

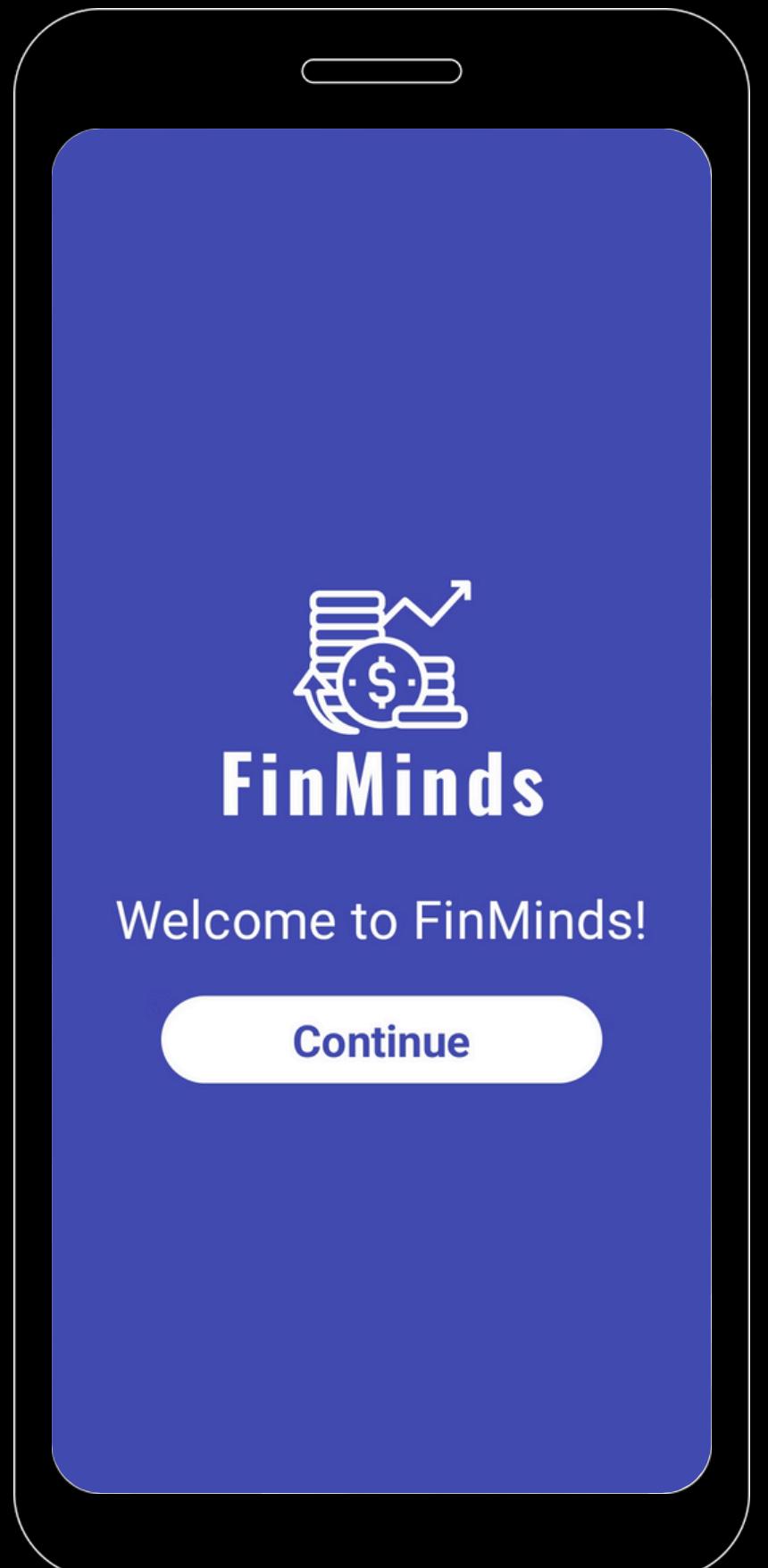
FINMINDS

Team 8

Topic - Pace undergraduate Student's Financial Literacy

CS-639

Submitted to - Prof. Christelle Scharff



TEAM MEMBERS

Lead Developer



Jimmy Karoly

UI Developer



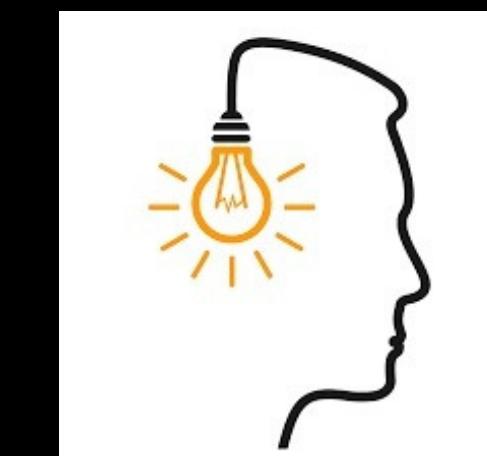
Dhyey Dave

Overview

FinMinds, is a mobile app designed to enhance financial literacy among college students. The app aims to provide an accessible platform for users to gain knowledge and stay updated with the latest financial news.

Problem Statement and Approach

A 2018 study found that only 40% of incoming students had taken a personal finance course. The issue starts in high school, where only five states in the US require a financial education course. There isn't a push in most schools for financial education, and some students go to college unprepared to manage their finances. An app with small educational lessons and quizzes can help a student gain control of their finances.



