

Wanderlust: A Personalized Travel Planning And Tracking App

1. INTRODUCTION

1.1 OVERVIEW:

In today's fast-paced world, travellers are always looking for ways to simplify and streamline their trip preparations. A travel planner app is a perfect solution for these needs, offering numerous benefits that make the planning process easier and more enjoyable.

The Wanderlust Personal Travel Planner is here to help make our life easier. This printable planner has everything we need to organize and visualize our trip, from booking activities and accommodations to tracking our budget. It's the perfect way to stay organized and enjoy our vacation without any stress.

The app was designed with a clean and modern interface, featuring intuitive icons and buttons that make it easy for users to navigate the app. It was developed using the latest Android development technologies, optimized for performance and security, and tested thoroughly to ensure that it functioned properly on different devices.

1.2 Purpose:

The purpose of Wanderlust, a personalized travel planning and tracking app, is to make travel planning and tracking easier, more organized, and more personalized for users. The app is designed to provide users with a comprehensive solution for all their travel needs, from planning and booking their trip to tracking their expenses and documenting their journey.

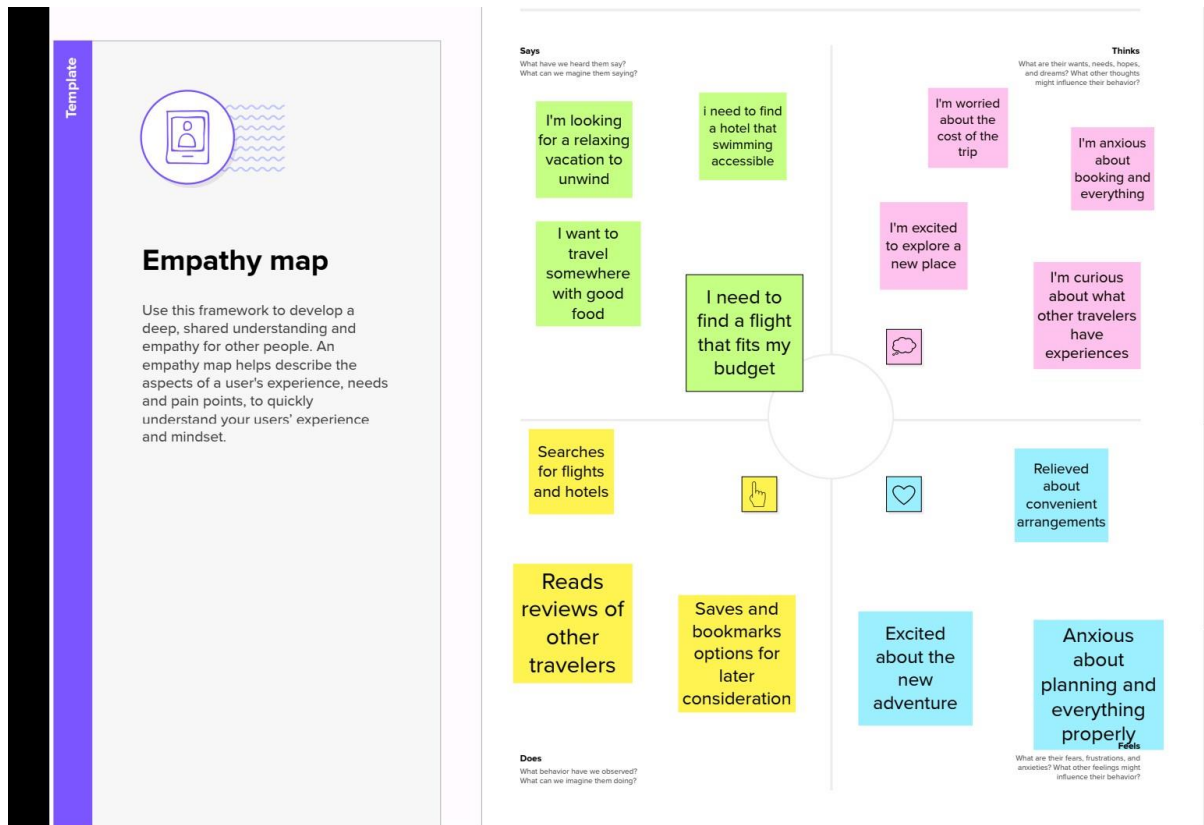
The app's personalized travel recommendations, based on user preferences and travel history, help users discover new destinations and experiences that match their interests and budget. The trip planning tools enable users to organize and manage their entire trip itinerary, including flights, hotels, and activities, all in one place.

