LETTER OF TRANSMITTAL

Dear IMU,

I am pleased to present the attached report on the building POS system that was recently implemented. This report provides a detailed analysis of the procedures for building the system.

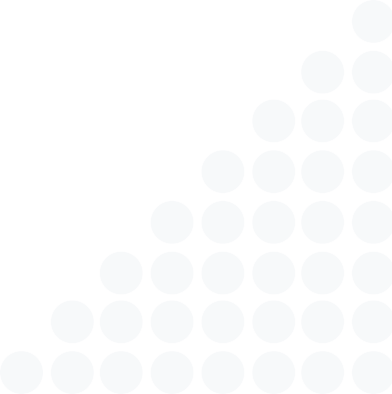
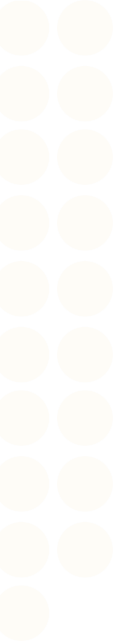
As you will see, the building POS system has proven to streamline many of our daily processes and improve efficiency. However, we have also identified a number of areas where the system could be improved, and we have included recommendations for addressing these issues in the report.

We hope that this report will be useful to you as you consider the future of the building POS system in your organization. If you have any questions or would like to discuss the report further, please do not hesitate to contact me.

Sincerely,

Mohammad Mashayekh

9 Jan 2023



POINT OF SALES SYSTEM

technical report

**Table of**

|  |  |
| --- | --- |
| **Contents**  TABLE OF CONTENTS | 3 |
| List of Figures | 4 |
| GLOSSARY | 5 |
| ABSTRACT | 6 |
| INTRODUCTION | 6 |
| Methodology | 8 |
| Results | 9 |
| Discussion | 20 |
| Conclusions | 21 |
| Results | 21 |
| Recommendations  References | 22 |
| Appendices | 24 |
|  |  |

**List of**

**Figures**

|  |  |  |
| --- | --- | --- |
| Figure | Description | Page |
| 1 | All tables in database | 9 |
| 2 | Table of items | 9 |
| 3 | transactions table | 10 |
| 4 | transactions table | 10 |
| 5 | users\_transactions table | 11 |
| 6 | login page | 12 |
| 7 | dashboard page | 13 |
| 8 | all items page | 13 |
| 9 | item single page | 14 |
| 10 | edit item page | 14 |
| 11 | edit item page | 15 |
| 12 | all transactions page | 15 |
| 13 | transaction single page | 16 |
| 14 | edit transaction page | 16 |
| 15 | selling dashboard page | 17 |
| 16 | all users page | 17 |
| 517 | user single page | 18 |
| 18 | edit user page | 18 |
| 19 | create user page | 19 |
| 20 | model code/1 | 24 |
| 21 | model code/2 | 25 |
| 22 | view model | 26 |
| 23 | ajax get method | 27 |
| 24 | ajax put method | 28 |
| 25 | Ajax delete method | 29 |
| 26 | items index page | 29 |

**GLOSSARY**

|  |  |
| --- | --- |
| **POS** | Point Of Sales |

**ABSTRACT**

In this report I described how I used PHP and the MVC design to create a POS system that can process transactions and keep track of inventory and mange items. We used in front end HTML, CSS, Bootstrap, JavaScript and jQuery and in backend we used PHP, MySQL and XAMPP server.

Then we created database structure contains table for items and table for users and another table for transactions, and then we developed MVC application model. Then we created POS System can handle transactions and track inventory and mange items.

And this system will improve efficiency of sales and leading to increase profits also it is strong competitive advantage for the store.

# INTRODUCTION

Point of Sale (POS) is a sales-oriented activity that supports the transaction process, which includes customer administration, inventory management, reporting, and purchasing. The processing of returns and standards. This point of sale is where customers can pay retailers. [1](Marisa & Yuarita, 2017). Systems and technology are developing very quickly, which is essential for society's advancement in many spheres. For people, including entrepreneurs, technology can offer many conveniences in their daily lives. A point of sales application's benefits includes enhancing customer service through POS. A business can quickly complete an accurate, quick, and organized transaction process.

This promotes the consumer-centric orientation of commercial services and spurs demand. Making decisions and exercising control are two more advantages. Most of the time, the controlling process can be completed rapidly because all reports are made promptly available, enabling group and individual decision-making. [2](Sani, Pradana, & Rusdianto, 2018)

### Problem

Inside King Abdullah Business Park, HTU opened a store. We were hired by the HTU IMU to create a Point of Sale (POS) web application to manage inventory and sales transactions. Numerous personnel in a variety of roles will work on the POS.

### Objectives

1. Create a Point of Sale (POS) website that addresses the challenges that HTU faces in managing the store at King Abdullah Business Park.
2. handle customer transactions, as well as tracking sales and inventory.
3. Providing the administrator with real-time information about the store's operations.



### Methodology

1. Tools and software used:

Our system is a web based application and the main technologies and tools that are associated with this project are:

* For Front-end we will use: HTML, CSS, Bootstrap, JavaScript and jQuery.
* For Back-end we will use: PHP, MySQL, Xampp Server.

1. Steps of project production:
2. Plan the POS system's functionality by deciding what features and functionalities you want it to have. This could involve creating reports, processing transactions, and maintaining inventory, among other things.
3. Create the development environment: Install a database management system like MySQL and a local server like Apache or Nginx on your computer.
4. To create the database: Make a database schema with tables for holding data on goods, clients, transactions, and any other pertinent information.
5. Create the models: The data and business logic of an MVC application are represented by the models. You might develop models for goods, clients, and transactions for a POS system.
6. Create the views: In an MVC application, the views serve as the user interfaces through which users can communicate with the system. You could design views for the transaction history, the product catalog, and the main POS screen in a POS system.
7. Implement the controllers: In an MVC application, the controllers manage user input and interactions and act as a liaison between the models and views. You may design controllers for a POS system to handle transactions, manage products, and produce reports.



# Result

We started by creating a database to link it to the site that we want to build, and w the tables were as shown in figure (1):

