Virtual Internship Program

Project-1

TRAVEL APP: WANDERLUST

Smart Internz

By,

YASIKA.M(814722104188) -AUTCS22188

VAISHALI.K(814722104179) -AUTCS22174

SWETHA.B(814722104169) -AUTCS221691

SRIMOHANRAJ(814722104158) -AUTCS22158

Aim

To develop a user-friendly **Travel App** in Android Studio using Kotlin that provides seamless navigation, destination details (e.g., Bali and Singapore), and an intuitive interface for user authentication and exploration of curated travel content.

Abstract

The proposed Travel App is a mobile application designed to simplify travel planning and enhance user experiences by providing quick access to curated destination information. Developed in Android Studio with Kotlin, the app includes key features such as a login page, registration functionality, and a visually appealing home screen showcasing destinations like Bali and Singapore.

The app integrates Firebase Authentication for secure user management and Google Places API for dynamic destination details. Users can explore detailed descriptions and images of destinations, navigate through a clean UI, and access travel recommendations. The app is designed with Material Design components, ensuring an intuitive interface.

This project demonstrates the application of modern Android development practices, including the use of Kotlin’s coroutines for smooth asynchronous operations, RecyclerView for dynamic content display, and Glide for image loading. It serves as a practical solution to inspire users to plan their travels conveniently while offering a

scalable foundation for adding advanced features like bookings and personalized itineraries.

Project Overview

Objective:
The Travel App allows users to log in or register, and view curated destination information for places like Bali and Singapore, with features to explore travel packages and itineraries.

Key Features:

1. User Authentication (Login and Registration)
2. Destination Showcase for Bali and Singapore (with images and details)
3. Modular and user-friendly design using Kotlin.

Technologies Used:

- **Programming Language:** Kotlin
- **Backend:** Firebase Authentication (for user management)
- **API Integration:** Google Places API for destination details
- **Libraries:**
    - Glide/Picasso: For loading destination images
    - Retrofit: For API calls
    - Material Design Components: For UI

App Flow

1. Login Page

- Allows users to sign in with email and password.
- Displays error messages for invalid inputs or authentication failures.

Key UI Elements:

- Email and Password fields
- Login button
- "Forgot Password?" and "Register Here" links

Login_activity.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:id="@+id/main"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:background="@drawable/login_background"
   android:padding="16dp"
   tools:context=".loginActivity">

   <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:gravity="center"
      android:orientation="vertical">

