

# ROYAL POS

---

Technical Report

# Letter Of Transmittal

Dear IMU,

I am pleased to present the attached report on the recently implemented building POS system. This report contains a thorough examination of the system's construction procedures.

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

We hope you find this report useful as you consider the future of your organization's building POS system. Please do not hesitate to contact me if you have any questions or would like to discuss the report further.

Regards,

Mohammad Borini

9 Jan 2023

# Table of contents

Subject number	Subject name	Page number
1	Executive summary	5
2	<b>Introduction</b>	5
3	Problem	6
4	Objectives	6
5	Methodology	7
5.1	Technical Feasibility	7
5.2	For Front-end we used	7
5.3	<b>For Back-end we used</b>	7
6	Result	8-19
7	Discussion	20
8	Conclusion	21
9	Recommendation	21
10	References	22
11	<b>Appendix</b>	23

## GLOSSARY

POS	Point of sales
MVC	Model-control-view

List Of Figures:

Subject number	Subject name	Page number
1	work Flow Diagram	7
2	login page	8
3	QR code	8
4	dashboard	9
5	user report page	9
6	dashboard side bar	10
7	porfile page	10
8	<b>all users</b>	11
9	create new user	11
10	single user information	12
11	create and send message	12
12	edit user information	13
13	selling side bar	13
14	selling dashboard	14
15	list of daily sales	14
16	list of transactions	15

## List Of Figures:

<b>Subject number</b>	<b>Subject name</b>	<b>Page number</b>
17	impty message	15
18	content of message	16
19	all transactions	16
20	edit transactions	17
21	user's profile page	17
22	all items	18
23	create new item page	18
24	<b>item's information page</b>	19
25	Edit item page	19
26	MVC pattern	23
27	barcode analytic	23
28	barcode analytic	24
29	html script for all transactions code	24
30	PHP code to get and show all transactions	25
31	PHPcode to get all transactionsfrom database	25
32	the database	26
33	items table in database	26

## 1. Executive summary

In this report, I described how I used PHP and the MVC design to create a POS system that can process transactions, manage inventory, and process transactions. We used HTML, CSS, Bootstrap, JavaScript, and jQuery on the front end, and PHP, my SQL, and XAMPP server on the back end.

Then we created a database structure that includes tables for items, users, and transactions, and we developed an MVC application model. Then we designed a POS system that can handle transactions, track inventory, and manage items.

And this system will increase sales efficiency, resulting in increased profits and a strong competitive advantage for the store.

## 2. Introduction

To properly record customer and product activity, almost all retail-based operations require a Point of Sale (POS) System.[1] Every business owner wants to know which product sells the most, which customers buy frequently, how much inventory is on hand, what method of payment is preferred by customers (cash, credit, checks, or debit), and how much we actually sold on a given day.[2]

## 3. Problem

Our project's motivation is to develop a Point of Sale (POS) website that effectively addresses the challenges that HTU faces in managing the store at King Abdullah Business Park. The POS system will be in charge of processing customer transactions, tracking sales and inventory, and providing real-time data about the store's performance to the administrator. We hope to improve the efficiency and productivity of the store's operations, as well as the overall customer experience, by automating these processes. Furthermore, the POS website will be designed to be user-friendly and intuitive, allowing employees of various roles to easily use it. Overall, our goal is to create a dependable and efficient POS system that will allow HTU to effectively manage its operations.

## 4. Objectives

- 1-To improve efficiency and accuracy by automating the process of handling customer transactions and tracking sales data.
- 2-To provide real-time data about the store's performance, including sales, inventory levels, and customer information, to the administrator.
- 3-To maintain and update the POS system on a regular basis to ensure that it remains reliable and effective in meeting the needs of the store.
- 4-To ensure that the POS system is user-friendly and intuitive, allowing employees in a variety of roles to use it easily.

### Goals of the Project:

The goal of the POS system project is to create a web application that can handle customer transactions, track sales and inventory, and provide real-time data to the administrator at HTU's King Abdullah Business Park store.

## 5. Methodology

### 1. Technical Feasibility

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

### 2. For Front-end we used :

- HTML
- CSS
- Bootstrap
- JQuery
- JavaScript

### 3. For Back-end we used :

- PHP
- SQL

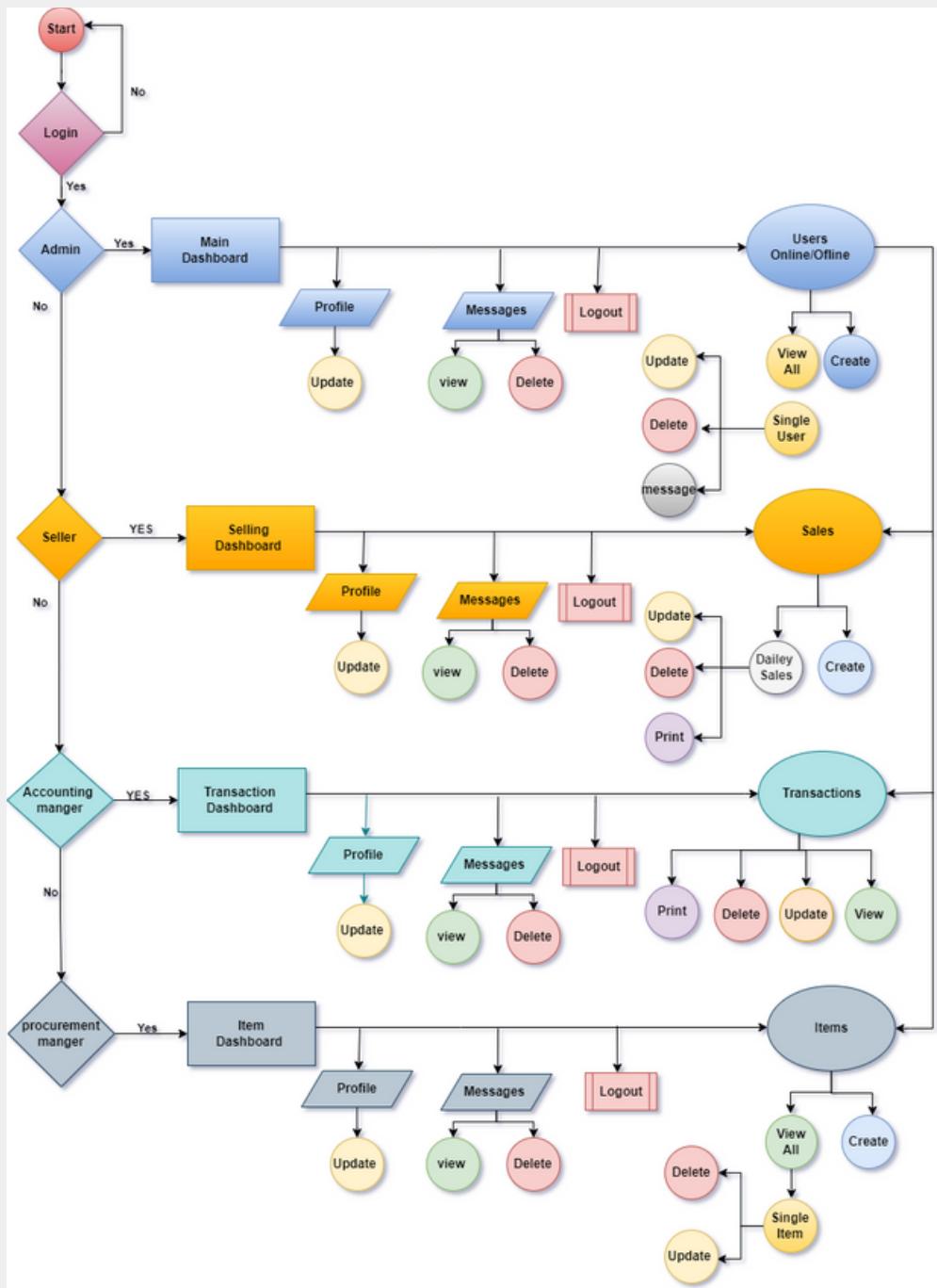


Figure [1]: work Flow Diagram

In figure (1) shows the flow work of POS system that shows the role of users and the dashboard that will redirect the users depend on role and shows all functionality of each users.

