Web Development for Ecommerce using Neo4j and Leaflet

By
Devansh Maru
(19CEUBG055)

A project submitted
In
partial fulfilment of the requirements
for the degree of
BACHELOR OF TECHNOLOGY
in
Computer Engineering

Prof. Apurva Mehta

Assistant Professor,

Dept. of Computer Engineering

Faculty of Technology

Dharmsinh Desai University, Nadiad

Shri. Rajendra Gaikwad
SCI/ENGR-SE
Space Applications Centre,
Ahmedabad

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University
April 2023

CERTIFICATE

This is to certify that the project work titled

Web development for E-commerce using Neo4j and leaflet

is the bonafide work of

Devansh Pareshbhai Maru(19CEUBG055)

carried out in the partial fulfillment of the degree of Bachelor of Technology in Computer Engineering at Dharmsinh Desai University in the academic session

December 2022 to April 2023

Prof. Apurva A Mehta
Assistant Professor
Dept. of Comp. Engg.

Dr. C. K. Bhensdadia
Head,
Dept. of Computer Engg.

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University
April 2023

भारत सरकार
अंतरिक्ष विभाग
अंतरिक्ष उपयोग केन्द्र
आंबावाडी विस्तार डाक घर,
अहमदाबाद-380 015. (भारत)
दूरभाष : +91-79-26913050, 26913060
वेबसाईट : www.sac.isro.gov.in/www.sac.gov.in

Government of India
Department of Space
SPACE APPLICATIONS CENTRE
Ambawadi Vistar P.O.
Ahmedabad - 380 015. (INDIA)
Telephone : +91-79-26913050, 26913060
website : www.sac.isro.gov.in/www.sac.gov.in

Scientific Research and Training Division (SRTD)
Research, Outreach and Training Coordination Group (RTCG)
Management and Information Systems Area (MISA)

CERTIFICATE

This is to certify that **Mr. Devansh Pareshbhai Maru**, a student of B. Tech (Computer Engineering) of Dharmsinh Desai University, Nadiad, Gujarat has completed four months (06 December-2022 to 06 April-2023) project on **"Web Development for E-Commerce using Neo4J and Leaflet"** under the supervision of **Sri. Rajendra N Gaikwad**, SCI/ENGR-SE, EPSA-VRG-CGDD, Space Applications Centre (ISRO), Ahmedabad. The research work was carried out through Scientific Research and Training Division (SRTD) of Space Applications Centre, Ahmedabad.

डॉ सर्वेश्वर व्यास / Dr. S P Vyas
प्रधान, वैज्ञानिक अनुसन्धान एवं प्रशिक्षण प्रभाग
Head, Scientific Research and Training Division
एसआरटीडी-आरटीसीजी मीसा/ SRTD-RTCG-MISA
अंतरिक्ष उपयोग केंद्र (इसरो)
Space Applications Centre(ISRO)
अंतरिक्ष विभाग / Department of Space
भारत सरकार / Government of India
अहमदाबाद / Ahmedabad - 380015

ACKNOWLEDGEMENTS

I express my gratitude towards My Family for their support, **DR. S.P. VYAS**, Head of ERTD/VRG/EPSA, **SEU Ravi Patel, SEU Nishi Hathi, SEU Akanksha Wankhade** for all necessary support.

It is my pleasure to express my deep and sincere gratitude to my guide, **SHRI RAJENDRA GAIKWAD.** SCI/ENGR-SE EPSA-VRG-CGDD, SAC-ISRO, for giving me the opportunity to do work and providing valuable guidance throughout the internship. His dynamism, vision, sincerity to work, and motivation have profoundly influenced me. He has taught me the methodology to carry out and present the work as clearly as possible. I learned a lot of new technologies and frameworks. Working with different type of software and frameworks and new technologies is really exhilarating. To work and learn under his direction was a wonderful privilege and honour.

I want to convey my sincere appreciation and special thanks to **PROF. APURVA A. MEHTA**, my internal guide, for always supporting me. I would also like to thank **PROF. C.K. BHENSDADIA**, HOD of the CE Department, for allowing me this valuable internship at such a prestigious organization.

I am extremely thankful to all my professors at DDU for giving me valuable knowledge and for their consistent support.

Devansh P. Maru

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

This document outlines the functionality of an e-commerce website from the buyer, seller, and admin perspective. The buyer module includes features such as product display, category-wise search, product search, and cart and wishlist management. The seller module includes product addition, category management, and offer management. The admin module includes a dashboard, sales and user management.

The buyer module allows users to view all available products on the homepage and select a specific product for more information. Users can add the product to their cart or wishlist, view offers related to the product, and enroll in those offers. Additionally, users can search for products based on keywords and view products by category.

The cart module allows users to add products to their cart, view them on a cart page, and transfer products from the cart to the wishlist. Similarly, the wishlist module allows users to add products to their wishlist, view them on a wishlist page, and transfer products from the wishlist to the cart. The profile management module allows users to manage their account and view transaction history.

The seller module allows sellers to add new products to the database along with product details such as name, price, category, image, and description. Sellers can also view detailed information about their products and edit or delete them as needed. The category module allows sellers to add or delete categories, while the offer module allows them to view and approve offers made by buyers.

The admin module includes a dashboard that displays total sales, the number of products, users, and sales on a particular day, and a chart of month-wise sales. The sales management module allows admins to view transaction details, such as product names, prices, quantities, total prices, transaction IDs, and dates. The user management module allows admins to view user details and take appropriate actions as needed.

Overall, this e-commerce website allows buyers to view and purchase products, sellers to manage their products and categories, and admins to manage sales and users. The platform is designed to be user-friendly, with easy navigation and clear instructions for each module. With this platform, sellers can increase their reach and buyers can access a wide range of products and services.

CHAPTER 1
INTRODUCTION

E-commerce application in which user can buy or sell any product. Also, admin can monitor all the sales details and to do user management.

Basically, three types of users are in web Application:

- Admin
- Buyer
- Seller

Brief Features:

Buyer:

- In this web application, Buyer can login to their account and see all the product which is listed by the all the seller across the web application.
- User can also filter products category wise.
- User can add specific product to the cart, whishlist and take part in the bid of any product.

Seller:

- Seller can add the product which is listed on the user side.
- Also, they can add the category of the product.
- Seller can sell their products via auctions.

Admin:

- Admin can do analyse of all sales and done user management.

CHAPTER 2
ABOUT THE SYSTEM AND SRS

Tools/Technologies

Backend:

PHP: PHP is used as backend in whole project. All authentication is done in PHP.

Frontend:

HTML5, CSS, Bootstrap: Whole frontend is made using HTML, Bootstrap and customised CSS.

Tools:

Visual Studio Tools, Xammp local server (MySQL and Apache server)

How Run project on local server

- To run this project, we need to run the apache server and MySQL server on xammp application panel.
- Need to shift project data folder inti htdocs Folder in the xammp folder.
- Port no:8080
- Then On the browser type **http://localhost:8080/ecommerce/** which take us on home page of the website.

Note: MySQL server generally works on port no:80 but firstly check the port and according to that use that port.

Functional Requirements:

A Functional Requirement (FR) is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behaviour, and outputs.

In this FR all the functionality is describe as per user role.

R1 Registration:

R.1.1 Sign up a User:

A user that has no existing account can sign in to our website and will be added to the database.

Input:

o Name, Email, Address, Phone no, address, password

Output:

o User is redirected to the login page and the user info is added to the database.

R.1.2 Login by user:

Existing user can login with his/her credentials.

Input:

o Email, Password

Output:

o If the credentials are correct user is redirected to the home page else the error message is displayed

Buyer side

R2 View Product:

R.2.1 product details:

- User can see the details of different product.

Input:

- Select the product, product name

Output:

- Open the product view page

R3 Cart Management:

R.3.1 Add to cart:

- User can add the product into the cart.

Input:

- Product id and quantity

Output:

- Information message and product added to cart.

R.3.2 Delete the product from cart

- User can delete the product from the cart.

Input:

- Product id and quantity

Output:

- Display the message and delete the item from cart

R.3.3 update quantity:

- User can update the quantity in the cart.

Input:

- Select the quantity

Output:

- Display the quantity and add it into the total price.

R.3.4 move to wishlist:

- User can move the product from cart to the wishlist.

Input:

- Product id and quantity

Output:

- Display the message and product move to the wishlist

R4 wishlist management:

R.4.1 Add to wishlist:

- User can add specific product to the wishlist.

Input:

- Product id

Output:

- Display the message and product added to the wishlist.

R.4.2 Remove product from wishlist:

- User can remove product from the wishlist.

Input:

- Product id

Output:

- Display message and remove product from wishlist

R.4.3 move to the cart:

- User can add the specific product into the cart.

Input:

- Product id and quantity

Output:

- Display the message and product move from wishlist to cart.

R5 offer module

R.5.1 offer on the product:

- User can offer on the specific product.

Input:

- offer amount

Output:

- Successfully enrol in offer and show he status on result page.

R6 Profile management:

R.6.1 edit Profile:

- User can edit their profile details like mobile number, address, photo etc.

Input:

- Details that to be change

Output:

- Update the specific field.

Seller side

R7 product management:

R.7.1 Add product:

- Seller can add new product which can be display at user side

Input:

- Name, category, price, photo, Description

Output:

- Display Message and product add to the database

R.7.2 Update product:

- Seller can update the product which is inserted by himself/herself.

Input:

- Name, category, price, photo, Description (whichever want be changed)

Output:

- Display Message and Update the product details

R.7.3 View Product Description:

- Seller can show the full description

Input:

- Product id

Output:

- Display the product description

R.7.4 Delete Product:

- Seller can delete their product.

Input:

- Product id

Output:

- Display message and delete product

R8 category management:

R.8.1 Add Category:

- Seller can add the Category of product.

Input:

- Category name

Output:

- Display message and add the category

R.8.2 Delete Category:

- Seller can delete Category

Input:

- Category id

Output:

- Display message and delete from database.

R9 Bid Module:

R.9.1 View product offer:

- Seller can view bid on the specific product

Input:

- Product id

Output:

- Product name, description, expected price, User details whoever bid on that specific product

R.9.2 sold product in offer module:

- Seller can sell products to specific user

Input:

- Product id, User id, bid amount

Output:

- Close the bid and update the status

Admin side

R10 User management:

R.10.1 Edit User information:

- Admin can Edit user information

Input:

- Email, first name, last name, address, contact no (whichever needs to change)

Output:

- Display message and Update the information

R.10.2 Delete User:

- Admin can delete user.

Input:

- User id

Output:

- Display message and delete user.

R.10.3 View Cart:

- Admin can see the cart of specific user at admin side

Input:

- User id

Output:

- Product name, quantity etc.

R11 Sales management:

R.11.1 View Transaction:

- Admin can see all the transaction whichever done across the website.

Input:

- sales id

Output:

- date, Buyer Name, transaction id, amount

CHAPTER 3
ANALYSIS

Diagram
Use-case Diagram

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

Use-case diagrams illustrate and define the context and requirements of either an entire system or the important parts of the system. You can model a complex system with a single use-case diagram or create many use-case diagrams to model the components of the system. You would typically develop use-case diagrams in the early phases of a project and refer to them throughout the development process.

