LALLY SCHOOL
OF MANAGEMENT

SYLLABUS

Web Science Systems Development

ALT TITLE: Advanced Web Systems Development: Theory, Systems, and Practice in Full-Stack and Agentic AI Development

ITWS-4500
Room Location: Lally102
Class Time: Tuesday/Friday 12:00 PM – 1:50 PM (01); 2:00 PM – 3:50 PM (02);
RPILMS

INSTRUCTOR (REQUIRED)

Instructor Name: Jason Kuruzovich **Office Location:** *Pitt 2206*
Tel. No.: 518-698-9910 **Email Address:** kuruzj@rpi.edu*
Office Hours: Tuesday 9:00 - 11:00 (in person); Teams by appointment
Book online appointment: https://bit.ly/jason-rpi

COURSE DESCRIPTION:

Building on the technology covered in Web Systems Development, students will be exposed to current technologies, frameworks, and practices in the area of Web development. Types of topics included will be HTML5/CSS3, APIs for data, Node.js, MongoDB, Express.js, and R. Methodology to be explored will be application design, software versioning, and team development. Lab intensive, this course is intended to provide a foundation for the advanced courses in Data Science and Advanced Web Science.

Proposed:

Building on the foundations established in Web Systems Development, this course focuses on the **design and implementation of modern, production-quality web applications as integrated systems**. Students will examine **architectural and design patterns** that underlie contemporary web platforms, emphasizing system decomposition, data flow, scalability, reliability, and security.

The course adopts a full-stack perspective, integrating modern frontend frameworks, backend services, databases, and APIs into cohesive and maintainable applications. Core technologies include HTML5/CSS3, JavaScript-based frontend frameworks, Node.js, Express.js, MongoDB, and related tooling. Students will also explore user experience design, data modeling, and performance considerations that influence real-world application behavior.

In addition, the course introduces intelligent and automated system components, including AI-assisted development workflows and agent-based automation, and examines how such capabilities can be safely and effectively incorporated into web applications. Emphasis is placed on testing, containerization, continuous integration and deployment, and cloud-based deployment and scaling using contemporary platforms and practices.

The course is lab-intensive and project-driven, with students working collaboratively to design, build, deploy, and present a full-stack web application. By the end of the course, students will be prepared to reason about architectural trade-offs, communicate technical decisions, and operate complex web systems in real-world environments.

REQUIRED TEXT(S):

The reading materials will be tailored to provide the needed conceptual background for the various technologies examined. Links to the required readings will be available via the LMS.

STUDENT LEARNING OUTCOMES

1. Students will demonstrate knowledge of the theoretical foundations of Web Science.
2. Students will demonstrate knowledge of a broad range of technologies that must be deployed to design and build web applications.
3. Students will demonstrate knowledge of a broad range of issues that must be addressed to design and build web applications.
4. Students will be able to investigate and present on topics in Web Science at a level so that other students obtain an understanding of the key issues.
5. Students will develop team-building techniques in order to develop and deploy real-world web applications.

Proposed:

1. *Architectural Theory and Systems Thinking.* Students will demonstrate knowledge of the theoretical foundations of modern web application architecture, including architectural patterns, system decomposition, data flow, and trade-offs related to scalability, reliability, and security.

2. *Full-Stack Technology Integration.* Students will demonstrate the ability to design and implement full-stack web applications using a modern frontend, backend services, databases, and APIs, integrating multiple technologies into a coherent and maintainable system.

3. *Design, UX, and Data Considerations.* Students will demonstrate understanding of the technical, data, and user-experience considerations that influence the design of production-quality web applications, including state management, performance, usability, and data modeling.

4. *Intelligent and Automated Systems.* Students will demonstrate the ability to incorporate intelligent components into web applications, including AI-assisted development workflows and agent-based automation, while reasoning about correctness, reliability, and human oversight.

5. *Deployment, Scaling, and Operations*. Students will demonstrate the ability to deploy, test, and operate web applications in cloud environments, applying containerization, CI/CD pipelines, and scaling patterns to support real-world usage.

6. *Technical Communication and Analysis*. Students will be able to investigate, analyze, and clearly communicate architectural and technical decisions, including trade-offs and alternatives, in a manner that enables others to understand and evaluate complex systems.

7. *Collaborative Development and Professional Practice*. Students will demonstrate effective collaboration and professional software development practices, including version control, testing, documentation, and team-based project execution.

COURSE ASSESSMENT MEASURES

Assessment	Description	Learning Outcome #s
Lab Report	Each *individual* will submit one lab project per week, assigned the week prior.	1, 2, 3
Quizzes	Each *individual* will take two quizzes, one around the midpoint of the semester and the other after all lectures have been given.	1, 2, 3
Project	Each *group* will design, implement, iterate on, and present a semester-long web application project. The group will include overall and individual contributions.	1, 2, 3, 4, 5
Participation	Each *individual* will participate in class via the asking and answering of questions during lectures.	3, 4

GRADING CRITERIA

GRADING CRITERIA

Although Practicum in Management is a team-based class, grading is done individually for each student. Make sure you meet the requirements, including the requirements on the project itself such as presentations, written reports, project timeline updates, customer/client meetings, etc.

