LATEST NEWS HEADLINES

Study Material

Design App

Submit by:

Sheik Mohamed .S

Abdul Rahuman .J

Satheesh .D

Mohideen Meera Sahib .S

An Android Application for keeping up with Latest Headline

INTRODUCTION

Overview:

Our Project is about a update a news to delivery in current time and also delivery a correct news based on a location

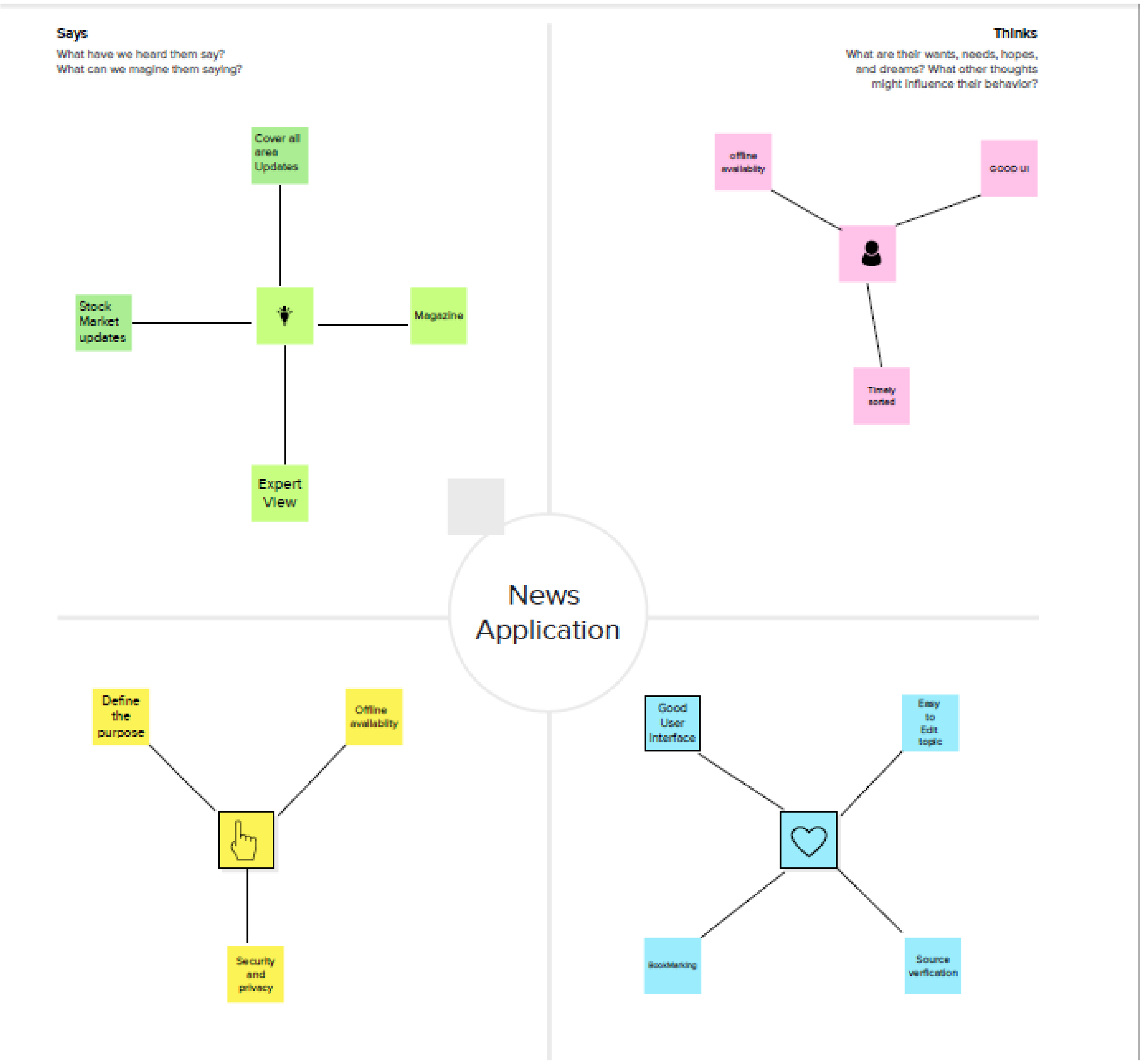
Tag Filter them based on your favourite

Purpose:

