1.INTRODUCTION

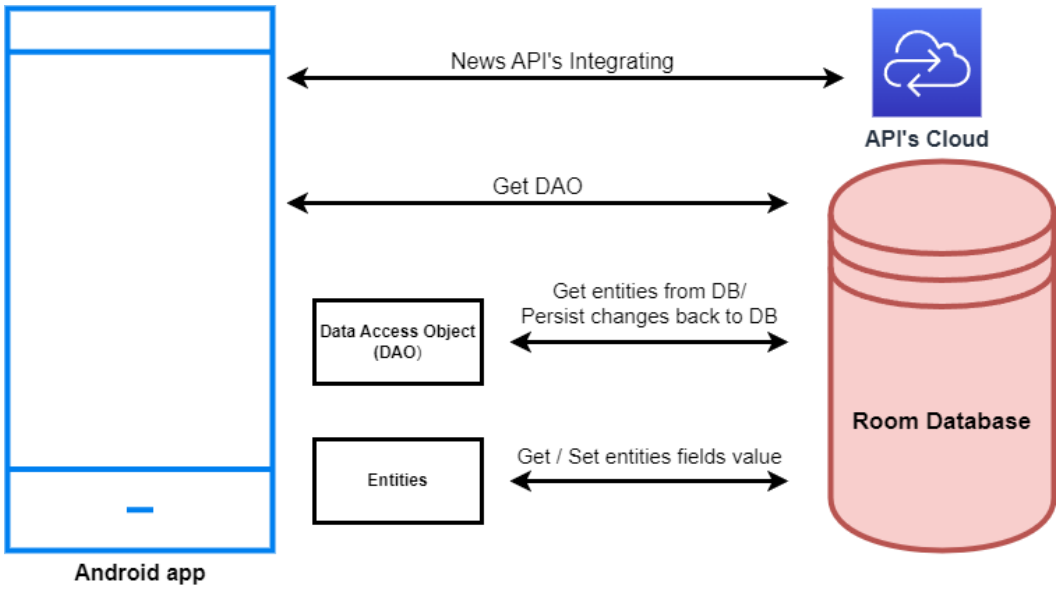
1.1 PROJECT OVERVIEW

An Android Application For Keeping Up With The Latest Headlines

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.

Architecture :



Learning Outcomes :

By end of this project:

- You'll be able to work on Android studio and build an app.
- You'll be able to integrate the database accordingly.
- You'll be able to integrate the API's accordingly.

Project Workflow:

- Users register into the application.
- After registration , user logins into the application.
- User enters into the main page

Note:

To complete the project you need to finish up the tasks listed below:

Tasks:

1.Required initial steps
2.Creating a new project.
3.Adding required dependencies.
4.Adding permissions
5.Creating the database classes.
6.Creating API Service and required classes for integrating API
7.Building application UI and connecting to database.
8.Modifying AndroidManifest.xml
9.Running the application.

