



Ollscoil
Teicneolaíochta
an Atlantaigh

Atlantic
Technological
University

Profession Practice in IT

DRIVING THEORY TEST WEBSITE

Group Members:

Conor Murphy & Rohan Sikder

Group Supervisor:

Martin Kenirons

GitHub Repository:

<https://github.com/ConorPadraigMurphy/ProfessionalPractice>

Computing in Software Development

18/02/2023

Contents

| | |
|---|----|
| Introduction | 2 |
| GitHub Repository | 2 |
| Architectural Overview of the System | 2 |
| What is MERN? | 3 |
| React.js | 3 |
| Express.js & Node.js | 3 |
| MongoDB | 3 |
| Functional Breakdown of our Project | 3 |
| Goals | 3 |
| Objectives | 3 |
| Data Structures and Algorithms | 4 |
| FISHER-YATES SHUFFLE (FISHER-YATES_SHUFFLE, 2022) | 4 |
| Project Plan and Design Methodology | 5 |
| Our Kanban board history | 6 |
| What is a Kanban Board? | 11 |
| Agile Methodology | 11 |
| What is Agile methodology? | 11 |
| Possible Technologies | 11 |
| MEAN | 12 |
| MEVN | 12 |
| LAMP | 12 |
| Why Choose MERN? | 12 |
| Database Design | 12 |
| Screen Layout | 15 |
| Testing Plans | 18 |
| Postman Testing: | 18 |
| Selenium Testing: | 19 |
| Deployment | 21 |
| Limitations and Known Bugs | 22 |

Introduction

The following documentation is based on the Profession Practice in IT project we have received in our course Computing in Software Development. I, Conor Murphy(G00399177) and Rohan Sikder(G00389052) have been paired in a team for this assignment. For this project we have decided to create a website that users can use to study and practice the driver's theory exam. To accomplish this, we will be using MERN. Our requirements for our website are to have two main features. These are being able to go through the questions and study the answers as well as being tested on the study material for the theory test. Of course, over time we may develop more minor details as the project develops, but for now we have our main requirements for a functioning Drivers Theory website.

GitHub Repository

<https://github.com/ConorPadraigMurphy/ProfessionalPractice>

Architectural Overview of the System

As mentioned above for our project we will be using the MERN architecture, MERN stands for MongoDB, Express, React and Node after the four technologies that make up the stack. The MERN architecture that we are using enables us to easily construct a three-tier architecture, also known as the front end, backend, and the database. We do all this by using JavaScript. We have decided to use MERN seeing as we have previous knowledge of the architecture thanks to modules that we have had during our time in Atlantic Technological University. Using this knowledge, it has given us the know-how on how we can build a database to store our data, as well as develop a frontend and backend to utilize our data.

WHAT IS MERN?

React.js

The top level of the MERN stack is React.js, this is a JavaScript framework used for creating client-side applications in HTML. React lets us build complex interfaces using simple components then connect them to our data using our back-end server and render them. React is great for handling data-driven interfaces without the use of a huge amount of code and includes fantastic support for error handling, events, lists, forms, etc.

Express.js & Node.js

The next level of MERN is the Express server-side framework that runs inside a node.js server. Express is great for URL routing and dealing with HTTP requests and HTTP responses. By making use of GETs and POSTs on our React.js frontend which gives us the power to allow us to access and/or update data in our mongo database.

MongoDB

This level of the application is where we store our data. Mongo stores this data in JSON-like documents, meaning that, if need be, fields can vary from document to document as well as that we can change the data structure if needed.

FUNCTIONAL BREAKDOWN OF OUR PROJECT

At the beginning of our project, it was important we laid out the requirements we needed for our website to have the proper functionality. In this section of the documentation, I will be discussing our Goals, Objectives, and Project deliverables.

Goals

The goal of our project is to make a functional website that users can use to study for their driver's theory test as well as do mock exams in preparation for their actual drivers theory test. Where users can take a mock theory test as well as study for the questions that come up that may come up on the test.

Objectives

We have two main objectives for our project. One is to develop a theory test where users can test themselves on drivers' theory questions. We can do this by developing a mongo database that holds the questions as well as the answers to each question with one of which being marked correct so that we can compare it against what answer the user has picked and mark it as correct or incorrect. Once the user finishes the theory test, we want to show the sure a result e.g., X/20. We can do this by incrementing a score integer every time a user gets a question correct.

Data Structures and Algorithms

FISHER-YATES SHUFFLE (FISHER-YATES_SHUFFLE, 2022)

An algorithm we implemented in our project was the Fisher-Yates Shuffle, this algorithm was used to shuffle the questions given to the user, how this algorithm works by getting all the index of elements of the array from the database and firstly getting the current index which is initially the length of the array and swapping with a random index then not using both of those indexes again, having two arrays keeping track of this. This is repeated till the length of the index is zero. Pseudocode and implementation snippet below:

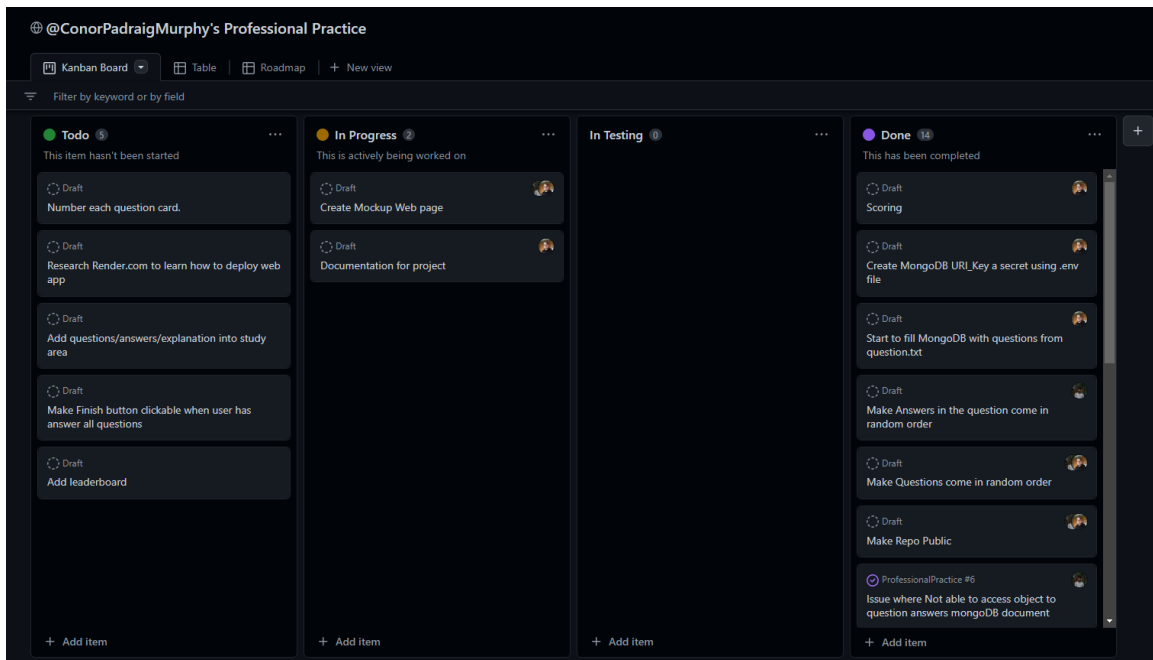
```
-- To shuffle an array a of n elements (indices 0..n-1):  
for i from n-1 downto 1 do  
    j ← random integer such that 0 ≤ j ≤ i  
    exchange a[j] and a[i]
```

```
1 shuffle(array) {  
2     //Gets ArrayIndex length  
3     let currentIndex = array.length, randomIndex;  
4     //console.log(currentIndex);  
5  
6     // Loops While there remain elements to shuffle.  
7     while (currentIndex !== 0) {  
8  
9         // Pick a remaining element.  
10        //Finding a random element  
11        randomIndex = Math.floor(Math.random() * currentIndex);  
12        currentIndex--;  
13  
14        // And swap it with the current element.  
15        [array[currentIndex], array[randomIndex]] = [  
16            array[randomIndex], array[currentIndex]];  
17    }  
18  
19    //Returns shuffled array  
20    return array;  
21 }
```

