Connecting Intelligence

iCET CERTIFIED MASTER

FINAL PROJECT

Introduction

The objective of this final coursework is twofold. Firstly, it provides us with an opportunity to showcase the culmination of our learning and skills acquired throughout the course. Secondly, it serves as a platform to demonstrate our ability to conceive, design, implement, and evaluate a software solution to address a real-world problem or fulfill a specific need.

In this report, we will present our chosen software project, detailing its objectives, requirements, design considerations, implementation details, testing methodologies, and evaluation criteria. Additionally, we will reflect on the challenges encountered during the development process and discuss the lessons learned. Finally, we will conclude with insights gained from this experience and potential avenues for future work.

Through this final coursework, we aim to not only demonstrate our proficiency in software development but also to underscore our commitment to continuous learning and improvement in this dynamic field.

1. Project

Feel free to choose a project topic, but be sure to incorporate some innovative elements.

2. Evaluation Criteria

Criteria	Considerations	Weight
Functionality	• Does the software fulfill its intended purpose? • Are all specified features implemented correctly? • Does the software handle edge cases and errors gracefully?	15 %
User Interface (UI) Design	• Is the user interface intuitive and easy to navigate? • Does the UI follow design principles such as consistency and responsiveness?	10%
Code Quality:	• Is the code well-organized and modular? • Are appropriate design patterns and best practices followed? • is the codebase maintainable and easily extensible?	25%

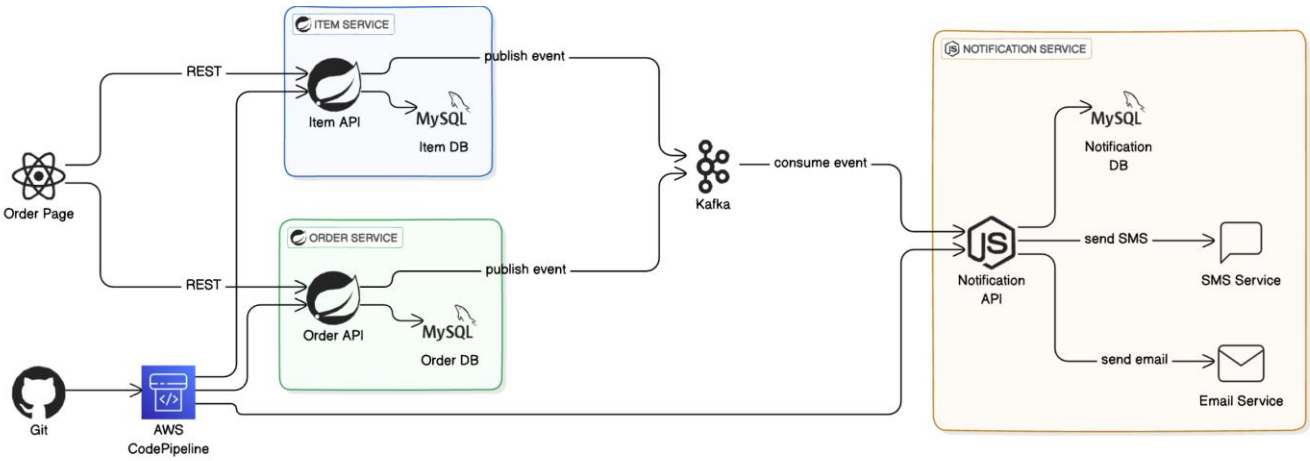
	• Are comments and documentation provided where necessary?	
Performance	• Does the software respond quickly to user interactions? • Does the software handle large datasets or high loads effectively?	5%
Testing	• Are unit tests, integration tests, and/or end-to-end tests implemented?	5%
Security	• Does the software implement appropriate security measures to protect against common vulnerabilities (e.g., injection attacks, cross-site scripting)? • Are user inputs validated and sanitized to prevent security breaches? • Is sensitive data handled securely (e.g., encryption of passwords)?	10%
Scalability:	• Is the software designed to scale effectively as the user base or data volume grows?	5%
Documentation (ER, UseCase, Highlevel Architecture, API Documentation),	• Is there developer documentation detailing the architecture, APIs, and implementation details? • Is the documentation clear, concise, and up-to-date?	15%
Version Control	• Was version control (e.g., Git) utilized properly?	5%
Project Management:	• Were project management methodologies (e.g., Agile, Scrum) effectively applied?	5%

3. Technology Selection

4. Design Architecture

Design your idea in to software solution adherering event driven microservice architecture. Use UI SPA framework with styling library to create responsive views if applicable.

Event Driven Microservice Architecture



5. Documentation

You are requested to provide detailed documentation considering the below facts.

- **Title Page:** A title page with the project title, the student's name, the diploma program, the institution, and the date.
- **Abstract:** A concise summary of the project, including its objectives, methods, and findings.
- **Table of Contents:** A detailed listing of all sections and subsections with corresponding page numbers.
- **Introduction:** An introduction to the project, providing background information, the problem statement, objectives, and scope.
- **Literature Review:** A review of relevant literature and previous work in the field to provide context and justify the project's approach.
- **Methodology:** Description of the methods and tools used to develop the project, including software development methodologies, programming languages, frameworks, and technologies.
- **System Design:** Detailed documentation of the system architecture, including diagrams, data flow, components, and interfaces.
- **Implementation:** Description of the implementation process, including coding practices, algorithms used, database design, and testing methodologies.
- **Conclusion:** Summary of the project's achievements, contributions to the field, and future directions.
- **References:** A list of all sources cited in the document, following a specific citation style (e.g., APA, IEEE).
- **Appendices:** Additional materials such as code snippets, user manuals, screenshots, or data sets that support the main document.

6. Time frame and Submission Process

Date	Milestone
As provided on the lms	Submission of ER Diagram and Use case Diagram. ex : 103_ERD_<Student Reg No>_<First Name>
As provided on the lms	Submission of Completed Document. ex : 103_PROJECT_DOC_<Student Reg No>_<First Name>
As provided on the lms	Submission of Completed Project. ex : 103_PROJECT _<Student Reg No>_<First Name>.zip

7. Sample ER Diagram