## 6.Result

Figure (2) shows initial page that allows the user to interact with the system. So, there are four user roles: Admin, seller, account supervisor procumbent.

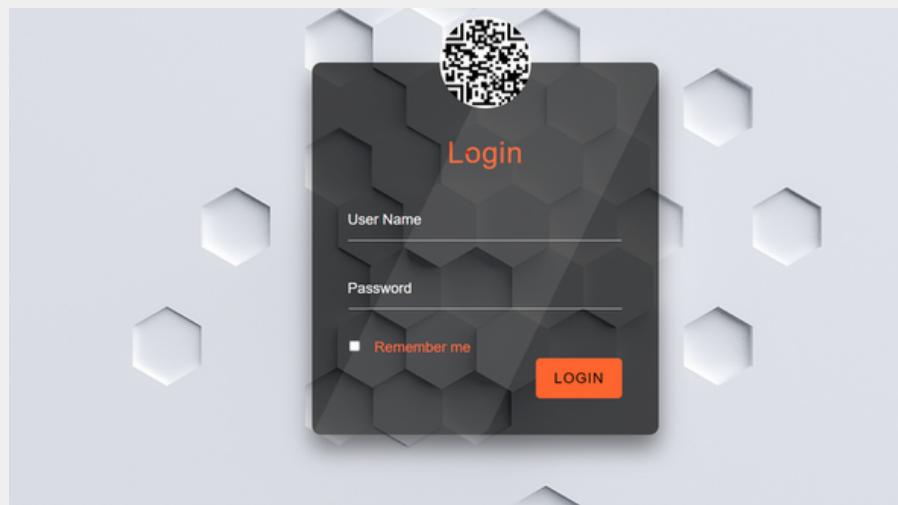


Figure [2]: login page

Figure (3) shows QR code that allows to provide easy access to the website.

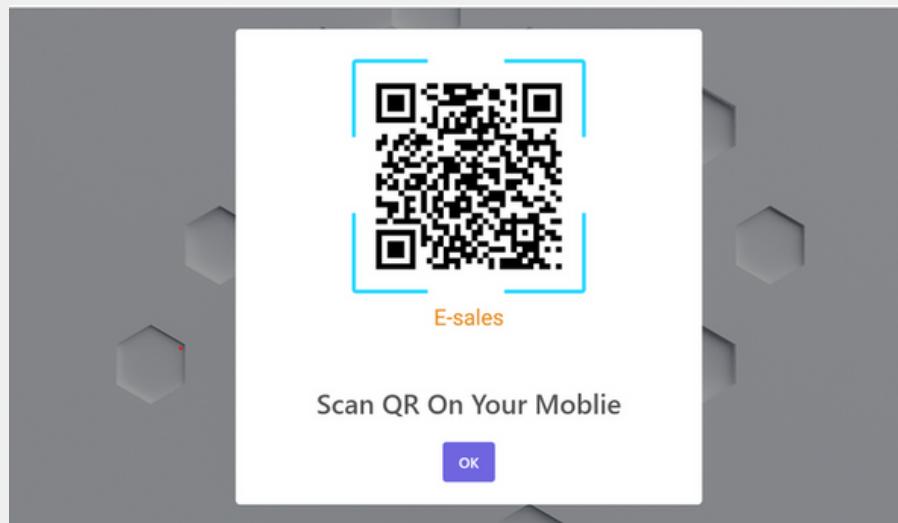


Figure [3]: QR Code for website

Figure [4] shows admin dashboard that if the user is an administrator in the system, it will redirect to the dashboardpage, where the administrator may get some fast information such as "Total item number" and "Total Users number" and "Top 5 Expensive Item to Buy"... and so on. It also has a sidebar buttonthat displays a list of all pages.

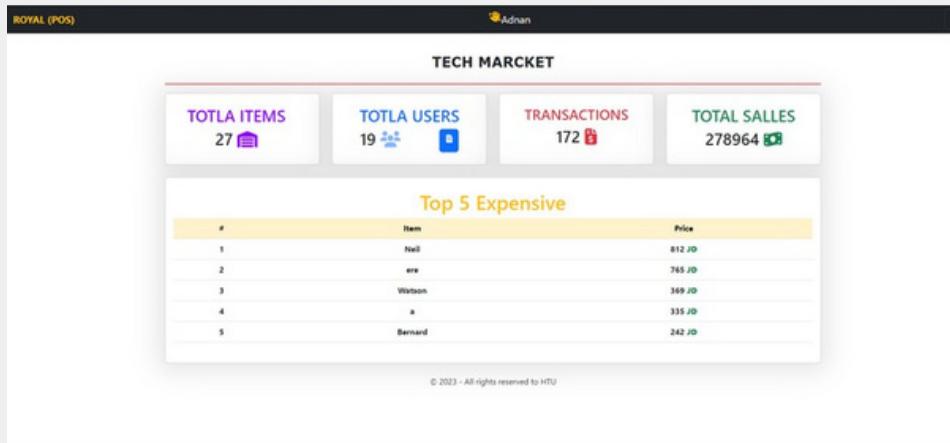


Figure [4]: dashboard page

Figure (5) shows page for all users report and the ability of print it.

user_id	user_name	display Name	email	role	salary	last login	lastlogout	created at	update at
129	Adnan	Adnan	adnan@gmail.com	Admin	100	2023-01-09 00:32:42	2023-01-09 00:28:51	2023-01-07 00:25:30	2023-01-09 00:36:43
130	khaled	khaled	khalid@gmail.com	Admin	100			2023-01-07 00:25:30	2023-01-09 00:34:08
132	fares	fares	fares@gmail.com	Seller	250	2023-01-08 15:43:34	2023-01-08 15:43:58	2023-01-07 00:25:30	2023-01-08 16:49:51
133	mohammad	mohammad	mohammad@yahoo.com	Accountant	100			2023-01-	2023-01-

Figure [5]: All users report pages

Figure (6) shows sidebar has the profilepage and the logout button. Also It includes the pages that the user may accessbased on the role he holds. The administrator has complete accessto the system and can do anything. Only the system administrator has access to the system's users.

