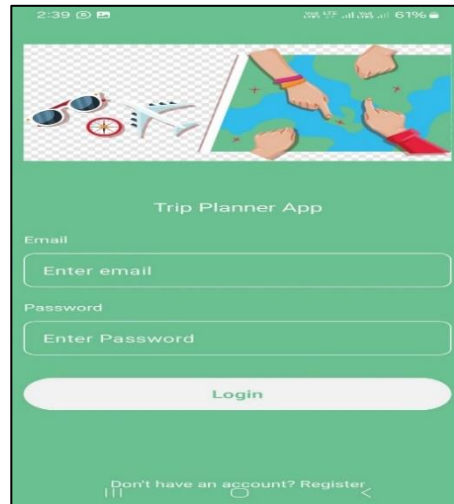


Figure 35. Login Page

4. Home Page –

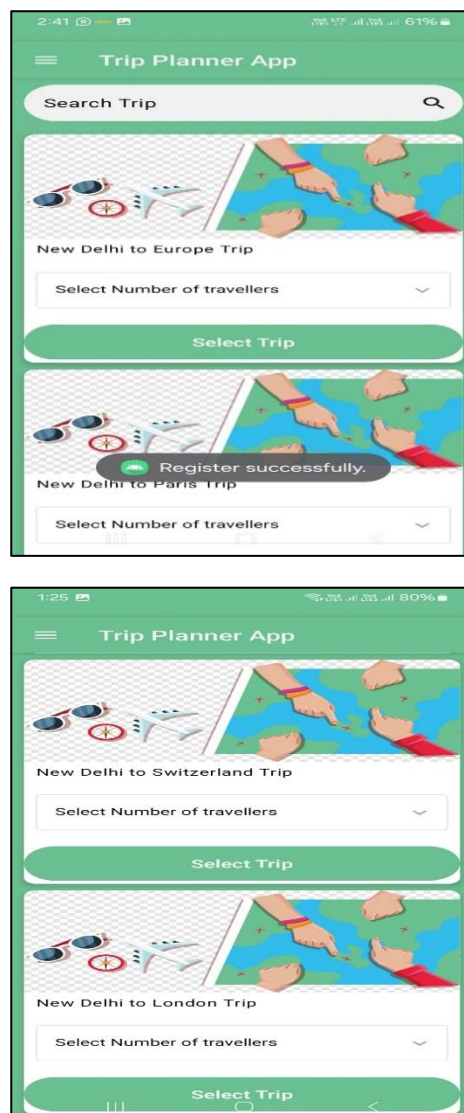


Figure 36. Home Page I

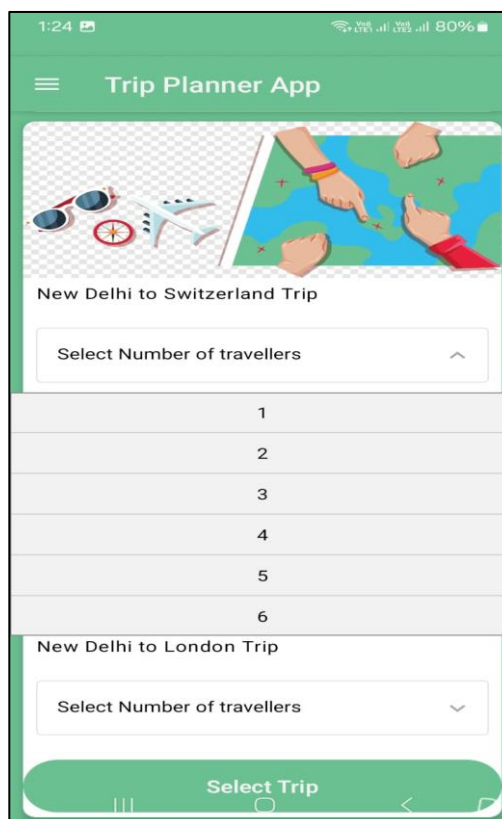
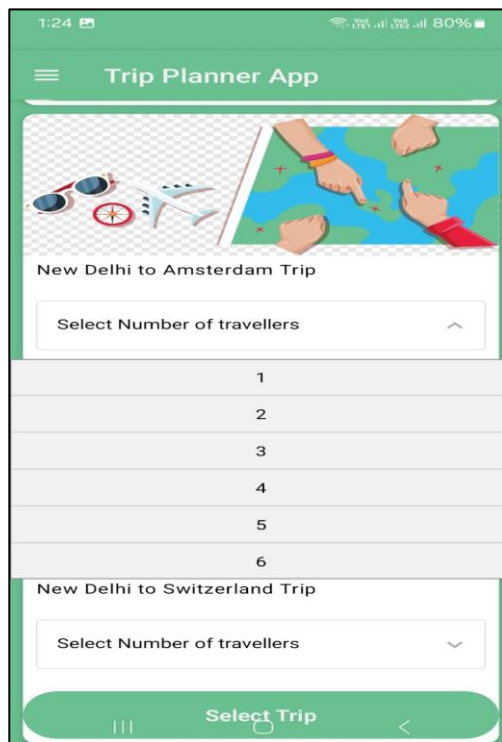