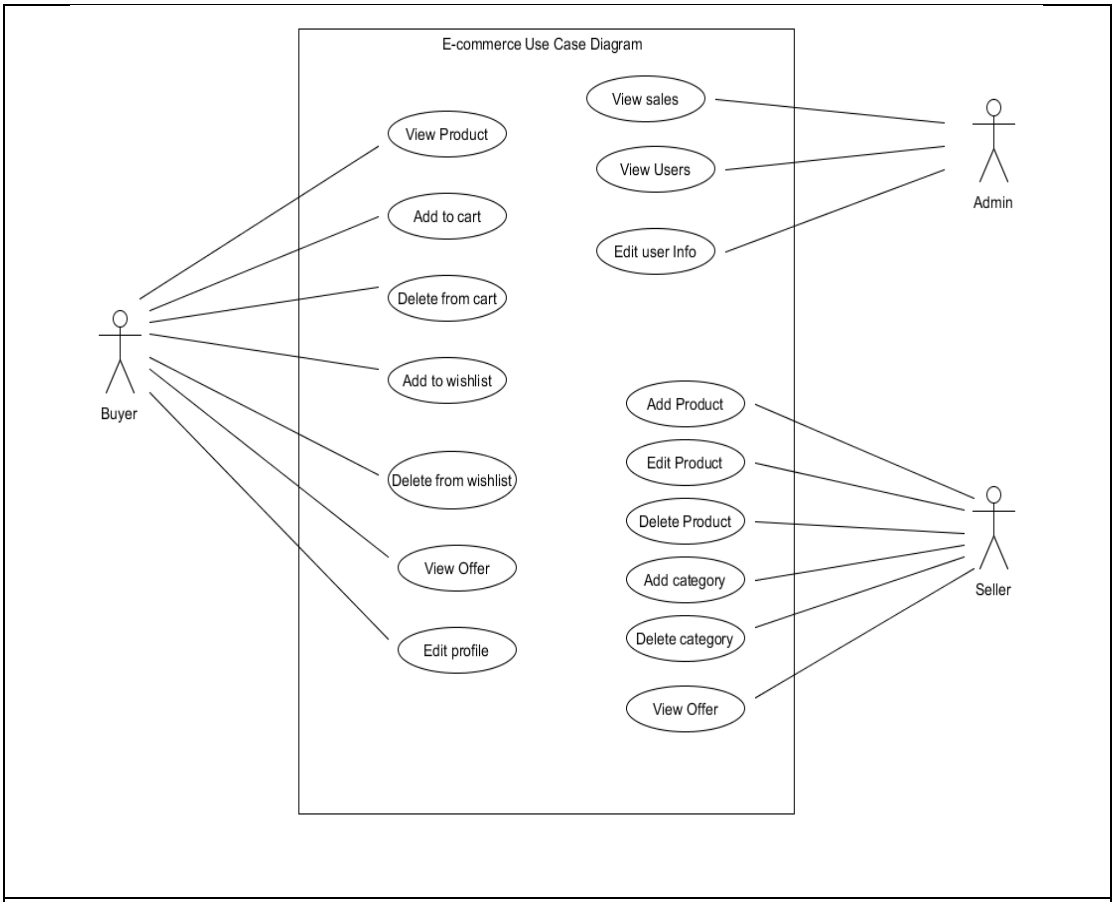


Fig:1.0 Use case Diagram

ER Diagram

ER Diagram gives a better understanding of the information to be stored in a database. It reduces complexity and allows database designers to build databases quickly. It helps to describe elements using Entity-Relationship models. It allows users to get a preview of the logical structure of the database.

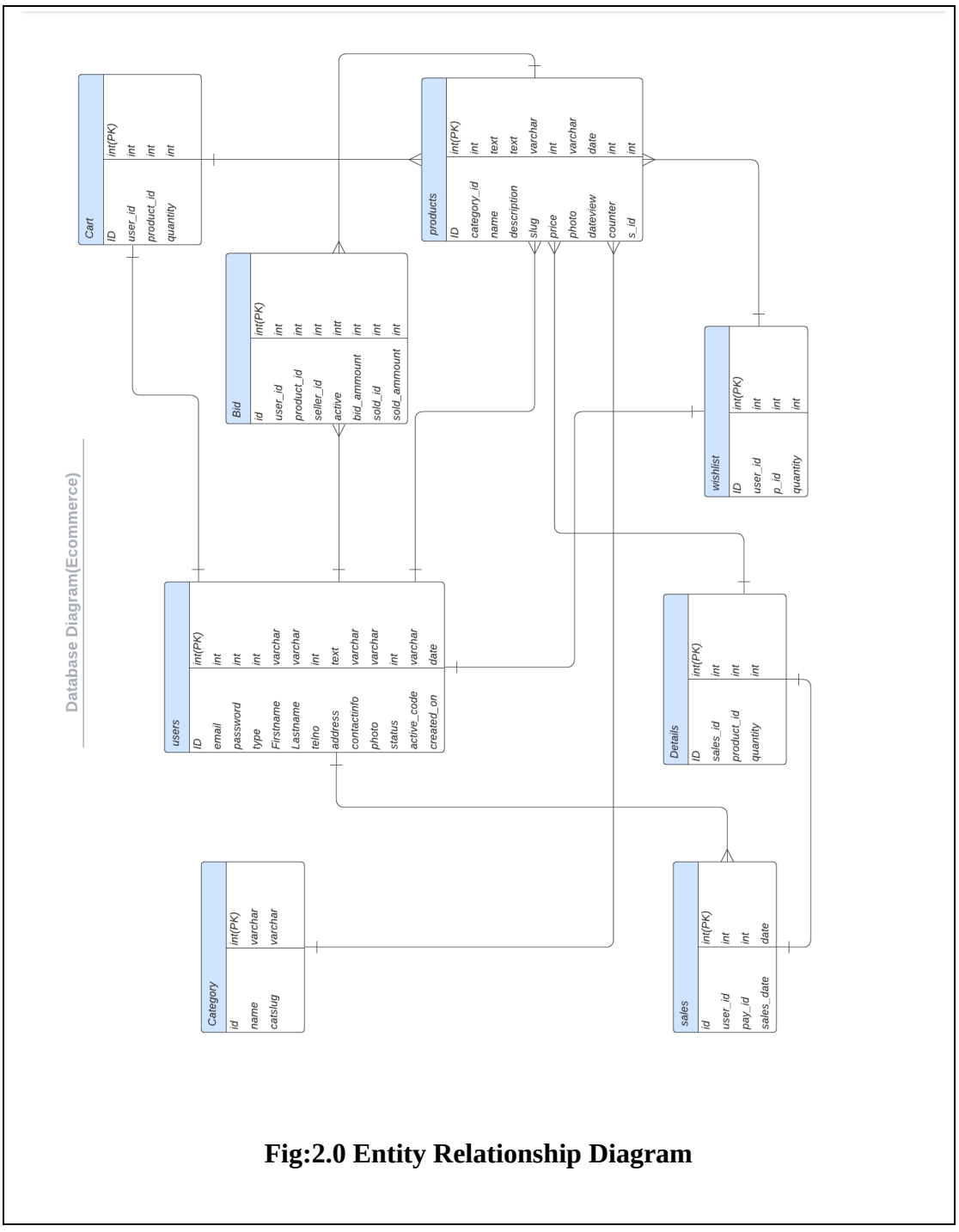


Fig:2.0 Entity Relationship Diagram

Data Flow Diagram

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both. It shows how data centres and leaves the system, what changes the information, and where data is stored. The objective of a DFD is to show the scope and boundaries of a system. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

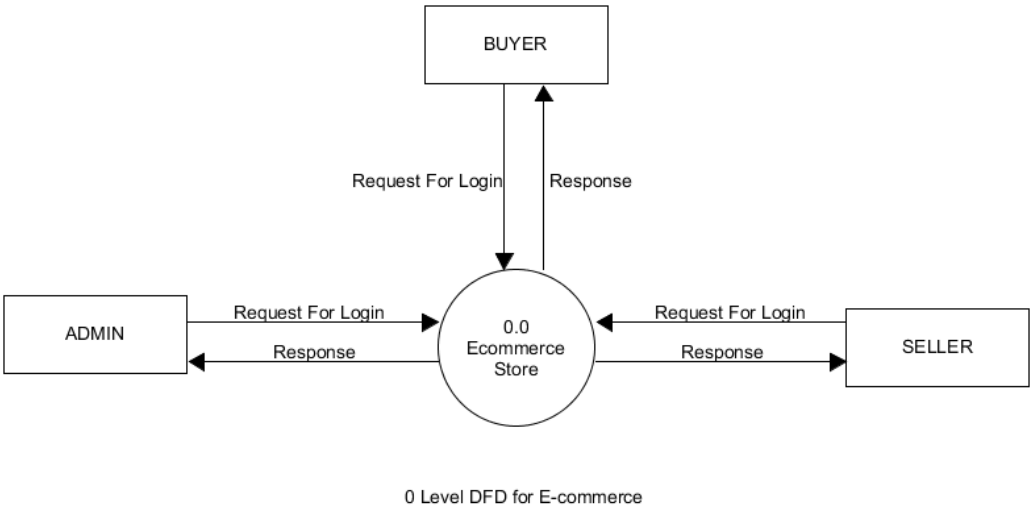


Fig:3.0 0 Level DFD for E-commerce

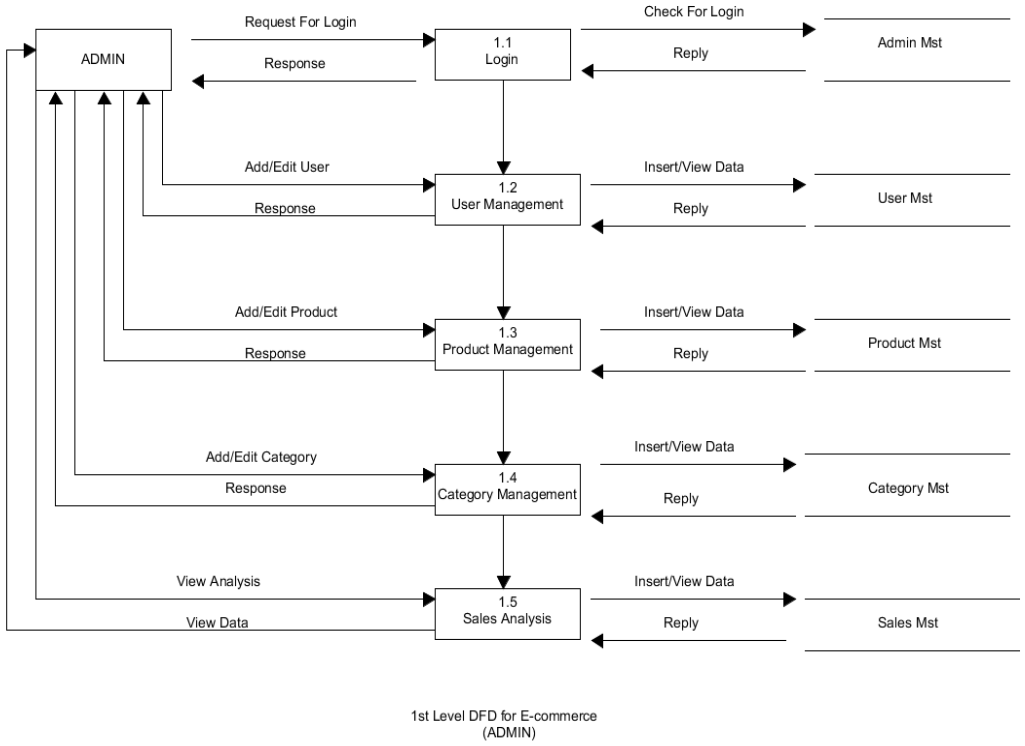


Fig:3.1 1st Level DFD for E-commerce(Admin)

In above fig 3.1, admin's details are fetching from database. After authentication and authorization user related data, product related data, Category and sales related data is stored respectively database tables.

