Project Report

1. Introduction

1.1 Overview

A brief description about your project

1.2 Purpose

The use of this project. What can be achieved using this

2 Problem definition & design thinking

2.1 Empathy map

Paste the empathy map screenshot

2.2 Ideation & brainstorming map

Paste the ideation & brainstorming map screenshot

3 Result

Final findings (output) of the project along with screenshot

4 Advantages & Disadvantages

List of advantages and disadvantages of the proposed solution

5 Applications

The areas where this solution can be applied

6 Conclusion

Conclusion summarising the entire work and findings

7 Future Scope

Enhancements that can be made in the future

8 Appendix

A. Source code

Attach the code for the solution built

TRAVEL PLANNING APP

INTRODUCTION

1.1 Overview

A travel planning app is an application for planning travel reservation, tracking loyalty points and browsing travel packages. A travel planning software package can be used by travel agencies, travel suppliers, and consumers. Some travel planning applications are specifically designed for use by travel agencies, while others are designed for individual consumers. And, the ravel planning app is always used to search hundreds of travel sites instantly in an attempt to offer every possible deal on a restaurants or reservation of hotels available on the same location, and it will also help you to organize everything in one place.

1.2 Purpose

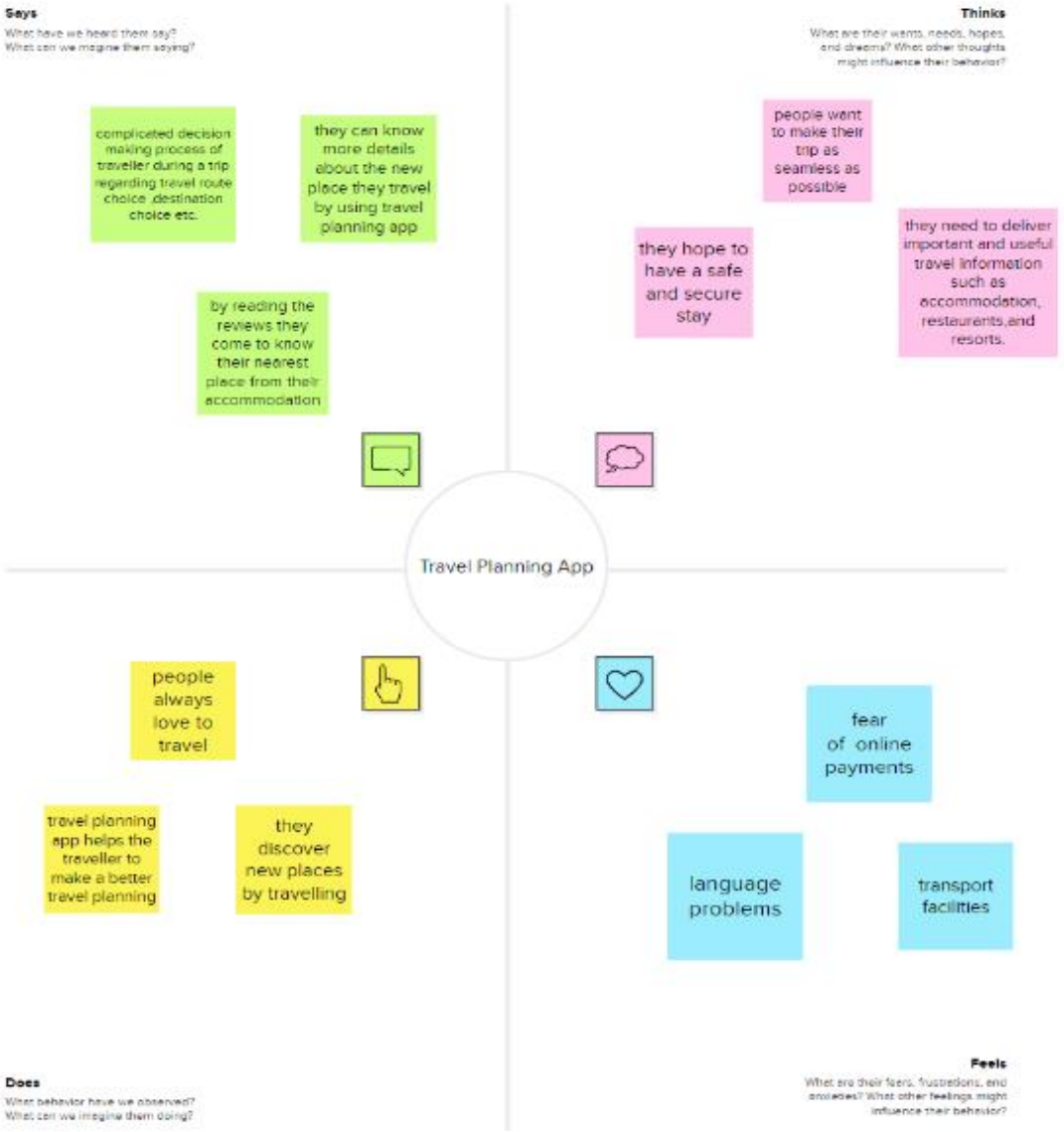
A mobile travel app is a software application created specifically for use on smaller devices such as smartphones and tablets that help business travellers to plan and manage their travel while on the go. Planning travel and reviewing itineraries are the key features available in most mobile travel applications.

A travel app helps them to organize their trip smoothly, swiftly, and cost efficiently, from finding a hotel. Travel apps now include all sophisticated features to provide customers with fast and accurate service at their fingers touch.

