

CHAPTER 1**INTRODUCTION**

The electricity billing management system is made to keep the records about the bills of the customers. The admin can manage all the accounts and the registered users and customers can only manage their own accounts. The whole world is shifting from manual to automated, in every sector the technology. The Electricity Billing Management System stands out as a crucial solution for upgrading and improving the management of electricity usage and billing processes

This system helps in maintaining the bills and the payments of a customer. Admin and customers have a different interface and different privileges according to their needs. A different operation is there for customer to check the customers electricity bills and dues.

As the demand for electricity continues to rise and billing data becomes more complex, it's essential to move away from old-fashioned paper-based billing approaches into a digital platform to ensure smooth and efficient billing operations.

1.1 OBJECTIVES

The objective of the Electricity Billing Management System is to catalyse¹ the transition from manual, paper-driven processes to a dynamic digital ecosystem. This transition seeks to revolutionize user experience through an interactive easy to use interface, expedite data entry and processing, and implement cost-effective measures through cutting-edge technological integration. Harnessing the power of advanced analytics, it provides actionable insights to inform strategic decision making. Additionally, it endeavours to alleviate administrative burdens, enabling staff to allocate time judiciously to more value-adding endeavours, thereby bolstering overall operational efficiency, the system is committed to fostering a culture of sustainability by promoting energy conservation practices and facilitating the integration of renewable energy sources into the billing framework.

1.2 SCOPE

Our electricity billing management system offers a user-friendly platform to streamline billing processes and manage electricity usage effectively. With a focus on simplicity and efficiency, our system enables easy generation of invoices, tracking of payments, and

monitoring of electricity consumption. It ensures organized data storage for customer information and billing records while providing fast search and update capabilities. Designed for intuitive use, our system enhances efficiency by minimizing administrative tasks and facilitating clear reporting on usage trends and payment status. Scalable to various needs, it caters to both small-scale operations and larger utility companies, making electricity management straightforward and accessible for all users.

1.3 PRODUCT PERSPECTIVE

The Electricity Billing Management System stands as a transformative force in modernizing traditional electricity billing processes. By digitizing and streamlining operations, it simplifies the interactive management of electricity consumption data and billing procedures. The system boasts a user-centric interface, enhancing accessibility and usability for both administrators and consumers alike. The system optimizes productivity and efficiency in managing electricity billing tasks. Through centralized data storage and real-time analysis capabilities, it empowers administrators with invaluable insights for informed decision-making. Moreover, the system alleviates administrative burdens by automating repetitive tasks and reducing manual intervention.

CHAPTER 2**SYSTEM REQUIREMENTS**

System requirement specification is a detailed statement of the effects that a system is required to achieve. A good specification gives a complete statement of what the statement is to do, without any commitment as to how the system is to do it. It consists only externally observable behaviour and omits any design or implementation bias.

2.1 HARDWARE REQUIREMENTS

DEVICE	: Laptop-PH315-13
PROCESSOR	: Intel(R) Core(TM) i5-11500H CPU @ 2.20GHz 2.21
GHz RAM	: 16.00 GB

2.2 SOFTWARE REQUIREMENTS

OPERATING SYSTEM	: Windows11
FRONT END	: HTML
BACK END	: PHP, MYSQL
SOFTWARE USED	: XAMPPSERVER, Visual Studio

2.2.1 MYSQL:

MySQL, a widely acclaimed relational database management system, serves as the backbone for storing, organizing, and retrieving structured data in various applications. Renowned for its scalability, performance, and robustness. With its adherence to SQL standards and extensive feature set, MySQL offers developers powerful tools for managing data efficiently. XAMPP, on the other hand, is a popular web server solution stack package crafted by Apache Friends. It integrates MySQL alongside Apache HTTP Server, PHP, and Perl, providing developers with a comprehensive local server environment for testing and development purposes. This all-in-one package simplifies the setup process, allowing users to quickly establish a local server environment without the need for individual component installation and configuration.

2.2.2 HTML

HTML provides the basic structure and content of web pages. It defines the layout of elements such as text, images, and links, organizing them into a structured hierarchy. In HTML, developers can create static web pages and define the overall layout and visual structure of the application.