The screenshot shows a dark-themed dashboard titled "Selling Dashboard". At the top right are "Profile", "Logout", and a close button. A sidebar on the right lists "Dashboard", "Users", "Items", "Selling", and "Transactions", each with an icon. The main area displays a table with columns "Name", "Quantity", and "Price". The data is as follows:

Name	Quantity	Price	Action
Watch	68	30 JOD	Quantity
holder	1	5 JOD	Quantity
Gell	94	5 JOD	Quantity
Stocker	99	5 JOD	Quantity
vga	96	7 JOD	Quantity

Figure [6]: slide bar for dashboard

Figure (7) shows user's profile page and the user can modify his or her information on the profile page

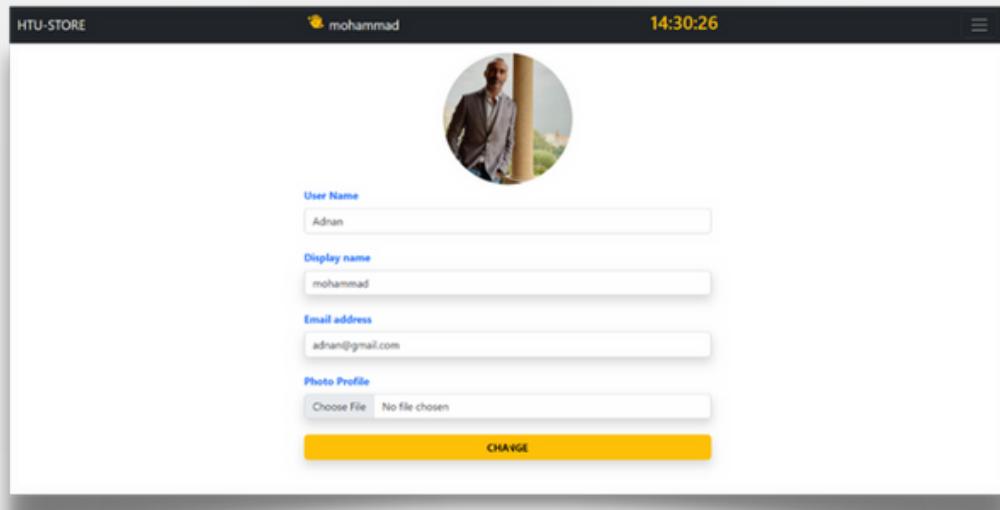


Figure [7]: user's profile page

Figure (8) shows all users page. The user can add new users to the system by pressing the add button Display information about a single user and the status of the users if the other user online or offline.

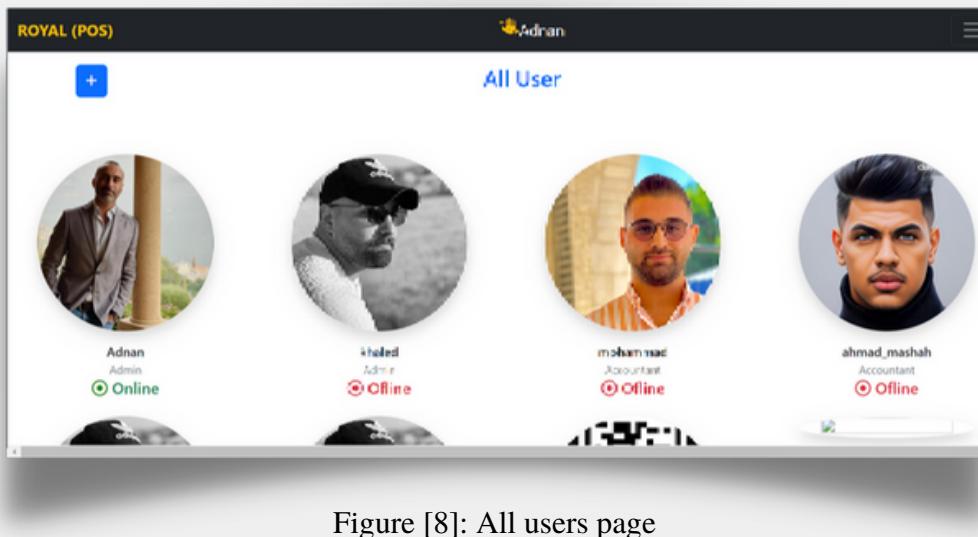


Figure [8]: All users page

Figure (9) shows create item page. The user with role admin or procurement can add new items to the system

Figure (9): create new item

Figure (10) shows user information page the Admin role has the authority to see user information, modify user information, and delete the user from the system

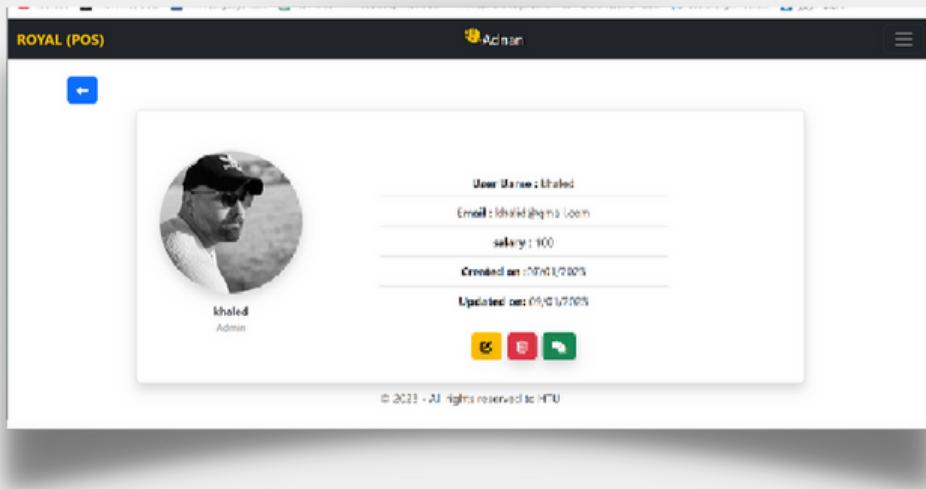


Figure [10]:user information page

Figure (11) shows pop send massage, the users can send the messages to admin. And the admin can read them.

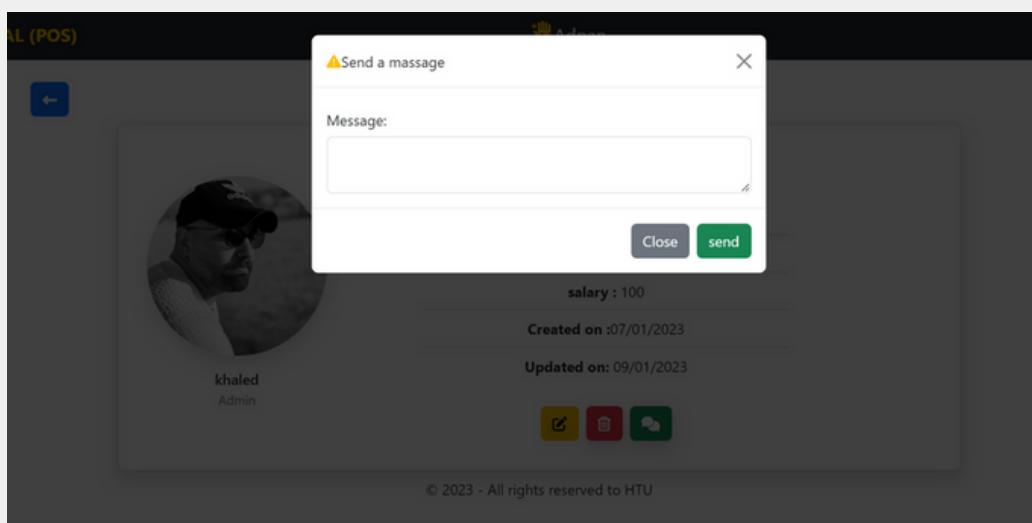


Figure [11]: create and send message

Figure (12) edit user information page that can user modify the information of user