      <ImageView
         android:layout_width="160dp"
         android:layout_height="160dp"
```

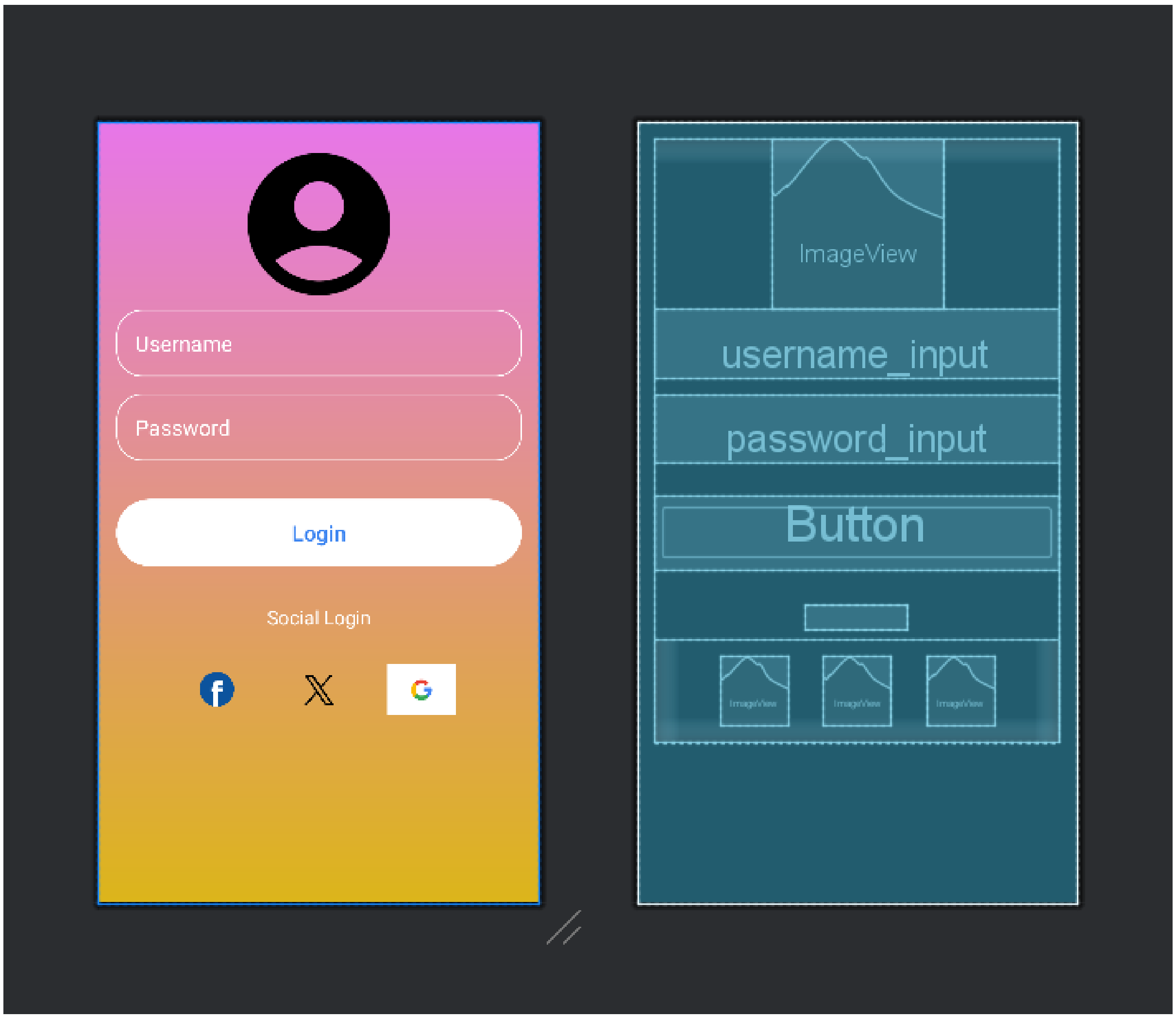
```
    android:src="@drawable/icon_account_circle" />

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Username"
    android:textColor="@color/white"
    android:textColorHint="@color/white"
    android:textSize="20sp"
    android:background="@drawable/rounded_corner"
    android:padding="18dp"
    android:inputType="text"
    android:id="@+id/username_input"/>

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:hint="Password"
    android:textColor="@color/white"
    android:textColorHint="@color/white"
    android:textSize="20sp"
    android:background="@drawable/rounded_corner"
    android:padding="18dp"
    android:inputType="textPassword"
    android:id="@+id/password_input"/>

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Login"
    android:backgroundTint="@color/white"
    android:textColor="#3B84F1"
```

```
        android:padding="18dp"
        android:layout_marginTop="32dp"
        android:textSize="20sp"
        android:id="@+id/login_btn"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Social Login"
        android:textSize="18sp"
        android:textColor="@color/white"
        android:layout_marginTop="32dp"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="8dp"
        android:gravity="center">

        <ImageView
            android:layout_width="64dp"
            android:layout_height="64dp"
            android:layout_margin="16dp"
            android:src="@drawable/facebook"/>

        <ImageView
            android:layout_width="64dp"
            android:layout_height="64dp"
            android:layout_margin="16dp"
            android:src="@drawable/twitter"/>

        <ImageView
```

```
            android:layout_width="64dp"
            android:layout_height="64dp"
            android:layout_margin="16dp"
            android:src="@drawable/google"/>
        </LinearLayout>
    </LinearLayout>