2.2.3 CSS

CSS complements HTML by adding style and presentation to web pages. It allows developers to customize the appearance of HTML elements, including colors, fonts, spacing, and layout. CSS enables the creation of visually appealing and responsive designs, enhancing the user experience across different devices and screen sizes.

2.2.4 JAVA SCRIPT

JavaScript is a versatile programming language that adds interactivity and dynamic behavior to web pages. It runs on the client side, enabling developers to create interactive elements such as animations, form validations, and event handling. JS can manipulate HTML and CSS elements in real time, making web applications more engaging and responsive.

2.2.5 PHP

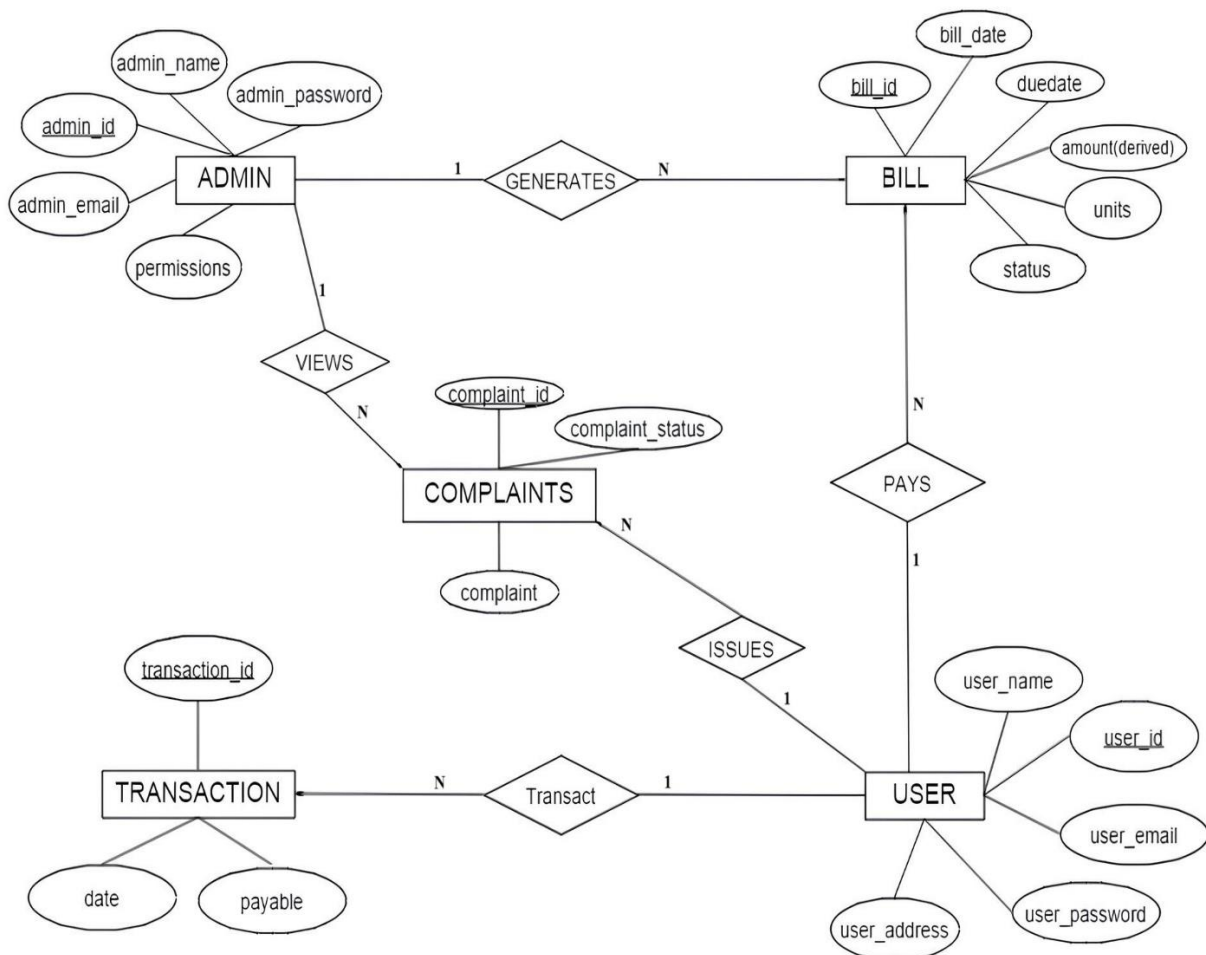
PHP is a server-side scripting language commonly used for backend web development. It allows developers to generate dynamic web content, interact with databases, and handle user input securely. Developers can create dynamic web pages, process form data, authenticate users, and perform server-side operations such as file handling and session management. PHP integrates seamlessly with databases like MySQL, enabling the retrieval and manipulation of data to power dynamic web applications.

