1. INTRODUCTION

1.1OVERVIEW

Nowadays, 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 fulfil 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 i.e. 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 users's and route's textual descriptions to the topical package model to get user package and route package using topical package space.

1.2PURPOSE

The purpose of a wanderlust travel and tracking app is to help travellers plan and keep track of their trips. Such an app can provide a range of features, including:

Trip planning: The app can help travellers plan their trips by suggesting destinations, creating itineraries, and providing information about hotels, restaurants, and attractions.

Travel tracking: The app can track travellers' movements and help them keep track of where they have been and where they still need to go.

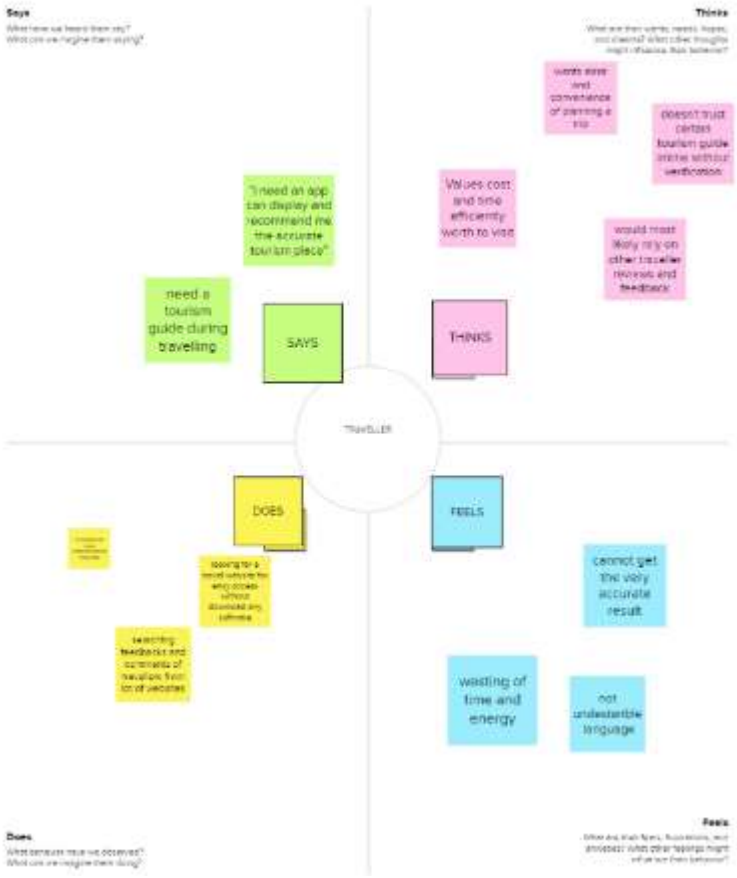
Social sharing: The app can allow travellers to share their experiences with friends and family by posting photos, updates, and reviews.

Safety and security: The app can provide information on safety and security issues in different destinations, as well as emergency contact information and other resources.