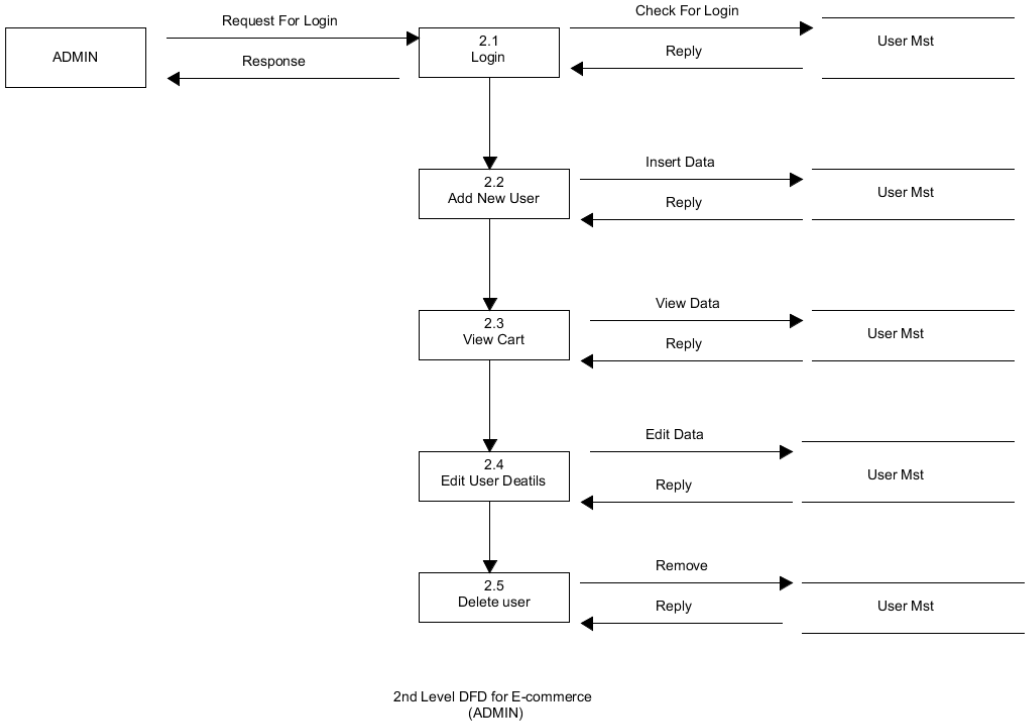


Fig:3.2 2nd Level DFD for E-commerce(Admin)

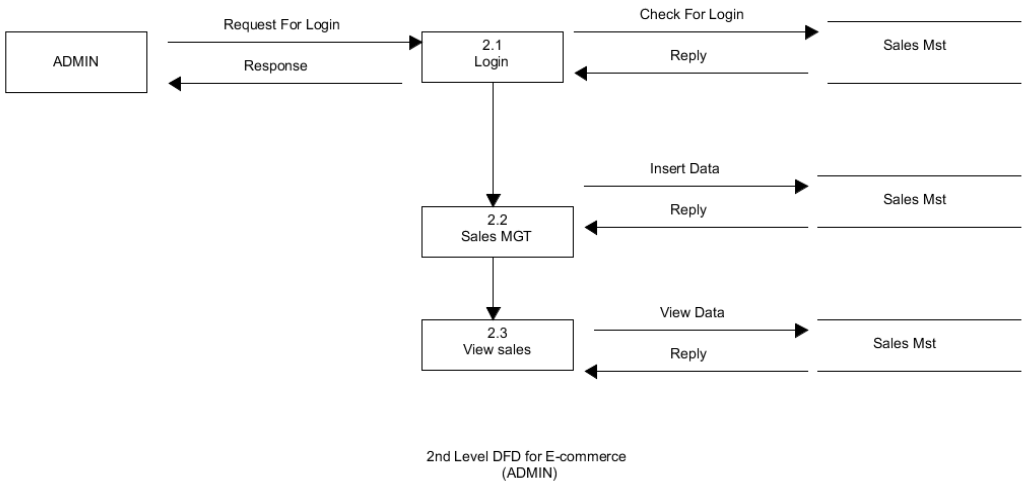


Fig:3.3 2nd Level DFD for E-commerce(Admin)

In above fig 3.2/fig 3.3, data of the user can store at user table and related users' cart and other details are store at user table and after fetching from that table.

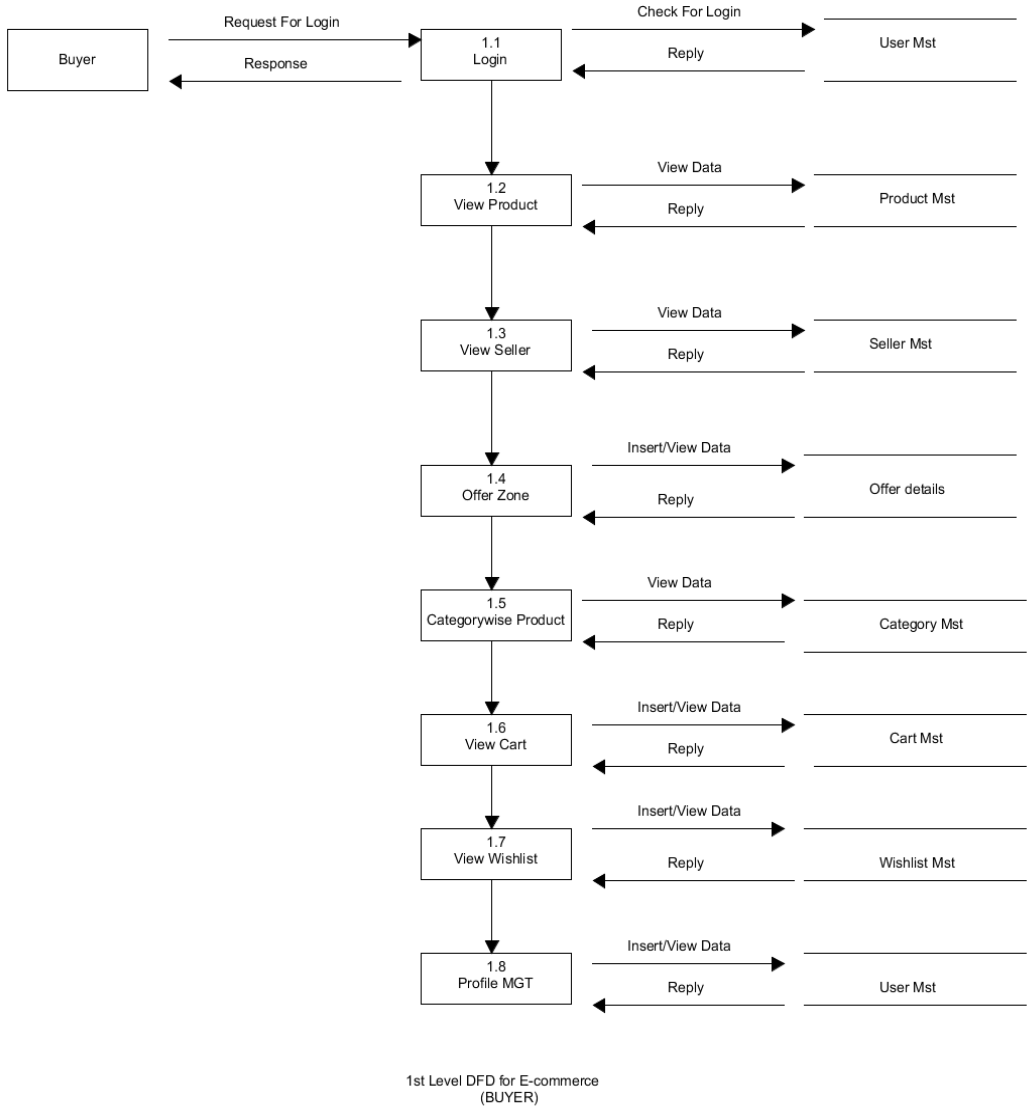


Fig:3.4 1st Level DFD for E-commerce(Buyer)

In above fig 3.4, users' details are fetching from database. After authentication and authorization product details, seller details, cart and wishlist related data can store at respective tables.

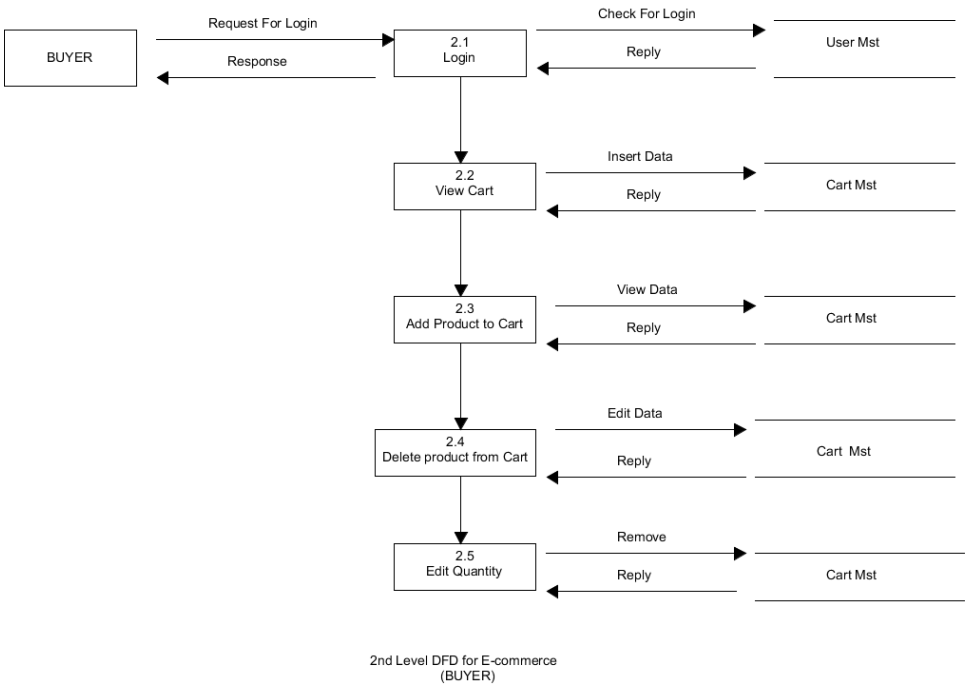


Fig:3.5 2nd Level DFD for E-commerce(Buyer)

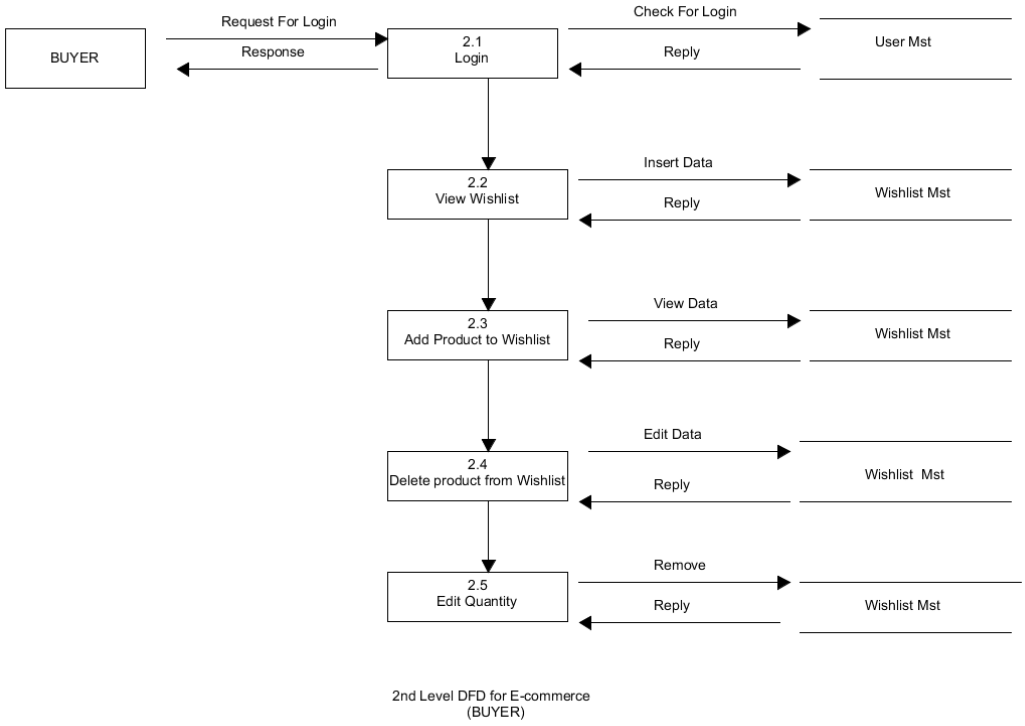


Fig:3.6 2nd Level DFD for E-commerce(Buyer)

In previous fig 3.5/fig 3.6, users can add the product's data into cart tables and view, delete and edit that data in cart table. Same as cart, whishlist is implement and data flow is same as cart.

