1.INTRODUCTION

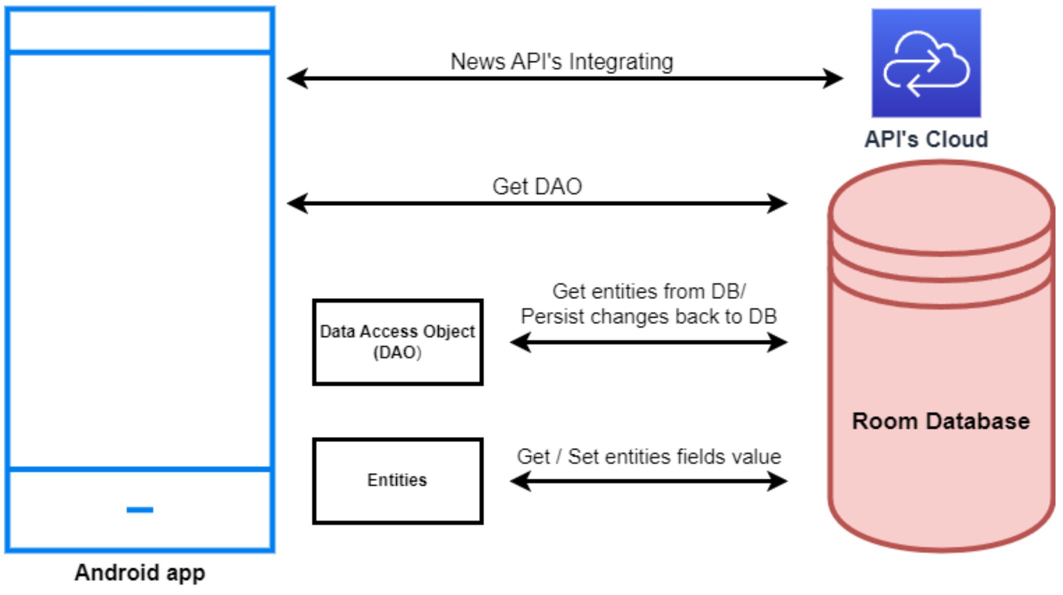
1.1 PROJECT OVERVIEW

AN ANDROID APPLICATION FOR KEEPING UP WITH THE LATEST HEADLINES

Project Description:

The app's main feature is displaying a list of news articles, each with a title, image, and brief description. Users can scroll through the list of articles and tap on an article to view more details. The app uses the Jetpack Compose UI toolkit to build the UI and it uses the coil library to load images. The app fetches data from a remote server using Retrofit library and demonstrates how to use the Jetpack Compose UI toolkit for Android development.

Architecture :



Learning Outcomes :

By end of this project:

- You'll be able to work on Android studio and build an app.

- You'll be able to integrate the database accordingly.
- You'll be able to integrate the API's accordingly.

Project Workflow:

- Users register into the application.
- After registration , user logins into the application.
- User enters into the main page

Note:

To complete the project you need to finish up the tasks listed below:

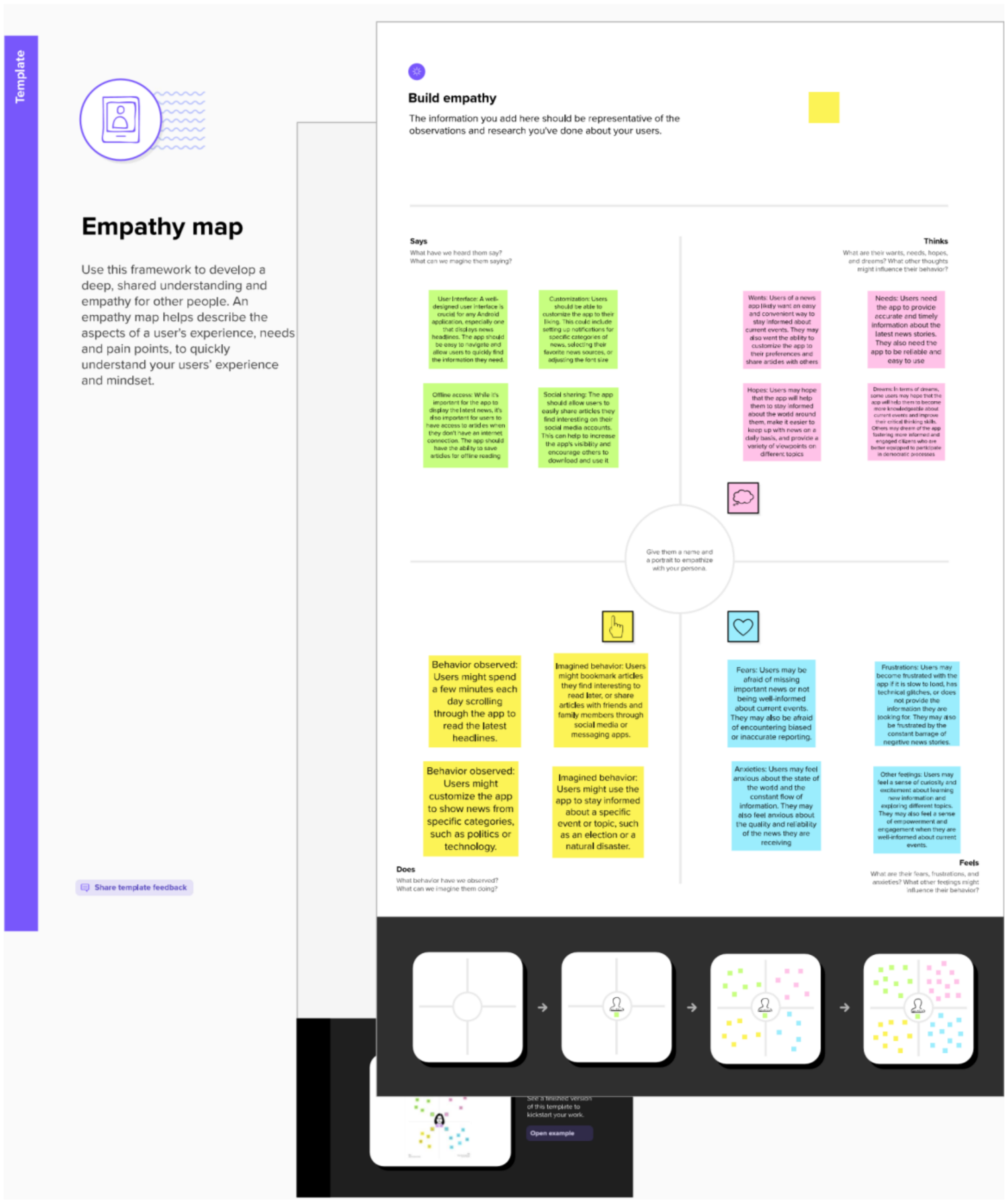
Tasks:

1.Required initial steps
2.Creating a new project.
3.Adding required dependencies.
4.Adding permissions
5.Creating the database classes.
6.Creating API Service and required classes for integrating API
7.Building application UI and connecting to database.
8.Modifying AndroidManifest.xml
9.Running the application.

1.2 PURPOSE OF PROJECT :

The purpose of an Android application for keeping up with the latest headlines is to provide users with a convenient and efficient way to stay informed about current events and news stories. With so much information available online, it can be challenging to keep up with the latest news and events happening around the world.

2-PROBLEM DEFINATION AND DESIGN THINKING:



Empathy map

Use this framework to develop a deep, shared understanding and empathy for other people. An empathy map helps describe the aspects of a user's experience, needs and pain points, to quickly understand your users' experience and mindset.

Build empathy

The information you add here should be representative of the observations and research you've done about your users.

Says

What have we heard them say?
What can we magine them saying?

User Interface: A well-designed user interface is crucial for any Android application, especially one that displays news headlines. The app should be easy to navigate and allow users to quickly find the information they need.

