CS-2 PROJECT

YEARLONG WALKTHROUGH

D. P. Raagas

OVERVIEW

- This school year, you will undertake a yearlong programming project that allows you to design, build, and refine your own software application.
- Work in pairs or individual.
- For Quarter 1, the focus is on understanding the problem, planning a possible solution, and documenting initial ideas.

SCOPE

Your project must reflect your understanding of key programming concepts, including but not limited to:

1. Programming Fundamentals: data types, input/output, conditionals, loops, functions
2. Data Structures and File Handling: arrays/lists, dictionaries/objects, JSON
3. Modularization and Collaboration: reusable code, GitHub version control
4. Data Validation, Privacy, and Security: input/output verification, data privacy in algorithms, basic cybersecurity threats and mitigation
5. Programming Ethics and Principles

PROJECT CYCLE PER QUARTER

- Q1 - Project Proposal (Conceptualization) - First Draft
- Q2 - Project Proposal (Conceptualization) - Final Draft
- Q3 - Initial Development and Core Feature Implementation
- Q4 - Full Feature Implementation, Testing, and UI Design, Finalization, Documentation, and Presentation

SCHEDULE

- In Quarter 1 (Weeks 1–7), the students' task is to conceptualize his/her project and submit a first-draft of his/her Project Proposal via a GitHub repository. No actual coding is required yet.

SCHEDULE – QUARTER 1

- Weeks 1 to 2 Introduction to the Yearlong Project; explore sample topics
- Week 3 Brainstorm and select a topic
- Week 4 Define problem, users, and objectives
- Week 5 Plan features, inputs/outputs, Draft proposal and set up GitHub repository
- Week 6 Finalize proposal and review feedback
- Week 7 Submit proposal via GitHub (deadline)

EXPECTATION

What You Need to Submit:

1. Project Proposal Document (as text - uploaded in Github)
2. GitHub Repository with initial structure

DOCUMENTATION REQUIREMENT

Document (Initial Draft) must include:

- Project Title
  - A tentative yet meaningful title that creatively reflects the core problem the project seeks to address. This may evolve as the project develops.

- Problem Statement
  - An initial articulation of the real-world issue the project aims to solve. This draft version should clearly explain the context and relevance of the problem, with room for refinement later.

DOCUMENTATION REQUIREMENT

- Project Objectives
  - A preliminary set of specific, realistic goals that guide how the project plans to address the identified problem. These objectives may be adjusted as the project becomes more defined.

- Planned Features
  - An early list of key functionalities the project is expected to include. These features are meant to tackle the problem and may be revised as the design becomes clearer.

DOCUMENTATION REQUIREMENT

- Planned Inputs and Outputs
- A draft description of what data or interaction the user will provide (inputs), and what the system is expected to generate or do in return (outputs). These elements may be refined as development progresses.

- Logic Plan:
- Pseudocode or Flowchart to describe the expected program flow.

DOCUMENTATION REQUIREMENT

- GitHub Repository Link (The link to the public GitHub repository containing the draft proposal, README, and initial commits.)

SAMPLE PROJECT IDEAS

You may create your own original project. These samples are for inspiration.

- Data Insights Dashboard (Data Provided): Analyze, sort, and summarize a JSON dataset.
- Personal Expense Tracker: Track and categorize expenses, generate summaries.
- Student Grades Manager: Input grades, compute averages, print report cards.
- Quiz Game Engine: Loads questions from JSON, computes scores.
- Career Recommender: Suggest careers based on user interest mapping.
- Weekly Task Organizer: Sort and manage tasks by category.

SAMPLE PROBLEM STATEMENT

You may create your own original project. These samples are for inspiration.

- Title: Weekly Task Organizer
- Problem statement:
- Everyday people are faced with different demands at work or in school. These demands may entail several tasks to get one job done. Sometimes in a day people have to work on more than 1 jobs. The problem crops up in scheduling the tasks in multiple jobs. Essentially, a weekly organizer is one tool that can help optimize and expedite daily work. Our project aims to develop a weekly organizer that can help individuals maximize productivity.

RUBRICS

- The project title is clear, creative, and relevant to the problem (10%)
- The problem statement is clearly defined and well-explained (30%)
- The project objectives are specific, realistic, and aligned with the problem (30%)
- The proposed features are logical and appropriate for solving the problem (20%)
- The GitHub repository is properly set up (README, initial structure, commits) (10%)