NAAN MUDHALVAN SCHEME - 2023

Wanderlust: A Personalized Travel Planning and Tracking App

SUBMITTED BY

TEAM LEADER:

T.MOHANRAJA – ASMSU2022010814 (NAANMUDHALVAN ID)

TEAM MEMBER:

C.MATHAN– ASMSU2022010811 (NAANMUDHALVAN ID)

S.MICHAEL FERNANDO – ASMSU2022010812 (NAANMUDHALVAN ID)

U.MOHAMEDTHAMEEM ANSARI – ASMSU2022010813 (NAANMUDHALVAN ID)

DEPARTMENT OF COMPUTER SCIENCE

ADITANAR COLLEGE OF ARTS AND SCIENCE

VIRAPANDIANPATNAM – 628216

Project Report Template

1. INTRODUCTION

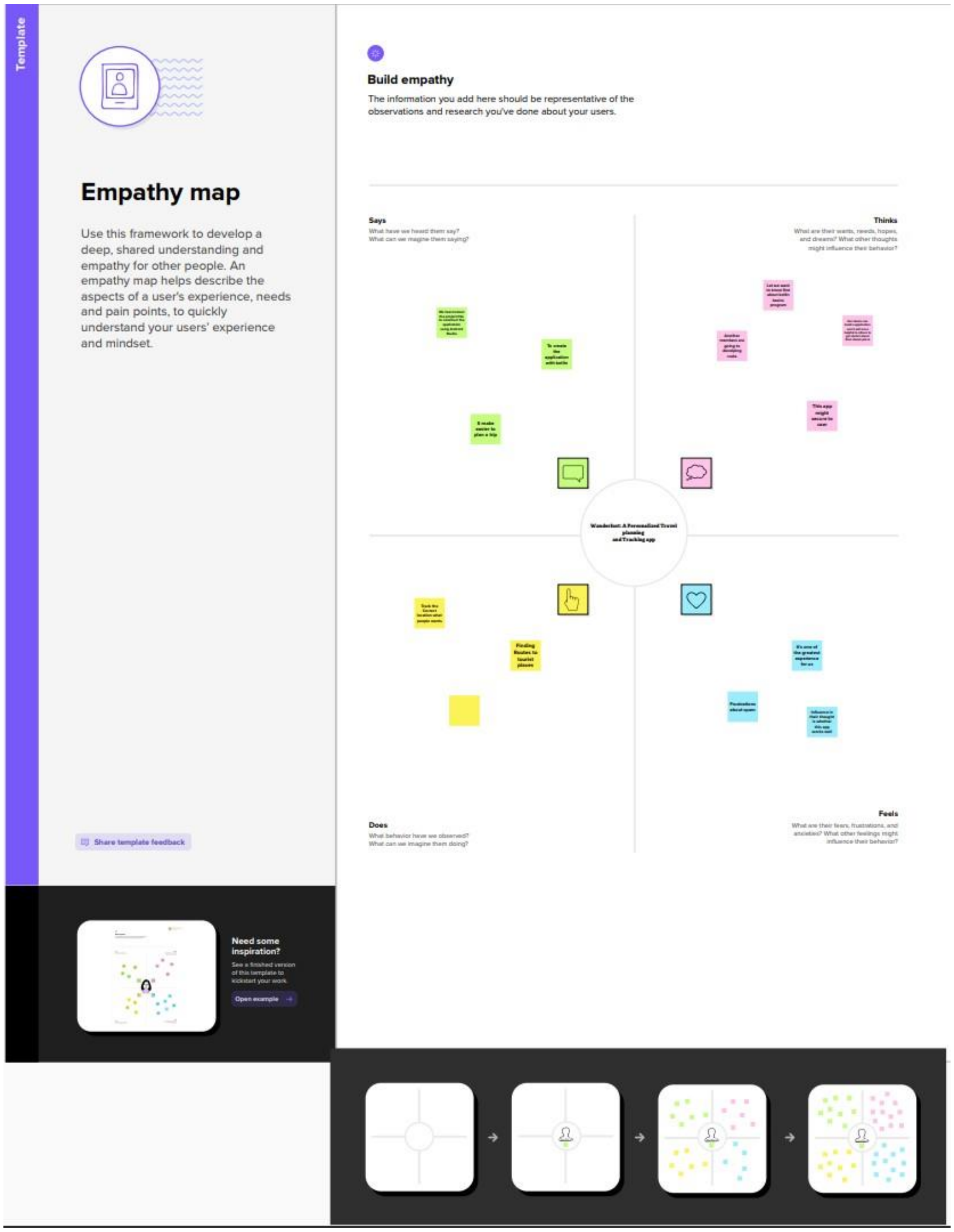
1.1)Overview :-

In simple words, Travel is the moving of objects or people between two geographical locations of relative distance. At times Introduction to Travel can also include movements between different destinations with short stop over's on the way to the final destination to be reached. By the way, Today's Adventurers and Nature lovers are undertake travel in search for new grounds for adventure activities and nature surrounding environments. A lot of Travel is usually undertaken as tourists who look for leisure, vacations and holidays. This usually acts as the base for boosting up the Tourism industry. According to this generation travel. People must need an application or websites to search about their travelling places to gain information about the new places where the people not gone to the places before. So that the application was created.