Vision

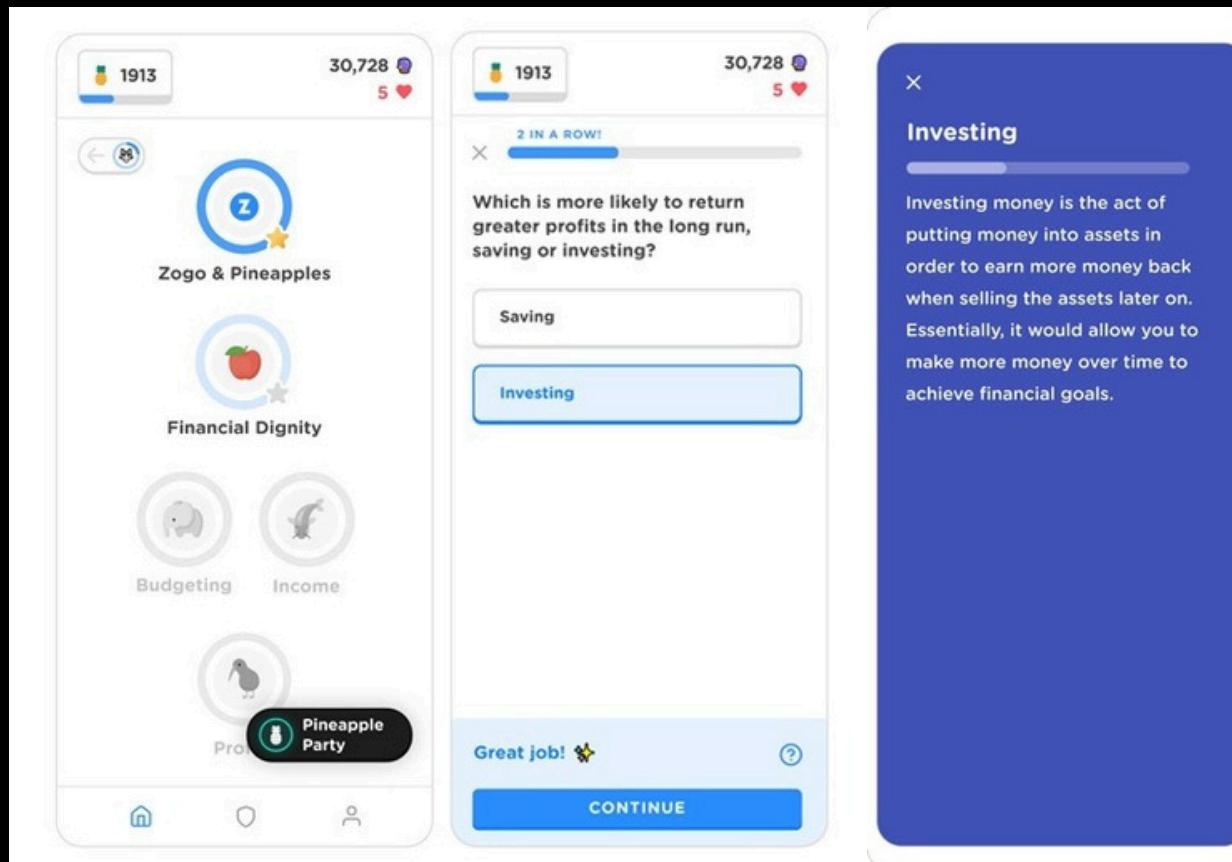
We aim to develop a mobile application that empowers users to easily enhance their understanding of financial matters and cultivate sound financial knowledge. Our goal is to create an enjoyable, user-friendly platform that specifically caters to young individuals, making financial education accessible and engaging. By doing so, we aim to assist them in establishing a solid groundwork for achieving financial prosperity.