Wanderlust also offers real-time travel alerts to keep users informed of any travel disruptions or changes to their itinerary. The expense tracking feature helps users stay on budget by monitoring their expenses throughout the trip. And the travel journal feature allows users to document their journey and share their experiences with friends and family.

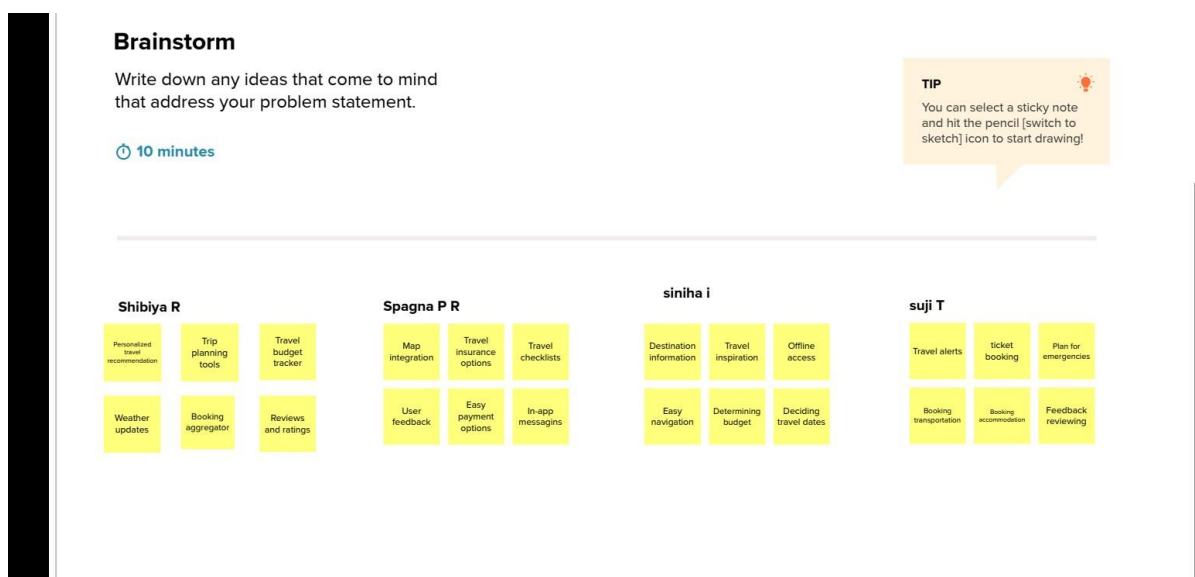
Overall, the purpose of Wanderlust is to provide a seamless and personalized travel planning and tracking experience for users, making it easier and more enjoyable to explore new destinations and create unforgettable memories.

2. PROBLEM DEFINITION AND DESIGN THINKING

2.1 Empathy map:



2.2 Ideation & Brainstorming Map:




3. RESULT

Output:

RegisterActivity & LoginActivity:

9:07

24% 8:05



Register

Username

Email


Password

Register

[Have an account?](#) [Log in](#)

8:05

24% 8:05



Register

Username
shibiya

Email
shibi@gmail.com


Password

Register

[Have an account?](#) [Log in](#)

8:05

24% 8:05



Login

Username
shibiya

Password

Successfully log in

Login

[Register](#) [Forget password?](#)

MainActivity:

Wanderlust Travel



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



Kochi

Super saver pack with Rs.4000
7days/2persons

BaliActivity, ParisActivity & SingaporeActivity:



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 Tegallalang 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

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.

Visit the Orsay Museum, which houses a large collection of impressionist art.

Have dinner at a local French restaurant.

Day 3: French Culture and Food

Visit the Montmartre neighborhood to see the famous Basilique du Sacré-Cœur and Place du Tertre.

Explore the historic neighborhood of Le Marais.

Try some delicious French pastries at a local bakery.

Have dinner at a brasserie to taste some classic French cuisine.