CHAPTER 3**DESIGN AND IMPLEMENTATION**

Database design involves creating a detailed plan for how data will be stored and organized in a database. It includes deciding how information will be structured logically and physically, and determining the necessary storage settings.

3.1 ER DIAGRAM

An entity relationship model is a schematic way of describing and defining a business process. An ER model is typically implemented as a database. The main component of ER models entity set and relationship set.

**Fig: 3.1 ER diagram**

3.2 RELATIONSHIP:

ENTITY 1	ENTITY 2	RELATIONSHIP	DESCRIPTION	CARDINAL
Admin	Bill	Generates	1 Admin can generate N number of Bill	1:N
Admin	Complaints	Views	1 Admin can view N number of Complaints	1:N
User	Bill	Pays	1 User can pay N number of Bill	1:N
User	Complaints	Issues	1 User can issue N number of complaints	1:N
User	Transaction	Transact	1 User can transact N number of transactions	1:N

Table 1: Relationships in the Relation Schema

3.3 RELATIONAL SCHEMA

It formulates all the constraints that are to be applied on the data. A database schema defines all the entities and the relationship among them. It contains a descriptive detail of the database. It shows how the data will be stored in secondary storage.

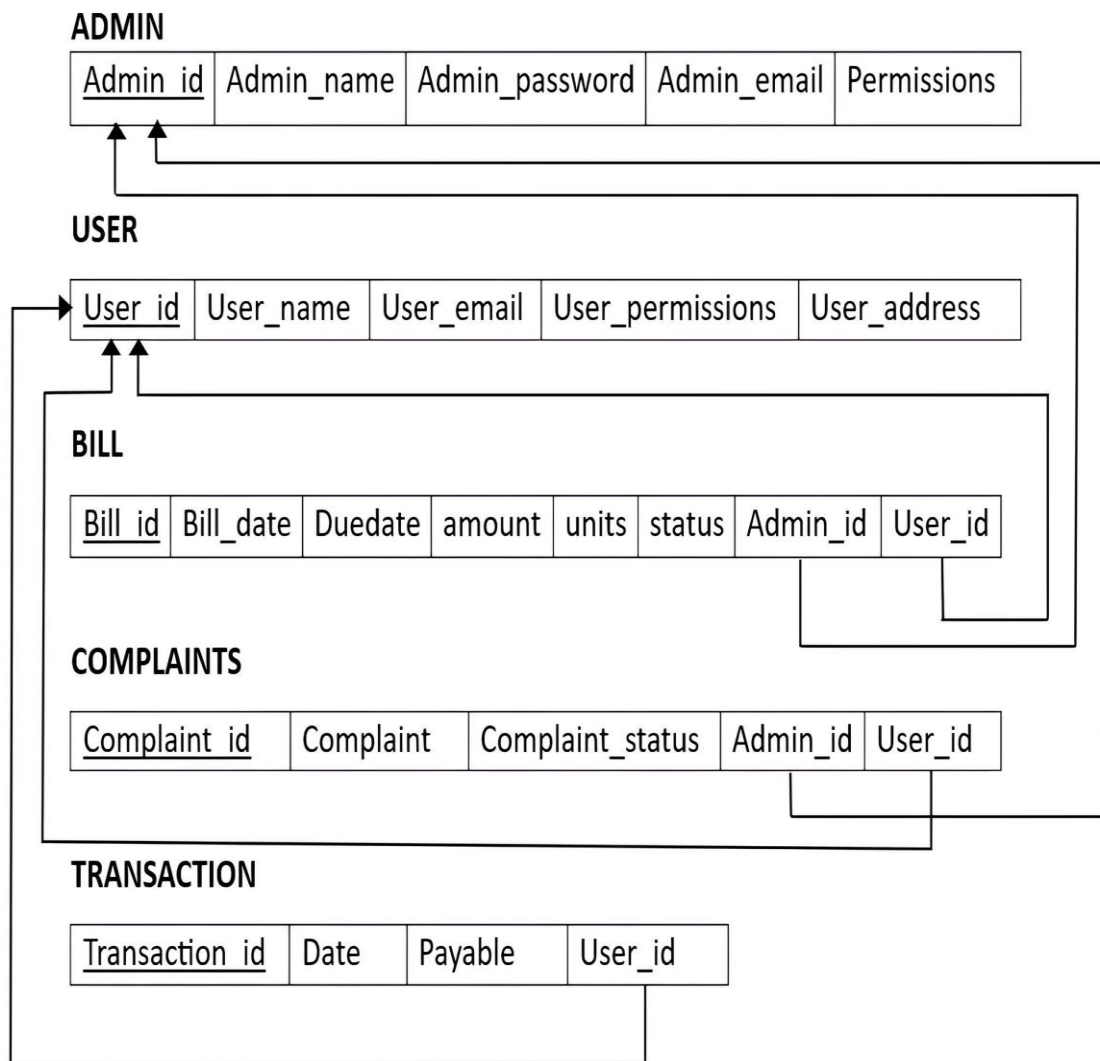


Fig: 3.2 Schema Diagram

3.4 IMPLEMENTATION

- Here we discuss about the codes that are used in the implementation.
- We have used MYSQL as the back end to store data and implemented PHP codes in VS CODE text editor.

3.4.1 PHP CONNECTION CODE

```
<?php
```

```
$host='localhost'; # MySQL Host
```

```
$mysql_user="root";# MySql Username
```

```
$mysql_pwd=""; # MySql Password

$dbms="ebillsystem"; # Database

$con = mysqli_connect($host,$mysql_user,$mysql_pwd,$dbms);

if (!$con) die('Could not connect: ' . mysqli_error($con));

mysqli_select_db($con,$dbms) or die("cannot select DB" . mysqli_error($con));

