

Technologies Used / References

Backend

- **Apollo Server**: An industry-standard implementation of the GraphQL schema. We use this to build our API specification and the code that runs the actual SQL queries to the database on top of Apollo. (*We specifically use the apollo-server-express*)
- **Knex.js & Objection.js**: A SQL query builder & ORM implementation for Node.js. (*Allows us to build SQL queries, dealing with SQL injection prevention and graph inserts for inserts like customer purchases*)
- **Express**: Framework for Node.js used to route to our Apollo implementation and login system.
- Other tools the NPM packages used: mysql2, cors, bcrypt, jsonwebtoken, express-session, body-parser, gql

Frontend

- **Next.js**: A React framework that allows us to use React easily, with server-side rendering and deploy it in one command.
- **React**: A JavaScript library for building user interfaces.
- **Apollo Client**: A GraphQL client interface, with mutation validation.
- **React Bootstrap**: React implementation of the popular Bootstrap CSS framework used for UI components
- Other tools the NPM packages used: axios, graphql, js-cookie, next-apollo, react-dom, react-icons, react-responsive, sass

Other link

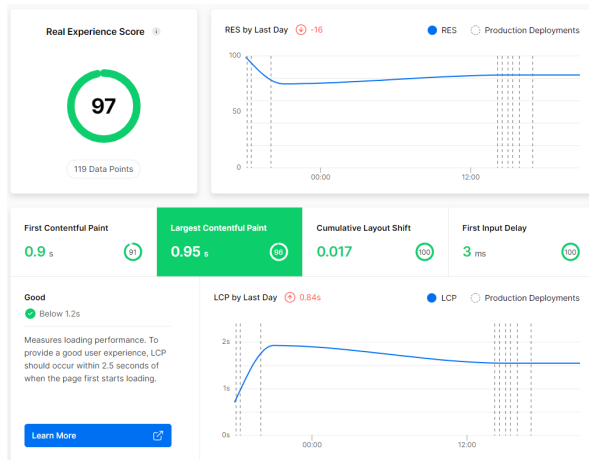
- <https://stackoverflow.com/questions/27012854/change-iso-date-string-to-date-object-javascript>
- <https://www.apollographql.com/blog/modularizing-your-graphql-schema-code-d7f71d5ed5f2/>
- <https://flaviocopes.com/graphql-auth-apollo-jwt-cookies/>

Images sources:

- <https://www.istockphoto.com/photos/single-rose?phrase=single%20rose&sort=best>
- <https://www.sizzix.co.uk/664390/sizzix-thinlits-die-set-8pk-tulip>
- <https://www.ikea.com/gb/en/>
- <https://www.lowes.com/pd/Galanz-4-0-cu-ft-Retro-Mini-Refrigerator-with-Dual-Door-Tue-Freezer-in-Red/5001372703>
- <https://www.samsung.com/uk/refrigerators/family-hub-fridge-freezers/>

Report

Over the past few weeks, we have assembled a complete, fully-functional web based order creation and management system for Generic Flatpack Furniture Co®; over the course of the next few pages, we will go into detail on the development process for our system, and guide you through the process of using it from both a staff and customer perspective.



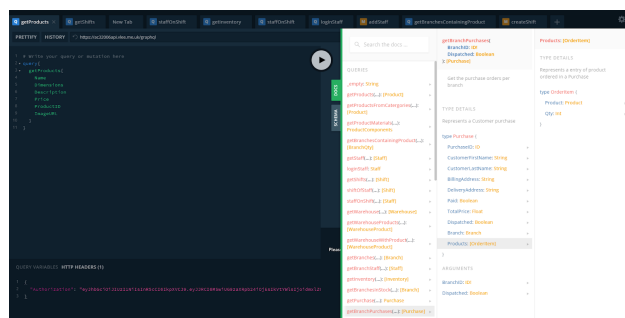
Vercel statistics for website performance

Front-end Development

We have built a modern JAM (Javascript, API and Markup) stack using Next.js to create our website. Next.js is a flexible implementation of React (a library for creating user interfaces) that can be used to efficiently create dynamic web pages. React is declarative, making code easier to debug and component-based allowing us to manage things like state easily and intuitively. Apollo Client is then used to make and cache GraphQL queries and mutations to our database.

