



सेल SAIL

स्टील अथॉरिटी ऑफ इण्डिया लिमिटेड
STEEL AUTHORITY OF INDIA LIMITED

Advanced Electricity Data Integration and Management Platform

Name: Aruf Khan

College: Vellore Institute of
Technology, Vellore, TN-632014

College Regn: 22BCE0578

BSP Roll No: P-24/9441

Date: 10th July, 2024

ABSTRACT

This project involves the development of an application designed to perform various operations on a comprehensive database. The database contains detailed information about electricity usage, billing, meter types, solar panel types, phase details, and addresses for various plots across city sectors. The primary objective of the application is to facilitate efficient data management through functionalities such as searching, updating, saving, checking status, error reporting, clearing fields, and exiting the application. By utilizing robust database management techniques and real-time data processing, this application ensures accurate and up-to-date information handling, thereby enhancing the overall management of electricity-related data for the city. The project was developed using a combination of five key applications and two coding languages, ensuring a seamless integration between the database, compiler, server, and user interface. Through meticulous debugging, complex logic implementation, and extensive testing, the application delivers a user-friendly experience with an intuitive frontend and well-documented code for ease of maintenance and future enhancements.

TABLE OF CONTENTS

1. Title Page
2. Abstract
3. Table of Contents
4. Introduction
5. Methodology
6. Technical Specifications
7. Project Description
8. Implementation
9. Results
10. Discussion
11. Conclusion
12. Appendices
13. References
14. Acknowledgements

INTRODUCTION

The electricity data management project at Bhilai Steel Plant is a response to the critical need for modernizing outdated technology systems plagued by numerous errors, severely impacting basic operational efficiency. The previous system's challenges necessitated a comprehensive overhaul to effectively manage essential aspects of electricity data for city sectors. This includes metrics on electricity usage, billing details, meter types, solar panel specifications, phase specifics, and plot addresses across various city sectors.

The primary objective of this project is to develop a robust application capable of performing essential data management functions such as searching, updating, saving, checking status, error reporting, field clearing, and application navigation. It aims to provide real-time operational support to over 10,000 executives within the plant, including SAIL executives and general managers of the C&IT department.

Technologically, the project leverages HP VICTUS PC with Windows 11 as the hardware foundation, complemented by a software toolkit featuring Eclipse, VS Code, Toad for Oracle, Java JDBC, SQL, MySQL, and Java SDE resources. Development primarily utilizes JSP (JavaServer Pages) alongside Oracle Procedure and other coding languages like Java and SQL, ensuring robust real-time data processing capabilities and an intuitive user interface designed for usability and scalability.

METHODOLOGY

The working for this project followed a systematic approach aimed at effectively managing electricity data and ensuring seamless user interaction. Initially, extensive planning sessions were held to define project objectives and gather detailed requirements. This phase laid the foundation for designing a robust system architecture, focusing on the integration of backend logic developed in Java with frontend components built using JSP, HTML, and CSS. The development process proceeded in phases, facilitated by tools like Eclipse IDE and JDK for Java development, ensuring systematic implementation of features and functionalities. Rigorous testing procedures were then applied to validate the system's performance and functionality, with frequent debugging sessions addressing any identified issues promptly. Throughout the project, meticulous documentation of technical specifications and user manuals was maintained, supported by continuous feedback from stakeholders and supervisors. This iterative approach ensured that the final solution met both technical requirements and user expectations effectively.

TECHNICAL SPECIFICATIONS

1. Toad for Oracle (Version: 17.0.353.2906_x64_En)
2. Eclipse IDE (Version: 2024-06)
3. VS Code
4. Apache Tomcat (Version: 9.0.89)
5. JDK (Version: 21)
6. Oracle Database (Version: OraDB21Home1)
7. Quest Software

Programming Languages:

1. JSP (JavaServerPages)
2. SQL
3. Java
4. HTML
5. CSS

PROJECT DESCRIPTION

The project utilized the following technologies:

- Toad for Oracle (Version: DBAPlus_Subscription_17.0.353.2906_x64_En) for comprehensive database management.
- Eclipse IDE (Version: 2024-06) for robust Java development.
- VS Code for lightweight code editing.
- Apache Tomcat (Version: 9.0.89) for deploying Java-based web applications.
- JDK (Version: 21) for Java development environment setup.
- Oracle Database (Version: OraDB21Home1) for efficient data storage and retrieval.
- Quest Software for additional database management functionalities.

Programming languages employed include:

- JSP (JavaServerPages) for dynamic web page generation.
- SQL for database query handling.
- Java for backend logic and application development.
- HTML for structuring web pages.
- CSS for styling and enhancing the frontend user interface.

IMPLEMENTATION

The operations in the project are performed through a combination of frontend HTML forms and backend Java Servlets/JSPs that interact with the Oracle database using JDBC (Java Database Connectivity). Here's a detailed breakdown of how each operation is handled:

1. Data Retrieval:

- **Fetch Data Operation:** When a user initiates a request to fetch data for a specific sector and plot, the application first establishes a connection to the Oracle database using JDBC.

- **SQL Queries:** Depending on whether the data exists in the `ELEC_MASTER` table or needs to be fetched from the `TP_MAS` table, corresponding SQL queries are executed.

- **ResultSet Handling:** The retrieved data from the database ResultSet is processed to construct a JSON response that includes all relevant fields such as sector, plot, consumer details, meter information, and additional attributes based on specific business rules and conditions.

- **Error Handling:** Exception handling ensures that any errors during database interaction or query execution are captured and appropriately handled, returning error messages in the JSON response for debugging and user feedback.

2. Data Saving and Updating:

- **Save/Update Data Operation:** When a user submits updated data through the frontend form, the application first verifies the HTTP request method and retrieves form parameters using Servlet request objects.