Day 4: Architecture and Gardens

Visit the Palace of Versailles, a UNESCO World Heritage site, and explore its beautiful gardens.

Walk along the Champs-Élysées and stop at the Arc de Triomphe.

Visit the Sainte-Chapelle, a beautiful Gothic chapel with stunning stained-glass windows.

Have dinner at a local restaurant in the 7th arrondissement.

Day 5: Shopping and Sightseeing

Visit the Notre-Dame Cathedral and climb up to the top for a stunning view of the city.

Explore the Latin Quarter and visit the Pantheon.

Go shopping at the famous Galeries Lafayette or Printemps department stores.

Have dinner at a local bistro.

Day 6: Parisian Parks and Museums

Visit the Musée Rodin and explore its beautiful gardens.

Stroll through the Luxembourg Gardens and visit the Luxembourg Palace.

Visit the Centre Pompidou, a modern art museum in the Marais neighborhood.

Have dinner at a local restaurant in the Latin Quarter.



Singapore



Day 1:

Morning: Visit Gardens by the Bay and marvel at the Supertree Grove and the Flower Dome and Cloud Forest conservatories.

Afternoon: Explore the Marina Bay Sands complex, which includes a casino, luxury shopping mall, and observation deck with a stunning view of the city.

Day 2:

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

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

Day 3:

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

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

Day 4:

Morning: Visit the Singapore Zoo and admire the wildlife, including orangutans, tigers, and elephants.

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.

Day 5:

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

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

Day 6:

Morning: Explore the trendy neighborhood of Tiong Bahru, known for its hip cafes and boutiques, as well as its Art Deco architecture.

Afternoon: Visit the National Gallery Singapore, which houses the largest public collection of modern art in Singapore and Southeast Asia.

Day 7:

Morning: Take a day trip to the nearby island of Pulau

KochiActivity.kt:

6:41

4G

Kochi



Kochi (also known as Cochin) is a city in southwest India's coastal Kerala state. It has been a port since 1341, when a flood carved out its harbor and opened it to Arab, Chinese and European merchants. Sites reflecting those influences include Fort Kochi, a settlement with tiled colonial bungalows and diverse houses of worship. Cantilevered Chinese fishing nets, typical of Kochi, have been in use for centuries.

Trip location: Kochi
Destinations covered: 2 Days
Start Point: Folklore Museum
End Point: Wonderla Amusement park

Sightseeing places:
Folklore Museum
Mattancherry palace
Chottanikara Bahavathi Amman Temple
Wonderla Amusement park

Folklore Museum



What's Special: We came across this gem purely by chance - having no plans on a lazy afternoon.. The museum is housed in an architecturally classic building of three floors wherein all the rooms and even the hallways are replete with numerous artifacts ranging from masks to ancient armaments. A must see!!

Mattancherry palace



What's Special: The Mattancherry Palace is a prominent tourist attraction of Kochi. The palace is built in traditional Kerala style, with a sloping tiled roof and a courtyard in the middle. There is a small temple in the courtyard dedicated to the protective goddess of the Kochi Royal family. There are two more temples in the temple

6:43

4G

Chottanikara Bahavathi Amman Temple



What's Special: Chottanikara Bhagavathi Amman is a powerful goddess and generous to her bhaktas.. Negative energy gets cleansed once you enter the temple.. The rituals are unique

Wonderla Amusement park



What's Special: Fun day with family. Wonderla Amusement Park Had an awesome day with my whole family. There were lots of rides for kids too. Recoil ride was really amazing. Better to take fastrack tickets on crowded days. Excellent hygiene and safety standards.



4. ADVANTAGES AND DISADVANTAGES

Advantages:

Wanderlust, a personalized travel planning and tracking app, offers a range of advantages to users, including:

Personalized Travel Recommendations: Wanderlust uses AI and machine learning algorithms to provide personalized travel recommendations based on user preferences and travel history, helping users discover new destinations and experiences that match their interests and budget.

Comprehensive Trip Planning Tools: The app provides users with a range of tools to plan and organize their entire trip itinerary, including flights, hotels, and activities, all in one place.

Real-time Travel Alerts: Wanderlust sends real-time travel alerts to users, such as flight delays or gate changes, helping users avoid potential travel disruptions and stay on top of their itinerary.