Customization: Users should be able to customize the app to their liking. This could include setting up notifications for specific categories of news, selecting their favorite news sources, or adjusting the font size

Offline access: While it's important for the app to display the latest news, it's also important for users to have access to articles when they don't have an internet connection. The app should have the ability to save articles for offline reading

Social sharing: The app should allow users to easily share articles they find interesting on their social media accounts. This can help to increase the app's visibility and encourage others to download and use it

Thinks

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

Wants: Users of a news app likely want an easy and convenient way to stay informed about current events. They may also went the ability to customize the app to their preferences and share articles with others

Needs: Users need the app to provide accurate and timely information about the latest news stories. They also need the app to be reliable and easy to use

Hopes: Users may hope that the app will help them to stay informed about the world around them, make it easier to keep up with news on a daily basis, and provide a variety of viewpoints on different topics

Dreams: In terms of dreams, some users may hope that the app will help them to become more knowledgeable about current events and improve their critical thinking skills. Others may dream of the app fostering more informed and engaged citizens who are better equipped to participate in democratic processes

Give them a name and a portrait to empathize with your persona.

Behavior observed: Users might spend a few minutes each day scrolling through the app to read the latest headlines.

Imagined behavior: Users might bookmark articles they find interesting to read later, or share articles with friends and family members through social media or messaging apps.

Behavior observed: Users might customize the app to show news from specific categories, such as politics or technology.

Imagined behavior: Users might use the app to stay informed about a specific event or topic, such as an election or a natural disaster.

Does

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

Fears: Users may be afraid of missing important news or not being well-informed about current events. They may also be afraid of encountering biased or inaccurate reporting.

Frustrations: Users may become frustrated with the app if it is slow to load, has technical glitches, or does not provide the information they are looking for. They may also be frustrated by the constant barrage of negative news stories.

Anxieties: Users may feel anxious about the state of the world and the constant flow of information. They may also feel anxious about the quality and reliability of the news they are receiving