Grading Schedule

SUMMARY OF COURSE ASSESSMENT MEASURES

Assessment	Weight
Lab Assignments	**30%**
Semester Group Project	**30%**
– Overall System Architecture & Performance	15%
– Individual Contribution & Professional Practice*	15%
Quizzes	**30%**
Participation	**10%**
Total	**100%**

*Will incorporate peer feedback, project commits, instructor observation, individual summary of contribution.

Grades

>93	A
90-92	A-
87-89	B+
83-86	B
80-82	B-
77-79	C+
73-76	C
70-72	C-
66-69	D
0-65	F

ATTENDANCE POLICY

Active participation is essential in a project-based consulting course, both for your learning and for the success of your team and client engagement. Attendance will be taken at all class sessions, workshops, and presentations.

- **Absences:** More than **two unexcused absences** will result in a **5% reduction of your final grade for each additional unexcused absence.**
- **Notification:** If you must miss class, you are expected to notify both the instructor and your teammates in advance whenever possible.
- **Making Up Missed Work:** Students missing class should (a) provide an asynchronous update on project progress in Webex related to their group tasks, and (b) coordinate with teammates to confirm additional plans for the following week.
- **Excused Absences:** Excused absences (e.g., illness, religious observance, professional obligation) must be communicated to the instructor in a timely manner and will not impact your grade.

Because team collaboration are central to the course, **poor participation, lack of communication, or repeated absences may result in a more significant reduction to your grade** regardless of overall class performance.

OTHER COURSE POLICIES

Students must follow all institute safety and health protocols at all times throughout the semester. There will be no exception to these requirements at any time.

ACADEMIC INTEGRITY

Student-teacher relationships are built on trust. For example, students must trust that teachers have made appropriate decisions about the structure and content of the courses they teach, and teachers must trust that the assignments that students turn in are their own. Acts that violate this trust undermine the educational process.

The Rensselaer Handbook of Student Rights and Responsibilities and the Graduate Student Supplement (For 6000 level and above courses) define various forms of Academic Dishonesty and you should make yourself familiar with these. In this class, all assignments that are turned in for a grade must represent the student's own work. In cases where help was received, or teamwork was allowed, a notation on the assignment should indicate your collaboration. Submission of any assignment that is in violation of this policy will result in (1) an academic (grade) penalty and (2) reporting to Lally's Associate Dean of Academic Affairs and either the Dean of Students (for Undergraduates) or the Dean of Graduate Education (for Graduate students).

In this course, the academic penalty for a first offense is *loss of a letter grade for the course.* A second offense will result in failure of the course as noted in Lally's Three Strikes Policy.

If you have any questions concerning this policy before submitting an assignment, please ask for clarification.

ACADEMIC ACCOMMODATIONS

Rensselaer Polytechnic Institute strives to make all learning experiences as accessible as possible. If you anticipate or experience academic barriers based on a disability, please let me know immediately so that we can discuss your options.

To establish reasonable accommodations, please register with The Office of Disability Services for Students (mailto:dss@rpi.edu; 518-276-8197; 4226 Academy Hall). After registration, make arrangements with me as soon as possible to discuss your accommodations so that they may be implemented in a timely fashion."

Tentative Course Calendar

Week	Class Date	Topic
1	1/13/26	Course Introduction: Web Applications as Architectural Patterns; Group Formation
1	1/16/26	Lab 1: MERN Stack Foundations
2	1/20/26	Architectural Theory: Patterns, System Decomposition, and Project Scoping
2	1/23/26	Lab 2: MERN Backend – Node.js & Express Architecture
3	1/27/26	Frontend Architecture Patterns; Project Proposal Presentations
3	1/30/26	Lab 3: Frontend Architecture with React
4	2/3/26	Backend Architecture Patterns: Services, APIs, and Data Flow
4	2/6/26	Lab 4: Data Architecture with MongoDB and PostgreSQL
5	2/10/26	Identity, Authentication, and Mobile Architecture Patterns
5	2/13/26	Lab 5: Authentication and Mobile Application Development
6	2/17/26	Agentic Architecture Patterns I: Agents, Tools, and Workflows
6	2/20/26	Quiz 1: Architectural Foundations and System Design
7	2/24/26	Midterm Project Presentations
7	2/27/26	Project Work Day
8	3/10/26	Agentic Architecture Patterns II: Orchestration, Memory, and Control
8	3/13/26	Lab 6: Implementing Agentic AI Workflows
9	3/17/26	Continuous Integration and Testing as Architecture
9	3/20/26	Lab 7: CI Pipelines and Automated Testing
10	3/24/26	Deployment Architecture: Infrastructure as Code and Cloud Patterns
10	3/27/26	Lab 8: Containerization and Web Services
11	3/31/26	Deployment Architecture II: Performance, Caching, and Efficiency
11	4/3/26	Lab 9: Caching, Scaling, and Reliability
12	4/7/26	AI-Assisted Development: Human–AI Collaboration Patterns
12	4/10/26	Lab 10: AI-Assisted Coding and Testing
13	4/14/26	System Integration, Evolution, and Course Synthesis
13	4/17/26	Lab 11: Build Something Interesting
14	4/21/26	Quiz 2: Comprehensive Architectural and Systems Reasoning
14	4/24/26	Project Work Day: Final Integration and Hardening
15	4/28/26	Final Project Presentations: Evaluation, Feedback, and Reflection