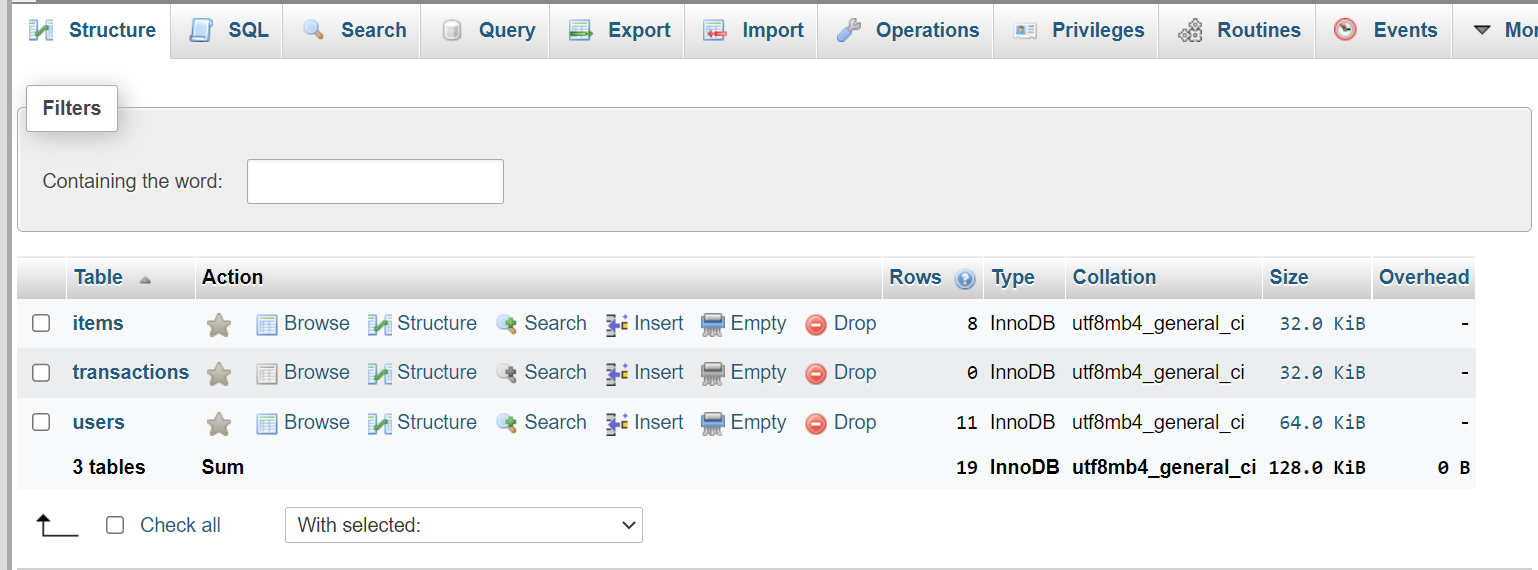


Figure (1): All tables in database

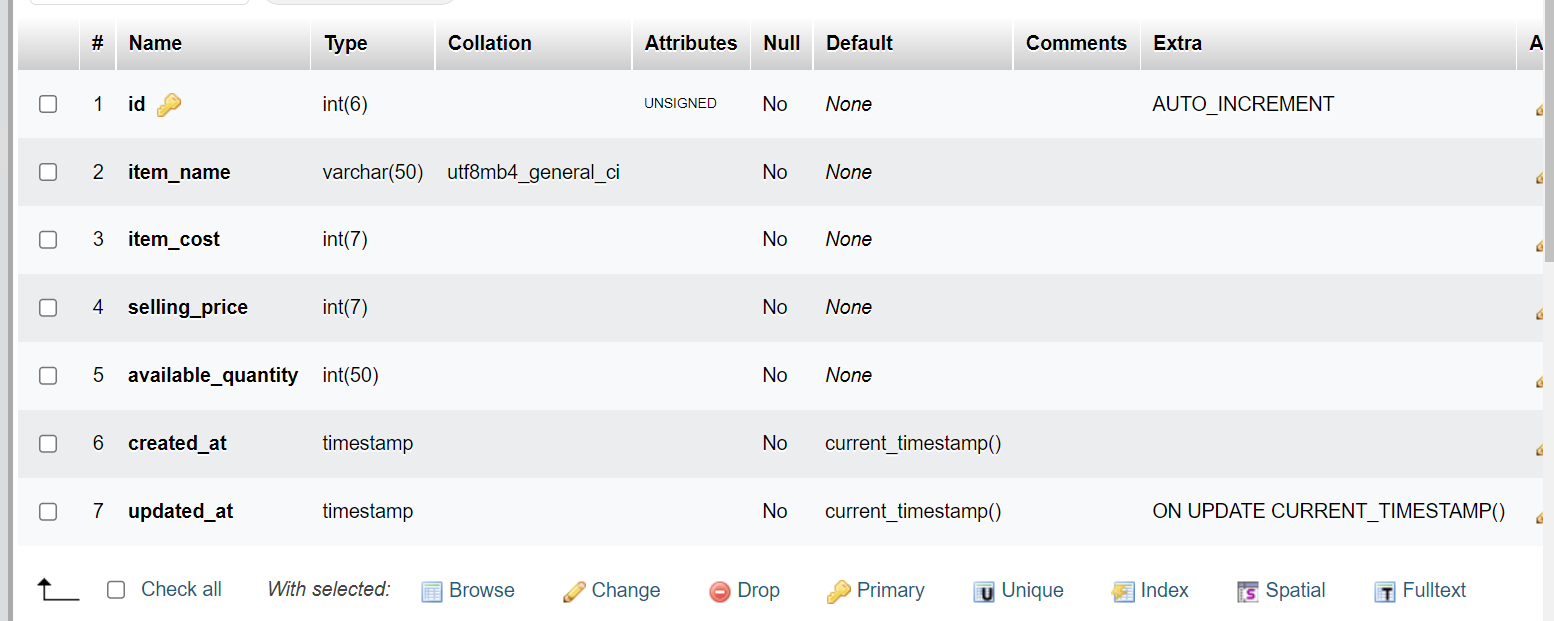


Figure (2): Table of items

The table in which a new item is added, changed, and used on the website is shown in figure (2).

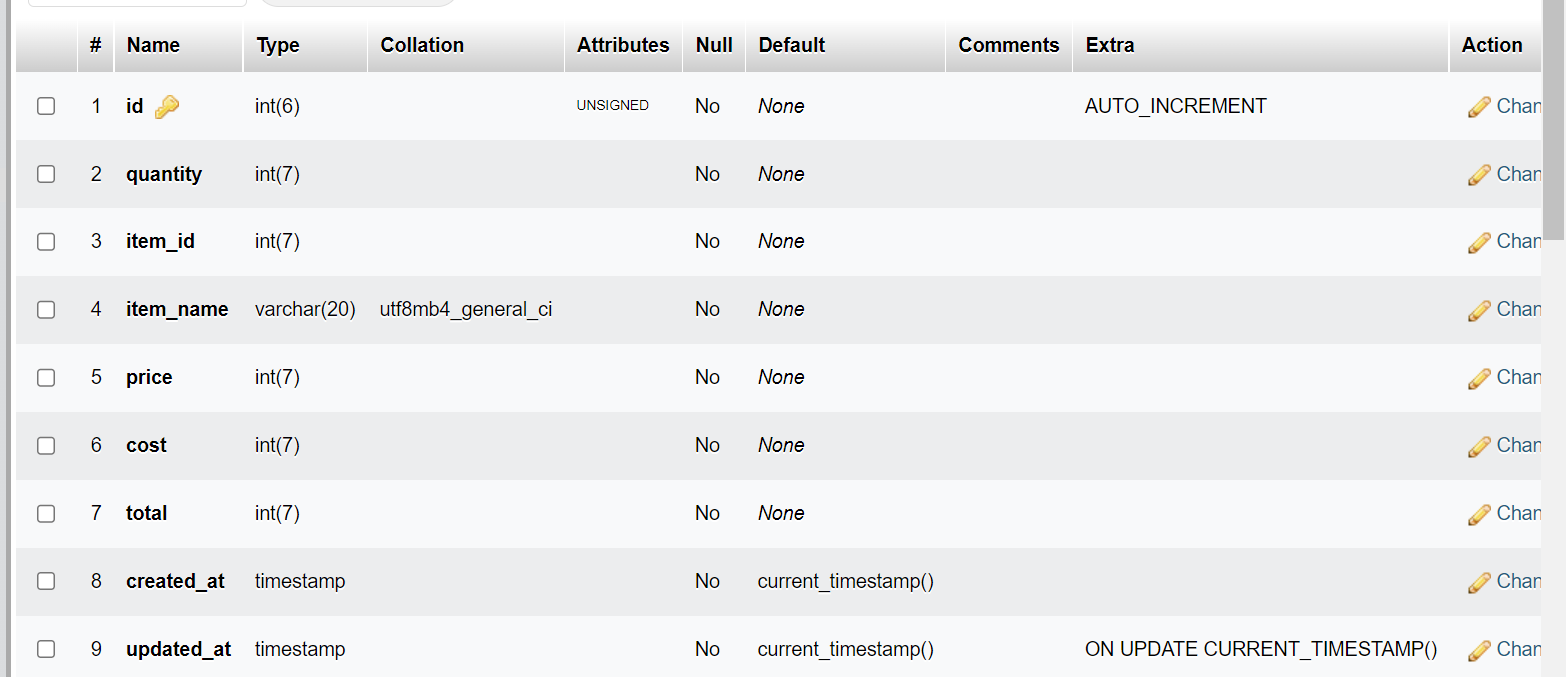


Figure (3): transactions table

This table is used on the site to manage transactions in terms of adding a new transaction, modifying or deleting it is shown in figure (3).