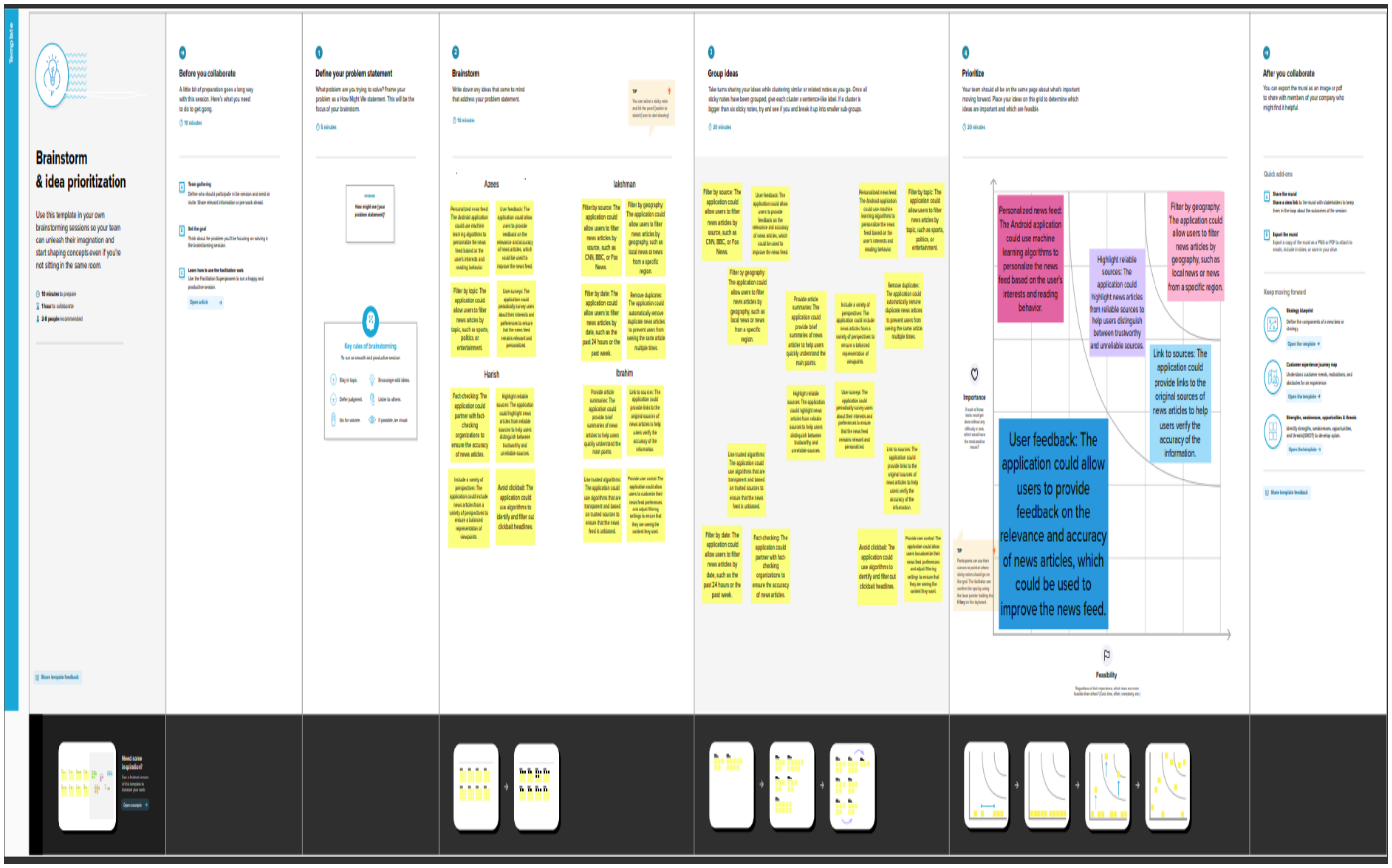
Other feelings: Users may feel a sense of curiosity and excitement about learning new information and exploring different topics. They may also feel a sense of empowerment and engagement when they are well-informed about current events.

Feels

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

See a finished version of this template to kickstart your work.

2.2 IDEATION AND BRAINSTORM:



Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

Azees

Personalized news feed: The Android application could use machine learning algorithms to personalize the news feed based on the user's interests and reading behavior.

User feedback: The application could allow users to provide feedback on the relevance and accuracy of news articles, which could be used to improve the news feed.

Filter by topic: The application could allow users to filter news articles by topic, such as sports, politics, or entertainment.

User surveys: The application could periodically survey users about their interests and preferences to ensure that the news feed remains relevant and personalized.

lakshman

Filter by source: The application could allow users to filter news articles by source, such as CNN, BBC, or Fox News.

Filter by geography: The application could allow users to filter news articles by geography, such as local news or news from a specific region.

Filter by date: The application could allow users to filter news articles by date, such as the past 24 hours or the past week.

Remove duplicates: The application could automatically remove duplicate news articles to prevent users from seeing the same article multiple times.

Harish

Fact-checking: The application could partner with fact-checking organizations to ensure the accuracy of news articles.

Highlight reliable sources: The application could highlight news articles from reliable sources to help users distinguish between trustworthy and unreliable sources.

Include a variety of perspectives: The application could include news articles from a variety of perspectives to ensure a balanced representation of viewpoints.

Avoid clickbait: The application could use algorithms to identify and filter out clickbait headlines.

Ibrahim

Provide article summaries: The application could provide brief summaries of news articles to help users quickly understand the main points.

Link to sources: The application could provide links to the original sources of news articles to help users verify the accuracy of the information.

Use trusted algorithms: The application could use algorithms that are transparent and based on trusted sources to ensure that the news feed is unbiased.

Provide user control: The application could allow users to customize their news feed preferences and adjust filtering settings to ensure that they are seeing the content they want.

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Personalized news feed: The Android application could use machine learning algorithms to personalize the news feed based on the user's interests and reading behavior.

