CUSTOMER SYNC

Application

ABSTRACT

This Customer Management System performs CRUD operations using JSP/Servlet for the backend and MySQL for the database.

Purushotham Palla

Full Stack Web Developer

Customer Management System Documentation

Contents

1. Introduction

- **Description:** This project is a CRUD (Create, Read, Update, Delete) application for managing customer data. The system utilizes a MySQL database, with the backend developed using JSP/Servlet and the frontend built with HTML, CSS. The application includes secure JWT-based authentication and integrates with a remote API for synchronizing customer data.

2. Technologies Used

- **Database: MySQL:**
  An overview of the MySQL database used to store customer data, including the schema design and key tables.
- **Backend: JSP/Servlet:**
  Describes the backend technologies used, focusing on the choice between JSP/Servlet. This section covers the rationale for the chosen technology and its role in the application.
- **Frontend: HTML, CSS:**
  A brief introduction to the frontend stack, explaining the use of HTML for structure, CSS for styling.
- **Authentication: JWT (JSON Web Tokens):**
  An explanation of JWT authentication, its importance in securing the application, and how it is implemented.

3. Data-Base Development

Create the Users Table: Ensure your `users` table in MySQL has columns for an auto-incremented ID, login ID, and password. You can also add additional fields as needed.

```
sql
CREATE TABLE users (
    id INT PRIMARY KEY AUTO_INCREMENT,
    login_id VARCHAR(255) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL
);
```

phurushotham2001@gmail.com

Create the CustomerDetails Table:

here is the SQL command to create the `customerdetails` table with the specified columns:

```
CREATE TABLE customerdetails (
   id INT AUTO_INCREMENT PRIMARY KEY,
   firstname VARCHAR(255) NOT NULL,
   lastname VARCHAR(255) NOT NULL,
   street VARCHAR(255),
   address VARCHAR(255),
   city VARCHAR(255),
   state VARCHAR(255),
   email VARCHAR(255) NOT NULL,
   phone VARCHAR(20),
   uuid VARCHAR(250)
);
```

4. Backend Development

- **Setting Up the Backend Environment:**
  Instructions for setting up the backend development environment, including tools, dependencies, and configuration settings.

Follow some steps:

Creating java project

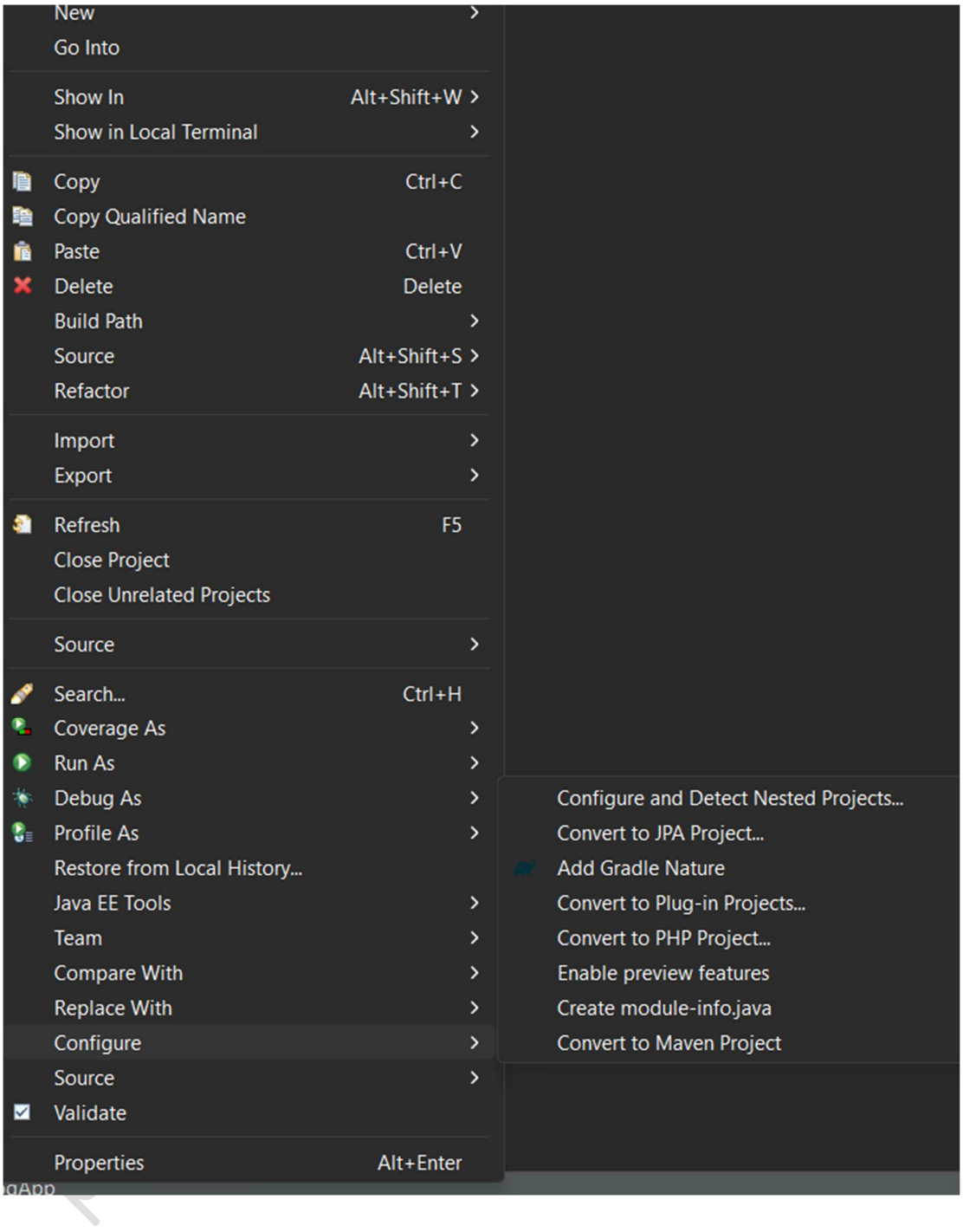
Follow some steps:

Create Java Dynamic Web Project

Click on project

It shows

purushotham2001@gmail.com



Click on configure and convert to maven project

In pom.xml it shows and add dependencies

Add Dependencies

For Json

```
<dependency>
        <groupId>org.json</groupId>
        <artifactId>json</artifactId>
        <version>20231013</version>
</dependency>
```

For httpclient

```
<dependency>
        <groupId>org.apache.httpcomponents</groupId>
        <artifactId>httpclient</artifactId>
        <version>4.5.13</version>
    </dependency>
```

For Jaxb-api

```
<dependency>
        <groupId>javax.xml.bind</groupId>
        <artifactId>jaxb-api</artifactId>
        <version>2.3.1</version>
    </dependency>
```

purushotham2001@gmail.com

For JWT

```
<dependency>
        <groupId>io.jsonwebtoken</groupId>
        <artifactId>jjwt</artifactId>
        <version>0.9.1</version>
    </dependency>
```

For mysql

```
<dependency>
                <groupId>mysql</groupId>
                <artifactId>mysql-connector-java</artifactId>
                <version>8.0.30</version>
    </dependency >
```

