Exam project report

Progression:

Week 1:
I started off by reading through the assignment and planning what needed to be done. I split the project into different parts like the database, routes, front-end, tests, etc. I also went back to some old lessons since I had almost forgotten how to do certain things and which plugins were needed.

Week 2:
Starting by get all the necessary npm plugins and set up the SQL database connection and tables with their relationships. I had to update these tables later on. I worked my way down the assignment from top to bottom, starting with the **/init** route to set up roles, users, memberships, and product fetching so I had some data to work with.

Week 3-6:
During these weeks, I focused on creating CRUD functions and service files for the routes. The hardest route was the cart route due to the membership count update and checking product quantity.

Week 6-8:
I then started working on the admin dashboard front-end. This took longer than expected because, after doing some initial work, I found a video that showed what was needed. I had to remake my app to match the video, which also showed more functionality for the dashboard than I initially thought. For example, for the search function, I had to fetch the brand/category data dynamically instead of hardcoding values like 'Apple' and 'Samsung'. If a new brand or category was added, it needed to appear in the search bar. I did a lot of research on adding validation, Bootstrap modals, and tables. The first admin/products page was the hardest to do because of the search function, which needed to switch between the product table and the search results table.

I spent the rest of the time on Swagger, testing, and documentation, which went relatively smoothly.

Challenges:

They was many things that didnt go was plan. I had to keep changing the table relationships to get them to work properly.

POST for /Search and /products: Spent days figuring out the raw SQL queries for these routes, Couldn’t find much info online about raw SQL queries.

Cart and Order Routes: These routes were super hard and complex Had to:

- Update the purchaseCount for memberships based on how many purchases were made.
- Check and update product quantities after orders were placed.
- Calculate discounts for higher-level members.
- Generate order numbers and create new orders.
- Create a new table for order items.

Take sometimes to write these and was so many functions involved, making it a ton of work.

The frontend-end: The front-end took way more time than I thought, Getting buttons to work with event listeners was buggy. I found out that when populate the data, the brand and category did not goes in the same order everytime, because I have em hard coded. So had to fetch both and display in search so it would also show other new added aswell. Needed to add more functionality for the admin to update user details. Struggled a lot with modals, to getting the current product details to show for editing same with the adding modal sometimes those two crashes for no reason.

Got a lot of errors with EJS for rendering and displaying data, Sometimes the syntax seemed right, but it didn’t work until I restarted the app

Other things is getting use to new operator like **ternary** and **spread** The code often didn’t look clean or understandable to me, probably because my eyes weren’t used to these operators. Barely got used to arrow function.

