

# Daily News APP

Aswath A  
Barath Giri C S  
Shree Pranav S  
Namrutha K  
Ashvithaa V P

# DESCRIPTION

**News Aggregation:** A comprehensive news platform that gathers updates from a variety of sources, offering users convenient access to the latest stories across diverse categories such as politics, sports, technology, entertainment, business, and more. This ensures users stay informed with real-time updates.

**Personalized Feed:** Users can customize their news experience by selecting topics of interest, creating a tailored feed that prioritizes relevant and engaging content.

**Real-Time Notifications:** Stay ahead of the curve with instant alerts for breaking news, ensuring users are always aware of major events and developments as they unfold.

**Offline Mode:** Save articles for offline reading, enabling users to stay informed even without an active internet connection, providing flexibility and convenience.

**Multimedia Content:** Enrich your news experience with a variety of media formats, including videos, images, and infographics, complementing traditional text-based articles.

**Trusted Sources:** All content is curated from credible and verified outlets, ensuring the information delivered is reliable.

## Login Activity.java:

```
@file:Suppress("DEPRECATION")package com.example.newsheadlinesimport android.content.Contextimport android.content.Intentimport android.os.Bundleimport androidx.activity.ComponentActivityimport androidx.activity.compose.setContentimport androidx.compose.foundation.Imageimport androidx.compose.foundation.backgroundimport androidx.compose.foundation.layout.*import androidx.compose.foundation.shape.RoundedCornerShapeimport androidx.compose.material.*import androidx.compose.material.icons.Iconsimport androidx.compose.material.icons.filled.Lockimport androidx.compose.material.icons.filled.Personimport androidx.compose.material3.Buttonimport androidx.compose.material3.ButtonDefaultsimport androidx.compose.material3.Dividerimport androidx.compose.material3.ExperimentalMaterial3Apiimport androidx.compose.material3.Iconimport androidx.compose.material3.MaterialThemeimport androidx.compose.material3.Textimport androidx.compose.material3.TextFieldimport androidx.compose.material3.TextFieldDefaultsimport androidx.compose.runtime.*import androidx.compose.ui.Alignmentimport androidx.compose.ui.Modifierimport androidx.compose.ui.graphics.Colorimport
```

```

androidx.compose.ui.res.painterResourceimport
androidx.compose.ui.text.font.FontWeightimport
androidx.compose.ui.text.input.PasswordVisualTransformationimport
androidx.compose.ui.tooling.preview.Previewimport
androidx.compose.ui.unit.dpimport androidx.compose.ui.unit.spimport
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)
            }
        }
    }
    @OptIn(ExperimentalMaterial3Api::class)
    @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.headlineLarge
                )
            }
            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(containerColor = 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(containerColor = Color.Transparent)          )
Spacer(modifier = Modifier.height(12.dp))          if (error.isNotEmpty())
{          Text(text = error, color = MaterialTheme.colorScheme.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 logged in"
context.startActivity(Intent(context, MainActivity::class.java))          //
Finish the LoginActivity if needed          // (context as Activity).finish()
} else {          error = "Invalid username or password"          }
} else {          error = "Please fill all fields"          }          },          shape =
RoundedCornerShape(20.dp),          colors =
ButtonDefaults.buttonColors(containerColor = 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 = { /* Handle forgot password */ }) {                Text(text = "Forgot
password?", color = Color.Black)      }      }  }}

```

registration activity:

```

package com.example.newsheadlinesimport android.content.Contextimport
android.content.Intentimport android.os.Bundleimport
androidx.activity.ComponentActivityimport
androidx.activity.compose.setContentimport
androidx.compose.foundation.Imageimport
androidx.compose.foundation.backgroundimport
androidx.compose.foundation.layout.*import
androidx.compose.foundation.shape.RoundedCornerShapeimport
androidx.compose.material.*import androidx.compose.material.icons.Iconsimport
androidx.compose.material.icons.filled.Emailimport
androidx.compose.material.icons.filled.Lockimport
androidx.compose.material.icons.filled.Personimport
androidx.compose.runtime.*import androidx.compose.ui.Alignmentimport
androidx.compose.ui.Modifierimport androidx.compose.ui.graphics.Colorimport
androidx.compose.ui.res.painterResourceimport
androidx.compose.ui.text.font.FontWeightimport
androidx.compose.ui.text.input.PasswordVisualTransformationimport
androidx.compose.ui.tooling.preview.Previewimport

```

```

androidx.compose.ui.unit.dpimport androidx.compose.ui.unit.spimport
androidx.core.content.ContextCompatimport
com.example.newsheadlines.ui.theme.NewsHeadlinesTheme
class
RegistrationActivity : ComponentActivity() { private lateinit var dbHelper:
UserDatabaseHelper override fun onCreate(savedInstanceState: Bundle?)
{ super.onCreate(savedInstanceState) dbHelper =
UserDatabaseHelper(this) setContent
{ RegistrationScreen(this, dbHelper) } }@Composablefun
RegistrationScreen(context: Context, dbHelper: 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"

```



```

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)}

```

main activity:

```

package com.example.newsheadlinesimport android.content.Contextimport android.content.Intentimport
android.content.Intent.FLAG_ACTIVITY_NEW_TASKimport android.os.Bundleimport android.util.Logimport
android.widget.TextViewimport androidx.activity.ComponentActivityimport
androidx.activity.compose.setContentimport androidx.activity.viewModelsimport
androidx.compose.foundation.Imageimport androidx.compose.foundation.backgroundimport
androidx.compose.foundation.clickableimport androidx.compose.foundation.layout.*import
androidx.compose.foundation.lazy.LazyColumnimport androidx.compose.foundation.lazy.itemsIndexedimport
androidx.compose.foundation.selection.selectableimport
androidx.compose.foundation.shape.RoundedCornerShapeimport androidx.compose.material.Cardimport

