

WANDERLUST: A PERSONALIZED TRACKING PLANNING AND TRACKING APP

PROJECT PRESENTED BY:

TEAM ID: NM2023TMID10883

TEAM SIZE: 4

TEAM LEADER: KEERTHANA T

TEAM MEMBER:

❖ KAVIYA A

❖ KAVYA S

❖ KOPPERUMDEVI M



Edit with WPS Office

1. INTRODUCTION

1.1 OVERVIEW

A project built using the Android Compose UI toolkit.

It demonstrates how to create a simple travel app using the Compose libraries

It also features a personalized feed of recommended

Accommodations based on the locations.

1.2 PURPOSE

► Ease your travel worries, and relax and enjoy every moment of your trip – without worrying about a thing.



Edit with WPS Office

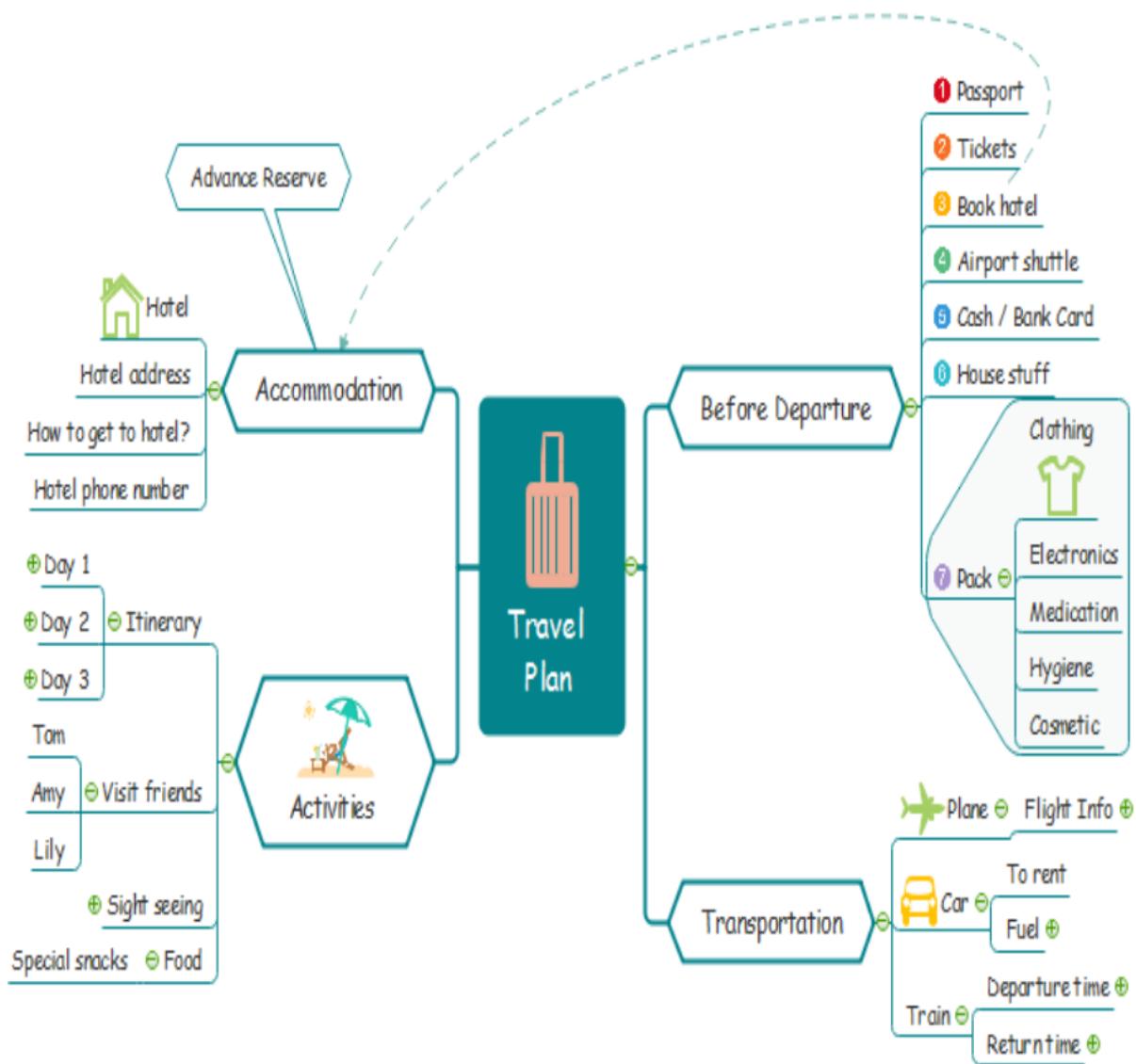
►This is the perfect tool for anyone who wants to have a worry-free vacation.