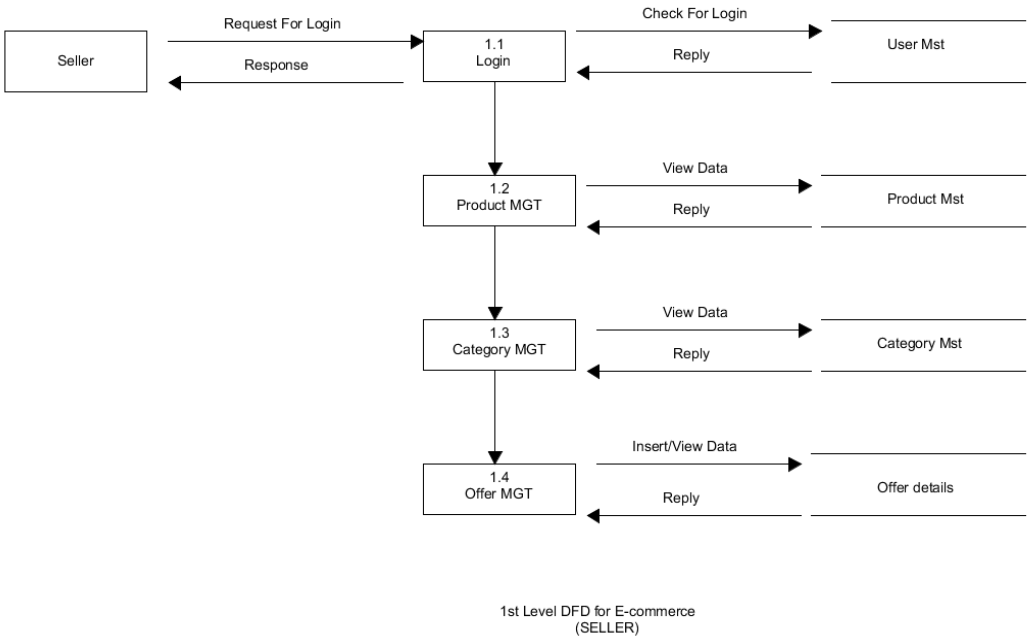


Fig:3.7 1st Level DFD for E-commerce(Seller)

In above fig 3.7, seller's details are fetching from database. After authentication and authorization product management, category management, offer management is fetching from respective tables.

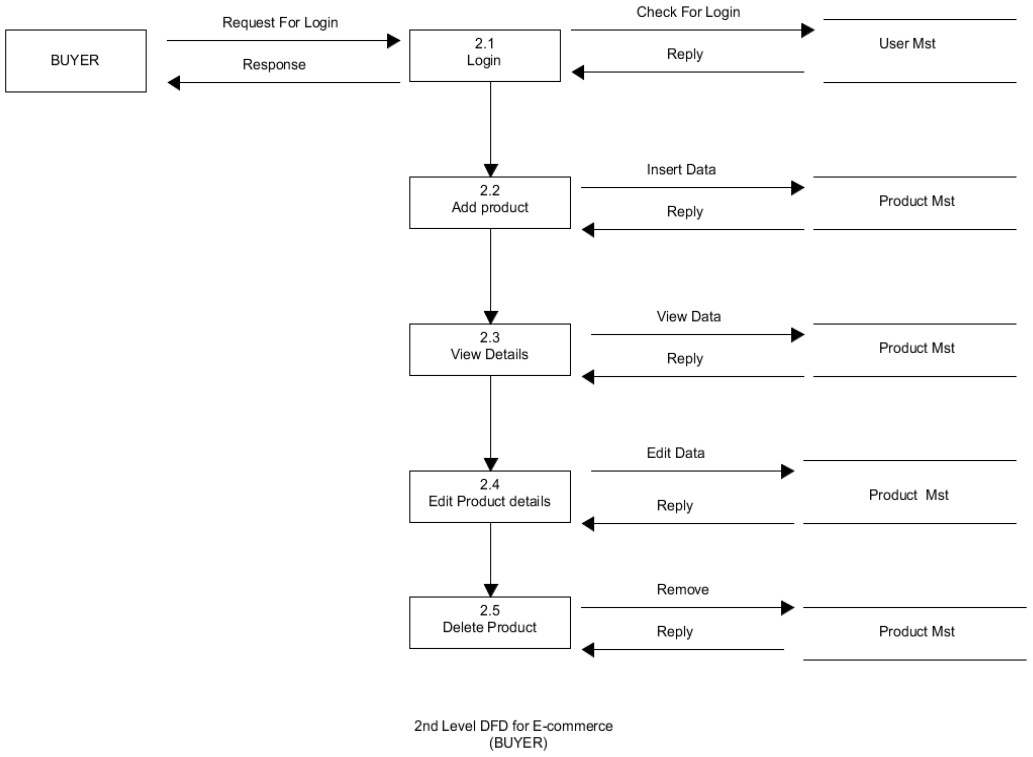


Fig:3.8 2nd Level DFD for E-commerce(Seller)

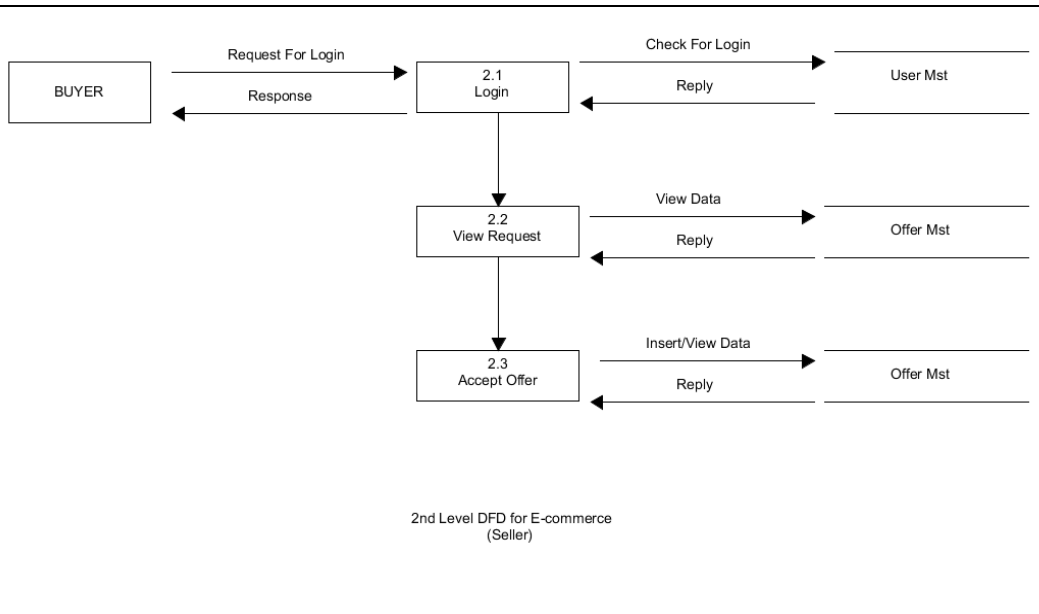


Fig:3.9 2nd Level DFD for E-commerce(Seller)

In above fig 3.8/fig 3.9, seller can add the product's data into products tables and view, delete and edit that data in product table. In offer zone seller can store their details.

CHAPTER 4
DESIGN

Data Dictionary

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(100)			No	*None*		AUTO_INCRI
2	product_id	int(100)			No	*None*		
3	seller_id	bigint(255)			No	*None*		
4	user_id	bigint(255)			No	*None*		
5	bid_ammount	bigint(255)			No	*None*		
6	active	int(10)			No	*None*		
7	sold_id	int(11)			No	*None*		
8	sold_ammount	int(255)			No	*None*		

Fig:4.1 bid table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCRE
2	user_id	int(11)			No	*None*		
3	product_id	int(11)			No	*None*		
4	quantity	int(11)			No	*None*		

Fig:4.2 cart table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCRI
2	name	varchar(100)	latin1_swedish_ci		No	*None*		
3	cat_slug	varchar(150)	latin1_swedish_ci		No	*None*		

Fig:4.3 category table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	None		AUTO_INCRE
2	sales_id	int(11)			No	None		
3	product_id	int(11)			No	None		
4	quantity	int(11)			No	None		

Fig:4.4 details table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	None		AUTO_INCREME
2	category_id	int(11)			No	None		
3	name	text	latin1_swedish_ci		No	None		
4	description	text	latin1_swedish_ci		No	None		
5	slug	varchar(200)	latin1_swedish_ci		No	None		
6	price	double			No	None		
7	photo1	varchar(200)	latin1_swedish_ci		No	None		
8	photo2	varchar(200)	latin1_swedish_ci		Yes	NULL		
9	date_view	date			No	None		
10	counter	int(11)			No	None		
11	s_id	int(20)			Yes	NULL		

Fig:4.5 product table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	None		AUTO_INCR
2	user_id	int(11)			No	None		
3	pay_id	varchar(50)	latin1_swedish_ci		No	None		
4	sales_date	date			No	None		

Fig:4.6 sales table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCR
2	email	varchar(200)	latin1_swedish_ci		No	*None*		
3	password	varchar(60)	latin1_swedish_ci		No	*None*		
4	type	int(1)			No	*None*		
5	firstname	varchar(50)	latin1_swedish_ci		No	*None*		
6	lastname	varchar(50)	latin1_swedish_ci		No	*None*		
7	address	text	latin1_swedish_ci		No	*None*		
8	contact_info	varchar(100)	latin1_swedish_ci		No	*None*		
9	photo	varchar(200)	latin1_swedish_ci		No	*None*		
10	status	int(1)			No	*None*		
11	activate_code	varchar(15)	latin1_swedish_ci		No	*None*		
12	reset_code	varchar(15)	latin1_swedish_ci		No	*None*		
13	created_on	date			No	*None*		
14	telno	int(255)			Yes	*NULL*		

Fig:4.7 user table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCRE
2	user_id	int(11)			No	*None*		
3	product_id	int(11)			No	*None*		
4	quantity	int(11)			No	*None*		

Fig:4.8 wishlist table

CHAPTER 5
IMPLEMENTATION DETAILS

Buyer Side

Product module:

In this module user can see all the product in the home page. When user click/select any specific product then detail description of that product will show on screen. On that page user can add product to cart, wishlist and view the offer and enrol in that offer.

Category module:

In this module user can see the product category wise.in the category section user can see all the available category.

Search Functionality:

In this functionality user can search for any keyword of the product name. If the searched product available, then display on the screen otherwise information message will display.

Cart module:

In this module user can add any product into cart and display it on the view cart page. On view cart page user can see all products that added in cart and at the end total is displayed. Also, beside every product one button is there via that product is transferred from cart to wishlist.

Wishlist module:

In this module user can add any product into wishlist and display it on the view wishlist page. On view wishlist page user can see all products that added in Wishlist. Also, beside every product one button is there via that product is transferred from wishlist to cart.

Profile Management:

In this module user can manage their account and user profile related information.

On this page user can see previous transaction history.

Seller Side

Product module:

In this module seller can add the new product which will be displayed on the buyer side and add to the database. Products details like product name, price, category, image and description. Also, seller can see the detailed description of the specific product and can edit any information about the product or delete the product.

Category:

In this section seller can add the new category of product. Same as seller can delete category.

Offer Module:

In this section seller can see all the request come from the buyer and approved the specific offer. After that process updated information will be displayed at buyer side.

Admin Side

Dashboard:

In this dashboard details like total sales, total number of products, total number of users, sales on particular day and display the chart of month wise sales.

Sales management:

In this module admin can see all the transaction that can make across the website along with transaction id. Also, admin can see the detailed transaction information in which products name, price, quantity, total price, transaction id, date of the transaction.

User Management:

In this module admin can done all the work which is related to user management. Admin can see all the details about the user and can take appropriate action on specific user.