Figure 37. Home Page (with dropdown I)

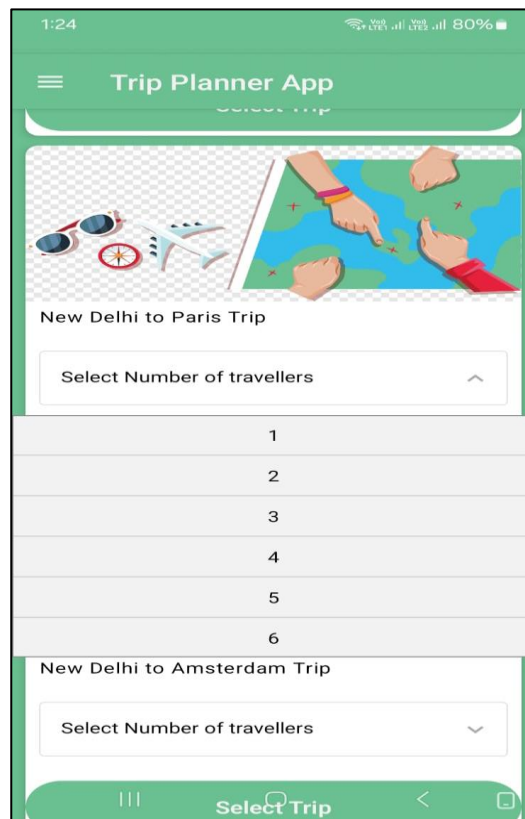


Figure 38. Home Page (with dropdown II)

