## WANDERLUST: A PERSONALIZED TRAVEL PLANNING AND TRACKING APP

This is to certify that this project report entitled “**Wanderlust: A personalized Travel Planning and Tracking App**” is a bonafide record of work done by

M. Yamini

S .Pradeepa

M .Yoheeswari

M. Gayathri

**1. INTRODUCTION**

**1.1 OVERVIEW**

The Travel Android App Project is an exciting and innovative app development project aimed at providing an all-in-one travel solution for Android users. The app is designed to provide users with access to a wide range of travel information, including flight schedules, hotel bookings, travel guides, weather information, and more.

The app will feature a user-friendly interface that will help users effortlessly plan trips, book hotels and flights, and discover new destinations. The app will also integrate with popular travel providers, such as Expedia and Booking.com, to provide users with the best travel deals and discounts.

In addition, the app will feature personalized recommendations and trip suggestions, based on user preferences and past travel history. The app will also provide users with the ability to share travel experiences with friends and family through social media integration.

Overall, the Travel Android App Project aims to provide a comprehensive travel platform that will help users plan, book, and enjoy their travel experiences to the fullest. It is an exciting opportunity to develop an app that will make travel planning and booking easier, more accessible, and more enjoyable for Android users.

**PROJECT WORKFLOW**

As an AI language model, I cannot access actual live scenarios. However, I can provide you with a possible workflow for a personalized travel planning and tracking app.

**1**. User registration and profile creation: The user signs up for the app and creates their profile, which includes details such as name, age, gender, preferences, budget, and travel history.

**2**. Trip planning: The user selects their travel destination, trip duration, preferred mode of transportation, and accommodation type. Based on these details, the app suggests personalized itineraries, activity options, and recommend booking options.

**3.** Booking and payment: The user selects their preferred itinerary and books their flights, accommodations, tours, and activities directly through the app. The app securely processes their payment and sends a confirmation message.

**4**. Packing checklist and reminders: The app generates a packing checklist based on the user's destination, time of year, and activity plans. It sends reminders to the user to ensure they pack everything they need for their trip.

**5**. Travel tracking and sharing: The app provides users with real-time updates on their trip, including flight schedules, hotel check-in/out, and local weather. Users can also share their travel experiences with family and friends by posting photos, videos, and blogs directly from the app.

**6.** Expense tracking and management: The app allows users to track their expenses in real-time, including food, transportation, and excursion costs. Users can set budgets for their trip and get alerts when they are close to exceeding them.

**7.** User feedback and ratings: The app prompts users to rate their travel experiences, accommodations, tours, and activities. This feedback can help

future users make informed decisions when planning their trips.

**8.** Loyalty Program and Discounts: The app can provide users with discounts and rewards points based on their travel activities, such as booking flights, hotel rooms, or tours. These points can be redeemed for future trips or other travel-related perks.

**9**. Customer support: The app offers customer support and assistance to users in case of any issues such as cancellations or booking changes.

**10.** Security and Data Protection: The app ensures that user data is protected and secure by encrypting personal information such as credit card details and passport information.

**1.2 PURPOSE**

The purpose of a personalized travel planning and tracking app is to assist travelers in planning and organizing their trips in a more convenient, efficient and enjoyable manner. It provides travelers with personalized trip recommendations, bookings, and tracking of travel activities. Here are some specific purposes of a personalized travel planning and tracking app:

**1.** Trip Planning: It helps travelers plan their trips by suggesting personalized itineraries based on their preferences, budget, and travel history.

**2.** Booking and Payment: The app enables travelers to book their flights, accommodations, tours, and activities directly from the app, making the process more convenient, efficient and time-saving.

**3.** Trip Tracking: The app provides real-time updates on travel activities such as flight schedules, hotel check-in/out, and local weather to help travelers stay organized and up to date.

**4.** Expense Tracking: The app helps travelers track their expenses in real-time, allowing them to stick to their budget and make informed decisions along the way.

