

WEB 3430: FULL STACK JAVASCRIPT DEVELOPMENT

3 CREDIT HOURS - FALL 2021 - ONLINE

COURSE SYLLABUS

INSTRUCTOR

Abdulmalek Al-Gahmi, PhD

CONTACT INFORMATION

Office: CAE 166

Email: aalgahmi@weber.edu (*Use this email instead of Canvas Inbox*)

OFFICE HOURS (ONLINE VIA THE ZOOM LINK BELOW)

Mon/Tue: 10:00am – 11:15am

Wed: 10:00am – 12:30pm

Zoom URL: <https://weber.zoom.us/j/95044844491>

CO-REQUISITES

WEB 3200: Dynamic Languages for Web Development

COURSE DESCRIPTION

Modern web development increasingly involves using end-to-end JavaScript-based technologies such as MongoDB, Express.js, Angular, React, Vue.js, Node.js, etc. This course introduces development techniques that capitalize on the strengths of every layer in this JavaScript-based full-stack.

LEARNING OBJECTIVES

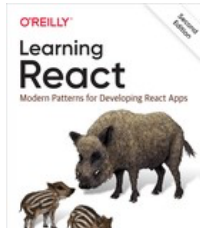
At the conclusion of this class, the students will be able to:

- Differentiate between front-end and back-end web development;
- List the pros and cons of Single Page Applications (SPA's);
- Distinguish between public-facing and private pages;
- Create and manage a NoSQL database using MongoDB;
- Build a secure full-stack web application using JavaScript;
- Design and consume a REST API; and
- Deploy a JavaScript-based full-stack application.

LEARNING RESOURCES

TEXTBOOKS

The following textbooks will be used in this course; they are available online.



Learning React (2nd Edition)

By: Alex Banks, Eve Porcello

Pub. Date: June 2020

ISBN: 9781492051725

Available for free in [O'Reilly Safari Learning Platform](#) via the library



Eloquent JavaScript (3rd Edition)

By: Marijn Haverbeke

Pub. Date: December 2018

ISBN: 978-1593279509

Available for free in <http://eloquentjavascript.net>

CANVAS

Canvas is where course modules, assignments, grades, and announcements will reside. It can be accessed from <https://canvas.weber.edu>. For Canvas-related technical support, please click the HELP link in the top right corner of your screen. You can also call WSU Online at (801) 626-6499 or email wsuonline@weber.edu.

PLURALSIGHT

A channel of relevant video-based Pluralsight tutorial courses has been created and will be used to supplement the learning materials of this course. Invitations to Pluralsight will be sent out during the first week of the course.

RECOMMENDED DEVELOPMENT ENVIRONMENT

This course uses the following software applications to set up a simple development environment that exposes students to essential tools/skills like GitHub, and the command-line.

- Text editor: [Visual Studio Code](#)
- JavaScript: [Node.js](#)
- Git: XCode Command-line Tools (MacOS) or [Git for Windows](#) (Windows)

ONLINE RESOURCES

- Software/Frameworks (the MERN stack)
 - M: [MongoDB Community](#)
 - E: [Express](#)

- R: [React](#)
- N: [Node.js](#)
- [Emmet Cheat Sheet](#)
- [GitHub](#) and [Git Cheat Sheet](#)
- Cloud-based serverless hosting:
 - Netlify: <https://www.netlify.com>
 - Heroku: <https://www.heroku.com>
 - Firebase: <https://firebase.google.com>

LEARNING ACTIVITIES

READINGS (LEARNING)

Weekly reading assignments will be posted to Canvas. Students are highly recommended to read the assigned materials ahead of time.

PRACTICE ACTIVITIES (PRIMARILY LEARNING)

There will be 10 practice activities in which students are asked to submit the code they wrote as they follow along with the lectures. These activities are worth 10% of the final grade.

ASSIGNMENTS (LEARNING AND ASSESSMENT)

There will be 6 programming assignments accounting for 45% of the final grade. Up to 10% of the submitted assignments' grades will be towards making sure that the code is easily readable, clearly documented, and properly indented. All assignments are GitHub assignments and come with starter code.

PROJECT (LEARNING AND ASSESSMENT)

There will be a project worth 30% of the final grade where students put everything they learned in this class together by creating a fully functional full-stack web application of their choosing.

EXAM (ASSESSMENT ONLY)

There will be one exam accounting for 15% of the final grade testing your JavaScript skills.

GRADING

SCALE

The final grade will be calculated based on the following scale with the passing grade being C or above.

A: 100 – 94

A-: <94 – 90

B+: <90 – 87

B: <87 – 84

B-: <84 – 80

C+: <80 – 77

C: <77 – 74

C-: <74 – 70

D+: <70 – 67

D: <67 – 64

D-: <64 – 60

E: <60

DISTRIBUTION

The final grade is broken down as:

10% Activities

45% Assignments

30% Project

15% Exam

POLICIES/STATEMENTS

EXTRA CREDIT

No extra credit is available beyond what is already specified above.

