

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the text [Date].

[Date]

Just for Drinks

Created by: Niall McDonagh

(Jamie Gallagher)

Written By: Niall McDonagh

Supervisor:

John French

Submission Link:

<https://github.com/NiallMcDonaghNMD/PPITGroupProject22>

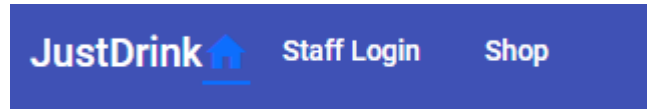
Several thin, curved lines in shades of blue and grey that originate from the bottom left and sweep upwards and to the right.

User

Disclaimer:

This has been a one-person project primarily (95%).

Jamie contributed to the navigational bar code by implementing it.



As shown, he implemented the JustDrink and the home page button, with the layout for the navigation bar navigational sections, which I edited.

The rest of the project has been completed by myself (Niall McDonagh).

I'm sure the git commits will show this clearly.

Project Requirements:

The initial plan for the project was to have a fully functioning clone of JustEat, however instead of food the main items would be Alcohol.

A working frontend connected to a MySQL database was required. Being able to login/register an account, shopping cart, having a staff login section where you could adjust pricing for each shops items, Nice GUI and sleek animations were key focuses for the project.

However, due to limitations and conflicts/arising problems I, Niall, had to adjust the project significantly.

Technologies Used:

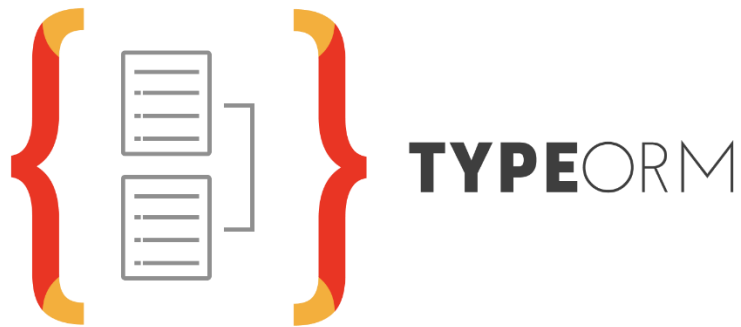


Nest JS - <https://nestjs.com/>

As we required a backend to talk to the frontend/database, We decided on Nest. Nest is a framework that uses either Express or Fastify, HTTP Server Frameworks to build Node.js server side applications.

Although we have used Express in our course previously, I wanted to try to implement and use a different alternative than what we have used in the past. Primarily to broaden my knowledge base and fundamental understanding of how “a bread can be sliced in several ways”.

Nest has proven to be a very powerful framework.



TypeORM - <https://typeorm.io/>

TypeORM was by far the most difficult part of this project to understand and implement. Primarily because this was completely alien to myself. However, in saying this I found that once I was able to implement it and read up on solutions to any issues I encountered, I was able to see how powerful a “middleware” piece of software can be.

TypeORM is a Object Relational Mapper. It allows you to use JavaScript/TypeScript and connect that code directly to a relational database, like MySQL or MongoDB. Once setup it works flawlessly and works very well. It had a lot of functionality built into it for any occasion.



Postman - <https://www.postman.com/>

Postman is an Application Programming Interface. It allows you to send HTTP requests to any website **WITHOUT** the need for a frontend to utilise. It allows you to make folders and save specific queries which is a very nice feature. Initially, I was primarily working on the backend of the application, and while Jamie was working on the frontend, I needed a fast and efficient method to test what I worked on. I found Postman to be exactly what I needed, and will use it in the future when making any web applications for ease of testing.

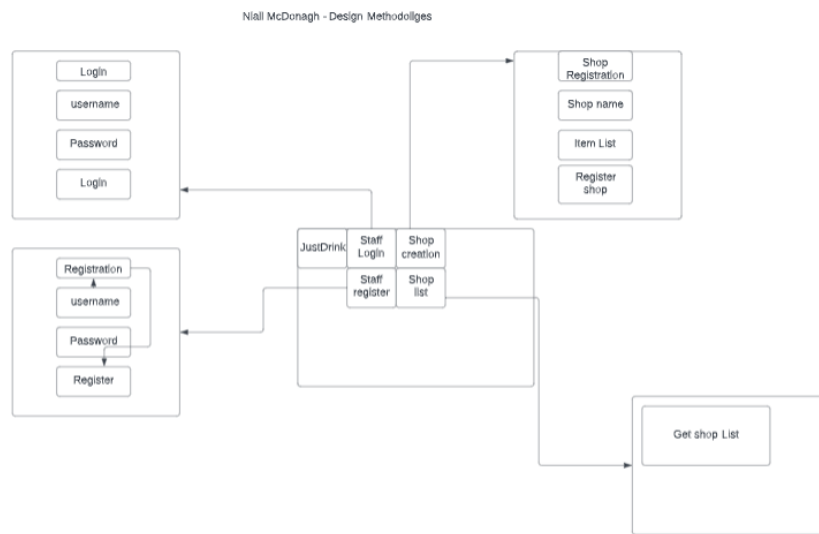


DBeaver - <https://dbeaver.io/>

DBeaver is a database tool that can be used with several types of databases. It has a very usable GUI that allows you to interact with databases in a more efficient manner than with using the MySQL command line console method.

I used it several times in conjunction with Postman to insert and see data being put correctly into the appropriate table/database. It was straightforward to setup and I found it useful as I grew tiresome of using the command line for MySQL consistently.

