

# COS 221 ASSIGNMENT 5

Group name: Null & Void

Group members:

Diya Narotam u23533596

Chinmayi Santhosh u24585671

Shavir Vallabh u23817146

Ayush Beekum u23596351

Jaitin Moodally u23621372

## Task 1: Research

### Research on Online Shopping and Price Comparison Platforms

In recent years there has been a drastic increase in the retail and e-commerce industry growth. The major cause behind this is the profound development of the internet and its ability to connect consumers with desired products across the world. With consumers now well-acquainted to online shopping at the comfort of their homes, there is a rising demand for advanced technological solutions which could make this more efficient. The COVID-19 pandemic has further contributed to this advancement, not only by making it convenient but also a necessity for many modern households [1].

Retail refers to the buying and selling of goods to the public and when this is done through the internet it is known as e-commerce [2]. The growing popularity of e-commerce is attributed to unmatched convenience, product variety, competitive pricing, fast delivery and customer feedback mechanisms at a click of a button. E-commerce websites keep evolving every day with clothing, food and electronics being amongst the most popular products. Well-known platforms such as Amazon, E-Bay and Alibaba are examples of well-known e-commerce companies that have dominated with their seamless user experience (UX) design [1].

Products and retailers in the e-commerce industry can be generally classified into specific categories. Some of the main categories for products include:

- Consumer Electronics – smartphones, laptops, wearable technology
- Apparel and Accessories – shoes, clothing, jewellery
- Home Goods – furniture, kitchenware, cleaning appliances
- Health and Beauty – skincare, supplements, wellness products
- Food and Groceries – including both non-perishables and fresh goods
- Digital Goods – software, e-books, online services

Retailers can also be categorized. As mentioned above, popular marketplaces like Amazon allow for third-party sellers to list products. Brand-specific stores like Apple sell directly to consumers. Niche retailers have formed specialized online environments like Etsy for handmade goods and crafts [1].

Several factors contribute to the way in which products and services are categorized and rated. One of the most popular forms is giving a rating through a star rating system, typically out of five. With this system, customers can provide and give a written review on products. Most e-commerce websites have products categorised as “best sellers” or “most popular” based on the sales volume. Algorithmic formed recommendations are also a common form of categorization and the user’s search history and clicks provide similar product suggestions. These personalization techniques improve user engagement. Recommendation engines enhance the shopper’s experience by presenting alternative products, often by using collaborative filtering [3].

Modern e-commerce platforms also thrive on their UX features like community interaction, verified user reviews and product filtering for specific criteria. UX has become a critical component of online retail success. According to Junfeng's comparative study on UX design tools, over 70% of users rated ease of use and interface clarity as the most critical factors influencing their satisfaction and continued usage of a design platform [4].

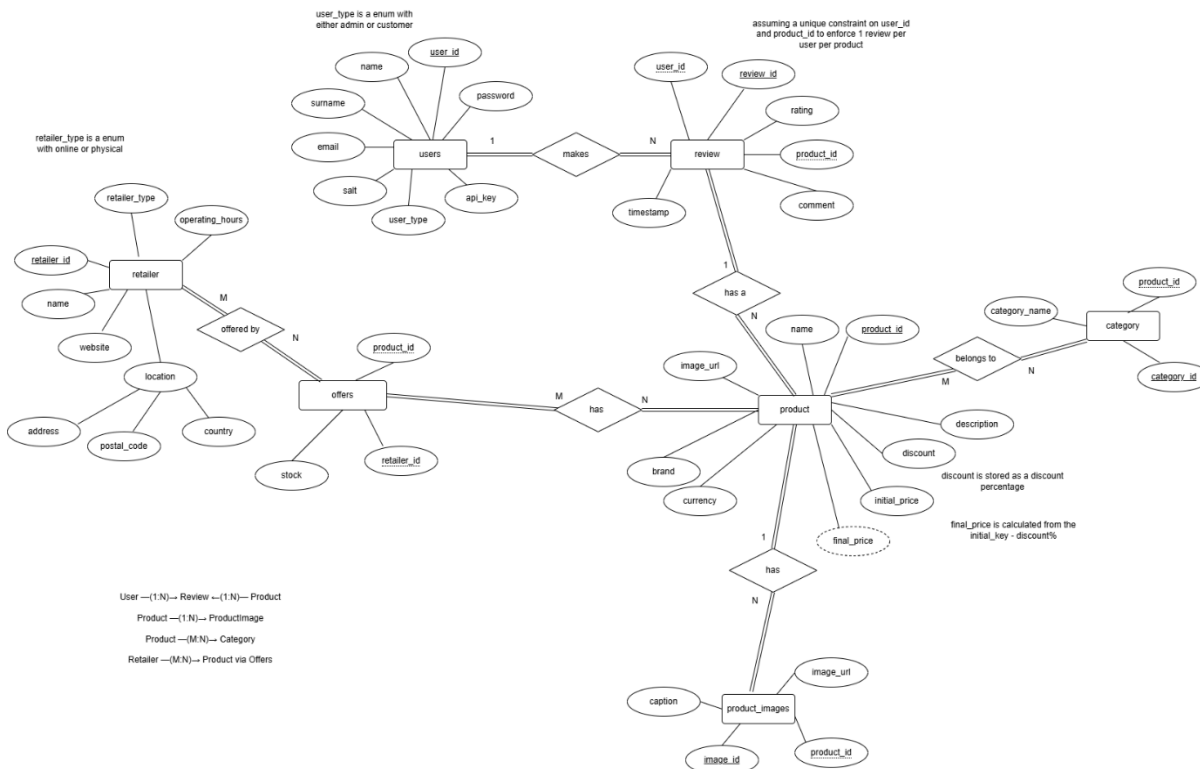
As CompareIt enters this vast and competitive space, its rise in the industry will depend on offering a user friendly UX design with frequent data aggregation and community engagement. By educating ourselves on existing competition platforms, CompareIt can become a brand-new generational tool for price comparison and product discovery based on the community and its consumer’s needs.