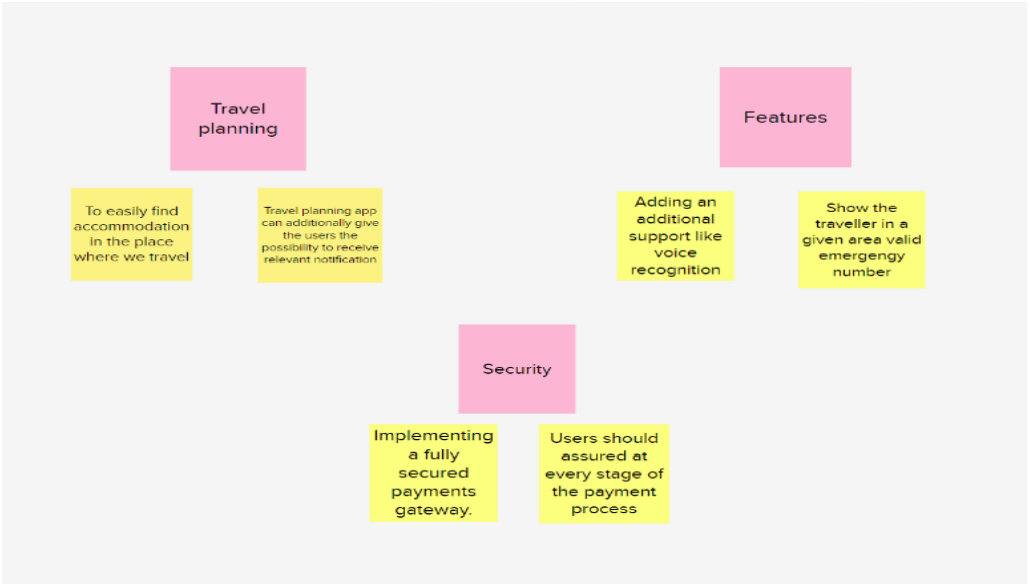
With specialized travel app, we can provide the customers with real time tracking experience. Now a days, mobile solution for travel companies Offer numerous benefits that improve and a ease travel arrangements.

