“Solve It with Code”: Project Guidelines

Project Overview

This project challenges you to **identify a real-world problem from your daily life** and design a **computational solution** using programming principles. Your final output will include an **explanation of the problem**, a **detailed algorithm design**, and a **presentation of your solution**.

G - Goal

Your goal is to **apply computational thinking** to solve a real-life problem using programming. Your solution should be **logical, structured, and feasible** within the given constraints.

- Clearly define the problem.
- Design an algorithmic solution.
- Represent the solution through **flowcharts and pseudocode**.
- Effectively communicate the process and impact of your solution.

R - Role

You are acting as a **problem solver, developer, and presenter**.

- Work individually or in pairs to research and design a logical solution.
- Use programming logic and computational thinking to break down the problem.
- Be prepared to **justify your choices and approach** in your final presentation.

A - Audience

Your **target audience** includes **teachers, peers, and potentially the users of your solution**. Your final presentation should be clear and accessible to someone unfamiliar with the problem.

S - Situation

- The problem must be **authentic**, meaning it should come from **your daily life, school, or community**.
- Your solution should be something that **a program, algorithm, or software** can realistically address.
- Think about challenges in areas such as **time management, studying, organization, daily routines, or school-related tasks**.

P - Product

Your final submission will include:

1. **Problem Identification**
    - Describe the real-world problem you are solving.
    - Explain why it is important and who it affects.
2. **Solution Design**

- **Flowchart**: Visual representation of your solution.
- **Pseudocode**: A structured step-by-step breakdown of how your program would work.
- **Logic Explanation**: A description of the decision-making process within your solution.

3. **Presentation Board**
   - Summary of your problem and solution.
   - Visual aids (flowcharts, diagrams, pseudocode snippets).
   - Explanation of how your solution is **practical and effective**.
4. **(Optional) Prototype/Demo**
   - If feasible, a basic working **prototype** of your solution in a programming language of your choice.

GRASP Framework

(To be individually filled out as your role is different from each other).

- G
- R
- A
- S
- P

Assessment Criteria (Rubric)

Criteria	Exceeding Expectations (3 points)	Meeting Expectations (2 points)	Approaching Expectations (1 point)
Problem Identification	Clearly defines a **real, relevant** problem with strong justification.	Identifies a **clear problem**, but lacks some depth in explanation.	Problem is **vague or unrealistic**.
Solution Design	Flowchart and pseudocode are **detailed, logical, and correctly structured**.	Flowchart and pseudocode are **mostly correct but need refinement**.	Flowchart and pseudocode are **incomplete or unclear**.
Logic Explanation	Clearly explains **how the solution works**, with well-thought-out logic.	Provides a **basic** explanation of how the logic works.	Minimal explanation of how the algorithm works.
Self-Regulation	The project follows a **clear plan with success criteria** and incorporates **feedback effectively**.	The project has a **plan and success criteria**, but does **not** incorporate feedback.	The project lacks **clear planning, success criteria, or feedback integration**.
Collaboration	Team shares responsibility equally and makes **substantive decisions together**, showing **strong interdependence**.	Team shares responsibility but lacks **clear interdependence**.	Team attempts collaboration but **work is not well distributed**.
Presentation & Communication	The project is communicated in a **highly effective and engaging manner**.	The project is **clearly presented but lacks engagement**.	Presentation lacks clarity, organization, or engagement.
Creativity & Feasibility	Solution is **innovative, realistic, and has clear applicability**.	Solution is **feasible**, though **not highly innovative**.	Solution lacks **realistic feasibility or development**.

Timeline & Milestones

- **Block 1:** Identify and define your problem.
- **Block 2:** Design your algorithm (flowchart & pseudocode).
- **Block 3:** Design your algorithm (flowchart & pseudocode).
- **Block 4:** Develop your presentation board and refine your explanation.
- **Block 5:** Present your project to the class.

This project will be completed **within one full cycle (5 blocks) during class time** and must be finished before **April**.

Final Notes

- This is a **thinking-based project**, not just coding. Even if you don't write actual code, you must demonstrate **logical problem-solving**.
- Work **collaboratively if in pairs**, ensuring both members contribute.
- Be ready to **answer questions** about your solution in your presentation.

Let's start solving real-world problems with computational thinking!