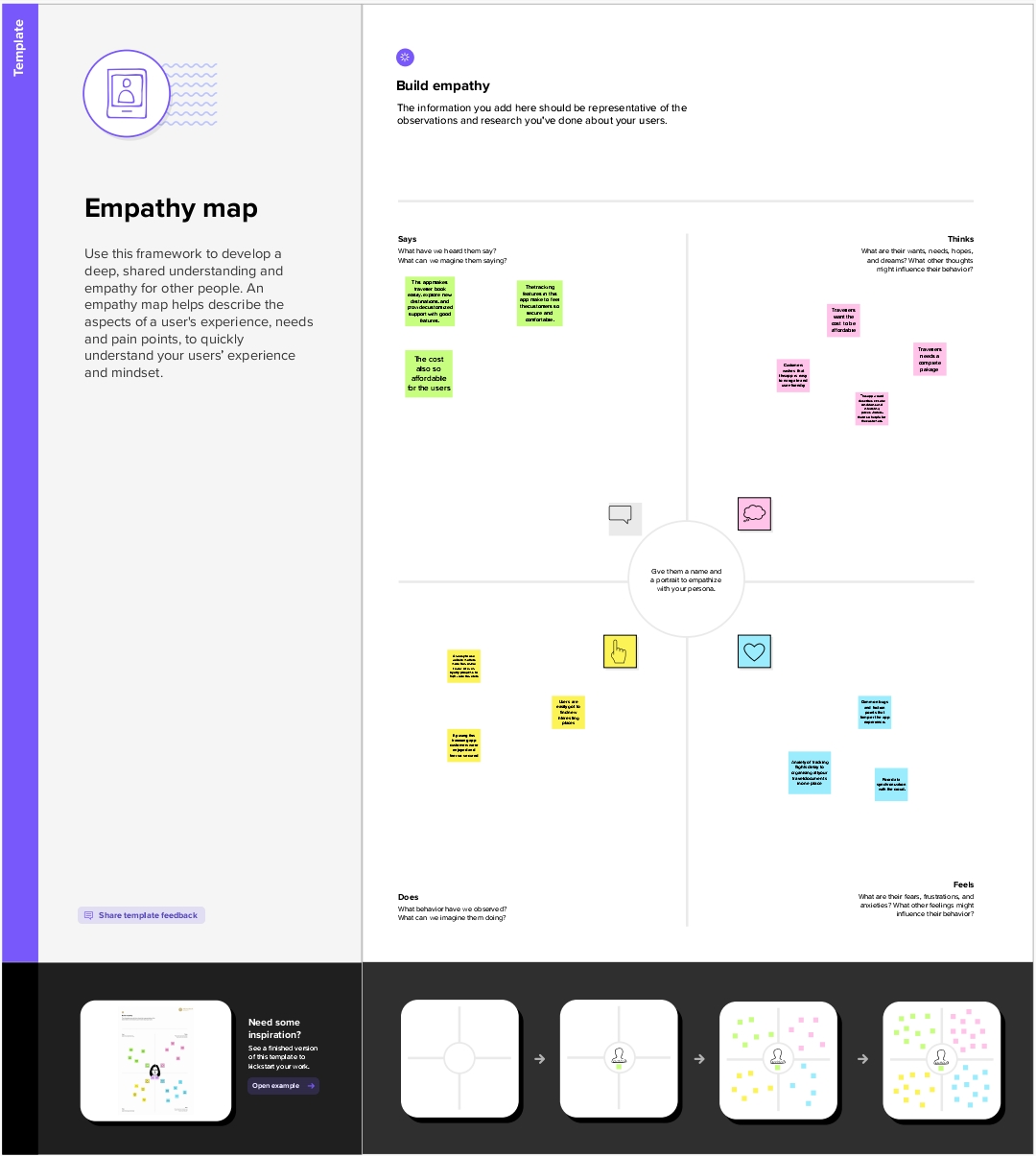
**5.** Personalization: The app provides personalized recommendations based on each user's preferences, making the travel experience unique and enjoyable.

**6.** Loyalty Program and Discounts: The app can offer travelers discounts and rewards points based on their travel activities, such as booking flights, hotel rooms or tours, providing them with a sense of value for their purchase.

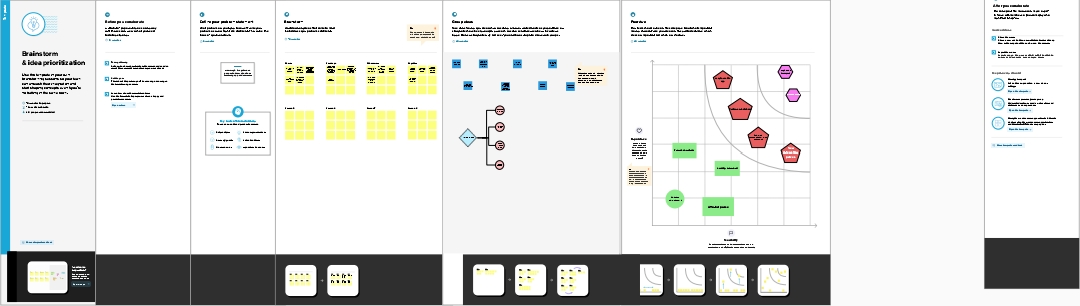
Overall, a personalized travel planning and tracking app aims to make traveling easier, more convenient, and enjoyable for travelers. It ensures that travelers have a seamless experience before, during, and after their trip. The main purpose of a travel app is to provide valuable information about travel destinations, flights, accommodations, local transportation, tourist attractions, restaurants, and other relevant details that can help travelers have a successful and enjoyable journey. It can also help travelers monitor their travel arrangements and keep all their travel-related documents in one place, making it easier for them to access and keep track of important information.