## Bibliography:

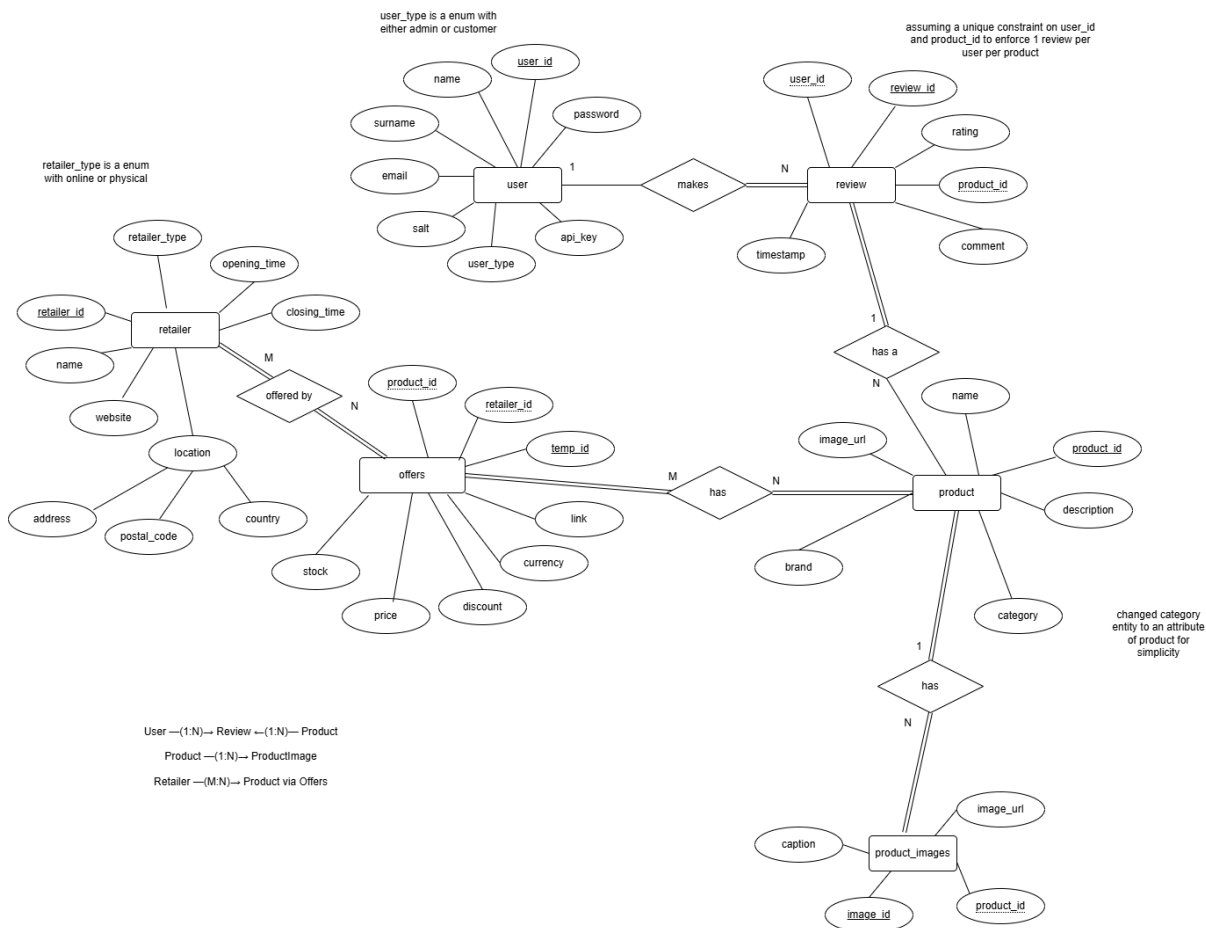
- [1] H. Ramcharran, "E-Commerce Growth and the Changing Structure of the Retail Sales Industry," *International Journal of E-Business Research (IJEER)*, vol. 9, no. 2, pp. 17–28, Apr. 2013. [Online]. Available: <https://www.igi-global.com/article/content/78296>
- [2] "RETAIL | meaning in the Cambridge English Dictionary," *Cambridge Dictionary*. [Online]. Available: <https://dictionary.cambridge.org/dictionary/english/retail>
- [3] Cambridge Dictionary, "E-COMMERCE | meaning in the Cambridge English Dictionary," *Cambridge.org*, Oct. 23, 2019. [Online]. Available: <https://dictionary.cambridge.org/dictionary/english/e-commerce>
- [4] J. Wang, "A Comparative Research on Usability and User Experience of User Interface Design Software," *Int. J. Adv. Comput. Sci. Appl. (IJACSA)*, vol. 13, no. 8, Jan. 2022. [Online]. Available: [https://www.researchgate.net/publication/363274896\\_A\\_Comparative\\_Research\\_on\\_Usability\\_and\\_User\\_Experience\\_of\\_User\\_Interface\\_Design\\_Software](https://www.researchgate.net/publication/363274896_A_Comparative_Research_on_Usability_and_User_Experience_of_User_Interface_Design_Software)