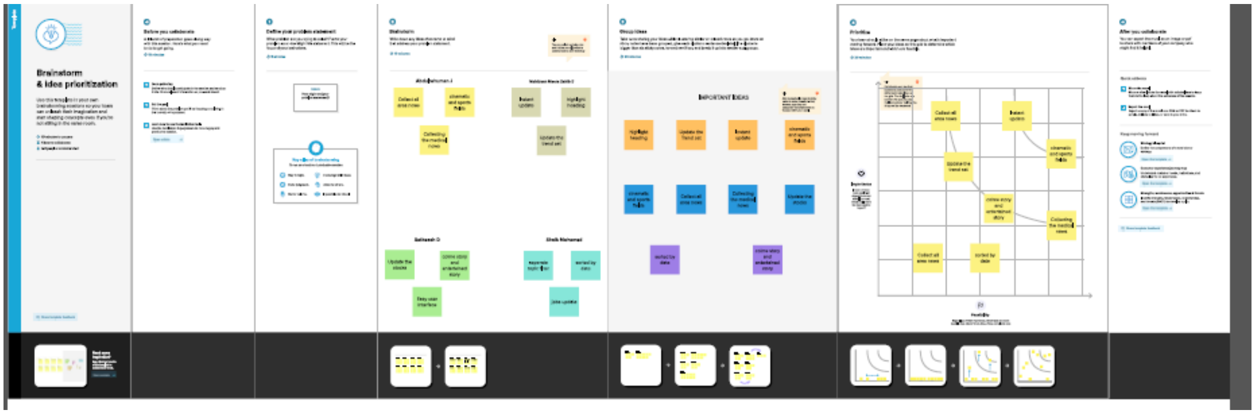
The purpose is delivery the update and correct time and stock market and sports news, foreign news highlight the main topic

Problem Definition &Design Thinking

Empathy map:

Says
What have we heard them say?
What can we magine them saying?
Cover all area Updates
Stock Market updates
Magazine
Expert View
Thinks
What are their wants, needs, hopes, and dreams? What other thoughts might influence their behavior?
offline availability
GOOD UI
Timely sorted
News Application
Define the purpose
Offline availablity
Security and privacy
Good User Interface
Easy to Edit topic
BookMarking
Source verfication

