

EECS 485: Web Systems Syllabus

The University of Michigan, Fall 2024

A holistic course of modern web systems and technologies, covering front end and back end. Build an Instagram clone in the first half of the semester and a Google clone in the second.

Instructors Andrew DeOrio <awdeorio@umich.edu>, John Kloosterman <jklooste@umich.edu> and Raed Almomani <almomani@umich.edu>.

Instruction

Lectures will be in-person. You may attend any lecture section or watch any of the recordings, even if it's different from your registered section. We don't take attendance in lecture. Studio recordings from a past semester are posted for your reference, but exam questions will be drawn from the live lectures given this semester and not these past recordings.

Labs will be in-person, and one lab will be recorded. Expect hands-on practice with concepts from lecture and project introduction material. You may attend any lab section or watch the recordings, even if it's different from your registered section. We don't take attendance in lab.

Projects will be completed in groups and turned in to an autograder. If you'd like, we'll help you find a group.

Exams will be in-person on paper.

Office hours will be hybrid, with in-person on Mondays, Wednesday, and Thursdays, and remote on Tuesdays and Fridays.

Computer and network recommendations

Make sure you have a laptop consistent with [CAEN recommendations](#).

Test your internet connection with the [U-M Custom Speedtest website](#) and make sure it meets the [minimum requirements](#) for all UM or CAEN services. You'll need more bandwidth if there will be multiple simultaneous users in your household.

Resources for help with computing equipment:

- Information and Technology Services (ITS) [Laptop loaner program](#)

- College of Engineering (CoE) [Office of Student Affairs](#), email requests to coe-studentaffairs@umich.edu

Communication

eecs485.org links to all course resources and is required reading.

[Ed](#) is the course discussion forum, best for technical questions and project updates. Do not publicly post your code. (Follow the Q&A Forum link on the sidebar of the home page to join).

eecs485staff@umich.edu reaches the course staff, best for non-technical questions.

[Individual professor](#) email addresses are best for confidential matters.

Overview

EECS 485 integrates concepts from multiple computer science disciplines used in web applications ("front end"), web services ("back end") and web semantics ("extracting knowledge and information"). Within these three themes, the course covers concepts such as:

Web Applications	Web Distributed Systems	Web Semantics
<ul style="list-style-type: none">• Static pages• Server-side dynamic pages• Client-side dynamic pages• Sessions• REST APIs• Security	<ul style="list-style-type: none">• Distributed compute• Distributed storage• DNS• Cloud computing: IaaS, PaaS, and SaaS• Parallel programming	<ul style="list-style-type: none">• Text analysis for web search• Link analysis for web search• Recommender systems• Ads and Auctions

Students build a social media application similar to Instagram or Facebook and a search engine like Google or Bing. They gain experience with practical web technologies such as:

<ul style="list-style-type: none">• HTML and CSS• Python• JavaScript• Flask (server-side dynamic pages)• React (client-side dynamic pages)	<ul style="list-style-type: none">• Amazon Web Services• SQL• Version control with Git• Shell scripting• Linux system administration
--	--

Objectives

By the end of this course, successful students will feel comfortable learning a new web technology independently using developer documentation. They will be able to:

<ul style="list-style-type: none">• Design and implement a web application• Balance the trade-offs between static pages, server-side dynamic pages and client-side dynamic pages• Use JavaScript and asynchronous programming to invoke remote procedure calls in web user interfaces• Use JSON for messaging and data exchange	<ul style="list-style-type: none">• Design and implement a parallel program using processes, threads and sockets• Process web data using a parallel compute framework such as MapReduce• Use IaaS and PaaS to scale a web application• Balance the trade-offs of different distributed storage systems	<ul style="list-style-type: none">• Design and implement a pipeline of MapReduce programs to build a web search inverted index• Rank search results using text and link analysis• Balance the trade-offs of different auction mechanisms for web ad sales• Describe how web recommendations are made• Describe the societal impact of web technology
--	---	--

Prerequisites

The prerequisite for EECS 485 is EECS 281.

This course requires substantial independent learning. We'll provide you with a starting point, but expect to learn independently a lot of HTML, CSS, SQL, Python, and Javascript.

Exams

There will be one midterm and one final examination. Exam dates are posted on eecs485.org. There will be no make-up exams. It is your responsibility to verify you can attend both exams.

Practice exams for many previous semesters are in the [EECS 485 Exam Repository](#). We'll support exams from the most recent two semesters in office hours.

Labs

Labs will contain hands-on practice with concepts from lecture and helpful project introduction material.

Projects

This course contains 5 programming projects. The first three focus on web apps, while the last two focus on web distributed systems and web semantics.

[Project 1: Templated Static Site Generator](#). An Instagram clone implemented with a templated static site generator.

[Project 2: Server-side Dynamic Pages](#). An Instagram clone implemented with server-side dynamic pages.

[Project 3: Client-side Dynamic Pages](#). An Instagram clone implemented with client-side dynamic pages.

[Project 4: MapReduce](#). A single machine, multi-process, multi-threaded server that executes user-submitted MapReduce jobs.

[Project 5: Search Engine](#). A scalable search engine similar to Google or Bing.

We improve the projects each semester, so they may change any time before the project release date.

Project Groups

Project 1 is independent. The remaining projects are completed in groups of 2-3.

You may modify group membership between projects. You must not modify group membership during a project. In exceptional cases, you may request group dissolution via email to the faculty. If the dissolution is granted, all group members may use previously shared code and no group member may join another group for remainder of the project.

Each group turns in one solution to the autograder. You must register your group on the autograder. All group members earn the same grade on a project. We may adjust the grades of students who substantially under-participate based on feedback from their group members.

