Task W03

Open-Source Course Task Overview

This semester, many task assignments will center on a shared, student-driven open-source software project involving students from CIS 3970 and CIS 4000. Together, you will define a project vision, establish a public GitHub repository, and collaboratively build the project throughout the semester, and possibly beyond. The goal is to create something long-lived and reusable, where current and future students can continue contributing, and be a legacy of what NSU CIS students can achieve.

The group will choose the project idea. A game was the most popular suggestion among in-person students (like RuneScape's development), but the group makes the final decision collectively. Because this project is for coursework, it must include Java and database components, but you are encouraged to add other technologies as needed. Web front ends, additional languages, tooling, artwork, security features, and more are possibilities. This is not a traditional group project. Think of it as a real open-source project with contributors.

Organizing as a Team

One required kickoff meeting will be held before this assignment is due. I ask everyone in to remain flexible in scheduling this meeting. The in-person students know that everyone has different schedules and other responsibilities, and there are currently 11 people in these classes, so scheduling will be difficult, but achievable, and everyone should respect time requirements. After that, communication will be based on what the group decides.

Purpose of the Meeting:

- Introductions
- Establishing a shared vision
- Discussing strengths and interests
- Selecting an open-source license
- Discuss what is needed
- Potentially assigning initial tasks (e.g., creating the GitHub repository)
- After this meeting, I am not requiring additional group meetings unless extraordinary circumstances arise. Any meetings will be at the group's discretion.
- Communication channels are up to the group (Discord, Teams, group chat, etc.).
- Collaboration can happen through contributions, not necessarily meetings.

Project Leadership & Workflow

- Graduate students will manage the project:
    - Justin Dysarz (Senior Manager)
    - Brady Benoit (Junior Manager)

- Their role is to:
    - Contribute to the project
    - Manage merges and branches
    - Help me track progress and contributions
    - Report issues and project status to me
    - Help me make decisions about the project

All students need to learn how GitHub workflows operate (branching, merging, forking, pull requests). Justin and Brady will handle official merges, guided by team input.

What Counts as a Contribution?

There is a place for every skill set:

- Coding features
- Designing or improving the database
- Fixing bugs
- Improving security
- Creating artwork or assets
- Writing documentation
- Building a website
- Reviewing or testing code
- And others…

Some contributions may be rejected or revised; that is normal and part of open-source development. A rejected contribution is still considered a contribution. If you need help figuring out what you want to contribute, please ask the team. Team members can form subgroups to work on contributions.

Professionalism is required. Be respectful, open to feedback, and drama-free. Internal disputes may require conflict resolution, but remember that the project's success depends on collaboration and mutual support. Just like work environments, if personal issues arise, you need to communicate with your team and managers. “Disappearing” or ignoring the project can lead to being laid off (i.e., fired), which may make it impossible to pass the course.

Grading Structure

Task assignments may fall into one of the following formats. You will need to read any instructions I provide to know the requirements for the entire assignment.

- Mixed Mode
    - Part 1: Linux/tools or course-related work (graded objectively)
        - Most Linux/tools will be tutorial-oriented for introductory quick learning

        - Course-related work would be typical assignments you would expect in a course, but most likely still application-oriented
    - Part 2: Description and listing of your contribution to the shared project
        - Will not require formal report writing—a description and bullet listing of your contributions is likely sufficient
- Backed by GitHub activity and manager reports
    - Graded on effort, impact, learning, teamwork, amount, and team opinions, but I do not know the details until we make progress
    - Contributions vary in size and form; the key is meaningful progress
- Single Mode – Topic-Based
    - One focused task covering the course material
    - Mostly application-oriented and graded objectively
- Single Mode – Shared Project
    - Reporting and reflecting on your contributions to the open-source project

The Big Picture

The team meets once, decides what to build, makes it public, and spends the semester building something real together. You will apply course concepts in an authentic setting, gain experience with open-source workflows, and leave behind a project others can see, use, and extend.