5. Detail Page –



Figure 39. Details Page I



Figure 40. Details Page II

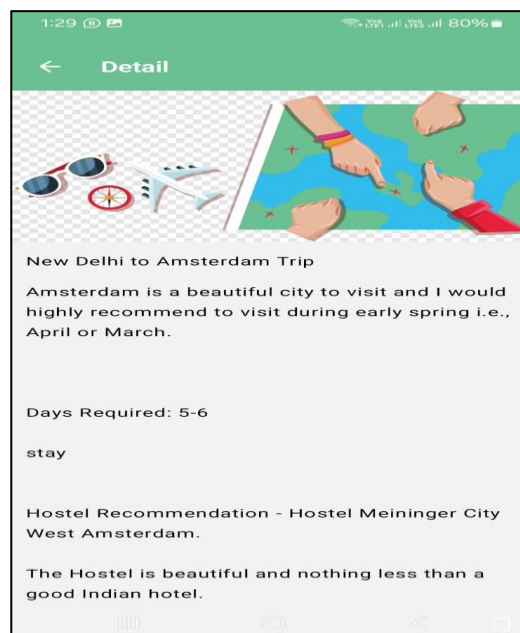


Figure 41. Details Page III

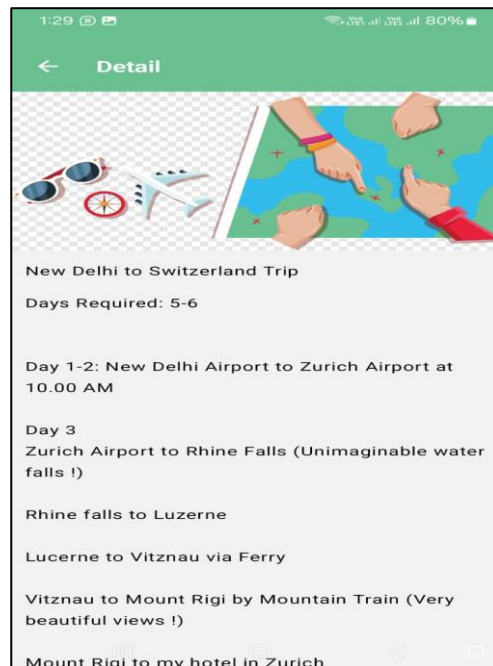


Figure 42. Details Page IV



Figure 43. Details Page V

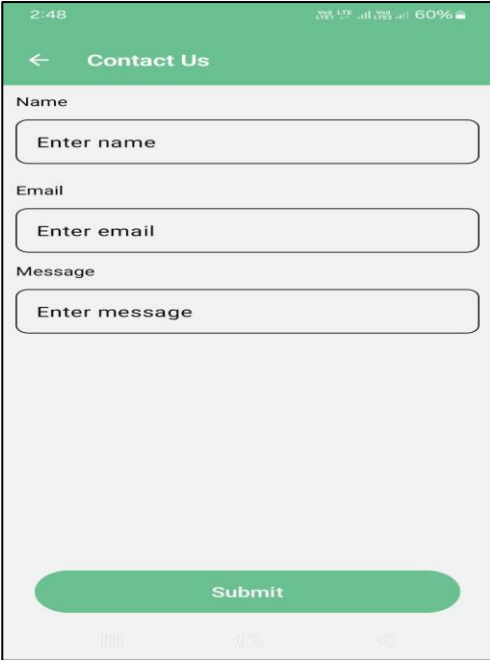
6. Transportation Booking Page –



A mobile app interface for a transportation booking page. The page has a green header with a back arrow and the title "Transportation Booking". Below the header, there are four input fields: "Name" with placeholder "Enter name", "Email" with placeholder "Enter email", "Mobile Number" with placeholder "Enter mobile", and "Address" with placeholder "Enter address". At the bottom, there is a green "Submit" button. The bottom navigation bar shows three icons: a home icon, a camera icon, and a back arrow.

Figure 44. Transportation Booking Page

7. Contact Us Page –



A mobile app interface for a contact us page. The page has a green header with a back arrow and the title "Contact Us". Below the header, there are three input fields: "Name" with placeholder "Enter name", "Email" with placeholder "Enter email", and "Message" with placeholder "Enter message". At the bottom, there is a green "Submit" button. The bottom navigation bar shows three icons: a home icon, a camera icon, and a back arrow.