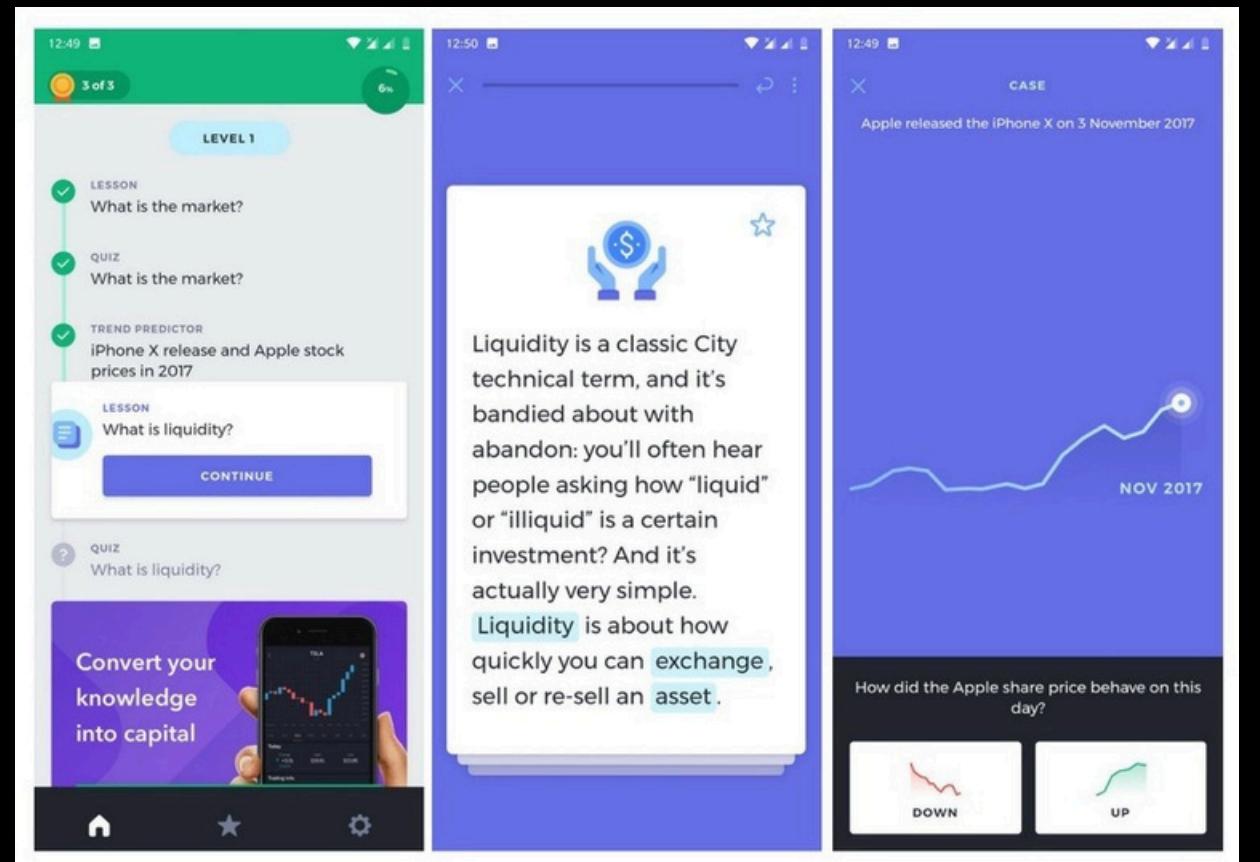
Market Research

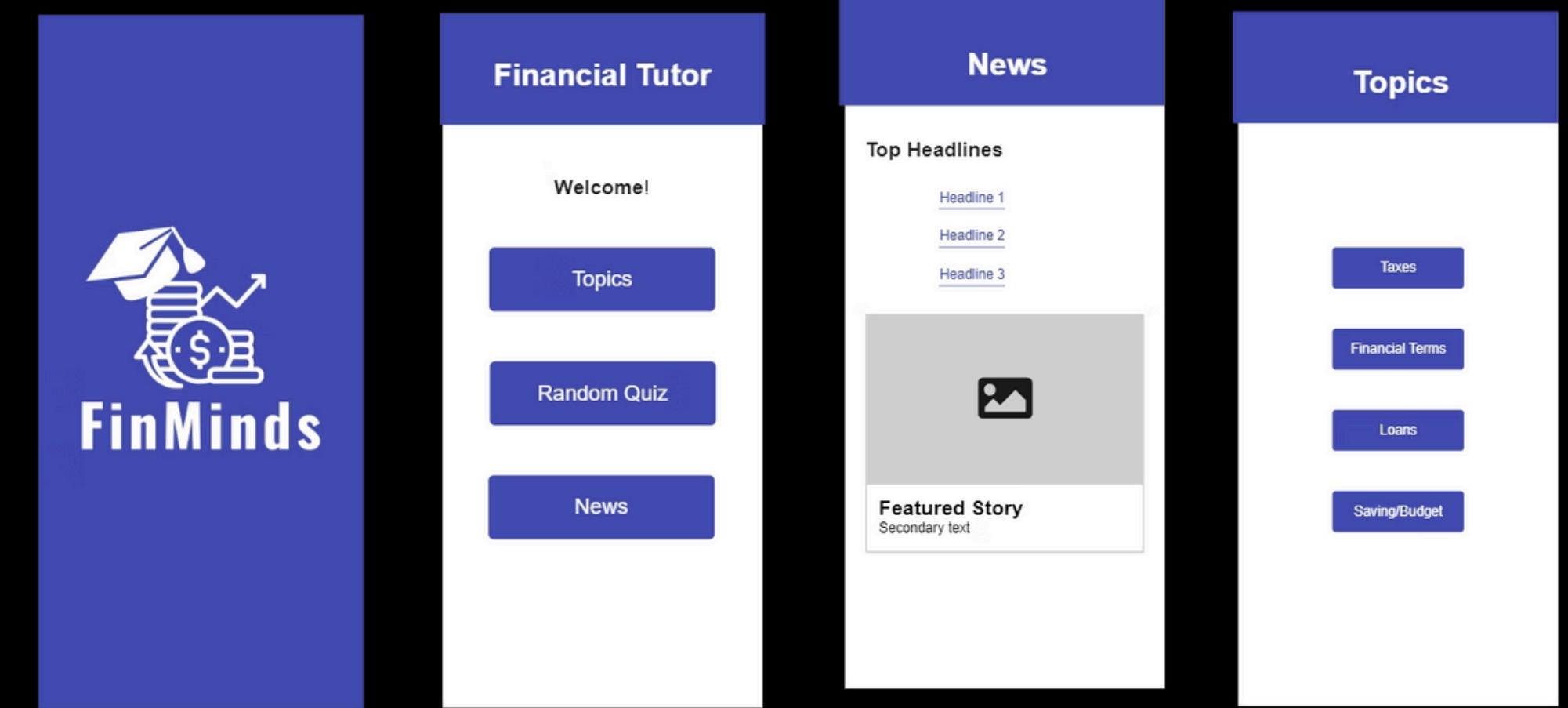
There are two apps we found on the market which could have similar functionality as of our app.