**2. PROBLEM DEFINITION AND DESIGN THINKING.**

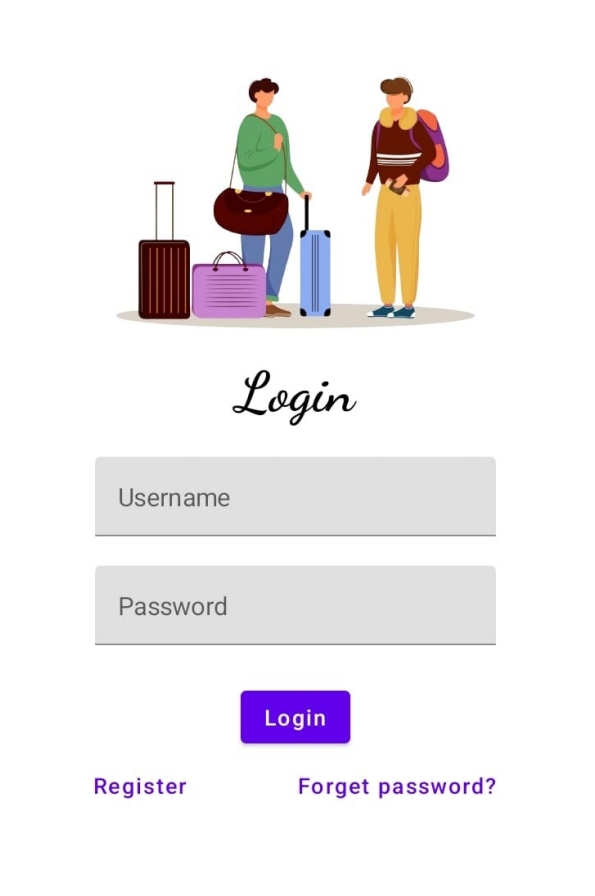
**2.1 EMPATHY MAP**

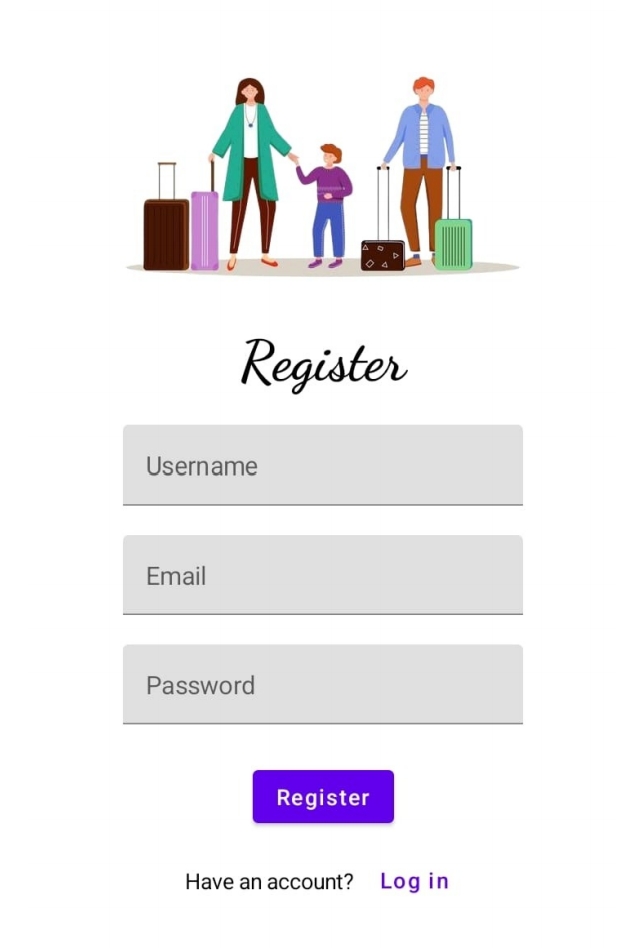


**2.2 IDEATION BRAINSTORMING**

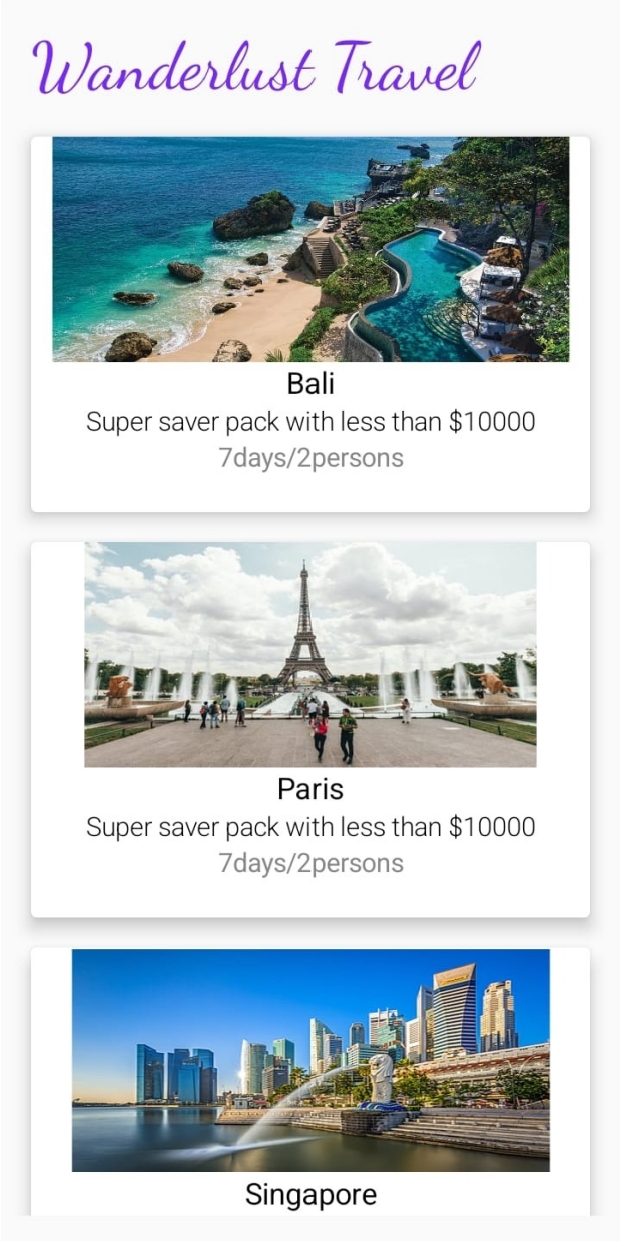


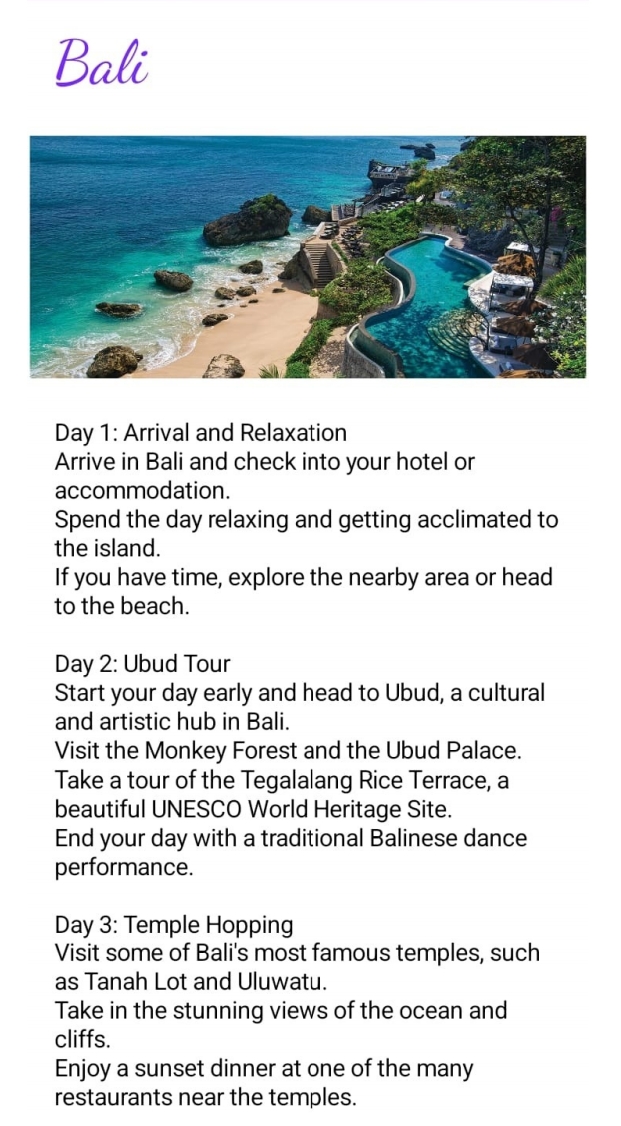
**3.RESULT**

**LOGIN PAGE**

**REGISTER PAGE**

**MAIN PAGE**

****

**LOCATION PAGE**

**4. ADVANTAGES AND DISADVANTAGES**

**ADVANTAGES**

**1. Customization**: A personalized travel app allows users to create their own itinerary based on their personal preferences and interests. This ensures that the trip is tailored to their liking.