Backend development

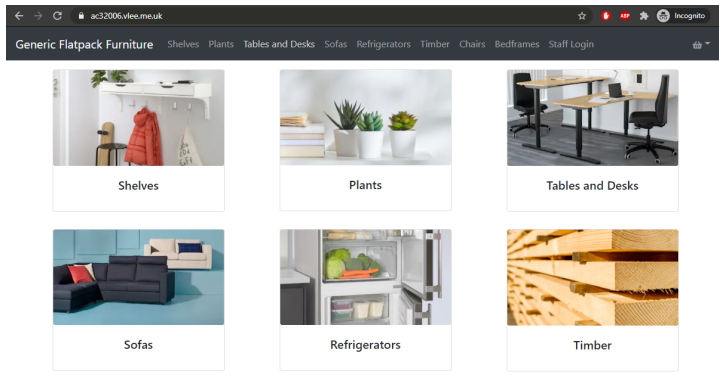
The website the users interact with connects to a backend API that does the actual interfacing with the database to store, retrieve and limit view of data. This API is built on top of an industry GraphQL schema allowing future development teams to easily interface with the website we've created today to create new user experiences such as mobile applications and integrating voice assistants. GraphQL allows us to query only the fields we use in our specific page and add additional fields without breaking prior code.



GraphQL API Playground

User Guide

Customer



Viewing items in a category

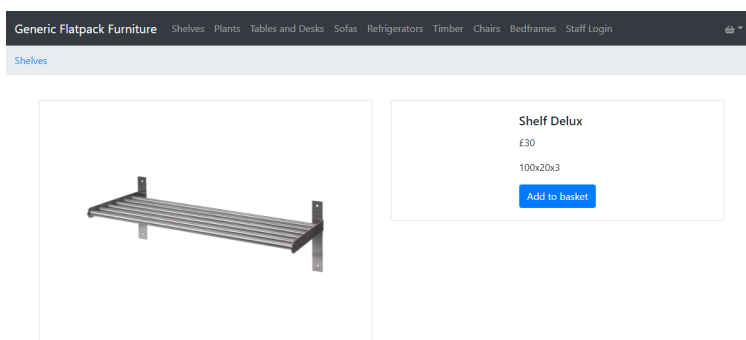
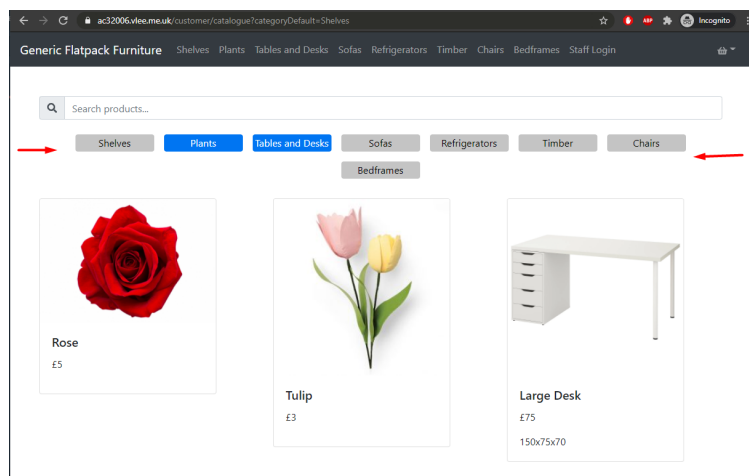
On the main home page, select a category to view by clicking on any of the pictures or titles underneath.

Viewing products from multiple categories

You can toggle on and off the tokens at the top of the page when viewing a category to get products of one or more categories.

View a product

Click on a product you want to view.



To add a product to the basket

Click on the **Add to basket** button.

Editing the number of items in a basket.

