

Database and Backend Developer Course

essentials

#	resource
1	The current version of this syllabus
2	Welcome Video
3	What should you do the first week
4	Instructor: Ron Zacharski ron.zacharski@gmail.com , 575.680.4041
5	Experience Point Sheet
6	The UMW Database Slack Workspace

Course Description

This course provides an introduction to databases and backend web development. It covers Postgresql, Javascript, HTML/CSS, and Elasticsearch.

An asynchronous online class

This class is asynchronous meaning there is no mandatory real-time interaction. You will be working through the Inquiryum Backend Developer Fundamentals Course. You can watch the videos anytime you want. You can play them at a faster speed, you can rewatch them or pause them. You can work on the course material in 20 minute blocks throughout a day, or devote a large contiguous block of time once per week. When you need help you can use the class Slack channel to get assistance from me or your classmates.

The advantages of this approach is that it allows you great flexibility in when you want to work on the material and for how long. And, as described below under mastery learning, it allows you to work at your own pace.

Optional Zoom Study Groups

See the description of the study groups on the [What to expect page](#).

Instructor availability

I will be sitting at my laptop on the [UMW Database Slack workspace](#) on Monday through Wednesdays from 11am until 2pm ET. This means that if you message me, I will respond immediately unless I am helping another student. Excluding those times, my next level of availability is Monday through Thursday from 11am to 4pm. My average response time during that period is 15 minutes. Feel free to message me outside of those times but my response delay might be significant. Often I turn off Slack notifications at midnight. There may be times during

Friday through Sunday when I don't have cell coverage and I will not be able to receive your message.

The above hours may be subject to change if other times benefit more students. These changes will be announced in the Slack channel.

Expected Outcomes

JavaScript / Node.JS

JavaScript, often abbreviated JS, is the core programming language of the web, running both in a user's browser and on the backend. It is a high-level, just-in-time compiled language. Node.js is a runtime environment that allows JavaScript to be executed outside of browser typically on the server side.

Students should be able to

- write complex command line programs in JavaScript.
- write code that executes asynchronously
- write a scalable, maintainable, backend server.
- write code that interacts with a database server
- test and evaluate code using an industry-standard test environment.

PostgreSQL

PostgreSQL is one of the most popular free and open-source relational database management systems, and is available on wide variety of operating systems.

Students should be able to

- design and implement databases based on an English description of the data.
- understand database normalization and design database that meet the normalization requirements.
- demonstrate a good understanding of SQL including subqueries, joins, and transactions.

HTML/CSS

HTML and CSS define the user interface of a web application. Understanding both are of major importance to web developers.

Students should be able to

- Read and write HTML and CSS
- Utilize semantic tags to provide clear structure to HTML
- Manage the layout of a complex web page using CSS

Tasks and Projects

The majority of effort in the course is in working on tasks and project, which have different levels of expected knowledge and independence.

Tasks

- Closely mirror the examples shown in the course videos.
- The majority of the code is provided to you.
- Where appropriate, test code is also provided enabling you to get instant feedback.
- You are to:
 - write a small amount of code to complete the task
 - evaluate what you have done by running the tests

Tasks are evaluated on a four tier system.

- **tier 1** the task code works and all code is 100% correct. The code is formatted correctly (indentation follows programming standards, comments when needed, no debugging print statements remain, no old code lines commented.) - (100% xp)
- **tier 2** the task code works and all code is 100% correct. However, the code is not formatted correctly -(65-75% xp)
- **tier 3** most of the task code works. However, there are coding errors or missing functions. - (25-50% xp)
- **tier 4** the code mostly does not work or was not submitted. - (0% xp)

For example, if a task is worth 20 points, you might get 20 for tier 1, 15 for tier 2, 7 for tier 3, and zero for tier 4.

You can submit up to two tasks per week.

Projects

- Follow examples shown in the course videos.
- Build off of concepts and skills you learned completing prerequisite tasks.
- Project definition provides either
 - an API specification and test file
 - a video showing an example of a completed project
- You are to
 - design and create the database to support the project
 - write the Javascript code to implement the specification
 - test and evaluate you code.
 - move you code to a production environment.

Projects are evaluated as to whether the code works or not (passes the necessary tests). If it does not you will fix your code and resubmit. You can submit as many times as necessary to pass the tests. The test set that is used to evaluate your code may be larger than that given to you to aid you in development. For some projects there are optional parts for which you will get additional points.

You must complete the prerequisite tasks before starting a project and you cannot submit more than 2 projects per week.

Core Competency Task

Often when you are applying for a job, you are given a timed programming task. The core competency task is designed to help you prepare for this. The task involves writing a backend for a web service in JavaScript and PostgreSQL. You can take the test at any time (after you meet the prerequisites), but once you start you will have four hours to complete it. By *any time* I mean that you can decide to take it at 1 in the afternoon on a particular Tuesday, or 1am on a Sunday morning.

