

Project Development Phase Project Report Documentation

Date	9 November 2023
Team ID	SI-GuidedProject-587104-1696656485
Project Name	Android Application for Latest Headlines
Team Members	Aarush Jain, Hemantha Kumar Karthikeya, Sanjana Sivakumar, Siddhi Kadam
Maximum Marks	10 Marks

1. INTRODUCTION

1.1 Project Overview:

The Android application aims to provide users with a seamless experience in keeping up with the latest headlines. Leveraging the Jetpack Compose UI toolkit, the app presents a user-friendly interface for viewing news articles. The integration of the Coil library facilitates efficient image loading, enhancing the visual appeal of the articles.

1.2 Purpose:

The primary purpose of the application is to offer users a centralized platform for staying informed about current events through a curated list of news articles. The utilization of the Jetpack Compose UI toolkit showcases modern Android development practices, emphasizing the adaptability and efficiency of this framework. Additionally, the integration of the Retrofit library enables the app to fetch real-time data from a remote server, ensuring that users have access to the most up-to-date information.

By combining these technologies, the application seeks to deliver a user-centric experience that is both visually engaging and technologically advanced. Through this project, we aim to demonstrate the seamless integration of Jetpack Compose, Coil, and Retrofit, showcasing their collective potential in creating a robust and dynamic news consumption platform for Android users.

2. LITERATURE SURVEY

2.1 Existing problem

In today's era of the internet, a multitude of news articles and gossip pieces are easily accessed, by a simple click of a button. But how many of these news articles are written by verified news outlets, and how many sources of the gossip pieces have verifiable sources? Sometimes, the amount of unorganized news floating through the web seems to be too much to handle, and this discourages people from even knowing the events occurring all around the world.

2.2 References

1. **Jetpack Compose Documentation:**

- [Jetpack Compose](<https://developer.android.com/jetpack/compose>) - Official documentation for Jetpack Compose, the UI toolkit used for building the app's user interface.

2. **Retrofit Documentation:**

- [Retrofit](<https://square.github.io/retrofit/>) - Documentation for Retrofit, the library used for making network requests and fetching data from a remote server.

3. **Coil Documentation:**

- [Coil](<https://coil-kt.github.io/coil/>) - Documentation for Coil, the library used for loading images efficiently in the app.

4. **NewsAPI Documentation:**

- [NewsAPI](<https://newsapi.org/docs/get-started>) - Documentation for NewsAPI, which is integrated into the app to fetch the latest news articles.

5. **Android SQLite Documentation:**

- [SQLite on Android](<https://developer.android.com/training/data-storage/sqlite>) - Android's official documentation on SQLite, which is used for local storage in the app.

6. **Android Hilt Documentation:**

- [Hilt](<https://developer.android.com/training/dependency-injection/hilt-android>) - Documentation for Hilt, the dependency injection library used in the app.

7. **Android Activity and Intent Documentation:**

- [Activities](<https://developer.android.com/guide/components/activities/intro-activities>) - Overview of Android activities, which are used for navigation between different screens in the app.

- [Intents](<https://developer.android.com/guide/components/intents-filters>) - Documentation on Android Intents, used for communication between different components of the app.

8. **Android ViewModel Documentation:**

- [ViewModel](<https://developer.android.com/topic/libraries/architecture/viewmodel>) - Documentation for Android's ViewModel, which is used to manage UI-related data.

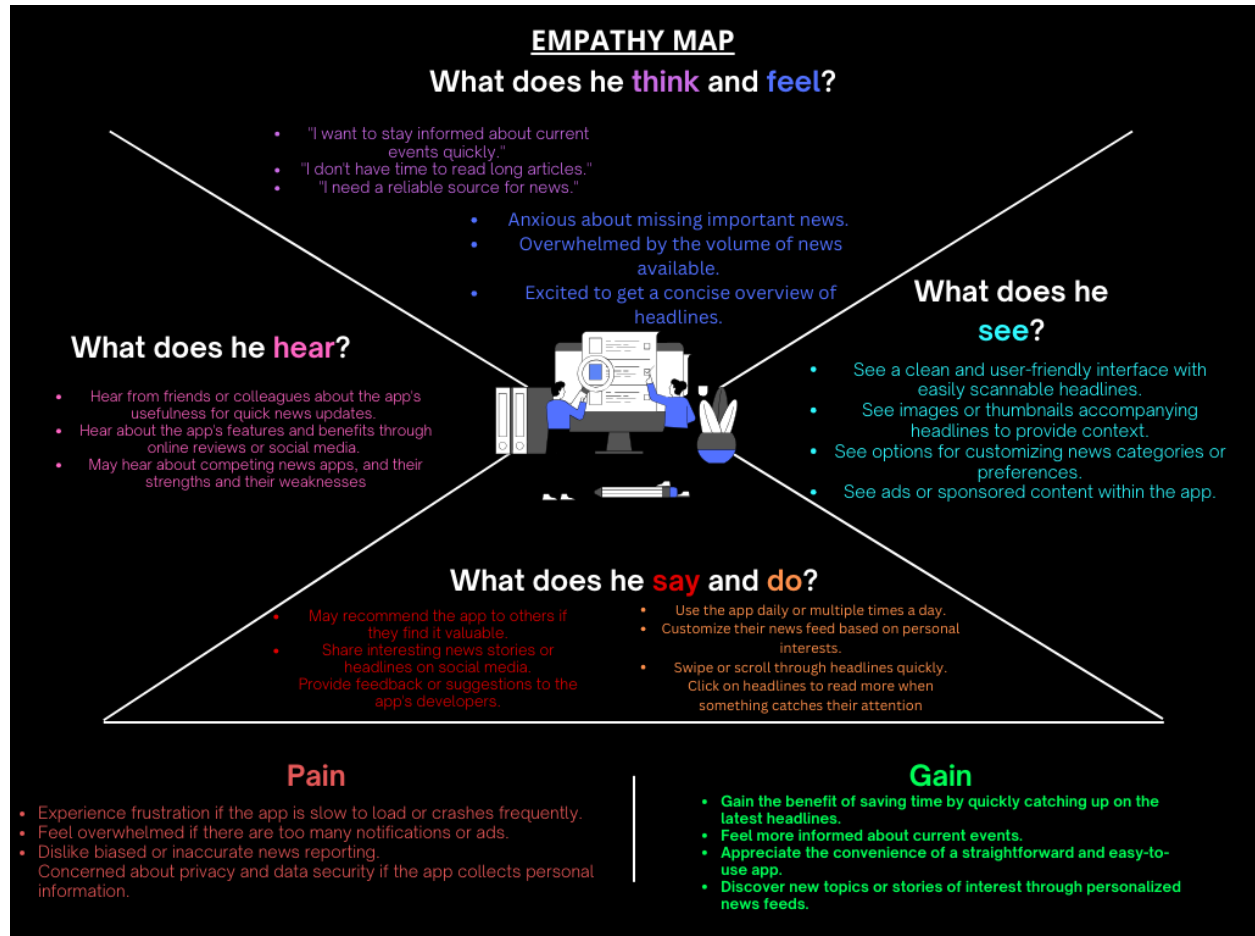
2.3 Problem Statement Definition

An Android application for keeping up with the latest headlines, can be just the solution for this problem, to address information overload, user engagement, data retrieval, image loading, user interaction, UI development, performance and responsiveness.