For those retaking the course, see the policy on [code reuse](#).

Project Grading

We use a web-based autograder to evaluate correctness and style.

Before the deadline, you may submit up to 3 times per day. After each submission, the autograder shows the results of the public tests released with the project.

After the deadline, the autograder shows the results of private tests.

Your final project score is a combination of public and private tests. We use the submission that received the combined best score. If multiple submissions share the best score, we grade the last.

Grades

Letter grades are assigned on a curve. In general, our goal is be consistent with [past semesters](#).

We calculate your total weighted score using these weights.

Assignment	Percentage of Grade
Programming Projects	5 projects x 10% each = 50%
Midterm Exam	25%
Final Exam	25%

Curve

Sometimes the difficulty of exams varies from semester to semester. We use a curve to maintain consistent letter grades despite exam variation. In general, our goal is be consistent with [past semesters](#), where a median grade is 3.3 (B+).

Minimum scores

You are guaranteed to pass EECS 485 with a C or better if:

- Your total weighted project score is at least 65.0%; and
- Your total weighted exam score is within 1.5 standard deviations of the mean or above 60%, whichever is in your favor.

If your scores are below these thresholds, your maximum letter grade may be a C-.

Deadlines and Extensions

We do not accept late work.

We will consider extension requests made at least two weeks in advance, for example, for religious holidays or planned medical procedures.

For medical or personal emergencies we will consider exceptions on a case-by-case basis. Please provide documentation of the emergency. In the case of weekend project deadlines, extension requests must be sent by noon on the preceding Friday in order to be considered.

Projects are due on Sunday, but staff will support projects through Friday 5PM in office hours and on the forums. If you choose to work on the project after Friday 5PM, staff will not be available during that time for questions.

We expect everyone to submit Project 1 by the due date, regardless of whether you are registered, on the waitlist, or are temporarily unable to register for another reason. To cover unforeseen circumstances, students adding EECS 485 late may petition for additional time to turn in Project 1. The extension is limited to 48 hours from time of registration. If you receive this extension, you must work alone on Project 2.

Regrades

Exams are graded by hand. Request a regrade via Gradescope. Your score may go up or down.

Projects are automatically graded. We do not accept regrade requests.

In all cases regrade requests are due no later than 7 days after a grade is released unless a shorter deadline is specified.

Policies

Academic Integrity

We encourage collaboration in EECS 485, especially on concepts, tool, specifications, and strategies.

All work you submit must be your own or your group's. Collaboration must not result in code that is identifiably similar to other solutions, past or present.

You are still responsible for following these rules even after finishing the course.

If you are unsure about what constitutes an honor code violation, please contact the course staff with questions.

Code reuse

If you are retaking the course, you may work in a group or you may reuse your old code, but not both. When reusing your own code, you must work alone and the code must have been wholly written according to rules outlined in this syllabus. It is possible for instructors to miss an honor

code violation in a previous term, but catch and report it when the code is reused. Also note that there may be changes to project specifications each semester.

Code publishing

You may not make your code publicly available in any form, for example, in a public GitHub repository or personal website. You are still responsible for following these rules even after finishing the course.

If you are recruiting and would like to show companies your work in EECS 485, our recommendation is to create a Git repository with only a README file and no code. Take a screenshots of your solution put those in the README. Include a description of the languages, libraries, frameworks, and technologies you used. Another option is to privately demo or send your code to a recruiter.

Generative AI (Copilot, ChatGPT)

The use of generative AI tools such as GitHub Copilot and ChatGPT is permitted for projects. Some examples of responsible use of generative AI include autocompleting repetitive/boilerplate code and generating usage examples for library functions. Creating large sections of code you do not understand yourself is using generative AI in an irresponsible way.

Remember that during the exams you will write code without the help of generative AI tools.

Recordings

Course lectures may be audio/video recorded and made available to other students in this course and students on the waitlist. As part of your participation in this course, you may be recorded. If you do not wish to be recorded, please contact your [instructor](#) the first week of class to discuss alternative arrangements.

Students may not record or distribute any class activity without written permission from the instructor, except as necessary as part of approved accommodations for students with disabilities. Any approved recordings may only be used for the student's own private use.

Accommodations for Students with Disabilities

If you think you need an accommodation for a disability, please let your instructor know during the first three weeks of the semester. Some aspects of this course may be modified to facilitate your participation and progress. As soon as you make us aware of your needs, we can work with the Services for Students with Disabilities (SSD) office to help us determine appropriate academic accommodations. SSD (734-763-3000; <https://ssd.umich.edu>) typically recommends

accommodations through a Verified Individualized Services and Accommodations (VISA) form. Any information you provide is private and confidential and will be treated as such.

Commitment to Equal Opportunity

As indicated in the [General Standards of Conduct for Engineering Students](#), we are committed to a policy of equal opportunity for all persons and do not discriminate on the basis of race, color, national origin, age, marital status, sex, sexual orientation, gender identity, gender expression, disability, religion, height, weight, or veteran status. Please feel free to contact us with any problem, concern, or suggestion. We ask that all students treat each other with respect.

Students' Mental Health and Well-being

Students may experience stressors that can impact both their academic experience and their personal well-being. These may include academic pressure and challenges associated with relationships, mental health, alcohol or other drugs, identities, finances, etc.

If you are experiencing concerns, seeking help is a courageous thing to do for yourself and those who care about you. If the source of your stressors is academic, please contact me so that we can find solutions together. For personal concerns, U-M offers many resources, some of which are listed at [well-being services for students at U-M](#).

