

Billing And Inventory Management System

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Engineering

by

**Aarya Khatate
Chetan Chaudhari**

**Vinay Gone
Yashraj Patil**

Supervisor:

Asst. Prof. Tayyabali Sayyad



UNIVERSITY OF MUMBAI

Billing And Inventory Management System

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Engineering

by

**Aarya Khatate
Chetan Chaudhari
Vinay Gone
Yashraj Patil**

Supervisor:- Asst. Prof. Tayyabali Sayyad



Department of Information Technology

**Don Bosco Institute of Technology
Vidyavihar Station Road, Mumbai - 400070**

2024-2025

DON BOSCO INSTITUTE OF TECHNOLOGY

Vidyavihar Station Road, Mumbai - 400070

Department of Information Technology

CERTIFICATE

This is to certify that the project entitled “**Billing and Inventory Management System** ” is a bonafide work of the following students:

Aarya Khatate	28
Chetan Chaudhari	10
Vinay Gone	63
Yashraj Patil	43

submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Undergraduate** in **Bachelor of Information Technology**.

Date: / /

(Prof. Tayyabali Sayyad)
Supervisor

(Prof. Prasad Padalkar)
HOD, IT Department

(Dr. Sudhakar Mande)
Principal

DON BOSCO INSTITUTE OF TECHNOLOGY

Vidyavihar Station Road, Mumbai - 400070

Department of Information Technology

Project Report Approval for S.E.

This project report entitled “**Billing And Inventory Management System**” is approved for the Mini Project of **Second Year Semester III in Information Technology**

(Examiner's Name and Signature)

1. _____

2. _____

(Supervisor's Name and Signature)

1. _____

(Chairman)

1. _____

Date:

Place:

DON BOSCO INSTITUTE OF TECHNOLOGY

Vidyavihar Station Road, Mumbai - 400070

Department of Information Technology

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact / source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Name	Roll No.	Signature
Aarya Khatate	28	
Chetan Chaudhari	10	
Vinay Gone	63	
Yashraj Patil	43	

Date:

Abstract

This project presents a comprehensive Billing and Inventory Management System designed for small to medium-sized businesses, built using JavaFX and MySQL. The system enables shop owners and their employees to efficiently manage product inventory, generate customer bills, and handle user management. Each shop owner and their respective employees have restricted access to their shop's data, ensuring data privacy and security across multiple users. Shop owners have full control over product inventory, bill generation, and employee management, while employees are limited to bill creation functionalities.

The system is designed to simplify inventory management by automatically logging low-quantity alerts and ensuring that the inventory is updated in real time. Additionally, the system logs every critical activity, such as product updates, bill creation, user login/logout, and employee management. These logs serve as an audit trail, providing transparency and accountability for all system users.

The project is implemented with a MySQL database, which stores all relevant data such as products, users, bills, and logs. The application follows the MVC (Model-View-Controller) architecture, with a user-friendly interface created in Scene Builder and implemented using JavaFX. The database design ensures that each user's actions are accurately tracked, and product data is consistently tied to the relevant shop owner. The system's extensibility and real-time processing make it a valuable tool for shop owners looking to streamline their operations and enhance their overall efficiency.

Keywords: Billing System, Inventory Management, JavaFX, MySQL, User Management, Real-time Updates, Activity Logs, Multi-user System, Data Security

Contents

1	Introduction	10
2	Review	11
3	Analysis and Design	12
3.1	Entity Relationship Diagram	12
3.2	Database Design	13
3.2.1	Owner Table	13
3.2.2	Product Table	13
3.2.3	Employee Table	14
3.2.4	Order Table	14
3.2.5	Bill Table	14
3.2.6	Logs Table	15
3.3	Triggers	15
3.3.1	After Owner Insert	15
3.3.2	After Owner Update	16
3.3.3	After Owner Delete	16
3.3.4	After Employee Insert	17
3.3.5	After Employee Update	17
3.3.6	After Employee Delete	18
3.4	Database Hosting	18
4	Implementation	19
4.1	Login	19
4.2	Registration	20
4.3	Dashboard	23
4.4	Inventory	24
4.5	Bill	26
4.6	Employees	28

4.7	Notification Alerts	30
4.8	Logs	31
4.9	Profile	31
4.10	Change Password	32
4.11	Forgot Password	33
4.12	Logout	35
5	Results and Discussion	36
5.1	Intermediate Results and Analysis	36
5.1.1	Inventory Management Module	36
5.1.2	Billing Module	37
5.1.3	Logs and Notifications	37
5.2	Final Results and Analysis	37
5.2.1	System Accuracy	37
5.2.2	Logging and Notifications	38
5.2.3	System Performance	38
5.3	Comparison with Existing Applications	38
5.3.1	Feature Comparison	38
5.3.2	Performance Comparison	38
5.4	Summary	39
6	Conclusion & Future Work	40
6.1	Conclusion	40
6.2	Lessons Learned from Project Management	41
6.2.1	Requirement Gathering and Understanding	41
6.2.2	Time Management	41
6.2.3	Collaboration and Communication	41
6.3	Future Work	41
6.3.1	Enhanced User Role Management	41
6.3.2	Advanced Analytics and Reporting	42
6.3.3	Mobile Application Integration	42
6.3.4	Integration with Payment Gateways	42
6.3.5	Data Security and Encryption	42
6.4	Summary	42

Chapter 1

Introduction

From recent times, Billing has been a part of day-to-day professional life of a shop owner. Along with Billing, keeping a track of the Inventory is also as important to keep the working of shop streamlined. In today's competitive business environment, efficient billing and inventory management are crucial for smooth operation of businesses. Managing these processes is time consuming and hectic. Any errors in these processes can lead to stock shortages, economic losses and customer dissatisfaction. Owners in the earlier times had to manage these both tasks manually which was a tough task. With the rise of computers and internet, applications have been developed for shopkeepers, helping them manage both these tasks effectively, either independently or together. With improvement in technology, these applications have developed a lot making the job of shop owner more and more easier.

Bill-n-Box is one such desktop application whose main functionality is managing inventory and billing. It will help the shop owners, streamline their workflow by creating bills and managing inventory along with various other functionalities. It is a one stop solution for billing and inventory management. Along with these functionalities, several other functionalities such as receiving notifications for low quantity and expiry date will help owner remain alert avoiding customer dissatisfaction. Also managing multiple employees in the same application will make shop owner's work a bit more easy. Providing a logs system will help the shop owner keep track of actions performed by employees on the application. Overall, it will help the shop owner have a bit less stressful professional life.

Chapter 2

Review

Inventory management and billing are critical components of any retail, whole-sale oriented business. With development in technology many applications have emerged which provide facilities to manage inventory and create bills efficiently. These solutions are typically designed for larger businesses, but often require significant training and customization. Many of the existing solution either lack flexibility or are complex for the users with minimal technical expertise. Also many existing systems do not provide other functionalities other than billing and inventory management, like low stock and nearing expiry date alerts, managing employees, etc. This part is also very crucial for any small scale business. This project seeks to address these gaps by developing a user friendly desktop application integrating inventory management and billing. Other important features like analysis, employee management and alerts for low stock and nearing expiry dates are also included. Thus this application will help the businesses, especially the smaller ones, to have a track of their inventory and create bills efficiently, also manage employees and keep their workflow under control, enhancing customer satisfaction and contributing to the overall growth of the business.