Highlight reliable sources: The application could highlight news articles from reliable sources to help users distinguish between trustworthy and unreliable sources.

Filter by geography: The application could allow users to filter news articles by geography, such as local news or news from a specific region.

Link to sources: The application could provide links to the original sources of news articles to help users verify the accuracy of the information.

User feedback: The application could allow users to provide feedback on the relevance and accuracy of news articles, which could be used to improve the news feed.

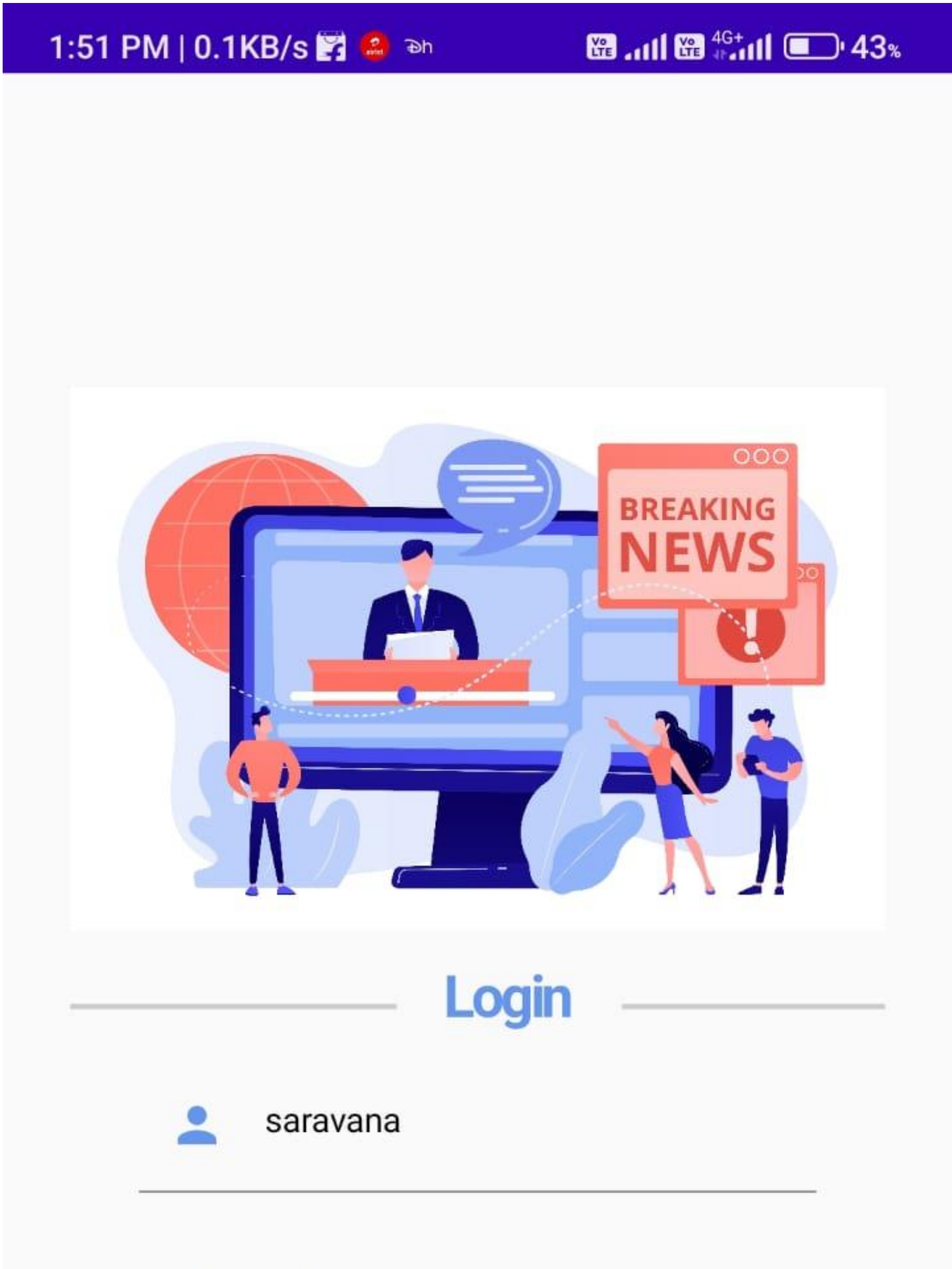
3-RESULT:

1:51 PM | 0.0KB/s 43%

Sign Up

username

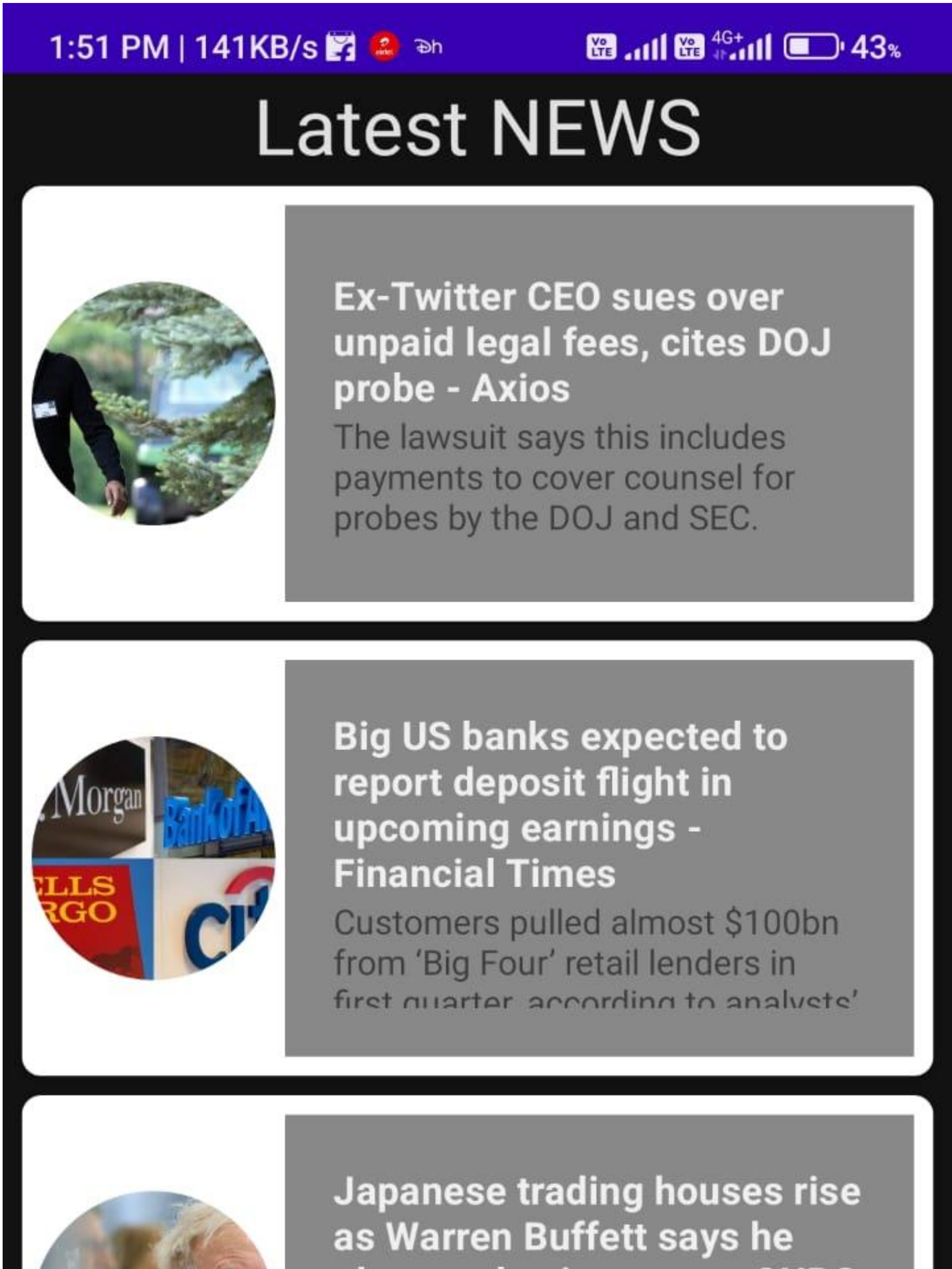
password


1:51 PM | 0.1KB/s
43%
BREAKING NEWS
Login
saravana

1:51 PM | 98.9KB/s 43%

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.



1:51 PM | 141KB/s 43%

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

4-ADVANTAGE AND DISADVANTAGE:

ADVANTAGES OF AN ANDROID APPLICATION FOR KEEPING UP WITH THE LATEST HEADLINES :

Convenience: With an Android app for news, you can get the latest headlines and news updates anytime and anywhere, without having to go to a website or turn on the TV.

Customization: Many Android news apps allow you to customize the types of news you receive, so you can get news that is relevant to your interests and preferences.

Speed: News apps are designed to deliver news quickly and efficiently, so you can get the latest updates as soon as they are available.

Notifications: With a news app, you can receive notifications for breaking news and important stories, so you never miss an important update

DISADVANTAGES AN ANDROID APPLICATION FOR KEEPING UP WITH THE LATEST HEADLINES :

Battery usage: Constantly receiving news updates can be a drain on your device's battery life, particularly if you have notifications turned on for each news update.

Storage space: Depending on the app, news articles can take up a significant amount of storage space on your device. This can be particularly problematic if you have limited storage available.

Privacy concerns: Many news apps require access to your location, browsing history, and other personal data in order to provide personalized news updates. This can raise privacy concerns for some users.

Information overload: With so many news articles available, it can be easy to become overwhelmed by the amount of information being presented. This can make it difficult to stay focused on the most important news stories.

5-APPLICATION:

An Android application for keeping up with the latest headlines can be a useful tool for anyone who wants to stay informed about current events. Here are some features that could be included in such an application:

The application should allow users to select the news sources and topics that interest them. This could be achieved through a simple interface that lets users browse and select from a list of available sources and topics.