**2. Saves time and money:** The app can suggest the best routes to take, the best time to travel and the best places to eat, stay and visit, which can save users both time and money.

**3. Convenience:** Planning a trip can be time-consuming and stressful. With a personalized travel app, users can plan their trip on-the-go, in real-time.

**4. Better travel experience:** By providing customized recommendations, a personalized travel app can enhance the travel experience by suggesting places that may not be found in guidebooks or online reviews.

**5. User-friendly:** Travel apps are designed to be user-friendly and straightforward, making it easy for users to navigate and explore new places.

**6. Access to local knowledge:** The app can provide users with insights and tips from local experts and residents, giving travelers an authentic and immersive experience.

**7. Safety and security:** A personalized travel app can also offer safety and security features, such as risk assessments and emergency contacts, ensuring that users have a safe and enjoyable trip.

**DISADVANTAGES**

**1. Inaccurate information:** Personalized travel apps may rely on user-generated content, which can be inaccurate or out of date. This could lead to disappointment or frustration for travelers who rely on this information to make decisions.

**2. Cost:** Some personalized travel apps may require a subscription or premium features that come with a cost, which may not be affordable for all travelers.

**3. Technical issues:** Like all technology, travel apps can experience technical issues that could prevent users from accessing certain features or information.

**4. Dependency on technology:** Travelers may become too reliant on the app and miss out on the spontaneous discoveries and experiences that make travel so exciting.

**5. Privacy risks:** Travel apps may collect and store personal information that could be at risk of being accessed or stolen by hackers.

**6. Limited range:** Some travel apps may only cover a certain geographic area or focus on a specific type of travel, which could limit its usefulness for some travelers.

**7. User preference:** Personalized travel apps tailor recommendations based on user preferences, which can be limiting in terms of discovering new experiences and expanding interests.

**5. APPLICATION**

**1.** Flight search engine that allows users to compare flight prices and travel routes from different airlines.

**2.** Hotel booking system that allows users to search for accommodation intheir desired location, compare prices and amenities, and book their stay.

**3.** Travel itinerary planner that helps users plan their trip, including transportation, accommodation, and activities.

**4.** Destination guides that provide information on popular tourist attractions, local cuisine, and cultural customs.

**5.** Language translator that helps users communicate with locals in foreign countries.

**6.** Currency converter that converts currency rates in real-time

**7.** Weather forecast to help users plan their trip based on the weather conditions.

**8.** GPS navigation system that provides directions to popular tourist attractions.

**9.** Local events calendar that suggests events, festivals, and concerts happening nearby.

**10**. Offline access to maps, destination information, and travel guides for users without internet access.

**ABOUT ANDROID STUDIO APP APPLICATION**

Android Studio is the official integrated development environment (IDE) for developing Android applications. It is based on the IntelliJ IDEA and specifically designed for Android development.

Some of the key features of Android Studio include:

**1. User interface designer:** You can easily drag and drop UI elements with the built-in user interface designer.

**2. Code editor:** The code editor in Android Studio is customizable and supports syntax highlighting, code completion, and refactoring.

**3. Android emulator:** Android Studio comes with a built-in emulator to test and run your application on different virtual devices.

**4. Gradle build system:** Android Studio uses Gradle as the build system to manage dependencies and create different build versions of your application.

**5. Debugging tools:** You can easily debug your application with Android Studio’s built-in debugging tools.

**6. Android SDK and API support:** Android Studio comes with support for the latest SDK and APIs, making it easy to integrate new features into your application.

Overall, if you're interested in developing Android applications, Android Studio is the go-to IDE for beginners and experts alike.

**6.CONCLUSION**

Over the years, the travel industry has undergone a massive transformation, with digital solutions driving growth and changing the way people travel. Personalized travel planning and tracking app is one such solution that has revolutionized the travel experience for millions of people globally.

The primary objective of a personalized travel planning and tracking app is to provide a complete and seamless travel experience to the users. The app does this by offering a suite of services that caters to all aspects of travel, from trip planning to real-time tracking, itinerary management, and much more.

The app's first benefit is the personalized trip planning services, which ensure travelers get tailored recommendations that suit their unique needs and preferences. For instance, users can easily set their preference for accommodation type, transport medium, and activities they love to engage in while on the road. The app will then offer a comprehensive list of recommended locations, top attractions, and restaurants that suit their specific needs. This personalized approach helps travelers save time, reduce stress, and make confident decisions about their travel plans.