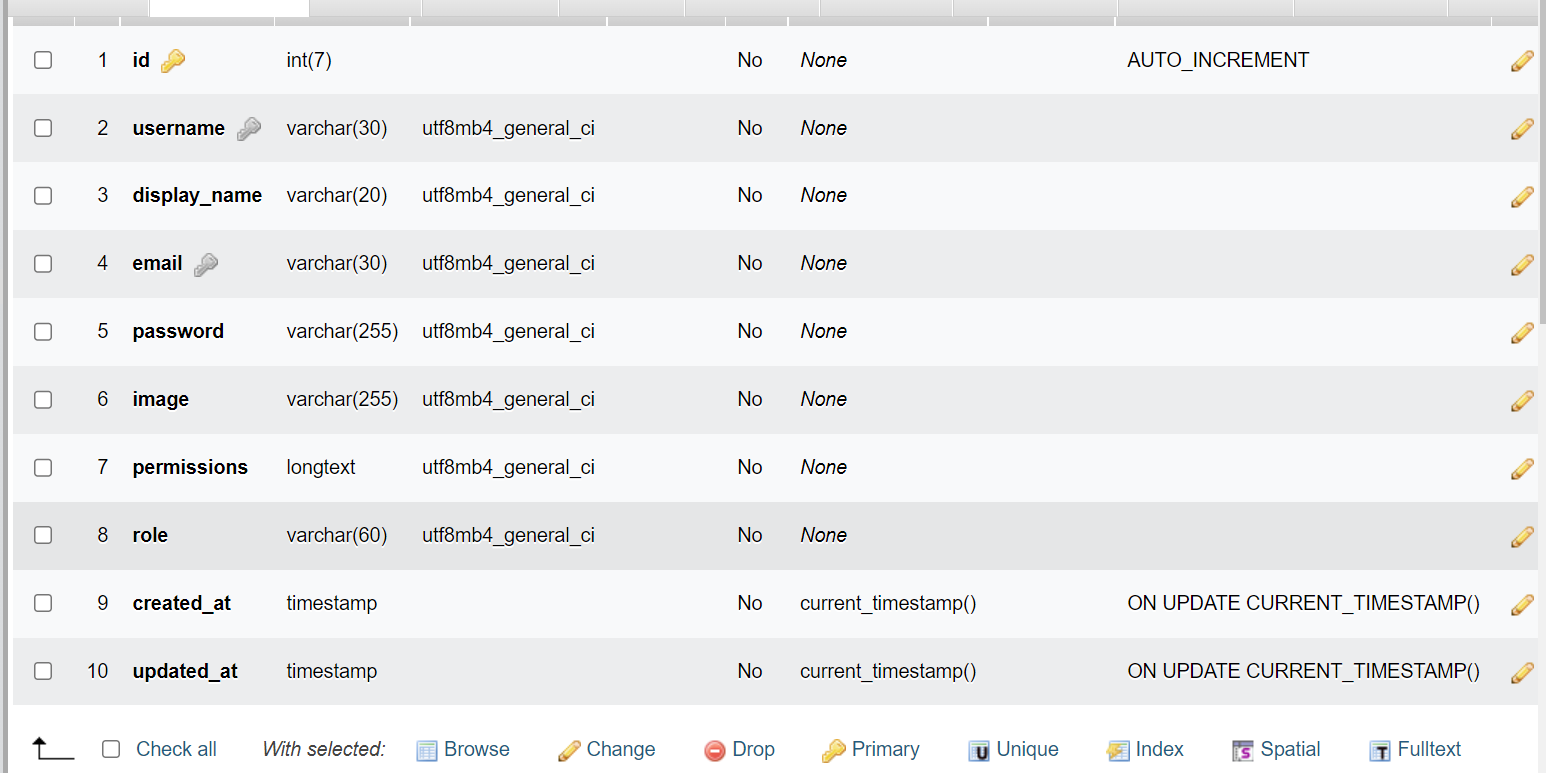


Figure (4): users table

In this table, users are managed by adding a new user, modifying or deleting it, and verifying whether the user is registered on the site or not in order to grant him the appropriate powers based on user role which shown in figure (4).

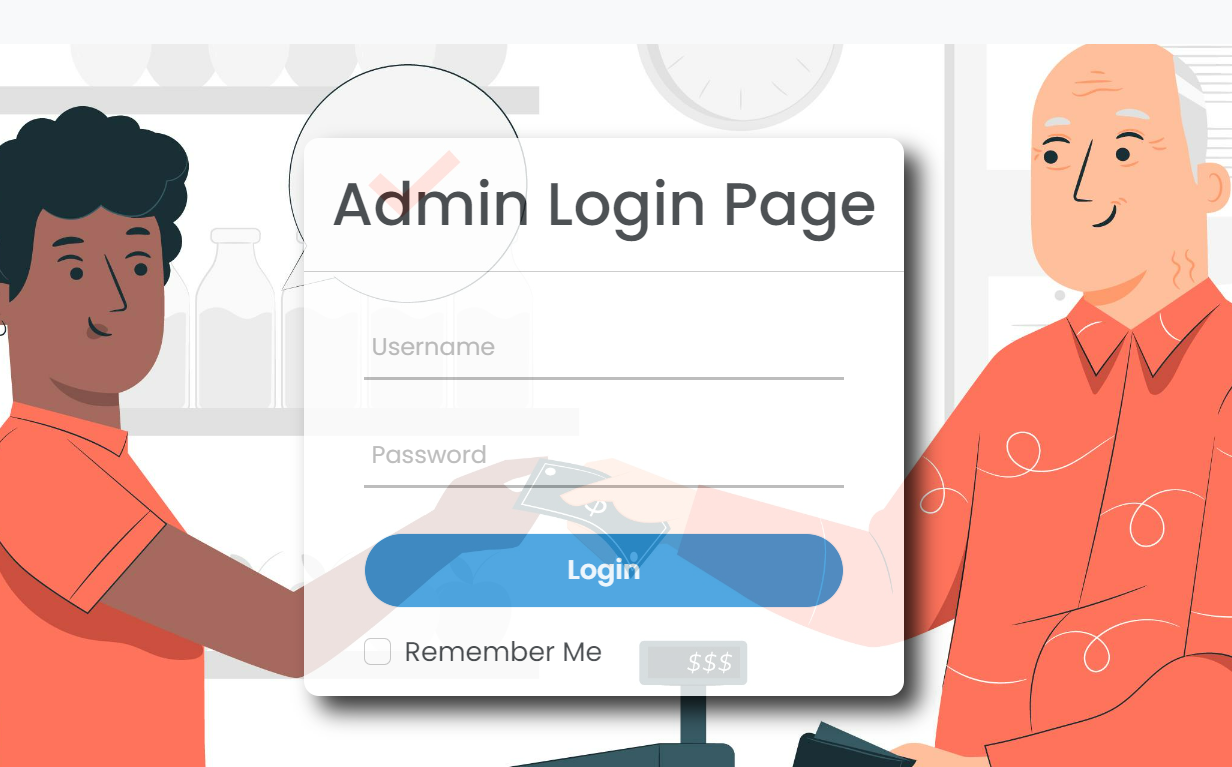


Figure (6): login page

figure (6) shows the login page, which is the first page that appears to the user when opening the site, through which the username and password that allows him to enter are verified if the information entered coincides with the database.