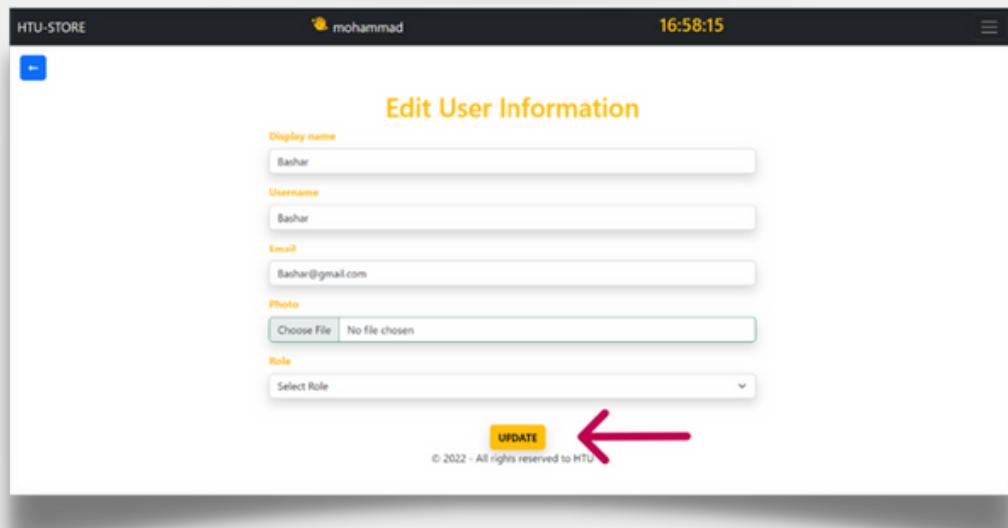


Figure [12]: edit user information

Figure (13) and Figure (14) depicts the selling dashboard page, which allows admin or seller to make new transaction.

photo	Item Name	Item Quantity	Item Price	Quantity
Mouse	Mouse	168	15 JOD	Quantity
Smart Watch	Smart Watch	83	30 JOD	Quantity
Phone Holder	Phone Holder	99	5 JOD	Quantity
Gell	Gell	95	5 JOD	Quantity
Stocker	Stocker	100	5 JOD	Quantity
vga to hdmi	vga to hdmi	99	7 JOD	Quantity
HDMI cable	HDMI cable	68	5 JOD	Quantity

Figure [13]: selling dashboard

The total will appear at the top of the table, and the save button will move the transaction to the daily sales, which he may view by hitting the all sales button.

The screenshot shows a "Selling Dashboard" interface. At the top, there is a search bar labeled "Search Barcode OR Name...". Below the search bar is a table with columns: photo, Item Name, Item Quantity, Item Price, and Add To Cart. The table contains five rows of data:

photo	Item Name	Item Quantity	Item Price	Add To Cart
	Mouse	169	15 JOD	<input type="button" value="Quantity"/> <input style="background-color: #007bff; color: white; border: none; padding: 2px 5px; margin-left: 10px;" type="button" value="ADD"/>
	Smart Watch	83	30 JOD	<input type="button" value="Quantity"/> <input style="background-color: #007bff; color: white; border: none; padding: 2px 5px; margin-left: 10px;" type="button" value="ADD"/>
	Phone Holder	99	5 JOD	<input type="button" value="Quantity"/> <input style="background-color: #007bff; color: white; border: none; padding: 2px 5px; margin-left: 10px;" type="button" value="ADD"/>
	Gell	95	5 JOD	<input type="button" value="Quantity"/> <input style="background-color: #007bff; color: white; border: none; padding: 2px 5px; margin-left: 10px;" type="button" value="ADD"/>
	Stocker	100	5 JOD	<input type="button" value="Quantity"/> <input style="background-color: #007bff; color: white; border: none; padding: 2px 5px; margin-left: 10px;" type="button" value="ADD"/>

Red arrows highlight the "Quantity" input field in the first row, the "Add" button in the first row, and the "All Sales" button in the top right corner.

Figure [14]: selling dashboard

Figure (15) shows the dairy scales buy the logged in user buy the day he enters the system he can also remove the most recent transaction.

The screenshot shows a table titled "Sunday 8th of January 2023 11:35:15 PM" displaying a list of sales transactions. The table has columns: Item, Quantity, Price, Date, and Action buttons (Edit, Delete). The data in the table is as follows:

Item	Quantity	Price	Date	Action
Mouse	11	22	08/01/2023	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Mouse	11	44	08/01/2023	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Mouse	11	44	08/01/2023	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Mouse	11	77	08/01/2023	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Mouse	11	88	08/01/2023	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Mouse	11	77	08/01/2023	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Mouse	11	22	08/01/2023	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Mouse	11	594	08/01/2023	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Mouse	11	11	08/01/2023	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Mouse	11	11	08/01/2023	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Watch	30	60	2023-01-09 00:35:26	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Figure [15]: list of daily sales

Figure (16) shows list of transactions. When an accountant logs into the system, he or she is sent to the transaction dashboard page.

The screenshot displays a table titled "All Transactions" under the "Mariam" user profile. The table has columns: Trans #, Created by, Item Name, Quantity, Total, Created At, and Updated At. Each row represents a purchase record. The data includes items like headphones, speakers, and mice from various users (USER\_1, USER\_13, Mariam) at different dates and times.

Trans #	Created by	Item Name	Quantity	Total	Created At	Updated At
330	USER_1	headphones	2	40	2023-01-16 20:50:12	2023-01-16 20:50:12
325	Mariam	speaker	1	15	2023-01-11 17:40:46	2023-01-11 17:40:46
324	Mariam	speaker	1	15	2023-01-11 17:40:02	2023-01-11 17:40:02
323	Mariam	speaker	1	15	2023-01-11 17:39:43	2023-01-11 17:39:43
320	USER_13	speaker	1	15	2023-01-11 11:42:21	2023-01-11 11:42:21
319	Mariam	speaker	1	15	2023-01-10 18:20:44	2023-01-10 18:20:44
316	Mariam	Mouse	2	22	2023-01-09 13:38:06	2023-01-09 13:38:12
313	Mariam	Mouse	2	22	2023-01-09 13:35:12	2023-01-09 13:35:20
312	Mariam	Mouse	2	22	2023-01-09 13:34:32	2023-01-09 13:34:49
311	Mariam	Mouse	1	11	2023-01-09 13:34:27	2023-01-09 13:34:27
310	Mariam	Mouse	1	11	2023-01-06 01:03:59	2023-01-06 01:03:59
309	Mariam	holder	1	5	2023-01-06 01:03:43	2023-01-06 01:03:43
308	Mariam	JBL	1	40	2023-01-06 01:03:41	2023-01-06 01:03:41
307	Mariam	Watch	3	90	2023-01-06 01:03:38	2023-01-06 01:03:38
306	Mariam	Mouse	1	11	2023-01-06 01:03:34	2023-01-06 01:03:34
305	Mariam	Mouse	1	11	2023-01-06 01:02:24	2023-01-06 01:02:24

Figure [16]: list of transactions

Figure (17) pop message, if the admin doesn't have message and check to message button it's will appear