Front End Interface

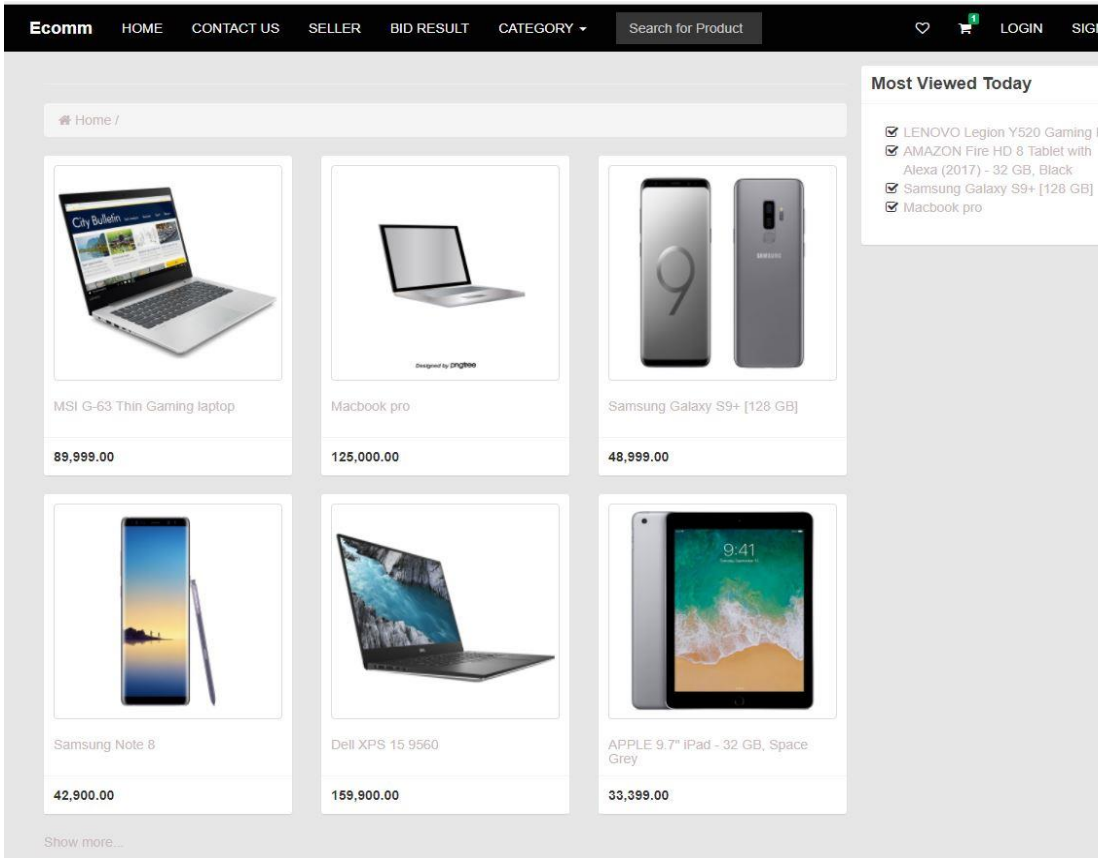


Fig 5.1 Home Page

In this page buyer can see all the available products across the website. Button of seller, offer zone, category, wishlist, cart is shown at the top of the page.

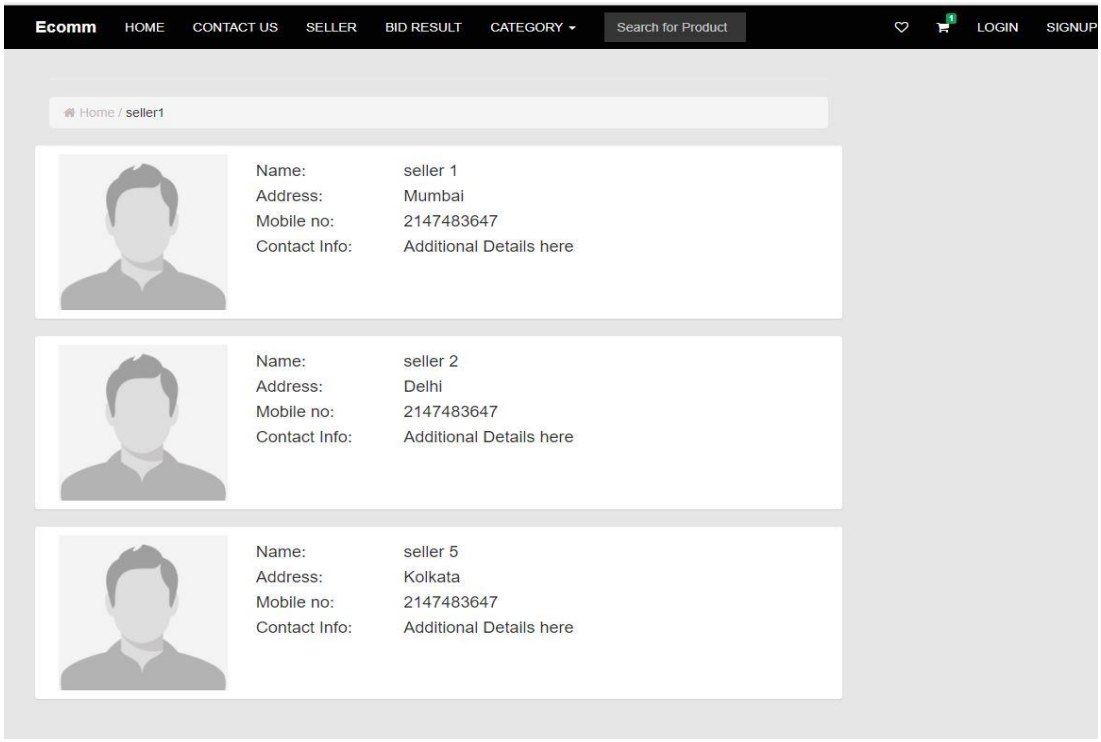


Fig 5.2 Seller Page (Buyer)

In this page user can see al the seller across the website.

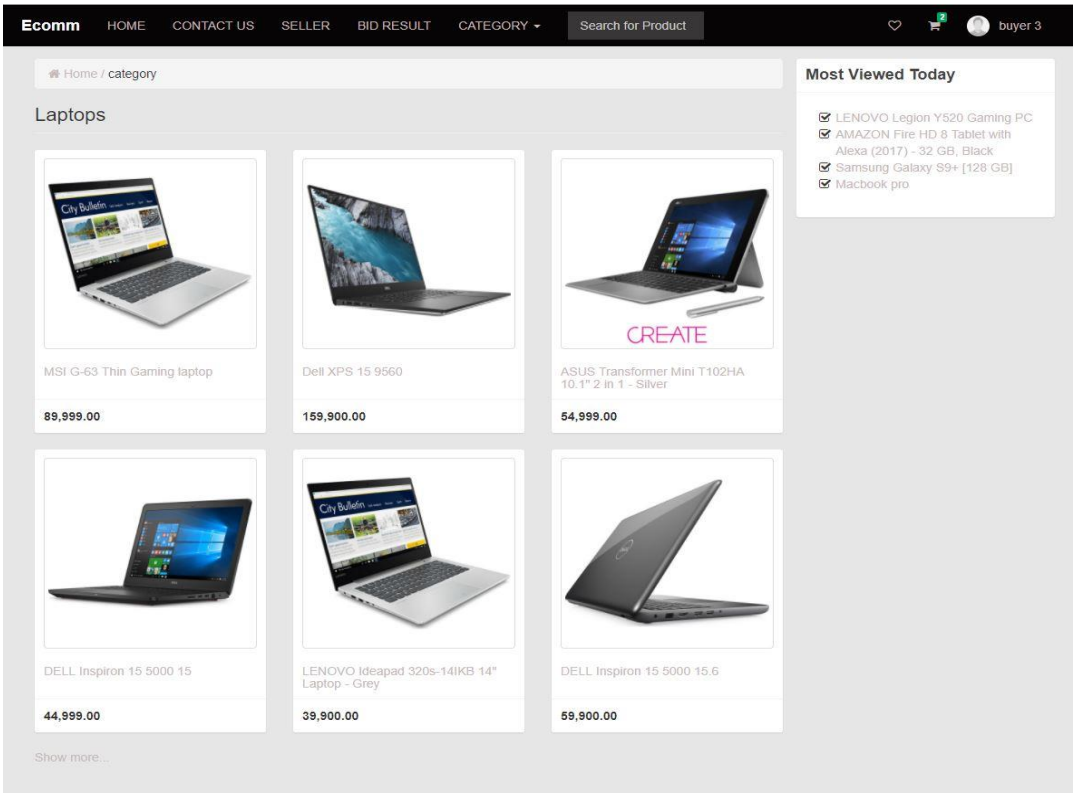


Fig 5.3 Category Page (Buyer)

In this category page user can filter the product across different available categories.

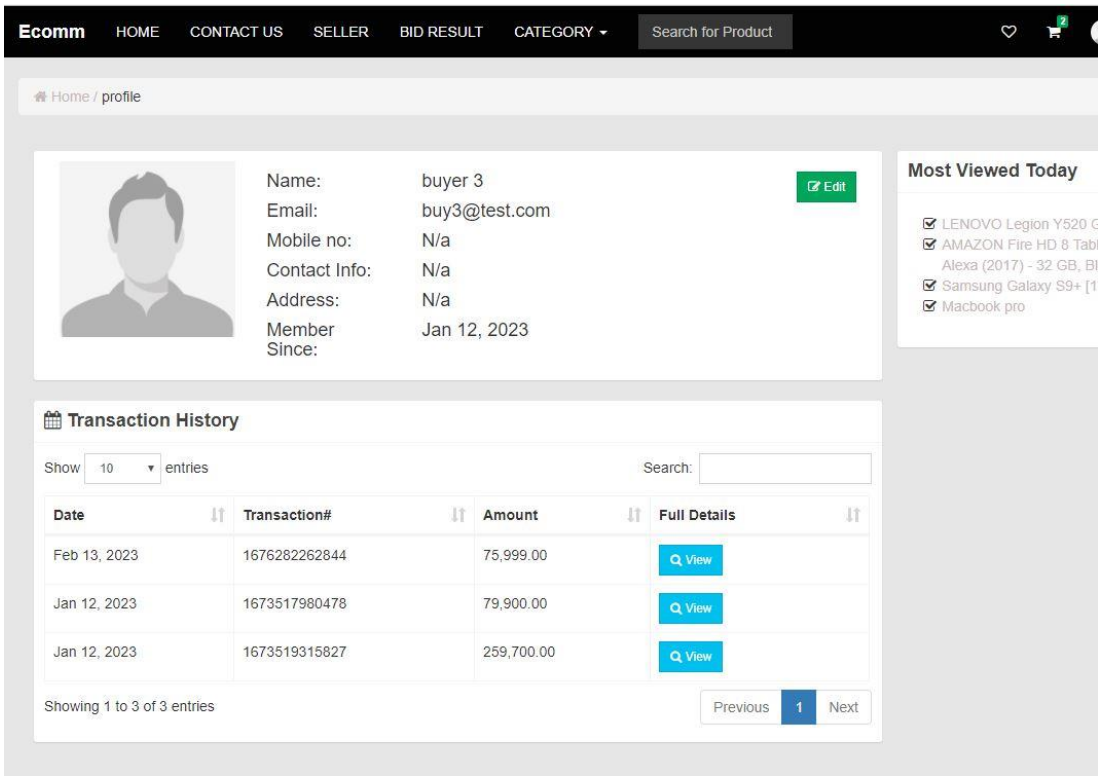


Fig 5.4 Profile Page (Buyer)

In the profile page user can see their profile and previous transaction.

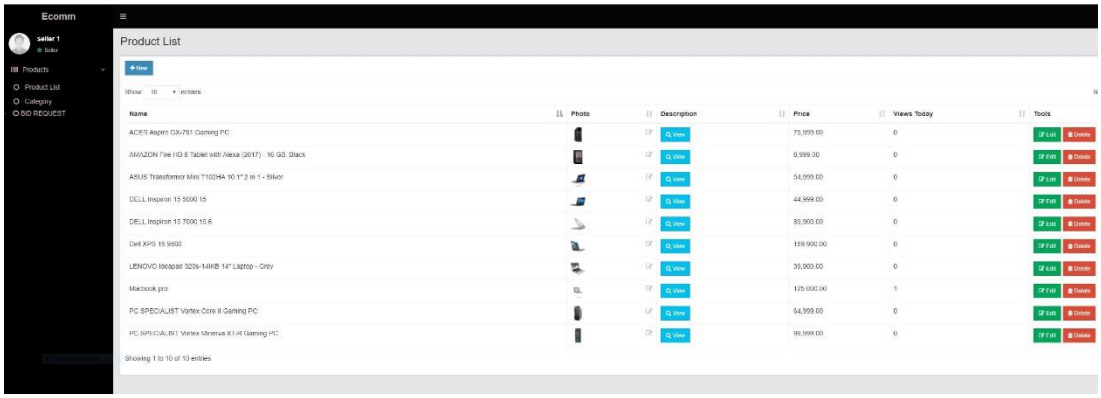


Fig 5.5 Product Page (Seller)

In this page seller can add new product and view all product that add by himself/herself.