</RelativeLayout>
```

LoginActivity.kt:

```
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import android.util.Log
import android.widget.Toast

class loginActivity : AppCompatActivity() {
    private lateinit var Username: EditText
    private lateinit var password: EditText
    private lateinit var Login: Button

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.login_main)

        Username= findViewById<EditText>(R.id.username_input)
        password = findViewById<EditText>(R.id.password_input)
        Login = findViewById<Button>(R.id.login_btn)
    }
    private fun validation(){
        if(Username.text.isEmpty()) {
            Toast.makeText(this, "Enter Email Id",
Toast.LENGTH_SHORT).show()
        }
        else if(password.text.isEmpty()) {
            Toast.makeText(this,"Enter password",
Toast.LENGTH_SHORT).show()
        }
        else{
            val intent= Intent(this, loginActivity::class.java)
            startActivity(intent)
        }
    }
}
```



2. Registration Page

- Allows new users to create an account with email and password.
- Validates inputs (e.g., matching passwords, valid email format).

Key UI Elements:

- Name, Email, Password, and Confirm Password fields
- Register button

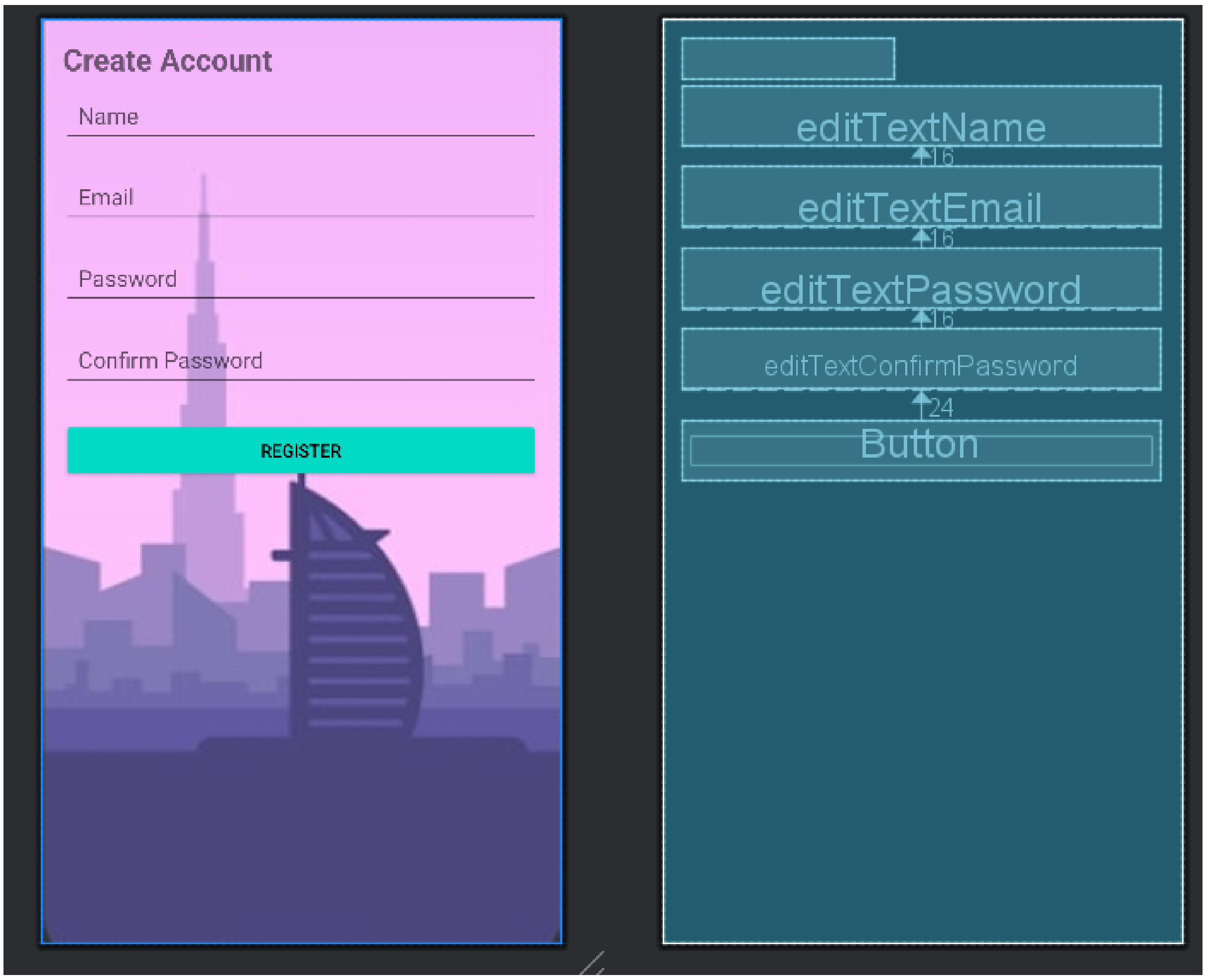
RegisterActivity.kt:

```
package com.example.travelapp

import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity

class RegisterActivity : AppCompatActivity() {
    private lateinit var editTextName: EditText
    private lateinit var editTextEmail: EditText
    private lateinit var editTextPassword: EditText
    private lateinit var editTextConfirmPassword: EditText
    private lateinit var buttonRegister: Button

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_register)

        editTextName = findViewById(R.id.editTextName)
        editTextEmail = findViewById(R.id.editTextEmail)
        editTextPassword = findViewById(R.id.editTextPassword)
        editTextConfirmPassword =
findViewById(R.id.editTextConfirmPassword)
        buttonRegister = findViewById(R.id.buttonRegister)

        buttonRegister.setOnClickListener {
            registerUser()
        }
    }

    private fun registerUser() {
        val name = editTextName.text.toString()
        val email = editTextEmail.text.toString()
        val password = editTextPassword.text.toString()
        val confirmPassword = editTextConfirmPassword.text.toString()

        // Simple validation
        if (name.isEmpty() || email.isEmpty() || password.isEmpty() ||
confirmPassword.isEmpty()) {
            Toast.makeText(this, "Please fill all fields",
Toast.LENGTH_SHORT).show()
            return
        }

        if (password != confirmPassword) {
            Toast.makeText(this, "Passwords do not match",
Toast.LENGTH_SHORT).show()
            return
        }

        // TODO: Add registration logic (e.g., save to database)

        Toast.makeText(this, "Registered successfully!",
Toast.LENGTH_SHORT).show()
        finish() // Close activity after registration
    }
}
```

register_activity.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:background="@drawable/login">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Create Account"
        android:textSize="24sp"
        android:textStyle="bold"
        android:layout_marginBottom="24dp"/>

    <EditText
        android:id="@+id/editTextName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Name"
        android:padding="12dp"
        android:layout_marginTop="36dp"/>

    <EditText
        android:id="@+id/editTextEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:layout_below="@id/editTextName"
        android:padding="12dp"
        android:layout_marginTop="16dp"/>

    <EditText
        android:id="@+id/editTextPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:inputType="textPassword"
        android:layout_below="@id/editTextEmail"
        android:padding="12dp"
        android:layout_marginTop="16dp"/>

    <EditText
        android:id="@+id/editTextConfirmPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Confirm Password"
        android:inputType="textPassword"
        android:layout_below="@id/editTextPassword"
        android:padding="12dp"
        android:layout_marginTop="16dp"/>

    <Button
        android:id="@+id/buttonRegister"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Register"
        android:layout_below="@id/editTextConfirmPassword"
```

```
        android:layout_marginTop="24dp"
        android:textColor="@android:color/black"

android:backgroundTint="@color/design_default_color_secondary"/
>
</RelativeLayout>
```



Home_activity.xml:

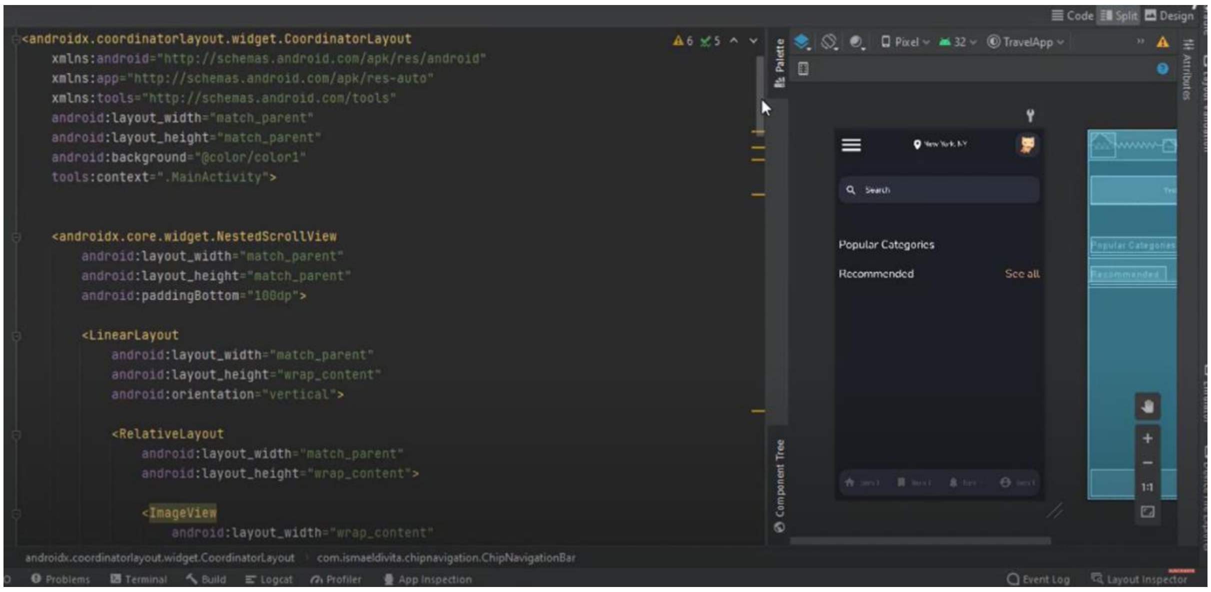
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
android:orientation="vertical"
android:padding="16dp"
android:background="#E3A8ED">

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"
    android:text="Wanderlust"
    android:textAlignment="center"
    android:textColor="@color/design_default_color_primary_dark"
    android:textSize="32sp"
    android:textStyle="bold" />

<EditText
    android:id="@+id/searchBar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Search for destinations..."
    android:padding="12dp"
    android:backgroundTint="@color/black"
    android:layout_marginBottom="16dp"/>

