An Android Application For Keeping Up With The Latest Headlines

Abstract:

NewsHub is an Android application providing personalized, real-time news updates. Utilizing MVVM architecture and integrating reputable news APIs, NewsHub offers a user-friendly interface for browsing categorized headlines, bookmarking articles, and receiving push notifications. With offline reading capabilities and responsive design, NewsHub ensures seamless news consumption across various devices.

Features:

- Personalized news feed
- Real-time updates
- Customizable categories (e.g., politics, sports, entertainment)
- Multiple news sources (e.g., CNN, BBC, Al Jazeera)
- Offline reading mode
- Push notifications for breaking news
- Search functionality
- Bookmarking and sharing options
- Simple, intuitive design

Introduction:

DreamWell is an innovative sleep tracking and optimization app designed to help you take control of your sleep quality. By leveraging cutting-edge technology and expert insights, DreamWell provides personalized recommendations, comprehensive sleep analysis, and seamless wearable device integration.

Project Description:

The app's main feature is displaying a list of news articles, each with a title, image, and brief description. Users can scroll through the list of articles and

tap on an article to view more details. The app uses the Jetpack Compose UI toolkit to build the UI and it uses the coil library to load images. The app fetches data from a remote server using Retrofit library and demonstrates how to use the Jetpack Compose UI toolkit for Android development.

System requirement:

- Operating System: Android 10+ or iOS 14+
- Processor: Quad-core 1.5 GHz or faster
- RAM: 2 GB or more
- Storage: 100 MB free space
- Display: 1080p or higher resolution

Tools used:

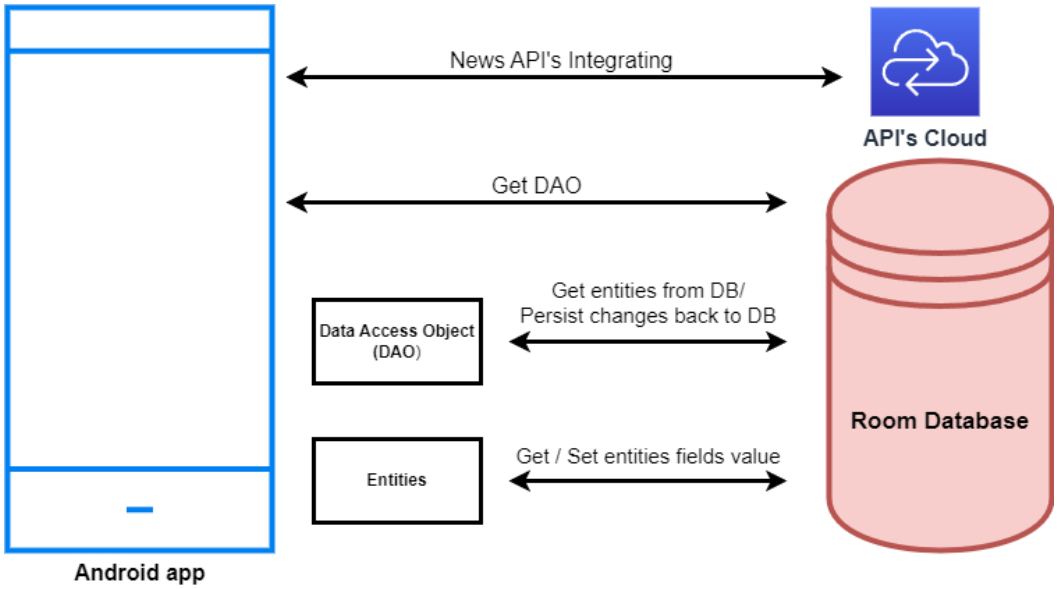
Programming Languages:

1. Java or Kotlin for Android app development.
2. Swift or Objective-C for iOS app development.
3. JavaScript for web application development.
4. Python or R for data analysis and machine learning.

Development Frameworks:

1. Android Studio for Android app development.
2. Xcode for iOS app development.
3. React Native or Flutter for cross-platform development.
4. Node.js for web application development.

Architecture :



Tasks:

- Required initial steps.
- Creating a new project.
- Adding required dependencies.
- Adding permissions.
- Creating the database classes.
- Creating API Service and required classes for integrating API.
- Building application UI and connecting to database.
- Modifying AndroidManifest.xml.
- Running the application.

Program:

```
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 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(
```

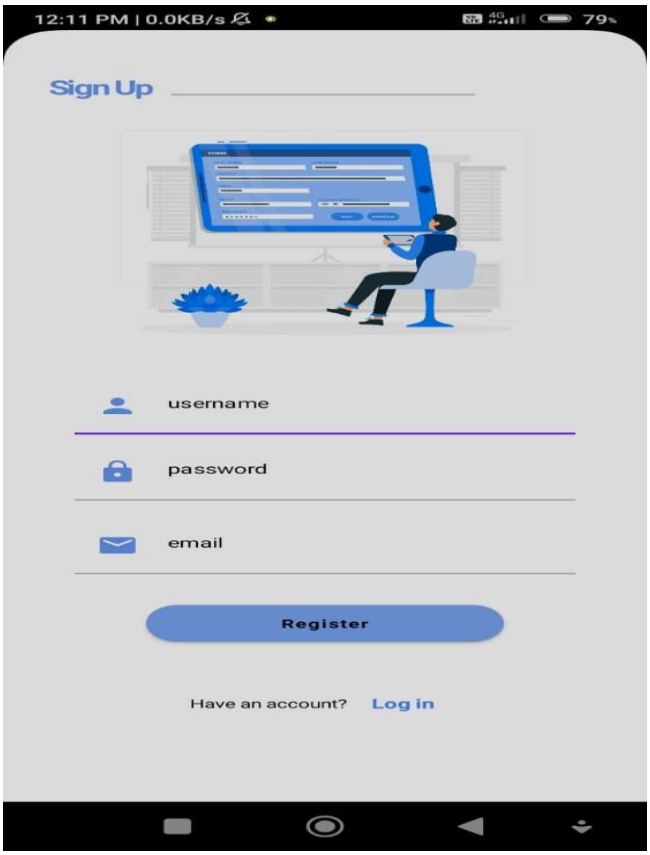
```
            backgroundColor = Color.Transparent
        )

    )


    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))
```