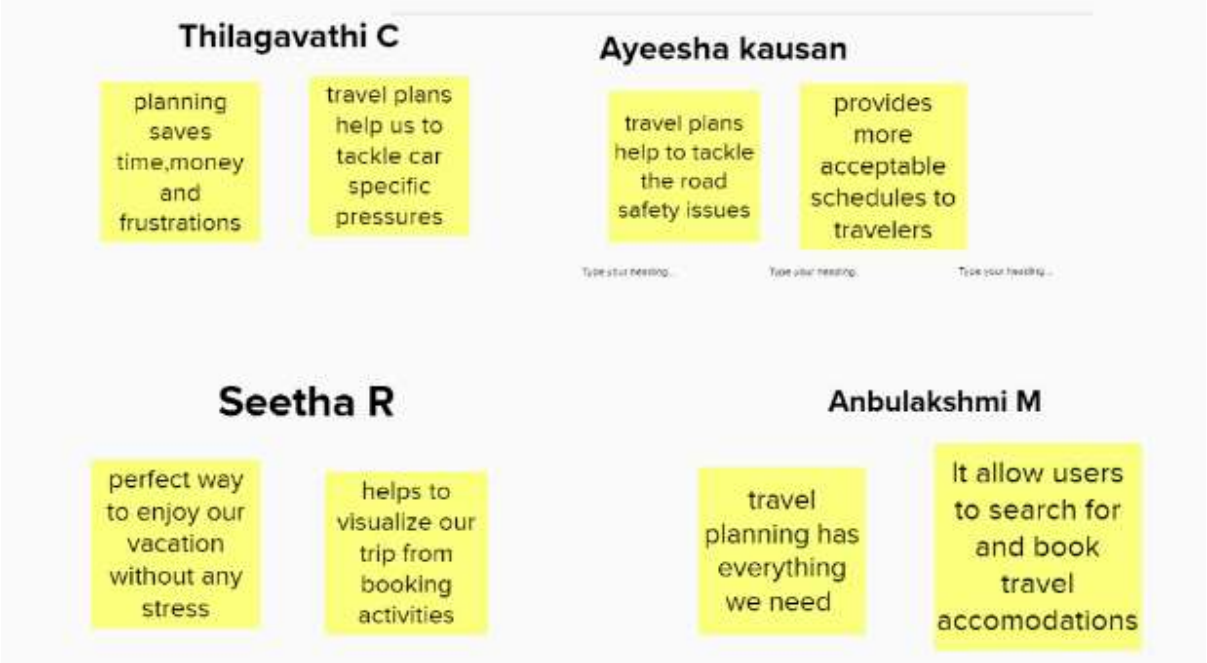
Overall, the purpose of a wanderlust travel and tracking app is to make travel easier, more convenient, and more enjoyable for travellers, while also providing them with useful information and resources to enhance their travel experiences.

