

***Wanderlust: A Personalized Travel Planning and  
Tracking App***

**Presented by**

**Team ID: NM2023TMID34969**

**Team Leader: THAJMUL AHAMED**

**Team members:**

**AswinSaba R**

**Kabilan S**

**Mohamed Rifayath R**

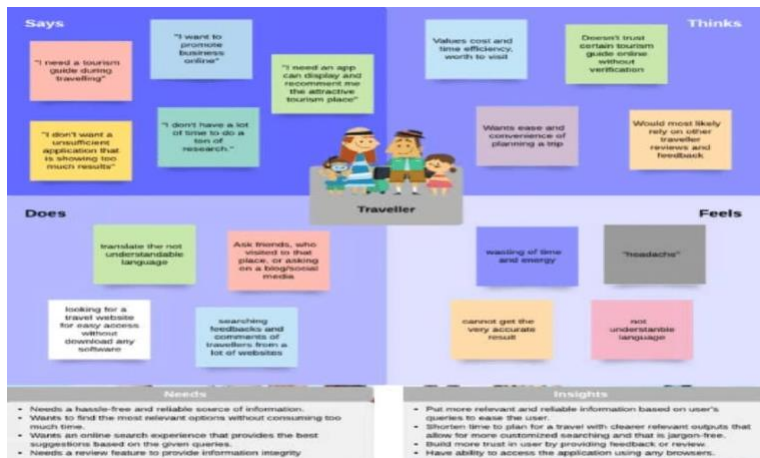
## INTRODUCTION:

Software has changed how people travel, where they travel, when they travel, and even how they share their travels with others. It has also allowed for the democratization of travel, enabling travelers to easily book flights, hotels, and adventure tours. With the addition of new technology, travel has become more comfortable, providing travelers with the opportunity to access the Internet in-flight, book travel accommodations in advance, and see reviews of restaurants and tourist attractions.

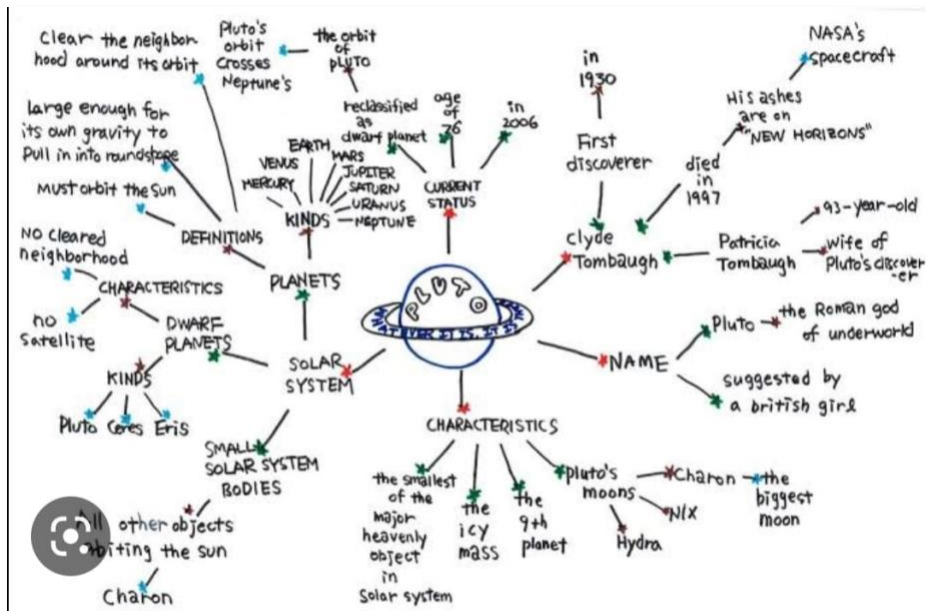
### 1:2 purpose

Travel tracking is used for many purposes. They were initially created to help companies manage the location of their travellers from a duty of care and security perspective. This is still the most common use of travel trackers.

### Empathy Map



## Brainstorm Map



### Advantages:

- Communicate when it matters most.
- Improves Safety and Security.
- Accelerate your Crisis Response.
- Create a Monitoring and Tracking Policy.
- Obtain Consent.
- Use Verified Monitoring Technology.

### Disadvantages:

- Assuming always-on data connection.
- Assuming high speed network access.
- UX design too small and packed.
- Poor data synchronization with the cloud.
- Requesting too many permissions.
- +1 (Android specific bonus) App cannot be moved to SD card.

**Application:**

Travel apps are a source of inspiration and extremely useful for booking and managing logistics — even making restaurant reservations or finding a great fitness class wherever in the world you’re flying to next.

From apps that help with everything from day-of hotel bookings and last-minute flight changes to those that serve as guidebooks and foreign-language dictionaries, travel is infinitely easier thanks to technology.

**Conclusion:**

Tourism Mobile apps help users for planning travel, accommodation bookings, ticket bookings, cab booking, route mapping, and more. However, these are just the basic uses of tourism mobile apps, and more tourist services can be delivered.

**Future scope:**

Technology is continuously evolving and will continue to do so in the coming era. The change in the technology is also impacting the industries like cosmetic industry, health industry, education, entertainment, the travel industry and many more.

**Appendix:**

Begin the process of developing your app with a development team that will provide the design and code for it.

**Source 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?,
```

```
)
```

```
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)
```

```
}
```

```
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
}
}
}
}

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()
    }

    @SuppressWarnings("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
    }

    @SuppressWarnings("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
}

```

```

}

```

```

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,  
    onChange = { username = it },  
    label = { Text("Username") },  
    modifier = Modifier.padding(10.dp)  
        .width(280.dp)  
)
```

```
TextField(  
    Value = password,  
    onChange = { 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)
}
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))
}

}

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."
    )

```

```

    }
}
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-Élysées 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 7<sup>th</sup> 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”



```

    )
}
}
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 “

)

}

}

#### **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>

```







