Logo, company name

Description automatically generated

**Grocery Database: Healthy Outlet**

A PROJECT REPORT SUBMITTED TO

**THE NATIONAL INSTITUTE OF ENGINEERING, MYSURU**

(An Autonomous Institute under VTU, Belagavi)

In partial fulfilment of the requirements for Project work (Database Laboratory CS5L02),   
fifth semester

**Bachelor of Engineering**

**in**

**Computer Science and Engineering**

*Submitted by*

**Adeep Krishna Keelar (4NI19CS007)**

**Ananya Mathur (4NI19CS022)**

**Athmeeya M (4NI19CS027)**

Under the Guidance of

Ushadevi A

Assistant professor

Nivedha S

Assistant professor

Amrutha Sree V M

Assistant Professor

Logo

Description automatically generated

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**2021-2022

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**THE NATIONAL INSTITUTE OF ENGINEERING**



***CERTIFICATE***

This is to certify that the project work entitled **Movie Database** is a work carried out by **Adeep Krishna Keelar (4NI19CS007), Ananya Mathur (4NI19CS022) and Athmeeya M (4NI19CS027)** in partial fulfilment for the project work (Database Laboratory – CS5L02), fifth semester, Computer Science & Engineering, The National Institute of Engineering **(**Autonomous Institution under Visvesvaraya Technological University, Belagavi) during the academic year 2021-2022. It is certified that all corrections and suggestions indicated for the Internal Assessment have been incorporated in the report deposited in the department library. The project work report has been approved in partial fulfilment as per academic regulations of The National Institute of Engineering, Mysuru.

**Signature of the Internal Guides** **Signature of the HoD**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Nivedha S Ushadevi A

Assistant professor Assistant professor

Amrutha Sree V M

Assistant Professor

Dr. V K Annapurna  
Professor and Head  
Dept. of CS&E  
NIE, Mysuru

**Signature of the Examiners with Dates**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name:  
Designation:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name:  
Designation:

**Acknowledgements**

Firstly, we would like to thank the Department of Computer Science and Engineering for providing us an opportunity to implement this project as a part of the laboratory component. We thank our faculty assigned to our batch, Ms. Nivedha S, Mrs. Ushadevi A and Mrs. Amrutha Sree VM for their help and support throughout the entire semester and the project phase. Their patience and guidance have motivated us to complete the project. We would like to thank the non-teaching staff of the department of CS&E for their support throughout our endeavour.

Secondly, we acknowledge Dr. Phaneendra HD the course instructor for DBMS CS5C03 and we acknowledge the help from Mr. Narender M, the DBMS Lab Co-ordinator.

We acknowledge the support from Mr. Yogesh M.J., Dr Abhinandan SP and Mr. KG Rajaram for their support, guidance and motivation throughout the coursework.

We acknowledge the insight, support and help from our friends, Abhilash Hegde, Kinar S, P Pradyoth, Abdul Shakur, Vinay P, Hari Charan, namely a few.

Finally, I thank my team mates, Ananya Mathur and Athmeeya M for having the faith to trust in me and helping me throughout the crucial parts of the development of this project.

Thanks and Cheers!

* Adeep Krishna Keelar (4NI19CS007)
* Ananya Mathur (4NI19CS022)
* Athmeeya M (4NI19CS027)

**Table Of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Chapter Name** | **Page No.** |
| 1 | Introduction | 6 |
| 2 | System Analysis | 8 |
| 2.1 | Existing System | 8 |
| 2.2 | Proposed System | 8 |
| 2.3 | System Requirements | 9 |
| 3 | System Design | 11 |
| 3.1 | System Architecture | 11 |
| 3.2 | Entity-Relationship Diagram | 12 |
| 4 | System Implementation | 13 |
| 4.1 | Starting the Application | 13 |
| 4.2 | Creating Tables to Store Data | 14 |
| 4.3 | Concepts used in Database | 14 |
| 4.4 | Tools Used for Implementation | 15 |
| 5 | System Testing | 16 |
| 6 | Results | 17 |
| 7 | Conclusion and Future Enhancements | 21 |
| 8 | References | 22 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Description** | **Page No.** |
| 3.1 | 3 Tier Architecture | 11 |
| 3.2 | ER- Diagram | 12 |
| 4.1, 4.2 | XAMPP Control Panel, htdocs file | 13 |
| 4.3 | Various Tables for Grocery Database | 14 |
| 4.4 | VS Code Editor with all files | 15 |
| 4.5 | Git showing change in code | 16 |
| 6.1 | Main Display Page of the Site | 17 |
| 6.2 | Login/registration page | 17 |
| 6.3 | Registration Form | 17 |
| 6.4 | When customer adds an item to the cart | 18 |
| 6.5 | Displaying details of product | 18 |
| 6.6 | Checkout Page | 18 |
| 6.7 | Payment Page | 19 |
| 6.8 | Updating account for customer | 19 |
| 6.9 | Admin Login | 19 |
| 6.10 | Admin Main Page with Features | 20 |
| 6.11 | View All Products | 20 |
| 6.12 | View All Payments being displayed | 20 |

# Chapter 1

**Introduction**

Groceries play an essential part in every person’s life. It may be basic vegetables and fruits, essential food items, domestic products, everybody needs to have these in the house for their daily purposes. Usually these are purchased at the local shop by the people who go to the shop and purchase these there or they go to the local supermarket. However, for those who cannot leave the house due to physical ailments and for third party elements who are linking shops and people online, this can be a hurdle for simple grocery shops that have been faithful servers for the customers who have been purchasing from them for years before.

Hence, our project focuses on the Grocery Database for a fictious local shop “Healthy Outlet”. The project provides a front-end site for the customers and guest users to make their purchases from other various fictious elements who supply for the shop. The front-end allows customers to save their details, this helps regular customers to easily order and for new customers, they can register their details. The site dynamically links the categories of the products given in the shop and various fictious brands along with their products for the customers to choose from.