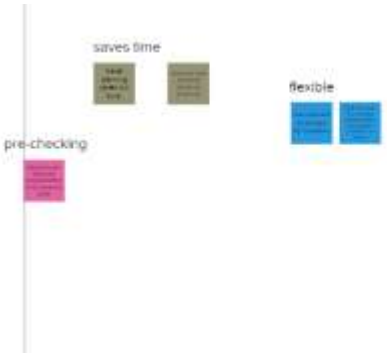
2.PROBLEM DEFINITION & DESIGN THINKING

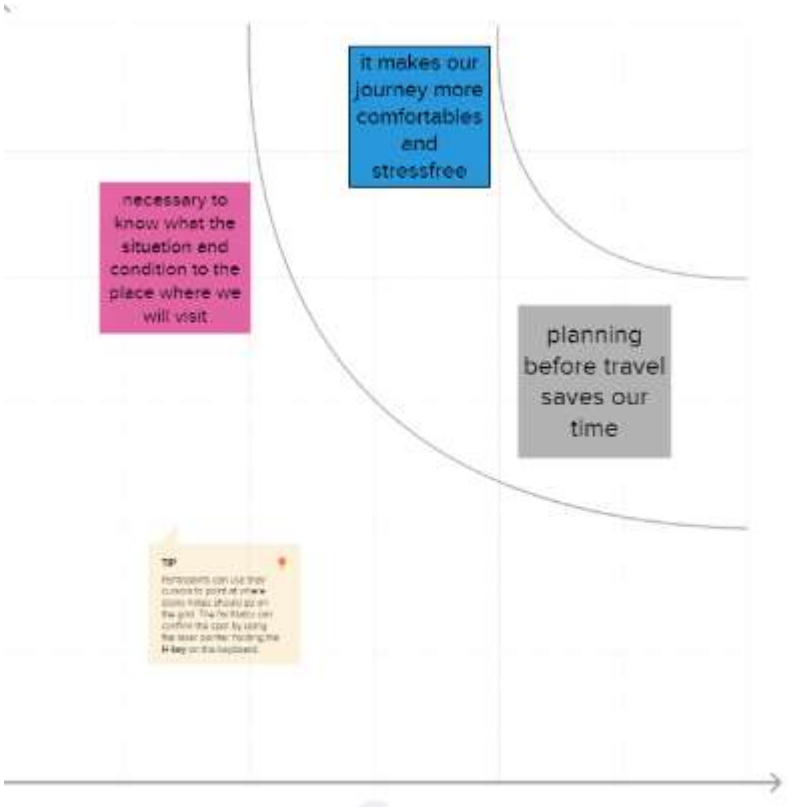
2.1 EMPATHY MAP



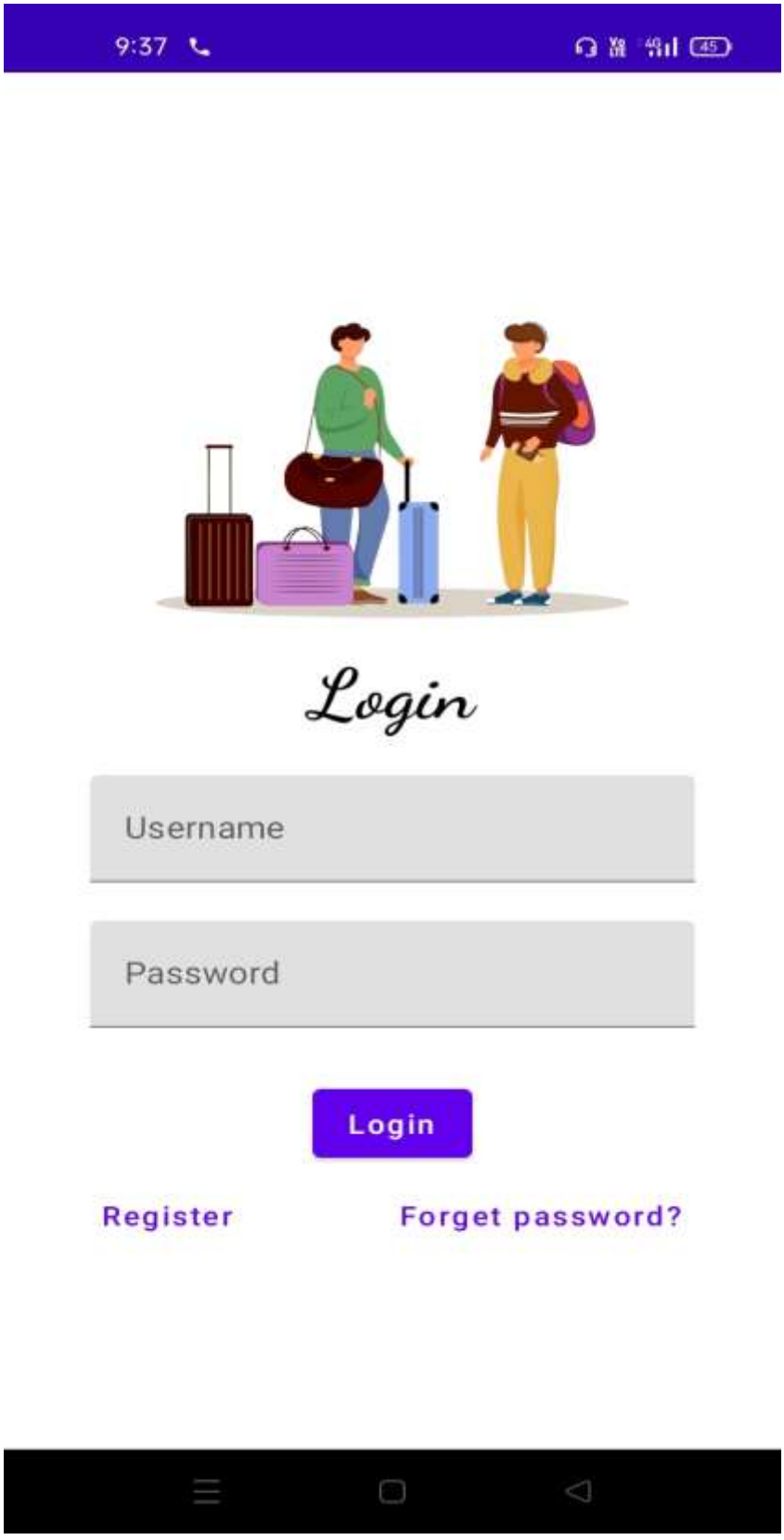
2.2 BRAINSTROMING MAP




saves time
flexible