Brainstorming



Brainstorm & idea prioritization

Before you collaborate

Define your problem statement

Brainstorm

Abdul Rahman J

Collect all area news

cinematic and sports fields

Collecting the medical news

Instant update

Highlight heading

Update the trend set

Satheesh D

Update the stocks

comic story and entertained story

Easy user interface

Sheik Mohamad

separate topic filter

sorted by date

Jobs update

Group ideas

IMPORTANT IDEAS

Highlight heading

Update the trend set

Instant update

cinematic and sports fields

cinematic and sports fields

Collect all area news

Collecting the medical news

Update the stocks

sorted by date

comic story and entertained story

Prioritize

Collect all area news

Instant update

cinematic and sports fields

Update the trend set

comic story and entertained story

Colouring the medical news

Collect all area news

sorted by date

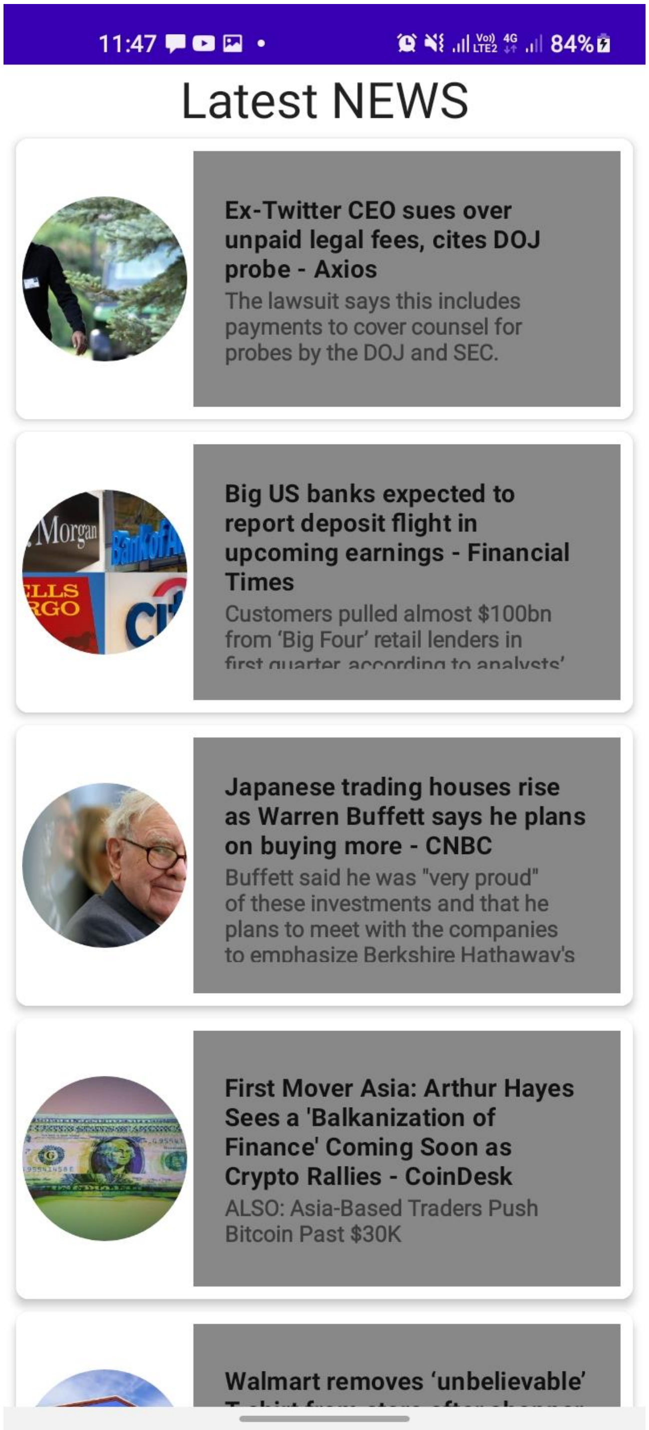
Importance

Feasibility

After you collaborate

Result:

Output of a Application



Latest NEWS

Ex-Twitter CEO sues over unpaid legal fees, cites DOJ probe - Axios
The lawsuit says this includes payments to cover counsel for probes by the DOJ and SEC.

Big US banks expected to report deposit flight in upcoming earnings - Financial Times
Customers pulled almost $100bn from ‘Big Four’ retail lenders in first quarter, according to analysts’

Japanese trading houses rise as Warren Buffett says he plans on buying more - CNBC
Buffett said he was "very proud" of these investments and that he plans to meet with the companies to emphasize Berkshire Hathaway's

First Mover Asia: Arthur Hayes Sees a 'Balkanization of Finance' Coming Soon as Crypto Rallies - CoinDesk
ALSO: Asia-Based Traders Push Bitcoin Past $30K

Walmart removes ‘unbelievable’

ADVANTAGES:

- ➢ Update the news

- Filter based on UX
- Good User Interface

Disadvantages:

- Lot of News
- High quality so data speed is fast
- Not filter the notification

Application:

The app can use on different sector include health ,sports, astrology and also update stock market

Conclusion:

The main focus of this application is to connect news articles from all around the world and derived it to user as fast as possible in best visualize way.

