# Assignment 4 – Project Demo 4 [25%]

This assignment relates to the following Course Learning Requirements:

CLR 1: Work effectively, individually and in a team, to solve problems and deliver a reasonable software product to a client.

CLR 2: Analyze, develop, test, and deliver a software product using the agile software development methodology.

CLR 3: Communicate effectively with a client to determine system requirements and to shape software to meet those needs.

CLR 4: Present prototypes and the final product to the client.

CLR 6: Include discussion of sustainable practices relating to software design, implementation, testing and maintenance.

CLR 7: Describe how experiential learning informs the professional development process.

Objective of this Assignment:

Demo part 4 marks the culmination of the project, where the final coding and implementation activities take place. This phase is crucial for completing the development process, optimizing code, and ensuring the project aligns with the proposed solutions and user requirements.

1. **Code Finalization:**
   * Complete the coding phase and ensure all features are implemented and functional.
   * Conduct thorough testing to identify and resolve any remaining issues.
2. **Optimization:**
   * Optimize code for performance, scalability, and efficiency.
   * Address any technical debt accumulated during the development process.
3. **Documentation and Reporting:**
   * Provide a final update to project documentation, including any changes made during the final coding phase.
   * Generate reports summarizing the project's development process, challenges faced, and solutions implemented.

**Assignment Tasks:**

**Part 1: Code Development**

In Task 1 of Demo 4, we focus on finalizing the development of the project. The goal is to complete the coding phase, ensuring that all proposed solutions and features outlined in the design are transformed into fully functional and optimized code.

1. **Code Finalization:**
   * Complete the coding phase by ensuring that all proposed features are fully implemented.
   * Address any outstanding issues or incomplete functionalities.
2. **Optimization:**
   * Optimize the codebase for performance, scalability, and efficiency.
   * Identify and eliminate any bottlenecks or areas where code improvements can be made.
3. **Documentation Update:**
   * Provide a final update to project documentation, including any changes made during the final coding phase.
   * Generate report summarizing the overall development process, challenges faced, and solutions implemented.
   * Include insights into the project's evolution from the initial design to the final implementation.
   * Reflect on the overall journey, highlighting achievements and lessons learned.

**Submission and Reporting Requirements for the Part 1:**

**Code Submission:**

* **File Format:** Provide the developed code files in a compressed ZIP format.
* **Organization:** Ensure the code files are organized logically, with a clear directory structure.
* **Documentation:** Include a README file within the ZIP archive that provides an overview of the code structure and instructions for running the application.

**Project Report:**

* **Content:** The project report should encompass all relevant information about the code developed in this part.
* **Feature Coverage:** Clearly specify which features have been implemented in this phase.
* **Challenges and Solutions:** Document any challenges encountered during coding and implementation. Describe the solutions or workarounds implemented**.**

**Architecture Changes:**

* **Updates to Diagrams:** If there are any changes to the architectural diagrams, UML diagrams, or other design documents, provide an updated version.
* **Justification:** Clearly explain the reasons behind any alterations to the initial design or architecture.

**Version Control:**

* **Include Commit History:** If using version control systems (e.g., Git), include a link to the commit history.
* **Commit Messages:** Ensure each commit message is descriptive, indicating the purpose of the changes made.

**Part 2: Video Presentation**

**Objective:**

Produce a recorded video presentation (10-15minutes) that effectively communicates the project's concept, includes a demonstration, and highlights individual contributions from each group member.

**Video Presentation Guidelines:**

1. **Content:**
   * Provide an overview of the project concept, its objectives, and key functionalities.
   * Include a demonstration of the project reports, Task 1, showcasing its main features.
   * Each group member should discuss their specific contributions to the project.
2. **Duration:**
   * The video presentation should be between 10-15 minutes in length.
3. **Recording Format:**
   * Record the video in a clear and audible manner, ensuring both video and audio components are of high quality.
4. **Individual Contributions:**
   * Each group member must contribute to the presentation. Discuss your role, tasks, and any challenges you encountered.
5. **Submission Method:**
   * To record the video, consider using a platform like Zoom, record group meeting and upload the recording as part of your submission.
6. **Clarity and Professionalism:**
   * Ensure clarity in communication and present in a professional manner.