## Task 2: (E)ER-Diagram (30 marks)

Initial:



Final:



### Task 3: (E)ER-diagram to Relational Mapping

- Step-by-step conversion of the (E)ER-diagram to a relational model.
- For each entity/relationship, specify:
  - Tables created.
  - Primary/foreign keys.
  - Handling of multivalued attributes, weak entities, and n-ary relationships.
- Explain choices where multiple mapping options existed (e.g., "Used a junction table for M:N relationships").

## Steps:

Step 1: mapping of strong(regular) entity types and simple attributes

Step 2: Add the Weak entities and connect relationship with their strong entity, in our ER we do not implement weak entities hence no entities will be in the schema

Step 3: No one-to-one relationships

Step 4: Connected the 1-N relationships

User—(1:N)→review←(1:N)—product

product—(1:N)→product\_image

Step 5: Creating relationships from the M-N relationships

product—(M:N)→category

retailer—(M:N)→product through offers

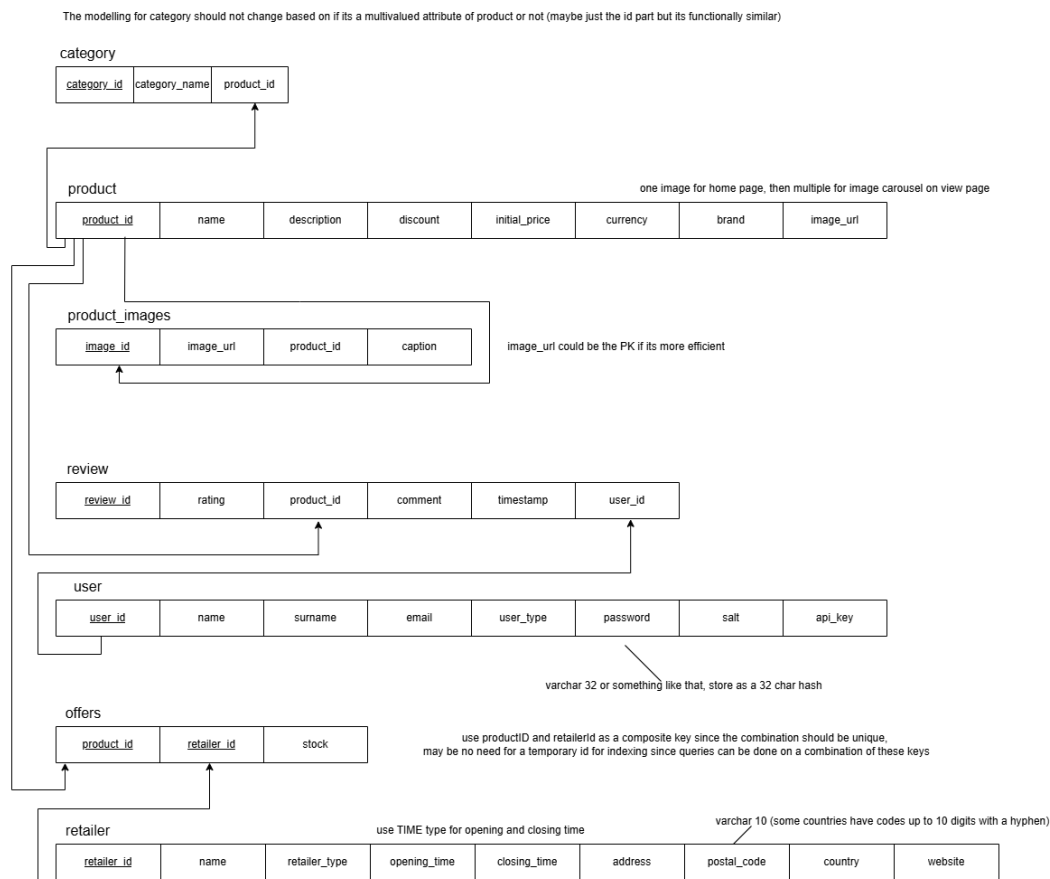
Step 6: Mapping of multivalued attributes, we do not use multivalued attributes hence are not in the schema