Chapter 3

Analysis and Design

3.1 Entity Relationship Diagram

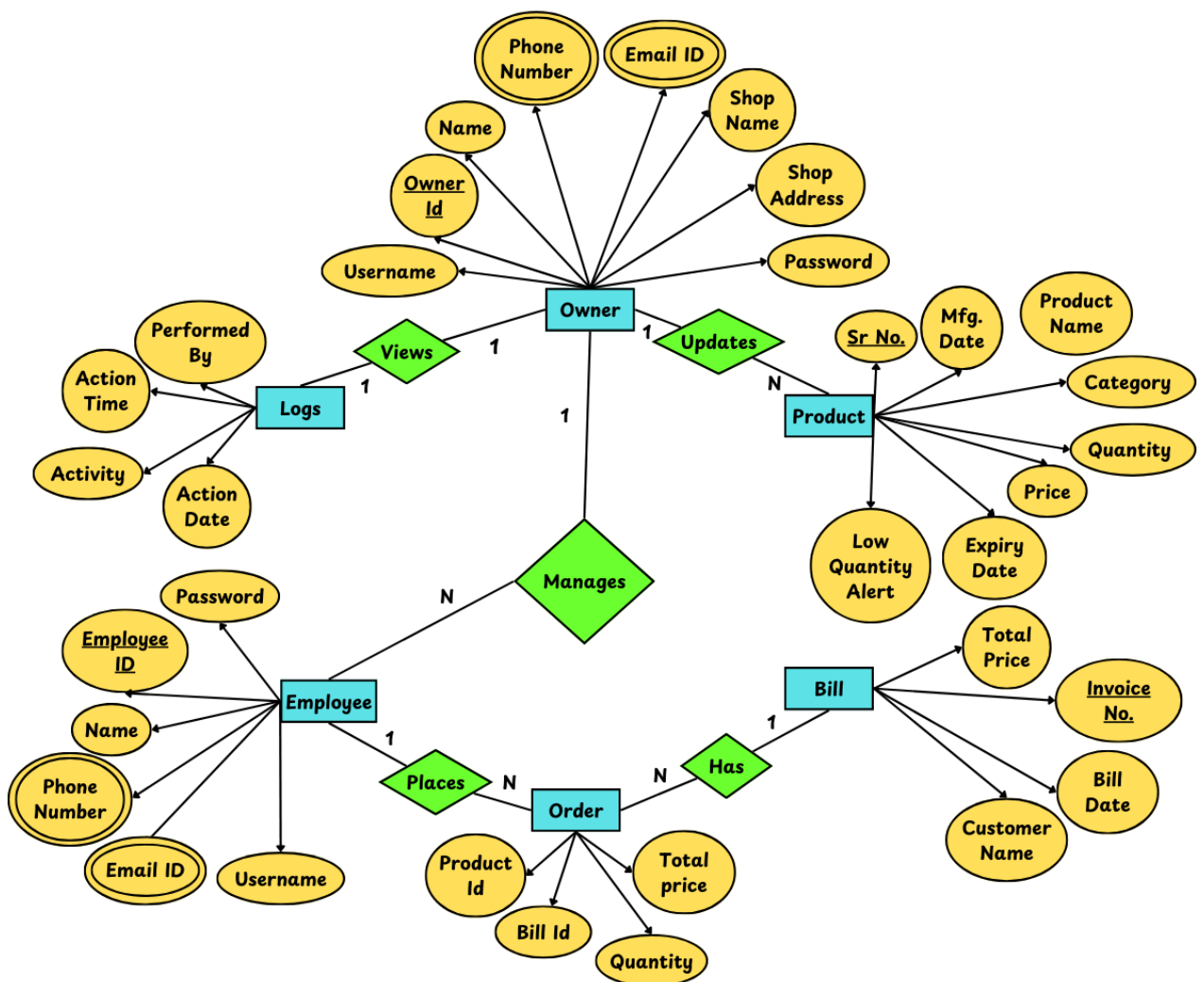


Figure 3.1: ER Diagram

The Entity-Relationship (ER) diagram represents a billing and inventory management system that defines relationships between entities: Owner, Product, Employee, Order, Bill, and Logs, each with specific attributes. Overall, the ER diagram efficiently shows how owners manage employees and products, orders link to bills, and logs ensure accountability within the system.

3.2 Database Design

3.2.1 Owner Table

	Field	Type	Null	Key	Default	Extra
►	OwnerID	int	NO	PRI	NULL	auto_increment
	Username	varchar(50)	YES	UNI	NULL	
	Password	varchar(50)	YES		NULL	
	Name	varchar(50)	YES		NULL	
	EmailID	varchar(50)	YES		NULL	
	PhoneNo	varchar(10)	YES		NULL	
	ShopName	varchar(50)	YES		NULL	
	ShopAddress	varchar(100)	YES		NULL	

Figure 3.2: Owner Table

The Owner entity manages the system and includes attributes like OwnerId, Name, Phone Number, and Shop Address. Each owner can manage multiple employees and products, creating a 1:N relationship.

3.2.2 Product Table

	Field	Type	Null	Key	Default	Extra
►	SrNo	int	NO	PRI	NULL	auto_increment
	ProductName	varchar(100)	NO		NULL	
	Category	varchar(50)	NO		NULL	
	Quantity	int	NO		0	
	Price	decimal(10,2)	NO		NULL	
	MfgDate	date	YES		NULL	
	ExpDate	date	YES		NULL	
	LowQuantityAlert	int	YES		1	
	OwnerID	int	YES		NULL	

Figure 3.3: Product Table

The Product entity tracks Sr No., Product Name, Category, Quantity, Price, and OwnerId. Products are linked to owners in a 1:N relationship and are central to inventory and billing.

3.2.3 Employee Table

	Field	Type	Null	Key	Default	Extra
►	EmpID	int	NO	PRI	<small>NULL</small>	auto_increment
	Username	varchar(50)	YES		<small>NULL</small>	
	Password	varchar(50)	YES		<small>NULL</small>	
	Name	varchar(50)	YES		<small>NULL</small>	
	EmailID	varchar(50)	YES		<small>NULL</small>	
	PhoneNo	varchar(10)	YES		<small>NULL</small>	
	OwnerID	int	YES	MUL	<small>NULL</small>	

Figure 3.4: Employee Table

The Employee entity holds details like Employee ID, Name, Phone Number, and Username, with employees reporting to a single owner. Employees can place orders on behalf of customers.

3.2.4 Order Table

	Field	Type	Null	Key	Default	Extra
►	OrderID	int	NO	PRI	<small>NULL</small>	auto_increment
	BillID	int	YES	MUL	<small>NULL</small>	
	SrNo	int	YES	MUL	<small>NULL</small>	
	ProductName	varchar(100)	YES		<small>NULL</small>	
	Quantity	int	NO		<small>NULL</small>	
	TotalPrice	decimal(10,2)	NO		<small>NULL</small>	
	OwnerID	int	YES		<small>NULL</small>	

Figure 3.5: Order Table

The Order entity ties products to billing, with attributes like ProductId, Quantity, and BillId, forming a link between sales and bills.

3.2.5 Bill Table

	Field	Type	Null	Key	Default	Extra
►	BillID	int	NO	PRI	<small>NULL</small>	auto_increment
	EmpId	int	YES	MUL	<small>NULL</small>	
	OwnerID	int	YES	MUL	<small>NULL</small>	
	Time	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
	CustomerName	varchar(255)	YES		<small>NULL</small>	
	Amount	decimal(10,2)	YES		<small>NULL</small>	
	InvoiceNo	varchar(10)	YES		<small>NULL</small>	

Figure 3.6: Bill Table

The Bill entity records transactions, including Invoice No., Bill Date, and Total Price. Bills link to multiple orders in a 1:N relationship.

3.2.6 Logs Table

	Field	Type	Null	Key	Default	Extra
►	id	int	NO	PRI	HULL	auto_increment
	date	date	NO		HULL	
	time	time	NO		HULL	
	User	varchar(25)	YES		HULL	
	activity	text	NO		HULL	
	OwnerID	int	YES		HULL	

Figure 3.7: Logs Table

Finally, the Logs entity tracks actions such as updates or transactions, recording who performed the activity, ensuring transparency.

3.3 Triggers

3.3.1 After Owner Insert

```
-- Trigger for logging after an owner insert
DELIMITER //
CREATE TRIGGER after_owner_insert
AFTER INSERT ON Owner
FOR EACH ROW
BEGIN
    INSERT INTO logs (date, time, User, activity, OwnerID)
    VALUES (CURRENT_DATE, CURRENT_TIME,
            NEW.Username,
            CONCAT('Owner registered: ', NEW.Username),
            NEW.OwnerID);
END; //
DELIMITER ;
```

Figure 3.8: Trigger After Owner Insert

This trigger is used to update the logs table when any insertion is done in the Owner Table.

3.3.2 After Owner Update

```
-- Trigger for logging after an owner update
DELIMITER //
CREATE TRIGGER after_owner_update
AFTER UPDATE ON Owner
FOR EACH ROW
BEGIN
    INSERT INTO logs (date, time, User, activity, OwnerID)
    VALUES (CURRENT_DATE, CURRENT_TIME,
            NEW.Username,
            CONCAT('Owner with ID ', NEW.Username, ' updated'),
            NEW.OwnerID);
END; //
DELIMITER ;
```

Figure 3.9: Trigger After Owner Update

This trigger is used to update the logs table when any update is done in the Owner Table.

3.3.3 After Owner Delete

```
-- Trigger for logging after an owner delete
DELIMITER //
CREATE TRIGGER after_owner_delete
AFTER DELETE ON Owner
FOR EACH ROW
BEGIN
    INSERT INTO logs (date, time, User, activity, OwnerID)
    VALUES (CURRENT_DATE, CURRENT_TIME,
            OLD.Username,
            CONCAT('Owner with ID ', OLD.Username, ' deleted'),
            OLD.OwnerID);
END; //
DELIMITER ;
```

Figure 3.10: Trigger After Owner Delete

This trigger is used to update the logs table when any deletion is done in the Owner Table.