Expense Tracking: The app's expense tracking feature enables users to monitor their expenses throughout the trip, helping them stay on budget and avoid overspending.

Travel Journal: The app's travel journal feature allows users to document their journey and share their experiences with friends and family, helping them create lasting memories of their trip.

Time-saving: Wanderlust saves time by eliminating the need to visit multiple websites and apps to plan and organize a trip.

Overall, Wanderlust offers users a convenient, personalized, and comprehensive solution for travel planning and tracking, making it easier and more enjoyable to explore new destinations and create unforgettable memories.

Disadvantages:

While Wanderlust, a personalized travel planning and tracking app, offers many advantages, there are also some potential disadvantages to consider:

App Glitches: Like any app, Wanderlust may experience occasional glitches or technical issues that could impact its usability or accuracy.

Limited Availability: The app may not be available in all countries or regions, limiting its usefulness for some users.

Data Usage: The app requires a stable internet connection, and frequent use of the app may consume significant data, which could result in high data charges.

Dependence on Technology: Some users may prefer more traditional methods of travel planning and tracking, such as using a paper map or journal.

Additional Costs: While the app provides a convenient way to plan and book travel, some users may incur additional fees or charges, such as booking fees or transaction fees.

Overall, the potential disadvantages of Wanderlust are relatively minor compared to the many benefits the app offers. However, users should be aware of these issues and weigh them against the app's advantages when deciding whether to use Wanderlust for their travel planning and tracking needs.

5. APPLICATIONS

Wanderlust, a personalized travel planning and tracking app, has many practical applications for users. Some of the key ways in which users can use Wanderlust include:

Trip Planning: Wanderlust can be used to plan and organize entire trips, from booking flights and hotels to reserving activities and creating an itinerary.

Personalized Travel Recommendations: The app's AI and machine learning algorithms provide personalized travel recommendations,

helping users discover new destinations and experiences that match their interests and budget.

Real-time Travel Alerts: Wanderlust sends real-time travel alerts to users, such as flight delays or gate changes, helping users stay informed and avoid potential travel disruptions.

Expense Tracking: The app's expense tracking feature enables users to monitor their expenses throughout the trip, helping them stay on budget and avoid overspending.

Travel Journal: The app's travel journal feature allows users to document their journey and share their experiences with friends and family, helping them create lasting memories of their trip.

Solo Travel: Wanderlust can be especially useful for solo travelers, providing a convenient and organized way to plan and track their trips.

Group Travel: The app can also be used for group travel, allowing users to collaborate and coordinate with other travelers to plan and organize their trips.

Overall, Wanderlust has a wide range of practical applications for users, from planning and organizing trips to documenting and sharing travel experiences. Whether you're traveling solo or in a group, Wanderlust can help you make the most of your journey.

6. CONCLUSION

Wanderlust, a personalized travel planning and tracking app, offers a comprehensive and personalized solution for users to plan, organize, and track their travels. With its AI and machine learning algorithms, the app provides personalized travel recommendations, helping users discover new destinations and experiences that match their interests and budget.

The app's range of features, including trip planning tools, real-time travel alerts, expense tracking, and a travel journal, make it a convenient and organized way to plan and track your travels. Whether you're traveling solo or in a group, Wanderlust can help you make the most of your journey.

While there may be some potential disadvantages to using the app, such as occasional glitches or data usage, the app's benefits outweigh its potential drawbacks. Overall, Wanderlust is a practical and useful tool for anyone looking to plan and organize their travels and create unforgettable memories.

7. FUTURE SCOPE

The future scope of Wanderlust, a personalized travel planning and tracking app, is promising as the travel industry continues

to grow and evolve. Here are some potential future developments for the app:

Integration with Smart Devices: With the increasing popularity of smart devices like smartwatches and smart speakers, Wanderlust could potentially integrate with these devices, making it even easier for users to access their travel plans and receive real-time travel alerts.

Enhanced AI Capabilities: As AI and machine learning techniques continue to advance, Wanderlust could enhance its algorithms to provide even more personalized travel recommendations based on user preferences and travel history.

Virtual Reality Travel Planning: Virtual reality (VR) technology has the potential to transform the way users plan and experience their travels. Wanderlust could potentially integrate VR technology, allowing users to explore destinations and experiences virtually before they book their trip.

