**CONTENTS**

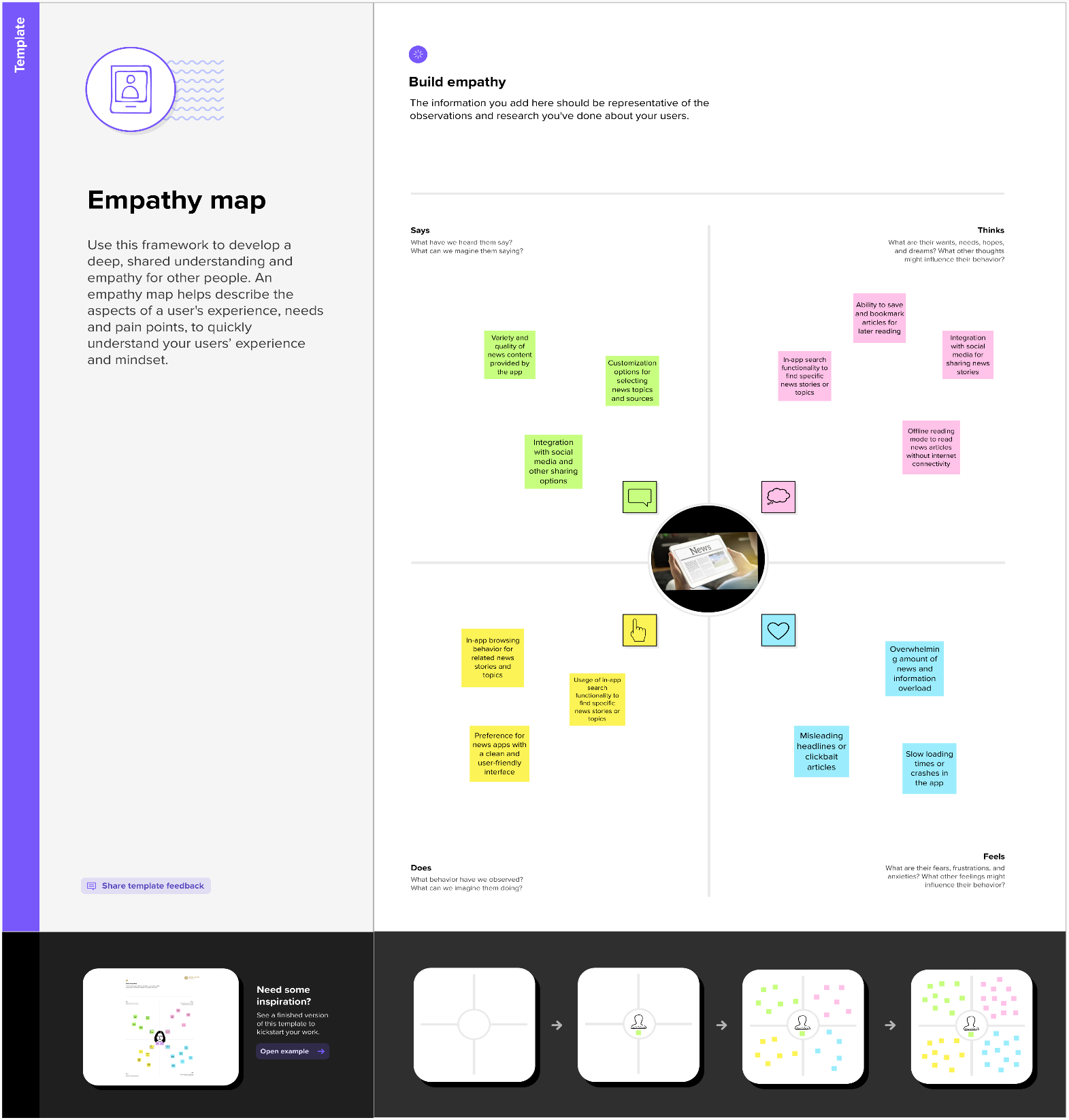
1. **INTRODUCTION** 
   1. Overview
   2. Purpose
2. **PROBLEM DEFINITION & DESIGN THINKING**
   1. Empathy Map
   2. Ideation & Brainstorming Map
3. **RESULT**
4. **ADVANTAGES & DISADVANTAGES**
5. **APPLICATIONS**
6. **CONCLUSION**
7. **FUTURE SCOPE**
8. **APPENDIX**
   1. Source Code