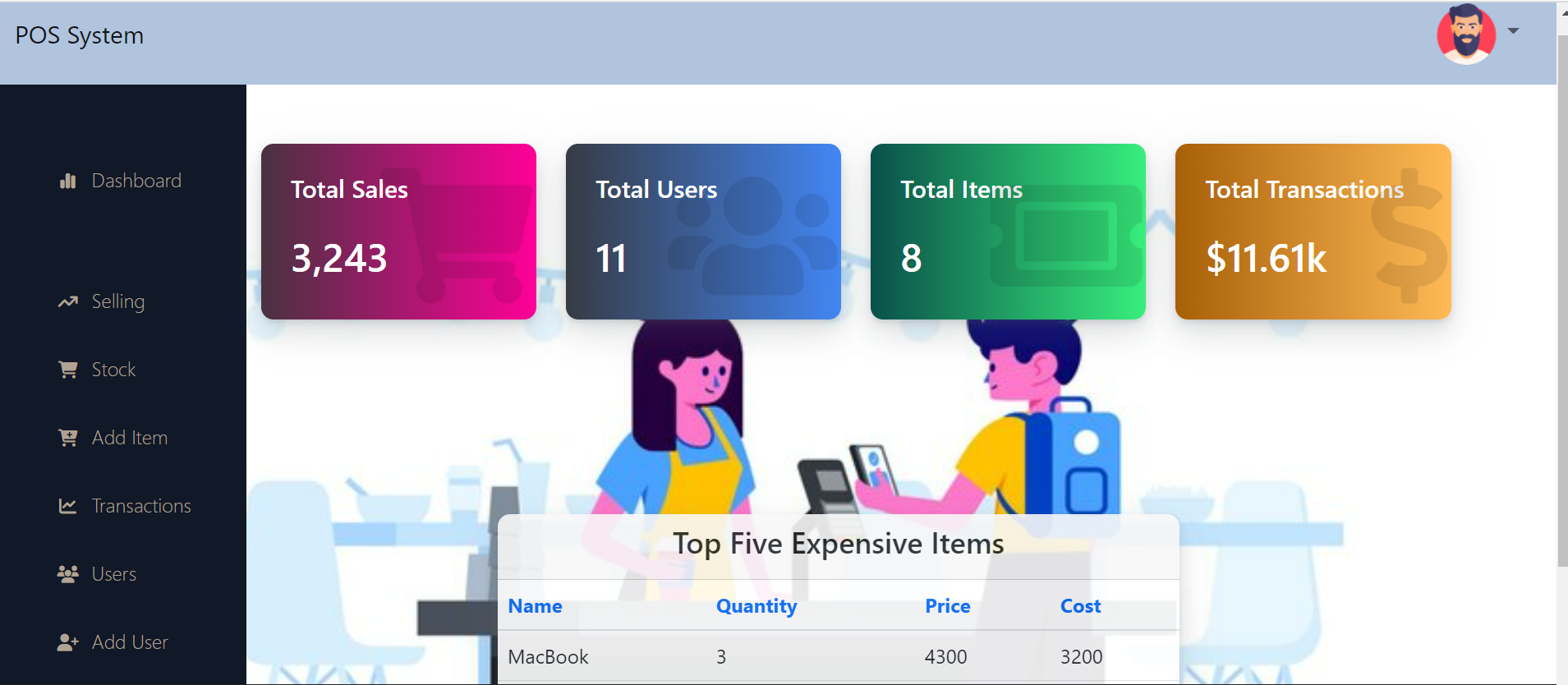


Figure (7): dashboard page

Dashboard page, which is a straightforward dashboard and the first page the admin sees after verifying his information, contains the following data: Total transactions, total products sold, and total sales and table of top five most expensive goods to purchase, total users, revenues, and total purchases, and this page no one can reach it just admin Figure (7).

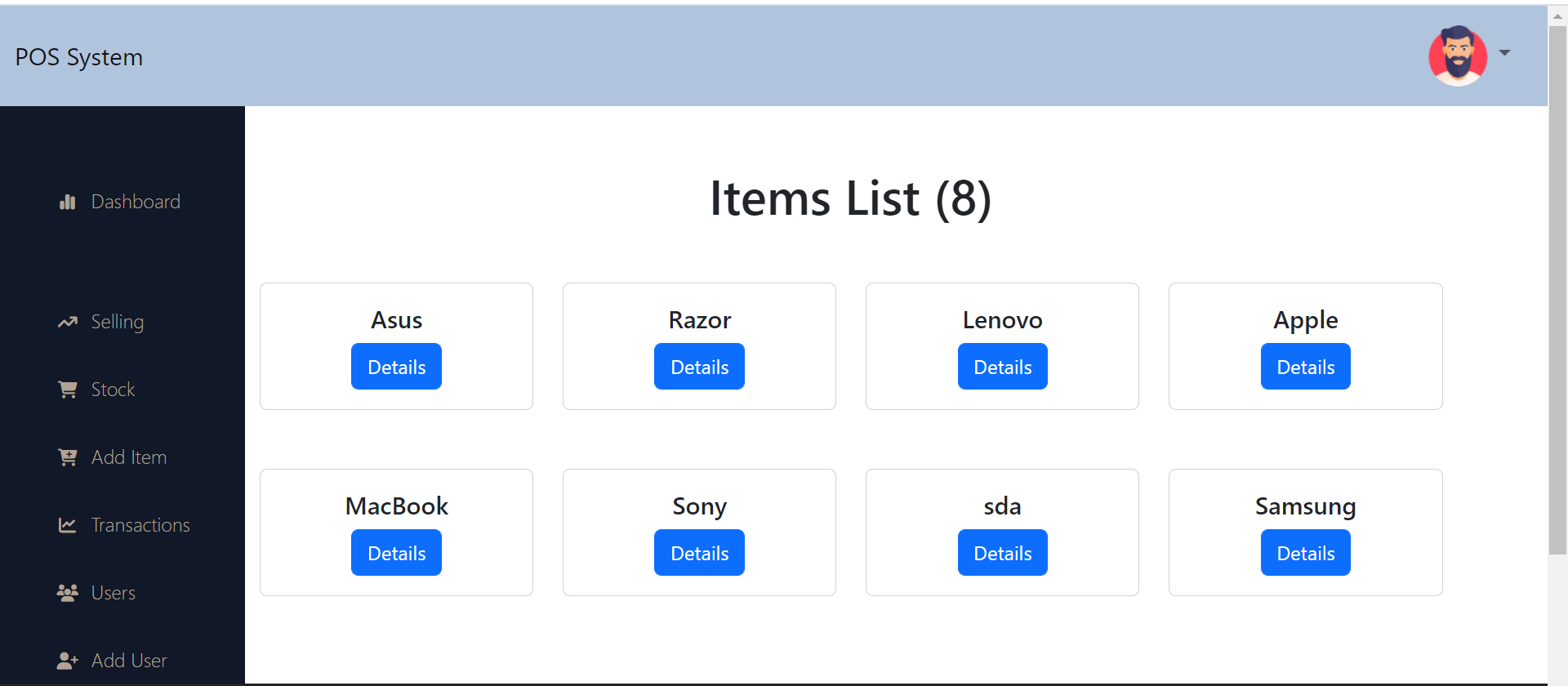


Figure (8): all items page

A stock management page is depicted in Figure (8), which lists every item and offers all CRUD functionalities (through subpages) for each item. Only administration and procurement could access it.

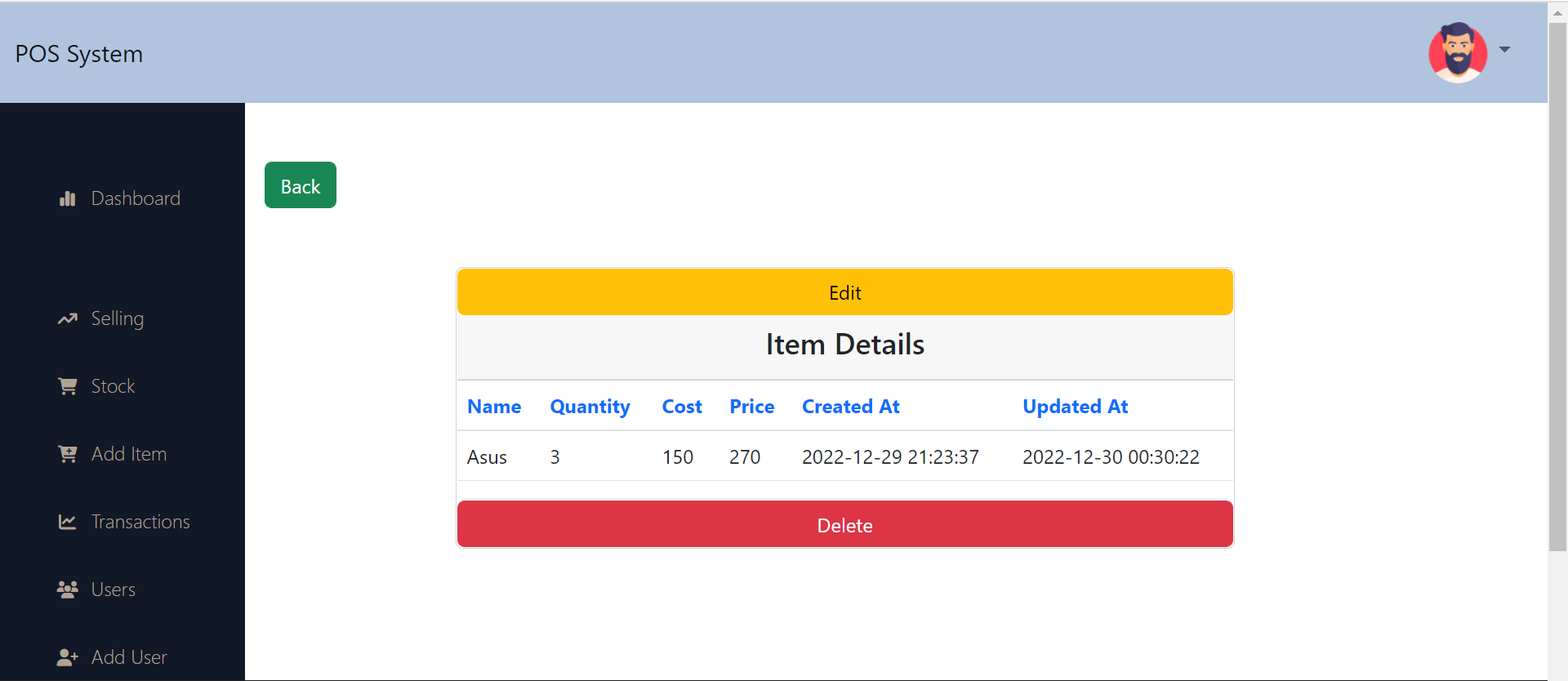


Figure (9): item single page

In figure (9) shows item single page, all product information appears and contains two buttons, one for modification and the other for deletion.