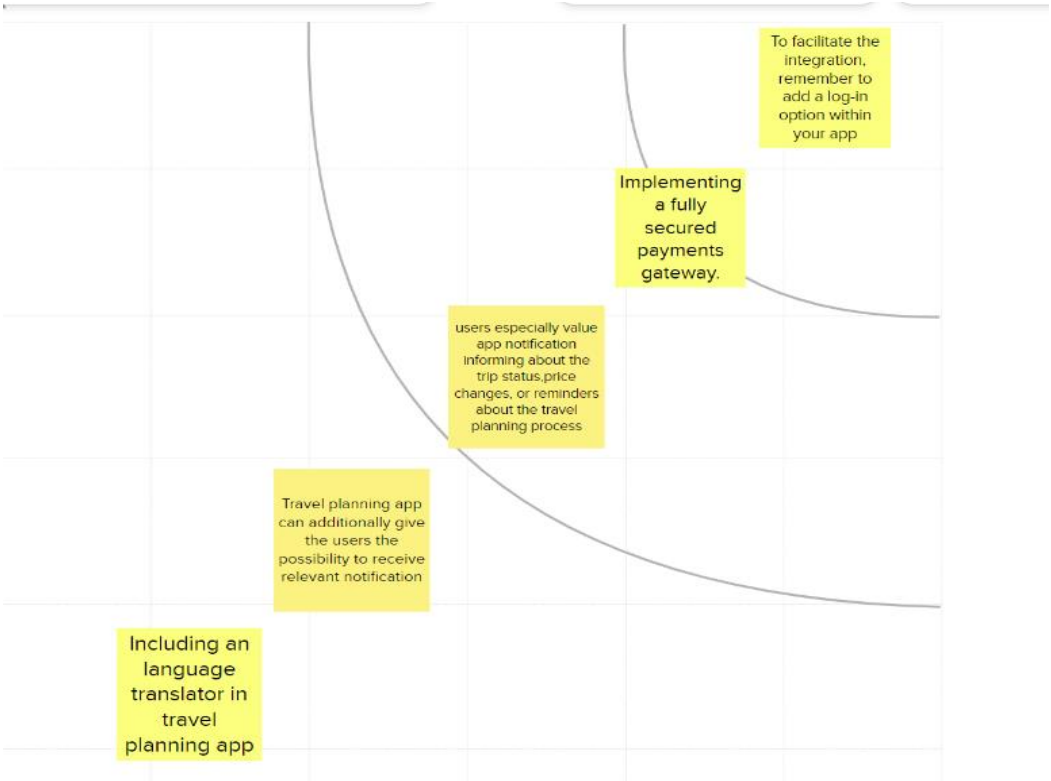
2 Problem Definition & Design Thinking

Empathy map

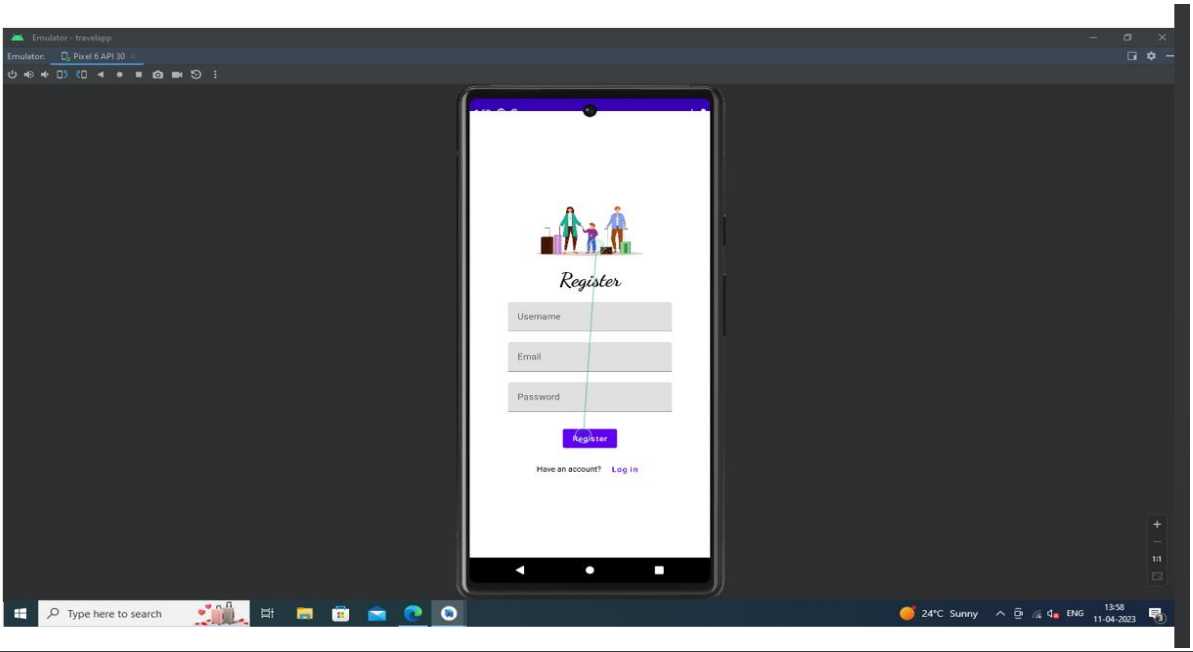


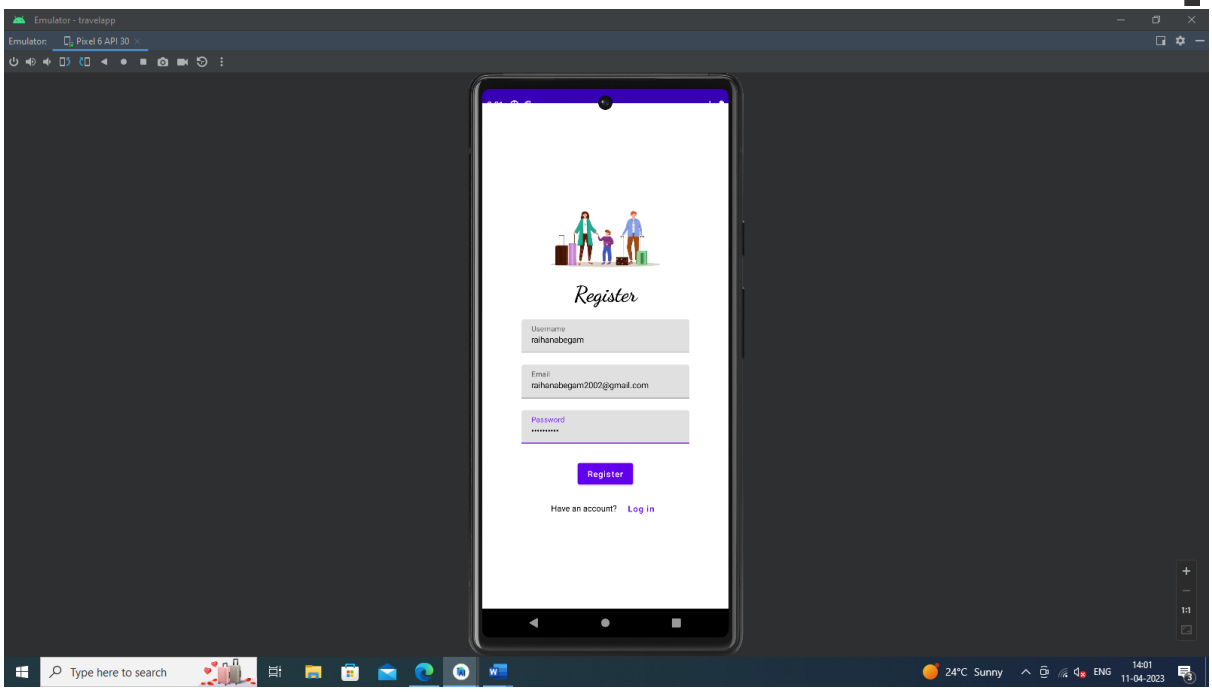
Brainstorming

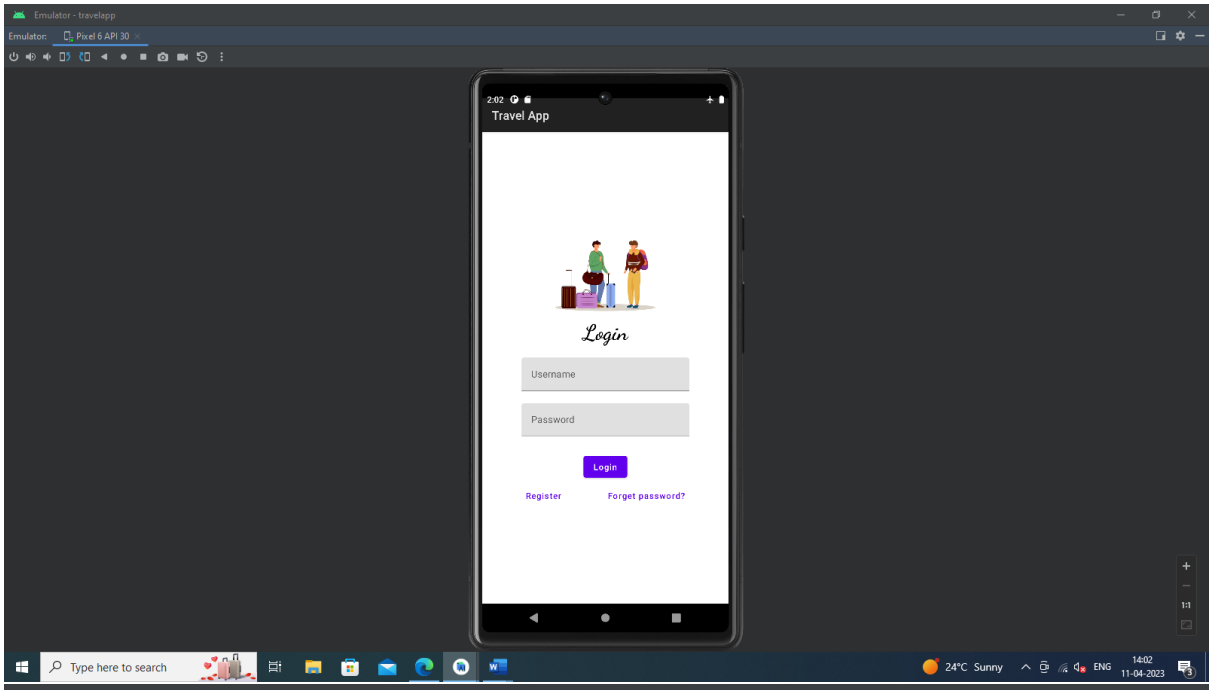

Travel planning
To easily find accommodation in the place where we travel
Travel planning app can additionally give the users the possibility to receive relevant notification
Features
Adding an additional support like voice recognition
Show the traveller in a given area valid emergengy number
Security
Implementing a fully secured payments gateway.
Users should assured at every stage of the payment process