Project Plan and Design Methodology

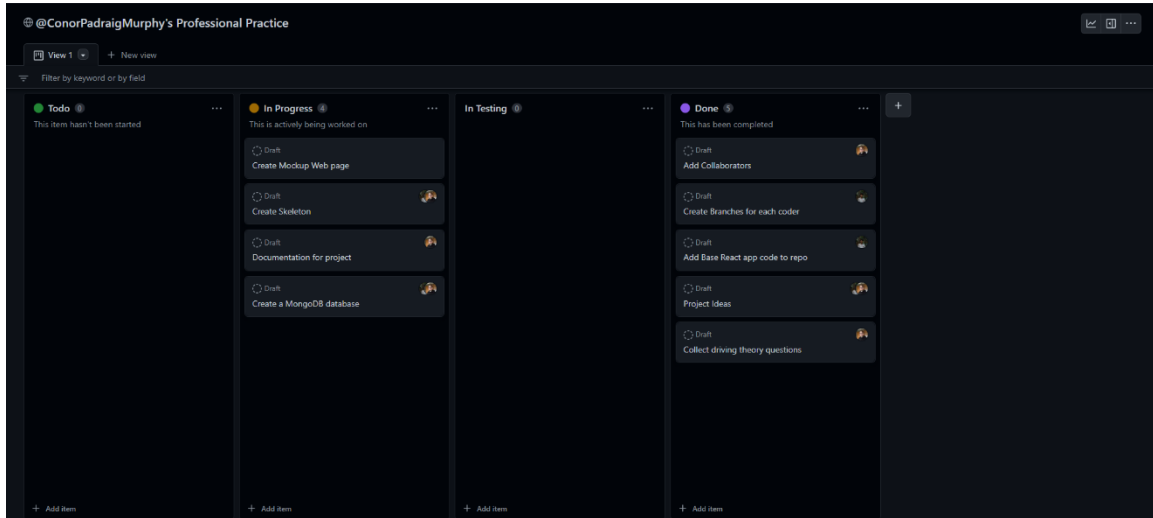
In past modules we have used various services to track project progress and to track issues such as Jira and Microsoft Projects. But for our project we have decided to use GitHub Projects which allows use to tie our GitHub repository directly to our GitHub projects Kanban board.

Using GitHub projects allows us to add tasks to a board where we can see what tasks we have To-do, what tasks we have In Progress, what aspects we need to test and what has been finished.