pre-checking

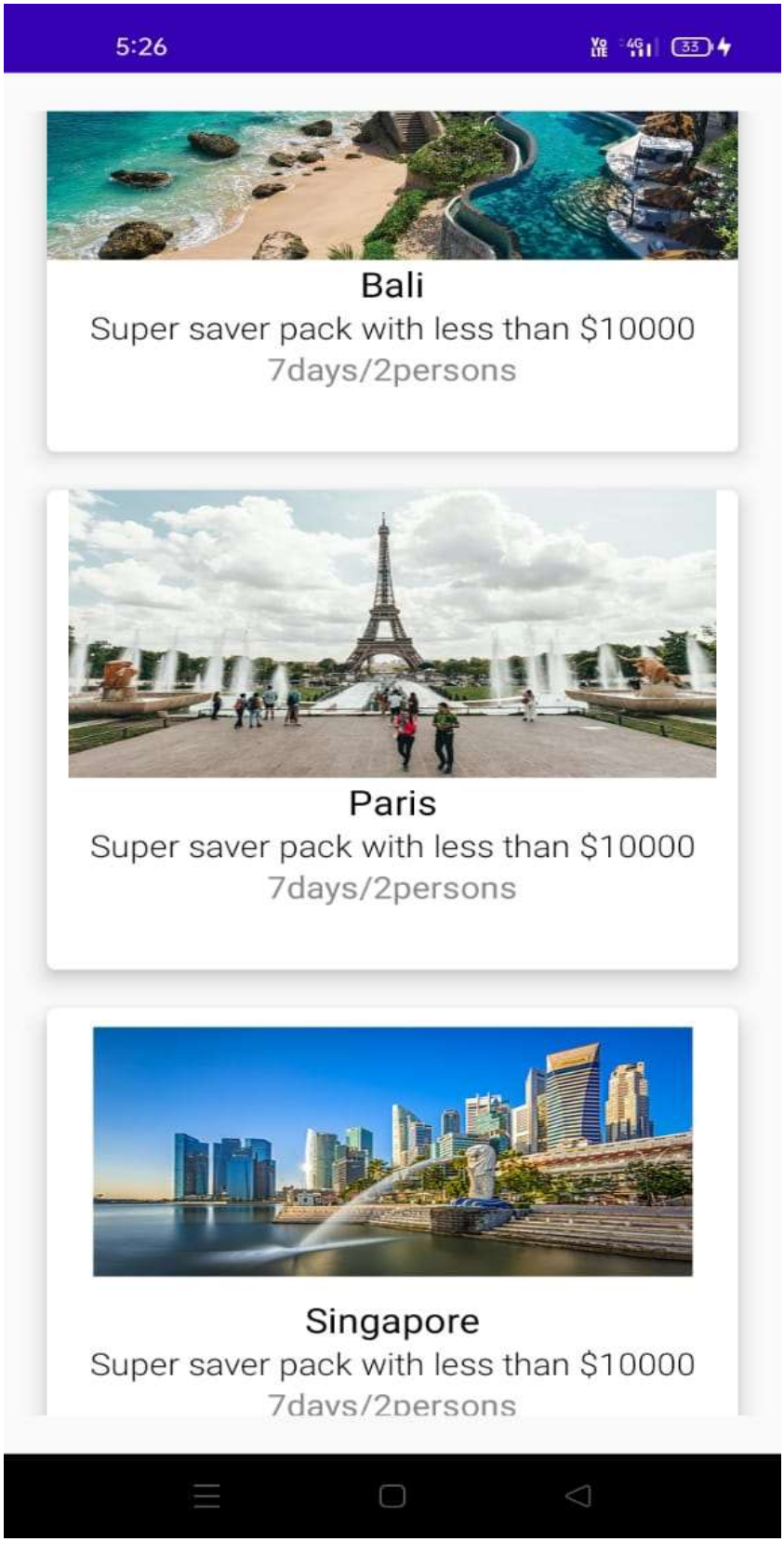

it makes our journey more comfortables and stressfree
necessary to know what the situation and condition to the place where we will visit
planning before travel saves our time