In addition to planning, the app also offers itinerary management services where travelers can easily manage their entire travel schedule in one place. They can also include all their travel details such as flight timings, hotel reservations, and other important bookings, conveniently available in a personalized itinerary. This feature ensures the traveler remains organized and has all the relevant information they need at their fingertips.

Real-time tracking is another feature that is increasingly valuable to users of a personalized travel planning and tracking app. Travelers can track their journey from start to finish with live updates on their mobile devices. Through the app, travelers can receive real-time notifications about flight delays, weather updates, and any other essential travel information that they need to keep their trip on track. The app's tracking feature helps travelers stay on top of their route and ensures they are always aware of any unexpected changes.

The app also offers safety and privacy features tailored to the user. Safety features such as location tracking and emergency contacts can help put travelers at ease, giving them peace of mind when exploring new and unfamiliar places. At the same time, privacy features help protect user data by ensuring that only the user can access their travel information.

Finally, the app offers personalized recommendations on popular tourist destinations, restaurants, and travel essentials. This feature helps users make informed and tailor-made decisions about the places they visit, eat, and shop. The app narrows down recommendations based on the users' prior experiences and preferences, ensuring that travelers enjoy a truly personalized travel experience.

In conclusion, a personalized travel planning and tracking app is an essential tool for modern-day travelers. It offers a suite of personalized services that cater to all aspects of travel, from trip planning to itinerary management, real-time tracking, safety and privacy, and personalized recommendations. The app simplifies travel by providing travelers with the best travel experience possible while reducing stress and providing peace of mind. With this app, users can travel efficiently and confidently, ensuring a memorable travel experience.

**7.FUTURESCOPE**

As with any technology, there is always room for improvement, and personalized travel planning and tracking apps are no exception. In the future, some enhancements that can be made to the app include:

**1. Augmented Reality (AR) Integration:** AR is increasingly becoming popular in the travel industry, and integrating it into a personalized travel planning and tracking app can create an immersive experience for travelers. For instance, with AR, users can virtually explore a destination, preview activities, and sightsee without leaving their home.

**2. Artificial Intelligence (AI)-driven Personalization:** The use of AI could transform the way personalized travel planning and tracking apps function. By analyzing user data such as their travel patterns, activities, and preferences, an AI-powered app can offer more accurate recommendations and personalized trip plans that cater to the traveler's unique needs and preferences.

**3. Seamless Integration with Travel Suppliers:** To enable a smoother and more seamless travel experience, personalized travel planning and tracking apps can integrate with airlines, hotels, rental car companies, and other travel suppliers. Through these integrations, users can quickly book their preferred options and have it reflected in their personalized itinerary.

**4. Real-time Translation:** Real-time translation can help eliminate language barriers and make it easier for users to communicate in new and unfamiliar locations. For instance, users can get instant translations of signage, menus, and conversations with local people, making it easier for them to navigate and enjoy their travel experience.

**5. Social Networking Integration:** Personalized travel planning and tracking apps can incorporate social networking functionality, where users can connect with other travelers and locals in their destination. With social networking integration, users can share their experiences, get recommendations, and make new friends while on the road.

**6. Intelligent and Dynamic Pricing:** Apps can utilize big data and AI to provide users with real-time price incentives based on demand and supply dynamics. This approach can provide users with more affordable travel options while generating additional revenue for service providers.

**7. Environmental Sustainability:** As sustainable tourism becomes increasingly popular, personalized travel planning and tracking apps can integrate sustainability features that promote eco-friendly travel choices. These features could include recommendations for eco-friendly accommodations, carbon offsetting, and other initiatives that encourage responsible travel.

**8.APPENDIX**

**A.SOURCE CODE**

**DATA CLASS CODE**

package com.example.travelapp

import androidx.room.ColumnInfo

import androidx.room.Entity

import androidx.room.PrimaryKey

@Entity(tableName = "user\_table")

data class User(

@PrimaryKey(autoGenerate = true) val id: Int?,

@ColumnInfo(name = "first\_name") val firstName: String?,

@ColumnInfo(name = "last\_name") val lastName: String?,

@ColumnInfo(name = "email") val email: String?,

@ColumnInfo(name = "password") val password: String?,

)

**USERDAO INTERFACE CODE**

package com.example.travelapp

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 CLASS CODE

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 DATA BASE HELPER CLASS CODE**

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"

}

override fun onCreate(db: SQLiteDatabase?) {

val createTable = "CREATE TABLE $TABLE\_NAME (" +

"$COLUMN\_ID INTEGER PRIMARY KEY AUTOINCREMENT, " +

"$COLUMN\_FIRST\_NAME TEXT, " +

"$COLUMN\_LAST\_NAME TEXT, " +

"$COLUMN\_EMAIL TEXT, " +

"$COLUMN\_PASSWORD TEXT" +

")"

db?.execSQL(createTable)