1.2)Purpose :-

- Travelling is used to feel the Emotions of the World.
- Modern life tends to be busy and pressured with little time for yourself.
- Can experience a new places information that you have not gone as before
- Many tourism trips are taken where the main purpose is relaxation, rest and enjoyment.
- Travelling can improve your communications skills.
- Travelling can help you make Memorable Memories.

2. 1)Empathy Map:-



Template

Empathy map

Use this framework to develop a deep, shared understanding and empathy for other people. An empathy map helps describe the aspects of a user's experience, needs and pain points, to quickly understand your users' experience and mindset.

Share template feedback

Need some inspiration?

See a finished version of this template to kickstart your work.

Open example

Build empathy

The information you add here should be representative of the observations and research you've done about your users.

Says

What have we heard them say?
What can we imagine them saying?

Thinks

What are their wants, needs, hopes, and dreams? What other thoughts might influence their behavior?

Does

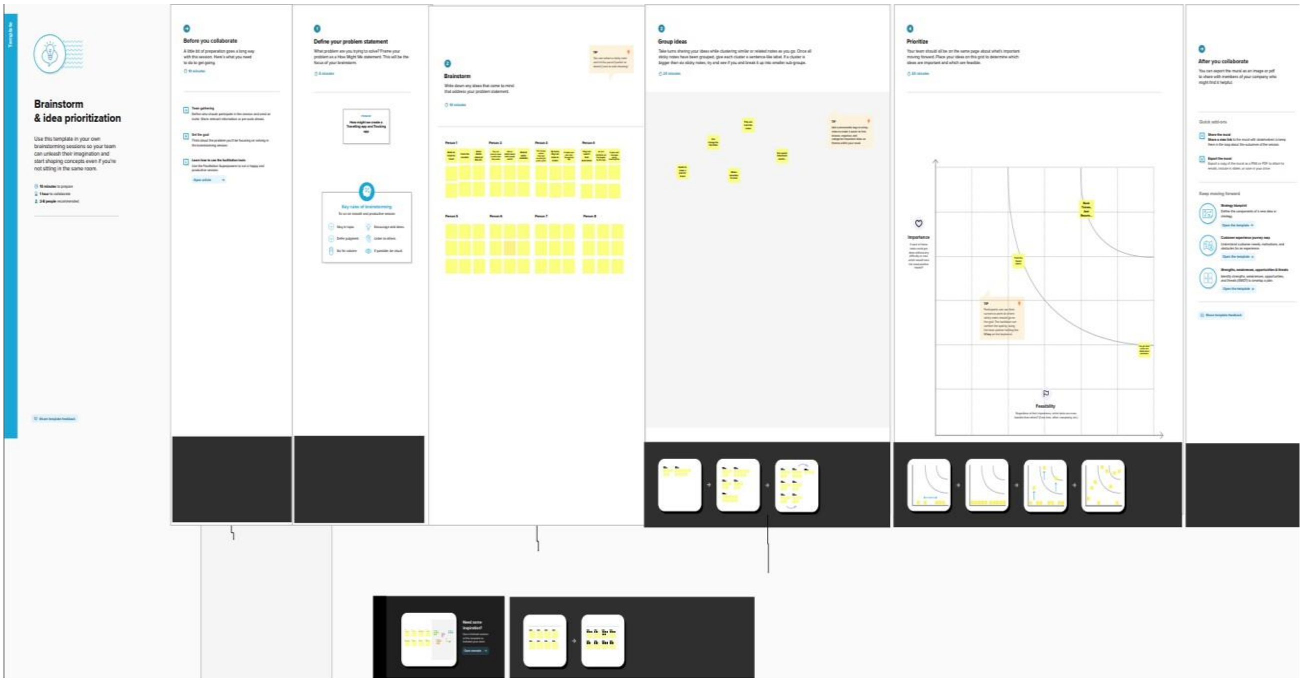
What behavior have we observed?
What can we imagine them doing?

Feels

What are their fears, frustrations, and anxieties? What other feelings might influence their behavior?