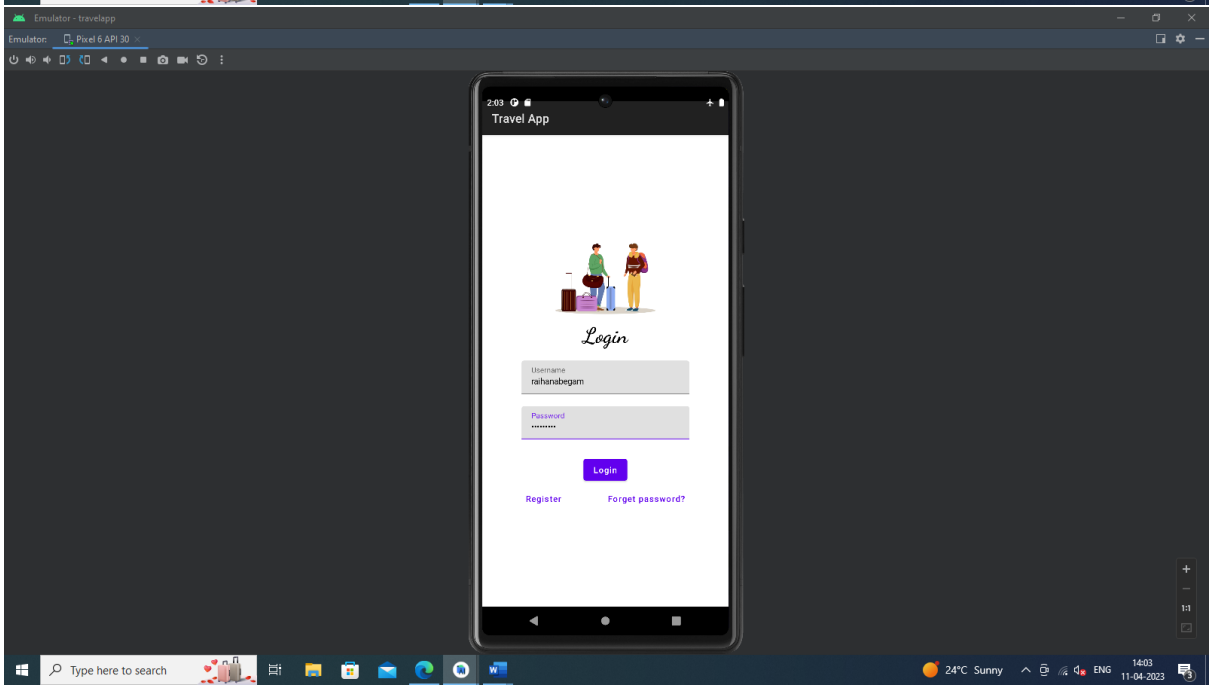

To facilitate the integration, remember to add a log-in option within your app
Implementing a fully secured payments gateway.
users especially value app notification informing about the trip status,price changes, or reminders about the travel planning process
Travel planning app can additionally give the users the possibility to receive relevant notification
Including an language translator in travel planning app

