

CSC491H1 S

Capstone Design Project

Winter 2026 Syllabus

Lecture Time & Location

CSC491H1 S & CSC20261

Section	Day & Time	Delivery Mode & Location
All Sections	Thursday, 6:00 PM - 9:00 PM	In Person: SM 2360

Refer to ACORN for the most up-to-date information about the location of the course meetings.

Classes, Tutorials & Workshops will be held in the [Department of Computer Science Innovation Lab \(DCSIL\)](#) located on the 2nd floor in the Gerstein Science Information Center, 9 King's College Circle

Gerstein Science Information Center Hours of Operation

- M-Thurs 8:30am-11pm, Fri 8:30am-10pm, Sat 9am-10pm, Sun 10am-10pm – Except Holidays

For more up-to-date information, please refer to the [Library Calendar](#).

Course Contacts

Course Website: <https://q.utoronto.ca/courses/417516>

Instructor: Atoosa Nasiri

Email: atoosa.nasiri@mail.utoronto.ca

Office Hours and Location: Office hours are by appointment only. - To ensure you receive dedicated time and focused feedback, office hours are held strictly by appointment. - To Book: Email the teaching team at least 24 hours in advance with a brief summary of what you need to discuss.

Communication Policies

In a professional engineering environment, communication is not an afterthought—it is a requirement. In this course, we treat your outreach as a reflection of your professional readiness. You are expected to be clear, concise, and proactive. Primary Communication Channels:

- Email (Private Matters): For grades, personal emergencies, or sensitive team issues.
- In-Person (General): Brief questions before/after class or during designated breaks.
- GitHub Issues: All technical decisions, roadmapping, and project-related blockers must be documented here to maintain a clear audit trail.
- Quercus: Visit for official announcements and formal feedback.

Course Overview

In this course, students design and build a software application that implements the business idea developed in CSC454/2526 (The Business of Software). CSC491 focuses on turning validated ideas into **working products**, emphasizing MVP-driven development, iterative delivery, and decision-making under uncertainty.

Working within a theme defined in CSC454, students will take ownership of their product by:

- Defining and refining the problem scope
- Making deliberate design and technical trade-offs
- Developing and iterating on a functional minimum viable product (MVP)
- Adapting and pivoting based on feedback, constraints, and evidence

Students are expected to deliver progress in small, tangible, demo-ready increments and to communicate that progress clearly to stakeholders. Work will be shared through multiple formats, including written documentation, oral presentations, in-class demonstrations, and any other meaningful or innovative forms students choose to effectively convey their work and learning.

Class time prioritizes doing over listening. Rather than focusing on long lectures, fixed readings, or assessment based on deterministic content, the course emphasizes project-focused workshops, short targeted lectures, and guided discussions. The goal is to help students learn how to operate under uncertainty—making decisions with incomplete information, navigating ambiguity, and adapting as real-world constraints evolve—reflecting the realities of building software products and businesses.

Lastly, industry guest speakers may be invited to share real-world perspectives on product development and startup execution. These sessions are intended to expose students to how teams make decisions in practice—how trade-offs are handled, how uncertainty is managed, and how products evolve under real constraints such as time, funding, and customer expectations.

What this course is not!

This is not a traditional lecture-driven course, a step-by-step tutorial on specific technologies, or a place to build a fully polished, production-ready product. It is not about chasing features, writing code in isolation, or following a fixed set of instructions. In this course, requirements will evolve, assumptions will be challenged, and trade-offs are unavoidable. Success in this course is measured by evidence of learning, iteration, and progress—not by perfection, scale, or completeness.

Class Attendance & Professional Responsibility

Attendance is expected and viewed as a professional commitment to your teammates. Although a formal roll call is not taken, this is a selective senior course in which students are

expected to manage their time responsibly. Repeated lateness, uncommunicated absences, or failure to engage directly impacts your individual evaluation and team-based grading.

Illness, Emergencies, and Personal Circumstances :You will not be penalized for circumstances beyond your control, such as illness, injury, or emergencies. Do not attend class if you are ill. If you require flexibility or must miss a session, notify the teaching team as early as possible. You do not need to share sensitive personal details; please explain the situation and the expected duration so that we can arrange alternative participation or support, where appropriate.

Communication Protocol: Clear and timely communication is essential for adequate support. Please use email for all inquiries or accommodation requests to ensure a consistent written record. We ask that you communicate any concerns, absences, or requests well in advance of class; given the complexity of the course and our shared busy schedules, last-minute requests are complex to accommodate and may not be addressed in time.

Required Materials & Software

There are no required materials per se, but you must bring a laptop to class with a fully functional development environment for your project. iPads/tablets are not recommended as the primary platform unless you can develop for them.

All resources and readings are optional. I firmly believe that everyone learns in different ways that may change over time. For some people, reading is a fantastic approach. For others, reading can be tiresome and ineffective. I recommend skimming most resources to start, referring back to them later if you find a topic you want to learn more about.