Augmented Reality Travel Tracking: Augmented reality (AR) technology could also enhance the app's travel tracking capabilities, allowing users to view their travel itinerary and real-time travel alerts in an augmented reality environment.

Sustainable Travel Planning: As sustainability becomes an increasingly important factor in travel planning, Wanderlust could

incorporate features that help users choose eco-friendly travel options and reduce their carbon footprint.

Overall, the future scope of Wanderlust is promising as technology continues to evolve and new trends emerge in the travel industry. By adapting to these changes and incorporating new features and technologies, Wanderlust can continue to provide a comprehensive and personalized solution for travel planning and tracking.

8. APPENDIX

A. Source Code:

LoginActivity.kt

```
package com.example.travelapp

import android.content.Context
import android.content.Intent
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.material.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.input.PasswordVisualTransformation
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.core.content.ContextCompat

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



```

        setContent {
            LoginScreen(this, databaseHelper)
        }
    }
}

@Composable
fun LoginScreen(context: Context, databaseHelper: UserDatabaseHelper) {

    var username by remember { mutableStateOf("") }
    var password by remember { mutableStateOf("") }
    var error by remember { mutableStateOf("") }

    Column(
        modifier = Modifier.fillMaxSize().background(Color.White),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {

        Image(painterResource(id = R.drawable.trav), contentDescription =
            "")

        Text(
            fontSize = 36.sp,
            fontWeight = FontWeight.ExtraBold,
            fontFamily = FontFamily.Cursive,
            text = "Login"
        )

        Spacer(modifier = Modifier.height(10.dp))

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

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

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

        Button(
            onClick = {
                if (username.isNotEmpty() && password.isNotEmpty()) {
                    val user = databaseHelper.getUserByUsername(username)
                    if (user != null && user.password == password) {
                        error = "Successfully log in"
                        context.startActivity(

```

```

                Intent(
                    context,
                    MainActivity::class.java
                )
            )
            //onLoginSuccess()
        }
        else {
            error = "Invalid username or password"
        }

        } else {
            error = "Please fill all fields"
        }
    },
    modifier = Modifier.padding(top = 16.dp)
) {
    Text(text = "Login")
}
Row {
    TextButton(onClick = {context.startActivity(
        Intent(
            context,
            RegisterActivity::class.java
        )
    )})
    { Text(text = "Register") }
    TextButton(onClick = {
    })

    {
        Spacer(modifier = Modifier.width(60.dp))
        Text(text = "Forget password?")
    }
}
}
}

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

```

RegisterActivity.kt:

```

package com.example.travelapp

import android.content.Context
import android.content.Intent
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.material.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color

```

```

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

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

@Composable
fun RegistrationScreen(context: Context, databaseHelper:
UserDatabaseHelper) {

    var username by remember { mutableStateOf("") }
    var password by remember { mutableStateOf("") }
    var email by remember { mutableStateOf("") }
    var error by remember { mutableStateOf("") }

    Column(
        modifier = Modifier.fillMaxSize().background(Color.White),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {

        Image(painterResource(id = R.drawable.tra), contentDescription =
        "")

        Text(
            fontSize = 36.sp,
            fontWeight = FontWeight.ExtraBold,
            fontFamily = FontFamily.Cursive,
            text = "Register"
        )

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

        TextField(
            value = email,
            onValueChange = { email = it },
            label = { Text("Email") },

```

```

        modifier = Modifier
            .padding(10.dp)
            .width(280.dp)
    )

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

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

    Button(
        onClick = {
            if (username.isNotEmpty() && password.isNotEmpty() &&
email.isNotEmpty()) {
                val user = User(
                    id = null,
                    firstName = username,
                    lastName = null,
                    email = email,
                    password = password
                )
                databaseHelper.insertUser(user)
                error = "User registered successfully"
                // Start LoginActivity using the current context
                context.startActivity(
                    Intent(
                        context,
                        LoginActivity::class.java
                    )
                )
            } else {
                error = "Please fill all fields"
            }
        },
        modifier = Modifier.padding(top = 16.dp)
    ) {
        Text(text = "Register")
    }
    Spacer(modifier = Modifier.width(10.dp))
    Spacer(modifier = Modifier.height(10.dp))

    Row() {
        Text(
            modifier = Modifier.padding(top = 14.dp), text = "Have an
account?"
        )
    }

```

```

        TextButton(onClick = {
            context.startActivity(
                Intent(
                    context,
                    LoginActivity::class.java
                )
            )
        })

        {
            Spacer(modifier = Modifier.width(10.dp))
            Text(text = "Log in")
        }
    }
}

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

```

MainActivity.kt:

```

package com.example.team12travelapp

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))

Card(
    modifier = Modifier
        .fillMaxWidth()
        .height(250.dp)
        .clickable {
            context.startActivity(
                Intent(context, KochiActivity::class.java)
            )
        },
    elevation = 8.dp
)
{
    Column(
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Image(
            painterResource(id = R.drawable.kochi),
contentDescription = "",
            modifier = Modifier

```

```

                .height(150.dp)
                .scale(scaleX = 1.2F, scaleY = 1F)
            )

            Text(
                text = "Kochi",
                fontSize = 18.sp
            )

            Text(
                text = "Super saver pack with Rs.4000",
                fontWeight = FontWeight.Light,
                fontSize = 16.sp,
                textAlign = TextAlign.Center,
            )

            Text(
                text = "2day/person", color = Color.Gray,
                fontSize = 16.sp
            )
        }
    }
}
}
}
}

```

BaliActivity.kt:

```

package com.example.travelapp

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.MaterialTheme
import androidx.compose.material.Surface
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.scale
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.travelapp.ui.theme.TravelAppTheme

class BaliActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {

```

```

        TravelAppTheme {
            // A surface container using the 'background' color from
the theme
            Surface(
                modifier = Modifier.fillMaxSize(),
                color = MaterialTheme.colors.background
            ) {
                PlaceOne()
            }
        }
    }
}

@Composable
fun PlaceOne() {
    Column(modifier = Modifier.background(color = Color.White)
        .padding(20.dp)
        .verticalScroll(rememberScrollState()))
    {
        Text(
            fontSize = 40.sp,
            color = Color(android.graphics.Color.rgb(120, 40, 251)),
            fontFamily = FontFamily.Cursive,
            text = stringResource(id = R.string.place_1),
        )
        Image(
            painterResource(id = R.drawable.bali), contentDescription = "",
            modifier = Modifier
                .padding(16.dp)
                .fillMaxWidth()
                .height(200.dp)
                .scale(scaleX = 1.2F, scaleY = 1F)
        )
        Text(
            color=Color.Black,
            text = "Day 1: Arrival and Relaxation\n" +
                "Arrive in Bali and check into your hotel or\n" +
                "accommodation.\n" +
                "Spend the day relaxing and getting acclimated to the\n" +
                "island.\n" +
                "If you have time, explore the nearby area or head to\n" +
                "the beach.\n" +
                "\n" +
                "Day 2: Ubud Tour\n" +
                "Start your day early and head to Ubud, a cultural and\n" +
                "artistic hub in Bali.\n" +
                "Visit the Monkey Forest and the Ubud Palace.\n" +
                "Take a tour of the Tegalalang Rice Terrace, a\n" +
                "beautiful UNESCO World Heritage Site.\n" +
                "End your day with a traditional Balinese dance\n" +
                "performance.\n" +
                "\n" +
                "Day 3: Temple Hopping\n" +
                "Visit some of Bali's most famous temples, such as\n" +
                "Tanah Lot and Uluwatu.\n" +
                "Take in the stunning views of the ocean and cliffs.\n"
        )
    }
}

```

```

+
        "Enjoy a sunset dinner at one of the many restaurants
near the temples.\n" +
        "\n" +
        "Day 4: Waterfalls and Beaches\n" +

        "Take a day trip to Bali's beautiful waterfalls, such
as Tegenungan or Gitgit.\n" +
        "Spend the afternoon at one of Bali's world-renowned
beaches, like Seminyak or Nusa Dua.\n" +
        "\n" +
        "Day 5: Island Hopping\n" +

        "Take a day trip to one of Bali's neighboring islands,
such as Nusa Lembongan or Gili Islands.\n" +
        "Snorkel or scuba dive in the clear waters and relax on
the beach.\n" +
        "\n" +
        "Day 6: Cultural Activities\n" +
        "Visit a traditional Balinese village and learn about
the island.\n" +
        "\n" +
        "Day 7: Departure\n" +
        "Explore the surrounding area and take in the stunning
sunset views.\n" +
        "Have dinner at a local restaurant before returning to
your accommodation."
    )
}
}

```

ParisActivity.kt:

```

package com.example.travelapp

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.MaterialTheme
import androidx.compose.material.Surface
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.scale
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.travelapp.ui.theme.TravelAppTheme

```

```

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

@Composable
fun Greeting() {
    Column(
        modifier = Modifier.background(color = Color.White)
            .padding(20.dp)
            .verticalScroll(rememberScrollState())
    ) {
        Text(
            fontSize = 40.sp,
            color = Color(android.graphics.Color.rgb(120, 40, 251)),
            fontFamily = FontFamily.Cursive,
            text = stringResource(id = R.string.place_2),
        )
        Image(
            painterResource(id = R.drawable.paris), contentDescription =
""",
            modifier = Modifier
                .padding(16.dp)
                .fillMaxWidth()
                .height(200.dp)
                .scale(scaleX = 1.2F, scaleY = 1F)
        )
        Text(
            color=Color.Black,
            text = "Day 1: Arrival and Introduction\n" +

                "Check into your accommodation and freshen up\n" +
                "Take a stroll around the neighborhood to get
acquainted\n" +

                "Visit the Eiffel Tower, preferably in the evening when
it is lit up\n" +

                "Have a relaxing dinner at a nearby restaurant\n" +

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

                "Visit the Louvre Museum to see some of the world's
most famous art pieces\n" +
                "Stroll through the Tuileries Garden and the Place de
la Concorde\n" +

                "Visit the Orsay Museum, which houses a large
collection of impressionist art\n" +

```



```

        "Have dinner at a local French restaurant\n" +

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

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

        "\n" +

        "Day 4: Architecture and Gardens\n" +

        "Visit the Palace of Versailles, a UNESCO World
Heritage site, and explore its beautiful gardens\n" +
        "Walk along the Champs-Élysées and stop at the Arc de
Triomphe\n" +
        "Visit the Sainte-Chapelle, a beautiful Gothic chapel
with stunning stained-glass windows\n" +
        "Have dinner at a local restaurant in the 7th
arrondissement\n" +

        "\n" +
        "Day 5: Shopping and Sightseeing\n" +

        "Visit the Notre-Dame Cathedral and climb up to the top
for a stunning view of the city\n" +
        "Explore the Latin Quarter and visit the Panthéon\n" +
        "Go shopping at the famous Galeries Lafayette or
Printemps department stores\n" +
        "Have dinner at a local bistro\n" +
        "\n" +
        "Day 6: Parisian Parks and Museums\n" +

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

        "\n" +
        "Day 7: River Cruise and Farewell\n" +

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

```

SingaporeActivity.kt:

```
package com.example.travelapp

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.MaterialTheme
import androidx.compose.material.Surface
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.scale
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.travelapp.ui.theme.TravelAppTheme

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

@Composable
fun Greeting2() {
    Column(
        modifier = Modifier.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.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
    }

    @SuppressWarnings("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
    }
}

```

AndroidManifest.xml:

```

<?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>

```

KochiActivity.kt:

```

package com.example.team12travelapp

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.team12travelapp.ui.theme.Team12travelappTheme

class KochiActivity : ComponentActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {

        super.onCreate(savedInstanceState)

        setContent {

            Team12travelappTheme {

                // A surface container using the 'background' color from
the theme

                Surface(

                    modifier = Modifier.fillMaxSize(),

                    color = MaterialTheme.colors.background

                ) {

                    Greeting3("Android")

                }

            }

        }

    }

}

@Composable
fun Greeting3(name: String) {

```

```

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 = "Kochi",

    )

    Image(

        painterResource(id = R.drawable.kochi), contentDescription =
        "",

        modifier = Modifier

            .padding(16.dp)

            .fillMaxWidth()

            .height(200.dp)

            .scale(scaleX = 1.2F, scaleY = 1F)

    )

    Text(

        color= Color.Black,

        text ="Kochi (also known as Cochin) is a city in southwest
India's coastal Kerala state. It has been a port since 1341, when a flood
carved out its harbor and opened it to Arab, Chinese and European
merchants. Sites reflecting those influences include Fort Kochi, a
settlement with tiled colonial bungalows and diverse houses of worship.
Cantilevered Chinese fishing nets, typical of Kochi, have been in use for
centuries."