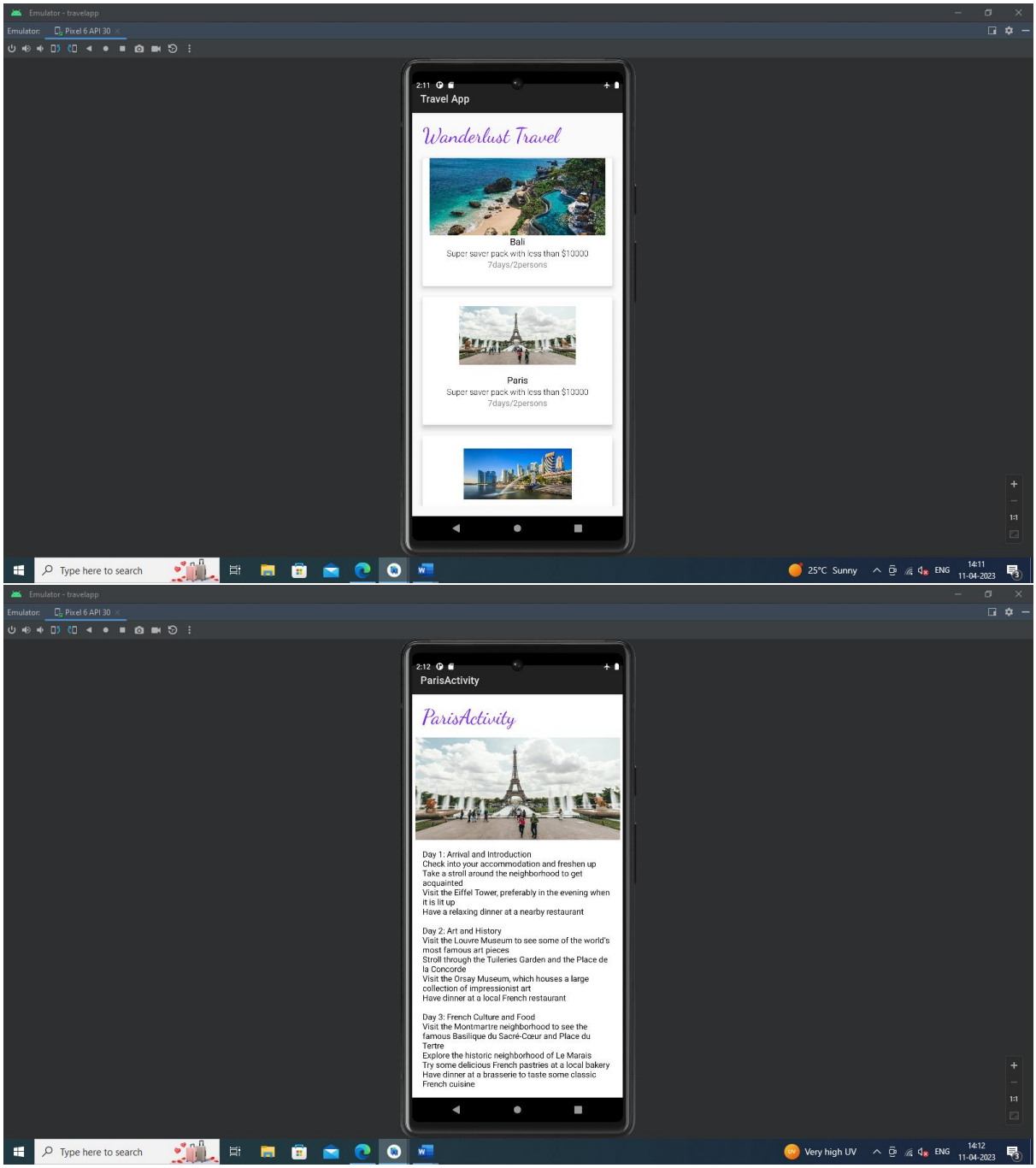
3 Result


Emulator - travelapp
Emulator: Pixel 6 API 30
Register
Username
Email
Password
Register
Have an account? Log in
Type here to search
24°C Sunny
ENG
13:58
11-04-2023


Emulator - travelapp
Emulator: Pixel 6 API 30
Register
Username
raihanabegam
Email
raihanabegam2002@gmail.com
Password
Register
Have an account? Log in
Type here to search
24°C Sunny
ENG
14:01
11-04-2023


Emulator - travelapp
Emulator: Pixel 6 API 30
2:02
Travel App
Login
Username
Password
Login
Register
Forget password?
Type here to search
24°C Sunny
ENG
14:02
11-04-2023


Emulator - travelapp
Emulator: Pixel 6 API 30
2:03
Travel App
Login
Username
raihanabegam
Password
Login
Register
Forget password?
Type here to search
24°C Sunny
ENG
14:03
11-04-2023



4 Advantages and Disadvantages

- Travel planning app is a user-friendly app
- Travel planning app are beneficial for both the business owners and users
- The most popular features of such apps are the capabilities they have to record and store the data of users

- The app can be used 24/7
- The travel planning app helps you to acquire any kind of information regarding the travel plan
- The app is convenient to use without worrying about the issues of time zone

Disadvantages

- Travel planning app have real security and privacy issues
- Android-based apps failed to receive a passing security grade
- Requesting too many permissions
- Assuming always-on data connection
- Assuming high speed network access

5 Applications

- A travel planning app takes the hassle out of planning your next adventure
- It Build space allows users to customize their itineraries based on their preferences
- Travel Planning app development has made it possible for users to track their expenses and set budget limits. This feature particularly helpful for those looking to stick to a specific budget while on vacation
- Get real view pictures of destination
- Users get to know about all Information about the destination they planned to travel

6 Conclusion

The travel planning app helps the people to reach their destination with the proper guidance. it gives detailed information including text and images. The travel planning app provide the users with location Based information which can be browsed.

The travel planning app serves as a useful tool for planning a travel. Travel planning app enable users for easy accessibility and can be run from any part of the world. All over the travel planning app plays an important role in guiding the users to plan their loved destination**.**

7 Future Scope

- This application can easily Implemented under various situation
- We can add new features as and when we required
- Reusability of this application is also available
- Tourism group can use it for managing their location, hotel details
- Any tourist agency can make use of it for saving customer details in database
- The covering access range can be increased
- Rating system can also be imbedded according to the user's satisfaction
- Apart from Android, it can also be mad for Windows and IOS users
- Navigation system can also be integrated for a particular place.

8 Appendix

Source code

Gradle scripts

```
plugins {
    id 'com.android.application'
    id 'org.jetbrains.kotlin.android'
}

android {
    namespace 'com.example.travelapp'
    compileSdk 33

    defaultConfig {
```

```
        applicationId "com.example.travelapp"
        minSdk 21
        targetSdk 33
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
        vectorDrawables {
            useSupportLibrary true
        }
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-
optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    kotlinOptions {
        jvmTarget = '1.8'
    }
    buildFeatures {
        compose true
    }
    composeOptions {
        kotlinCompilerExtensionVersion '1.2.0'
    }
    packagingOptions {
        resources {
            excludes += '/META-INF/{AL2.0,LGPL2.1}'
        }
    }
}

dependencies {

    implementation 'androidx.core:core-ktx:1.7.0'
    implementation 'androidx.lifecycle:lifecycle-runtime-ktx:2.3.1'
    implementation 'androidx.activity:activity-compose:1.3.1'
    implementation "androidx.compose.ui:ui:$compose_ui_version"
    implementation "androidx.compose.ui:ui-tooling-
preview:$compose_ui_version"
    implementation 'androidx.compose.material:material:1.2.0'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
    androidTestImplementation "androidx.compose.ui:ui-test-
junit4:$compose_ui_version"
    debugImplementation "androidx.compose.ui:ui-
tooling:$compose_ui_version"
    debugImplementation "androidx.compose.ui:ui-test-
manifest:$compose_ui_version"
    // Adding Room dependencies
    implementation 'androidx.room:room-common:2.5.0'
    implementation 'androidx.room:room-ktx:2.5.0'
```

```
}
```

User data class

```
plugins {
    id 'com.android.application'
    id 'org.jetbrains.kotlin.android'
}

android {
    namespace 'com.example.travelapp'
    compileSdk 33

    defaultConfig {
        applicationId "com.example.travelapp"
        minSdk 21
        targetSdk 33
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
        vectorDrawables {
            useSupportLibrary true
        }
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-
optimize.txt'), 'proguard-rules.pro'
        }
```

```
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    kotlinOptions {
        jvmTarget = '1.8'
    }
    buildFeatures {
        compose true
    }
    composeOptions {
        kotlinCompilerExtensionVersion '1.2.0'
    }
    packagingOptions {
        resources {
            excludes += '/META-INF/{AL2.0,LGPL2.1}'
        }
    }
}

dependencies {

    implementation 'androidx.core:core-ktx:1.7.0'
    implementation 'androidx.lifecycle:lifecycle-runtime-ktx:2.3.1'
    implementation 'androidx.activity:activity-compose:1.3.1'
    implementation "androidx.compose.ui:ui:$compose_ui_version"
    implementation "androidx.compose.ui:ui-tooling-
preview:$compose_ui_version"
    implementation 'androidx.compose.material:material:1.2.0'
    testImplementation 'junit:junit:4.13.2'
```

```
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'

    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'

    androidTestImplementation "androidx.compose.ui:ui-test-
junit4:$compose_ui_version"

    debugImplementation "androidx.compose.ui:ui-
tooling:$compose_ui_version"

    debugImplementation "androidx.compose.ui:ui-test-
manifest:$compose_ui_version"

    // Adding Room dependencies

    implementation 'androidx.room:room-common:2.5.0'

    implementation 'androidx.room:room-ktx:2.5.0'



}
```

