1. INTRODUCTION:

1.1 Overview

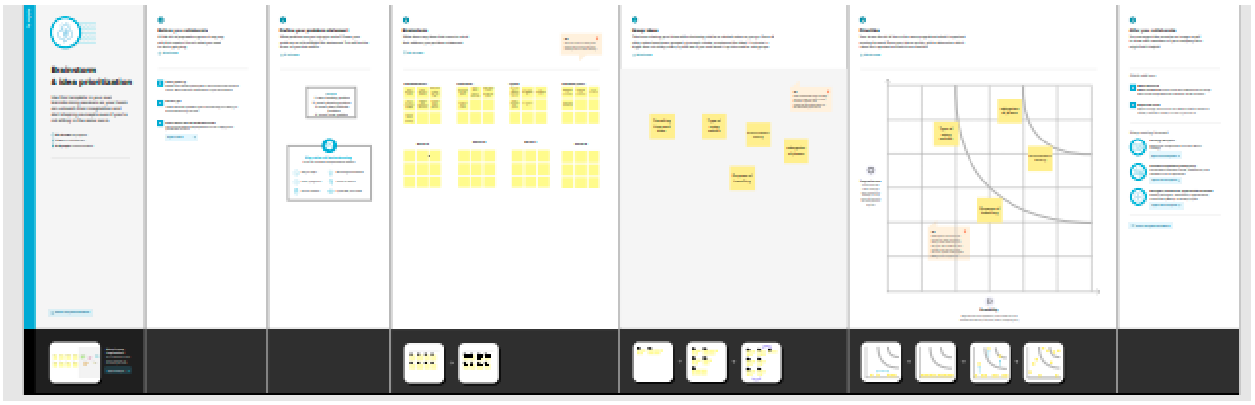
Now a days, people are interested in travelling and searching for different tourist location for travel planning according to their interests. Social media has come out as a medium to fulfill the continuous needs for automatic travel recommendation. It offers great opportunities to address many challenging problems, like GPS estimation and travel recommendation. Travelogue websites are basically blogs which offers rich descriptions about landmarks and also about the travelling experience which are written by users. These data are not only useful for determining POIs. points of interest but also gives an opportunity to recommend personalized travel POIs and routes based on user's interest. Existing studies on travel recommendation use the different types of social media data, GPS trajectory, check-in-data, geo tag and blogs which are used for mining famous travel POIs and routes. The existing system for travel recommendation has certain flaws due to which it cannot meet user's personal requirements. Personalized recommendation of travel system uses location-based collaborative filtering method in order to recommend the POIs and optimized routes by mining user's travel history. In this method, social similar users are mapped based on the location co-occurrence of previously visited POIs and then POIs are ranked according to the similar users travel history. There are two problems in automatic travel recommendation that needs to be discussed when compared with static existing travel recommendation approach. First, the recommended POIs should be personalized to user interest since different users may prefer different types of POIs. Second, it is important to recommend a sequential travel route that is a sequence of POIs rather than individual POI. Existing system on travel recommendation typically comprises of two problems. The first problem, most of the travel recommendation system focuses only on user topical interest mining without considering other crucial attributes like consumption capability of the user. And for the second problem, existing systems focuses more on famous route mining rather than considering user travel interest. To solve the above enlisted challenges, the new system proposes Topical Package Model method which automatically mines user travel interest from two types of social media data, different user contributed photos and travelogues. For the first problem, it considers user's topical interest with the attribute like consumption capability and preference of visiting time of user. As it is difficult to measure the similarity directly between user and route, the proposed system uses topical package model which maps both user’s and route’s textual descriptions to the topical package model to get user package and route package using topical package space.

1.2 Purpose

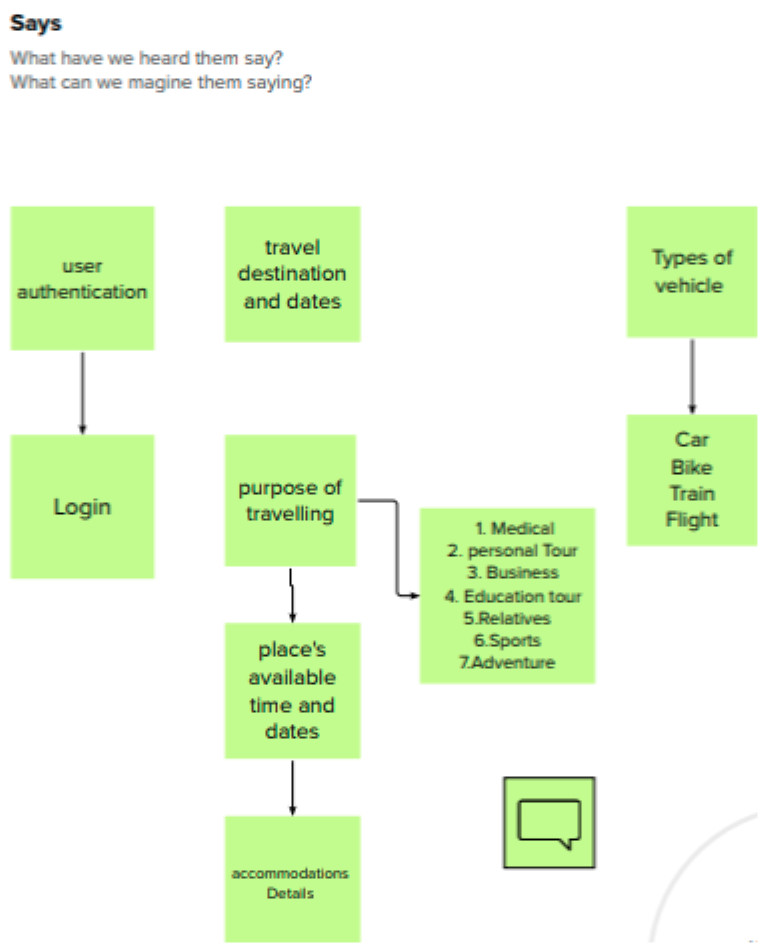
To mine users travel interest, a Topical Package Model(TPM) has been used where the data is fetched from community contributed photos and travelogues. To address the existing first challenge, Our system considers not only user’s topical interest but also the consumption capability and preference of visiting time and season. It is difficult to map the users point of interest(POI), so here the users point of interest(POI) will be input to the topical package space model , and the algorithm will return an optimized route for each day as well as a personalized itinerary First, tags of user’s photo set are mapped to topical package space to get user’s topical interest distribution. It is difficult to get user’s consumption capability directly from the textual descriptions of photos. But the topics user interested in could somehow reflect these attributes. For example, if a user usually takes part in luxurious activities like Golf and Spas, he is more likely to be rich. Our system combines user topical interest and the cost, time, season distribution of each topic to mine user’s consumption capability, preferred visiting time and season. After user package mining, Our system ranks famous routes through measuring user package and routes package. At last, Our system optimizes the top ranked routes through social similar users’ travel records in this city. Social similar users are measured by the similarity of user packages.