Step 7: No N-ary relationships

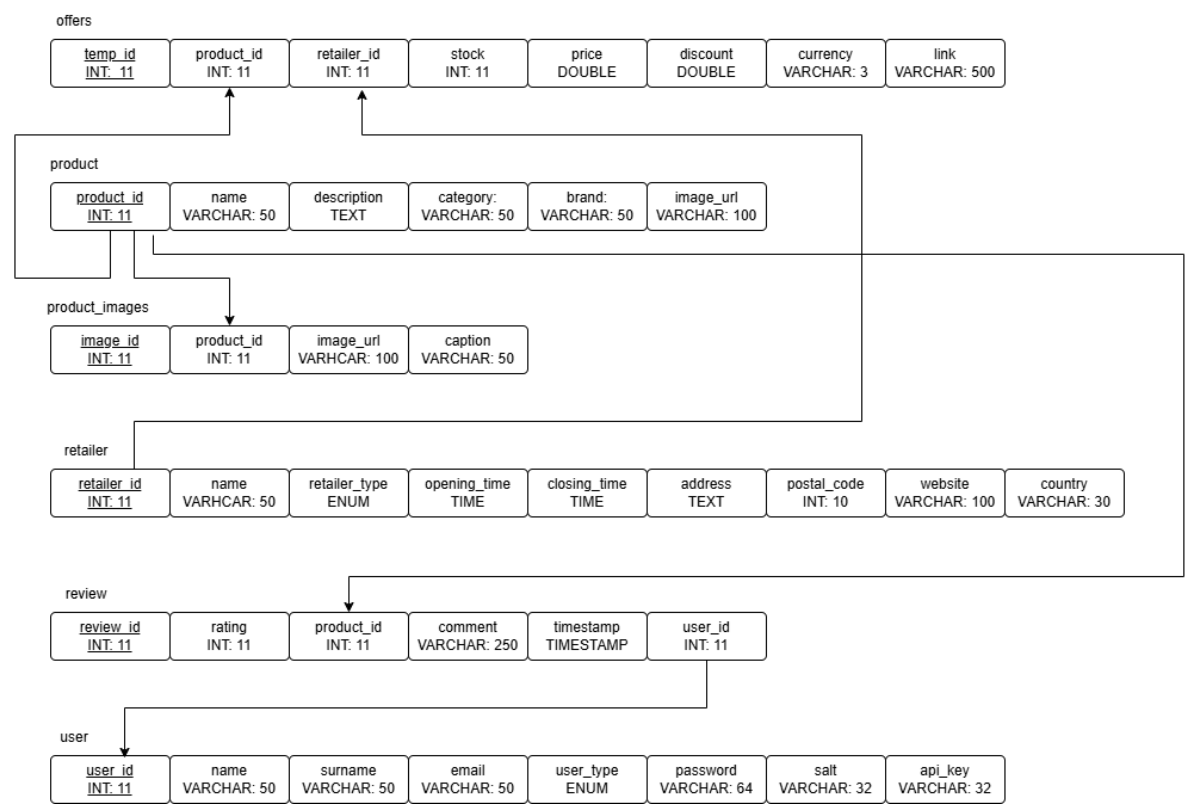
Step 8: No specialisation or generalisation

