

CMSI 2021

WEB APPLICATION DEVELOPMENT

<http://dondi.lmu.build/fall2023/cmsi2021>

Fall 2023—Featherston 330 (01); Doolan 219 (02); 2 semester hours
MW 3:40–4:50pm (01); MW 12:15–1:25pm (02)

John David N. Dionisio, Ph.D.
dondi@lmu.edu

Office Hours MW 5–7pm
or by appointment (*don't hesitate to ask!*)

Doolan 102
(310) 338-5782

Objectives and Outcomes

This is a two-unit practicum course, meaning you will receive training and gain experience building systems according to modern best practices. All computer science majors must complete this or Mobile Application Development (CMSI 2022) in order to experience full-stack development *before* undertaking the rigorous courses in the junior and senior years. Long after the course concludes, my hope is that you will be able to:

1. Complete web application projects as full-stack developers
2. Demonstrate useful skills in front-end, back-end, API, and database development
3. Participate in an agile development process
4. Demonstrate competency in a web application stack, using modern frameworks such as React
5. Show knowledge of established technology company processes related to business and software development concerns, such as ideation, pitching, and presenting

In addition to the course-specific content, you are also expected to:

6. Follow disciplinary best practices throughout the course

Prerequisites/Prior Background

You should have knowledge and experience equivalent to at least one semester of computer programming, but more importantly you must bring: (1) a growth mindset, (2) a respectful attitude, and (3) a willingness to work hard, to never give up, to accept feedback, and to strive to avoid imposter syndrome and stereotype threat. Make sure you take the time for effective self-study. Interact fully with classmates and teammates: the computing industry is one of the most collaborative fields in which to work, and your course experience should reflect this.

Materials and Texts

Most of the course content will be found online—and no different from what actual web app developers use for their own learning. It's a practicum course after all! Links to these resources are on the course website. In addition, do not hesitate to look for further information regarding the concepts, techniques, tools, and paradigms in the course.

As a practicum, we focus on specific technologies to keep things cohesive and concrete. For this offering, these are the React on the front end and Firebase on the back. Code will be written in CSS, JavaScript, and JSX (which looks a lot like, but isn't exactly, HTML). As such, you'll need:

- A web browser (pretty sure you have one of those already 😊)
- A source code editor, ideally one that supports the Prettier code formatter plug-in
- A GitHub account for storing your code
- A Google ID for use with Firebase

Course Work and Grading

Your final grade will be based on the percentage of the points you get for the following deliverables against the total number of possible points:

| | |
|----------------------------|-----|
| Standalone vanilla web app | 100 |
| Generic API-backed web app | 100 |
| Firebase-backed web app | 100 |
| Your own web app | 200 |

Total 500 points

Percentages $\geq 90\%$ get an A– or better; $\geq 80\%$ get a B– or better; $\geq 70\%$ get a C– or better. I may nudge grades upward based on qualitative considerations such as degree of difficulty, effort, class participation, time constraints, and overall attitude toward the course.

Disciplinary Best Practices (Objective 6)

Appearance of these in the syllabus constitutes fair warning of the consequences of not heeding them.

Punctuality

An assignment's number is its due date in *mmd* format, and it is always due by 11:59:59.999pm of that date. Point values are based on the state of your assignments at that moment. If there is an issue that prevents you from submitting an assignment on time (e.g., personal or family issues, sickness, conference attendance, can't-miss world bosses, job interviews, a family trip, or emergencies), let the instructor know ahead of time.

Clean Code

The *quality* of your code, not just its *correctness*, will play a part in determining your grade. Please refer to the course website's resources and notes on clean code for information on expectations of code quality.

Version Control

Version control is an indispensable part of today's computer science landscape in industry, the academe, and the open source community. We use version control heavily in this course: make sure that you get the hang of it.

None of the assignments can be completed (well) overnight; they should be the result of steady progress from the moment they are assigned to the date they are due. "One and done" submissions will negatively affect the final score.

Workload Expectations

In line with the *LMU Credit Hour Policy*, the workload expectation for this two-credit-hour course is a minimum of $2 \times 3 = 6$ hours of work by an average student per week, including the time that we spend together in the classroom.

Attendance

Attendance at all sessions is expected, but not absolutely required. If you must miss class (for reasons such as those listed in the *Punctuality* section, or anything else), it is your responsibility to notify me about this and keep up with the course.

The last day to add or drop a class without a grade of W is September 1. The withdrawal or credit/no-credit deadline is November 17.

Academic Honesty

Loyola Marymount University is a community dedicated to academic excellence, student-centered education, and the Jesuit and Marymount traditions. As such, the University expects all members of its community to act with honesty and integrity at all times, especially in their academic work. Academic honesty requires that all members of the LMU community act with integrity, respect their own intellectual and creative work as well as that of others, acknowledge sources consistently and completely, act honestly during exams and on assignments, and report results accurately. As an LMU Lion, by the Lion's Code, you are pledged to join the discourse of the academy with honesty of voice and integrity of scholarship.

