Syllabus — TCSS 460 A Sp 26

Course Information

Institution : School of Engineering and Technology, University of Washington Tacoma

Course : TCSS 460 A — Client/Server Programming For Internet Applications

Credits : 5

Quarter : Spring 2026

Schedule : T/Th 3:40 – 5:40 PM, JOY 207

Prerequisites : TCSS 360 — Software Development and Quality Assurance

Instructor

Instructor : Charles Bryan — cfb3@uw.edu

Office : CP 216, Discord, or Zoom

Office Hours : By appointment — no, seriously, just ask and I'll work out a time to meet

---

Course Description

Examines the languages and techniques for internet client/server application programming. Includes languages like CGI, Perl, XML, JavaScript, and DHTML, and topics like scripts, queries, forms, data access, redirection, firewalls, proxies, hypermedia, cookies, and gateways.

In practice, this course is a project-based introduction to full-stack web development. Students design and build a complete web application — from a RESTful API backed by a relational database to an interactive front-end — using modern tools and frameworks (currently TypeScript, Express, PostgreSQL, Prisma, React, and Next.js). The focus is on understanding client/server architecture, not memorizing any single technology.

Learning Objectives

By the end of this course, students will be able to:

1. Design and implement RESTful web APIs using a server-side framework (e.g., Express)
2. Model and access relational data using an ORM (e.g., Prisma) backed by a relational database (e.g., PostgreSQL)
3. Implement authentication and authorization using token-based and federated identity patterns (e.g., JWT, OAuth2)
4. Build interactive front-end applications using a component-based framework (e.g., React/Next.js)
5. Deploy full-stack applications to cloud infrastructure
6. Transfer object-oriented programming skills to a new language and ecosystem (e.g., Java to TypeScript)
7. Collaborate in teams using version control workflows, sprint milestones, and code review (e.g., Git, GitHub)

Course Outcomes

Relationship of course to program outcomes — UWT Student Learning Goals that this course contributes to:

Inquiry and Critical Thinking : Students will acquire skills and familiarity with modes of inquiry and examination from diverse disciplinary perspectives, enabling them to access, interpret, analyze, quantitatively reason, and synthesize information critically.

Communication/Self-Expression : Students will gain experience with oral, written, symbolic, and artistic forms of communication and the ability to communicate with diverse audiences. They will also have the opportunity to increase their understanding of communication through collaboration to solve problems or advance knowledge.

Grading

Component	Weight
Group Project	50%
Check-Off Assignments	25%
Reading Quizzes	20%
AI Diary & Course Reflection	5%

Group Project (50%)

A full-stack web application built in 9 sprints (Sprint 0–8) over the quarter. Groups form at the end of Week 1. Back-end sprints (Weeks 3–6) are followed by front-end sprints (Weeks 7–10). Each sprint is graded independently and not all sprints are weighted equally. All sprints are due Sunday at the end of the sprint week.

Check-Off Assignments (25%)

Individual assignments that verify your understanding of key tools and concepts. Each check-off includes a hands-on task — you will build something and demonstrate your understanding to a peer. Your lowest check-off score will be dropped.

Reading Quizzes (20%)

Weekly Canvas quizzes based on the assigned concept readings (Weeks 1–8). Quizzes are open-note, timed, and drawn from question pools — you may not see the same questions on each attempt. You get two attempts and the higher score is kept. Your lowest quiz score will be dropped.

AI Diary & Course Reflection (5%)

A weekly log of how you use AI coding tools throughout the quarter, submitted during finals week alongside a written course reflection and peer ratings. Details and a template will be provided separately.

Late Work

No late work is accepted for any component. The dropped quiz and dropped check-off provide a safety net for illness, emergencies, or a rough week.

Schedule

Week	Dates	Lecture Focus	Quiz	Check-Off	Sprint
1	Mar 31 – Apr 4	Express basics, HTTP methods, params, body	Quiz 1	—	—
2	Apr 7 – Apr 11	API testing, Middleware	Quiz 2	Running Web API + API Testing	Sprint 0
3	Apr 14 – Apr 18	Proxy pattern, DB + Prisma intro	Quiz 3	Proxy API	Sprint 1
4	Apr 21 – Apr 25	Prisma deep dive, migrations	Quiz 4	Prisma Intro	Sprint 2
5	Apr 28 – May 2	Auth concepts, auth-squared	Quiz 5	Auth-Squared	Sprint 3
6	May 5 – May 9	Evolution of web, HTML/React crash course	Quiz 6	React Tutorial	Sprint 4
7	May 12 – May 16	OAuth2, NextAuth	Quiz 7	TBD	Sprint 5
8	May 19 – May 23	FE patterns, accessibility, performance	Quiz 8	TBD	Sprint 6
9	May 24 – May 30	—	—	TBD	Sprint 7

Week	Dates	Lecture Focus	Quiz	Check-Off	Sprint
10	May 31 – Jun 6	—	—	—	Sprint 8
Finals	Jun 9 – Jun 11	Reflection			

All coursework is due Sunday at 11:59 PM at the end of the listed week, unless otherwise stated.

Policies

Attendance

Attendance is encouraged but not required. Lectures will be recorded and lecture recaps are available on the course site. That said, recordings are a backup — not a substitute. Technical issues occasionally prevent recordings, and the in-class experience includes live coding, Q&A, and discussions that don't translate well to a recording.

AI Tools

AI coding tools are not just allowed — they are part of the curriculum. You will learn to use agentic coding tools alongside traditional development practices. See the AI Coding Assistant guides for setup and usage.

The goal is not to avoid AI, but to understand what it produces. Check-offs are verified in person — if you cannot explain your code, it does not matter who or what wrote it.

Academic Integrity

This course follows the University of Washington Tacoma Academic Integrity Policy. Violations will be reported to the Office of Student Conduct.

Communication

Discord is the primary communication channel for this course. Use it for questions, discussion, and peer help. For private or sensitive matters, email the instructor directly.

The instructor will make every effort to respond to messages within **24 hours on weekdays**. Responses on weekends may be delayed.

Group Conflict Resolution

Working in teams is a core part of this course and a core part of professional software development. Groups are expected to resolve disagreements internally when possible. That said, **do not wait until a situation becomes unmanageable** — come talk to the instructor early if your group is struggling. The peer ratings in the final reflection provide an additional channel for feedback on group dynamics.

Accessibility

Students with disabilities are encouraged to contact Disability Resources for Students (DRS) to establish accommodations. Please share your accommodation letter with the instructor as early as possible so we can support your learning.

Campus Resources

For campus-wide policies, resources, and student support information, see the UW Tacoma e-Syllabus.

Required Materials

All course materials are provided through this site and Canvas. There is no required textbook. Software installation guides are available in the Guides section.