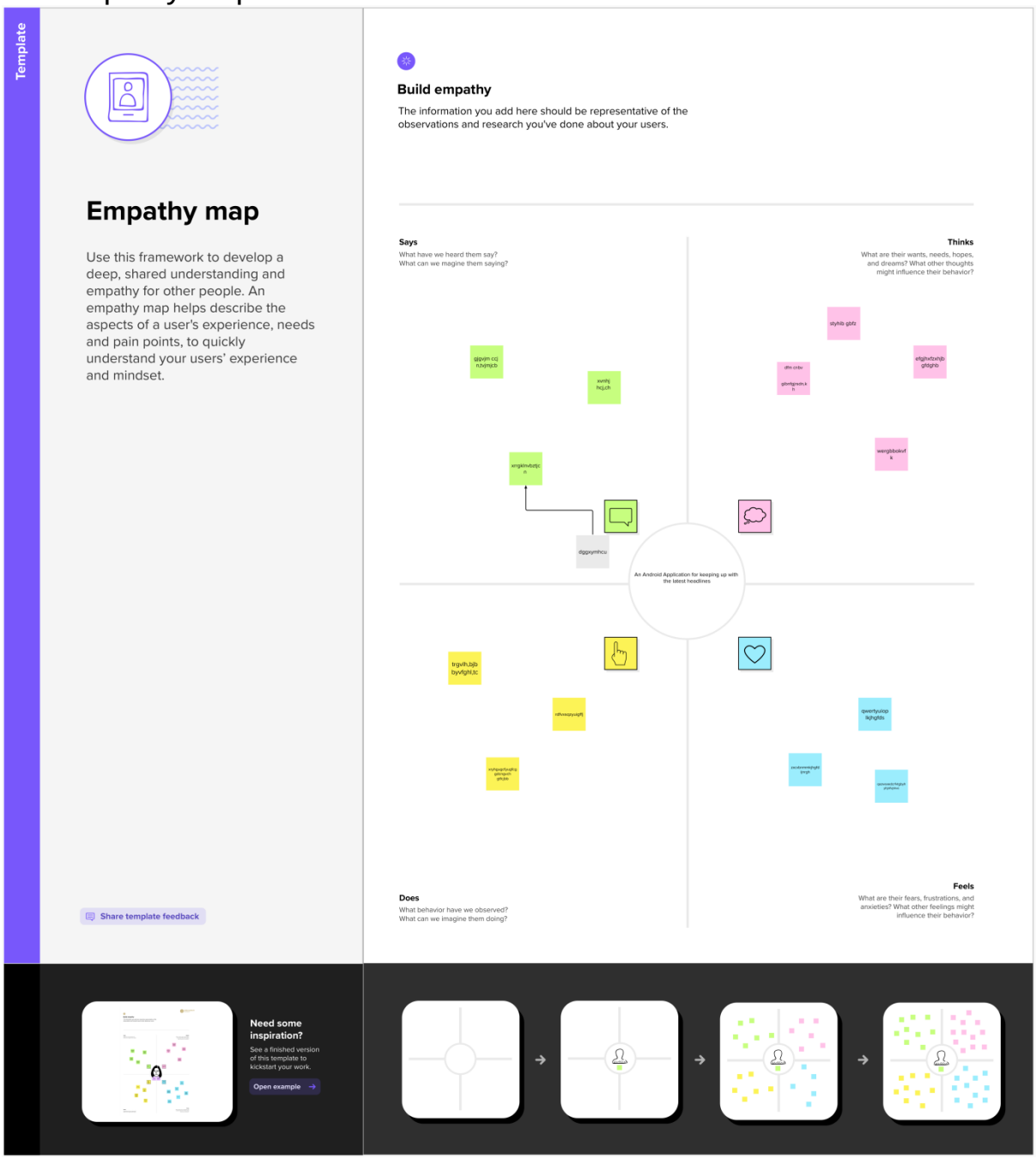
1.2

1.3 Purpose of project

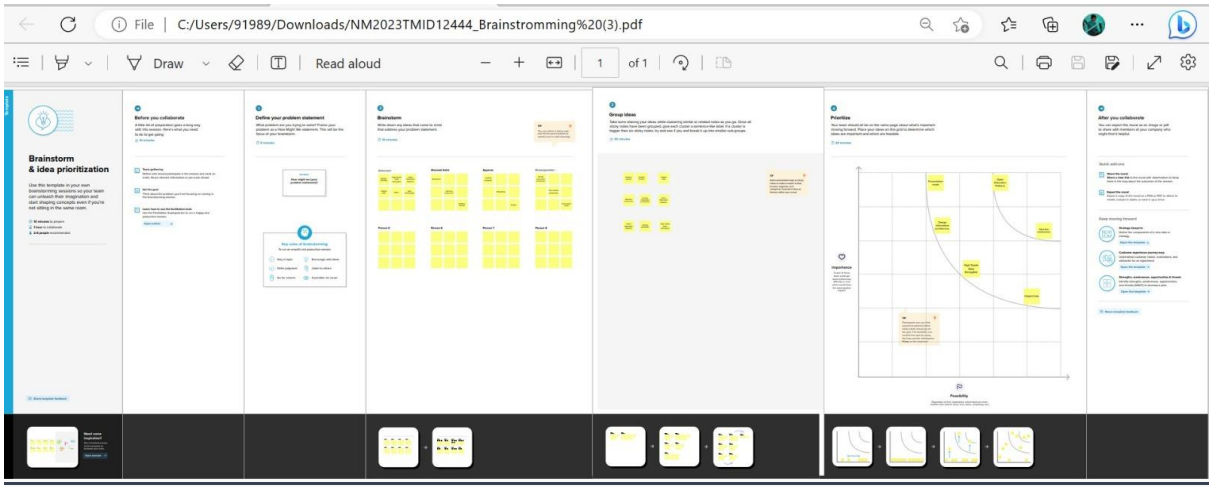
- Convenience: Many people lead busy lives and may not have the time to browse multiple news sources every day. An Android app that delivers the latest headlines directly to their device can be a convenient way for them to stay up-to-date with current events.
- Personalization: A news app can be customized to show news stories that are relevant to the user's interests. By allowing users to select the topics they care about, the app can provide a personalized news experience.
- Accessibility: An Android app for news can make it easier for people with disabilities to access news content. For example, a visually impaired person can use text-to-speech software to have the news stories read to them.
- Real-time updates: With a news app, users can receive real-time updates on breaking news stories as they happen. This can be especially useful in situations where timely information is critical.
- Multilingual support: An Android news app can support multiple languages, making it accessible to users around the world. This can help people stay informed about global events and news stories.