Wanderlust: A Personalized Travel planning
and Tracking app
[illegible]
To create the application with kotlin
It make easier to plan a trip
[illegible]
Another members are going to developing code
[illegible]
This app might secure to user
Track the Current location when people wants
Finding Routes to tourist places
It's one of the greatest experience for us
Frustrations about spam
Influence in their thought is whether this app works well

2.2)Brainstorming Map:-



Brainstorm
& idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

How might we create a Travelling app and Tracking app

Key rules of brainstorming

Brainstorm

Write down any ideas that come to mind that address your problem statement.

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

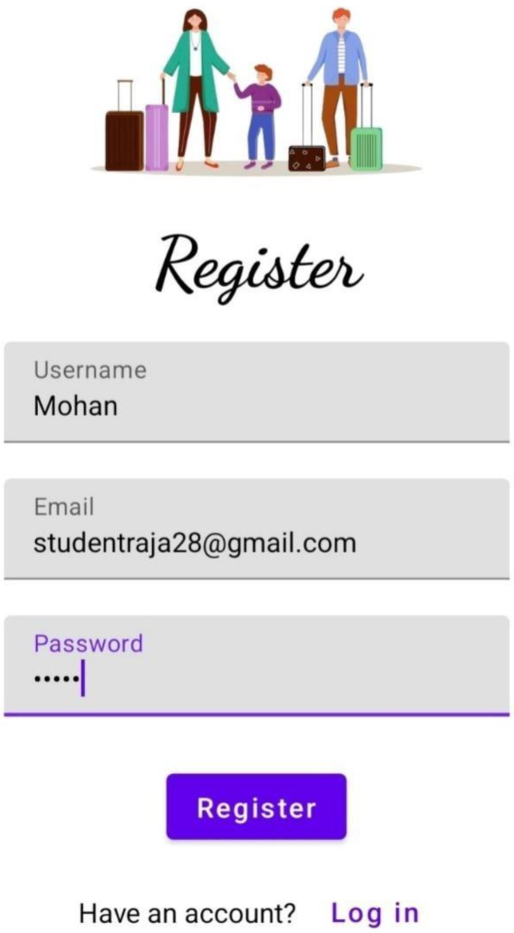
Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