At final your code looks like

```
<project
xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSche
ma-instance"
xsi:schemaLocation="http://maven.apache.o
rg/POM/4.0.0
https://maven.apache.org/xsd/maven-
4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>CustomerSync</groupId>
  <artifactId>CustomerSync</artifactId>
  <version>0.0.1-SNAPSHOT</version>
```

gotham2001@gmail.com

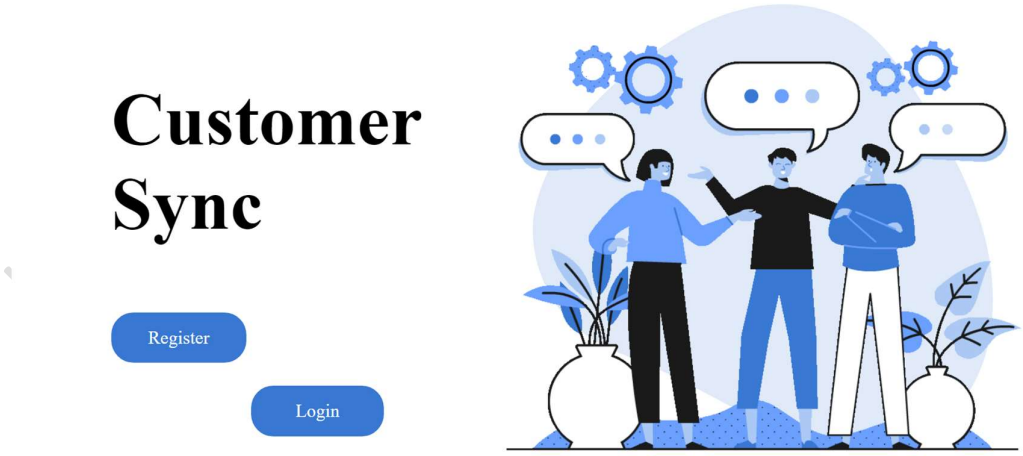
```
  <packaging>war</packaging>
  <build>
    <plugins>
      <plugin>
        <artifactId>maven-compiler-
plugin</artifactId>
        <version>3.8.1</version>
        <configuration>
          <release>17</release>
        </configuration>
      </plugin>
      <plugin>
        <artifactId>maven-war-
plugin</artifactId>
        <version>3.2.3</version>
      </plugin>
    </plugins>
  </build>


  <dependencies>
      <!--
https://mvnrepository.com/artifact/io.jso
nwebtoken/jjwt -->
    <dependency>
        <groupId>io.jsonwebtoken</groupId>
        <artifactId>jjwt</artifactId>
        <version>0.9.1</version>
    </dependency>

    <dependency>
          <groupId>mysql</groupId>
          <artifactId>mysql-connector-
java</artifactId>
```

```
            <version>8.0.30</version>
    </dependency >

    <!--
https://mvnrepository.com/artifact/javax.
xml.bind/jaxb-api -->
    <dependency>
        <groupId>javax.xml.bind</groupId>
        <artifactId>jaxb-api</artifactId>
        <version>2.3.1</version>
    </dependency>

    <!--
https://mvnrepository.com/artifact/org.ap
ache.httpcomponents/httpclient -->
    <dependency>

<groupId>org.apache.httpcomponents</group
Id>

<artifactId>httpclient</artifactId>
        <version>4.5.13</version>
    </dependency>

    <!--
https://mvnrepository.com/artifact/org.js
on/json -->
    <dependency>
        <groupId>org.json</groupId>
        <artifactId>json</artifactId>
        <version>20231013</version>
    </dependency>
```