1. Zogo



2. Investmate





Initial Design

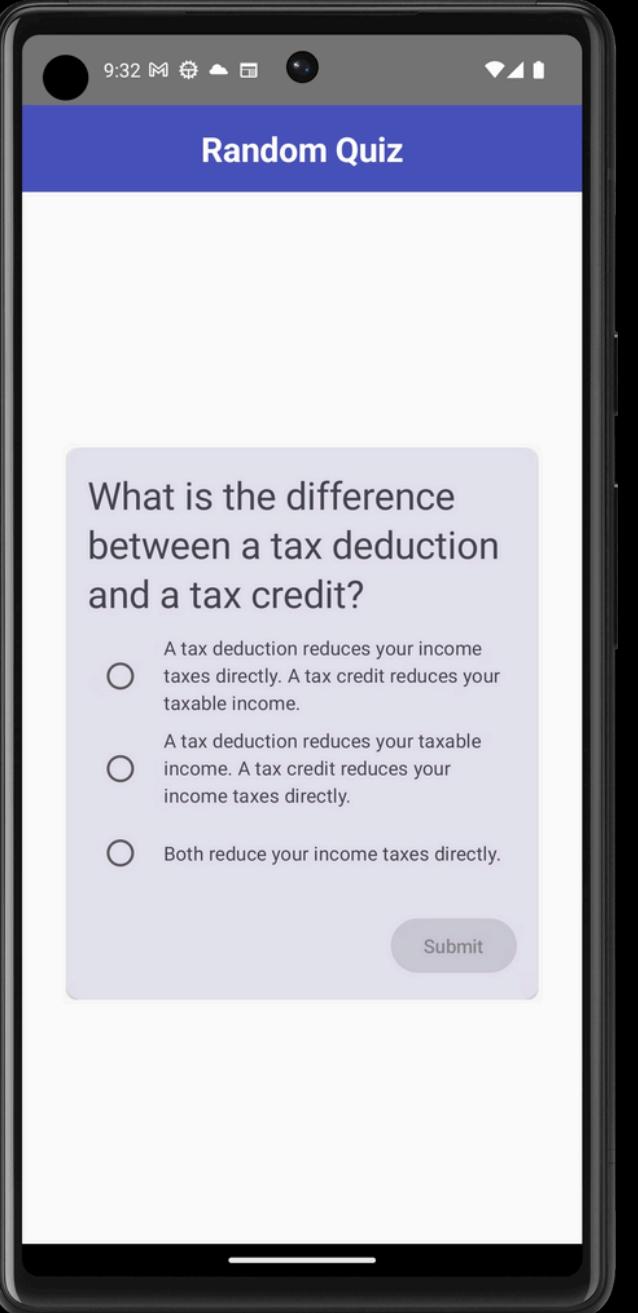
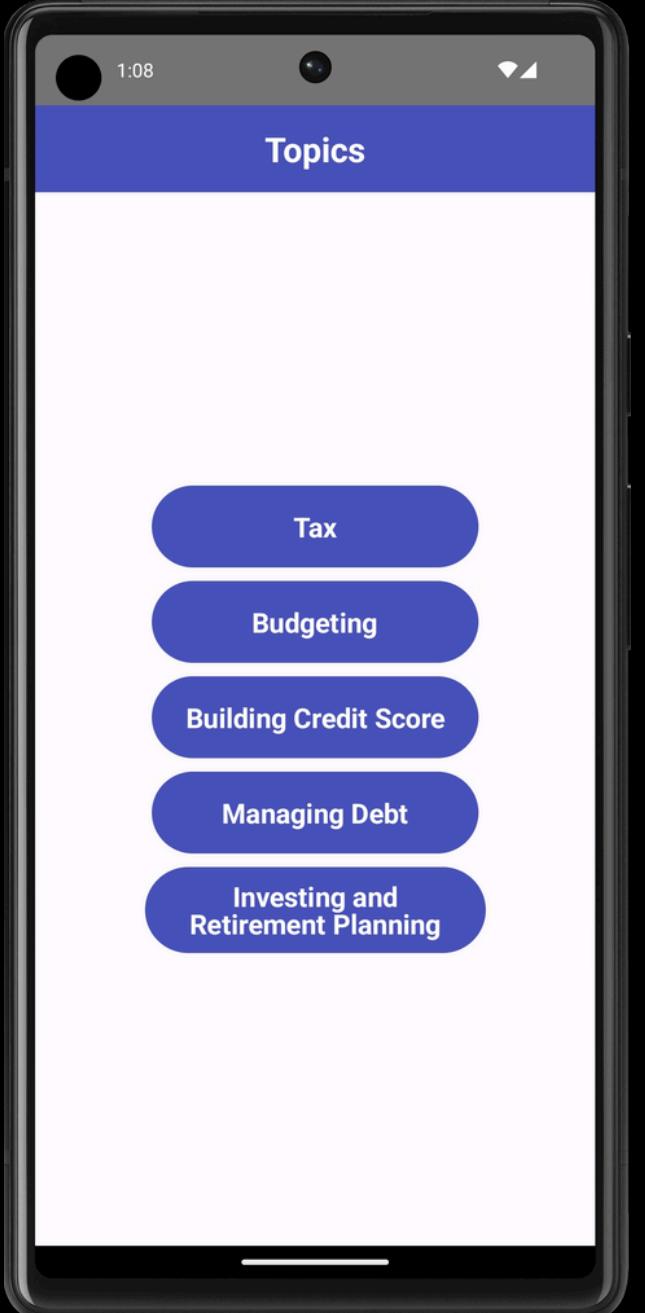
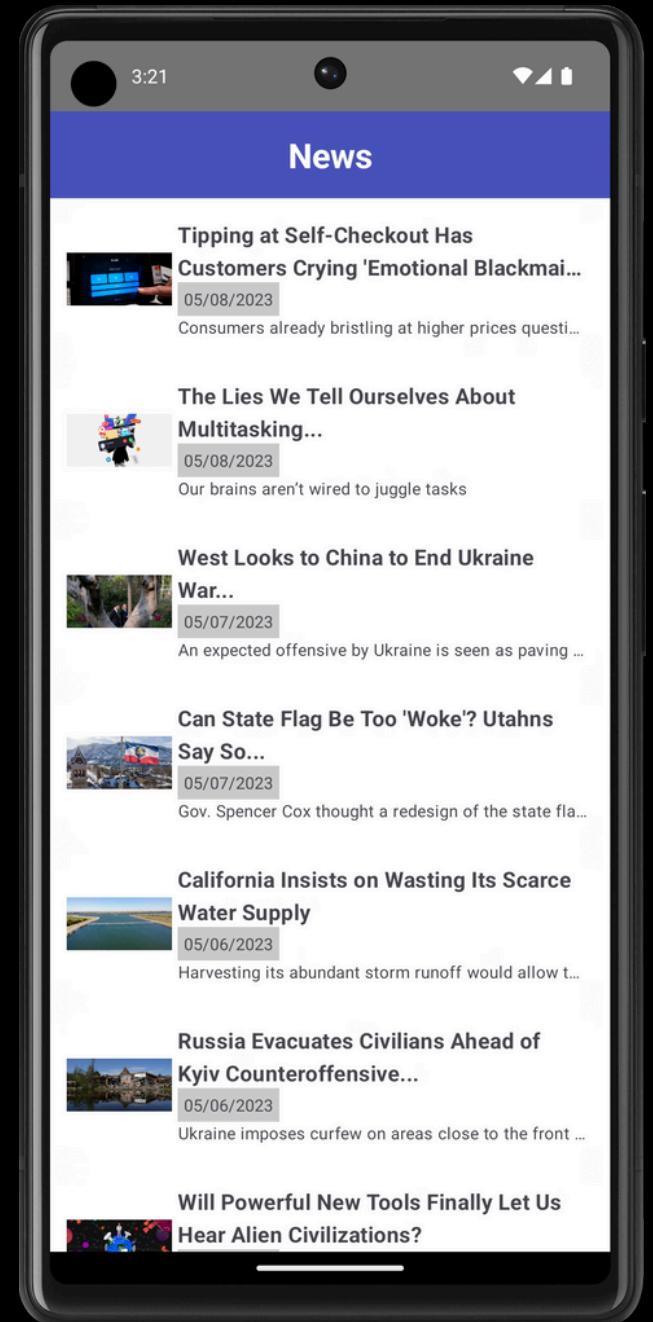