Step 9: No unions

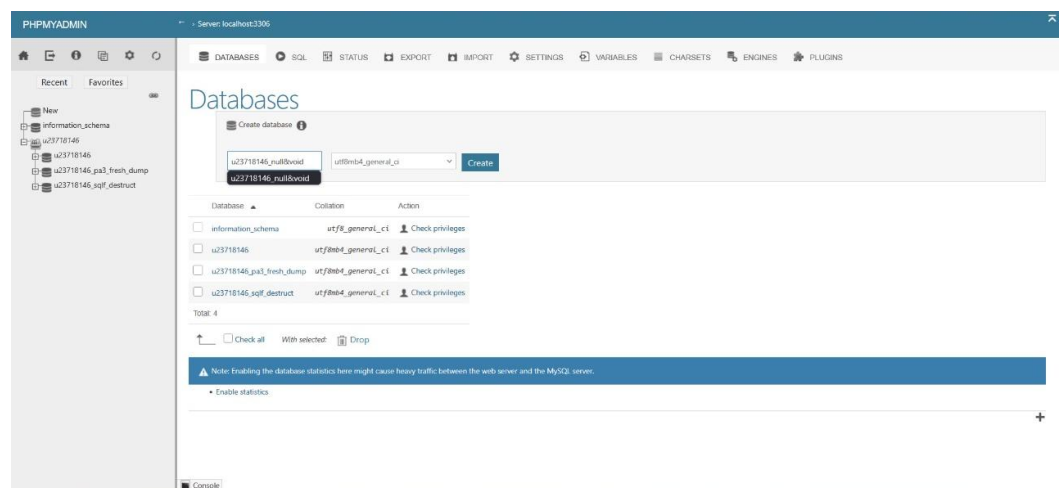
## Task 4: Relational Schema Old relational schema:



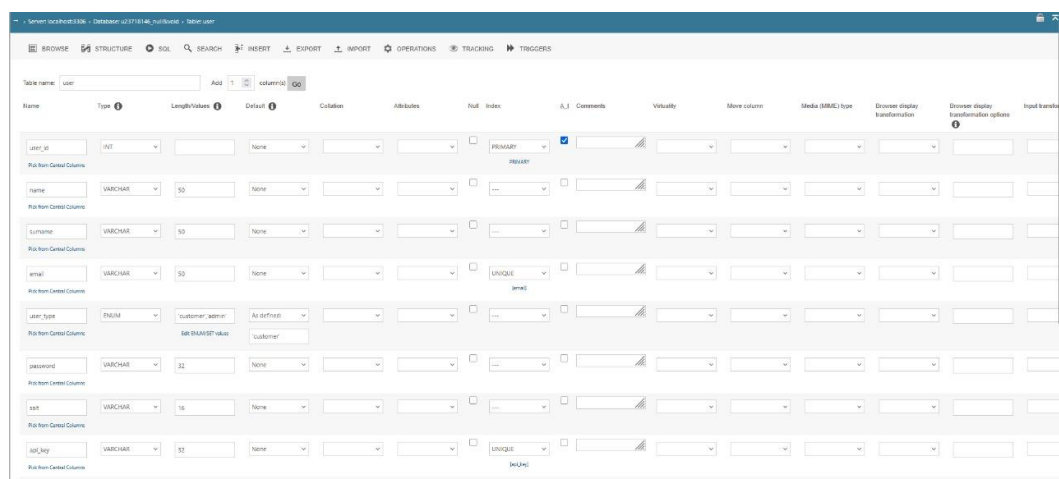
Final relational schema:



Database creation:



User table:



Products table:

Server: localhost:3306 • Database: u23718146\_null8void • Table: product

BROWSE STRUCTURE SQL SEARCH INSERT EXPORT IMPORT OPERATIONS TRACKING TRIGGERS

Table name:  Add 1 column(s) Go

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A.I.	Comments	Virtuality	Move column	Media (MIME) type	Browser display transformation
product_id	INT		None			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>					
name	VARCHAR	50	None			<input type="checkbox"/>		<input type="checkbox"/>					
description	TEXT		None			<input type="checkbox"/>		<input type="checkbox"/>					
discount	FLOAT		As defined			<input type="checkbox"/>		<input type="checkbox"/>					
initial_price	DOUBLE		None			<input type="checkbox"/>		<input type="checkbox"/>					
currency	VARCHAR	3	As defined			<input type="checkbox"/>		<input type="checkbox"/>					
brand	VARCHAR	50	None			<input type="checkbox"/>		<input type="checkbox"/>					

Offers table:

Server: localhost:3306 • Database: u23718146\_null8void • Table: offers

BROWSE STRUCTURE SQL SEARCH INSERT EXPORT IMPORT OPERATIONS TRACKING TRIGGERS

Table name:  Add 1 column(s) Go

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A.I.	Comments	Virtuality	Move column	Media (MIME) type
product_id	INT		None			<input type="checkbox"/>	INDEX	<input type="checkbox"/>				
retailer_id	INT		None			<input type="checkbox"/>	INDEX	<input type="checkbox"/>				
stock	INT		As defined			<input type="checkbox"/>		<input type="checkbox"/>				

Structure

Table comments:

Collation:

Storage Engine:

Review table:

Server: localhost:3306 • Database: u23718146\_null8void • Table: review

BROWSE STRUCTURE SQL SEARCH INSERT EXPORT IMPORT OPERATIONS TRACKING TRIGGERS

Table name:  Add 1 column(s) Go

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A.I.	Comments	Virtuality	Move column	Media (MIME) type	Browser display transformation
review_id	INT		None			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>					
rating	SET	"1","2","3","4","5"	None			<input type="checkbox"/>		<input type="checkbox"/>					
product_id	INT		None			<input type="checkbox"/>	INDEX	<input type="checkbox"/>					
comment	VARCHAR	250	None			<input checked="" type="checkbox"/>		<input type="checkbox"/>					
timestamp	TIMESTAMP		CURRENT_TIMESTAMP			<input type="checkbox"/>		<input type="checkbox"/>					
user_id	INT		None			<input type="checkbox"/>	INDEX	<input type="checkbox"/>					

Structure

## Product\_image table:

The screenshot shows the SQL Server Enterprise Designer interface for the 'product\_images' table. The table has four columns: 'image\_id' (INT, PRIMARY KEY), 'image\_url' (VARCHAR(100)), 'product\_id' (INT), and 'caption' (VARCHAR(50)). The 'image\_id' column is marked as the primary key. The 'product\_id' column is marked as a foreign key. The 'caption' column is marked as a unique key. The table is stored in the 'InnoDB' storage engine. The interface includes a 'Table name' field, a 'Columns' list, and a 'Structure' section for defining table properties.

## Retailer table:

The screenshot shows the SQL Server Enterprise Designer interface for the 'retailer' table. The table has seven columns: 'retailer\_id' (INT, PRIMARY KEY), 'name' (VARCHAR(50)), 'retailer\_type' (ENUM with values 'online', 'physical'), 'opening\_time' (TIME), 'closing\_time' (TIME), 'address' (TEXT), and 'postal\_code' (INT(10)). The 'retailer\_id' column is marked as the primary key. The 'retailer\_type' column is marked as an ENUM type. The 'opening\_time' and 'closing\_time' columns are marked as TIME types. The 'address' column is marked as a TEXT type. The 'postal\_code' column is marked as an INT(10) type. The table is stored in the 'InnoDB' storage engine. The interface includes a 'Table name' field, a 'Columns' list, and a 'Structure' section for defining table properties.

## Task 5: Web-Based Application

### • User Authentication and Management

The application features both login and signup functionality integrated within the graphical user interface (GUI). As the GUI is client-side based, user registration or login for administrative access is not available via the frontend. Instead, administrator credentials are inserted directly into the backend database. When an administrator logs in, they are redirected to the admin dashboard. Regular customers, upon successful login, are returned to the homepage and are provided with the option to log out at any time.

### • Product and Category Management

Only administrators can add, edit, or delete products and categories. These operations allow them to directly modify the database contents related to products and retailers. Customers do not have access to these functionalities.

### • Management of Stockists, E-commerce Platforms, Product Details, and Brands

Administrators can manage both the list of products and their associated stockists (e.g., retailers or e-commerce platforms). This includes editing product descriptions, categories, prices, images,



and brand information. The interface is designed to be user-friendly, enabling database manipulation without requiring any SQL knowledge.

- **Product Display: Pricing, Images, and Descriptions**

Product prices are shown on the product listing page. Clicking on a product image opens a full-sized view, allowing users to browse the product gallery. Each product entry also displays relevant brand information and a detailed description.

- **Product Sorting and Filtering**

Users can sort and filter products on the product page using various criteria. These include keyword searches via a search bar, filtering by product category, price range, and sorting by product name in ascending or descending order.

- **Database Updates (Insert, Edit, Delete)**

As mentioned under product and category management, administrators can perform all necessary database operations such as inserting new entries, editing existing data, or deleting records. These actions affect the underlying database tables directly.

- **Product View: Images, Prices Across Stockists**

The product detail page displays a large image of the product, followed by a list of price offers from different retailers. This section is fully functional with integrated API endpoints. (Postman request examples can be provided upon request.)