While modern successful third-party companies such as Food Basket, Amazon Groceries and more have made an impact in the market, our application aids a shop directly, thus the shop makes it to the internet which in turn will be seen all around the world. While receiving better feedback and more success, they can expand their business thus helping the local shop to expand their reaches and when someone makes it to the internet, they are also educated digitally for the modern world. A modern solution for the busy people who are very busy also helps those with work. This is also help promote employment for people.

Feedback systems play an important role in taking decisions, the fictional application also has a feedback system that stores the feedback from the customers or guests who have gone through the site. All kinds of feedback, from criticism regarding the working and rendering of the site to the glitches that prevail to hinder the smooth functioning of the site to complaints regarding the lack or too many choices to ridiculous pricing of products to suggestions on how to improve the application with better and enhanced features to queries to even good feedback, feedback plays an important role, it helps one to analyse themselves and to improve upon their faults.

A Database Management System refers to the technology used for storing and retrieving user’s data with utmost efficiency while considering appropriate security features. It consists of programs that manipulate the database.

Database Systems are meant to increase the ease of maintaining large amounts of information. Management refers to defining structures that store these large volumes of data and providing the user with tools to manipulate according to their needs, while providing a sense of security that the data will not be leaked or be fallen to the wrong hands.

This project attempts to recreate an e-commerce site for Groceries which is “supposedly” owned by a certain Healthy Outlet supermarket, which provides the opportunity to their customers to make their purchases on the internet. This project isn’t hosted on the internet and runs on a local server; however, it can be easily hosted and it would be able to function properly to the needs of the users.

The project Grocery Database uses various technologies that work smoothly to provide the experience required for the user to do their requirements easily. The front end applications were built using HTML5 (Hypertext Markup Language 5), CSS3 (Cascading Style Sheets 3), some amount of Vanilla JavaScript, the middleware code along with some of the supporting front end was done using PHP7.0 and the backend was provided by MySQL or MariaDB, the feature database of the XAMPP Application. The programs or the code while being saved on the local system was also made open-source by being pushed to the GitHub through the version-controlling application Git. The code editor used was Visual Studio Code.

# Chapter 2

**System Analysis**

People currently go out to make purchases at the local shop. Due to problems such as illnesses, very busy schedules, emergency situations, lockdown and more, they may not be able to do so and one cannot survive without essentials in the modern house. While many successful applications such as Food Basket and Amazon Groceries and many other delivery services have made a trend in the market, they usually have their own storage units or decided centres where they make their purchases or buy goods in bulk. The stored products when ordered are carefully packed and then delivered to the destination. This system makes a very large profit as the initial amount sent for the mass bulk purchases are usually less and some expenses set up for the packing and labour, customers end up paying a lot in the false thought that they are saving by the discounts in the prices, without realising they have been swindled. This isn’t a crime; this is a marketing strategy executed right on the lines of business.

However, with more and more users shifting towards these applications, they feel it isn’t necessary to visit their local shops that have been providing services. Many shop keepers have been running shops for decades and have served honestly, yet they suffer from the competition. Not all markets, shops or even local supermarkets suffer with lack of customers and even shut down due to losses. These centres may not be recognised beyond a certain level and hence lose a lot of customers.

**Proposed System**

The current systems are indeed very well built by the top-class engineers with very good planning and software engineering techniques. There are no flaws in the existing system that our proposed system offers to size up, the proposed system is similar to the current e-commerce sites. It offers most of all the options present in the sites, from viewing various products and brands and searching for some products. In order to purchase from the site, one needs to log in, if not having an account, they would have to make an account. Upon logging in, they can make their purchases and check out and opt for either an offline payment or payment via PayPal (not implemented, only typo) or enter the details where the payment details are given. Similarly, they can give feedback to the site which is received.

The front-end part of the project was built using HTML5, CSS3 and some amount of Vanilla JavaScript. Multiple pages such as the login page, checkout page, my account page and more are connected by hyperlinks within the code using the anchor tags. These pages have various functions for the user to interact with the web site.

The backend or the MySQL/MariaDB is a part of the XAMPP Application which provides local hosting for applications within the system. The database holds the data entered by the user in various elements and many parts of the site where information is displaced are dynamically done so by connecting the front-end to the back-end by the middleware scripting language PHP. These scripts are embedded into the HTML documents that are served by the web server, the interpreter then coverts the scripts into commands that the computer executes. PHP has the primary advantage as not only as a server-client scripting language but also has the ability to extract data from databases easily, hence is very popular in web development but suffers lack of popularity due to hard semantics unlike the JavaScript flavours such as Node.js, etc,.

The system also has an administrator, who has the power to insert, delete and view various aspects of the data stored. Some of the features or powers of the admin are –

* Admin has the power to insert products of his choice to the database.
* Admin has the power to view, edit or delete the products entered. Editing involves changing the details of the product from the name to the price and more.
* Admin has the power to insert, view, edit or delete various categories and brands of the products listed.
* Admin has the power to check out the Customer details and delete Customer details.
* Admin can view the Orders made and receive Payment details.

**System Requirements**

1. **Software Requirements:**

**Operating System:** Any Operating System with the ability to support latest applications or applications on versions that are at least six or seven years back would be able to run this project smoothly. Listing a few -

* Windows 8 and above: Windows 8, Windows 8.1, Windows 10, Windows 11.
* Mac OS 10 and above
* 64-bit Ubuntu 14.04+, Debian 8+, openSUSE 13.3+, or Fedora Linux 24+, Manjaro Linux, Raspbian, etc,.