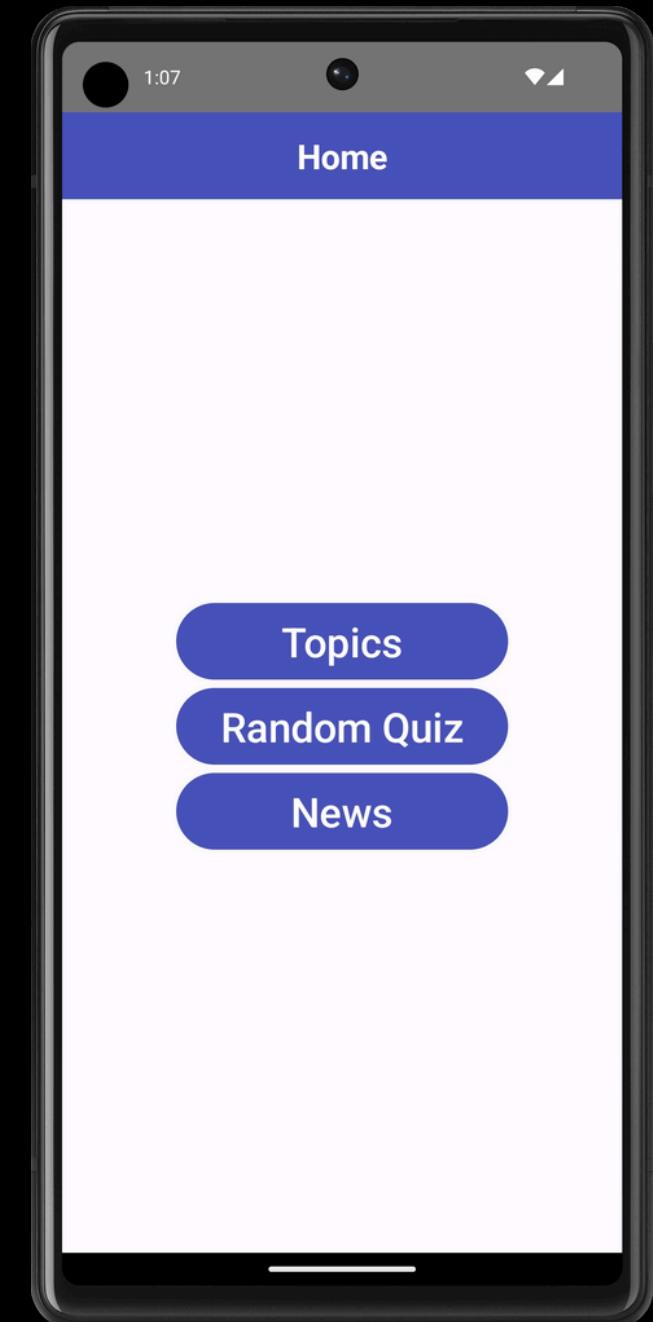
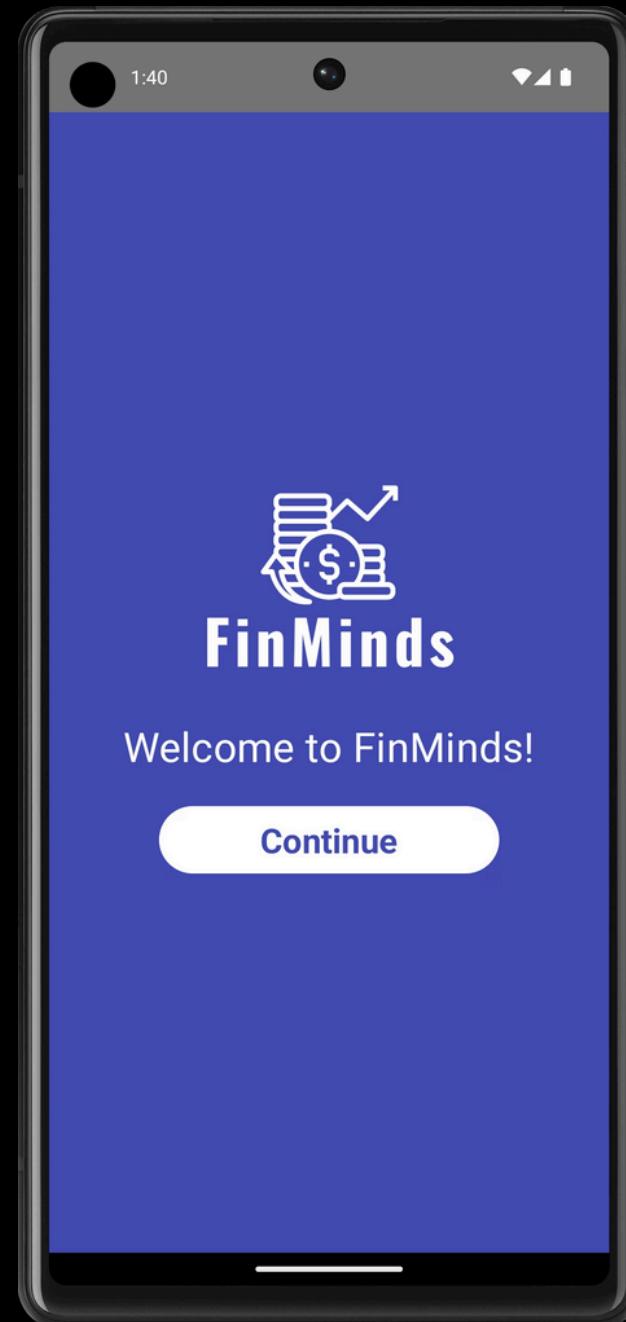
This image shows two detailed wireframes for the "Taxes" section of the application.