Graphical user interface, application

Description automatically generated

Figure (10): edit item page

Figure (10) depicts the edit item page, which allows admin or procurement to make changes to products.

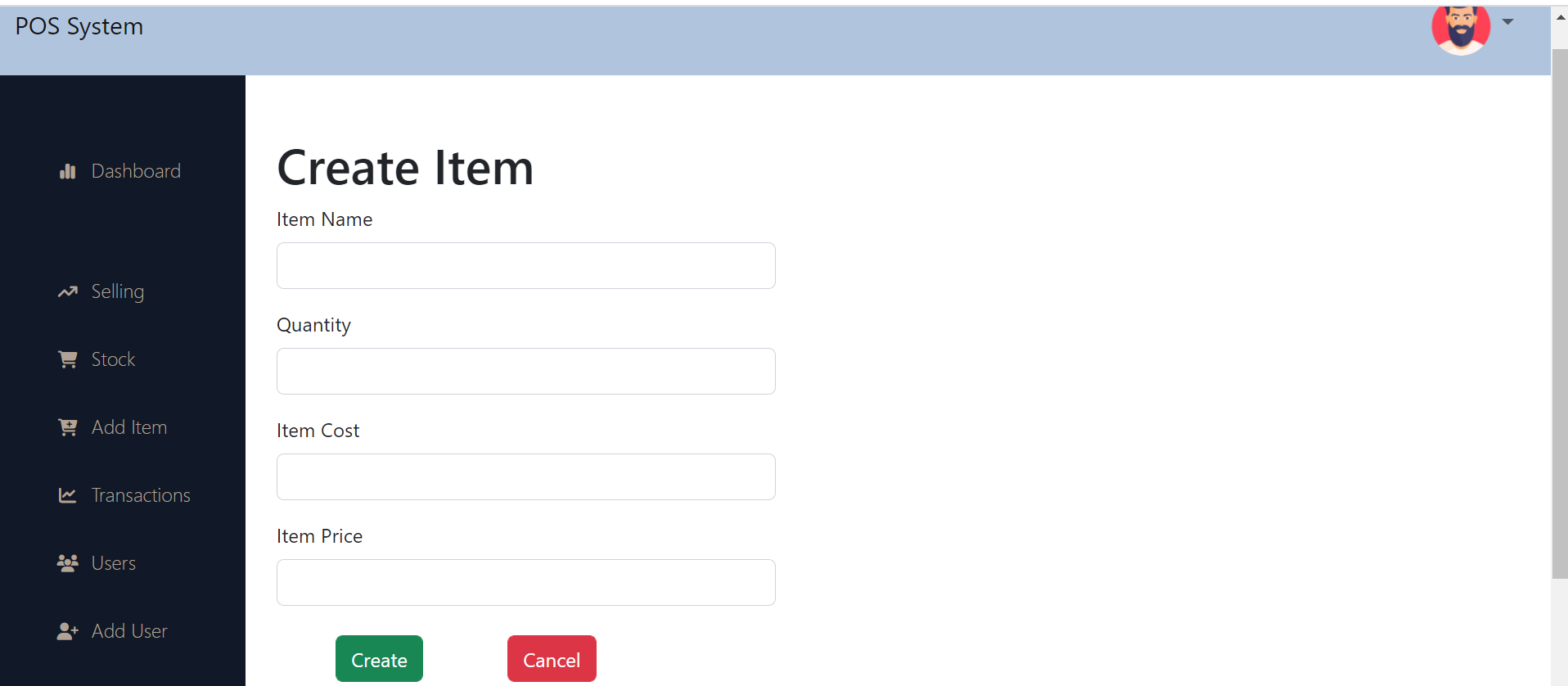


Figure (11): create item page

Figure (11) depicts the create item page, which allows admin or procurement to create items.

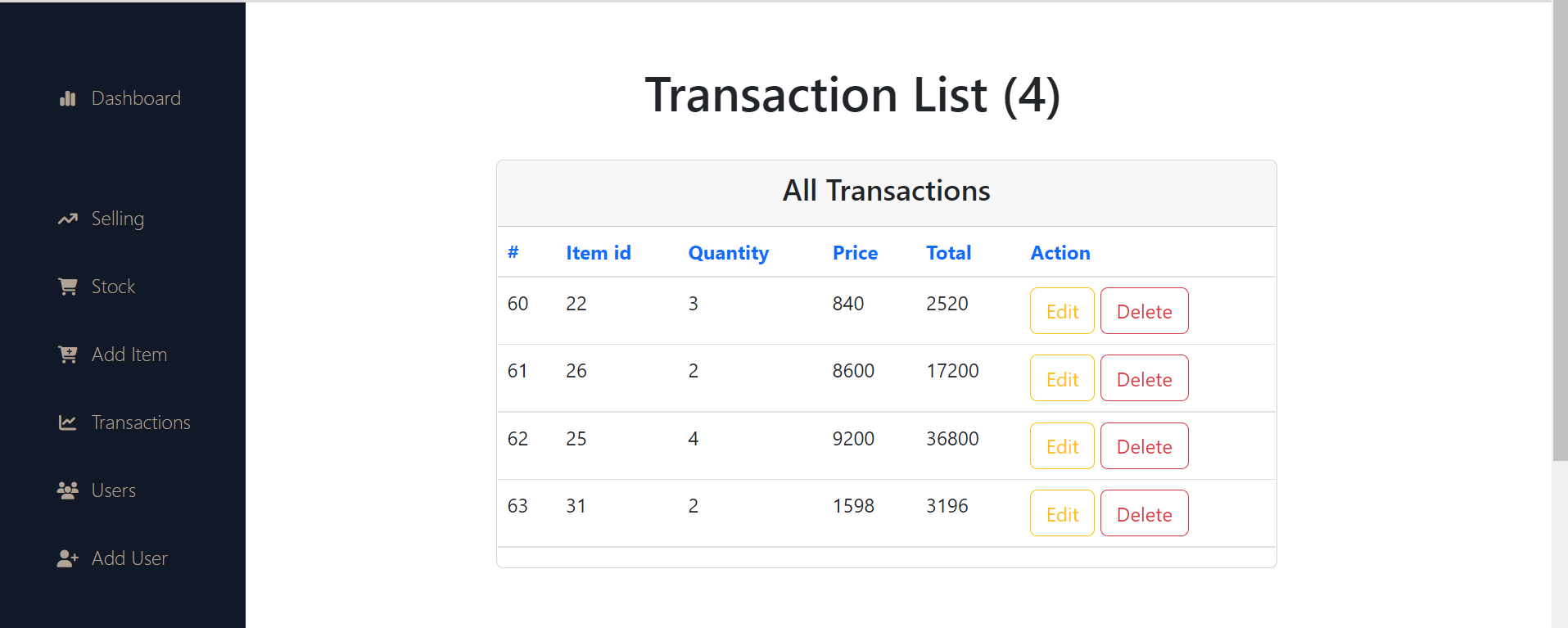


Figure (12): all transactions page

Figure (12) depicts the all transactions page, which allows admin or accountant to edit or delete transaction.