2.PROBLEM DEFINITION & DESIGN THINKING

2.1 EMPATHY MAP



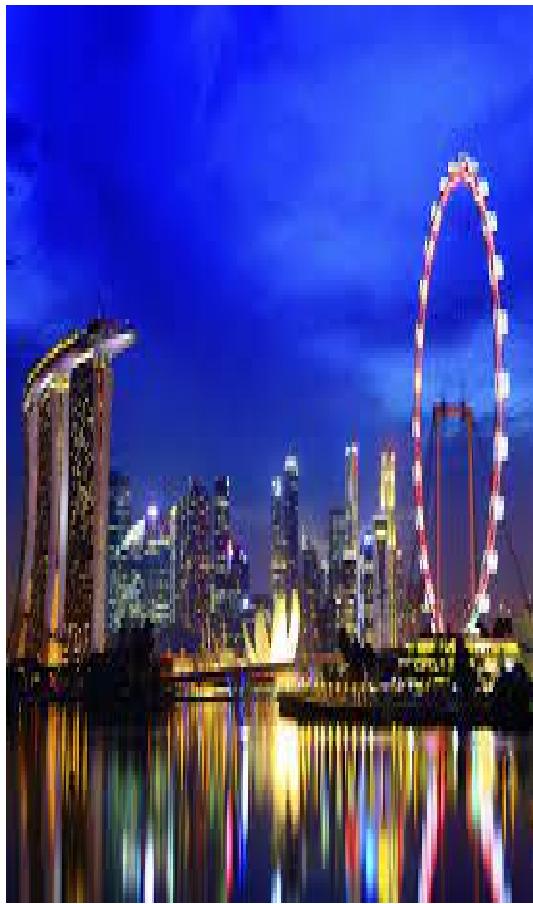
Edit with WPS Office



2.2 IDEATION & BRAINSTORMING MAP



Edit with WPS Office



TRAVEL HACKS

01

ACCOMMODATION

Sign up for a loyalty program to get discounts on accommodation. Book in advance.

02

SIGHTSEEING

Sign up for a free walking tour and pay only if you like the tour. The amount of tips is up to you.



04

SOUVENIRS

Visit only crafting stores or flea markets if you hunt for cheap but authentic stuff-to-take-home.

03

FOOD

Talk to locals about their favorite spots to eat - they are normally cheaper and have tastier food.

3.RESULT



Edit with WPS Office

Thus, the above project is successfully executed and the output is verified.

4. ADVANTAGES & DISADVANTAGES

ADVANTAGES

- *Breaking from this routine helps you reset and relax.*
- *This can improve your problem-solving skills and make you a*



Edit with WPS Office

creative thinker

- *Getting out of your comfort zone and experiencing*
- *new things mean you have to adapt and be resourceful*

DISADVANTAGES

- *Environmental disadvantages of travelling*
- *Long flight and road trips contribute to air pollution caused by transportation*
- *Tourism contributes also to plastic pollution*
- *Pollution most of the plastics used in tourism is meant to be used only once and then be thrown away.*

5.APPLICATIONS



Edit with WPS Office

A travel planning app is an application for “booking travel reservation, tracking loyalty points, and browsing travel packages”. A travel planning software packages can be used by travel agencies, travel suppliers, and consumers

6.CONCLUSION

Having done this project for a hackathon, time was naturally very tight and as a team we had very little time to brainstorm and truly iron out our idea before we need to begin research

Compromised-having a strong foundation and understanding of the problem space is critical to delivering a problem solving and user centric solution.

7.FUTURE SCOPE

VIRTUAL REALITY: The app could incorporate virtual reality, allowing users to see



Edit with WPS Office

a 3D view of their travel destination before they arrive, making it easier for them to plan their itinerary

8.APPENDIX

A. SOURCES 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"
```



Edit with WPS Office

```
    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>
```

Color.kt

```
package com.example.travelapp.ui.theme

import androidx.compose.ui.graphics.Color

val Purple200 = Color(0xFFBB86FC)
val Purple500 = Color(0xFF6200EE)
val Purple700 = Color(0xFF3700B3)
val Teal200 = Color(0xFF03DAC5)
```

Shape.kt

```
package com.example.travelapp.ui.theme
```



Edit with WPS Office

```
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material.Shapes
import androidx.compose.ui.unit.dp

val Shapes = Shapes(
    small = RoundedCornerShape(4.dp),
    medium = RoundedCornerShape(4.dp),
    large = RoundedCornerShape(0.dp)
)
```

Theme.kt

```
package com.example.travelapp.ui.theme

import androidx.compose.foundation.isSystemInDarkTheme
import androidx.compose.material.MaterialTheme
import androidx.compose.material.darkColors
import androidx.compose.material.lightColors
import androidx.compose.runtime.Composable

private val DarkColorPalette = darkColors(
    primary = Purple200,
    primaryVariant = Purple700,
    secondary = Teal200
)

private val LightColorPalette = lightColors(
```



Edit with WPS Office

```

primary = Purple500,
primaryVariant = Purple700,
secondary = Teal200

/* Other default colors to override
background = Color.White,
surface = Color.White,
onPrimary = Color.White,
onSecondary = Color.Black,
onBackground = Color.Black,
onSurface = Color.Black,
*/
)

@Composable
fun TravelAppTheme(darkTheme: Boolean = isSystemInDarkTheme(), content: @Composable () -> Unit) {
    val colors = if (darkTheme) {
        DarkColorPalette
    } else {
        LightColorPalette
    }
}

MaterialTheme(
    colors = colors,
    typography = Typography,

```



Edit with WPS Office

```
    shapes = Shapes,  
    content = content  
)  
}
```

Type.kt

```
package com.example.travelapp.ui.theme  
  
import androidx.compose.material.Typography  
import androidx.compose.ui.text.TextStyle  
import androidx.compose.ui.text.font.FontFamily  
import androidx.compose.ui.text.font.FontWeight  
import androidx.compose.ui.unit.sp  
  
// Set of Material typography styles to start with  
val Typography = Typography(  
    body1 = TextStyle(  
        fontFamily = FontFamily.Default,  
        fontWeight = FontWeight.Normal,  
        fontSize = 16.sp  
)  
  
/* Other default text styles to override  
button = TextStyle(  
    fontFamily = FontFamily.Default,  
    fontWeight = FontWeight.W500,
```



Edit with WPS Office

```
    fontSize = 14.sp  
),  
caption = TextStyle(  
    fontFamily = FontFamily.Default,  
    fontWeight = FontWeight.Normal,  
    fontSize = 12.sp  
)  
*/  
)
```

BailActivity.kt

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.remember  
  
import androidx.compose.foundation.verticalSc
```

```
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?) {
```



Edit with WPS Office

```
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() {
```



Edit with WPS Office

```
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.bal), contentDescription = "",
        modifier = Modifier
            .padding(16.dp)
            .fillMaxWidth()
            .height(200.dp)
            .scale(scaleX = 1.2F, scaleY = 1F)
    )
}
```



Edit with WPS Office

)

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" +



Edit with WPS Office

"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" +



Edit with WPS Office

```
"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."  
)  
}  
}
```

LoginActivity.kt

```
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.*
```



Edit with WPS Office

```
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)  
  
        }  
  
    }  
  
}
```



Edit with WPS Office

```
@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,
```



Edit with WPS Office

```
    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 = {
```