1. **Hardware Requirements:**

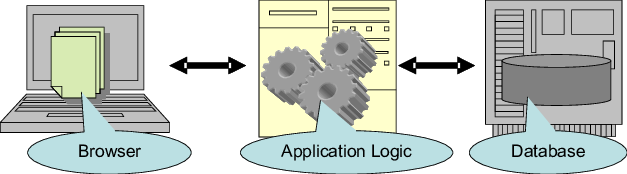
* Intel Pentium 6000, i3 (preferably 10th generation), i5 (7th generation and above), AMD Ryzen 3/5/7 or equivalents.
* Preferably 4GB RAM or above for smooth functioning.
* 1920 x 1080p monitor for better viewing experience.

1. **Software Applications Used:**

* Visual Studio Code
* XAMPP Server
* Git for Version Control (Pushing code to GitHub repo)
* An internet browser – Microsoft Edge 10, Chrome, Opera Mini, Tor.
* Preferably internet connection.

# Chapter 3

**System Design**

**3.1 System Architecture**

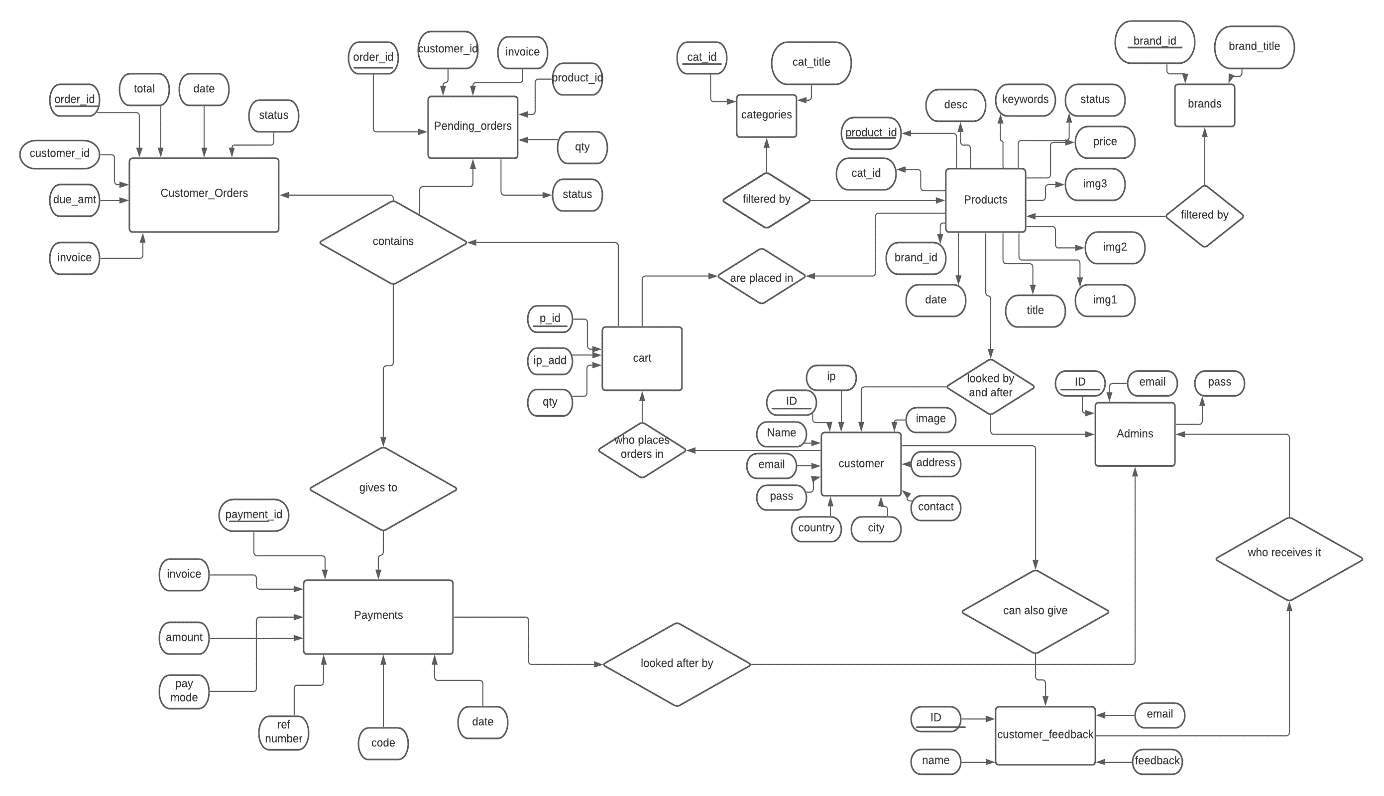
**Fig 3.1 3-Tier Architecture**

This system uses the Three-Tier Database Architecture as depicted in Fig 3.1, where in the application logic lives in the middle-tier, hosted by Apache Server Service of XAMPP. The user interface is available to the end user when they access our website using any client-side web browser application which server the website up to them. The database too, is hosted on XAMPP and is separated from the website and is not directly accessible by the user. The queries given by the user are first processed by the logic present in our application layer before being sent to the database system. This architecture was implemented since it runs faster, is more secure since the end-user cannot directly communicate with the database server and has better performance.

The Data Tier consists of the MYSQL database is hosted on the Apache service. The Logic Tier consists of the application logic written using PHP for the backend which resides on the Apache Server. The Client or User Tier consists of the front-end of the website which the user is presented with which is fetched by the Client-side web browser applications and is implemented using HTML5, CSS3 and JS.

\

* 1. **Entity Relationship Diagram**



**Fig 3.2 ER Diagram**

This is the ER diagram that serves as a blueprint for the project. It describes the structure of our database and includes the two main components required which are the entity set and the relationship set.

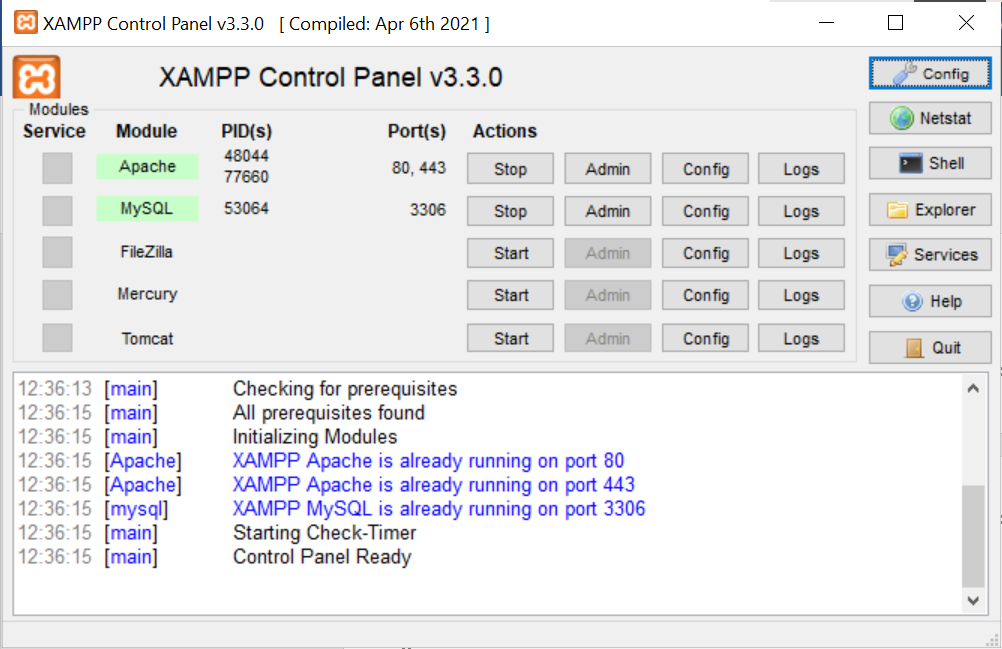
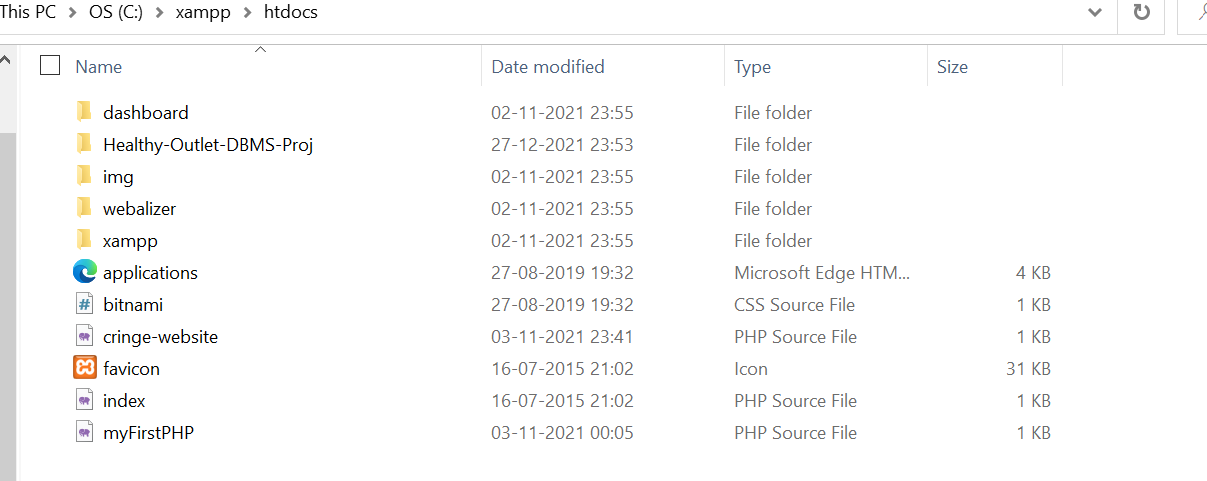
Basically, the entity Product has several attributes which serves as the primary and foreign keys to several other entities. The entity Product has a relation with Brand and Categories. Products are placed in another entity cart which takes in a few attributes, which has a relationship with customer\_orders and pending\_orders. This in turn arises with Payments which can be done only if a customer has logged into the system. All details and extra parts of the system are maintained and looked upon by the Admin.

**Chapter 4**

# System Implementation

**4.1 Starting the Application**

In order to first start the system, press the XAMPP Control Panel in the Search Tab and start Apache and MySQL modules, this allows the user to run the browser on a server on the local system, thus having scripting support. Create the folder in the htdocs of the XAMPP file in the memory drive of the system, otherwise the files will not be located by the server.