LATE POLICY

Quizzes and final project cannot be made up unless arrangements are made to take them ahead of time. Late assignments will be accepted with a 10% penalty per day up to 5 days to provide for unforeseen circumstances.

ACCOMMODATIONS FOR SICK STUDENTS

Students who are sick or need to be in quarantine for an extended period must stay at home and not attend in-person classes. Reasonable accommodations may be provided to these students upon request.

ALLOCATED TIME

You should anticipate spending two to three hours of study per week for each credit hour of a university course. Computer and programming classes typically require time in the upper range. An online class is not easier; it is harder. You have to do more reading and learning on your own.

TIPS FOR SUCCESS

- One cannot learn a new software and/or a programming language by just reading textbooks or watching videos; practice is critical when learning. So write as many programs as you can.
- The most effective way to get a C or above in this class is to stay current with the course topics and submit assignments on time. Your grades will be based on the degree to which you fulfilled the requirements of this course and not on you needing to get a C or above.

INCOMPLETE GRADES

An “Incomplete” may be given only when the student, having satisfactorily completed approximately 80% of the required work, is unable to complete the

classwork for a legitimate reason (such as illness or accident) and can reasonably finish on his/her own.

COURSE FEES

Course fees are designed to cover the costs of lab equipment maintenance and replacement including desktop and server computer systems and software; consumable materials and supplies; and support for lab aides, student tutors, and online instructional resources.

ACADEMIC DISHONESTY

Students are expected to maintain academic ethics and integrity in regards to performing their own work. The WSU Student Code specifically prohibits the following activities:

- a. Cheating, which includes but is not limited to the following examples:
 - i) Copying from another student's test;
 - ii) Using materials during a test not authorized by the person giving the test;
 - iii) Collaborating with any other person during a test without authorization;
 - iv) Knowingly obtaining, using, buying, selling, transporting, or soliciting in whole or in part the contents of any test without authorization of the appropriate University official;
 - v) Bribing any other person to obtain any test;
 - vi) Soliciting or receiving unauthorized information about any test;
 - vii) Substituting for another student or permitting any other person to substitute for oneself to take a test;
 - viii) Knowingly obtaining academic credit for work that is not one's own regardless of the source of the work;
 - ix) Knowingly involved in arranging fraudulent academic credit or false transcripts.
- b. Plagiarism, which is the unacknowledged (uncited) use of any other person's or group's ideas or work. This includes purchased or borrowed papers;
- c. Collusion, which is the unauthorized collaboration with another person in preparing work offered for credit;
- d. Falsification, which is the intentional and unauthorized altering or inventing of any information or citation in an academic exercise, activity, or record-keeping process;
- e. Giving, selling, or receiving unauthorized course or test information;
- f. Using any unauthorized resource or aid in the preparation or completion of any course work, exercise, or activity;

- g. Infringing on the copyright law of the United States which prohibits the making of reproductions of copyrighted material except under certain specified conditions.

School of Computing policy dictates that any verifiable evidence of student academic cheating, as defined and determined by the instructor, will result in:

1. an automatic failing grade for the class and
2. a report to the Dean of Students that will include the student's name and a description of the student's dishonest conduct.

ACCOMMODATIONS FOR DISABILITIES

Any student requiring accommodations or services due to a disability must contact Disability Services in Room 181 of the Student Services Center (or Room 256 at the Davis Campus). Disability Services can also arrange to provide course materials (including this syllabus) in alternative formats upon request. You can also call 801-626-6413 (Ogden) or 801-395-3442 (Davis) or visit <http://www.weber.edu/ssd> for more details.

DISCLAIMER

The instructor reserves the right to make changes to this syllabus, as he sees fit, anytime during this class.

SCHEDULE

The following is a tentative high-level weekly schedule of this class; it is subject to change at any time. Always refer to Canvas for more details and due dates.

#	Week of	Topic	Activity	Assignment
1	Aug 30	Introduction; NPM; JavaScript Review	#1	
2	Sep 6	React (with Netlify deployment)	#2	#1
3	Sep 13		#3	
4	Sep 20		#4	#2
5	Sep 27		#5	
6	Oct 4		#6	#3
7	Oct 11		MongoDB/Mongoose	
8	Oct 18	#7		#4
9	Oct 25	Node.js and Express (Server-rendered pages)	#8	#5
10	Nov 1	Building a REST API	#9	
11	Nov 8	Sessions and user authentication	#10	#6
12	Nov 15	Putting it all together		
13	Nov 22	Cloud-based serverless deployment (Heroku and Firebase)		
14	Nov 29	Project		
15	Dec 6			
16	Dec 13	Exam		