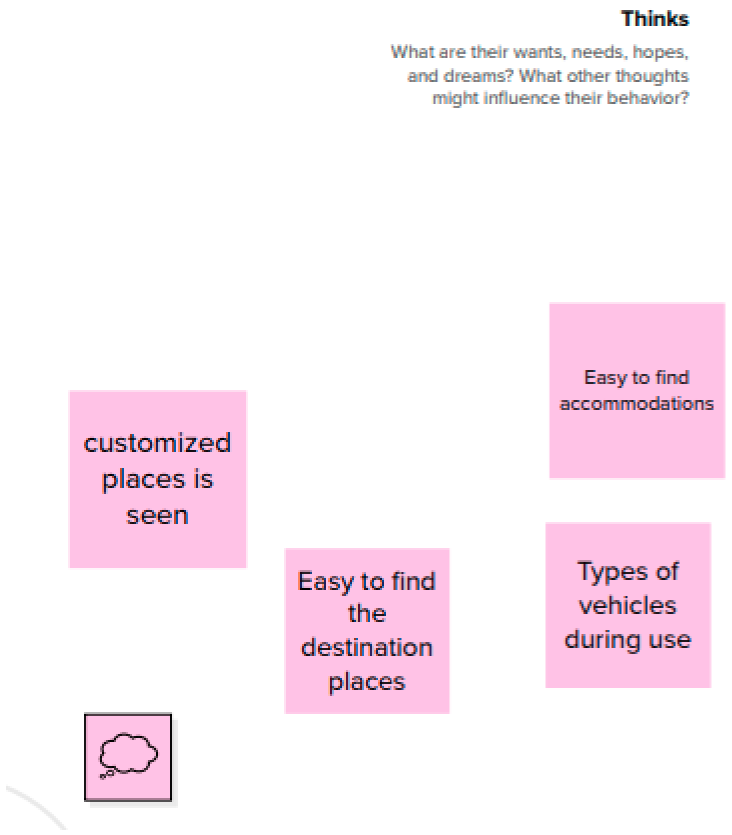
2. PROBLEM DEFINITION & DESIGN THINKING

2.1 Brain storm:



2.2 Empathy Map:


Says
What have we heard them say?
What can we magine them saying?
user authentication
Login
travel destination and dates
purpose of travelling
place's available time and dates
accommodations Details
1. Medical
2. personal Tour
3. Business
4. Education tour
5.Relatives
6.Sports
7.Adventure
Types of vehicle
Car
Bike
Train
Flight



Thinks

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

Easy to find accommodations

customized places is seen

Easy to find the destination places

Types of vehicles during use

Discuss about my project. with my guide

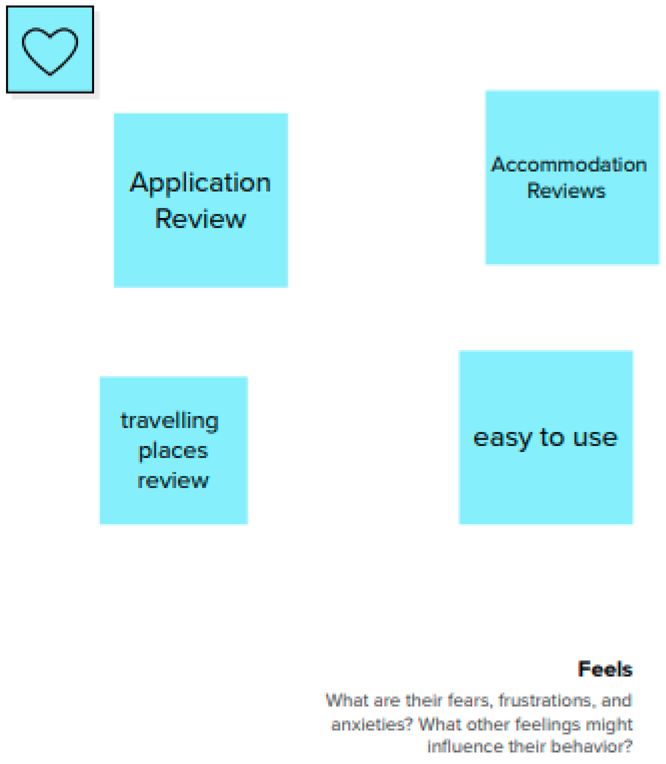
Discuss about my project with my brother

