**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

"Jnana Sangama", Belagavi: 590 018



Database Management Systems Mini Project report on

**“XXXXXXXXXXXX”**

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

**BACHELOR OF ENGINEERING**

**IN**

**ARTIFICIAL INTELLIGENCE & MACHINE LEARNING**

By

**XXXXXXXXXX 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 **XXXXXXXXXXXXX** is a bonafide work carried out by **XXXXXXXX (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

1. Implementation 12

5.1 Database Schema

5.2 Table Design

1. Testing 13
2. 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 : >=1GB
* 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:**

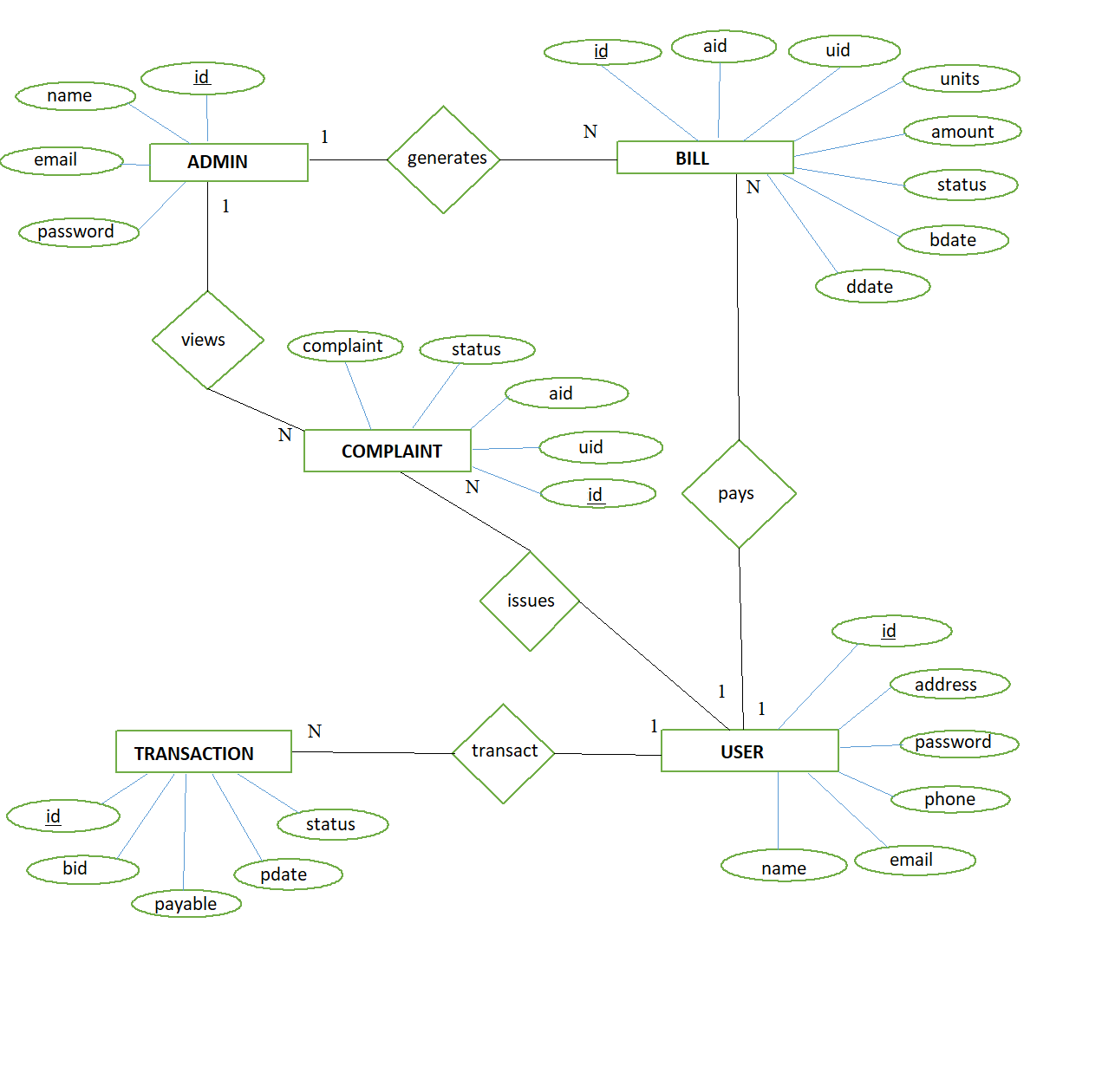
* Realibilty: Database updating should follow transaction processing to avoid data inconsistency.
* Availabilty: The project will be deployed on a public shared server so it will be available all the time and will be accessible anywhere of the world using internet.
* Security: We have implemented a lot of security mechanism to avoid to hack the system by outer world.
* Maintainabilty: 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 Compatibilty: The project being web based required compatibility with at least the popular web browsers. Microsft 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:

**ADMIN**

|  |  |  |  |
| --- | --- | --- | --- |
| Id | name | email | password |

**BILL**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id | aid | Uid | units | amount | status | bdate | ddate |

**COMPLAINT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Uid | aid | status | complaint |

**TRANSACTION**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | bid | payable | pdate | status |

**USER**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | name | email | phone | password | address |

**LOGS**

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Uid | action | Createdate |

**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.

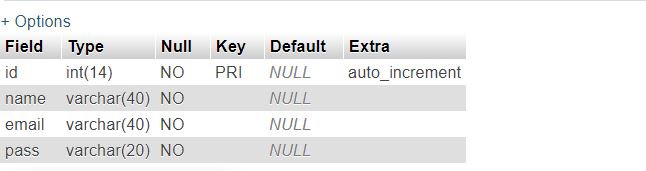
MySQL is multithreaded, multi user SQL database management System (DBMS). The basic program run as server providing multiuser access to a number of databases. MySQL is a database. The data in a MySQL is stored in a Database objects called tables. A table is a collection of related data entries and it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

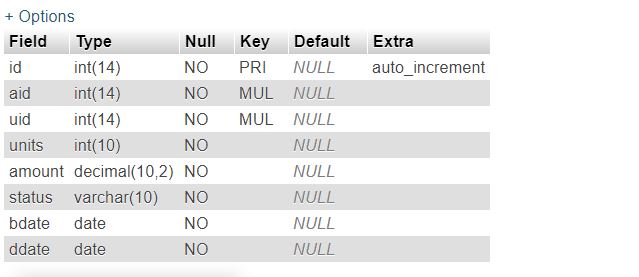
CREATE TABLE admin (id int(14) NOT NULL, name varchar(40) NOT NULL, email varchar(40) NOT NULL, pass varchar(20) NOT NULL);



**Figure 4.1: Creation of Admin Table**

* Bill

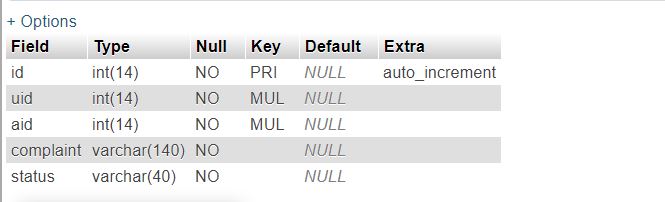
CREATE TABLE bill (id int(14) NOT NULL, aid int(14) NOT NULL, uid int(14) NOT NULL, units int(10) NOT NULL, amount decimal(10,2) NOT NULL, status varchar(10) NOT NULL, bdate date NOT NULL, ddate date NOT NULL);



**Figure 4.2: Creation of User Table**

* Complaint

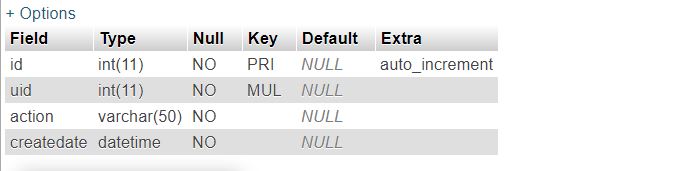
CREATE TABLE complaint (id int(14) NOT NULL, uid int(14) NOT NULL, aid int(14) NOT NULL, complaint varchar(140) NOT NULL, status varchar(40) NOT NULL);



**Figure 4.3: Creation of Complaint Table**

* Logs

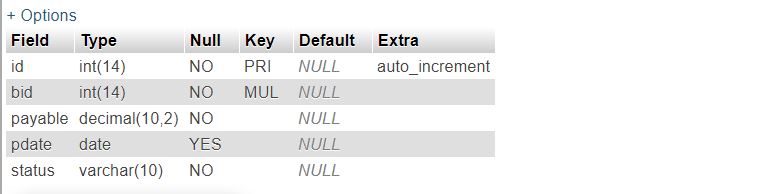
CREATE TABLE logs (id int(11) NOT NULL, uid int(11) NOT NULL,action varchar(50) NOT NULL, createdate datetime NOT NULL);



**Figure 4.4: Creation of Logs Table**

* Transaction

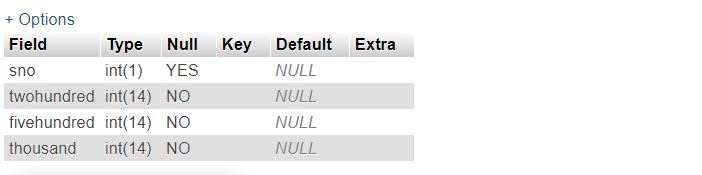
CREATE TABLE transaction (id int(14) NOT NULL, bid int(14) NOT NULL, payable decimal(10,2) NOT NULL, pdate date DEFAULT NULL, status varchar(10) NOT NULL);



**Figure 4.5: Creation of Transaction Table**

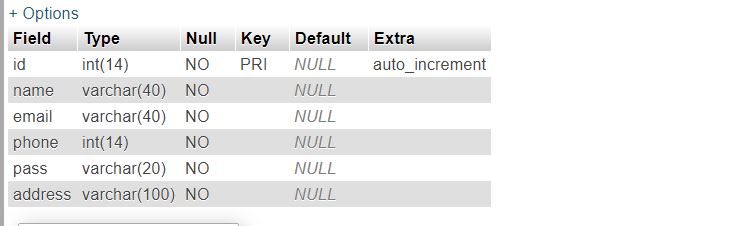
* Unitsrate

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



**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); 

**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');



**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');

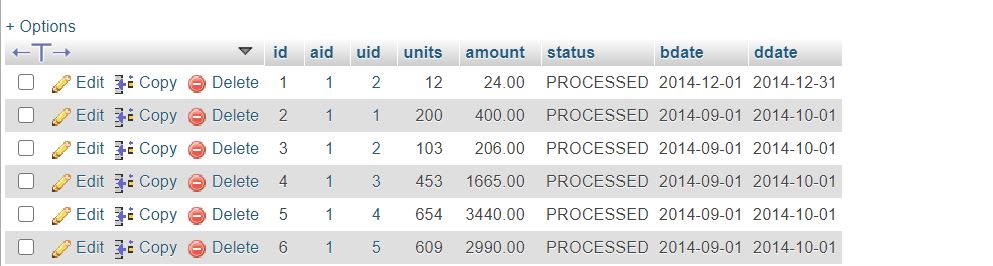
INSERT INTO `bill` (`id`, `aid`, `uid`, `units`, `amount`, `status`, `bdate`, `ddate`) VALUES(2, 1, 1, 200, '400.00', 'PROCESSED', '2014-09-01', '2014-10-01');

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');



**Figure 4.9: Inserting values to the Bill Table**

* Complaint

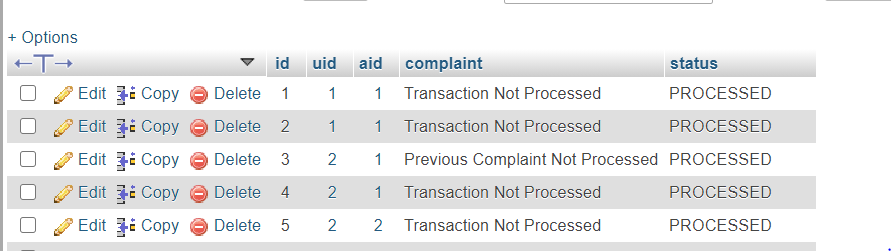
INSERT INTO `complaint` (`id`, `uid`, `aid`, `complaint`, `status`) VALUES(1, 1, 1, 'Transaction Not Processed', 'PROCESSED');

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');



**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');



**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');



**Figure 4.12: Inserting values to Transaction Table**

* Unitsrate

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



**Figure 4.13: Inserting values to Unitsrate Table**

* User

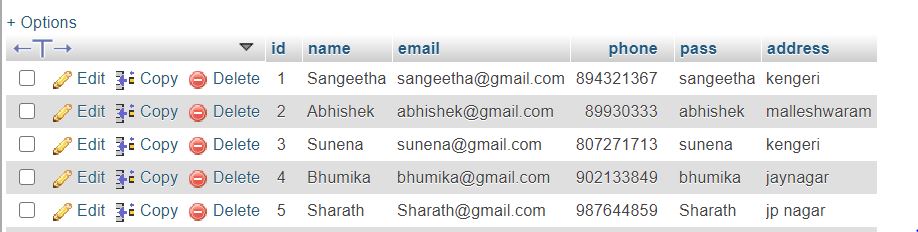
INSERT INTO `user` (`id`, `name`, `email`, `phone`, `pass`, `address`) VALUES(1, 'Sangeetha', 'sangeetha@gmail.com', 894321367, 'sangeetha', 'kengeri');

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');

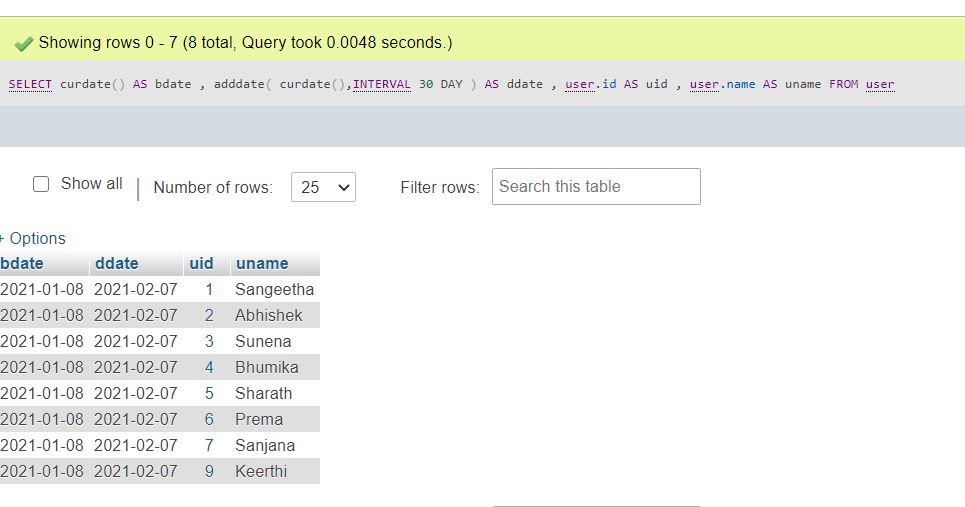
INSERT INTO `user` (`id`, `name`, `email`, `phone`, `pass`, `address`) VALUES(5, 'Sharath', 'Sharath@gmail.com', 987644859, 'Sharath', 'jp nagar');



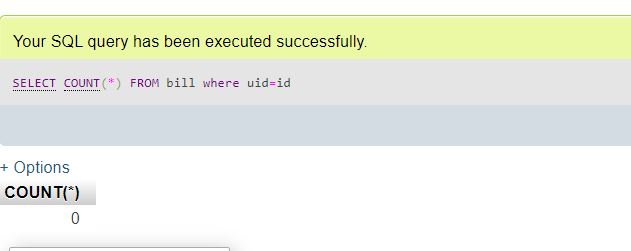
**Figure 4.14: Inserting values to User Table**

**4.3 Queries:**

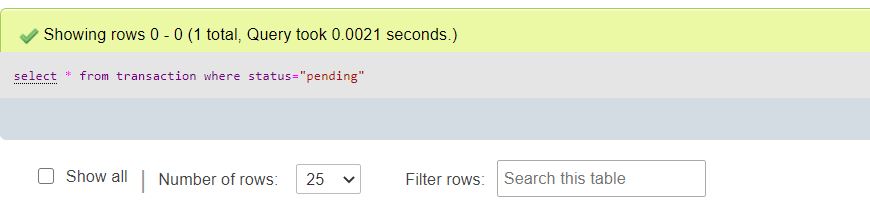
1. SELECT curdate() AS bdate , adddate( curdate(),INTERVAL 30 DAY ) AS ddate , user.id AS uid , user.name AS uname FROM user;



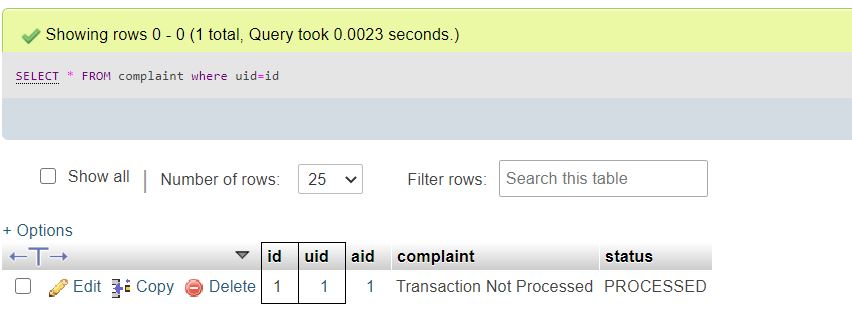
2. SELECT count(\*) FROM bill where uid=id;



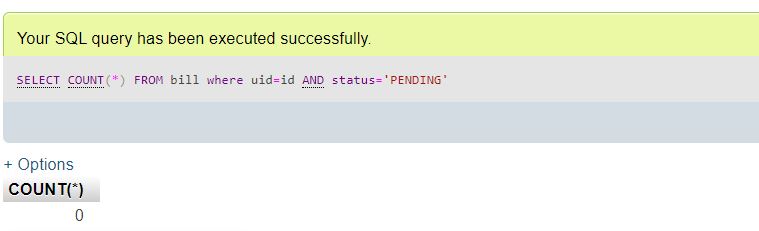
3. SELECT \* FROM transaction where status=”pending”;



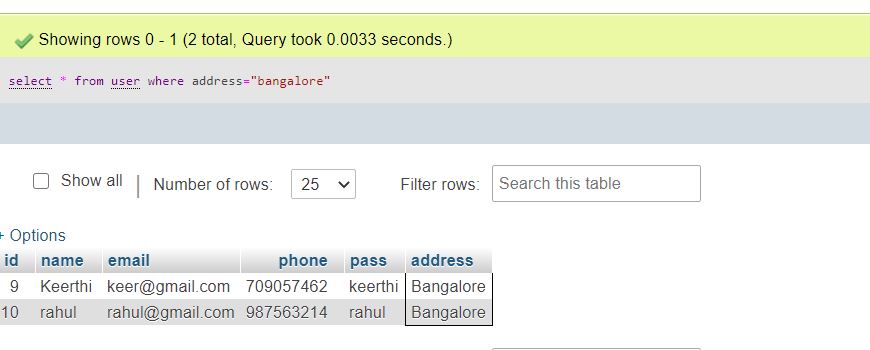
4. SELECT \* FROM complaint where uid=id;



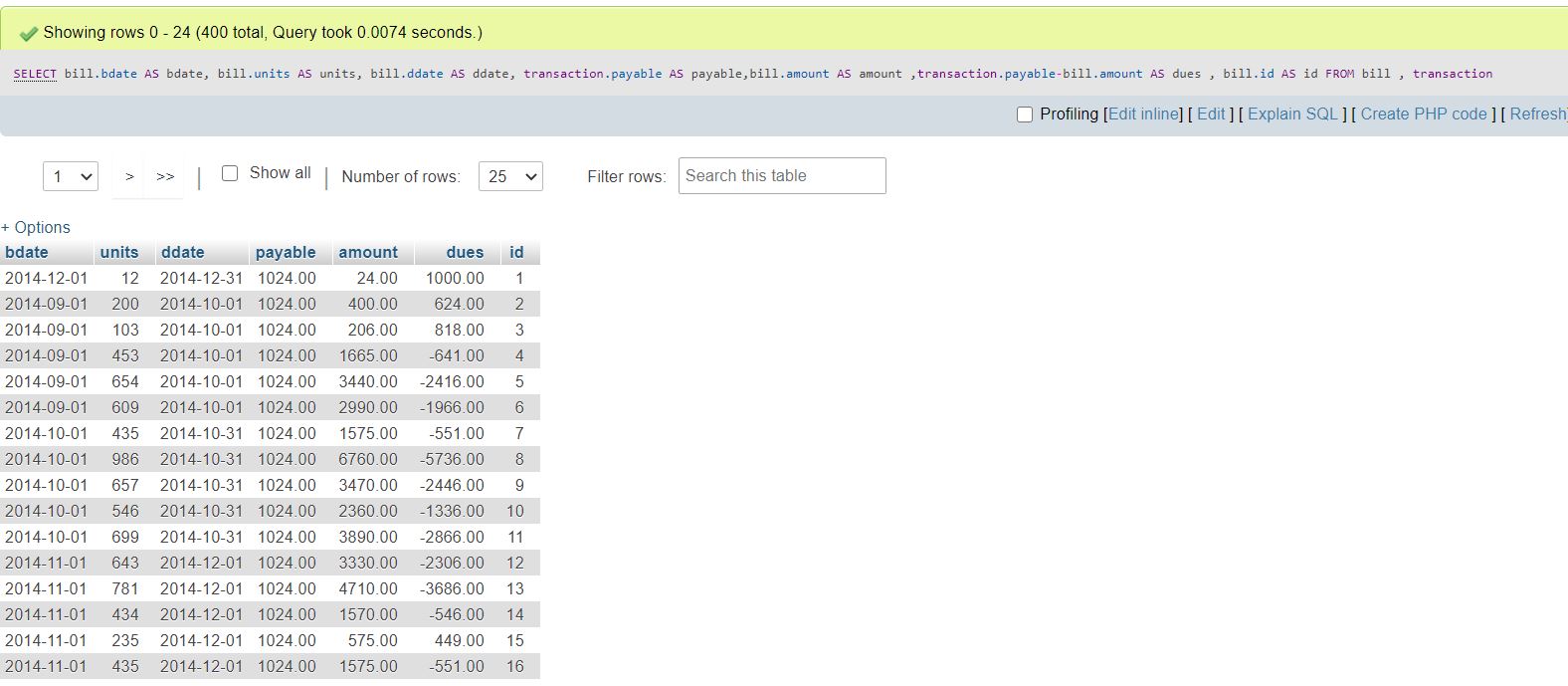
5. SELECT COUNT(\*) FROM bill where uid={$id} AND status='PENDING';



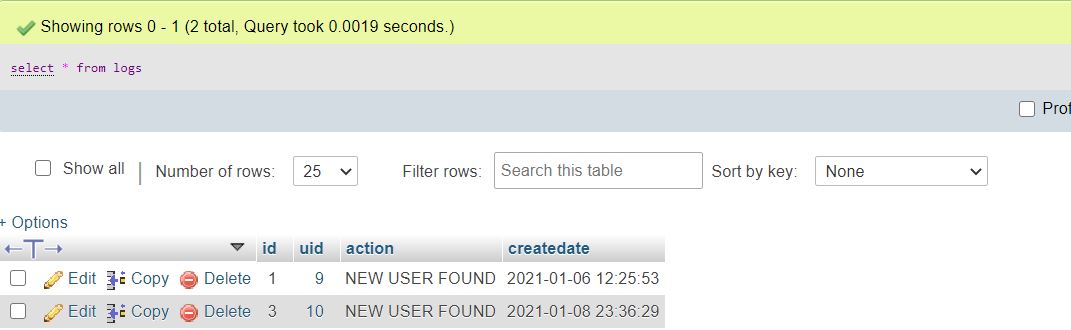
6. Select \* from user where address = “bangalore”;



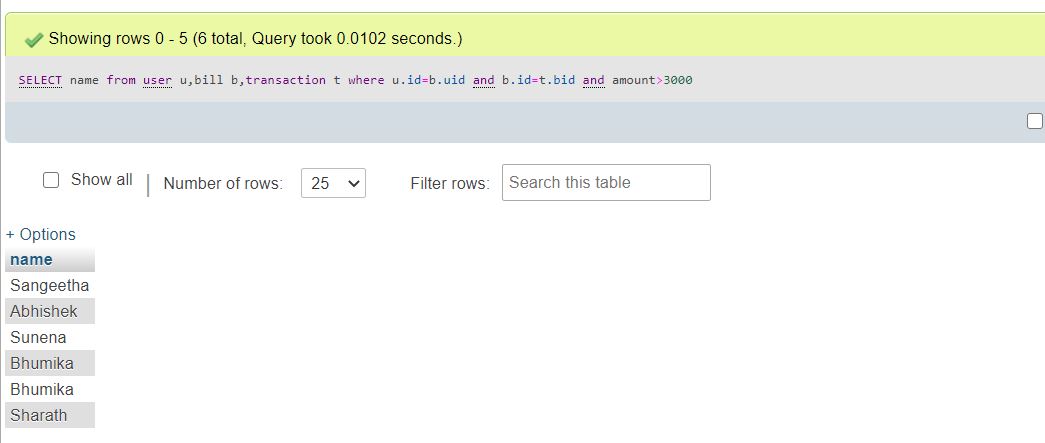
7. SELECT bill.bdate AS bdate, bill.units AS units, bill.ddate AS ddate, transaction.payable AS payable, bill.amount AS amount ,transaction.payable-bill.amount AS dues , bill.id AS id FROM bill , transaction;



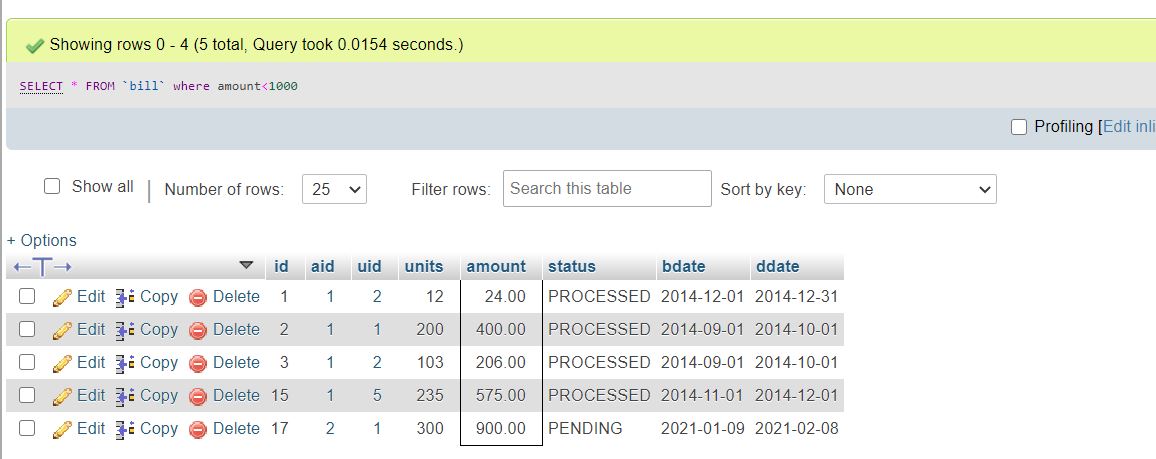
8. SELECT \* FROM logs;



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



10.SELECT \* FROM bill where amount<1000;



**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 <img>. 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.

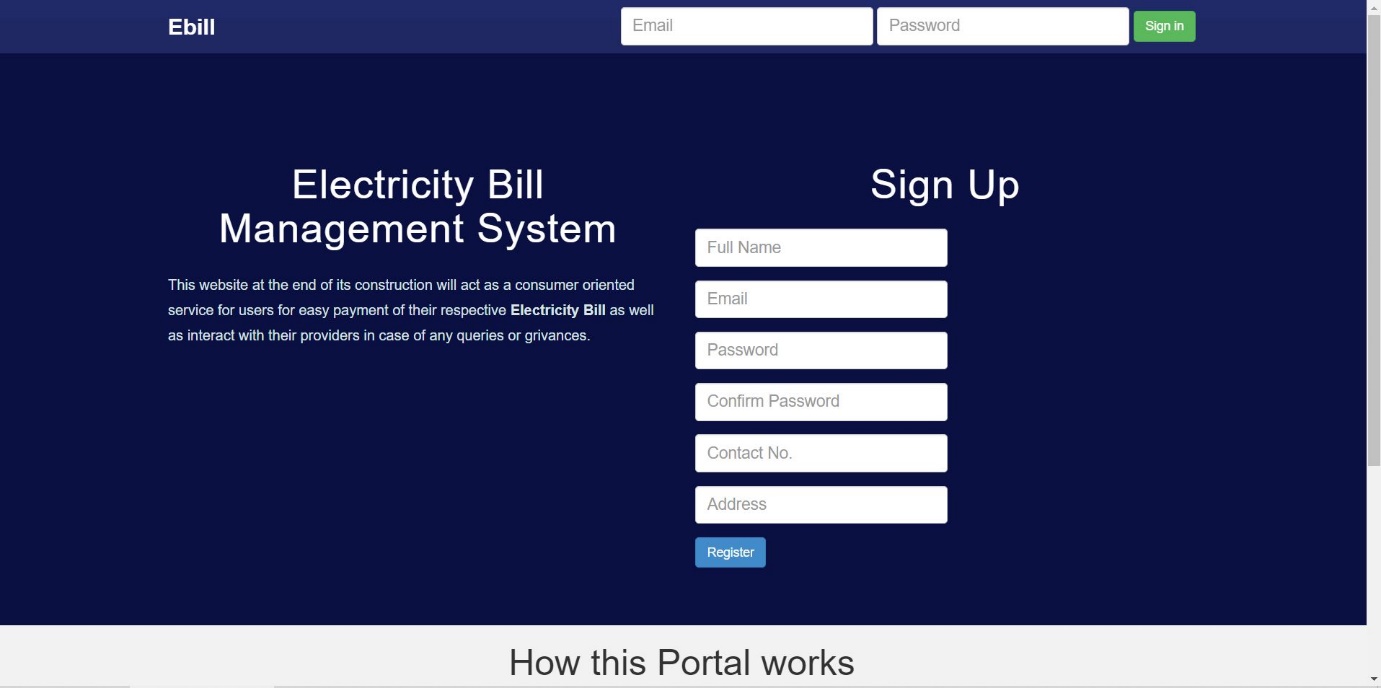
Bootstrap is a HTML, CSS & JS Library that focuses on simplifying the development of informative web pages .The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge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

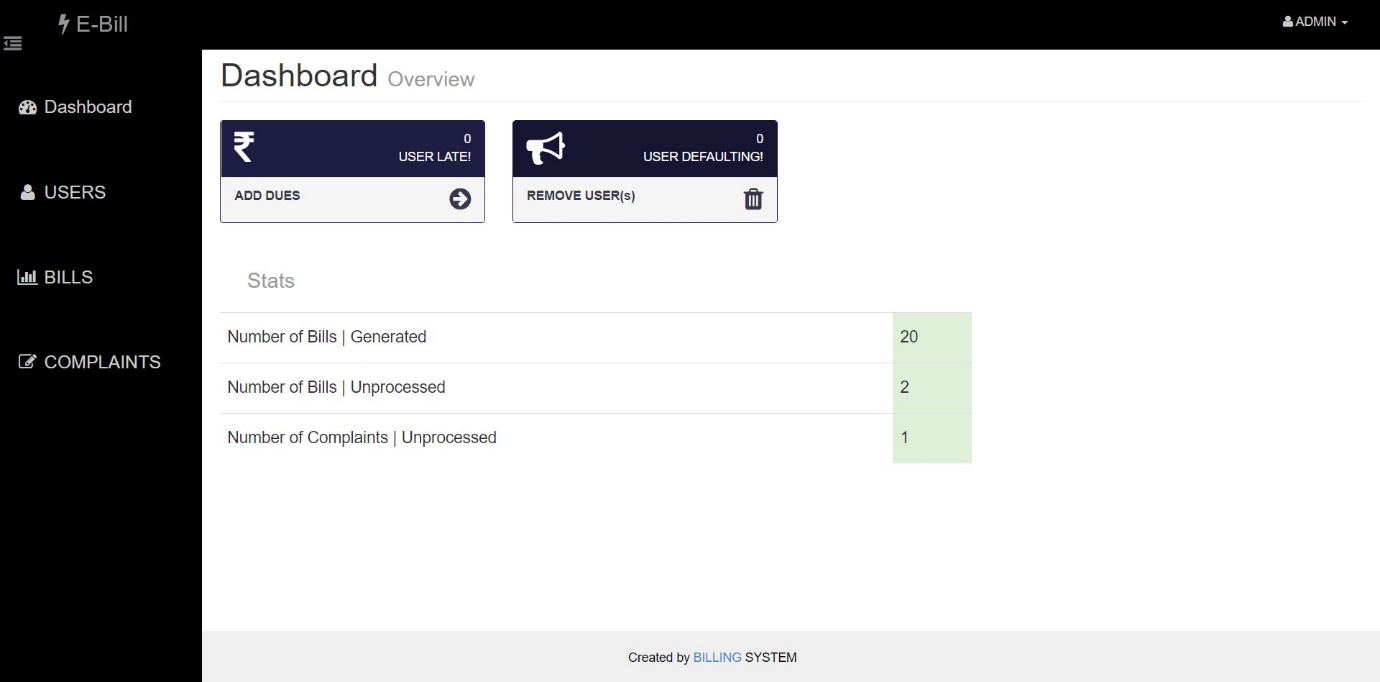
**CHAPTER 5:**

**RESULTS AND DISCUSSIONS**

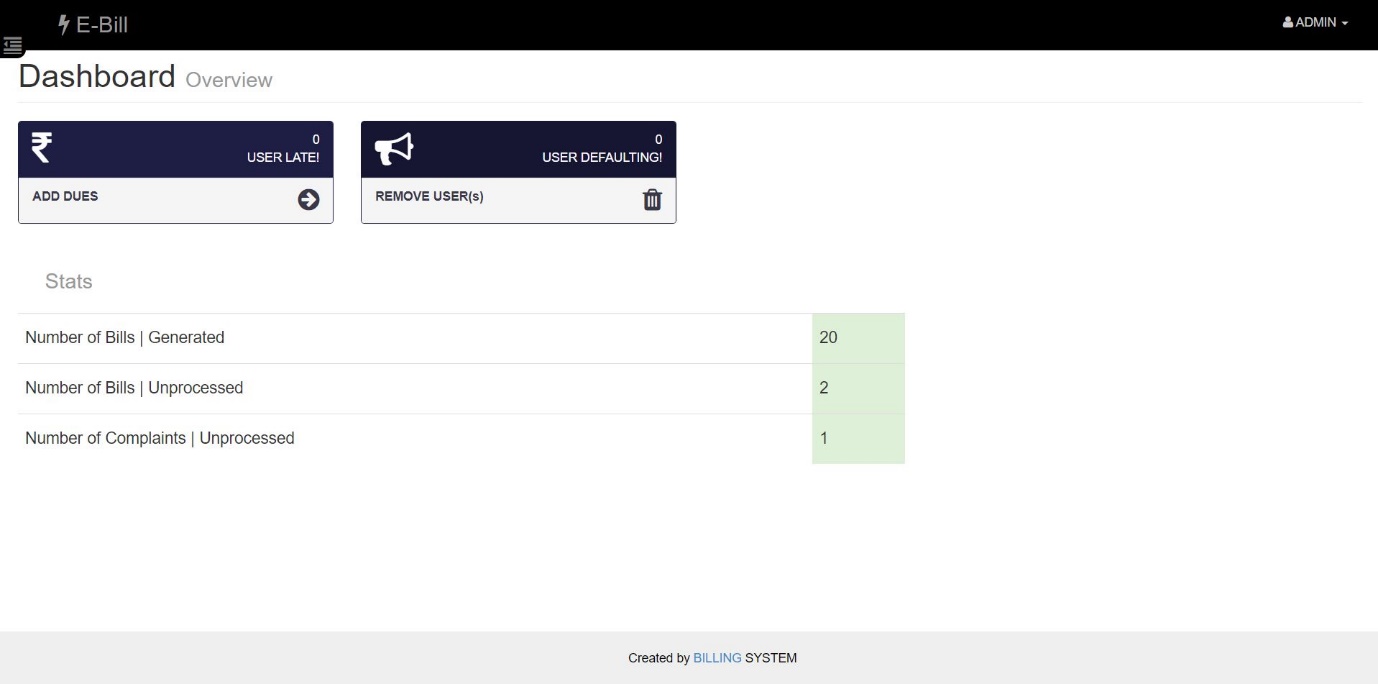
**SNAPSHOTS OF WEBPAGES:**

****

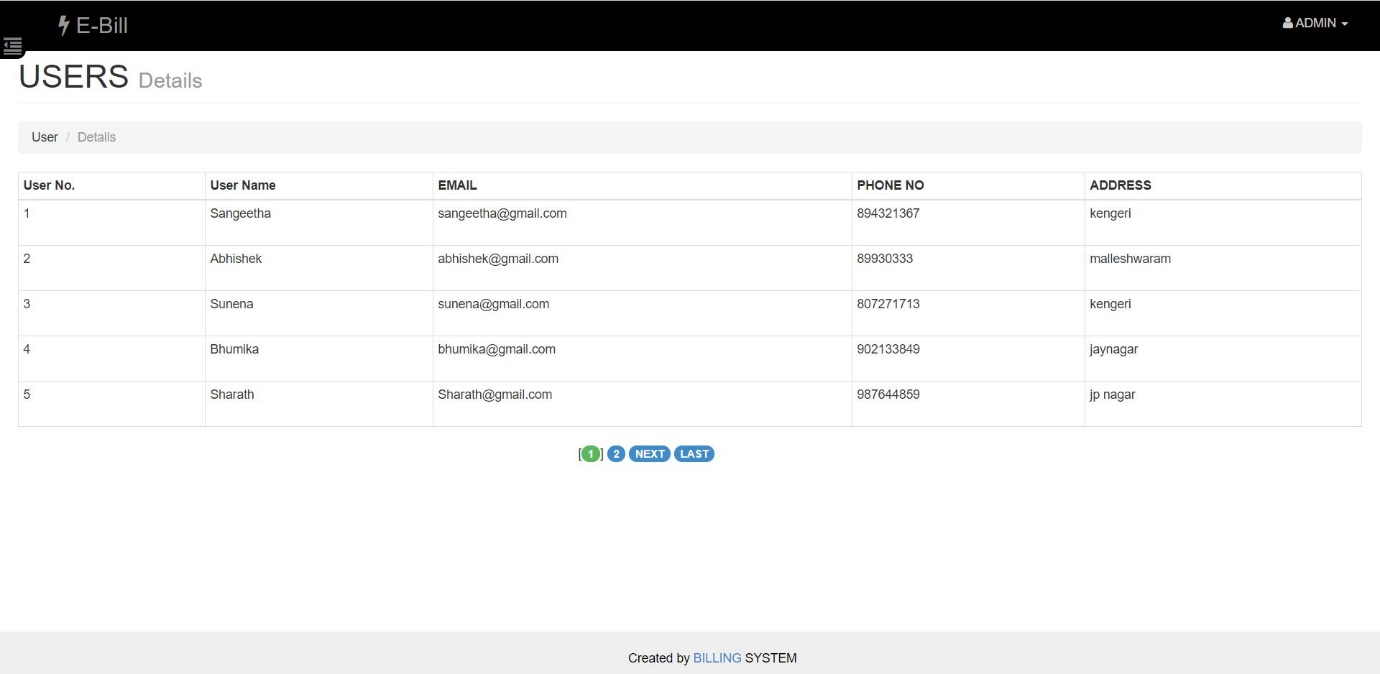
**Figure 5.1:** Home page

****

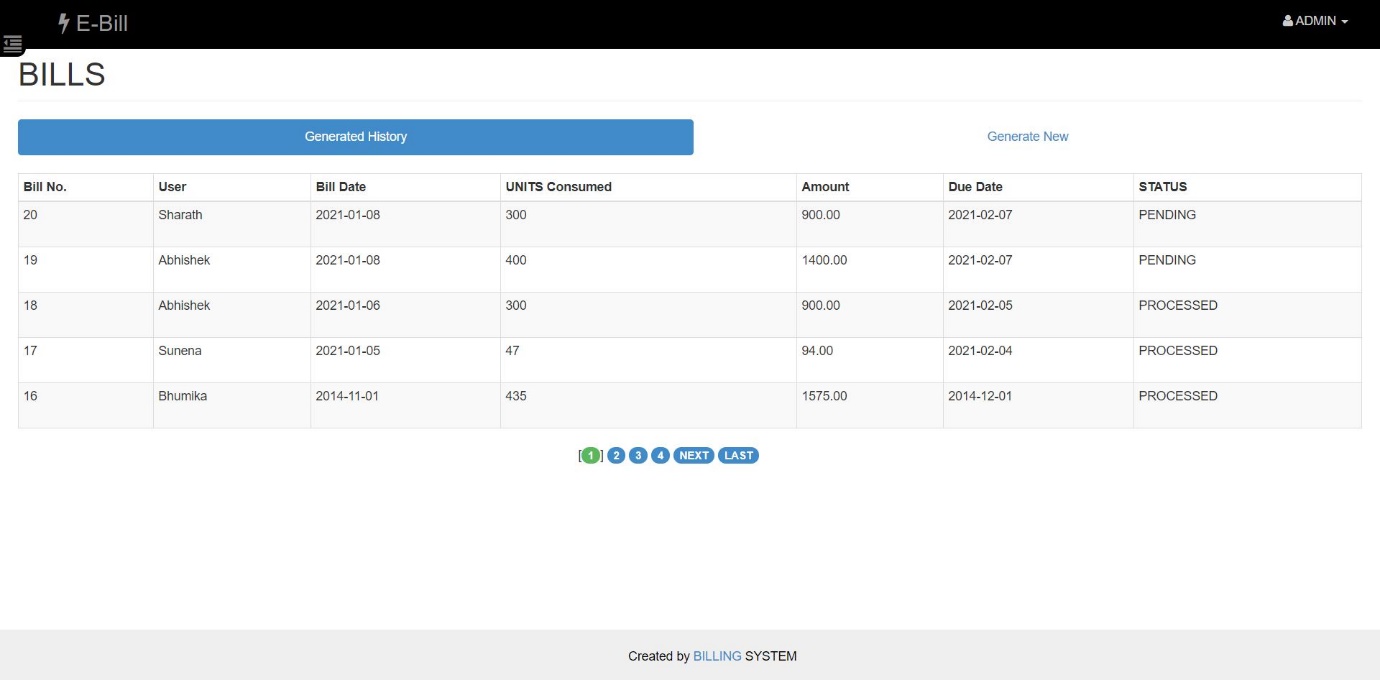
**Figure 5.2:** Admin page



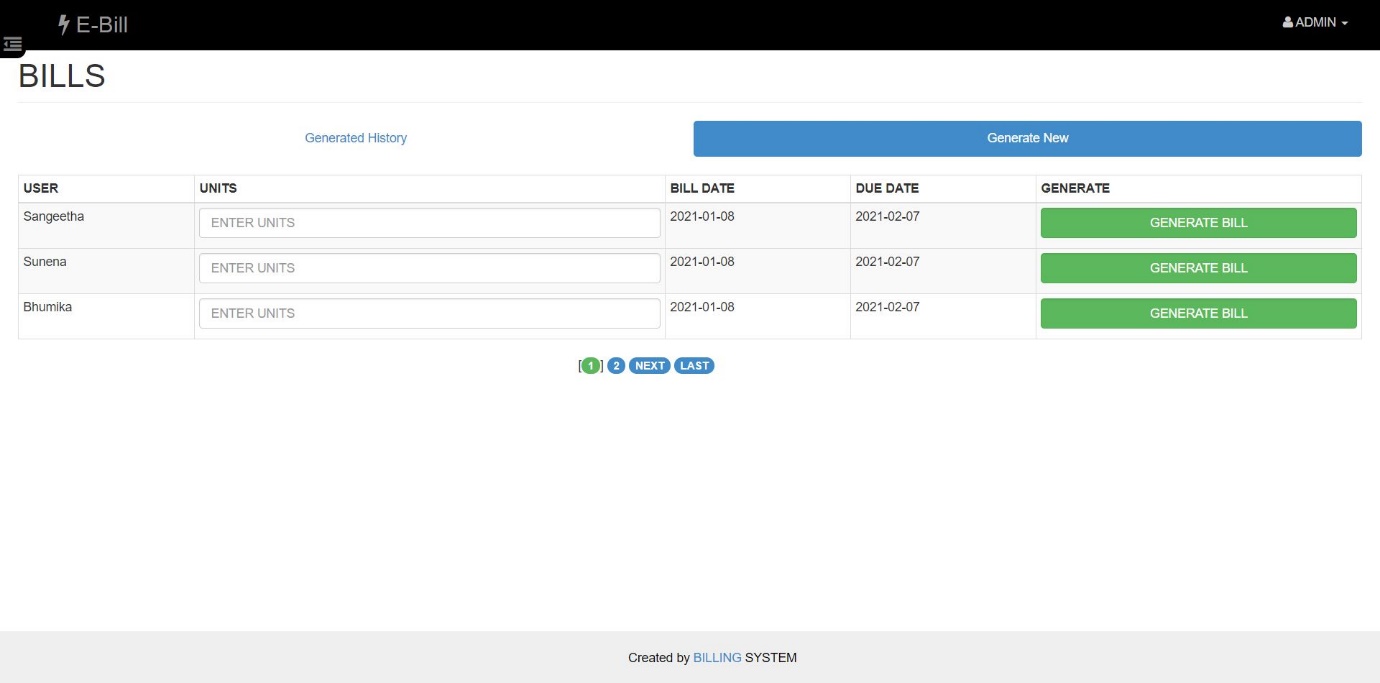
**Figure 5.3:** Admin Dashboard



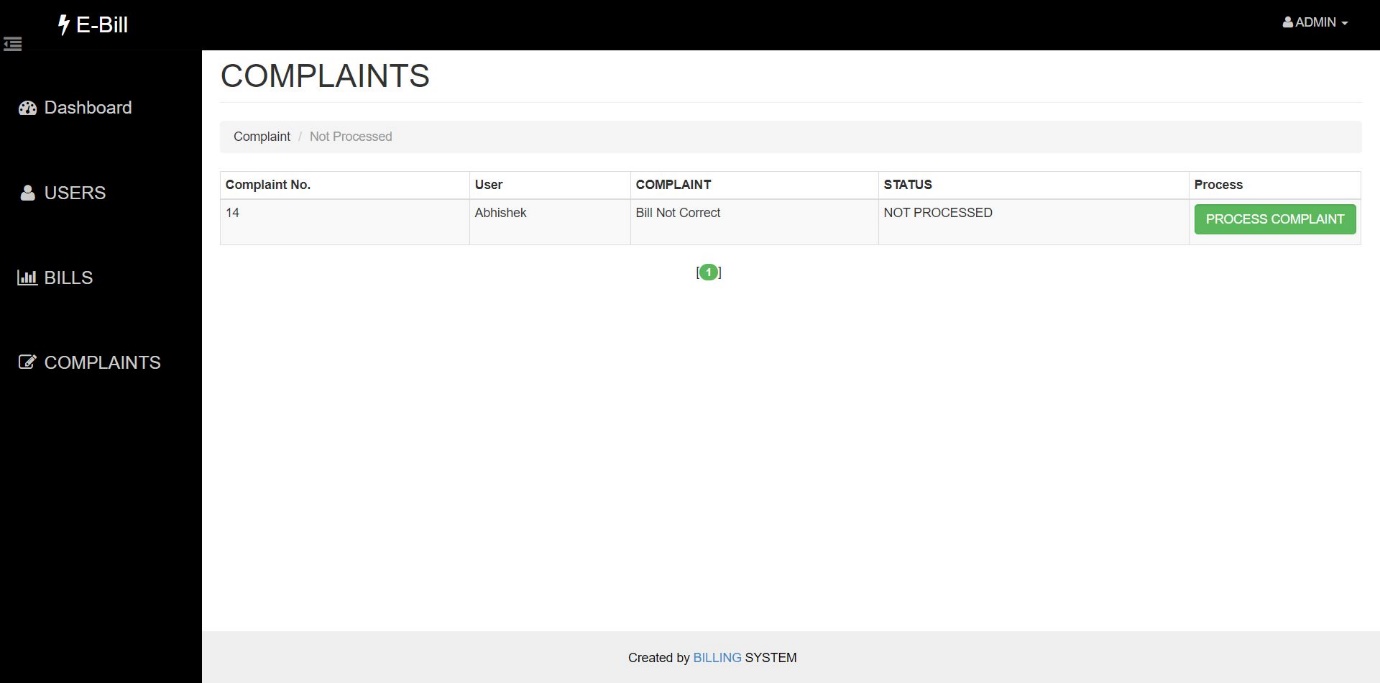
**Figure 5.4:** User Details



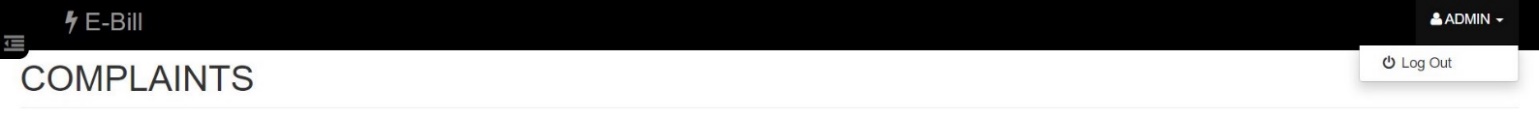
**Figure 5.5:** Generated bill history



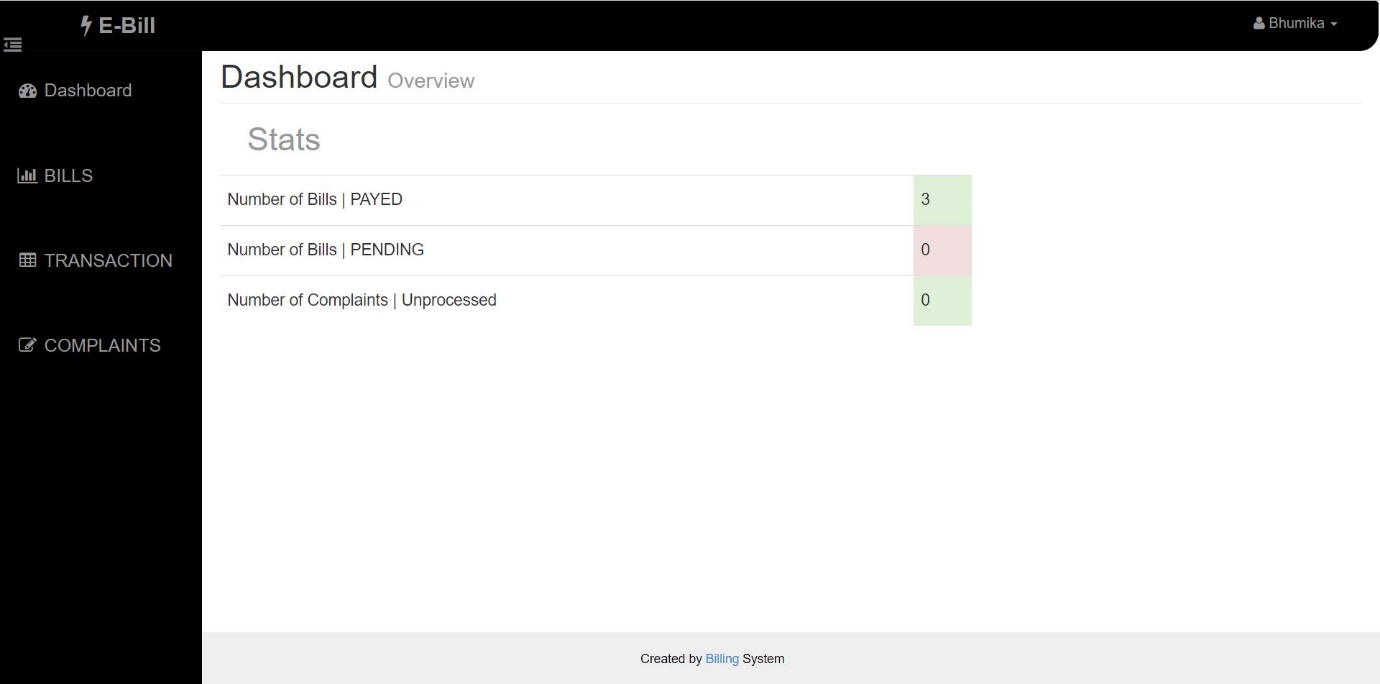
**Fig.5.6:** Generating new bill



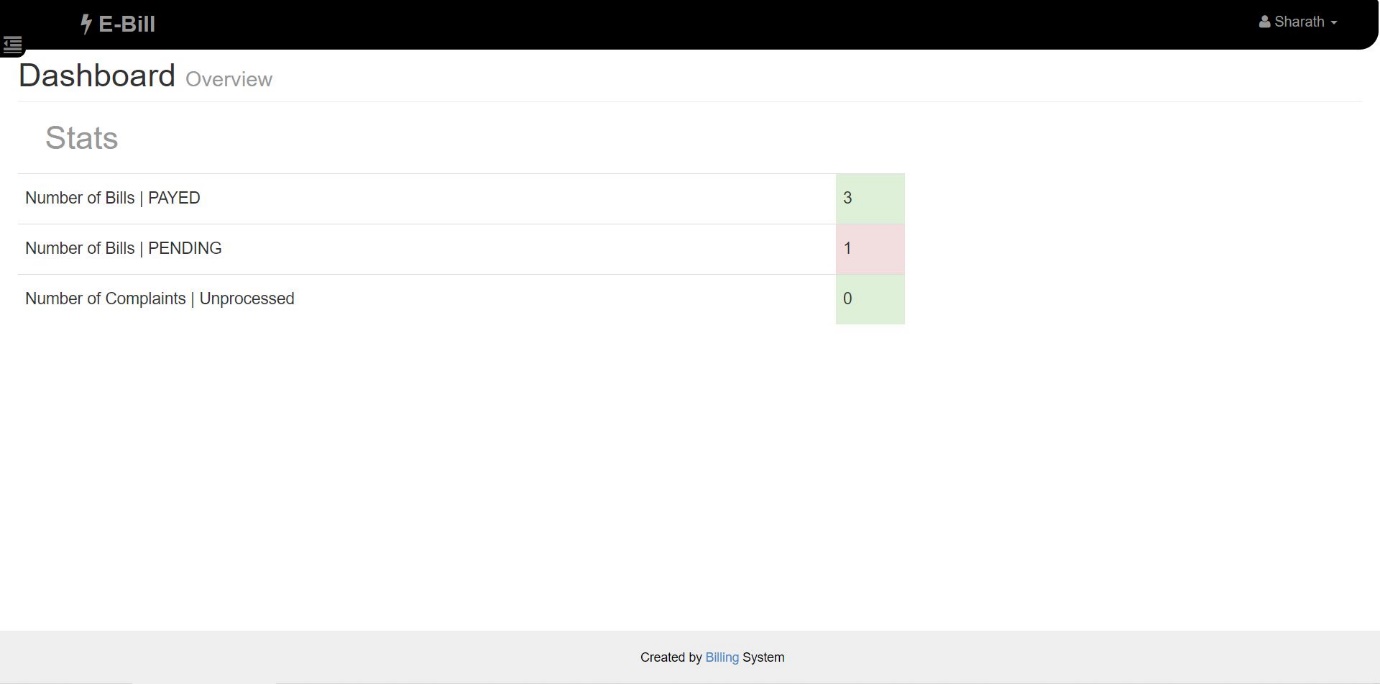
**Figure 5.7:** Complaints page



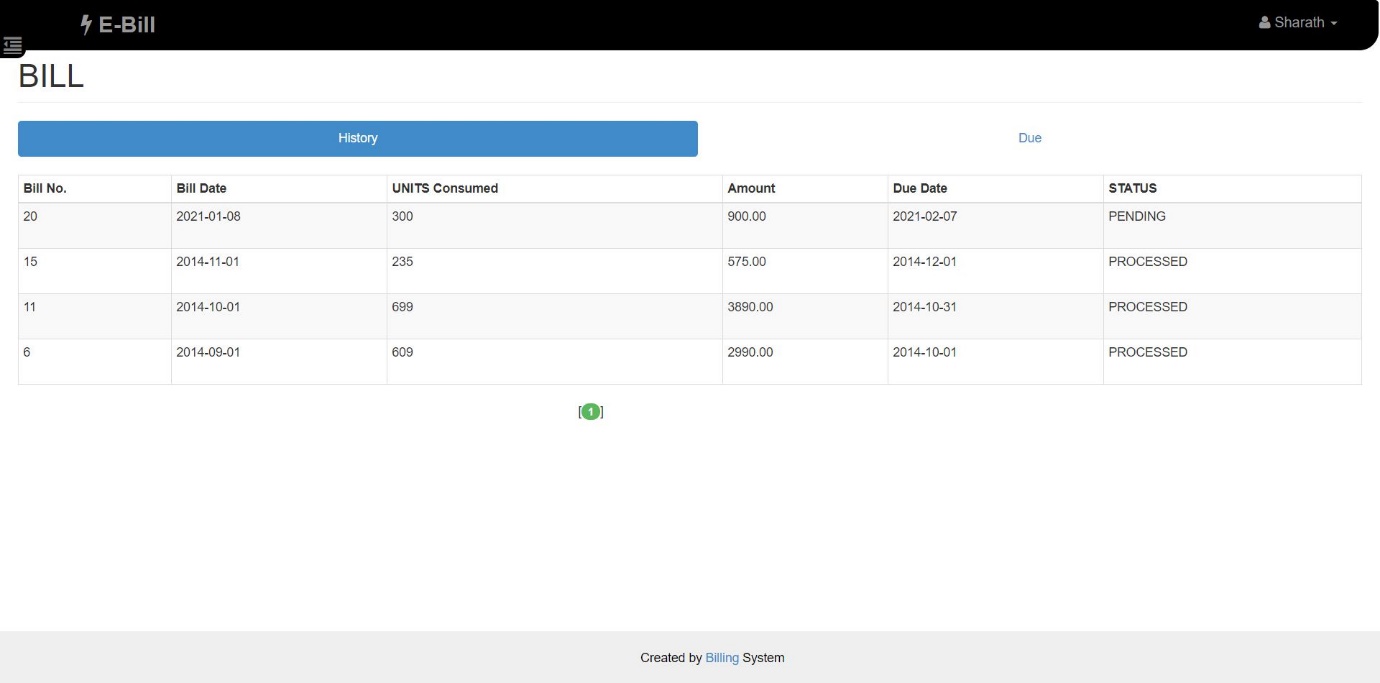
**Fig.5.7** Admin logout



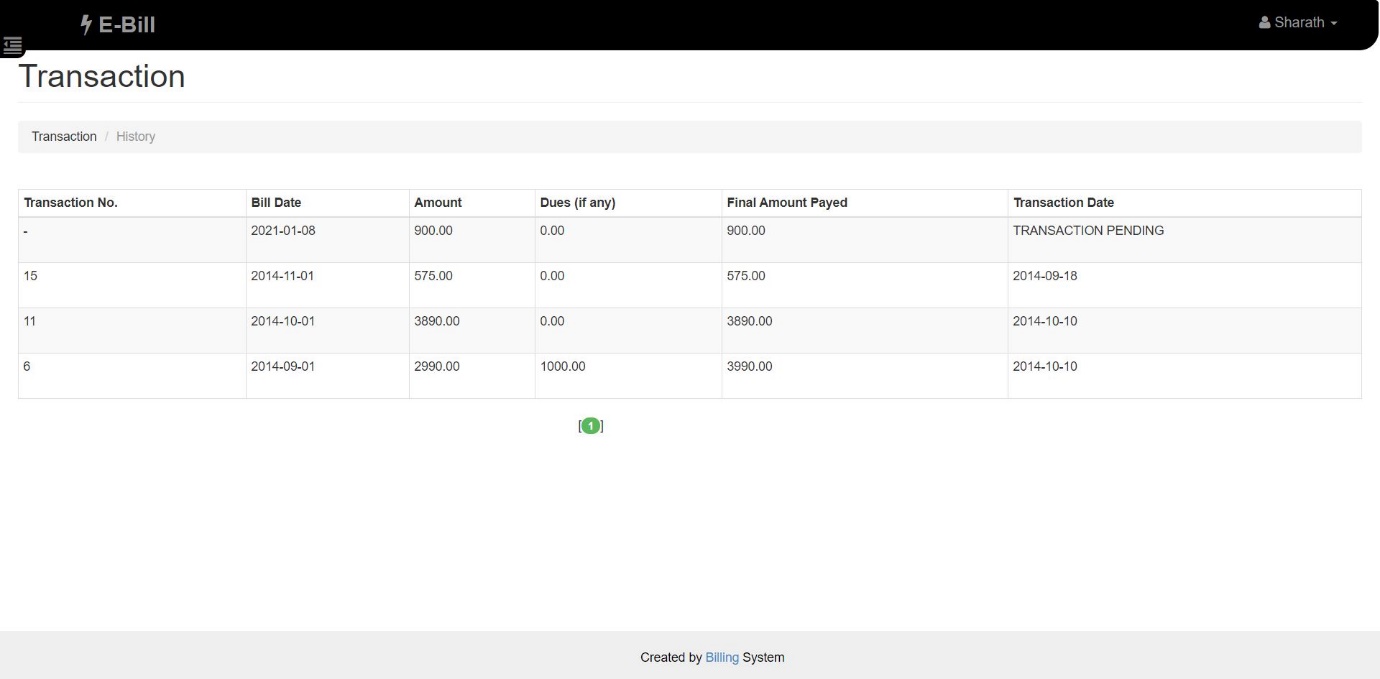
**Figure 5.8:** User page



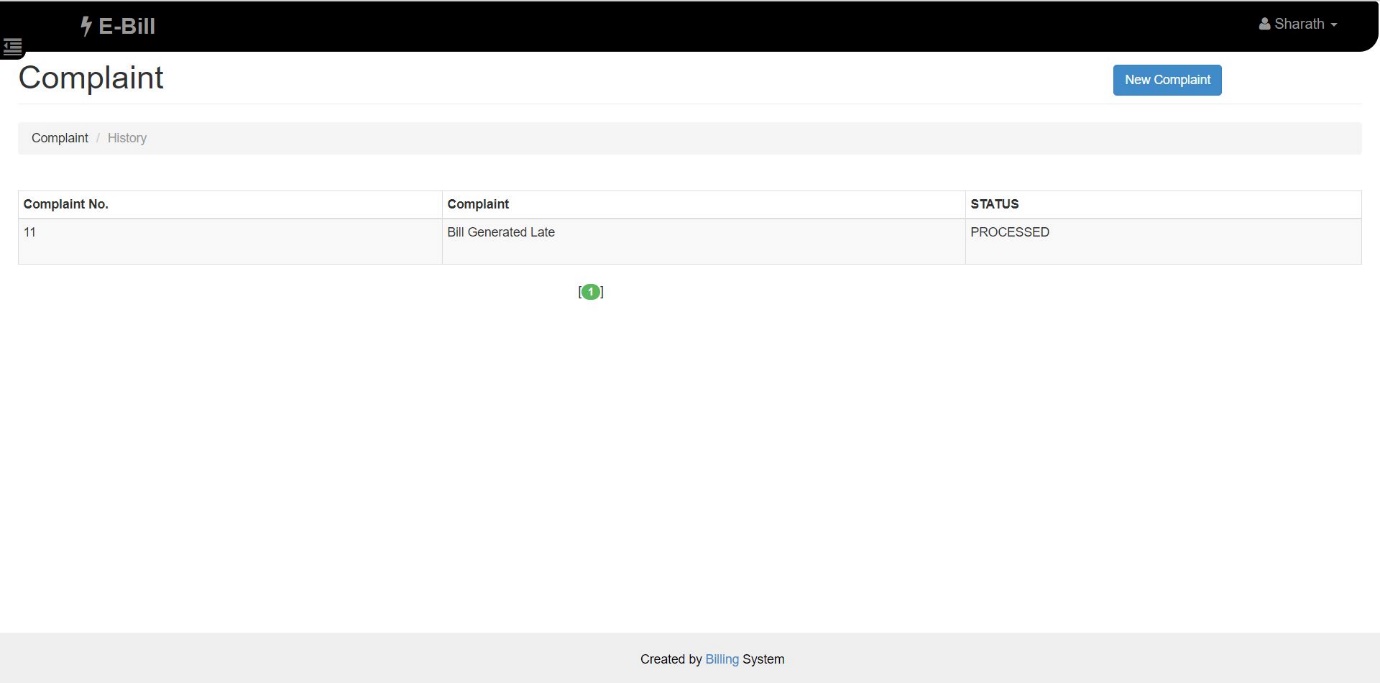
**Figure 5.9:** User dashboard



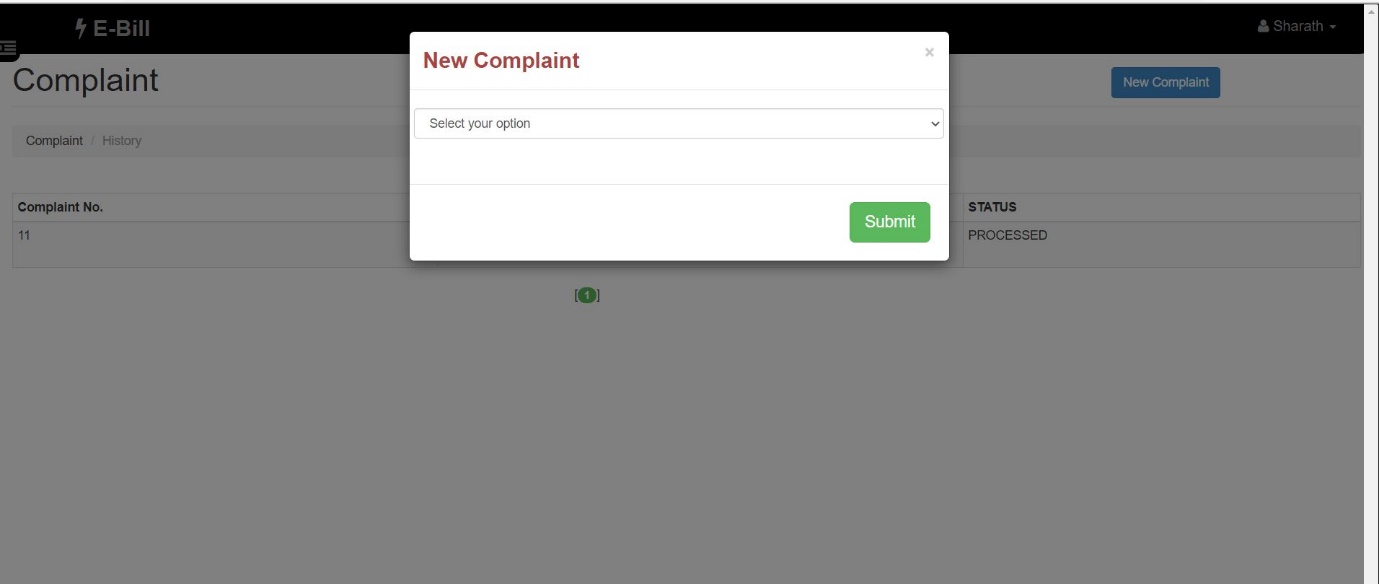
**Figure 5.10:** User bill history



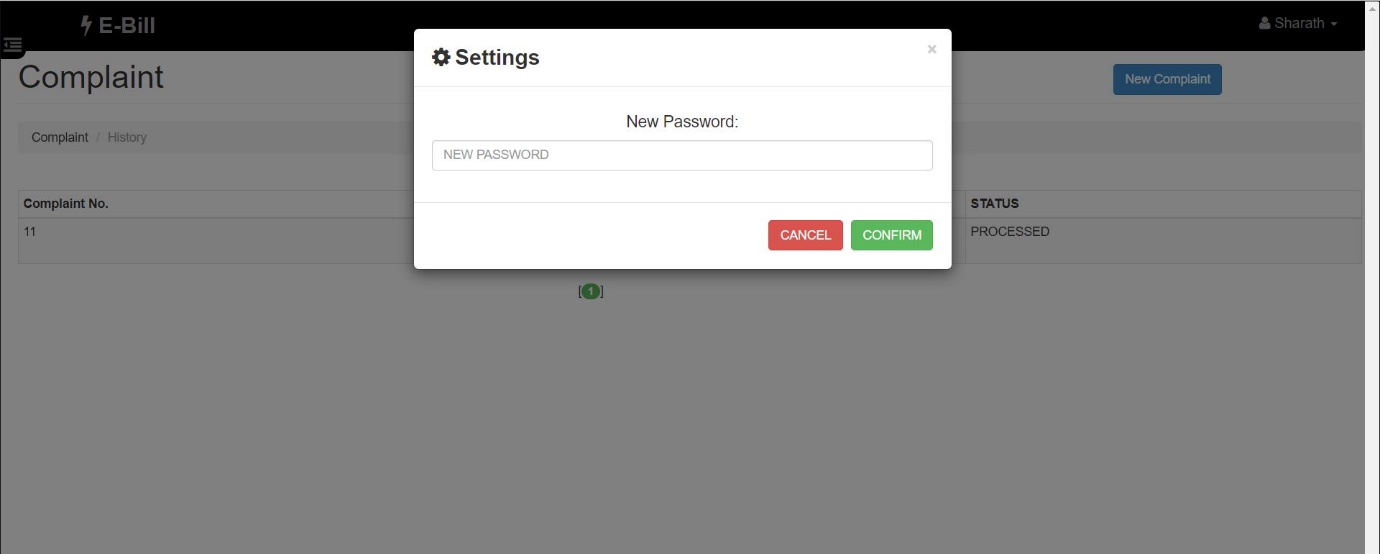
**Figure 5.11:** User transaction history



**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](http://www.php.net)
  + www.mysql.com
  + [www.xamppserver.com](http://www.xamppserver.com)
  + [www.w3schools.com](http://www.w3schools.com)
  + [www.stackoverflow.com](http://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