User Dao

```
import androidx.room.*



@Dao

interface UserDao {



    @Query("SELECT * FROM user_table WHERE email = :email")

    suspend fun getUserByEmail(email: String): User?



    @Insert(onConflict = OnConflictStrategy.REPLACE)

    suspend fun insertUser(user: User)



    @Update

    suspend fun updateUser(user: User)



    @Delete

    suspend fun deleteUser(user: User)

}
```

User Database

```
package com.example.travelapp


import android.content.Context
import androidx.room.Database
import androidx.room.Room
import androidx.room.RoomDatabase


@Database(entities = [User::class], version = 1)
abstract class UserDatabase : RoomDatabase() {


    abstract fun userDao(): UserDao


    companion object {


        @Volatile
        private var instance: UserDatabase? = null


        fun getDatabase(context: Context): UserDatabase {
            return instance ?: synchronized(this) {
                val newInstance = Room.databaseBuilder(
                    context.applicationContext,
                    UserDatabase::class.java,
                    "user_database"
                ).build()
                instance = newInstance
                newInstance
            }
        }
```

```
    }

}
```

User Database Helper

```
package com.example.travelapp


import android.annotation.SuppressLint

import android.content.ContentValues

import android.content.Context

import android.database.Cursor

import android.database.sqlite.SQLiteDatabase

import android.database.sqlite.SQLiteOpenHelper


class UserDatabaseHelper(context: Context) :

    SQLiteOpenHelper(context, DATABASE_NAME, null, DATABASE_VERSION) {


    companion object {

        private const val DATABASE_VERSION = 1

        private const val DATABASE_NAME = "UserDatabase.db"


        private const val TABLE_NAME = "user_table"

        private const val COLUMN_ID = "id"

        private const val COLUMN_FIRST_NAME = "first_name"

        private const val COLUMN_LAST_NAME = "last_name"

        private const val COLUMN_EMAIL = "email"

        private const val COLUMN_PASSWORD = "password"

    }
```

Login Activity

```
package com.example.travelapp


```

```
import android.content.Context
import android.content.Intent
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.material.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.input.PasswordVisualTransformation
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.core.content.ContextCompat


class LoginActivity : ComponentActivity() {
    private lateinit var databaseHelper: UserDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = UserDatabaseHelper(this)
        setContent {
            LoginScreen(this, databaseHelper)
```

```
        }
    }
}
@Composable
fun LoginScreen(context: Context, databaseHelper: UserDatabaseHelper) {

    var username by remember { mutableStateOf("") }
    var password by remember { mutableStateOf("") }
    var error by remember { mutableStateOf("") }

    Column(
        modifier = Modifier.fillMaxSize().background(Color.White),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {

        Image(painterResource(id = R.drawable.trav), contentDescription =
"")

        Text(
            fontSize = 36.sp,
            fontWeight = FontWeight.ExtraBold,
            fontFamily = FontFamily.Cursive,
            text = "Login"
        )
        Spacer(modifier = Modifier.height(10.dp))

        TextField(
            value = username,
            onValueChange = { username = it },
            label = { Text("Username") },
```

```
            modifier = Modifier.padding(10.dp)
                .width(280.dp)
        )

        TextField(
            value = password,
            onValueChange = { password = it },
            label = { Text("Password") },
            visualTransformation = PasswordVisualTransformation(),
            modifier = Modifier.padding(10.dp)
                .width(280.dp)
        )

        if (error.isNotEmpty()) {
            Text(
                text = error,
                color = MaterialTheme.colors.error,
                modifier = Modifier.padding(vertical = 16.dp)
            )
        }

        Button(
            onClick = {
                if (username.isNotEmpty() && password.isNotEmpty()) {
                    val user = databaseHelper.getUserByUsername(username)
                    if (user != null && user.password == password) {
                        error = "Successfully log in"
                        context.startActivity(
                            Intent(
                                context,
```

```
                            MainActivity::class.java
                        )
                    )
                    //onLoginSuccess()
                }
                else {
                    error =  "Invalid username or password"
                }

            } else {
                error = "Please fill all fields"
            }
        },
        modifier = Modifier.padding(top = 16.dp)
    ) {
        Text(text = "Login")
    }
    Row {
        TextButton(onClick = {context.startActivity(
            Intent(
                context,
                RegisterActivity::class.java
            )
        )}
        )
        { Text(text = "Register") }
        TextButton(onClick = {
        })

        {
```

```
                Spacer(modifier = Modifier.width(60.dp))
                Text(text = "Forget password?")
            }
        }
    }
}
private fun startMainPage(context: Context) {
    val intent = Intent(context, MainActivity::class.java)
    ContextCompat.startActivity(context, intent, null)
}
```