An abundance of information can lead to information overload, making it difficult for users to filter and access the most relevant and important news stories. Keeping users engaged with the app is crucial. A static list of headlines may not be sufficient to retain user interest. To solve this problem, the app needs to provide an engaging and visually appealing user interface that presents news articles in an easy-to-digest format. The app must fetch the latest news articles from a remote server. Ensuring a smooth and efficient data retrieval process is essential for providing users with up-to-date information. This requires integrating external libraries like Retrofit for API communication. Displaying news articles with images is more engaging and informative. Handling image loading efficiently and asynchronously is vital to prevent lag and ensure a seamless user experience. The app incorporates the Coil library to address this issue. Users should be able to browse through the list of news articles, read article titles, view associated images, and read brief descriptions. Additionally, they should be able to tap on an article to access more detailed information about it. Creating a modern and intuitive user interface is essential for user satisfaction. Jetpack Compose is chosen as the UI toolkit to build the application, and it should be utilized effectively to design a visually appealing and user-friendly interface. The app must be responsive and perform efficiently, even on devices with varying specifications. This involves optimizing code, handling network requests gracefully, and minimizing any potential UI lag.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

Step 2: Brainstorm, Idea List, Grouping



Step-3: Idea Prioritization



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

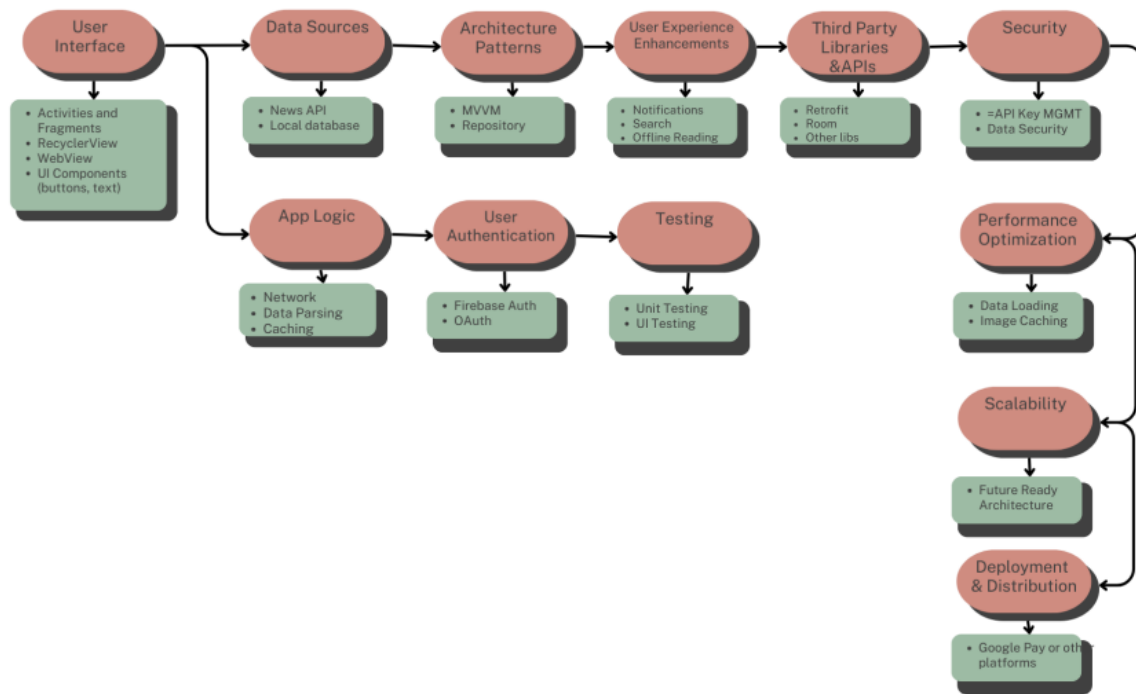
- Search Bar
- Filtering
- News outlet buttons
- Saved user data in a local database

4.2 Non-Functional requirements

- Dark Mode
- Light Mode

5. PROJECT DESIGN

5.1 Data Flow Diagrams & User Stories



User Type	Functional Requirement (Epic)	User Story Number	User Story/Task	Acceptance Criteria	Priority	Release
Customer	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password	I can access my account/ dashboard	HIGH	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email and click confirm	HIGH	Sprint-1
	Login	USN-3	As a user, I can log into the application by entering my username and password	I can enter a valid username and password to log in, after which I will be redirected to the dashboard.	HIGH	Sprint-1

	Dashboard	USN-4	As a user, I want to read news articles on various topics	The dashboard displays a list of up-to-date news articles from various topics or categories for	HIGH	Sprint-1
--	-----------	-------	---	---	------	----------

				easy navigation.		
		USN-5	As a user, I want to search for specific news articles using keywords	I can enter search keywords in the search bar and relevant news articles are displayed.	HIGH	Sprint-2
		USN-6	As a user, I want to share news articles on social media	I can select a news article and also choose a social media platform to share it.	MEDIUM	Sprint-3
		USN-7	As a user, I want to receive notifications about breaking news	When enabled, I can receive timely notifications about breaking news and be up to date.	MEDIUM	Sprint-2

		USN-8	As a user, I want to add chosen news articles into a label "Favourites"	I can mark and unmark an article as my favourite and store them in the "Favorites" section.	HIGH	Sprint-2
		USN-9	As a user, I want	The "Favorites"	HIGH	Sprint-2

			to access "Favourites" news articles	section displays all news articles marked as favorites and can be clicked to read its full content.		
--	--	--	--------------------------------------	---	--	--

5.2 Solution Architecture

1. User Interface (UI):

- Activities and Fragments: Android app UI components for displaying news headlines, article details, and user interactions.
- RecyclerView: To display a list of news headlines efficiently.
- WebView: For displaying full news articles when a headline is selected.
- User Interface Components: Buttons, navigation elements, and other UI elements for user interaction.

2. Data Sources:

- News API: To fetch news headlines and articles from various news sources. You may consider using a reputable news API that provides structured data.
- Local Database: To store and cache news data, providing offline access and reducing the frequency of API requests.

3. App Logic:

- Network Communication: A component to manage HTTP requests to the news API, handling API endpoints and data retrieval.
- Data Parsing: Parsing and formatting news data retrieved from the API.
- Caching: Managing the local database for storing and retrieving news data.

4. Architecture Patterns:

- Model-View-ViewModel (MVVM) Architecture: This architectural pattern separates the UI from the business logic and data access. It involves ViewModel classes that expose data to the UI and handle user interactions.
- Repository Pattern: To abstract data sources, making it easy to switch between local and remote data sources.

5. User Authentication:

- If the app offers features like saving favorite articles or customizing news sources, you may need user authentication components.
- Firebase Authentication or OAuth could be used for user authentication.

6. User Experience Enhancements:

- Notifications: To notify users of new articles or updates.
- Search Functionality: If you want to allow users to search for specific news topics.
- Offline Reading: Downloading articles for offline reading.