}

override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {

db?.execSQL("DROP TABLE IF EXISTS $TABLE\_NAME")

onCreate(db)

}

fun insertUser(user: User) {

val db = writableDatabase

val values = ContentValues()

values.put(COLUMN\_FIRST\_NAME, user.firstName)

values.put(COLUMN\_LAST\_NAME, user.lastName)

values.put(COLUMN\_EMAIL, user.email)

values.put(COLUMN\_PASSWORD, user.password)

db.insert(TABLE\_NAME, null, values)

db.close()

}

@SuppressLint("Range")

fun getUserByUsername(username: String): User? {

val db = readableDatabase

val cursor: Cursor = db.rawQuery("SELECT \* FROM $TABLE\_NAME WHERE $COLUMN\_FIRST\_NAME = ?", arrayOf(username))

var user: User? = null

if (cursor.moveToFirst()) {

user = User(

id = cursor.getInt(cursor.getColumnIndex(COLUMN\_ID)),

firstName = cursor.getString(cursor.getColumnIndex(COLUMN\_FIRST\_NAME)),

lastName = cursor.getString(cursor.getColumnIndex(COLUMN\_LAST\_NAME)),

email = cursor.getString(cursor.getColumnIndex(COLUMN\_EMAIL)),

password = cursor.getString(cursor.getColumnIndex(COLUMN\_PASSWORD)),

)

}

cursor.close()

db.close()

return user

}

@SuppressLint("Range")

fun getUserById(id: Int): User? {

val db = readableDatabase

val cursor: Cursor = db.rawQuery("SELECT \* FROM $TABLE\_NAME WHERE $COLUMN\_ID = ?", arrayOf(id.toString()))

var user: User? = null

if (cursor.moveToFirst()) {

user = User(

id = cursor.getInt(cursor.getColumnIndex(COLUMN\_ID)),

firstName = cursor.getString(cursor.getColumnIndex(COLUMN\_FIRST\_NAME)),

lastName = cursor.getString(cursor.getColumnIndex(COLUMN\_LAST\_NAME)),

email = cursor.getString(cursor.getColumnIndex(COLUMN\_EMAIL)),

password = cursor.getString(cursor.getColumnIndex(COLUMN\_PASSWORD)),

)

}

cursor.close()

db.close()

return user

}

@SuppressLint("Range")

fun getAllUsers(): List<User> {

val users = mutableListOf<User>()

val db = readableDatabase

val cursor: Cursor = db.rawQuery("SELECT \* FROM $TABLE\_NAME", null)

if (cursor.moveToFirst()) {

do {

val user = User(

id = cursor.getInt(cursor.getColumnIndex(COLUMN\_ID)),

firstName = cursor.getString(cursor.getColumnIndex(COLUMN\_FIRST\_NAME)),

lastName = cursor.getString(cursor.getColumnIndex(COLUMN\_LAST\_NAME)),

email = cursor.getString(cursor.getColumnIndex(COLUMN\_EMAIL)),

password = cursor.getString(cursor.getColumnIndex(COLUMN\_PASSWORD)),

)

users.add(user)

} while (cursor.moveToNext())

}

cursor.close()

db.close()

return users

}

}

**LOGINACTIVITY CODE**

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)}

**REGISTERACTIVITYCODE**

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)

}

**MAINACTIVITY DATABASE CODE**

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))

}

}

}

**BALIACTIVITY DATABASE CODE**

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."

)

}

}

**PARISACTIVITY DATABASE CODE**

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()

}

}

}

}

}

@Composable

fun Greeting() {

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\_2),

)

Image(

painterResource(id = R.drawable.paris), 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 Introduction\n" +

"Check into your accommodation and freshen up\n" +

"Take a stroll around the neighborhood to get acquainted\n" +

"Visit the Eiffel Tower, preferably in the evening when it is lit up\n" +

"Have a relaxing dinner at a nearby restaurant\n" +

"\n" +

"Day 2: Art and History\n" +

"Visit the Louvre Museum to see some of the world's most famous art pieces\n" +

"Stroll through the Tuileries Garden and the Place de la Concorde\n" +

"Visit the Orsay Museum, which houses a large collection of impressionist art\n" +

"Have dinner at a local French restaurant\n" +

"\n" +

"Day 3: French Culture and Food\n" +

"Visit the Montmartre neighborhood to see the famous Basilique du Sacré-Cœur and Place du Tertre\n" +

"Explore the historic neighborhood of Le Marais\n" +

"Try some delicious French pastries at a local bakery\n" +

"Have dinner at a brasserie to taste some classic French cuisine\n" +

"\n" +

"Day 4: Architecture and Gardens\n" +

"Visit the Palace of Versailles, a UNESCO World Heritage site, and explore its beautiful gardens\n" +