Figure 45. Contact Us Page

8. Log out Page –

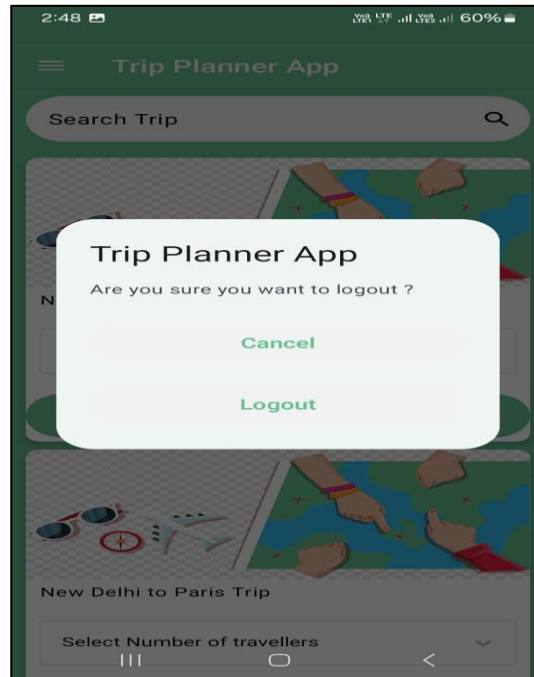


Figure 46. Log Out Page

GIT

1st commit

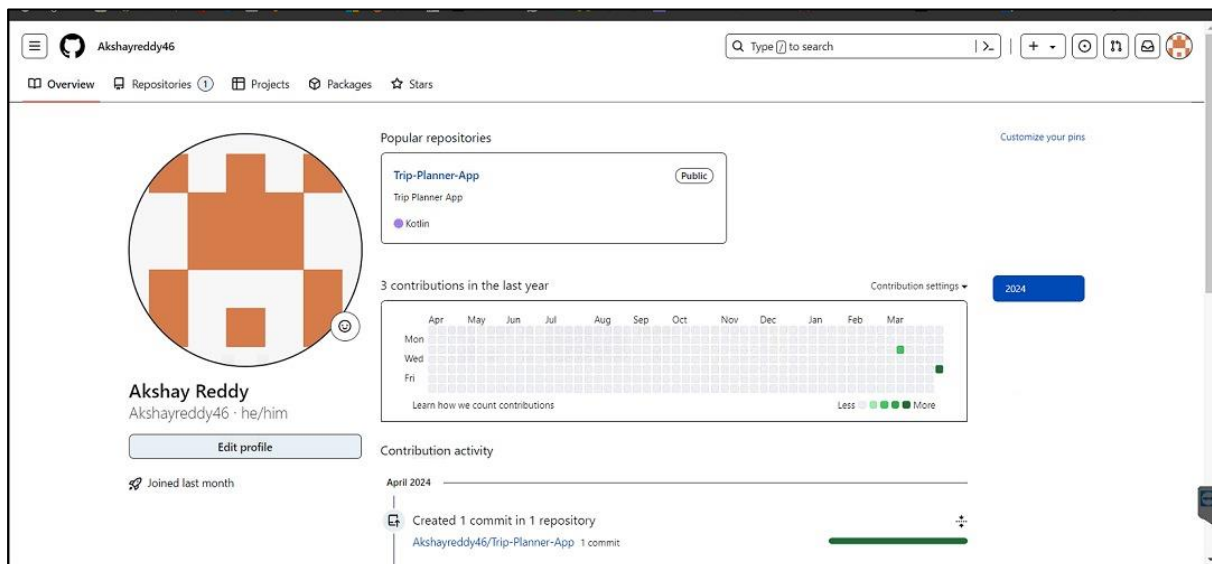


Figure 47. Ist Commit (overview)


```

Akshayreddycj@Akshayreddy MINGW64 /C:/Users/aksha/Downloads/Trip Planner App/Tri
p Planner App (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        app/
        build.gradle
        gradle.properties
        gradle/
        gradlew
        gradlew.bat
        settings.gradle

nothing added to commit but untracked files present (use "git add" to track)

Akshayreddycj@Akshayreddy MINGW64 /C:/Users/aksha/Downloads/Trip Planner App/Tri
p Planner App (master)
$ git add .
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the n
ext time Git touches it
warning: in the working copy of 'app/build.gradle', LF will be replaced by CRLF
the next time Git touches it
warning: in the working copy of 'app/google-services.json', LF will be replaced
by CRLF the next time Git touches it
warning: in the working copy of 'app/proguard-rules.pro', LF will be replaced b
y CRLF the next time Git touches it
warning: in the working copy of 'app/src/androidTest/java/com/tripplanner/Examp
leInstrumentedTest.kt', LF will be replaced by CRLF the next time Git touches i
t
warning: in the working copy of 'app/src/main/java/com/tripplanner/MainActivity

```

```

Akshayreddycj@Akshayreddy MINGW64 /C:/Users/aksha/Downloads/Trip Planner App/Tri
p Planner App (master)
$ git commit -m "uploading code"
[master (root-commit) 8888318] uploading code
58 files changed, 2742 insertions(+)
create mode 100644 .gitignore
create mode 100644 app/.gitignore
create mode 100644 app/build.gradle
create mode 100644 app/google-services.json
create mode 100644 app/proguard-rules.pro
create mode 100644 app/src/androidTest/java/com/tripplanner/ExampleInstrumente
dTest.kt
create mode 100644 app/src/main/AndroidManifest.xml
create mode 100644 app/src/main/java/com/tripplanner/MainActivity.kt
create mode 100644 app/src/main/java/com/tripplanner/routing/Navigation.kt
create mode 100644 app/src/main/java/com/tripplanner/routing/screen.kt
create mode 100644 app/src/main/java/com/tripplanner/ui/bookScreen/BookScreen.
kt
create mode 100644 app/src/main/java/com/tripplanner/ui/contact_us/ContactUsSc
reen.kt
create mode 100644 app/src/main/java/com/tripplanner/ui/detail/DetailScreen.kt
create mode 100644 app/src/main/java/com/tripplanner/ui/login/LoginScreen.kt
create mode 100644 app/src/main/java/com/tripplanner/ui/main/MainScreen.kt
create mode 100644 app/src/main/java/com/tripplanner/ui/model/TripModel.kt
create mode 100644 app/src/main/java/com/tripplanner/ui/profileScreen/Profiles
creen.kt
create mode 100644 app/src/main/java/com/tripplanner/ui/register/Registerscre
en.kt
create mode 100644 app/src/main/java/com/tripplanner/ui/searchListing/SearchSc
reen.kt
create mode 100644 app/src/main/java/com/tripplanner/ui/splash/SplashScreen.kt
create mode 100644 app/src/main/java/com/tripplanner/ui/theme/color.kt
create mode 100644 app/src/main/java/com/tripplanner/ui/theme/Theme.kt