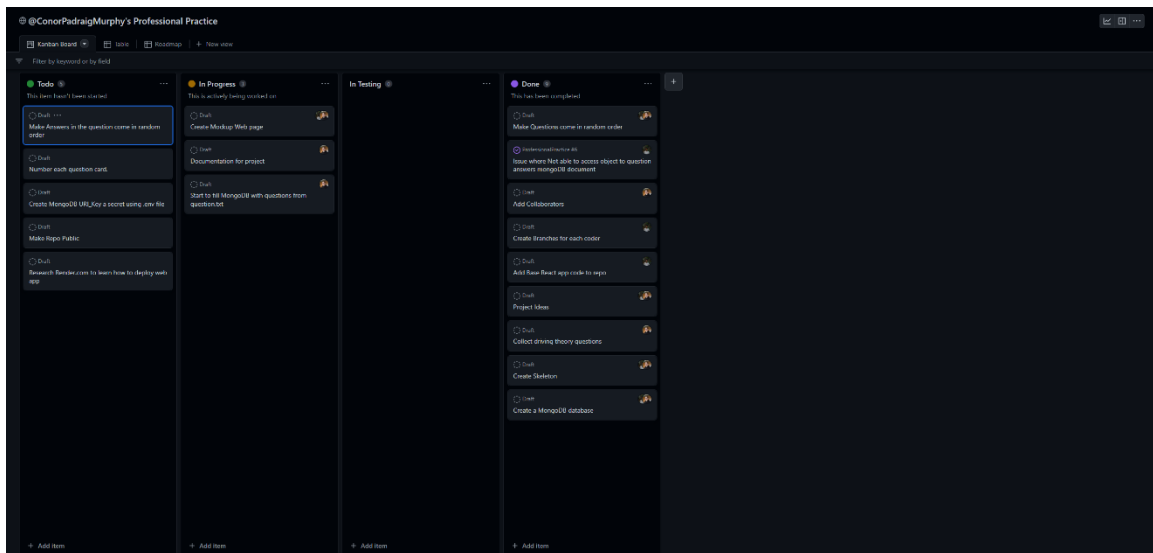


Picture 1: Example Kanban board

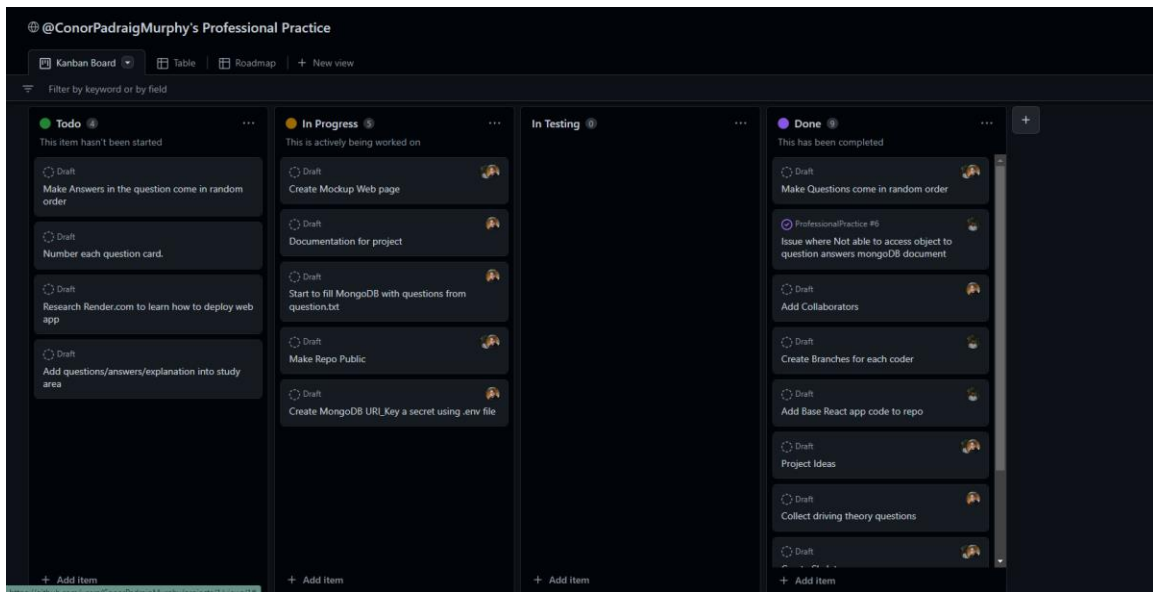
Our Kanban board history



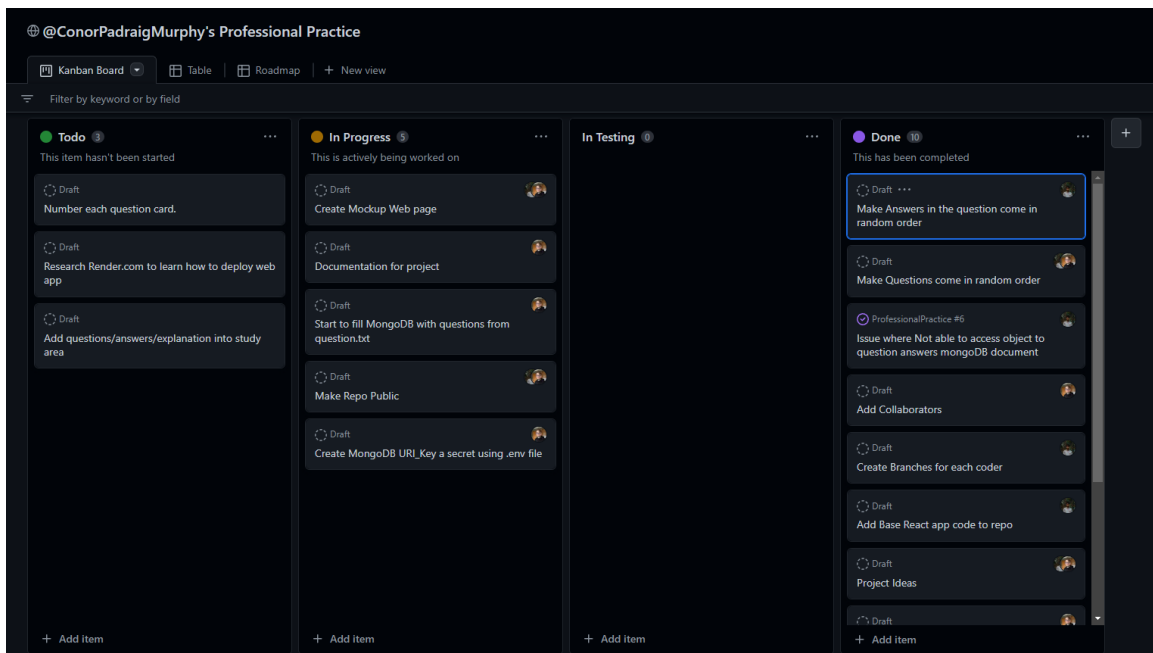
Picture 2 : 22/02/2023



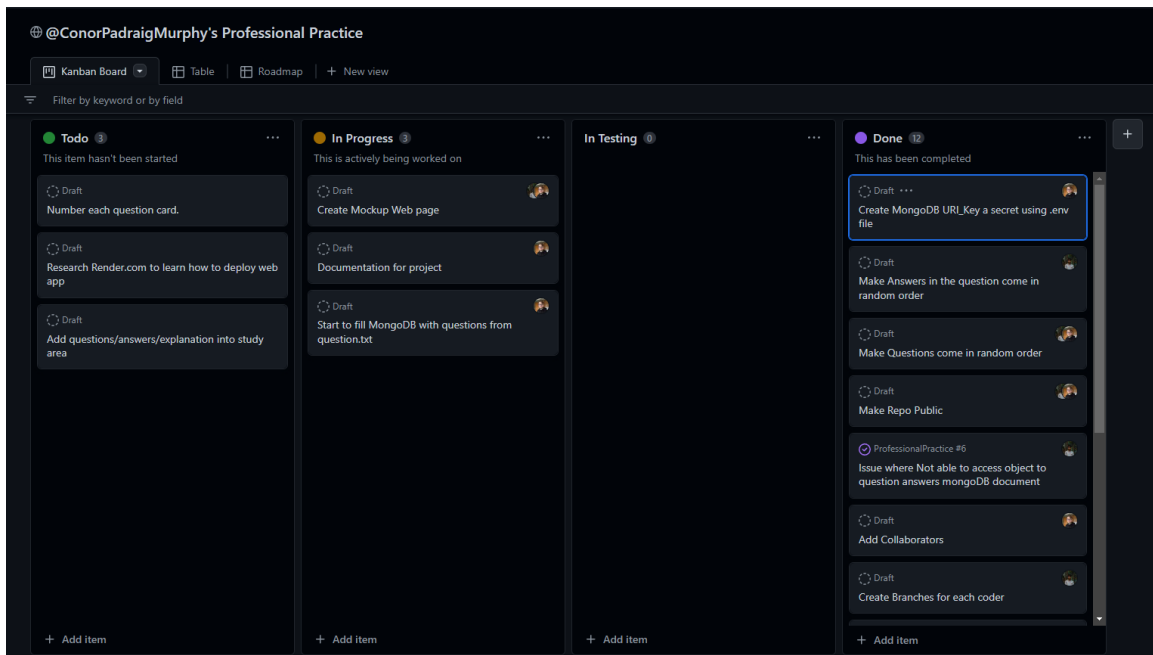
Picture 3: 09/03/2023



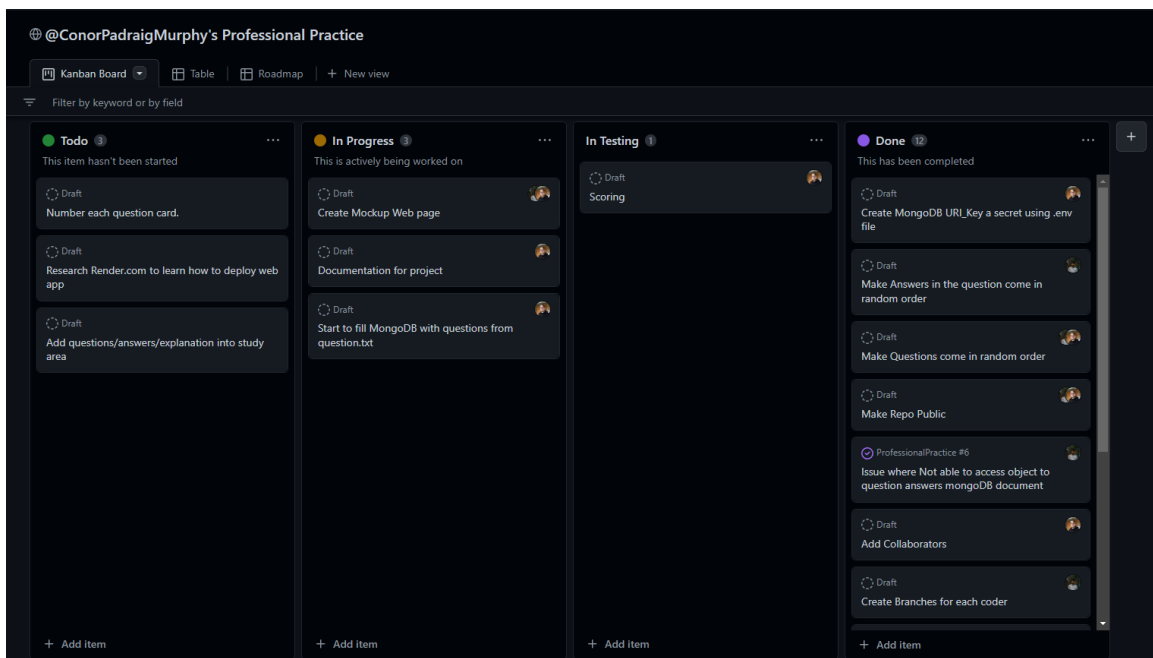
Picture 4: 11/03/2023



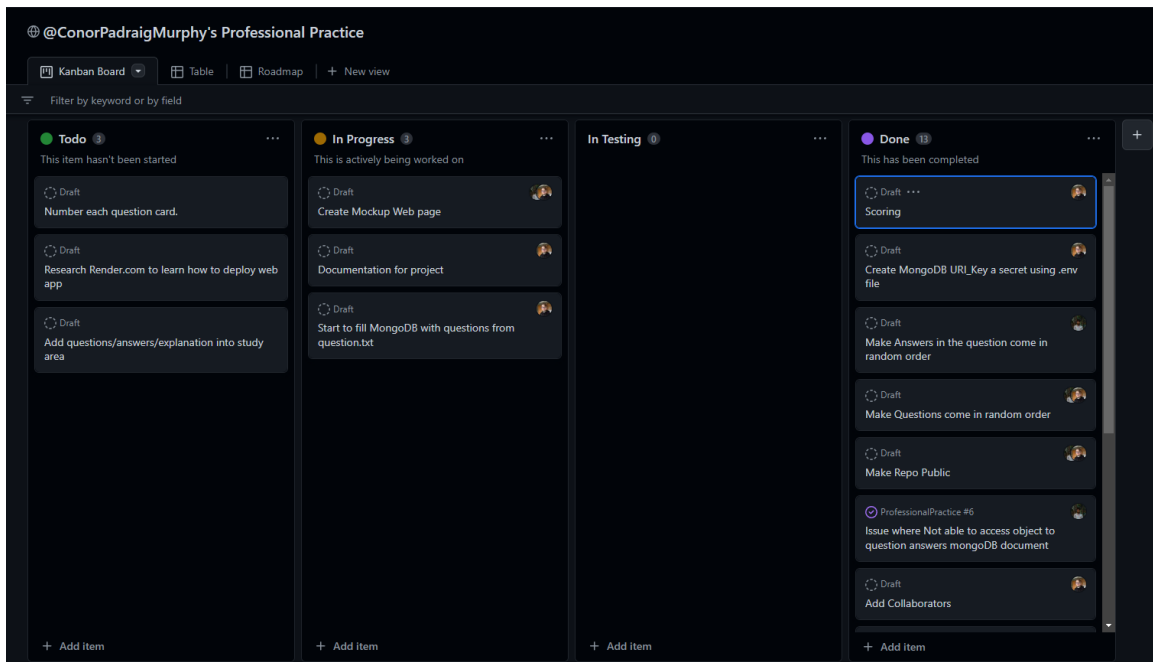