Future scope:

A mobile news app has simple goals capture and repeat the process several times a day. Pretty straight forward .but no that simple in the real world for a news provider the smartphone screen is the most challenging environment ever seen

Source Code:

Android Mainfest.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">
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission
android:name="android.permission.ACCESS_WIFI_STATE"/>
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@drawable/news"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.NewsHeadlines"
        tools:targetApi="31">
        <activity
            android:name=".DisplayNews"
            android:exported="false"
            android:label="@string/title_activity_display_news"
            android:theme="@style/Theme.NewsHeadlines" />
        <activity
            android:name=".RegistrationActivity"
            android:exported="false"
            android:label="@string/title_activity_registration"
            android:theme="@style/Theme.NewsHeadlines" />
        <activity
            android:name=".MainPage"
            android:exported="false"
            android:label="@string/title_activity_main_page"
            android:theme="@style/Theme.NewsHeadlines" />
        <activity
            android:name=".LoginActivity"
            android:exported="true"
            android:label="@string/app_name"
            android:theme="@style/Theme.NewsHeadlines">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

MainActivity.kt:

```
package com.example.newsheadlines

import android.content.Context
import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.widget.TextView
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.viewModels
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.itemsIndexed
import androidx.compose.foundation.selection.selectable
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material.Card
import androidx.compose.material.MaterialTheme
import androidx.compose.material.Surface
import androidx.compose.material.Text
import androidx.compose.runtime.*
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
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
import androidx.compose.ui.viewinterop.AndroidView
import androidx.core.text.HtmlCompat
import coil.compose.rememberImagePainter
import coil.size.Scale
import coil.transform.CircleCropTransformation
import com.example.example.Articles
import com.example.newsheadlines.ui.theme.NewsHeadlinesTheme


class MainPage : ComponentActivity() {
    val mainViewModel by viewModels<MainViewModel>()
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            NewsHeadlinesTheme {
                // A surface container using the 'background' color
from the theme
                Surface(color = MaterialTheme.colors.background)
{
                    Column() {


                        Text(text = "Latest NEWS", fontSize = 32.sp,
modifier = Modifier.fillMaxWidth(), textAlign =
TextAlign.Center)

                        MovieList(applicationContext, movieList =
mainViewModel.movieListResponse)
                        mainViewModel.getMovieList()
                    }
                }
            }
        }
    }
}

@Composable
fun MovieList(context: Context, movieList: List<Articles>) {
    var selectedIndex by remember { mutableStateOf(-1) }
    LazyColumn {

        itemsIndexed(items = movieList) {
                index, item ->
            MovieItem(context,movie = item, index, selectedIndex)
{ i ->
                selectedIndex = i
            }
        }
    }

}

@Composable
fun MovieItem(context: Context) {
    val movie = Articles(
        "Coco",
        "",
        " articl"
    )


    MovieItem(context,movie = movie, 0, 0) { i ->
```

```
        Log.i("wertytest123abc", "MovieItem: "
                +i)
    }
}

@Composable
fun MovieItem(context: Context, movie: Articles, index: Int,
selectedIndex: Int,
              onClick: (Int) -> Unit)
{

    val backgroundColor = if  (index == selectedIndex)
MaterialTheme.colors.primary else
MaterialTheme.colors.background

    Card(
        modifier = Modifier
            .padding(8.dp, 4.dp)
            .fillMaxSize()
            .selectable(true, true, null,
                onClick = {
                    Log.i("test123abc", "MovieItem:
$index/n$selectedIndex")
                })
```

```
      .clickable { onClick(index) }
      .height(180.dp), shape = RoundedCornerShape(8.dp),
elevation = 4.dp
  ) {
    Surface(color = Color.White) {

      Row(
        Modifier
          .padding(4.dp)
          .fillMaxSize()

      )
      {
        Image(
          painter = rememberImagePainter(
            data = movie.urlToImage,
            builder = {
              scale(Scale.FILL)
              placeholder(R.drawable.placeholder)
              transformations(CircleCropTransformation())
            }
          ),
          contentDescription = movie.description,
          modifier = Modifier
            .fillMaxHeight()
            .weight(0.3f)
        )


        Column(
          verticalArrangement = Arrangement.Center,
          modifier = Modifier
            .padding(4.dp)
            .fillMaxHeight()
            .weight(0.8f)
            .background(Color.Gray)
            .padding(20.dp)
            .selectable(true, true, null,
              onClick = {
                Log.i("test123abc", "MovieItem:
$index/n${movie.description}")
                context.startActivity(
                  Intent(context, DisplayNews::class.java)

.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK)
                    .putExtra("desk",
movie.description.toString())
```

```
                                    .putExtra("urlToImage",
movie.urlToImage)
                                    .putExtra("title", movie.title)
                            )
                        })
            ) {

                Text(
                    text = movie.title.toString(),
                    style = MaterialTheme.typography.subtitle1,
                    fontWeight = FontWeight.Bold
                )

                HtmlText(html = movie.description.toString())
            }
        }
    }
}
@Composable
fun HtmlText(html: String, modifier: Modifier = Modifier) {
    AndroidView(
        modifier = modifier
            .fillMaxSize()
```

```
              .size(33.dp),
          factory = { context -> TextView(context) },
          update = { it.text = HtmlCompat.fromHtml(html,
HtmlCompat.FROM_HTML_MODE_COMPACT) }
      )
   }
}
```