?>
```

3.4.2 CODE FOR TABLE CREATION

```
CREATE TABLE `ADMIN` (
`ID` INT(14) NOT NULL,
`NAME` VARCHAR(40) NOT NULL,
`EMAIL` VARCHAR(40) NOT NULL,
`PASS` VARCHAR(20) NOT NULL );

CREATE TABLE `USER` (
`ID` INT(14) NOT NULL,
`NAME` VARCHAR(40) NOT NULL,
`EMAIL` VARCHAR(40) NOT NULL,
`PHONE` VARCHAR(255) NOT NULL,
`PASS` VARCHAR(20) NOT NULL,
`ADDRESS` VARCHAR(100) NOT NULL );

CREATE TABLE `BILL` (
`ID` INT(14) NOT NULL,
`AID` INT(14) NOT NULL,
`UID` INT(14) NOT NULL,
`UNITS` INT(10) NOT NULL,
```



```
`AMOUNT` DECIMAL(10,2) NOT NULL,  
`STATUS` VARCHAR(10) NOT NULL,  
`BDATE` DATE NOT NULL,  
`DDATE` DATE NOT NULL );
```

```
CREATE TABLE `COMPLAINT` (  
`ID` INT(14) NOT NULL,  
`UID` INT(14) NOT NULL,  
`AID` INT(14) NOT NULL,  
`COMPLAINT` VARCHAR(140) NOT NULL,  
`STATUS` VARCHAR(40) NOT NULL );
```

```
CREATE TABLE `TRANSACTION` (  
`ID` INT(14) NOT NULL,  
`BID` INT(14) NOT NULL,  
`PAYABLE` DECIMAL(10,2) NOT NULL,  
`PDATE` DATE DEFAULT NULL,  
`STATUS` VARCHAR(10) NOT NULL );
```

```
CREATE TABLE `UNITSRATE` (  
`SNO` INT(1) DEFAULT NULL,  
`TWOHUNDRED` INT(14) NOT NULL,  
`FIVEHUNDRED` INT(14) NOT NULL,  
`THOUSAND` INT(14) NOT NULL );
```

CHAPTER 4**TESTING AND ANALYSIS**

Testing and analysis play a crucial role in ensuring the quality and reliability of any software system, including an electricity management system developed using HTML, CSS, JavaScript, PHP, MySQL, and XAMPP. Let's break down the testing and analysis process in a simpler, easy-to-understand way:

4.1 Purpose of Testing

The main goal of testing is to find and fix any errors or weaknesses in the electricity management system. It's like double-checking to make sure everything works as expected and meets the needs of users. Testing helps ensure that the system does what it's supposed to do and doesn't crash unexpectedly.