- **Product Reviews and Ratings**

Only logged-in users are allowed to submit reviews for products. This ensures that reviews can be accurately attributed to specific users, facilitating product rating calculations and data analytics. Each review is timestamped to help users assess the relevance and recency of the feedback.

- **Review and Rating Functionality**

As previously mentioned, registered users can rate and review products, and all submissions are tracked in the database for analysis and display.

- **Dashboard and Data Visualization**

The homepage serves as a basic dashboard by showcasing featured products and offering users an introduction to the platform through an intuitive and visually appealing design.

- **SQL Injection Prevention**

To ensure security, all database interactions use prepared statements to protect against SQL injection attacks.

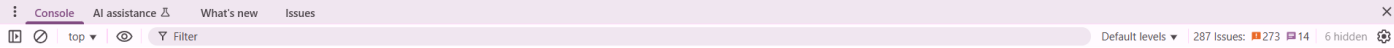
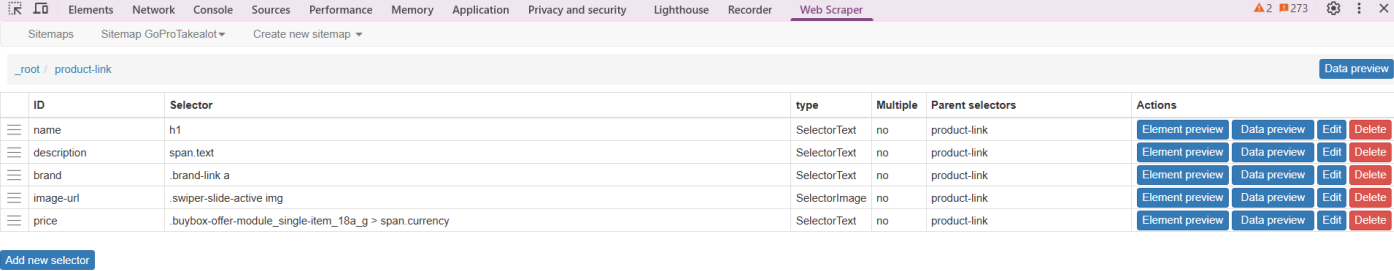
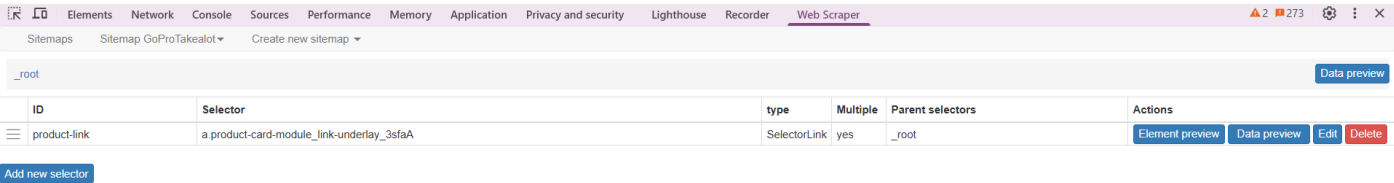
## Task 6: Data (20 marks)

- Explanation of data sources:

For our data collection we have used Web Scrapper.io(which is a web scrapping tool) to extract product related information from websites. The following steps were followed:

Initially we had to install the Web Scraper extension for Google Chrome to facilitate our data extraction

We then created multiple sitemaps which corresponded to our product category and website name. Within each sitemap we included a product link which had multiple customized selectors to pinpoint the specific datapoints that we required in correspondence to the database. A screenshot of our general setup has been attached below:



The data was then extracted by executing the scraper which automatically navigated through the configured sitemaps to retrieve the required datapoints from each product which we then carefully organized into a excel sheet to work with.

Using the excel document, we used a visual basic macro to transform the spreadsheet to an array of JSON data. From there we automated iteration through the data making a request to the

api.php to insert the products. The same procedure was done to offers but the product name was used to determine the product\_id and then insert the offers accordingly.

## Task 7: Analyse and Optimise (10 marks)

A query being executed before performance review:

✓ Showing rows 0 - 24 (66 total, Query took 0.0016 seconds.)