LoginActivity.kt:

```
package com.example.newsheadlines

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.foundation.shape.RoundedCornerShape
import androidx.compose.material.*
import androidx.compose.material.icons.Icons
import androidx.compose.material.icons.filled.Lock
import androidx.compose.material.icons.filled.Person
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
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.core.content.ContextCompat.startActivity
import com.example.newsheadlines.ui.theme.NewsHeadlinesTheme

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
            .fillMaxHeight()
            .fillMaxWidth()
            .padding(28.dp),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center)

    {
        Image(
            painter = painterResource(id = R.drawable.news),
            contentDescription = "")
    Spacer(modifier = Modifier.height(10.dp))



    Row {
      Divider(color = Color.LightGray, thickness = 2.dp, modifier =
Modifier
        .width(155.dp)
        .padding(top = 20.dp, end = 20.dp))
      Text(text = "Login",
        color = Color(0xFF6495ED),
        fontWeight = FontWeight.Bold,
        fontSize = 24.sp,style = MaterialTheme.typography.h1)
      Divider(color = Color.LightGray, thickness = 2.dp, modifier =
Modifier
        .width(155.dp)
        .padding(top = 20.dp, start = 20.dp))


    }


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


    TextField(
      value = username,
    onValueChange = { username = it },
    leadingIcon = {
        Icon(
            imageVector = Icons.Default.Person,
            contentDescription = "personIcon",
            tint = Color(0xFF6495ED)
        )
    },
    placeholder = {
        Text(
            text = "username",
            color = Color.Black
        )
    },
    colors = TextFieldDefaults.textFieldColors(
        backgroundColor = Color.Transparent
    )

)



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

```
        TextField(
            value = password,
            onValueChange = { password = it },
            leadingIcon = {
                Icon(
                    imageVector = Icons.Default.Lock,
                    contentDescription = "lockIcon",
                    tint = Color(0xFF6495ED)
                )
            },
            placeholder = { Text(text = "password", color = Color.Black) },
            visualTransformation = PasswordVisualTransformation(),
            colors = TextFieldDefaults.textFieldColors(backgroundColor =
Color.Transparent)
        )



        Spacer(modifier = Modifier.height(12.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,
                        MainPage::class.java
                    )
                )
                //onLoginSuccess()
            } else {
                error = "Invalid username or password"
            }
        } else {
            error = "Please fill all fields"
        }
    },
    shape = RoundedCornerShape(20.dp),
    colors = ButtonDefaults.buttonColors(backgroundColor = Color(0xFF77a2ef)),
    modifier = Modifier.width(200.dp)
    .padding(top = 16.dp)
) {
    Text(text = "Log In", fontWeight = FontWeight.Bold)
}