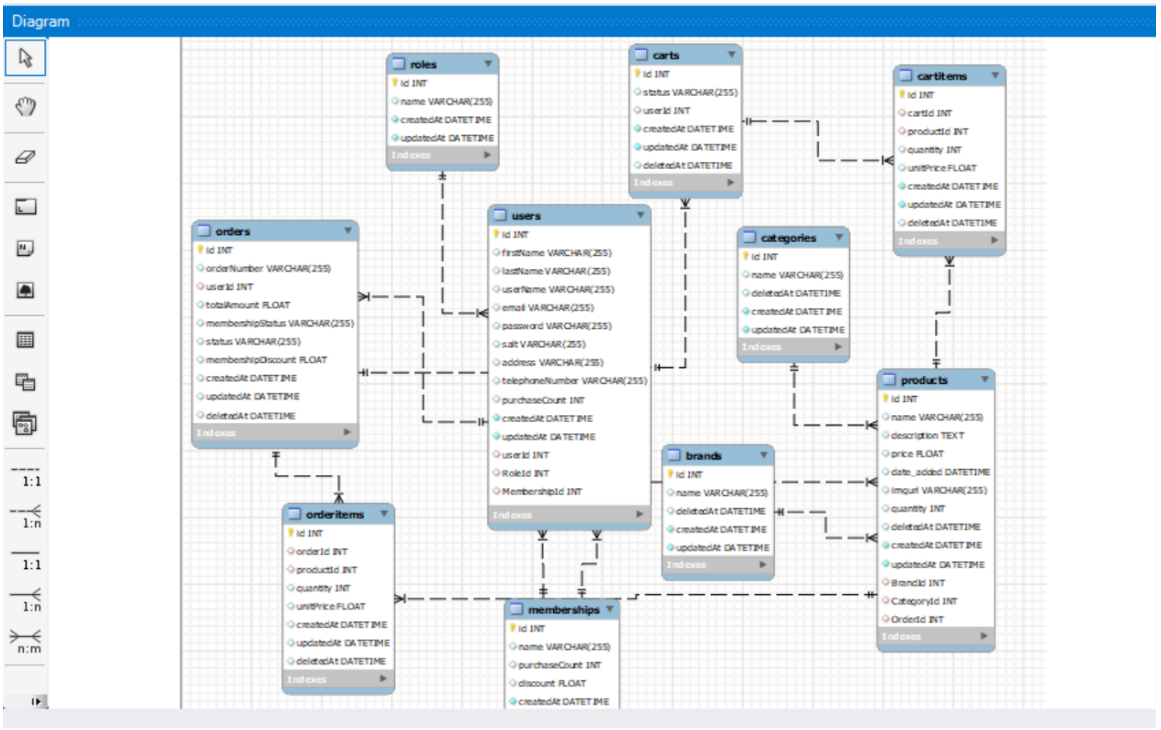
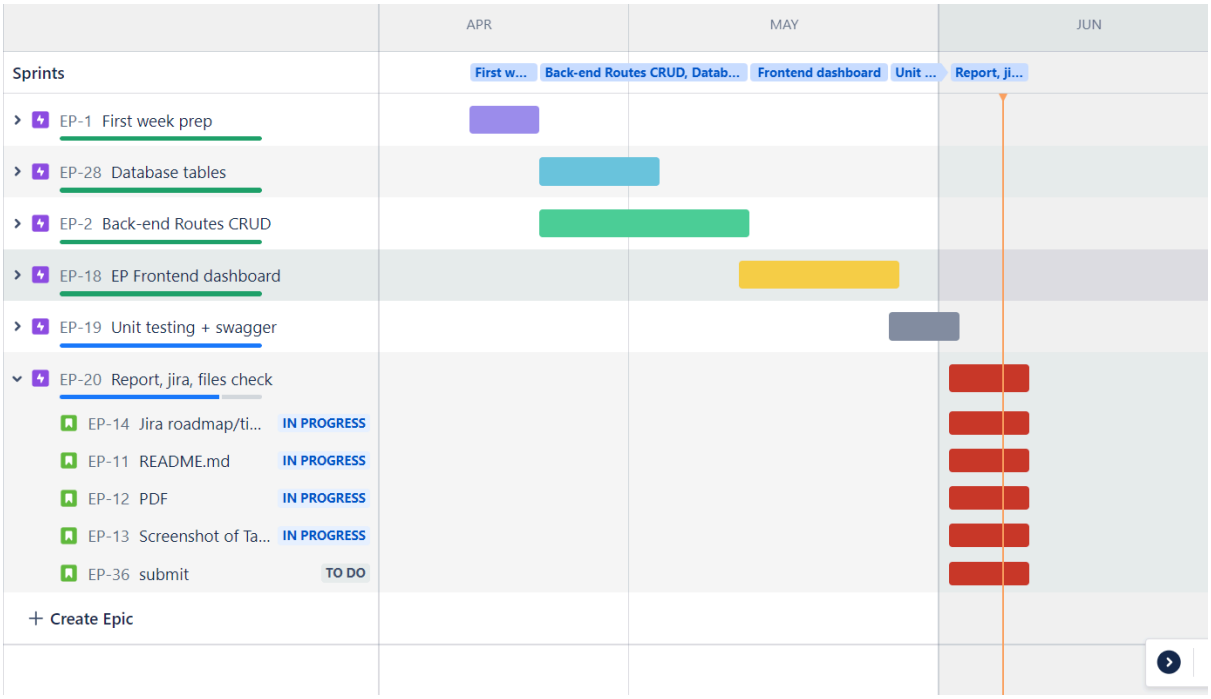


Table Relationships

- Brand - Product (One-to-Many): Each brand can have many products, but each product belongs to only one brand
- Category - Product (One-to-Many): Each category can have many products, but each product belongs to only one category
- Role - User (One-to-Many): Each role can be assigned to many users. These define what users can do, like being an admin or a customer.
- Product - Brand/Category (Many-to-One): Category/brand can contain many products.
- Product - Order (Many-to-One): Many products can appear in many orders.
- OrderItem - Product (Many-to-One): These are the specific products within an order, detailing what was bought, the quantity, and the price
- Membership - User (One-to-Many): These are the levels of membership users can have
- User: Each user has a role and a membership level, and they can have a cart and place orders.
- Cart - CartItem (One-to-Many): These are the shopping carts where users can add products before buying. Each cart can contain many items.
- Cart Items: These are the individual products added to a shopping cart. Each cart item is associated with one cart and one product.

- **Roadmap/ Timeline**



Sprints