**INTRODUCTION**

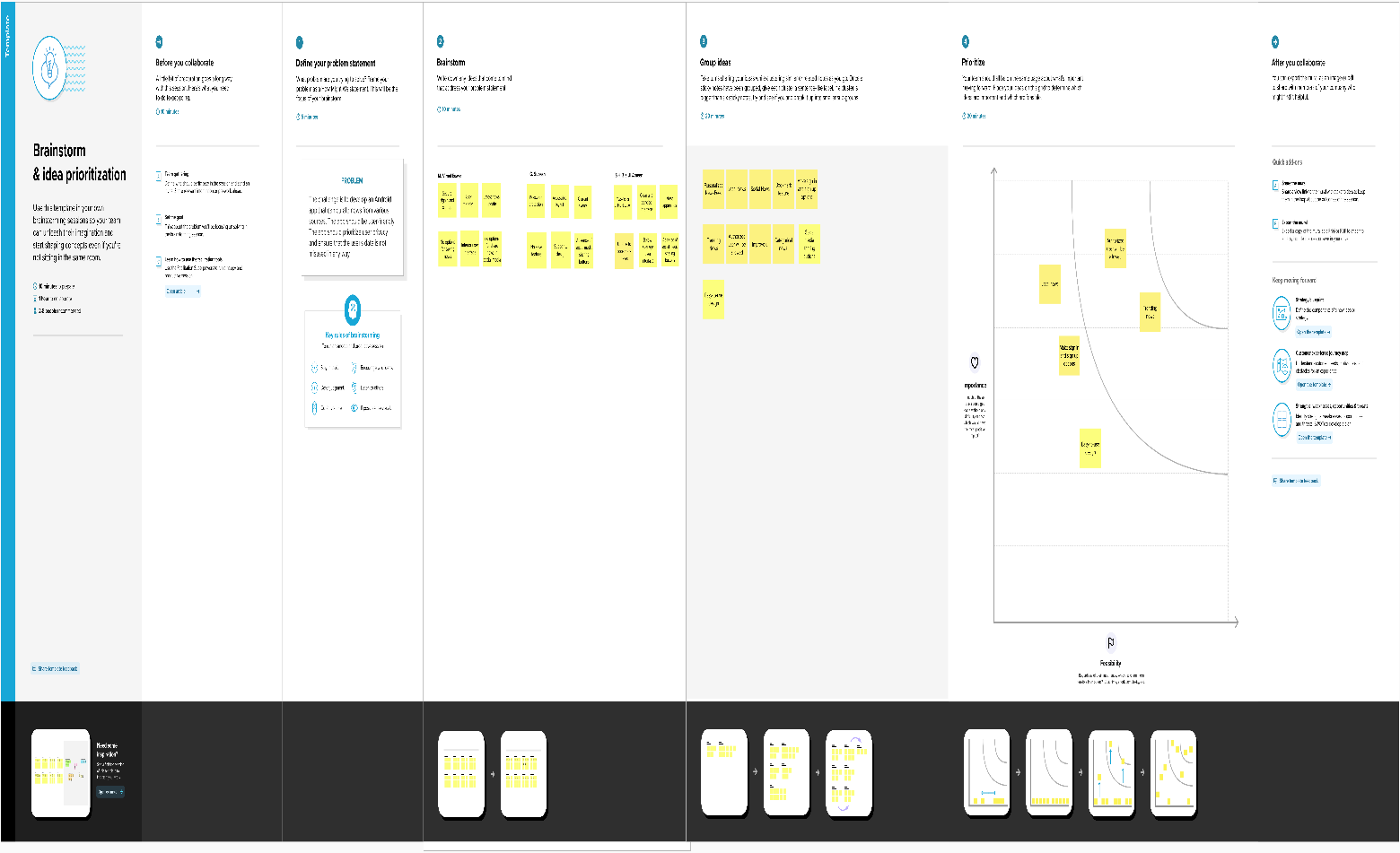
The main objective of News App is to provide the users with easy access to the latest news from a variety of reliable sources for user-friendly and efficient manner. This app should aim to offer a clean and organized interface, with a focus on delivering news articles quickly and efficiently.