3. RESULT

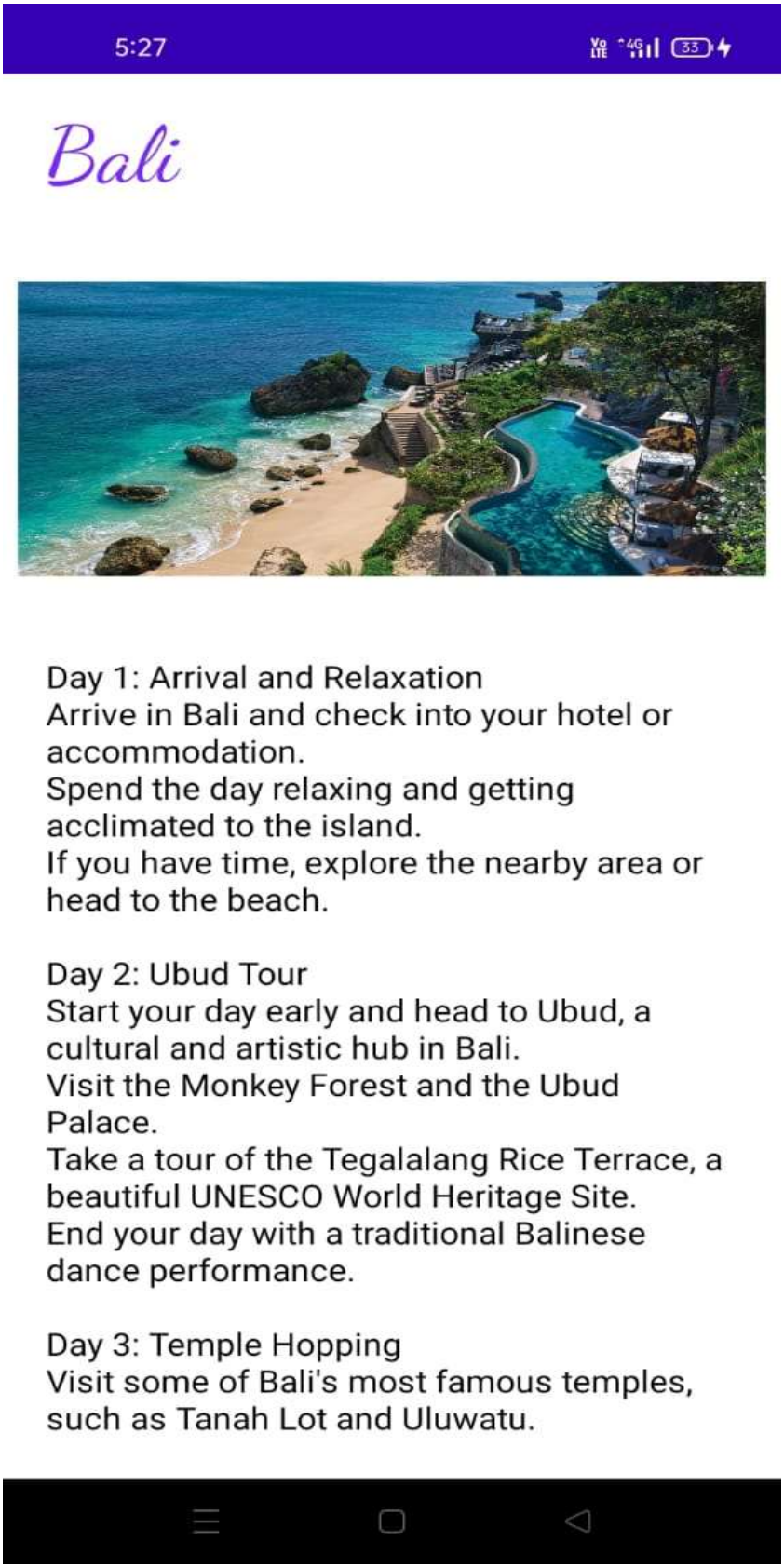


LOGINPAGE

REGISTER PAGE

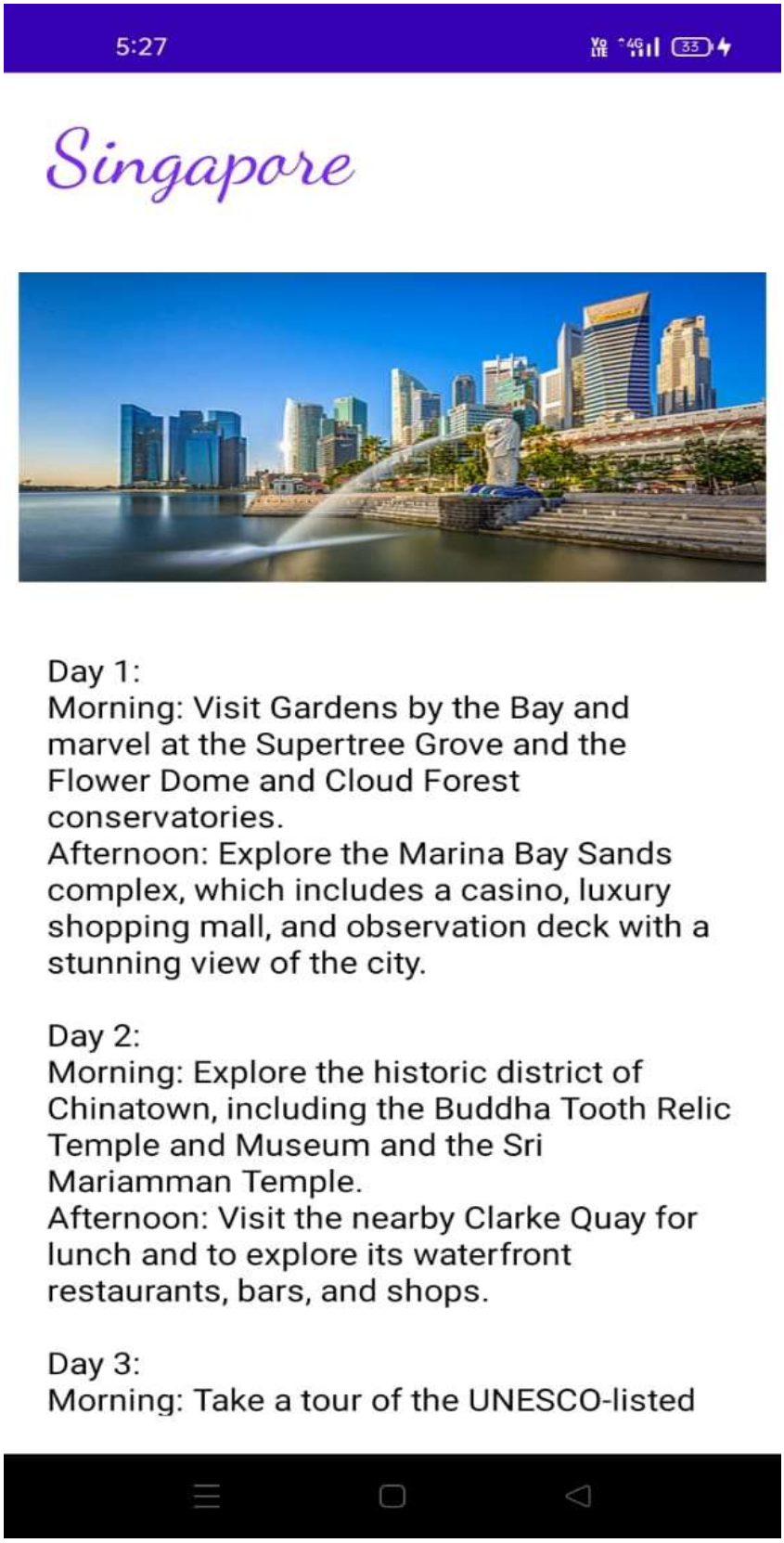

5:26
Bali
Super saver pack with less than $10000
7days/2persons
Paris
Super saver pack with less than $10000
7days/2persons
Singapore
Super saver pack with less than $10000
7days/2persons

MAIN PAGE



BALI LOCATION PAGE

PARIS LOCATION PAGE



SINGAPORE LOCATION PAGE

4. ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

A wanderlust travel and tracking app can offer several advantages to travellers, including

Helps plan and organize trips: A good wanderlust travel and tracking app can help travellers plan and organize their trips by providing information on destinations, flights, accommodations, and activities. It can also help create a travel itinerary, make reservations, and keep track of bookings.

Provides real-time information: A travel app can provide real-time information about flight delays, cancellations, gate changes, and other important updates. This can help travellers stay informed and adjust their plans accordingly.