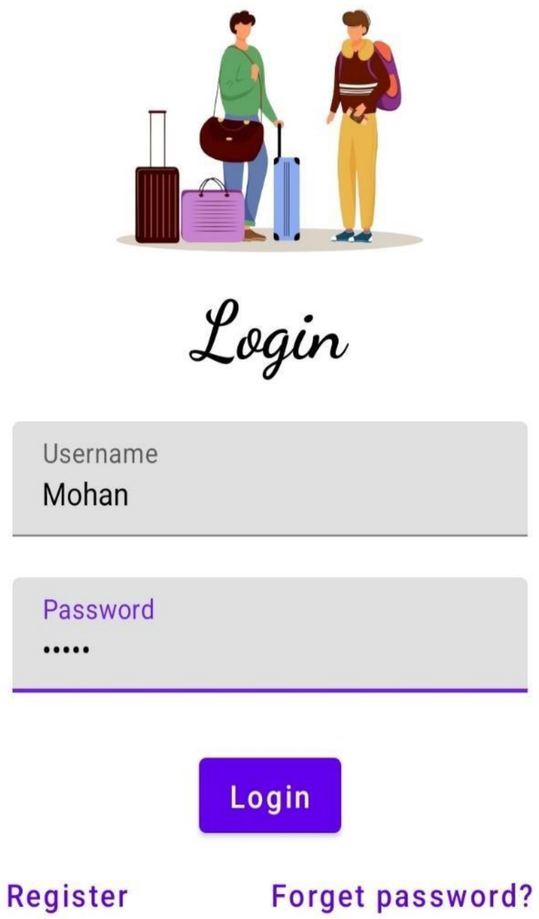
3.Result :-



Username
Mohan

Email
studentraja28@gmail.com

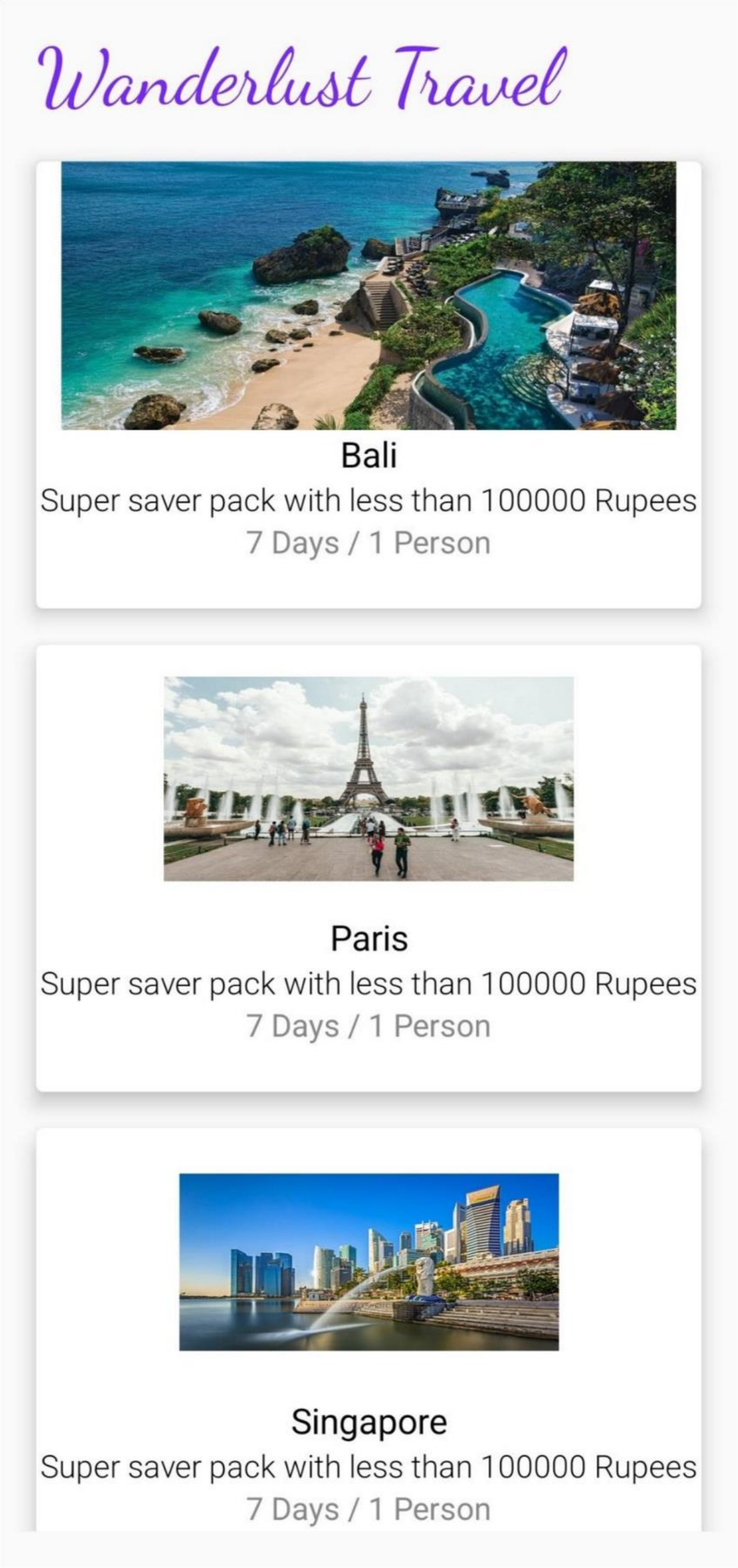
Password
•••••



Login

Username
Mohan

Password
•••••



Wanderlust Travel

Bali

Super saver pack with less than 100000 Rupees

7 Days / 1 Person

Paris

Super saver pack with less than 100000 Rupees

7 Days / 1 Person

Singapore

Super saver pack with less than 100000 Rupees

7 Days / 1 Person

4.ADVANTAGES:-

1. Breaking from this routine helps you **reset and relax**

2. Getting out of your comfort zone and you can gain new information about new places through this app

DISADVANTAGES:-

- Unexpected **costs and emergencies.**

- Not being able to exercise regularly.

5. Application:-

This section should include **a clear and brief description about your Travelling procedures**. One important purpose of this section is to convince the customer that your work is valid. Here, customers can get information about their dream places.

6. Conclusion :-

If an app is being built primarily for the Android platform, then using Travelling app is an easy choice. Because of

Google's widespread adoption, any app based on Travelling Principles is going to feel you're your native app.

7. Future Scope:-

After introduction of Further Travel planning Apps, It become one of the most wanted App. The Android OS is capable of execution of High-Quality. The Enhancement of future scope will made efficient booking services and apps that will enable the tourists to create versatile itineraries with reasonable price. The Mobile Application can be accessed virtually from Remote Place with no requirement of additional Hardware which result in faster downloading and good user interface along with increase in number of user. The Mobile App are on the Best way to move forward as the 90% of Travelling Information application very efficiently. Travelling app will suggest good worldwide places for such tour circumstances; they can search a places from any part of the Globe.