Title	Author	Remarks (Where to find the title?)
Shape Up	Ryan Singer (Basecamp)	https://basecamp.com/shapeup
The Lean Startup	Eric Ries	http://theleanstartup.com
The Indie Hackers Podcast	Courtland Allen	https://www.indiehackers.com/podcast or wherever you normally listen to podcasts
The Architecture of Open Source Applications, Volume II	John O'Duinn	http://aosabook.org/en/index.html
Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification	Joy Buolamwini, Timnit Gebru	http://proceedings.mlr.press/v81/buolamwini18a.html

Assessment Summary

The assessment for this course is structured around the iterative release cycle rather than traditional high-stakes deadlines. It is designed to mirror the professional software lifecycle, rewarding consistent momentum, technical justification, and the ability to refine your system through repeated workflows.

You are evaluated on your ability to build functional software while justifying the architectural decisions that keep it running. The grading for this course is designed to mirror the professional software lifecycle, rewarding consistent progress, technical depth, and clear communication.

Your final grade is composed of three evaluation streams summarized in the table:

Component	Weight	Focus & Deliverables
 Development Cycle (A1–A10)	50%	Engineering Foundations: Architecture, CUJs, and release assets. Sprint Demos: Live code presentations and iterative feature builds.
 Team Check-ins	20%	10 check-ins (2% each) to clear blockers, verify momentum, and ensure collaborative progress.
 Final Live Demo	30%	Final system launch, comprehensive technical report of iterations, and personal engineering reflection.

Late Submissions Policy

Asynchronous Delivery Policies (Quercus/GitHub): Includes GitHub Releases (A1-A10) and Progress Review Quercus Submission

- The Sliding Scale: Assignments may be submitted up to five days after the deadline with a 10% penalty per day.
- Hard Cut-off: Submissions more than five days late will receive a score of 0.
- Mandatory Completion: Due to the cumulative nature of the course, even if a submission receives a zero, it must be completed to satisfy dependencies for future milestones.

Synchronous Delivery Policies (In-Person/Live): Includes Engineering Foundations Presentations, Live Demos, and Team Check-ins

- Live Execution: Because these require the teaching team’s presence for marking, missing a scheduled slot typically results in a score of 0 for that component.
- Pulse Check Absences: If a team misses a Check-in class, you must proactively reach out to the teaching staff. Unexcused absences or failure to communicate regarding an absence result in a 0 for the live portion (1% of the total grade).
- Consistency: Repeated absences without communication will significantly impact your final grade beyond the individual assignment weight.

Course Schedule

Week	Topic	In-Class Activity	Assignment / Milestone
Week 1	The Practitioner Mindset	Introduction to course, Course Policies and Administrative Review, Assignment Rubrics & Delivery	-
Week 2	Product-Market Fit & Roadmap	Guest Speaker: Defining CUJs (Critical User Journeys) & lean project management.	A1: Team Formation & Diversity Analysis Team Check-ins & Review [01]
Week 3	Architectural Foundations	Workshop: Setting up the Walking Skeleton (CI/CD and Cloud Landing Zone).	A2: Rapid MVP & CUJ Framework Team Check-ins & Review [02]
Week 4	Technical Decision Making	The Buy vs. Build debate; Architectural Decision Records (ADRs).	A3: Cloud Architecture & Deployment Team Check-ins & Review [03]
Week 5	Reliability & Observability	Beyond unit tests: Logging, Tracing, and Proactive Monitoring in the Cloud.	A4: Demo 1 (Surface Product) Team Check-ins & Review [04]
Week 6	The Art of Technical Narrative	Storytelling for Engineers: How to demo to stakeholders vs. peers.	A5: Demo 2 (Core Feature) Team Check-ins & Review [05]
Week 7	Reading Week	No Class - Use for technical debt cleanup.	-
Week 8	Competitive Resilience	Analysis of existing market solutions; Security & Stress-testing your logic.	A6: Demo 3 (Backend Integration) Team Check-ins & Review [06]
Week 9	Data Strategy & AI/MLOps	Building data pipelines; Integrating ML models; Ethical AI and Data Privacy.	A7: Advanced CUJ & Refinement Team Check-ins & Review [07]
Week 10	Product Validation & Scale	Cost optimization (FinOps), handling traffic spikes, and load testing.	A8: Demo 4 (Integration/Edges) Team Check-ins & Review [08]
Week 11	Enterprise Integration & Safety	Guest Speaker: AI Safety, Fraud Management, or Governance in Large Systems.	A9: Demo 5 (Beta Release) Team Check-ins & Review [09]
Week 12	The Final Pitch	Poster Showcase & Live MVP Demo.	A10: Launch Prep & Promo Video Team Check-ins & Review [10]
Week 13	The Final Pitch	Poster Showcase & Live MVP Demo.	Final Product Live Demo