[Student name]

[Batch]

[Date]

PROJECT PROPOSAL

[Project Name]

Table of content

1. Summary

Provide a brief overview of the project for your instructor or reviewer. Write this section last, keeping it concise (no more than one page). Include:

- The purpose of your software project (e.g., to develop a web app, mobile app, or tool).
- The main deliverable (e.g., a working prototype, source code, documentation).
- How the project aligns with course objectives or personal learning goals.
- Key benefits (e.g., solving a specific problem, gaining experience with a technology).

2. Introduction

Introduce the context of your project. Include:

- Why you chose this project: motivation (e.g., interest in a specific technology, addressing a real-world issue).

- The target users or audience for your software (e.g., students, hobbyists, a specific community).
- How the project supports your academic or career goals.

3. Problem Statement

Describe the problem your project addresses. Include:

- The specific issue or need (e.g., lack of an efficient tool, a gap in existing software).
- Who is affected by this problem (e.g., target users or stakeholders).
- Evidence of the problem (e.g., personal experience, research, or user feedback).
- Why this problem is worth solving for your final project.

4. Goals and Objectives

List specific, measurable goals for your project. Examples:

- Develop a [web/mobile/desktop] application using [language/framework].
- Implement [specific feature, e.g., user login system] by [milestone date].

- Ensure the application meets [specific requirement, e.g., usability or performance metric].

5. Technical Requirements

Specify the technical components of your project. Include:

- Programming language(s) and framework(s) (e.g., Python with Flask, JavaScript with React).
- Development tools (e.g., IDEs, version control like Git).
- External dependencies (e.g., APIs, libraries, or databases).

6. Timetable

Provide a schedule for your project, broken into phases. Example:

Phase	Description	Start Date	End Date

7. Resources

List resources you will need. For a student project, include:

- Software: free/open-source tools or specific licenses required.
- Time commitment: estimated hours for each phase.
- Support: instructor guidance, peer feedback, or access to tutorials/documentation.

8. Appendix

Include optional supporting materials, such as:

- Wireframes or mockups of the user interface.
- Pseudocode or algorithms for key functionality.
- References or research supporting your problem statement.
- Preliminary code snippets or proof-of-concept work.