7. Testing:

- Unit Testing: Writing unit tests for app logic, especially data retrieval and parsing components.
- UI Testing: Creating UI tests to ensure the app's UI components work correctly.

8. Third-Party Libraries and APIs:

- Consider using libraries like Retrofit for network requests, Room for local database management, and libraries for image loading, like Picasso or Glide.
- Ensure compliance with the terms of use of the news API and any other third-party services you utilize.

9. Security:

- Ensure that sensitive data like API keys are stored securely, and API requests are made over HTTPS.
- Implement appropriate security measures to protect user data and privacy.

10. Performance Optimization:

- Implement strategies for efficient data loading, image caching, and smooth UI interactions.

11. Scalability:

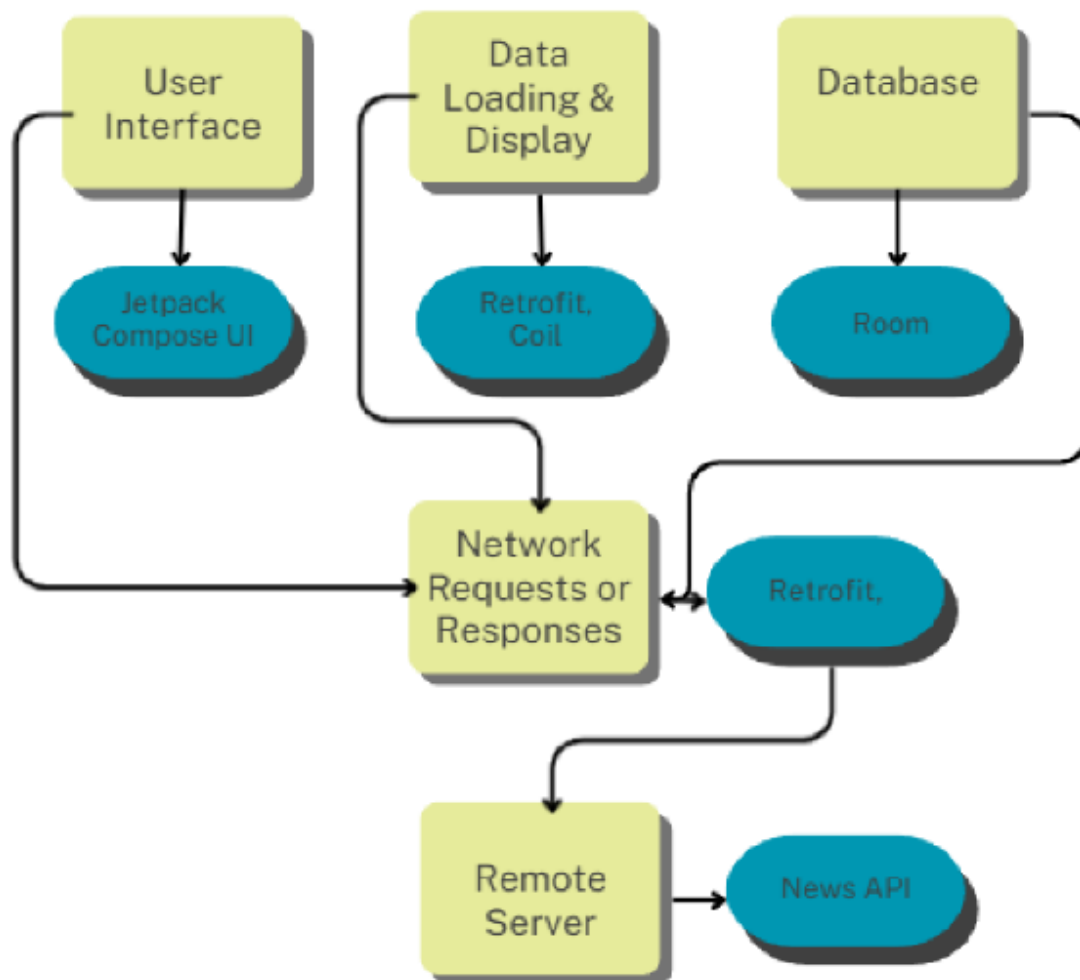
- Ensure that the architecture allows for easy scaling if you plan to add more features or support a larger user base in the future.

12. Deployment and Distribution:

- Consider publishing the app on Google Play Store or other distribution platforms

6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture



Component	Description	Technology
User Interface (UI)	The front-end part of the application, responsible for displaying news articles and handling user interactions.	Jetpack Compose, XML (for layouts)
Data Fetching Service	Responsible for fetching news articles from external sources.	Retrofit (HTTP client), RESTful APIs
Image Handling Service	To load and cache images associated with news articles.	Coil (Image loading library)
Local Data Storage	Cache or store data locally on the device for offline access.	Room (Local database)
User Authentication	User login and authentication for personalized experiences.	Firebase Authentication (if required)
External APIs	Third-party APIs for fetching news articles and related data.	News API, OpenWeatherMap API, or other news sources APIs
Analytics	Collecting user behavior data for analysis and improvements.	Firebase Analytics, Google Analytics (optional)

Characteristic	Description	Technology
User-Friendly UI	An intuitive and visually appealing user interface for news browsing.	Jetpack Compose, XML (for layouts)
Real-Time Updates	Frequent updates to provide users with the latest news headlines.	Retrofit (for real-time data fetching), Push Notifications
Offline Access	The ability to read previously fetched articles when offline.	Local database (Room), caching mechanisms
Personalization	Customization based on user preferences and behavior.	User profiles, machine learning (if applicable)
Security	Ensuring user data and interactions are secure.	SSL/TLS, Firebase Authentication (if applicable)
Scalability	The application's ability to handle increased user load.	Scalable server infrastructure, CDN (for images), Load balancing
Cross-Platform Support	The ability to expand to other platforms (iOS, Web, etc.).	Kotlin Multiplatform Mobile (if applicable)
Analytics and Insights	Gathering user behavior data for app improvement and advertising.	Firebase Analytics, Google Analytics (optional)
Monetization	Strategies to generate revenue from the app.	Freemium model, Advertisements, Affiliate Marketing, Data Analytics, etc.

6.2 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	3	High	Aarush Jain
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application.	2	High	Aarush Jain
Sprint-1	Login	USN-3	As a user, I can log into the application by entering my username and password.	2	High	Sanjana SivaKumar
Sprint-1	Dashboard	USN-4	As a user, I want to read news articles on various topics.	3	High	Siddhi Kadam
Sprint-2		USN-5	As a user, I want to search for specific news articles using keywords.	1	High	Hemantha Kumar Karthikeya
Sprint-3		USN-6	As a user, I want to share news articles on social media	3	Medium	Aarush Jain
Sprint-2		USN-7	As a user, I want to receive notifications about breaking news.	3	Medium	Siddhi Kadam
Sprint-2		USN-8	As a user, I want to add chosen news articles into a label "Favourites".	2	High	Hemantha Kumar Karthikeya
Sprint-2		USN-9	As a user, I want to access "Favourites" news articles.	2	High	Sanjana SivaKumar

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	10	7 Days	24 Oct 2023	31 Oct 2023	10	02 Nov 2023
Sprint-2	8	6 Days	01 Nov 2023	06 Nov 2023		
Sprint-3	3	3 Days	07 Nov 2023	10 Nov 2023		

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{20}{10} = 2$$

6.3 Sprint Delivery Schedule

- Sprint 1
 - Total Story Points: 10
 - Duration: 7 days
 - Velocity=10/7=1.43 (approximately)
- Sprint 2
 - Total Story Points: 8
 - Duration: 6 days
 - Velocity = 8/6 =1.33 (approximately)
- Sprint 3
 - Total Story Points: 3
 - Duration: 3 days
 - Velocity = 3/3 =1

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

SearchAdapter (Adapter for RecyclerView):

The SearchAdapter class is responsible for handling the items displayed in the search results RecyclerView. It dynamically inflates different view types (Source or Article) based on the viewType and binds the data accordingly.

