RROJECT

REPORT TEMPLATE

Project Name:

An Android Application For Keeping Up With The Latest Headlines

Submitted By ,

RameshKumar   -   TEAM LEADER

Sabeshen   -   TEAM MEMBER

Sahadevan   -   TEAM MEMBER

Philo Jebas Abishek   -   TEAM MEMBER

1. <u>INTRODUCTION :</u>

Overview:

This News App Is A Personalized That Lets You Follow Trending Topics, Entertaining Contents , Knowledge gaining News and Live Cricket Updates.

Top Stories brings you the latest, breaking news from our trusted global network of journalists. My News is a feature allowing you to personalize your news offering. Never Miss A Breaking News Story Or Important Update Again. Personalize Your Experience And Choose The News You Want By Category: Politics, Business, Health, Entertainment And More. Get The News You Want, The Way You Want.

1.2 <u>Purpose</u>

This App provide The Regular updates And Info About The Trending Topics Or Events In The User's Device. There Are Many Sites That Provide Free App For iPhone, Android, And Other Platforms. The User Gets Free App Downloads From The Websites. There Are Many Features In The Applications That Give Live Details To The User. When We talk About journalism and Its Functions, The Few Things That Pop Into Our Heads Are That It Helps In Educating People, Informing Them, And Guiding Them In Forming Their Opinions. In A Nutshell, Its Primary Function Is To Disseminate Information On A Larger Scale And To Reach People All Over The World. The News Is Most Relied On As People Believe It Is The Most Accurate And Objective Item To Trust. News In Print, Audio, Or Audio-visual Form Must Be Objective.

The user can set the home screen widgets of these apps that help in quick access of the application and stories. From there the user can easily read the news in a single click. This applications also allows to post comments to the users on the stories. This applications provide the latest information about the world. The user doesn't need to refresh the app window, they get automatically updated with the time. The user can get regular updates of sports, lifestyle, business, and other things on their device with easy navigation.
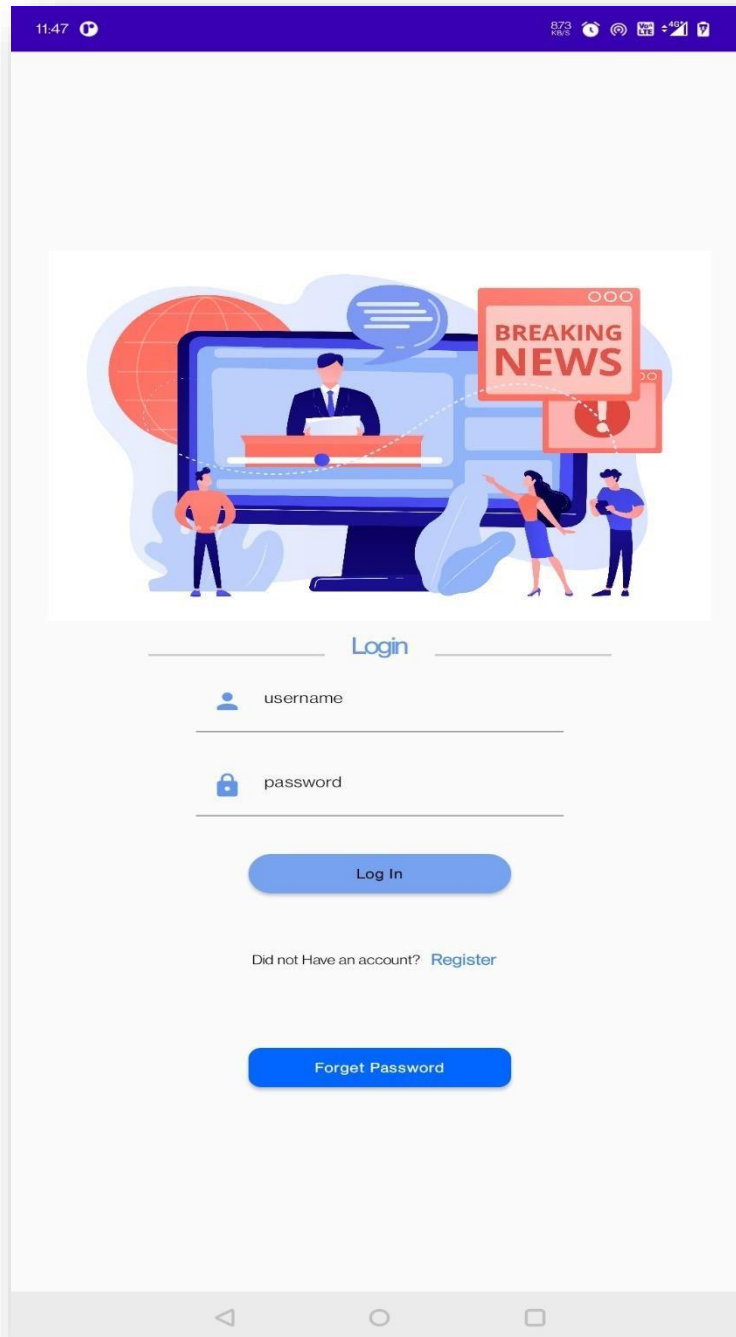
PROBLEM DEFINITION AND DESIGN THINKING