Discuss about with my teammates

If refer Google about my projec

Does

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

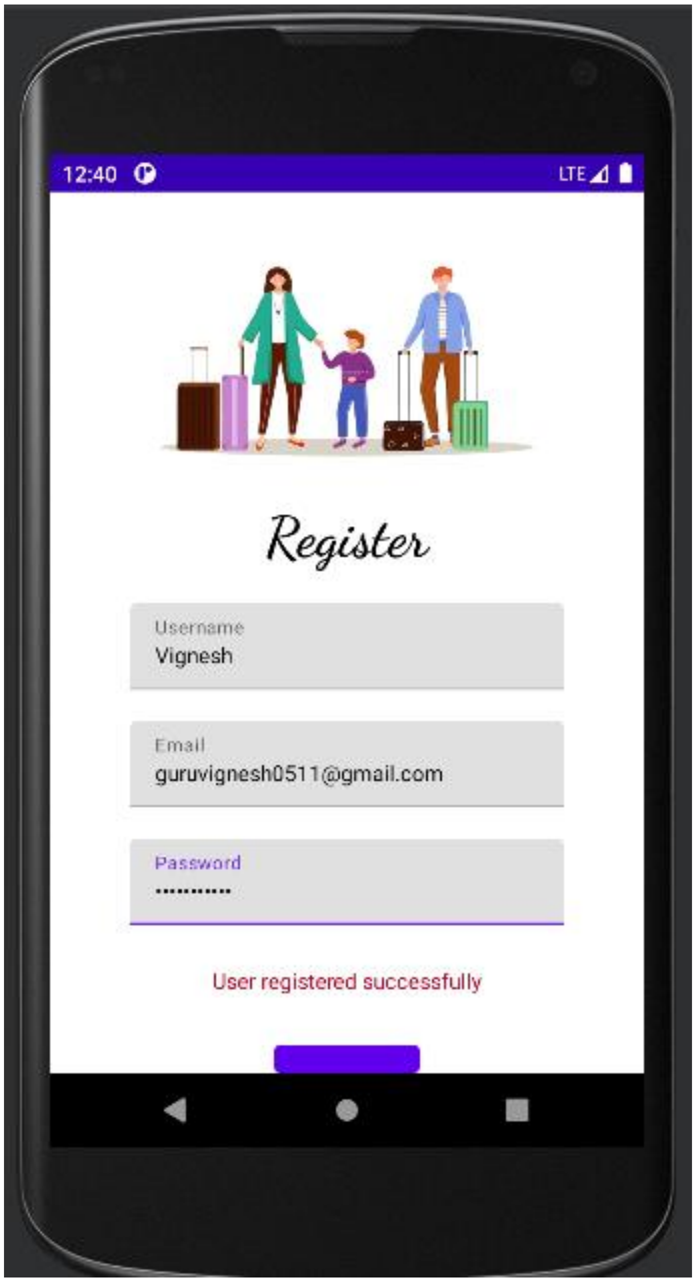


Feels

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

3. RESULT

Screenshot:



12:40

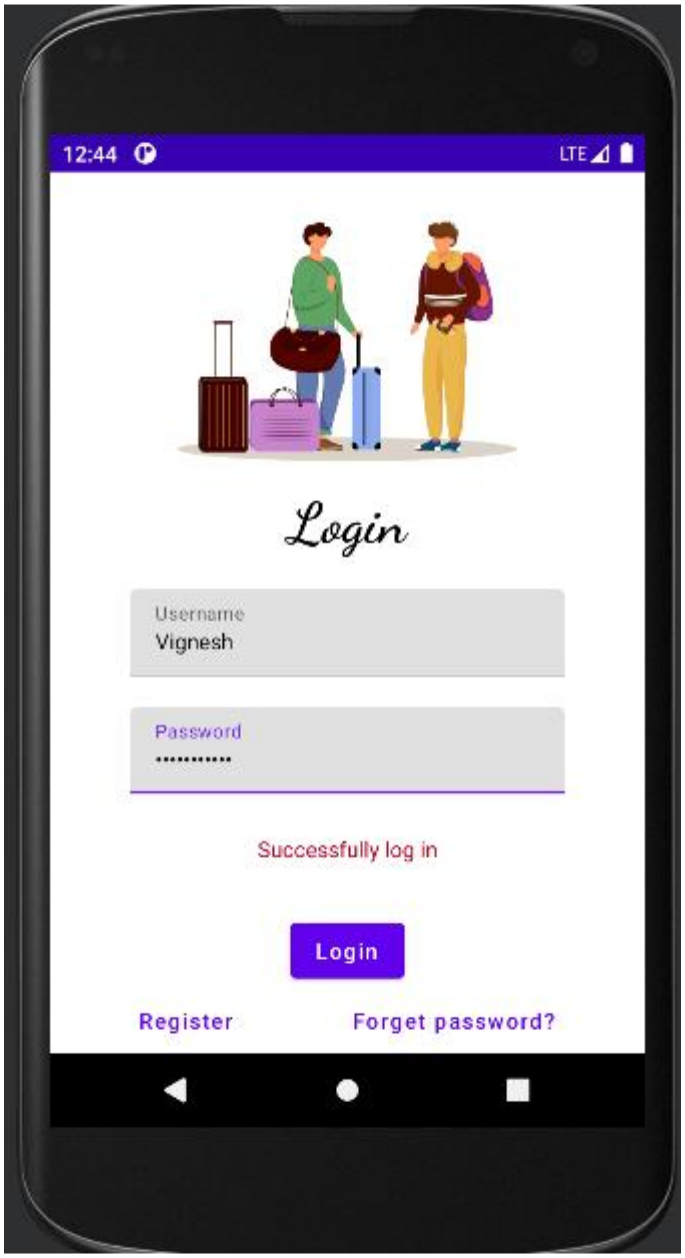
Register

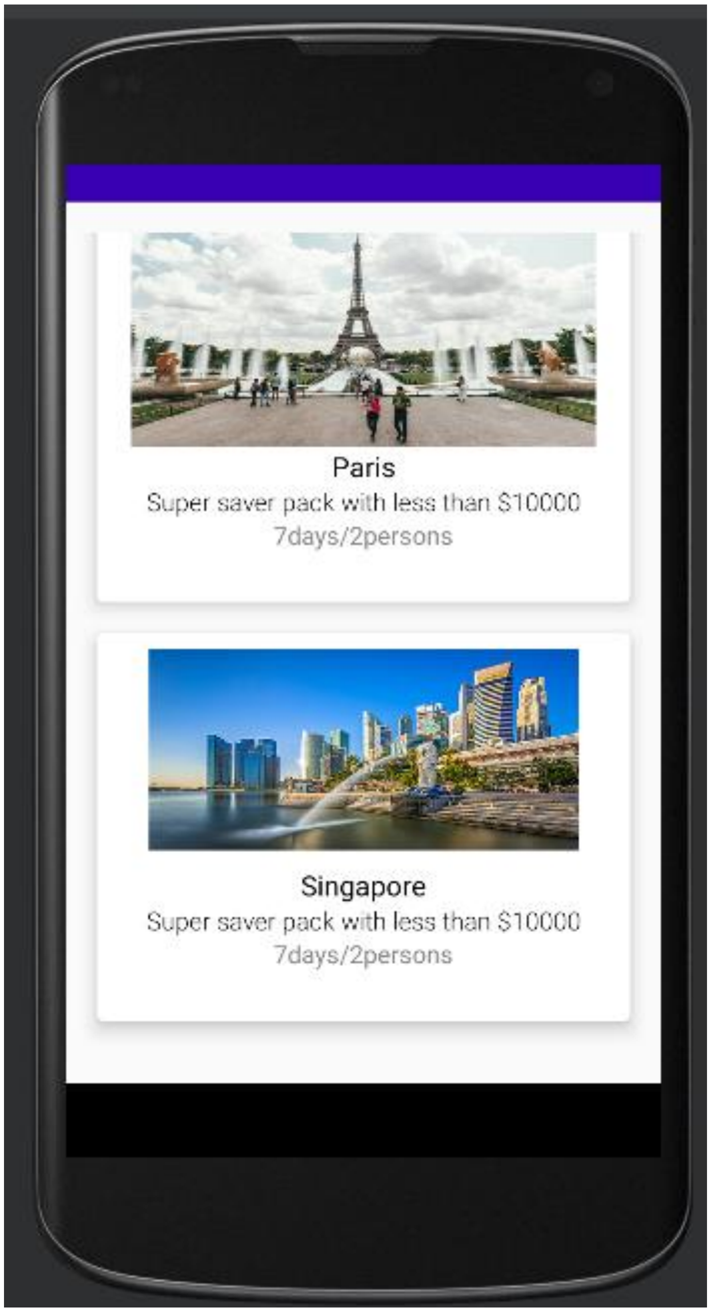
Username
Vignesh

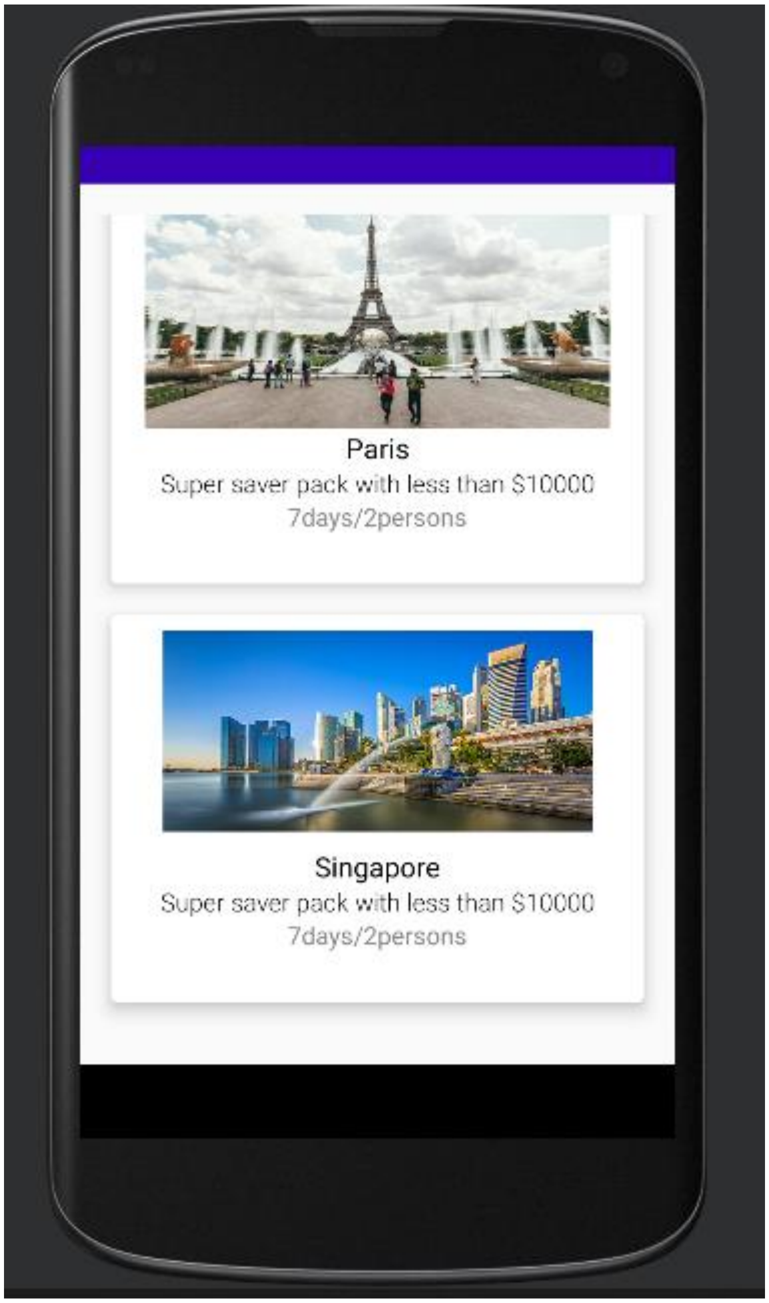
Email
guruvignesh0511@gmail.com

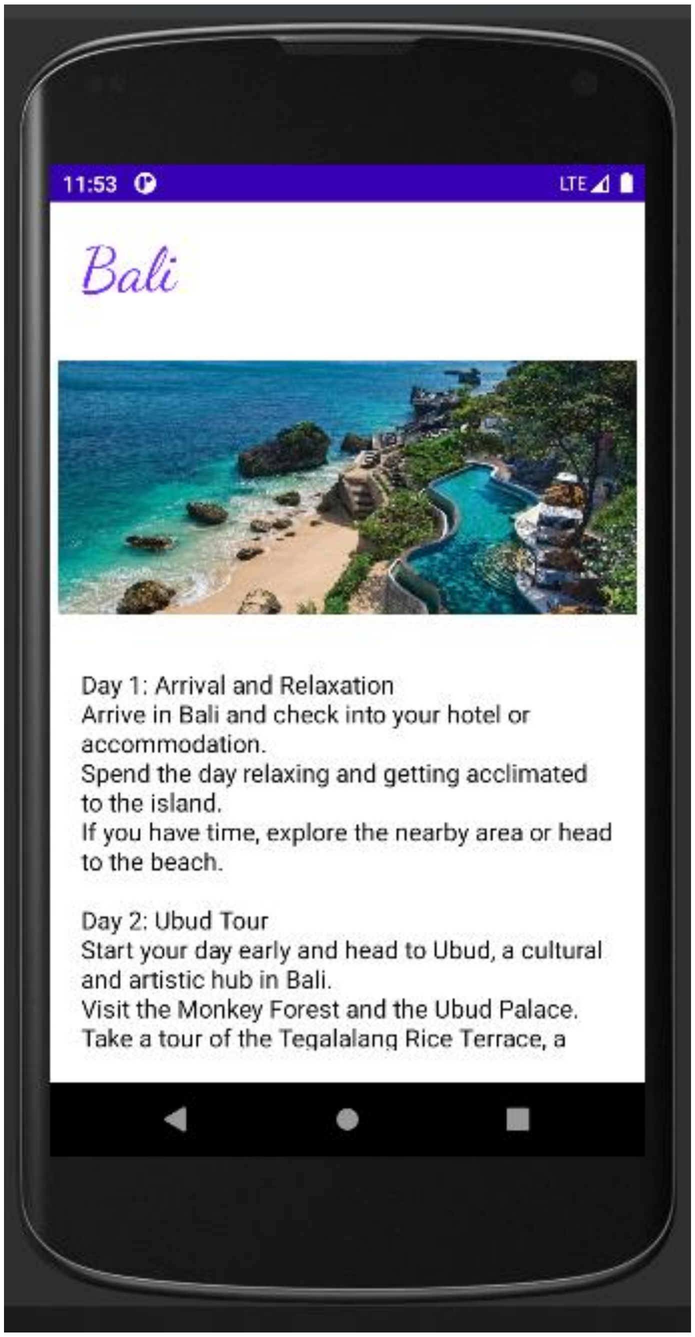
Password
•••••••••••

User registered successfully


12:44
LTE
Login
Username
Vignesh
Password
•••••••••••
Successfully log in
Login
Register
Forget password?


Paris
Super saver pack with less than $10000
7days/2persons
Singapore
Super saver pack with less than $10000
7days/2persons

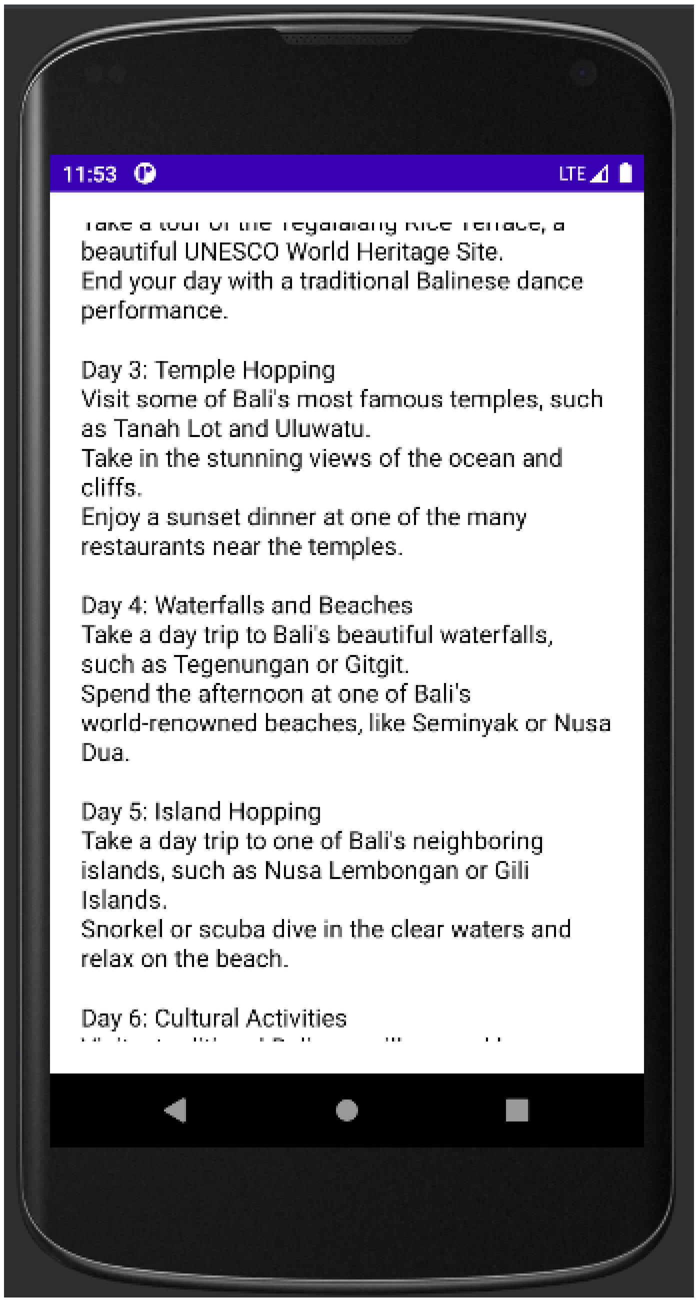

Paris
Super saver pack with less than $10000
7days/2persons
Singapore
Super saver pack with less than $10000
7days/2persons



Bali

Day 1: Arrival and Relaxation
Arrive in Bali and check into your hotel or accommodation.
Spend the day relaxing and getting acclimated to the island.
If you have time, explore the nearby area or head to the beach.

Day 2: Ubud Tour
Start your day early and head to Ubud, a cultural and artistic hub in Bali.
Visit the Monkey Forest and the Ubud Palace.
Take a tour of the Tegalalang Rice Terrace, a