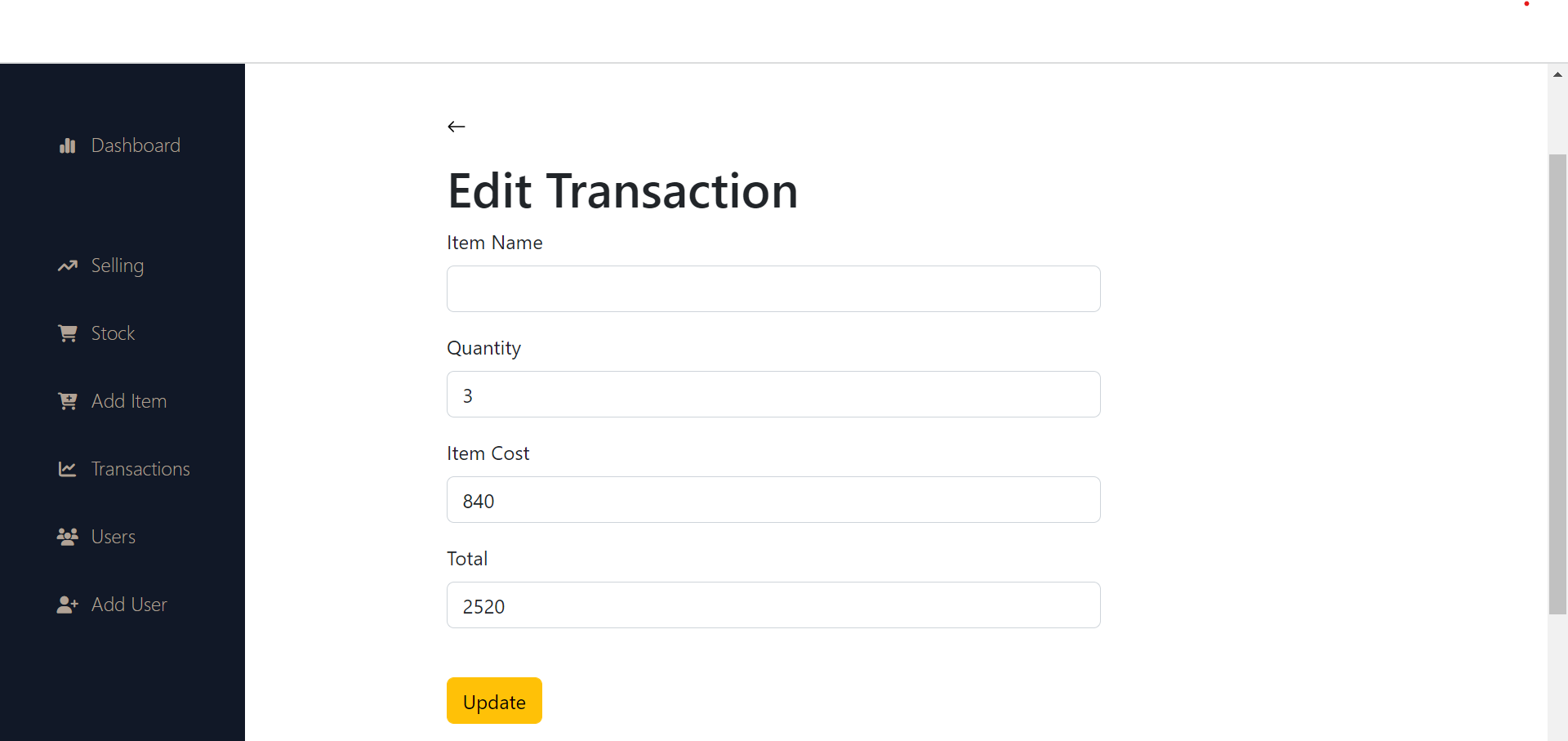


Figure (14): edit transaction page

Figure (14) depicts the edit transaction page, which allows admin or procurement to make changes to transaction.

Graphical user interface, application

Description automatically generated

Figure (15): selling dashboard page

Figure (15) depicts the selling dashboard page, which allows admin or seller to make new transaction.

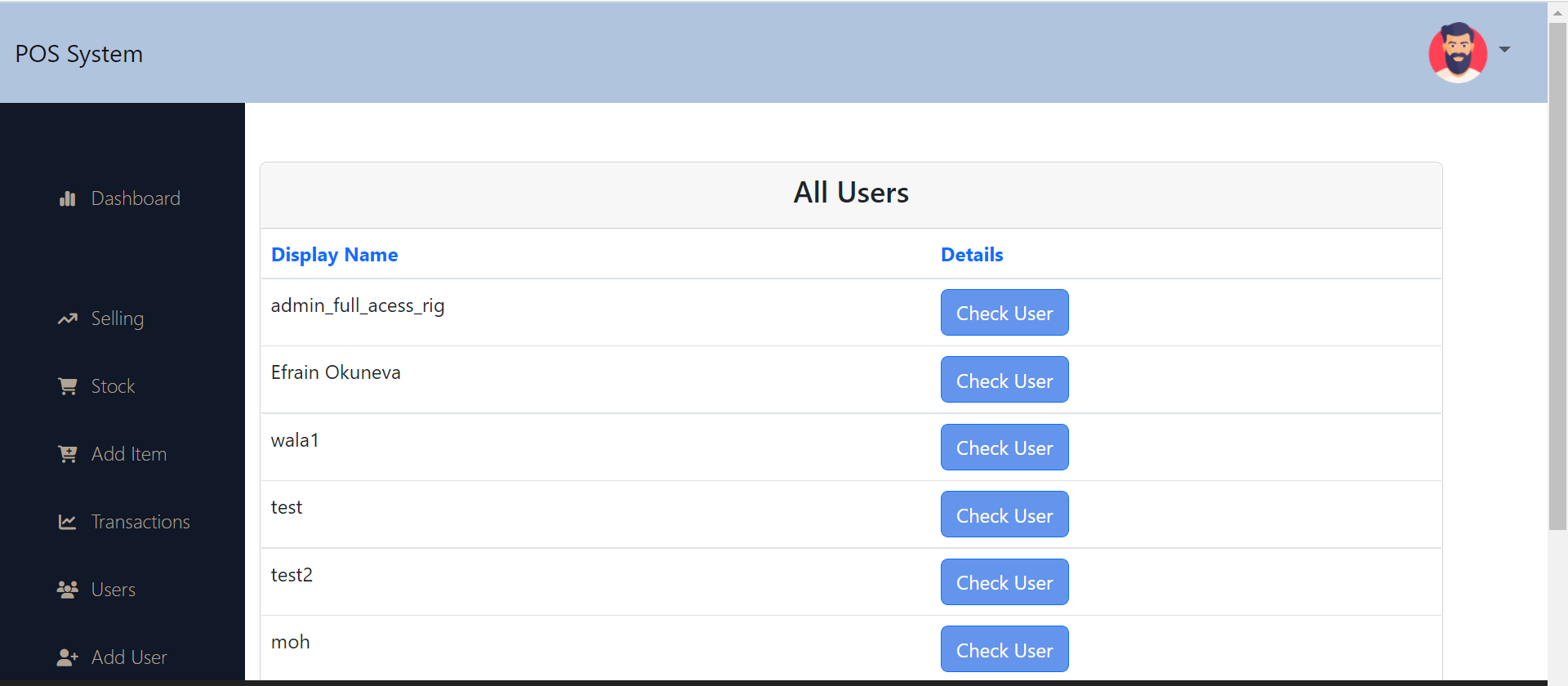


Figure (16): all users page

In figure (16) shows all users page, contain check user when click on it will show user single page.

