

Main

Folio: Student Academic Planning with Task-Completion Encouragement
First Semester

Project Team:

Nicholas Dunzelman (ndunzelman2023@fit.edu)

Caleb Brooks (cbrooks2022@fit.edu)

Faculty Advisor:

David Luginbuhl (dluginbuhl@fit.edu)

Client/User:

Jackson Neering (Student at Florida Institute of Technology)

Date(s) of meeting(s) with the advisor and client for developing this plan:

Nicholas Dunzelman met with Dr. Luginbuhl on August 25, 2026. The team had a discussion with the potential user on August 28, 2026.

Goal and Motivation

Our goal with Folio is to help students who are struggling or falling behind to turn their course information into a manageable plan and follow through on it. Existing student feedback identified late awareness, stress, a lack of advice, and inconsistent use of Canvas as common problems. In the August 28 discussion, the potential user described the hardest part of getting back on track as the dread of catching up and consuming course material as deadlines approach. He said a self-imposed reward system could motivate him to complete his work. After recognizing the problem, students may still be unsure of the next step or how to follow through.

Canvas and similar systems show grades, deadlines, missing work, and course content. Still, they do not provide a student-owned workflow for choosing the work to address, creating and revising an improvement plan, scheduling tasks from multiple active plans into one schedule, and optionally sharing information from their plan with an instructor for suggestions. Folio addresses this while keeping the student in control of the plan and what information is shared.

Approach (Key Features of the System)

The user can manually enter or import course schedules, assessments, deadlines, and scores, then choose which assignment, test, or other work to address. This course information is used as a planning context to help the student choose and schedule their next actions. Folio does not use the information to infer studying, attendance, participation, or effort outside of the system.

The user can choose a planning template, edit suggested tasks, and schedule tasks from active plans into a daily or weekly schedule. The user can export a plan as a PDF or DOCX, review a prepared email before sending it to an instructor, then accept, reject, or adapt the instructor's suggestions, and decide whether to revise the plan. The student remains the owner of the plan, and instructor involvement is optional and limited to information the student deliberately chooses to share.

The user can view scheduled tasks, reschedule tasks, receive configurable reminders, mark tasks complete, see a progress bar for each plan, and receive congratulations on task completion. The user can also play a simple one-player tic-tac-toe game tied to task completion. Completing one task earns a move; otherwise, the user loses the turn, and the computer moves. A new game starts after each game ends, the player is randomly selected, and wins, losses, and ties are tracked. The user can earn XP, points, levels, badges, and cosmetic rewards separately. The planning features remain usable without the game or rewards.

Novel Features/Functionalities

Folio's proposed novelty is the combination of student-owned academic planning, cross-plan scheduling, student-controlled instructor feedback, plan revision, and task-completion encouragement into a single workflow. Tasks from multiple active plans feed into a single schedule, while the student can export a plan and request instructor suggestions without needing a professor dashboard or exposing private progress. The task-completion-based tic-tac-toe game and separate XP, points, levels, badges, and cosmetic rewards encourage, while remaining distinct from the core planning workflow. The individual ideas of calendars, reminders, task lists, or rewards are not claimed as novel by themselves.

Algorithms and Tools

- Streamlit and Python for web application, planning interface, scheduling, reminders, progress, and game and reward logic.
- Pandas for manual/CSV course data validation and SQLite for local data storage.
- Scheduling and conflict detection logic for combining tasks from active plans without overriding student choices.
- Playwright for PDF export, Python-docx for DOCX export, and Python email tools for preparing instructor emails for student review.
- Tic-tac-toe game state and turn rules for earned and lost turns, computer moves, random starting player, new games, and win/loss/tie tracking. The computer-move algorithm will be compared and selected during Milestone 1.
- Python rules for XP, points, levels, badges, progress bars, and congratulatory messages.
- GitHub for software development collaboration.

Technical Challenges

- Data quality and scheduling: validate manual fields, preserve source labels, handle duplicate or conflicting schedule data, and resolve task collisions across active plans without overriding student choices.
- Privacy and export: keep private progress, rewards, and game information separate from academic information the student chooses to share, and allow the student to review exported plans and prepared emails before sharing them with an instructor.

- Game and reward correctness: prevent duplicate earned tic-tac-toe moves, handle lost turns, randomly select the starting player, track legal moves and win/loss/tie states, and keep XP, points, levels, badges, cosmetics, and congratulations separate from the core planning workflow.

Milestone 1 (Sep 28)

1. Compare and select technical tools for planning and scheduling, manual/CSV data handling and storage, PDF/DOCX/email export, and tic-tac-toe and rewards.
2. Provide “hello world” demos for the planning and scheduling interface, CSV to database workflow, PDF/DOCX and email preparation, and tic-tac-toe.
3. Resolve technical challenges involving data quality and schedule conflicts, privacy and export, and game and reward correctness.
4. Compare and select collaboration tools for development, documents and presentations, communication, and task calendar.
5. Create Requirement Document.
6. Create Design Document.
7. Create Test Plan.

Milestone 2 (Oct 26)

1. Implement, test, and demo manual/CSV entry of course schedules, assessments, deadlines, and scores; selection of the assignment, test, or coursework to address; input validation; storage.
2. Implement, test, and demo: planning template selection; editing suggested tasks; combining tasks from active plans into a single daily or weekly schedule; PDF and DOCX export; prepared instructor email; and student-controlled plan revision after feedback.

Milestone 3 (Nov 23)

1. Implement, test, and demo daily and weekly task views, rescheduling, configurable reminders, task completion, plan progress bars, and completion congratulations.
2. Implement, test, and demo tic-tac-toe and XP, points, levels, badges, and cosmetic rewards; integrate these features with the complete version-1 workflow while keeping the planning functions usable independently of the game and rewards.

Task matrix for Milestone 1

Task	Nicholas Dunzelman	Caleb Brooks
Compare and Select Technical Tools		Planning and scheduling, manual/CSV data and storage, PDF/DOCX/email export, tic-tac-toe, and rewards

"Hello World" Demos	PDF/DOCX and email preparation	Planning and scheduling interface, CSV to database workflow, tic-tac-toe
Resolve Technical Challenges	Privacy and export	Data quality and scheduling, game and reward correctness
Compare and Select Collaboration Tools	Software documentation and presentations, communication, and task calendar	Software development
Requirement Document	Write 50%	Write 50%
Design Document	Write 50%	Write 50%
Test Plan	Write 50%	Write 50%

Approval from Faculty Advisor

I have discussed with the team and approve this project plan. I will evaluate the progress and assign a grade for each of the three milestones.

Signature: _____ Date: _____