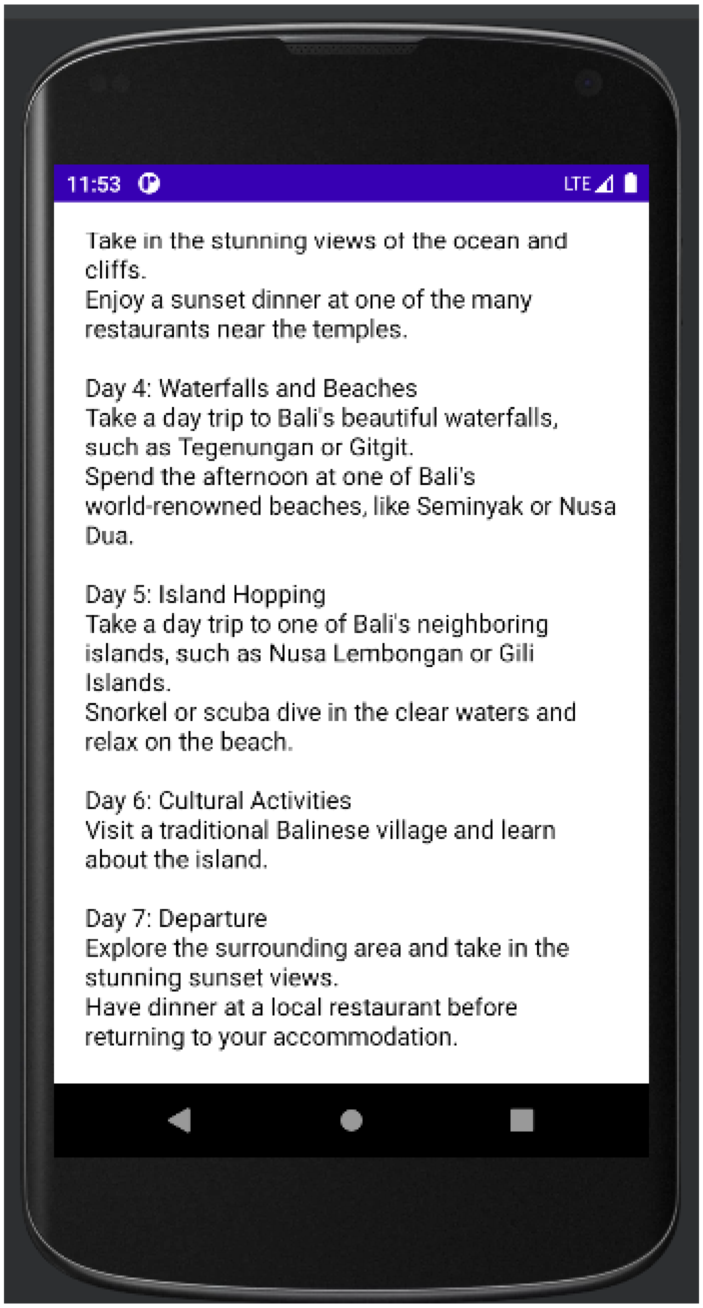
Take a tour of the Tegalalang Rice Terrace, a beautiful UNESCO World Heritage Site.
End your day with a traditional Balinese dance performance.

Day 3: Temple Hopping
Visit some of Bali's most famous temples, such as Tanah Lot and Uluwatu.
Take in the stunning views of the ocean and cliffs.
Enjoy a sunset dinner at one of the many restaurants near the temples.

Day 4: Waterfalls and Beaches
Take a day trip to Bali's beautiful waterfalls, such as Tegenungan or Gitgit.
Spend the afternoon at one of Bali's world-renowned beaches, like Seminyak or Nusa Dua.

Day 5: Island Hopping
Take a day trip to one of Bali's neighboring islands, such as Nusa Lembongan or Gili Islands.
Snorkel or scuba dive in the clear waters and relax on the beach.

Day 6: Cultural Activities



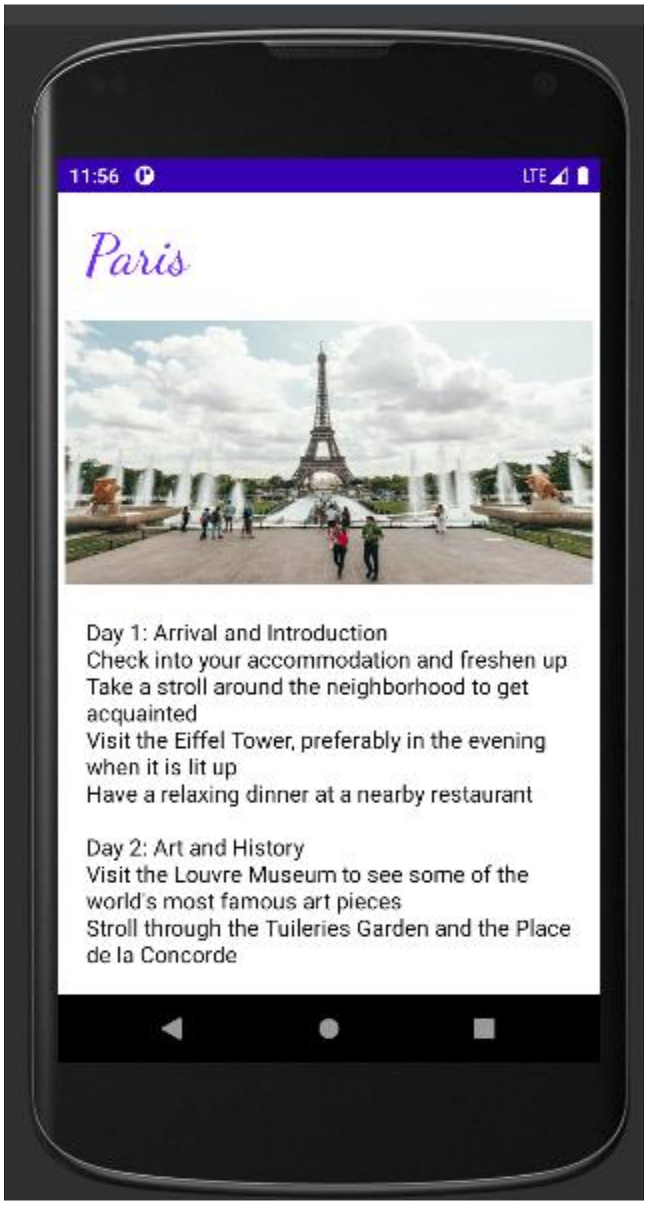
Take in the stunning views of the ocean and cliffs.
Enjoy a sunset dinner at one of the many restaurants near the temples.

Day 4: Waterfalls and Beaches
Take a day trip to Bali's beautiful waterfalls, such as Tegenungan or Gitgit.
Spend the afternoon at one of Bali's world-renowned beaches, like Seminyak or Nusa Dua.

Day 5: Island Hopping
Take a day trip to one of Bali's neighboring islands, such as Nusa Lembongan or Gili Islands.
Snorkel or scuba dive in the clear waters and relax on the beach.

Day 6: Cultural Activities
Visit a traditional Balinese village and learn about the island.

