

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi: 590 018



Database Management Systems Mini Project report on

“XXXXXXXXXXXXXX”

Submitted in partial fulfillment of the requirement for the award of Degree of

**BACHELOR OF ENGINEERING
IN
ARTIFICIAL INTELLIGENCE & MACHINE LEARNING**

By

**XXXXXXXXXXXX
XXXXXXX 1AY18AIXXX
XXXXXXX 1AY18AIXXX**

Under the guidance of

Mr / Mrs. XXXXXXXX



**DEPARTMENT OF ARTIFICIAL INTELLIGENCE &
MACHINE LEARNING
ACHARYA INSTITUTE OF TECHNOLOGY
(Affiliated to Visvesvaraya Technological University, Belagavi)
2022-2023**

ACHARYA INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi)
Soladevanahalli, Bangalore – 560090

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING



Certificate

Certified that the Database Management Systems mini project entitled **XXXXXXXXXXXXXX** is a bonafide work carried out by **XXXXXXXXX (1AY18AI000) & XXXXXXX(1AY18AI000)** of Fifth semester in partial fulfillment for the award of degree of **Bachelor of Engineering in Artificial Intelligence & Machine Learning** of the **Visvesvaraya Technological University, Belagavi**, during the year **2022-2023**. It is certified that all corrections/ suggestions indicated for internal assessments have been incorporated in the Report deposited in the departmental library. The Mini Project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for the **Bachelor of Engineering Degree**.

Signature of Guides

Signature of H.O.D

Name of the examiners

Signature with date

1.

2.

ACKNOWLEDGEMENT

I express my gratitude to our institution and management for providing us with good infrastructure, laboratory, facilities and inspiring staff whose gratitude was of immense help in completion of this mini-project successfully.

I am grateful to the institute Acharya Institute of Technology and management with its ideas and inspiration for having provided us with the good infrastructure, laboratory, facilities and inspiring staff which has made this seminar report complete successfully.

I would like to express my sincere gratitude to **Dr. Rajath Hegde M M**, Principal, AIT for all the facilities that he has extended throughout my work.

I heartily thank and express my sincere gratitude to **Dr. Vijayashekhar S Sankannavar**, Associate Professor and Head, Dept. of AIML, AIT for his valuable support and a constant source of enthusiastic inspiration to steer us forward.

I would like to express my sincere gratitude to the Internal Guide **Mr/Ms. XXXX**, Assistant Professor, Dept. of AIML, AIT for his valuable guidance and support.

I would like to express my sincere gratitude to the Mini-project Coordinator **Mr. Jovin Deglus** Assistant Professor, Dept. of AIML, AIT for their valuable guidance and support.

Finally, I would like to express my sincere gratitude to my parents, all teaching and non-teaching faculty members and friends for their moral support, encouragement and help throughout the completion of the mini-project.

XXXXXXX

1AY18AI000

ABSTRACT

The main aim and objective was to plan and program system application .We have to apply the best software engineering practice for system application. I developed a “Electricity Bill Management System” using PHP and SQL. The purpose of the Electricity Bill Management System is to automate the existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for long period with easy accessing and the manipulation of the same. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources. The organization can maintain computerized records without redundant entries.

The project is totally build at administrative end and thus only administrator is guaranteed the access. This project provides the different module for employees to check the customer's details if their job requires. Admin, employees and customers all have a different interface and different privileges according to their needs. Basically the project describes how to manage for good performance and better services.

LIST OF FIGURES

Fig 1.1 Architecture

3

LIST OF TABLES

Table 2.1 Student Table

8

TABLE OF CONTENTS

Acknowledgement	i	
Abstract	ii	
List of Figures	iii	
List of Tables	iv	
Chapters	Title	Page No.
1.	Introduction	1
2.	Requirements Specification	2
3.	Tools and Technologies Used	10
4.	Design	
	4.1 Data Flow Diagram	11
	4.2 Use case Diagram	
	4.3 Class Diagram	
	4.4 ER Diagram	
5.	Implementation	12
	5.1 Database Schema	
	5.2 Table Design	
6.	Testing	13
7.	Conclusion	14
	Appendices	
	Bibliography	

CHAPTER 1

INTRODUCTION

This project entitled “**Electricity Bill Management System**” aims to generate electricity bill with all the charges and penalty. Manual system that is employed is extremely laborious and quit inadequate. It only makes the process more difficult and hard. So, we aims to develop a system that is mean to partially computerized the work performed in the Electricity Board like generating monthly electricity bill, record of consuming unit of energy, store record of the customer and previous unpaid record. This has been developed to overwrite the problems prevailing in the practicing manual system. This software is supported to eliminate in some cases reduce the hardship faced by this existing system. Moreover this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

The Application is reduced as much as possible to avoid errors while entering the data. No formal knowledge is np[eeded for the user to use this system. Thus, by this all it proves it is user friendly. This is designed to assist in a strategic planning, and will help you ensure that your organization is equipped with the right level of information and details for our future goals. Also, for those busy executive who are always on the go, our systems come with the remote access features. Which will allow us to manage our workforce anytime, at all time. These systems will ultimately allow you to better manage resources.

ABOUT THE PROJECT:

Home page: After entering into the system home page will appear, In home page there is a user register option, admin and user signup options .And there are steps to know how this portal works .

Admin page: As we login to the admin page we can manage all details like generating the bill, processing the user complaints and so on.

User page: As we login to the user page we can see the user details such as number of bills paid, pending. User can pay the bills here. And user can also raise the complaints.

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

Requirements specification is a specification of software requirements and hardware requirements required to do the project.

2.1 Hardware Requirements Specification

Hardware Requirements are the hardware resources that are need to do the project work. These resources are a computer resource provides functions and services to do the project. Hardware resources required for our project are shown below.

- Processor : Intel i3 core
- RAM : $\geq 1\text{GB}$
- Hard disk : Minimum 10 GB
- Keyboard : QWERTY

2.2 Software Requirements Specification

Software Requirements are the software resources that are need to do the project work. These resources are installed on a computer in order to provide functions, services, hardware accessing capabilities to do the project.

In our project we used the following software resources.

- Operating System: Windows 10
- Xampp Server 3.2.4.0 tool
- Apache web server
- MySQL database server
- PHP for web page development
- phpMyadmin for interfacing PHP and MySQL

2.3 FUNCTIONAL REQUIREMENTS:

Functional requirements specify a function that system or a system component must be able to perform. It can be documented in various ways.

- Sign Up: The user should be able to setup a new account.
- Sign In: All the users and admin should be able to log onto the system by a email and password for each profile.
- Update Profile: Users of the system should be able to update their profile information including changing of password.
- Pay Bill: Users should be able to pay their electricity bills.
- View Bill: Users should be able view their electricity bill at the beginning of the month.
- Logout: Both the users and admin should be able to log out of the system.
- Calculate: The system must be able to calculate the bill of the users.
- Make Complaint: The customers should be able to make and submit complaints.
- Process Complaint: The admin should be able to process the users complaint.

2.4 NON-FUNCTIONAL REQUIREMENTS:

- Reliability: Database updating should follow transaction processing to avoid data inconsistency.
- Availability: The project will be deployed on a public shared server so it will be available all the time and will be accessible anywhere of the world using internet.
- Security: We have implemented a lot of security mechanism to avoid to hack the system by outer world.
- Maintainability: It is very easy to maintain the system. The system has been developed on php so anyone who has the knowledge of php, can easily maintain the system.
- Portability: Yes this system is portable and we can switch the servers very easily.
- Browser Compatibility: The project being web based required compatibility with at least the popular web browsers. Microsoft windows XP and above, Linux and Macintosh being the current popular operating system and Microsoft Internet

Explorer, Mozilla Firefox, Opera, Safari and Google Chrome being the currently popular web browsers.

