

Project Report Template

1. INTRODUCTION

1.1 Overview

1.2 Purpose

2. PROJECT DEFINITION AND DESIGN THINKING

2.1 Empathy Map

2.2 Ideation & Brainstorming Map

3. RESULT

3.1 Login page of project

3.2 Sign page of project

3.3 Activity intro page

3.4 Brief news page

4. ADVANTAGES & DISADVANTAGES

5. APPLICATIONS

6. CONCLUSION

7. FUTURE SCOPE

8. APPENDIX

1.INTRODUCTION

The world becomes more interconnected, staying up-to-date on the latest news and events has become increasingly important. With so many different sources of news available, it can be overwhelming to try and keep track of everything that's going on. That's where a news headlines application can come in handy. A news headlines application is a tool that aggregates news articles from a variety of sources and presents them to users in an easy-to-digest format. These apps typically allow users to customize their news feed by selecting the topics and sources they're interested in, and they often provide push notifications to alert users of breaking news.

There are many different news headlines applications available on the market today, each with its own set of features and capabilities. Some of the most popular news apps include Apple News, Google News, and Flipboard. In this article, we'll take a closer look at what these apps have to offer and how they can help you stay informed on the latest news and events.

Apple News

Apple News is a news headlines application that's available on iOS devices. It's a free app that aggregates news articles from a variety of sources, including major news outlets like CNN and the New York Times, as well as smaller, niche publications. The app allows users to customize their news feed by selecting topics and sources they're interested in, and it provides a "For You" section that features personalized news recommendations based on the user's reading history. One unique feature of Apple News is its integration with Apple's Siri virtual assistant. Users can ask Siri to read them the latest news headlines, and the app will automatically open and begin playing a news briefing. Apple News also provides push notifications for breaking news stories, and it allows users to save articles to read later.

Google News

Google News is a news headlines application that's available on both iOS and Android devices. Like Apple News, it's a free app that aggregates news articles from a variety of sources. Google News uses artificial intelligence to analyze user behavior and provide personalized news recommendations. It also provides a "For You" section that features news articles based on the user's interests and reading history.

One unique feature of Google News is its "Full Coverage" feature, which provides a comprehensive overview of a news story by pulling together articles from a variety of sources and displaying them in a single feed. Google News also provides push notifications for breaking news stories, and it allows users to save articles to read later.

Flipboard

Flipboard is a news headlines application that's available on both iOS and Android devices. It's a free app that allows users to create their own custom news magazines by selecting topics and sources they're interested in. Flipboard pulls articles from a variety of sources, including major news outlets and smaller, niche publications. One unique feature of Flipboard is its user interface, which presents news articles in a magazine-style format that's easy to read and visually appealing. Flipboard also allows users to save articles to read later, and it provides push notifications for breaking news stories.

In conclusion, a news headlines application can be a useful tool for staying up-to-date on the latest news and events. Apple News, Google News, and Flipboard are just a few examples of the many news apps available on the market today. Whether you're looking for personalized news recommendations or comprehensive

coverage of a breaking news story, there's likely a news headlines application out there that will meet your needs.

1.1 OVERVIEW

An application headlines report is an overview of the latest news and developments in the world of applications. In recent years, the popularity of mobile and web applications has skyrocketed, and the industry has seen a surge in innovation, competition, and investment. This report aims to provide an overview of the most significant headlines in the application industry in the recent past, covering various aspects such as new application launches, updates, trends, mergers and acquisitions, funding, and regulations.

New application launches

One of the most significant developments in the application industry is the launch of new applications that offer innovative solutions to various problems. In recent years, we have seen the rise of various applications that offer new ways of doing things, making life easier for people. For instance, the launch of Clubhouse, an audio-only social media platform, has garnered significant attention. The platform allows users to join virtual rooms and participate in discussions on various topics. Similarly, the launch of an application called Calm, which offers meditation and sleep assistance, has been a hit among people looking to improve their mental health and well-being.

Updates

Application updates are another essential aspect of the industry. Regular updates help applications to improve their features, fix bugs, and enhance their overall user experience. In recent years, we have seen significant updates to various popular applications. For instance, the social media platform Instagram launched a significant update that saw the introduction of Reels, a new feature that allows users to create short-form videos. Similarly, the popular video-conferencing application Zoom introduced new features that allow users to add virtual backgrounds and improve the overall quality of video and audio.