Design Methodologies Implemented:



Architecture of the Solution:



Angular - <https://angular.io/>

Similarly to Nest, Angular is a framework. However, instead of Nest it is a Web Application Framework, a “frontend” compared to a “backend” like Nest.

Angular uses components, which could be compared to building blocks of the application. Each component typically has 3 sections – CSS, HTML and TypeScript pages respectively. Angular interlinks these automatically when you create them using the Angular CLI, which makes it very easy to get started with it. It also builds your application with everything you need to start off.

One of the nicest parts about Angular is its built in Routing capabilities. It allows for effortless interlinking between components (pages) and is fairly robust. It is what the project uses for its webpages, and made making several elements and interlinking them straightforward.



Bcrypt - <https://www.npmjs.com/package/bcrypt>

Bcrypt is a set of functions that allow you to hash and unhash passwords or other information.

Bcrypt uses salt, which is a random piece of data that is used to hash passwords. Salt allows for a more secure storage of passwords, and less risk of exposing the original plaintext password. It then uses hashing with conjunction of the salt to encrypt the password **one-direction**. Thus, the password cannot be decrypted. I used this for the user login/password system, to ensure user data is protected. I enjoyed learning about encryption through using bcrypt, and has opened my mind to the world of cybersecurity.

express

Express - <https://expressjs.com/>

Express is a web application framework that provides a host of HTTP/web functions and abilities, such as routing and receiving/sending data to the web client.

In the project, express is not directly used majorly, however it is a core element of Passport, which we do use for an element of encryption/dealing with user data.



Passport - <http://www.passportjs.org/>

Passport is an authentication service that uses Node Js. The project uses Passport in conjunction with jwt and bcrypt for the user/login system.



JWT - <https://jwt.io/>

JWT (JSON Web Tokens) is a two way method of verifying data, enabling the data to be digitally signed. Typically, this is encrypted.



Wampserver - <https://www.wampserver.com/>

I used wampserver as the primary way of running the MySQL database. Although you can use other programs/software to do this, I was familiar with wampserver already and thus did not have to worry about learning new software for the database element.

Project Management Style:

Initially, We were going to have a team meeting twice a week so have a consistent schedule, and to update each other on the progress we made individually, and plan accordingly for the project as a whole. However, A lack of participation and consistency made this difficult.

The source code was managed via GitHub. I originally planned to use GitHub's built in milestone/goal tracker however once the project was made clear to me that it was going to be a single person project, I decided to not plan it out there.

I feel this was a mistake, primarily because having it planned out and all the features marked out properly allows for good organisation.

Limitations:

The biggest limitation was scope. Myself and Jamie originally planned to have a fully fledged functioning website that you could select a region/location of a shop closest to you and buy alcohol with a nice functioning GUI and a user login/registration element as well as a administrative/staff login or registration element that once logged in you could adjust stock for each shop and add or delete items accordingly.

We decided that I would focus on the backend systems, setting up the database and connecting it to the frontend that Jamie would work on. Once I had the backend setup however there was nothing to link it to. Jamie was inconsistent and made it difficult for me to get solid portions of a certain section finished. Several times I needed some element completed and found either no response or a response that yielded no results.

Although I still feel group work **can** be beneficial, and more productive and useful than working with yourself on a singular entity, when it goes downhill it can have some nasty consequences.

Conclusions and Recommendations:

Originally, we had the following plan and design for the website:

- MySQL database connected via TypeORM and NEST JS to the frontend of the website
- The website being a store front for alcohol, with various features such as:
 - List of various shops in various cities/towns/regions

- Shopping cart and checkout/payment process
- List within a specific shop of alcohol – price, image and quantity left within the shop
- User Account to save information from the shop
- Nice homepage with information/background of the shop, carousels switching between various images

However, due to Jamie leaving and or not contributing to the project, I had to massively reduce the scale as the workload was going to be fairly significant. I found it difficult to grasp certain concepts, and having a teammate that I could ask consistently would have been of great benefit.

I personally learnt a lot during this project. As I mentioned previously I do find group projects to be beneficial **IF** all members try to contribute their fair share of the workload. Otherwise, it becomes exhaustive and or mentally draining for the people who have to pick up the slack.

Im satisfied that I have learnt more elements of development. Some frameworks can be better than others and I definitely realised that while working with this project. Specifically I found Angular to be of massive help and allowed me to focus on the more difficult or tedious elements of the project.

I also seem to enjoy the challenge of working with backend systems. I have been wondering what fields I may personally go into after my degree, and after working with this project I can see the appeal of working on the “hidden” elements of a project/piece of work.

I would **recommend** to try different ways of implementing a solution. I found it improves your thought process and ability to mentally approach a task in a different way, which typically yields better results from my experience. Although I was originally frustrated with the “lesser” known aspects of some of the components I had to learn from scratch, once I found the appropriate knowledge on how to implement certain aspects It was very enjoyable to learn.

Learnings and findings:

One of the biggest learning outcomes for me was how important communication and planning is in a group project. Without proper planning and somewhat daily communication it can be difficult to find motivation and direction for the project you are working on. Starting the project I found is one of the most difficult aspects to mentally get past, primarily because I felt that I was facing a massive hurdle. However, after concluding the project I have realised that If proper planning and organisational methods were used, the project would not have felt so daunting.

References:

Programming’s fun Tutorial series for NEST JS, MySQL and Angular:

https://www.youtube.com/watch?v=6ySmc_vvssA&list=PLsjmv9aDmNDD0EZQU5yZaI0hNTL1umKF9&ab_channel=Programming%27sFun