3.3.4 After Employee Insert

```
-- Trigger for logging after an employee insert
delimiter ..
CREATE TRIGGER after_employee_insert
AFTER INSERT ON Employee
FOR EACH ROW
BEGIN
    DECLARE ownerUsername VARCHAR(255);

    -- Fetch the Username of the Owner based on the OwnerID of the new Employee
    SELECT Username INTO ownerUsername FROM Owner WHERE OwnerID = NEW.OwnerID;

    -- Ensure that NEW.Username is not NULL
    IF NEW.Username IS NOT NULL THEN
        -- Insert into logs with the fetched ownerUsername
        INSERT INTO logs (date, time, user, activity, OwnerID)
        VALUES (CURRENT_DATE, CURRENT_TIME, ownerUsername,
            CONCAT('Employee created: ', NEW.Username),
            NEW.OwnerID);
    ELSE
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Username cannot be NULL in logs for employee creation';
    END IF;
END..
delimiter ;
```

Figure 3.11: Trigger After Employee Insert

This trigger is used to update the logs table when any insertion is done in the Employee Table.

3.3.5 After Employee Update

```
-- Trigger for logging after an employee update
DELIMITER //
CREATE TRIGGER after_employee_update
AFTER UPDATE ON Employee
FOR EACH ROW
BEGIN
    INSERT INTO logs (date, time, User, activity, OwnerID)
    VALUES (CURRENT_DATE, CURRENT_TIME,
        (SELECT Username FROM Owner WHERE OwnerID = NEW.OwnerID),
        CONCAT('Employee with Username ', NEW.Username, ' updated'),
        NEW.OwnerID);
END; //
DELIMITER ;
```

Figure 3.12: Trigger After Employee Update

This trigger is used to update the logs table when any update is done in the Employee Table.

3.3.6 After Employee Delete

```
-- Trigger for logging after an employee delete
DELIMITER //
CREATE TRIGGER after_employee_delete
AFTER DELETE ON Employee
FOR EACH ROW
BEGIN
    INSERT INTO logs (date, time, User, activity, OwnerID)
    VALUES (CURRENT_DATE, CURRENT_TIME,
            (SELECT Username FROM Owner WHERE OwnerID = OLD.OwnerID),
            CONCAT('Employee with Username ', OLD.Username, ' deleted'),
            OLD.OwnerID);
END; //
DEFINER :

```

Figure 3.13: Trigger After Employee Delete

This trigger is used to update the logs table when any deletion is done in the Employee Table.

3.4 Database Hosting

Database has been hosted on aiven cloud.

<https://mysql-service-for-billnbox-bill-n-box.i.aivencloud.com>

Chapter 4

Implementation

4.1 Login

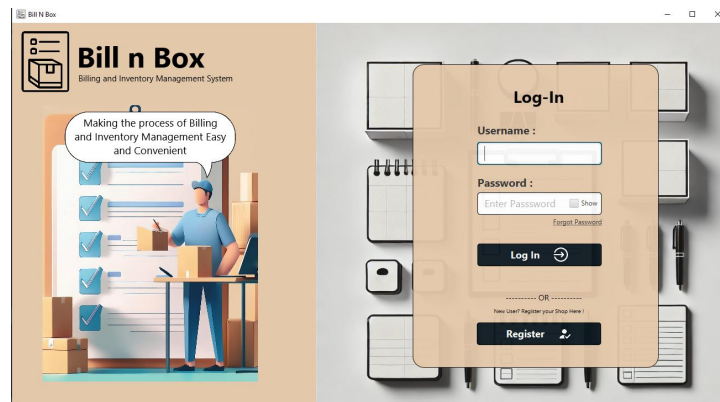


Figure 4.1: Login page

The main function of this page is to accept credentials from the existing user and allow access to the user if the credentials are valid. The page consists of two text fields for 'username' and 'password' each. There is a button named 'Log-In'. After entering valid credentials and clicking on the Log-In button, this application will grant access to the user and direct the user to the dashboard. Depending on the credentials entered, the application whether the user is an owner or an employee and will direct the user to the respective dashboard. If the credentials entered by the user do not match, a message will be displayed to the user saying 'Invalid Username or Password'. There is also a label named 'Forgot Password?'. The user will be directed to the forgot password page after clicking on this label.

There is a button named 'Register' which on clicked will allow registration for a new owner. After clicking on this button, the user will be directed to the

registration page.

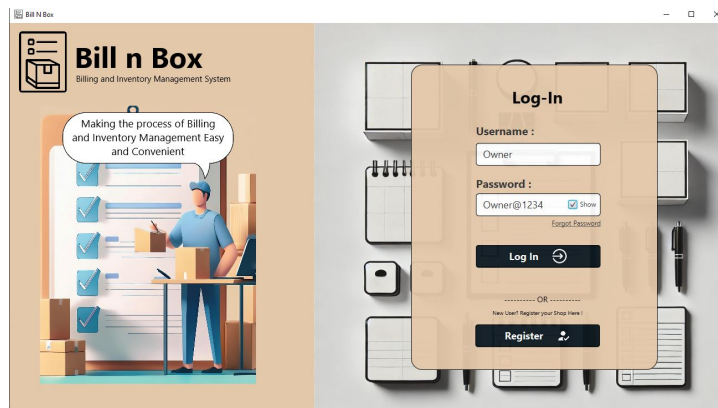


Figure 4.2: Login page

There is a checkbox named 'Show' at the end of the password field. When the checkbox is unchecked, the password is displayed in dotted format and when the checkbox is checked, the password is displayed in text format.

4.2 Registration

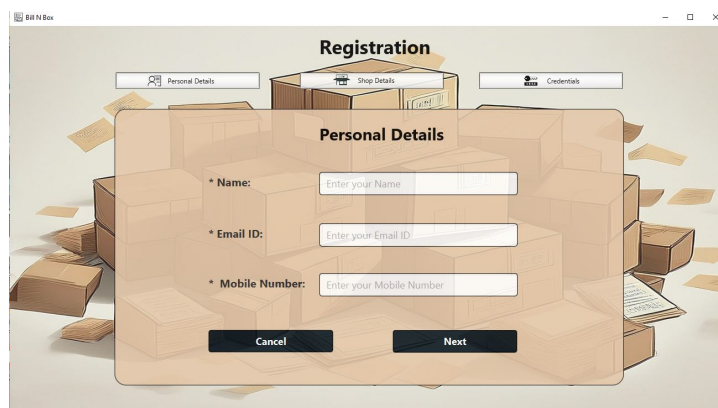


Figure 4.3: Registration page

After clicking on the 'Register' button from the Log-In page, the user gets directed to this page. This allows the new users to get access of the application. This page consists of three text fields namely 'Name', 'Email-Id' and 'Phone Number'. The Phone number should consist of exactly ten digits. If the phone number entered by the user fail to satisfy this condition, a message will be displayed to the user saying 'Invalid Mobile Number '. The Email-Id should follow the format where it ends with '@gmail.com'. If Email-Id entered

by the user fail to follow this format, a message will be displayed to the user saying 'Invalid Email-Id'. It also has two buttons namely 'Cancel' and 'Next'. The Cancel button directs the user back to the login page whereas after completing these fields and clicking on the Next button, the user is directed to the next page of registration form. At the top, three horizontal bars are present indicating three stages of registration form. The name of three horizontal bars are 'Personal Details', 'Shop Details' and 'Credentials'. Currently, the bar named Personal Details is blinking indicating that currently personal details are being accepted.