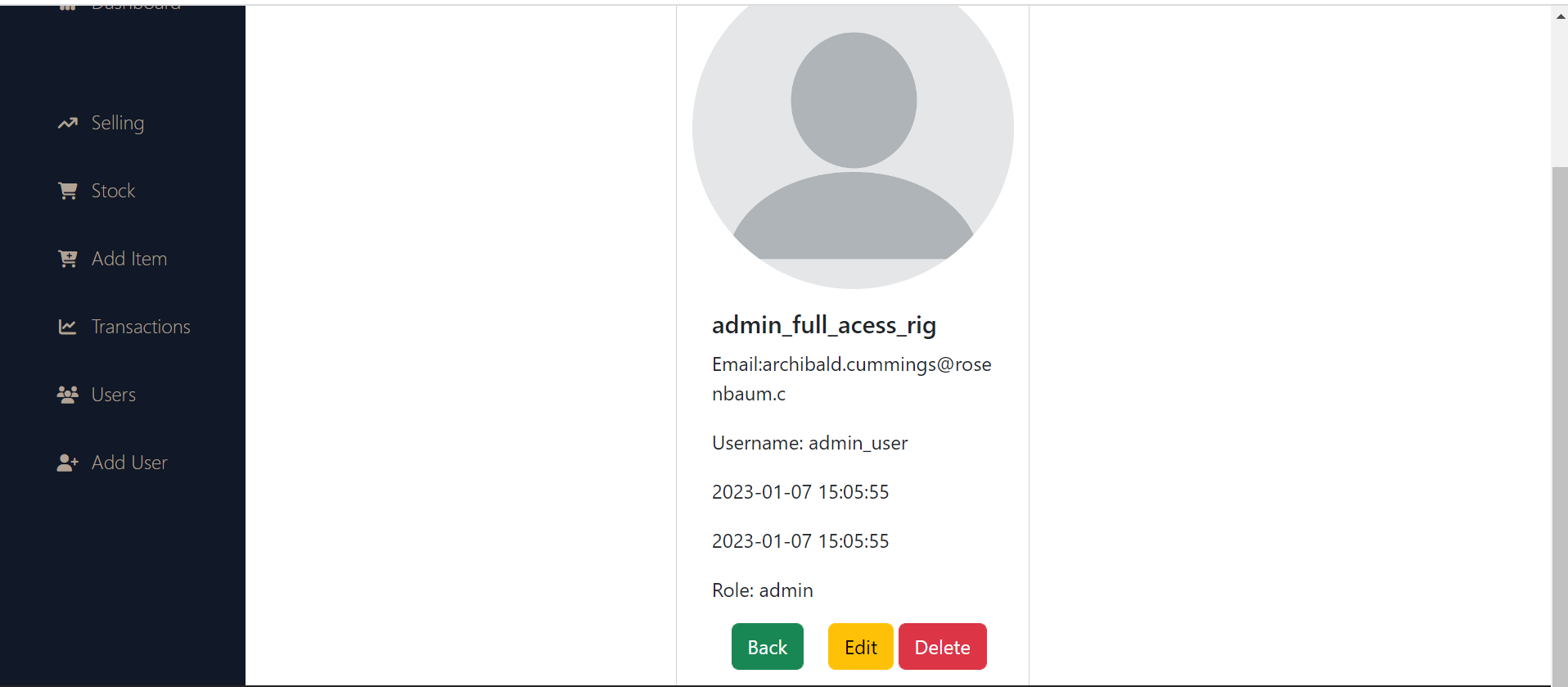


Figure (17): user single page

In figure (17) shows user single page, all user information appears and contains three buttons, one for modification and the other for deletion and another button for back to all users page.

Graphical user interface, text, application, email

Description automatically generated

Figure (18): edit user page

Figure (18) depicts the edit user page, which allows admin changes to user.

Graphical user interface, text, application

Description automatically generated

Figure (19): create user page

Figure (19) depicts create user page, which allows admin create new user.

**Discussion**

In this report, I describe how I used PHP and the MVC design to create a POS (Point of Sale) system that can process transactions and keep track of inventory.

-To store the data for the POS system, I created the database structure. Tables for users, items, and transactions are included in this.

-then I developed the MVC application's models. To retrieve and store data, the models must communicate with the database.

-After that, I made the MVC application's views. The views are in charge of presenting the user interface and enabling user interaction with the system.

-the controllers for your MVC application were then written by me. The controllers are in charge of receiving user input and sending it to the proper model or view, as well as managing communication between the models and views.

-Finally, I tested and fixed the MVC application to make sure it was working properly.

**Conclusions**

In conclusion, the development of a Point of Sale (POS) system is crucial for HTU's store at King Abdullah Business Park in order to effectively handle customer transactions, track sales and inventory, and provide real-time data to the administrator. The POS system will improve efficiency, accuracy, and the overall customer experience, leading to increased profits and a stronger competitive advantage for the store. The scope of the project includes the development of a web application that meets these requirements and does not include physical hardware or integration with external systems. By meeting the objectives of the project, the POS system will enable HTU to effectively manage its store and drive business success.

## Recommendations

The following suggestions can be made based on the POS system project's goals:

1. To give customers a convenient checkout experience, we can integrate a variety of payment methods into the system, including cash, credit and debit cards, and mobile payments.
2. To safeguard sensitive store data, stop fraud, and secure data breaches, implement strong security measures.
3. Maintaining and updating the POS system on a regular basis will help to ensure its dependability and ability to serve the needs of the store.
4. To continuously enhance the POS system's usability and functionality, conduct user testing and solicit input from staff and consumers.
5. F. Marisa and T. G. Yuarita, “Perancangan Aplikasi Point of Sales (Pos) Berbasis Web Menggunakan Metode Siklus Hidup Pengembangan Sistem,” J. Teknol. Dan Manaj. Inform., vol. 3, no. 2, 2017.
6. A. S. Sani, “Pembangunan sistem informasi point of sales terintegrasi dalam lingkup rumah makan beserta cabangnya (studi kasus: RM. Pecel Pincuk bu Tinuk),” 2018.

## Appendix

In figure (20) and figure (21) shows part of code for model to call with database:

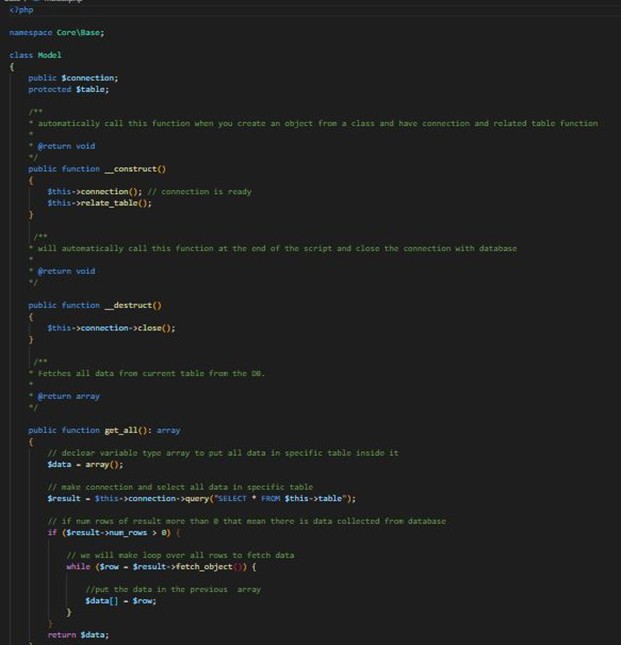


Figure (20): model code



Figure (21): model code

In figure (22) shows view class that responsible for included the php html template

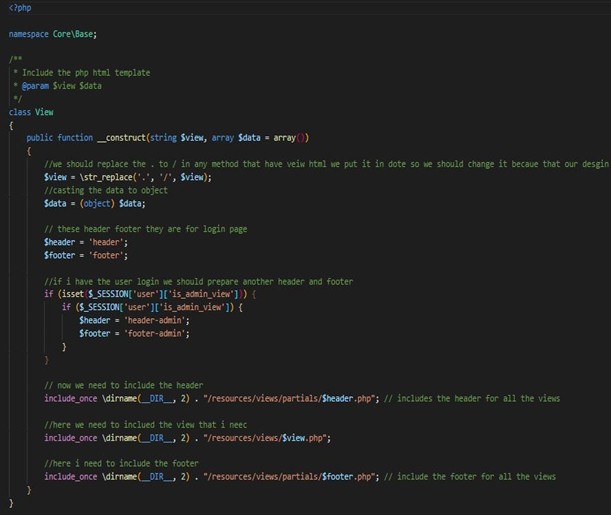


Figure (22): view model

In figure (23) the jquery and ajax code.

