

IN3210 Mobile Applications Development

Assignment 2

Cross-Platform Mobile Development with React Native

Objective :

Design and develop a **functional mobile application** using React Native (via Expo CLI or React Native CLI). The application should follow best practices & Standards in **UI development, navigation, API integration, state management, and data persistence**.

Topic Selection - Based on Last Digit of Index Number

Last Digit	Domain / App Theme	Example App Idea
0, 5	Campus & Education Tech	“UoM Course Finder” or “SkillUp” – Browse and save online courses or campus events
1, 6	Health & Wellness	“FitBuddy” – Track exercises, water intake, and wellness tips
2, 7	Entertainment & Media	“StreamBox” – Browse trending movies, songs, or podcasts
3, 8	Travel & Transport	“GoMate” – View public transport schedules or explore destinations
4, 9	Sports & Lifestyle	“Sportify” – View matches, players, and scores

Key Requirements:

User Authentication

- Implement a **user registration** and **login** flow.
- Use **React Hooks** to handle form data and validations (e.g., with Yup or custom validation).
- On successful login, navigate to the home screen.
- The logged-in user’s **name or username** should be visible in the app’s header or profile area.

Tip: Store authentication state locally according to the best security practices.

Navigation Structure

- Use a navigation library such as **React Navigation** or **Expo Router**.
- Implement **stack, Bottom tab or Drawer navigations** according to your suitability.

Home Screen (Dynamic Item List)

- Display a list of relevant items which fetched from an API according to your domain:
- Each item must appear as a **card** containing:
 - Image / Icon
 - Title
 - Description or status (e.g., “Active”, “Upcoming”, “Popular”)

Item Interaction and State Management

- When a user **taps an item**, open the **Details Screen**.
- Manage state using a **state management solution** such as Redux Toolkit.

Favourites

- Allow users to **mark items as favourites** and view them in a separate screen.
- Persist favourite items in a proper way.

Styling and UI

- Apply consistent and visually clean styles.
- Utilize Feather Icons for all iconographic elements.
- Maintain responsive design for various screen sizes.

Bonus Features (for additional marks)

- Dark mode toggle.

Key Considerations :

- Feature-based commits: Make commits related to specific features.
- Proper validations: Implement appropriate input validations.
- Decoupled, testable, and reusable code: Structure your code to be modular and easy to test.
- Best Practices and industry Standards: Adhere to best practices and coding standards.

Use dummy APIs for **user authentication and data fetching**.

Example:

- <https://free-apis.github.io/#/>
- <https://dummyjson.com/docs>

Example public APIs per domain:

Domain	Sample Public API
Education	<u>Open Library API</u>
Health	<u>API Ninjas Fitness API</u>
Entertainment	<u>The Movie Database API</u>
Transport	<u>Transport API UK</u> or mock JSON
Sports	<u>TheSportsDB</u>

Deliverables :

Submit a ZIP file containing:

- URL to your **public GitHub repository** with all project files.
- **Screenshots** of key app screens (Login, Home, Details and etc).
- **Short demo video (≤2 minutes)** showing the app's core flow.

Evaluation Criteria :

Criteria	Marks
Authentication & Validation	15
Navigation Implementation	10
API Integration & Data Display	15
State Management	15
UI/UX Design & Responsiveness	15
Code Quality & Best Practices/Standards	20
Demo Video	5
Bonus Feature	5

Deadline : 23rd November, 2025