Offers personalized recommendations: A good travel app can offer personalized recommendations for activities, restaurants, and other attractions based on the traveller's preferences and interests. This can help travellers discover new experiences and make the most of their trip.

Keeps track of expenses: A travel app can help travellers keep track of their expenses by providing a budget tracker or expense log. This can help them stay within their budget and avoid overspending.

Provides safety and security features: A travel app can offer safety and security features such as emergency contacts, location tracking, and safety tips. This can help travellers stay safe and secure while traveling.

DISADVANTAGES:

There are a few potential disadvantages of using a wanderlust travel and tracking app, including:

Dependence on Technology: The app may require a stable internet connection and smartphone, which can be a disadvantage if you are traveling to remote or underdeveloped areas where internet connectivity may be limited or non-existent.

Over-Reliance on Apps: Relying too much on the app to plan your trip may limit your ability to explore and discover new places, as you may only visit destinations that the app recommends.

Privacy Concerns: The app may collect and store personal data, such as your location, travel habits, and preferences, which may be a concern for users who value their privacy.

Cost: Some travel apps require a subscription fee or may charge for premium features, which can be a disadvantage if you are on a tight budget.

User Experience: The app may not be user-friendly or may not have all the features you need, which can be frustrating and limit your ability to plan and enjoy your trip.

Environmental Impact: Using an app to track and plan your travels may contribute to the negative impact of tourism on the environment, such as over-tourism and carbon emissions from travel.

5. APPLICATIONS

There are a few potential disadvantages of using a wanderlust travel and tracking app, including:

Dependence on Technology: The app may require a stable internet connection and smartphone, which can be a disadvantage if you are traveling to remote or underdeveloped areas where internet connectivity may be limited or non-existent.

Over-Reliance on Apps: Relying too much on the app to plan your trip may limit your ability to explore and discover new places, as you may only visit destinations that the app recommends.

Privacy Concerns: The app may collect and store personal data, such as your location, travel habits, and preferences, which may be a concern for users who value their privacy.

Cost: Some travel apps require a subscription fee or may charge for premium features, which can be a disadvantage if you are on a tight budget.

User Experience: The app may not be user-friendly or may not have all the features you need, which can be frustrating and limit your ability to plan and enjoy your trip.

Environmental Impact: Using an app to track and plan your travels may contribute to the negative impact of tourism on the environment, such as over-tourism and carbon emissions from travel.

6. CONCLUSION

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 (e.g., meeting, conference, etc.) 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 SCOPES

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 smartphone 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, Text Image Ciphering and Hybrid Key Distribution Systems.

8.APPENDIX

A. SOURCE CODE

LOGIN ACTIVITY

```
package
com.example.travela
pp
                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.PasswordVisualTransfor
                mation
                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)
}
```

REGISTER ACTIVITY

```
package
com.example.travela
pp
                    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.PasswordVisualTransfor
                    mation
                    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)
}
```

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))
}
}
}
```

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."
)
}
}
```

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 "
)
}
}
```

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.trave
lapp
                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_FIR
ST_NAME)),
lastName =
cursor.getString(cursor.getColumnIndex(COLUMN_LAS
T_NAME)),
email =
cursor.getString(cursor.getColumnIndex(COLUMN_EM
AIL)),
password =
cursor.getString(cursor.getColumnIndex(COLUMN_PAS
SWORD)),
)
}
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_FIR
ST_NAME)),
lastName =
cursor.getString(cursor.getColumnIndex(COLUMN_LAS
T_NAME)),
email =
cursor.getString(cursor.getColumnIndex(COLUMN_EM
AIL)),
password =
cursor.getString(cursor.getColumnIndex(COLUMN_PAS
SWORD)),
)
}
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_FIR
ST_NAME)),
```

```
lastName =
cursor.getString(cursor.getColumnIndex(COLUMN_LAS
T_NAME)),
email =
cursor.getString(cursor.getColumnIndex(COLUMN_EM
AIL)),
password =
cursor.getString(cursor.getColumnIndex(COLUMN_PAS
SWORD)),
)
users.add(user)
} while (cursor.moveToNext())
}
cursor.close()
db.close()
return users
}
}
```