AN ANDROID APPLICATION FOR KEEPING UP WITH THE LATEST HEADLINES

DESCRIPTION :

Stay informed and connected with the latest news across the globe using Headlines Hub, the ultimate Android application designed to bring you real-time updates and trending stories right to your fingertips. Whether you are a news junkie or an occasional reader, Headlines Hub offers a user-friendly experience tailored to meet your needs.

Features:

1. Personalized News Feed: Customize your feed with topics and sources you care about.
2. Real-Time Updates: Get instant notifications for breaking news.
3. Multiple Sources: Access news from leading publications and websites.
4. Offline Reading: Save articles for offline reading.
5. Search Function: Find specific news stories and topics.
6. Daily Briefings: Receive summarized news updates.

Creating the database classes:

***create user data class**

```
package com.example.newsheadlines
import androidx.room.ColumnInfo
import androidx.room.Entity
import androidx.room.PrimaryKey
@Entity(tableName = "user_table")
data class User(
@PrimaryKey(autoGenerate = true)
val id: Int?,
@ColumnInfo(name = "first_name")
val firstName: String?,
@ColumnInfo(name = "last_name")
val lastName: String?,
@ColumnInfo(name = "email") val
email: String?,
@ColumnInfo(name = "password")
val password: String?,
)
```

***Create an userDao interface**

```
package com.example.newsheadlines
import androidx.room.*
@Dao
interface UserDao {
@Query("SELECT * FROM user_table WHERE
email = :email")
suspend fun getUserByEmail(email: String):
User?
@Insert(onConflict =
OnConflictStrategy.REPLACE)
suspend fun insertUser(user: User)
@Update
suspend fun updateUser(user: User)
@Delete
suspend fun deleteUser(user: User)
}
```

***Create an user database class**

```
package com.example.newsheadlines
import android.content.Context
import androidx.room.Database
import androidx.room.Room
import androidx.room.RoomDatabase
@Database(entities = [User::class], version = 1)
abstract class UserDatabase : RoomDatabase() {
    abstract fun userDao(): UserDao
    companion object {
        @Volatile
        private var instance: UserDatabase? = null
        fun getDatabase(context: Context):
                UserDatabase {
            return instance ?: synchronized(this)
```

```
                {
                    val newInstance =
Room.databaseBuilder(
    context.applicationContext,
    UserDatabase::class.java,
    "user_database"
).build()
instance = newInstance
newInstance
                }
            }
        }
```

***Create an user database helper class**

```
package com.example.newsheadlines
import android.annotation.SuppressLint
import android.content.ContentValues
import android.content.Context
import android.database.Cursor
import
android.database.sqlite.SQLiteDatabase
import
android.database.sqlite.SQLiteOpenHelper
class UserDatabaseHelper(context: Context)
:
SQLiteOpenHelper(context,
DATABASE_NAME, null, DATABASE_VERSION)
{
companion object {
private const val DATABASE_VERSION =
1
private const val DATABASE_NAME =
"UserDatabase.db"
```

```
private const val TABLE_NAME = "user_table"
        private const val COLUMN_ID = "id"
    private const val COLUMN_FIRST_NAME =
                "first_name"
    private const val COLUMN_LAST_NAME =
                "last_name"
    private const val COLUMN_EMAIL = "email"
        private const val COLUMN_PASSWORD =
                "password"
                }
override fun onCreate(db: SQLiteDatabase?) {
        val createTable = "CREATE TABLE
                $TABLE_NAME (" +
            "$COLUMN_ID INTEGER PRIMARY KEY
                AUTOINCREMENT, " +
            "$COLUMN_FIRST_NAME TEXT, " +
            "$COLUMN_LAST_NAME TEXT, " +
                "$COLUMN_EMAIL TEXT, " +
            "$COLUMN_PASSWORD TEXT" +
                ")"
        db?.execSQL(createTable)
                }
```

```
override fun onUpgrade(db: SQLiteDatabase?,
oldVersion: Int, newVersion: Int) {
db?.execSQL("DROP TABLE IF EXISTS
$TABLE_NAME")
onCreate(db)
}
fun insertUser(user: User) {
val db = writableDatabase
val values = ContentValues()
values.put(COLUMN_FIRST_NAME,
user.firstName)
values.put(COLUMN_LAST_NAME,
user.lastName)
values.put(COLUMN_EMAIL, user.email)
values.put(COLUMN_PASSWORD,
user.password)
db.insert(TABLE_NAME, null, values)
db.close()
}
@SuppressLint("Range")
fun getUserByUsername(username: String):
User? {
```

```
val db = readableDatabase
val cursor: Cursor = db.rawQuery("SELECT * FROM $TABLE_NAME WHERE $COLUMN_FIRST_NAME = ?", arrayOf(username))
var user: User? = null
if (cursor.moveToFirst()) {
user = User(
id = cursor.getInt(cursor.getColumnIndex(COLUMN_ID)),
firstName = cursor.getString(cursor.getColumnIndex(COLUMN_FIRST_NAME)),
lastName = cursor.getString(cursor.getColumnIndex(COLUMN_LAST_NAME)),
email = cursor.getString(cursor.getColumnIndex(COLUMN_EMAIL)),
password = cursor.getString(cursor.getColumnIndex(COLUMN_PASSWORD)),
)
}
cursor.close()
```

```
        db.close()
        return user
    }
    @SuppressLint("Range")
    fun getUserById(id: Int): User? {
        val db = readableDatabase
        val cursor: Cursor = db.rawQuery("SELECT * FROM
        $TABLE_NAME WHERE $COLUMN_ID = ?",
            arrayOf(id.toString()))
        var user: User? = null
        if (cursor.moveToFirst()) {
            user = User(
                id =
        cursor.getInt(cursor.getColumnIndex(COLUMN_ID)),
                firstName =
        cursor.getString(cursor.getColumnIndex(COLUMN_FIRST_
        NAME)),
                lastName =
        cursor.getString(cursor.getColumnIndex(COLUMN_LAST_
        NAME)),
                email =
        cursor.getString(cursor.getColumnIndex(COLUMN_EMAIL
        )),
                password =
        cursor.getString(cursor.getColumnIndex(COLUMN_PASS
        WORD)),
            )
        }
        cursor.close()
```

```
        db.close()
        return user
    }
    @SuppressLint("Range")
    fun getAllUsers(): List<User> {
        val users = mutableListOf<User>()
        val db = readableDatabase
        val cursor: Cursor =
            db.rawQuery("SELECT * FROM
                $TABLE_NAME", null)
            if (cursor.moveToFirst()) {
                do {
                    val user = User(
                        id =
cursor.getInt(cursor.getColumnIndex(COLU
                        MN_ID)),
                        firstName =
cursor.getString(cursor.getColumnIndex(CO
                        LUMN_FIRST_NAME)),
                        lastName =
cursor.getString(cursor.getColumnIndex(CO
                        LUMN_LAST_NAME)),
                        email =
cursor.getString(cursor.getColumnIndex(CO
                        LUMN_EMAIL)),
```

```
                password =
cursor.getString(cursor.getColumnIndex(COLUM
                N_PASSWORD)),
                )
                users.add(user)
            } while (cursor.moveToNext())
        }
        cursor.close()
        db.close()
        return users
    }
}
```

Creating API service and Required classes for integrating API

***Database fir news Integration into project:**

```
package com.example.newsheadlines
import retrofit2.Retrofit
import
retrofit2.converter.gson.GsonConverterFactory
import retrofit2.http.GET
interface ApiService {
 //@GET("movielist.json")
 @GET("top-headlines?
country=us&category=business&apiKey=684cb8
93caf7425abeffad82ac1d0f4e")
 ///@GET("search?q=chatgpt")
 suspend fun getMovies() :News
 companion object {
```

```
var apiService: ApiService? = null
    fun getInstance() : ApiService {
        if (apiService == null) {
            apiService = Retrofit.Builder()
                //
.baseUrl("https://howtodoandroid.com/apis/")
                .baseUrl("https://newsapi.org/v2/")
                //.baseUrl("https://podcast-
episodes.p.rapidapi.com/")

.addConverterFactory(GsonConverterFactory.cr
eate())
                .build().create(ApiService::class.java)
            }
        return apiService!!
        }
    }
}
```

***Create model data class:**

```
package com.example.newsheadlines
    data class Movie(val name: String,
                     val imageUrl: String,
                     val desc: String,
                     val category: String)
```

***Create News data class:**

```
package com.example.newsheadlines
import com.example.example.Articles
import
com.google.gson.annotations.SerializedName
data class News (
@SerializedName("status") var status:String?=
null,
@SerializedName("totalResults") var
totalResults : Int?              = null,
@SerializedName("articles") var articles    :
ArrayList<Articles> = arrayListOf()
)
```

*** Create Source data class:**

```
package com.example.example
import
com.google.gson.annotations.SerializedName
data class Source (
@SerializedName("id"   ) var id   : String? = null,
@SerializedName("name" ) var name : String? =
null
)
```

***Create Artical data class:**

```
package com.example.example

import com.google.gson.annotations.SerializedName

data class Articles (
@SerializedName("title"       ) var title : String? = null,
@SerializedName("description" ) var description : String? = null,
@SerializedName("urlToImage"  ) var urlToImage  : String? = null,
)
```

Create main view model class

```
package com.example.newsheadlines
import android.util.Log
import androidx.compose.runtime.getValue
import
androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.setValue
import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import com.example.example.Articles
import kotlinx.coroutines.launch
class MainViewModel : ViewModel() {
var movieListResponse:List<Articles> by
mutableStateOf(listOf())
var errorMessage: String by
mutableStateOf("")
fun getMovieList() {
viewModelScope.launch {
```

```
val apiService = ApiService.getInstance()
                try {
            val movieList = apiService.getMovies()
          movieListResponse = movieList.articles
                }
                    catch (e: Exception) {
                errorMessage = e.message.toString()
                    }
                }
            }
        }
```

Building Application UI and connecting to database

* creating login activity. Kt with database

```
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.PasswordVisualT
ransformation
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.startActi
vity
```

```
import
com.example.newsheadlines.ui.theme.NewsHea
dlinesTheme

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(backgroundCo
lor = 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 username4 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)
}
```

***Creating Registration Activity.kt with database:**

```
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.RoundedC
ornerShape
import androidx.compose.material.*
import
androidx.compose.material.icons.Iconsimport
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(backgroundCo
lor = 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(backgroundCo
lor = 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)
}
```

***Creating MainPage. Kt file:**

```
package com.example.newsheadlines

import android.content.Context
import android.content.Intent
import
android.content.Intent.FLAG_ACTIVITY_NEW_TA
SK
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.selectab
le
import
androidx.compose.foundation.shape.RoundedC
ornerShape
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) }
    )
}
}
```

***Creating Display News .kt file:**

```
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.Arrangeme
nt
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.NewsHea
dlinesTheme
```

```
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 urlImage =
getIntent().getStringExtra("urlToImage")
                        Log.i("test123abc", "MovieItem:
                $desk")
```

```
Column(Modifier.background(Color.Gray).paddin
g(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) }
    )
}
```

Modifying Android 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:supportsRtl="true"

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

android:label="@string/title_activity_display_ne
ws"

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>
```

Login

username

passwvrd

Lvg In

Sign up

Fvrgvt passwvrd ?

Sign Up

princy

.....

rakkiprincy449@gmail.com

Register

Have an account? Log in

Login

princy

••••••••••

Log In

Sign up

Forgot password ?

Latest NEWS

Southwest plane heading to Indy hit by gunfire at Dallas airport - FOX 59 Indianapolis

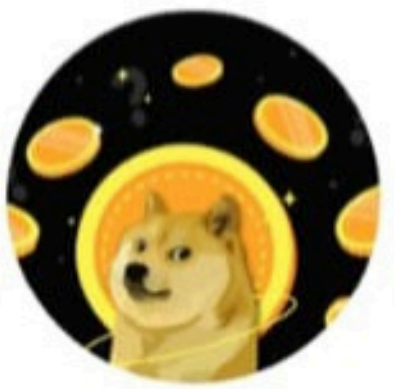
DALLAS, Texas – Police are investigating after a Southwest airplane was reportedly struck by gunfire at a Dallas airport Friday night. Passengers reported at around 9:15 p.m., the plane was preparing to take off from Dallas to Indianapolis when it was by a bu...

Austria says Russia to cut off gas supplies on Saturday - POLITICO Europe

Austrian Chancellor Karl Nehammer says there is a secure supply of alternative fuel and "no one will freeze this winter."

Prediction: Nvidia Stock Is Going to Stall Out on Nov. 20 - Yahoo Finance

The perfect storm may be brewing for Nvidia.



Dogecoin Price Flashes Bull Flag On The Hourly Chart, Can It Rally To $1? | Bitcoinist.com - Bitcoinist

Another DOGE rally may be imminent after the Dogecoin price flashed a bullish pattern on the hourly chart.

Latest NEWS

Just Before Biden's Term Ends, Taiwan Semiconductor Lands $6.6 Billion in US Funding to Build Advanced Chip Plants - Benzinga

The U.S. will provide up to $6.6 billion in funding to Taiwan Semiconductor to build chip facilities, creating jobs and boosting national security.

A Powerful AI Breakthrough Is About to Transform the World - The Wall Street Journal

The technology driving ChatGPT is capable of so much more. What's coming next will make talking bots look like mere distractions.

What is Bluesky, the fast-growing social platform welcoming fleeing X users? - The Associated Press

Disgruntled X users are once again flocking to Bluesky, a newer social media platform that grew out of the former Twitter before billionaire Elon Musk took it over in 2022. While it remains small compared to established online spaces such as X, it has emerged...

Major Trump Media shareholder sells nearly entire stake - Reuters

Trump Media & Technology Group's key shareholder, ARC Global Investments, has unloaded nearly all its stake in the media company, it said in a regulatory filing on Thursday.

Latest NEWS

Supermicro Stock Surges on Reports of Plan To Avoid Delisting - Investopedia

Super Micro Computer shares soared in extended trading Friday following reports the company is expected to file a plan by Monday that could help it avoid delisting.

US Inflation on Strong Trajectory to 2% Goal, Fed's Collins Says - Yahoo Finance

(Bloomberg) -- Federal Reserve Bank of Boston President Susan Collins said she sees inflation heading back to the central bank's 2% goal even though the data...

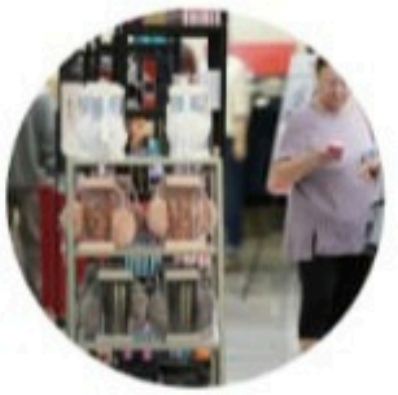
American Airlines flight from Hawaii to California narrowly avoids mountain collision - San Francisco Chronicle

An American Airlines flight from Hawaii to California narrowly avoided a mountain collision after the pilot missed a turn, prompting an FAA investigation.

Lilly Sues U.S. to Change Hospital Drug Discount Payments - WSJ - The Wall Street Journal

null

Latest NEWS



Retail sales up solidly in October as Americans showed continued willingness to spend - The Associated Press

Americans stepped up their spending at retailers last month in the latest sign that healthy consumer spending is driving the economy's steady growth. Retail sales rose 0.4% from September to October, a solid increase though less than the previous month's rebu...

S&P 500 Gains and Losses Today: Palantir Stock Jumps as Listing Heads to Nasdaq - Investopedia

The S&P 500 fell 1.3% on Friday, Nov. 15, after the Fed chair struck a cautious tone about additional rate cuts and a report showed strong retail sales in October.

Elon Musk's companies are soaring in value - Axios

Musk's devoted considerable time lately to his alliance with Trump — in the meantime, his companies are attracting a deluge of capital.

Disney, Comcast, Lionsgate and WBD Ad Spend on Elon Musk's X Falls 98% - TheWrap

The companies collectively spent less than $3.3 million on the social media platform from January to September - a 98% drop from $170 million a year ago

Latest NEWS

Tesla Stock Edges Up. Are Trump EV Tax-Credit Concerns Weighing on the Shares? - Barron's

null

Walmart Just dropped epic deals for Black Friday — I've handpicked 47 sales to shop now - Tom's Guide

Save big on TVs, laptops, appliances and toys

5 Bank Stocks That Could Get a Trump Boost - Barron's

null

Trump's Post-Election Rally Fades, Dollar Surges To Over 1-Year Highs, Powell Puts December Interest Rate Cut Into Question: This Week In The Market - Benzinga

Wall Street faces reality check as post-election rally falters. Republicans' win gives Trump freedom to shape economic policy, but uncertainty remains.

Southwest plane heading to Indy hit by gunfire at Dallas airport - FOX 59 Indianapolis

DALLAS, Texas – Police are investigating after a Southwest airplane was reportedly struck by gunfire at a Dallas airport Friday night. Passengers reported at around 9:15 p.m., the plane was preparing to take off from Dallas to Indianapolis when it was by a bu...