Created by: Bryan Loo Chun Wai

Mobile Development Final Project Report

Table of Contents

Section 1 – Concept Development

1.1) Project Idea and Motivation

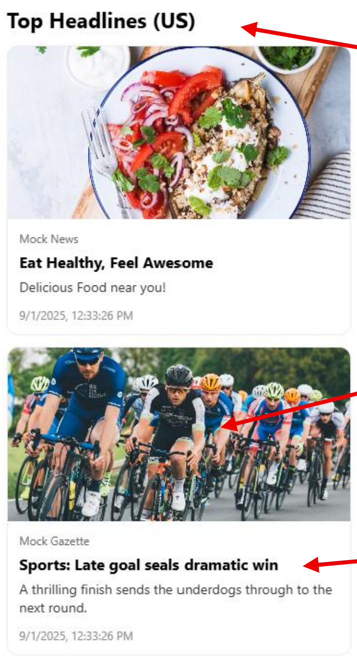
The idea for NewsDeck arose from the observation that most news applications only focus on providing an endless stream of articles, often leaving users overwhelmed rather than informed. With the constant influx of online media, there is a strong need for a tool that not only delivers articles but also helps users understand patterns and trends in what they consume.

The motivation was therefore twofold:

Educational motivation – to practice developing a cross-platform React Native application with modular design, applying advanced programming concepts such as contexts, hooks, and services.

Practical motivation – to create an app that allows users to browse the news in an intuitive interface, while also offering analytics-driven insights (e.g., article volumes, trending topics, and category distribution).

By combining a traditional feed with a lightweight analytics dashboard, NewsDeck aims to bridge the gap between raw information consumption and meaningful interpretation.


Top Headlines (US)

Mock News
Eat Healthy, Feel Awesome
Delicious Food near you!
9/1/2025, 12:33:26 PM

Mock Gazette
Sports: Late goal seals dramatic win
A thrilling finish sends the underdogs through to the next round.
9/1/2025, 12:33:26 PM

Aim was to create a articles page similar to this sketch.

Images sparks users' interest in the article.

Only the headline and a short description of the article is displayed.

1.2) Objectives and Goals

The main objectives of NewsDeck can be summarized as:

To design a modular and maintainable news application with clear separation of concerns.

To implement essential user functions: register/login, browse articles, save/remove favorites, and update profile settings.

To provide an Analytics (Stats) screen that visually represents trends, categories, and topics in a report-friendly format.

To demonstrate best practices in React Native development, such as context providers for global state and error handling for API calls.

To support theming (light/dark modes), enhancing accessibility and user control over the app's interface.

1.3) Innovation and Creativity

Whereas many student projects limit themselves to static article fetching or basic CRUD operations, NewsDeck introduces several innovative features:

Analytics Dashboard: A custom StatsScreen that calculates article volumes, category shares, and trending topics.

Report-Ready Presentation: Charts and KPIs designed with academic reports and demonstrations in mind.

Cross-Cutting Contexts: Use of ThemeContext and FavouriteContext for consistent state across screens, demonstrating knowledge of global state management.

Account Stack Features: In addition to authentication, the app includes FAQ, feedback, and account update/change/delete functionalities, which show depth beyond minimal login flows.

This mix of utility and creativity ensures that NewsDeck is not just another "news reader" but a news analysis tool.

Section 2 – Wireframing

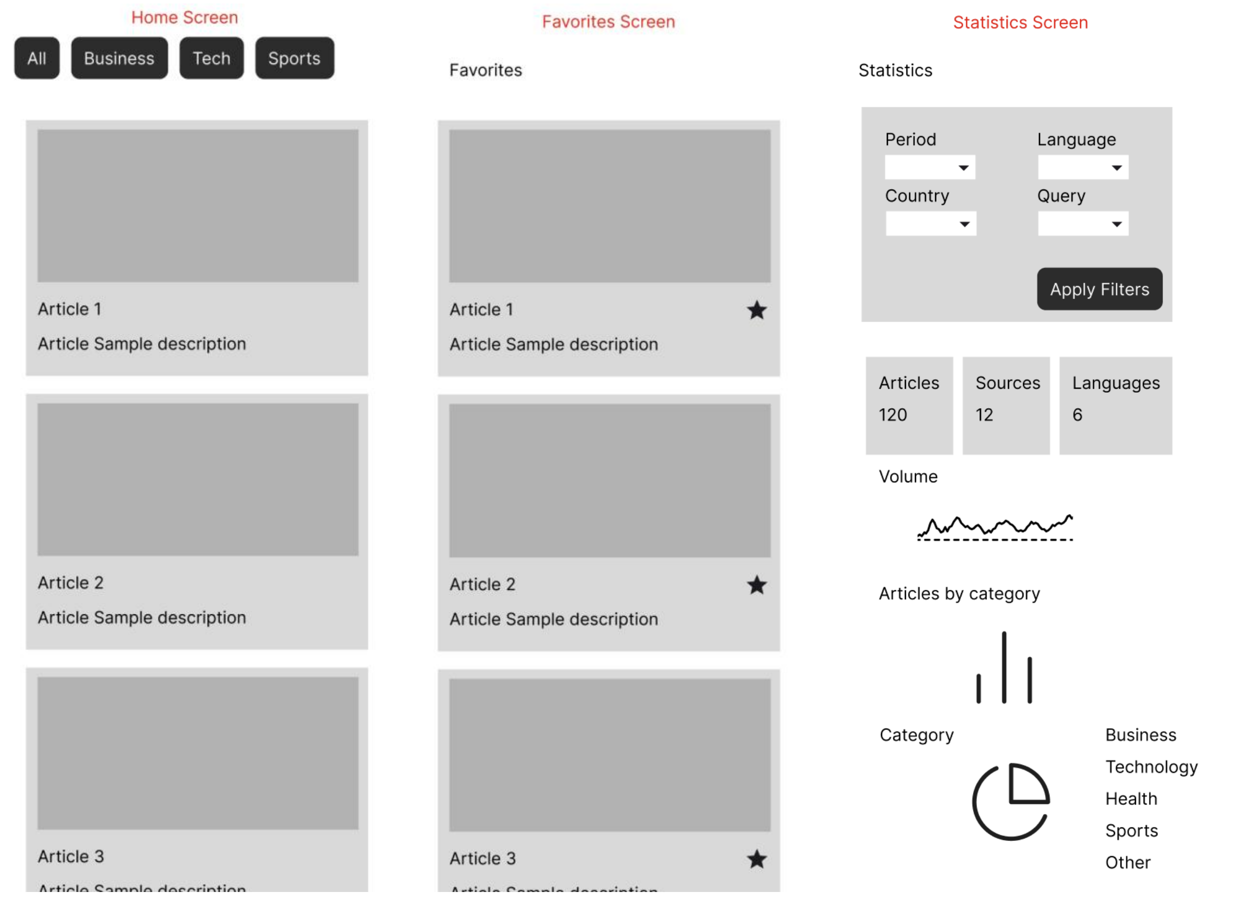
2.1) Early Wireframes and Layout Sketches

Initial wireframes were produced to establish the layout and flow of the app before any code was written. These low-fidelity sketches included:

A bottom navigation bar with four tabs: Home, Favourites, Analytics, and Account.