* 1. **Overview:**
* Providing a personalized experience: The News app should allow users to customize their news feed based on their interests, location, or other preferences.
* Offering real-time updates: This app should be updated frequently to ensure that users have access to the latest news as it happens.
* Ensuring accuracy and reliability: It should only source news from credible and reliable sources to ensure that users receive accurate and trustworthy information.
* Enhancing engagement: This app should provide users with the ability to interact with news stories by commenting, sharing, or reacting to articles.
  1. **Purpose:**
* Providing users with a convenient and efficient way to access news articles from different sources.
* Delivering news stories in real-time to ensure users stay up-to-date on current events.
* Providing a platform for users to engage with news stories, share articles with others, and provide feedback or comments.
* Offering a platform for publishers to distribute their content and reach a wider audience.

**PROBLEM DEFINITION & DESIGN THINKING**

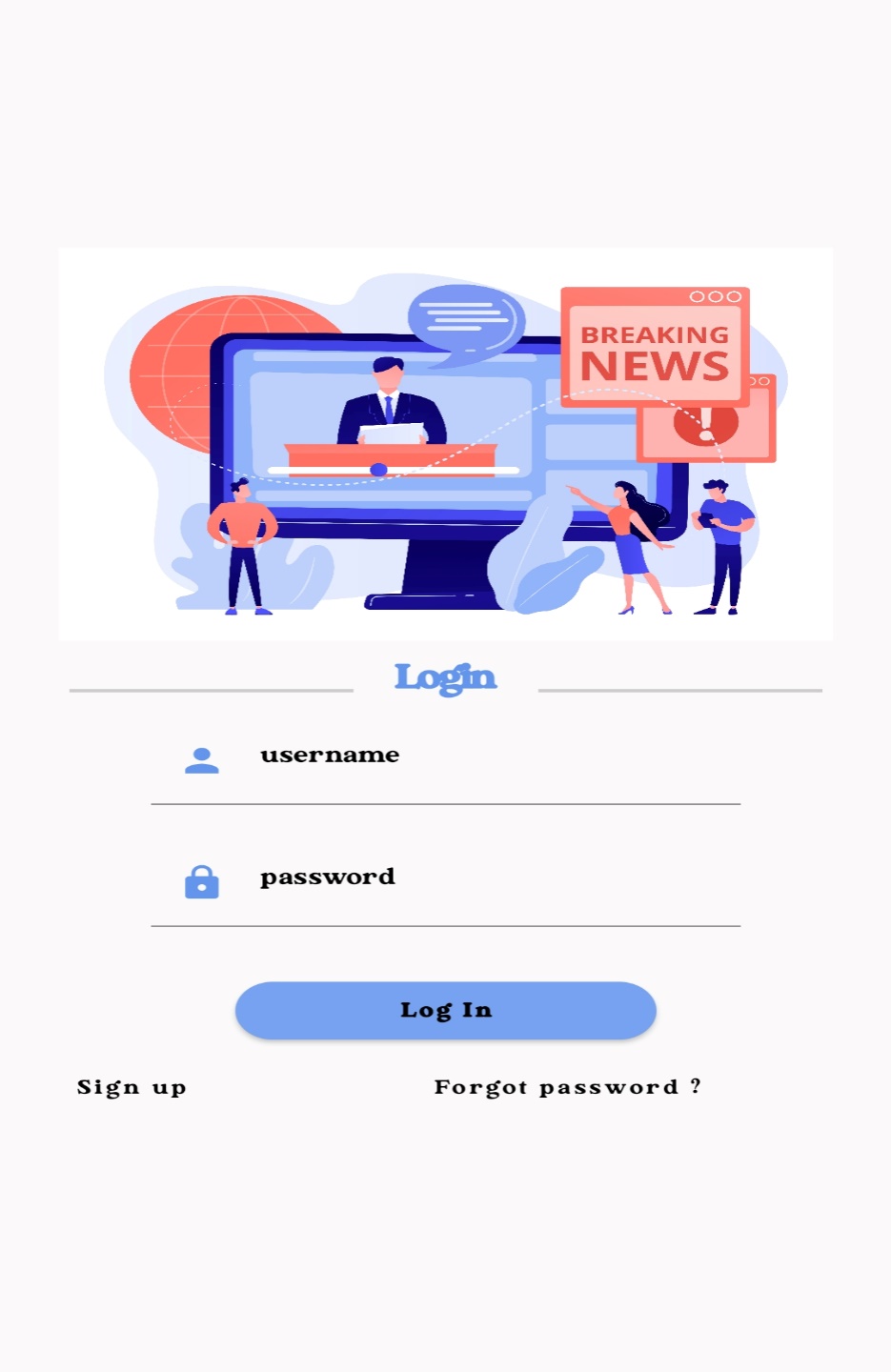
**2.1 Empathy Map:**

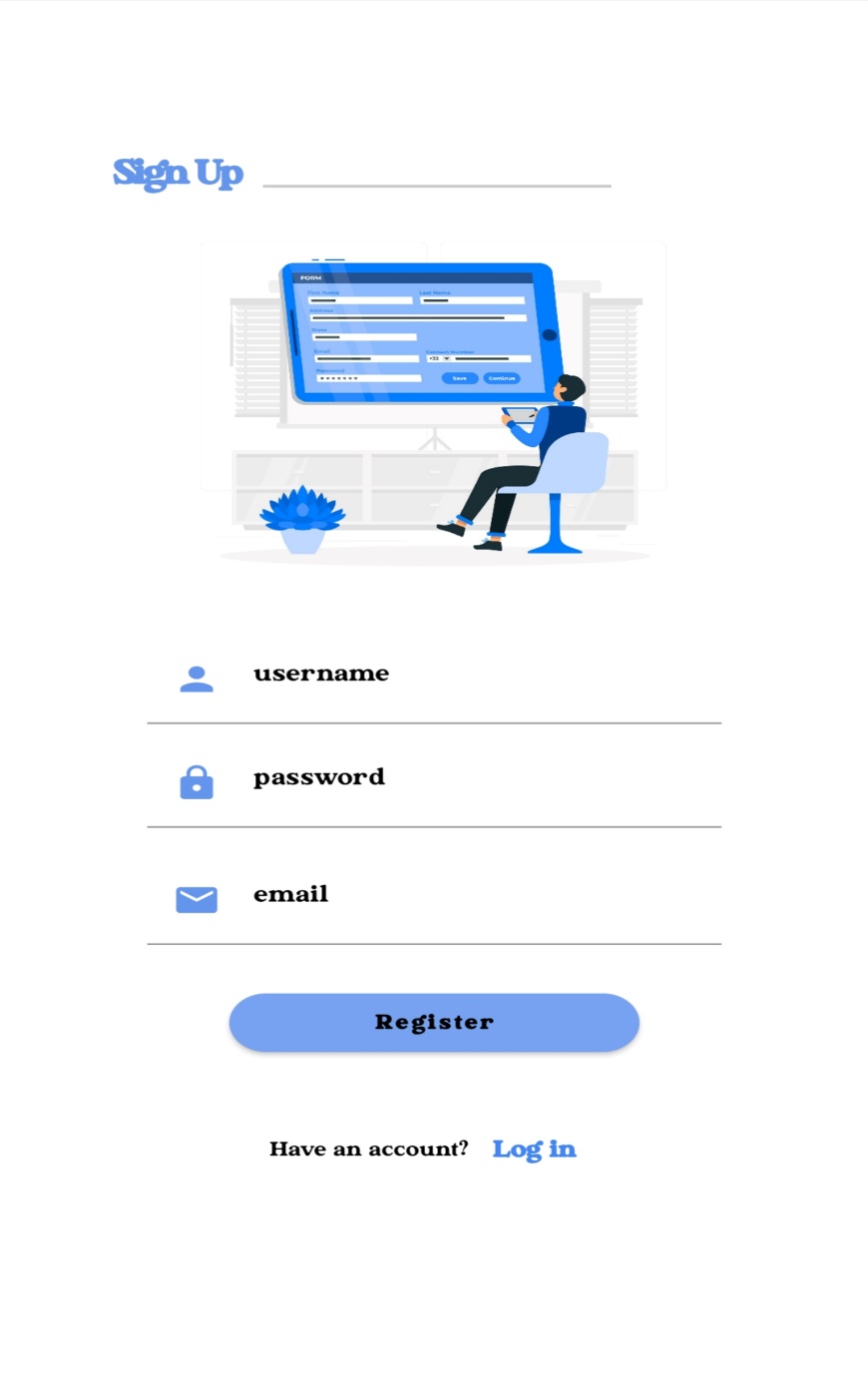
**2.2 Ideation & Brainstorming Map:**



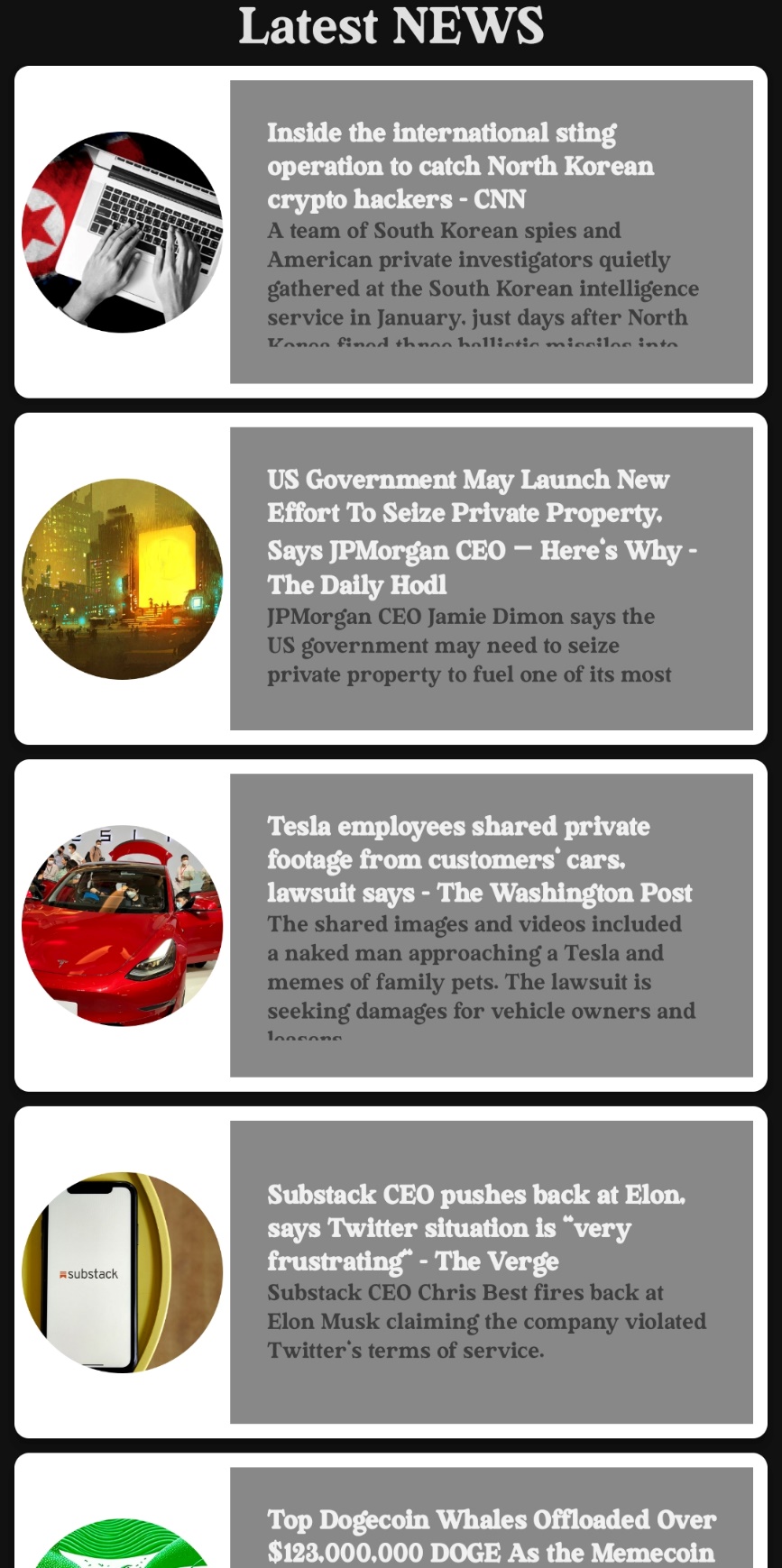
**RESULT**

**Sign in Page:**

****

**Sign up page:**

**Home Page:**

****

**News Page:**

****

**ADVANTAGES & DISADVANTAGES**

* 1. **Advantages:**