2.Problem Definition & Design Thinking

2.1 Empathy map:



2.2 Ideation&Brainstroming map:


File | C:/Users/91989/Downloads/NM2023TMID12444_Brainstromming%20(3).pdf
Draw
Read aloud
1
of 1
Brainstorm
& idea prioritization

3.Result

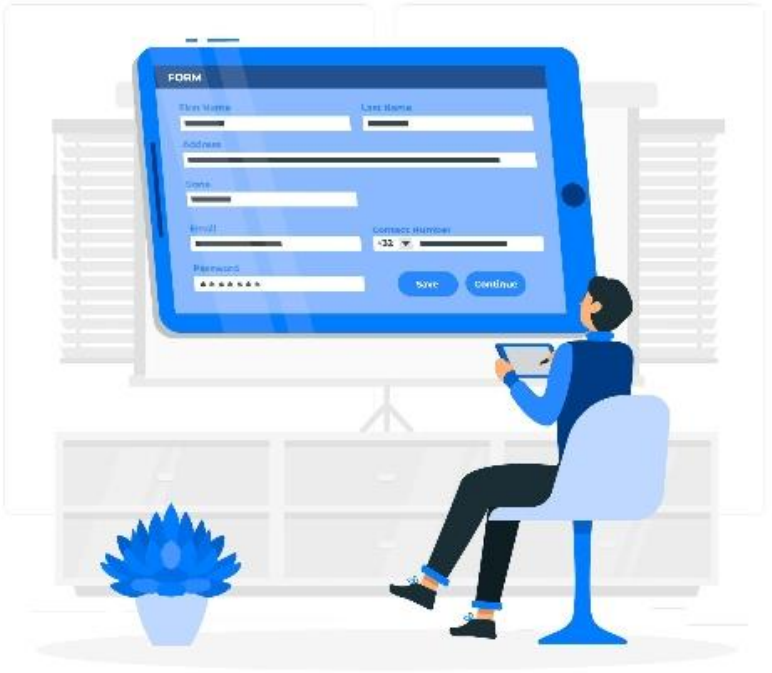
Login

username

password

Log In

Sign Up



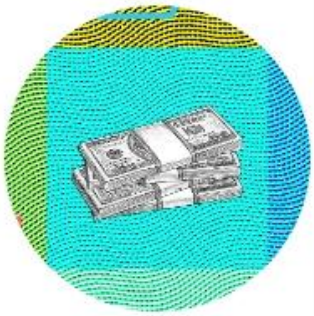
username

password

email

Register

Latest NEWS



Stock Market Today: Dow Futures Edge Up After Bank of America, Goldman Sachs Earnings - The Wall Street Journal

Live updates on stocks and financial news, including the

Stocks making the biggest moves premarket: Johnson & Johnson, Goldman Sachs, Sunrun and more - CNBC

Shares of Goldman Sachs shed nearly 4% after missing revenue

Bank of America, Goldman Sachs, J&J, Lockheed, Netflix, and More Market Movers - Barron's

Bank of America, Goldman Sachs, and Johnson & Johnson post first-quarter earnings that

Stock Market Today: Dow Futures Edge Up After Bank of America, Goldman Sachs Earnings - The Wall Street Journal

Live updates on stocks and financial news, including the S&P 500, Dow and Nasdaq Composite.