- **Prepared Statements:** PreparedStatement are used to execute parameterized SQL queries for either updating existing records in the

`ELEC_MASTER` table or inserting new records into it, depending on whether the record already exists.

- Transaction Management: The application ensures transactional integrity by handling database connections, statement executions, and result set closures within try-catch-finally blocks to guarantee resources are properly released and transactions are committed or rolled back as necessary.

- Feedback and Response: Upon successful data update or insertion, a confirmation message is generated and displayed to the user through a dynamically generated HTML response, indicating the success or failure of the operation based on the number of rows affected by the SQL statement.

3. Security Measures:

- Prevention of SQL Injection: Parameterized queries in JDBC are employed to prevent SQL injection attacks, where user input is sanitized and validated to ensure only expected data types and formats are accepted.

- Session Management: Sessions are managed securely to maintain user authentication and authorization throughout the application session, ensuring that only authorized users can access and modify electricity usage data.

4. Frontend Integration:

- HTML Forms: User-friendly HTML forms are designed to collect and display input fields based on the database schema, providing an intuitive interface for users to interact with and input or update data effectively.

- JSON Responses: Responses from backend operations are formatted as JSON objects, facilitating seamless integration with

frontend JavaScript frameworks or AJAX calls for dynamic updates and real-time data rendering without full page reloads.

Overall, the project leverages a robust architecture combining frontend HTML forms with backend Java Servlets/JSPs and Oracle database connectivity via JDBC to efficiently manage and process electricity usage data, ensuring reliability, security, and user-friendly functionality across all operational aspects.

RESULTS

Project Outcomes: The implementation of the project has significantly improved data management and handling capabilities at Bhilai Steel Plant. By migrating from outdated systems to a modernized application, the project has streamlined operations related to electricity usage data. Real-time data processing capabilities, efficient database querying through optimized SQL queries, and enhanced user interface design have collectively contributed to better management of consumer information, meter details, and solar panel data. The project's robust error handling mechanisms and secure transaction management ensure data integrity and reliability, crucial for operational continuity.

Impact: The benefits extend beyond operational efficiency to substantial improvements in decision-making and resource utilization. Bhilai Steel Plant now enjoys quicker access to accurate consumer data, facilitating prompt decision-making processes. The city sectors served also benefit from improved service delivery and reliability in electricity supply management. This modernization initiative not only enhances customer satisfaction through better service responsiveness but also supports environmental sustainability efforts through more effective management of solar panel installations. Overall, the project has bolstered data-driven operations, enhancing both operational efficiency and service quality for stakeholders at Bhilai Steel Plant and the city sectors it serves.

DISCUSSION

The project aimed to revolutionize data management within Bhilai Steel Plant and city sectors by implementing a robust system for electricity usage tracking and analysis. Leveraging technologies such as Oracle Database, Eclipse IDE, Apache Tomcat, and JavaServer Pages (JSP), the system successfully achieved its objectives of enhancing operational efficiency and data accuracy. Through extensive testing and iterative development cycles, we ensured that the application met industry standards for reliability and performance.

Challenges encountered during implementation included initial setup complexities and integration issues with legacy systems, which required meticulous debugging and refinement. User feedback was pivotal in refining the system's user interface and functionality, ensuring intuitive navigation and ease of use for plant executives and administrators.

Results showed significant improvements in data accessibility and real-time monitoring capabilities, empowering decision-makers with timely insights into electricity consumption patterns and operational efficiencies. However, scalability under high workload conditions emerged as a potential limitation, suggesting future enhancements in system architecture and performance optimization.


CONCLUSION

In conclusion, this project marks a transformative milestone in enhancing data management and operational efficiency at Bhilai Steel Plant and city sectors. By leveraging cutting-edge technologies including Oracle Database, Eclipse IDE, and Apache Tomcat, coupled with robust development in JavaServer Pages (JSP) and SQL, we have successfully created a sophisticated platform for real-time electricity usage tracking and analysis. The implementation process, though challenging, was navigated with precision, addressing initial setup complexities and integrating seamlessly with existing infrastructure.

The outcomes have been profound, facilitating streamlined operations, enhanced data accuracy, and empowered decision-making capabilities for plant executives and administrators. The project's success is underscored by its ability to deliver actionable insights into consumption patterns, thereby optimizing resource allocation and promoting sustainable practices.

Looking forward, this initiative sets a precedent for future technological advancements in industrial data management, promising continued innovation and growth. With its proven impact on operational excellence and strategic planning, this project exemplifies our commitment to excellence and leadership in the realm of smart infrastructure management.

APPENDICES

**सल SAIL**

Electricity Master Data Entry

Sector Plot ☐ BSP qtr

SearchSaveClearExit

Consumer No

Location

Name

Physical Sector

Street

QR No

QR Type

Type

Address L1

Address L2

Address L3

Status

From TP Master

Meter No

Meter Make

Meter KW

Sanc Load (KW)

Phase

Multi Factor

Solar

Solar Meter No

Solar Meter Make

Solar Meter KW

Solar Sanc Load (KW)

Solar Phase

Solar Multi Factor

Solar Meter Status

Solar AMP

Elec Code

Std Light Units

Std Power Units

Load KW

Consumer Type


Tariff Category

Supply Purpose

Consumer Category

Assd Consumption

Remarks

**सल SAIL**

Electricity Master Data Entry

Sector Plot ☐ BSP qtr

SearchSaveClearExit

Consumer No

Location

Name

Physical Sector

Street

QR No

QR Type

Type

Address L1

Address L2

Address L3

Status

From TP Master

Meter No

Meter Make

Meter KW

Sanc Load (KW)