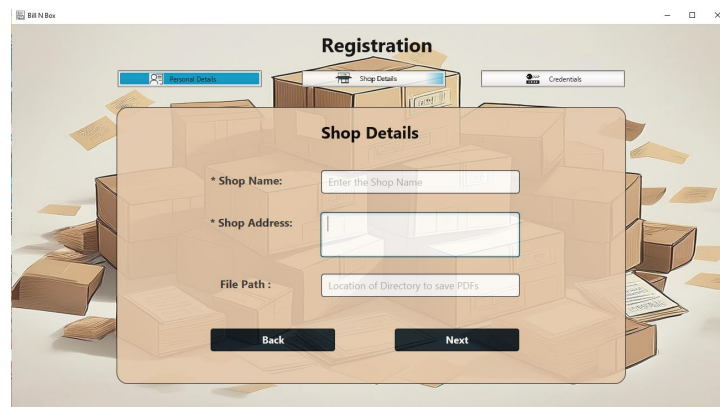
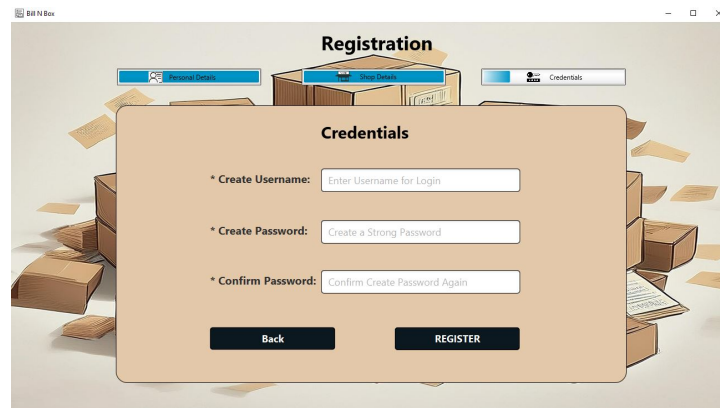


Figure 4.4: Registration page

After clicking on the Next button from the last registration page, user gets directed to this page. This page consists of three text fields named 'Shop Name', 'Shop Address' and 'File Path'. In file path, the path where the PDF of bill is to saved is accepted. The horizontal bar on the top for Personal details is completely highlighted indicating that this part of the form has been completed whereas the bar named 'Shop Details' is blinking which indicates that the current form accepts shop details. This page consists of two buttons named 'Back' and 'Next'. Clicking on Back button, the user gets directed to the last registration page whereas after filling the required information and clicking on the Next button directs the user to the next registration page.

After clicking on the Next button from the last registration page, user gets directed this page. This page contains two text fields named 'Username' and 'Password'. These fields allow user to create an username and a password for their application account. When the user enters any username, it checked that whether that username is already registered for other user. If this is the case, a



The screenshot shows a web application window titled "Registration". At the top, there are three horizontal tabs: "Personal Details", "Shop Details", and "Credentials". The "Credentials" tab is currently active and highlighted. Below the tabs, the "Credentials" section contains three input fields:

- * Create Username: Enter Username for Login
- * Create Password: Create a Strong Password
- * Confirm Password: Confirm Create Password Again

 At the bottom of the form, there are two buttons: "Back" and "REGISTER". The background of the page features a 3D illustration of cardboard boxes and scattered papers.

Figure 4.5: Registration page

message will be displayed to the user saying 'Username not available'. The entered password should have minimum eight characters. If the password entered by the user does not follow this condition, a message will be displayed to the user saying 'Password should consist of atleast 8 characters'. Also the password should include atleast one Uppercase letter, Digit and Special Character. If the user created password fails to fulfill this condition, a message will be displayed to the user saying 'Password should consist of atleast one Uppercase Alphabet, Digit and Special Character'. There are buttons on this page named 'Back' and 'REGISTER'. The horizontal bars at top show first two bars for Personal Details and Shop Details completely highlighted whereas the bar for Current Details is blinking which indicates that first two parts of the registration form have been completed whereas the third part is ongoing. The Back button directs the user to the last registration page. After filling the required information and clicking on the Register button, the user gets successfully registered as an owner in the application.

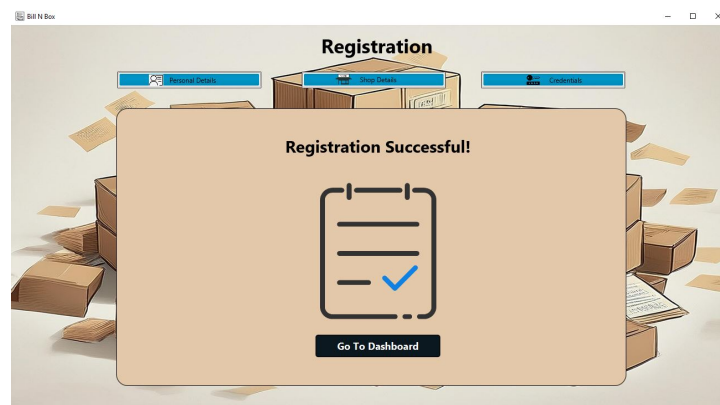


Figure 4.6: Registration Successful page

After clicking on the REGISTER button from the last page, the user gets directed to this page. This page contains a message showing Registration Successful and a button named 'Go To Dashboard' which on clicking directs the user to the dashboard. The three horizontal bars at the top are completely highlighted indicating that the registration form has been completed.

4.3 Dashboard

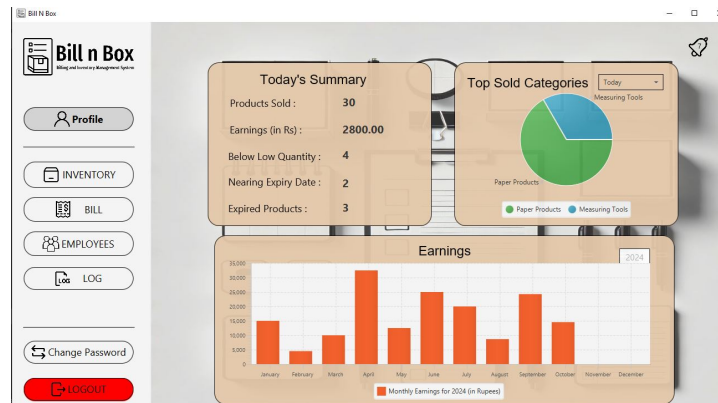


Figure 4.7: Owner Dashboard

After a successful login or registration by the user, the owner gets directed to the dashboard. The dashboard shows various analytics such as 'Today's Summary', a pie chart showing 'Top Sold' products, a graph showing 'Earnings'. A drop down list for Top Sold products is provided where user can choose from the drop down list and see daily, weekly, monthly or yearly top sold products. A text field for changing the year is available for the 'Earnings' graph. The dashboard consists of buttons named 'Profile', 'Inventory', 'Bill', 'Employees', 'Log', 'Notifications', 'Change Password' and 'Logout' directing to the respective pages.

After a successful login, employee is directed to the dashboard. The dashboard shows various analytics such as 'Today's Summary', a pie chart showing 'Top Sold' products, a graph showing 'Earnings'. A drop down list for Top Sold Products is provided where user can choose from the drop down list and see daily, weekly, monthly or yearly top sold products. A text field for changing the year is available for the 'Earnings' graph. This page consists of 'Profile', 'Inventory', 'Bill', 'Notifications', 'Change Password' and 'Logout' button directing to the respective pages.

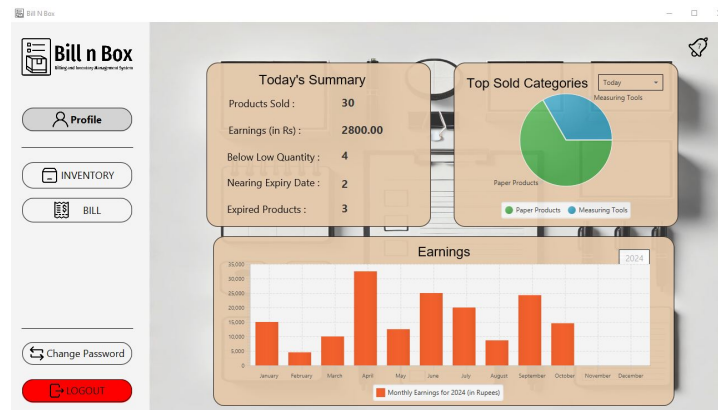


Figure 4.8: Employee Dashboard

4.4 Inventory

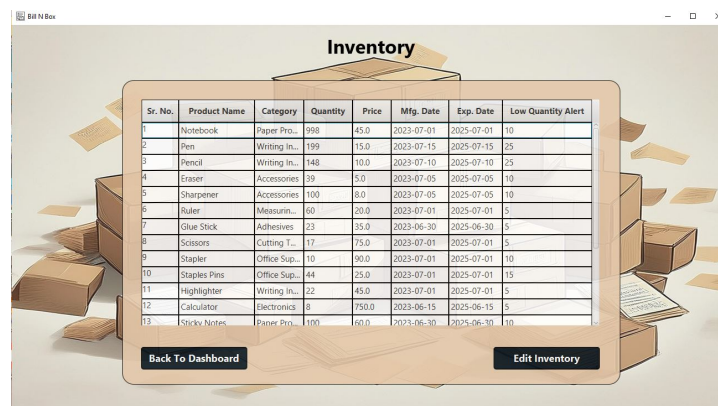


Figure 4.9: Inventory

The user gets directed to the inventory page after clicking on the button named 'Inventory' on the dashboard. This page consists of a table which displays various aspects of the inventory. The table consists of columns namely 'Sr. No.' which displays serial number of the product, 'Product Name' which displays name of the product, 'Category' which displays the category to which the product belongs, 'Quantity' which displays the quantity of product available in the stock, 'Price' which displays the price of the product per piece, 'Mfg. Date' which displays the manufacturing date of the product, 'Exp. Date' which displays the expiry date of the product, 'Low Quantity Alert' which displays at what quantity of a product will the application give a low quantity alert. This allows the user keep a track of inventory and check if quantity of any product is low or if any product is nearing its Expiry Date. The page consists of two buttons which are 'Back To Dashboard' and 'Edit Inventory'. The Back To Dashboard

button directs the user back to dashboard while the Edit Inventory button allows the user to make changes in the inventory.

Inventory

Sr. No.	Product Name	Category	Quantity	Price	Mfg. Date	Exp. Date	Low Quantity Alert
1	Notebook	Paper Pro...	998	45.0	2023-07-01	2025-07-01	10
2	Pen	Writing In...	199	15.0	2023-07-15	2025-07-15	25
3	Pencil	Writing In...	148	10.0	2023-07-10	2025-07-10	25
4	Eraser	Accessories	39	5.0	2023-07-05	2025-07-05	10
5	Sharpener	Accessories	100	8.0	2023-07-05	2025-07-05	10
6	Ruler	Measur...	60	20.0	2023-07-01	2025-07-01	5
7	Glue Stick	Adhesives	23	35.0	2023-06-30	2025-06-30	5
8	Scissors	Cutting T...	17	75.0	2023-07-01	2025-07-01	5
9	Stapler	Office Sup...	10	90.0	2023-07-01	2025-07-01	10
10	Staples Pins	Office Sup...	44	25.0	2023-07-01	2025-07-01	15
11	Highlighter	Writing In...	22	45.0	2023-07-01	2025-07-01	5
12	Calculator	Electronics	8	750.0	2023-06-15	2025-06-15	5

Figure 4.10: Edit Inventory

This functionality is available only for owner. After the owner clicks on Edit Inventory button, the table becomes editable allowing user to make changes in the inventory. This page consists of four buttons named 'Add New Product', 'Delete', 'Discard Changes' and 'Save Changes'. Add New Product button directs the owner To the Add New Product Page. Delete button allows owner to delete a product from the inventory table. The delete button is initially disabled, it only becomes clickable when any product has been selected. Save changes button allows the user to save the changes they made into the inventory table. Discard button will reset the unsaved changes in the inventory.

New Stock Entry

Product Name	Category	Quantity	Price	Mfg. Date	Exp. Date	Low Quantity Alert
No content in table						

Product Name :
Mfg. Date :

Category :
Exp. Date :

Quantity :
Low Quantity Alert :

Price :

Figure 4.11: Add New Product

After clicking on the Add New Product button on the inventory page, user is directed to this page. This page consists text fields to accept 'Product Name', 'Category', 'Quantity', 'Price', 'Mfg. Date', 'Exp. Date', 'Low Quantity

Alert'. After entering these fields user will click on the 'Add' button. After clicking on the Add button, its data will be displayed in the table on the top of the same page. The table consists of columns such as 'Product Name' which displays the Name of the product, 'Category' displaying category of the added product, 'Quantity' displaying quantity of the product, 'Price' displaying the price of product, 'Mfg. Date' and 'Exp. Date' displaying the manufacturing and expiry date respectively and 'Low Quantity Alert' displaying at what minimum quantity will the user get notification for the newly added product. If any invalid input is entered, error is displayed to the user saying "Invalid Input".

4.5 Bill

Figure 4.12: Bill

The Bill page is loaded when user clicks on the 'Bill' button on the dashboard. This page allows the Owner and Employee to create bills. This page consists of text field at top which accepts 'Name of the Customer' and 'Date'. In the Date field, current date is set by default. Below there is a table consisting of columns named 'Product Name', 'Quantity', 'Price per Quantity', 'Total Price'. The table is initially empty. Below there are four text fields namely 'Product Name', 'Quantity', 'Price' and 'Grand Total'. When user starts entering product name, he/she gets suggestion of products whose names similar to what the user has entered. Depending on the name of the product entered, the price is automatically filled by fetching it from the inventory. After user enters the quantity of the product, the grand total gets displayed. After filling this information and clicking on the 'Add to Bill' button, a row gets added in the column in the above table. User can add more products if wanted. Entered product can be removed from the table by selecting that particular row from the table and clicking the

'Delete' button. After selecting all the products wanted, user can click on the 'Generate Bill' button. If any invalid input is entered, error is displayed to the user saying "Invalid Input". If user wants to discard the bill, he/she can click on the 'Discard Bill' button. After clicking on Discard Bill button the bill gets cancelled and user is directed back to the dashboard.

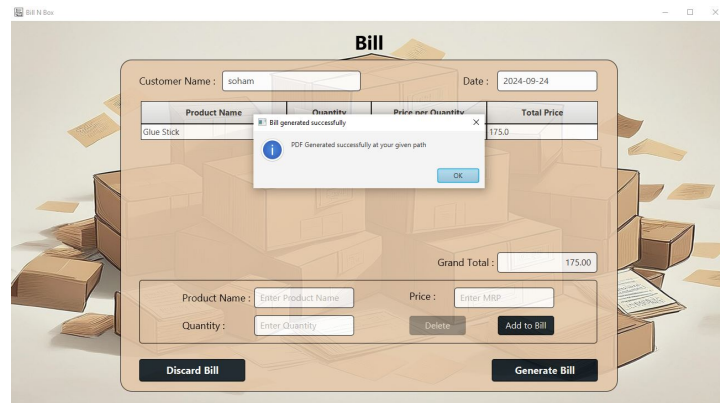


Figure 4.13: Bill created alert

After clicking on the Generate Bill button, an alert comes telling the user that bill is created successfully and PDF of that bill generated on the path given by the user. After clicking the OK button, user is redirected to the dashboard.

Alice's Grocery Shop
 123 Stationery Lane, Suite 4B, Kurla - 400079
 Contact: 9998453218 Email: alicethompson@groceryshop.com

Bill Receipt

Bill ID: 149 Date: 09/10/2024 19:14:12

Customer Name: Aaryan
Contact Details:

Product Name	Quantity	Price per Quantity	Total Price
Wheat 1kg	1	100.00	100.00
Basmati Rice 1kg	1	90.00	90.00
Toor Dal 1kg	1	100.00	100.00
Lentils 500g	2	75.00	150.00
Parle-G Biscuits 500g	5	50.00	250.00

Grand Total: Rs. 690.00

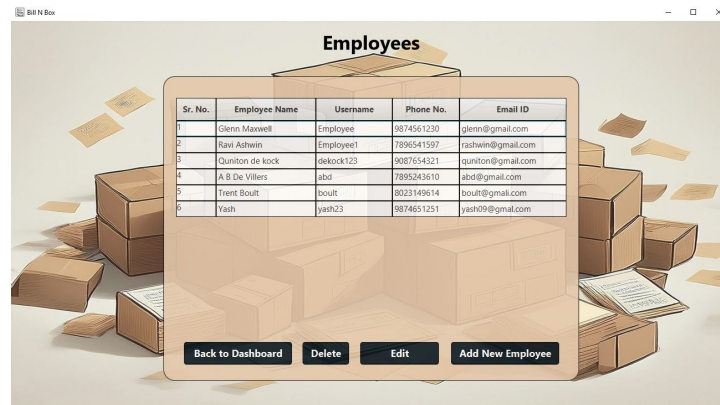
Thank you for your purchase!
Please visit us again.

Figure 4.14: Bill

Bill is generated and is saved to the path given by the user in PDF format. Bill consists of Shop name, Shop address, Contact number and Email-Id of

the shop owner. It consists of the Bill ID, Date , customer Name. A table consisting of columns such as 'Product Name', 'Quantity', 'Price per Quantity', 'Total Price'. This table contains information of the products purchased by the customer. Grand Total is displayed below the table.

4.6 Employees



Sr. No.	Employee Name	Username	Phone No.	Email ID
1	Glenn Maxwell	Employee	9874561230	glenn@gmail.com
2	Ravi Ashwin	Employee1	7896541597	rashwin@gmail.com
3	Quinton de kock	decock123	9087654321	quinton@gmail.com
4	A B De Villiers	abd	7895243610	abd@gmail.com
5	Trent Boult	boult	8023149614	boult@gmail.com
6	Yash	yash23	9874651251	yash09@gmail.com

Figure 4.15: Employees

This functionality is available only for the owner. After clicking on the Employees button on the dashboard, the owner will be directed to the Employees page where the owner will be able to see to the data of the employees under him/her. The table consists of columns such as 'Sr. No.' displaying serial number, 'Employee Name' displaying name of employees, 'Username' displaying username of employees, 'Phone No.' which displays the phone number of employees, 'Email-Id' displaying the Email-Id of employees. This page has four buttons. 'Back To Dashboard' button takes you back to the dashboard. 'Delete' button allows the owner to delete and employee from the table. 'Edit' button allows the owner to edit the data of any employee and save it. 'Add New Employee' button directs the owner to the Add New Employee page.