Picture 5: 11/03/2023 - changes 1



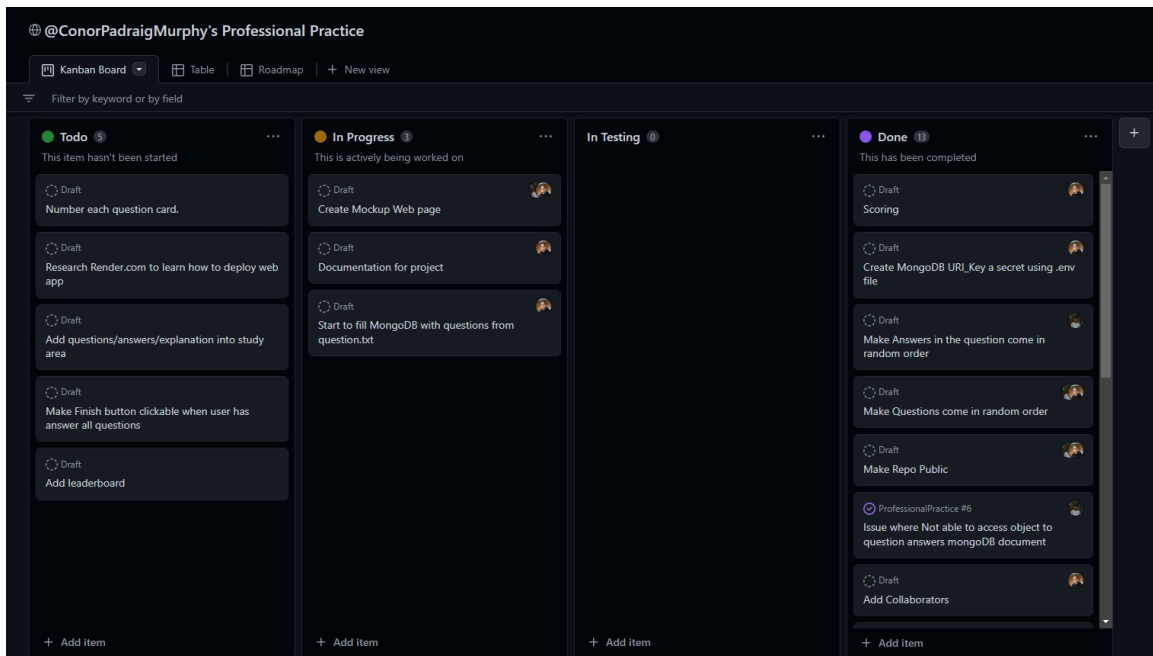
Picture 6: 11/03/2023 - changes 2



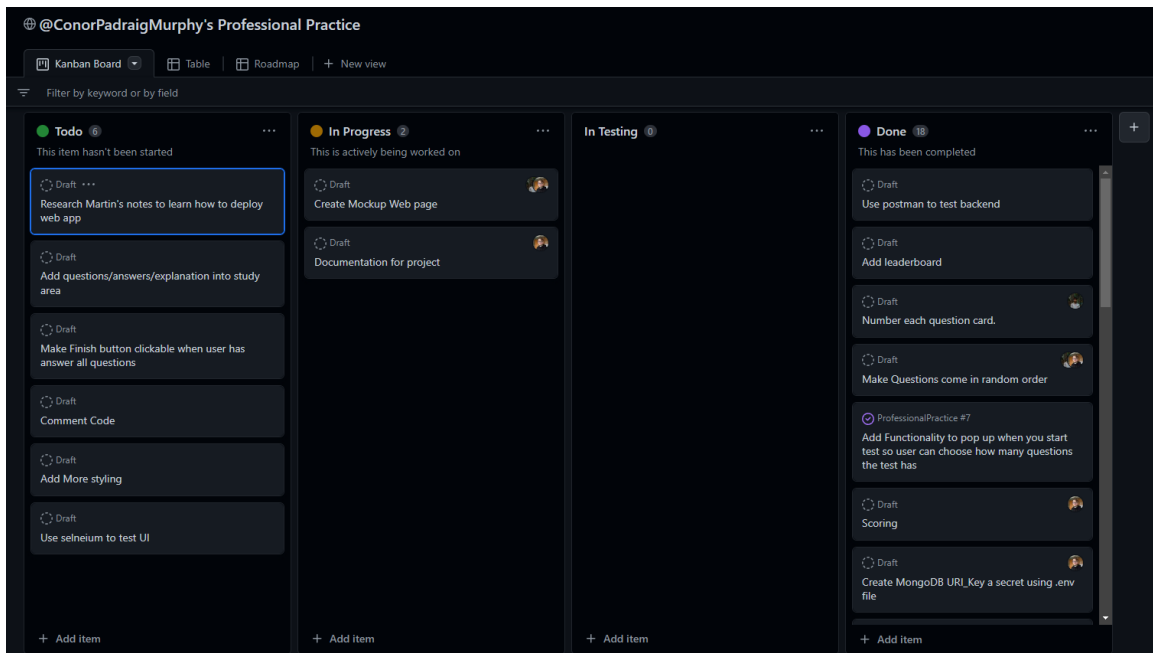
Picture 7: 11/03/2023 - changes 3



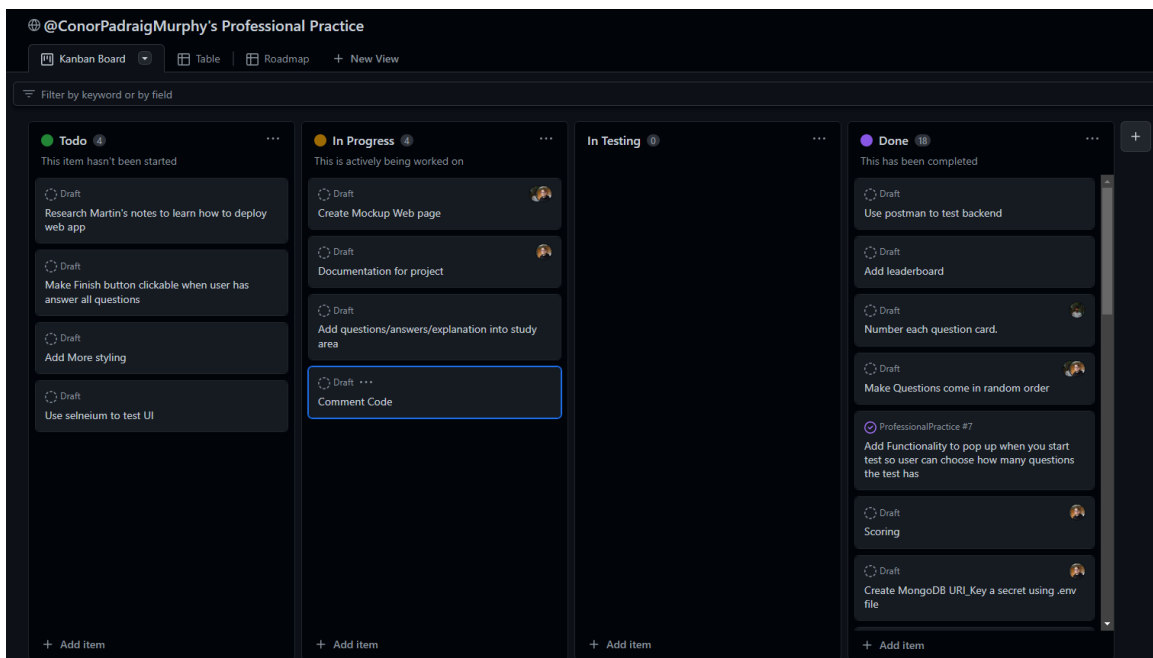
Picture 8: 11/03/2023 - changes 4



Picture 9: 19/03/2023



Picture 10: 02/04/2023



WHAT IS A KANBAN BOARD?