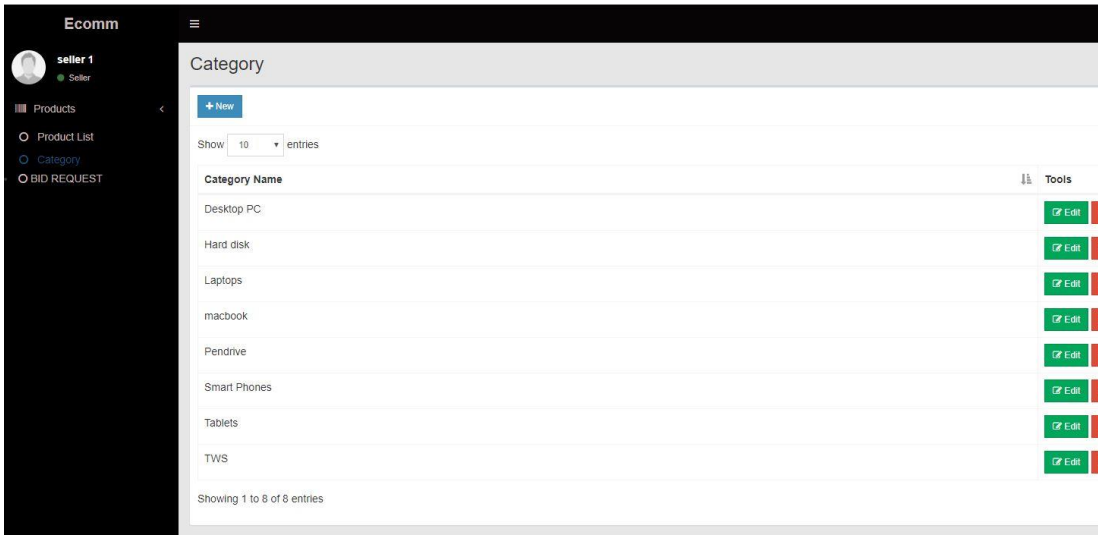


Fig 5.6 Category Page (Seller)

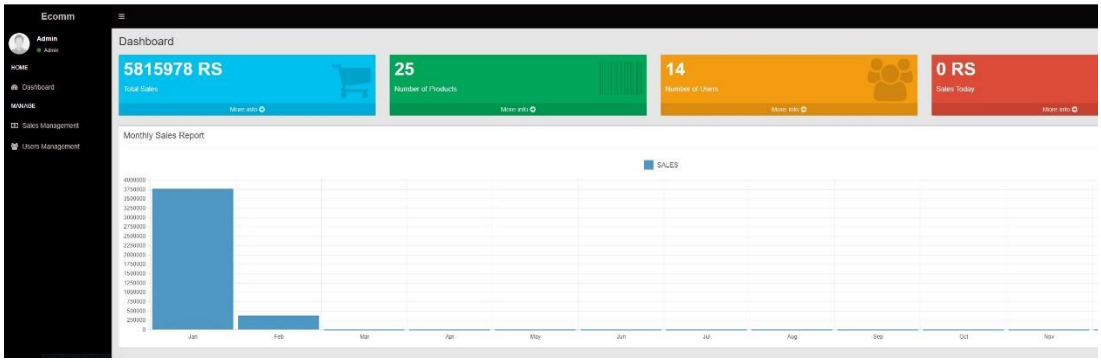


Fig 5.7 Sales Analysis Page (Admin)

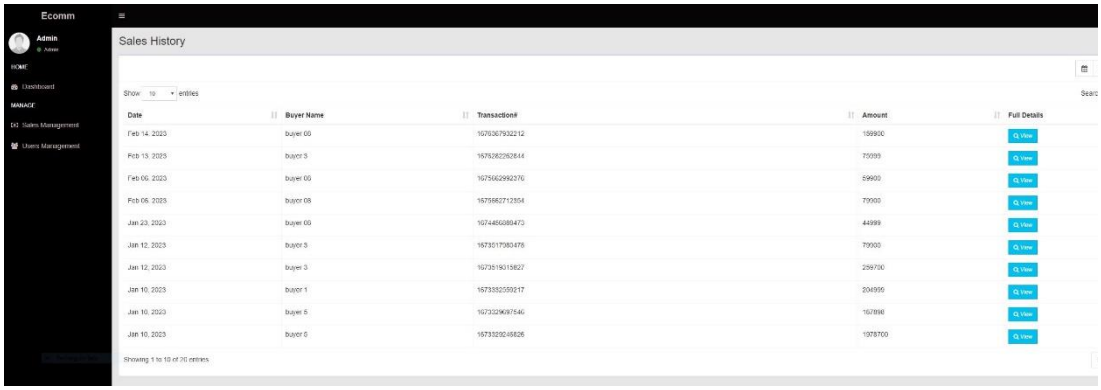


Fig 5.8 Sales History Page (Admin)

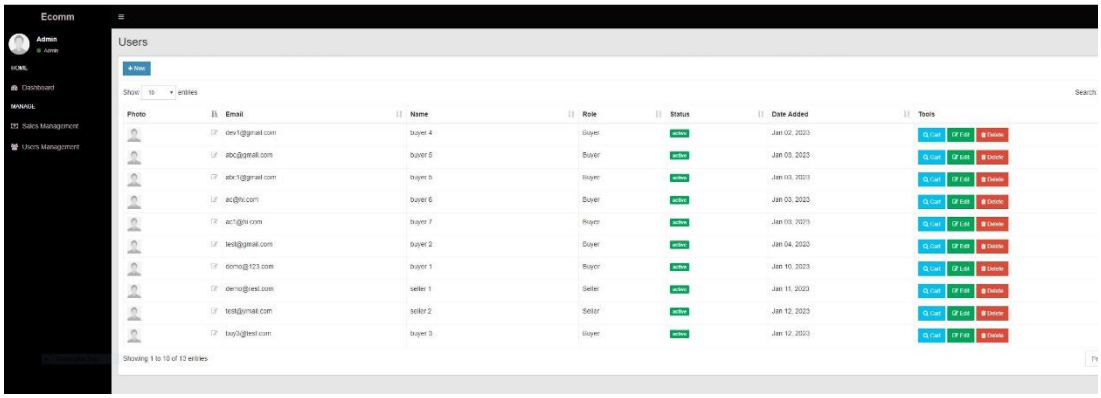


Fig 5.9 User Mgt Page (Admin)

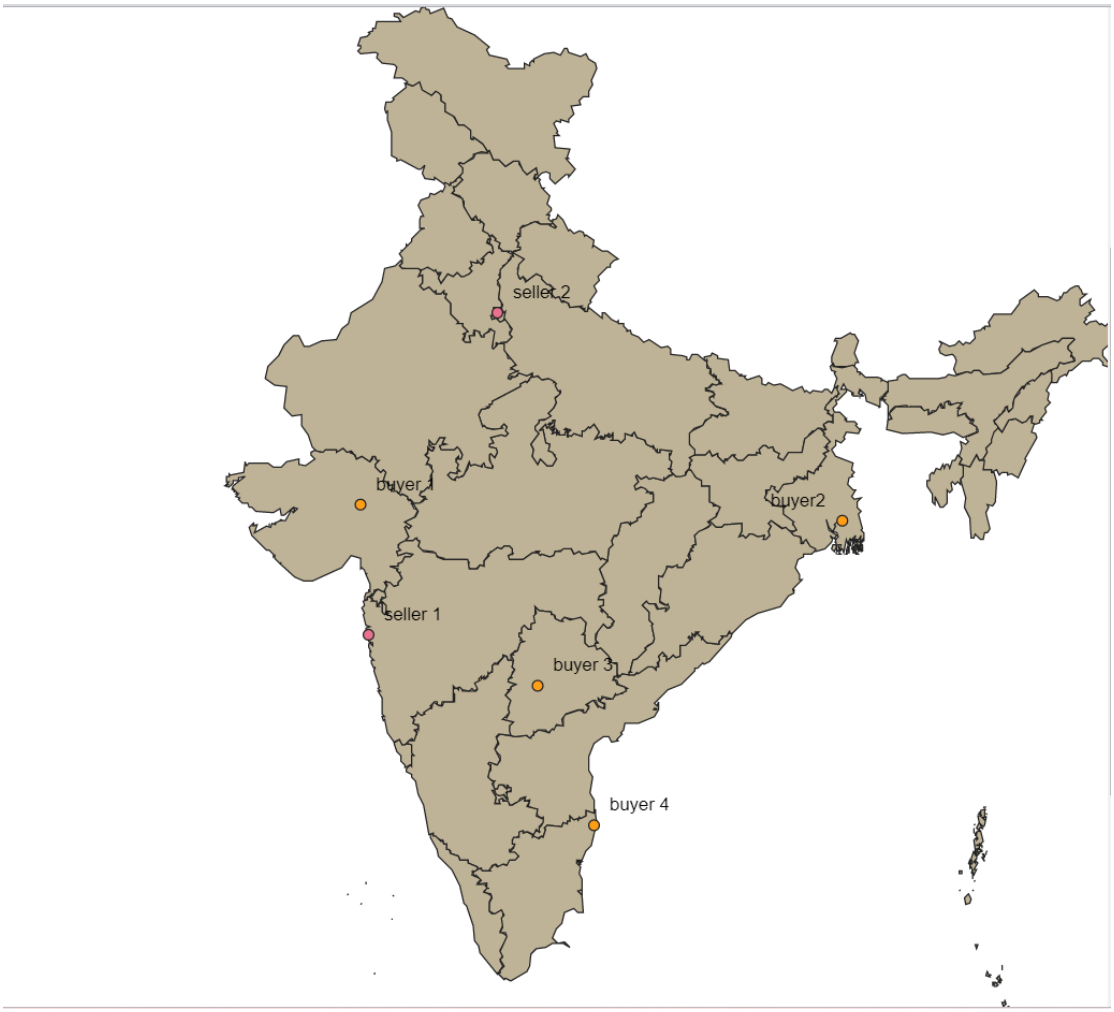
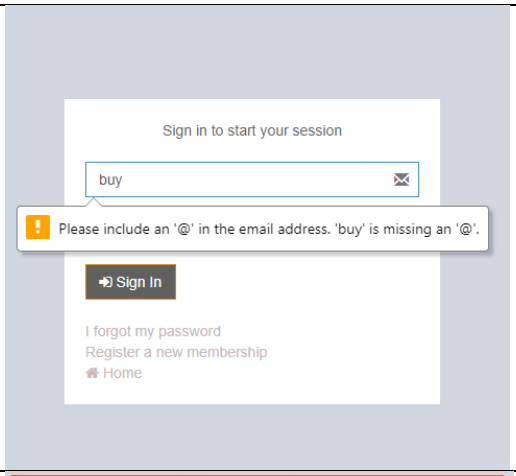
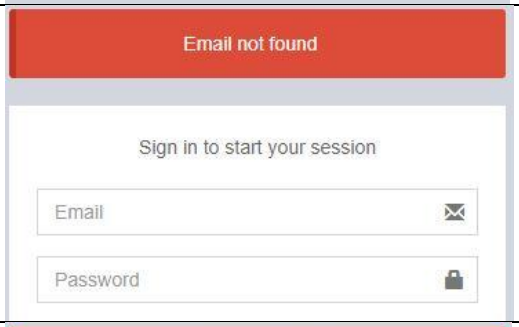
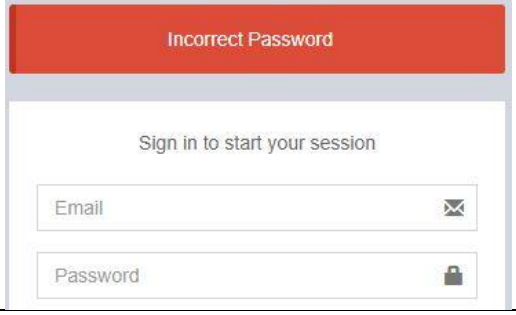
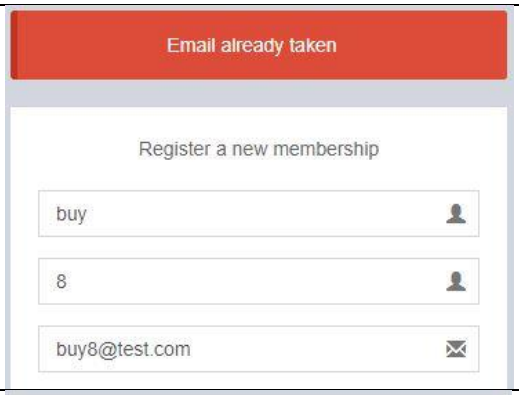
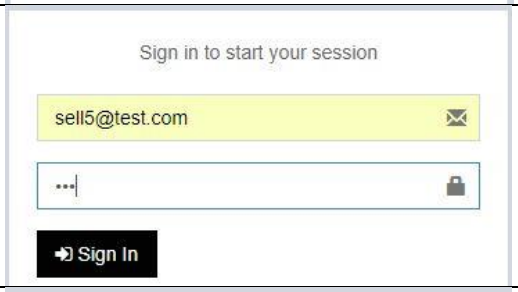
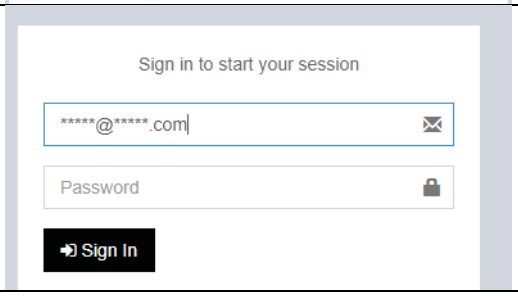
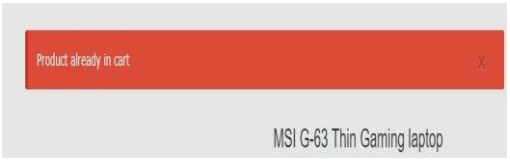
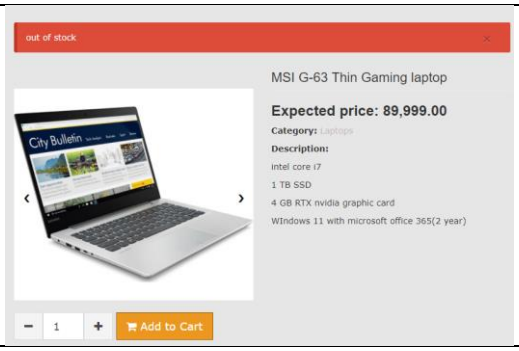