        +"\n"+"\\n"+

        "Trip location: Kochi"+"\\n"+

        "Destinations covered: 2 Days"+"\\n"+

        "Start Point: Folklore Museum"+"\\n"+

        "End Point: Wonderla Amusement park"+"\\n" +"\\n"    +

```

```

        "Sightseeing places:"+"\n"+
        "    Folklore Museum"+"\\n"+
        "    Mattancherry palace"+"\\n"+
        "    Chottanikara Bahavathi Amman Temple"+"\\n"+
        "    Wonderla Amusement park"+"\\n"+"\\n"
    )

    Text(
        fontSize = 35.sp,
        color = Color(android.graphics.Color.rgb(120, 40, 251)),
        fontFamily = FontFamily.Cursive,
        text = "Folklore Museum",
    )

    Image(
        painterResource(id = R.drawable.mus), contentDescription = "",
        modifier = Modifier
            .padding(16.dp)
            .fillMaxWidth()
            .height(200.dp)
            .scale(scaleX = 1.2F, scaleY = 1F)
    )

    Text(
        color= Color.Black,
        text = "\\n"+"What's Special: We came across this gem purely by
chance - having no plans on a lazy afternoon.. The museum is housed in an
architecturally classic building of three floors wherein all the rooms and
even the hallways are replete with numerous artifacts ranging from masks to
ancient armaments. A must see!!"
        +"\\n"+"\\n"
    )

    Text(
        fontSize = 35.sp,
        color = Color(android.graphics.Color.rgb(120, 40, 251)),

```

```

        fontFamily = FontFamily.Cursive,

        text = "Mattancherry palace",

    )

    Image(

        painterResource(id = R.drawable.palace), contentDescription =
""

        modifier = Modifier

            .padding(16.dp)

            .fillMaxWidth()

            .height(200.dp)

            .scale(scaleX = 1.2F, scaleY = 1F)

    )

    Text(

        color= Color.Black,

        text = "\n"+"What's Special: The Mattancherry Palace is a
prominent tourist attraction of Kochi. The palace is built in traditional
Kerala style, with a sloping tiled roof and a courtyard in the middle.
There is a small temple in the courtyard dedicated to the protective
goddess of the Kochi Royal family. There are two more temples in the temple
compound, one dedicated to Lord Shiva and the other to Lord Krishna."

        +"\n"+"

    )

    Text(

        fontSize = 35.sp,

        color = Color(android.graphics.Color.rgb(120, 40, 251)),

        fontFamily = FontFamily.Cursive,

        text = "Chottanikara Bahavathi Amman Temple",

    )

    Image(

        painterResource(id = R.drawable.temple), contentDescription =
""

        modifier = Modifier

            .padding(16.dp)

            .fillMaxWidth()

```

```

        .height(200.dp)

        .scale(scaleX = 1.2F, scaleY = 1F)

    )

    Text(

        color= Color.Black,

        text = "\n"+"What's Special: Chottanikara Bhagavathi Amman is a
powerful goddess and generous to her bhakthas.. Negative energy gets
cleansed once you enter the temple.. The rituals are unique"

        +"\n"+" \n"

    )

    Text(

        fontSize = 35.sp,

        color = Color(android.graphics.Color.rgb(120, 40, 251)),

        fontFamily = FontFamily.Cursive,

        text = "Wonderla Amusement park",

    )

    Image(

        painterResource(id = R.drawable.wonder), contentDescription =
"" ,

        modifier = Modifier

            .padding(16.dp)

            .fillMaxWidth()

            .height(200.dp)

            .scale(scaleX = 1.2F, scaleY = 1F)

    )

    Text(

        color= Color.Black,

        text = "\n"+"What's Special:Fun day with family. Wonderla
Amusement Park Had an awesome day with my whole family.There were lots of
rides for kids too.Recoil ride was really amazing.Better to take fastrack
tickets on crowded days.Excellent hygiene and safety standards."

        +"\n"+" \n"

    )

```