Day 7: Departure
Explore the surrounding area and take in the stunning sunset views.
Have dinner at a local restaurant before returning to your accommodation.



Paris

Day 1: Arrival and Introduction
Check into your accommodation and freshen up
Take a stroll around the neighborhood to get acquainted
Visit the Eiffel Tower, preferably in the evening when it is lit up
Have a relaxing dinner at a nearby restaurant

Day 2: Art and History
Visit the Louvre Museum to see some of the world's most famous art pieces
Stroll through the Tuileries Garden and the Place de la Concorde

4. ADVANTAGES & DISADVANTAGES

Advantages:

Real-time updates: Wanderlust provides real time updates on gate changes, and other important travel information, so users can stay informed and adjust their plans accordingly.

- **Budget detail** : Wanderlust helps users their travel expenses and stay within their budget by providing estimated costs for activities, accommodations, and transportation Social sharing: Wanderlust allows users to share their travel experiences with friends and family through social media platforms, providing inspiration and recommendations for future trips.
- **Customizable maps**: Wanderlust provides customizable maps that allow users to mark their favorite spots, track their travels, and share their maps with others.
- Overall, Wanderlust is a valuable tool for travelers who want to plan personalized trips, stay informed, and track their experiences. With its user-friendly interface and many features, it is a great app for anyone with a case of wanderlust.

Disadvantage:

- **Data privacy concerns**: As the app would be collecting personal information about users' travel plans, preferences, and even location, there may be concerns about data privacy and security. Users may be hesitant to share such information with the app, which could limit its user base.
- **Competition:** The travel industry is already crowded with various travel apps and websites. Wanderlust would need to offer something unique and valuable to stand out and attract users.
- **User engagement:** While the idea of personalized travel planning and tracking may sound appealing, getting users to consistently engage with the app could be a challenge. Users may lose interest in the app after planning a few trips or finding that the app does not meet their specific needs.
- **Technical challenges**: Developing a personalized travel planning and tracking app would require a significant investment of time and resources. There could be technical challenges in integrating various data sources, ensuring a seamless user experience, and maintaining the app's functionality.
- **Dependence on travel industry**: The app's success could be dependent on the state of the travel industry. Any downturn or restrictions on travel due to pandemics, natural disasters or political issues can affect the app's performance and revenue.

5. APPLICATION

It includes countries like Paris, Bali, Singapore, we don’t know new places there when we go on a tour, we have divide days separately for each of these places to easily go to the tourist places and for example, when we go to Paris, it is given from 1 to 7 days which places we can visit in this for 2 people. It costs 10000 dollars. That way you can know how much each places costs.

6. CONCLUSION

In this paper, Our system presents a personalized travel itinerary recommendation system by implementing topical package model using data mining form social media: travelogues and community-contributed photos. The advantages of our work are that firstly, the system automatically mines user ’s topical preferences including the point of interest, cost and time and secondly the recommendation is not only providing point of interest but also travel sequence order, considering both the popularity and user’s travel preferences at the same time. The system also provides user with flexibility to freeze a day or two for his/her personal work and successfully managed to show the travel itinerary and hotel bookings for comfortable stay in a single framework. Out project mines and ranks famous routes based on the similarity between user and route package and then optimizes the top ranked famous routes according to social similar users’ travel records thereby providing user with the most efficient and feasible route.

7. FUTURE SCOPE

The current project gives user its own personalized travel itinerary based on his or her travel interests and point of interests along with hotel stay information. For future work, more type of data for mining user interest can be used and also the system can provide new features which include providing air ticket details for a more convenient tour planning. Also a more detailed input can be taken from the user, asking the user about its eating preferences and based on that the system can suggest restaurants near every point of interests. Other miscellaneous things such as, giving the user specific privileges to tailor the itinerary by removing or replacing a particular place in the trip, can be added in the future. As the web-surfing era is about to end, the website can be converted into faster and easily accessible smart phone application and expand the project by providing itineraries for every place in the world. Also the website can be made more secured and different attacks are prevented using techniques like CAPTCHA [19], Text Image Ciphering [20] and Hybrid Key Distribution Systems [21].

8. APPENDIX

8.1 Source code:

Main page:

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.TravelApp"
        tools:targetApi="31">
        <activity
            android:name=".RegisterActivity"
            android:exported="false"
            android:label="RegisterActivity"
            android:theme="@style/Theme.TravelApp"                                  />
        <activity
            android:name=".SingaporeActivity"
            android:exported="false"
            android:label="@string/title_activity_singapore"
            android:theme="@style/Theme.TravelApp"                                  />
        <activity
```

```
            android:name=".ParisActivity"
            android:exported="false"
            android:label="@string/title_activity_paris"
            android:theme="@style/Theme.TravelApp"                                         />
        <activity
            android:name=".BaliActivity"
            android:exported="false"
            android:label="@string/title_activity_bali"
            android:theme="@style/Theme.TravelApp"                                         />
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:label="@string/app_name"
            android:theme="@style/Theme.TravelApp"/>
        <activity
            android:name=".LoginActivity"
            android:exported="true"
            android:label="@string/app_name"
            android:theme="@style/Theme.TravelApp">
            <intent-filter>
                <action              android:name="android.intent.action.MAIN"             />

                <category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>
    </application>


</manifest>
```

Login page:

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.TravelApp"
        tools:targetApi="31">
        <activity
            android:name=".RegisterActivity"
            android:exported="false"
            android:label="RegisterActivity"
            android:theme="@style/Theme.TravelApp" />
        <activity
            android:name=".SingaporeActivity"
            android:exported="false"
            android:label="@string/title_activity_singapore"
            android:theme="@style/Theme.TravelApp" />
        <activity
            android:name=".ParisActivity"
            android:exported="false"
            android:label="@string/title_activity_paris"
            android:theme="@style/Theme.TravelApp" />
        <activity
            android:name=".BaliActivity"
            android:exported="false"
            android:label="@string/title_activity_bali"
            android:theme="@style/Theme.TravelApp" />
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:label="@string/app_name"
            android:theme="@style/Theme.TravelApp"/>
        <activity
            android:name=".LoginActivity"
            android:exported="true"
            android:label="@string/app_name"
            android:theme="@style/Theme.TravelApp">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Travels:

```
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)
```

```
}
```

Bali Activity:

```
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."

