Project descriptions

# Week5

Using the Flask framework, Mixins and Bootstrap, I added the ability to create, display, update and delete case notes to an existing CSS Grid website for a law firm. Users had to be logged into see or alter case notes. The Flask-WTF extension was used to protect against Cross-Site Request Forgery by encrypting passwords before they are stored in the database.

# Week4

Myself and seven other team members designed a relational database and created an Entity Relational Diagram (ERD), with cardinality, for an auto dealership that sold and serviced vehicles. I then built the database using the tables, primary keys, foreign keys, field names and data types specified in the ERD.

# Week3

Created an object-oriented program that displayed and calculated the return on investment for potential rental real estate investment. The program imported 15 different user inputs into dictionaries, validated the inputs and then calculated the return on investment.

# Week2

Create an object-oriented program in Python that kept tract of the parking tickets sold and the parking spaces available in a parking garage. This was a pair programming project that used dictionaries and github as the code repository.

# Week1

Wrote a program that created a responsive design multi-page law website from a mock-up using HTML5, CSS and CSS Grid. This was a team project with four members and github was used as the code repository.

Week 5

# Using Flask, recreate the NZA law website. In this project, you will need to create routes that allow "lawyers" to create accounts and be able to sign in. "Lawyers" should also be able to write "case notes" about their recent cases. Each "lawyer" should have the ability to Create, Retrieve, Update, and Delete "case notes". This project again requires collaboration of the group. The best way to do that without conflicts while using git/GitHub would be to talk about how the files should be structured before work begins. As an added challenge, deploy (host) your project on heroku. Once completed send either the GitHub project link to the assignment link or the heroku link.

Week 4

**Week 4 - ERD Assignment**

**Part 1: USE ERD to Create SQL Database**

*Part 2: Create Database for Car Dealership using SQL*

*Car Dealership*

*Create an ERD for a car dealership. The dealership sells both new and used cars, and it operates a service facility. Base your design on the following business rules:*

*A salesperson may sell many cars, but each car is sold by only one salesperson.*

*A customer may buy many cars, but each car is bought by only one customer.*

*A salesperson writes a single invoice for each car he or she sells.*

*A customer gets an invoice for each car he or she buys.*

*A customer may come in just to have his or her car serviced; that is, a customer need not buy a car to be classified as a customer.*

*When a customer takes one or more cars in for repair or service, one service ticket is written for each car.*

*The car dealership maintains a service history for each of the cars serviced. The service  records are referenced by the car’s serial number.*

*A car brought in for service can be worked on by many mechanics, and each mechanic may work on many cars.*

*A car that is serviced may or may not need parts (e.g., adjusting a carburetor or cleaning a fuel injector nozzle does not require providing new parts).*

*After the tables have been created, use SQL to add data to the database*

Each person will need to create their own ERD diagram using Lucidchart and create their own SQL database.

Week 3

# **Object Oriented Programming - Calculation of Rental Income (individual project)**

Using Visual Studio Code/Jupyter Notebook, and Object Oriented Programming create a program that will calculate the Return on Investment(ROI) for a rental property. Attached to the assignment is a youtube video which explains how to calculate ROI and is the method you will use in your programs. This project will be completed individually, though you can feel free to share ideas with your fellow students. Once completed, commit the project to github and submit the link to this assignment.

[[](https://www.youtube.com/watch?v=T_7vhsSBi7c)](https://www.youtube.com/watch?v=T_7vhsSBi7c" \o "Calculating Numbers on a Rental Property [Using The Four Square Method!]" \t "_blank)

[Calculating Numbers on a Rental Property [Using The Four Square Method!]](https://www.youtube.com/watch?v=T_7vhsSBi7c" \o "Calculating Numbers on a Rental Property [Using The Four Square Method!]" \t "_blank)

[YouTube video](https://www.youtube.com/watch?v=T_7vhsSBi7c" \o "Calculating Numbers on a Rental Property [Using The Four Square Method!]" \t "_blank)

# Week2 Object Oriented Parking Garage - Pair programming

Due Jul 27, 10:30 AM

Your assignment for today is to create a parking garage class to get more familiar with Object Oriented Programming(OOP). This project will be a pair programming project. This means, that one person(The driver) will code the project while the other person(The navigator) brainstorms and guides to a working solution. Each of you should share/switch these roles every 30mins-1hr Make sure to: create a github repository, share the link, fork the code, clone it and begin working. When code has been updated, you will need to pull down the changes. Here's an article on doing so -- <https://stackoverflow.com/questions/3903817/pull-new-updates-from-original-github-repository-into-forked-github-repository> Your parking gargage class should have the following methods: - takeTicket - This should decrease the amount of tickets available by 1 - This should decrease the amount of parkingSpaces available by 1 - payForParking - Display an input that waits for an amount from the user and store it in a variable - If the payment variable is not empty then -> display a message to the user that their ticket has been paid and they have 15mins to leave - This should update the "currentTicket" dictionary key "paid" to True -leaveGarage - If the ticket has been paid, display a message of "Thank You, have a nice day" - If the ticket has not been paid, display an input prompt for payment - Once paid, display message "Thank you, have a nice day!" - Update parkingSpaces list to increase by 1 - Update tickets list to increase by 1 You will need a few attributes as well: - tickets -> list - parkingSpaces -> list - currentTicket -> dictionary When the project is completed, commit the final changes, sync all pull requests, and each member should submit their respective GitHub links(though the code in each should be the same)

# Week1 Law site

**Wrote a program that created a responsive design multi-page law website from a mock-up using HTML5, CSS and CSS Grid. This was a team project with four members and github was used as the code repository.**

In this project, you will be tasked with creating a new website for a “law firm”. This website’s mock design is also included in this file. This will be a group project in which each person should share an equal role in the finished product. One person from the group (Original Team Member) will need to create the starter files in their Github Profile. From there each person will need to fork & clone the repository and begin working on their portion of the product. When each of you have completed your parts of the project, commit them back to github and submit a pull request to the original team member’s repository. Each person will then need to submit the original team member’s github link to google classroom. Also, to help with workflow management, you may use github's "projects" tab to seperate and manage what has been done. You will need to create 3 boards which are "Work to be done", "In-Progress", and "Completed". TASK: - Create Website using CSS3. ChallengeCreate a responsive design using Bootstrap or CSS Grid \*\* On the following monday, each group will be required to present a 5-10min presentation of the project. Including any learning experiences along the way and how those were handled \*\* IMPORTANT NOTES : EACH member of the team must commit work. Also, a README file will need to show each of your names and what file/code you completed. The project is not considered complete until it has been checked in (uploaded) to your GitHub account and the original team member's link posted to the Google Classroom assignment(each person individually). All photos can be placeholder images HELPFUL HINT: The original repository holder should create all files and split the work based on the team's agreement. Splitting up the work inside of the CSS and HTML files is recommended! \*\* Please check the rubric for specific requirements\*\*