This page consists of the same table that was on the employees page but now owner can make changes in it as the table has become editable. After making changes, owner can save the changes by clicking on the 'Save' button. This page also consists of the other three buttons that were on the Employees page namely 'Back To Dashboard', 'Delete' and 'Add New Product' having the same functionality. After making changes and clicking on the Save button, the owner is redirected to the Employees page.

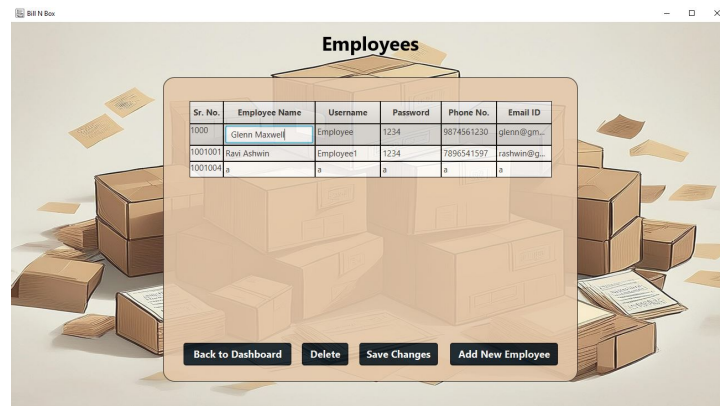


Figure 4.16: Edit Employees

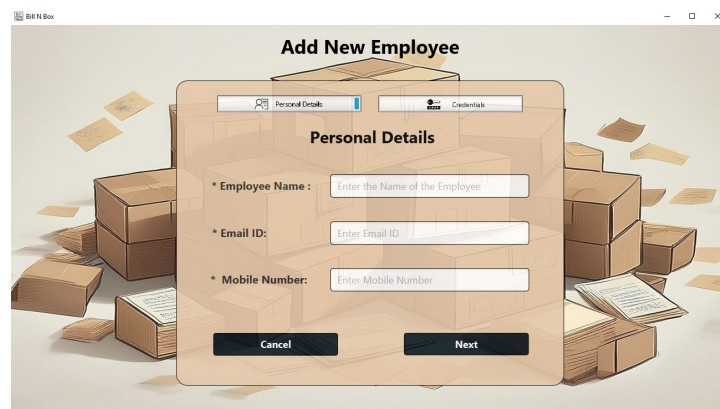


Figure 4.17: Add New Employee

Owner is directed to this page after clicking on the Add New Employee button on the Employees page. This page consists of three text fields for accepting 'Name', 'Email-Id' and 'Phone Number' of the employee. There are two buttons namely 'Back' and 'Next'. The Back button takes the owner back to the Employees page. After filling all the information in the given text fields and on the Next button, owner is directed to the next page. There are two horizontal bars at the top named 'Personal Details' and 'Credentials'. The bar named Personal Details is blinking indicating that currently employee's personal details are being filled. If user enters any invalid input in mobile number like more or less than 10 digits or any other character other than digits, an error message is displayed to the user saying 'Invalid Mobile Number'. Also if the entered Email-Id does not satisfy the required pattern, an error message saying 'Invalid Email-Id' is displayed to the user.

Owner is directed to this page after clicking Next button on the last page. This page has two fields for accepting 'Username' and 'Password'. If the username entered by the user is already taken by some other user, a message saying

Figure 4.18: Add New Employee

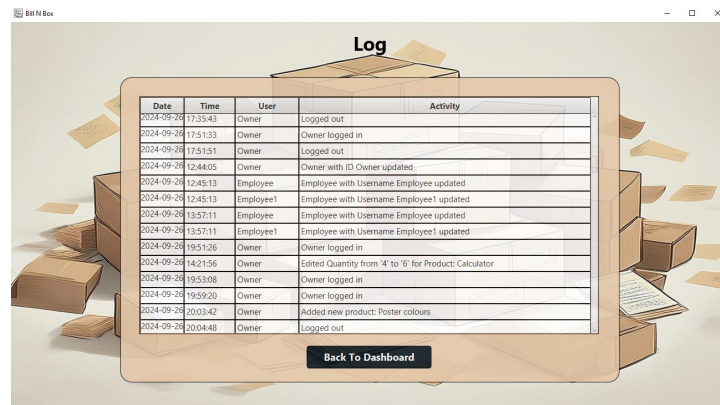
'Username Not Available' is displayed to the user. If entered password does not satisfy the criteria of minimum eight characters, a error saying 'Password should consist of atleast 8 characters' is displayed to the user. Also password should consist of atleast one digit, special character and Uppercase alphabet. If password does not satisfy this criteria , a message saying 'Password should consist of atleast one Uppercase Alphabet, Digit and Special Character'. There are two buttons. 'Back' button directs the owner back to the last page. After filling the information and clicking on the 'Add Employee' button, the employee gets added in the employee table and owner is directed to the Employees page. Out of the two horizontal bars at the top, the bar named 'Personal Details' is completely highlighted indicating that that part of the form is completed while the bar named 'Credentials' is blinking which indicates that Credentials for the Employee are being entered.

4.7 Notification Alerts

Figure 4.19: Notifications Page

After clicking on the 'Notifications' button from the dashboard, the user will be directed to the Notifications page. This page shows alert message to the user for low quantity, nearing expiry date and expired products. It consists of a button named 'Back to Dashboard' which directs the user back to the dashboard.

4.8 Logs



Date	Time	User	Activity
2024-09-26	17:35:43	Owner	Logged out
2024-09-26	17:51:33	Owner	Owner logged in
2024-09-26	17:51:51	Owner	Logged out
2024-09-26	12:44:05	Owner	Owner with ID Owner updated
2024-09-26	12:45:13	Employee	Employee with Username Employee1 updated
2024-09-26	12:45:13	Employee1	Employee with Username Employee1 updated
2024-09-26	13:57:11	Employee	Employee with Username Employee1 updated
2024-09-26	13:57:11	Employee1	Employee with Username Employee1 updated
2024-09-26	19:51:26	Owner	Owner logged in
2024-09-26	14:21:56	Owner	Edited Quantity from '4' to '6' for Product: Calculator
2024-09-26	19:53:08	Owner	Owner logged in
2024-09-26	19:59:20	Owner	Owner logged in
2024-09-26	20:03:42	Owner	Added new product: Poster colours
2024-09-26	20:04:48	Owner	Logged out

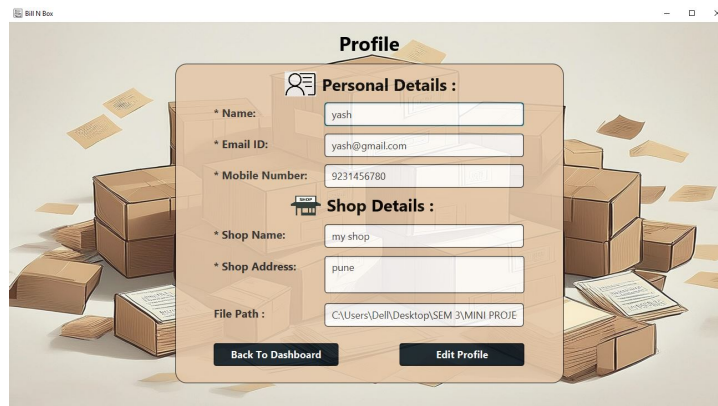
Back To Dashboard

Figure 4.20: Logs

This functionality is available only for the owner. Owner gets directed to this page after clicking on the logs button from dashboard. This page consists of a table consisting of columns named 'Date' which displays the date on which a particular activity was performed, 'Time' displaying the time on which a particular was performed, 'User' displaying the person by which the activity was performed and 'Activity' which displays what activity was performed. There is a button named 'Back To Dashboard' which directs the owner back to the dashboard.

4.9 Profile

Profile page consists of text fields namely 'Name' displaying name of the user, 'Email-Id' and 'Mobile Number' displaying email-id and phone number of the user, 'Shop Name' and 'Shop Address' which displays the name and address of the shop and 'File Path' which displays the path at which the PDF of created bills are saved. Initially the existing values are displayed. After clicking on 'Back To Dashboard' button the user is directed to the dashboard. The 'Edit Profile' button makes these text fields editable.