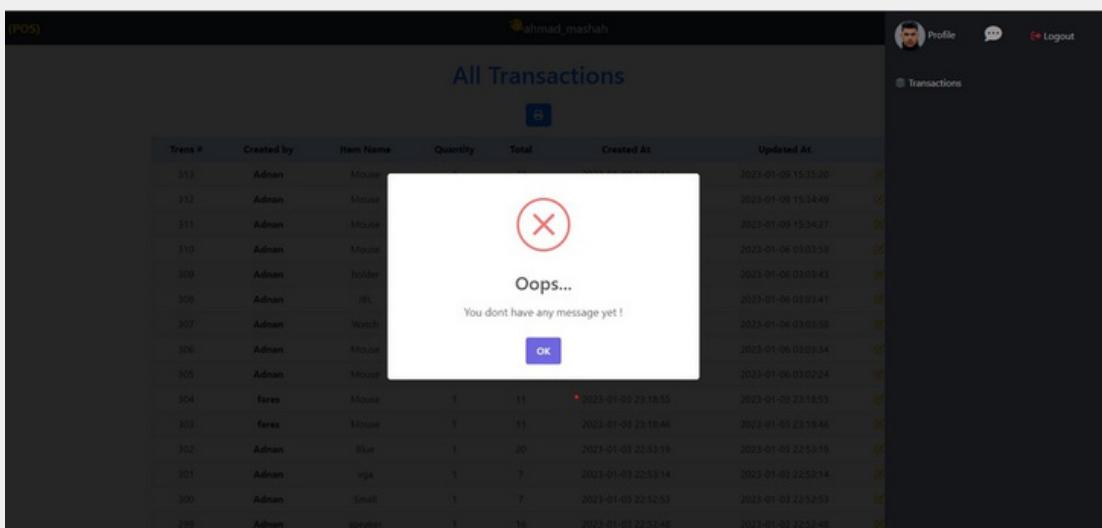


Figure [17]: message

Figure (17) shows pop message, this page displays all transactions in the system that include the name of the user that he created. He can also delete and alter any transactions

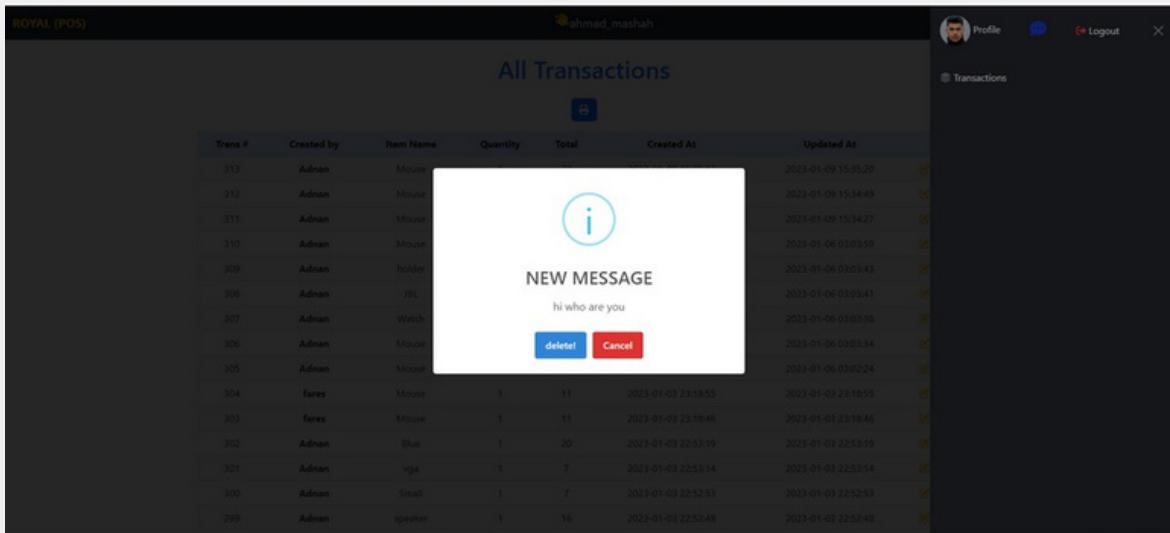


Figure [18]: pop for new message

Figure (18) shows all transaction report, that can user print the all transactions.

Trans #	Created by	Item Name	Quantity	Total	Created At	Updated At
313	Adnan	Mouse	2	22	2023-01-09 15:35:12	2023-01-09 15:35:20
312	Adnan	Mouse	2	22	2023-01-09 15:34:32	2023-01-09 15:34:49
311	Adnan	Mouse	1	11	2023-01-09 15:34:27	2023-01-09 15:34:27
310	Adnan	Mouse	1	11	2023-01-06 03:03:59	2023-01-06 03:03:59
309	Adnan	holder	1	5	2023-01-06 03:03:43	2023-01-06 03:03:43
308	Adnan	JBL	1	40	2023-01-06 03:03:41	2023-01-06 03:03:41
307	Adnan	Watch	3	90	2023-01-06 03:03:38	2023-01-06 03:03:38
306	Adnan	Mouse	1	11	2023-01-06 03:03:34	2023-01-06 03:03:34
305	Adnan	Mouse	1	11	2023-01-06 03:02:24	2023-01-06 03:02:24
304	fares	Mouse	1	11	2023-01-03 23:18:55	2023-01-03 23:18:55
303	fares	Mouse	1	11	2023-01-03 23:18:46	2023-01-03 23:18:46
302	Adnan	Blue	1	20	2023-01-03 22:53:19	2023-01-03 22:53:19
301	Adnan	vga	1	7	2023-01-03 22:53:14	2023-01-03 22:53:14
300	Adnan	Small	1	7	2023-01-03 22:52:53	2023-01-03 22:52:53
299	Adnan	speaker	1	16	2023-01-03 22:52:48	2023-01-03 22:52:48

Figure [19]: all transactions

Figure (19) shows the edit transaction page. The user can modify the transactions.

The screenshot shows a web application interface titled "Edit Transaction". At the top, there is a header with "HTU-STORE", a user profile icon for "mohammad", the time "15:24:18", and a menu icon. The main content area has a title "Edit Transaction" in yellow. It contains four input fields: "Item Name" with value "nest", "price" with value "4", "Item Quantity" with value "1", and "Total" with value "4". Below these fields is a yellow "SAVE" button. At the bottom of the page, a copyright notice reads "© 2022 - All rights reserved to HTU".

Figure [20]: edit transactions

Figure (21) shows the user's profile page, the user can modify his information and update profile photo

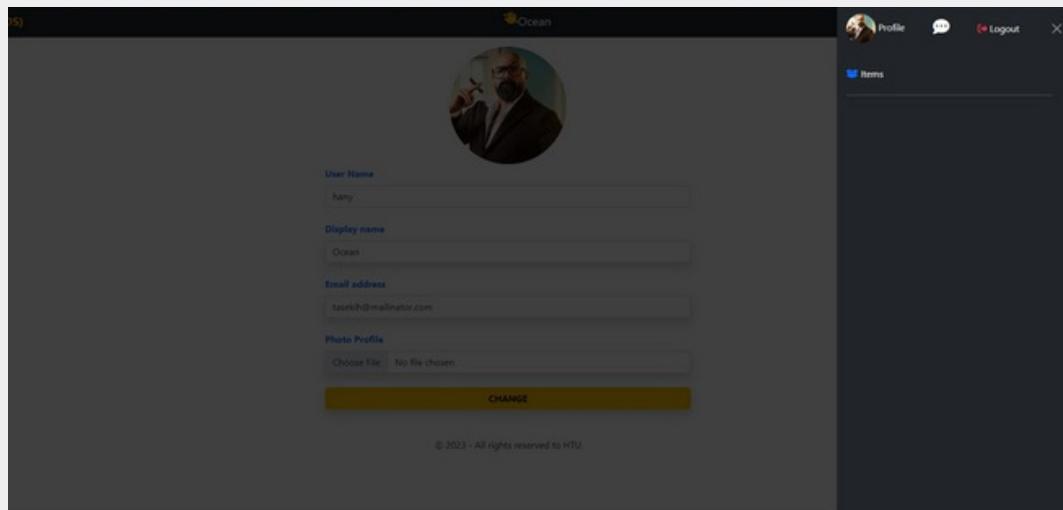


Figure [21]: user's profile page

Figure (22) shows all items page, this page displays all items in the database.