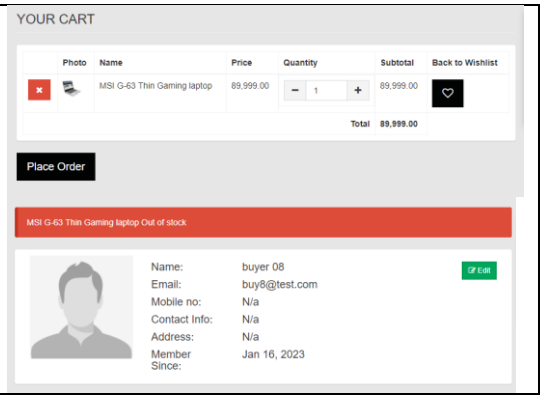
Fig 5.10 Users on map (Admin)

CHAPTER 6
TESTING

MANUAL TESTING

Test case name	Test name number	Result	Photo
User validation	Testcase1	pass	
	Testcase2	pass	
	Testcase3	pass	

	Testcase4	pass	
	Testcase5	pass	
	Testcase6	pass	
Item already into cart	Testcase1	pass	
Transaction Message	Testcase1	pass	
Validation of quantity	Testcase1	pass	

	Testcase2	pass	

CHAPTER 7
CONCLUSION AND FUTURE EXTENSION

Conclusion

The website is an online marketplace that caters to both buyers and sellers. The website offers a product module that allows users to browse through various products and view product details, add items to the cart or wishlist, and enroll in offers. The category module categorizes products for easy navigation, and the search functionality allows users to search for specific products. The cart and wishlist modules enable users to manage their preferred items.

On the seller side, the website allows sellers to add new products, categorize them, and manage offers. The seller can also edit or delete products and categories.

On the other hand, the admin side of the website provides dashboard features that show details about total sales, total number of products, users, and sales. The sales management module allows admins to view detailed transaction information, and user management enables the admin to take appropriate action on specific users. Overall, the website offers a convenient platform for buying and selling products.

Future extension

Product Reviews and Ratings: Allow users to leave reviews and ratings for the products they have purchased. This can help other users in making their buying decision.

Advanced Search Functionality: Enhance the search functionality by including filters for price range, brand, color, size, etc. This will make it easier for users to find the exact product they are looking for.

Multi-language Support: Add support for multiple languages to cater to users from different regions and countries.

Mobile App: Develop a mobile application for the website to make it more accessible and user-friendly for users who prefer to shop on their mobile devices.

Live Chat Support: Implement a live chat support feature to assist users in real-time with their queries and concerns related to the products or the website.

Personalized Recommendations: Utilize machine learning algorithms to provide personalized product recommendations to users based on their previous purchases and browsing history. This can increase customer engagement and sales.

Introduction to Neo4j Graph Database

Comparison with different Database:

Neo4j	MongoDB	PostgreSQL	MySQL
Build on java and Scala	Build on c++ ,Go,javascript,python	Build on c	Build on c and c++
Not support Map reduce	Support Map reduc–e	Not support Map reduce	Not support Map reduce
It can have foreign key between two different tables.	It can't have foreign key between two different tables.	It can have foreign key between two different tables. (we can use join query for this)	It can have foreign key between two different tables. (we can use join query for this)
Does not support Standard SQL query.	Support read only SQL query	Support SQL query	Support SQL query
Graph based DBMS	Document based DBMS	Relational DBMS	Relational DBMS
Does not supporting Partitioning Methods	Support sharding partitioning method.	Supports horizontal partitioning, vertical partitioning	Supports horizontal partitioning, sharding with MySQL cluster or MySQL Fabric.

Comparison Between MySQL And Neo4j(Performance):

Depth	Execution Time-MySQL	Execution Tine-Neo4j
2	0.016	0.01
3	30.267	0.168
4	1543.505	1.359
5	>1 hour	2.132

Graph based database

❖ **In which scenario, we can use Neo4j database?**

- Analytics and data science
- Fraud Detection
- Knowledge Graph
- Real time Recommendations
- Supply chain management
- Identity and access management
- Social Networking

❖ **Advantages:**

- Flexible data model
- Real time insights
- High availability
- Connected and semi structure data
- Easy retrieval
- No joins

❖ **Disadvantages:**

- Graph size
- No security provided at data level
- No data encryption

❖ **Building Blocks:**

- Nodes
- Properties
- Relationships
- Labels
- Data browser

Installation of Neo4j:

1. If it is not already installed, get **OpenJDK 17** or **Oracle Java 17**.

2. Download the latest release from /download-center [Neo4j Download Center]. Select the appropriate ZIP distribution.

    a. Make sure to download Neo4j from /download-center[Neo4j Download Center] and always check that the SHA hash of the downloaded file is correct:
    b. To find the correct SHA hash, go to Neo4j Download Center and click on SHA_256 which will be located below your downloaded file.
    c. Using the appropriate commands for your platform, display the SHA-256 hash for the file that you downloaded.
    d. Ensure that the two are identical.

3. Right-click the downloaded file, click Extract All.

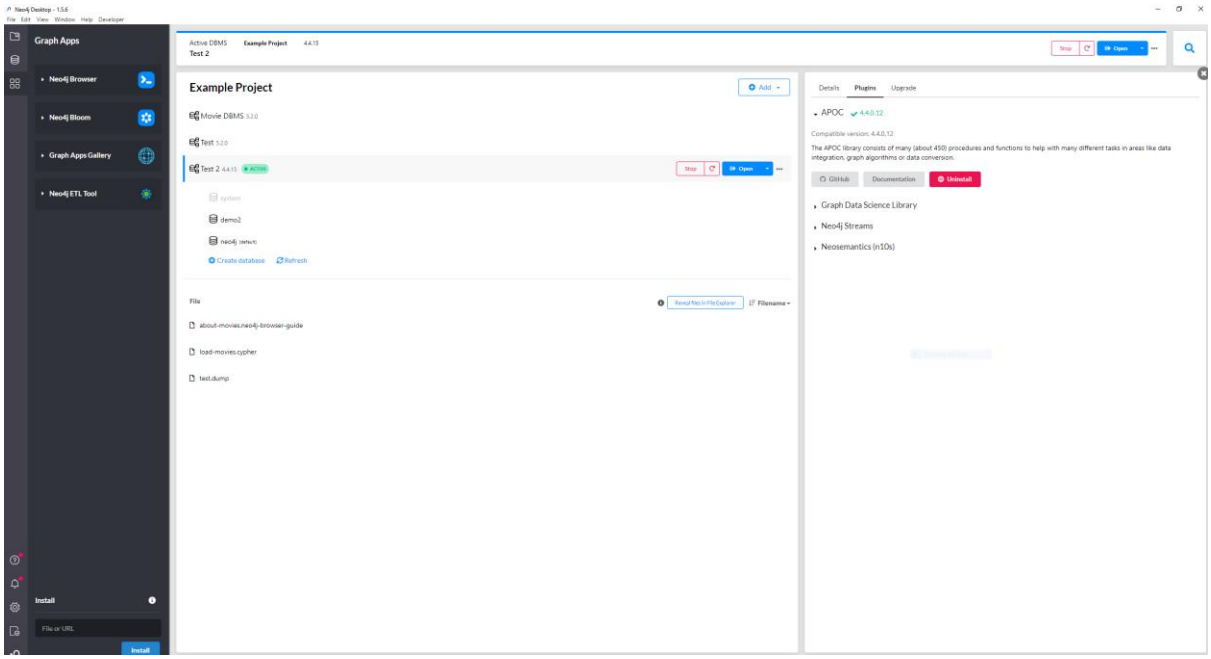
    a. Place the extracted files in a permanent home on your server, for example D:\neo4j\. The top level directory is referred to as NEO4J_HOME.
    b. To run Neo4j as a console application, use: <NEO4J_HOME>\bin\neo4j console.
    c. To install Neo4j as a service use: <NEO4J_HOME>\bin\neo4j windows-service install.
    d. For additional commands and to learn about the Windows PowerShell module included in the Zip file, see **Windows PowerShell module**.

4. Visit **http://localhost:7474** in your web browser.

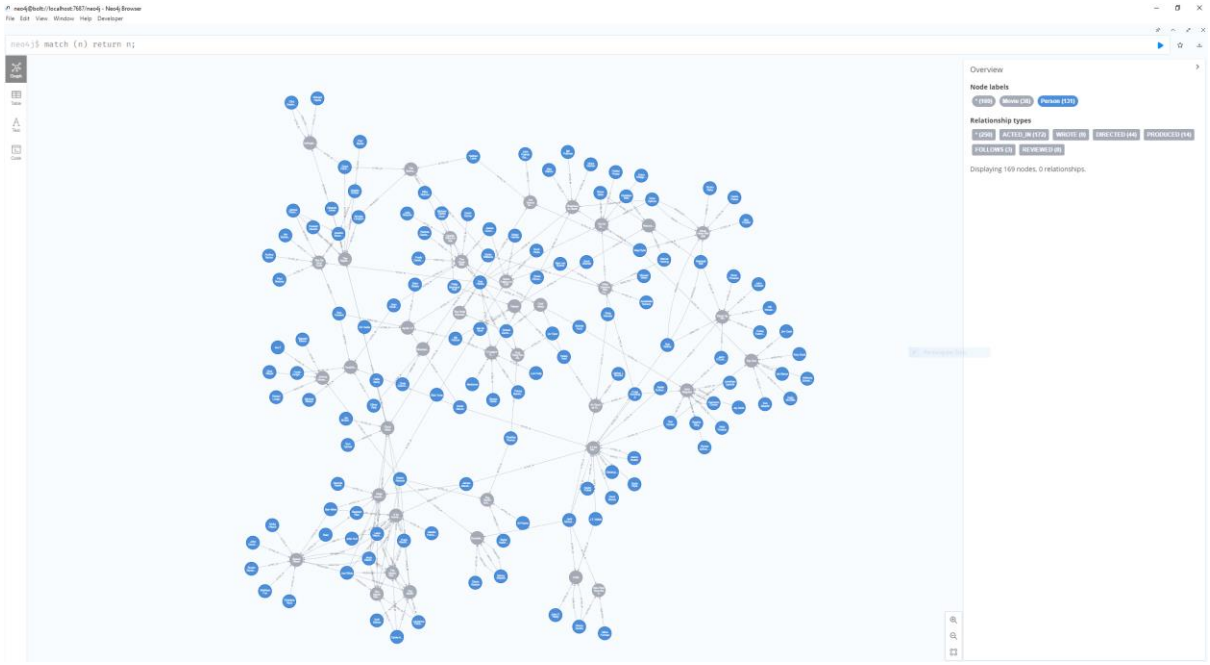
5. Connect using the username 'neo4j' with default password 'neo4j'. You'll then be prompted to change the password.