8.Appendix

Source code :-

https://github.com/Mohanraja07/Wanderlust-A-Personalized-Travel-Planning-and-Tracking-App

CODE :-

User.kt

```
package com.example.travelapp
 import androidx.room.ColumnInfo
import androidx.room.Entity import
androidx.room.PrimaryKey

@Entity(tableName = "user_table") data
class User(
@PrimaryKey(autoGenerate = true) val id: Int?,
@ColumnInfo(name = "first_name") val firstName: String?,
@ColumnInfo(name = "last_name") val lastName: String?,
@ColumnInfo(name = "email") val email: String?,
@ColumnInfo(name = "password") val password: String?,

)
```

UserDao.kt

```
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)
```

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) 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
 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?.exec
TABLE IF EXISTS $TABLE_NAME")          onCreate(db)
    }  fun insertUser(user: User) { val db =
writableDatabase val values = ContentValues()
        values.put(COLUMN_FIRST_NAME, user.firstName)
values.put(COLUMN_LAST_NAME, user.lastName)         values.put(COLUMN_EMAIL,
user.email)         values.put(COLUMN_PASSWORD, user.password)
db.insert(TABLE_NAME, null, values)         db.close()
    }

@SuppressLint("Range")
fun getUserByUsername(username: String): User? { val db = readableDatabase
```

```
val cursor: Cursor = db.rawQuery("      SELECT * FROM    $TABLE_NAME          WHERE          $C
?", arrayOf(username)) var
user: User? = null                      {                                IRST_NA
        if (cursor.moveToFirst())                                        ST_NAME )),
user = User( id =                       nIndex(COLUMN_                   SSWORD)
cursor.getInt(cursor.getColum           r.getColumnInd   ID)), ex(COLUMN_F
firstName = cursor.getString(curso      .getColumnInde   x(COLUMN_LA                     MI
lastName = cursor.getString(cursor      tColumnIndex(C   OLUMN_EMAIL
email = cursor.getString(cursor.ge      .getColumnInde   x(COLUMN_PA                     )
password = cursor.getString(cursor )
        }
        cursor.close()
db.close() return user
    }
@SuppressLint("Range")
fun getUserById(id: Int): User? { val
db = readableDatabase val cursor:
Cursor = db.rawQuery("
arrayOf(id.toString())) var user:
User? = null
        if (cursor.moveToFirst())       SELECT * FROM                    WHERE
user = User(
                                        id = {                $TABLE_NAME                        $(
   cursor.getInt(cursor.getColumnIndex(
                                 firstName =          COLUMN_ID
cursor.getString(cursor.getColumnIndex(
                                  lastName =
                                                                )),
cursor.getString(cursor.getColumnIndex(                                           COLUMN_FIRST_NAME
                                     email =           COLUMN_LAST_NAME                              )
cursor.getString(cursor.getColumnIndex(              COLUMN_EMAIL          )),
password = cursor.getString(cursor.getColumnIndex(COLUMN_PASSWORD)),
)
        }
        cursor.close()
db.close() return user
    }


@SuppressLint("Range") fun
getAllUsers(): List<User> { val
users = mutableListOf<User>() val
db = readableDatabase
val cursor: Cursor = db.rawQuery("SELECT * FROM $TABLE_NAME", null)
if (cursor.moveToFirst()) { do {
val user = User(
id = cursor.getInt(cursor.getColumnIndex(COLUMN_ID)),
firstName = cursor.getString(cursor.getColumnIndex(COLUMN_FIRST_NAME)),
lastName = cursor.getString(cursor.getColumnIndex(COLUMN_LAST_NAME)),
email = cursor.getString(cursor.getColumnIndex(COLUMN_EMAIL)), password
= cursor.getString(cursor.getColumnIndex(COLUMN_PASSWORD)),
)
                users.add(user)
            } while (cursor.moveToNext())
        }
        cursor.close()
db.close() return users
    }


}
```

LoginActivity.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 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)
}
```

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 import
androidx.compose.ui.Modifier import
androidx.compose.ui.graphics.Color import
androidx.compose.ui.layout.ContentScale import
androidx.compose.ui.res.painterResource import
androidx.compose.ui.text.font.FontFamily
import
androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.input.PasswordVisualTransformation
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp import
androidx.compose.ui.unit.sp
import androidx.core.content.ContextCompat
```

```
class RegisterActivity : ComponentActivity() { private
lateinit var databaseHelper: UserDatabaseHelper override fun
onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
        databaseHelper = UserDatabaseHelper(this)
setContent {
RegistrationScreen(this, databaseHelper) }
}
}