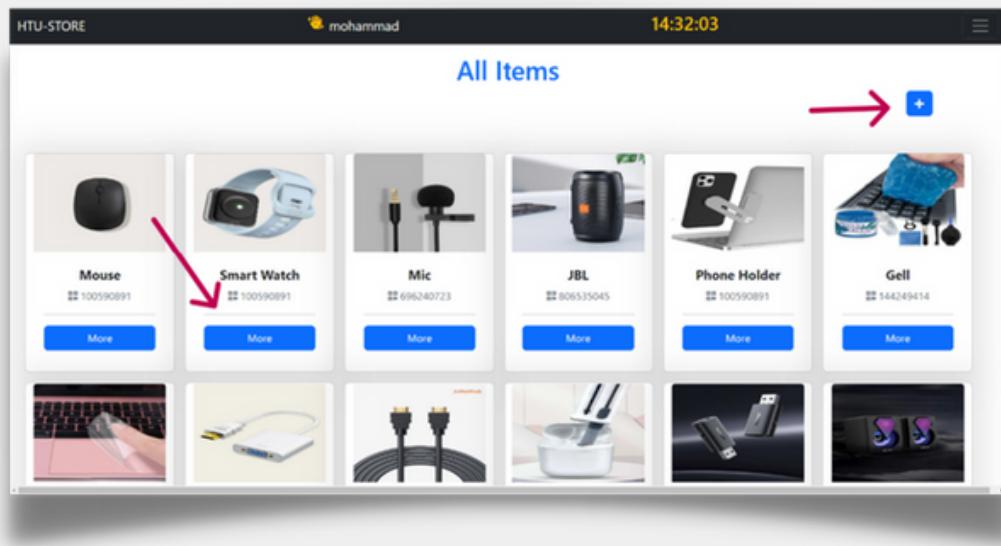


Figure [22]: all items

Figure (23) create new item page. The admin and the procurement add new item.

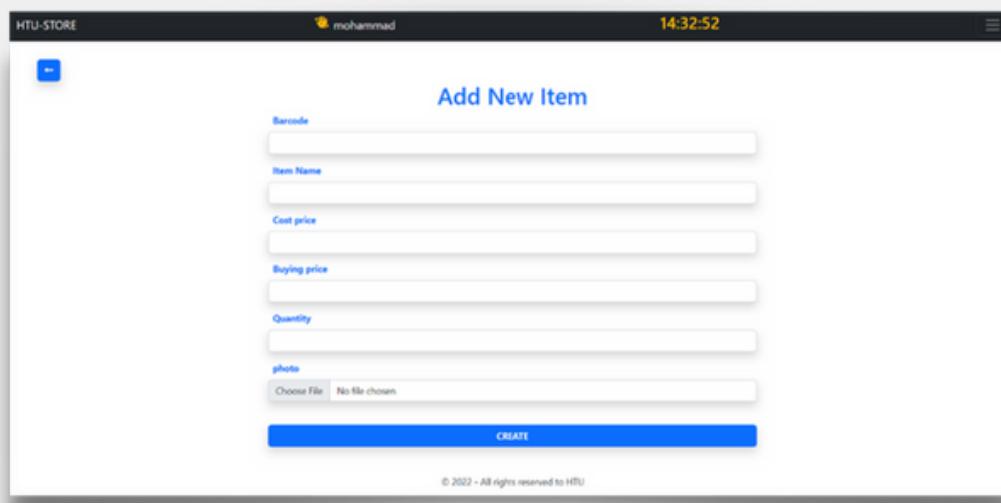


Figure [23]: create new item page

Figure (24) shows item information page He may also provide complete details for any item and edit or change it

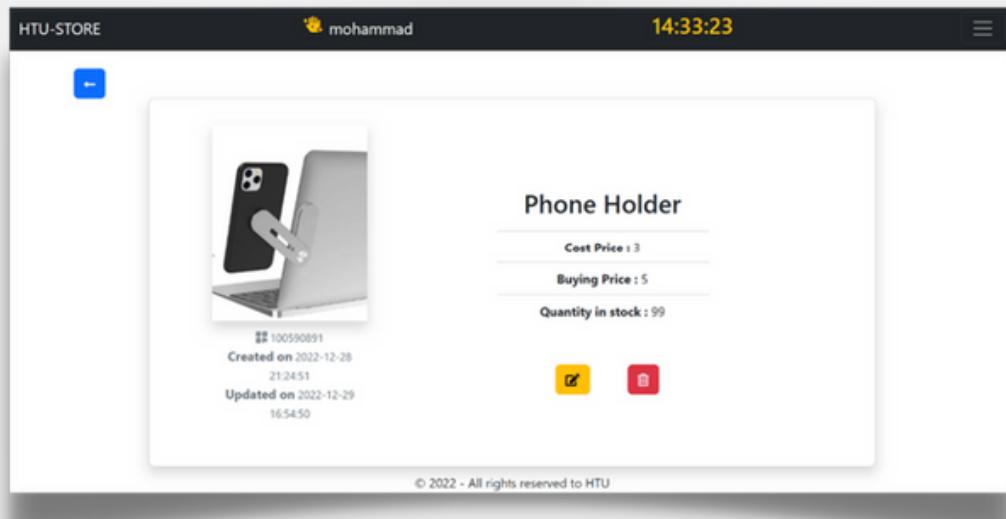


Figure [24]: item's information page

Figure (25) shows edit item page , This is the form for updating item information.

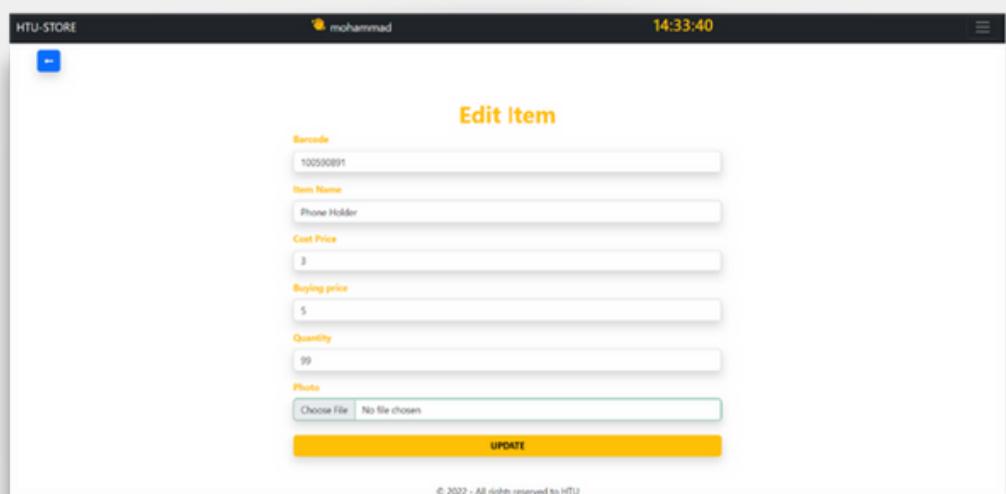


Figure [25]: Edit item page

## 7.Discussion

In this report, I built a POS (Point of Sale) system that has the ability to process transactions and track inventory using MVC pattern and PHP, and I followed these steps:

- I designed the database schema to store the information for the POS system. This includes tables for products, users, and transactions.
- Then I created the models for the MVC application. The models are responsible for interacting with the database to retrieve and store data.
- After that, I created the views for MVC application. The views will be responsible for displaying the user interface and allowing the user to interact with the system.
- Then I created the controllers for your MVC application. The controllers will be responsible for managing the communication between the models and the views, and for handling user input and directing it to the appropriate model or view.
- Finally, I Tested and debugged MVC application to ensure that it is functioning correctly.