6. Stop the server by typing Ctrl-C in the console.

Neo4j Desktop:



Neo4j Browser:



Cypher Query Language:

To know the version of neo4j:

- Call dbms.components() yield name,versions,edition unwind versions return name,version,edition ;

Following is the syntax for creating a node using Cypher Query Language.

- CREATE (node_name);

Following is the syntax to create a relationship using the CREATE clause.

- CREATE (node1)-[:RelationshipType]->(node2)

Following is the syntax for the MERGE command.

- MERGE (node: label {properties . . . . . })

Following is the syntax for setting a property.

- MATCH (node:label{properties . . . . . })
  SET node.property = value
  RETURN node

Following is the query to delete all the nodes and the relationships in the database using the DELETE clause.

- MATCH (n) DETACH DELETE n

Following is the syntax to remove a property of a node using the REMOVE clause.

- MATCH (node:label{properties . . . . . })
  REMOVE node.property
  RETURN node

Following is the query which returns all the nodes in Neo4j database.

- MATCH (n) RETURN n

Following is the syntax of the WHERE clause.

- MATCH (label)
  WHERE label.country = "property"
  RETURN label

Following is the syntax of the count function.

- MATCH (n { name: 'A' })-->(x)
  RETURN n, count(*)

Comparison Installation of APOC (Awesome Procedures On Cypher) Library:

Step I:

Download required library from the sources as:

1. APOC library
2. Graph Algorithms

Make sure to library release major and minor version number matches with Neo4j version

Place downloaded files in the following location:

<Neo4j Path>\plugins

Step II:

Modify the Neo4j Configuration file, present in the location :

<Neo4j Path>\conf\neo4j.conf

1) Uncomment **dbms.directories.plugins=plugins**
   allows Neo4j to load this plugins directory to server

2) Uncomment and modify as following

   dbms.security.procedures.unrestricted=apoc.*

   dbms.security.procedures.whitelist=apoc.*,apoc.coll.*,apoc.load.*

   loads unrestricted and white-listed procedures/ plugins to the server

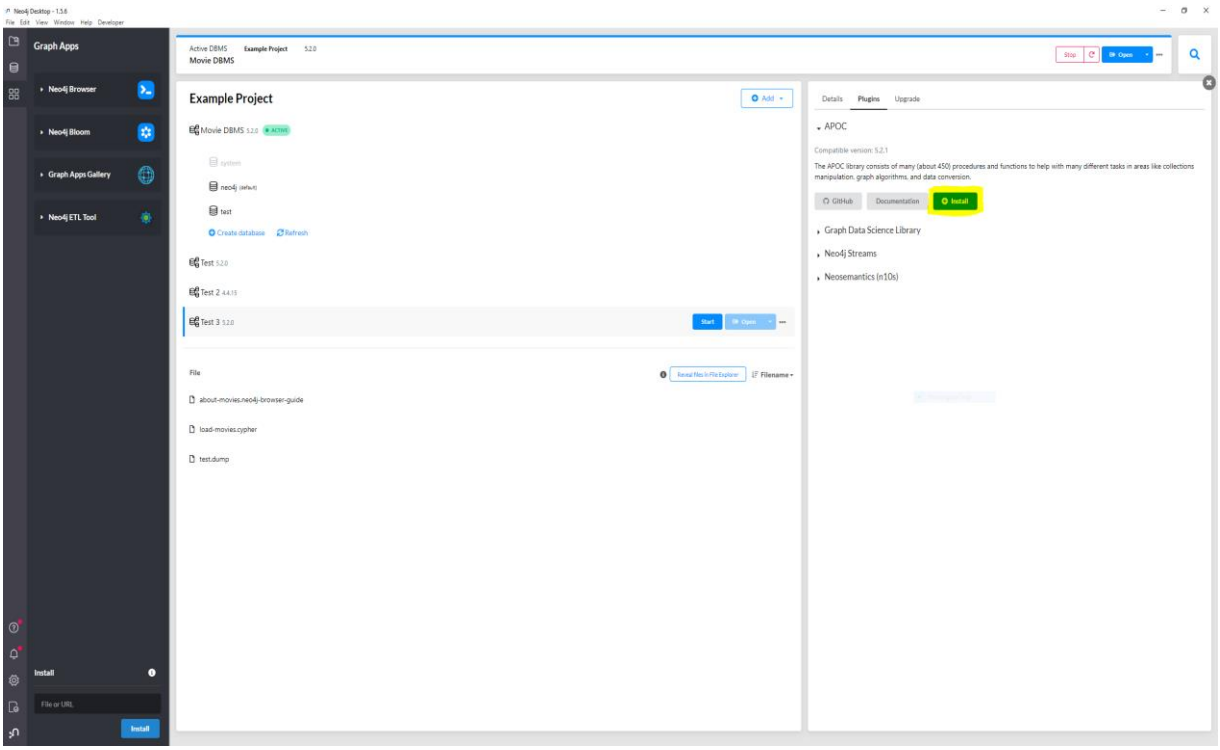
3) Add the following lines

   apoc.import.file.enabled=true

   apoc.export.file.enabled=true

   Useful for reading and writing data from and to files respectively

Direct Download APOC library in Neo4j Desktop:



We can download jar file for different category from here.

APOC Procedures:

No.	Procedure name	Description
1	"apoc.load.csv"	"apoc.load.csv('urlOrBinary',{config}) YIELD lineNo, list, map - load CSV from URL as stream of values, config contains any of: {skip:1,limit:5,header:false,sep:'TAB',ignore:['tmp'],nullValues:['na'],arraySep:';',mapping:{years:{type:'int',arraySep:'-',array:false,name:'age',ignore:false}}"
2	"apoc.load.driver"	"apoc.load.driver('org.apache.derby.jdbc.EmbeddedDriver') register JDBC driver of source database"
3	"apoc.load.jdbc"	"apoc.load.jdbc('key or url','table or statement', params, config) YIELD row - load from relational database, from a full table or a sql statement"
4	"apoc.load.directory.async.remove"	"apoc.load.directory.async.remove(name) YIELD name, status, pattern, cypher, urlDir, config, error - Remove a folder listener by name and return remaining listeners, if any"

There are more procedures are available for different use in apoc library.

Connection of Neo4j And MongoDB:

Taking advantage of the strengths of multiple database technologies is the concept of polyglot persistence. To enable polyglot persistence the application needs to store data in multiple databases, each with its own data model (graph vs. document). Being able to connect MongoDB to Neo4j and synchronize data automatically makes this process much simpler.

Prerequisites:

You should have an understanding of MongoDB, Neo4j, be familiar with both the document data model and property graph data model and have MongoDB and Neo4j installed.

To Install Mongo Connector:

- How can I install Mongo Connector without internet access?

  On a server have internet access:

  Python -m pip install –download /path/to/some/dir mongo-connector

  On offline server:

  Python -m pip install –ignore -installed—no-index—find-links /path/to/some/dir mongo-connector

Installing

You must have Python installed to use Neo4j Doc Manager.

It is recommended to install using pip the Python package manager.

pip install neo4j-doc-manager –pre

Using Neo4j Doc Manager

- Ensure that a Neo4j instance is running. If authentication is enabled for Neo4j, set the NEO4J_AUTH environment variable, containing username and password:

  export NEO4J_AUTH=user: password

- Ensure that MongoDB is running a replica set. To initiate a replica set, start MongoDB with this command:

  mongod --replSet myDevReplSet

- Then open mongo-shell and run:

  rs.initiate()

- Start the mongo-connector service with the following command:

mongo-connector -m localhost:27017 -t http://localhost:7474/db/data -d neo4j_doc_manager

Data synchronization

With the neo4j_doc_manager service running, any documents inserted into MongoDB will be converted to a property graph structure and immediately inserted into Neo4j as well. Document keys will be turned into nodes. Nested values on each key will become properties.

Connection of Neo4j And MySQL:

After installation of the Neo4j APOC kit, download the respective RDBMS JDBC driver .jar and install into **$NEO4J_HOME\plugins\.**

After installing APOC and copying the RDBMS vendor .jar to **$NEO4J_HOME\plugins\,** restart Neo4j.

The apoc.load.jdbc stored procedure is used to connect over JDBC and takes 2 arguments,namely:
Connection string
SQL statement or table

You may need to first manually load the driver by calling:

- call apoc.load.driver('com.mysql.jdbc.Driver')

The following example would connect to a MySQL database named proddb1 as user root with password=football and select all movies from the 'movies' table where the studio column was defined to be 'MGM Studios'. Using this data we would then create nodes in Neo4j for all the movies meeting this criteria and define the title property.

- CALL apoc.load.jdbc('jdbc:mysql://localhost:3306/proddb1?user=root&password=football','select title from movies where studio=\'MGM Studios\') YIELD row
- CREATE (n:Movies {name:row.title}

BIBLIOGRAPHY

"How Do I Use Cypher to Connect to a RDBMS Using JDBC - Knowledge Base." *Neo4j Graph Data Platform*, https://neo4j.com/developer/kb/how-do-i-use-cypher-to-connect-to-a-rbms-using-jdbc/.

"How Much Faster Is a Graph Database, Really?" *Neo4j Graph Data Platform*, 23 Aug. 2019, https://neo4j.com/news/how-much-faster-is-a-graph-database-really/.

"Neo4j And Mongodb - Developer Guides." *Neo4j Graph Data Platform*, https://neo4j.com/developer/mongodb/.

"System Properties Comparison Mongodb vs. MySQL vs. neo4j." *MongoDB vs. MySQL vs. Neo4j Comparison*, https://db-engines.com/en/system/MongoDB%3BMySQL%3BNeo4j.

Bhardwaj, Shikha. "Opencart Auction Module: Online Bidding Extension." *Webkul Blog*, 26 Dec. 2022, https://webkul.com/blog/opencart-auction-version2/.

CodexWorld. "Load More Data from Database Using JQuery Ajax PHP MySQL." *CodexWorld*, 22 Apr. 2018, https://www.codexworld.com/load-more-data-using-jquery-ajax-php-from-database/.

"How to Pass JavaScript Variables with Ajax Calls." *Tutorials Point*, https://www.tutorialspoint.com/How-to-pass-JavaScript-Variables-with-AJAX-calls.

"Language Reference." *Php*, https://www.php.net/manual/en/langref.php.

Mark Otto, Jacob Thornton. "Bootstrap." *Bootstrap · The Most Popular HTML, CSS, and JS Library in the World.*, https://getbootstrap.com/.

"An Open-Source JavaScript Library for Interactive Maps." *Leaflet*, https://leafletjs.com/.

PHP Functions, https://www.w3schools.com/php/php_functions.asp.

Prajapati, Ankit. "Load More Data Using JQuery, Ajax, and Php." *Coding Birds Online*, 17 Aug. 2022, https://codingbirdsonline.com/load-more-data-using-jquery-ajax-and-php/.