The HomeScreen arranged with a large headline card followed by smaller paired article cards.

A StatsScreen with placeholders for KPI cards, charts, and filters.



Description:

The first screen on the left is the HomeScreen, the aim of the screen is to display the list of articles based on the topic and is presented with the heading as well as a short description.

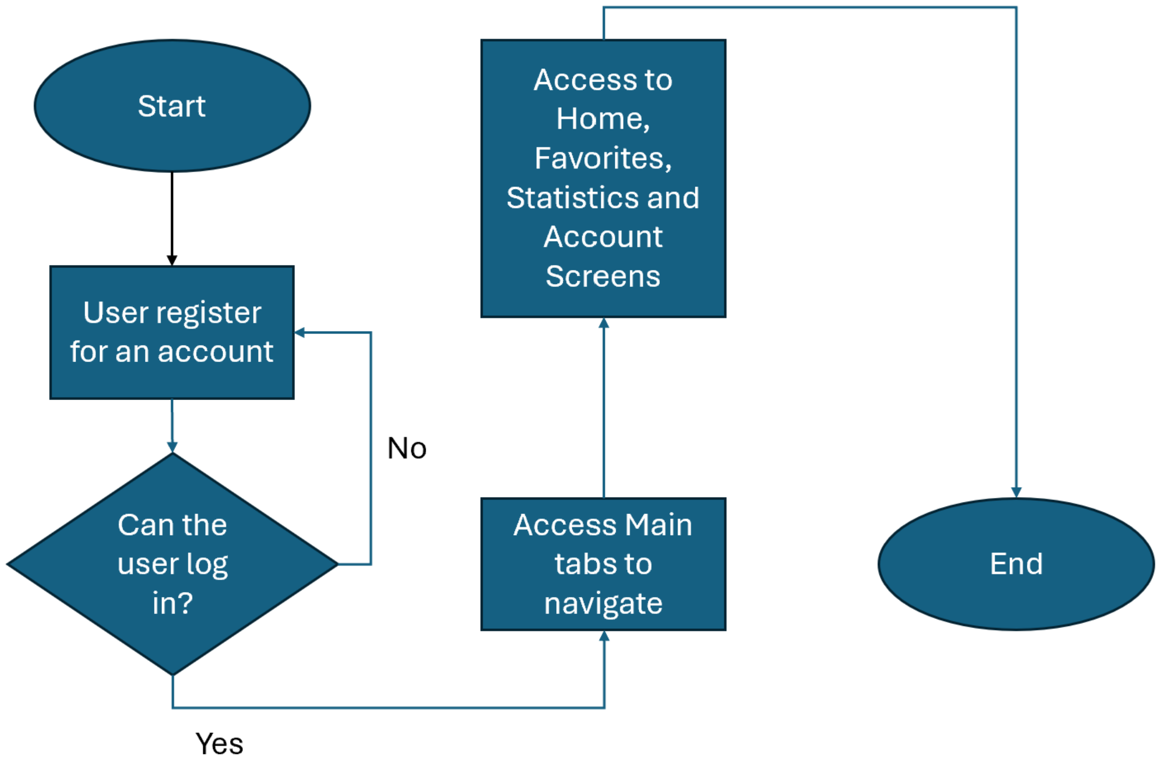
The second screen is the FavoritesScreen, the aim of this screen is to allow the user to view the selected article that the user has liked for easy access

The last screen on right is the StatisticsScreen, the aim of this screen is to allow the user to view the top topics viewed by other users.

2.2) User Flow Diagrams

A user flow diagram was created to map out navigation between screens. It highlighted how the user moves from login/register into the tabbed application, and how each feature could be accessed within 1–2 taps.

This simple flow guaranteed intuitive navigation.


Start
User register for an account
Can the user log in?
No
Yes
Access Main tabs to navigate
Access to Home, Favorites, Statistics and Account Screens
End

Description:

The diagram above demonstrates how a user might navigate and interact with my application, from registration to navigation access of the main tabs.

2.3) Reflection on Wireframing

Wireframing proved crucial for aligning the conceptual design with technical feasibility. It allowed early testing of visual hierarchy (e.g., whether the analytics KPIs should be at the top or bottom) and avoided costly redesigns later in development. The wireframes also formed the foundation for building the first working prototype.

Section 3 – User Feedback

3.1) Feedback Collection Methods

User feedback was gathered informally through peer reviews and usability testing sessions. Classmates and friends were asked to:

Navigate through the app's tabs.

Star/unstar articles and check the Favourites list.

Interpret the Analytics screen charts.

3.2) Key Insights from Feedback

Theme toggle was initially confusing when placed in the Home tab; users suggested moving it into Settings.

Top Words feature in the StatsScreen was found unhelpful, as common words diluted meaningful trends.

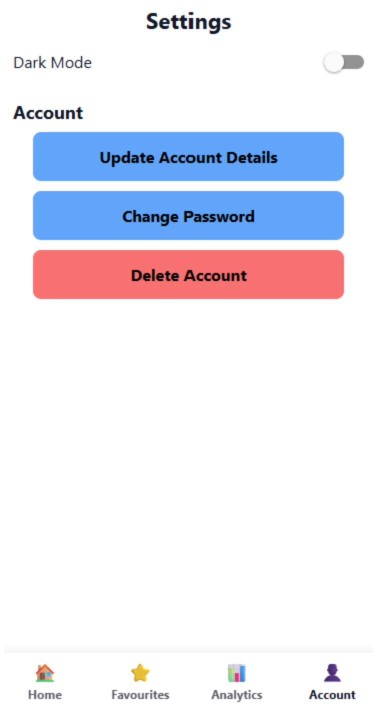
Testers valued supporting screens like FAQ and Feedback for enhancing usability.

3.3) Design Changes Based on Feedback

The theme toggle was relocated into Account → Settings for better discoverability.