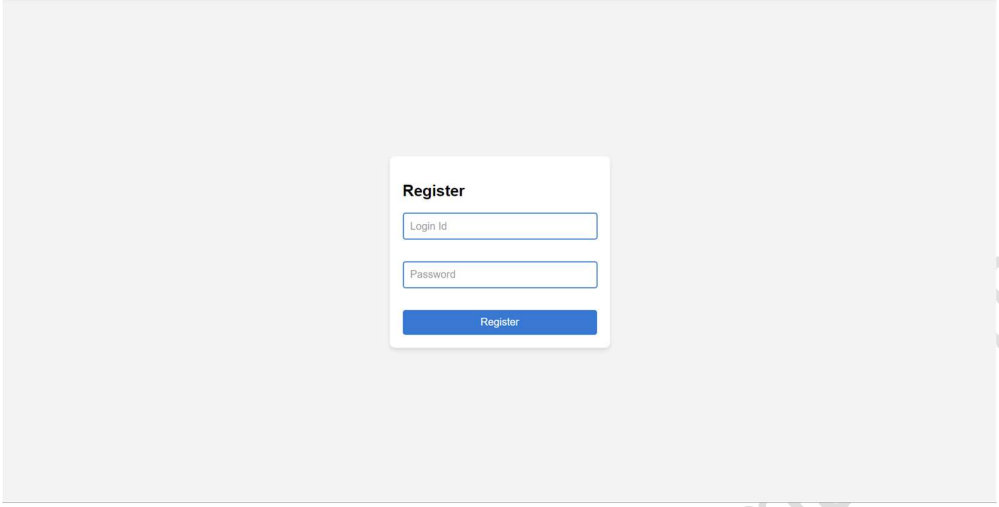
```
  </dependencies>


</project>
```

ham2001@gmail.

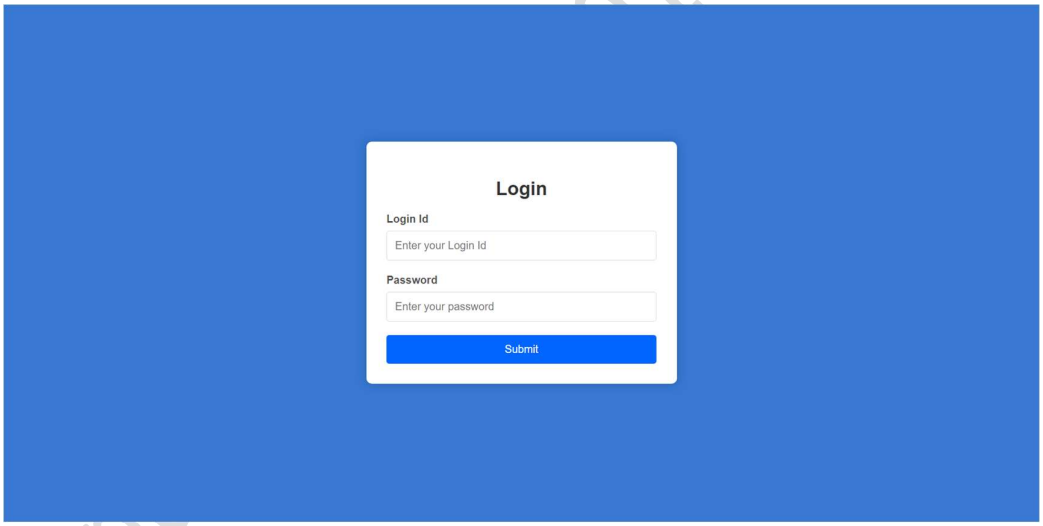
Index.jsp



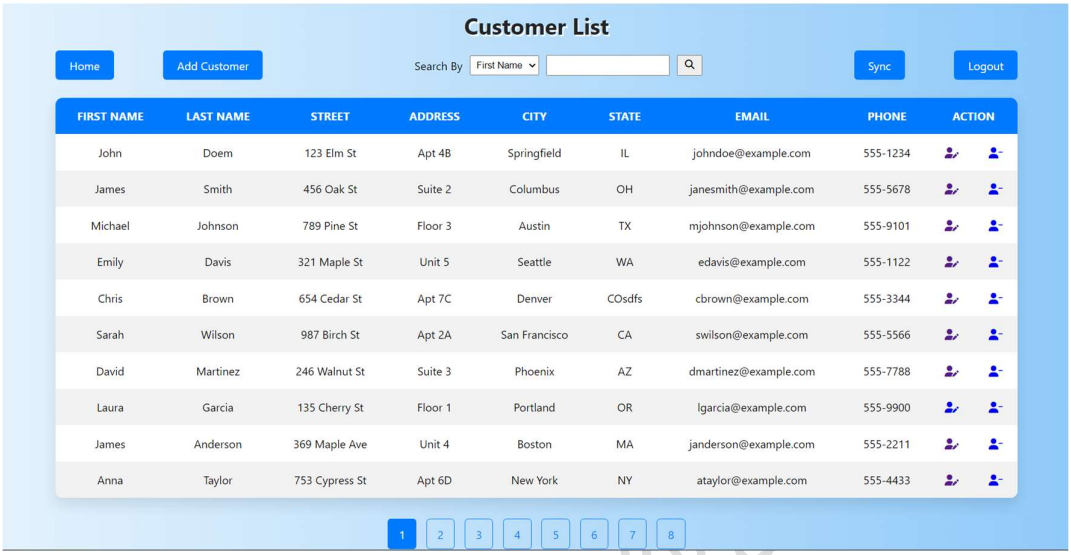
Register.jsp



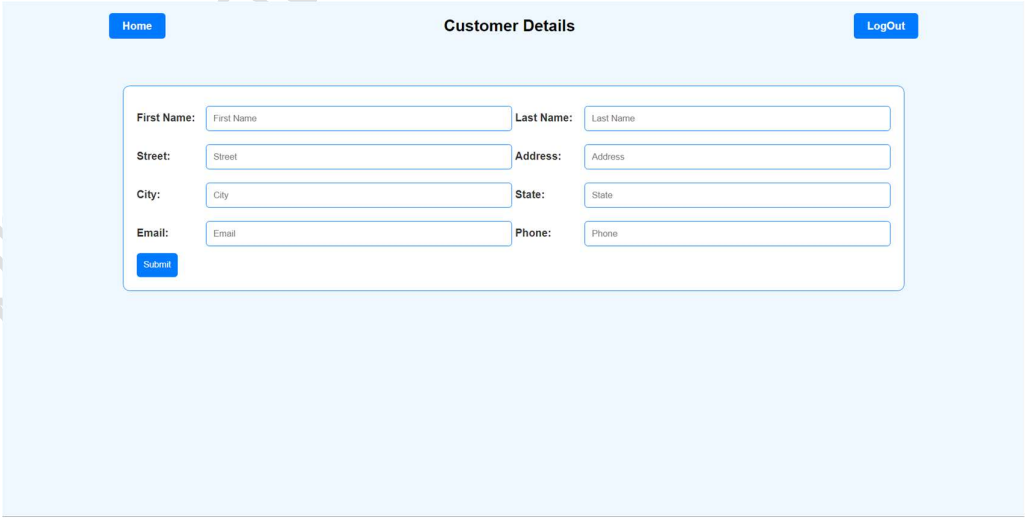
Login.jsp



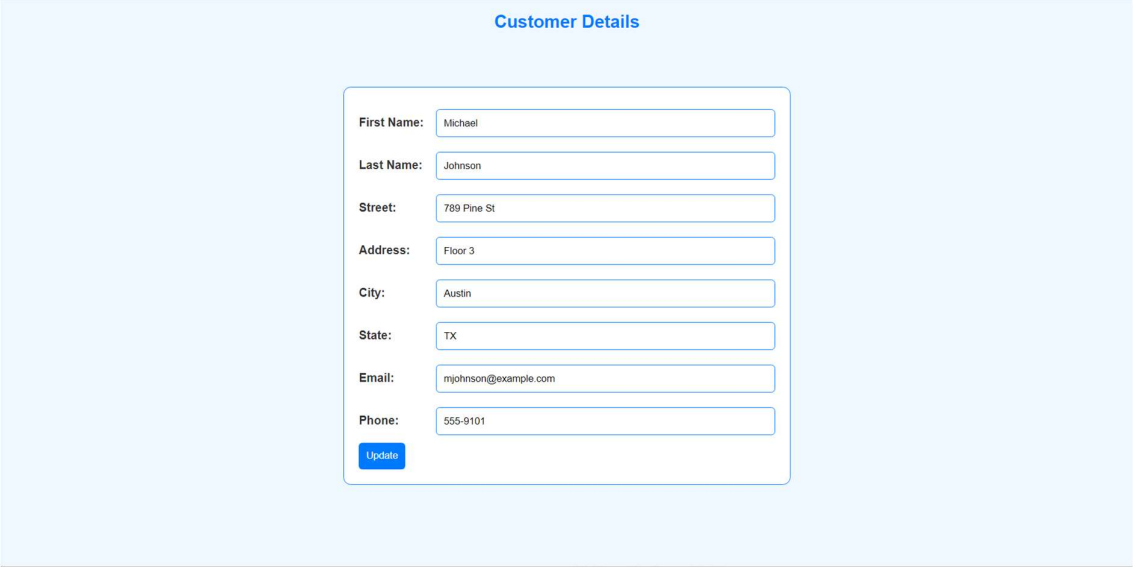
customerlist.jsp



- API Specifications
    - **Create Customer:** Detailed specifications for the API that handles creating new customer records.



- Update Customer



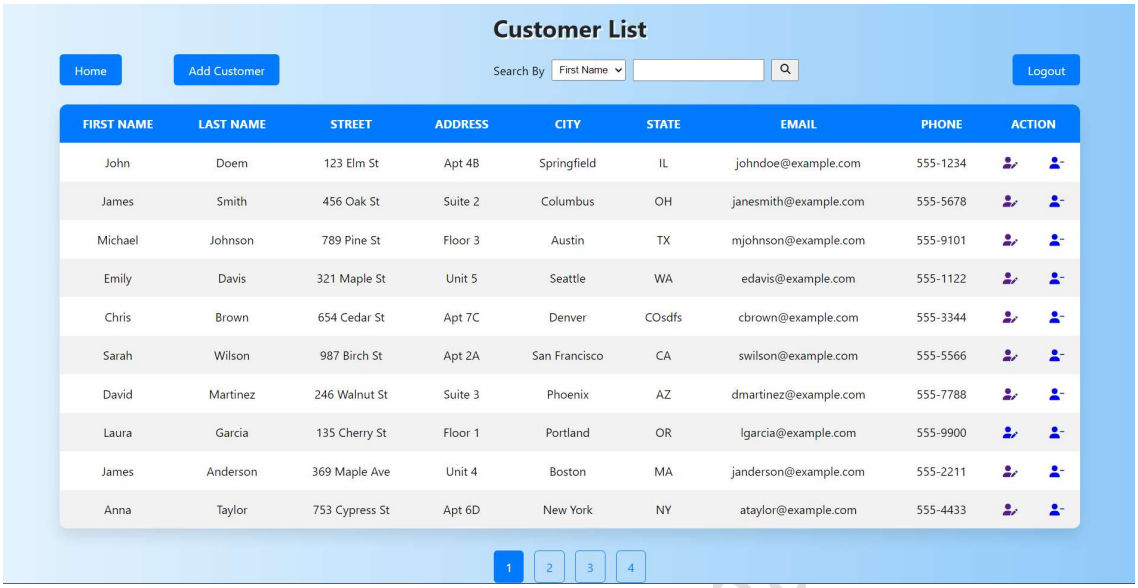
- Get Customer List (with Pagination, Sorting, and Searching)
- Get Customer by ID
- Delete Customer

- JWT Authentication Implementation

4. Data Synchronization Feature

- Integration with Remote API
- Sync Button Implementation
- Data Update Logic

purushotham2001@



Project Git-hub Link : https://github.com/Purushotham-Palla/customersync

Project video link :

https://drive.google.com/file/d/1FMM8hlaWRSFCO4ZJ8yMFvEcw8lkZVXf_/view?usp=drivesdk

ppurushotham2001@gm