Edit with WPS Office

```
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 {
```



Edit with WPS Office

```

    TextButton(onClick = {context.startActivity(
        Intent(
            context,
            RegisterActivity::class.java
        )
    )}
)
{
    Text(text = "Register")
}

TextButton(onClick = {
})
{
}

Spacer(modifier = Modifier.width(60.dp))

Text(text = "Forgot password?")
}
}

private fun startMainPage(context: Context) {
    val intent = Intent(context, MainActivity::class.java)
    ContextCompat.startActivity(context, intent, null)
}

```

MainActivity.kt

android.content.Context



Edit with WPS Office

```
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
```



Edit with WPS Office

```
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
```



Edit with WPS Office

```
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))  
    }  
}
```



Edit with WPS Office

```
// 01

Card(
```

modifier = Modifier

```
.fillMaxWidth()
```

.height(250.dp)

```
.clickable {
```

context.startActivity(

```
Intent(context, BaliActivity::class.java)
```

```
)
```

```
,
```

elevation = 8.dp

```
)
```

```
{
```

Column(

```
horizontalAlignment = Alignment.CenterHorizontally
```

```
) {
```



Edit with WPS Office

`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,
```



Edit with WPS Office

```
        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()
```



Edit with WPS Office

```
.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)
)
```



Edit with WPS Office

```
.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(
```



Edit with WPS Office

```
        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)
```



Edit with WPS Office

```
)  
  
},  
  
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),
```



Edit with WPS Office

```
        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  
    )  
}
```



Edit with WPS Office

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

ParisActivity.kt

```
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
```



Edit with WPS Office

```
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 {

```



Edit with WPS Office

```
TravelAppTheme {  
    // A surface container using the 'background' color from the theme  
  
    Surface(  
        modifier = Modifier.fillMaxSize(),  
  
        color = MaterialTheme.colors.background  
    ) {  
  
        Greeting()  
  
    }  
  
}  
  
}  
  
}  
  
}  
  
}  
  
@Composable  
  
fun Greeting() {  
  
    Column(  
        ...  
    ) {  
  
        Text("Hello, world!")  
  
        Text("This is a sample application")  
  
        Text("...")  
  
    }  
}
```



Edit with WPS Office

```
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)
)
```



Edit with WPS Office

)

Textl

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" +



Edit with WPS Office

"\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" +



Edit with WPS Office

"\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" +



Edit with WPS Office

```
"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"  
}  
}  
}
```

RegisterActivity.kt

```
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
```



Edit with WPS Office

```
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) {
```



Edit with WPS Office

```
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 },
```



Edit with WPS Office

```
        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)  
)
```



Edit with WPS Office

```
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(

```



Edit with WPS Office

```
        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  
        )  
    })  
}
```



Edit with WPS Office

```
        )
    )
})

{
    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)
}
SingaporeActivity.kt
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
```



Edit with WPS Office

```
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
```



Edit with WPS Office

```
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()

                }
            }
        }
    }
}
```



Edit with WPS Office

```
@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 = "",
```



Edit with WPS Office

```
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" +



Edit with WPS Office

"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" +



Edit with WPS Office

"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 "

)

}

}

User.kt

package com.example.travelapp



Edit with WPS Office

```
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.kt

```
package com.example.travelapp
```

```
import androidx.room.*
```



Edit with WPS Office

```
@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)
}
```

UserDatabase.kt

```
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)
```



Edit with WPS Office

```
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  
            }  
        }  
    }  
}
```

UserDatabaseHelper.kt

```
package com.example.travelapp
```



Edit with WPS Office

```
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 (" +
```



Edit with WPS Office

```

"$COLUMN_ID INTEGER PRIMARY KEY AUTOINCREMENT, " +
"$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()
}

```



Edit with WPS Office

```

@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 getUserId(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

```



Edit with WPS Office

```

    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)),
            )
            users.add(user)
        } while (cursor.moveToNext())
    }
    cursor.close()
    return users
}

```



Edit with WPS Office

```
        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  
  
}  
  
}
```



Edit with WPS Office



Edit with WPS Office