CHAPTER 3

SYSTEM DESIGN

3.1 Entity-Relationship Diagram

The entity-relationship diagram, also known as the E-R Diagram, is a high level database design, which shows the database in diagrammatic approach. It consists of entities, relationships, attributes and associations. The E-R Diagram for the project is shown in the figure 3.1 below:

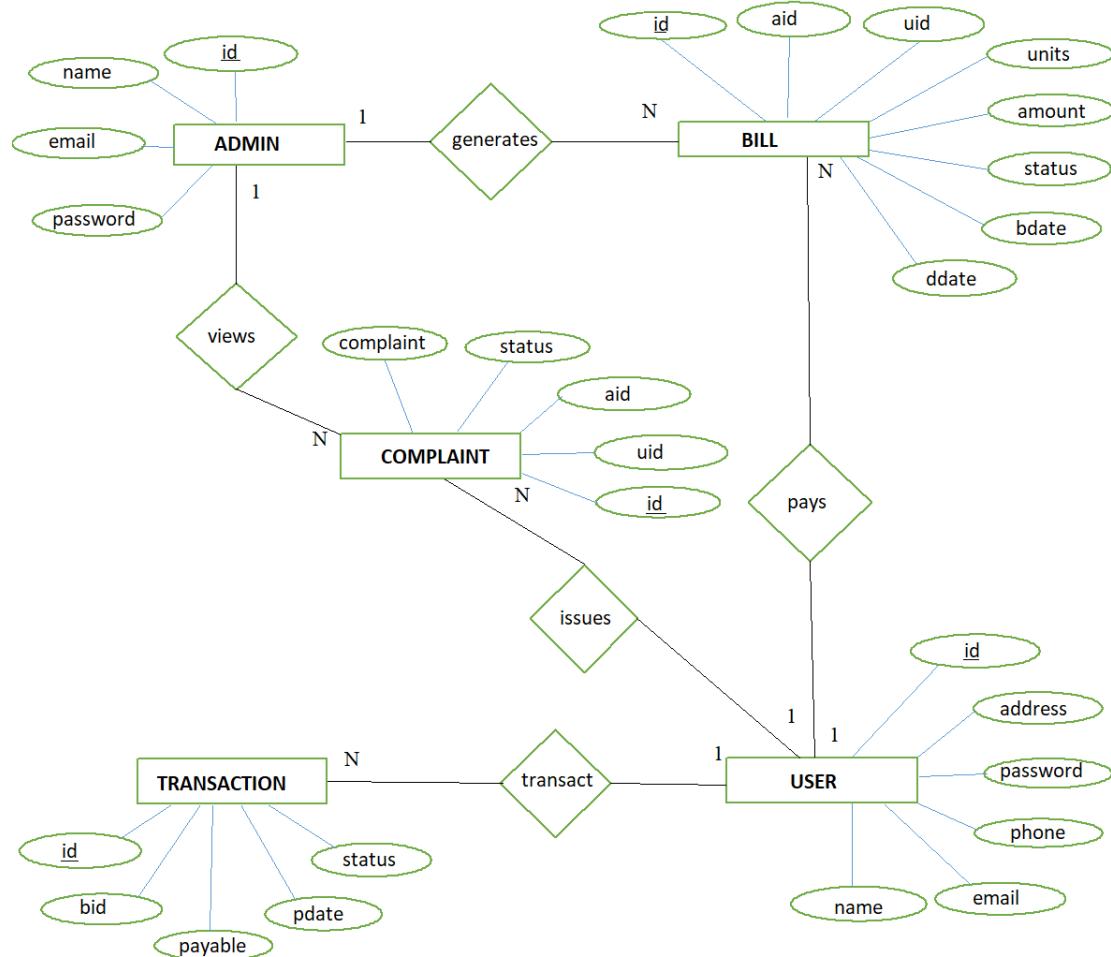


Figure 3.1: E-R Diagram of the Electricity Bill Management System

3.2 Schema Diagram

A schema diagram is an illustrative display of most aspects of a database schema. A schema construct is a component of the schema, or an object within the schema. The schema diagram of the database system is illustrated in figure 3.2:

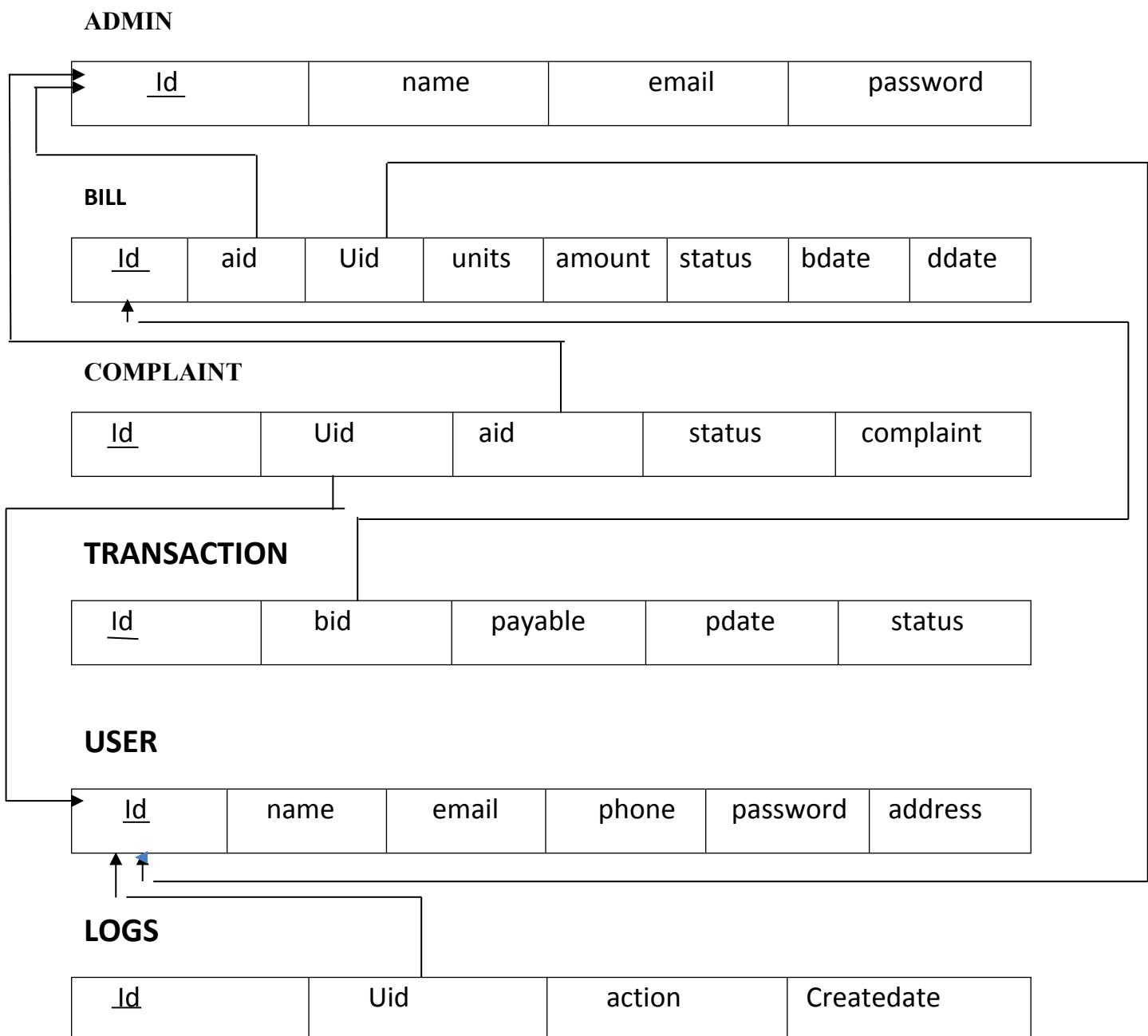


Figure 3.2: Schema Diagram of the Electricity Bill Management System

CHAPTER 4

IMPLEMENTATION

Databases are the storehouses of data used in the software systems. The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system. Two essential settings for a database are

Primary key- the field that is unique for all the record occurrences.

Foreign key-the field used to set relation between tables.

MySQL is multithreaded, multi user SQL database management System (DBMS). The basic program run as server providing multiuser access to a number of databases. MySQL is a database. The data in a MySQL is stored in a Database objects called tables. A table is a collection of related data entries and it consists of columns and rows. The databases are useful when storing information categorically.

4.1 CREATION OF TABLES:

The Tables created are:

- Admin
- Bill
- Complaint
- Logs
- Transaction
- Unitsrate
- User

➤ Admin

```
CREATE TABLE admin (id int(14) NOT NULL, name varchar(40) NOT NULL,
email varchar(40) NOT NULL, pass varchar(20) NOT NULL);
```

+ Options

Field	Type	Null	Key	Default	Extra
id	int(14)	NO	PRI	NULL	auto_increment
name	varchar(40)	NO		NULL	
email	varchar(40)	NO		NULL	
pass	varchar(20)	NO		NULL	

Figure 4.1: Creation of Admin Table

➤ Bill

```
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);
```

+ Options

Field	Type	Null	Key	Default	Extra
id	int(14)	NO	PRI	NULL	auto_increment
aid	int(14)	NO	MUL	NULL	
uid	int(14)	NO	MUL	NULL	
units	int(10)	NO		NULL	
amount	decimal(10,2)	NO		NULL	
status	varchar(10)	NO		NULL	
bdate	date	NO		NULL	
ddate	date	NO		NULL	

Figure 4.2: Creation of User Table

➤ Complaint

```
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);
```

+ Options					
Field	Type	Null	Key	Default	Extra
id	int(14)	NO	PRI	NULL	auto_increment
uid	int(14)	NO	MUL	NULL	
aid	int(14)	NO	MUL	NULL	
complaint	varchar(140)	NO		NULL	
status	varchar(40)	NO		NULL	

Figure 4.3: Creation of Complaint Table

➤ Logs

```
CREATE TABLE logs (id int(11) NOT NULL, uid int(11) NOT NULL,action
varchar(50) NOT NULL, createdate datetime NOT NULL);
```

+ Options					
Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
uid	int(11)	NO	MUL	NULL	
action	varchar(50)	NO		NULL	
createdate	datetime	NO		NULL	

Figure 4.4: Creation of Logs Table

➤ Transaction

```
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);
```

+ Options					
Field	Type	Null	Key	Default	Extra
id	int(14)	NO	PRI	NULL	auto_increment
bid	int(14)	NO	MUL	NULL	
payable	decimal(10,2)	NO		NULL	
pdate	date	YES		NULL	
status	varchar(10)	NO		NULL	

Figure 4.5: Creation of Transaction Table

➤ Unitsrate

```
CREATE TABLE unitsrate (sno int(1) DEFAULT NULL, twohundred int(14)
NOT NULL, fivehundred int(14) NOT NULL, thousand int(14) NOT NULL);
```

+ Options

Field	Type	Null	Key	Default	Extra
sno	int(1)	YES		NULL	
twohundred	int(14)	NO		NULL	
fivehundred	int(14)	NO		NULL	
thousand	int(14)	NO		NULL	

Figure 4.6: Creation of Unitsrate Table

➤ User

```
CREATE TABLE user (id int(14) NOT NULL, name varchar(40) NOT NULL,
email varchar(40) NOT NULL, phone int(14) NOT NULL, pass varchar(20) NOT
NULL, address varchar(100) NOT NULL);
```

+ Options

Field	Type	Null	Key	Default	Extra
id	int(14)	NO	PRI	NULL	auto_increment
name	varchar(40)	NO		NULL	
email	varchar(40)	NO		NULL	
phone	int(14)	NO		NULL	
pass	varchar(20)	NO		NULL	
address	varchar(100)	NO		NULL	

Figure 4.7: Creation of User Table

4.2 Insertion Of Values:

➤ Admin

```
INSERT INTO `admin` (`id`, `name`, `email`, `pass`) VALUES(1, 'raksha',
'raksha@gmail.com', 'raksha123');
```

```
INSERT INTO `admin` (`id`, `name`, `email`, `pass`) VALUES(2, 'rutu',
'rutu@gmail.com', 'rutu123');
```

	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	id	name	email	pass
	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1	raksha	raksha@gmail.com	raksha123
	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	2	rutu	rutu@gmail.com	rutu123

Figure 4.8: Inserting values to the Admin Table

➤ Bill

```
INSERT INTO `bill` (`id`, `aid`, `uid`, `units`, `amount`, `status`, `bdate`, `ddate`)
VALUES(1, 1, 2, 12, '24.00', 'PROCESSED', '2014-12-01', '2014-12-31');
```

```
INSERT INTO `bill` (`id`, `aid`, `uid`, `units`, `amount`, `status`, `bdate`, `ddate`)
VALUES(2, 1, 1, 200, '400.00', 'PROCESSED', '2014-09-01', '2014-10-01');
```

```
INSERT INTO `bill` (`id`, `aid`, `uid`, `units`, `amount`, `status`, `bdate`, `ddate`)
VALUES(3, 1, 2, 103, '206.00', 'PROCESSED', '2014-09-01', '2014-10-01');
```

```
INSERT INTO `bill` (`id`, `aid`, `uid`, `units`, `amount`, `status`, `bdate`, `ddate`)
VALUES(4, 1, 3, 453, '1665.00', 'PROCESSED', '2014-09-01', '2014-10-01');
```

```
INSERT INTO `bill` (`id`, `aid`, `uid`, `units`, `amount`, `status`, `bdate`, `ddate`)
VALUES(5, 1, 4, 654, '3440.00', 'PROCESSED', '2014-09-01', '2014-10-01');
```

```
INSERT INTO `bill` (`id`, `aid`, `uid`, `units`, `amount`, `status`, `bdate`, `ddate`)
VALUES(6, 1, 5, 609, '2990.00', 'PROCESSED', '2014-09-01', '2014-10-01');
```

	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	id	aid	uid	units	amount	status	bdate	ddate
	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1	1	2	12	24.00	PROCESSED	2014-12-01	2014-12-31
	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	2	1	1	200	400.00	PROCESSED	2014-09-01	2014-10-01
	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	3	1	2	103	206.00	PROCESSED	2014-09-01	2014-10-01
	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	4	1	3	453	1665.00	PROCESSED	2014-09-01	2014-10-01
	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	5	1	4	654	3440.00	PROCESSED	2014-09-01	2014-10-01
	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	6	1	5	609	2990.00	PROCESSED	2014-09-01	2014-10-01

Figure 4.9: Inserting values to the Bill Table

➤ Complaint

```
INSERT INTO `complaint` (`id`, `uid`, `aid`, `complaint`, `status`) VALUES(1, 1, 1, 'Transaction Not Processed', 'PROCESSED');
```

```
INSERT INTO `complaint` (`id`, `uid`, `aid`, `complaint`, `status`) VALUES(2, 1, 1, 'Transaction Not Processed', 'PROCESSED');
```

```
INSERT INTO `complaint` (`id`, `uid`, `aid`, `complaint`, `status`) VALUES(3, 2, 1, 'Previous Complaint Not Processed', 'PROCESSED');
```

```
INSERT INTO `complaint` (`id`, `uid`, `aid`, `complaint`, `status`) VALUES(4, 2, 1, 'Transaction Not Processed', 'PROCESSED');
```

```
INSERT INTO `complaint` (`id`, `uid`, `aid`, `complaint`, `status`) VALUES(5, 2, 2, 'Transaction Not Processed', 'PROCESSED');
```



The screenshot shows the 'complaint' table in MySQL Workbench. The table has columns: id, uid, aid, complaint, and status. The data is as follows:

	<input type="checkbox"/>	Edit	Copy	Delete	id	uid	aid	complaint	status
	<input type="checkbox"/>				1	1	1	Transaction Not Processed	PROCESSED
	<input type="checkbox"/>				2	1	1	Transaction Not Processed	PROCESSED
	<input type="checkbox"/>				3	2	1	Previous Complaint Not Processed	PROCESSED
	<input type="checkbox"/>				4	2	1	Transaction Not Processed	PROCESSED
	<input type="checkbox"/>				5	2	2	Transaction Not Processed	PROCESSED

Figure 4.10: Inserting values to the Complaint Table

➤ Logs

```
INSERT INTO `logs` (`lid`, `id`, `action`, `createdate`) VALUES(1, 9, 'NEW USER FOUND', '2021-01-06 12:25:53');
```



The screenshot shows the 'logs' table in MySQL Workbench. The table has columns: lid, id, action, and createdate. The data is as follows:

	<input type="checkbox"/>	Edit	Copy	Delete	lid	id	action	createdate
	<input type="checkbox"/>				1	9	NEW USER FOUND	2021-01-06 12:25:53

Figure 4.11: Inserting values to Logs Table

➤ Transaction

```
INSERT INTO `transaction` ('id', 'bid', 'payable', 'pdate', 'status') VALUES(1, 1, '1024.00', '2015-01-06', 'PROCESSED');
```

```
INSERT INTO `transaction` ('id', 'bid', 'payable', 'pdate', 'status') VALUES(2, 2, '1400.00', '2014-10-10', 'PROCESSED');
```

```
INSERT INTO `transaction` ('id', 'bid', 'payable', 'pdate', 'status') VALUES(3, 3, '1206.00', '2014-10-10', 'PROCESSED');
```

```
INSERT INTO `transaction` ('id', 'bid', 'payable', 'pdate', 'status') VALUES(4, 4, '2665.00', '2014-10-10', 'PROCESSED');
```

```
INSERT INTO `transaction` ('id', 'bid', 'payable', 'pdate', 'status') VALUES(5, 5, '4440.00', '2014-10-10', 'PROCESSED');
```

```
INSERT INTO `transaction` ('id', 'bid', 'payable', 'pdate', 'status') VALUES(6, 6, '3990.00', '2014-10-10', 'PROCESSED');
```

+ Options								
	← T →	▼	id	bid	payable	pdate	status	
<input type="checkbox"/>				1	1	1024.00	2015-01-06	PROCESSED
<input type="checkbox"/>				2	2	1400.00	2014-10-10	PROCESSED
<input type="checkbox"/>				3	3	1206.00	2014-10-10	PROCESSED
<input type="checkbox"/>				4	4	2665.00	2014-10-10	PROCESSED
<input type="checkbox"/>				5	5	4440.00	2014-10-10	PROCESSED
<input type="checkbox"/>				6	6	3990.00	2014-10-10	PROCESSED

Figure 4.12: Inserting values to Transaction Table

➤ Unitsrate

```
INSERT INTO `unitsrate` ('sno', 'twohundred', 'fivehundred', 'thousand') VALUES(1, 2, 5, 10);
```

+ Options	sno	twohundred	fivehundred	thousand
	1	2	5	10

Figure 4.13: Inserting values to Unitsrate Table

➤ User

```
INSERT INTO `user` (`id`, `name`, `email`, `phone`, `pass`, `address`) VALUES(1, 'Sangeetha', 'sangeetha@gmail.com', 894321367, 'sangeetha', 'kengeri');
```

```
INSERT INTO `user` (`id`, `name`, `email`, `phone`, `pass`, `address`) VALUES(2, 'Abhishek ', 'abhishek@gmail.com', 89930333, 'abhishek', 'malleshwaram');
```

```
INSERT INTO `user` (`id`, `name`, `email`, `phone`, `pass`, `address`) VALUES(3, 'Sunena', 'sunena@gmail.com', 807271713, 'sunena', 'kengeri');
```

```
INSERT INTO `user` (`id`, `name`, `email`, `phone`, `pass`, `address`) VALUES(4, 'Bhumika', 'bhumika@gmail.com', 902133849, 'bhumika123', 'jaynagar');
```

```
INSERT INTO `user` (`id`, `name`, `email`, `phone`, `pass`, `address`) VALUES(5, 'Sharath', 'Sharath@gmail.com', 987644859, 'Sharath', 'jp nagar');
```

+ Options	← →	▼	id	name	email	phone	pass	address
<input type="checkbox"/>	Edit	Copy	1	Sangeetha	sangeetha@gmail.com	894321367	sangeetha	kengeri
<input type="checkbox"/>	Edit	Copy	2	Abhishek	abhishek@gmail.com	89930333	abhishek	malleshwaram
<input type="checkbox"/>	Edit	Copy	3	Sunena	sunena@gmail.com	807271713	sunena	kengeri
<input type="checkbox"/>	Edit	Copy	4	Bhumika	bhumika@gmail.com	902133849	bhumika	jaynagar
<input type="checkbox"/>	Edit	Copy	5	Sharath	Sharath@gmail.com	987644859	Sharath	jp nagar

Figure 4.14: Inserting values to User Table

4.3 Queries:

1. SELECT curdate() AS bdate , adddate(curdate(),INTERVAL 30 DAY) AS ddate , user.id AS uid , user.name AS uname FROM user;

Showing rows 0 - 7 (8 total, Query took 0.0048 seconds.)

```
SELECT curdate() AS bdate , adddate( curdate(),INTERVAL 30 DAY ) AS ddate , user.id AS uid , user.name AS uname FROM user
```

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

bdate	ddate	uid	uname
2021-01-08	2021-02-07	1	Sangeetha
2021-01-08	2021-02-07	2	Abhishek
2021-01-08	2021-02-07	3	Sunena
2021-01-08	2021-02-07	4	Bhumika
2021-01-08	2021-02-07	5	Sharath
2021-01-08	2021-02-07	6	Prema
2021-01-08	2021-02-07	7	Sanjana
2021-01-08	2021-02-07	9	Keerthi

2. SELECT count(*) FROM bill where uid=id;

Your SQL query has been executed successfully.

```
SELECT COUNT(*) FROM bill where uid=id
```

+ Options

COUNT(*)

0

3. SELECT * FROM transaction where status="pending";

```
Showing rows 0 - 0 (1 total, Query took 0.0021 seconds.)  
select * from transaction where status="pending"
```

Show all | Number of rows: 25 Filter rows: Search this table

4. SELECT * FROM complaint where uid=id;

```
Showing rows 0 - 0 (1 total, Query took 0.0023 seconds.)  
SELECT * FROM complaint where uid=id
```

Show all | Number of rows: 25 Filter rows: Search this table

+ Options

	id	uid	aid	complaint	status
<input type="checkbox"/> Edit Copy Delete	1	1	1	Transaction Not Processed	PROCESSED

5. SELECT COUNT(*) FROM bill where uid={\$id} AND status='PENDING';

```
Your SQL query has been executed successfully.  
SELECT COUNT(*) FROM bill where uid=id AND status='PENDING'
```

+ Options **COUNT(*)**
0

6. Select * from user where address = “bangalore”;

```

✓ Showing rows 0 - 1 (2 total, Query took 0.0033 seconds.)

select * from user where address="bangalore"

```

Show all | Number of rows: 25 Filter rows:

+ Options

id	name	email	phone	pass	address
9	Keerthi	keer@gmail.com	709057462	keerthi	Bangalore
10	rahul	rahul@gmail.com	987563214	rahul	Bangalore

7. SELECT bill.bdate AS bdate, bill.units AS units, bill.ddate AS ddate, transaction.payable AS payable, bill.amount AS amount ,transaction.payable-bill.amount AS dues , bill.id AS id FROM bill , transaction;

```

✓ Showing rows 0 - 24 (400 total, Query took 0.0074 seconds.)

SELECT bill.bdate AS bdate, bill.units AS units, bill.ddate AS ddate, transaction.payable AS payable,bill.amount AS amount ,transaction.payable-bill.amount AS dues , bill.id AS id FROM bill , transaction

```

Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

1 < > >> | Show all | Number of rows: 25 Filter rows:

+ Options

bdate	units	ddate	payable	amount	dues	id
2014-12-01	12	2014-12-31	1024.00	24.00	1000.00	1
2014-09-01	200	2014-10-01	1024.00	400.00	624.00	2
2014-09-01	103	2014-10-01	1024.00	206.00	818.00	3
2014-09-01	453	2014-10-01	1024.00	1665.00	-641.00	4
2014-09-01	654	2014-10-01	1024.00	3440.00	-2416.00	5
2014-09-01	609	2014-10-01	1024.00	2990.00	-1966.00	6
2014-10-01	435	2014-10-31	1024.00	1575.00	-551.00	7
2014-10-01	986	2014-10-31	1024.00	6760.00	-5736.00	8
2014-10-01	657	2014-10-31	1024.00	3470.00	-2446.00	9
2014-10-01	546	2014-10-31	1024.00	2360.00	-1336.00	10
2014-10-01	699	2014-10-31	1024.00	3890.00	-2866.00	11
2014-11-01	643	2014-12-01	1024.00	3330.00	-2306.00	12
2014-11-01	781	2014-12-01	1024.00	4710.00	-3686.00	13
2014-11-01	434	2014-12-01	1024.00	1570.00	-546.00	14
2014-11-01	235	2014-12-01	1024.00	575.00	449.00	15
2014-11-01	435	2014-12-01	1024.00	1575.00	-551.00	16

8. SELECT * FROM logs;

```

✓ Showing rows 0 - 1 (2 total, Query took 0.0019 seconds.)
select * from logs
 Prof

 Show all | Number of rows: 25 ▾ Filter rows: Search this table Sort by key: None ▾

+ Options
← → id uid action createdate
 Edit Copy Delete 1 9 NEW USER FOUND 2021-01-06 12:25:53
 Edit Copy Delete 3 10 NEW USER FOUND 2021-01-08 23:36:29

```

9. SELECT name from user u,bill b,transaction t where u.id=b.uid and b.id=t.id and amount>3000;

```

✓ Showing rows 0 - 5 (6 total, Query took 0.0102 seconds.)
SELECT name from user u,bill b,transaction t where u.id=b.uid and b.id=t.id and amount>3000
 Prof

 Show all | Number of rows: 25 ▾ Filter rows: Search this table

+ Options
name
Sangeetha
Abhishek
Sunena
Bhumika
Bhumika
Sharath

```

10.SELECT * FROM bill where amount<1000;

The screenshot shows a MySQL query results page. At the top, a green bar indicates "Showing rows 0 - 4 (5 total, Query took 0.0154 seconds.)". Below this, the SQL query "SELECT * FROM `bill` where amount<1000" is displayed. There are buttons for "Profiling [Edit ini]" and "Show all" (unchecked). A dropdown menu shows "Number of rows: 25" and a "Sort by key: None" dropdown. A search bar says "Filter rows: Search this table". The main area displays a table with columns: id, aid, uid, units, amount, status, bdate, and ddate. The data is as follows:

	<input type="checkbox"/>	Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	id	aid	uid	units	amount	status	bdate	ddate
1	<input type="checkbox"/>	Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	1	1	2	12	24.00	PROCESSED	2014-12-01	2014-12-31
2	<input type="checkbox"/>	Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	2	1	1	200	400.00	PROCESSED	2014-09-01	2014-10-01
3	<input type="checkbox"/>	Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	3	1	2	103	206.00	PROCESSED	2014-09-01	2014-10-01
15	<input type="checkbox"/>	Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	15	1	5	235	575.00	PROCESSED	2014-11-01	2014-12-01
17	<input type="checkbox"/>	Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	17	2	1	300	900.00	PENDING	2021-01-09	2021-02-08

4.4 Front-End Details:

PHP:

PHP known as “Hypertext Preprocessor”. It is a server-side scripting language usually written in an HTML context. Unlike an ordinary HTML page, a PHP script is not sent directly to a client by the server; instead, it is parsed by the PHP binary or module, which is server-side installed. HTML elements in the script are left alone, but PHP code is interpreted and executed. PHP code in a script can query databases, create images, read and write files, talk to remote servers – the possibilities is endless. The output from PHP code is combined with the HTML in the script and the result sent to the user’s web-browser, therefore it can never tell the user whether the web-server uses PHP or not, because the entire browser sees is HTML.

PHP’s support for Apache and MySQL .Apache is now the most-used web-server in the world, and PHP can be compiled as an Apache module. MySQL is a powerful free SQL database, and PHP provides a comprehensive set of functions for working with it.

PHP supports an extensive list of databases and web-servers.

Basic PHP Syntax :

A PHP scripting block always starts with <?php and ends with ?>. A PHP scripting block can be placed anywhere in the document.

A PHP file normally contains HTML tags, just like an HTML file, and some PHP scripting code.

HTML:

HTML or Hyper Text Markup Language is the standard markup language used to create web pages. HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent empty elements and so are unpaired, for example . The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags). The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language.

CSS:

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content.

JAVASCRIPT:

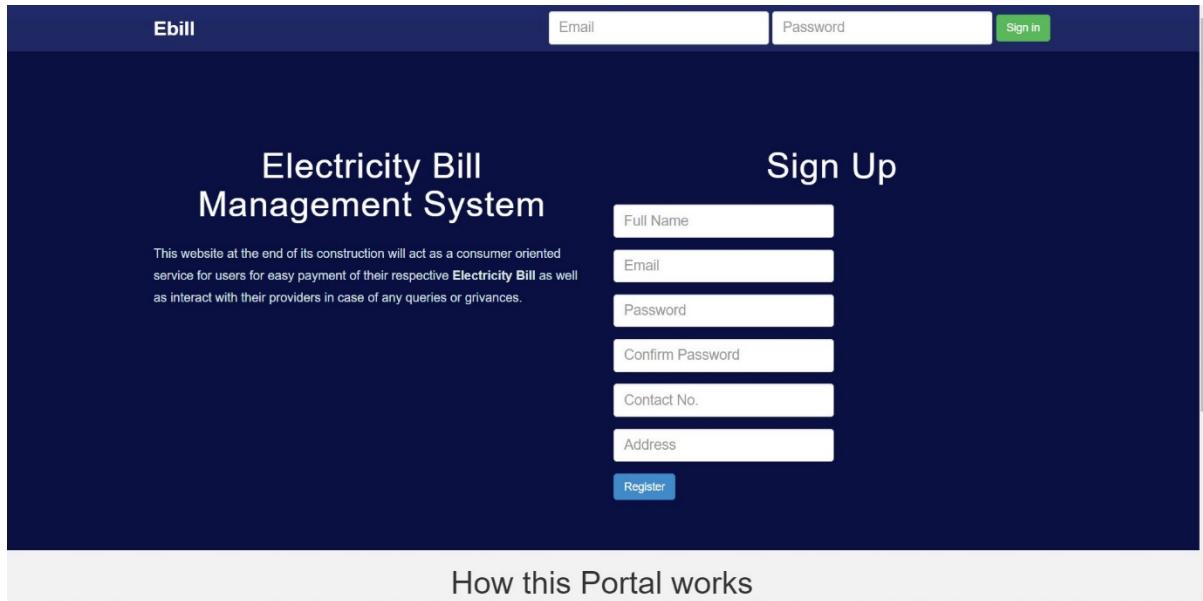
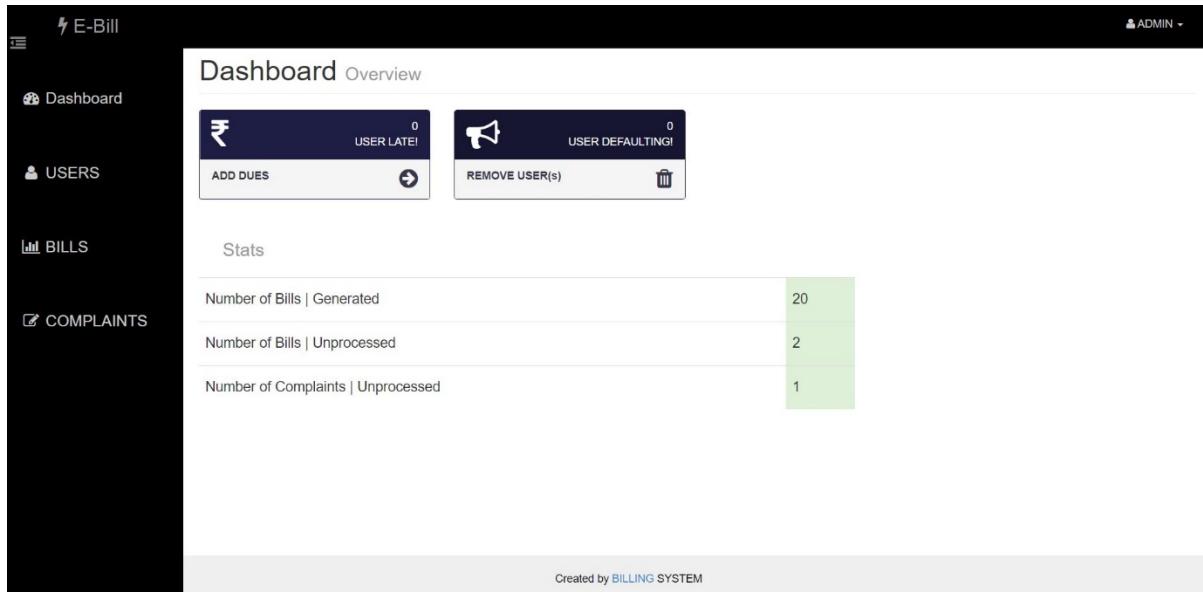
JavaScript (JS) is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side network programming (with Node.js), game development and the creation of desktop and mobile applications.

BOOTSTRAP:

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

Bootstrap is a HTML, CSS & JS Library that focuses on simplifying the development of informative web pages .The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking.

The most prominent components of Bootstrap are its layout components, as they affect an entire web page. The basic layout component is called "Container", as every other element in the page is placed in it. Developers can choose between a fixed-width container and a fluid-width container.

CHAPTER 5:**RESULTS AND DISCUSSIONS****SNAPSHOTS OF WEBPAGES:****Figure 5.1:** Home page**Figure 5.2:** Admin page

ELECTRICITY BILL MANAGEMENT SYSTEM

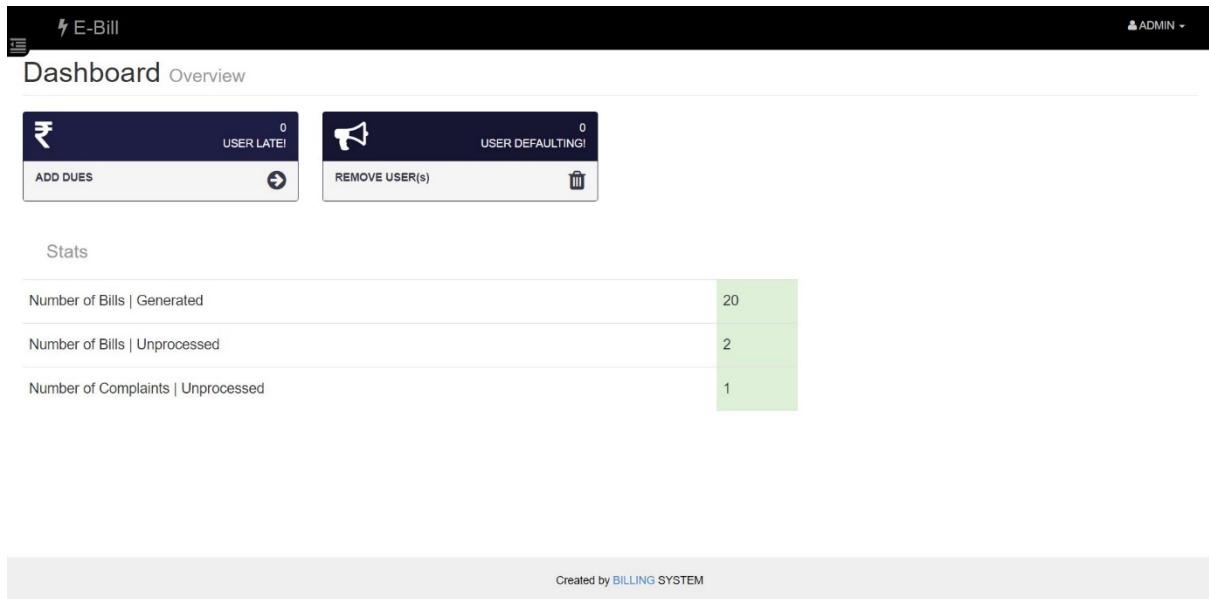


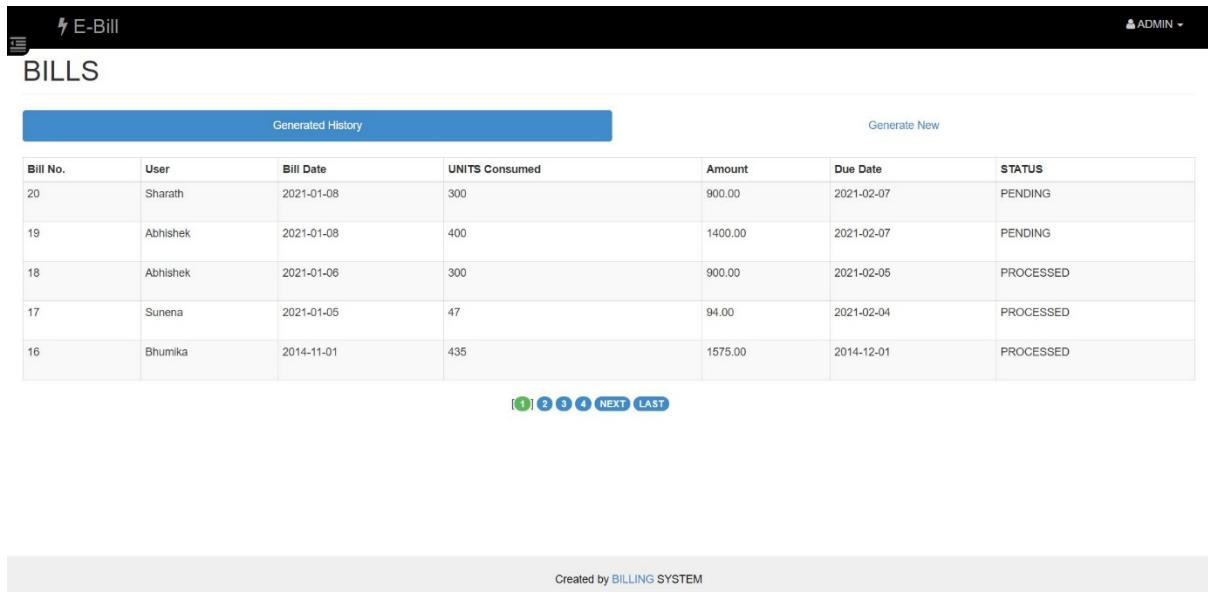
Figure 5.3: Admin Dashboard

The screenshot shows the 'USERS Details' page. The header includes the 'E-Bill' logo and 'ADMIN' access. The main content displays a table of user details with columns: User No., User Name, EMAIL, PHONE NO, and ADDRESS. There are five entries in the table, each corresponding to a user from 1 to 5. At the bottom of the table, there are navigation links for 'NEXT' and 'LAST' pages, with the current page number '2' highlighted. A footer at the bottom indicates the system was 'Created by BILLING SYSTEM'.

User No.	User Name	EMAIL	PHONE NO	ADDRESS
1	Sangeetha	sangeetha@gmail.com	894321367	kengeri
2	Abhishek	abhishek@gmail.com	89930333	malleshwaram
3	Sunena	sunena@gmail.com	807271713	kengeri
4	Bhumika	bhumika@gmail.com	902133849	Jaynagar
5	Sharath	Sharath@gmail.com	987644859	JP Nagar

Figure 5.4: User Details

ELECTRICITY BILL MANAGEMENT SYSTEM



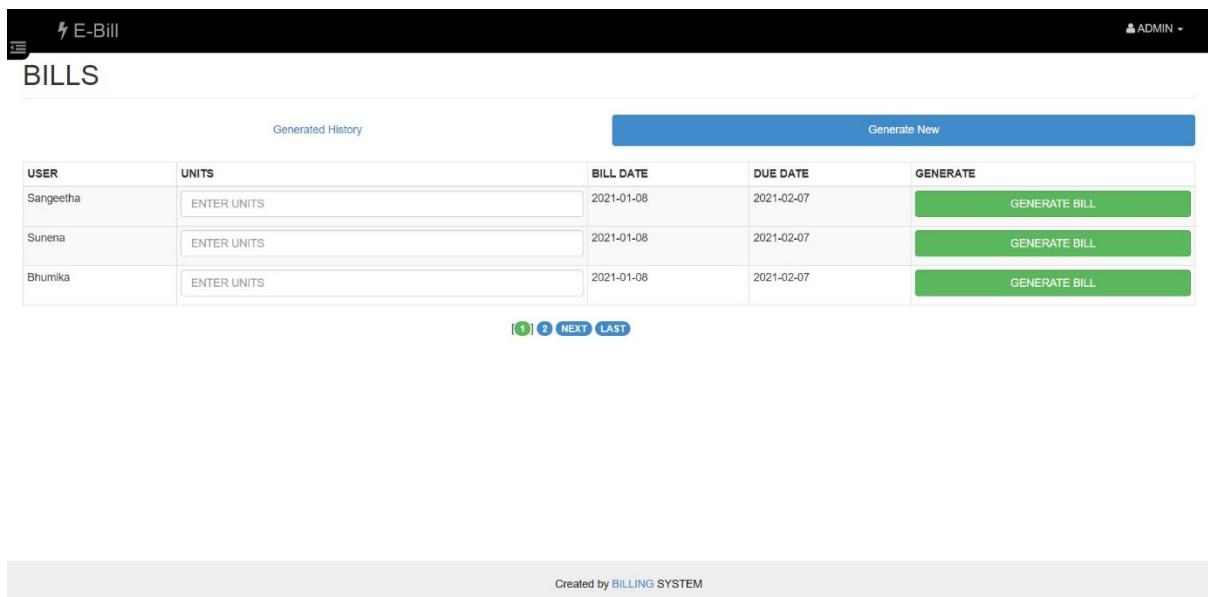
The screenshot shows a table titled 'Generated History' with the following data:

Bill No.	User	Bill Date	UNITS Consumed	Amount	Due Date	Status
20	Sharath	2021-01-08	300	900.00	2021-02-07	PENDING
19	Abhishek	2021-01-08	400	1400.00	2021-02-07	PENDING
18	Abhishek	2021-01-06	300	900.00	2021-02-05	PROCESSED
17	Sunena	2021-01-05	47	94.00	2021-02-04	PROCESSED
16	Bhumika	2014-11-01	435	1575.00	2014-12-01	PROCESSED

[1] [2] [3] [4] [NEXT] [LAST]

Created by BILLING SYSTEM

Figure 5.5: Generated bill history



The screenshot shows a table titled 'Generate New' with the following data:

USER	UNITS	BILL DATE	DU DATE	GENERATE
Sangeetha	ENTER UNITS	2021-01-08	2021-02-07	GENERATE BILL
Sunena	ENTER UNITS	2021-01-08	2021-02-07	GENERATE BILL
Bhumika	ENTER UNITS	2021-01-08	2021-02-07	GENERATE BILL

[1] [2] [NEXT] [LAST]

Created by BILLING SYSTEM

Fig.5.6: Generating new bill

Complaint No.	User	COMPLAINT	STATUS	Process
14	Abhishek	Bill Not Correct	NOT PROCESSED	<button>PROCESS COMPLAINT</button>

Figure 5.7: Complaints page

Number of Bills PAYED	3
Number of Bills PENDING	0
Number of Complaints Unprocessed	0

Figure 5.8: User page

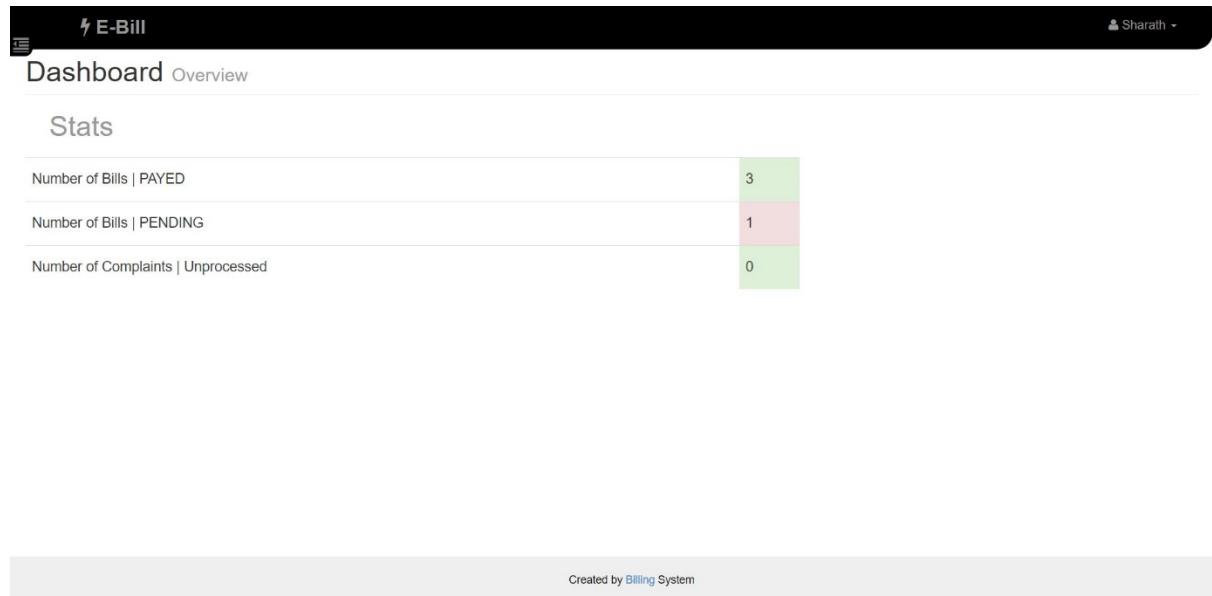


Figure 5.9: User dashboard

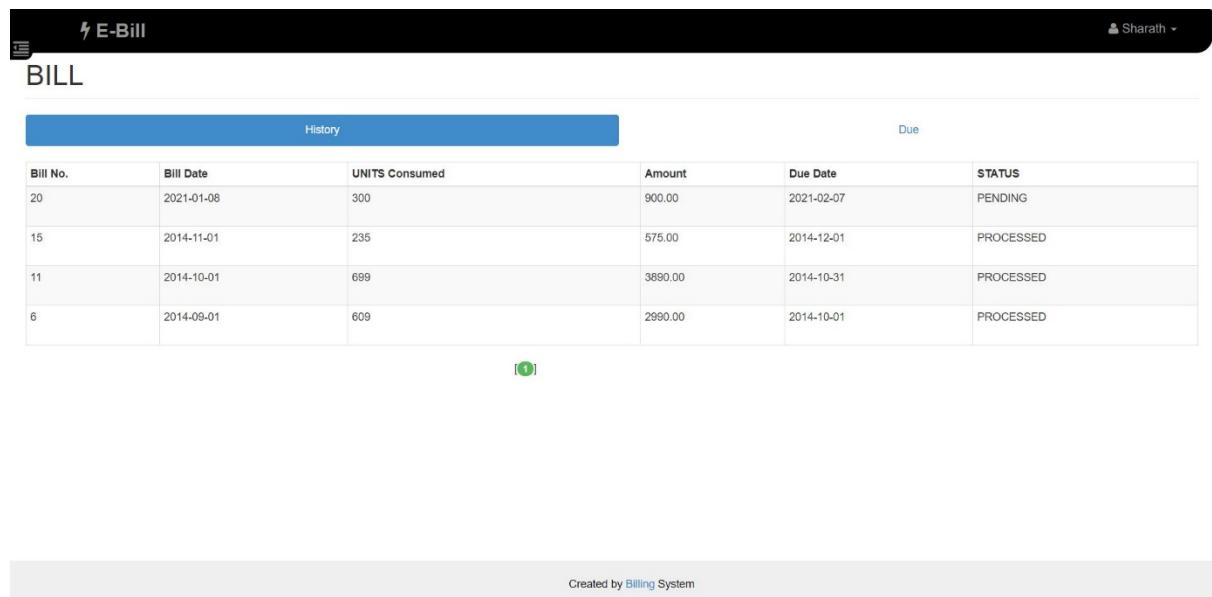


Figure 5.10: User bill history

ELECTRICITY BILL MANAGEMENT SYSTEM

The screenshot shows the 'Transaction' section of the system. At the top, there's a header with the E-Bill logo and a user profile for 'Sharath'. Below the header, the title 'Transaction' is displayed above a breadcrumb navigation bar ('Transaction / History'). A table lists transaction details:

Transaction No.	Bill Date	Amount	Dues (if any)	Final Amount Payed	Transaction Date
-	2021-01-08	900.00	0.00	900.00	TRANSACTION PENDING
15	2014-11-01	575.00	0.00	575.00	2014-09-18
11	2014-10-01	3890.00	0.00	3890.00	2014-10-10
6	2014-09-01	2990.00	1000.00	3990.00	2014-10-10

A green circular icon with the number '1' is located at the bottom right of the table. At the bottom of the page, a footer bar indicates it was 'Created by Billing System'.

Figure 5.11: User transaction history

The screenshot shows the 'Complaint' section of the system. At the top, there's a header with the E-Bill logo and a user profile for 'Sharath'. Below the header, the title 'Complaint' is displayed above a breadcrumb navigation bar ('Complaint / History'). A blue button labeled 'New Complaint' is visible. A table lists complaint details:

Complaint No.	Complaint	STATUS
11	Bill Generated Late	PROCESSED

A green circular icon with the number '1' is located at the bottom right of the table. At the bottom of the page, a footer bar indicates it was 'Created by Billing System'.

Figure 5.12: User Complaint history

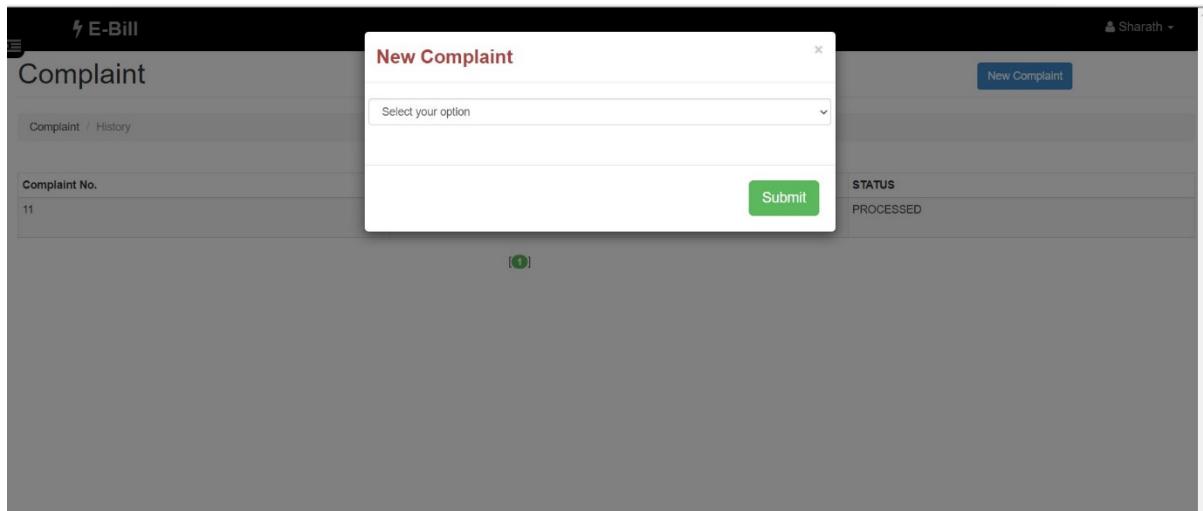


Figure 5.13: Generating new complaint

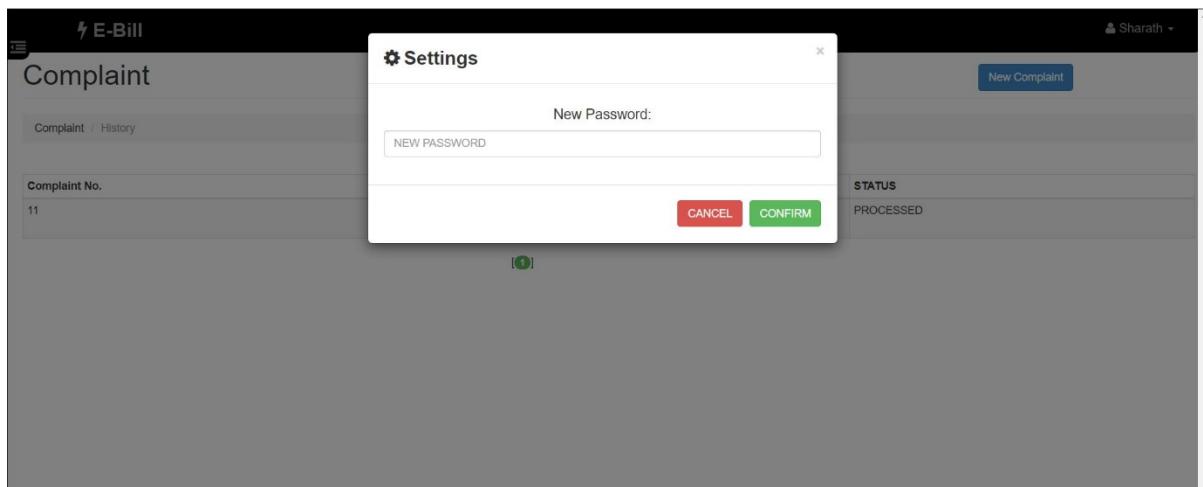


Figure 5.14: User change password



Figure 5.15: User logout

CONCLUSION AND FUTURE SCOPE:

The Project “**Electricity bill Management System**” is designed to provide usage of electricity by customer in one month by Inserting, Retrieving and Updating the Details are easy when it is compared to the manual update and storing. This project reports, before suggesting the solution they were using a manual process on meter reading, amount calculation, billing customer and so on. The interaction between customer and Electricity Board was very poor and it took much longer to respond to customer queries. The User Interface of it is very friendly and can be easily used by anyone. This project helps to store usage of electricity by all the customer in electricity board in organized manner and to replace old paper work system. It also provides knowledge about the latest technology used in developing web enable application and client-server technology that will be great demand in future. Further we can give more advance software for Electricity Billing System including more facilities. We can also host the platform on online servers to make it accessible worldwide. Also we can implement the mechanism for taking backup of codebase and database on regular basis on different servers.

REFERENCES

Web references

- www.php.net
- www.mysql.com
- wwwxamppserver.com
- www.w3schools.com
- www.stackoverflow.com

Book references

- PHP: the Complete Reference by Steven Holzner
- PHP & MySQL Web development by Luke Welling & Laura Thompson
- DBMS: Ramakrishnan - Database Management Systems 3rd Edition