KONGU ENGINEERING COLLEGE
(Autonomous)
Perundurai, Erode – 638060

DEPARTMENT OF COMPUTER APPLICATIONS

A MICRO LEVEL PROJECT REPORT
FOR
ELECTRICITY BILL MANAGEMENT SYSTEM

ADVANCED JAVA PROGRAMMING
LABORATORY (22MCL21)

MAY - 2024

Submitted by

DHYANESHVAR PRASAATH JM (23MCR019)
JEYSARAVANAN S (23MCR037)
MAONJKUMAR V (23MCR052)

KONGU ENGINEERING COLLEGE

(Autonomous)

Perundurai, Erode – 638060

Transform Yourself

DEPARTMENT OF COMPUTER APPLICATIONS

BONAFIDE CERTIFICATE

Name : DHYANESHVAR PRASAATH J M (23MCR019)
JEYSARAVANAN S (23MCR037)
MANOJKUMAR V (23MCR052)

Course Name : ADVANCED JAVA PROGRAMMING LABORATORY

Course Code : 22MCL21

Semester : II

Certified that, this is a bonafide record of work for micro level project done by the above students for **22MCL21-ADVANCED JAVA PROGRAMMING LABORATORY** during the academic year **2023- 2024.**

Submitted for the Viva Voce Examination held on _____

Lab Incharge **Head of the Department**

(Dr.A.TAMILARASI)

INDEX

ABSTRACT

The Electricity Bill Management System (EBMS) is a simplified approach to effectively managing electricity bills. It integrates MySQL for database management with Thymeleaf for the frontend and the Spring Boot framework for backend operations. With this all-inclusive method, billing is made easier for administrators and customers alike, guaranteeing a seamless and user-friendly experience.

The fundamental feature of EBMS is its capacity to enable users to enter their electricity consumption in units, which promotes easy communication with the system. Using predetermined tariff rates, EBMS uses MySQL to automatically determine the amount that must be paid upon submission. Through this integration, billing data can be processed in real-time, and the customer can see the calculated amount right away. Through the utilization of MySQL's extensive database management features, EBMS guarantees data accuracy and integrity, thereby augmenting billing operations transparency.

SOFTWARE REQUIREMENT SPECIFICATION

A declaration that describes the capability that a system needs in order to satisfy the user's requirements is known as a system requirement. The system requirements for specific machines, software or business operations are general. Taking it all the way down to the hardware and coding that operates the software. System requirements are the most efficient way to address user needs while lowering implementation costs.

Hardware Requirements

The hardware for the system is selected considering the factors such as CPU processing speed, memory access, peripheral channel access speed, printer speed; seek time & relational data of hard disk and communication speed etc. Below is the minimum hardware requirement of the project.

Processer	:	Intel Core i5
RAM	:	4GB RAM
Hard Disk	:	320GB
Monitor	:	15''VGA Monitor

Software Requirements

The software for the project is selected considering the factors such as working front end environment, flexibility in the coding language, database knowledge of enhances in backend technology etc.

Operating System	:	Windows 11
Front End	:	Thymeleaf
Back End	:	Spring Boot
Database	:	MySQL
Tools	:	IntelliJ IDEA Community Edition

MODULE DESCRIPTION

Bill Calculation:

Bill Calculation module facilitates the seamless integration of consumer input, typically in the form of electricity units consumed, with predefined tariff rates. Upon receiving input from the user interface, the module swiftly processes this data, performs the necessary calculations based on the configured rates, and generates the payable amount in real-time.

Key Features:

- **Dynamic Calculation:** The module dynamically calculates the amount to be paid based on the electricity units entered by the consumer, ensuring accuracy and transparency in billing.
- **Tariff Rate Integration:** It integrates seamlessly with the tariff rate database to fetch the current rates and apply them to the calculation process.
- **Instantaneous Response:** The real-time nature of the module ensures that consumers receive immediate feedback on the payable amount upon entering their electricity usage, enhancing user experience and satisfaction.
- **Error Handling:** Robust error handling mechanisms are implemented to handle invalid inputs and unexpected scenarios, ensuring reliability and preventing erroneous calculations.
- **Scalability:** The module is designed to scale efficiently to handle a large volume of concurrent requests, making it suitable for deployment in systems serving a wide consumer base.
- **Security:** Measures are implemented to ensure the security of sensitive data during the calculation process, protecting consumer privacy and financial information.

Database Module:

The Database module within the Electricity Billing Management System (EBMS) serves as the backbone for storing and managing crucial billing data. Seamlessly integrated with MySQL, this module efficiently handles the storage and retrieval of consumer-related information, tariff rates, and billing calculations.

Key Features:

- **Data Storage and Retrieval:** Facilitates efficient storage and retrieval of consumer input data, leveraging MySQL for reliable management of electricity units consumed and billing amounts.
- **Tariff Rate Management:** Dynamically fetches and applies current tariff rates to billing calculations, ensuring accuracy by maintaining an up-to-date record of pricing structures.
- **Real-time Data Processing:** Enables instantaneous processing of billing data, providing consumers with immediate feedback on payable amounts upon entering electricity usage, thereby enhancing user experience and satisfaction.
- **Error Handling and Data Integrity:** Implements robust error handling mechanisms to address invalid inputs and unexpected scenarios, enforcing data integrity constraints to maintain reliability and prevent erroneous calculations.
- **Scalability and Security:** Designed for efficient scaling to handle large volumes of concurrent requests, with stringent security measures ensuring confidentiality and integrity of sensitive consumer data, safeguarding privacy and financial information.

SOURCE CODE

Pom.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
	<modelVersion>4.0.0</modelVersion>
	<parent>
		<groupId>org.springframework.boot</groupId>
		<artifactId>spring-boot-starter-parent</artifactId>
		<version>3.2.5</version>
		<relativePath/> <!-- lookup parent from repository -->
	</parent>
	<groupId>com.microproject</groupId>
	<artifactId>demo</artifactId>
	<version>0.0.1-SNAPSHOT</version>
	<name>ElectricityBillManagementSystem</name>
	<description>Demo project for Spring Boot</description>
	<properties>
		<java.version>17</java.version>
	</properties>
	<dependencies>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-data-jdbc</artifactId>
		</dependency>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-data-jpa</artifactId>
		</dependency>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-thymeleaf</artifactId>
		</dependency>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-web</artifactId>
		</dependency>

		<dependency>
			<groupId>com.mysql</groupId>
			<artifactId>mysql-connector-j</artifactId>
```

```
                <scope>runtime</scope>
            </dependency>
            <dependency>
                <groupId>org.projectlombok</groupId>
                <artifactId>lombok</artifactId>
                <optional>true</optional>
            </dependency>
            <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-test</artifactId>
                <scope>test</scope>
            </dependency>
        </dependencies>

        <build>
            <plugins>
                <plugin>
                    <groupId>org.springframework.boot</groupId>
                    <artifactId>spring-boot-maven-plugin</artifactId>
                    <configuration>
                        <excludes>
                            <exclude>
                                <groupId>org.projectlombok</groupId>
                                <artifactId>lombok</artifactId>
                            </exclude>
                        </excludes>
                    </configuration>
                </plugin>
            </plugins>
        </build>

</project>
```

ElectricityBillManagementSystemApplication.java:

```
package com.microproject.demo;


import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;


@SpringBootApplication
public class ElectricityBillManagementSystemApplication {
```

```
	public static void main(String[] args) {
		SpringApplication.run(ElectricityBillManagementSystemApplication.class, args);
	}

}
```

BillRepository.java:

```
package com.microproject.demo;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface BillRepository extends JpaRepository<Bill, Integer> {
   // You can add custom query methods here if needed
}
```

Bill.java:

```
package com.microproject.demo;

import jakarta.persistence.*;
import lombok.Data;

import static jakarta.persistence.GenerationType.SEQUENCE;

@Data
@Entity
@Table(name = "BillTable")
public class Bill {
   @Id
   @Column(name = "id")
   @GeneratedValue(strategy=SEQUENCE, generator="ID_SEQ")
   private int id;
   @Column(name="unitsConsumed")
   private int consumedUnits;
   @Column(name="billAmount")
   private double billAmount;

}
```

ElectricityBillController.java :

```
package com.microproject.demo;

import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;

@Controller
public class ElectricityBillController {

    @Autowired
    private BillRepository billRepository;

    @GetMapping("/")
    public String showElectricityBillForm() {
        return "electricity-bill-form";
    }

    @PostMapping("/")
    public String calculateElectricityBill(@RequestParam("units") int consumedUnits, Model model) {
        double unitRate1 = 3.50;
        double unitRate2 = 4.00;
        double unitRate3 = 5.20;
        double unitRate4 = 6.50;
        double billAmount;

        if (consumedUnits <= 50) {
            billAmount = consumedUnits * unitRate1;
        } else if (consumedUnits <= 150) {
            billAmount = (50 * unitRate1) + ((consumedUnits - 50) * unitRate2);
        } else if (consumedUnits <= 250) {
            billAmount = (50 * unitRate1) + (100 * unitRate2) + ((consumedUnits - 150) * unitRate3);
        } else {
            billAmount = (50 * unitRate1) + (100 * unitRate2) + (100 * unitRate3) + ((consumedUnits - 250) *
unitRate4);
        }
```

```
        // Create a new Bill object and set its properties
        Bill bill = new Bill();
        bill.setConsumedUnits(consumedUnits);
        bill.setBillAmount(billAmount);

        // Save the Bill object to the database using the repository
        billRepository.save(bill);

        // Add attributes to the model
        model.addAttribute("units", consumedUnits);
        model.addAttribute("billAmount", billAmount);

        return "electricity-bill-result";
    }
}
```

Application.properties:

```
spring.application.name=ElectricityBillManagementSystem
server.port=7500
spring.datasource.url=jdbc:mysql://localhost:3306/electricitybill
spring.datasource.username=root
spring.datasource.password=dhyanesh

spring.jpa.show-sql=true
spring.jpa.generate-ddl=true
spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
```

Electricity-bill-form.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Electricity Biling System</title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <style>
        body {
```

```
            background-color: #f2f7ff;
            font-family: 'Poppins', sans-serif;
        }

        h2 {
            font-size: 3rem;
            font-weight: 700;
            text-align: center;
            margin-bottom: 3rem;
        }

        .custom-form-label {
            font-weight: bold;
            font-size: 1.2rem;
        }

        input[type=number] {
            font-size: 1.2rem;
            padding: .75rem .75rem;
        }

        button[type=submit] {
            font-size: 1.2rem;
            padding: .75rem 3rem;
        }
    </style>
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=swap"
rel="stylesheet">
</head>
<body>
    <header>
        <nav class="navbar navbar-expand-lg navbar-light bg-light">
            <a class="navbar-brand" href="#">Electricity Bill Calculator</a>
        </nav>
    </header>
```

```
    <div class="container mt-5">
        <div class="row justify-content-center">
            <div class="col-md-6">
                <h2>Electricity Bill Calculator</h2>
                <form method="post" action="/" class="card p-3 shadow">
                    <div class="form-group">
                        <label for="units" class="custom-form-label">Enter Total Consumed Units:</label>
                        <input type="number" class="form-control" id="units" name="units" required>
                    </div>
                    <div class="text-center">
                        <button type="submit" class="btn btn-primary">Calculate Bill</button>
                    </div>
                </form>
            </div>
        </div>
    </div>
    <footer class="pt-4 my-md-5 pt-md-5 border-top">
        <div class="row">
            <div class="col-12 col-md">
                <small class="d-block mb-3 text-muted">&copy; 2024 Micro-Project</small>
            </div>

<!--            <div class="col-6 col-md">-->
<!--                <h5></h5>-->
<!--                <ul class="list-unstyled text-small">-->
<!--                    <li><a class="text-muted" href="#"></a></li>-->
<!--                    <li><a class="text-muted" href="#"></a></li>-->
<!--                </ul>-->
<!--            </div>-->
        </div>
    </footer>
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>
```

Electricity-bill-result.html:

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.w3.org/1999/xhtml">
<head>
    <meta charset="UTF-8">
    <title>Electricity Billing Amount</title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <style>
        body {
            background-color: #f2f7ff;
            font-family: 'Poppins', sans-serif;
        }

        .custom-result-heading {
            font-weight: 700;
            font-size: 3rem;
            margin: 0 0 2rem;
        }

        .custom-form-label {
            font-weight: bold;
            font-size: 1.2rem;
            margin-bottom: .5rem;
        }

        strong {
            font-size: 1.2rem;
            font-weight: 700;
        }
    </style>
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=swap"
rel="stylesheet">
</head>
<body>
<header>
```

```
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
      <a class="navbar-brand" href="#">Electricity Billing Amount</a>
    </nav>
</header>
<div class="container mt-5">
    <div class="row justify-content-center">
      <div class="col-md-6">
        <div class="card p-3 shadow">
          <h2 class="text-center mb-4 custom-result-heading">Your Electricity Billing Amount</h2>
          <div class="mb-3">
            <p class="custom-form-label mb-0">Total Consumed Units:</p>
            <p class="custom-result-box"><strong th:text="${units}"></strong></p>
          </div>
          <div>
            <p class="custom-form-label mb-0">Total Bill Amount:</p>
            <p class="custom-result-box"><strong th:text="${billAmount}"></strong></p>
          </div>
        </div>
      </div>
    </div>
</div>
<footer class="pt-4 my-md-5 pt-md-5 border-top">
    <div class="row">
      <div class="col-12 col-md">
        <small class="d-block mb-3 text-muted">&copy; 2024 Micro-Project</small>
      </div>

<!--       <div class="col-6 col-md">-->
<!--          <h5>About</h5>-->
<!--          <ul class="list-unstyled text-small">-->
<!--            <li><a class="text-muted" href="#"></a></li>-->
<!--            <li><a class="text-muted" href="#"></a></li>-->
<!--          </ul>-->
<!--       </div>-->
    </div>
</footer>
```

```
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>
```

Style.css:

```
/* Apply a max-width to the form */
form {
  max-width: 500px;
  margin: 0 auto;
}

/* Apply a grid layout to the form elements */
form div {
  display: grid;
  grid-template-columns: 1fr 1fr;
  grid-gap: 10px;
}

/* Add vertical padding to the inputs */
input[type="text"] {
padding: .375rem .75rem;
}

/* Add margin to the form labels */
label {
margin-bottom: .5rem;
}

/* Apply mobile-first responsive design using media queries */
@media screen and (max-width: 600px) {
  /* Modify the form width and grid layout to fit smaller screens */
  form {
     max-width: 350px;
  }
```

```
  form div {
     display: block;
  }

  /* Reduce the font size of the labels */
  label {
     font-size: 14px;
  }
}

/* Add blue boxes to the result values */
.custom-result-box {
  border: 2px solid #2296f3;
  padding: 1rem;
  color: #2296f3;
  font-weight: bold;
  text-align: center;
}
```

SCREENSHOTS

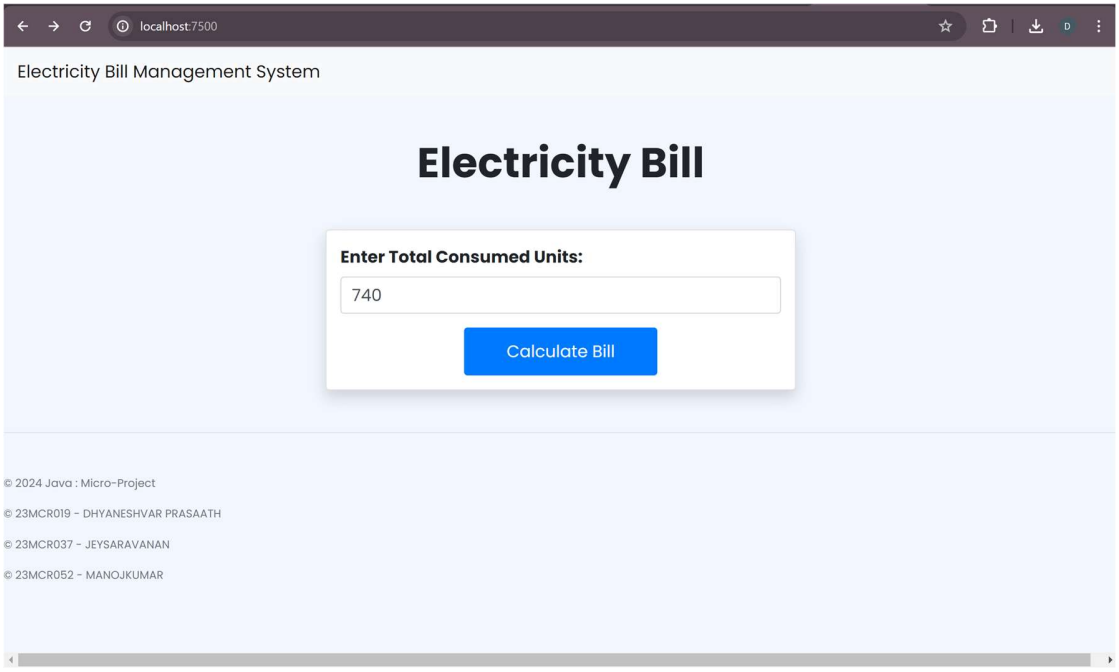

Figure 1.1

Figure 1.1 represents the electricity bill, consumed unit interface.

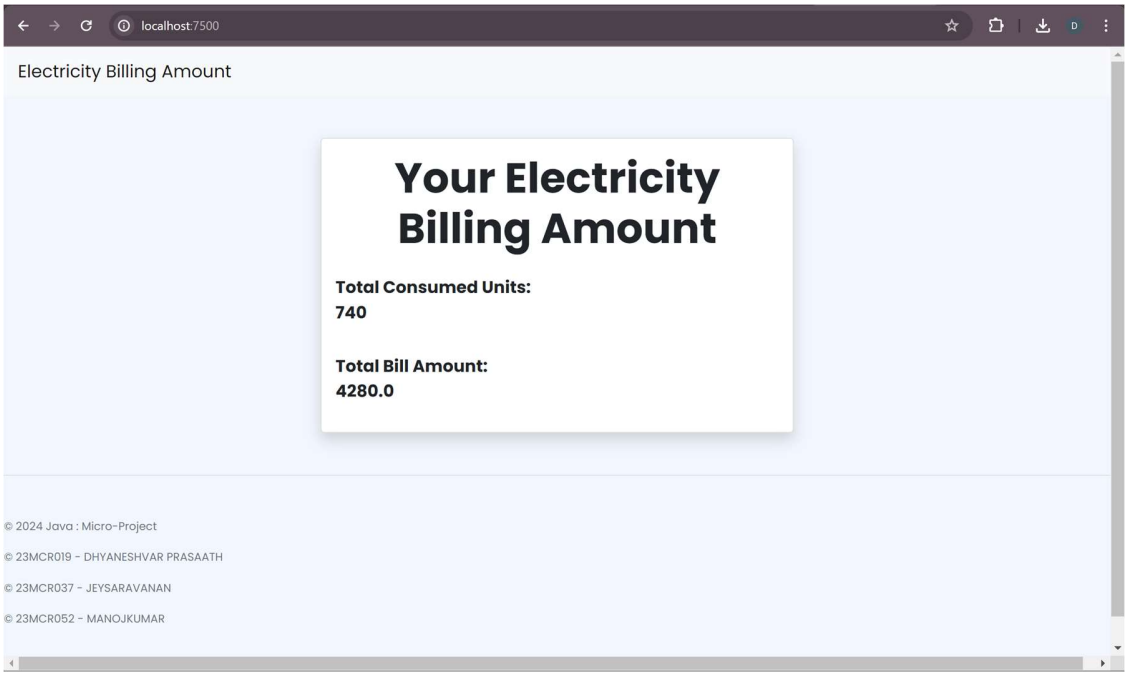

Figure 1.2

Figure 1.2 represents the total bill amount based on the units entered.

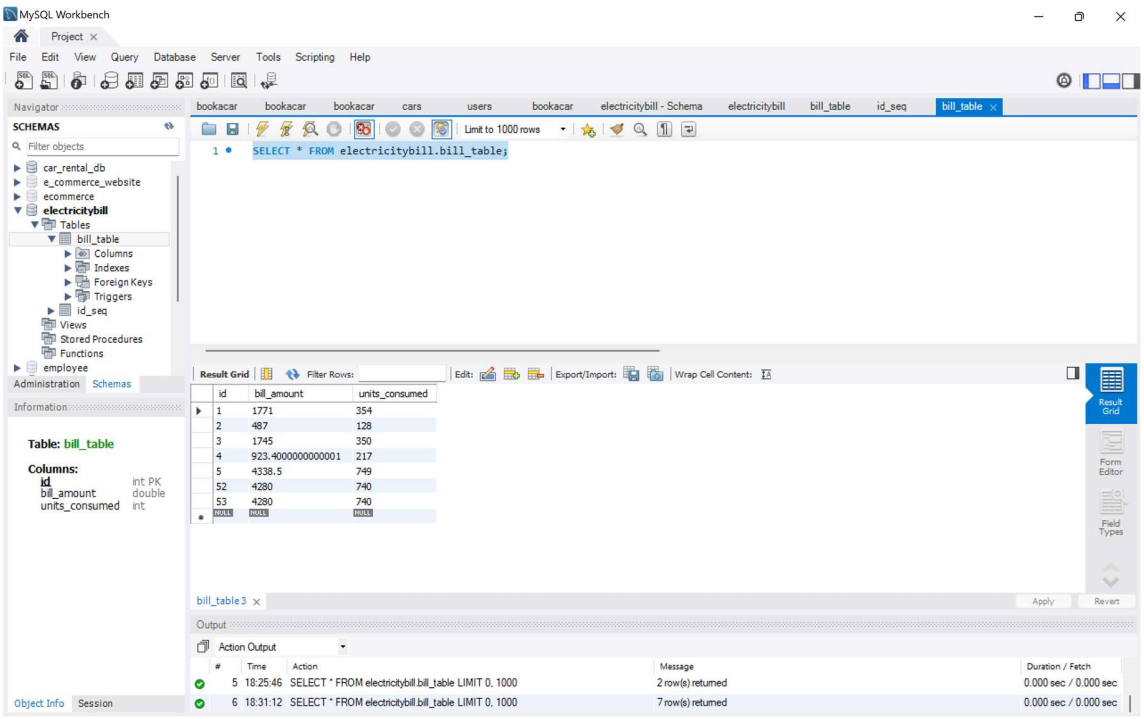

Figure 1.3

Figure 1.3 represents data stored in the MySQL workbench.

CONCLUSION

The Electricity Billing Management System (EBMS) offers a streamlined solution for managing electricity bills efficiently, leveraging MySQL for database management alongside Thymeleaf for the frontend and the Spring Boot framework for backend operations. By integrating these technologies, EBMS ensures a simplified billing process for both administrators and customers, enhancing overall user experience.

In summary, EBMS guarantees accuracy, dependability, and transparency in the management of electricity bills in addition to streamlining the billing process. It is evidence of the ability of integrated technologies to improve billing management system effectiveness and user satisfaction.

REFERENCES

1. https://docs.spring.io/spring-boot/docs/current/reference/html/web.html
2. https://docs.spring.io/spring-framework/reference/web/webmvc-view/mvc-thymeleaf.html
3. https://www.baeldung.com/spring-boot-jsp
4. https://spring.io/guides/gs/accessing-data-mysql
5. https://docs.spring.io/spring-boot/docs/current/reference/pdf/spring-boot-reference.pdf