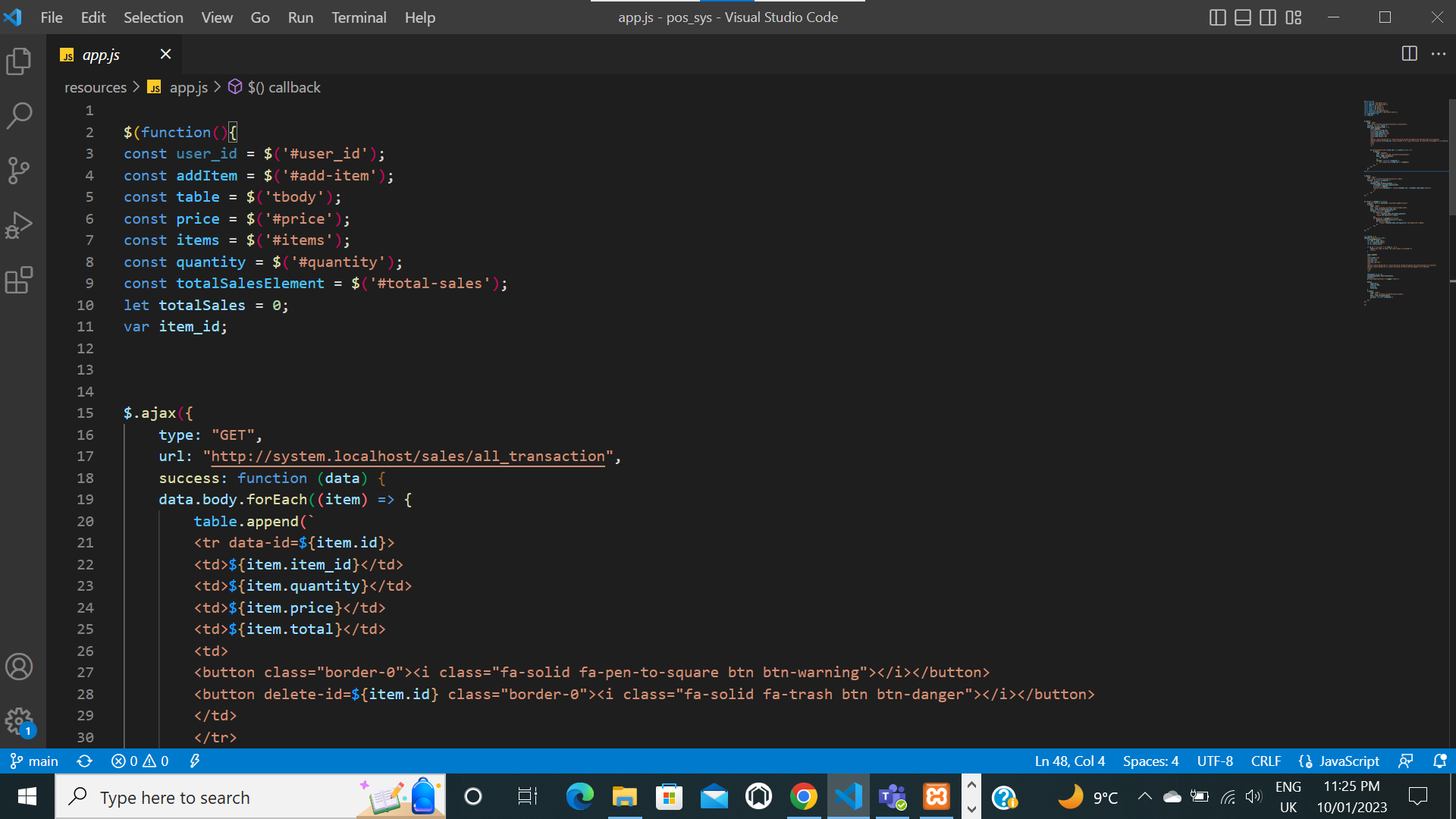


Figure (23): ajax get method

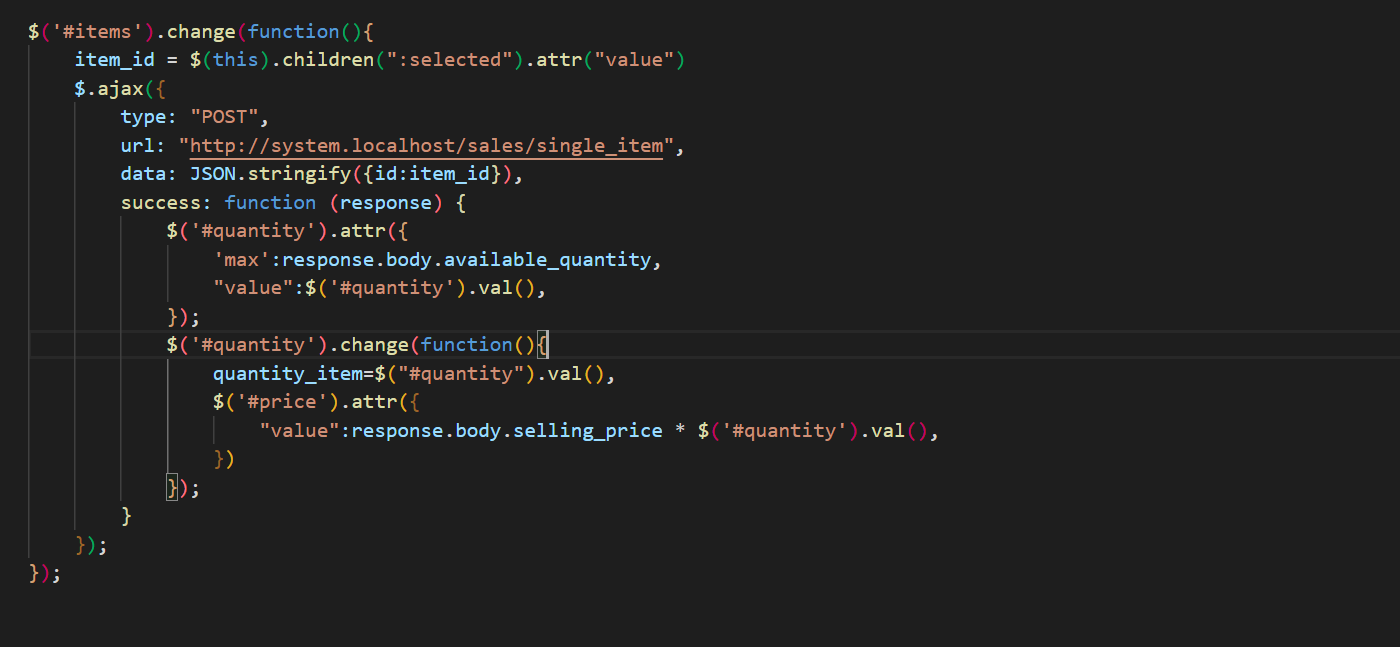


Figure (24): ajax post method

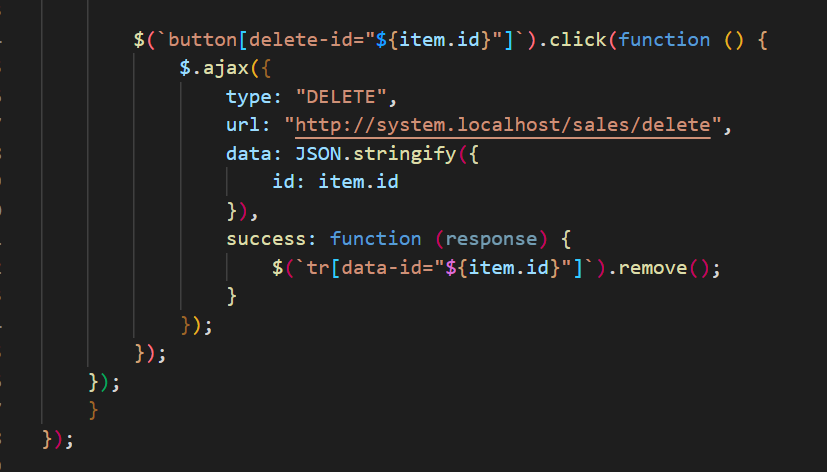


Figure (25): ajax delete method

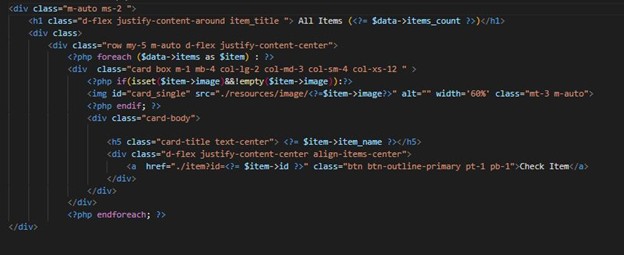


Figure (26): items index page