"Walk along the Champs-Elysées and stop at the Arc de Triomphe\n" +

"Visit the Sainte-Chapelle, a beautiful Gothic chapel with stunning stained-glass windows\n" +

"Have dinner at a local restaurant in the 7th arrondissement\n" +

"\n" +

"Day 5: Shopping and Sightseeing\n" +

"Visit the Notre-Dame Cathedral and climb up to the top for a stunning view of the city\n" +

"Explore the Latin Quarter and visit the Panthéon\n" +

"Go shopping at the famous Galeries Lafayette or Printemps department stores\n" +

"Have dinner at a local bistro\n" +

"\n" +

"Day 6: Parisian Parks and Museums\n" +

"Visit the Musée Rodin and explore its beautiful gardens\n" +

"Stroll through the Luxembourg Gardens and visit the Luxembourg Palace\n" +

"Visit the Centre Pompidou, a modern art museum in the Marais neighborhood\n" +

"Have dinner at a local restaurant in the Latin Quarter\n" +

"\n" +

"Day 7: River Cruise and Farewell\n" +

"Take a boat cruise along the Seine River to see the city from a different perspective\n" +

"Visit the Musée de l'Orangerie, which houses Monet's famous water lilies paintings\n" +

"Have a farewell dinner at a Michelin-starred restaurant"

)

}

}

**SINGAPORE ACTIVITY DATABASE CODE**

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()

}

}

}

}

}

@Composable

fun Greeting2() {

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\_3),

)

Image(

painterResource(id = R.drawable.singapore), contentDescription = "",

modifier = Modifier

.padding(16.dp)

.fillMaxWidth()

.height(200.dp)

.scale(scaleX = 1.2F, scaleY = 1F)

)

Text(

color = Color.Black,

text = "Day 1:\n" +

"Morning: Visit Gardens by the Bay and marvel at the Supertree Grove and the Flower Dome and Cloud Forest conservatories.\n" +

"Afternoon: Explore the Marina Bay Sands complex, which includes a casino, luxury shopping mall, and observation deck with a stunning view of the city.\n" +

"\n" +

"Day 2:\n" +

"Morning: Explore the historic district of Chinatown, including the Buddha Tooth Relic Temple and Museum and the Sri Mariamman Temple.\n" +

"Afternoon: Visit the nearby Clarke Quay for lunch and to explore its waterfront restaurants, bars, and shops.\n" +

"\n" +

"Day 3:\n" +

"Morning: Take a tour of the UNESCO-listed Botanic Gardens, one of the world's most famous and significant tropical gardens.\n" +

"Afternoon: Head over to the National Museum of Singapore, which houses a vast collection of historical and cultural artifacts.\n" +

"\n" +

"Day 4:\n" +

"Morning: Visit the Singapore Zoo and admire the wildlife, including orangutans, tigers, and elephants.\n" +

"Afternoon: Head over to Sentosa Island and relax at one of its many beaches or try some of the many attractions such as Universal Studios Singapore or Adventure Cove Waterpark.\n" +

"\n" +

"Day 5:\n" +

"Morning: Go on a nature walk at MacRitchie Reservoir, which offers hiking trails and stunning views of the city skyline.\n" +

"Afternoon: Visit Little India, a vibrant and colorful neighborhood, and explore the shops, temples, and food stalls.\n" +

"\n" +

"Day 6:\n" +

"Morning: Explore the trendy neighborhood of Tiong Bahru, known for its hip cafes and boutiques, as well as its Art Deco architecture.\n" +

"Afternoon: Visit the National Gallery Singapore, which houses the largest public collection of modern art in Singapore and Southeast Asia.\n" +

"\n" +

"Day 7:\n" +

"Morning: Take a day trip to the nearby island of Pulau Ubin, where you can rent a "

)

}

}

**ANDROID MANIFEST.XML CODE**

<?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"

android:theme="@style/Theme.TravelApp"

tools:targetApi="31">

<activity

android:name=".RegisterActivity"

android:exported="false"

android:label="RegisterActivity"

android:theme="@style/Theme.TravelApp" />

<activity

android:name=".SingaporeActivity"

android:exported="false"

android:label="@string/title\_activity\_singapore"

android:theme="@style/Theme.TravelApp" />

<activity

android:name=".ParisActivity"

android:exported="false"

android:label="@string/title\_activity\_paris"

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=".MainActivity"

android:exported="true"

android:label="@string/app\_name"

android:theme="@style/Theme.TravelApp"/>

<activity

android:name=".LoginActivity"

android:exported="true"

android:label="@string/app\_name"

android:theme="@style/Theme.TravelApp">

<intent-filter>

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

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>