Code:

<https://github.com/smartinternz02/SI-GuidedProject-587104-1696656485/tree/main/MyNewsApp/app/src/main/java/com/aarush/newsapp/ui/articles>

SearchActivity (Activity that handles search functionality):

The SearchActivity class manages the UI components and user interactions related to the search feature. It utilizes the SearchAdapter for displaying search results in a RecyclerView.

Code: <https://github.com/smartinternz02/SI-GuidedProject-587104-1696656485/tree/main/MyNewsApp/app/src/main/java/com/aarush/newsapp/ui/articles>

CustomSearchBar (Custom Search Bar Widget):

The CustomSearchBar class is a custom widget that extends LinearLayout to encapsulate the search functionality. It includes a SearchView and an additional button (ImageView). It provides methods to customize the search bar, set listeners, and handle user interactions.

Code: <https://github.com/smartinternz02/SI-GuidedProject-587104-1696656485/tree/main/MyNewsApp/app/src/main/java/com/aarush/newsapp/widget>

Signup Feature (signupForm Activity):

The signupForm activity handles user registration, storing the user's name, username, and password in a SQLite database. It also includes a password visibility toggle for the user to show or hide the entered password.

Code: <https://github.com/smartinternz02/SI-GuidedProject-587104-1696656485/blob/main/MyNewsApp/app/src/main/java/com/aarush/newsapp/signupForm.kt>

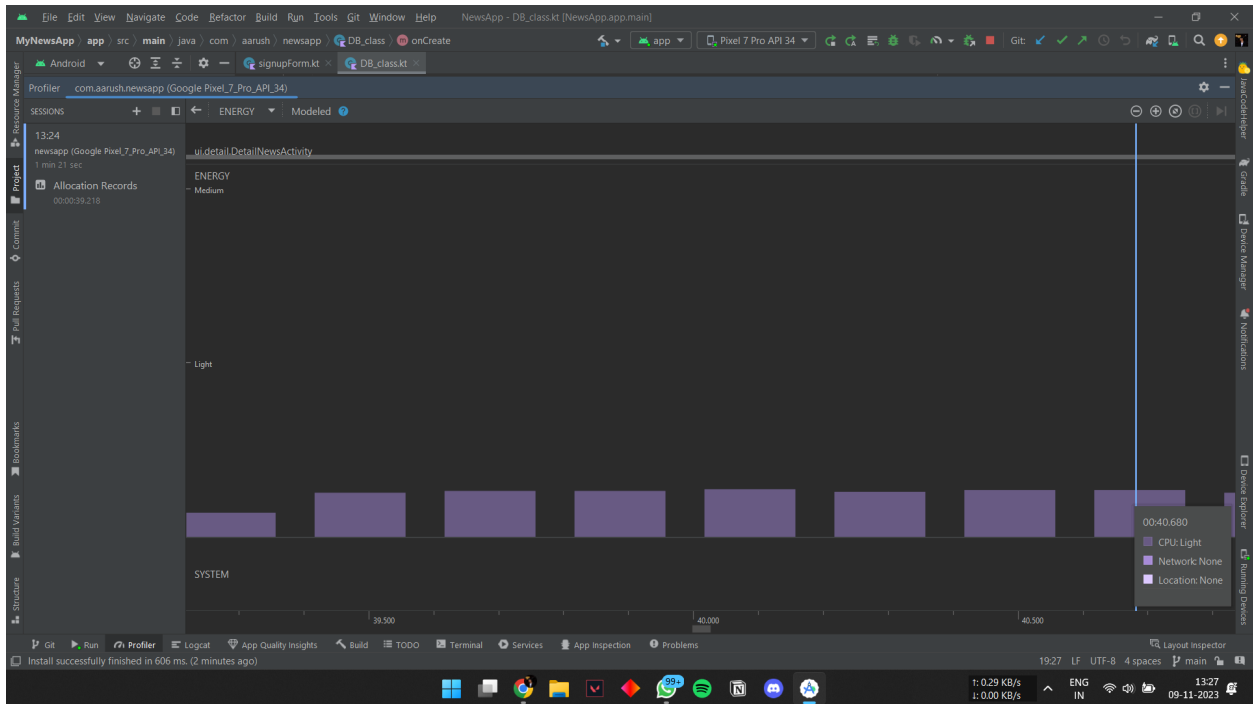
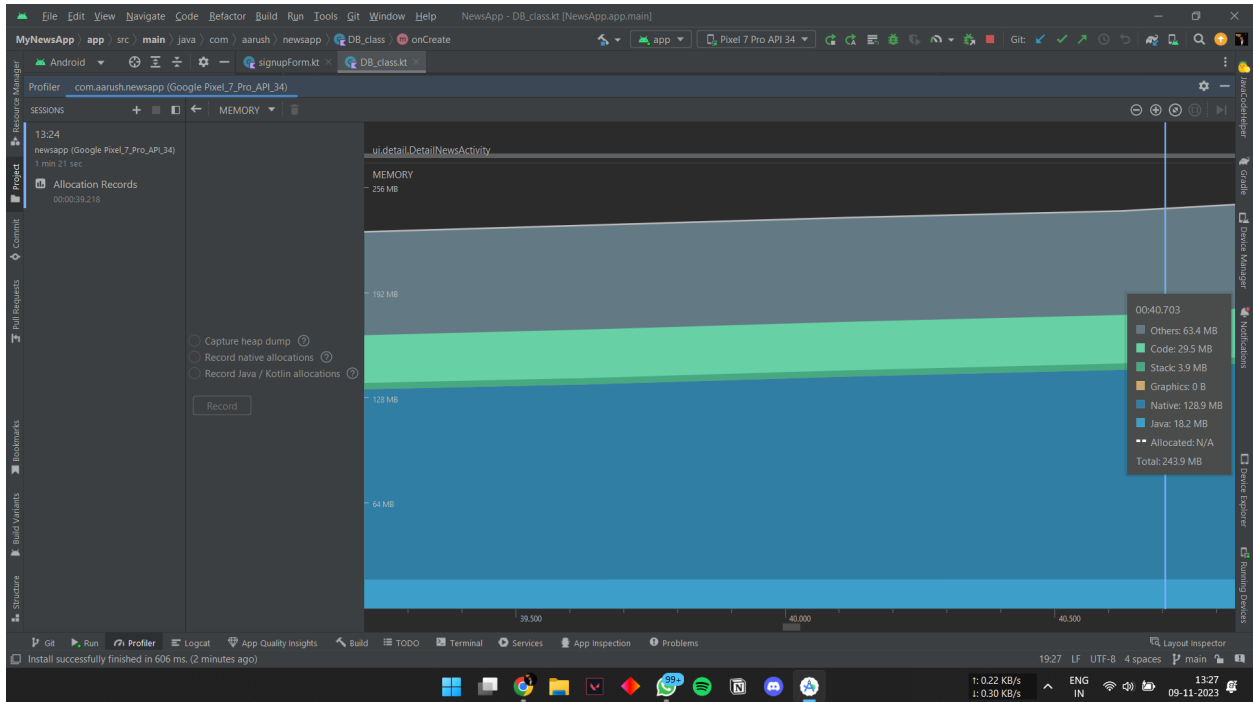
Login Feature (login_form Activity):

The login_form activity handles user login, checking the entered username and password against the data stored in the SQLite database. It also includes a password visibility toggle and a link to navigate to the signupForm activity.