Mastery Learning

Traditional classes are time-based learning. You spend a specific amount of time on a topic and then you move on to the next topic. For example, in a traditional programming course you might cover *while* and *for* loops in week 5, take a quiz on them, and then move on to ArrayLists in week 6. Suppose you got a 75% on that quiz in week 5. That means that you did not learn 25% of the material. Then perhaps in week 10 you take a test on inheritance and get an 80% (you did not master 20% of the material). These gaps in your mastery start adding up, and eventually, in either in some future class or on the job, you hit a wall because your current task requires that you are skilled in areas that you failed to master.

This class doesn't work like that.

In contrast to time-based learning, in mastery Learning you stay on the topic until you master it. You work at your own pace. This online class is based on this approach. You stay on a topic until you master it. The lectures are a set of videos (mostly screencasts) that you can watch at anytime. If the material is easy for you, you can speed up the videos and watch them at 1.5 speed. If you find the material challenging, you can rewatch the videos, google for more information, interact with other learners on the Slack channel, or request a private Zoom conference with the instructor.

Obviously, the work-at-your-own pace approach will collide with the end of the semester and there will be some material that you will not cover. The course is designed so that the essential core information is presented first, to enable you to develop solid foundational skills with no gaps.

Mastery Learning Difficulties

This course is work at your own pace. Other courses you might be taking have fixed deadlines. So, for example, you might have a gnarly project for a programming class due this week and a big operating systems project due next week. It is likely that you will work on those projects since they have immediate deadlines and ignore working on this course. It is human nature. To give you some flexibility but to help you stay focused, there will be both recommended and hard deadlines for each project. For example, one project has a recommended deadline of the fourth week of class and a hard deadline of the eighth week.

By **hard deadline** I mean that any work received after the deadline will not receive any xp.

By **recommended deadline** I mean a suggested deadline that is designed to help you get an A in the course and equally distribute the work throughout the semester.

There is nothing to prevent you from doing all the course work in 12 weeks, get an A, and spend the remainder of the semester focused on your other classes.

The design of hard and soft deadlines

If you work at a pace set by the hard deadlines you will pass the course with a D. In a sense, the hard deadlines are provided as a safety net to have people stay on track to pass the course.

If you work at a pace set by the soft deadlines you will get an A in the class.

The course material

Order	Lesson
1	JumpStart
2	Tasks
3	Projects

Again, the class is work-at-your-own pace, but we provide a suggested schedule below.

Week-by-Week

Week	Date	Unit	Topics
1	25 Jan	Intro	Intro, MVC, RESTful, Gitlab lab
2	1 Feb	SQL	creating and inserting data into a databas. Update, alter, delete
3	8 Feb	Javascript	Intro to Javascript
4	15 Feb	JavaScript	NodeJS, using PostgreSQL with Node
5	22 Feb	Deployment	Google Cloud, Nginx reverse proxy
6	1 Mar	SQL	3NF, joins
7	8 Mar	SQL	transactions
8	15 Mar	Javascript	Intro to React and Nextjs
9	22 Mar	JavaScript	Full Stack Web Development
10	29 Mar	JavaScript	cookies
11	5 Apr	JavaScript	-
12	12 Apr	ElasticSearch	noSQL, Elastic Search
13	19 Apr	Projects	-
14	26 Apr	Projects	-

Deadlines hard and soft

Meeting the hard deadlines should put you around a C for the course. To be considered to be on time for a hard deadline, the code must be submitted **before** the date indicated.

Project	Recommended Deadline	Hard Deadline
1 Workshop v1	27 Feb	13 Mar
2 Workshop v2	6 Mar	27 Mar
3 NextJS	20 Mar	16 Apr
4 Complete Web App	27 Mar	30 Apr
5 SuperSearch	10 Apr	-
6 ElasticSearch	30 Apr	-

Required materials

Google Cloud Account (or equivalent)

Laptop

[Inquiryum's Backend Developer Fundamentals course](#)

Slack

Slack is a work chat application that many tech companies use. We are going to be using Slack in a number of ways. If you have a particular programming question you can ask it in a general channel and hopefully you will get an answer or suggestion quickly from either myself or fellow learners.

[Sign up for Slack here.](#)

Okay but how do I pass?

Grading is based on a method developed by Professor Lee Sheldon at Indiana University. It is based on obtaining experience points (XP). The number of XP determines what level you are at. You start the class at Level Zero and with 0 XP. The level you obtain at the end of the semester determines your final grade. Here is the chart:

Level	XP	Grade
Zero	0	F
One	240	D
Two	340	D+
Three	450	C-
Four	550	C
Five	650	C+
Six	750	B-
Seven	850	B
Eight	925	B+
Nine	1000	A-
Ten	1100	A

Here are the ways of earning XP:

- there will be at least 10 tasks. On average each will be worth 22xp
- there are 6 PostgreSQL projects. On average, each is worth 125xp
- there is an ElasticSearch project worth 150xp
- the core competency programming challenge is worth 150xp

Honor Code Policy

[Computer Science Department Honor Code Policy](#)

The amendments to the Computer Science Department policy are as follows (the numbers related to the numbers in the policy):

1. In the initial comment block of any submitted assignment, type *I hereby swear upon my word of honor that I have neither given nor received unauthorized aid on this work* followed by your name.
2. I am more flexible than the policy "you are *not* to communicate to others in any way about your assignment." My rule of thumb is What would a responsible adult do on the job? If you have a deadline on the job at a startup and didn't know how to do something, the responsible thing wouldn't be to sit at your workstation just getting more and more frustrated and depressed and missing the deadline. The responsible person would get whatever help was necessary to complete the task. On the other hand, a responsible person wouldn't let someone else do all the work and present it as his own. That would be a violation of this policy. (See the Slack section of the syllabus)
3. Regarding "Remember that *giving* unauthorized help violates the Honor Code just as much as *receiving* unauthorized help does." Again, I refer to the 'responsible adult' mentioned above. I would like people to help each other but yet do the work to learn the material. Sharing a complete assignment violates this point, but helping a person debug a function is fine.

4. Sadly, this contradicts what you want to do in your professional life. In your professional life, you want to post solutions to things you figured out as a way of helping people in the community. In fact, we are going to be using some material people posted in this class. However, to prevent plagiarism, you will only post your material to a private github repository. Sorry.
5. You should acknowledge the people that helped you in writing in your submission. For example, "Ann Mulkern helped me with the code to divide the dataset into training and testing sets"
6. All the rest of the conditions of the computer science policy hold as is.

Avatar names, pseudonyms, noms de plume

During the first week of class you will need to fill out the [Avatar Form](#) for your avatar name, pseudonym, whatever. This is the name that will appear on the Experience Point Google Spreadsheet that will be viewable by everyone in the class. If you wish to remain anonymous, don't share your avatar name with anyone. To further protect the anonymity of those who wish to remain anonymous, the spreadsheet will also be populated by fictitious avatar names.

Accessibility Statement

The Office of Disability Resources has been designated by the college as the primary office to guide, counsel, and assist students with disabilities. If you receive services through the Office of Disability Resources and require accommodations for this class, make an appointment with me as soon as possible to discuss your approved accommodation needs. Bring your accommodation letter, along with a copy of our class syllabus with you to the appointment. I will hold any information you share with me in strictest confidence unless you give me permission to do otherwise.

If you have not made contact with the Office of Disability Resources and have reasonable accommodation needs, (note taking assistance, extended time for tests, etc.), I will be happy to refer you. The office will require appropriate documentation of disability

Title IX Statement

University of Mary Washington faculty are committed to supporting students and upholding the University's Policy on Sexual and Gender Based Harassment and Other Forms of Interpersonal Violence. Under Title IX and this Policy, discrimination based upon sex or gender is prohibited. If you experience an incident of sex or gender based discrimination, we encourage you to report it. While you may talk to me, understand that as a "Responsible Employee" of the University, I MUST report to UMW's Title IX Coordinator what you share. If you wish to speak to someone confidentially, please contact the below confidential resources. They can connect you with support services and help you explore your options. You may also seek assistance from UMW's Title IX Coordinator. Please visit <http://diversity.umw.edu/title-ix/> to view UMW's Policy on Sexual and Gender Based Harassment and Other Forms of Interpersonal Violence and to find further information on support and resources.

Resources

Tiffany W. Oldfield, J.D.
Title IX Coordinator
Office of Title IX
Fairfax House
540-654-5656
toldfiel@umw.edu

Myranda Thomson
Title IX Deputy for Students
Area Coordinator
540-654-1184
mthomson@umw.edu

Confidential Resources

On Campus

Talley Center for Counseling Services
Lee Hall 106

Student Health Center
Lee Hall 112

Off-Campus

Empowerhouse
540-373-9373

RCASA
540-371-1666

Recording Policy

Classroom material in this course may be copied by students enrolled in the course for the personal, educational use of that student or for all students presently enrolled in the class only, and may not be further copied, distributed, published or otherwise used for any other purpose without the express written consent of the course instructor. Distribution or sale of class material is prohibited without the written permission of the instructor and other students who are recorded. Distribution without permission is a violation of copyright law. This policy is consistent with [UMW's Policy on Recording Class and Distribution of Course Materials](#).
Academic Integrity