```

Figure 48. Ist Commit (Master)

```

Akshayreddycj@Akshayreddy MINGW64 /C/Users/aksha/Downloads/Trip Planner App/Trip Planner App (master)
$ git status
On branch master
nothing to commit, working tree clean

Akshayreddycj@Akshayreddy MINGW64 /C/Users/aksha/Downloads/Trip Planner App/Trip Planner App (master)
$ git remote add origin https://github.com/Akshayreddy46/Trip-Planner-App.git

Akshayreddycj@Akshayreddy MINGW64 /C/Users/aksha/Downloads/Trip Planner App/Trip Planner App (master)
$ git push origin master
Enumerating objects: 100, done.
Counting objects: 100% (100/100), done.
Delta compression using up to 8 threads
Compressing objects: 100% (78/78), done.
Writing objects: 100% (100/100), 151.86 KiB | 2.62 MiB/s, done.
Total 100 (delta 9), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (9/9), done.
To https://github.com/Akshayreddy46/Trip-Planner-App.git
 * [new branch]      master -> master

Akshayreddycj@Akshayreddy MINGW64 /C/Users/aksha/Downloads/Trip Planner App/Trip Planner App (master)
$

```

Figure 49. Ist Commit (Master II)

```

Akshayreddycj@Akshayreddy MINGW64 /C/Users/aksha/Downloads/Trip Planner App/Trip Planner App
$ git config --global user.name "Akshayreddy46"

Akshayreddycj@Akshayreddy MINGW64 /C/Users/aksha/Downloads/Trip Planner App/Trip Planner App
$ git config --global user.email "akshayreddybadwali@gmail.com"

Akshayreddycj@Akshayreddy MINGW64 /C/Users/aksha/Downloads/Trip Planner App/Trip Planner App
$ git status
fatal: not a git repository (or any of the parent directories): .git

Akshayreddycj@Akshayreddy MINGW64 /C/Users/aksha/Downloads/Trip Planner App/Trip Planner App
$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in C:/Users/aksha/Downloads/Trip Planner App/Trip Planner App/.git/

Akshayreddycj@Akshayreddy MINGW64 /C/Users/aksha/Downloads/Trip Planner App/Trip Planner App (master)
$

```

Figure 50. Ist Commit (Master III)