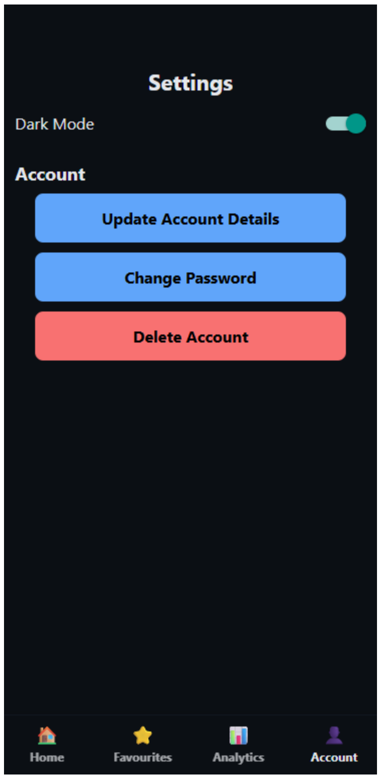
The StatsScreen replaced "Top Words" with Trending Topics, which aggregated article headlines into higher-value insights.

FAQ and Feedback screens were implemented to improve user support.


Settings

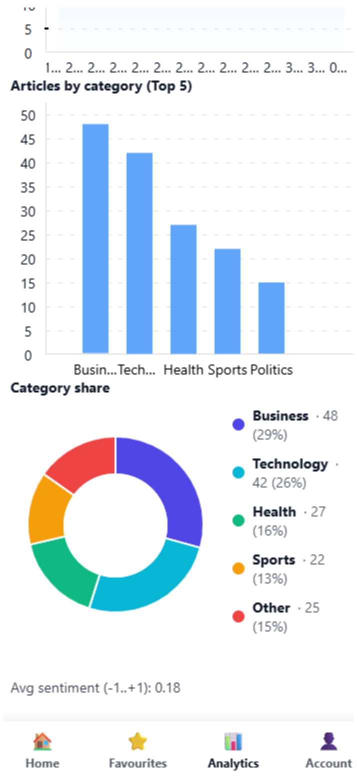
Dark Mode

Account


Settings

Dark Mode

Account



● Business · 48 (29%)
● Technology · 42 (26%)
● Health · 27 (16%)
● Sports · 22 (13%)
● Other · 25 (15%)

Avg sentiment (-1..+1): 0.18

Section 4 – Prototyping

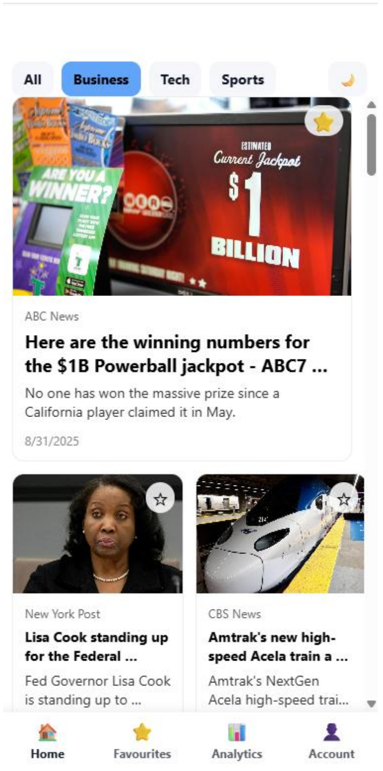
4.1) Transition from Wireframes to Prototype

The initial prototype was built in Expo Snack using static data, focusing on layout and navigation. Once the UI was validated, dynamic features such as API fetching and favorites were added.

4.2) Demo Screenshots of Prototype

Key prototypes included:

1) HomeScreen with live articles from NewsAPI.



ABC News

Here are the winning numbers for the $1B Powerball jackpot - ABC7 ...

No one has won the massive prize since a California player claimed it in May.

8/31/2025

New York Post

Lisa Cook standing up for the Federal ...

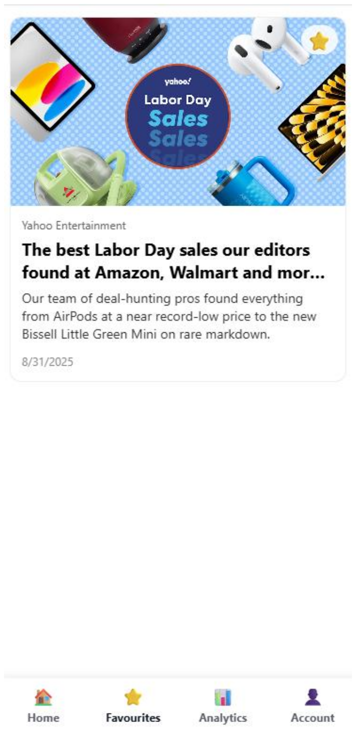
Fed Governor Lisa Cook is standing up to ...

CBS News

Amtrak's new high-speed Acela train a ...

Amtrak's NextGen Acela high-speed trai...

2) Favourites screen rendering starred articles.



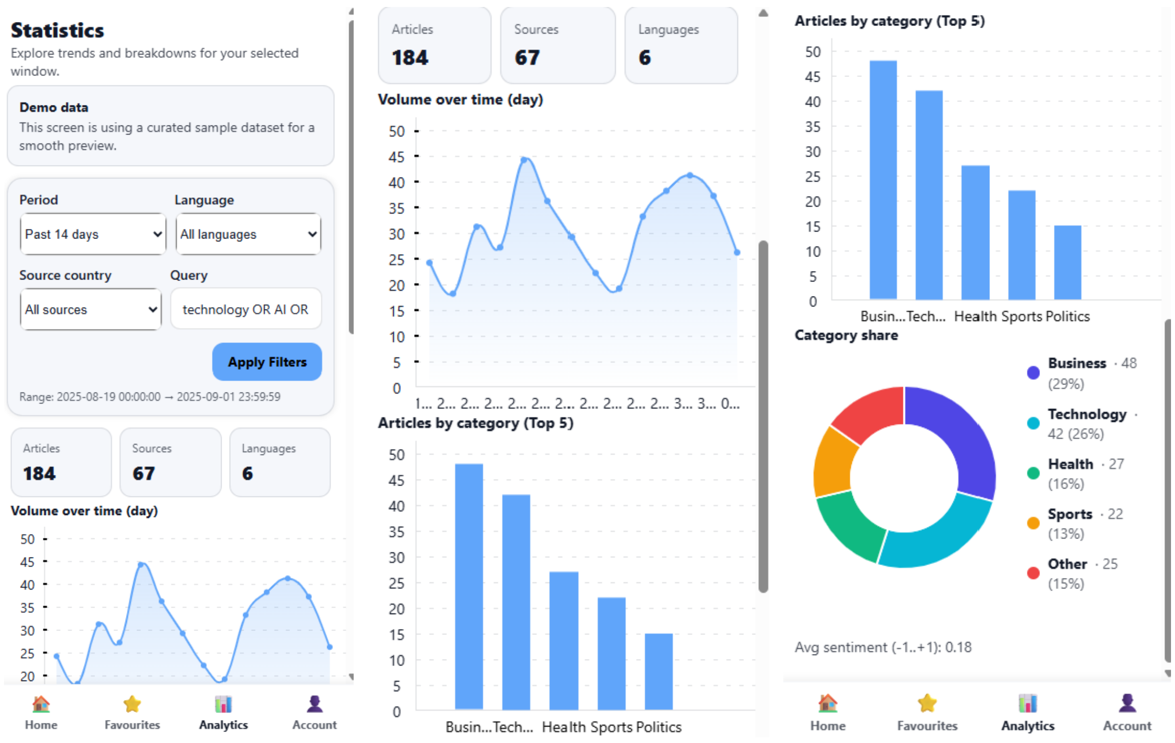
Yahoo Entertainment

The best Labor Day sales our editors found at Amazon, Walmart and mor...

