WILFRID LAURIER UNIVERSITY
LAURIER
Inspiring Lives.

Project Outlines
CP476A - Internet Computing (Winter 2026)
by Dr. Mustafa Daraghmeh

Group Project: Full-Stack Web Application
Weight: 15% of final grade (5% per milestone)
Team Size: Up to 3 students
Submission (all milestones): MyLS → Dropbox → Project Milestone 01 / 02 / 03 (Final)

Project Description and Purpose

The group project serves as the **capstone experience** for CP476A - Internet Computing. Through this project, students will integrate the core concepts learned throughout the course to design, implement, test, and document a **full-stack web application**. The project emphasizes not only technical implementation, but also **software design thinking, planning, teamwork, documentation, and accountability**.

Each team will build a web application that combines:

- a functional **front-end** interface,
- **server-side processing**,
- and a **relational database** backend.

The application must support a **clear and complete user workflow** (for example, creating, viewing, updating, and deleting data) and demonstrate appropriate **input validation** and **basic security hygiene** consistent with material covered in the course.

Students are **free to choose the project topic** they wish to develop. However, all projects are subject to instructor approval to ensure feasibility, appropriate scope, and alignment with course learning outcomes.

Tools, Resources, and Constraints

Students are expected to build their project using the **technologies and approaches covered in CP476A**.

The use of **external libraries, frameworks, tools, APIs, or templates that are not covered in the course is not permitted unless explicitly approved by the instructor in advance**.

Students may consult reputable documentation and learning resources to understand syntax and concepts; however, the project code itself must be written by the students and must rely on **course-covered technologies** unless approval has been granted.

If a team is unsure whether a tool or resource is permitted, they must ask for clarification **before using it**. Approval must be obtained in writing via email.

GitHub, Project Management, and Accountability

Each group must create and maintain a **GitHub repository** (or GitHub Organization repository) for their project. GitHub will be used not only for version control, but also as a primary mechanism for **project management and individual accountability**.

GitHub Projects (Kanban Board)

Teams must use **GitHub Projects (Kanban)** to plan, track, and manage project work throughout the term.
At a minimum, the Kanban board must include the following columns:

- Backlog
- Ready
- In Progress
- In Review
- Done

Tasks should move through these columns as work progresses. Each task should clearly indicate **what needs to be done**, **who is responsible**, and **when it is complete**. The board must be actively maintained across all milestones.

Activity Blog / Wiki

Each group must maintain an **activity blog** using either the GitHub Wiki or a dedicated Markdown file within the repository. The activity blog documents the **process** of the project, not just the final result.
Entries must include:

- minutes of weekly (or regular) meetings,
- task assignments,
- progress updates,
- important design and architecture decisions,
- challenges encountered and how they were addressed.

This documentation supports transparency, teamwork evaluation, and individual contribution assessment.

Milestone-Based Project Structure

The project is divided into **three milestones**, each worth **5%** of the final course grade.
Each milestone is graded **out of 100**, and the score is then converted to 5%.
All milestones must be submitted via **MyLS → Dropbox** under the corresponding milestone folder.

Individual Contribution and Fair Assessment (applies to all milestones)

Although the project is completed in groups, **individual contributions matter**.
Individual grades may be adjusted based on evidence such as:

- GitHub commit history (quality + consistency, not only quantity),
- task ownership and movement in the Kanban board,
- and documentation in the activity blog/wiki showing assignments and completed responsibilities.

Submission Packaging (applies to all milestones)

For each MyLS dropbox submission:

- **A single ZIP** containing (**LastName_FirstName_ID__M#.zip**):

- A PDF report for that milestone (or the required PDFs)
- A links.txt with:
  - GitHub repo link
  - GitHub Project board link
  - Wiki/docs link
- For Milestone 03: include a demo video file **or** a link for the demo video

Milestone 01 – Project Planning and Design

Due: Friday, January 30, 11:59 PM

Submission Folder: Project Milestone 01

Purpose of Milestone 01

Milestone 01 focuses on **planning and design**. At this stage, students are not expected to implement a working system. Instead, they must demonstrate that they have a **clear, feasible, and well-thought-out plan** for the application they intend to build.

This milestone ensures that:

- the project idea is realistic,
- requirements are clearly defined,
- the user workflow is understood,
- and the team is organized and ready to begin development.

Deliverables (what to submit)

A single PDF (or well-organized set of PDFs) **plus** GitHub links (repo + Project board + wiki/docs):

1. **Project proposal (1–2 pages)**
   - Problem statement, motivation, target users
   - App concept and scope (what is *in* / *out*)
   - **Features list** (**Must Have**: Essential for product viability; **Should Have**: Very important, adds significant value, but product can launch without it; **Could Have**: Nice additions, low impact if omitted.)
2. **Requirements (user stories)**
   - **Minimum: 5 user stories**
   - Each story includes: *role, goal, value* and **clear acceptance criteria**
3. **Wireframes**
   - Wireframes for all key screens in the primary workflow (e.g., list/create/edit/view)
   - Show navigation between screens
4. **Data planning (lightweight, but explicit)**
   - Identify key data entities (e.g., 2–4 main tables you expect)
   - Identify key relationships at a high level (no full ERD required yet)
5. **Team plan**
   - Roles (can rotate), communication method, meeting cadence
   - Definition of “Done” for tasks (e.g., code + tested + documented)
6. **GitHub setup evidence**

- Repo created with a clear README skeleton
- GitHub Projects Kanban with required columns and initial tasks created/assigned

7. **Activity blog/wiki initialized**
   - At least **one** planning entry: meeting minutes + initial task assignments

Milestone 01 Rubric (100 points → converted to 5%)

- **Idea clarity & feasibility (15 pts)**
  Clear problem, realistic scope for term timeline, coherent users/workflow.
- **User stories + acceptance criteria quality (25 pts)**
  Specific, testable acceptance criteria; covers end-to-end workflow.
- **Wireframes & workflow clarity (20 pts)**
  Key screens covered, navigation makes sense, supports user stories.
- **Data planning (10 pts)**
  Reasonable entities/relationships consistent with proposed features.
- **Team plan & execution readiness (10 pts)**
  Roles, cadence, communication, and definition of done.
- **GitHub repo + Kanban setup quality (15 pts)**
  Correct columns, tasks created, ownership shown, board usable.
- **Activity blog/wiki quality (5 pts)**
  Meeting notes + decisions + assignments recorded.