2nd commit

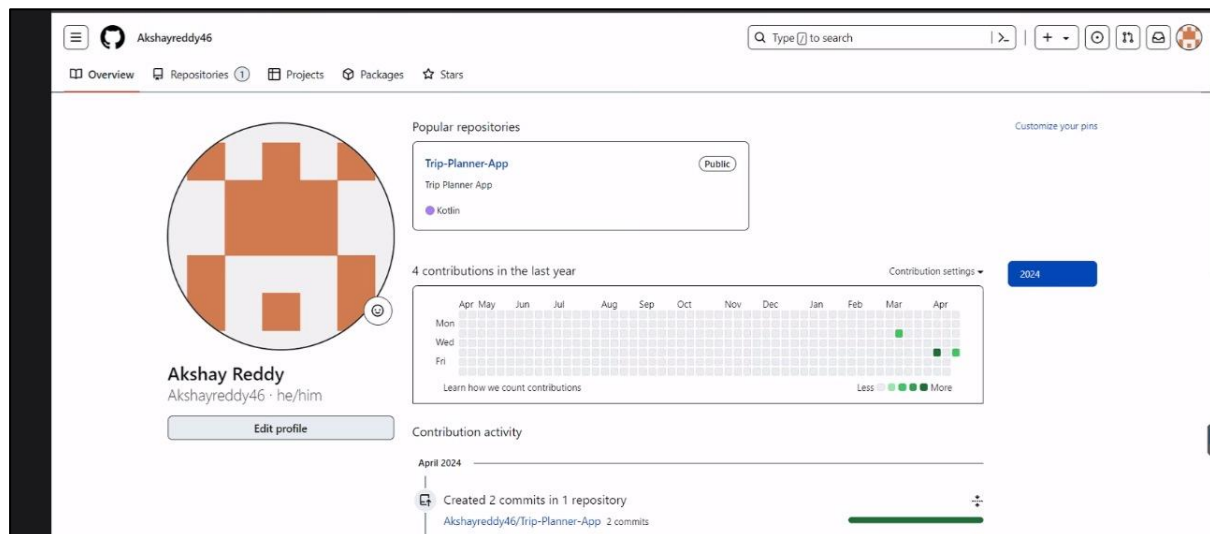


Figure 51. IInd Commit (Overview)

```
Akshayreddycj@Akshayreddy MINGW64 ~/Downloads/2nd Commkit/clone/Trip-Planner-App (master)
$ git push
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (11/11), 841 bytes | 841.00 KiB/s, done.
Total 11 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
To https://github.com/Akshayreddy46/Trip-Planner-App.git
  8888318..0217c4d master -> master

Akshayreddycj@Akshayreddy MINGW64 ~/Downloads/2nd Commkit/clone/Trip-Planner-App (master)
$ git commit -m "second commit"
[master 0217c4d] second commit
1 file changed, 1 insertion(+), 1 deletion(-)

Akshayreddycj@Akshayreddy MINGW64 ~/Downloads/2nd Commkit/clone/Trip-Planner-App (master)
$ git add .
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/build.gradle', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/google-services.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/proguard-rules.pro', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/androidTest/java/com/tripplanner/ExampleInstrumentedTest.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/tripplanner/MainActivity.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/tripplanner/ui/theme/color.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/tripplanner/ui/theme/Theme.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/tripplanner/ui/theme/Type.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/tripplanner/utlis/RoundedButton.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/drawable-v24/ic_launcher_foreground.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/drawable/baseline_person_24.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/drawable/ic_launcher_background.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/mipmap-anydpi-v26/ic_launcher.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/mipmap-anydpi-v26/ic_launcher_round.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/mipmap-anydpi-v33/ic_launcher.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/values/colors.xml', LF will be replaced by CRLF the next time Git touches it
```

Figure 52. IInd Commit (Master)

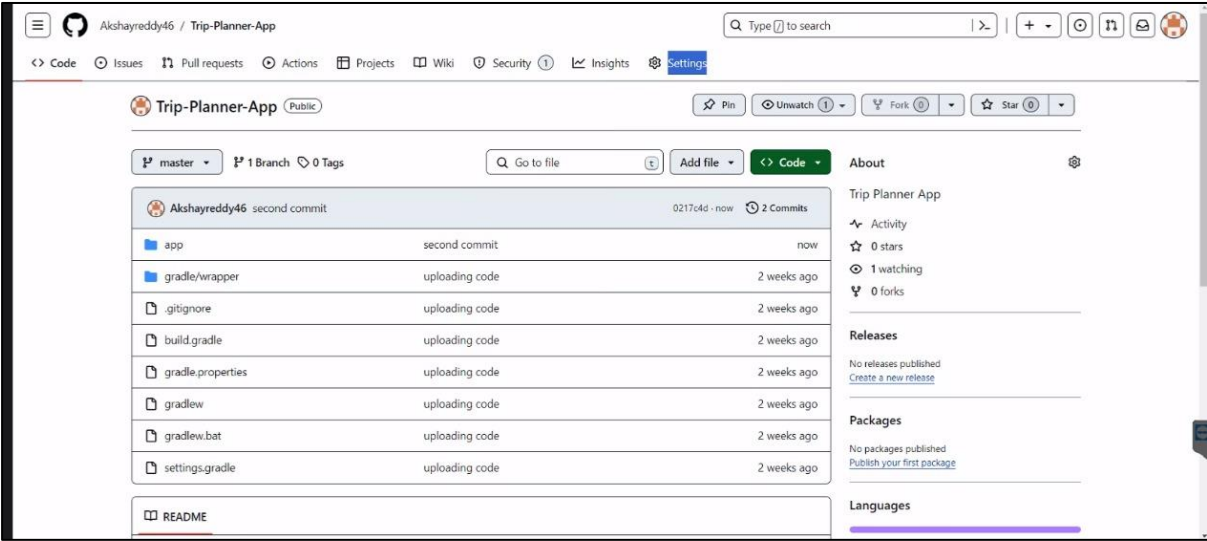


Figure 53. IInd Commit (Code)