## 8. Conclusion

In order to efficiently manage client transactions, track sales and inventory, and give the administrator real-time data, a Point of Sale (POS) system is essential for HTU's store in King Abdullah Business Park. The POS system will boost productivity, accuracy, and the entire customer experience, boosting sales and giving the store a greater competitive edge. The creation of a web application that complies with these specifications is included in the project's scope; physical hardware or system integration are not. The POS system will enable HTU to successfully run its store and advance commercial success by realizing the project's goals.

## 9. Recommendation

Based on the objectives of the POS system project, the following recommendations can be made:

- we can Incorporate various payment methods into the system, including cash, credit and debit cards, and mobile payments, to provide customers with a convenient checkout experience.
- Implement robust security measures to protect sensitive store data, and prevent fraud and security breaches.
- Regularly maintain and update the POS system to ensure it remains reliable and effective in meeting the needs of the store.
- Conduct user testing and gather feedback from employees and customers to continuously improve the functionality and usability of the POS system.

## 10. References

- [1] H. Hendriyanto and P. A. Cakranegara, “Web-Based Online Sales Information System Using PHP and MYSQL Database in Nara Collection,” JMKSP J. Manaj. Kepemimp. Dan Supervisi Pendidik., vol. 7, no. 1, pp. 35–52, 2022.
- [2] I. A. Dewi, Y. Miftahuddin, M. A. Fattah, C. B. Palenda, and S. F. Erawan, “Point of Sales System in InHome Café Website using Agile Methodology,” J. Innov. Community Engagem., vol. 1, no. 1, pp. 01–19, 2021.
- [3] M. Jailia, A. Kumar, M. Agarwal, and I. Sinha, “Behavior of MVC (Model View Controller) based Web Application developed in PHP and. NET framework,” presented at the 2016 International Conference on ICT in Business Industry & Government (ICTBIG), 2016, pp. 1–5.
- [4] G. Ballard, “Managing work flow on design projects,” CIB Rep., pp. 17–30, 2001.
- [5] H.-F. Tsai, B. R. Chang, H.-C. Huang, and C.-F. Huang, “Implementation of mobile point of sale cashier management system,” Sens. Mater., vol. 31, no. 8, pp. 2463–2476, 2019.

## 11.Appendix

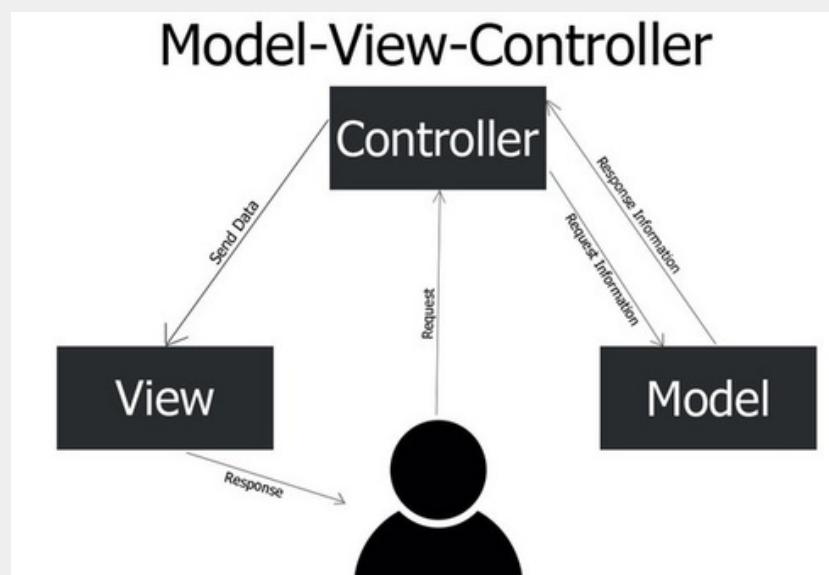


figure [26]: MVC pattern

figure (27) shows graph between the number of scan and the time and we can see from the graph the highest number of scan in Jun 9 2023 with six scans.