Taxes Screen: This screen displays a placeholder image, the text "Information about topic here", and a large play button icon.

Taxes Quiz Screen: This screen lists three questions, each with three choices:

- Question 1:** Choice 1, Choice 2, Choice 3
- Question 2:** Choice 1, Choice 2, Choice 3
- Question 3:** Choice 1, Choice 2, Choice 3

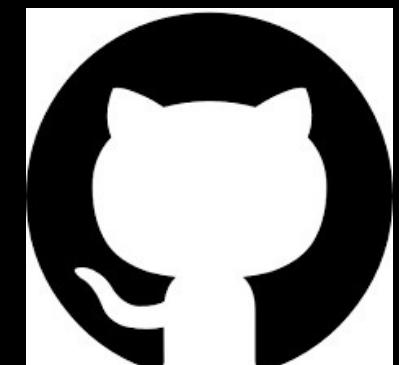
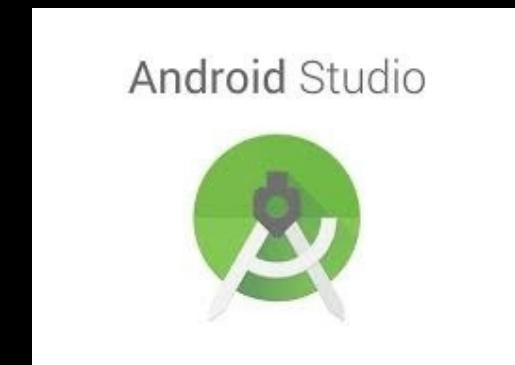
A "miro" watermark is visible in the bottom right corner of the second screen.



Final Design

Technologies and Tools

- The programming language used in our app is **Kotlin**
- We built the UI using **Jetpack Compose**
- The IDE used to build this project was **Android Studio**
- Collaboration and integration application used - **GitHub**
- We used the **NewsAPI** for the news sections
- A **FireBase** database was used to store the random Questions for quiz.



Why Kotlin?

1. Concise syntax
2. Interoperability with Java.
3. Null safety.
4. Functional programming features.
5. Android Studio integration.



Why Jetpack Compose?

- **Declarative UI:** Jetpack Compose uses a state-based approach for UI development.
- **Efficient development:** Jetpack Compose reduces code and promotes component reuse.
- **Real-time previews:** Developers can preview UIs in Android Studio during development.
- **Improved performance:** Jetpack Compose optimizes animations and layout for faster rendering.
- **Simplified testing:** Declarative programming enables easier unit testing of UI components.
- **Backward compatibility:** Jetpack Compose supports older Android versions for wider device compatibility.



Features Overview

Our app consists of three primary features.

1. Topics - These give insights into essential issues such as tax, budgeting, etc.
2. Random Quiz - This feature allows users to test their knowledge about financial literacy.
3. News - This feature helps users be updated with the latest news from *The Wall Street Journal*.

