**PPROJECT REPORT** **Project Report**

1.**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, hotels, and adventure tours. With the addition of new technology, travel has become more comfortable, book travel accommodations in advance, and see reviews of restaurants and tourist attractions.  
  
It is also changing how companies in the travel industry operate, enabling real-time monitoring of inventory, pricing, and demand. It enables travel companies to offer greater value to their customers by collecting data about their customers,

2.**Purpose**:

There is no one-size-fits-all answer to this question, as the purpose of travel in tourism will vary depending on what kind of travel you are. However, some common purposes of travel include exploring a new place, gaining knowledge and understanding about a culture or region, and spending time with friends or family.

There are many different types of travel out there, so it’s important to think about what your specific purpose for travel might be. If you’re looking to relax and enjoy the scenery, then vacationing may be the best type of travel for you. On the other hand, if you’re interested in learning more about a particular subject or cultural experience, then educational tourism may be more appropriate.

Whatever your reasons for traveling – whether it’s recreation or education – make sure to consider all the different options available to you before making your decision. There is no one right way to explore a new destination, learn about a new culture, or spend time with loved ones – and that’s what makes traveling so special!

3.Problem definition and Design thinking:

Empathy map:

Graphical user interface, application

Description automatically generated

**Ideation & Brainstorming map:**

Graphical user interface, application

Description automatically generated

**Result:** Login page:

**Graphical user interface, application, Teams

Description automatically generated**

Register page:

Graphical user interface

Description automatically generated with low confidence

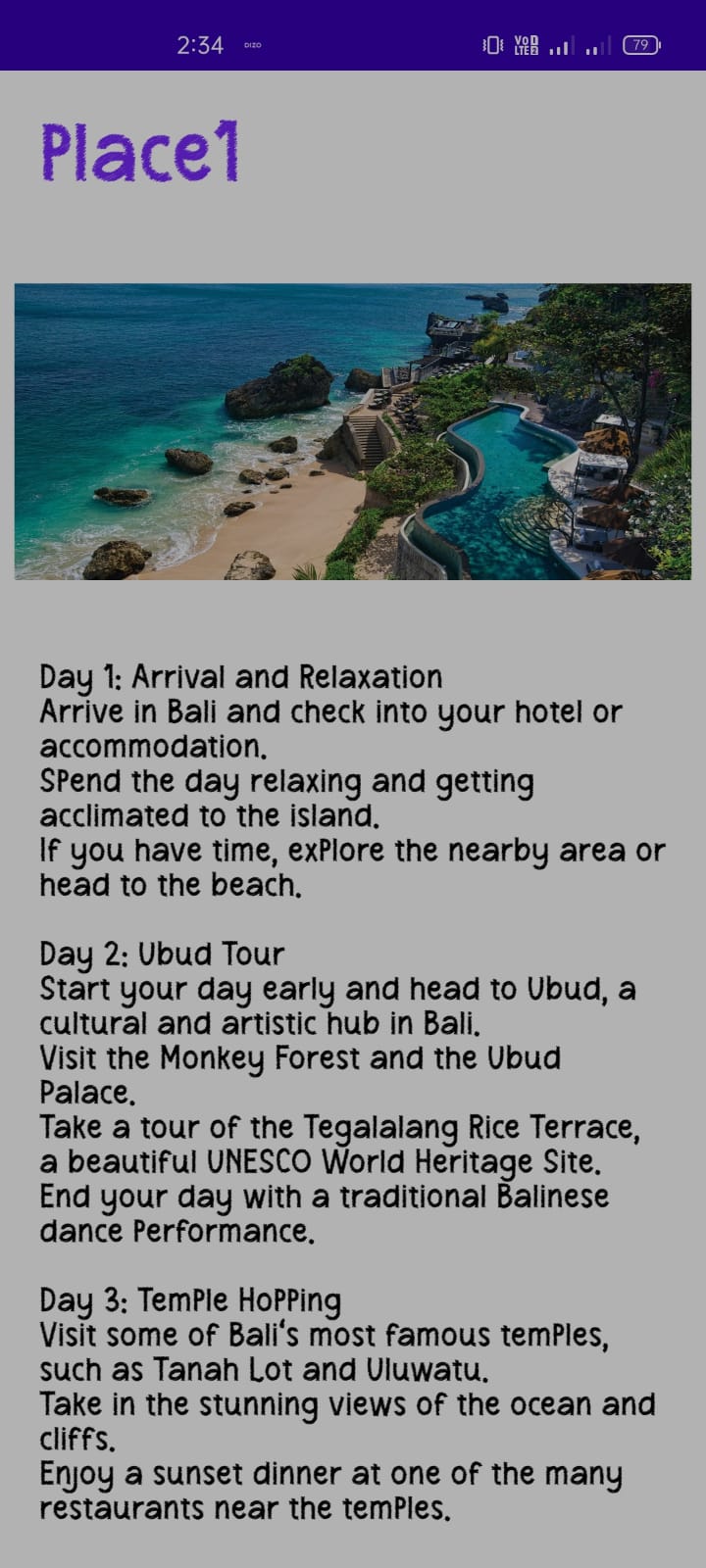
Register page:

Graphical user interface, application

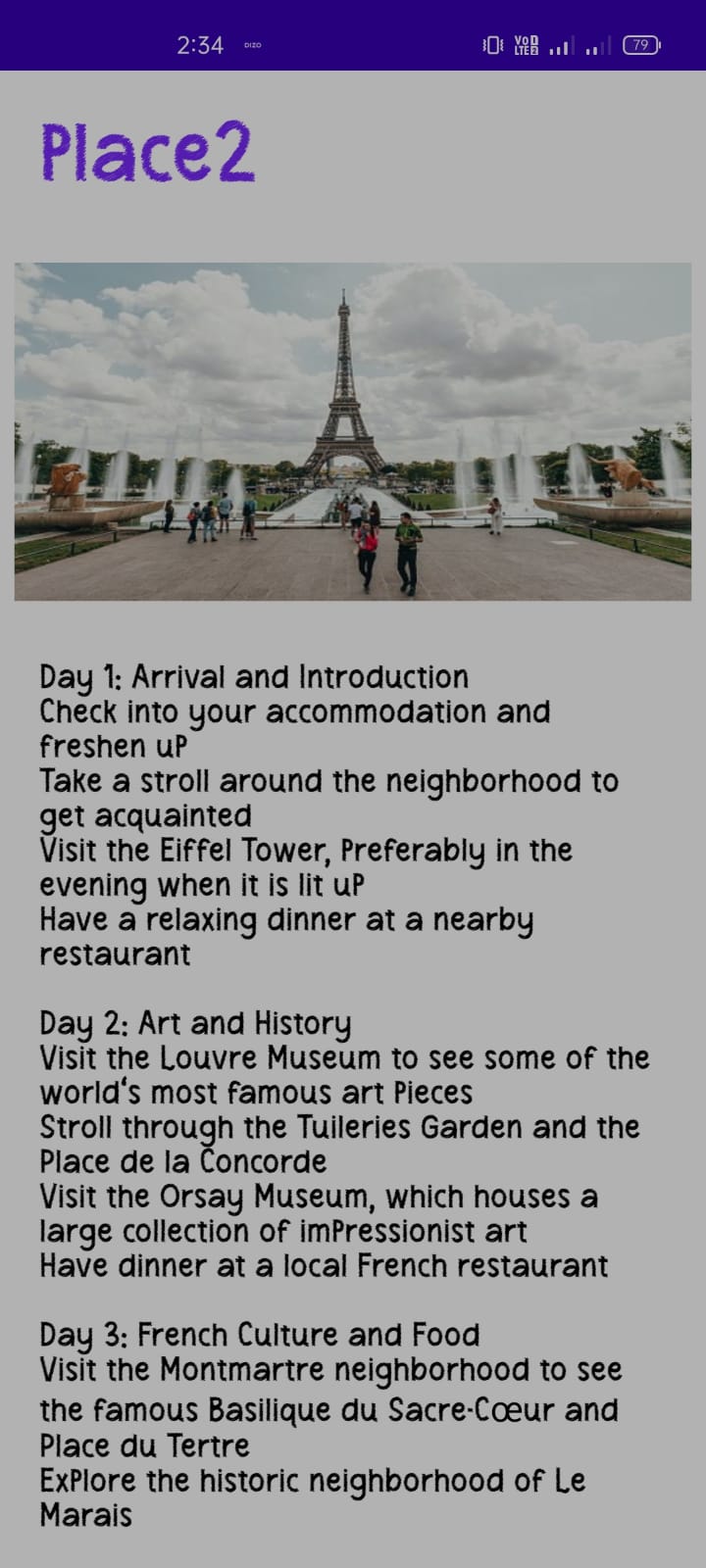
Description automatically generated

Place searching:

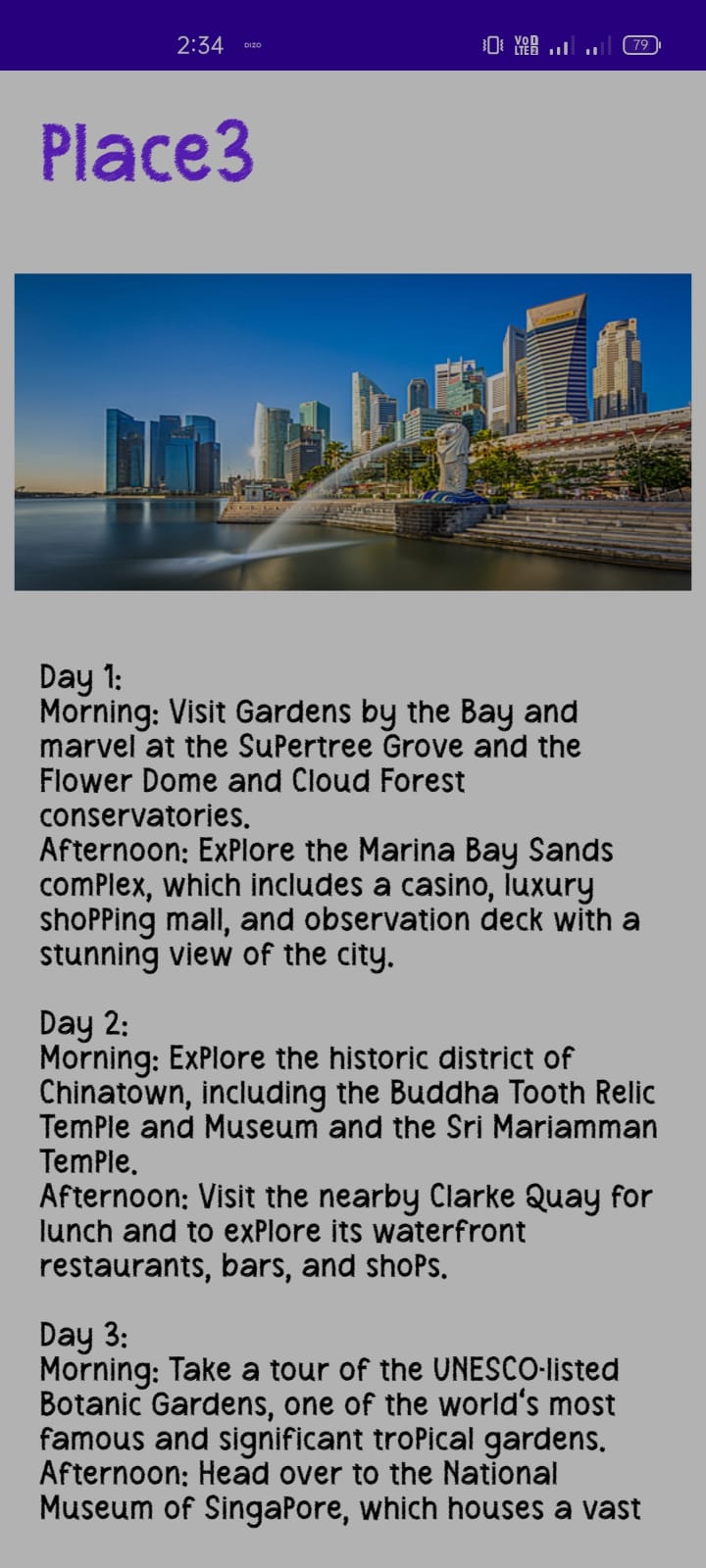
Place 1:



Place 2:



Place 3:



Advantage:

* **Tourism brings in money –** Tourism brings in money because people from all over the world travel to different countries. As we already saw, tourism is a huge industry, which means that it has a lot of benefits, including economic and social ones. It provides jobs in hotels, restaurants, and other tourism-related industries. It also helps sustain services so that tourists can enjoy their vacations even more.
* **Tourism provides jobs –** Tourism can provide a significant source of income for local communities. As a result, tourism often generates jobs in various fields like the hospitality and agriculture industries. Tourism also helps to improve the housing market, which will lead to economic growth and job opportunities in other industries

Disadvantage:

* Can cause environmental damage – There are many disadvantages to tourism, such as the effects on the environment. Tourists will often cause damage during their visit in an attempt to find a photo opportunity. The result is that tourists are often responsible for deforestation and over-exploitation of natural resources. Destroying habitats could lead to extinction of endangered species, which could have adverse consequences on biodiversity.
* Commercialization – Developed countries continue to grow in popularity and can become very crowded. This has led to changes in people’s attitudes and behaviours, including the development of commercialism. This new form of capitalism has had a direct impact on how tourism is conducted. A common example is the growing prevalence of high-priced souvenirs that are sold by tour guides around major tourist destinations.

Conclusion:

we can say that**tourism is a very productive activity** both for the tourist and the government. As they support each other simultaneously. Also, the government should consider improving the conditions of the country as more and more number of tourist visit their country.

**Future scope:**

IN future add rating for each place safety for customers understanding. Provide full support to the customers via artificial intelligence. We provide all information to customer via argument reality technology.

Appendix:

**Gradle scripts > build.gradle(Module :app)**

plugins **{** id 'com.android.application'  
 id 'org.jetbrains.kotlin.android'  
**}**android **{** namespace 'com.example.travelapp'  
 compileSdk 33  
  
 defaultConfig **{** applicationId "com.example.travelapp"  
 minSdk 21  
 targetSdk 33  
 versionCode 1  
 versionName "1.0"  
  
 testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"  
 vectorDrawables **{** useSupportLibrary true  
 **}  
 }** buildTypes **{** release **{** minifyEnabled false  
 proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'  
 **}  
 }** compileOptions **{** sourceCompatibility JavaVersion.*VERSION\_1\_8* targetCompatibility JavaVersion.*VERSION\_1\_8* **}** kotlinOptions **{** jvmTarget = '1.8'  
 **}** buildFeatures **{** compose true  
 **}** composeOptions **{** kotlinCompilerExtensionVersion '1.2.0'  
 **}** packagingOptions **{** resources **{** excludes += '/META-INF/{AL2.0,LGPL2.1}'  
 **}  
 }  
}**dependencies **{** implementation 'androidx.core:core-ktx:1.7.0'  
 implementation 'androidx.lifecycle:lifecycle-runtime-ktx:2.3.1'  
 implementation 'androidx.activity:activity-compose:1.3.1'  
 implementation "androidx.compose.ui:ui:$compose\_ui\_version"  
 implementation "androidx.compose.ui:ui-tooling-preview:$compose\_ui\_version"  
 implementation 'androidx.compose.material:material:1.2.0'  
 testImplementation 'junit:junit:4.13.2'  
 androidTestImplementation 'androidx.test.ext:junit:1.1.5'  
 androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'  
 androidTestImplementation "androidx.compose.ui:ui-test-junit4:$compose\_ui\_version"  
 debugImplementation "androidx.compose.ui:ui-tooling:$compose\_ui\_version"  
 debugImplementation "androidx.compose.ui:ui-test-manifest:$compose\_ui\_version"  
  
  
 // Adding Room dependencies  
 implementation 'androidx.room:room-common:2.5.0'  
 implementation 'androidx.room:room-ktx:2.5.0'  
**}**

**User data Create class:**

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?,  
  
 )

**Create an UserDao interface:**

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)  
}

**Create an UserDatabase class:**

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  
 **}** }  
 }  
}

**Creating LoginActivity.kt with database:**

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)  
}

**Creating RegisterActivity.kt with database:**

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)  
}

### Creating MainActivity.Kt File:

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*))  
 **}** }  
}