A Kanban board is management tool regularly used in the agile methodology to help developers visualize the work that is being done, what has been done as well as what has yet to be started. It can be used to maximize workflow as well as help us visualize what needs to be done on a daily/weekly basis.

The Kanban board uses cards and columns to allow us to track out tasks and issues as you can see in the above picture we have four columns, To-do, In Progress, In Testing and Done. And in each of the columns we have cards such as in the In Progress column we have a card titled “Create Mockup Webpage” then within this card we would have a description of what needs to be done.

AGILE METHODOLOGY

What is Agile methodology?

The agile methodology is an iterative approach to keeping project management organized and on track so that we meet the requirements and goals we set for ourselves. Rather than doing everything in one large development cycle we use agile to deliver development updates in small increments. Using a Kanban board is one of the agile frameworks we used for our project which allowed us to increment our workflow allowing us not to fall behind when it came to working on our Professional Practice in IT project as well as projects for other modules.

The agile methodology was developed in response to the waterfall methodology. And so, because of this we have a more organized, fluid form of workflow while managing our project. It specifically designed because a software development plan or the technology used to develop can change drastically during the time it takes to develop a project. Obviously with the timeline of our project we don't need to worry too much about technology changing but as it has been a methodology, we have used frequently it was an obvious choice for us.

Possible Technologies

So, for our project we used MERN, which as detailed above is a combination of Mongo, Express, React and Node.js. But of course, there are other technologies available to make an application like ours. But what are they and how do they compare to MERN?

MEAN

First, we have the MEAN stack. It is like the MERN stack but instead of using React.js like we do in MERN we use Angular instead. Angular of course is a popular front-end framework. Angular was developed by Google engineers and is still supported by them. Angular is a great framework and helps use address a variety of issues we can run into when using SPA's (Single page applications). Like MERN, MEAN also uses Express.js, Mongo and Node.js which makes it very easy for developers to switch between the two.

MEVN

MEVN, again is like the previous stacks I have talked about (MERN & MEAN). But rather than using React and Angular it uses Vue on the frontend. Vue.js is an open-source model used for building SPA's. It was created by Evan You and is maintained by him and his team.

LAMP

LAMP stack is made up of Linux, Apache, MySQL, and PHP. LAMP was one of the first open-source stacks made and to be available to developers worldwide and has remained one of the popular stacks to use since its creation. LAMP runs very efficiently on operating systems and is great at handling dynamic pages and applications.

WHY CHOOSE MERN?

The reason we chose MERN over any of the above stacks is because both of us had a good knowledge of how MERN worked as we had to use it for a project in our previous semester of year three. Mongo was the perfect database for use to use because none of our information that we had to sort through or would contain anything that we would need a specific or complex query for as we were just displaying questions and answers. As well as displaying names and cores for our leader board. When it came to designing the webpage React was the perfect framework to make our drivers theory test as React is great for creating complex or simple applications that connect backing into the backend and render it, which is all we needed for our application.

Database Design

For our database, like we have mentioned above. We decided to use Mongo. Mongo stores data using JSON like documents. Meaning that when it comes to the fields in our database they can vary from entry to entry.

For our database we initially knew we would need to store the question, and four answers that would be displayed in our application so that the user can choose an answer. In addition, we added a field in our database for the correct answer. So that when the user chose an answer, we could compare what they chose against the correct answer we had stored for that question.

```
_id: ObjectId('6403bf45aaa631d5eld30ed1')
question: "What should a driver do where a section of road ahead has a shallow fl..."
answer1: "Increase speed and use a higher gear."
answer2: "Decrease speed and use a higher gear."
answer3: "Drive in a low gear as slowly as possible keeping the revs high."
answer4: "Use a high gear and maintain the same speed keeping the engine revs lo..."
correctAnswer: "Drive in a low gear as slowly as possible keeping the revs high."
```

```
_id: ObjectId('64049966e7661d8bd8ae7613')
question: "If a vehicle is driven with low oil pressure, what effect does this ha..."
answer1: "It increases fuel consumption."
answer2: "It increases fuel efficiency."
answer3: "It increases wear and tear on the engine."
answer4: "It increases exhaust emissions."
correctAnswer: "It increases wear and tear on the engine."
```

```
_id: ObjectId('6404a35fe7661d8bd8ae7625')
question: "What should a driver do if they wish to drive across a busy road and t..."
answer1: "Sound the horn and proceed slowly to cross the road."
answer2: "Take good observation, wait for a clear break in the traffic and proce..."
answer3: "Slowly force their way across using hazard warning lights if necessary..."
answer4: "Switch on headlights and use hand signals to show other drivers their ..."
correctAnswer: "Take good observation, wait for a clear break in the traffic and proce..."
explanation: "Always read the road ahead and be prepared to react to any traffic sit..."
```

```
_id: ObjectId('640cc2c327f29a6f6cd63c47')
question: "What should a driver do if a system warning light lights up on the das..."
answer1: "Continue to drive and listen for any unusual sounds."
answer2: "Continue for a distance to see if the light goes out."
answer3: "Stop and check the problem."
answer4: "Carry on and check the problem later."
correctAnswer: "Stop and check the problem."
explanation: "If a system warning light lights up while you are driving, you should ..."
```

Picture 10: Questions database entry example

With the image above you can see how we have structured the fields in your database. All of our data types in our questions document are “String” because all we are storing are question and answers and the id is using mongos default for the id, “ObjectId”.

```
_id: ObjectId('640d247189d8f2705f1e5df8')
user: "Rohan"
score: 7
__v: 0
```

```
_id: ObjectId('640d2f2d89d8f2705f1e5e26')
user: "John"
score: 0
__v: 0
```

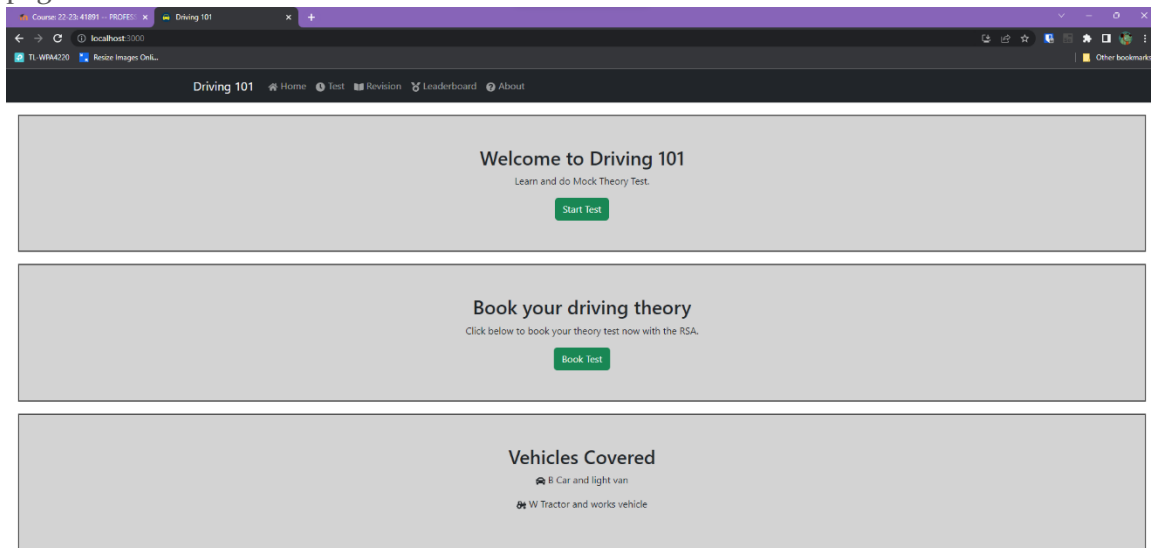
Picture 11: Leaderboard example entry

We also have a leaderboard system with our website that allows users to enter their name at the end of the test and like an old arcade game displays your name and score for others to see when they visit the website. To do this we store the users score in another document separate from our main question database and store the users chosen name. You can see in the image above how we have our entries structured. Just like the questions document the ID is set as “ObjectId”, for the name we have it set as “String” and for the score itself we are using datatype “Int32”