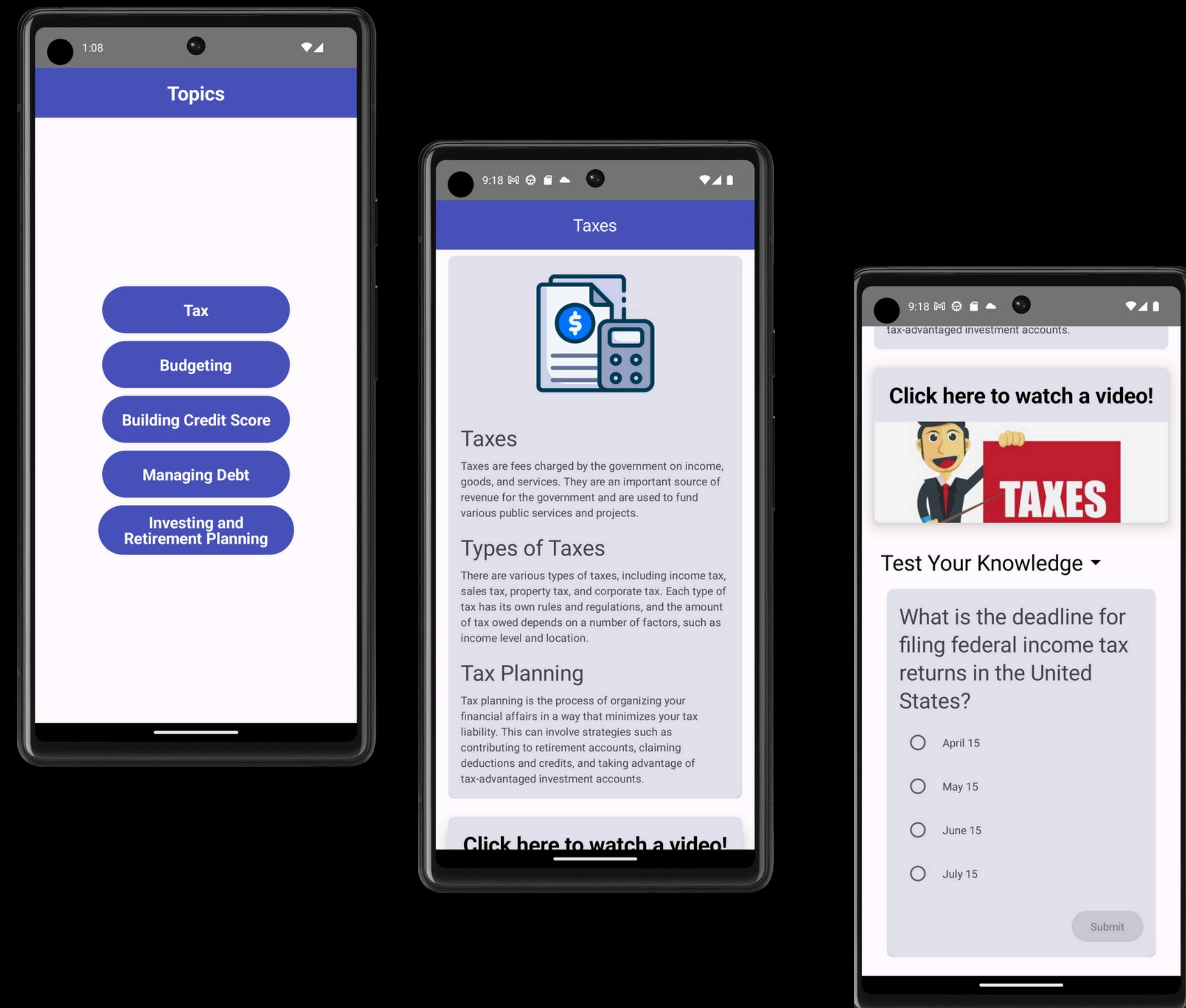


Topics

The topics Feature has a list of topics which could be navigated to their respective activities with intent and onclick functionality.

Each topic has 3 sub features/parts

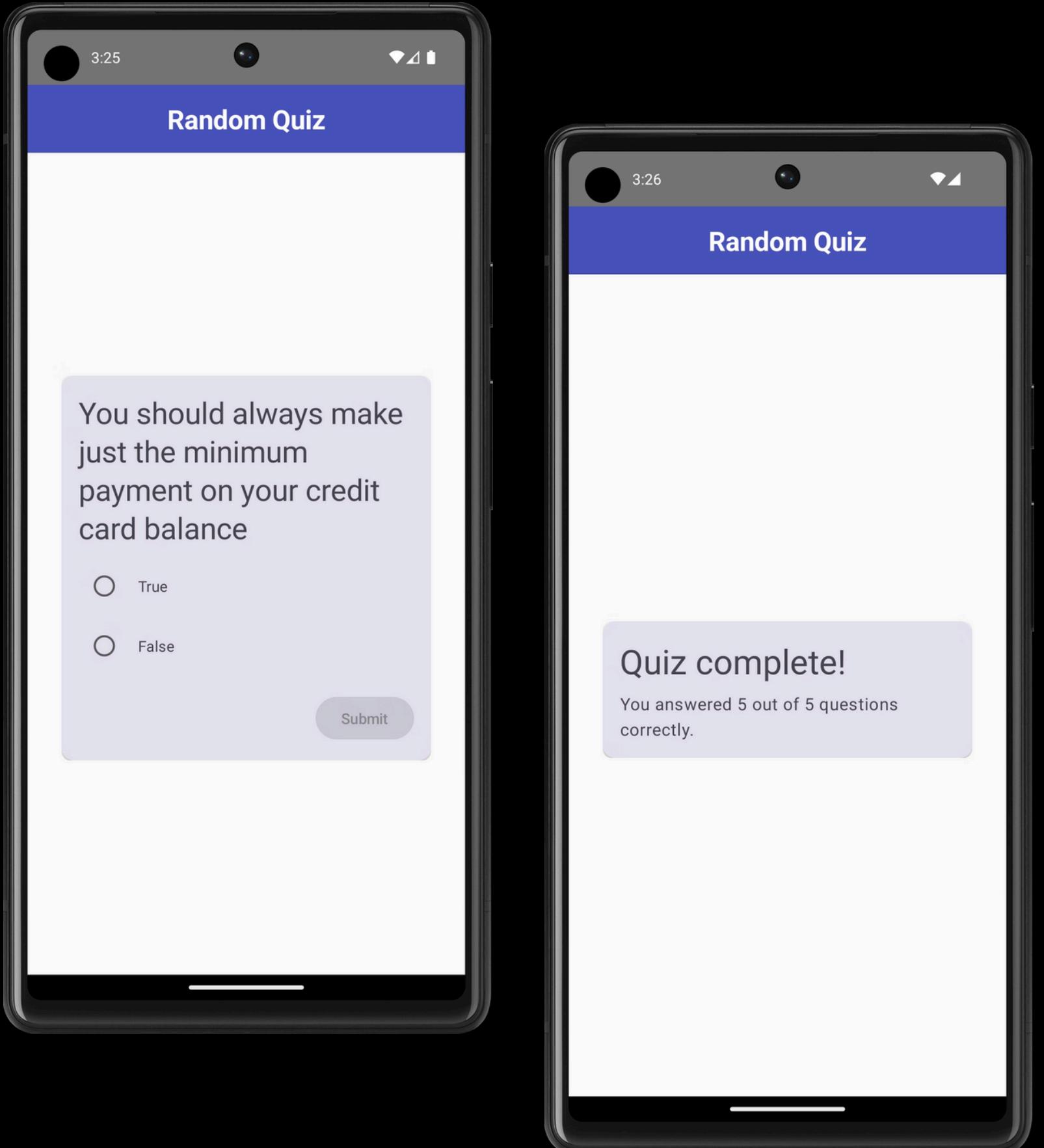
1. First part displays basic information about the selected topic.
2. Second part is an on clickable placeholder card for a YouTube video which redirects the user to give youtube link.
3. The last part is a limited question quiz (3 questions), user is allowed to test his/her knowledge in this feature.