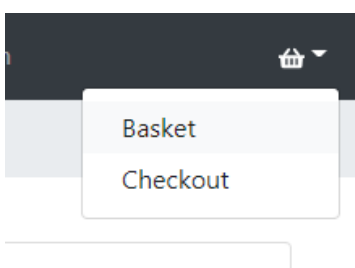
On the product page, use the **+ / -** buttons.

View Basket

Click on the basket icon on the top right and then click on **Basket**.

Checkout

Click on the basket icon on the top right and then click on **Checkout**.



Using the Basket


The basket page will display any and all unique items that you have selected from the catalogue, as well as the total cost for all items.

Increasing and Decreasing Quantity

Click on the + and - buttons to increase or decrease the quantity of an item by one respectively. Pressing the - button when the item's quantity is at 1 will delete that item from the basket.

+

-



Large Desk


5

150x75x70

£75

+

-



Desk Chair

2

50x50x60

£250

Total cost: £875

Advance To Checkout

Empty Basket

If nothing has been selected from the catalogue, or all items in the basket have been removed, the basket will display an 'empty' message.

Basket

The basket is currently empty.

Total cost: £0

Advance To Checkout

Advance to Checkout

Click on the '**Advance to Checkout**' button to proceed to the checkout page where you can enter details for ordering. If the basket is empty, the checkout page will return a message stating that you need items in the basket to checkout. If no branches stock the items that are in your basket a 'not in stock' message will be displayed.

Checkout

First Name

Enter your first name.

Hugh

Last Name

Enter your last name.

Mungous

Delivery Address

Enter your delivery address.

Flat 1

1 Example Street

Exampleton

EX1 EX1

Dispatch Branch

Not all branches will have all the items in your basket in stock

Dundee ▾

Billing Address

Enter your billing address.

Flat 1

1 Example Street

Exampleton

EX1 EX1

Credit/Debit card Checkout

Staff

Staff Login

Email address

We'll never share your email with anyone else.

Password

Sign in

Log in

Click on **“Staff Login”** in the navbar and enter your login details. Then press enter or click **“Sign in”**. This will take you to the Shifts page. You'll notice the links in the navbar change to only show staff-related pages.

Join/Leave a shift

To join a shift, select your branch from the dropdown menu (“Dundee” by default) and click **“Join shift”** on the shift you want to join. The page will then refresh, and the button will change to say **“Leave shift”**. You can then click this button to leave that shift.

Shifts

Dundee					
Date	Start Time	End Time	Slots left	Staff required for shift	Choose shift
2020/12/25	7:00:00 PM	7:01:00 PM	3	5	Leave shift
2020/12/25	12:00:00 PM	7:00:00 PM	5	6	Join shift
2020/11/24	12:00:00 PM	7:00:00 PM	10	10	Join shift

Add Shift

Start Time

End Time

Staff Required

Submit

Create a shift

To create a shift, select your branch from the dropdown menu (“Dundee” by default). Select the start/end date/time for the shift by clicking on the calendar icons, enter the number of staff you need for the shift, and then click on **“Submit”** to enter the shift into the system.

Add a staff member

To add a new staff member, click on **“Admin”** in the navbar. You'll then see a form saying **“Create Staff Member”**. Fill out their details and click **“Submit”** - this will add them to the table of staff and give them a login.

Create Staff Member

First Name

Last Name

Phone Number

National Insurance Number

Email address

Password

Address

Wage (£)

Position

Position

Submit

Assign/Unassign staff members from branches

Each staff member can be assigned to the two branches. To assign them to a branch, click **“Assign or unassign staff members”** at the bottom of the **“Create Staff Member form”**. Then, select the branch you want to assign them to from the dropdown (**“Dundee”** by default). You can then click **“Assign”** next to a staff member to assign them to that branch, or **“Unassign”** to unassign them from that branch.