4.2 Types of Testing:

There are different types of tests we can perform to make sure our electricity management system works correctly. Each type of test focuses on a specific aspect of the system to ensure it meets the requirements.

4.2.1 System Testing: This type of testing checks the entire electricity management system to make sure it meets all the requirements and works as expected. For example, we'll make sure that users can log in, view their electricity usage, and pay their bills without any problems.

4.2.2 Integration Testing: Integration testing checks how different parts of the system work together. We'll test how the HTML, CSS, JavaScript, PHP, and MySQL components interact to ensure they communicate correctly and share data seamlessly.

4.2.3 Unit Testing: Unit testing is done for individual parts, or modules, of the system. It checks that each part works correctly on its own before integrating it with other parts. For instance, we'll test the login form to ensure that users can log in successfully and that any errors are handled properly.

4.2.4 Using XAMPP for Testing: XAMPP provides a convenient environment for testing our electricity management system locally before deploying it to a live server. It includes

Apache, MySQL, PHP, and other tools needed to run the system on our computer, making it easy to test and debug any issues. Testing and analysis ensure that our electricity management system functions smoothly and meets the needs of users. By testing each component and the system as a whole, we can identify and fix any problems before they impact users. XAMPP provides a convenient platform for testing our system locally, ensuring a smooth deployment process.

CHAPTER 5

SCREENSHOTS

5.1 DATABASE PAGE

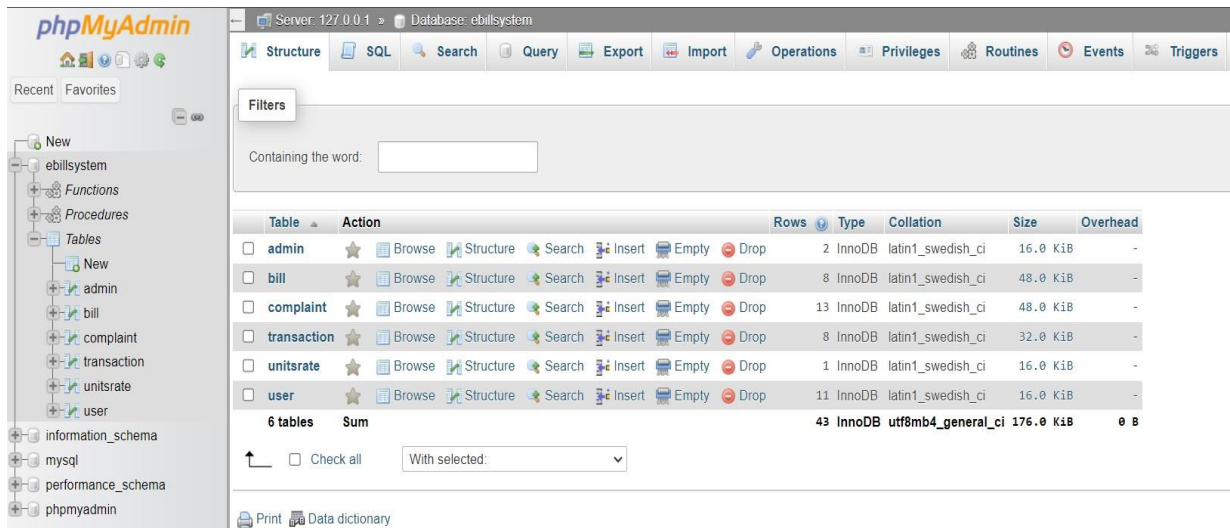


Fig: 5.1 Database Page

5.2 LOGIN PAGE

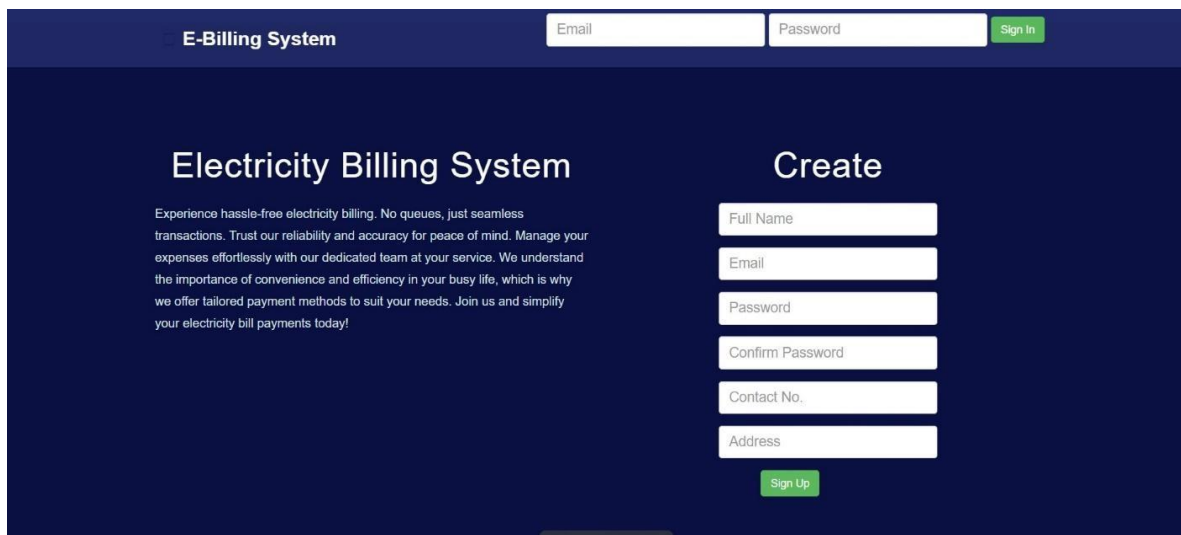


Fig: 5.2 Login Page

5.3 ADMIN DASHBOARD

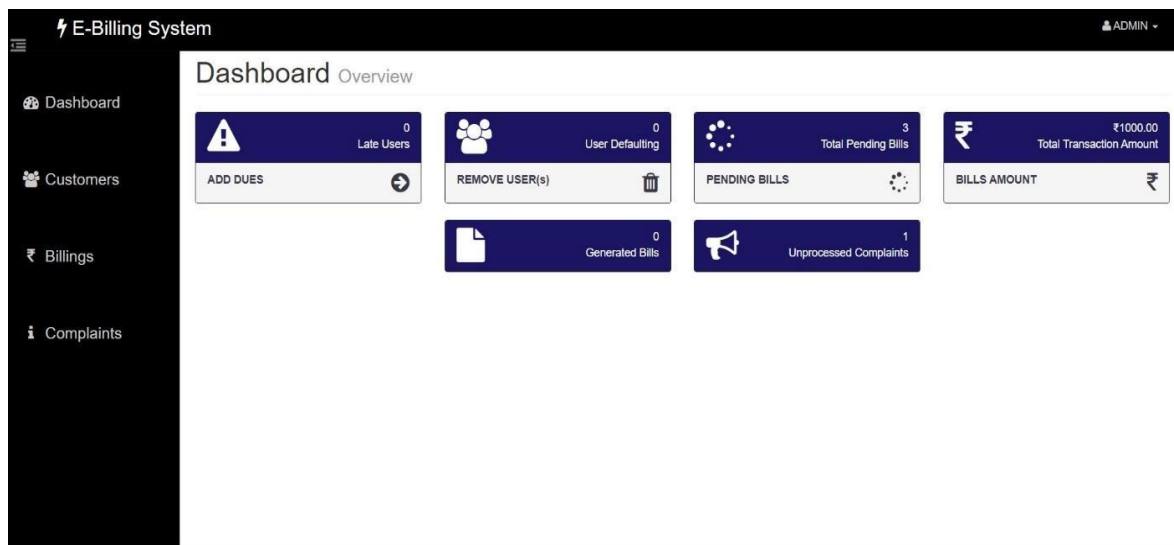


Fig: 5.3 Admin Dashboard

5.4 ADMIN_CUSTOMER DETAILS

The Admin Customer Details page displays the following table:

Sl.no	Name	Email	Contact	Address	Delete user
1	AB de Villiers	MR360@gmail.com	7450002145	17 Allen At Earth,Sullia	Delete
2	Ambani	Ambani@gmail.com	7854547855	3961 Ambani Villa,Sullia	Delete
3	Cristiano Ronaldo	goat@gmail.com	7012569980	7 Goat Street Cr7,Sullia	Delete
4	Kylian Mbappe	mbappe@gmail.com	7012458888	10 Santiago Barnebue,Sullia	Delete
5	Lionel Messi	Pessi@gmail.com	7012565800	4830 Chor Bazar Street,Sullia	Delete

Page navigation: 1 2 3 NEXT LAST

Fig: 5.4 Admin_Customer Details

5.5 ADMIN_BILLS HISTORY

Bills						
Bills History				Generate New Bill		
Bill No.	Customer Name	Date	UNITS Consumed	Amount	Due Date	Status
BN_28	Rahul	2024-03-16	100	₹477.00	2024-04-15	PENDING
BN_27	Rohit	2024-03-16	250	₹1295.00	2024-04-15	PENDING
BN_26	Virat Kohli	2024-03-16	82	₹398.34	2024-04-15	PENDING
BN_25	AB de Villiers	2024-03-16	360	₹1957.20	2024-04-15	PENDING

Fig: 5.5 Admin_Bills History

5.6 ADMIN_BILL GENERATION

Bills					
Bills History			Generate New Bill		
USER	UNITS	BILL DATE	DUE DATE	GENERATE	
Ambani	280	2024-03-16	2024-04-15	GENERATE BILL	
Cristiano Ronaldo	ENTER UNITS	2024-03-16	2024-04-15	GENERATE BILL	
Kyllan Mbappe	ENTER UNITS	2024-03-16	2024-04-15	GENERATE BILL	
Lionel Messi	ENTER UNITS	2024-03-16	2024-04-15	GENERATE BILL	

Fig: 5.6 Admin_Bill Generation

5.7 ADMIN_COMPLAINT DETAILS

Complaints				
Complaint / Not Processed				
Complaint No.	User	Complaint	Status	Process
CA-12	AB de Villiers	Bill Generated Late	NOT PROCESSED	PROCESS COMPLAINT

Fig: 5.7 Admin_Complaint Details

5.8 CUSTOMER PROFILE



Fig: 5.8 Customer Profile

5.9 CUSTOMER DASHBOARD

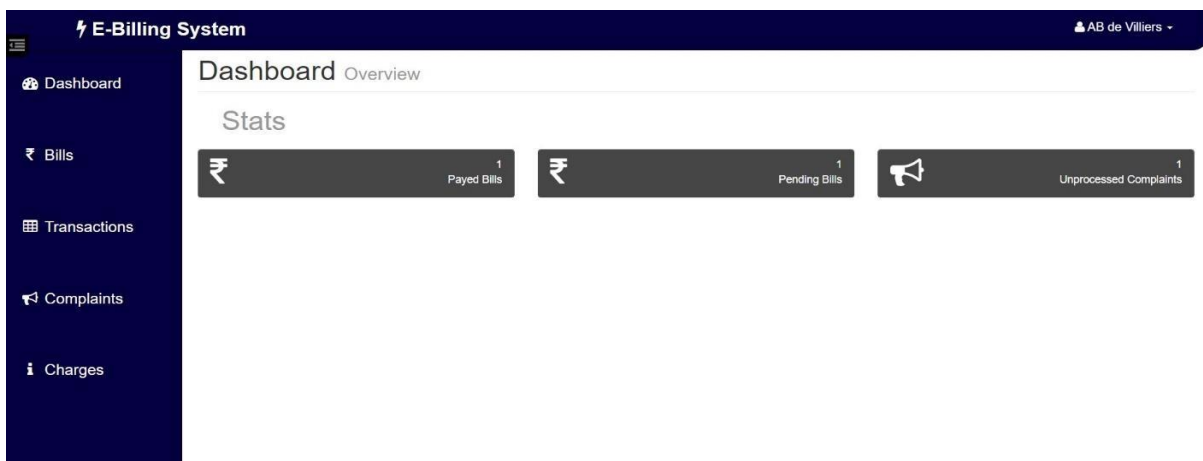


Fig: 5.9 Customer Dashboard

5.10 CUSTOMER BILL HISTORY

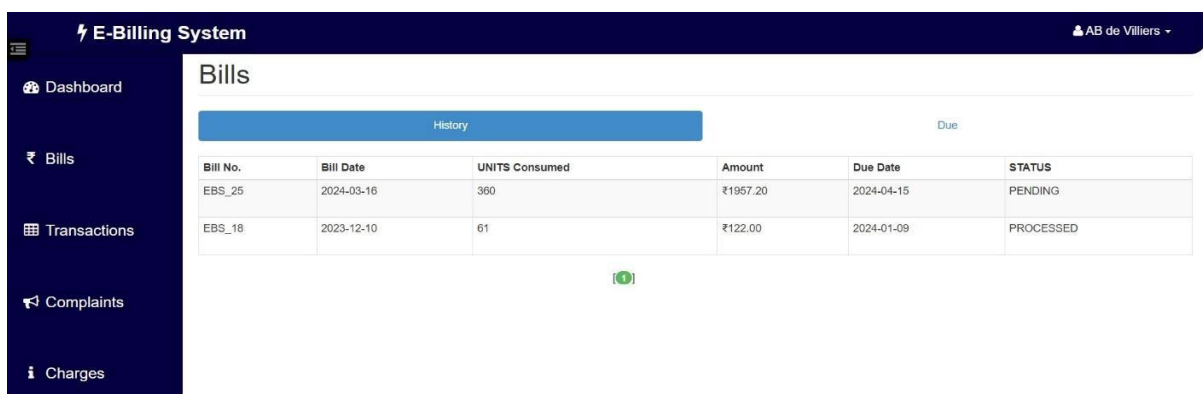


Fig: 5.10 Customer Bill History

5.11 CUSTOMER TRANSCATIONS

Transaction No.	Bill Date	Amount	Dues (if any)	Final Amount Paid	Transaction Date
-	2024-03-16	₹1957.20	₹0.00	₹1957.20	TRANSACTION PENDING
TRN_18	2023-12-10	₹122.00	₹0.00	₹122.00	2024-01-10

Fig: 5.11 Customer Transactions

5.12 CUSTOMER COMPLAINT

Complaint No.	Bill Date	Status	Description
CA-12		NOT PROCESSED	
CA-6		PROCESSED	Bill Not Correct
CA-2		PROCESSED	Transaction Not Processed
CA-1		PROCESSED	Transaction Not Processed

Fig: 5.12 Customer Complaint

5.13 CHARGES DEATILS

Calculate the energy cost based on the usage of units:

- First 100 units: Rs. 4.22 per unit
- Next 100 units: Rs. 5.02 per unit
- Remaining units: Rs. 5.87 per unit
- Add the fixed fee: Rs. 40
- Add the energy duty: Rs. 0.15 per unit

Total energy bill = Energy cost + Fixed fee + Energy duty

Fig: 5.12 Charges Details

CHAPTER 6**CONCLUSION**

In conclusion, the testing and analysis phase of our electricity billing management system project has been good in ensuring its functionality and user satisfaction. Through various types of testing such as system testing, integration testing, and unit testing, we have examined different aspects of the system to identify and rectify any potential issues. Utilizing XAMPP as our testing environment has provided us with a convenient platform to simulate real-world usage scenarios and debug any anomalies before deployment.

Looking ahead, there are several avenues for future enhancements to further improve the system's efficiency and user experience. Firstly, Making the user to provide signup message, Further, integrating advanced analytics capabilities could offer users insights into their electricity consumption patterns, helping them make informed decisions to reduce energy usage and costs. Introducing features for online payment options and mobile accessibility would enhance the accessibility of the system.

REFERENCE

- 1.Elmasri Navathe "Fundamental of Database System 3rd edition". Pearson Education, 2000.
2. Ramakrishnan and Gehrke "Database management systems" 3rd edition. McGrawHill, 2002.
3. Apna College MYSQL - https://youtu.be/hlGoQC332VM?si=IVnKfSl_UgLGUWpf
4. Code with Harry Frontend tutorial - https://youtu.be/x5XgoTjGTu4?i=0WgFsN9F_OE2zrp