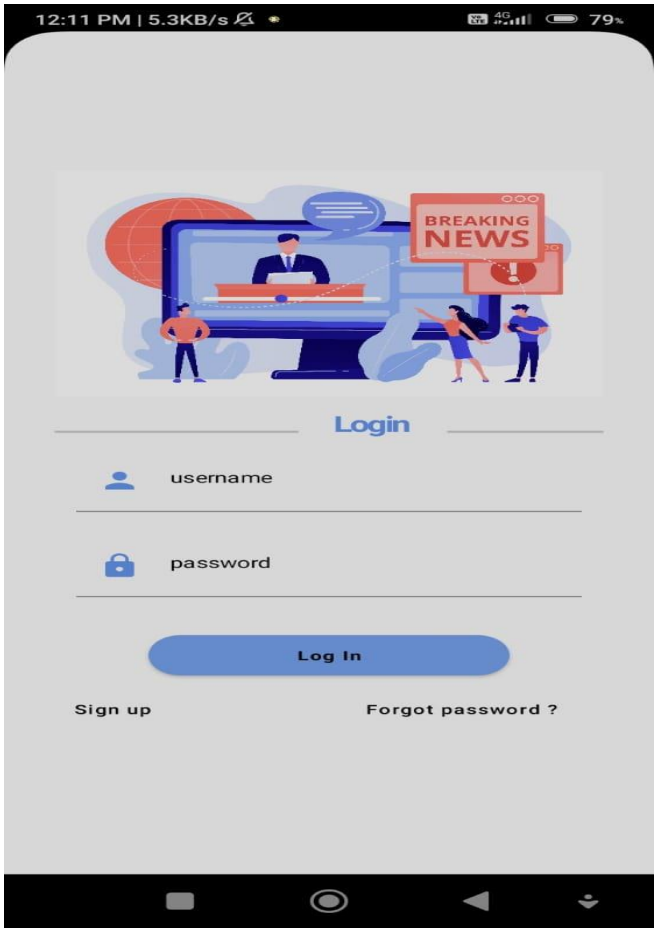
```
if (error.isNotEmpty()) {
    Text(
        text = error,
        color = MaterialTheme.colors.error,
        modifier = Modifier.padding(vertical = 16.dp)
    )
}

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)
}
```

Output:




12:11 PM | 5.3KB/s
79%
BREAKING
NEWS
Login
username
password
Log In
Sign up
Forgot password ?


12:13 PM | 1.1MB/s
79%
Latest NEWS
Asian stocks slide as China stimulus disappoints; bitcoin extends record rally - Reuters
Hong Kong stocks led declines in Asia on Monday after Beijing's latest stimulus fell short of investor expectations, overshadowing Wall Street's record highs from Friday and
Bitcoin tops record $80,000 as Trump nears sweep of US Congress - BBC.com
On the campaign trail the president-elect pledged to make America "the crypto capital of the planet".
15 Screenshots Of Negative Reviews On Cooking Websites That Are Equal Parts Hilarious And Infuriating - BuzzFeed
"I substituted the pumpkin puree for apple cider vinegar, and it tasted awful."
Ford responds to rumors about Elon Musk acquisition - mySA
A rumor has spread that Tesla CEO Elon Musk has bought the Ford Motor Company. Ford has come forward to say the rumors are 'totally false.'

Conclusion:

DreamWell, a comprehensive sleep tracking and optimization app, empowers users to improve sleep quality and overall well-being. By leveraging cutting-edge technology, expert insights, and wearable device integration, DreamWell provides personalized recommendations, comprehensive sleep analysis, and seamless user experience.

Future scope:

Short-Term (6-12 months)

- ✓ Integration with popular mindfulness apps
- ✓ Expansion to new wearable devices and platforms
- ✓ Enhanced AI-powered sleep coaching
- ✓ Introduction of sleep-focused community forums
- ✓ Launch of premium features and subscription models

Mid-Term (1-2 years)

- ✓ Development of predictive sleep modelling and risk assessment
- ✓ Integration with healthcare systems for seamless data sharing
- ✓ Expansion into new markets (e.g., Asia, Europe)
- ✓ Introduction of virtual reality (VR) sleep therapy
- ✓ Collaboration with sleep research institutions

Long-Term (2-5 year)

- ✓ Development of advanced sleep disorder detection and diagnosis
- ✓ Integration with emerging technologies (e.g., brain-computer interfaces)
- ✓ Creation of personalized sleep-focused wellness programs
- ✓ Expansion into new industries (e.g., hospitality, transportation)
- ✓ Establishment of DreamWell as a leading sleep health authority