Profile

Personal Details :

* Name:

* Email ID:

* Mobile Number:

Shop Details :

* Shop Name:

* Shop Address:

File Path :

[Back To Dashboard](#) [Edit Profile](#)

Figure 4.21: Profile



Profile

Personal Details :

* Name:

* Email ID:

* Mobile Number:

Shop Details :

* Shop Name:

* Shop Address:

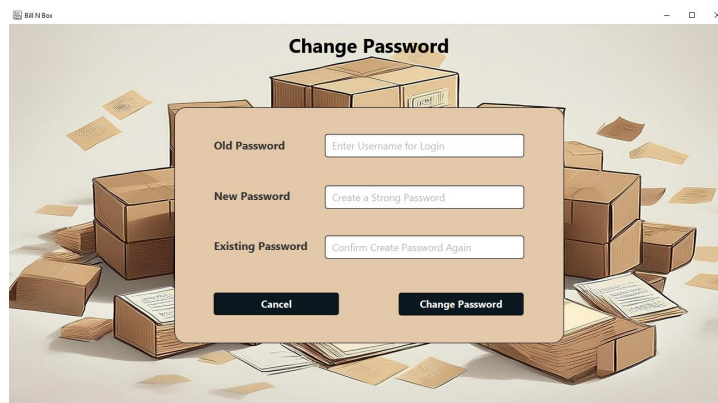
File Path :

[Back To Dashboard](#) [Save Changes](#)

Figure 4.22: Edit Profile

After clicking the 'Edit Profile' button the text fields become editable. After making the changes, user can click on the 'Save Changes' button to make the changes permanent. The 'Back To Dashboard' button redirects the user back to dashboard.

4.10 Change Password



Change Password

Old Password

New Password

Existing Password

[Cancel](#) [Change Password](#)

Figure 4.23: Change Password

After the user clicks on the 'Change Password' button on the Dashboard, the user is directed to this page. This page consists of three text fields named 'Current Password' where user should enter the current password, 'New Password' where user should enter the new password and 'Confirm Password' where user should re-enter the new password. After filling these fields and clicking on the 'Change Password' the password is changed and user is directed to the next page. 'Cancel' button directs the user back to the dashboard.

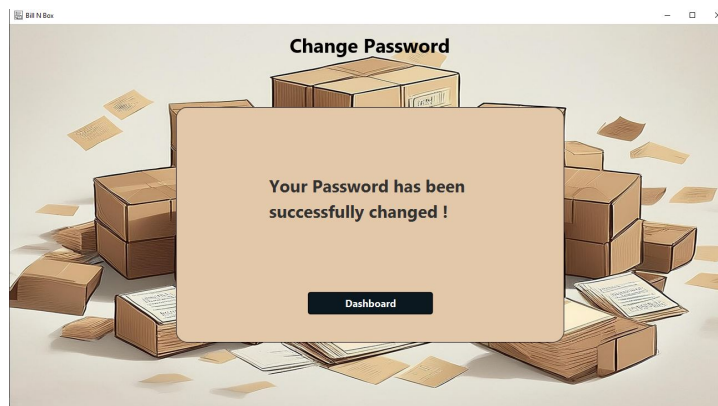


Figure 4.24: Password Change Successful

This page displays a message telling that password has been changed successfully. 'Dashboard' button redirects the user back to dashboard.

4.11 Forgot Password

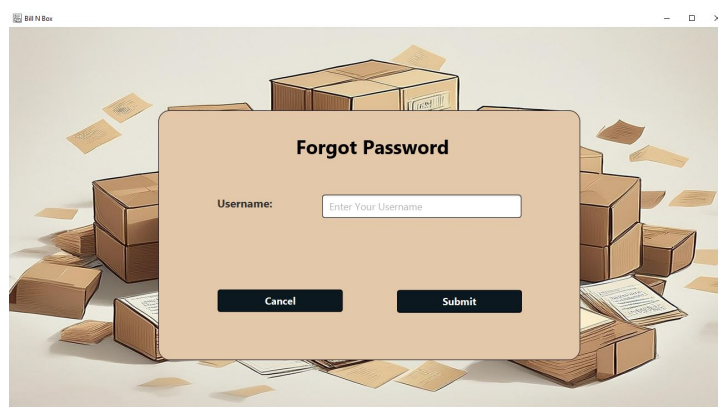


Figure 4.25: Forgot Password

This page is loaded when the user clicks on the 'Forgot Password' label from the Login page. This page helps the user to get access of the application in

case he/she is not able to recollect their password. This page consists of a text field to accept username from the user. There are two buttons on this page namely 'Back' which directs the user back to the Login page and 'Submit' which generates a mail and sends it on the email which is registered with the username that user has entered. A page giving message conveying the same is also displayed to the user when the user clicks on Submit button. The page displaying the button also consists of a button named 'Login Page' which directs the user back to the Login Page.

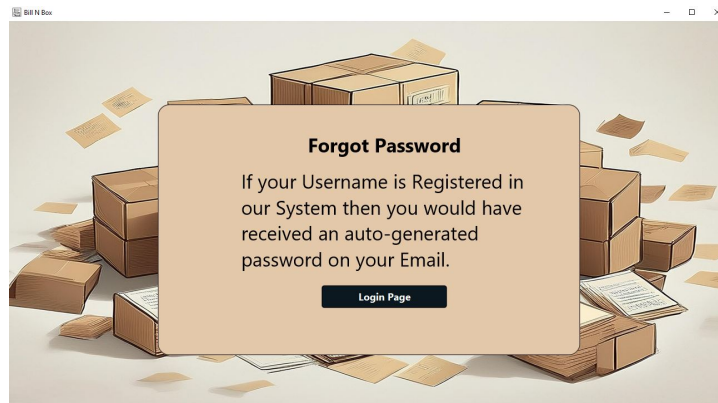


Figure 4.26: Forgot Password

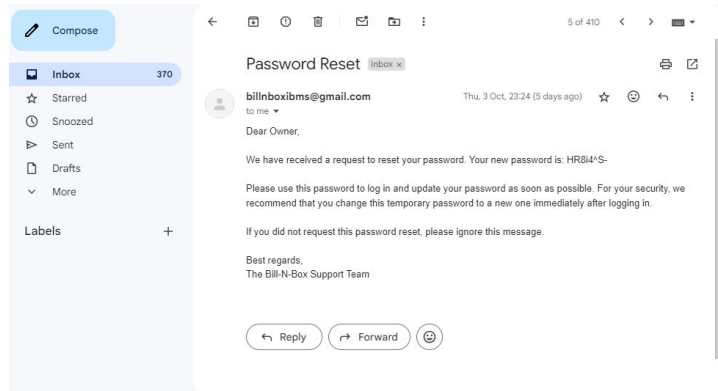


Figure 4.27: Mail received for Forgot Password

This email is sent to the user after user clicks on the Submit button from the Forgot Password page. This mail consists of a new password which is generated in a random manner. Using this password, the user can login into his/her application account. The mail is sent with a title 'Password Reset'. The Password is sent with a formal message for the user.

4.12 Logout

After the user clicks on the 'LOGOUT' button on the dashboard, user is directed this page. This page consists of a message asking the user whether he/she wants to logout. After clicking 'NO' button, the user is redirected back to dashboard.'YES' button directs the user to the Login page after a successful logout.

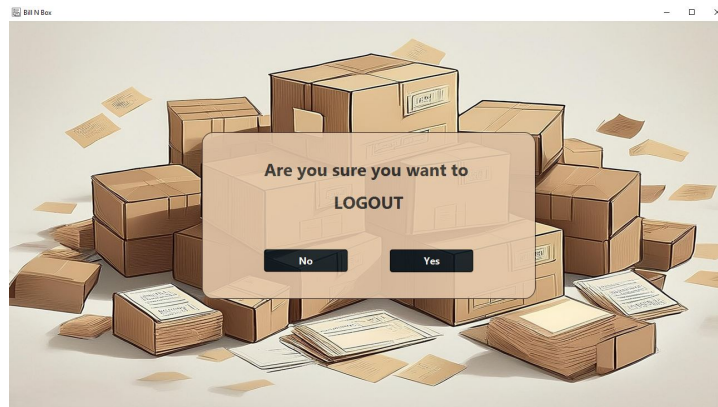


Figure 4.28: Logout Confirmation

Chapter 5

Results and Discussion

This chapter presents the intermediate and final results of the developed billing and inventory management system, along with their analysis. Additionally, a comparison between the developed application and existing solutions in the market is discussed.

5.1 Intermediate Results and Analysis

This section outlines the intermediate results obtained during the implementation and testing of various modules, including the inventory management system, billing system, logs, and notifications.