Trends

The application industry is constantly evolving, and there are several trends that are shaping the industry's future. One of the most significant trends in recent years has been the rise of on-demand services. Applications such as Uber, Grubhub, and DoorDash have revolutionized the way people access various services such as transportation and food delivery. Another significant trend is the use of artificial intelligence and machine learning in applications. Applications that use AI and ML are becoming increasingly popular, as they offer users personalized experiences and recommendations based on their preferences.

Mergers and Acquisitions

Mergers and acquisitions are a common occurrence in the application industry. In recent years, we have seen several high-profile mergers and acquisitions that have reshaped the industry. For instance, in 2020, Microsoft acquired ZeniMax Media, the parent company of popular video game publisher Bethesda Softworks. Similarly, the social media giant Facebook acquired Giphy, a popular GIF search engine, in a deal worth \$400 million.

Funding

Funding is crucial for the growth and success of applications, and the industry has seen a surge in investment in recent years. In 2020, the total investment in the application industry reached \$73 billion, up from \$56 billion in 2019. The investment has been driven by the increasing demand for new and innovative applications, as well as the potential for high returns. Some of the most significant funding rounds in recent years include the \$1.5 billion raised by Epic Games, the developer of the popular video game Fortnite, and the \$2.5 billion raised by the Chinese ride-hailing giant Didi Chuxing.

Regulations

Regulations are another crucial aspect of the application industry. As the industry grows, governments and regulatory bodies are becoming increasingly concerned about issues such as data privacy and security. In recent years, several countries have introduced new regulations aimed at protecting user data and ensuring that applications operate in a fair and transparent manner. For instance, in 2018, the European Union introduced the General Data Protection Regulation (GDPR), which gives users more control over their data and imposes significant fines on companies that fail to comply.

1.2 purposes

A headlines application report is a document that describes the purpose, design, implementation, and evaluation of a software application that generates news headlines automatically. The purpose of such an application is to assist journalists, news editors, and media companies in generating headlines quickly and efficiently, based on the content of news articles. In this report, we will discuss the purpose of a headlines application, the design and implementation of such an application, and the evaluation of its effectiveness.

Purpose of a headlines application:

The purpose of a headlines application is to provide a tool for journalists and editors to quickly generate headlines for news articles. In traditional journalism, headlines are often written manually by journalists or editors, which can be a time-consuming process. However, with the increasing amount of news content being published on the internet, the need for automated tools to generate headlines has become more important. A headlines application can help to reduce the time and effort required to generate headlines by automatically analyzing the content of news articles and generating headlines that are relevant, concise, and attention-grabbing. The application can also help to ensure that headlines are consistent in style and tone across different news articles.

Design and implementation of a headlines application:

The design and implementation of a headlines application involve several components, including data acquisition, text analysis, and headline generation. Data acquisition involves collecting news articles from various sources, such as RSS feeds, social media platforms, and news websites. The collected data is then preprocessed to remove irrelevant information and convert it into a standardized format for analysis.

Text analysis involves using natural language processing (NLP) techniques to analyze the content of news articles and extract key information, such as the main topic, sentiment, and tone. NLP algorithms can also identify important keywords and phrases that can be used to generate headlines. Headline generation involves using machine learning algorithms, such as neural networks or decision trees, to generate headlines

based on the analyzed content of news articles. These algorithms can learn from existing headlines to generate new ones that are relevant and attention-grabbing.

Evaluation of a headlines application:

The effectiveness of a headlines application can be evaluated based on several metrics, including headline quality, accuracy, and relevance. Headline quality refers to the readability, clarity, and conciseness of the generated headlines. A good headline should be easy to understand, informative, and attention-grabbing.

Accuracy refers to the ability of the headlines application to accurately reflect the content of news articles. The headlines should be relevant to the content of the article and not misleading or sensationalized.

Relevance refers to the ability of the headlines application to generate headlines that are relevant to the target audience. The headlines should be tailored to the interests and preferences of the audience and reflect the tone and style of the news outlet.

Conclusion:

In conclusion, a headlines application can provide significant benefits for journalists, editors, and media companies by reducing the time and effort required to generate headlines for news articles. The design and implementation of a headlines application involve several components, including data acquisition, text analysis, and headline generation, and the effectiveness of the application can be evaluated based on headline quality, accuracy, and relevance. With the increasing amount of news content being published on the internet, the need for automated tools to generate headlines has become more important, and a headlines application can help to ensure that news outlets remain competitive and relevant in the digital age.

2. PROJECT DEFINITION AND DESIGN THINKING

Empathy map

Use this framework to develop a deep, shared understanding and empathy for other people. An empathy map helps describe the aspects of a user's experience, needs and pain points, to quickly understand your users' experience and mindset.

[Share template feedback](#)

Build empathy

The information you add here should be representative of the observations and research you've done about your users.

Says

What have we heard them say?
What can we imagine them saying?

Thinks

What are their wants, needs, hopes, and dreams? What other thoughts might influence their behavior?

Does

What behavior have we observed?
What can we imagine their doing?

Feels

What are their fears, frustrations, and unmet needs? What other feelings might influence their behavior?

Give them a name and a portrait to empathize with your persona.

Need some inspiration?
See a finished version of this template to kickstart your work.

[Open example](#)

2.1 Empathy Map

Brainstorm & idea prioritization

Use this template to generate ideas, brainstorm, and prioritize them. It helps you to generate ideas, brainstorm, and prioritize them. It helps you to generate ideas, brainstorm, and prioritize them.

[View example](#)

Define your goals and objectives

Use this template to define your goals and objectives. It helps you to define your goals and objectives. It helps you to define your goals and objectives.

[View example](#)

Brainstorm

Use this template to brainstorm ideas. It helps you to brainstorm ideas. It helps you to brainstorm ideas.

[View example](#)

Idea prioritization

Use this template to prioritize ideas. It helps you to prioritize ideas. It helps you to prioritize ideas.

[View example](#)

Implementation

Use this template to implement ideas. It helps you to implement ideas. It helps you to implement ideas.


[View example](#)

Review


Use this template to review ideas. It helps you to review ideas. It helps you to review ideas.


[View example](#)

3. RESULT



Login

 username

 password

Log In

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

3.1 Login page of project

Sign Up



username



password



email

Register

Have an account? [Log in](#)

3.2 sign page of project

Latest NEWS



Top Crypto Trader Issues Bitcoin Warning, Says BTC Flashing Vibes of March 2020 Meltdown - The Daily Hodl

A closely followed crypto trader is



Saudi oil giant Aramco posts record \$161.1 billion profit for 2022 - CNBC

Saudi state-controlled Aramco achieved a record profit last year, boosted by higher energy prices

Silicon Valley Bank: the spectacular unravelling of the tech industry's banker - Financial Times

News, analysis and comment from the Financial Times, the



Meta looking into new social platform that could rival Twitter - The Hill

Meta, Facebook's parent company, is looking into a new social media platform that could



Crypto Exhales as USDC Stablecoin Rebounds

3.3 activity intro page

Top Crypto Trader Issues Bitcoin Warning, Says BTC Flashing Vibes of March 2020 Meltdown - The Daily Hodl

A closely followed crypto trader is issuing an alert to Bitcoin (BTC) holders, saying that the king crypto's current market structure looks similar to its price action prior to a deep plunge about three years ago.



3.4 Brief news page

4.ADVANTAGES AND DISADVANTAGES

Creating a headlines application can have several advantages and disadvantages, some of which are listed below:

Advantages:

Quick access to news: A headlines application can provide users with quick and easy access to news stories from various sources in one place, making it easier for users to stay informed.

Personalization: A headlines application can allow users to customize the types of news they want to see, based on their interests, location, and other preferences.

Real-time updates: With a headlines application, users can receive real-time updates on breaking news stories and events, which can be particularly useful during times of crisis or emergencies.

Increased engagement: A headlines application can help increase engagement with news content, as users can easily browse and share stories with others.

Disadvantages:

Overwhelming amount of information: A headlines application can present users with an overwhelming amount of information, making it difficult for them to filter out the noise and find stories that are relevant to their interests.

Bias: The sources of news stories in a headlines application may have their own biases, which can affect the presentation of news and potentially influence users' perspectives.

Lack of context: A headlines application may only provide a brief summary of a news story, without providing the necessary context or background information needed to fully understand the story.

Dependence on technology: A headlines application requires a stable internet connection and a device to access it, which can limit access to news for people in areas with poor internet connectivity or who cannot afford a device.

Overall, creating a headlines application can have benefits and drawbacks, and it is important to consider these factors when developing such an application.

5. APPLICATION PREVIEWS

In recent times, there have been several significant news headlines that have garnered worldwide attention. From political upheavals to technological advancements, these events have shaped the global landscape in various ways. In this article, we will look at some of the latest news headlines that have dominated the headlines in recent months.

COVID-19 Pandemic: The ongoing COVID-19 pandemic continues to be a major news headline worldwide. With the emergence of new variants and the slow rollout of vaccines in some parts of the world, the pandemic continues to impact daily life and the global economy. The emergence of the Omicron variant in late 2021 and early 2022 also sparked concerns, with countries implementing new restrictions and travel bans.

Political Turmoil in Myanmar: Myanmar has been in a state of political unrest since the military coup in February 2021. The military junta has cracked down on pro-democracy protesters and opposition leaders, resulting in widespread human rights abuses and international condemnation. The country's leader, Aung San Suu Kyi, remains under house arrest, and the future of democracy in Myanmar remains uncertain.

Climate Change: Climate change continues to be a pressing global issue, with record-breaking heatwaves, wildfires, and floods affecting various parts of the world. The United Nations Climate Change Conference (COP26) held in November 2021 saw world leaders gathering to discuss ways to address climate change and implement the goals of the Paris Agreement.

Technological Advancements: In the world of technology, several significant advancements have made headlines in recent months. In December 2021, SpaceX successfully launched its Starship rocket prototype, marking a significant step towards space exploration. Meanwhile, the metaverse and its potential impact on various industries, including gaming and advertising, continue to be a hot topic of discussion.

Sports: In the world of sports, the Winter Olympics held in Beijing in February 2022 was a major event, with athletes from around the world competing in various winter sports. The tournament was held amid controversy, with concerns over human rights abuses and the COVID-19 pandemic. In football, the 2022 World Cup in Qatar was also a significant event, with the tournament being held in November and December due to concerns over the extreme heat in Qatar during the summer months.

Global Economy: The global economy has been impacted by various factors, including the COVID-19 pandemic and supply chain disruptions. In November 2021, the world's largest economies signed the historic "Build Back Better World" (B3W) infrastructure plan, aimed at providing an alternative to China's Belt and Road Initiative. Meanwhile, inflation and rising prices have become a concern for consumers worldwide.

6.CONCLUSION

In conclusion, creating a headlines application can be a challenging but rewarding process. A headlines application is a great way to stay up to date with the latest news and trends in various industries. By following the steps outlined in this article, you can develop a headlines application that is both functional and user-friendly. The first step in creating a headlines application is to define your target audience and their needs. This will help you design the application in a way that meets their requirements and provides value. You should also research the competition and analyze their strengths and weaknesses to differentiate your application from theirs.

Once you have a clear understanding of your target audience and competition, the next step is to choose the right platform and technology stack. The platform and technology stack you choose will depend on your budget, resources, and development expertise. It is important to choose a platform and technology stack that are scalable, secure, and can support future updates and enhancements.

Designing the user interface and user experience of your headlines application is critical to its success. A well-designed user interface will make your application easy to use and intuitive for your target audience. The user experience should be engaging and encourage users to spend more time on the application. You should also consider adding features like push notifications, social media sharing, and personalization to enhance the user experience.

Developing the functionality of your headlines application will require you to work closely with your development team. It is important to prioritize the features that are most important to your target audience and develop them first. You should also conduct regular testing and quality assurance to ensure that your application is bug-free and performs well.

Launching your headlines application is an exciting milestone, but it is not the end of the journey. You should continue to monitor user feedback and usage patterns to identify areas for improvement. Regular updates and enhancements will keep your application relevant and engaging for your target audience.

7. FUTURE ENHANCEMENTS

A some general enhancements that could be beneficial for any application that involves creating or curating headlines:

Natural Language Processing (NLP) capabilities: NLP is a branch of artificial intelligence that focuses on the interaction between computers and human language. Incorporating NLP capabilities into your application can help improve the accuracy and relevance of the headlines generated by the app, as well as enable features like sentiment analysis and entity recognition.

Machine Learning (ML) algorithms: ML algorithms can be trained to analyze large amounts of data and generate insights that can help optimize the headlines created by the app. For example, the app could use ML to analyze user engagement data and identify which types of headlines are most effective in driving clicks and shares.

Content personalization: Personalization is becoming increasingly important in today's digital landscape. By leveraging data about a user's interests, browsing behavior, and demographic information, your app could create tailored headlines that are more likely to resonate with the user.

Integration with social media platforms: If your app is designed to help users create headlines for social media posts, integrating it with social media platforms like Facebook, Twitter, and Instagram could be beneficial. This would allow users to easily share their headlines on these platforms and track engagement metrics.

Collaboration features: If your app is intended for use by teams or multiple users, incorporating collaboration features could be helpful. For example, users could be able to share headline ideas and collaborate on headlines in real-time.

Image and video recognition: In addition to text-based headlines, your app could also incorporate image and video recognition capabilities. This would allow users to generate headlines based on visual content, which can be especially useful for platforms like Instagram and TikTok.

Trend analysis: Finally, incorporating trend analysis capabilities into your app could help users stay on top of the latest news and trends in their industry. This could include features like real-time news alerts, social media monitoring, and trend tracking.

8. APPENDIX

```
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?,

)

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)

}

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

        }

    }

}

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_PASSWORD)),
)
}
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(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)),
)
users.add(user)
} while (cursor.moveToNext())
}
cursor.close()
db.close()
}

```

```

return users
}

}

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=684cb893caf7425abeffad82ac1d0f4e")

    /// @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.create())

                .build().create(ApiService::class.java)

            }

            return apiService!!

        }

    }

}

package com.example.newsheadlines

data class Movie(val name: String,

```

```

val imageUrl: String,
val desc: String,
val category: String)

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()
)

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
)

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,
)

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()

            }

        }

    }

}

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)
}

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",

        "",

        " article1"

    )

    MovieItem(context, movie = movie, 0, 0) { i ->

        Log.i("wertetest123abc", "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) }
    )
}
}

package com.example.newsheadlines

import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.widget.TextView
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column

```

```

import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.padding
import androidx.compose.material.MaterialTheme
import androidx.compose.material.Surface
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.compose.ui.viewinterop.AndroidView
import androidx.core.text.HtmlCompat
import coil.compose.rememberImagePainter
import com.example.newsheadlines.ui.theme.NewsHeadlinesTheme

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

```

```

Log.i("test123abc", "MovieItem: $desk")

Column(Modifier.background(Color.Gray).padding(20.dp),                horizontalAlignment =
Alignment.CenterHorizontally, verticalArrangement = Arrangement.Center) {

    Text(text = ""+title, fontSize = 32.sp)

    HtmlText(html = desk.toString())

    /* AsyncImage(

    model = "https://example.com/image.jpg",

    contentDescription = "Translated description of what the image contains"

    )*/

    Image(

    painter = rememberImagePainter(uriImage),

    contentDescription = "My content description",

    )

    }

    // Greeting(desk.toString())

    }

    }

    }

    }

    }

    }

    @Composable

    fun Greeting(name: String) {

    // Text(text = "Hello $name!")

    }

    @Preview(showBackground = true)

    @Composable

    fun DefaultPreview() {

    NewsHeadlinesTheme {

    // Greeting("Android")

```

```

}

}

@Composable
fun HtmlText(html: String, modifier: Modifier = Modifier) {
    AndroidView(
        modifier = modifier,
        factory = { context -> TextView(context) },
        update = { it.text = HtmlCompat.fromHtml(html, HtmlCompat.FROM_HTML_MODE_COMPACT) }
    )
}

<?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_news"
            android:theme="@style/Theme.NewsHeadlines" />

```

```

<activity
    android:name=".RegistrationActivity"
    android:exported="false"
    android:label="@string/title_activity_registration"
    android:theme="@style/Theme.NewsHeadlines" />

<activity
    android:name=".MainPage"
    android:exported="false"
    android:label="@string/title_activity_main_page"
    android:theme="@style/Theme.NewsHeadlines" />

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

    <intent-filter>

        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />

    </intent-filter>

</activity>

</application>

</manifest>

```