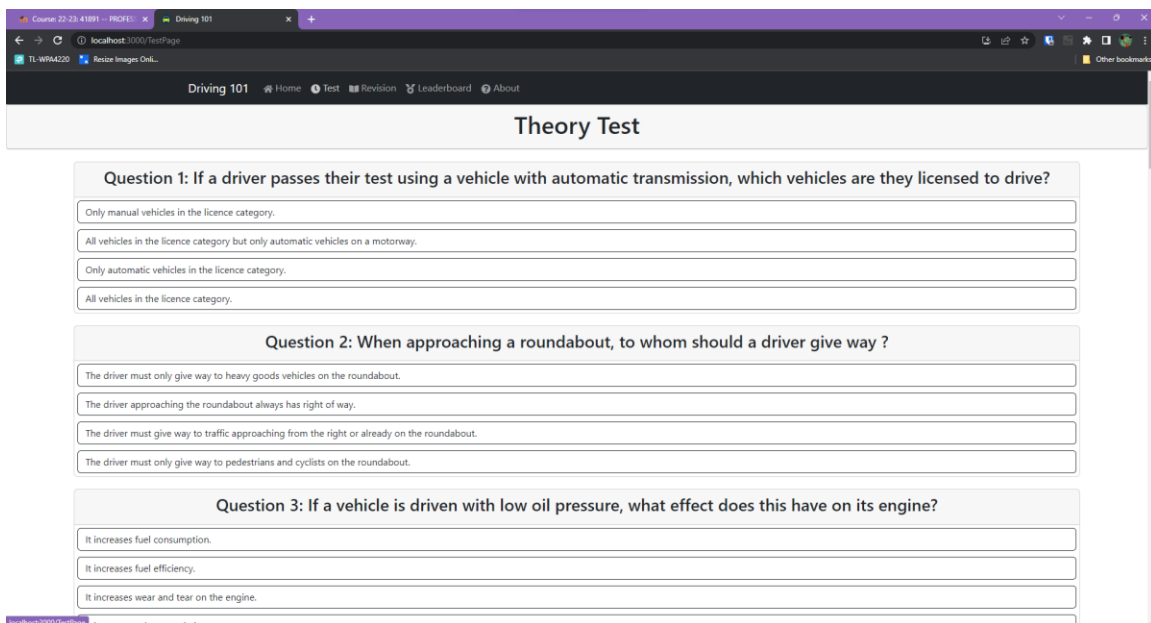
We have made sure for our database we only have the absolute essentials required so we are not storing any information that isn't required.

Screen Layout

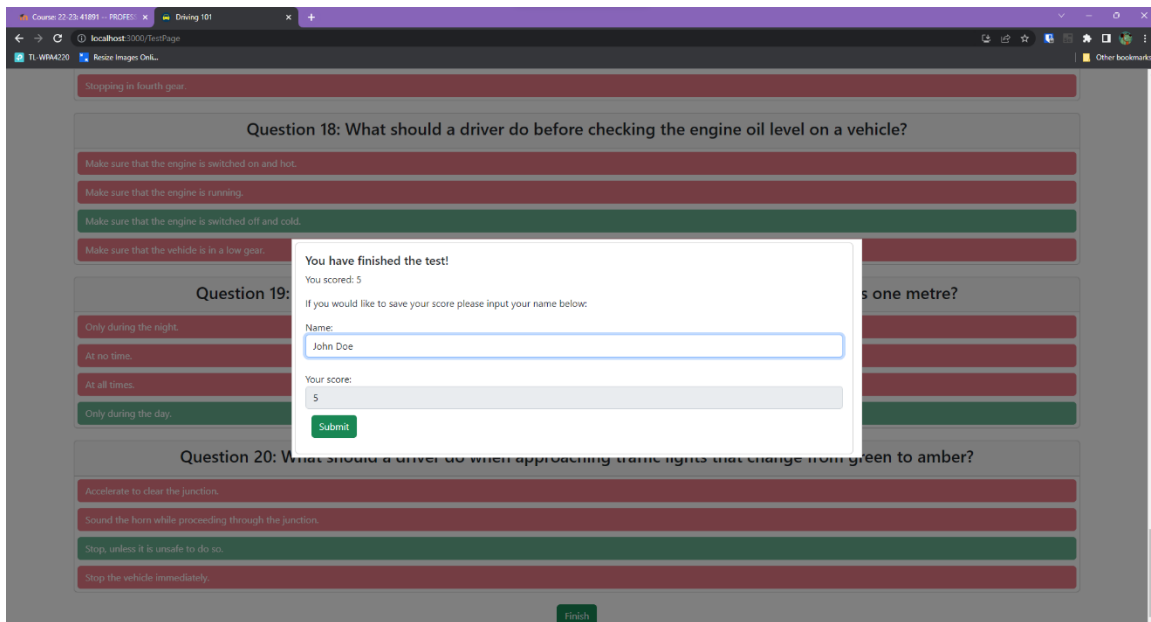
From our home page the user can see we have a navbar where they can navigate back to our home page or navigate to our Test page, Revision page, Leaderboard page and About page.



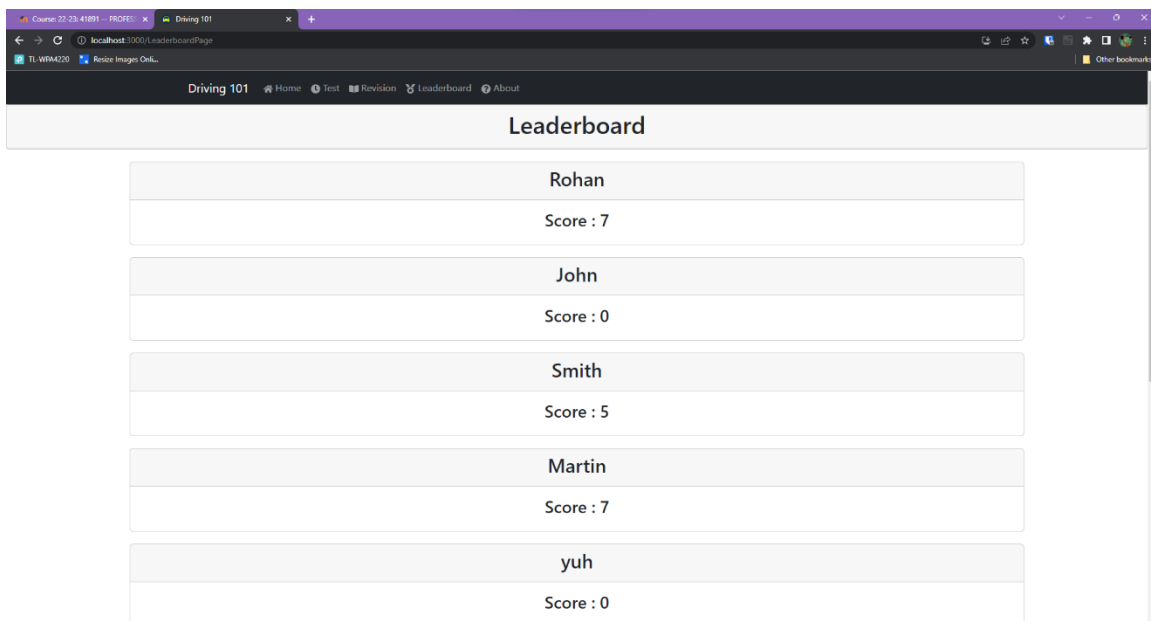
Selecting our Test page will bring you to our default test page where you can be tested on twenty theory questions where afterwards of you click finish you will be prompted to enter your name to post your score and name to our leaderboard.



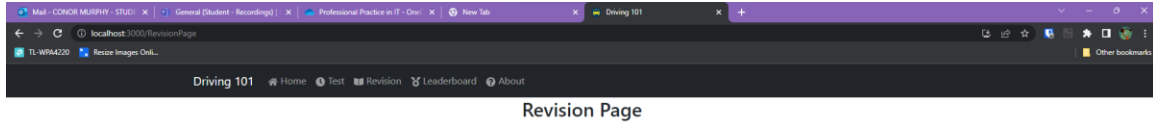
In this picture you can see the user has finished their questions and clicked finish and has been prompted to enter their name so that their score can be posted to our leaderboard.



