

Wanderlust: A Personalized Travel Planning and Tracking App

Submitted by.

Lisha P

Manimekala K

Gnana Subitha M

Esther S

Project Report Description

1.INTRODUCTION

- 1.1 Description about your project Overview A brief.
- 1.2 Purpose The use of This project. WWhat can be achieved using this.

2.Problem Definition & Design Thinking

2.1 Empathy Map

Paste the empathy map screenshot

2.2 Ideation & Brainstorming map

Paste the Ideation & brainstorming map screenshot

3.RESULT

Final findings (output) of the project along with screenshots.

4.ADVANTAGES & DISADVANTAGES

List of advantages and disadvantages of the proposed solution

5.APPLICATION

The areas where this solution can be applied

6.CONCLUSION

Conclusion summarizing the entire work and findings

7.FUTURE SCOPE

Enhancements that can be made in the future

1. INTRODUCTION

1.1 Overview:

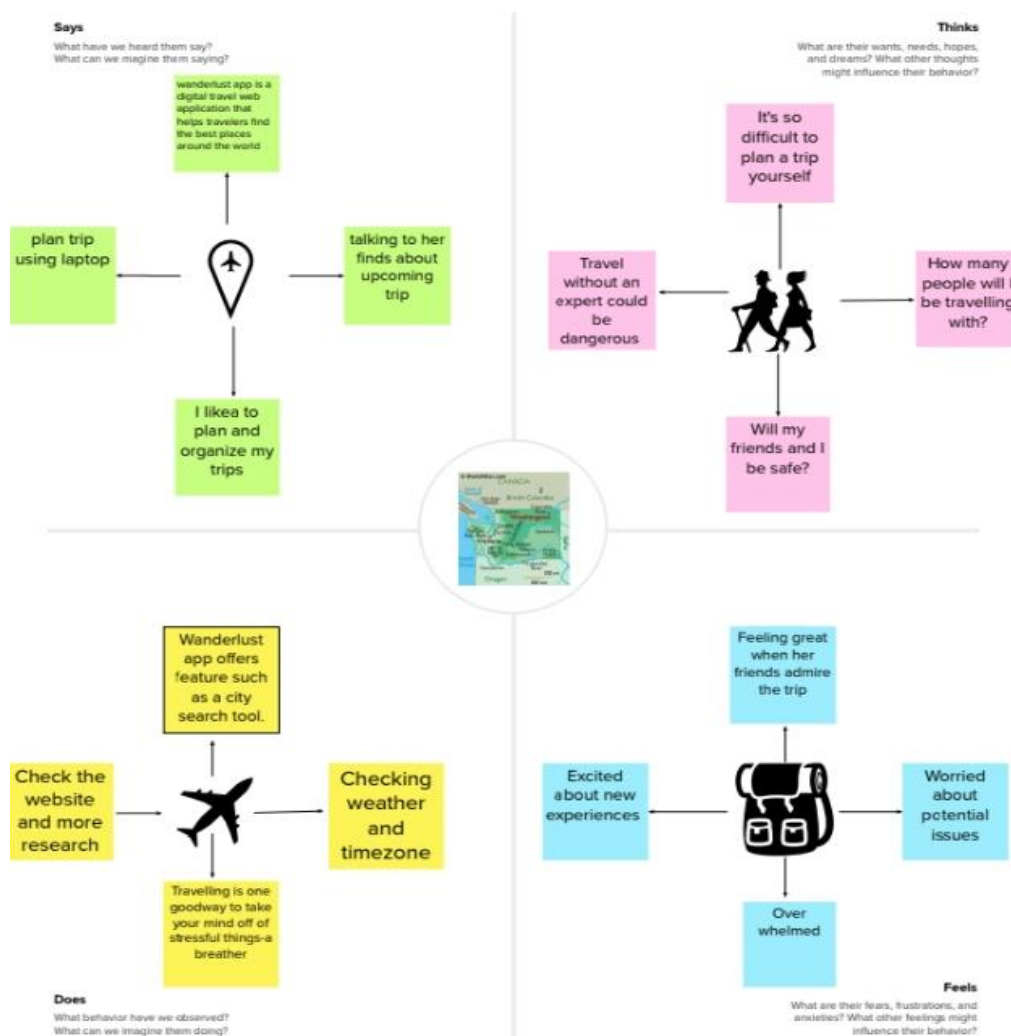
- A travel planning app is an application for booking travel reservations, tracking loyalty points, and browsing travel packages.
- An android application for travellers which allow them to save their journey experiences at one place in an organisable way.
- One can also seek new restaurants to eat at and support local businesses as a tourist in these destinations.
- Travel planner apps, one can research all about the new destinations that don't enjoy the limelight as the most popular tourist places do.

1.2. Purpose:

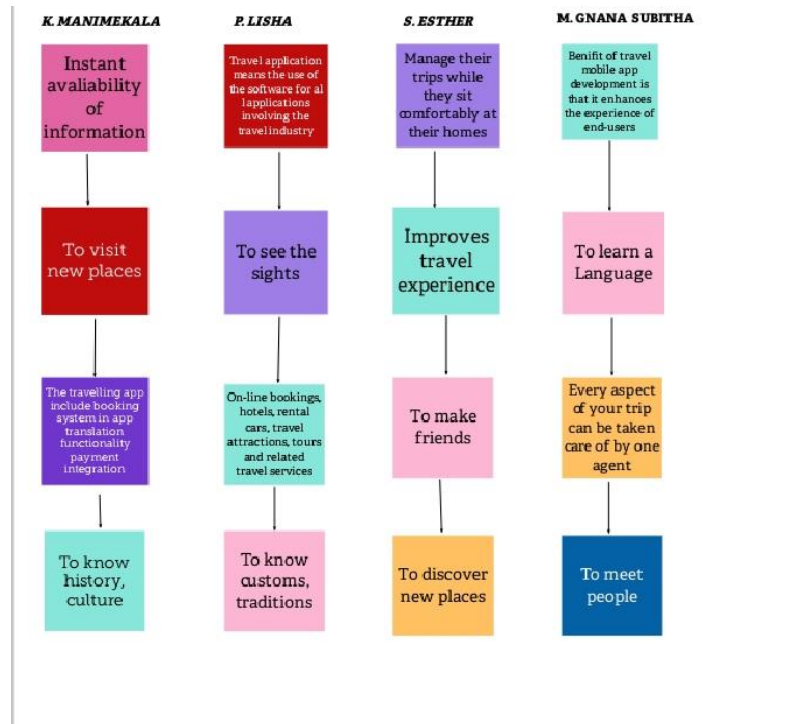
- The easiest way one could stay updated about these regulations was through the regular updates on travel apps.
- Even at a time when travel was restricted, trip-planning apps became the new source of information on the various travel regulations that each country had imposed.
- Along with its benefits during the pandemic, travel planner apps come with great benefits.
- Useful safety information safety is a primary concern if you are planning a recent trip...

2. PROBLEM DEFINITION & DESIGN THINKING

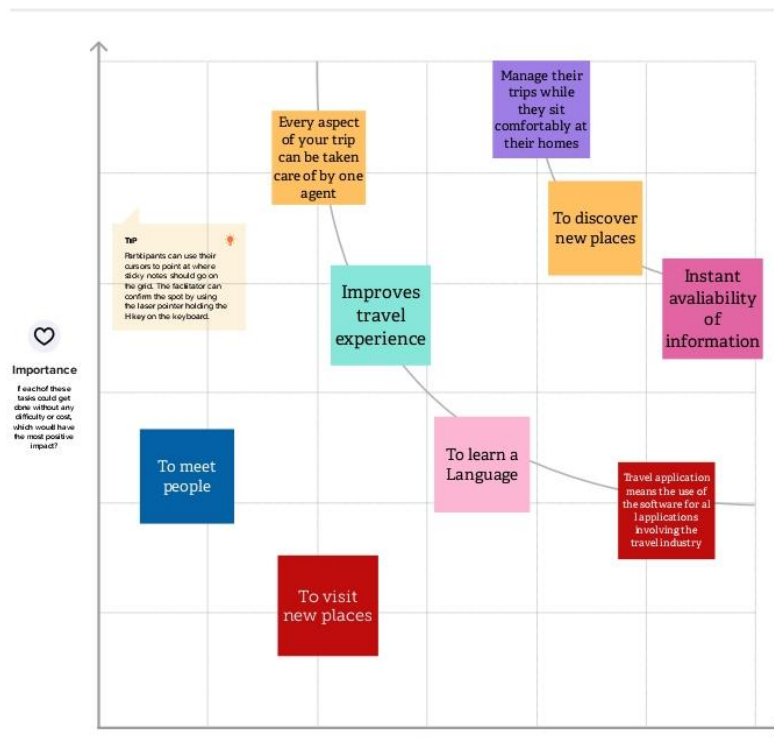
2.1 Empathy Map



2.2 Ideation & Brainstroming Map







3 RESULT



Register

Username

Email

Password

Register

Have an account? [Log in](#)



Login

Username

Password

Login

[Register](#)

[Forget password?](#)

Wanderlust Travel



Bali

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



Paris

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



Singapore

Bali



Day 1: Arrival and Relaxation

Arrive in Bali and check into your hotel or accommodation.

Spend the day relaxing and getting acclimated to the island.

If you have time, explore the nearby area or head to the beach.

Day 2: Ubud Tour

Start your day early and head to Ubud, a cultural and artistic hub in Bali.

Visit the Monkey Forest and the Ubud Palace.

Take a tour of the Tegalalang Rice Terrace, a 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

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 Sacre-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

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 a proboscis monkey and a

4. ADVANTAGES

- **Travel application is one place to get every service regarding travelling.**
- **Tourist places, ticket booking, hotel booking, restaurant detail and transportation booking-you can put all these things at a place by creating travel app.**
- **An app will allow working more in the digital arena and make the tourism company's marketing perfectly!**
- **If users have your travel app on their mobile, you will be able to send the notification regarding your upcoming offer and newly-added services.**
- **For numerous advantages of travel apps, it becoming more popular among travellers day by day.**

DISADVANTAGES

- Assuming always-on data connection. It's not always on.
- When it comes to travelling abroad, roaming forces data networks off for most users.
- The biggest issue for many was price. Travel booking sites aggregate flight, lodging, and other costs in real time across countless airline carriers and hotels.
- While there are many important advantages to world travel, it's not all rainbows and unicorns! You 'll soon become painfully aware that foreign travel for any length of time also has many challenges and some drawbacks.
- Being away from your home country on a big trip for a long time, perhaps in a place where you do not speak the native language can be hard at times.

5 APPLICATIONS

Bus booking:

- Book bus tickets with red bus India .

Car Rentals:

- Partnered with world leading B2B transport solutions.

Flight Booking:

- 950+ Airlines aggregated from multiple OTA and GDS. IATA not mandatory.

Hotel Booking:

- 8,50,000+ inventories aggregated from multiple Bedbanks, OTA and GDS.

6 CONCLUSION

- The availability of travel planning apps helps people navigate through government policies during these difficult times.
- The convenience of booking flight/train tickets, reserving hotel rooms, hiring rental cars, and charting out a travel itinerary from the comfort of one's home on a single app has boosted the travel planner app industry.
- Travel agencies have recognized the merit of travel planning apps and are scouting the best means to learn how to create a travel app.
- The app development tools and case studies will help one have a better understanding of the app they wish to create and the requirements one must consider.

7 FUTURE SCOPE

- The current project gives user its own personalized travel itinerary based on his or her travel interests and point of interests along with hotel stay information.
- For futures 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 places in the world.

8 APPENDIX

A Source Code

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>
```

Main Activity.kt:

```
package com.example.travelapp

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import
androidx.compose.foundation.layout.fillMaxSize
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.tooling.preview.Preview
import
com.example.travelapp.ui.theme.TravelAppThem
e
```

```
class MainActivity : 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("Android")
    }
}
}
}
```

@Composable

```
fun Greeting(name: String) {
    Text(text = "Hello $name!")
}
```

@Preview(showBackground = true)

```
@Composable
fun DefaultPreview() {
    TravelAppTheme {
        Greeting("Android")
    }
}
```

Register Activity.kt:

```
package com.example.travelapp

import android.content.Context
import android.content.Intent
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
```

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

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

@Composable

```
fun RegistrationScreen(context: Context,  
    databaseHelper: UserDatabaseHelper) {  
  
    var username by remember { mutableStateOf("") }  
    var password by remember { mutableStateOf("") }
```

```
var email by remember { mutableStateOf("") }
```

```
var error by remember { mutableStateOf("") }
```

```
Column(
```

```
    modifier =
```

```
    Modifier.fillMaxSize().background(Color.White),
```

```
    horizontalAlignment =
```

```
    Alignment.CenterHorizontally,
```

```
    verticalArrangement = Arrangement.Center
```

```
) {
```

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

```
    Text(
```

```
        fontSize = 36.sp,
```

```
        fontWeight = FontWeight.ExtraBold,
```

```
        fontFamily = FontFamily.Cursive,
```

```
        text = "Register"
```

)

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

TextField(

value = username,

onValueChange = { username = it },

label = { Text("Username") },

modifier = Modifier

.padding(10.dp)

.width(280.dp)

)

TextField(

value = email,

onValueChange = { email = it },

label = { Text("Email") },

modifier = Modifier

```
        .padding(10.dp)
        .width(280.dp)
    )
```

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

```
    if (error.isNotEmpty()) {
        Text(
```

```
        text = error,  
        color = MaterialTheme.colors.error,  
        modifier = Modifier.padding(vertical = 16.dp)  
    )  
}
```

```
Button(  
    onClick = {  
        if (username.isNotEmpty() &&  
password.isNotEmpty() && email.isNotEmpty()) {  
            val user = User(  
                id = null,  
                firstName = username,  
                lastName = null,  
                email = email,  
                password = password  
            )  
            databaseHelper.insertUser(user)
```



```
        error = "User registered successfully"
        // Start LoginActivity using the current
context
        context.startActivity(
            Intent(
                context,
                LoginActivity::class.java
            )
        )

    } else {
        error = "Please fill all fields"
    }
},
    modifier = Modifier.padding(top = 16.dp)
) {
    Text(text = "Register")
}
```

```
Spacer(modifier = Modifier.width(10.dp))
```

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

```
Row() {
```

```
    Text(
```

```
        modifier = Modifier.padding(top = 14.dp), text  
= "Have an account?"
```

```
    )
```

```
    TextButton(onClick = {
```

```
        context.startActivity(
```

```
            Intent(
```

```
                context,
```

```
                LoginActivity::class.java
```

```
            )
```

```
        )
```

```
    })
```

```
{
```

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

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

Login activity.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
import androidx.room.*

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,
        onChange = { username = it },
        label = { Text("Username") },
        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()) {
            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)
}

```

@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)
```

```
}
```