@Composable
fun RegistrationScreen(context: Context, databaseHelper: UserDatabaseHelper) {
 var username by remember { mutableStateOf("") } var
password by remember { mutableStateOf("") } var email
by remember { mutableStateOf("") } var error by
remember { mutableStateOf("") }
Column(
modifier = Modifier.fillMaxSize().background(Color.White),
horizontalAlignment = Alignment.CenterHorizontally,
verticalArrangement = Arrangement.Center     ) {
Image(painterResource(id = R.drawable.tra), contentDescription = "")
        Text( fontSize =
36.sp,
fontWeight = FontWeight.ExtraBold,
fontFamily = FontFamily.Cursive, text =
"Register"
)

        Spacer(modifier = Modifier.height(10.dp))
        TextField( value =
username,
onValueChange = { username = it }, label =
{ Text("Username") }, modifier = Modifier
                .padding(10.dp)
                .width(280.dp)
        )
        TextField( value =
email,
onValueChange = { email = it }, label = {
Text("Email") }, modifier = Modifier
                .padding(10.dp)
                .width(280.dp)

   TextField( value = password, onValueChange
= { password = it }, label = {
Text("Password") },
visualTransformation = PasswordVisualTransformation(),
modifier = Modifier
                .padding(10.dp)
                .width(280.dp)
        )


if (error.isNotEmpty()) {
            Text(
text = error,
color = MaterialTheme.colors.error,
modifier = Modifier.padding(vertical = 16.dp)
            )
        }


        Button(
onClick = {
if (username.isNotEmpty() && password.isNotEmpty() && email.isNotEmpty()) {
val user = User( id = null,
firstName = username,
lastName = null,
email = email,
password = password
)
                    databaseHelper.insertUser(user)
error = "User registered successfully"
// Start LoginActivity using the current context
context.startActivity(
                        Intent(
context,
LoginActivity::class.java
                        )
                    )


                } else {
                    error = "Please fill all fields"
} },
modifier = Modifier.padding(top = 16.dp)
        ) {
Text(text = "Register")
}
Spacer(modifier = Modifier.width(10.dp))
        Spacer(modifier = Modifier.height(10.dp))
        Row() {
Text(
modifier = Modifier.padding(top = 14.dp), text = "Have an account?"
)
             TextButton(onClick = {
             context.startActivity(
                            Intent(
                           context,
LoginActivity::class.java
                    )
                )
})