5.1.1 Inventory Management Module

The inventory management module was designed to allow shop owners to manage their products effectively. The system enabled the owners to add, update, and delete products from the inventory.

During the testing phase, the system correctly maintained the product inventory and successfully implemented the low quantity alert feature. Each product added or updated was associated with the respective owner using the ‘OwnerID’. This ensured that each owner could only access and manage products belonging to their shop.

Challenges: One challenge was ensuring the proper logging of product changes (addition, update, deletion) in the ‘logs’ table, which tracks all modifications made to the inventory.

5.1.2 Billing Module

The billing module was responsible for generating and storing bills created by both owners and employees. Each bill was linked to the respective owner using the 'OwnerID', ensuring data integrity.

Intermediate Results: The billing system functioned smoothly, with all bills being correctly generated and saved in the 'Bill' table, containing information such as 'InvoiceNo', 'BillDate', 'CustomerName', and 'TotalPrice'. The session management was implemented through the 'SessionManager' class, which ensured that each user, based on their role, had appropriate access to data related to their shop.

Challenges: Initial challenges involved ensuring that employees could only create bills and not modify inventory data, which was resolved through role-based access control in the system.

5.1.3 Logs and Notifications

The logging system was implemented to track user activities, such as login, logout, product updates, bill creation, and employee management. The system stored logs in the 'logs' table with details like date, time, user, activity, and owner information.

A notification system was also introduced, where a bell icon on the dashboard displayed a small circle indicating the number of pending notifications. This feature was tested and verified to update dynamically as events occurred.

Challenges: One challenge was ensuring real-time updates to the notification count, which was successfully achieved using database triggers and UI updates.

5.2 Final Results and Analysis

The final results of the system after complete development and testing are discussed here, focusing on the accuracy, performance, and overall system efficiency.

5.2.1 System Accuracy

The system accurately managed the inventory and billing processes, ensuring that owners could modify the 'Product' and 'Bill' tables, while employees were restricted to bill creation. The implementation of the 'OwnerID' and

‘OwnerUsername’ fields ensured that the data was properly segregated between owners and employees.

5.2.2 Logging and Notifications

The logging system provided a detailed record of all actions performed by users, including inventory updates, bill creation, and user activity. The notification system was effective in displaying real-time alerts on the dashboard, enhancing the overall user experience.

5.2.3 System Performance

The system demonstrated high performance in terms of speed and resource utilization. The use of MySQL for database management and efficient querying allowed the system to handle large datasets with minimal latency.

Challenges: Ensuring seamless performance across different user roles required careful management of database queries and session states, which was achieved through role-based access control and the ‘SessionManager’ class.

5.3 Comparison with Existing Applications

This section compares the developed system with other existing billing and inventory management solutions in terms of features, performance, and overall efficiency.

5.3.1 Feature Comparison

Compared to existing billing and inventory management systems, the developed application offers unique features. Many other systems do not provide this level of customization, especially in terms of restricting product categories or implementing specific business rules like low quantity alerts.

Additionally, the logging of every activity, including user actions and inventory changes, is a key feature that distinguishes the system from other basic billing solutions.

5.3.2 Performance Comparison

In terms of performance, the developed system is competitive with existing applications. The use of session management ensures that only relevant data is ac-

cessible to the correct users, while the efficient database design supports quick data retrieval and updates.

Compared to other systems, the developed solution also provides real-time notifications for stock and product alerts, which enhances its usability for small business owners. The ability to log and track all user activities provides additional accountability, a feature not commonly found in many existing applications.

5.4 Summary

The developed billing and inventory management system successfully meets the project objectives, providing accurate and efficient management of product inventory, billing, and user activity. The system's unique features, such as role-based access control and logging, differentiate it from other similar applications. The final implementation delivered high performance and user satisfaction, with a robust notification system and detailed logs to ensure smooth operations and accountability.

Chapter 6

Conclusion & Future Work

This chapter summarizes the key findings and conclusions drawn from the results and discussions presented in the previous chapters. Additionally, it reflects on the lessons learned in terms of project management and outlines potential areas for future improvement and development.

6.1 Conclusion

The development of the billing and inventory management system has successfully met the primary objectives of providing a robust, role-based access control system for shop owners and employees. The system effectively manages product inventories, billing processes, and user activities, while ensuring data integrity and accountability through comprehensive logging.

Key achievements include:

- Successful implementation of a role-based system where owners can manage inventory, bills, and employees, and employees can only generate bills.
- Comprehensive logging of all actions performed by users, ensuring a traceable history of system activities.
- Implementation of real-time notifications for low stock alerts and other system events, enhancing user experience.

Overall, the system demonstrated reliability and efficiency in handling typical business operations for small to medium-sized shops, with a user-friendly interface and real-time updates.

6.2 Lessons Learned from Project Management

During the course of this project, several lessons were learned regarding effective project management, particularly in software development.

6.2.1 Requirement Gathering and Understanding

One of the most significant lessons was the importance of thoroughly understanding and documenting the requirements before the development phase.

6.2.2 Time Management

Allocating sufficient time for each phase of the project, particularly the testing and debugging phases, proved to be crucial. Underestimating the time required to test and resolve bugs, especially in modules such as logging and notifications, led to some delays. Future projects would benefit from a more realistic allocation of time for each task.

6.2.3 Collaboration and Communication

Effective communication is the key to success in any collaborative effort. By maintaining open channels, expressing ideas clearly, and listening to the perspectives of others, we can not only overcome challenges but also elevate our project outcomes. Each team member brings unique strengths and insights, and when we communicate with respect and positivity, we build trust, strengthen relationships, and create a synergistic environment where creativity thrives.

Strong communication correlates directly with strong project performance. When we align our goals, share feedback constructively, and support one another, we turn potential obstacles into learning opportunities and set the stage for collective achievement.

6.3 Future Work

While the system met its current objectives, several areas of improvement and extension were identified that could be explored in future versions.

6.3.1 Enhanced User Role Management

Future work could explore more granular user roles, allowing for different levels of access for employees. For example, certain employees could be allowed to

update inventory or view logs, depending on their permissions.

6.3.2 Advanced Analytics and Reporting

The system could be enhanced with advanced analytics and reporting features, providing owners with insights into sales trends, inventory performance, and customer preferences. This would help improve decision-making and business optimization.

6.3.3 Mobile Application Integration

A potential future enhancement could be the development of a mobile version of the system, allowing owners and employees to manage their shop operations from anywhere. Mobile notifications for stock alerts, sales reports, and other events would improve accessibility and user convenience.

6.3.4 Integration with Payment Gateways

Integrating the system with online payment gateways could provide an additional feature for owners, allowing them to offer more payment options to customers and streamline the billing process.

6.3.5 Data Security and Encryption

Future versions of the system could incorporate stronger data security features, such as encryption of sensitive information and more advanced authentication mechanisms to protect against unauthorized access.

6.4 Summary

In conclusion, the billing and inventory management system successfully achieved its goals and provided a reliable, user-friendly solution for managing business operations. The lessons learned from this project in terms of requirement gathering, time management, and communication will be valuable in future endeavors. Several potential areas for improvement have been identified, and future work could focus on enhancing user roles, integrating mobile and analytics features, and improving data security.

References

- [1] J. Jenkov: *JavaFX Tutorials*: <https://jenkov.com/tutorials/javafx/index.html>
- [2] Video Tutorial: <https://youtu.be/lkov5shwRQs?feature=shared>
- [3] Video Tutorial: <https://www.youtube.com/watch?v=V9nDH2iBJSM>
- [4] Video Tutorial: <https://www.youtube.com/watch?v=9ntKSLLDeSs>
- [5] YouTube Playlist: https://youtube.com/playlist?list=PLZPZq0r_RZOM-8vJA3NQFZB7JroDcMwev&feature=shared

Github Repository

<https://github.com/Ai-Chetan/Bill-n-Box>

Acknowledgements

We, the undersigned, would like to extend our sincere gratitude to everyone who supported us throughout the completion of our college mini project. First and foremost, we are deeply thankful to our project guide Asst. Prof. Tayyabali Sayyad, for their expert guidance, constructive feedback, and constant encouragement. Their insights into software development and practical knowledge helped us navigate the challenges we encountered during the development of this billing and inventory management system.

We would also like to thank our college for providing us with the resources and opportunity to work on this project. The experience has enhanced our understanding of software design, teamwork, and project management. Finally, we are immensely grateful to our families and friends for their unwavering support, understanding, and motivation throughout this project.

(_____)
Aarya Khatate - 28

(_____)
Chetan Chaudhari - 10

(_____)
Vinay Gone - 63

(_____)
Yashraj Patil - 43

Date: