

***Dev Duel***

- Created a server that interfaces with Github's API using Express, to fetch Github user data and compare the data of two users
- Designed and implemented a webpage for client interaction with the server, using HTML, CSS, JavaScript, and jQuery

*Environment:* VS Code, JavaScript, HTML, CSS

[Github Link](#)

***Quizler***

- Built a command line application for creating and taking quizzes using functional programming techniques and the external libraries Inquirer and Vorpai

*Environment:* VS Code, JavaScript

[Github Link](#)

***Copy Page***

- Utilized HTML and CSS to recreate a web page design from Apple's website

*Environment:* VS Code, HTML, CSS

[Github Link](#)

***Declarative CalculateTotal***

- Implemented a JavaScript program to calculate the total price of items, including the tax on each item, using declarative programming
- Wrote type signatures for lambda functions to clarify their parameter and return types

*Environment:* VS Code, JavaScript

[Github Link](#)

***Imperative CalculateTotal***

- Wrote a JavaScript function to calculate the total price of items, including the tax on each item, using imperative programming

*Environment:* VS Code, JavaScript

[Github Link](#)

***Twitter API***

- Implemented a RESTful Web Service that models the basic functionality of Twitter's API
- Designed an ERD and generated a PostgreSQL database with JPA, including embedded objects to maintain the mental model of Java objects
- Architected the API following Onion principles for separation of concerns using Spring Boot, Lombok, and Mapstruct
- Collaborated with and assisted team members through Slack and Zoom to resolve merge conflicts through Git, and used the Kanban project board from Github to organize responsibilities

*Environment:* Eclipse, Spring Boot, JPA, Git, Github

[Github Link](#)

### **Quiz API**

- Built a RESTful Web Service to simulate the backend for a quiz-like application
- Generated a PostgreSQL database with JPA
- Utilized Spring Boot, Lombok, and Mapstruct to build the API following Onion principles for separation of concerns

*Environment:* Eclipse, Java, Spring Boot, JPA, PostgreSQL

[Github Link](#)

### **OrgChart**

- Implemented a Java data structure to store a company's organizational hierarchy of employees and managers, using Java Collections

*Environment:* Eclipse, Java

[Github Link](#)

### **Schema Design**

- Built a PostgreSQL database schema using PGAdmin to represent people, locations, and interests
- Wrote an SQL JOIN statement to connect people who are in the same specified location and share a specified interest

*Environment:* PGAdmin, PostgreSQL, SQL

[Github Link](#)

### **Fibonacci/FizzBuzz**

- Created Java methods to calculate elements and index-based slices of the Fibonacci sequence
- Generated a Java method to determine whether each number in an array is divisible by three, five, or both, and print a message accordingly for each number

*Environment:* Eclipse, Java

[Github Link](#)

### **Rationals**

- Implemented Java classes to represent rational numbers, and methods to perform mathematical operations on the rationals, including conversion of a rational to a simplified rational

*Environment:* Eclipse, Java

[Github Link](#)

**University of North Carolina at Chapel Hill - Chapel Hill, NC**

*Computer Science Student*

**08/16 - 05/20**

### **MiniJava Compiler**

- Developed a program in Java capable of compiling programs written using a subset of the Java language into assembly language
- Studied in-depth the theory of compilation and various implementations of compilers, and designed sets of rules to represent the syntax of the MiniJava language
- Implemented scanning, parsing, identification, type checking and code generation through the use of Abstract Syntax Trees

*Environment:* Eclipse, Java

[Github Link](#)