Assign or unassign staff members			
Dundee ▾			
Staff ID	First Name	Last Name	Toggle
1	Vincent	Lee	Unassign
5	Max	Kelly	Unassign
6	Bob	Bob	Unassign
7	Max	Kelly	Assign
8	Max	Fyall	Assign
9	Nick	DeVilliers	Assign
10	Dora	the Explorer	Assign
11	jpg	png	Assign
12	jpg	png	Assign
13	Maxwill	Kelly	Assign
14	Maxwill	Kelly	Assign

Purchases per branch

Dundee ▾

Purchase No.15

Ordered By:
Jon Jim

Billing Address:
Bill Address

Delivery Address:
Delivery Address

Paid?:
Yes

Total Order Price:
£20

Dispatch Order

Products Ordered

Name	Price	Colour	Weight	Dimensions	Qty Ordered
Shelf V1	£10	White	0.2	100x20x2	1
Rose	£5	Red	0.05		2

Dispatch customer orders

When a customer makes an order, it will show up in the **“Purchases”** page when signed in as a staff member. Orders are made directly from branches, so you’ll need to select a branch at the top of the screen from the dropdown menu (**“Dundee”** by default). You can then click **“Dispatch Order”** on an order to dispatch it.

Search a branch’s inventory

Each branch carries their own inventory from which purchases and orders are made. To search this inventory, click on **“Inventory”** in the navbar to get to the inventory page. Click on the dropdown menu in the top left to select a branch (**“Dundee”** by default). You can then use the search bar at the top right to search for products in that branch’s inventory.

Inventory

Dundee ▾							
Search Products							
Product ID	Name	Category	Price	Weight	Colour	Dimensions	Quantity in stock
1	Shelf V1	1	10	0.2	White	100x20x2	60
4	Rose	2	5	0.05	Red		40
3	Shelf Basic +	1	5	0.1	Wood	50x20x1	49
16	Herman Miller Cosm	7	1059.99	10	Mineral	52x48x98	5
12	2x4 Pine	6	10	2			0

Inventory

Dundee ▾							
Search Products							
Product ID	Name	Category	Price	Weight	Colour	Dimensions	Quantity in stock
1	Shelf V1	1	10	0.2	White	100x20x2	60
4	Rose	2	5	0.05	Red		40
3	Shelf Basic +	1	5	0.1	Wood	50x20x1	49
16	Herman Miller Cosm	7	1059.99	10	Mineral	52x48x98	5
12	2x4 Pine	6	10	2			0

Adjust item quantity in a branch’s inventory

To adjust the quantity of an item in a branch’s inventory, type the new quantity of that item into the **“Update quantity”** field of that item in the table. Then click **“Update”**. The page will refresh and the quantity will reflect the new quantity of that item.

Technical Documentation

User Management

User management is done via the staff table, where passwords are Bcrypt hashes with 12 passes. Emails should be unique per staff member.

To add a staff Member manually via SQL you can run the following command:

```
INSERT INTO `mydb`.`Staff` (`FirstName`, `LastName`, `PhoneNumber`,  
`NINumber`, `Address`, `Wage`, `Position`, `Email`, `Password`)  
VALUES ('FName', 'LName', 'PNum', 'NINum', 'Address', 'Wage', 'Pos',  
'Email', 'BcryptHash');
```

Add/remove Staff to shift

To add a staff member to a shift you can do the following insert:

```
INSERT INTO `mydb`.`StaffShifts` (`ShiftID`, `StaffID`) VALUES  
('ShiftID', 'StaffID');
```

To remove a staff member from a shift, do the following delete:

```
DELETE FROM `mydb`.`StaffShifts` WHERE (`ShiftID` = '2') and  
(`StaffID` = '1');
```

Product Management

To add a product to the system into the database system you need to manually execute SQL insert commands, you can do so:

```
INSERT INTO `mydb`.`Products` (`Name`, `Category`, `Price`,  
`Description`, `Weight`, `Colour`, `Dimensions`, `ImageURL`) VALUES  
('Name', 'CatInt', 'PriceFloat', 'Description', 'Weight', 'Colour',  
'Dimension', 'URL to Image');
```

Add Product to Branch Inventory

To add a product to a Branch's inventory you can execute the following SQL:

```
INSERT INTO `mydb`.`Inventory` (`ProductID`, `BranchID`, `QTY`)  
VALUES ('ProductID', 'BranchID', 'QtyInt');
```

Update qty product in Inventory of a branch

To update a product in an inventory you can do the following if you know the InventoryID:

```
UPDATE `mydb`.`Inventory` SET `QTY` = 'NewQTY' WHERE (`InventoryID` = 'ID');
```

If you don't know the Inventory ID you can do the following:

```
UPDATE `mydb`.`Inventory` SET `QTY` = 'NewQTY' WHERE (`ProductID` = 'ID') and (`BranchID` = 'ID');
```

Create a Customer Purchase

To create a customer's order you first need to state the details of the customer and the branch they're ordering to get a PurchaseID.

```
INSERT INTO `mydb`.`Purchases` (`CustomerFirstName`, `CustomerLastName`, `BillingAddress`, `DeliveryAddress`, `Paid`, `TotalPrice`, `BranchID`, `Dispatched`) VALUES ('FName', 'LName', 'Address', 'Address', 'PaidBool', 'TotalPrice', 'BranchID', 'DispatchedBool');
```

After you can retrieve the PurchaseID, you can insert into the ProductPurchases the items for the order, with the product being the inventoryID of the branch store of the product.

```
INSERT INTO `mydb`.`ProductPurchases` (`InventoryID`, `PurchaseID`, `QTY`) VALUES ('InventoryID', 'PurchaseID', 'NumberofItemToOrder');
```

Then you need to update the QTY in the Inventory for a branch's store of that product, you can do so using the following update query.

```
UPDATE Inventory
SET Inventory.QTY = Inventory.QTY - {Number of product purchased}
WHERE Inventory.ProductID = {ProductID}
and Inventory.BranchID = {BranchID}
```


Indices

Each table that contains a foreign key also has that said field in its own index as we regularly will query joining tables for said foreign keys.

Purchases

Search (Type: Index)

This is used for allowing a customer to retrieve their order details after placing an order, and we ask for these fields to verify a purchaser before showing order details

Fields

- PurchaseID
- Firstname
- Lastname

Branches

We use this to get the purchases made for a branch, so we can see what branches have what purchases they need to complete

Fields

- BranchID

Products

Name

Allows us to find products via their name

Fields

- Name

Category

Allows us to find a product group by their category, that is an int.

Fields

- Category

Staff

Email

As we use email to login, and they are also unique, we make an index of them for getting the details when we log in a user.

Fields

- Email

Advanced Queries

Return Staff that is a staff of a branch

```
SELECT Staff.*, Branch.*
FROM mydb.Staff
JOIN BranchStaff
ON BranchStaff.StaffID = Staff.StaffID
JOIN Branch
ON Branch.BranchID = BranchStaff.BranchID
```

Return Staff on Shift where the start dateTime is after 2020-12-25 19:00:00

```
SELECT Staff.*, Shifts.*
FROM mydb.Staff
JOIN StaffShifts
ON StaffShifts.StaffID = Staff.StaffID
JOIN Shifts
ON Shifts.ShiftID = StaffShifts.ShiftID
where Shifts.Start > "2020-12-25 19:00:00"
```

Get Products in a Branch 1's Inventory where they have no inventory left

```
SELECT Products.*, Inventory.QTY
FROM Products
JOIN Inventory
ON Inventory.ProductID = Products.ProductID
JOIN Branch
ON Branch.BranchID = Inventory.BranchID
WHERE Branch.BranchID = 1
AND Inventory.QTY <= 0
```

Get Product in Purchase, where the PurchaseID = 9

```
SELECT Products.*
FROM Products
JOIN Inventory
ON Inventory.ProductID = Products.ProductID
JOIN ProductPurchases
ON ProductPurchases.InventoryID = Inventory.InventoryID
JOIN Purchases
ON Purchases.PurchaseID = ProductPurchases.PurchaseID
WHERE Purchases.PurchaseID = 9
```