Random Quiz

The random quiz feature has a quiz component that pulls questions from a Firebase real-time database.

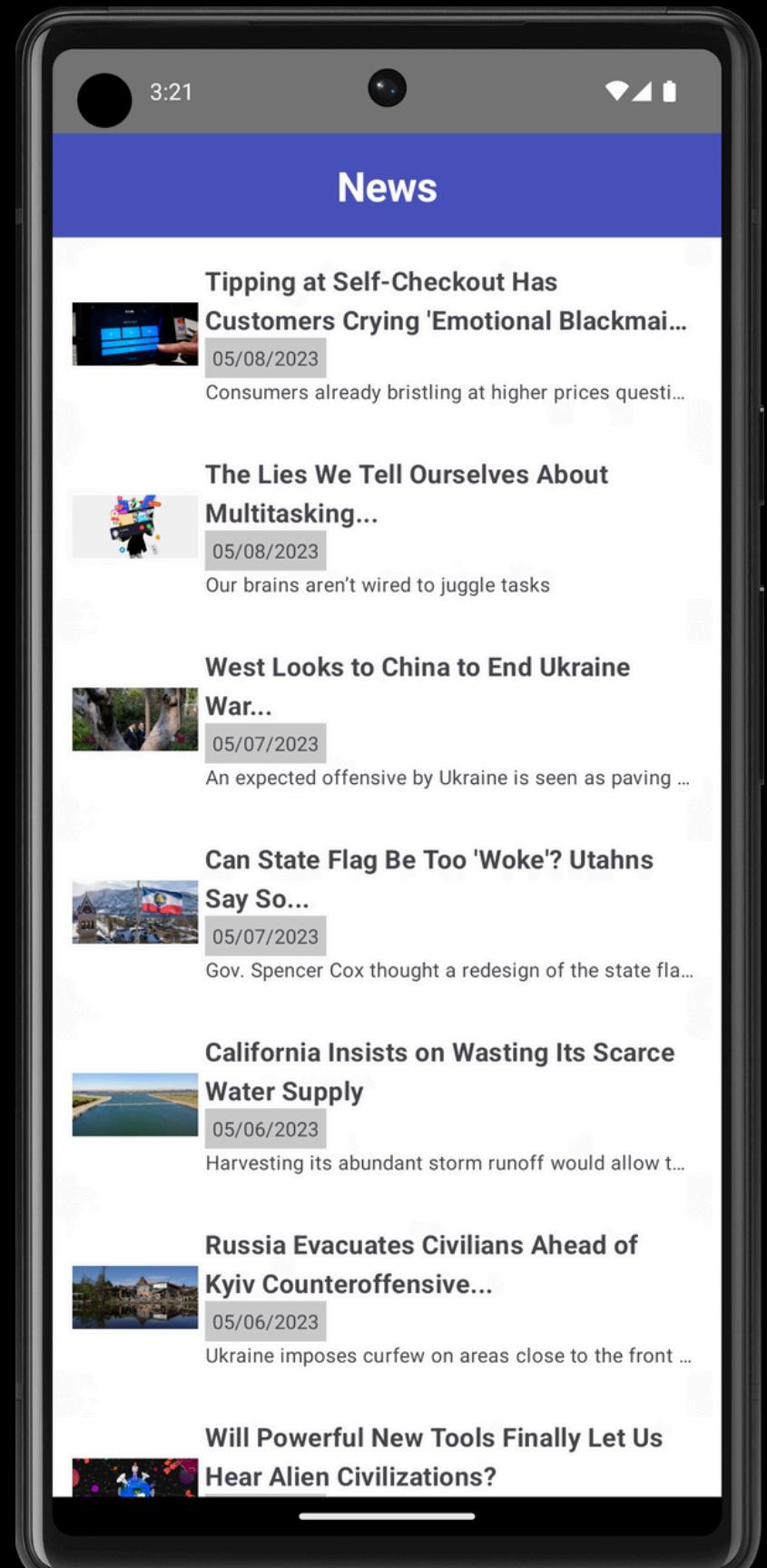
- The purpose of this component is to give the user the ability to jump right into a quiz to test their knowledge.
- The quiz function retrieves the data from the database and puts it in a list. Before the quiz composable is rendered, the list is shuffled to make the questions appear in a different order each time.



News

The News sections displays a list of articles from *The Wall Street Journal*

- The news screen is meant to give a user a quick look at top news stories related to business.
- Each story is clickable and will open the story in the phone's web browser.
- *The Wall Street Journal* was chosen because it generally had up-to-date articles about different business topics



Problems in Development

API Usage

When we were testing the API we didn't realize that we had accidentally made an infinite API call. This in turn made me hit my free account limit, and until we figured it out that we had coded the activity wrong.

Deciding the Scope

For the news section, we wanted to add a source selection to change the listed stories. We did try at first, but realized that we should focus on what we could accomplish in the time frame and keep working on it as a personal project if we wanted.

Future Scope of App

Random Quiz

Right now, the random quiz section has a limited amount of questions. We would have liked to expand the database and even add a custom question feature that allows a user to add a specific question they would like to study.

News

We would have liked to add a source and a search function to the news section. The source function would allow the user to get stories from different news outlets and a search function would help if they were interested in a certain topic.

Topics

The topic part could be redesigned to engage users in form of levels. starting from basic concepts to advanced concepts along with a reward point functionality to keep track of the user progress.

References

-
- 01 Jetpack Compose - <https://developer.android.com/jetpack/compose/documentation>
 - 02 Kotlin Codelabs - <https://developer.android.com/courses/android-basics-kotlin/unit-1>
 - 03 News API - <https://newsapi.org/>
 - 04 Github - <https://github.com/PAJK87/CS639-Final>
-



THANK YOU