Project Initial Proposal Submission:

Title:

# **Student Task Manager**

## Project Description.

Students often miss deadlines because assignments, exams, and projects are spread across many different places like emails, syllabi, and notes. Our project, the **Student Task Manager**, solves this by giving one simple desktop app where students can add tasks, set due dates, mark them complete, and see everything in a clean list or calendar view. The app will be built with Java, JavaFX, and Scene Builder inside IntelliJ, so it runs locally, works offline, and is easy to demonstrate.

The main version of the project will focus on the basics: creating, editing, deleting, and completing tasks, with data stored in a local SQLite database. Users will also see a simple calendar view to keep track of upcoming deadlines and get small reminders when tasks are due. Keeping the scope small makes the app realistic to finish in one semester and ensures our team can deliver a working and reliable product.

If time allows, we are ready to extend the system with extra features such as email login, cloud storage, CSV export, or external APIs (like Google Calendar integration). These will only be added if the base system is completed and tested. This shows that the design is flexible, but our first goal is a working core version that can be built and presented within the semester.

## WRSPM Analysis

## World Assumptions

* Students have laptops with Java installed.
* Students can run a desktop app and save data locally.
* The app will store tasks in a local database on the same machine.
* Students want a simple tool, not something with too many complicated features.

## User Requirements

* Create an account (basic login for single user).
* Add a task with title and due date.
* Edit or delete a task.
* Mark a task as complete or not complete.
* See all tasks in a list view.
* View tasks in a simple calendar style view.
* Get a reminder inside the app on the due date when the app is open.
* Data saves automatically and reloads when the app opens.

## Specifications and Interface Needs

* JavaFX user interface designed with Scene Builder.
* Main screen shows a task list with add, edit, delete, and complete buttons.
* Add/Edit form includes title, due date, optional notes, and priority.
* Calendar view shows days with tasks.
* Input validation (task title cannot be empty, due date must be valid).
* Clear success and error messages.
* Works on standard laptop screen size.

## Program and Hardware

* Language: Java 17
* UI: JavaFX with FXML (Scene Builder)
* IDE: IntelliJ IDEA
* Database: SQLite (local file, for example tasks.db)
* Build tool: Maven or Gradle with JavaFX and SQLite JDBC libraries
* Architecture: Simple MVC (Model, View, Controller)
* Notifications: In-app alerts (no email required for core version)
* Target OS: Windows and macOS laptops

## Team Meetings

* Meeting times: (example): Not decided yet.
* Meeting method: Zoom and In-Person.