<ImageButton
    android:id="@+id/Image"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:scaleType="centerCrop"
    android:background="@drawable/bali"
    android:contentDescription="Image of the bali">
</ImageButton>
```

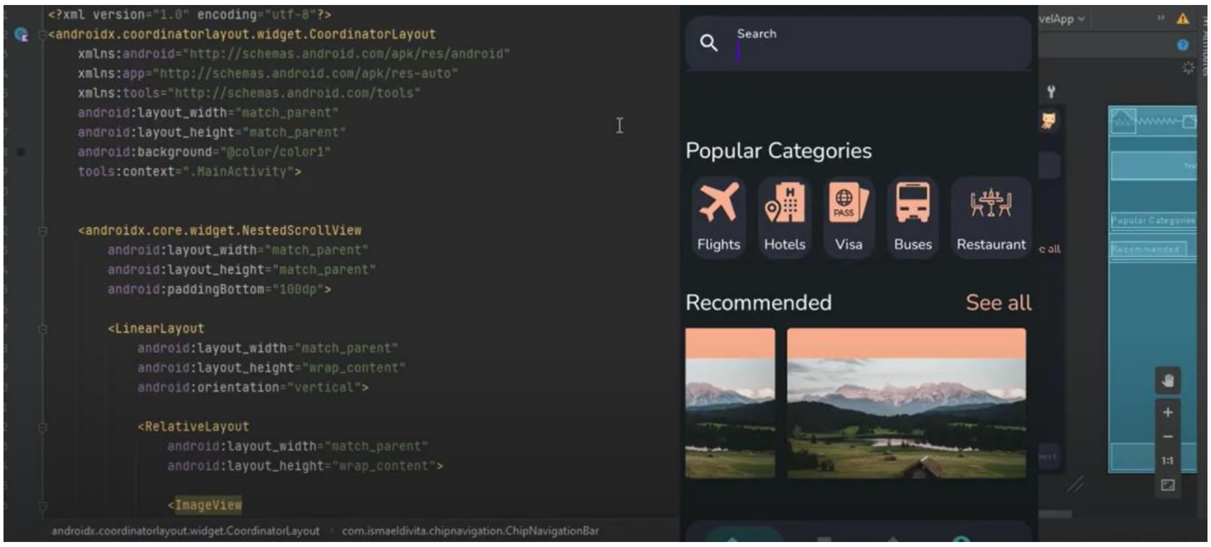
```
    <ImageButton
        android:id="@+id/Image1"
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:scaleType="centerCrop"
        android:background="@drawable/paris1"
        android:contentDescription="Image of the effiel tower">
    </ImageButton>

    <ImageButton
        android:id="@+id/Image2"
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:scaleType="centerCrop"
        android:background="@drawable/singapore"
        android:contentDescription="Image of the singapore">

    </ImageButton>


</LinearLayout>
```

HomeActivity.kt:

```
package com.example.myapplication

import android.content.Intent
import android.os.Bundle
import android.widget.EditText
import android.widget.ImageButton
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {
    private lateinit var searchBar: EditText
    private lateinit var baliImageButton: ImageButton
    private lateinit var parisImageButton: ImageButton
    private lateinit var singaporeImageButton: ImageButton

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.home_activity) // Ensure this points to
your layout file

        // Initialize views
        searchBar = findViewById(R.id.searchBar)
        baliImageButton = findViewById(R.id.Image)
        parisImageButton = findViewById(R.id.Image1)
        singaporeImageButton = findViewById(R.id.Image2)

        // Set click listeners for image buttons
        baliImageButton.setOnClickListener {
            navigateToDestination("Bali", R.drawable.bali) // Ensure the
drawable exists
        }

        parisImageButton.setOnClickListener {
            navigateToDestination("Paris", R.drawable.paris1) // Ensure
the drawable exists
        }

        singaporeImageButton.setOnClickListener {
            navigateToDestination("Singapore", R.drawable.singapore) //
Ensure the drawable exists
        }
    }
    private fun navigateToDestination(destinationName: String,
imageResId: Int) {
        startActivity(intent)
    }

}
```



```
<androidx.coordinatorlayout.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/color1"
    tools:context=".MainActivity">


    <androidx.core.widget.NestedScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:paddingBottom="100dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <RelativeLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content">

                <ImageView
                    android:layout_width="wrap_content"
```



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/color1"
    tools:context=".MainActivity">


    <androidx.core.widget.NestedScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:paddingBottom="100dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <RelativeLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content">

                <ImageView
```

androidx.coordinatorlayout.widget.CoordinatorLayout
com.ismaeldivita.chipnavigation.ChipNavigationBar
Search
Popular Categories
Flights
Hotels
Visa
Buses
Restaurant
Recommended
See all

Best Practices

- Use **Kotlin coroutines** for asynchronous operations (e.g., API calls).
- Apply **Material Design Guidelines** for a polished UI.
- Implement **ViewModel and LiveData** for managing UI-related data.
- Use **Firebase Authentication Rules** for secure data handling.