Registration Activity

```
package com.example.travelapp


import android.content.Context
import android.content.Intent
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.material.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.text.font.FontWeight
```

```
import androidx.compose.ui.text.input.PasswordVisualTransformation

import androidx.compose.ui.tooling.preview.Preview

import androidx.compose.ui.unit.dp

import androidx.compose.ui.unit.sp

import androidx.core.content.ContextCompat


class RegisterActivity : ComponentActivity() {

    private lateinit var databaseHelper: UserDatabaseHelper

    override fun onCreate(savedInstanceState: Bundle?) {

        super.onCreate(savedInstanceState)

        databaseHelper = UserDatabaseHelper(this)

        setContent {

            RegistrationScreen(this, databaseHelper)

        }

    }

}


@Composable

fun RegistrationScreen(context: Context, databaseHelper:
UserDatabaseHelper) {


    var username by remember { mutableStateOf("") }

    var password by remember { mutableStateOf("") }

    var email by remember { mutableStateOf("") }

    var error by remember { mutableStateOf("") }


    Column(

        modifier = Modifier.fillMaxSize().background(Color.White),

        horizontalAlignment = Alignment.CenterHorizontally,

        verticalArrangement = Arrangement.Center

    ) {
```

```
        Image(painterResource(id = R.drawable.tra), contentDescription =
"")


        Text(
            fontSize = 36.sp,
            fontWeight = FontWeight.ExtraBold,
            fontFamily = FontFamily.Cursive,
            text = "Register"
        )


        Spacer(modifier = Modifier.height(10.dp))
        TextField(
            value = username,
            onValueChange = { username = it },
            label = { Text("Username") },
            modifier = Modifier
                .padding(10.dp)
                .width(280.dp)


        )


        TextField(
            value = email,
            onValueChange = { email = it },
            label = { Text("Email") },
            modifier = Modifier
                .padding(10.dp)
                .width(280.dp)
        )
```

```
        TextField(
            value = password,
            onValueChange = { password = it },
            label = { Text("Password") },
            visualTransformation = PasswordVisualTransformation(),
            modifier = Modifier
                .padding(10.dp)
                .width(280.dp)
        )



        if (error.isNotEmpty()) {
            Text(
                text = error,
                color = MaterialTheme.colors.error,
                modifier = Modifier.padding(vertical = 16.dp)
            )
        }


        Button(
            onClick = {
                if (username.isNotEmpty() && password.isNotEmpty() &&
email.isNotEmpty()) {
                    val user = User(
                        id = null,
                        firstName = username,
                        lastName = null,
                        email = email,
                        password = password
                    )
                    databaseHelper.insertUser(user)
```

```
                    error = "User registered successfully"
                    // Start LoginActivity using the current context
                    context.startActivity(
                        Intent(
                            context,
                            LoginActivity::class.java
                        )
                    )

                } else {
                    error = "Please fill all fields"
                }
            },
            modifier = Modifier.padding(top = 16.dp)
        ) {
            Text(text = "Register")
        }
        Spacer(modifier = Modifier.width(10.dp))
        Spacer(modifier = Modifier.height(10.dp))

        Row() {
            Text(
                modifier = Modifier.padding(top = 14.dp), text = "Have an
account?"
            )
            TextButton(onClick = {
                context.startActivity(
                    Intent(
                        context,
                        LoginActivity::class.java
                    )
```

```
                )
            })


            {
                Spacer(modifier = Modifier.width(10.dp))
                Text(text = "Log in")
            }
        }
    }
}
private fun startLoginActivity(context: Context) {
    val intent = Intent(context, LoginActivity::class.java)
    ContextCompat.startActivity(context, intent, null)
}
package com.example.travelapp


import android.content.Context
import android.content.Intent
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.Card
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
```

```
import androidx.compose.ui.draw.scale
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontFamily
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


class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            TravelApp(this)
        }
    }



    @Composable
    fun TravelApp(context: Context) {
        Column(
            modifier = Modifier
                .padding(20.dp)
                .verticalScroll(rememberScrollState())

        ) {

            Text(
                fontSize = 40.sp,
```

```
                color = Color(android.graphics.Color.rgb(120, 40, 251)),
                fontFamily = FontFamily.Cursive,
                text = "Wanderlust Travel"
            )

            Spacer(modifier = Modifier.height(20.dp))

            // 01
            Card(
                modifier = Modifier
                    .fillMaxWidth()
                    .height(250.dp)
                    .clickable {
                        context.startActivity(
                            Intent(context, BaliActivity::class.java)

                        )
                    },
                elevation = 8.dp
            )
            {
                Column(
                    horizontalAlignment = Alignment.CenterHorizontally
                ) {
                    Image(
                        painterResource(id = R.drawable.bali),
contentDescription = "",
                        modifier = Modifier
                            .height(150.dp)
                            .scale(scaleX = 1.2F, scaleY = 1F)
                    )
```

```
                    Text(
                        text = stringResource(id = R.string.place_1),
                        fontSize = 18.sp
                    )



                    Text(
                        text = stringResource(id = R.string.description),
                        fontWeight = FontWeight.Light,
                        fontSize = 16.sp,
                        textAlign = TextAlign.Center,
                    )


                    Text(
                        text = stringResource(id = R.string.plan), color =
Color.Gray,
                        fontSize = 16.sp
                    )
                }
            }


            Spacer(modifier = Modifier.height(20.dp))



            //02
            Card(
                modifier = Modifier
                    .fillMaxWidth()
                    .height(250.dp)
                    .clickable {
```

```
                        context.startActivity(
                            Intent(context, ParisActivity::class.java)

                        )
                    },
                elevation = 8.dp
            )
            {
                Column(
                    horizontalAlignment = Alignment.CenterHorizontally
                ) {
                    Image(
                        painterResource(id = R.drawable.paris),
contentDescription = "",
                        modifier = Modifier
                            .height(150.dp)
                            .scale(scaleX = 1.2F, scaleY = 1F)
                    )

                    Text(
                        text = stringResource(id = R.string.place_2),
                        fontSize = 18.sp
                    )


                    Text(
                        text = stringResource(id = R.string.description),
                        fontWeight = FontWeight.Light,
                        fontSize = 16.sp,
                        textAlign = TextAlign.Center,
                    )
```

```
                    Text(
                        text = stringResource(id = R.string.plan), color =
Color.Gray,
                        fontSize = 16.sp
                    )
                }
            }

            Spacer(modifier = Modifier.height(20.dp))

            //03
            Card(
                modifier = Modifier
                    .fillMaxWidth()
                    .height(250.dp)
                    .clickable {
                        context.startActivity(
                            Intent(context, SingaporeActivity::class.java)

                        )
                    },
                elevation = 8.dp
            )
            {
                Column(
                    horizontalAlignment = Alignment.CenterHorizontally
                ) {
                    Image(
                        painterResource(id = R.drawable.singapore),
contentDescription = "",
```

```
                        modifier = Modifier
                            .height(150.dp)
                            .scale(scaleX = 1.2F, scaleY = 1F)
                    )

                    Text(
                        text = stringResource(id = R.string.place_3),
                        fontSize = 18.sp
                    )



                    Text(
                        text = stringResource(id = R.string.description),
                        fontWeight = FontWeight.Light,
                        fontSize = 16.sp,
                        textAlign = TextAlign.Center,
                    )


                    Text(
                        text = stringResource(id = R.string.plan), color =
Color.Gray,
                        fontSize = 16.sp
                    )
                }
            }

            Spacer(modifier = Modifier.height(20.dp))
        }
    }
}
```