* Convenience: A news app provides users with a quick and convenient way to access news articles from multiple sources in one place, without having to navigate through various websites or apps.
* Personalization: Users can customize their news feed by selecting topics or categories of interest, ensuring that they only receive news that is relevant to them.
* Real-time updates: A news app can deliver news stories in real-time, ensuring that users stay up-to-date on current events as they unfold.
* Privacy: Unauthorized person cannot access your app without enter correct username and password.
  1. **Disadvantages:**
* User engagement: A news app can provide users with features such as commenting, sharing, and saving articles, which can encourage engagement and interaction with the app.
* Technical issues: As with any app, technical issues such as bugs, crashes, or slow loading times can frustrate users and damage the app's reputation.
* Limited news sources: If the app only pulls news from a select few sources, users may not receive a diverse range of perspectives on news events. This could lead to a perception of bias or limited coverage.
* Limited functionality: If the app only provides basic features such as signing up and listing news articles, it may not be competitive with other news apps that offer more advanced features such as multimedia content, social sharing, or personalization.

**APPLICATIONS**

* General news: A news app can be used to deliver general news articles from a variety of sources, covering topics such as politics, world news, sports, and entertainment.
* Social news: A news app can incorporate social media features to allow users to share news articles with their social network or engage in discussions around specific news events.
* Education: A news app can be used in an educational setting to provide students with access to current events and news articles relevant to their coursework or research.
* Business: A news app can be used in a business setting to provide employees with access to news articles and updates related to their industry or company.
* Travel: A news app can provide users with access to news articles and updates related to their travel destination, including local news, events, and travel tips.
* Politics: A news app can be used in a political setting to provide users with access to news articles and updates related to politics, elections, and government policies.

**Conclusion**

We conclude that the News app would feature a sign-up and sign-in page, as well as a news page where articles are listed with current updates. Additionally, the app would allow users to view short notes on news articles when they select a particular news item. However, there are some potential limitations to this app, such as limited news sources, basic functionality, and lack of offline access. To improve the app, you plan to add the ability to save news articles, show brief news articles when a user selects a particular news item, improve the user interface, and show news articles based on categories. It can be applied in various areas where there is a need for access to up-to-date news and information, including general news, local news, niche news, social news, education, business, travel, and politics. The UI will be improved to make the app more visually appealing and user-friendly. The News app will not have offline access initially, but this will also be added in a future update.

**Future Scopes**

In future adding more sources to provide users with a wider range of news articles to choose from. To make the app more engaging for users and adding personalization features. Adding features that allow users to share news articles with their friends, or engage in discussions around specific news events. Adding audio and video clip features to the app for more user-friendly. To add more local news articles to provide users with up-to-date information on events and news in their local area and allow users to access news articles even when they are not connected to the internet.

**APPENDIX**

**Source Code:**

<https://github.com/VimalEswar/NewsHeadline>

**Code:**

1. 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)  
}

1. 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)  
}

1. 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*) **}** )  
 }  
}

1. 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*()) *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*) **}** )  
}