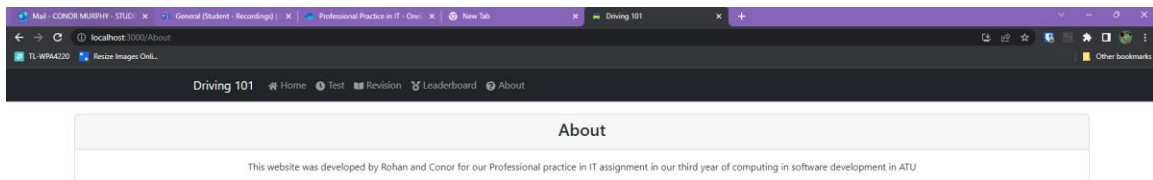
The following image is what the user will see when they navigate to our leaderboard page or when a user finish inputting their name after their test to post their score to the leaderboard.



When the user navigates to the revision page, they will be shown all the questions they could come up on the theory test and will also be given explanations to the answer to the question.



When the user navigates to the about page of our website, they are shown an about section letting them know who made the website as well as why it was made.



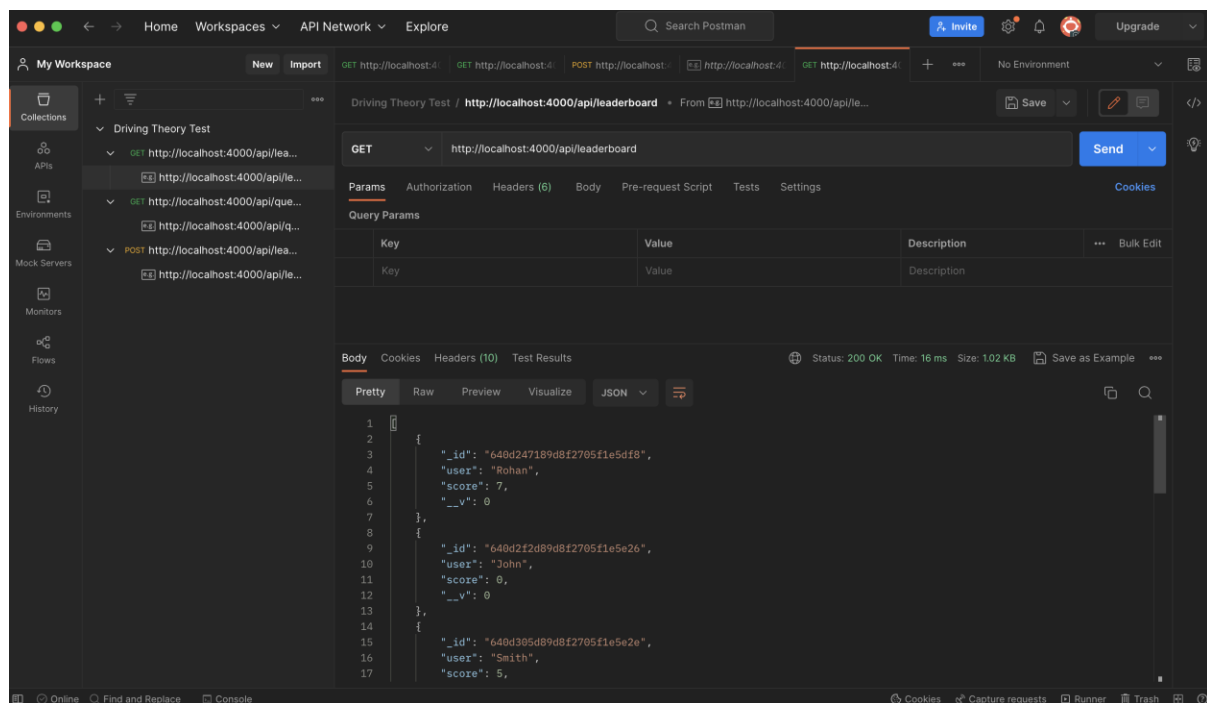
Testing Plans

As Testing is critical for any software development, we have chosen to use Postman and Selenium to test our Product. We will use Postman to test the backend and Selenium to test the frontend, these tests can be run after a developer has added code to the project.

POSTMAN TESTING:

Postman is used to test the backend for our own server API's, this includes our two GET requests to get Leaderboard and Test questions from the Mongo Database, and our POST request which posts the user's details to the Mongo database. Postman allows all tests to store as a collection in JSON format. Allowing the developer to run the collection and all the tests will be carried out.

Below is a screenshot of Postman, On the left we can see The Driving Theory Test Collection with two get requests at <http://localhost:4000/api/leaderboard> and <http://localhost:4000/api/questions> when send is clicked the GET Request is actioned and JSON data is retrieved along with Status: 200 OK, and other details in the body section, This is the same for the POST request except Query params are entered as the Database schema to allow test if Params are POSTED correctly to database.



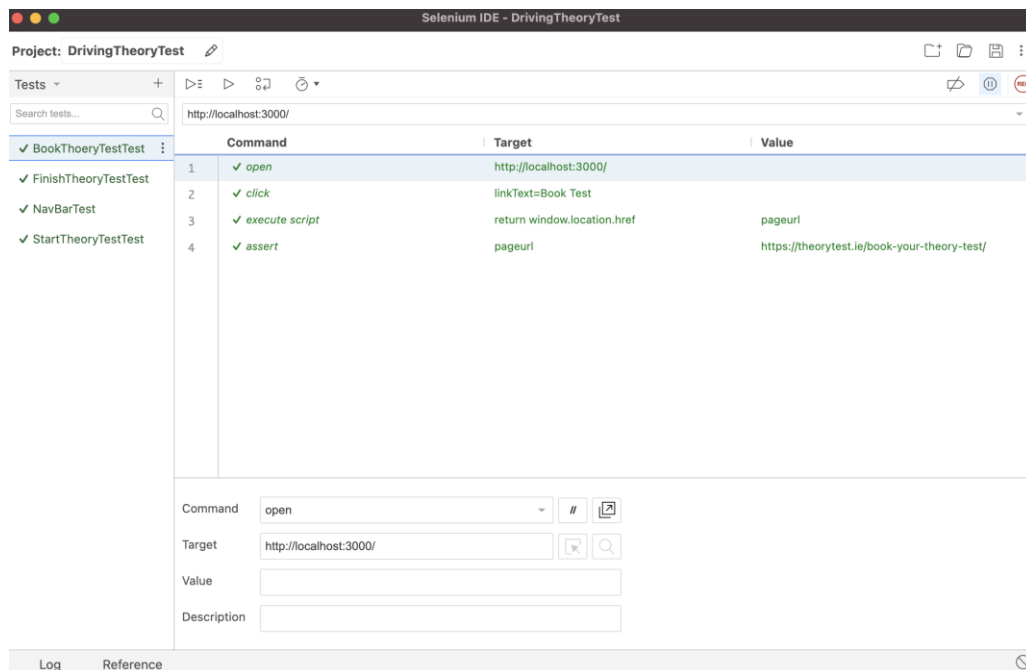
SELENIUM TESTING:

To Test the front-end of our website we used Selenium and the Selenium chrome IDE, this tests the functionality of our front-end checking if all buttons work and bring you/function as intended. Selenium can easily show where tests failed so bugs can be rectified at the source. As front-end testing can be tedious selenium makes this efficient as you can create tests to create step by step interactions on your web page to simulate a person navigating the website. Selenium makes testing front end very easy as current tests do not have to be changed unless functionality has changed, and developers and testers can create tests as they develop business requirements/functionality.