            )



          }

        }
```

Main activity:

```
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))
        }
    }
}
```

Paris Activity:

```
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"
        )
    }
}
```

Singapore Activity:

```
package com.example.travelapp


import android.os.Bundle

import androidx.activity.ComponentActivity

import androidx.activity.compose.setContent

import androidx.compose.foundation.Image

import androidx.compose.foundation.background

import androidx.compose.foundation.layout.*

import androidx.compose.foundation.rememberScrollState

import androidx.compose.foundation.verticalScroll

import androidx.compose.material.MaterialTheme

import androidx.compose.material.Surface

import androidx.compose.material.Text

import androidx.compose.runtime.Composable

import androidx.compose.ui.Modifier

import androidx.compose.ui.draw.scale

import androidx.compose.ui.graphics.Color

import androidx.compose.ui.res.painterResource

import androidx.compose.ui.res.stringResource

import androidx.compose.ui.text.font.FontFamily

import androidx.compose.ui.tooling.preview.Preview

import androidx.compose.ui.unit.dp

import androidx.compose.ui.unit.sp
import com.example.travelapp.ui.theme.TravelAppTheme


class SingaporeActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            TravelAppTheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                ) {
                    Greeting2()
                }
            }
        }
    }
}


@Composable
fun Greeting2() {


    Column(
        modifier = Modifier.background(color = Color.White)
            .padding(20.dp)
            .verticalScroll(rememberScrollState())
    ) {
        Text(
            fontSize = 40.sp,
            color = Color(android.graphics.Color.rgb(120, 40, 251)),
            fontFamily = FontFamily.Cursive,
            text = stringResource(id = R.string.place_3),
        )
        Image(
            painterResource(id = R.drawable.singapore), contentDescription = "",
            modifier = Modifier
                .padding(16.dp)
                .fillMaxWidth()
                .height(200.dp)
                .scale(scaleX = 1.2F, scaleY = 1F)
        )
        Text(
            color = Color.Black,
            text = "Day 1:\n" +

                    "Morning: Visit Gardens by the Bay and marvel at the Supertree Grove and the
Flower Dome and Cloud Forest conservatories.\n" +
                    "Afternoon: Explore the Marina Bay Sands complex, which includes a casino,
luxury shopping mall, and observation deck with a stunning view of the city.\n" +
                    "\n" +
                    "Day 2:\n" +
```

"Morning: Explore the historic district of Chinatown, including the Buddha Tooth Relic Temple and Museum and the Sri Mariamman Temple.\n" +

"Afternoon: Visit the nearby Clarke Quay for lunch and to explore its waterfront restaurants, bars, and shops.\n" +

"\n" +

"Day 3:\n" +

"Morning: Take a tour of the UNESCO-listed Botanic Gardens, one of the world's most famous and significant tropical gardens.\n" +

"Afternoon: Head over to the National Museum of Singapore, which houses a vast collection of historical and cultural artifacts.\n" +

"\n" +

"Day 4:\n" +

"Morning: Visit the Singapore Zoo and admire the wildlife, including orangutans, tigers, and elephants.\n" +

"Afternoon: Head over to Sentosa Island and relax at one of its many beaches or try some of the many attractions such as Universal Studios Singapore or Adventure Cove Waterpark.\n" +

"\n" +

"Day 5:\n" +

"Morning: Go on a nature walk at MacRitchie Reservoir, which offers hiking trails and stunning views of the city skyline.\n" +

"Afternoon: Visit Little India, a vibrant and colorful neighborhood, and explore the shops, temples, and food stalls.\n" +

```
                    "\n" +
                    "Day 6:\n" +


                    "Morning: Explore the trendy neighborhood of Tiong Bahru, known for its hip cafes
and boutiques, as well as its Art Deco architecture.\n" +
                    "Afternoon: Visit the National Gallery Singapore, which houses the largest public
collection of modern art in Singapore and Southeast Asia.\n" +
                    "\n" +
                    "Day 7:\n" +


                    "Morning: Take a day trip to the nearby island of Pulau Ubin, where you can rent a "
        )
    }
}
```

Register:

```
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)
}
```

User:

```
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?,


    )
```

User Dao:

```
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)
}
```

Data base:

```
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)
}
```

Database Helper:

```
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?.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()

    }


    @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 getUserById(id: Int): User? {
        val db = readableDatabase
        val cursor: Cursor = db.rawQuery("SELECT * FROM $TABLE_NAME WHERE
$COLUMN_ID = ?", arrayOf(id.toString()))
        var user: User? = null
        if (cursor.moveToFirst()) {
            user = User(
                id = cursor.getInt(cursor.getColumnIndex(COLUMN_ID)),
                firstName = cursor.getString(cursor.getColumnIndex(COLUMN_FIRST_NAME)),
                lastName = cursor.getString(cursor.getColumnIndex(COLUMN_LAST_NAME)),
                email = cursor.getString(cursor.getColumnIndex(COLUMN_EMAIL)),
                password = cursor.getString(cursor.getColumnIndex(COLUMN_PASSWORD)),
            )
        }
        cursor.close()
        db.close()
        return user
    }

    @SuppressLint("Range")
```

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

}
```

Example Instrumented Test:

```
package com.example.travelapp


import androidx.test.platform.app.InstrumentationRegistry
import androidx.test.ext.junit.runners.AndroidJUnit4


import org.junit.Test
import org.junit.runner.RunWith


import org.junit.Assert.*


/**
 * Instrumented test, which will execute on an Android device.
 *
 * See [testing documentation](http://d.android.com/tools/testing).
 */
@RunWith(AndroidJUnit4::class)
class ExampleInstrumentedTest {
    @Test
    fun useAppContext() {
        // Context of the app under test.
        val appContext = InstrumentationRegistry.getInstrumentation().targetContext
        assertEquals("com.example.travelapp", appContext.packageName)
    }
}
```

ExampleUnitTest:

```
package com.example.travelapp

import org.junit.Test

import org.junit.Assert.*

/**
 * Example local unit test, which will execute on the development machine (host).
 *
 * See [testing documentation](http://d.android.com/tools/testing).
 */
class ExampleUnitTest {
    @Test
    fun addition_isCorrect() {
        assertEquals(4, 2 + 2)
    }
}
```