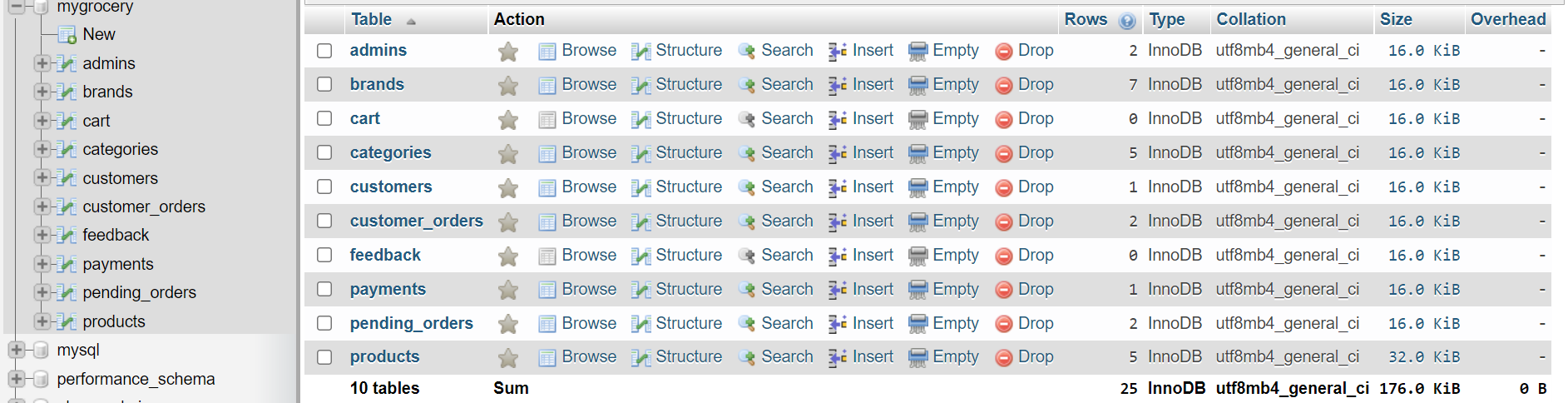
****  

**Fig 4.1, 4.2 XAMPP Control Panel, htdocs file**

In the browser, enter the localhost path and the phpadmin path or click on the admin button in the XAMPP control panel to easily access the project and the databases.

**4.2 Creating the Tables to store the data**

The project uses over 10 tables for the storing of various kinds of user entered data or admin entered data which isn’t visible to the User.



**Fig 4.3 Various Tables of Grocery Database**

1. admins – This table stores the values of the admins logging in with Administration power. They can insert, delete or view the data inserted into this database.
2. brands – This table stores the product’s brand details, with attributes of ID and name.
3. cart – This table stores the product id, the ip address of the user in case he is valid and quantity of the products he is wanting before checkout.
4. categories – This table stores the product’s category details, with attributes of ID and name.
5. customers – This table stores the details of the customers when they register for the first time. This table data is also used for the login and session\_starting functions and for the customer\_details, payments and pending\_details tables.
6. feedback – This table is used to store the feedback of the customer or guest.
7. Product – This table is used to store the details of the products inserted or modified by the admin or so.

**4.3 Concepts used in Database**

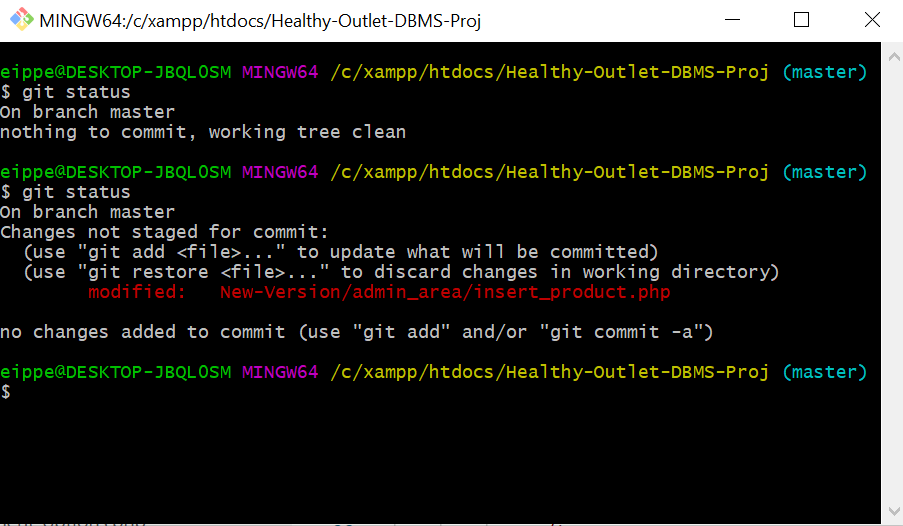
1. **Triggers:** Triggers are used to change the status of the order, when paid, the pending section changes to Complete. Triggers are also used to destroy the sessions when the customer logs out or deletes his account.
2. **Views:** Basic views of the tables are used in viewing functions of the admin section.
3. **Indexes:** Almost every table has an index, a primary key that auto increments and helps to sort and get data faster.
4. **Constraints:** Constraints such as PRIMARY keys, foreign keys are used to implement the database in 3NF format. Various other formats such as NOT NULL constraint, conditions are also used.

**4.4 Tools used for Implementation**

* **PHP 7.0:** It is the scripting language that is used in the front-end code that helps to connect it with the database. While several other popular options such as Node.js, Python3, Perl were considered, it was decided to use PHP as a traditional approach towards the development of the website.
* **HTML5:** Hypertext Markup Language 5 is the front-end language used to build the websites. It provides various tags that the browser sets up accordingly to fill the inserted data.
* **CSS3:** Cascading Style Sheets 3 is the extra feature of HTML5 that helps to providing styling abilities to the HTML pages.
* **JavaScript:** JS helps to bring some dynamic activity into the static page. Not much of JS was used, merely restricted to calling alert() and window.open() functions. An API called tinycloud was used to take in the description of the products in the insert\_products page.
* **Git:** Git is the version control application that helps to push the code of projects into GitHub.

****

**Fig 4.4 A screenshot of the VS Code editor with all the files**

****

**Fig 4.5 A screenshot of Git showing a change in the code**

**Chapter 5**

# System Testing

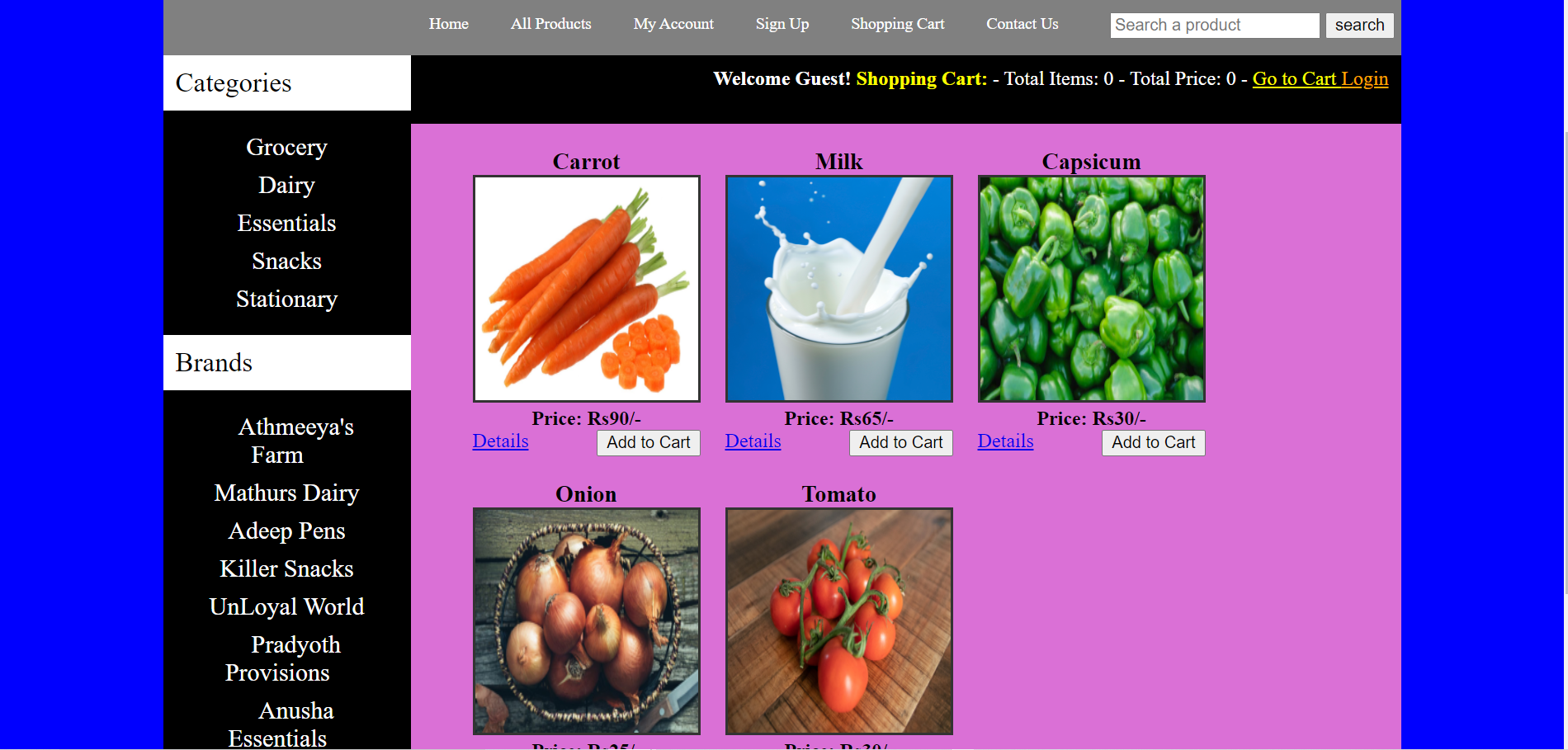
Grocery Database: Healthy Outlet consists of a lot of individual modules which come together to make the system work to make the system functional and efficient. Hence tests of various scenarios were required to test the functionality of each module individually as well as ensure the integration of said modules. Regression Tests were also carried out at various points to make sure that new features being implemented did not hamper the working and performance of existing ones.

Some of the tests that were carried out were –

* Testing if the PHP code was able to dynamically echo the various categories and brands of the database, further be able to display such when either one were removed accordingly.
* Testing views of the tables in the admin section according to the code.
* Testing the functioning of the IP function.
* Testing if the session was successfully started upon the login of a customer and the triggered destroying of the session when the user logged out or if the customer finished checking out of the cart, thus enabling him to make a new order.

**Chapter 6**

## Results

****

**Fig 6.1 Main display page of the site.**



**Fig 6.2 Login or Registration Page**

****

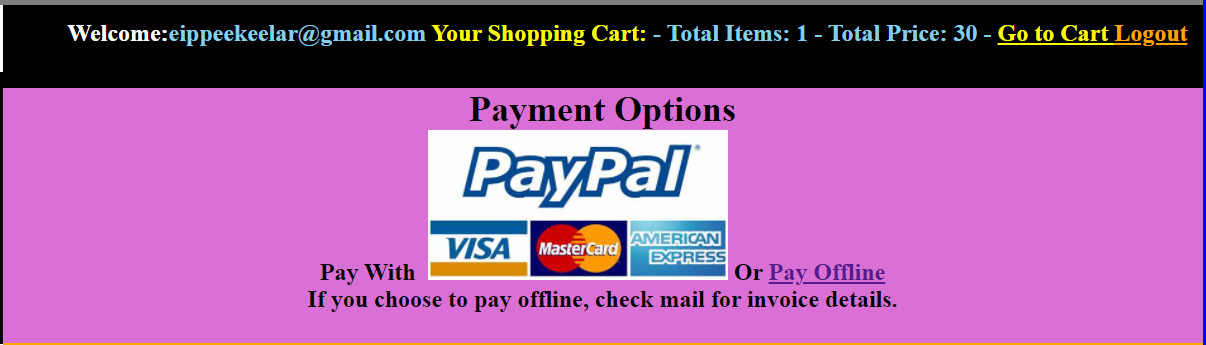
**Fig 6.3 Registration Form**

****

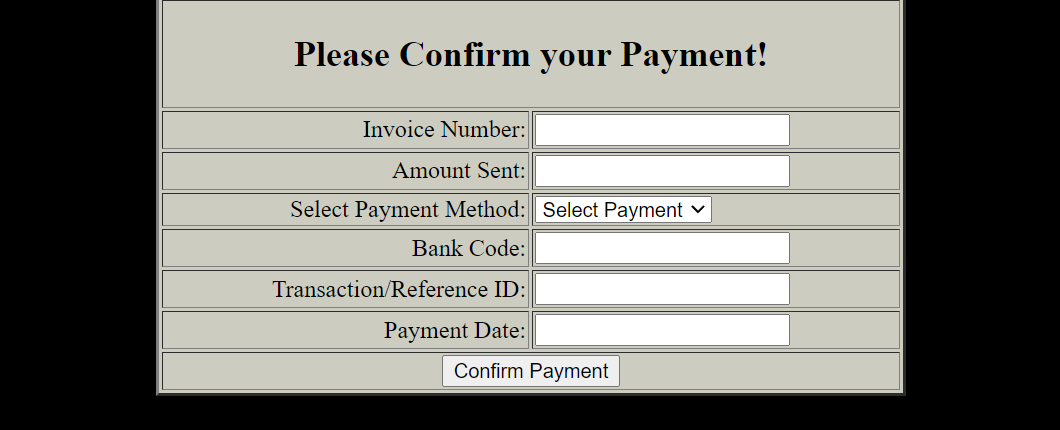
**Fig 6.4 When Customer Adds an item to the cart**

****

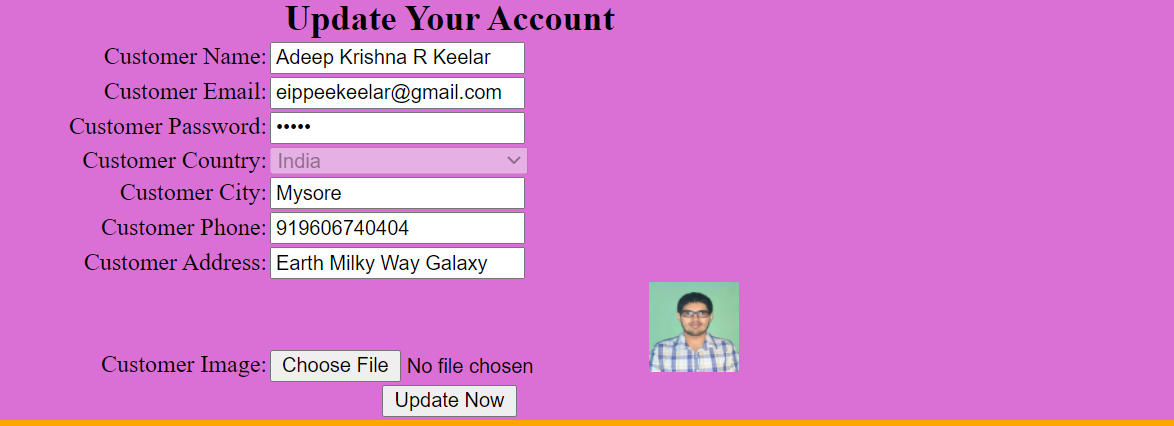
**Fig 6.5 Displaying some details or description of a product**

****

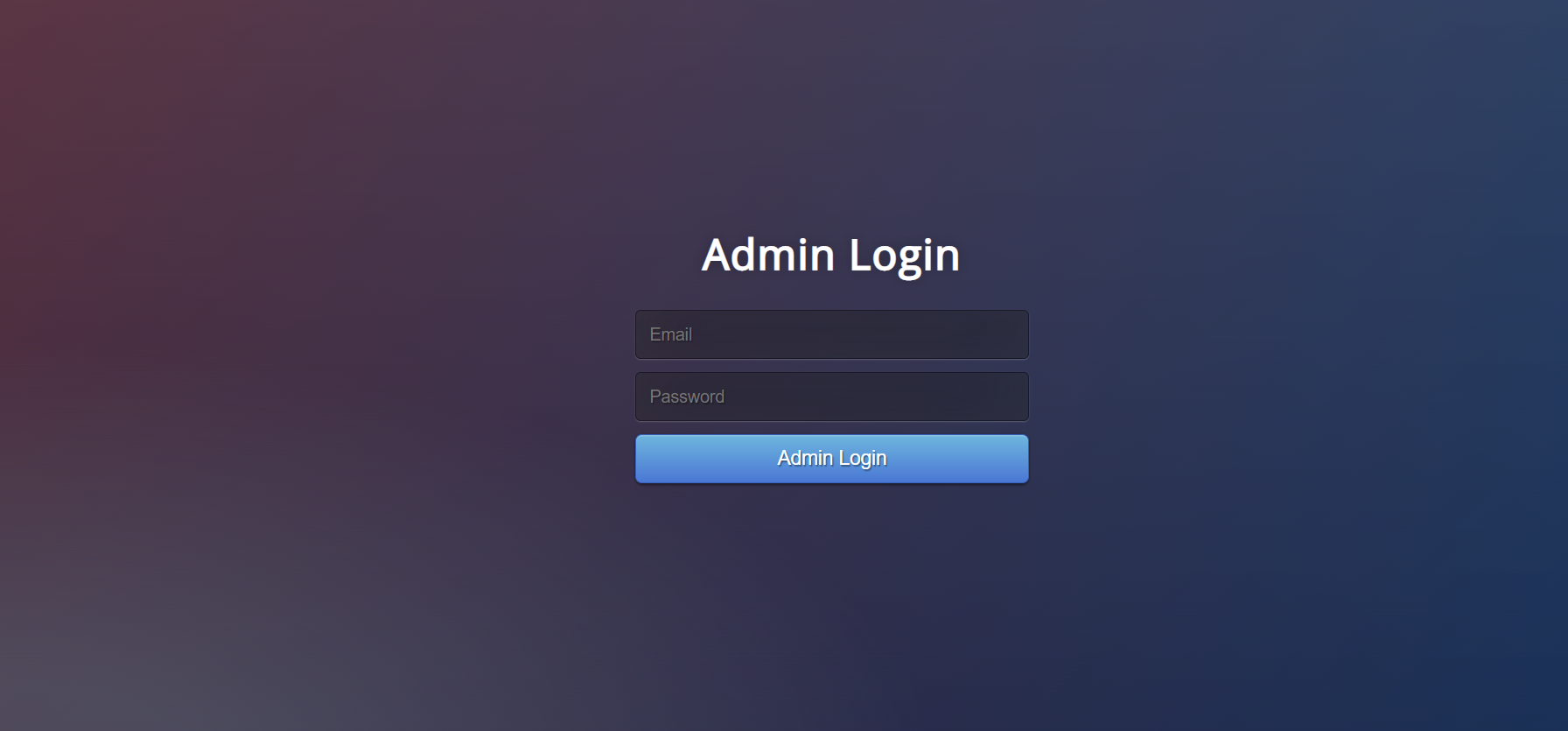
**Fig 6.6 Checkout Page**

****

**Fig 6.7 Payment Page**

****

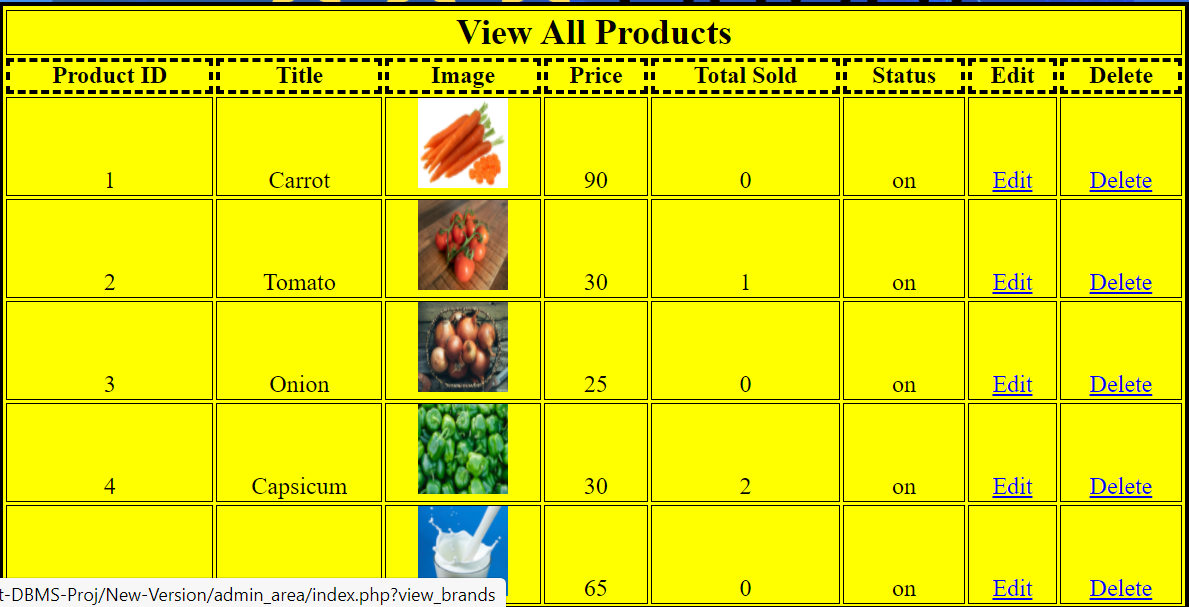
**Fig 6.8 Updating Account for Customer**

****

**Fig 6.9 Admin Login Page**

****

**Fig 6.10 Admin Main Page with features**

****

**Fig 6.11 Viewing All Products**

****

**Fig 6.12 All Payments being displayed**

# Conclusion and Future Enhancements

“Grocery Database: Healthy Outlet” was the implementation of an e-commerce website for a grocery-based supermarket Healthy-Outlet, done as a part of the project work for CS5L02 DBMS Laboratory work in a team of three. It was satisfying to see that the project turned out more successful than initial thought of and it was a good feeling to know that the efforts put was worth to see that we had managed to replicate at least a small implementation of a real e-commerce site. It makes us appreciate how the engineers in the industry put in a lot of planning and effort to make a site that is even more dynamic, powerful and intelligent.

Thinking of the future enhancements that the project could see –

1. The project could be hosted on the internet with some enhancements so that the modules of the PHP that was supposed to send a mail to the user could function perfectly. Further hosting on the internet would also allow this site to really function as a true web application for a shop. It was considered to use free hosting options such as FileZilla FTP and Outing.com but then dropped due to the changes required and the complications that needed more time for us to implement.
2. The project could do with more features, the Search tab could become dynamic in nature rather than using a Brute-Force approach to search through the keywords.
3. More features such as the Object being able to zoom in or give a 360-degree view, more detailed explanations.
4. The site could become a whole lot more dynamic by using strong web-oriented flavours of JS such as React.js, Angular.js, Vue.js, Bootstrap, etc,.
5. Some unsolved errors and glitches in the database that prevent the smooth functioning of this application, some undetectable errors in the PHP module, some glitches with the code not able to read the database entries, some JS elements not working properly.

## References

1. The Database Book: Principles and Practice using MYSQL, by Narain Gehani
2. Fundamentals of Database Systems, by Elmasri and Navathe
3. StackOverflow <https://stackoverflow.com>
4. MYSQL forums <https://forums.mysql.com>
5. W3Schools <https://www.w3schools.com>
6. PHP Documentation [PHP: Documentation](https://www.php.net/docs.php)
7. Online Ustaad Tutorials [Ecommerce Website in PHP - Urdu/Hindi - YouTube](https://www.youtube.com/playlist?list=PLU4yvac0MJbJeyd6BgKYxOGJAALu-eeN7)
8. Source code <https://github.com/AdeepKrishnaKeelar/Healthy-Outlet-DBMS-Project>