{
Spacer(modifier = Modifier.width(10.dp))
                Text(text = "Log in")
}
        }
    }
} private fun startLoginActivity(context: Context) { val
intent = Intent(context, LoginActivity::class.java)
    ContextCompat.startActivity(context, intent, null)
}
```

MainActivity.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.clickable
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.Card import
androidx.compose.material.Text import
androidx.compose.runtime.Composable import
androidx.compose.ui.Alignment import
androidx.compose.ui.Modifier import
androidx.compose.ui.draw.scale import
androidx.compose.ui.graphics.Color import
androidx.compose.ui.res.painterResource import
androidx.compose.ui.res.stringResource import
androidx.compose.ui.text.font.FontFamily import
androidx.compose.ui.text.font.FontWeight import
androidx.compose.ui.text.style.TextAlign import
androidx.compose.ui.unit.dp import
androidx.compose.ui.unit.sp


class MainActivity : ComponentActivity() { override
fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
setContent { TravelApp(this)
}
}




    @Composable
fun TravelApp(context: Context) {
Column( modifier = Modifier
                .padding(20.dp)
                .verticalScroll(rememberScrollState())
        ) {


Text(
fontSize = 40.sp,
color = Color(android.graphics.Color.rgb(120, 40, 251)),
fontFamily = FontFamily.Cursive, text
= "Wanderlust Travel"
)


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


// 01
Card(
modifier = Modifier
                    .fillMaxWidth()
                    .height(250.dp)
.clickable { context.startActivity(
                            Intent(context, BaliActivity::class.java)
                        )
},
elevation = 8.dp
            )
{
Column(
horizontalAlignment = Alignment.CenterHorizontally
                ) {
Image(
                        painterResource(id = R.drawable.bali), contentDescription = "",
modifier = Modifier
                            .height(150.dp)
                            .scale(scaleX = 1.2F, scaleY = 1F)
)
```

```
                    Text(
text = stringResource(id = R.string.place_1),
fontSize = 18.sp                    )



                    Text(
text = stringResource(id = R.string.description),
fontWeight = FontWeight.Light, fontSize = 16.sp,
textAlign = TextAlign.Center,
)


                    Text(
text = stringResource(id = R.string.plan), color = Color.Gray,
fontSize = 16.sp                    )
}
            }

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

//02
Card(
modifier = Modifier
                    .fillMaxWidth()
                    .height(250.dp)
.clickable { context.startActivity(
                            Intent(context, ParisActivity::class.java)

                        )
}, elevation =
8.dp
            )
{ Column(
horizontalAlignment = Alignment.CenterHorizontally
                ) {
Image(
                        painterResource(id = R.drawable.paris), contentDescription = "",
modifier = Modifier
                            .height(150.dp)
                            .scale(scaleX = 1.2F, scaleY = 1F)
                    )

                    Text(
text = stringResource(id = R.string.place_2),
fontSize = 18.sp
```

```
                    )



                    Text(
text = stringResource(id = R.string.description),
fontWeight = FontWeight.Light, fontSize = 16.sp,
textAlign = TextAlign.Center,
)


                    Text(
text = stringResource(id = R.string.plan), color = Color.Gray,
fontSize = 16.sp
                    )
}
            }

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

//03
Card(
modifier = Modifier
                    .fillMaxWidth()
                    .height(250.dp)
.clickable { context.startActivity(
                            Intent(context, SingaporeActivity::class.java)
                        )
},
elevation = 8.dp
            )
{
Column(
horizontalAlignment = Alignment.CenterHorizontally
                ) {
Image(
painterResource(id = R.drawable.singapore), contentDescription = "",
modifier = Modifier
                            .height(150.dp)
                            .scale(scaleX = 1.2F, scaleY = 1F)
)

Text(
text = stringResource(id = R.string.place_3),
fontSize = 18.sp )

 Text(
text = stringResource(id = R.string.description),
fontWeight = FontWeight.Light, fontSize = 16.sp,
textAlign = TextAlign.Center,
)

                    Text(
text = stringResource(id = R.string.plan), color = Color.Gray,
fontSize = 16.sp                     )
}
            }

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

BaliActivity.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.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.MaterialTheme import
androidx.compose.material.Surface import
androidx.compose.material.Text import
androidx.compose.runtime.Composable import
androidx.compose.ui.Modifier import
androidx.compose.ui.draw.scale import
androidx.compose.ui.graphics.Color import
androidx.compose.ui.res.painterResource import
androidx.compose.ui.res.stringResource import
androidx.compose.ui.text.font.FontFamily import
androidx.compose.ui.tooling.preview.Preview import
androidx.compose.ui.unit.dp import
androidx.compose.ui.unit.sp
import com.example.travelapp.ui.theme.TravelAppTheme


class BaliActivity : ComponentActivity() { override
fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState) setContent {
TravelAppTheme {
// A surface container using the 'background' color from the theme
Surface(
modifier = Modifier.fillMaxSize(), color
= MaterialTheme.colors.background
) {
PlaceOne()
}
            }
        }
}
}
 @Composable fun
PlaceOne() {
Column(modifier = Modifier.background(color = Color.White)
        .padding(20.dp)
        .verticalScroll(rememberScrollState())
    ) {
Text(
fontSize = 40.sp,
color = Color(android.graphics.Color.rgb(120, 40, 251)),
fontFamily = FontFamily.Cursive,
text = stringResource(id = R.string.place_1),
)
Image(
painterResource(id = R.drawable.bali), contentDescription = "",
modifier = Modifier
                .padding(16.dp)
                .fillMaxWidth()
                .height(200.dp)
                .scale(scaleX = 1.2F, scaleY = 1F)
        )
Text(
color=Color.Black,
text = "Day 1: Arrival and Relaxation\n" +

"Arrive in Bali and check into your hotel or accommodation.\n" +
"Spend the day relaxing and getting acclimated to the island.\n" +
"If you have time, explore the nearby area or head to the beach.\n" +
"\n" +

"Day 2: Ubud Tour\n" +

"Start your day early and head to Ubud, a cultural and artistic hub in Bali.\n" +
"Visit the Monkey Forest and the Ubud Palace.\n" +
"Take a tour of the Tegalalang Rice Terrace, a beautiful UNESCO World Heritage Site.\n" +
"End your day with a traditional Balinese dance performance.\n" +
"\n" +
"Day 3: Temple Hopping\n" +

"Visit some of Bali's most famous temples, such as Tanah Lot and Uluwatu.\n" +
"Take in the stunning views of the ocean and cliffs.\n" +
"Enjoy a sunset dinner at one of the many restaurants near the temples.\n" + "\n"
+
"Day 4: Waterfalls and Beaches\n" +
"Take a day trip to Bali's beautiful waterfalls, such as Tegenungan or Gitgit.\n" + "Spend
the afternoon at one of Bali's world-renowned beaches, like Seminyak or Nusa
Dua.\n" +
"\n" +
"Day 5: Island Hopping\n" +
"Take a day trip to one of Bali's neighboring islands, such as Nusa Lembongan or Gili
Islands.\n" +
"Snorkel or scuba dive in the clear waters and relax on the beach.\n" + "\n"
+
"Day 6: Cultural Activities\n" +
"Visit a traditional Balinese village and learn about the island.\n" +
"\n" +
"Day 7: Departure\n" +
"Explore the surrounding area and take in the stunning sunset views.\n" +
"Have dinner at a local restaurant before returning to your accommodation."
)



}
}
```