Bali Activity

```
package com.example.travelapp


import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.MaterialTheme
import androidx.compose.material.Surface
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.scale
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.travelapp.ui.theme.TravelAppTheme


class BaliActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
```

```
            TravelAppTheme {
                // A surface container using the 'background' color from
the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                ) {
                    PlaceOne()
                }
            }
        }
    }
}


@Composable
fun PlaceOne() {
    Column(modifier = Modifier.background(color = Color.White)
        .padding(20.dp)
        .verticalScroll(rememberScrollState())
    ) {
        Text(
            fontSize = 40.sp,
            color = Color(android.graphics.Color.rgb(120, 40, 251)),
            fontFamily = FontFamily.Cursive,
            text = stringResource(id = R.string.place_1),
        )
        Image(
            painterResource(id = R.drawable.bali), contentDescription = "",
            modifier = Modifier
                .padding(16.dp)
                .fillMaxWidth()
```

```
                .height(200.dp)
                .scale(scaleX = 1.2F, scaleY = 1F)
        )
        Text(
            color=Color.Black,
            text = "Day 1: Arrival and Relaxation\n" +

                    "Arrive in Bali and check into your hotel or
accommodation.\n" +
                    "Spend the day relaxing and getting acclimated to the
island.\n" +
                    "If you have time, explore the nearby area or head to
the beach.\n" +
                    "\n" +

                    "Day 2: Ubud Tour\n" +

                    "Start your day early and head to Ubud, a cultural and
artistic hub in Bali.\n" +
                    "Visit the Monkey Forest and the Ubud Palace.\n" +
                    "Take a tour of the Tegalalang Rice Terrace, a
beautiful UNESCO World Heritage Site.\n" +
                    "End your day with a traditional Balinese dance
performance.\n" +
                    "\n" +
                    "Day 3: Temple Hopping\n" +

                    "Visit some of Bali's most famous temples, such as
Tanah Lot and Uluwatu.\n" +
                    "Take in the stunning views of the ocean and cliffs.\n"
+
                    "Enjoy a sunset dinner at one of the many restaurants
near the temples.\n" +
                    "\n" +
                    "Day 4: Waterfalls and Beaches\n" +
```

```
                    "Take a day trip to Bali's beautiful waterfalls, such
as Tegenungan or Gitgit.\n" +

                    "Spend the afternoon at one of Bali's world-renowned
beaches, like Seminyak or Nusa Dua.\n" +

                    "\n" +

                    "Day 5: Island Hopping\n" +



                    "Take a day trip to one of Bali's neighboring islands,
such as Nusa Lembongan or Gili Islands.\n" +

                    "Snorkel or scuba dive in the clear waters and relax on
the beach.\n" +

                    "\n" +

                    "Day 6: Cultural Activities\n" +

                    "Visit a traditional Balinese village and learn about
the island.\n" +

                    "\n" +

                    "Day 7: Departure\n" +

                    "Explore the surrounding area and take in the stunning
sunset views.\n" +

                    "Have dinner at a local restaurant before returning to
your accommodation."

        )




    }

}
```

Paris Activity

```
package com.example.travelapp


import android.os.Bundle

import androidx.activity.ComponentActivity

import androidx.activity.compose.setContent
```

```
import androidx.compose.foundation.Image

import androidx.compose.foundation.background

import androidx.compose.foundation.layout.*

import androidx.compose.foundation.rememberScrollState

import androidx.compose.foundation.verticalScroll

import androidx.compose.material.MaterialTheme

import androidx.compose.material.Surface

import androidx.compose.material.Text

import androidx.compose.runtime.Composable

import androidx.compose.ui.Modifier

import androidx.compose.ui.draw.scale

import androidx.compose.ui.graphics.Color

import androidx.compose.ui.res.painterResource

import androidx.compose.ui.res.stringResource

import androidx.compose.ui.text.font.FontFamily

import androidx.compose.ui.tooling.preview.Preview

import androidx.compose.ui.unit.dp

import androidx.compose.ui.unit.sp

import com.example.travelapp.ui.theme.TravelAppTheme


class ParisActivity : ComponentActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {

        super.onCreate(savedInstanceState)

        setContent {

            TravelAppTheme {

                // A surface container using the 'background' color from
the theme

                Surface(

                    modifier = Modifier.fillMaxSize(),

                    color = MaterialTheme.colors.background

                ) {
```

```
                    Greeting()
                }
            }
        }
    }
```

Singapore Activity

```
package com.example.travelapp


import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.MaterialTheme
import androidx.compose.material.Surface
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.scale
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.travelapp.ui.theme.TravelAppTheme
```

```
class SingaporeActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            TravelAppTheme {
                // A surface container using the 'background' color from
the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                ) {
                    Greeting2()
                }
            }
        }
    }
```

Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        tools:targetApi="31">
```

```
        <activity
            android:name=".SingaporeActivity"
            android:exported="false"
            android:label="@string/title_activity_singapore"
            android:theme="@style/Theme.Travelapp" />
        <activity
            android:name=".BaliActivity"
            android:exported="false"
            android:label="@string/title_activity_bali"
            android:theme="@style/Theme.Travelapp" />
        <activity
            android:name=".RegisterActivity"
            android:exported="false"
            android:label="RegisterActivity"
            android:theme="@style/Theme.Travelapp" />
        <activity
            android:name=".ParisActivity"
            android:exported="false"
            android:label="@string/title_activity_paris"
             />
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:label="@string/app_name"
             />
        <activity
            android:name=".LoginActivity"
            android:exported="true"
            android:label="@string/app_name"
           >
```

```
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />


                <category android:name="android.intent.category.LAUNCHER"
/>
            </intent-filter>
        </activity>
    </application>


</manifest>
```