Phase

Multi Factor

Solar

Solar Meter No

Solar Meter Make

Solar Meter KW

Solar Sanc Load (KW)

Solar Phase

Solar Multi Factor

Solar Meter Status

Solar AMP

Elec Code

Std Light Units

Std Power Units

Load KW

Consumer Type

Tariff Category

Supply Purpose

Consumer Category

Assd Consumption

Remarks

C#ESTATE@XE - Toad for Oracle - [Schema Browser (C#ESTATE.ELEC_MASTER)]

Sign-in to continue Sign-in

File Edit Search Editor Session Database Debug View Team Coding Utilities Run Window Help

C#ESTATE@XE

Project Manager C#ESTATE

Tables

ELEC_MASTER Created: 04/07/2024 16:14:19 Last DDL: 04/07/2024 16:14:20

Primary Key: SECTOR, PLOT

Columns Indexes Constraints Triggers Data Script Partitioning Grants Synonyms Stats/Size Referential Used By Policies Auditing

Sort by PK Read Only Desc Auto Refresh

| SECTOR | PLOT | CONSUMER_NO | LOCATION | ADDRESS_L1 | ADDRESS_L2 | ADDRESS_L3 | METER_NO | METER_MAKE | METER_KW | NAME | CONSUMER_TYPE |
|--------|------|-------------|------------------|------------|------------|--------------------|---------------|------------|----------|---------------|---------------|
| AO | 110 | 155451 | Sector A | | | | NA | NA | 0 | VVV Corp | NA |
| A1 | 101 | CUST123456 | Main Street | BL | Building 1 | Floor 3, Apt 301 | MTR1234567890 | ABC Meters | 5 | John Doe | Residential |
| B2 | 202 | CUST789012 | Park Avenue | GR | Tower 2 | Floor 15, Apt 1503 | MTR9876543210 | TypeAlpha | 8 | Jane Smith | Commercial |
| C3 | 303 | CUST456789 | Sunset Boulevard | RD | House 3 | Unit 10 | MTR2468013579 | TypeBeta | 3 | Michael Brown | Industrial |
| D4 | 404 | CUST234567 | Ocean View Drive | SD | Villa 4 | Suite 401 | MTR1357924680 | TypeGamma | 6 | Emily Johnson | Residential |

Row 2 of 5 total rows

Ctrl: 4, Sel: 1 C#ESTATE@XE

localhost:8080 says
No electricity billing for this party.

सेल SAIL

Sector BA Plot 120 ☐ BSP qtr

Search Save Clear Exit

Consumer No CUST123456 Location Main Street

Name John Doe Physical Sector BL

Street BL QR No Building 1

QR Type Type Residential

From TP Master

Meter No MTR1234567890 Meter Make ABC Meters

Meter KW 5 Sanc Load (KW) 10.5

Phase 1 Multi Factor 2

localhost:8080 says
This is a terminated party. Please check.

सेल SAIL

Sector EA Plot 150 ☐ BSP qtr

Search Save Clear Exit

Consumer No CA/130 Location Sector C

Name LMN Inc Physical Sector

Street QR No

QR Type Type

Address L1 Address L2

Address L3 Status A

From TP Master

Meter No Meter Make

Meter KW Sanc Load (KW)

Phase Multi Factor

Data updated successfully

Back

Files:

index.jsp

```
<!DOCTYPE html>
<%@ include file="DBcon.jsp" %>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="new_design_02.css">
  <title>Electricity Data</title>
</script>
document.addEventListener("DOMContentLoaded", function() {
  function fetchData() {
    var sector = document.getElementById("sector").value;
    var plot = document.getElementById("plot").value;

    if (sector.length === 2 && plot.length === 3) {
      var xhr = new XMLHttpRequest();
      xhr.onreadystatechange = function() {
        if (xhr.readyState === 4 && xhr.status === 200) {
          var data = JSON.parse(xhr.responseText);

          console.log("Response received: ", data);

          if (data.error) {
            alert(data.error);
          } else if (data.message) {
            alert(data.message);
            if (data.clearFields) { // Check if fields should be cleared
              clearAllFields();
            }
          } else {
            document.getElementById("consumerNo").value = data.consumerNo || "";
            document.getElementById("location").value = data.location || "";
            document.getElementById("name").value = data.name || "";
            document.getElementById("physector").value = data.addressL1 || "";
            document.getElementById("street").value = data.addressL1 || "";
            document.getElementById("qrno").value = data.addressL2 || "";
            document.getElementById("type").value = data.consumerType || "";
            document.getElementById("addressL1").value = data.addressL1 || "";
          }
        }
      };
      xhr.open("GET", "data.php?sector=" + sector + "&plot=" + plot, true);
      xhr.send();
    }
  }
});
```



```

document.getElementById("addressL2").value = data.addressL2 || "";
document.getElementById("addressL3").value = data.addressL3 || "";
document.getElementById("status").value = data.status || "";
document.getElementById("meterno").value = data.meterNo || "";
document.getElementById("metermake").value = data.meterMake || "";
document.getElementById("meterKW").value = data.meterKW || "";
document.getElementById("sancload").value = data.loadSanc || "";
document.getElementById("phase").value = data.phase || "";
document.getElementById("multifactor").value = data.multFactor || "";
document.getElementById("solarmeterno").value = data.solarMeterNo || "";
document.getElementById("solarmetermake").value = data.solarMeterMake || "";
document.getElementById("solarMeterKW").value = data.solarMeterKW || "";
document.getElementById("solarsancload").value = data.solarLoadSanc || "";
document.getElementById("solarphase").value = data.solarPhase || "";
document.getElementById("solarmultifactor").value = data.solarMultFactor || "";
document.getElementById("solarmeterstatus").value = data.solarMeterStatus || "";
document.getElementById("solarAmp").value = data.solarAmp || "";
document.getElementById("elecCode").value = data.elecCode || "";
document.getElementById("stdLightUnits").value = data.stdLightUnits || "";
document.getElementById("stdPowerUnits").value = data.stdPowerUnits || "";
document.getElementById("loadKW").value = data.loadKW || "";
document.getElementById("consumerType").value = data.consumerType || "";
document.getElementById("tariffCatg").value = data.tariffCatg || "";
document.getElementById("supplyPur").value = data.supplyPur || "";
document.getElementById("consumerCatg").value = data.consumerCatg || "";
document.getElementById("assdConsp").value = data.assdConsp || "";
document.getElementById("remarks").value = data.remarks || "";
document.getElementById("sourceTable").value = data.sourceTable || "";

```

```

if (data.meterNo) {
    document.getElementById("meterno").disabled = true;
    document.getElementById("metermake").disabled = true;
    document.getElementById("meterKW").disabled = true;
}

```

```

if (data.enableFields) {
    document.getElementById("street").disabled = false;
    document.getElementById("qrno").disabled = false;
    document.getElementById("qrtype").disabled = false;
}

```

```

console.log("Populated data in form:", {
    consumerNo: data.consumerNo,
    location: data.location,
    name: data.name,
    physector: data.addressL1,
    street: data.addressL1,
    qrno: data.addressL2,
    type: data.consumerType,
    addressL1: data.addressL1,
    addressL2: data.addressL2,
    addressL3: data.addressL3,
    status: data.status,
    meterno: data.meterNo,
    metermake: data.meterMake,
    meterKW: data.meterKW,
    sancload: data.loadSanc,
    phase: data.phase,
    multifactor: data.multFactor,
    solarmeterno: data.solarMeterNo,
    solarmetermake: data.solarMeterMake,
    solarMeterKW: data.solarMeterKW,
    solarsancload: data.solarLoadSanc,
    solarphase: data.solarPhase,
    solarmultifactor: data.solarMultFactor,
    solarmeterstatus: data.solarMeterStatus,

```

```

        solarAmp: data.solarAmp,
        elecCode: data.elecCode,
        stdLightUnits: data.stdLightUnits,
        stdPowerUnits: data.stdPowerUnits,
        loadKW: data.loadKW,
        consumerType: data.consumerType,
        tariffCatg: data.tariffCatg,
        supplyPur: data.supplyPur,
        consumerCatg: data.consumerCatg,
        assdConsp: data.assdConsp,
        remarks: data.remarks,
        sourceTable: data.sourceTable
    });
    }
}
};
xhr.open("GET", "fetchData.jsp?sector=" + sector + "&plot=" + plot, true);
xhr.send();
}
}

function clearAllFields() {
    var inputs = document.querySelectorAll("input[type='text']");
    inputs.forEach(function(input) {
        input.value = "";
    });
    document.getElementById("street").disabled = true;
    document.getElementById("qrno").disabled = true;
    document.getElementById("qrtype").disabled = true;
}

// Remove event listeners from sector and plot input fields
document.getElementById("sector").removeEventListener("input", fetchData);
document.getElementById("plot").removeEventListener("input", fetchData);

// Add event listener to the Search button
document.getElementById("searchButton").addEventListener("click", fetchData);

document.getElementById("clearButton").addEventListener("click", function() {
    clearAllFields();
});

document.getElementById("exitButton").addEventListener("click", function() {
    window.close();
});

document.querySelector("form").addEventListener("submit", function(event) {
    // Form validation before submission
    var requiredFields = ["sector", "plot", "consumerNo", "location", "name"];
    for (var i = 0; i < requiredFields.length; i++) {
        if (document.getElementById(requiredFields[i]).value === "") {
            alert("Please fill all required fields.");
            event.preventDefault();
            return;
        }
    }
}
});
});
</script>
</head>
<body>
    <div class="heading">
        
        <h2>Electricity Master Data Entry</h2>
    </div>

```

[illegible]

[illegible]

```

</div>
<div>
  <label for="solarphase">Solar Phase</label>&emsp;&emsp;&emsp;&VeryThinSpace;
  <input type="text" id="solarphase" class="small-input" name="solarphase">
</div>
<div>
  <label for="solarmultifactor">Solar Multi Factor</label>&emsp;&emsp;&emsp;
  <input type="text" id="solarmultifactor" class="small-input" name="solarmultifactor">
</div>
<div>
  <label for="solarmeterstatus">Solar Meter Status</label>
  <input type="text" id="solarmeterstatus" class="small-input" name="solarmeterstatus">
</div>
<div>
  <label for="solarAmp">Solar AMP</label>&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&
  <input type="text" id="solarAmp" class="small-input" name="solarAmp">
</div>
</div>
</div>
<div class="lastdiv">
  <div class="form-grid">
    <div>
      <label for="elecCode">Elec Code</label>&emsp;&emsp;&emsp;&emsp;&MediumSpace;
      <input type="text" id="elecCode" class="small-input" name="elecCode">
    </div>
    <div>
      <label for="stdLightUnits">Std Light Units</label>&emsp;&emsp;&nbsp;
      <input type="text" id="stdLightUnits" class="small-input" name="stdLightUnits">
    </div>
    <div>
      <label for="stdPowerUnits">Std Power Units</label>&emsp;&emsp;
      <input type="text" id="stdPowerUnits" class="small-input" name="stdPowerUnits">
    </div>
    <div>
      <label for="loadKW">Load KW</label>&emsp;&emsp;&emsp;&emsp;&emsp;&nbsp;
      <input type="text" id="loadKW" class="small-input" name="loadKW">
    </div>
    <div>
      <label for="consumerType">Consumer Type</label>&emsp;&emsp;&VeryThinSpace;&VeryThinSpace;
      <input type="text" id="consumerType" class="small-input" name="consumerType">
    </div>
    <div>
      <label for="tariffCatg">Tariff Category</label>&emsp;&emsp;&emsp;
      <input type="text" id="tariffCatg" class="small-input" name="tariffCatg">
    </div>
    <div>
      <label for="supplyPur">Supply Purpose</label>&emsp;&emsp;
      <input type="text" id="supplyPur" class="small-input" name="supplyPur">
    </div>
    <div>
      <label for="consumerCatg">Consumer Category</label>
      <input type="text" id="consumerCatg" class="small-input" name="consumerCatg">
    </div>
    <div>
      <label for="assdConsp">Assd Consumption</label>
      <input type="text" id="assdConsp" class="small-input" name="assdConsp">
    </div>
  </div>
</div>
</div>
<div class="clearfix"></div>
<div class="fle">
  <div class="remarks">
    <label for="remarks">Remarks</label>
    <input type="text" id="remarks" name="remarks">
  </div>
</div>

```

```
</div>
</form>
</body>
</html>
```

SaveDataServlet.jsp

```
<%@ page import="java.sql.*, java.io.*, javax.servlet.*, javax.servlet.http.*" %>
<%@ include file="DBcon.jsp" %>
<%@page contentType="text/html; charset=iso-8859-1" pageEncoding="iso-8859-1"%>

<%
if (request.getMethod().equalsIgnoreCase("POST")){
    String sector = request.getParameter("sector");
    String plot = request.getParameter("plot");
    String consumerNo = (request.getParameter("consumerNo") == null || request.getParameter("consumerNo").trim().isEmpty()) ?
"NA" : request.getParameter("consumerNo");
    String location = (request.getParameter("location") == null || request.getParameter("location").trim().isEmpty()) ? "NA" :
request.getParameter("location");
    String name = (request.getParameter("name") == null || request.getParameter("name").trim().isEmpty()) ? "NA" :
request.getParameter("name");
    String status = (request.getParameter("status") == null || request.getParameter("status").trim().isEmpty()) ? "NA" :
request.getParameter("status");
    String sancload = (request.getParameter("sancload") == null || request.getParameter("sancload").trim().isEmpty()) ? "0" :
request.getParameter("sancload");
    String multifactor = (request.getParameter("multifactor") == null || request.getParameter("multifactor").trim().isEmpty()) ? "0" :
request.getParameter("multifactor");
    String remarks = (request.getParameter("remarks") == null || request.getParameter("remarks").trim().isEmpty()) ? "NA" :
request.getParameter("remarks");
    String sourceTable = request.getParameter("sourceTable");

    Connection conn = null;
    PreparedStatement stmt = null;

    try{
        conn = getConnection();

        if ("ELEC_MASTER".equals(sourceTable)) {
            String updateSql = "UPDATE ELEC_MASTER SET CONSUMER_NO = ?, LOCATION = ?, NAME = ?, STATUS = ?, METER_NO = ?,
METER_MAKE = ?, METER_KW = ?, LOAD_SANC = ?, PHASE = ?, MULT_FACTOR = ?, SOLAR_METER_NO = ?, SOLAR_METER_MAKE =
?, SOLAR_METER_KW = ?, SOLAR_LOAD_SANC = ?, SOLAR_PHASE = ?, SOLAR_MULT_FACTOR = ?, SOLAR_METER_STATUS = ?,
CONSUMER_TYPE = ?, TARIFF_CATG = ?, CONSUMER_CATG = ?, REMARKS = ? WHERE SECTOR = ? AND PLOT = ?";
            stmt = conn.prepareStatement(updateSql);
            stmt.setString(1, consumerNo);
            stmt.setString(2, location);
            stmt.setString(3, name);
            stmt.setString(4, status);
            stmt.setString(5, (request.getParameter("meterno") == null || request.getParameter("meterno").trim().isEmpty()) ? "NA" :
request.getParameter("meterno"));
            stmt.setString(6, (request.getParameter("metermake") == null || request.getParameter("metermake").trim().isEmpty()) ? "NA" :
request.getParameter("metermake"));
            stmt.setString(7, (request.getParameter("meterKW") == null || request.getParameter("meterKW").trim().isEmpty()) ? "0" :
request.getParameter("meterKW"));
            stmt.setString(8, sancload);
            stmt.setString(9, (request.getParameter("phase") == null || request.getParameter("phase").trim().isEmpty()) ? "0" :
request.getParameter("phase"));
            stmt.setString(10, multifactor);
            stmt.setString(11, (request.getParameter("solarmeterno") == null || request.getParameter("solarmeterno").trim().isEmpty()) ?
"NA" : request.getParameter("solarmeterno"));
            stmt.setString(12, (request.getParameter("solarmetermake") == null ||
request.getParameter("solarmetermake").trim().isEmpty()) ? "NA" : request.getParameter("solarmetermake"));
            stmt.setString(13, (request.getParameter("solarMeterKW") == null || request.getParameter("solarMeterKW").trim().isEmpty()) ?
"0" : request.getParameter("solarMeterKW"));
        }
    }
}
```

```

        stmt.setString(14, (request.getParameter("solarsanload") == null || request.getParameter("solarsanload").trim().isEmpty()) ?
"0" : request.getParameter("solarsanload"));
        stmt.setString(15, (request.getParameter("solarphase") == null || request.getParameter("solarphase").trim().isEmpty()) ? "0" :
request.getParameter("solarphase"));
        stmt.setString(16, (request.getParameter("solarmultifactor") == null ||
request.getParameter("solarmultifactor").trim().isEmpty()) ? "0" : request.getParameter("solarmultifactor"));
        stmt.setString(17, (request.getParameter("solarmeterstatus") == null ||
request.getParameter("solarmeterstatus").trim().isEmpty()) ? "NA" : request.getParameter("solarmeterstatus"));
        stmt.setString(18, (request.getParameter("consumerType") == null || request.getParameter("consumerType").trim().isEmpty())
? "NA" : request.getParameter("consumerType"));
        stmt.setString(19, (request.getParameter("tariffCatg") == null || request.getParameter("tariffCatg").trim().isEmpty()) ? "NA" :
request.getParameter("tariffCatg"));
        stmt.setString(20, (request.getParameter("consumerCatg") == null || request.getParameter("consumerCatg").trim().isEmpty())
? "NA" : request.getParameter("consumerCatg"));
        stmt.setString(21, remarks);
        stmt.setString(22, sector);
        stmt.setString(23, plot);
    } else if ("TP_MAS".equals(sourceTable)) {
        String insertSql = "INSERT INTO C##ESTATE.ELEC_MASTER (SECTOR, PLOT, CONSUMER_NO, LOCATION, NAME, STATUS,
METER_NO, METER_MAKE, METER_KW, LOAD_SANC, PHASE, MULT_FACTOR, SOLAR_METER_NO, SOLAR_METER_MAKE,
SOLAR_METER_KW, SOLAR_LOAD_SANC, SOLAR_PHASE, SOLAR_MULT_FACTOR, SOLAR_METER_STATUS, CONSUMER_TYPE,
TARIFF_CATG, CONSUMER_CATG, REMARKS) " +
        "VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
        stmt = conn.prepareStatement(insertSql);
        stmt.setString(1, sector);           // SECTOR
        stmt.setString(2, plot);             // PLOT
        stmt.setString(3, consumerNo);       // CONSUMER_NO
        stmt.setString(4, location);         // LOCATION
        stmt.setString(5, name);             // NAME
        stmt.setString(6, status);           // STATUS
        stmt.setString(7, (request.getParameter("meterno") == null || request.getParameter("meterno").trim().isEmpty()) ? "NA" :
request.getParameter("meterno"));
        stmt.setString(8, (request.getParameter("metermake") == null || request.getParameter("metermake").trim().isEmpty()) ? "NA" :
request.getParameter("metermake"));
        stmt.setInt(9, Integer.parseInt((request.getParameter("meterKW") == null ||
request.getParameter("meterKW").trim().isEmpty()) ? "0" : request.getParameter("meterKW"))));
        stmt.setDouble(10, Double.parseDouble(sanload));
        stmt.setInt(11, Integer.parseInt((request.getParameter("phase") == null || request.getParameter("phase").trim().isEmpty()) ?
"0" : request.getParameter("phase")));
        stmt.setDouble(12, Double.parseDouble(multifactor));
        stmt.setString(13, (request.getParameter("solarmeterno") == null || request.getParameter("solarmeterno").trim().isEmpty()) ?
"NA" : request.getParameter("solarmeterno"));
        stmt.setString(14, (request.getParameter("solarmetermake") == null ||
request.getParameter("solarmetermake").trim().isEmpty()) ? "NA" : request.getParameter("solarmetermake"));
        stmt.setInt(15, Integer.parseInt((request.getParameter("solarMeterKW") == null ||
request.getParameter("solarMeterKW").trim().isEmpty()) ? "0" : request.getParameter("solarMeterKW"))));
        stmt.setDouble(16, Double.parseDouble((request.getParameter("solarsanload") == null ||
request.getParameter("solarsanload").trim().isEmpty()) ? "0" : request.getParameter("solarsanload"))));
        stmt.setInt(17, Integer.parseInt((request.getParameter("solarphase") == null ||
request.getParameter("solarphase").trim().isEmpty()) ? "0" : request.getParameter("solarphase"))));
        stmt.setDouble(18, Double.parseDouble((request.getParameter("solarmultifactor") == null ||
request.getParameter("solarmultifactor").trim().isEmpty()) ? "0" : request.getParameter("solarmultifactor"))));
        stmt.setString(19, (request.getParameter("solarmeterstatus") == null ||
request.getParameter("solarmeterstatus").trim().isEmpty()) ? "NA" : request.getParameter("solarmeterstatus"));
        stmt.setString(20, (request.getParameter("consumerType") == null || request.getParameter("consumerType").trim().isEmpty())
? "NA" : request.getParameter("consumerType"));
        stmt.setString(21, (request.getParameter("tariffCatg") == null || request.getParameter("tariffCatg").trim().isEmpty()) ? "NA" :
request.getParameter("tariffCatg"));
        stmt.setString(22, (request.getParameter("consumerCatg") == null || request.getParameter("consumerCatg").trim().isEmpty())
? "NA" : request.getParameter("consumerCatg"));
        stmt.setString(23, remarks);
    } else {
        throw new SQLException("Invalid source table: " + sourceTable);
    }

    int rowsUpdated = stmt.executeUpdate();

```

```

%>
<!DOCTYPE html>
<html>
<head>
<title>Save Data</title>
<style>
  body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
  }
  .container {
    background-color: #fff;
    padding: 20px;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    text-align: center;
  }
  .container p {
    font-size: 18px;
    margin: 20px 0;
  }
  .button {
    background-color: #007BFF;
    color: white;
    padding: 10px 20px;
    text-decoration: none;
    border-radius: 5px;
    transition: background-color 0.3s ease;
  }
  .button:hover {
    background-color: #0056b3;
  }
</style>
</head>
<body>
<div class="container">
  <p><%= (rowsUpdated > 0) ? "Data updated successfully" : "Operation Done" %></p>
  <a href="javascript:window.history.back();" class="button">Back</a>
</div>
</body>
</html>
<%
} catch (Exception e) {
  e.printStackTrace();
%>
<!DOCTYPE html>
<html>
<head>
<title>Error</title>
<style>
  body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
  }
  .container {
    background-color: #fff;

```



```

padding: 20px;
border-radius: 5px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
text-align: center;
}
.container p {
font-size: 18px;
margin: 20px 0;
color: #e74c3c;
}
.button {
background-color: #007BFF;
color: white;
padding: 10px 20px;
text-decoration: none;
border-radius: 5px;
transition: background-color 0.3s ease;
}
.button:hover {
background-color: #0056b3;
}
</style>
</head>
<body>
<div class="container">
<p>Error: <%= e.getMessage() %></p>
<a href="javascript:window.history.back();" class="button">Back</a>
</div>
</body>
</html>
<%
} finally {
if (stmt != null) try { stmt.close(); } catch (SQLException e) { e.printStackTrace(); }
if (conn != null) try { conn.close(); } catch (SQLException e) { e.printStackTrace(); }
}
} else {
%>
<!DOCTYPE html>
<html>
<head>
<title>Error</title>
<style>
body {
font-family: Arial, sans-serif;
background-color: #f4f4f4;
display: flex;
justify-content: center;
align-items: center;
height: 100vh;
margin: 0;
}
.container {
background-color: #fff;
padding: 20px;
border-radius: 5px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
text-align: center;
}
.container p {
font-size: 18px;
margin: 20px 0;
color: #e74c3c;
}
.button {
background-color: #007BFF;
color: white;

```

```

padding: 10px 20px;
text-decoration: none;
border-radius: 5px;
transition: background-color 0.3s ease;
}
.button:hover {
background-color: #0056b3;
}
</style>
</head>
<body>
<div class="container">
<p>Invalid request method.</p>
<a href="javascript:window.history.back();" class="button">Back</a>
</div>
</body>
</html>
<%
}
%>

```

fetchData.jsp

```

<%@ page import="java.sql.*, org.json.JSONObject" %>
<%@ include file="DBcon.jsp" %>
<%
String sector = request.getParameter("sector");
String plot = request.getParameter("plot");

Connection conn = null;
PreparedStatement stmt = null;
ResultSet rs = null;
JSONObject jsonResponse = new JSONObject();

try {
conn = getConnection();

// Check if record exists in ELEC_MASTER
String sqlCheckELEC = "SELECT * FROM C##ESTATE.ELEC_MASTER WHERE SECTOR = ? AND PLOT = ?";
stmt = conn.prepareStatement(sqlCheckELEC);
stmt.setString(1, sector);
stmt.setString(2, plot);
rs = stmt.executeQuery();

if (rs.next()) {
// Record exists in ELEC_MASTER
jsonResponse.put("sourceTable", "ELEC_MASTER");
jsonResponse.put("sector", rs.getString("SECTOR"));
jsonResponse.put("plot", rs.getString("PLOT"));
jsonResponse.put("consumerNo", rs.getString("CONSUMER_NO"));
jsonResponse.put("location", rs.getString("LOCATION"));
jsonResponse.put("enableFields", true);

jsonResponse.put("addressL1", rs.getString("ADDRESS_L1"));
jsonResponse.put("addressL2", rs.getString("ADDRESS_L2"));
jsonResponse.put("addressL3", rs.getString("ADDRESS_L3"));
jsonResponse.put("meterNo", rs.getString("METER_NO"));
jsonResponse.put("meterMake", rs.getString("METER_MAKE"));
jsonResponse.put("meterKW", rs.getString("METER_KW"));
jsonResponse.put("name", rs.getString("NAME"));
jsonResponse.put("consumerType", rs.getString("CONSUMER_TYPE"));
jsonResponse.put("phase", rs.getString("PHASE"));
jsonResponse.put("supplyPur", rs.getString("SUPPLY_PUR"));
jsonResponse.put("tariffCatg", rs.getString("TARIFF_CATG"));
jsonResponse.put("consumerCatg", rs.getString("CONSUMER_CATG"));
jsonResponse.put("loadSanc", rs.getString("LOAD_SANC"));
}
}
}

```

```

jsonResponse.put("multFactor", rs.getString("MULT_FACTOR"));
jsonResponse.put("assdConsp", rs.getString("ASSD_CONSPN"));
jsonResponse.put("status", rs.getString("STATUS"));
jsonResponse.put("remarks", rs.getString("REMARKS"));
jsonResponse.put("solarMeterMake", rs.getString("SOLAR_METER_MAKE"));
jsonResponse.put("solarMeterKW", rs.getString("SOLAR_METER_KW"));
jsonResponse.put("solarLoadSanc", rs.getString("SOLAR_LOAD_SANC"));
jsonResponse.put("solarPhase", rs.getString("SOLAR_PHASE"));
jsonResponse.put("solarMultFactor", rs.getString("SOLAR_MULT_FACTOR"));
jsonResponse.put("solarMeterStatus", rs.getString("SOLAR_METER_STATUS"));
jsonResponse.put("solarMeterNo", rs.getString("SOLAR_METER_NO"));
jsonResponse.put("enableFields", true);
System.out.println("Debug: SECTOR value is " + rs.getString("SECTOR"));
System.out.println("Debug: LOCATION value is " + rs.getString("LOCATION"));
} else {
    // Record does not exist in ELEC_MASTER, check TP_MAS
    rs.close();
    stmt.close();
    String sqlCheckTPMas = "SELECT SECTOR, PLOT, SECTOR || '/' || PLOT AS CONSUMER_NO, LOCATION, NAME, ELEC_CODE,
STD_LUNITS, STD_PUNITS, LOAD_KW, STATUS, DECODE(SECTOR, 'AB', NULL, 'TRADE') AS TRADE, SUBSTR(NATURE, 1, 4) AS NATURE
FROM C##ESTATE.TP_MAS WHERE SECTOR = ? AND PLOT = ?";
    stmt = conn.prepareStatement(sqlCheckTPMas);
    stmt.setString(1, sector);
    stmt.setString(2, plot);
    rs = stmt.executeQuery();

    if (rs.next()) {
        jsonResponse.put("sourceTable", "TP_MAS");
        jsonResponse.put("sector", rs.getString("SECTOR"));
        jsonResponse.put("plot", rs.getString("PLOT"));
        jsonResponse.put("consumerNo", rs.getString("CONSUMER_NO"));
        jsonResponse.put("location", rs.getString("LOCATION"));
        jsonResponse.put("name", rs.getString("NAME"));
        jsonResponse.put("elecCode", rs.getString("ELEC_CODE"));
        jsonResponse.put("stdLightUnits", rs.getString("STD_LUNITS"));
        jsonResponse.put("stdPowerUnits", rs.getString("STD_PUNITS"));
        jsonResponse.put("loadKW", rs.getString("LOAD_KW"));
        jsonResponse.put("status", rs.getString("STATUS"));
        jsonResponse.put("trade", rs.getString("TRADE") != null ? rs.getString("TRADE") : "");
        jsonResponse.put("nature", rs.getString("NATURE"));
        jsonResponse.put("clearFields", true);

        if ("D".equals(rs.getString("STATUS"))) {
            jsonResponse.put("message", "This is a terminated party. Please check.");
            out.print(jsonResponse.toString());
            return;
        }

        if ("N".equals(rs.getString("ELEC_CODE"))) {
            jsonResponse.put("message", "No electricity billing for this party.");
            out.print(jsonResponse.toString());
            return;
        }

        String sectorValue = rs.getString("SECTOR");
        String locationValue = rs.getString("LOCATION");
        if ("01,02,04,05,06,07,08,09,10,RS,CC,HS,MS".contains(sectorValue)) {
            jsonResponse.put("addressL1", sectorValue);
        } else if ("LN".equals(sectorValue)) {
            if (locationValue != null && locationValue.length() >= 2) {
                jsonResponse.put("addressL1", locationValue.substring(0, 2));
            } else {
                jsonResponse.put("addressL1", "");
            }
        }
    }
}

```

```

String elecCode = rs.getString("ELEC_CODE");
String tariffCatg = "";
if ("3,4,5,6,7".contains(elecCode)) {
    tariffCatg = "LT1";
} else if ("0,2,9,8".contains(elecCode)) {
    tariffCatg = "LT2";
} else if ("1".equals(elecCode)) {
    tariffCatg = "LT3";
}
jsonResponse.put("tariffCatg", tariffCatg);

if ("HOUS,LEAS,LICE".contains(rs.getString("NATURE"))) {
    String sqlMeter = "SELECT METER_NO, METER_MAKE, METER_KW FROM C##ESTATE.ELEC_METER_MASTER_CURR
WHERE SECTOR || STREET || QRNO || QRTYPE = ?";
    PreparedStatement meterStmt = conn.prepareStatement(sqlMeter);
    meterStmt.setString(1, rs.getString("LOCATION"));
    ResultSet meterRs = meterStmt.executeQuery();

    if (meterRs.next()) {
        jsonResponse.put("meterNo", meterRs.getString("METER_NO"));
        jsonResponse.put("meterMake", meterRs.getString("METER_MAKE"));
        jsonResponse.put("meterKW", meterRs.getString("METER_KW"));
    } else {
        jsonResponse.put("meterNo", "");
        jsonResponse.put("meterMake", "");
        jsonResponse.put("meterKW", "");
    }
    meterRs.close();
    meterStmt.close();
} else {
    jsonResponse.put("message", "No such master exists.");
    jsonResponse.put("clearFields", true); // Add this line
}
}
} catch (Exception e) {
    jsonResponse.put("error", e.getMessage());
} finally {
    try {
        if (rs != null) rs.close();
        if (stmt != null) stmt.close();
        if (conn != null) conn.close();
    } catch (SQLException e) {
        jsonResponse.put("error", e.getMessage());
    }
}

out.print(jsonResponse.toString());
%>

```

REFERENCES

1. [Oracle Documentation](#)
2. [Java Documentation](#)
3. [W3Schools](#)
4. [Stack Overflow](#)
5. [Eclipse Documentation](#)
6. [Toad for Oracle Documentation](#)
7. [Apache Tomcat Documentation](#)
8. Company Policies and Guidelines

ACKNOWLEDGEMENT

I express my sincere gratitude to Shri Chandu Tembhurne, DGM(C&IT), Bhilai Steel Plant, Bhilai, for providing the opportunity to work on this project and for his valuable insights throughout its execution.

I am deeply thankful to Mr. Tejkaran Singh Hans, AGM(C&IT), Bhilai Steel Plant, Bhilai, for his guidance, support, and mentorship, which were invaluable in shaping this project.

I extend my appreciation to the entire team at Bhilai Steel Plant, whose expertise and assistance were instrumental in overcoming challenges and achieving the project's objectives.

I would also like to acknowledge the technical support provided by the IT department and the infrastructure facilities at Bhilai Steel Plant.

Lastly, I am grateful to my family and friends for their unwavering support and encouragement throughout this endeavor.