📱 Final Project Submission Guidelines

React Native / Mobile Application Development

📌 Overview

As the coursework for **React Native / Mobile App Development** is now complete, this **Final Project** will serve as a comprehensive assessment of your learning.

You are required to **design, build, and submit a complete Full-Stack Mobile Application** that demonstrates your understanding of mobile development, backend integration, and real-world application architecture.

🧱 Project Type & Technology Requirements

1️⃣ Application Type (Mandatory)

- The project **must be a Full-Stack Mobile Application**
- It must include:
    - A **Mobile App (Frontend)**
    - A **Server-side Backend**
    - A **Database**

❗ *Pure frontend or UI-only projects will NOT be accepted.*

2️⃣ Frontend (Mobile App)

You may choose **any one** of the following:

✅ **Recommended (Preferred):**

- **React Native**

✅ **Also Accepted:**

- Native **Android** (Java / Kotlin)
- Native **iOS** (Swift)

> Your choice should reflect your confidence and prior experience. Evaluation will focus on **quality and correctness**, not just the framework.

3️⃣ Backend (Mandatory)

Your app **must communicate with a backend server**.

Accepted backend technologies:

- Node.js (Express / Fastify / NestJS)
- Java Spring Boot
- Any equivalent server-side framework

Backend responsibilities must include:

- API endpoints (REST or GraphQL)
- Business logic
- Data validation
- Authentication / Authorization (if applicable)

4️⃣ Database (Mandatory)

You must use a **server-side database**, such as:

- MongoDB
- PostgreSQL
- MySQL
- Firebase / Supabase
- Any other structured or NoSQL database

> ❗ Local storage alone (AsyncStorage / SQLite only) is **not sufficient**.

🔑 Functional Requirements

Your project should clearly demonstrate:

- 📲 Real-world use case (not a demo or toy app)
- 🔄 API-based data flow between app and server
- 🧠 Proper state management
- 🧾 CRUD operations (Create, Read, Update, Delete)
- 🚨 Error handling and edge case management
- 🔐 Authentication (recommended but optional for simple apps)

🎨 UI / UX Expectations

- Clean and usable UI
- Consistent layout and spacing
- Meaningful loading states
- User-friendly navigation

- Responsive behavior across screen sizes

Visual perfection is not required, **usability is**.

📁 Submission Requirements

Each student must submit:

1️⃣ **GitHub Repository Link**

- Separate folders or repositories for:
  - `mobile-app`
  - `backend`
- Clean commit history is encouraged

2️⃣ **README.md (Mandatory)**

Your README must include:

- Project Title & Description
- Problem Statement
- Tech Stack Used
- Features Implemented
- How to run the project locally (steps)
- API documentation (basic)

3️⃣ **Demo Video (Mandatory)**

- 5–10 minutes
- Must show:

- App functionality
- Backend interaction
- Database operations

4️⃣ **PRD / Feature List**

- Clearly mention:
  - Core features
  - Optional/future features

📊 Evaluation Rubric (100 Marks)

Criteria	Marks
Project Idea & Use Case Clarity	10
Frontend Implementation (Mobile App)	25
Backend API Design & Logic	20
Database Design & Integration	15
Full-Stack Integration	15
UI/UX & User Flow	10
Code Quality & Project Structure	5
Total	**100**

🚫 Common Reasons for Rejection

- ❌ No backend
- ❌ No database

- ❌ App not functional
- ❌ Copied project without understanding
- ❌ Missing demo video or README
- ❌ Late submission

⏰ Submission Deadline

🗓️ **Final Submission Deadline:**

21st January (No Extensions)

Submission form - https://forms.gle/Eq2iqfTCnkvrRjcU7

Late submissions will **not be evaluated** under any circumstances.

📢 Final Notes

- This project is a **reflection of your real-world readiness**
- Focus on **clarity, correctness, and completeness**
- Simple ideas executed well score higher than complex but broken apps