Assessment 1

Total Marks 20

Assignment: Software requirements analysis and design (Full-Stack CRUD Application Development with DevOps Practices)

Objective

You have been provided with a starter project that includes user authentication using Node.js, React.js, and MongoDB. Your task is to extend this application by implementing **CRUD (Create, Read, Update, Delete)** operations of different features for a real-world application of your choice, while following industry best practices such as:

- Project Management with JIRA
- Requirement Diagram, Block Definition Diagram (BDD), Parametric Diagram using SysML
- Version Control using GitHub
- CI/CD Integration for Automated Deployment

GitHub link of the starter project: https://github.com/rajuiit/sdlapps

Requirement

1. Choose a Real-World Application

We will send you an email to choose a Real-World project. If you face any difficulties in choosing your project, please contact your tutor.

2. Project Design with SysML and Project Management with JIRA

- Draw a requirements diagram, Block Definition Diagram (BDD), and Parametric Diagram based on your project (Connect all functional features).
- Create a JIRA project and define:
  - Epic
  - User Stories (features required in your app)

  - Child issues or Subtasks (breaking down development work)
  - Sprint Implementation (organizing work into milestones)
- Provide your JIRA board URL in the project README.

3. Backend Development (Node.js + Express + MongoDB)

- Set up and configure the **MongoDB database connection.**
- Implement various backend functions for handling application data. Ensure that all functions are compatible with an **Application Programming Interface (API)** structure (Follow existing patterns used in the Task Manager App where applicable).
- Implement **CRUD** operations for creating, reading, updating, and deleting records for each functionality.

4. Frontend Development (React.js)

- Create a user-friendly interface to interact with your API endpoint (Follow task manager app).
- Implement different forms for adding, updating, and deleting records.
- Display data using tables, cards, or lists (Follow how we showed data in task manager app, try to implement better visualization for the frontend.)

5. Authentication & Authorization (Prerequisite Task)

- Ensure only authenticated users can access and perform CRUD operations. (Already developed in your project)
- Use JWT (JSON Web Tokens) for user authentication (Use the task manager one from .env file).

6. GitHub Version Control & Branching Strategy

- Use GitHub for version control and maintain:
  - main branch (stable production-ready code)
  - Feature branches for each new feature
- Follow proper commit messages and pull request (PR) for code reviews.

7. CI/CD Pipeline Setup

- Implement a CI/CD pipeline using GitHub Actions to:
  - Automatically run tests on every commit/pull request (**Optional**).
  - Deploy the backend to AWS. (Use the QUT provided EC2 instance)
  - Deploy the frontend to AWS.

- Document your CI/CD workflow in the README.

Submission Requirements

A report contains the following (Provide screenshots as evidence for each implemented task. **The screenshot should contain your username from JIRA, GITHUB, and AWS**):

- JIRA Project Management (Provide screenshots in the report of at least two epics, including user story, subtasks. **Please don't provide the User Authentication epic.** Provide your JIRA Board URL in the report and README file as well. Through the JIRA Board, we will systematically review the completeness of the project features, organised under Epics, User Stories, and Sub-tasks.
- Requirement diagram, Block Definition Diagram (BDD), Parametric Diagram (Using project features).
- GitHub Repository (backend/ and frontend/) link. We will review your code implementation, which you followed from the task description. We will also review your commits, main branch, feature branches, and pull requests. (**Please note that the authorisation (Log In, Registration) is the prerequisite for backend development.**)
- CI/CD pipeline details step by step screenshot.
- README.md with:
  - Project setup instructions.
  - Public URL of your project.
  - Provide a project-specific username and password if we need to access your dashboard.

Mark Distribution:

	Marks
• Project design with SysML	3
• Project Management with JIRA	4
• Backend Development (Node.js + Express + MongoDB)	3
• Frontend Development (React.js)	2
• GitHub Version Control & Branching Strategy	2
• CI/CD Pipeline Setup	5
• README.md and Report	1

Total Marks: 20

Assessment Criteria:

- Clarity and completeness of Jira board and SysML models.
- Adherence to Git best practices and practical contributions.
- Successful implementation, deployment and CI/CD pipeline.
- Problem-solving skills and the ability to go beyond basic requirements.