3rd commit

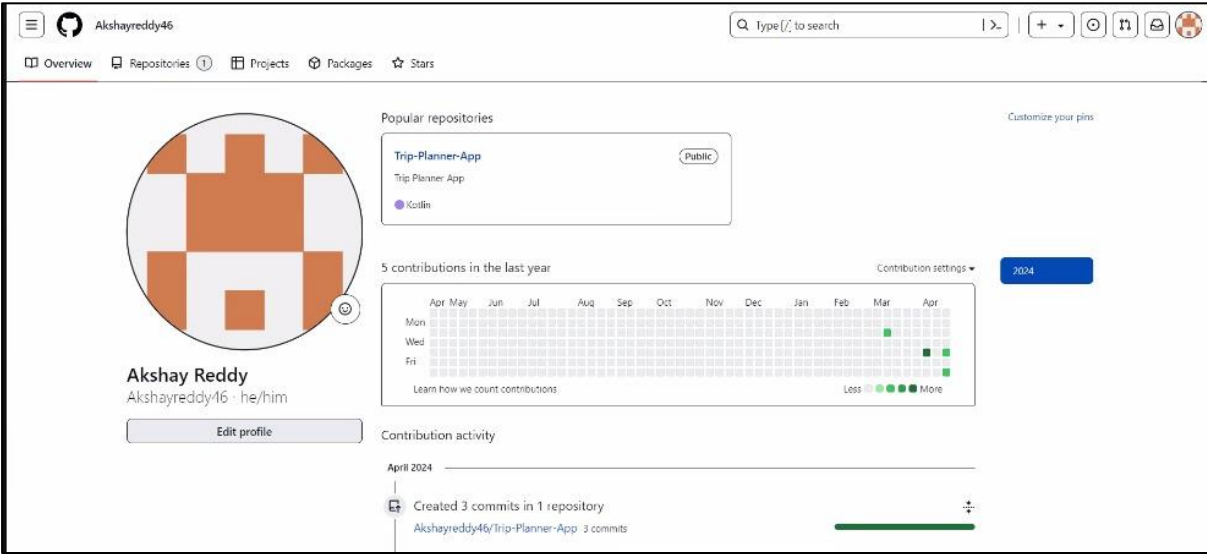


Figure 54. IIIrd Commit (Overview)

```
Akshayreddycj@Akshayreddy MINGW64 ~/Downloads/3rd Commit/Trip Planner App/clone/Trip-Planner-App (master)
$ git push
Enumerating objects: 82, done.
Counting objects: 100% (82/82), done.
Delta compression using up to 8 threads
Compressing objects: 100% (32/32), done.
Writing objects: 100% (45/45), 13.25 KiB | 714.00 KiB/s, done.
Total 45 (delta 18), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (18/18), completed with 16 local objects.
To https://github.com/Akshayreddy46/Trip-Planner-App.git
0217c4d..eb52496 master -> master

Akshayreddycj@Akshayreddy MINGW64 ~/Downloads/3rd Commit/Trip Planner App/clone/Trip-Planner-App (master)
$ |

Akshayreddycj@Akshayreddy MINGW64 ~/Downloads/3rd Commit/Trip Planner App/clone/Trip-Planner-App (master)
$ git commit -m "third commit"
[master eb52496] third commit
21 files changed, 1040 insertions(+), 361 deletions(-)
create mode 100644 app/src/main/java/com/tripplanner/ui/TripPreference/TripPreference.kt
create mode 100644 app/src/main/java/com/tripplanner/ui/drawer/NavDrawer.kt
create mode 100644 app/src/main/java/com/tripplanner/ui/drawer/TopBar.kt
create mode 100644 app/src/main/java/com/tripplanner/utils/Extension.kt
create mode 100644 app/src/main/java/com/tripplanner/utils/TripField.kt

Akshayreddycj@Akshayreddy MINGW64 ~/Downloads/3rd Commit/Trip Planner App/clone/Trip-Planner-App (master)
$ git push

Akshayreddycj@Akshayreddy MINGW64 ~/Downloads/3rd Commit/Trip Planner App/clone/Trip-Planner-App (master)
$ git add
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/build.gradle', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/google-services.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/proguard-rules.pro', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/androidTest/java/com/tripplanner/ExampleInstrumentedTest.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/tripplanner/MainActivity.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/tripplanner/ui/theme/Color.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/tripplanner/ui/theme/Theme.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/tripplanner/utils/RoundedButton.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/drawable-v24/ic_launcher_foreground.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/drawable/baseline_person_24.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/drawable/ic_launcher_background.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/mipmap-anydpi-v26/ic_launcher.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/mipmap-anydpi-v26/ic_launcher_round.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/mipmap-anydpi-v33/ic_launcher.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/values/colors.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/values/strings.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/values/themes.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/xml/backup_rules.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/xml/data_extraction_rules.xml', LF will be replaced by CRLF the next time Git touches it
```

Figure 55. IIIrd Commit (Master)

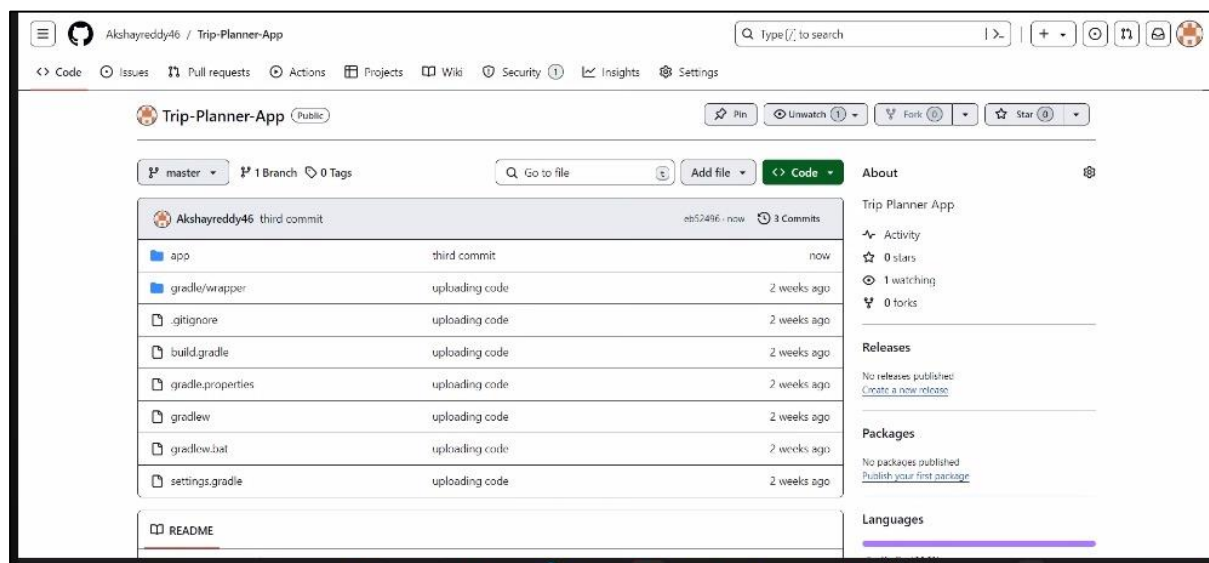


Figure 56. IIIrd Commit (Code)

Reference

- Noteboom, C., Ofori, M., Sutrave, K. and El-Gayar, O., 2021. Agile project management: A systematic literature review of adoption drivers and critical success factors.
- Haidabrus, B., Grabis, J. and Protsenko, S., 2021. Agile project management based on data analysis for information management systems. *Design, simulation, manufacturing: the innovation exchange*, pp.174-182.
- Thomas, C.G. and Devi, J., 2021. A study and overview of the mobile app development industry. *International Journal of Applied Engineering and Management Letters (IJAEML)*, 5(1), pp.115-130.
- Shah, K., Sinha, H. and Mishra, P., 2019, March. Analysis of cross-platform mobile app development tools. In *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)* (pp. 1-7). IEEE.
- Nayebi, M., Cho, H. and Ruhe, G., 2018. App store mining is not enough for app improvement. *Empirical Software Engineering*, 23, pp.2764-2794.
- Shchetynina, O., Kravchenko, N., Horbatiuk, L., Aliksieieva, H. and Mezhuyev, V., 2022. Trello as a Tool for the Development of Lifelong Learning Skills of Senior Students. *Postmodern Openings*, 13(2), pp.143-167.
- Flanagan, D., 2005. *Java in a Nutshell*. " O'Reilly Media, Inc."
- Eckel, B., 2003. *Thinking in JAVA*. Prentice Hall Professional.
- DiMarzio, J., 2016. *Beginning android programming with android studio*. John Wiley & Sons.
- Yener, M. and Dundar, O., 2016. *Expert Android Studio*. John Wiley & Sons.