EmpathyMap :
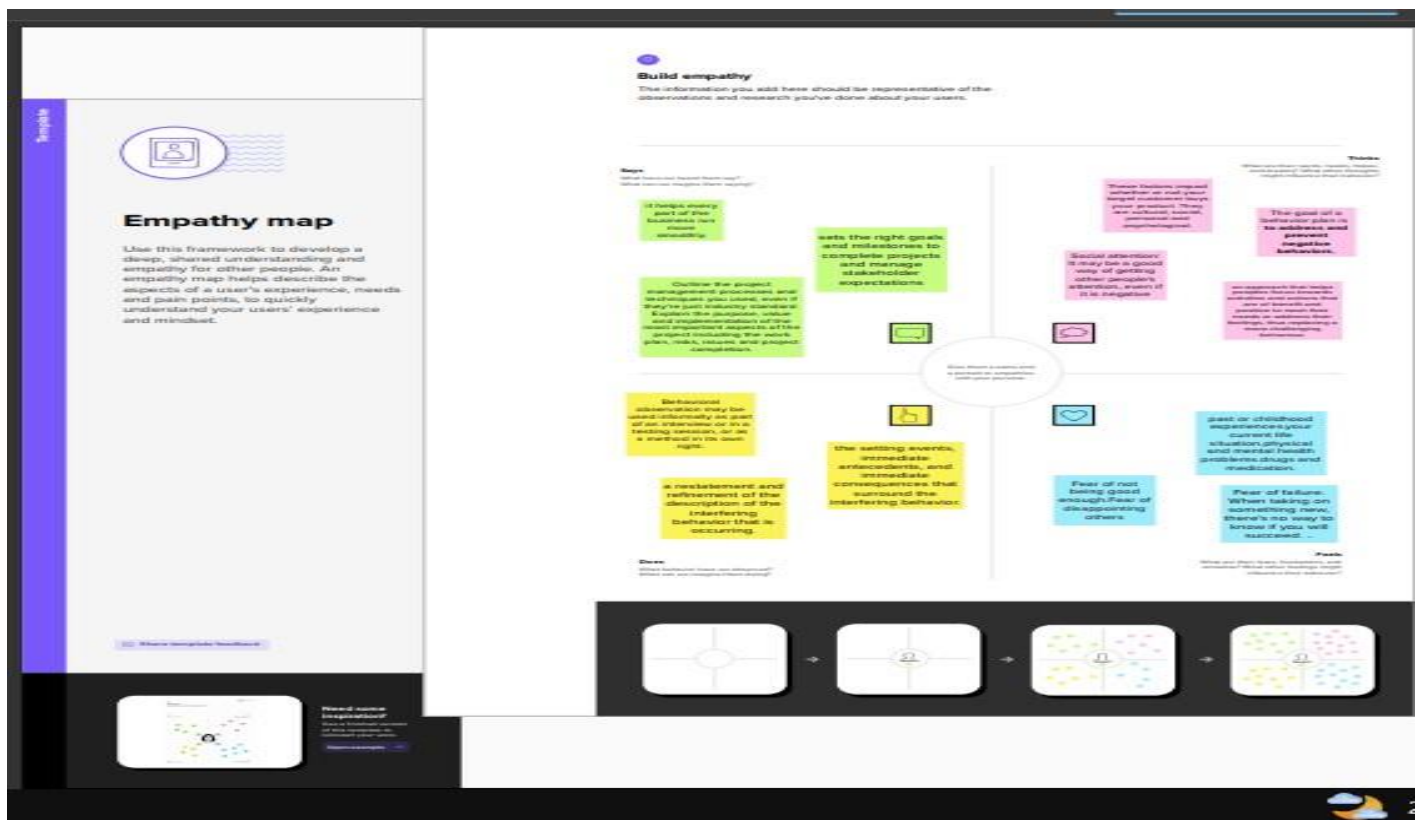