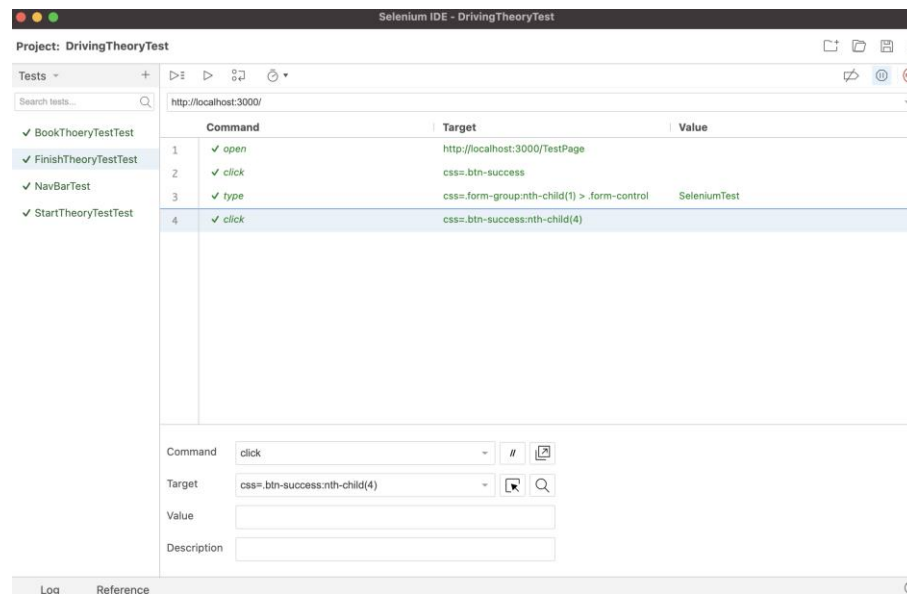
Below are screenshots of the Selenium IDE:

On the left you can see 4 Tests, BookTheoryTest, FinishTheoryTest, NavBarTest, and StartTheoryTest.

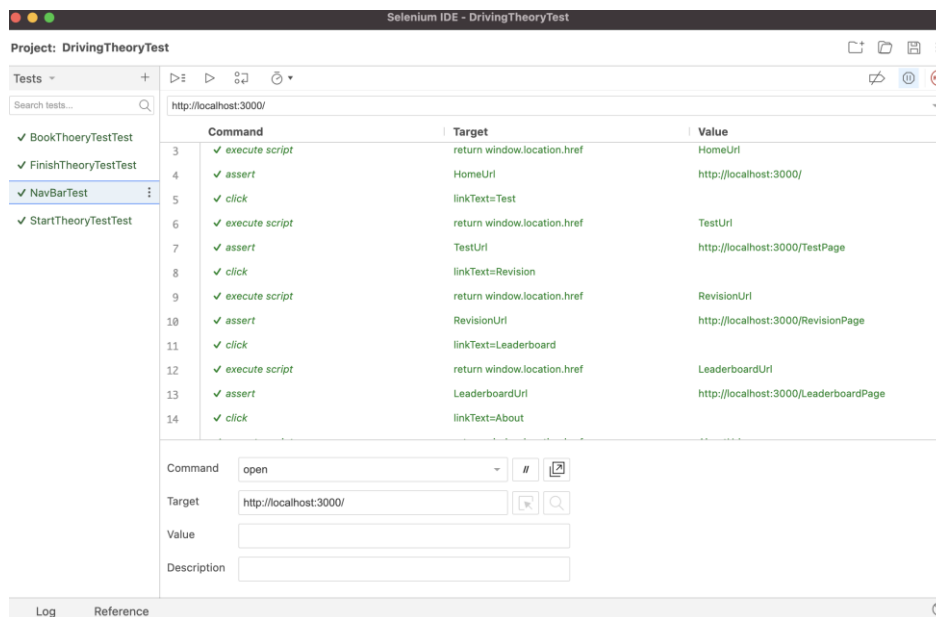
Respectively the first Test case checks if you can use the book theory test functionality where it opens the homepage, Clicks the book button this button's functionality is to redirect user to the RSA website where they can book a test, then selenium saves the URL of the page and checks it with what it should be.



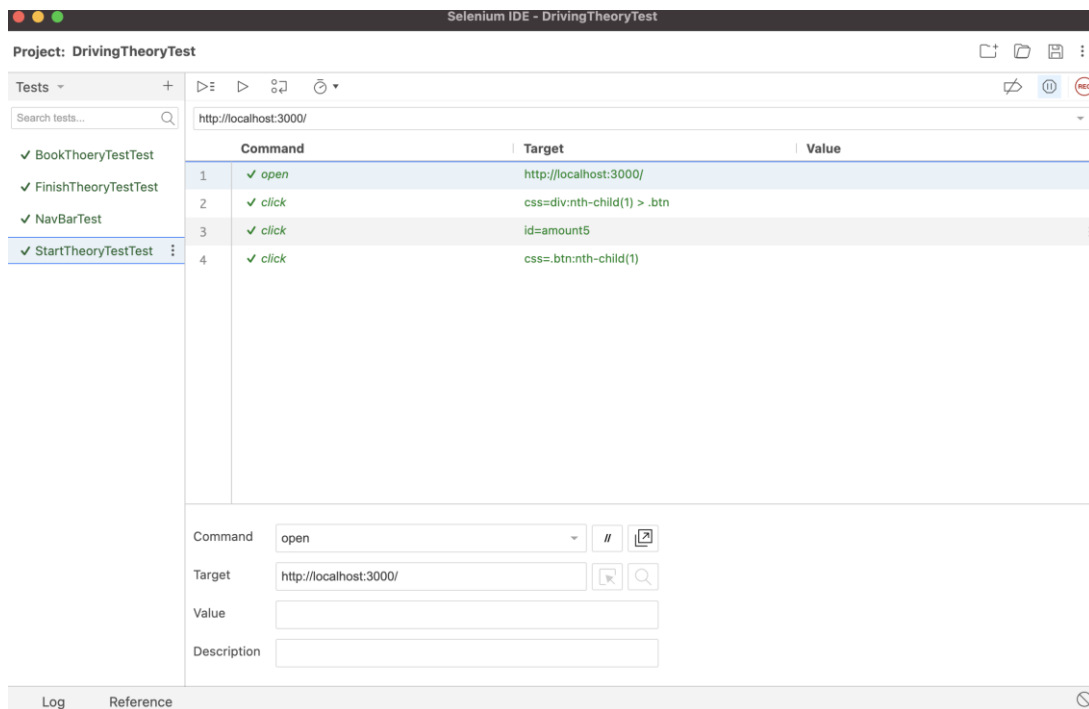
Second Test case finish's a Theory Test and entered a name into the input box so it can be posted to the Database.



Third Test case tests the Navigation Bar clicking into each option and checking if the URL match's what it should be for that page.



Finally, the fourth test case checks if you can start a test, Clicking Start Test, Choosing The question amount and beginning the test.



Deployment

We have chosen to deploy our website using Heroku and Docker, Heroku is a website which allows to run you node app application, Docker is a tool commonly used by many developers, Docker uses containers which you can build , store distribute your application in, This is done so anyone with docker can take this container with your project/application and can run it on any machine whether it be a different operating system or different versions of tools needed to run it. This makes. It makes a consistent environment regardless of the underlying infrastructure and create an error free environment so full focus can be at the scope of the project or the code.

Limitations and Known Bugs

As the project is we have met all the initial goals we set out for ourselves. But in the future, there would be some features we would like to add. For example, we would like to add an administration login so that rather than having to go to the database and input the questions there and admin could add questions, answers, and explanations so that the whole process can be easier and done entirely through our website. Now the only know bug/issue we have is that when the user navigates to the test page and if they scroll all the way down to the finish quiz button, they are prompted to enter their name to post their score which we would prefer if this was not the case, so we don't have false scores going to our leaderboard.

Conclusions from the Project

In conclusion, we're very pleased with how the project has gone. Fortunately, we never encountered any major issues that slowed down development. With using the MERN architecture we have broadened our knowledge on all its components. Hopefully in the future we hope to add some more functionality to the website such as some admin services where an administrator can add questions and answers through the website rather than having to input them into the database through Mongo. This will make the process much more straightforward. We're also happy with how we managed our time with the project. Having our weekly meetings with our supervisor helped us stay on track with our time management as well as manage our goals and specifications for the project and avoiding any scope creep. This project has been great to work on and has helped us develop our profession practice skills. Hopefully in the future we get another chance like this.