ParisActivity.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.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.MaterialTheme import
androidx.compose.material.Surface import
androidx.compose.material.Text import
androidx.compose.runtime.Composable import
androidx.compose.ui.Modifier import
androidx.compose.ui.draw.scale import
androidx.compose.ui.graphics.Color import
androidx.compose.ui.res.painterResource import
androidx.compose.ui.res.stringResource import
androidx.compose.ui.text.font.FontFamily import
androidx.compose.ui.tooling.preview.Preview import
androidx.compose.ui.unit.dp import
androidx.compose.ui.unit.sp
import com.example.travelapp.ui.theme.TravelAppTheme

class ParisActivity : ComponentActivity() { override
fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState) setContent {
TravelAppTheme {
// A surface container using the 'background' color from the theme
Surface(
modifier = Modifier.fillMaxSize(), color
= MaterialTheme.colors.background ) {
Greeting()
}
            }
        }
}
}

@Composable fun
Greeting() {
 Column(
modifier = Modifier.background(color = Color.White)
            .padding(20.dp)
            .verticalScroll(rememberScrollState())
```

```
    ) {
Text( fontSize =
40.sp,
color = Color(android.graphics.Color.rgb(120, 40, 251)),
fontFamily = FontFamily.Cursive,
text = stringResource(id = R.string.place_2),
)
Image(
painterResource(id = R.drawable.paris), contentDescription = "",
modifier = Modifier
                .padding(16.dp)
                .fillMaxWidth()
                .height(200.dp)
                .scale(scaleX = 1.2F, scaleY = 1F)
        )
Text(
color=Color.Black,
text = "Day 1: Arrival and Introduction\n" +

"Check into your accommodation and freshen up\n" +
"Take a stroll around the neighborhood to get acquainted\n" +
"Visit the Eiffel Tower, preferably in the evening when it is lit up\n" +
"Have a relaxing dinner at a nearby restaurant\n" +

"\n" +
"Day 2: Art and History\n" +

"Visit the Louvre Museum to see some of the world's most famous art pieces\n" +
"Stroll through the Tuileries Garden and the Place de la Concorde\n" +
"Visit the Orsay Museum, which houses a large collection of impressionist art\n" + "Have
dinner at a local French restaurant\n" +

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

"Visit the Montmartre neighborhood to see the famous Basilique du Sacré-Cœur and Place du
Tertre\n" +
"Explore the historic neighborhood of Le Marais\n" +
"Try some delicious French pastries at a local bakery\n" +
"Have dinner at a brasserie to taste some classic French cuisine\n" +
"\n" +

"Day 4: Architecture and Gardens\n" +

"Visit the Palace of Versailles, a UNESCO World Heritage site, and explore its beautiful
gardens\n" +
"Walk along the Champs-Elysées and stop at the Arc de Triomphe\n" +
"Visit the Sainte-Chapelle, a beautiful Gothic chapel with stunning stained-glass windows\n" +
"Have dinner at a local restaurant in the 7th arrondissement\n" +
"\n" +
"Day 5: Shopping and Sightseeing\n" +
"Visit the Notre-Dame Cathedral and climb up to the top for a stunning view of the city\n" +
"Explore the Latin Quarter and visit the Panthéon\n" +
"Go shopping at the famous Galeries Lafayette or Printemps department stores\n" +
"Have dinner at a local bistro\n" +
"\n" +
"Day 6: Parisian Parks and Museums\n" +

"Visit the Musée Rodin and explore its beautiful gardens\n" +
"Stroll through the Luxembourg Gardens and visit the Luxembourg Palace\n" +
"Visit the Centre Pompidou, a modern art museum in the Marais neighborhood\n" +
"Have dinner at a local restaurant in the Latin Quarter\n" +
"\n" +
"Day 7: River Cruise and Farewell\n" +

"Take a boat cruise along the Seine River to see the city from a different perspective\n" +
"Visit the Musée de l'Orangerie, which houses Monet's famous water lilies paintings\n" +
"Have a farewell dinner at a Michelin-starred restaurant"
)
}
}
```

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
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.backgroun
d(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 "
)
}
}
```

AndroidManifest.kt

```
<?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>

</application>
</activity>
```

[redacted]

```
</manifest>
```

THANK YOU

[redacted]