Academic dishonesty will be treated as an extremely serious matter, with serious consequences that can range from receiving no credit for assignments/tests to expulsion. It is never permissible to turn in any work that has been copied from another student or copied from any source (including the Internet) without properly acknowledging/citing the source. It is never permissible to work on an assignment, exam, quiz, or any project with another person unless your instructor has indicated so in the written instructions/guidelines. It is your responsibility to make sure that your work meets the standard of academic honesty set forth in the "Academic Honesty Policy" found at:

<https://academics.lmu.edu/honesty>

Technology Use and Academic Honesty

In this course, academic honesty includes the *appropriate* use of technology as an aid for learning and productivity. This includes but is not limited to generative artificial intelligence tools such as ChatGPT and GitHub Copilot.

Modern Generative AI provides fast solutions to a variety of computing problems, but should be used responsibly in the classroom setting to get the most out of your education. Generative AI will not always be helpful nor available, like in job interviews, technologies it hasn't seen, or extraordinary/unique scenarios. It cannot independently validate the code that it produces, which may contain bugs or security vulnerabilities. As such, you will still need to deeply understand the foundations of computing and should use it in your coursework sparingly, especially when its use might rob you of the same

learning and desired difficulty of problem solving that comes from programming on your own.

That said, here are some acceptable use cases of generative AI on programmatic assignments:

- Generating docstrings and type hints
- Getting examples/explanations for language mechanics/syntax (e.g., how to use the spread operator; how to assign variables directly from an array or object; how to implement an error boundary component)
- Interpreting error messages
- Generating unit tests for edge cases
- Brainstorming new ideas for apps or app names

TL;DR *Never use technology (including but not limited to AI) to avoid learning, but do use it to augment, amplify, or accelerate (your) learning.*

Course Evaluations

Student feedback provides valuable information for continued improvement. All students are expected to fairly and thoughtfully complete a course evaluation for this course. This semester, course evaluations will be administered online through the Blue™ evaluation system. You will receive an email notification at your Lion email address when the evaluation form is available. You may also access the evaluation form on Brightspace (<https://brightspace.lmu.edu>) dashboard during the evaluation period. Your responses will be anonymous and will not be linked to you in any way.

Special Accommodations

The Disability Support Services (DSS) Office offers resources to enable students with ADD/ADHD; physical, learning, and psychiatric disabilities; and those on the autism spectrum to achieve maximum independence while pursuing their educational goals. Staff specialists interact with all areas of the University to eliminate physical and attitudinal barriers. Students must provide documentation for their disability from an appropriate licensed professional. Services are offered to students who have established disabilities under state and federal laws. DSS personnel also advise students, faculty, and staff regarding disability issues. Students who need reasonable modifications, special assistance, academic accommodations or housing accommodations should direct their request to the DSS Office as soon as possible. All discussions will remain confidential. The DSS Office is located

on the 2nd floor of Daum Hall and may be reached by email at dsslmu@lmu.edu or phone at (310) 338-4216. Please visit <http://www.lmu.edu/dss> for additional information.

Topics and Important Dates

Correlated outcomes are shown for each topic. Specifics may change as the course progresses. University dates (italicized) are less likely to change.

| | |
|------------------------|--|
| August | HTML/CSS/JavaScript |
| September | React; network, authentication, and database APIs |
| <i>September 1</i> | <i>Last day to add or drop a class without a grade of W</i> |
| <i>September 4</i> | <i>Labor Day; no class</i> |
| October | Firebase and Firestore; Material UI; React Router |
| November | Testing, context, server/cloud deployment, additional topics as time allows |
| <i>November 17</i> | <i>Last day to withdraw from classes or apply for Credit/No Credit grading</i> |
| <i>November 22–24</i> | <i>Thanksgiving break; no class</i> |
| December | Ideation and design; pitch presentations and work sessions |
| <i>December 12, 13</i> | <i>Final web app presentations (based on LMU final exam schedule)</i> |
| <i>December 14</i> | <i>Final web app assignment due</i> |

You can view my class calendar and office hour schedule in any iCalendar-savvy client. Its subscription link can be found on the course web site (it's too long to provide in writing).

Tentative Nature of the Syllabus

If necessary, this syllabus and its contents are subject to revision. Students are responsible for any changes or modifications announced or distributed in class, emailed to students' LMU Lion accounts, or posted on LMU's course management system, Brightspace. If you are absent from a synchronous class session, it is the student's responsibility to check Brightspace and with the professor to see if you missed any important class announcements. Students should not rely on word-of-mouth from classmates.