Node.js + Flutter Course Project - Requirements & Assessment Guide

Group Size: Maximum 5 students **Lecturers:** ABDISALAN ABDULLAHI MOHAMED & SHARMAKE ALI KAHIE

1. Project Overview

Students are required to design and develop a complete backend system using Node.js, exposed through RESTful APIs, and consumed by a Flutter mobile application.

The project evaluates students' practical understanding of:

- Backend development with Node.js
- RESTful API design
- Database integration
- Authentication & authorization
- Team collaboration and project management using Jira

2. Technology Requirements

Backend (Mandatory)

- **Node.js**
- **Express.js**
- RESTful API architecture
- JWT-based or session-based authentication

Database (Choose ONE)

- MongoDB
- MySQL
- PostgreSQL

Frontend (Mandatory)

- **Flutter** (Mobile Application only)
- *Web frontends (HTML/CSS/JS, React, Vue, Angular) are NOT allowed.*

Project Management & Version Control (Mandatory)

- **Git & GitHub**
- **Jira** for:
    - Task creation
    - Task assignment
    - Progress tracking
    - Sprint management (if applicable)

All development tasks must be documented and managed through Jira.

3. Project Scope (Minimum Features)

3.1 User Management

- User registration
- User login
- Authentication (JWT or sessions)
- Role-based access control (Admin/User or similar)

3.2 CRUD Operations

Each system must support full CRUD functionality:

- Create data & Update data
- Read data
- Delete data

3.3 API Design Standards

- Well-structured RESTful endpoints
- Proper HTTP methods:
    - GET
    - POST
    - PUT/PATCH
    - DELETE

- Proper HTTP status codes:
  - 200 (OK)
  - 201 (Created)
  - 400 (Bad Request)
  - 401 (Unauthorized)
  - 404 (Not Found)
  - 500 (Internal Server Error)

3.4 Error Handling

- Centralized error handling middleware
- Request validation errors
- Clear and meaningful error messages

4. Project Examples (Choose Any Domain)

Students may choose any real-world system, such as:

- Student Management System
- Online Library System
- E-commerce Backend
- Hospital Appointment System
- Learning Management System

Creativity is encouraged, but the system must be practical, functional, and complete.

5. Code & Design Requirements

- Clean, readable, and well-documented code
- **Proper folder structure:**
  - routes
  - controllers
  - models
  - middleware

- Environment variable usage (.env file)
- .env.example file included (no real secrets)
- Clear separation of concerns
- Reusable middleware (auth, validation, error handling)

6. Group Work Rules

- **Maximum 5 students per group**
- All members must understand the entire system
- Each student will be individually questioned during evaluation
- Marks are given individually, not equally by default
- Jira activity will be used to assess individual contribution

7. Project Submission Requirements

Each group must submit:

7.1 GitHub Repository

- Public or private repository (with lecturer access)
- Proper commit history

7.2 Environment File

- .env.example file only
- **No real secrets or credentials**

8. Academic Integrity Policy

- **Plagiarism is strictly prohibited**
- Projects copied from the internet without understanding will receive **ZERO (0) marks**
- Each student must be able to clearly explain their contribution
- Inability to defend the project will result in mark reduction

9. Mandatory Pre-Submission Information

All students must submit the following information before the deadline:

Required Details

- Group members' full names and student IDs
- Selected project title/topic
- Group leader's name

Rules & Conditions

- ❌ **Late submissions will NOT be accepted**
- ❌ **Students who fail to submit will receive ZERO (0) marks**
- ❌ **Group changes after submission are strictly prohibited**
- Each student will be individually examined and graded
- 📅 **Demonstration Date:** 12 Feb 2026

IMPORTANT NOTICE

This instruction is final and non-negotiable. Failure to comply with any requirement may result in mark reduction or project rejection.

Lecturers: ABDISALAN ABDULLAHI MOHAMED & SHARMAKE ALI KAHIE