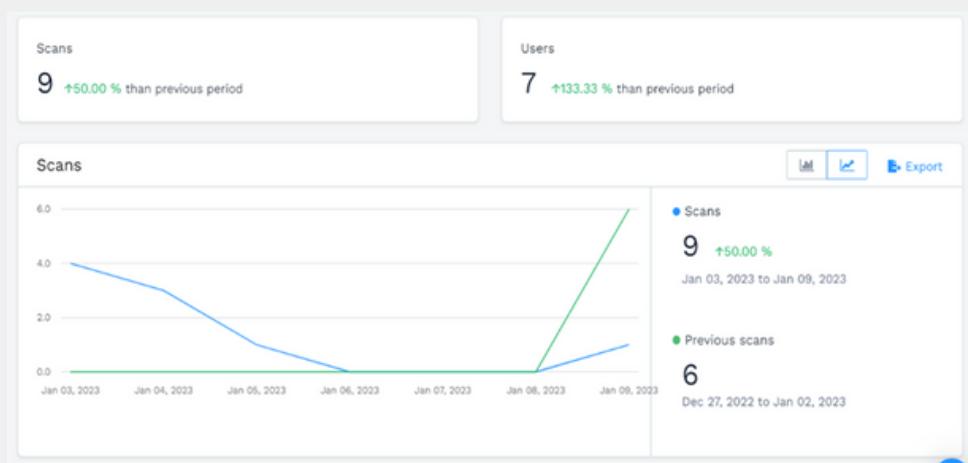


figure [27]: barcode analytic

# ROYAL pos

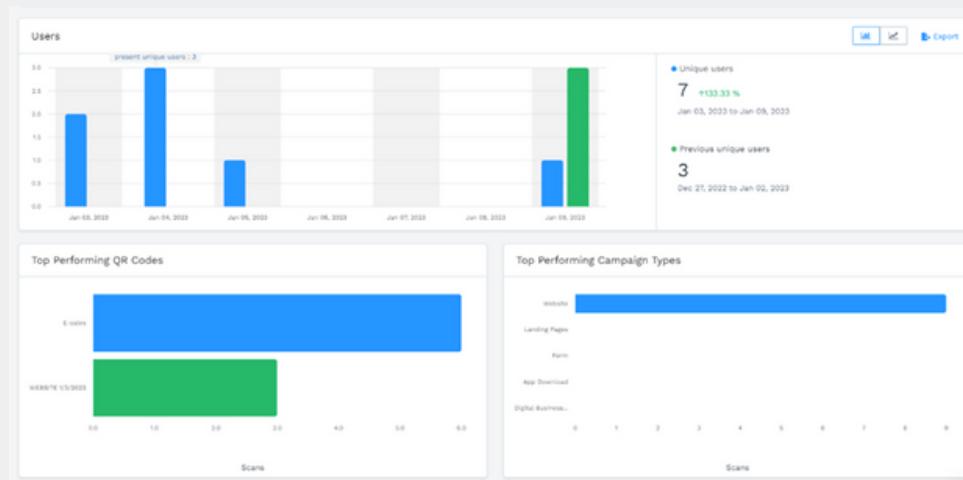


Figure [28]: barcode analytic

Part of code related to all transactions

```
</div>
<div class="d-flex container justify-content-center md-3 col-xxl-12">
    <table class="table table-striped frf text-center">

        <thead class="table-primary">
            <tr>
                <th scope="col">Trans #</th>
                <th scope="col">Created by</th>
                <th scope="col">Item Name</th>
                <th scope="col">Quantity</th>
                <th scope="col">Total</th>
                <th scope="col" class="hid">Created At</th>
                <th scope="col" class="hid">Updated At</th>
                <th scope="col" class="w_p"></th>
                <th scope="col" class="w_p"></th>
            </tr>
        </thead>
        <tbody>
            <tr class="text-center bolder"><?php foreach ($data->transaction as $transactions) : ?>
                <td><?php echo $transactions->transaction_id ?></td>
                <th scope="row"><?php echo $transactions->username ?></th>
                <td><?php echo $transactions->item_name ?></td>
                <td><?php echo $transactions->item_quantity ?></td>
                <td><?php echo $transactions->total ?></td>
                <td class="hid"><?php echo $transactions->created_at ?></td>
                <td class="hid"><?php echo $transactions->update_at ?></td>
                <td class="w_p">
                    <a href="/transaction/edit?id=<?php echo $transactions->id?>">
                        class="w-100 justify-content-center text-warning "><i class="fas fa-edit"/></a>
                </td>
                <td class="w_p">
                    <a href="/transaction/delete?id=<?php echo $transactions->id ?>" type="button" class="w-100"
                        data-bs-toggle="modal" data-bs-target="#staticBackdrop">
                        <i class="fas fa-trash-alt text-danger"/></a>
                    <div class="modal fade" id="staticBackdrop" data-bs-backdrop="static" data-bs-keyboard="false"
                        tabindex="-1" aria-labelledby="staticBackdropLabel" aria-hidden="true">
                        <div class="modal-dialog">
                            <div class="modal-content">
                                <div class="modal-header">
                                    <h1 class="modal-title fs-5" id="staticBackdropLabel">Attention</h1>
                                    <button type="button" class="btn-close" data-bs-dismiss="modal"
                                        aria-label="Close"></button>
                                </div>
                                <div class="modal-body">
                                    Are You Sure To Delete This Transaction?
                                </div>
                                <div class="modal-footer">
                                    <button type="button" class="btn btn-secondary"
                                        data-bs-dismiss="modal">No</button>
                                    <a href="/transaction/delete?id=<?php echo $transactions->id?>">
                                        class="btn btn-danger d-flex justify-content-center ">Yes Delete</a>
                                    </div>
                            </div>
                        </div>
                </td>
            <?php endforeach; ?>
        </tbody>
    </table>
</div>
```

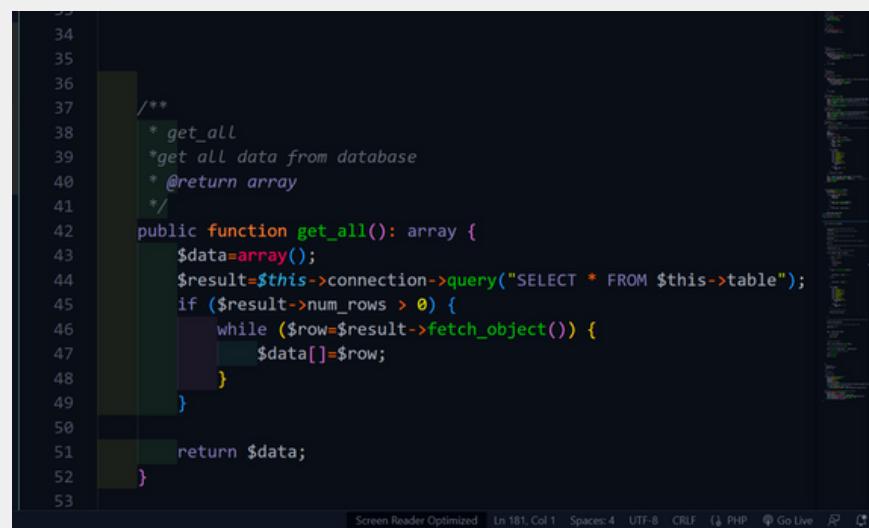
Figure [29]: html script for all transactions code

# ROYAL pos

```
    public function index()

    {
        /*PERMISSIONS
        $this->permissions(['transaction:read']);
        /*NEW TRANSACTION MODEL
        $transactions= new Transaction;
        /*NEW USER MODEL
        $user = new User;
        /* GET THE USER DISPLAY NAME AND THE PHOTO TO DISPLAY THEM IN HEADER
        $user_info=$user->get_by_id($_SESSION['user']['user_id']);
        $this->data['display_name']=$user_info->display_name;
        $this->data['photo']=$user_info->photo;
        $this->data['message']=$user_info->message;
        /* DISPLAY TRANSACTION PAGE
        $this->view = 'transactions.index';
        /* SELECT DATA FROM TRANSACTION_USER TABLE (REALATION TABLE)
        $result=$transactions->connection->query("SELECT users.username
        ,transactions.id,transactions.item_name, transactions.item_quantity,
        transactions.total,transactions.update_at,transactions.created_at,transaction_user.*"
        FROM transaction_user
        JOIN transactions ON transaction_user.transaction_id = transactions.id
        JOIN users ON transaction_user.user_id = users.id ORDER BY transaction_user.transaction_id DESC ");
        $transactions_info = array();
        /*LOOP ON DATA
        if ($result->num_rows > 0) {
            while ($row = $result->fetch_object()) {
                $transactions_info[] = $row;
            }
        }
        /*SET THE DATA IN ARRAY
        $this->data['transaction'] = $transactions_info;
    }
```

Figure [30]: PHP code to get and show all transactions



The screenshot shows a code editor window with a dark theme. The code is written in PHP and defines a public function named `get_all()`. The function uses a database connection to query all rows from the table associated with the current object. It then loops through the results, creating an array where each element is a row object. Finally, it returns this array. The code is annotated with JSDoc-style comments describing the function's purpose and return type.

```
34
35
36
37 /**
38 * get_all
39 *get all data from database
40 * @return array
41 */
42 public function get_all(): array {
43     $data=array();
44     $result=$this->connection->query("SELECT * FROM $this->table");
45     if ($result->num_rows > 0) {
46         while ($row=$result->fetch_object()) {
47             $data[]=$row;
48         }
49     }
50
51     return $data;
52 }
```

Figure [31]: PHPcode to get all transactionsfrom database

# ROYAL pos

The screenshot shows the MySQL Workbench interface. At the top, there is a search bar labeled "Containing the word:" followed by a text input field. Below this is a table listing the database schema:

Table	Action	Rows	Type	Collation	Size	Overhead
items		27	InnoDB	utf8mb4_general_ci	128.0 Kib	-
transactions		10	InnoDB	utf8mb4_general_ci	16.0 Kib	-
transaction_user		10	InnoDB	utf8mb4_general_ci	16.0 Kib	-
users		19	InnoDB	utf8mb4_general_ci	48.0 Kib	-
4 tables	Sum	66	InnoDB	utf8_general_ci	208.0 Kib	0 B

At the bottom left, there is a navigation icon (arrow pointing up) and a checkbox labeled "Check all". To the right of the checkbox is a dropdown menu labeled "With selected:".

Figure [32]: the database

The screenshot shows the MySQL Workbench interface displaying the "items" table. The table has the following columns: id, item\_name, item\_price, item\_quantity, total, created\_at, and update\_at. The data is as follows:

				id	item_name	item_price	item_quantity
				1	Mouse	11	10
				5	JBL	40	10
				43	JBL	40	4
				55	JBL	40	12
				58	JBL	40	3
				59	JBL	40	1
				62	JBL	40	1
				63	Watch	30	0
				66	JBL	40	9
				68	holder	5	11

Figure [33]:items table in database