BrainstormingMap
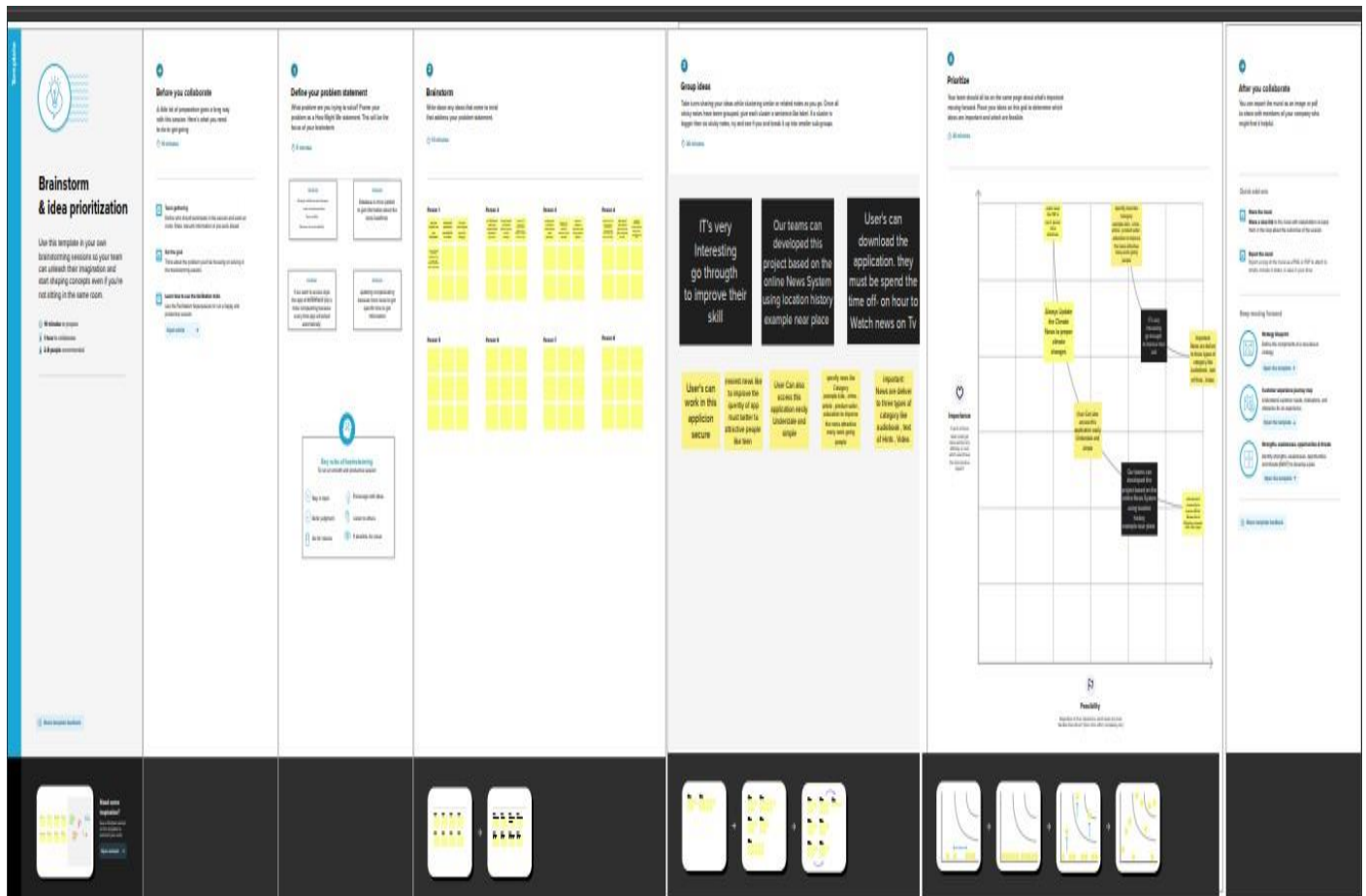
Result:

Sign Up _____

👤 username

🔒 password

✉ email

Register

Have an account? **Log in**

Latest NEWS

Warren Buffett-backed BYD announces new shock absorption tech for premium EVs - CNBC
Electric vehicle giant BYD is banking on new driver-assist technology to smooth out car rides and get the edge on its EV competition.

JetBlue to restore Worcester-Orlando route, says Massport - Worcester Telegram
Flights ended at start of pandemic.

US inflation expectations jump for first time in months, NY Fed survey shows - Fox Business
A New York Federal Reserve survey published Monday shows that consumers are worried about a potential credit crunch and higher inflation in coming years.

Fed rate rise fears weigh on US stocks - Financial Times
Equity traders return from Easter break to respond to robust jobs data

SpaceX eyeing 3rd week of April for Starship orbital launch, Elon Musk says - Space.com
An April 20 liftoff could really happen.

Warren, AOC press top Silicon Valley Bank depositors about any cushy treatment they received - CNBC
Progressive Democrats Sen. Elizabeth Warren and Rep. Alexandria Ocasio-Cortez demanded answers from SVB's 14 largest depositors.

JetBlue to restore Worcester-Orlando route, says Massport - Worcester Telegram

Flights ended at start of pandemic.

Stocks making the biggest moves midday: Micron, Pioneer Natural Resources, Block, AMC and more - CNBC

Micron's shares popped after its rival announced plans to cut memory chip production near term—a move analysts said could lead to a chipmaking sector rebound.

Advantage

Feature:

• **Personalized news:**

                Powered by the most powerful AI news engine from News Api this news app will deliver real-time, AI-curated news that interests you the most. The more you use Google to auto generated News, the better will know you the best while keeping your privacy secure.

• **News push notification:**

         Get critical, breaking news alerts pushed directly to your home screen.

Special :

The user doesn't need to refresh the app window, they get automatically updated with the time. The user can get regular updates of sports, lifestyle, business, and other things on their device with easy navigation.

Disdvantage :

Lack Of Focus :

Simply 'having an app' for the sake of it could prove counter-productive and a waste of financial resources: it's vital to know exactly what you want your app to achieve and have a business case to support it.

Cost :

Naturally developing an app carries a price tag, but your app could boost your bottom line by improving productivity (if a business app) or (if a customer based app) enhancing customer engagement and increasing sales. Also, working with an experienced mobile app development company means they can appraise you of what's possible within your budget and develop your app accordingly.

Complicated :

Developing an app isn't as easy as it may appear, but trusting in an experienced mobile apps for business expert will ensure much of the heavy lifting is done for you.

Updating :

Your app will require regular updates in line with revisions from platform providers such as Apple (iOS) and Google (Android). Again, companies providing mobile app development services know this and would put a package together incorporating ongoing updates.

Application :

The materials and methods section should include a clear and brief description of your research procedures. One important purpose of this section is to convince the readers that your work is valid. Another purpose is for researchers to use your methodology to guide his or her own experiments.

Conclusion :

If an app is being built primarily for the Android platform, then using News Healines Latest Updatation is an easy choice.

Because of Google's widespread adoption, any app based on News App Design principles is going to feel like a native app.

FutureScope

A mobile news app has simple goals: Capture and retain reader attention, and repeat the process, several times a day. Pretty straightforward. But not that simple in the real world. For a news provider, the smartphone screen is the the most challenging environment ever seen. There, chances are that a legacy media or a pure-player will find itself in direct competition, not only with the usual players in its field, but also with Facebook, Snapchat, Instagram and scores of gaming applications. Distraction is just one icon away; any weakness in functional or graphic design can be lethal.

They can measure how long the app is used, when, for how long, why and where people tend to drop it, what kind of news they like, if they hit a paywall and give up, and why they do so, etc. Similarly, when an app remains unopened for too long, smart tools can pinpoint the user and remind her of the product's benefits. These tools are as good as the people who (a) set the parameters, (b) monitor them on a daily basis, and (c) take appropriate action such as launching a broadside of supertargeted emails. But these are incremental measures, they don't breed exponential growth in viewership (and revenue).