**Submission:**

2 submissions:

1. **Word Doc/PDF of code, report, architecture changes, version control**
2. **Video recording of presentation**

Provide a descriptive report for each of above task. Each report should be in range of 1200-1600 words in length. Your project report is to be submitted in MS WORD or PDF format. It should be submitted with the following guidelines:

* 1. Include a cover page with the course code and course name, the project number, the title of your project, your student ID, your name, the instructor’s name, and the date the assignment is created. There must be one submission per group and each member should list their contributions.
  2. Ensure the first page of your assignment has the title at the top of the page and the sections of your report have headings and subheadings to chunk your paper into sections for each question you are writing about.
  3. Font should be Arial 12 or Calibri 12. Text must be double spaced, but table may be single spaced.
  4. You must cite all your sources of information using APA formatting.
  5. Create a separate reference page that lists all of your sources that you have cited in text. Sources include software used. Also, personal communications from a professional in the field count as a reference source. For citation and referencing examples, see https://owl.purdue.edu/owl/research\_and\_citation/apa\_style/apa\_formatting\_and\_style\_guide/general\_format.html

**Project Grading Rubric (25%)**

| **Criteria** | **Excellent**  **80-100%** | **Good**  **50-79%** | **Requires Improvement**  **<50%** | **Points** |
| --- | --- | --- | --- | --- |
| **Part 1: Code** | * Successfully completes the full feature set promised in the designated part of the project. Demonstrates a comprehensive understanding of the project's scope and successfully implements all planned features. * Rigorously tests each implemented feature, effectively identifying and resolving issues early in the development process. * Features developed align seamlessly with the proposed and agreed-upon design and requirement document. * Code development is closely aligned with the proposed and agreed-upon design and requirement documents. * Demonstrates a clear understanding of how the code fulfills the outlined specifications. * Code is of high quality, demonstrating efficiency, maintainability, and readability. * Well-documented code with meaningful comments and clear variable/method naming conventions. | * Implements a substantial portion of the promised features but may have minor omissions or areas that require further attention. Shows competency in initiating and completing feature implementation. * Conducts testing, catching, and resolving most issues during the development process. * Features developed generally align with the proposed and agreed-upon design and requirement documents but may have some room for improvement. * Code development generally aligns with the proposed and agreed-upon design and requirement documents. * Demonstrates an overall understanding of how the code fulfills the outlined specifications. * Code development generally aligns with the proposed and agreed-upon design and requirement documents. * Demonstrates an overall understanding of how the code fulfills the outlined specifications. | * Fails to complete a significant portion of the promised features. Demonstrates challenges in systematic initiation and completion of feature implementation. * Testing and issue resolution during the development process are inadequate. * Features developed may not fully align with the proposed and agreed-upon design and requirement documents. * Implements features in a disorderly or incomplete manner, missing critical or foundational components. * Testing and issue resolution during the development process are inadequate. * Features developed may not fully align with the proposed and agreed-upon design and requirement documents. * Implements features in a disorderly or incomplete manner, missing critical or foundational components. * Testing and issue resolution during the development process are inadequate. * Features developed may not fully align with the proposed and agreed-upon design and requirement documents. | /10 |
| **Comments** |  |  |  |  |
| **Part 1: Report** | * The project report is comprehensive and includes all relevant information about the code developed in this part. * Provides a thorough overview of the implemented features and their functionality. * Clearly specifies which features have been implemented in this phase. * Details how each implemented feature contributes to the overall project objectives. | * The project report is mostly comprehensive, covering most relevant information about the code developed in this part. * Provides a reasonable overview of the implemented features and their functionality. * The project report is mostly comprehensive, covering most relevant information about the code developed in this part. * Provides a reasonable overview of the implemented features and their functionality. | * The project report lacks comprehensiveness and may miss important information about the code developed in this part. * Offers limited insight into the implemented features and their functionality. * Fails to clearly specify which features have been implemented in this phase. * Lacks a clear connection between implemented features and overall project objectives. | /10 |
| **Comments** |  |  |  |  |
| **Part 2: Presentation** | * The video presentation is well-structured, engaging, and effectively communicates the project concept. * The demonstration is comprehensive, showcasing all key features. * Each group member articulates their contributions with clarity and enthusiasm. | * The video presentation is adequately structured and communicates the project concept. * The demonstration covers most key features but may lack depth. * Group members discuss their contributions with clarity, but some details may be missing. | * The video presentation lacks structure and fails to effectively communicate the project concept. * The demonstration is incomplete or unclear. * Group members struggle to articulate their contributions, and details are lacking. | /5 |
| **Comments** |  |  |  |  |
| **Total Points** |  |  |  | /25 |