The application should be able to send push notifications to users when new articles are published that match their selected topics or sources. This could be done using a simple algorithm that scans the user's feed and sends notifications for the most relevant articles.

The application should allow users to download articles for offline reading. This could be useful for users who want to read the news while commuting or when they don't have an internet connection.

6-CONCLUTION

In conclusion, developing an Android application for keeping up with the latest headlines is a smart investment in the current digital age. With an ever-increasing demand for news and information, such an application can offer a convenient and user-friendly way for people to stay up-to-date with the latest events from around the world.

To make the application a success, it is crucial to focus on providing a seamless user experience, incorporating features like customization, notifications, and easy navigation. By incorporating these key elements, the application can become an essential tool for users, providing them with a reliable and efficient way to stay informed about the latest news and current events.

7-FUTURE SCOPE:

An Android application for keeping up with the latest headlines can have several future scopes, such as:

Personalization: The application can use machine learning algorithms to understand the user's interests and preferences and personalize the news feed accordingly. This will enhance the user experience and keep them engaged with the application.

Multimedia Content: The application can include multimedia content like videos, podcasts, and infographics to enhance the user's engagement with the news.

Social Integration: The application can allow users to share news articles on social media platforms, comment on articles, and engage in discussions with other users. This can enhance the user's social experience and also attract new users to the application.

Augmented Reality: The application can use augmented reality (AR) technology to create immersive experiences for users, such as live events coverage, 360-degree videos, and interactive maps.

APPENDIX:

A.SOURCE CODE

MainPage.kt

```
package com.example.newsheadlines


import android.content.Context
import android.content.Intent
import android.content.Intent.FLAG_ACTIVITY_NEW_TASK
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)
}
```