CS211 - Full Stack Web Development - Spring 2025 - Course Rubric	
Learning standard mastery levels: M - (Mastery) Student demonstrates mastery of the standards assessed. PM - (Partial mastery) Student demonstrates partial mastery of the standards assessed. EU - (Emerging understanding) Student demonstrates an emerging understanding of the standards assessed. IE - (Insufficient evidence) Student has submitted insufficient evidence for the standards assessed. N/A - (Not assessed) This was not yet or will not be assessed for the student in the course	
Content & Knowledge (25%)	
Front-End Technologies & Client-Side Paradigms	
PM	HTML - Student can build websites using HTML. Student uses appropriate HTML elements for links, images, text, and other web content. Student writes HTML with accessibility in mind, including alt text on images.
EU	CSS - Student can apply custom styles to their website using CSS. Student can translate their design ideas into basic and advanced CSS attributes. Student effectively uses a variety of CSS selectors to target HTML elements based on class, pseudo class, type, hierarchy, etc.
EU	JavaScript - Student can write JavaScript to add interactive features to websites, such as click interactions, form submissions, animations, multimedia, and more. Student appropriately uses the computational structures of JavaScript: functions, objects, classes, etc.
N/A	JavaScript Frameworks - Student can use the Svelte framework (or another framework of their choice) to create a web application made up of components. Student's code leverages props and state in the React component lifecycle to render changes on their website.
Back-End Technologies & Server-Side Paradigms	
N/A	APIs and CRUD - Student can create and test an Application Programming Interface (API) for client/server communication in a full stack web application. Their API correctly implements the four basic operations of persistent storage: CRUD (Create, Read, Update, Delete)
N/A	Databases - Student can create and manage a database for persistent, server-side data storage for their web application. Student the difference between document-based databases (MongoDB) versus relational databases.
Version Control	
N/A	Git / GitHub - Student can use basic git commands (pull, push, commit, branch, merge) to upload and download code to and from GitHub. Student can maintain multiple working branches (or versions) of their code using git version control.
Skills & Practices (50%)	
Code Organization and Abstraction	
N/A	Code Organization - Student's code follows a semantically logical order and is well-organized with clearly named variables and functions. Student comments their code thoroughly, making their program readable by documenting their justification for making technical decisions. Student's coding project structure follows modern conventions for file organization.
N/A	Abstraction - Student uses programming libraries and higher-level computational structures, such as classes or higher-order functions, to reduce repetition in their code and make it more maintainable and extensible

Data Modeling and Manipulation	
N/A	React Component Hierarchy - Student models UI information as well-defined, modular React components. Student manipulates the data flow across multiple components by hoisting state, thinking critically about what information each component should be "responsible" for.
N/A	Database Scemas & CRUD paradigm - Student thoughtfully designs a database schema to model complex, interrelated information. Student designs an API that allows client applications to read and write this information, creating a communication protocol between client and server. Student can justify the technical design decisions made around database model property types and API uri paths and request parameters.
Debugging	
PM	Debugging - Student uses debugging techniques and technologies to pinpoint the cause of errors and persistently attempts multiple solutions for resolution. Student leverages the Chrome Developer Tools to read and interpret error messages, inspect HTML elements, experiment with CSS styles, set breakpoints, and more.
Project Management	
IE	Project Planning and Scoping - Student breaks project down into meaningfully useful steps, reevaluates and utilizes their project plan as they make progress. Student prioritizes and deprioritizes tasks to produce complete projects within the time allotted.
PM	Time Management - Student turns in completed work before the deadline or communicates if they need an extension.
Design & User Experience	
IE	Design and UX - Student's web UIs follow design principles such as visual hierarchy, contrast, and discoverability. Users can easily navigate the website and achieve desired goals. Student's UI adheres to universal design standards for accessibility.
Habits of Learning (25%)	
PM	Perseverance - Student engages with struggles and setbacks using a variety of approaches, learn more about themselves and the material in the process.
PM	Intentionality - Student makes thoughtful choices regarding their use of time, resources, and engagement with the class, in order to make progress.
IE	Citizenship - Student engages with the classroom and other people in a way that supports the learning and safety of everyone involved.
IE	Reflection and self-direction - Student reflects on successes and growth areas to create appropriate challenges and learning goals.