4.Advantages and Disadvantages

Advantages

Convenience: With an Android app, users can access the latest headlines and news stories from anywhere at any time, as long as they have an internet connection. This eliminates the need to visit different news websites or tune in to TV news broadcasts, making it a more convenient way to stay informed.

Customization: An Android news app can be customized to deliver news stories that are of interest to the user. For example, users can select the topics they want to follow, and the app will deliver news stories on those topics. This personalization can help users save time by avoiding irrelevant news stories.

Real-time updates: An Android news app can deliver real-time updates on breaking news stories. This can be especially useful in situations where timely information is critical, such as during natural disasters or political events.

Multilingual support: An Android news app can support multiple languages, making it accessible to users around the world. This can help users stay informed about global events and news stories.

Rich media: An Android news app can include multimedia elements such as photos, videos, and audio clips, making it a more engaging way to consume news content.

Cost-effective: Many Android news apps are available for free, making it a cost-effective way to stay informed

Disadvatages

Information overload: With so much news available at our fingertips, it can be easy to become overwhelmed by the volume of information. Users may find themselves spending too much time consuming news stories and struggling to filter out the most important information.

Inaccuracy: Not all news sources are trustworthy, and some may publish inaccurate or biased information. Users must be careful when selecting their news sources and should take the time to fact-check important stories.

Technical issues: An Android news app may experience technical issues or glitches that can impact its reliability or functionality. This could include slow loading times, crashes, or other performance issues.

Privacy concerns: Some Android news apps may collect personal data from users, such as their browsing history or location. Users should be careful to read the app's privacy policy and understand how their data will be used.

Dependency on technology: With the convenience of an Android news app comes the risk of becoming too dependent on technology. Users may find themselves relying too heavily on the app and neglecting other important activities or responsibilities.

5.Application

Google News: This app provides personalized news stories based on the user's interests and search history. It also includes news stories from a variety of sources, as well as real-time updates on breaking news.

Flipboard: This app allows users to create a customized news feed by selecting topics they are interested in. It also includes stories from a variety of sources and has a user-friendly interface.

Feedly: This app aggregates news stories from a variety of sources and allows users to organize them into categories. It also has a feature that suggests new sources based on the user's reading history.

BBC News: This app provides news stories from the British Broadcasting Corporation (BBC) and covers both UK and international news. It also includes live streaming of news events and video content.

Reuters News: This app provides news stories from the Reuters news agency and covers a range of topics, including business, finance, and politics. It also includes multimedia content, such as photos and videos.

6.conclusion

In conclusion, an Android application for keeping up with the latest headlines can be a convenient and personalized way to stay informed about the news stories that matter to us. Such applications offer several advantages, including real-time updates,

customization, and access to multimedia content. However, there are also potential disadvantages to consider, such as information overload, accuracy concerns, and technical issues. When selecting an Android news app, it's important to consider personal preferences, reliability of the news sources, and privacy concerns. Ultimately, a well-designed and reliable Android news app can be a valuable tool for staying informed about the world around us. In conclusion, an Android application for keeping up with the latest headlines can be a convenient and personalized way to stay informed about the

news stories that matter to us. Such applications offer several advantages, including real-time updates, customization, and access to multimedia content. However, there are also potential disadvantages to consider, such as information overload, accuracy concerns, and technical issues. When selecting an Android news app, it's important to consider personal preferences, reliability of the news sources, and privacy concerns. Ultimately, a well-designed and reliable Android news app can be a valuable tool for staying informed about the world around us.

7.Future scope

Artificial Intelligence (AI): AI-powered news apps could offer personalized news recommendations based on the user's behavior and preferences. These apps could also use AI to fact-check news stories and weed out fake news.

Voice Assistants: With the rise of voice assistants like Google Assistant and Amazon Alexa, news apps could integrate with these platforms to deliver news stories via voice commands.

Augmented Reality (AR): AR could be used to create immersive news experiences, allowing users to see news stories unfold in real-time and interact with news content in new ways.

Local News: There is a growing demand for hyper-local news stories that cover events in specific neighborhoods or communities. News apps could leverage location-based data to deliver customized local news stories to users.

Social Integration: News apps could integrate with social media platforms to allow users to share news stories with their friends and followers. This could help to increase the reach of important news stories and encourage more social discussion around current events.

8.Appendix

A.source code

Mainactivty

```
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",
        "",
        " 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) }
        )
    }
}
```

Loginativity

```
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)

}
```