Row(modifier = Modifier.fillMaxWidth()) {
    TextButton(onClick = {
        context.startActivity(
            Intent(
                context,
                RegistrationActivity::class.java
            ))})
    { Text(text = "Sign up",
        color = Color.Black
    )}

    Spacer(modifier = Modifier.width(100.dp))
```

```
            TextButton(onClick = { /* Do something! */ })
            { Text(text = "Forgot password ?",
                color = Color.Black
            )}
        }




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

RegistrationActivity.kt:

```
package com.example.newsheadlines

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.foundation.shape.RoundedCornerShape
import androidx.compose.material.*
import androidx.compose.material.icons.Icons
import androidx.compose.material.icons.filled.Email
import androidx.compose.material.icons.filled.Lock
import androidx.compose.material.icons.filled.Person
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
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 com.example.newsheadlines.ui.theme.NewsHeadlinesTheme

class RegistrationActivity : 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
            .background(Color.White)
            .fillMaxHeight()
            .fillMaxWidth(),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center)

    {
        Row {
            Text(
                text = "Sign Up",
                color = Color(0xFF6495ED),
                fontWeight = FontWeight.Bold,
                fontSize = 24.sp, style = MaterialTheme.typography.h1
            )
            Divider(
                color = Color.LightGray, thickness = 2.dp, modifier =
Modifier
                    .width(250.dp)
                    .padding(top = 20.dp, start = 10.dp, end = 70.dp)
            )

        }
```

```
Image(
    painter = painterResource(id = R.drawable.sign_up),
    contentDescription = "",
    modifier = Modifier.height(270.dp)
)

TextField(
    value = username,
    onValueChange = { username = it },
    leadingIcon = {
        Icon(
            imageVector = Icons.Default.Person,
            contentDescription = "personIcon",
            tint = Color(0xFF6495ED)
        )
    },
    placeholder = {
        Text(
            text = "username",
            color = Color.Black
        )
    },
        colors = TextFieldDefaults.textFieldColors(
            backgroundColor = Color.Transparent
        )

    )

    Spacer(modifier = Modifier.height(8.dp))

    TextField(
        value = password,
        onValueChange = { password = it },
        leadingIcon = {
            Icon(
                imageVector = Icons.Default.Lock,
                contentDescription = "lockIcon",
                tint = Color(0xFF6495ED)
            )
        },
        placeholder = { Text(text = "password", color = Color.Black) },
        visualTransformation = PasswordVisualTransformation(),
        colors = TextFieldDefaults.textFieldColors(backgroundColor =
Color.Transparent)
    )
```

```
        Spacer(modifier = Modifier.height(16.dp))


        TextField(
            value = email,
            onValueChange = { email = it },
            leadingIcon = {
                Icon(
                    imageVector = Icons.Default.Email,
                    contentDescription = "emailIcon",
                    tint = Color(0xFF6495ED)
                )
            },
            placeholder = { Text(text = "email", color = Color.Black) },
            colors = TextFieldDefaults.textFieldColors(backgroundColor =
Color.Transparent)
        )

        Spacer(modifier = Modifier.height(8.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"
                }
            },
            shape = RoundedCornerShape(20.dp),
            colors = ButtonDefaults.buttonColors(backgroundColor =
Color(0xFF77a2ef)),
            modifier = Modifier.width(200.dp)
                .padding(top = 16.dp)
        ) {
            Text(text = "Register", fontWeight = FontWeight.Bold)
        }

        Row(
            modifier = Modifier.padding(30.dp),
            verticalAlignment = Alignment.CenterVertically,
            horizontalArrangement = Arrangement.Center
        ) {
            Text(text = "Have an account?")

            TextButton(onClick = {
                context.startActivity(
                    Intent(
                        context,
                        LoginActivity::class.java
                    )
                )
            }) {
                Text(text = "Log in",
                    fontWeight = FontWeight.Bold,
                    style = MaterialTheme.typography.subtitle1,
                    color = Color(0xFF4285F4)
                )}

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