Milestone 02 – Front-End Implementation & Database Design

Due: Friday, February 27, 2026 – 11:59 PM

Submit: MyLS → Dropbox → **Project Milestone 02**

Purpose of Milestone 02

Milestone 02 focuses on **implementation progress**. By this stage, teams must demonstrate that their project has moved beyond planning into **actual development**.

This milestone evaluates whether students can:

- implement a functional user interface,
- design a correct and well-structured relational database,
- and set up the server-side foundation required for full integration later.

Deliverables (what to submit)

1. **Working front-end (in GitHub repo)**
   - Core screens implemented (matching Milestone 01 wireframes, updated if needed)
   - Primary workflow functional in the UI (even if still using mock data)
2. **Database design package (PDF)**
   - **ER diagram** (or clearly structured relational diagram)
   - SQL **CREATE TABLE** statements with:
     - primary keys, foreign keys (where relevant), constraints

        - sensible data types
3. **Back-end setup** (Node.js or PHP) **(in repo)**
    - Project skeleton and runnable server entry point
    - Initial routes/controllers structure (even if some endpoints are stubs)
4. **Updated GitHub Projects Kanban**
    - Active movement of tasks across columns, clear ownership, and meaningful status
5. **Updated activity blog/wiki**
    - At least **two** progress entries since Milestone 01: work completed, decisions made, blockers/resolutions
6. **README updated**
    - How to run front-end/back-end locally (basic steps)
    - Team member contributions summary (brief)

Milestone 02 Rubric (100 points → converted to 5%)

- **Front-end completeness & workflow (30 pts)**
  Core screens present; navigation and main interactions work.
- **Front-end quality (UI consistency + JS structure) (10 pts)**
  Readable code, consistent layout, meaningful form handling.
- **Database schema quality & normalization (25 pts)**
  Appropriate tables/relationships; constraints; avoids obvious redundancy.
- **SQL correctness & clarity (10 pts)**
  CREATE statements run cleanly; types/keys/constraints make sense.
- **Back-end setup progress (10 pts)**
  Runnable skeleton; clear route/controller organization.
- **Kanban usage & evidence of project tracking (10 pts)**

Activity blog/wiki quality (5 pts)

Milestone 03 – Full-Stack Integration, Testing Report, Final Demo & Presentation

Due: Friday, April 3, 2026 – 11:59 PM

Submit: MyLS → Dropbox → **Project Milestone 03 (Final)**

Purpose of Milestone 03

Milestone 03 represents the **final integrated system**. At this stage, the application must function end-to-end and be supported by testing evidence and clear documentation.

This milestone assesses students' ability to:

- integrate front-end, server, and database components,
- validate and test their system,
- and clearly communicate technical work through a professional demo.

Deliverables (what to submit)

1. **Fully functional full-stack application (GitHub repo)**
    - End-to-end workflow works against the database
    - **CRUD required** for core data objects (create/read/update/delete as appropriate)
2. **Testing summary report (PDF, recommended 2–5 pages)**
    - Test plan: key features + test cases
    - Results (pass/fail + notes)
    - Known issues/limitations (if any)
    - If you used automated tests, briefly explain how to run them
3. **Deployment/execution instructions (README)**
    - Clear steps to run locally from a clean machine setup
    - Include any environment/config notes
4. **Final GitHub Projects Kanban snapshot**
    - Board reflects completed work and remaining items (if any)
5. **Final activity blog/wiki reflection**
    - Summary of progress over the term, major decisions, and contribution evidence
6. **Final Demo video (max 7 minutes)**
   Must include (in this approximate order):
    - Project name + goal + target users (10–20 seconds)
    - Architecture overview (front-end/back-end / DB) (30–60 seconds)
    - Feature walkthrough (major workflow end-to-end)
    - Brief testing highlight (what you tested, one example)
    - Short closing: limitations + next steps
7. **Presentation artifact**
    - Slides (PDF) **or** a 1-page visual overview (architecture + features + responsibilities)

Milestone 03 Rubric (100 points → converted to 5%)

- **Full-stack functionality & completeness (35 pts)**
  End-to-end workflow works reliably; CRUD is implemented appropriately.
- **Code quality & maintainability (10 pts)**
  Clear structure, readable naming, minimal duplication, sensible separation of concerns.
- **Database integration quality (10 pts)**
  Correct queries, consistent data handling, constraints respected.
- **Input validation & basic security hygiene (10 pts)**
  Evidence of server-side validation and avoidance of common pitfalls (e.g., unsafe handling of input).
- **Testing report quality (15 pts)**
  Good coverage, clear results, honest limitations.
- **Demo video quality (15 pts)**
  Within 7 minutes; clear narration; shows core workflow and technical highlights.
- **Project tracking + documentation completeness (5 pts)**
  Kanban final state + blog/wiki + README are complete and consistent.