Code: https://github.com/smartinternz02/SI-GuidedProject-587104-1696656485/blob/main/MyNewsApp/app/src/main/java/com/aarush/newsapp/login_form.kt

8. PERFORMANCE TESTING

8.1 Performance Metrics

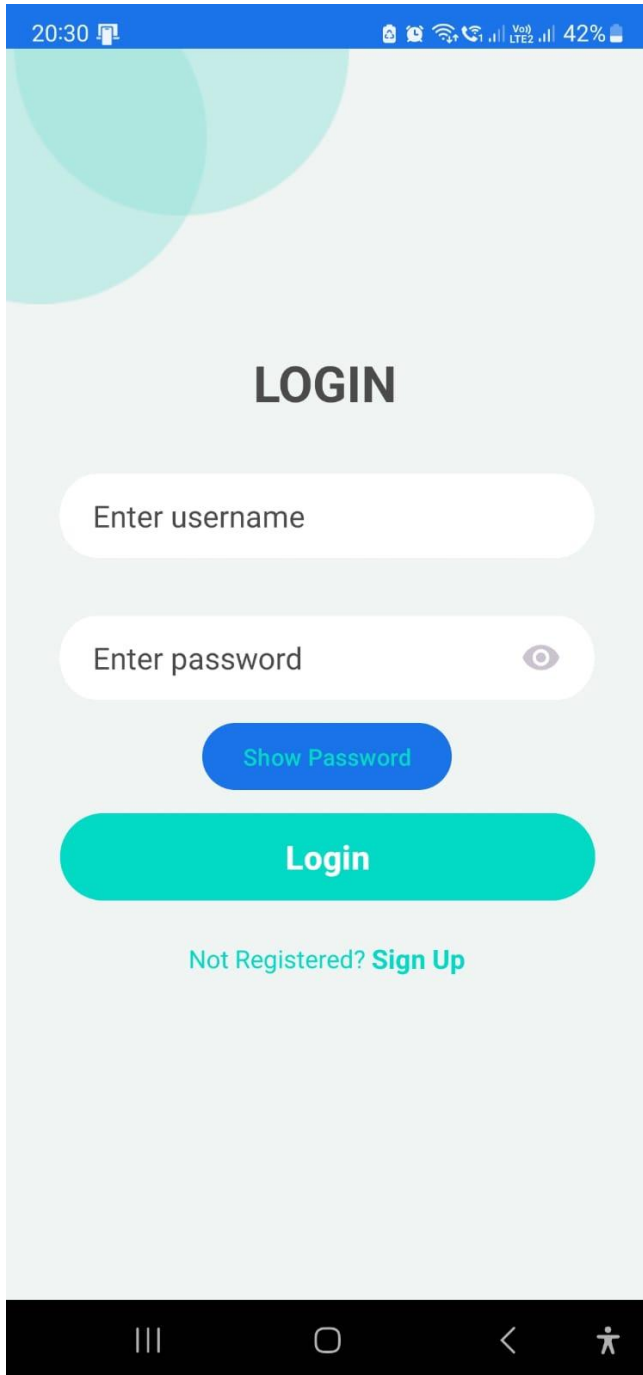


9. RESULTS

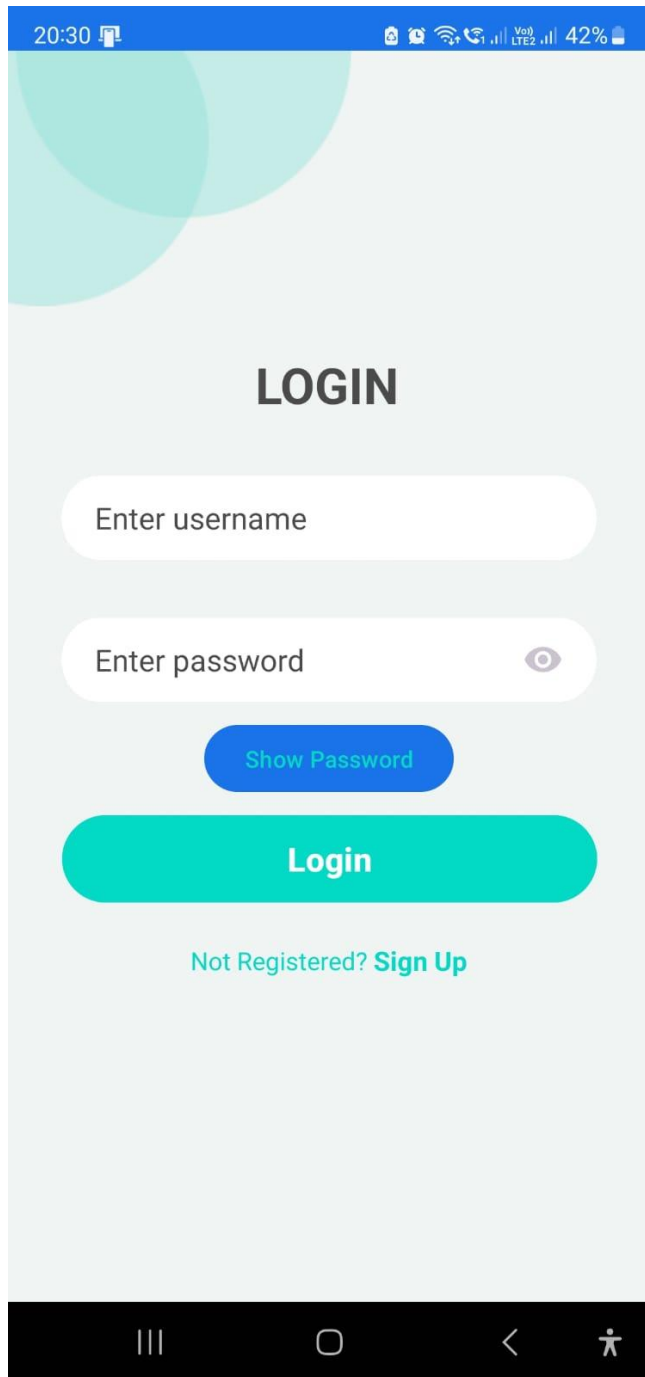
9.1 Output Screenshots

Screen 1: Login Screen

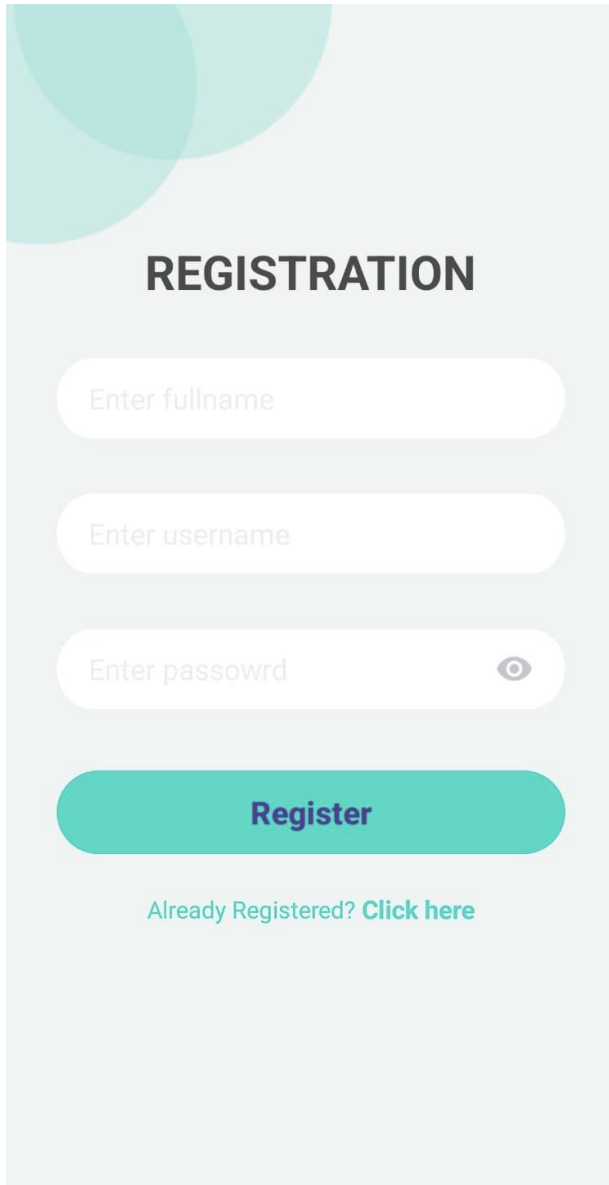
LIGHT MODE:




DARK MODE:



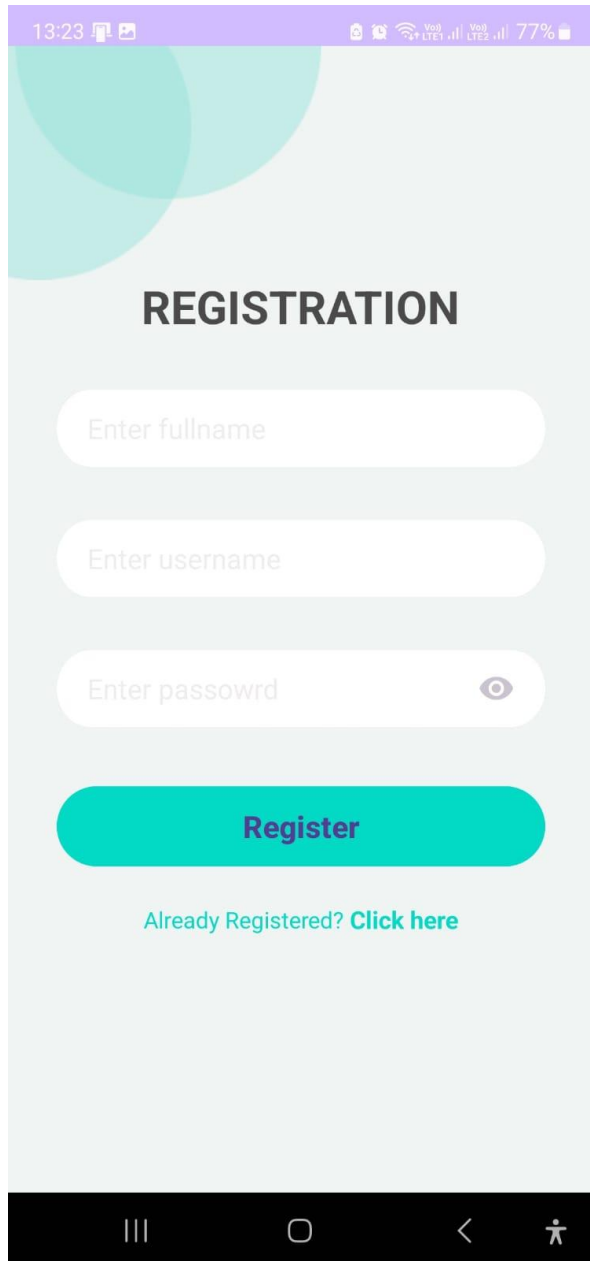
Screen 2:
LIGHT MODE:

The image shows a registration form on a light gray background. In the top-left corner, there are two overlapping teal circles. The form consists of three white rounded rectangular input fields stacked vertically. The first field is labeled 'Enter fullname', the second 'Enter username', and the third 'Enter passowrd' (note the typo). To the right of the password field is a small teal eye icon. Below the input fields is a teal rounded rectangular button with the text 'Register' in bold black font. At the bottom of the form, there is a link that says 'Already Registered? Click here' in teal text.