Our team of deal-hunting pros found everything from AirPods at a near record-low price to the new Bissell Little Green Mini on rare markdown.

8/31/2025

3) StatsScreen with charts and KPIs.



Statistics

Explore trends and breakdowns for your selected window.

Demo data

This screen is using a curated sample dataset for a smooth preview.

Period: Past 14 days

Language: All languages

Source country: All sources

Query: technology OR AI OR

Apply Filters

Range: 2025-08-19 00:00:00 → 2025-09-01 23:59:59

Articles 184

Sources 67

Languages 6

Volume over time (day)

Articles by category (Top 5)

Category share

Business · 48 (29%)

Technology · 42 (26%)

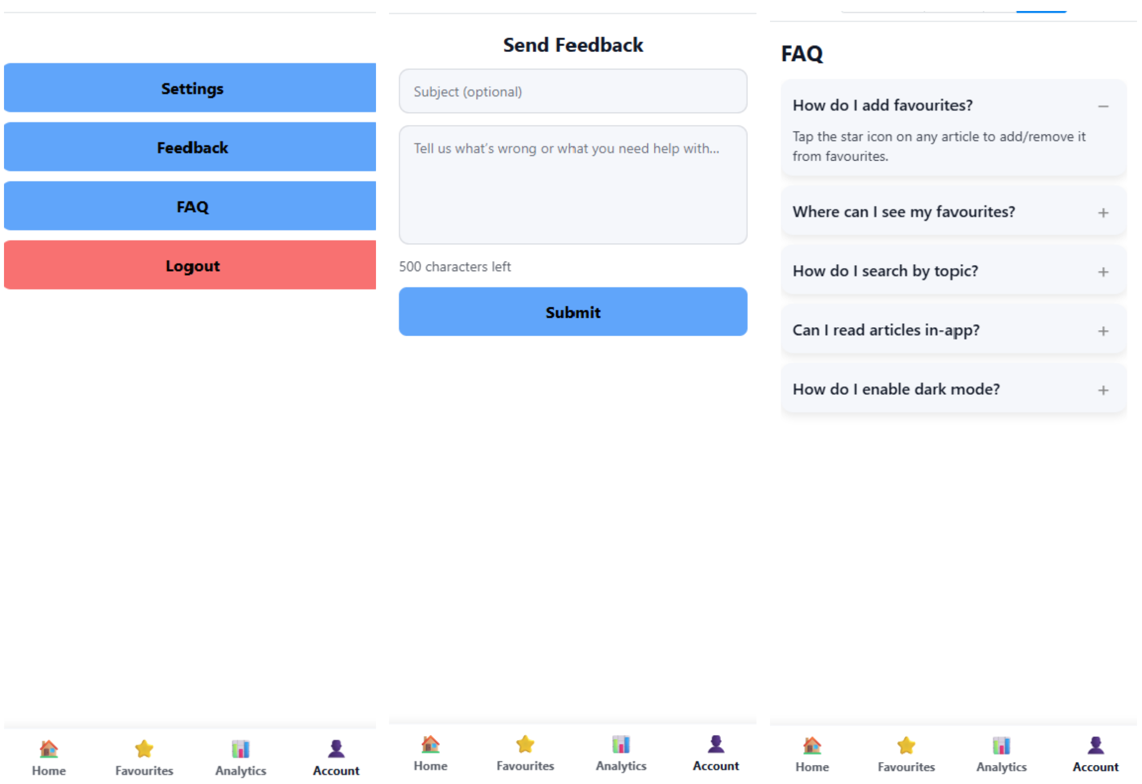
Health · 27 (16%)

Sports · 22 (13%)

Other · 25 (15%)

Avg sentiment (-1..+1): 0.18

4) Account stack with settings, feedback and FAQ



Settings

Feedback

FAQ

Logout

Send Feedback

Subject (optional)

Tell us what's wrong or what you need help with...

500 characters left

Submit

FAQ

How do I add favourites? −

Tap the star icon on any article to add/remove it from favourites.