SELECT \* FROM product WHERE category = 'Audio'

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

1 > >> ☐ Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	product_id	name	description	category	brand	image_url
<input type="checkbox"/> Edit Copy Delete	2	Razer Blackshark V2 X	The Sound of Esports Embrace the sound of esports...	Audio	Razer	https://m.media-amazon.com/images/I/71mzxWaotaL_A...
<input type="checkbox"/> Edit Copy Delete	116	VolkanoX VXH200 Active Noise Cancelling Bluetooth ...	Bluetooth Connectivity, Active Noise Cancellation ...	audio	VolkanoX	https://media.takealot.com/covers_images/e1a422fcc...
<input type="checkbox"/> Edit Copy Delete	117	VolkanoX VXH300 Hybrid Active Noise Cancelling Blu...	Wireless, Over-Ear Design, Superior Sound, Adjusta...	audio	VolkanoX	https://media.takealot.com/covers_images/d22c3b469...
<input type="checkbox"/> Edit Copy Delete	118	Sony WH-CH720 Over-Ear Noise Cancelling Wireless H...	Bluetooth headphones with Noise-cancelling technol...	audio	Sony	https://media.takealot.com/covers_images/3e7eb8b44...
<input type="checkbox"/> Edit Copy Delete	119	Amplify Chorus 3 Bluetooth Wireless Headphones	Hands-Free Calling Support, Aux Input, Up to 22 ho...	audio	Amplify	https://media.takealot.com/covers_images/2b7ee3b4f...

The same query being executed with “explain”:

Your SQL query has been executed successfully.

EXPLAIN SELECT \* FROM product WHERE category = 'Audio'

[\[Edit inline\]](#) [\[Edit\]](#) [\[Skip Explain SQL\]](#) [\[Analyze Explain at mariadb.org\]](#) [\[Create PHP code\]](#)

+ Options

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	product	ALL	NULL	NULL	NULL	NULL	343	Using where

The fact that type is “ALL” indicates that full table scan is currently inefficient and that key is “NULL” shows that no index is used.

ALTER TABLE product ADD INDEX (category);

The query above will optimize performance

SELECT BENCHMARK(100000, (SELECT \* FROM product WHERE category = Audio));

The next query above will record execution times, this was run before the optimization query and once after. The results after shows quicker compare quicker by in which the index reduced the search complexity from  $O(n)$  to  $O(\log n)$ . We have about 300 products in our product table and without the index, for example, we search through the category field it would take about 2-5 ms whereas adding index to category (shown in above query) the search time is reduced to about 0.5-1 ms, which is about 2.5x faster.

type: ref

key: category

rows: 30

Extra: Using index condition

The text above shows the output after testing

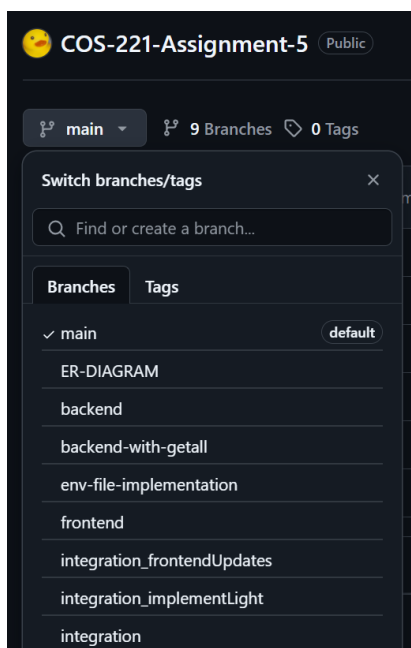
Why is it faster:

1. Uses a B-Tree index to jump directly to “audio” entries
2. Will only read relevant rows

## Task 8: Development (25 marks)

- Git usage:
  - The project uses Git for version control to ensure organized and collaborative development. We used a branching strategy for our repository with branches such as front end, back end and integration before fully committing and merging into the main branch after testing was done.

Screenshot of branches:



For information on our different branch commit history kindly view our github repo link

Code quality:

- Data validation techniques.
- Code structure/comments.

### Task 10: Bonus Task (15 marks)

As part of our bonus task, we implemented a **dark and light mode toggle** to enhance the user experience based on individual preferences. The **light mode was specifically designed with accessibility in mind**, particularly for users with **red-green color vision deficiency**, by avoiding the use of red and green within the same visual context or container.

To ensure content remains **perceivable**, we followed accessibility best practices such as:

- **Avoiding reliance on red text alone** to indicate errors by including **icons and descriptive labels** (e.g., "Error") alongside color.
- **Maintaining high contrast** between text and background elements in both light and dark modes for better readability.

In terms of security enhancements, we implemented the following:

- A **.env file** is used to securely manage and store confidential configuration variables, ensuring sensitive data is not hard-coded or exposed in the project workspace.
- We **do not store plaintext passwords** in the database. Instead, we hash passwords using the **SHA-256 algorithm**.
- To further strengthen security, we implemented **salting** by appending a **random 16-character string** to each password before hashing, protecting against common attacks such as rainbow table exploits.
- We also have a Wishlist mailing system to let users know when a product is available.

### Appendix

- GitHub repository link:

<https://github.com/ShavirV/COS-221-Assignment-5.git>