```

```

androidx.compose.material.MaterialThemeimport
androidx.compose.material.Surfaceimport androidx.compose.material.Textimport
androidx.compose.runtime.*import androidx.compose.ui.Modifierimport
androidx.compose.ui.graphics.Colorimport
androidx.compose.ui.text.font.FontWeightimport
androidx.compose.ui.text.style.TextAlignimport androidx.compose.ui.unit.dpimport
androidx.compose.ui.unit.spimport
androidx.compose.ui.viewinterop.AndroidViewimport
androidx.core.text.HtmlCompatimport coil.compose.rememberImagePainterimport
coil.size.Scaleimport coil.transform.CircleCropTransformationimport
com.example.example.Articlesimport
com.example.newsheadlines.ui.theme.NewsHeadlinesTheme
class MainActivity :
    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()
                        }
                    }
                }
            }
        }
    }
}

@Composablefun 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
            }
        }
    }
}

@Composablefun MovieItem(context: Context) {
    val movie =

```

```

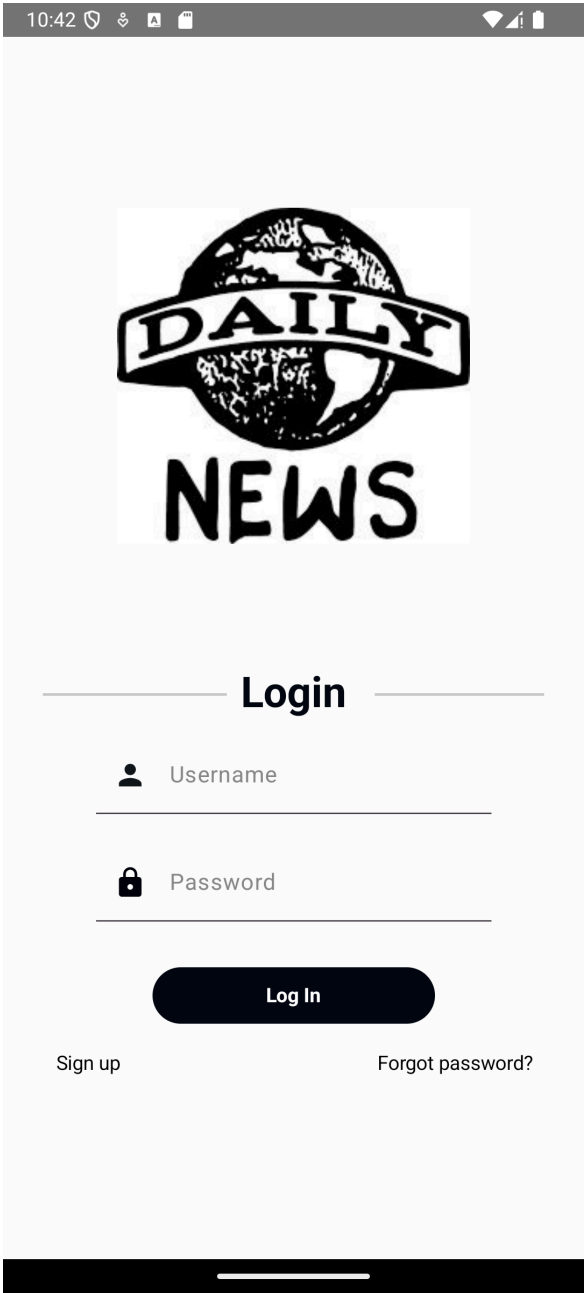
Articles("Coco", "", "articl" ) MovieItem(context, movie = movie, 0,
0) { i -> Log.i("wertyttest123abc", "MovieItem: " + i) } } @Composable fun
MovieItem(context: Context, movie: Articles, index: Int, selectedIndex: Int,
onClick: (Int) -> Unit) { val backgroundColor = if (index == selectedIndex)
MaterialTheme.colors.primary 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 = 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(tr
ue, true, null, onClick = { Log.i("test123abc",
"MovieItem: $index/n${movie.description}") context.startActivity(
Intent(context,
DisplayNews::class.java) .setFlags(Intent.FLAG_ACTIVITY_NE

```



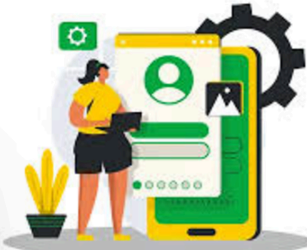
```
// Now we pass the API key correctly      val movieList =  
apiService.getMovies(apiKey = apiKey) // Pass the API key here  
movieListResponse = movieList.articles  } catch (e: Exception)  
{      errorMessage = e.message.toString()      Log.e("MainViewModel",  
"Error fetching movies: $errorMessage")  }  }  }}
```

OUTPUT:



10:44

## Sign Up



978597

Username

password

email

Register

Have an account? [Log in](#)



10:43



## Latest NEWS



**Russia's Gazprom stops the flow of natural gas to Austria, OMV utility says - The Associated Press**

Austrian utility OMV says supplies from Russia's state-owned natural



**Souring S&P 500 Profit Outlook a Bad Sign for Stock Market Rally - Bloomberg**

Wall Street analysts are quickly scaling back their forecasts for Corporate America's earnings growth



**Gold suffers worst week in 3 years as investors weigh Trump victory - Financial Times**

Dollar surge and Trump win brings metal's rally to a halt



**The Fed interrupted the stock market's Trump rally. What comes next. - MarketWatch**

Feedback loop between higher Treasury yields and wobbles in the equity market to remain center



**7 Social Security Changes Retirees Need to Know About**