REGISTRATION

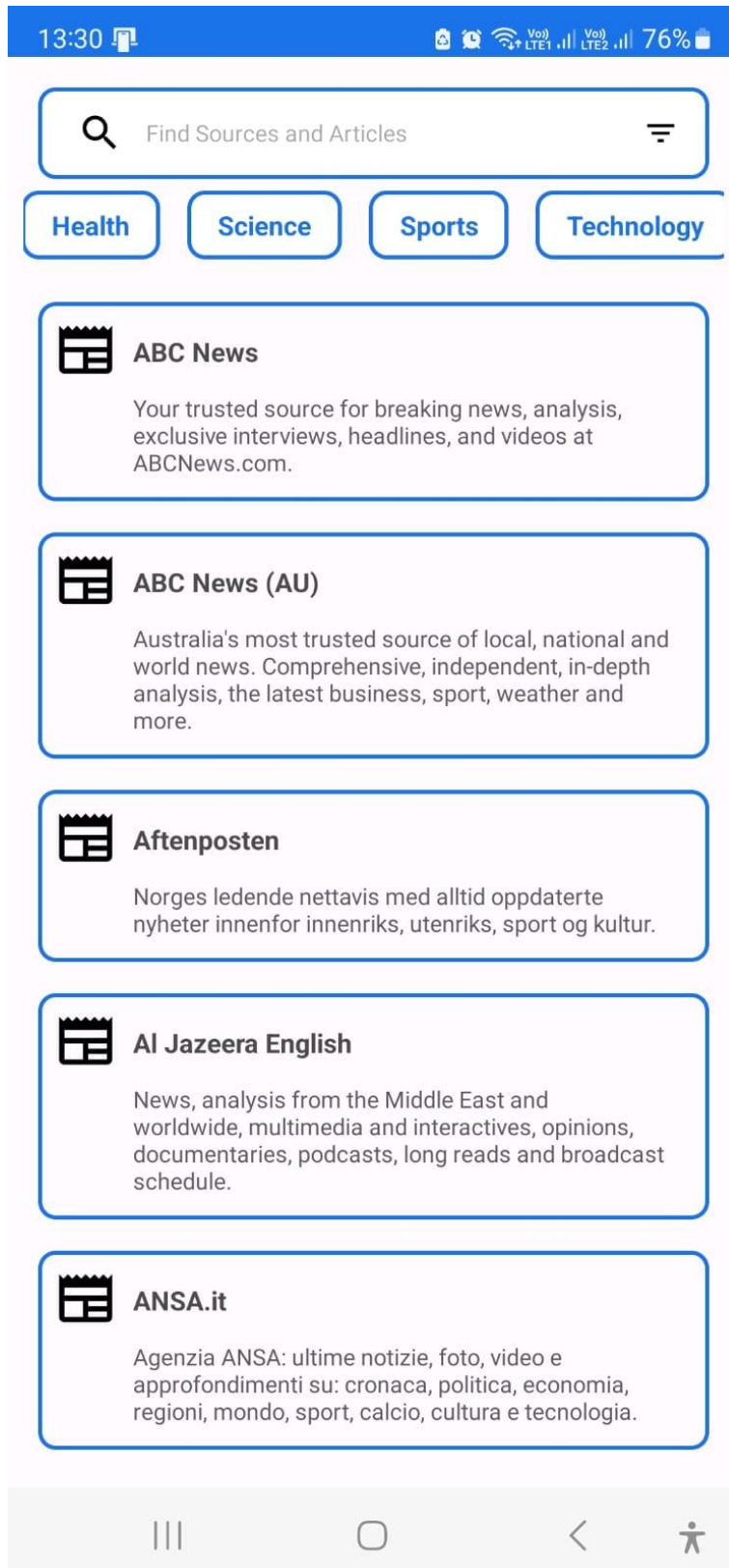
 
Register
[Already Registered? Click here](#)

DARK MODE:

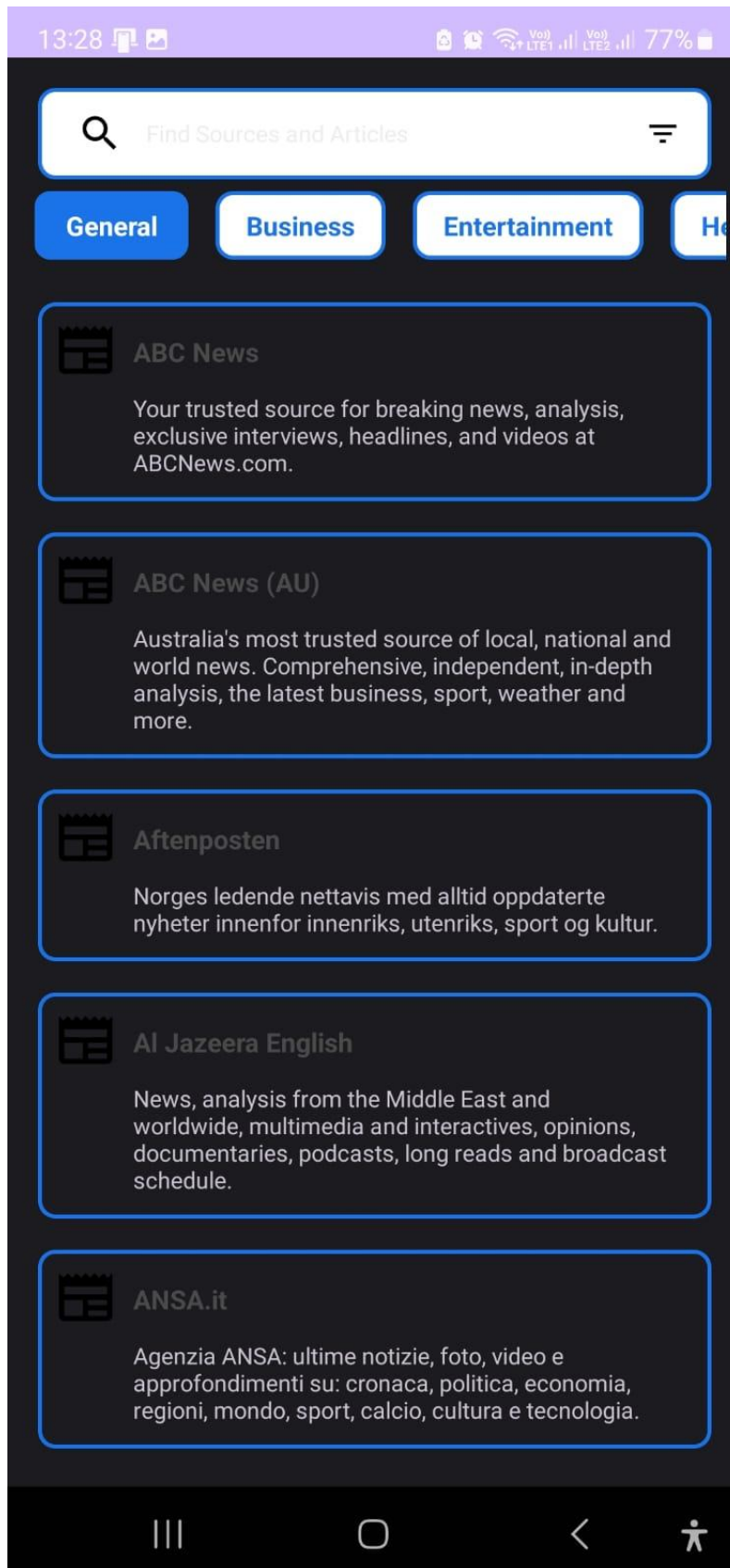


Screen 3:

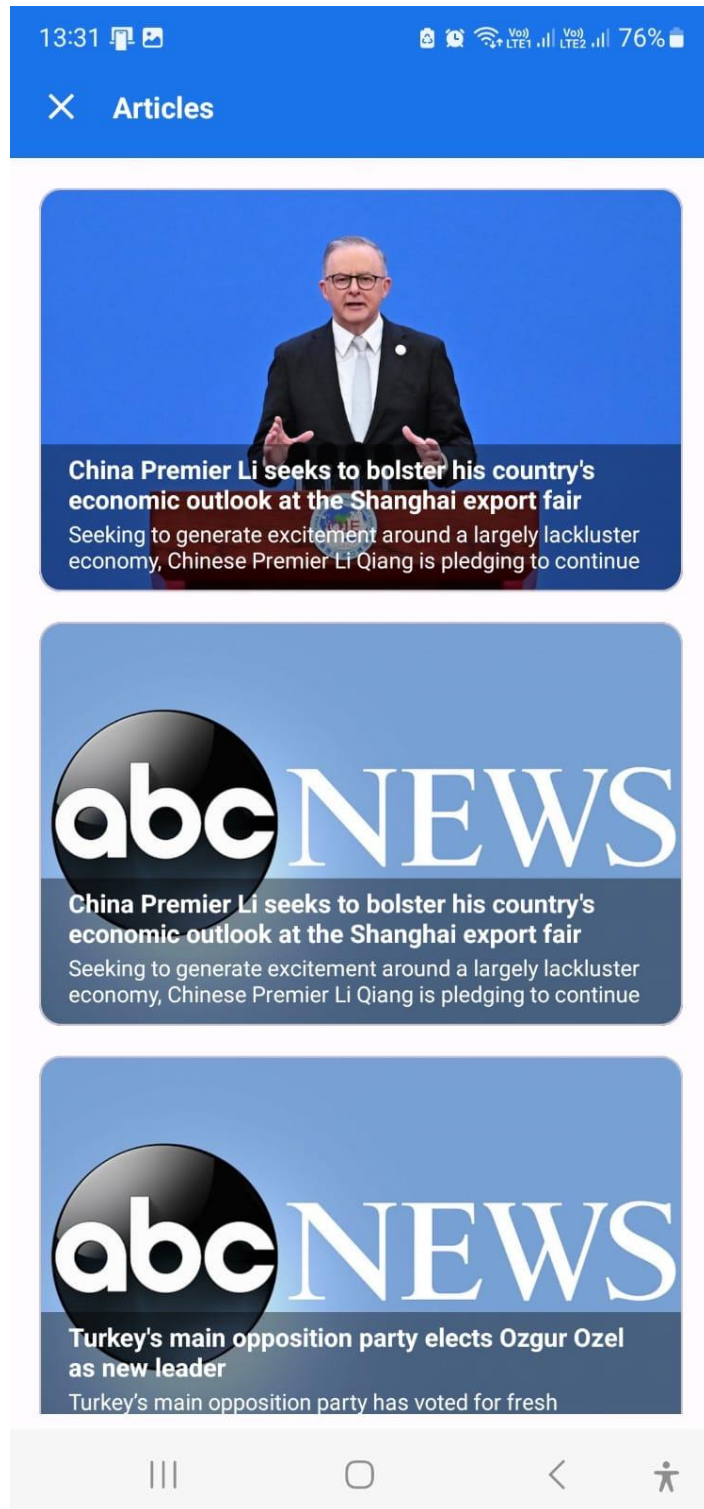
LIGHT MODE:



DARK MODE:



Screen 4:
LIGHT MODE:



DARK MODE:



Screen 5:
LIGHT MODE:



DARK MODE:

13:29

VoIP LTE1 VoIP LTE2 77%

× ABC News

≡ abc NEWS



China Premier Li seeks to bolster his country's economic outlook at Shanghai trade fair

Seeking to generate excitement around a largely lackluster economy, Chinese Premier Li Qiang is pledging to continue deepening reforms, expand free trade zones and relax market access for foreign investment

By The Associated Press
November 5, 2023, 1:13 PM



10. ADVANTAGES & DISADVANTAGES

Advantages:

User-Friendly Interface: The use of Jetpack Compose UI toolkit ensures a modern and interactive user interface.

Efficient Image Loading: Integration of the Coil library for image loading enhances the app's performance and user experience.

Data Fetching and Networking: Retrofit library simplifies data fetching from a remote server, making the app responsive and data-efficient.

Authentication and User Management: The inclusion of login and registration screens with SQLite ensures secure user authentication and management.

News Categorization: Categorizing news into subtabs makes it easy for users to find content relevant to their interests.

Search and Sorting: Inclusion of search and sort features adds flexibility and convenience for users to navigate through the news articles.

Disadvantages:

Dependency Management: Managing dependencies like Jetpack Compose, Retrofit, and Coil may lead to compatibility issues in the future.

Limited Offline Functionality: The app may have limited functionality without an internet connection since it relies heavily on remote data fetching.

In-App Browser Limitations: The in-app browser may not provide the same experience as external browsers, and some users may prefer external browsers for reading articles.

11. CONCLUSION

In conclusion, the Android Application for Keeping Up With the Latest Headlines presents a user-friendly and visually appealing solution for news enthusiasts. By leveraging XML files, the app's UI is crafted, ensuring an intuitive and engaging experience. The incorporation of the Coil library facilitates seamless loading and caching of images, enhancing the overall performance. The integration of SQLite as the database solution ensures efficient data storage and retrieval, enabling users to scroll through a vast array of news articles with ease. The addition of light and dark mode options enhances user experience, allowing customization based on preferences and lighting conditions. The search bar functionality adds a layer of convenience, enabling users to quickly find specific news outlets or articles of interest. This feature enhances the app's usability, making it easy for users to access information tailored to their preferences. This project successfully combines user interface design, efficient data handling, and search functionality to create a comprehensive news application. By providing a seamless experience and catering to user preferences through light and dark modes, the application is positioned to cater to a wide audience of news readers. The project's successful implementation of these features ensures a satisfying user experience, making it a valuable addition to the realm of Android news applications.

12. FUTURE SCOPE

Audio News and Podcast Integration: Explore the addition of audio news or podcast features to cater to users who prefer listening to news updates on the go.

Augmented Reality (AR) Features: Experiment with AR to provide interactive and immersive news experiences, such as AR-enhanced articles or visualizations.

Machine Learning for Content Recommendation: Implement machine learning algorithms to analyze user behavior and recommend personalized news content based on individual preferences.

Real-time News Updates: Integrate real-time updates for breaking news, ensuring users get the latest information as it happens.

User-Generated Content: Allow users to contribute content, such as submitting local news or opinion pieces, fostering community engagement.

Integration with Wearable Devices: Explore compatibility with wearable devices to deliver news updates directly to smartwatches or other wearable gadgets.

Blockchain for Authenticity: Consider leveraging blockchain technology to ensure the authenticity and traceability of news sources, combating misinformation.

Interactive Visualizations: Enhance the visualization of news stories with interactive charts, graphs, and infographics to make complex information more accessible.

Collaboration with Influencers: Collaborate with influencers or thought leaders in various domains to provide unique insights and perspectives on news stories.

In-Depth Analysis and Reports: Provide in-depth analysis and reports on trending topics or significant events, giving users a more comprehensive understanding of the news.

Gamification Elements: Introduce gamification elements to encourage user engagement, such as quizzes, challenges, or rewards for staying updated on news.

Integration with Smart Assistants: Enable integration with voice-activated smart assistants like Google Assistant or Amazon Alexa for a hands-free news browsing experience.

Virtual Reality (VR) News Experiences: Explore VR capabilities to create virtual environments for users to immerse themselves in news stories, especially for special features or events.

Global News Aggregator: Extend the app's reach by aggregating news from various international sources, providing a broader perspective on global events.

Community Forums: Introduce community forums or discussion boards within the app, allowing users to share opinions, discuss news, and engage in meaningful conversations.

13. APPENDIX

Source Code:

<https://github.com/smartinternz02/SI-GuidedProject-587104-1696656485/tree/main/MyNewsApp>

GitHub & Project Demo Link

GitHub link: <https://github.com/smartinternz02/SI-GuidedProject-587104-1696656485>

Project Demo Link: