

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

1.INTRODUCTION

1.1 OVERVIEW:

A news headlines app is a mobile application that provides users with a quick and easy way to stay up-to-date on current events. The app typically features a user-friendly interface that allows users to browse and read headlines from a variety of sources, including newspapers, news websites, and news agencies.

The purpose of a news headlines app is to provide users with an easy way to access breaking news and stay informed about important events happening around the world. Users can personalize their experience by selecting their preferred news sources, topics, and categories, which helps to ensure that they receive news that is relevant and interesting to them.

The app may also include additional features, such as the ability to save articles for later, share articles on social media, receive push notifications for breaking news, and customize the app's appearance to suit their preferences.

Overall, a news headlines app is a valuable tool for anyone who wants to stay informed about current events and news happening around the world, in a fast and convenient way.

2.2 PURPOSE:

The purpose of a news headlines app is to provide users with a quick and easy way to stay up-to-date on current events and news happening around the world.

The app allows users to access breaking news and headlines from a variety of sources, including newspapers, news websites, and news agencies, all in one place. This saves users the time and effort of having to visit multiple websites or read multiple newspapers to stay informed.

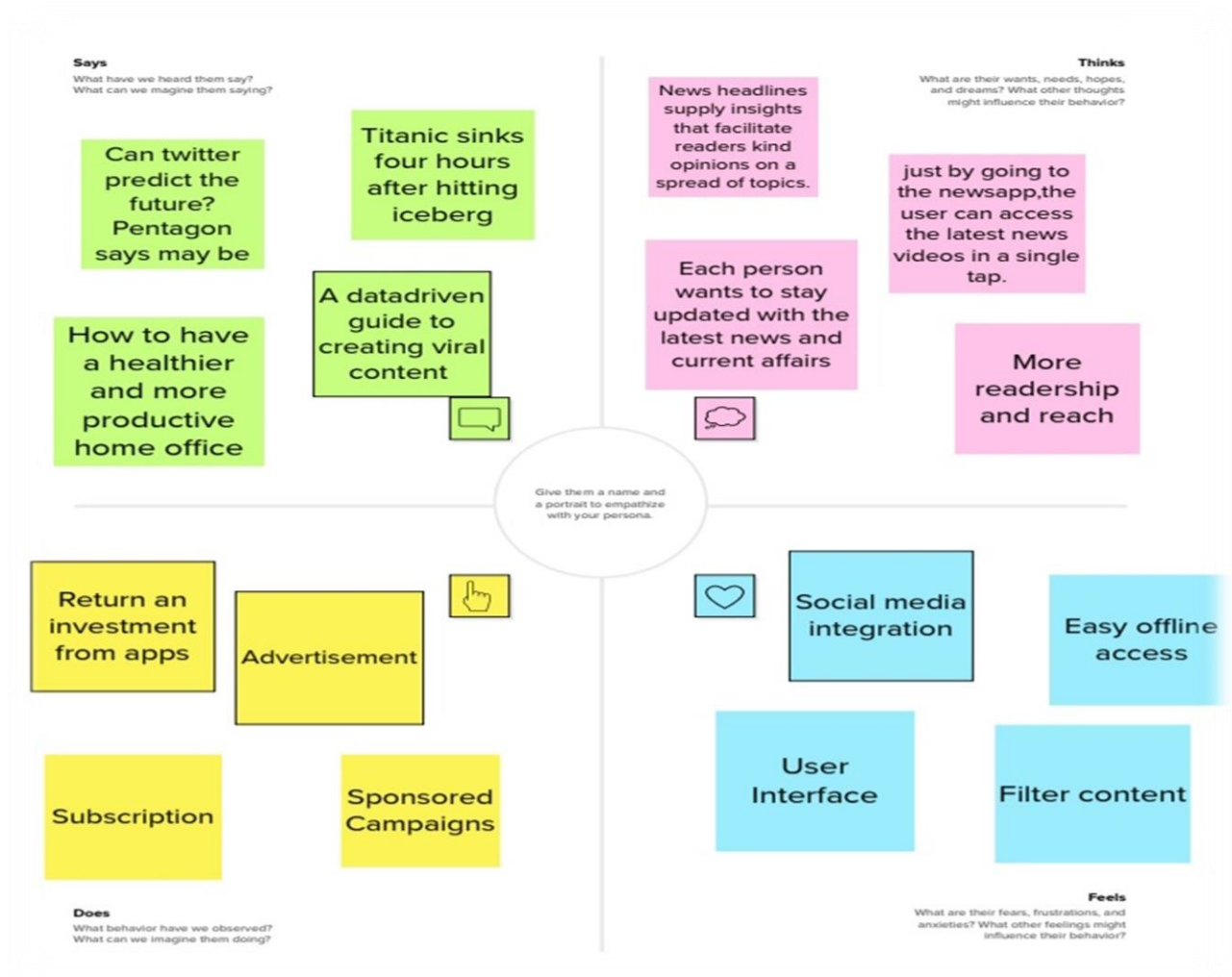
In addition, a news headlines app allows users to personalize their experience by selecting their preferred news sources, topics, and categories. This ensures that users receive news that is relevant and interesting to them, which can help them stay engaged and informed about the world around them.

Furthermore, a news headlines app can provide users with additional features, such as the ability to save articles for later, share articles on social media, receive push notifications for breaking news, and customize the app's appearance to suit their preferences.

Overall, the purpose of a news headlines app is to provide a convenient and customizable way for users to stay informed about current events and news happening around the world, in a fast and efficient manner.

2. PROBLEM DEFINITION & DESIGN THINKING

2.1 Empathy map:



2.2 Brainstrom:

JASPER

User interface	Filter content	Easy offline access
Premium content	Strengthens reading and writing skills	Provides entertainment and sports news
Improves vocabulary skills	Makes a good speaker	Get up to date with politics

JANEESH

Best source of general knowledge	Useful ideas for researches and projects	News apps are popular
People prefer mobile apps to websites	News app offers offer a better user experience	News app offers more personalization
News app nurtures engagement	News apps are easy to browse	News app are faster

JASWIN

News app is easy to monetize	stories, images, videos, ads, pop-ups	"digestible" feeds and chunks of headlines and stories.
Pick and save their favorites	Submit their own content	Access exclusive stories
Earn loyalty points and rewards	a news app allows you to collect data about the user's interests and behavior	people have a habit of using news apps to get their daily dose of information.

KRISHNA DEV


41% of mobile users' digital time is dedicated to news apps.	news app is a significant investment	Mobile apps are more convenient
Good user interface	Push notifications	Simple navigation
Access podcast	Change is needed and publishers must embrace mobile.	user can access to the content offline


3.RESULT


OUTPUT:


REGISTER ACTIVITY & LOGIN ACTIVITY:

Sign Up




 username

 password


 email


Register

Have an account? [Log in](#)



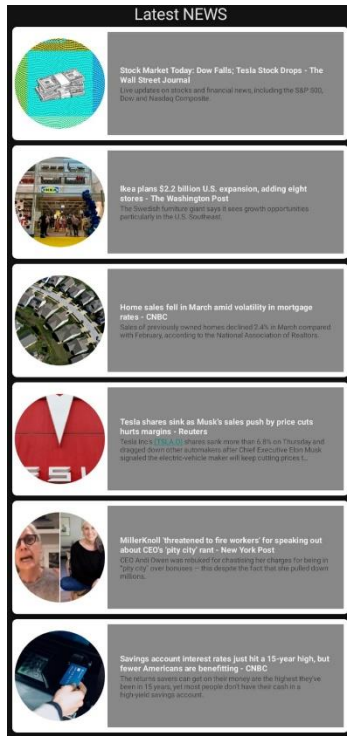
Login

 username

 password

Log In

[Sign up](#) [Forgot password ?](#)



4.ADVANTAGES AND DISADVANTAGES:

ADVANTAGES:

There are several advantages to using a news headlines app:

1. **Convenience:** One of the main advantages of a news headlines app is that it provides a convenient way to access news and current events. Users can easily browse and read headlines from a variety of sources, all in one place, which saves them time and effort.
2. **Personalization:** A news headlines app allows users to customize their experience by selecting their preferred news sources, topics, and categories. This ensures that

users receive news that is relevant and interesting to them, which can help them stay engaged and informed about the world around them.

3. Accessibility: A news headlines app is accessible to anyone with a smartphone or tablet, which means that users can stay informed about news and events from anywhere, at any time.
4. Timeliness: News headlines apps often provide users with real-time updates and push notifications for breaking news, which means that users can stay up-to-date on current events as they happen.
5. Cost-effective: Many news headlines apps are free to download and use, which makes them a cost-effective way to stay informed about news and events.

Overall, a news headlines app provides a convenient, personalized, and cost-effective way to stay informed about current events and news happening around the world.

DISADVANTAGES:

While there are several advantages to using a news headlines app, there are also some potential disadvantages that users should be aware of:

1. **Bias:** Many news headlines apps allow users to select their preferred news sources, topics, and categories. However, this can sometimes lead to a bias in the news that users receive, as they may only be exposed to news that aligns with their existing beliefs and opinions.
2. **Information overload:** A news headlines app can provide users with a lot of information, which can sometimes be overwhelming. Users may find it difficult to filter through all the headlines and articles to find the information that is most relevant and important to them.
3. **Accuracy:** While news headlines apps strive to provide accurate and reliable information, there is always the potential for inaccuracies and errors in reporting. Users should always verify the information they receive through multiple sources to ensure its accuracy.
4. **Distraction:** News headlines apps can be distracting, as users may be tempted to constantly check for updates and breaking news. This can be a particular problem for individuals who need to focus on work or other important tasks.
5. **Privacy concerns:** Some news headlines apps may collect user data, such as location and browsing history, which can raise privacy concerns for users who are sensitive about their personal information.

Overall, while news headlines apps provide a convenient and personalized way to stay informed about current events, users should also be aware of the potential biases, information overload, accuracy concerns, distraction, and privacy issues that may arise.

5. APPLICATIONS

News headlines have a wide range of applications, including:

1. **Staying informed:** The primary application of a news headlines app is to provide users with a convenient way to stay informed about current events and news happening around the world. The app allows users to browse and read headlines from a variety of sources, all in one place, which saves time and effort.
2. **Customization:** A news headlines app allows users to customize their experience by selecting their preferred news sources, topics, and categories. This ensures that users receive news that is relevant and interesting to them, which can help them stay engaged and informed about the world around them.
3. **Breaking news alerts:** Many news headlines apps provide users with real-time updates and push notifications for breaking news. This allows users to stay up-to-date on

current events as they happen, which can be particularly useful during emergencies or other rapidly evolving situations.

4. Saving articles: A news headlines app often allows users to save articles for later, which can be useful for individuals who want to read an article in-depth but do not have the time to do so immediately.
5. Sharing news: News headlines apps often allow users to share articles on social media, which can be useful for individuals who want to share news and current events with their friends and followers.

Overall, a news headlines app provides a range of applications that can be useful for individuals who want to stay informed and engaged with the world around them.

6. CONCLUSION

In conclusion, a news headlines app provides a convenient and personalized way for users to stay informed about current events and news happening around the world. The app allows users to access breaking news and headlines from a variety of sources, all in one place, which saves time and effort.

Users can customize their experience by selecting their preferred news sources, topics, and categories, which ensures that they receive news that is relevant and interesting to them. News headlines apps often provide real-time updates and push notifications for breaking news, which allows users to stay up-to-date on current events as they happen.

While there are potential drawbacks to using a news headlines app, such as bias and information overload, the benefits of staying informed and engaged with the world around us make a news headlines app a valuable tool for anyone who wants to stay informed about the latest news and current events.

7. FUTURE SCOPE

The future scope for a news headlines app is promising, as the demand for up-to-date and personalized news continues to grow. Here are some potential future developments for news headlines apps:

1. Artificial Intelligence (AI) integration: With the integration of AI, news headlines apps could provide more personalized news to users by analyzing their reading history and preferences. AI could also help to combat fake news and provide fact-checking in real-time.
2. Augmented Reality (AR) and Virtual Reality (VR) integration: AR and VR could enhance the user experience

of a news headlines app by providing a more immersive and interactive way of consuming news. For example, users could virtually visit the scene of a breaking news story or attend a virtual press conference.

3. Social media integration: News headlines apps could integrate with social media platforms to provide users with news stories that are trending or have gone viral. This would allow users to stay informed about news that is being discussed by their social media network.
4. Podcasts and audio news: The popularity of podcasts and audio news continues to grow, and news headlines apps could incorporate audio news features to cater to users who prefer to consume news through audio channels.
5. Localization: News headlines apps could incorporate localization features to provide users with news that is specific to their location. This could include local news stories, weather updates, and traffic information.

Overall, the future scope for news headlines apps is exciting, with the potential for increased personalization, interactivity, and localization. As technology continues to advance, news headlines apps will continue to evolve to meet the changing needs and preferences of users.

8. APPENDIX

Manifest.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_app_icon"
        android:label="@string/app_name"
        android:supportRtl="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>
```

DisplayNews.kt :

```
package com.example.newsheadlines
```

```

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.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.padding
import androidx.compose.material.MaterialTheme
import androidx.compose.material.Surface
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Preview
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 com.example.newsheadlines.ui.theme.NewsHeadlinesTheme

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

                    var desk = getIntent().getStringExtra("desk")
                    var title = getIntent().getStringExtra("title")
                    var uriImage = getIntent().getStringExtra("urlToImage")
                    Log.i("test123abc", "MovieItem: $desk")

                    Column(modifier.background(Color.Gray).padding(20.dp),
horizontalAlignment = Alignment.CenterHorizontally, verticalArrangement =
Arrangement.Center) {
                        Text(text = ""+title, fontSize = 32.sp)
                        HtmlText(html = desk.toString())
                        /* AsyncImage(
                            model = "https://example.com/image.jpg",
                            contentDescription = "Translated description of
what the image contains"
                        ) */
                        Image(
                            painter = rememberImagePainter(uriImage),
                            contentDescription = "My content description",

```

```

    )
    }
    // Greeting(desk.toString())
}
}
}
}

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

@Preview(showBackground = true)
@Composable
fun DefaultPreview() {
    NewsHeadlinesTheme {
        // Greeting("Android")
    }
}

@Composable
fun HtmlText(html: String, modifier: Modifier = Modifier) {
    AndroidView(
        modifier = modifier,
        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,
    onChange = { 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,
    onChange = { 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)
}

```

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",
        "",
        " articlel"
    )

    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) }
    )
}
}
}

```

Model.kt :

```

package com.example.newsheadlines

data class Movie(val name: String,
                 val imageUrl: String,
                 val desc: String,
                 val category: String)

```

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,
        onChange = { 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,
        onChange = { 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,
        onChange = { 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)
}
```