Where can I see my favourites? +

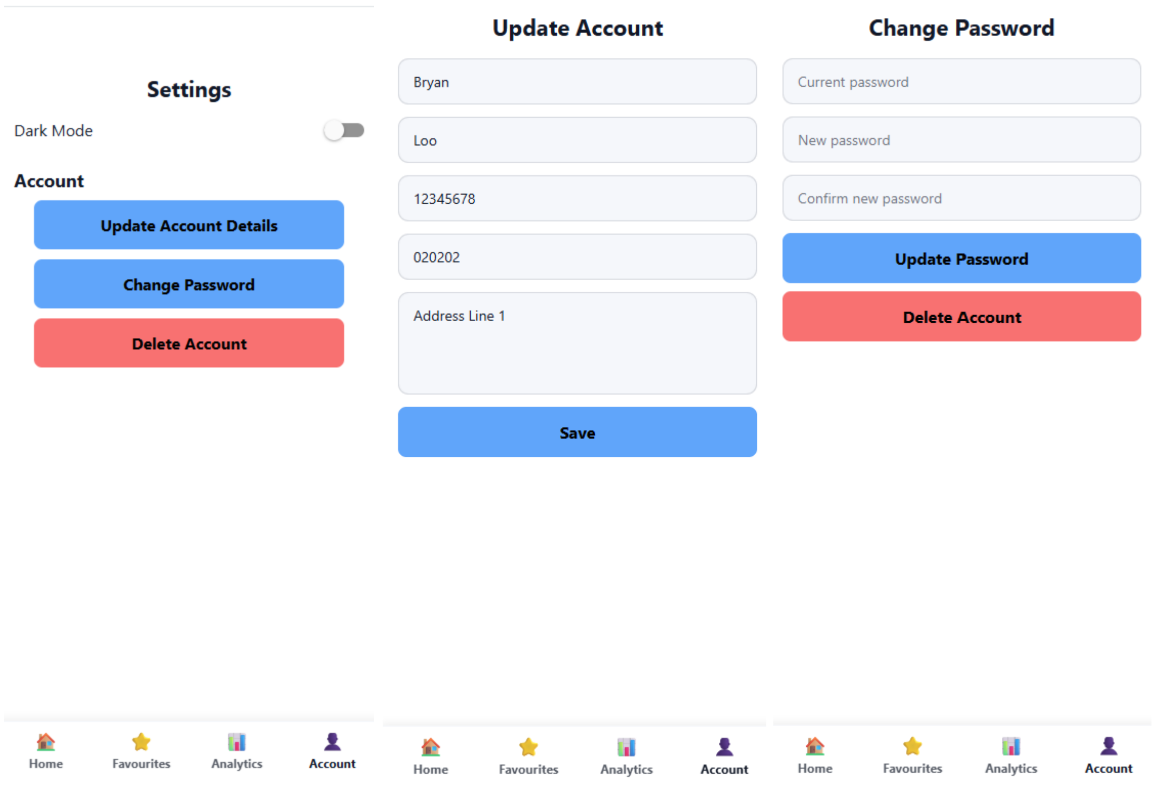
How do I search by topic? +

Can I read articles in-app? +

How do I enable dark mode? +

Home Favourites Analytics Account

5) Profile update/change password.



Settings

Dark Mode

Account

Update Account Details

Change Password

Delete Account

Update Account

Bryan

Loo

12345678

020202

Address Line 1

Save

Change Password

Current password

New password

Confirm new password

Update Password

Delete Account

Home Favourites Analytics Account

4.3) Challenges in Prototyping

Overcoming CORS restrictions on Expo Web required proxying.

Designing charts that would render consistently on mobile and web.

Ensuring the dark/light themes applied consistently across all screens.

Section 5 – Development

5.1) System Architecture Overview

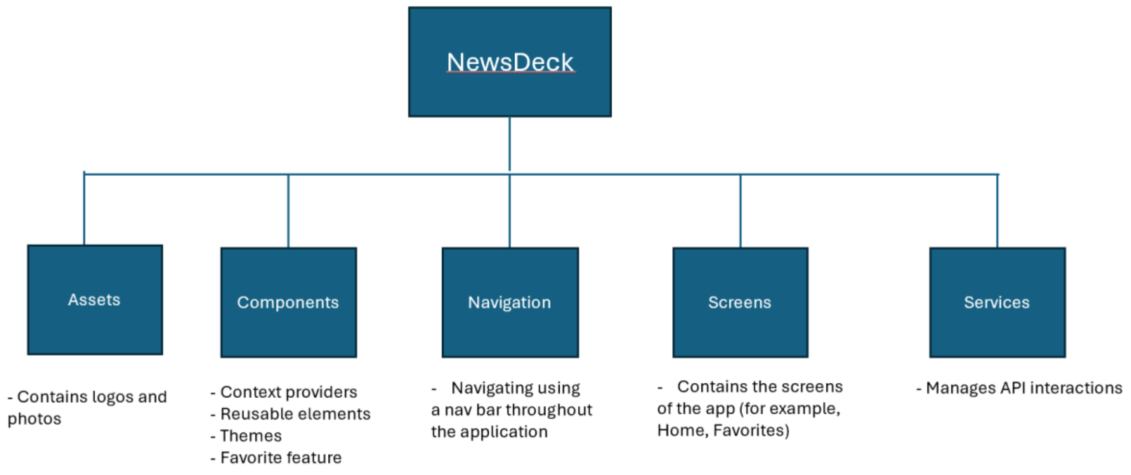
The app was structured with clear separation of concerns:

Services: news.js manages all API interactions.

Components: Context providers (Theme, Favourites) and reusable UI elements (Article card).

Screens: Encapsulate independent functionality (Home, Favourites, Stats, Account).

Root.js: Handles account management (in-memory) and navigation flow.


NewsDeck
Assets
Components
Navigation
Screens
Services
- Contains logos and photos
- Context providers
- Reusable elements
- Themes
- Favorite feature
- Navigating using a nav bar throughout the application
- Contains the screens of the app (for example, Home, Favorites)
- Manages API interactions

5.2) Implementation Details

5.2.1) API Fetching for Articles

NewsAPI is a simple HTTP-based REST API that provides access to real-time and historical news articles from thousands of online sources and blogs worldwide. It aggregates headlines, metadata, and content snippets from reputable publishers, allowing developers to query articles by parameters such as keywords, categories (e.g., business, technology, sports), countries, languages, and publication dates. NewsAPI is commonly used for building news feeds, content analysis tools, and data-driven dashboards because it returns standardized JSON responses, making integration straightforward.

getTopHeadlines() fetches category-based articles from NewsAPI.

searchEverything() allows date-ranged keyword queries for analytics.

Error handling ensures the app displays friendly messages when APIs fail.

```
// getTopHeadlines function retrieves the top articles
export async function getTopHeadlines({ country = "us", category } = {}) {
  const url = new URL(`${API_BASE}/top-headlines`);
  url.searchParams.set("country", country);
  if (category) url.searchParams.set("category", category);
  url.searchParams.set("pageSize", "30");
  url.searchParams.set("apiKey", API_KEY);
  const data = await fetchJson(url.toString());
  return Array.isArray(data?.articles) ? data.articles.filter((a) => !!a.urlToImage) : [];
}

// searchEverything function retrieves all the news
export async function searchEverything({ q = "news", fromISO, toISO, pageSize = 100 }) {
  const url = new URL(`${API_BASE}/everything`);
  url.searchParams.set("q", q);
  if (fromISO) url.searchParams.set("from", fromISO);
  if (toISO) url.searchParams.set("to", toISO);
  url.searchParams.set("sortBy", "publishedAt");
  url.searchParams.set("language", "en");
  url.searchParams.set("pageSize", String(pageSize));
  url.searchParams.set("apiKey", API_KEY);
  const data = await fetchJson(url.toString());
  return Array.isArray(data?.articles) ? data.articles : [];
}
```

5.2.2) Storage of User Details

Implemented as an in-memory object in Root.js.

Supports registration, login verification, update profile, change password, and delete account.

While insecure for production (plain-text passwords), it demonstrates full CRUD flow.

```
// Root function stores user registered data only once
export default function Root() {
  const [users, setUsers] = useState({});
  const [currentEmail, setCurrentEmail] = useState(null);
  const [screen, setScreen] = useState("login"); // 'login' | 'register' | 'app'

  const hasAnyAccount = Object.keys(users).length > 0;

  const addUser = (u) => {
    const key = (u.email || "").trim().toLowerCase();
    if (!key || users[key]) return false;
    setUsers((prev) => ({ ...prev, [key]: u }));
    return true;
  };
  const verifyLogin = (email, password) => {
    const key = (email || "").trim().toLowerCase();
    const rec = users[key];
    return !!rec && rec.password === password;
  };
  const onLoginSuccess = (email) => { setCurrentEmail(email); setScreen("app"); };
  const onLogout = () => { setCurrentEmail(null); setScreen("login"); };

  if (screen === "login") {
    return (
      <LoginScreen
        onSuccess={onLoginSuccess}
        goRegister={() => setScreen("register")}
        hasAnyAccount={hasAnyAccount}
        verifyLogin={verifyLogin}
      />
    );
  }
  if (screen === "register") {
    return (
      <RegisterScreen
        goLogin={() => setScreen("login")}
    addUser={addUser}
   />
  );
 }
 return (
  <Tabs
   onLogout={onLogout}
   currentEmail={currentEmail}
   users={users}
   setUsers={setUsers}
  />
 );
}
```

```
//Typical Article details
type Article = {
 source?: { id?: string|null; name?: string };
 author?: string|null;
 title?: string;
 description?: string|null;
 url?: string;           // unique; used as favourites key
 urlToImage?: string|null;  // used to decide which items to show in grid
 publishedAt?: string;     // ISO-8601
 content?: string|null;
}
```

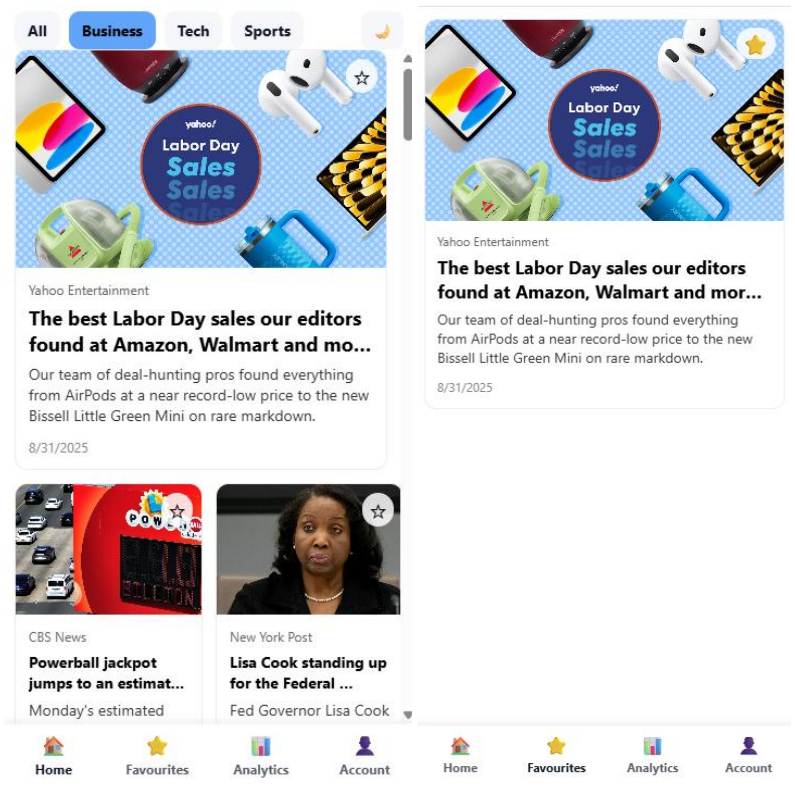
```
// User Model
type User = {
 email: string;
 password: string;
 firstName: string;
 lastName: string;
 phoneNumber: string;
 dateOfBirth: string;
 address: string;
 createdAt: number;
}
```

5.2.3) Favourites Feature

Managed globally via FavouriteContext.

Articles keyed by URL, enabling O(1) toggle operations.

Accessed across Home and Favourites screens seamlessly.



Yahoo Entertainment

The best Labor Day sales our editors found at Amazon, Walmart and mo...

Our team of deal-hunting pros found everything from AirPods at a near record-low price to the new Bissell Little Green Mini on rare markdown.

8/31/2025

CBS News

Powerball jackpot jumps to an estimat...

Monday's estimated

New York Post

Lisa Cook standing up for the Federal ...

Fed Governor Lisa Cook

Yahoo Entertainment

The best Labor Day sales our editors found at Amazon, Walmart and mor...

Our team of deal-hunting pros found everything from AirPods at a near record-low price to the new Bissell Little Green Mini on rare markdown.

8/31/2025

Description:

The user is able to star the article and it would be automatically added to the favorites screen. The user is also able to press on the article to view it in the website.

5.2.4) StatsScreen (Analytics Dashboard)

The StatsScreen is one of the most innovative aspects of the application, designed to move beyond simple article consumption and instead provide users with a high-level overview of news trends. At the top of the screen, three key performance indicators (KPIs) are displayed: the total number of articles retrieved, the number of unique sources, and the number of distinct languages represented within the dataset. These KPIs offer users a quick snapshot of the dataset's breadth and diversity, which is particularly useful when comparing news coverage over different timeframes.

Beneath the KPIs, the screen integrates multiple visualisations. A line chart shows article volume over time, allowing users to observe fluctuations in reporting activity and identify peaks of interest (e.g., surges in coverage following major events). A bar chart highlights the top sources by article count, which can reveal whether the dataset is dominated by a few outlets or is more evenly distributed. A donut chart visualises the share of articles across categories (such as business, technology, health, and politics), providing an immediate understanding of topical focus. Finally, a trending topics section extracts frequently co-occurring phrases from article titles and descriptions, offering insight into emergent themes beyond predefined categories.

To ensure that these computations remain performant, particularly when working with potentially large datasets, the component makes heavy use of memoisation techniques such as useMemo in React. This ensures that expensive operations (e.g., recalculating topic frequencies or aggregating by date) are only executed when the relevant input data changes. As a result, the StatsScreen balances data-rich insights with efficient rendering, making it both a practical and academically interesting example of combining analytics with user-facing design.

5.3) Design and Code Quality Practices

The development of NewsDeck was guided by a strong emphasis on code quality and maintainability. The application was structured according to a modular file hierarchy, separating services (API calls), components (reusable UI and contexts), screens (individual page logic), and navigation. This separation of concerns not only makes the codebase easier to read but also enables independent testing and future scalability.

At the implementation level, the project makes effective use of React hooks such as useCallback and useMemo. These hooks minimise unnecessary re-renders by ensuring that functions and computed values are only recreated when their dependencies change. This is particularly valuable in the Home and Stats screens, where repeated data transformations or re-renders could otherwise degrade performance.

Performance optimisation was also achieved through the careful use of FlatList for rendering lists of articles. Unlike simple array mapping, FlatList is designed to render only the items currently visible on the screen, recycling views as the user scrolls. This ensures smooth performance even when the API returns dozens of articles.

A further quality consideration was theming and responsiveness. Through the use of ThemeContext and a consistent colour palette, the application supports both light and dark modes, enhancing accessibility for different user preferences. Layout decisions were made with responsiveness in mind, ensuring that the interface works effectively across iOS, Android, and web platforms. Collectively, these practices align with professional software engineering standards and demonstrate that the project was not only functional but also designed for quality and sustainability.

5.4) Security and Privacy Considerations

While the current implementation is primarily a demonstration prototype, security and privacy were considered in planning for real-world deployment. At present, user account data is stored in-memory and passwords are handled as plain text. While acceptable for a classroom demonstration, this approach is insecure in production. A fully deployed system would require password hashing (e.g., using bcrypt) to prevent credential leaks in the event of a breach. Similarly, account authentication should rely on secure tokens (such as JWTs) stored in encrypted storage or secure keychains, rather than plain strings in state.

In terms of API security, the application currently embeds the NewsAPI key directly in the codebase, which poses a risk of exposure and rate-limit abuse. A best practice for production deployment would be to proxy API requests through a secure backend server, where keys are stored privately and requests can be monitored or cached for performance.

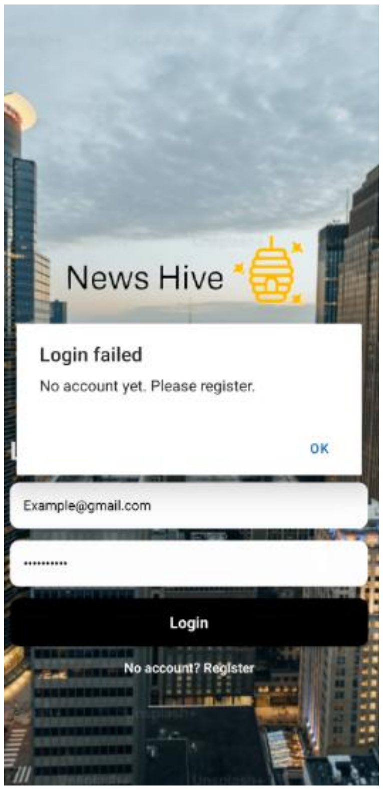
Finally, the application assumes that all communication occurs over HTTPS, ensuring encryption in transit. In a real-world scenario, this assumption would need to be enforced and validated, with certificate pinning used in mobile builds to mitigate man-in-the-middle attacks. These considerations reflect an awareness that while the current project demonstrates technical concepts effectively, security and privacy measures are essential prerequisites for moving from prototype to production-ready software.

Section 6 – Unit Testing

6.1) Testing Strategy and Tools

Manual testing performed in Expo iOS/Android simulators.

Iterative testing to verify error handling and UI flows.



News Hive

Login failed

No account yet. Please register.

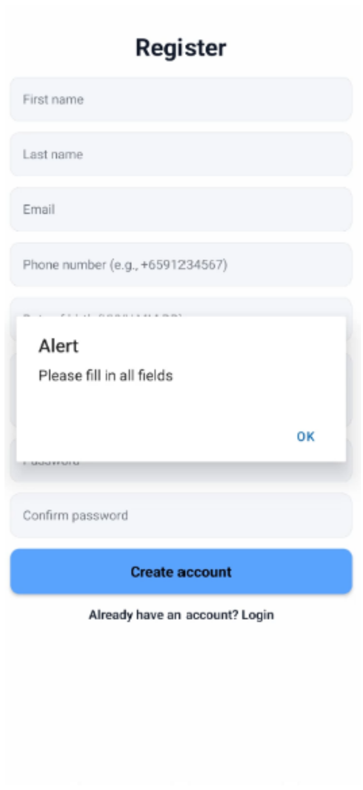
OK

Example@gmail.com

••••••••••

Login

No account? Register



Register

First name

Last name

Email

Phone number (e.g., +6591234567)

Alert

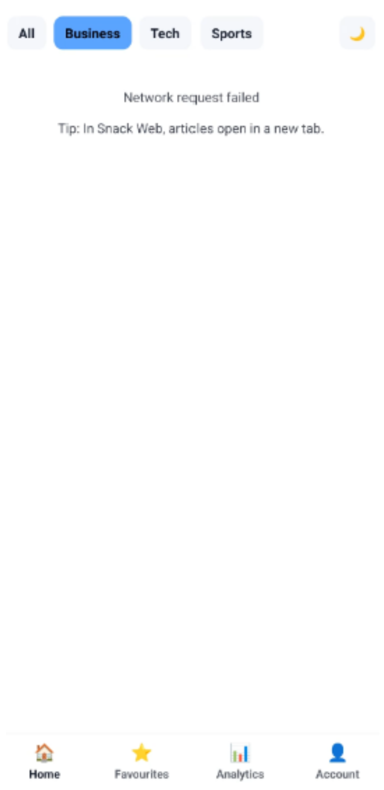
Please fill in all fields

OK

Confirm password

Create account

Already have an account? Login



Network request failed

Tip: In Snack Web, articles open in a new tab.

Description:

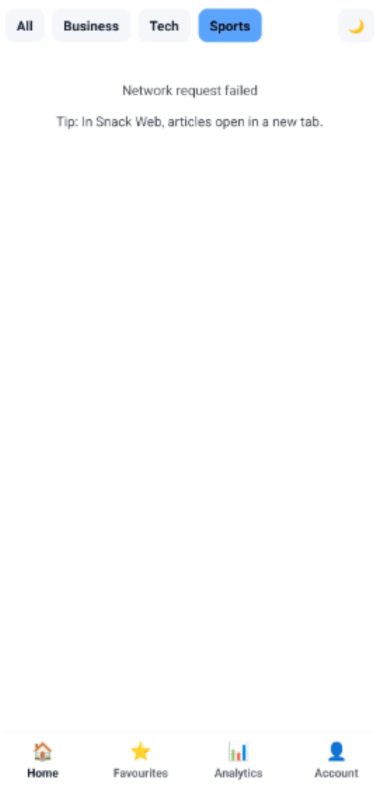
For testing I did manual testing on the UI elements. In the first screen I tested a failed login if the user did not register for an account and attempted to log in. In the second screen, I testing empty registration fields. And in the last screen I tested no internet to see if the articles were able to be retrieved despite the lost of internet.

6.2) Iterative Testing and Debugging Process

Tested API failure scenarios by disconnecting network.

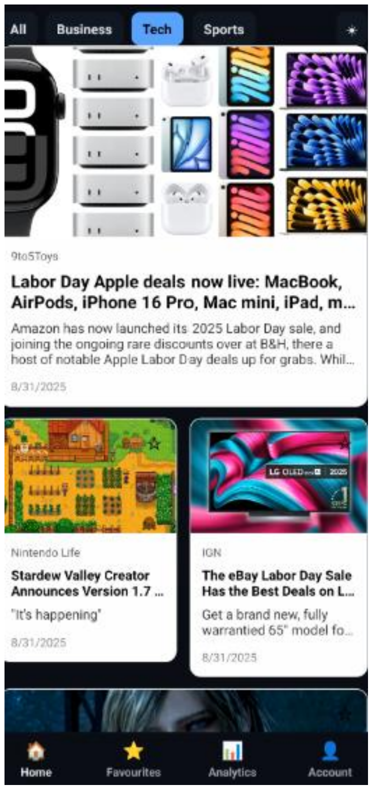
Verified invalid inputs trigger error alerts.

Debugged theming consistency across components.



Network request failed

Tip: In Snack Web, articles open in a new tab.



9to5Toys

Labor Day Apple deals now live: MacBook, AirPods, iPhone 16 Pro, Mac mini, iPad, m...

Amazon has now launched its 2025 Labor Day sale, and joining the ongoing rare discounts over at B&H, there a host of notable Apple Labor Day deals up for grabs. Whil...

8/31/2025

Nintendo Life

Stardew Valley Creator Announces Version 1.7 ...

"It's happening"

8/31/2025

IGN

The eBay Labor Day Sale Has the Best Deals on L...

Get a brand new, fully warrantied 65" model fo...

8/31/2025

Description:

In the first screen, there was no internet and the theme was set to light theme, and in the second screen I tested with internet and in the dark theme mode.

Section 7 – Evaluation

7.1) Strengths of the Application

One of the major strengths of NewsDeck lies in its clean and modular architecture, which demonstrates good software engineering practice. By separating concerns into services, components, contexts, and screens, the application achieves both clarity and maintainability, which would allow future developers to extend functionality with minimal code coupling.

A further strength is the inclusion of the StatsScreen, which sets this project apart from typical news reader applications. Instead of simply providing raw articles, the app integrates lightweight analytics in the form of KPIs, trend charts, and category breakdowns.

This design choice highlights creativity and innovation, while also ensuring that users are not just consumers of information but can also interpret patterns across the news cycle. In addition, the implementation of a theme toggle enhances accessibility, allowing users to choose between dark and light modes according to their preference or environmental context (e.g., night-time reading).

This reflects attention to user-centred design principles. Finally, the favourites feature adds a degree of personalisation, enabling users to curate their own customised feed of saved articles. This not only increases usability but also demonstrates how simple functionality can significantly improve user engagement.

7.2) Limitations and Challenges

Despite these strengths, NewsDeck also has several limitations. The most prominent technical constraint is that user accounts are stored in-memory only. This means that all data is lost when the application reloads, which restricts the app to being a demonstration rather than a production-ready solution.

Another limitation is that the StatsScreen filters are user interface only, meaning that while they appear functional, they do not dynamically modify queries to the API in the current demo version. This reduces the realism of the analytics and weakens the ability to carry out personalised trend exploration.

Furthermore, the charts used within the application are basic and lack interactivity such as tooltips, drill-downs, or hover states. While they effectively display aggregated data, they do not allow users to explore insights in greater depth. These limitations were partly due to the scope and time restrictions of the project but provide clear opportunities for further development.

7.3) User Experience Review

Feedback from peer testers indicated that the user experience was generally smooth and intuitive. Users consistently reported that navigation between tabs was straightforward and the layout of information was clear.

The horizontal filter chips on the HomeScreen were particularly praised for allowing rapid switching between categories such as Business, Technology, and Sports. Testers also expressed high satisfaction with the analytics dashboard, noting that it added genuine value compared to standard news apps by giving them a sense of “what is trending.”

The dark theme option was valued especially for use in low-light environments, which aligns with current accessibility best practices. Although some testers highlighted the need for persistent accounts and more advanced analytics, overall feedback confirmed that the application successfully balanced functionality with a clean and engaging interface.

Section 8 – Future Work

8.1) Persistence and Database Integration

A key area of future development is the integration of persistent storage. At present, accounts and favourites reset on reload due to their in-memory implementation. A short-term solution would be to use AsyncStorage to persist data locally on the device, ensuring that user preferences and saved articles are retained across sessions.

In the longer term, the application could be connected to cloud-based services such as Firebase Authentication and Firestore or MongoDB Atlas. This would enable secure account creation, real-time synchronisation of favourites across multiple devices, and robust scalability. Such integration would also align the app with professional development standards, where persistence is fundamental for usability and trust.

8.2) Enhanced Analytics and Charting

The StatsScreen could be expanded significantly in future iterations. Currently, the charts are basic and static, limiting their analytical depth. By adopting third-party charting libraries such as Recharts or Victory Native, the application could provide interactive and visually compelling charts with animations, legends, and tooltips.

Beyond charting, the analytics could incorporate natural language processing (NLP) techniques to perform sentiment analysis, clustering of related topics, or even identification of misinformation patterns.

These enhancements would not only increase the academic value of the project but also align the app more closely with real-world data science practices.

8.3) Deployment and Scalability

For broader impact, the application could be prepared for deployment and scalability. The Expo framework supports publishing as a Progressive Web App (PWA), which would allow users to access NewsDeck from desktop browsers in addition to mobile platforms.

Additionally, API calls currently go directly to NewsAPI, exposing the application to rate limits and potential key leakage. A scalable solution would involve implementing a backend proxy server, which could securely manage API keys, cache results to reduce redundant calls, and apply load balancing for performance.

With these improvements, NewsDeck would be well-positioned to transition from a student project into a production-ready application capable of supporting a wide user base.

Section 9 – Conclusion

9.1) Conclusion

NewsDeck demonstrates how an academic project can evolve beyond a simple news feed to become a modular, analytics-driven app. It applies separation of concerns, state management, theming, and report-ready analytics to deliver both functionality and innovation. With persistence, advanced analytics, and deployment, the project could mature into a fully production-ready solution.

9.2) References

Expo. (2025). Expo Documentation. Available at: https://docs.expo.dev

Facebook. (2025). React Native Documentation. Available at: https://reactnative.dev/docs

NewsAPI. (2025). News API Documentation. Available at: https://newsapi.org/docs

9.3) Video Demo Link

https://youtu.be/TGnkmvBHCR8