8. Appendix

Sourcecode :

https://github.com/RameshSam/NaanMudhalvan_Project
code :-

Loginactivity.kt:

```
package com.example.newsapplication


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
```

```kotlin
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.newsapplication.ui.theme.NewsApplicationTheme


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),
```

```kotlin
    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 = {
```

```kotlin
        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)
    )
```

```kotlin
        }

        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.padding(30.dp),
verticalAlignment = Alignment.CenterVertically,
horizontalArrangement = Arrangement.Center

    )
```

```kotlin
{
        Text(text = "Did not Have an account?")
TextButton(onClick = {
context.startActivity(                Intent(
context,

                RegistrationActivity::class.java
            )
          )
        }) {
      Text(text = "Register",                fontWeight =
FontWeight.Bold,              style =
MaterialTheme.typography.subtitle1,              color =
Color(0xFF4285F4)

          )}


    }


    /* Row(modifier = Modifier.fillMaxWidth()) {
TextButton(onClick = {
context.startActivity(                Intent(
context,

                RegistrationActivity::class.java
          )))})
        { Text(text = "Sign up",
color = Color.Black

        )} */
    Spacer(modifier = Modifier.width(50.dp))
Button(

      onClick = { /* Do something! */ },
shape = RoundedCornerShape(10.dp),
```

```kotlin
        colors = ButtonDefaults.buttonColors(backgroundColor = Color(0xFF0066ff)),        modifier =
Modifier.width(200.dp)

            .padding(top = 20.dp)

    ) {

        Text(text = "Forget Password", fontWeight = FontWeight.Bold , color = Color.White)

    }

  }

}

private fun startMainPage(context: Context) {    val

intent = Intent(context, MainPage::class.java)

   ContextCompat.startActivity(context, intent, null)

}
```

Mainactivity.kt :

```kotlin
package com.example.newsapplication


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
```

```kotlin
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.newsapplication.Articles

import com.example.newsapplication.ui.theme.NewsApplicationTheme


class MainPage : ComponentActivity() {
    val mainViewModel by viewModels<MainViewModel>()
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)        setContent {

        NewsApplicationTheme {
            // A surface container using the 'background' color from the theme
            Surface(color = MaterialTheme.colors.background) {
                Column() {
```

```kotlin
                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"
   )
```

```kotlin
    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)
```

```kotlin
                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
```

```kotlin
            )
            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) }
    )
  }
}
```

DisplayNews.kt

```kotlin
package com.example.newsapplication


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
```

```kotlin
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.newsapplication.ui.theme.NewsApplicationTheme class

DisplayNews : ComponentActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)        setContent {

        NewsApplicationTheme {

            // 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) {
```

```kotlin
                    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() {

  NewsApplicationTheme {

    //   Greeting("Android")

  }

}

@Composable

fun HtmlText(html: String, modifier: Modifier = Modifier) {

AndroidView(        modifier = modifier,
```

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

Registeractivity.kt

```kotlin
package com.example.newsapplication


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
```

```kotlin
import androidx.compose.ui.unit.sp import
androidx.core.content.ContextCompat

import com.example.newsapplication.ui.theme.NewsApplicationTheme


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",
```

```kotlin
        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

        )
```

```kotlin
            },
            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 = {
```

```kotlin
        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
```

```kotlin
                        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)
}
```

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_app_icon"
android:label="@string/app_name"        android:supportsRtl="true"
```

```xml
        android:theme="@style/Theme.NewsApplication"
        tools:targetApi="31">
<activity
        android:name=".DisplayNews"
android:exported="false"

        android:label="@string/title_activity_display_news"
android:theme="@style/Theme.NewsApplication" />

<activity
        android:name=".RegistrationActivity"
android:exported="false"

        android:label="@string/title_activity_registration"
android:theme="@style/Theme.NewsApplication" />

<activity
        android:name=".MainPage"
android:exported="false"

        android:label="@string/title_activity_main_page"
android:theme="@style/Theme.NewsApplication" />

<activity
        android:name=".LoginActivity"
android:exported="true"
android:label="@string/app_name"
android:theme="@style/Theme.NewsApplication">

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


</manifest>
```