

L1 – SPRING AND HIBERNATE LAB ASSESSMENT

Name: Nagarjun N S

Emp ID: 12228

Please perform the below tasks for the given case study which is assigned to your name.

Instructions:

- 1. Analyze the given problem using **W3H** technique.
- 2. Write an **algorithm** and draw the **flowchart** for the application.
- 3. Prepare **UML diagrams** for the given scenario [**Class Diagram, Use Case Diagram & Sequence Diagram**].
- 4. Create a web-based application for the given scenario using **Spring MVC and Hibernate**.
- 5. Use **MySQL** as your database.
- 6. Perform all the **CRUD** operations.
- 7. Make the application as user interactive and responsive.
- 8. Apply style using **CSS / Bootstrap**.
- 9. Perform the validations using **Javascript**.
- 10. Maintain the **Code Quality and the Coding Standard**.

Question:

| | | |
|-------|----------------------|--------------------------|
| 12228 | Nagarjun Suresh Babu | Tender Management System |
|-------|----------------------|--------------------------|

W3H:

| PROJECT NAME: Tender Management System | |
|---|---|
| What? 1 | How? 2 |
| <p>1. What are the modules are required? Ans: a) Admin (System) Module b) User Module</p> <p>2. What are the responsibilities of Admin (System)? Ans: a) Admin (System) will handle all the CRUD operations of the user database and the Tender database.</p> <p>3. What can the user do? Ans: a) The user can be able to bid for the property. b) User can be able to allocate and win the tender agreement of a property by providing the lowest amount quoted than others. c) User can be able to see the projects came for the Tender. d) User can be able to see the highest bid for a Tender. e) User can be able to see their profile. f) User can register themselves by providing details of their proofs like AADHAR number, PAN number, etc for verification</p> <p>4. What can the user see in their profile tab? Ans: a) User can be able to see their personal details. b) User can be able to see their Tender bidding history. c) They can update details of their account.</p> <p>5. What happens after the Tender is confirmed? Ans: Once the Tender is confirmed, the user is informed via SMS and the application notification, and they can pay the quoted amount to acquire the Tender.</p> <p>6. How many Tenders a user can avail? Is there any limit to it? Ans: A user can bid for many tenders. There is no limit to it.</p> <p>7. How to make the payment? Ans: Once the Tender is confirmed, then the user can pay via NEFT or in-person, because the amount will be huge.</p> | <p>1) Admin (System): a) Login: Method 1: The admin can login using their adminID, Email and password. Method 2: The admin can login using their username and password. Method 3: The admin can login to their account using social media accounts.</p> <p>b) Adding Tenders: Method 1: Admin can also add Tenders by using the TenderID, Tender Name, Scope of the tender, Tender Description, Tender provider’s Name and emailID and the location of the project and with budget quote Method 2: Admin can add Tenders with their Tender ID, Tender description and budget.</p> <p>c) Update / delete / view Tender Details: Method 1: Admin can be able to Update / delete / view Tender details by providing separate buttons with appropriate details. Method 2: Admin can be able to Update / delete / view Tender details by using the drop-down list with appropriate details.</p> <p>d) CRUD on User Accounts: Method 1: Admin can be able to Update / delete / view User account details by providing separate buttons. Method 2: Admin can be able to Update / delete / view User account details by using the drop-down list.</p> <p>2) User: a) Login: Method 1: The user can login by using their email and password. Method 2: The user can login by using mobile number and name and customer ID. Method 3: The user can login to their account using social media accounts.</p> <p>b) CRUD Operations: Method 1: User can perform CRUD Operations for their account settings and their personal details with separate buttons. Method 2: User can perform CRUD Operations for their account settings and personal details using a drop-down list and a confirm button.</p> <p>c) Check Tender status: Method 1: User can be able to check their Tender status using the “Bid for a Tender” option in the menu. Method 2: User can check their Tender status by calling the Tender Office.</p> |

| | |
|---|--|
| <p>1) Admin (System):</p> <p>a) Login: Method 1: The admin can login using their adminID, Email and password. (This method is used because the originality of the admin is verified).</p> <p>b) Adding Tenders: Method 1: Admin can also add Tenders by using the TenderID, Tender Name, Scope of the tender, Tender Description, Tender provider’s Name and emailID and the location of the project and with budget quote (This method is used because the admin can add the Tender by providing the Tender details.)</p> <p>c) Update / delete / view Tender Details: Method 1: Admin can be able to Update / delete / view Tender details by providing separate buttons with appropriate details. (Providing separate buttons with separate functionalities will help to use the application easily.)</p> <p>d) CRUD on User Accounts: Method 1: Admin can be able to Add / Update / Delete / View User account details by providing separate buttons. (Providing separate buttons with separate functionalities will help to use the application easily. And if in case some users don’t know to register themselves, they can use this method)</p> <p>2) User:</p> <p>a) Login: Method 1: The user can login by using their email and password. (Login to the system by using their email and password will be a hassle-free experience and results in a secure way of authentication. Because the email ID is already a unique and verified one.)</p> <p>b) CRUD Operations: Method 1: User can perform CRUD Operations for their account settings and their personal details with separate buttons. (Providing separate buttons with separate functionalities will help to use the application easily.)</p> <p>c) Check Tender status: Method 1: User can be able to check their Tender status using the “Bid for a Tender” option in the menu. (By providing a separate tracking status for each Tender will also help the customer to access Tender details easily and the user doesn’t need to bother by thinking questions like “what happened to the Tender I bided?”,etc.. By this, the tracking system will be very much easier for users.)</p> | <p>1) Admin:</p> <p>a) Login: Method 2: The admin can login using their username and password. (Admin can’t login using their username and password as the originality is not verified.)</p> <p>Method 3: The admin can login to their account using social media accounts. (Using Social media accounts to login will not be a feasible way and it is less secure).</p> <p>b) Adding Tenders: Method 2: Admin can add Tenders with their Tender ID, Tender description and budget. (If a Tender is added using the basic details like ID, description and budget, then these details are not sufficient for users to quote the budget.)</p> <p>c) Update / delete / view Tender Details Method 2: Admin can be able to Update / delete / view Tender details by using the drop-down list with appropriate details. (The drop-down list feature is not feasible, and it is hard to use, and it is time-consuming function too).</p> <p>d) CRUD on User Accounts: Method 2: Admin can be able to Update / delete / view User account details by using the drop-down list. (The drop-down list feature is not feasible, and it is hard to use, and it is time-consuming function too. Every time, the user must make 2 clicks for confirming an action.</p> <p>2) User:</p> <p>a) Login: Method 2: The user can login by using mobile number and name and customer ID. (User login using mobile number, name and customerID will not be much secure as email and password)</p> <p>Method 3: The user can login to their account using social media accounts. (Using social media accounts to login will be a security threat. Many personal details will be provided while sending a courier. Those details might have a chance to be leaked and unauthorized usage)</p> <p>b) CRUD Operations: Method 2: User can perform CRUD Operations for their account settings and personal details using a drop-down list and a confirm button. (The drop-down list feature is not feasible, and it is hard to use, and it is time-consuming function too).</p> <p>c) Check Tender status: Method 2: User can check their Tender status by calling the Tender Office. (If a user needs to call the office to track the Tender status, then there will be a huge traffic of calling to the office which results in loss of time and loss of productivity to the owner. So, it is not feasible.)</p> |
| Why? 3 | Why not? 4 |

ALGORITHMS:**ADMIN:**

Step 1: Start

Step 2: Go to Login page

Step 3: Enter the credentials

Step 4: After completing step 3, click login

Step 5: Go to homepage

Step 6: Click on “Add Tenders”

Step 7: Provide Tender Details and submit

Step 8: Logout

Step 9: Stop

USER:**Tender Bidding:**

step 1: Start

step 2: Login by entering your credentials

step 3: Go to homepage

step 4: Select “Bid for a Tender” option

step 5: Select a project

step 6: Enter the quote amount

step 7: Click “Submit”

step 8: Download Receipt

step 9: Logout and Wait for the result

step 10: stop

Update User Details:

step 1: Start

step 2: Login by entering your credentials

step 3: Go to homepage

step 4: select “Change Account Details”

step 5: Enter User ID

step 6: Enter your Mobile number

step 7: Enter OTP

step 8: Click Submit

step 9: Update Account details

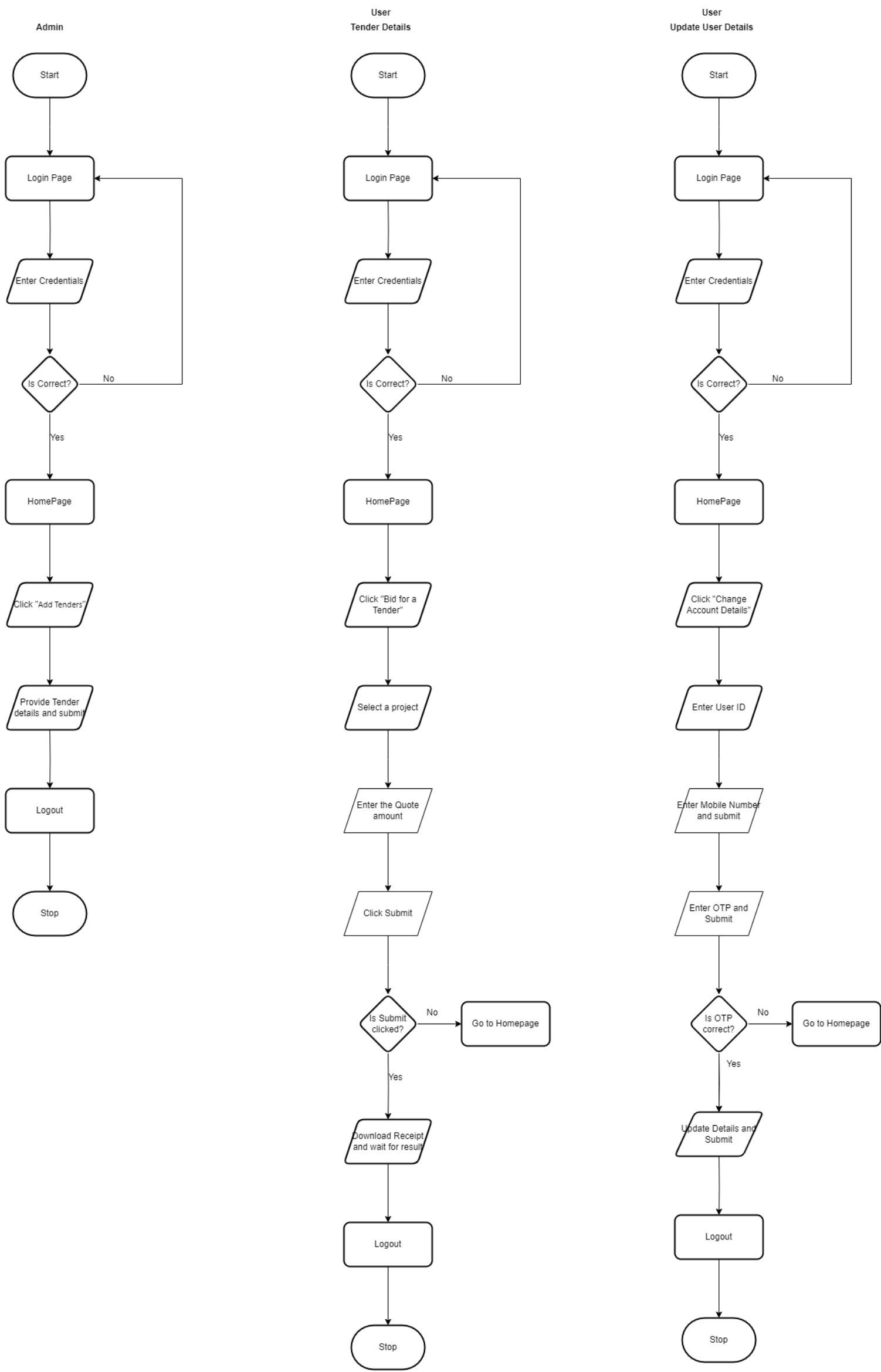
step 10: Confirm it

step 11: Success msg displayed

step 12: Logout

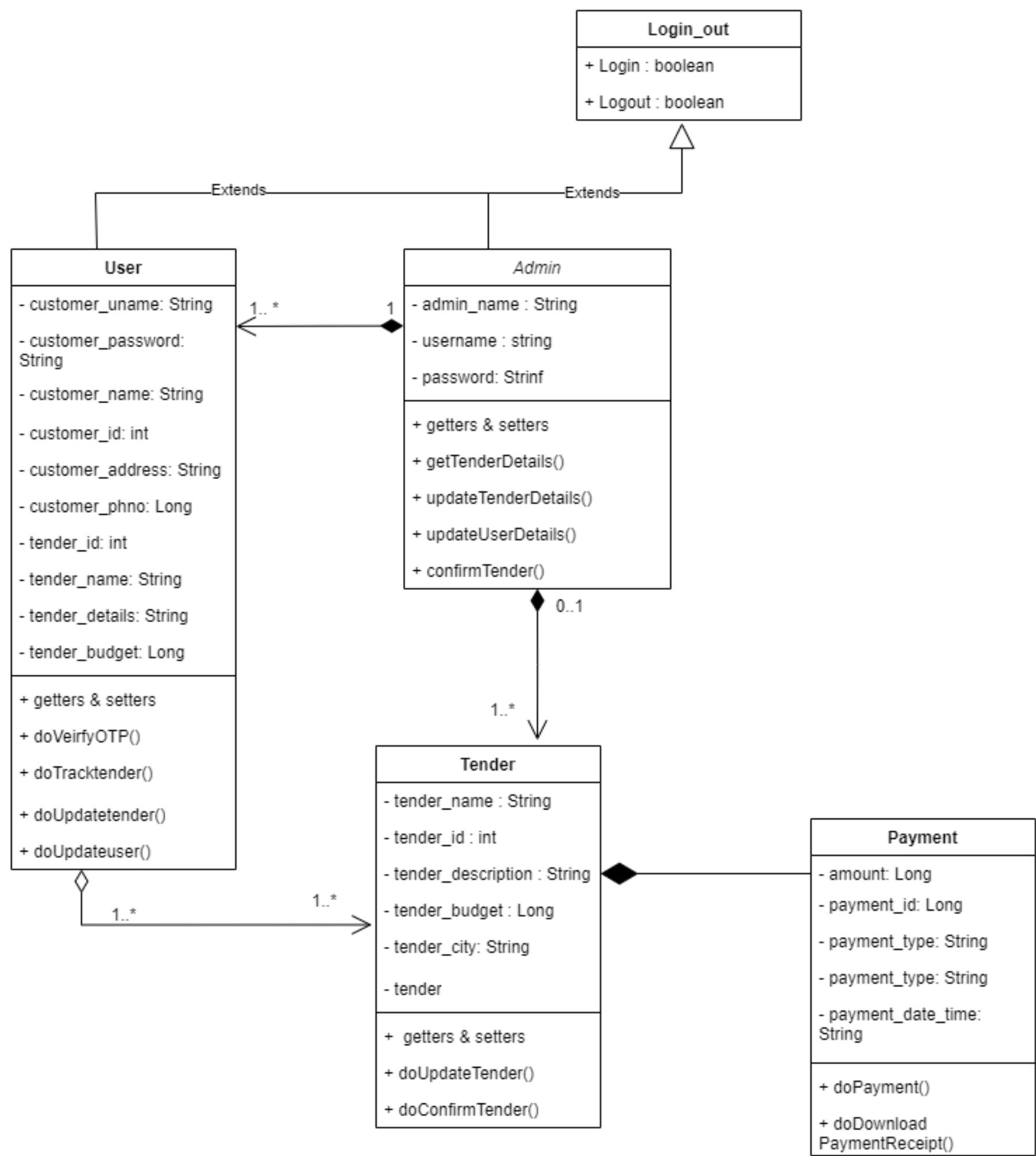
step 13: Stop

FLOWCHARTS:



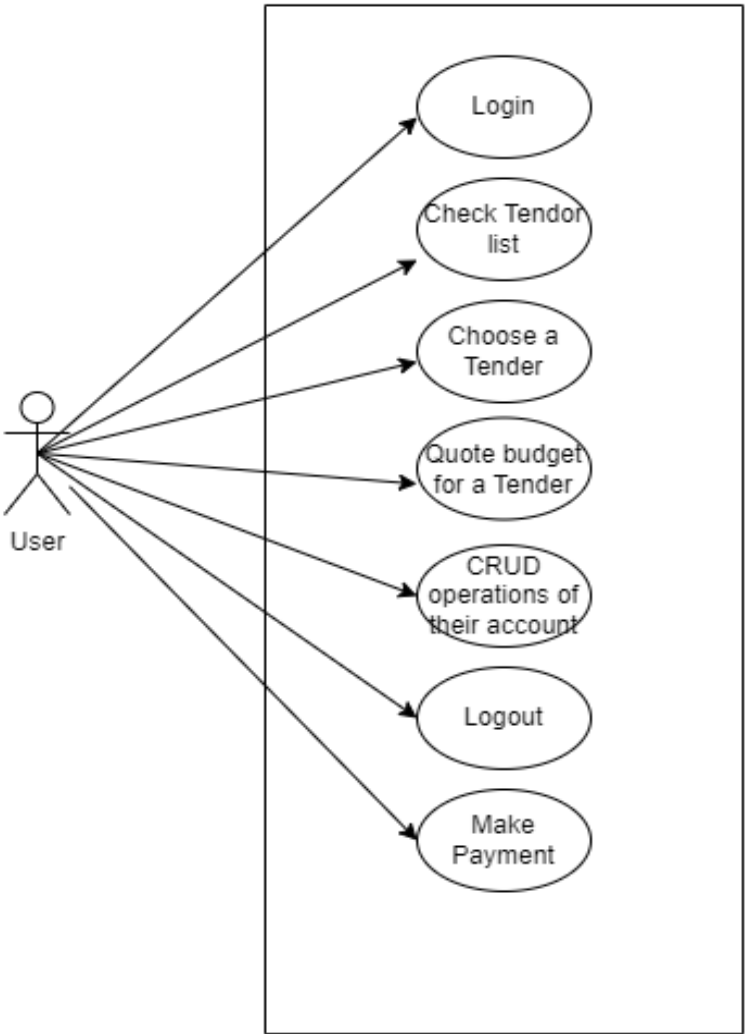
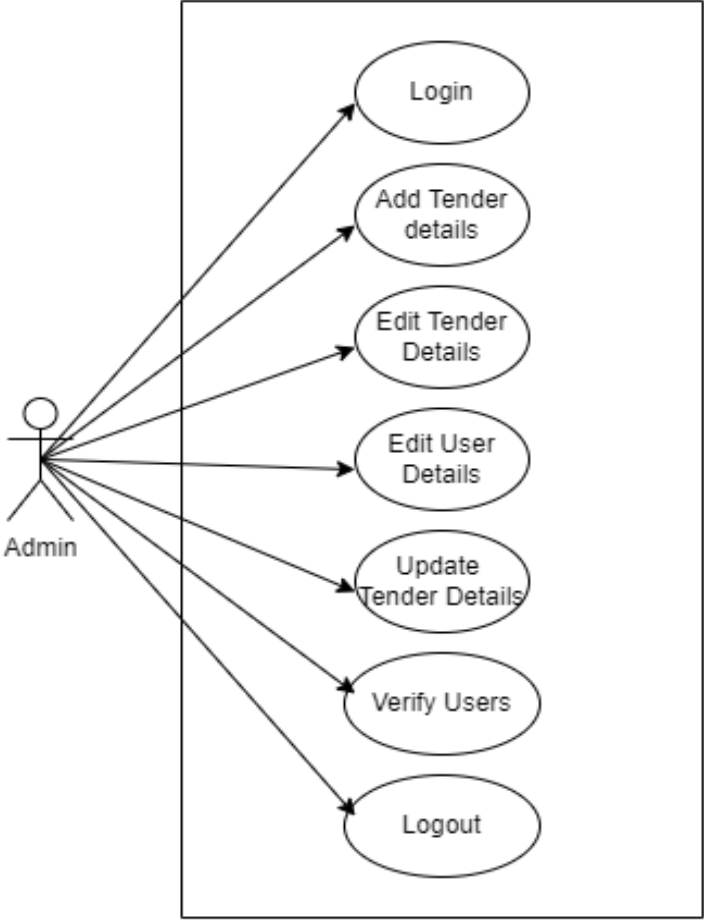
UML DIAGRAMS:
CLASS DIAGRAM:

TENDER MANAGEMENT SYSTEM CLASS DIAGRAM

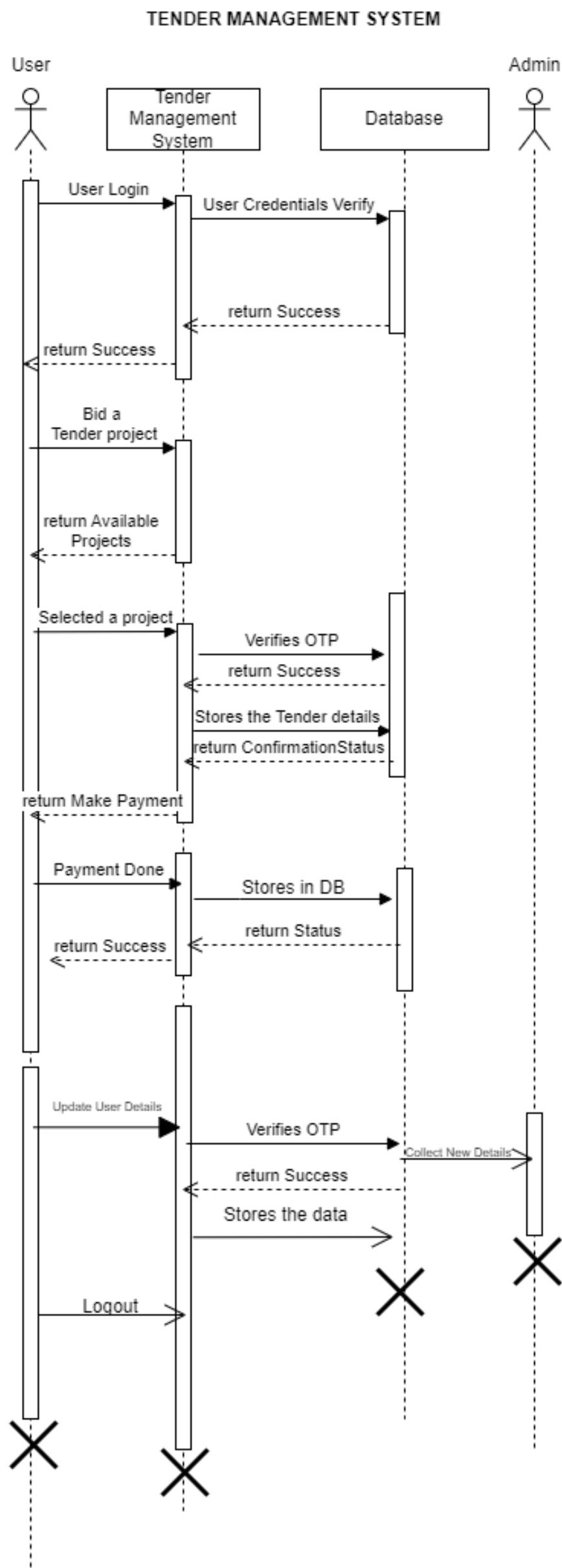


USE CASE DIAGRAM:

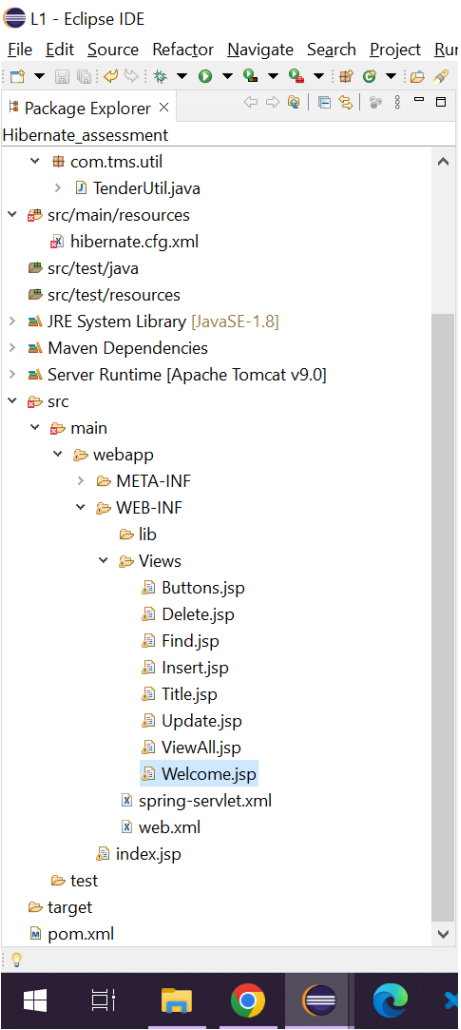
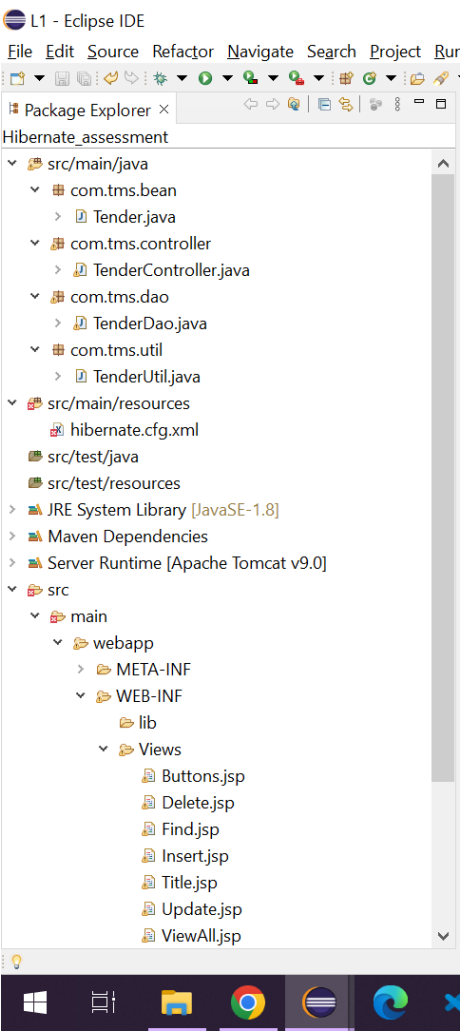
USE CASE DIAGRAM
Tender Management System



SEQUENCE DIAGRAM:



FOLDER STRUCTURE:



CODE:

com.tms.bean

Tender.java

package com.tms.bean;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

```
import javax.persistence.Table;

@Entity
@Table(name = "Tender_tbl")
public class Tender {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int tid;

    @Column(name = "Tender_Name")
    private String tname;

    @Column(name = "City")
    private String tcity;

    @Column(name = "Tender_Budget")
    private int tbudget;

    @Column(name = "Tender_Description")
    private String tdesc;

    @Column(name = "Tender_Owner")
    private String towner;

    public Tender() {
        super();
        // TODO Auto-generated constructor b
    }

    public Tender(int tid, String tname, String tcity, int tbudget, String tdesc, String towner) {
        super();
        this.tid = tid;
        this.tname = tname;
        this.tcity = tcity;
        this.tbudget = tbudget;
        this.tdesc = tdesc;
        this.towner = towner;
    }

    @Override
    public String toString() {
        return "Tender [tid=" + tid + ", tname=" + tname + ", tcity=" + tcity + ", tbudget=" + tbudget + ", tdesc="
        + tdesc + ", towner=" + towner + "]";
    }

    public int getTid() {
        return tid;
    }

    public void setTid(int tid) {
```

```
this.tid = tid;
```

```
}
```

```
public String getName() {
```

```
return tname;
```

```
}
```

```
public void setName(String tname) {
```

```
this.tname = tname;
```

```
}
```

```
public String getTcity() {
```

```
return tcity;
```

```
}
```

```
public void setTcity(String tcity) {
```

```
this.tcity = tcity;
```

```
}
```

```
public int getTbudget() {
```

```
return tbudget;
```

```
}
```

```
public void setTbudget(int tbudget) {
```

```
this.tbudget = tbudget;
```

```
}
```

```
public String getTdesc() {
```

```
return tdesc;
```

```
}
```

```
public void setTdesc(String tdesc) {
```

```
this.tdesc = tdesc;
```

```
}
```

```
public String getTowner() {
```

```
return towner;
```

```
}
```

```
public void setTowner(String towner) {
```

```
this.towner = towner;
```

```
}
```

```
}
```

com.tms.controller:

TenderController.java

package com.tms.controller;

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

import org.springframework.context.annotation.ComponentScan;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.servlet.ModelAndView;

import com.tms.bean.Tender;

import com.tms.dao.TenderDao;

import antlr.collections.List;

@Controller

//@ComponentScan(basePackages = {"com.tms.controller"})

public class TenderController {

// @Autowired

TenderDao dao = new TenderDao();

@GetMapping("/Title")

public ModelAndView loadBanner() {

ModelAndView mv = new ModelAndView("Title");

return mv;

}

@GetMapping("/Welcome")

public ModelAndView loadWelcome() {

ModelAndView mv = new ModelAndView("Welcome");

return mv;

}

@GetMapping("/Buttons")

public ModelAndView loadButtons() {

ModelAndView mv = new ModelAndView("Buttons");

return mv;

}

```
@GetMapping("/Insert")
```

```
public ModelAndView loadInsert() {
```

```
ModelAndView mv = new ModelAndView("Insert");
```

```
return mv;
```

```
}
```

```
@PostMapping("/Insertion")
```

```
public ModelAndView doInsert(@ModelAttribute("bean") Tender tend) {
```

```
int n = dao.insertTender(tend);
```

```
ModelAndView mv = new ModelAndView("Insert");
```

```
if(n>0) {
```

```
mv.addObject("Action","Success");
```

```
}
```

```
else {
```

```
mv.addObject("Action","Failure");
```

```
}
```

```
return mv;
```

```
}
```

```
@RequestMapping("Delete")
```

```
public ModelAndView loaddoDelete() {
```

```
java.util.List<Integer> idList = dao.idList();
```

```
ModelAndView mv = new ModelAndView("Delete");
```

```
mv.addObject("IdList",idList);
```

```
return mv;
```

```
}
```

```
@PostMapping("Deletion")
```

```
public ModelAndView doDelete(@ModelAttribute("bean") Tender tend) {
```

```
dao.tenderDelete(tend);
```

```
java.util.List<Integer> idList = dao.idList();
```

```
ModelAndView mv = new ModelAndView("Delete");
```

```
mv.addObject("Action","Success");
```

```
mv.addObject("IdList",idList);
```

```
return mv;
```

```
}
```

```
@RequestMapping("Update")
```

```
public ModelAndView loadUpdate() {
```

```
java.util.List<Integer> idList = dao.idList();
```

```
ModelAndView mv = new ModelAndView("Update");
```

```
mv.addObject("IdList",idList);
```

```
return mv;
```

```
}
```

```
@RequestMapping("Fetch")

public ModelAndView loaddoFetch(@RequestParam("tid") int id) {

Tender tend = dao.fetchOne(id);

ModelAndView mv = new ModelAndView("Update");

mv.addObject("tendfetch",tend);

return mv;

}


@PostMapping("Updation")

public ModelAndView doUpdate(@ModelAttribute("bean") Tender tend) {

dao.updateTender(tend);

java.util.List<Integer> idList = dao.idList();

ModelAndView mv = new ModelAndView("Update");

mv.addObject("Action","Success");

mv.addObject("IdList",idList);

return mv;

}


@RequestMapping("ViewAll")

public ModelAndView loadViewAll() {

java.util.List<Tender> TenderList = dao.fetchAll();

ModelAndView mv = new ModelAndView("ViewAll");

mv.addObject("TendList",TenderList);

return mv;

}


@RequestMapping("Find")

public ModelAndView loadFind() {

java.util.List<Integer> idList = dao.idList();

ModelAndView mv = new ModelAndView("Find");

mv.addObject("IdList",idList);

return mv;

}


@RequestMapping("FetchFind")

public ModelAndView loaddoFetchfind(@RequestParam("tid") int id) {

Tender tend = dao.fetchOne(id);

ModelAndView mv = new ModelAndView("Find");

mv.addObject("tendfind",tend);

return mv;

}


@PostMapping("Finding")

public ModelAndView dofinding(@ModelAttribute("bean") Tender tend) {
```

```
dao.findten(tend);  
java.util.List<Integer> idList = dao.idList();  
ModelAndView mv = new ModelAndView("Find");  
mv.addObject("TendfindList",idList);  
return mv;  
}  
}
```

Com.tms.dao:

TenderDao.java

```
package com.tms.dao;
```

```
import java.util.List;
```

```
import org.hibernate.Query;
```

```
import org.hibernate.Session;
```

```
import org.hibernate.Transaction;
```

```
import com.tms.bean.Tender;
```

```
import com.tms.util.TenderUtil;
```

```
public class TenderDao {
```

```
    public int insertTender(Tender tend) {
```

```
        int n=0;
```

```
        Session s = TenderUtil.openSession();
```

```
        Transaction tx =s.beginTransaction();
```

```
        n = (int) s.save(tend);
```

```
        tx.commit();
```

```
        return n;
```

```
    }
```

```
    public void tenderDelete(Tender tend) {
```

```
        Session session = TenderUtil.openSession();
```

```
        Transaction t = session.beginTransaction();
```

```
        session.delete(tend);
```

```
        t.commit();
```

```
        session.close();
```

```
    }
```

```
    public List<Integer> idList(){
```

```
        Session session = TenderUtil.openSession();
```

```
        Transaction t = session.beginTransaction();
```

```
Query query = session.createQuery("select tid from Tender");  
List<Integer> idList = query.list();  
  
t.commit();  
  
session.close();  
  
return idList;  
}
```

```
public Tender fetchOne(int id) {
```

```
    Session s = TenderUtil.openSession();  
    Transaction tx = s.beginTransaction();  
    Tender tend = s.find(Tender.class, id);  
    s.close();  
    return tend;  
}
```

```
public Tender updateTender(Tender tend) {  
    Session session = TenderUtil.openSession();  
    Transaction t = session.beginTransaction();  
    session.update(tend);  
    session.close();  
    return tend;  
}
```

```
public List<Tender> fetchAll(){  
    Session s = TenderUtil.openSession();  
    Transaction tx = s.beginTransaction();  
    List<Tender> TendList = s.createQuery("From Tender").list();  
    s.close();  
    return TendList;  
}
```

```
public void findten(Tender tend) {  
    Session session = TenderUtil.openSession();  
    Transaction t = session.beginTransaction();  
    session.save(tend);  
    t.commit();  
    session.close();  
}
```

```
}
```

Com.tms.util

TenderUtil.java


```
package com.tms.util;
```

```
import org.hibernate.Session;
```

```
import org.hibernate.SessionFactory;
```

```
import org.hibernate.cfg.Configuration;
```

```
public class TenderUtil {
```

```
public static Session openSession() {
```

```
Configuration cfg = new Configuration().configure("hibernate.cfg.xml");
```

```
SessionFactory sf = cfg.buildSessionFactory();
```

```
Session s = sf.openSession();
```

```
return s;
```

```
}
```

```
}
```

HIBERNATE CONFIGURATION FILE:

hibernate.cfg.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<session-factory>
<property name="connection.driver_class">com.mysql.jdbc.Driver</property>
<property name="connection.url">jdbc:mysql://localhost:3306/tender_db</property>
<property name="dialect">org.hibernate.dialect.MySQL8Dialect</property>
<property name="connection.username">root</property>
<property name="connection.password">root</property>
<property name="show_sql">>true</property>
<property name="hbm2ddl.auto">update</property>
<mapping class="com.tms.bean.Tender" ></mapping>
</session-factory>
</hibernate-configuration>
```

Views Folder files:

Buttons.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhY6hW+ALEwIH" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
YvpcrYf0tY3LHB60NNkmXc5s9fDVZLESaAA55NDz0xhy9GkcIdsLK1eN7N6jIeHz" crossorigin="anonymous"></script>
<meta charset="ISO-8859-1">
<style>
h1{
color:white
}
body{
background-image: linear-gradient(#200122,#6f0000);
/* background-image: linear-gradient(#0f0c29,#302b63,#24243e); */
background-repeat:no-repeat;
background-size: cover;
```

```
height:440px;
background-attachment: fixed;
}

#b1:hover,#b2:hover,#b3:hover,#b4:hover,#b5:hover{
transform:scale(1.01);
}
@keyframes ani{
from{
opacity:0;
}
to{
opacity:1;
}
}
#cont{
animation-name: ani;
animation-duration: 3s;
animation-timing-function: ease;
animation-direction: forwards;
animation-iteration-count: 1;
}
</style>
<title>TMS Dashboard</title>
</head>
<body>
<div id="cont" align="center">
<br>
<h1 align="center" class="display-4">Dashboard</h1>
<br>
<a href="Insert" id="b1" target="frame3"><input type="submit" class="btn btn-info" value="Insert"></a><br><hr>
<a href="Update" id="b2" target="frame3"><input type="submit" class="btn btn-info" value="Update"></a><br><hr>
<a href="Delete" id="b3" target="frame3"><input type="submit" class="btn btn-info" value="Delete"></a><br><hr>
<a href="Find" id="b4" target="frame3"><input type="submit" class="btn btn-info" value="Find"></a><br><hr>
<a href="ViewALL" id="b5" target="frame3"><input type="submit" class="btn btn-info" value="ViewALL"></a><br>
</div>
</body>
</html>
```

Delete.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page isELIgnored="false" %>
<!DOCTYPE html>
<html>
<head>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNLYT2bRjXh0JMhJY6hw+ALEwIH" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
YvpcrYf0tY3LHB60NNkmXc5s9fDVZLESaAA55NDz0xhy9GkcIdsLK1eN7N6jIeHz" crossorigin="anonymous"></script>

<style>
h1,p{
color:white
}
body{
background-image: linear-gradient(#200122,#6f0000);
background-repeat:no-repeat;
background-size: cover;
height:440px;
background-attachment: fixed;
}

#b1:hover,#b2:hover,#b3:hover,#b4:hover,#b5:hover{
transform:scale(1.01);
}
@keyframes ani{
from{
opacity:0;
}
to{
opacity:1;
}
}
#cont{
```

```
animation-name: ani;
animation-duration: 3s;
animation-timing-function: ease;
animation-direction: forwards;
animation-iteration-count: 1;
}
</style>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<center>
<form action="Deletion" method="post">
<h1 class="display-1">Delete Tender details by ID</h1>
<p class="display-3">Choose ID:</p>
<select name="tid">
<option selected>Select ID </option>
<c:forEach items="{IdList}" var="tid" >
<option value="{tid}">{tid}</option>
</c:forEach>
</select>
<br>
<br><input class="btn btn-danger" type="Submit" value="Delete" name="button">
</form>
<%
String result = (String) request.getAttribute("Action");
if(result!=null){

if(result.equals("Success")){
out.print("<h3><font color=green>Deleted Successfully!</font></h3>");
}
else{
out.print("<h3><font color=red>Deletion Failed!!!</font></h3>");
}
}
}%>
</center>
</body>
</html>
```

Find.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page isELIgnored="false" %>
<!DOCTYPE html>
<html>
<head>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNLYT2bRjXh0JMhJY6hW+ALEwIH" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
YvpcrYf0tY3LHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdsLK1eN7N6jIeHz" crossorigin="anonymous"></script>

<meta charset="ISO-8859-1">
<title>Insert title here</title>

<style>
h1,td,p{
color:white
}
body{
background-image: linear-gradient(#200122,#6f0000);
background-repeat:no-repeat;
background-size: cover;
height:440px;
background-attachment: fixed;
}

#b1:hover,#b2:hover,#b3:hover,#b4:hover,#b5:hover{
transform:scale(1.01);
}
@keyframes ani{
from{
opacity:0;
}
}
```

```
to{
opacity:1;
}
}
#cont{
animation-name: ani;
animation-duration: 3s;
animation-timing-function: ease;
animation-direction: forwards;
animation-iteration-count: 1;
}
</style>
</head>
<body>
<form action="FetchFind" method="post" align="center" modelAttribute="bean">
<h1 class="text-center display-1">Find Tender Details by ID</h1>
<p class="display-3 text-center">Choose ID:</p>
<select name="tid">
<option selected>Select ID </option>
<c:forEach items="{IdList}" var="tid" >
<option value="{tid}">{tid}</option>
</c:forEach>
</select>
<br>
<br><input class="btn btn-info" type="Submit" value="Find" name="button">
</form>

<form align="center" class="container" action="Finding" method="post" modelAttribute="bean">
<h1>Tender Details</h1>
<table border="1" class="table table-hover">
<tr>
<th scope="col">Tender ID</th>
<th scope="col">Tender Name</th>
<th scope="col">Tender City</th>
<th scope="col">Tender Budget</th>
<th scope="col">Tender Description</th>
<th scope="col">Tender Owner</th>
</tr>

<tr>
<td>{tendfind.getId() }</td>
<td>{tendfind.getName() }</td>
<td>{tendfind.getTcity() }</td>
<td>{tendfind.getTbudget() }</td>
<td>{tendfind.getTdesc() }</td>
<td>{tendfind.getTowner() }</td>
</tr>
</table>
</form>
</body>
</html>
```

Insert.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
YvpcrYf0tY3LHB60NNkMxc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeH2" crossorigin="anonymous"></script>
<meta charset="ISO-8859-1">
<title>TMS</title>
<style>
h1,p,td{
color:white
}
body{
background-image: linear-gradient(#200122,#6f0000);
background-repeat:no-repeat;
background-size: cover;
height:440px;
background-attachment: fixed;
}
```

```
#b1: hover, #b2: hover, #b3: hover, #b4: hover, #b5: hover{
transform: scale(1.01);
}
@keyframes ani{
from{
opacity: 0;
}
to{
opacity: 1;
}
}
#cont{
animation-name: ani;
animation-duration: 3s;
animation-timing-function: ease;
animation-direction: forwards;
animation-iteration-count: 1;
}
</style>
</head>
<body>
<form action="Insertion" method="post" modelAttribute="bean">
<h1 align="center" class="display-1">Tender Registration Form</h1>
<hr>
<table align="center">

<tr>
<td>
Tender ID:
</td>
<td>
<input type="text" disabled name="tid" placeholder="Tender ID" required>
</td>
</tr>
<tr>
<td>
Tender Project Name:
</td>
<td>
<input type="text" name="tname" placeholder="Tender Project Name" required>
</td>
</tr>
<tr>
<td>
Tender City:
</td>
<td>
<input type="text" name="tcity" placeholder="Project City" required>
</td>
</tr>
<tr>
<td>
Tender Budget:
</td>
<td>
<input type="text" name="tbudget" placeholder="Tender Budget" required>
</td>
</tr>
<tr>
<td>
Tender Description:
</td>
<td>
<textarea rows="2" cols="23" input type="text" name="tdesc" placeholder="Project Description" required></textarea>
<!-- <input type="textarea" name="tdesc" placeholder="Project Description" required> -->
</td>
</tr>
<tr>
<td>
Tender Acquired by:
</td>
<td>
<input type="text" name="towner" placeholder="Tender Acquired by" required>
</td>
</tr>
```

```
<br>
</table>
<br>
<div class="text-center">
<input type="submit" class="btn btn-success" name="button" value="Insert">
</div>

</form>
<%
String result = (String) request.getAttribute("Action");

if(result!=null){

if(result.equals("Success")){
    out.print("<h3 align=center><font color=green>Inserted Successfully!</font></h3>");
}
else{
    out.print("<h3 align=center><font color=red>Insertion Operation Failed!!!</font></h3></div>");
}

}

%>
</body>
</html>
```

Title.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNLYT2bRjXh0JMhJY6hW+ALEwIH" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
YvpcrYf0tY3LHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>
<style>
body{
background-image: linear-gradient(#200122,#6f0000);
/* background-image: linear-gradient(#00B4DB,#0083B0); */
background-repeat:no-repeat;
background-size: cover;
background-position: fixed;
}
a{
text-decoration:none;
color:white;
}
#title{
color:white;
}
@keyframes ani{
from{
opacity:0;
}
to{
opacity:1;
}
}
#title{
animation-name: ani;
animation-duration: 3s;
animation-timing-function: ease;
animation-direction: forwards;
animation-iteration-count: 1;
}
</style>
<title>TMS</title>
</head>
<body>
<br>
```

```
<hr>

<div id="title">
<a href="Welcome" target="frame3"><h1 class="display-5 fs-text-6 text-center">Tender Management System</h1></a>
</div>
<hr>

</body>
</html>
```

Update.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page isELIgnored="false" %>
<!DOCTYPE html>
<html>
<head>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNLYT2bRjXh0JMhJY6hW+ALEwIH" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
YvpcrYf0tY3LHB60NNkmXc5s9fDVZLESaAA55NDz0xhy9GkcIdsLK1eN7N6jIeHz" crossorigin="anonymous"></script>

<style>
h1,p,td,h3{
color:white
}
body{
background-image: linear-gradient(#200122,#6f0000);
background-repeat:no-repeat;
background-size: cover;
height:440px;
background-attachment: fixed;
}

#b1:hover,#b2:hover,#b3:hover,#b4:hover,#b5:hover{
transform:scale(1.01);
}
@keyframes ani{
from{
opacity:0;
}
to{
opacity:1;
}
}
#cont{
animation-name: ani;
animation-duration: 3s;
animation-timing-function: ease;
animation-direction: forwards;
animation-iteration-count: 1;
}
</style>
<meta charset="ISO-8859-1">
<title>TMS</title>
</head>
<body>
<center>
<form action="Fetch" method="post" modelAttribute="bean">
<h1 class="display-1">Update Tender Details by ID</h1>
<h3 class="display-2">Choose ID:</h3>
<select name="tid">
<option selected>Select ID </option>
<c:forEach items="{IdList}" var="tid" >
<option value="{tid}">{tid}</option>
</c:forEach>
</select>

<br><br><input class="btn btn-warning" type="Submit" value="Fetch" name="button">
</form>

<br>
<form action="Updation" method="post" modelAttribute="bean">
```

```
<table>
<tr>
<td>
Tender ID:
</td>
<td>
<input type="text" disabled name="tid" plcaeholder="Tender ID" value="${tendfetch.getTid() }" required>
</td>
</tr>
<tr>
<td>
Tender Project Name:
</td>
<td>
<input type="text" name="tname" plcaeholder="Tender Project Name" value="${tendfetch.getTname() }" required>
</td>
</tr>
<tr>
<td>
Tender City:
</td>
<td>
<input type="text" name="tcity" placeholder="Project City" value="${tendfetch.getTcity() }" required>
</td>
</tr>
<tr>
<td>
Tender Budget:
</td>
<td>
<input type="text" name="tbudget" placeholder="Tender Budget" value="${tendfetch.getTbudget() }" required>
</td>
</tr>
<tr>
<td>
Tender Description:
</td>
<td>
<textarea rows="2" cols="23" input type="text" name="tdesc" value="${tendfetch.getTdesc() }" placeholder="Project
Description" required></textarea>
</td>
</tr>
<tr>
<td>
Tender Owner:
</td>
<td>
<input type="text" name="towner" placeholder="Tender Owner Name" value="${tendfetch.getTowner() }" required>
</td>
</tr>
</table>
<br>
<div class="text-center">
<input type="submit" class="btn btn-warning" name="button" value="Update">
</div>
</form>
```

```
<%
String result = (String) request.getAttribute("Action");
if(result!=null){

if(result.equals("Success")){
out.print("<h3 align=center class=display-1><font color=green>Successfully Updated!</font></h3>");
}
else{
out.print("<h3 align=center class=display-1><font color=red>Updation Failure!!!</font></h3>");
}
}
%>
</center>
</body>
</html>
```


ViewAll.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page isELIgnored="false" %>
<!DOCTYPE html>
<html>
<head>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNLYT2bRjXh0JMhJY6hw+ALEwIH" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
YvpcrYf0tY3LHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdsLK1eN7N6jIeHz" crossorigin="anonymous"></script>

<style>
h1,p,th,td{
color:white
}
body{
background-image: linear-gradient(#200122,#6f0000);
background-repeat:no-repeat;
background-size: cover;
height:440px;
background-attachment: fixed;
}
</style>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<br>
<form align="center" class="container" action="ViewALL" method="post">
<h1>Tender Details</h1>
<table border="1" class="table table-hover">
<tr>
<th scope="col">Tender ID</th>
<th scope="col">Tender Name</th>
<th scope="col">Tender City</th>
<th scope="col">Tender Budget</th>
<th scope="col">Tender Description</th>
<th scope="col">Tender Owner</th>
</tr>

<c:forEach items="${TendList}" var="tend">
<tr>
<td>${tend.getId() }</td>
<td>${tend.getName() }</td>
<td>${tend.getTcity() }</td>
<td>${tend.getTbudget() }</td>
<td>${tend.getTdesc() }</td>
<td>${tend.getTowner() }</td>
</tr>
</c:forEach>
</table>
</form>
</body>
</html>
```

Welcome.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>TMS</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNLYT2bRjXh0JMhJY6hw+ALEwIH" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
YvpcrYf0tY3LHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdsLK1eN7N6jIeHz" crossorigin="anonymous"></script>
<style>
body{
background-image: linear-gradient(#200122,#6f0000);
/* background-image: linear-gradient(#0f0c29,#302b63,#24243e); */
```

```

background-repeat: no-repeat;
background-size: cover;
height: 240px;
background-position: fixed;
}
h1{
margin-top: 200px;
}

@keyframes ani{
from{
opacity: 0;
}
to{
opacity: 1;
}
}
a{
text-decoration: none;
color: white;
animation-name: ani;
animation-duration: 3s;
animation-timing-function: ease;
animation-direction: forwards;
animation-iteration-count: 1;
}
p{
text-decoration: none;
color: white;
animation-name: ani;
animation-duration: 2s;
animation-timing-function: ease;
animation-direction: forwards;
animation-iteration-count: infinite;
}
a: hover{
transform: scale(1.01);
}
</style>
</head>
<body>
<div>
<a href="Insert"><h1 class="display-2 text-center">Tender Management System</h1></a>
</div>
<p class="fs-5 text-center"> ≤- Select any one of the options to continue</p>
</body>
</html>

```

Spring-servlet.xml:

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:mvc="http://www.springframework.org/schema/mvc"
xsi:schemaLocation="
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd">

<!--Provide support for conversion, formatting and validation -->
<context:component-scan base-package="com.tms.controller" />
<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
<property name="prefix" value="/WEB-INF/Views/" />
<property name="suffix" value=".jsp" />
</bean>
<!-- <bean id="dao" class="com.tms.dao.TenderDao"></bean> -->
</beans>

```

web.xml:

```

<!DOCTYPE web-app PUBLIC
"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"

```

```
"http://java.sun.com/dtd/web-app_2_3.dtd" >
<web-app>
<display-name>Archetype Created Web Application</display-name>
<servlet>
<servlet-name>spring</servlet-name>
<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
<load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
<servlet-name>spring</servlet-name>
<url-pattern>/</url-pattern>
</servlet-mapping>
</web-app>
```

index.jsp:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<head>
<title>CRUD using Hibernate and Spring</title>
<frameset rows="30,90">
<frame src="Title" name="frame1"></frame>
<frameset cols="30,50">
<frame src="Buttons" name="frame2"></frame>
<frame src="Welcome" name="frame3"></frame>
</frameset>
</frameset>
</head>
```

pom.xml:

```
<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 http://maven.apache.org/maven-v4_0_0.xsd">
<modelVersion>4.0.0</modelVersion>
<groupId>com.spring</groupId>
<artifactId>Hibernate_assessment</artifactId>
<packaging>war</packaging>
<version>0.0.1-SNAPSHOT</version>
<name>Hibernate_assessment Maven Webapp</name>
<url>http://maven.apache.org</url>
<dependencies>
<dependency>
<groupId>junit</groupId>
<artifactId>junit</artifactId>
<version>3.8.1</version>
<scope>test</scope>
</dependency>
<!--https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-webmvc</artifactId>
<version>5.3.23</version>
</dependency>

<!-- https://mvnrepository.com/artifact/javax.servlet/jstl -->
<dependency>
<groupId>javax.servlet</groupId>
<artifactId>jstl</artifactId>
<version>1.2</version>
</dependency>
<!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->

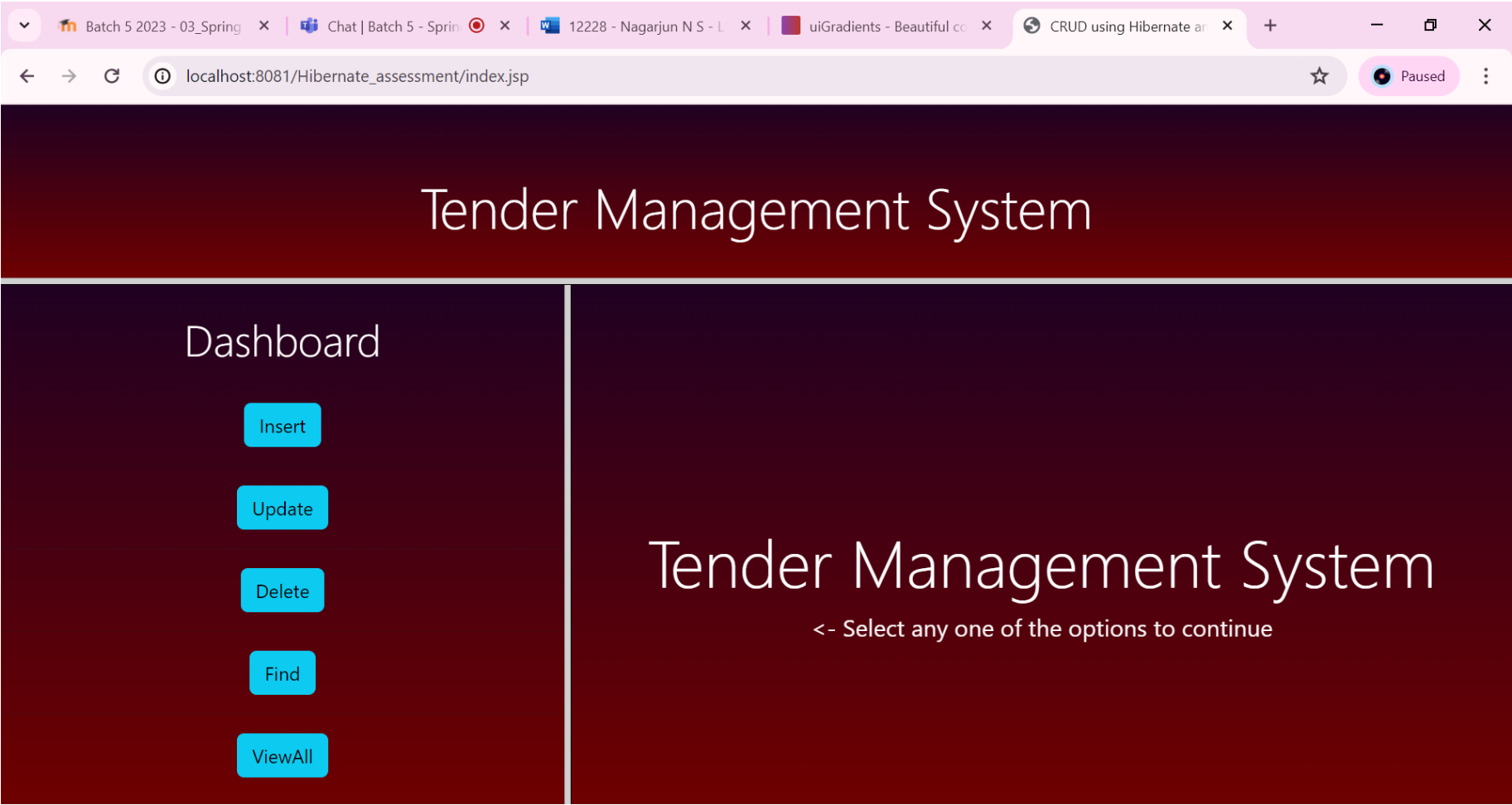
<!--<dependency>
<groupId>javax.servlet</groupId>
<artifactId>javax.servlet-api</artifactId>
<version>3.1.0</version>
<scope>provided</scope>
</dependency>-->

<!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->
<dependency>
<groupId>javax.servlet</groupId>
<artifactId>servlet-api</artifactId>
```

```
<version>3.0-alpha-1</version>
</dependency>
<dependency>
<groupId>org.hibernate</groupId>
<artifactId>hibernate-core</artifactId>
<version>5.5.4.Final</version>
</dependency>
<!-- https://mvnrepository.com/artifact/com.mysql/mysql-connector-j -->
<dependency>
<groupId>com.mysql</groupId>
<artifactId>mysql-connector-j</artifactId>
<version>8.0.33</version>
</dependency>
</dependencies>
<build>
<finalName>Hibernate_assessment</finalName>
</build>
</project>
```

SCREENSHOTS:

HOMEPAGE:



INSERTION:

Batch 5 2023 - 03_Spring

Chat | Batch 5 - Sprin

12228 - Nagarjun N S - L

uiGradients - Beautiful co

CRUD using Hibernate ar

localhost:8081/Hibernate_assessment/index.jsp

Paused

Tender Management System

Dashboard

Insert

Update

Delete

Find

ViewAll

Tender Registration Form

Tender ID:

Tender Project Name: Kalligudi Panchayat Road

Tender City: Virudhunagar

Tender Budget: 8967586

Tender Description: months for all over the panchayat

Tender Acquired by: VIP Constructions

Insert

Batch 5 2023 - 03_Spring

Chat | Batch 5 - Sprin

12228 - Nagarjun N S - L

uiGradients - Beautiful co

CRUD using Hibernate ar

localhost:8081/Hibernate_assessment/index.jsp

Paused

Tender Management System

Dashboard

Insert

Update

Delete

Find

ViewAll

Tender Registration Form

Tender ID:

Tender Project Name:

Tender City: Project City

Tender Budget: Tender Budget

Tender Description: Project Description

Tender Acquired by: Tender Acquired by

Insert

Inserted Successfully Done!

UPDATION:

Batch 5 2023 - 03_Spring & Hib

Chat | Batch 5 - Spring & Hi

12228 - Nagarjun N S - L1 - Hib

CRUD using Hibernate and Spri

localhost:8081/Hibernate_assessment/index.jsp

Paused

Tender Management System

Dashboard

Insert

Update

Delete

Find

ViewAll

Update Tender Details by ID

Choose ID:

Select ID

Fetch

Tender ID:

Tender Project Name:

Tender City:

Tender Budget:

Tender Description:

Project City

Tender Budget

Project Description

Batch 5 2023 - 03_Spring & Hib

Chat | Batch 5 - Spring & Hi

12228 - Nagarjun N S - L1 - Hib

CRUD using Hibernate and Spri

localhost:8081/Hibernate_assessment/index.jsp

Paused

Tender Management System

Dashboard

Insert

Update

Delete

Find

ViewAll

Update Tender Details by ID

Choose ID:

Select ID

Select ID

1

Fetch

Tender ID:

Tender Project Name:

Tender City:

Tender Budget:

Tender Description:

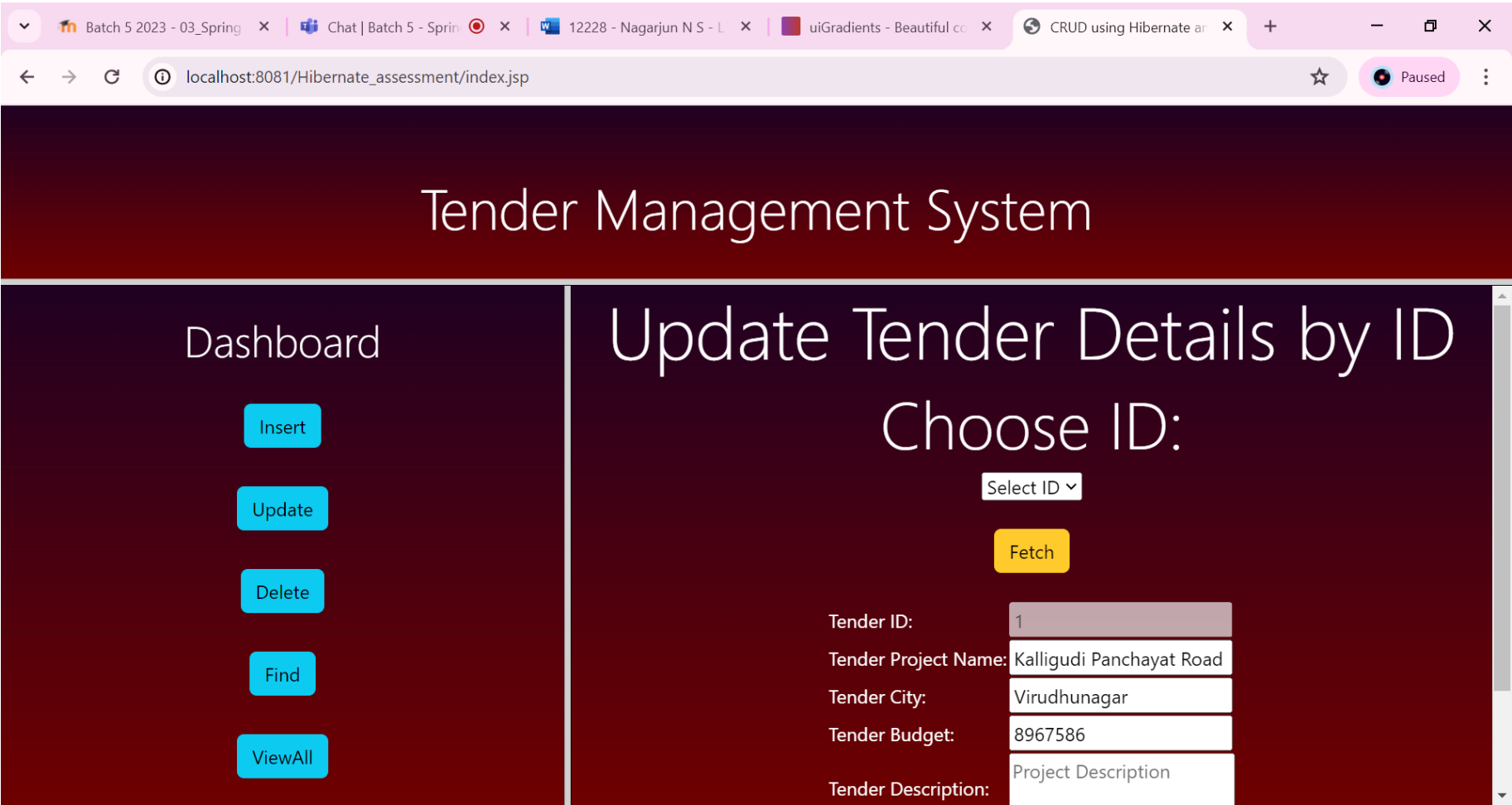
Project City

Tender Budget

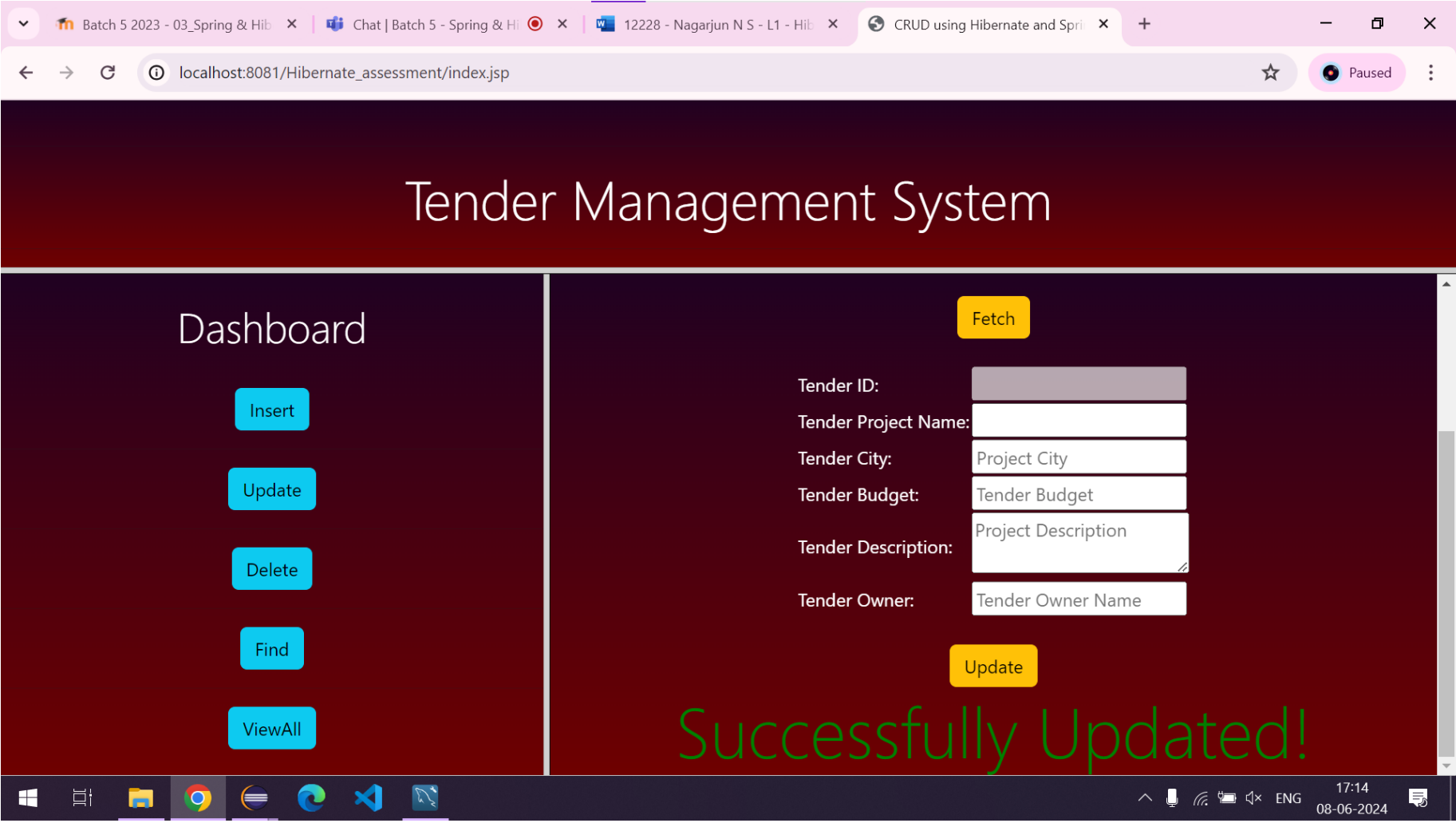
Project Description

Windows Taskbar

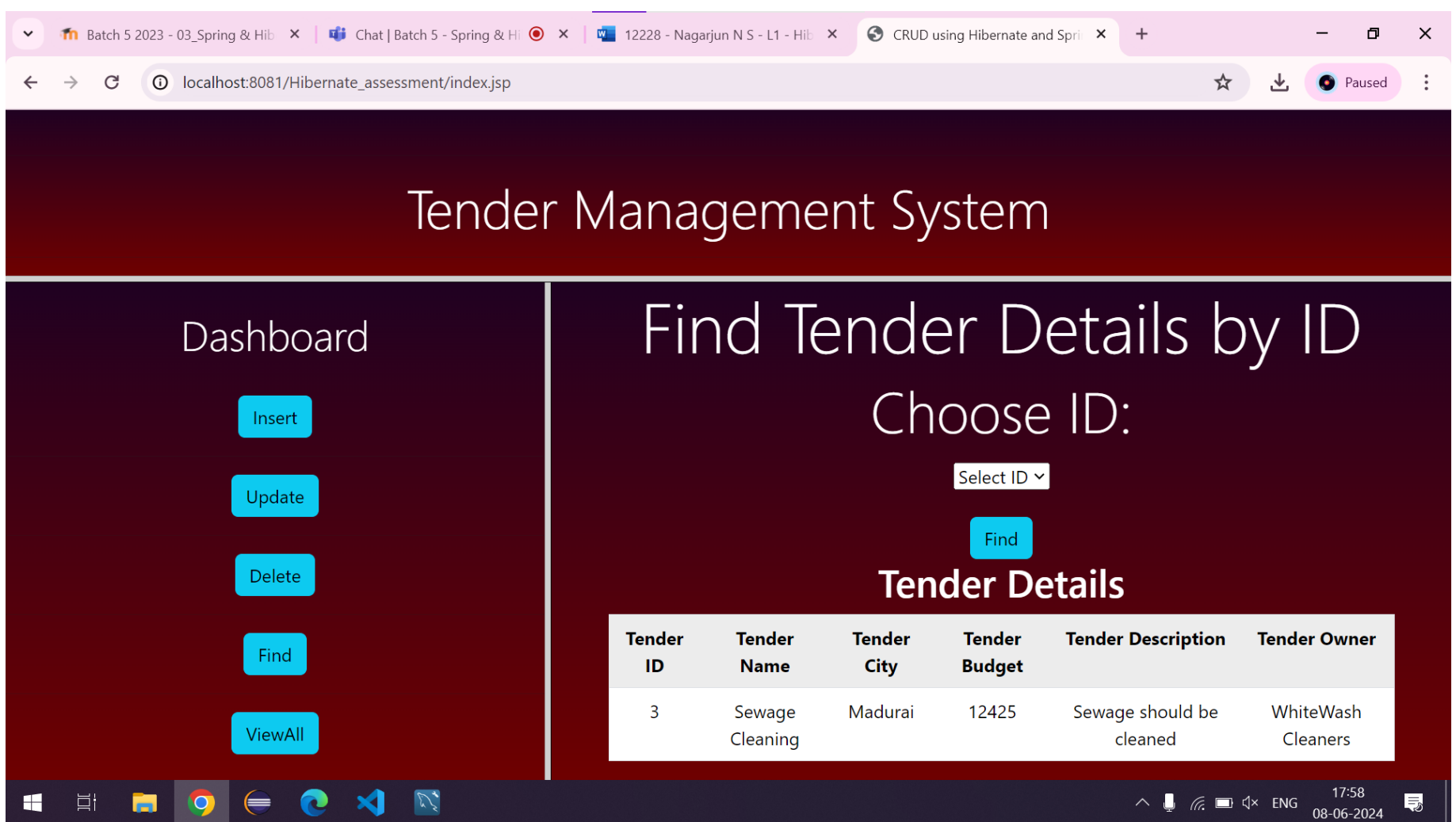
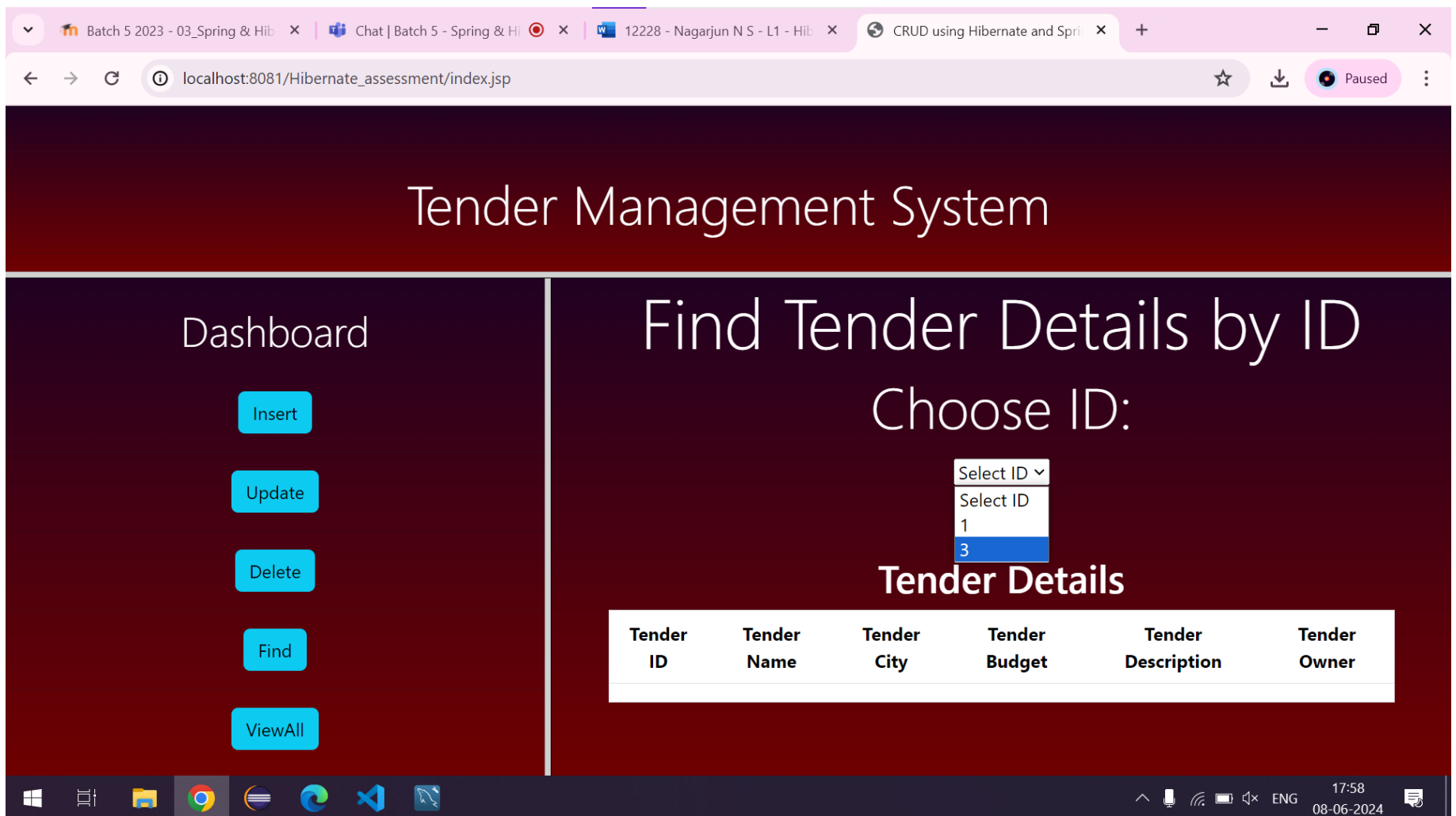
17:12 08-06-2024



City is Updated from “Virudhunagar” to “Kalligudi”:



FIND:



VIEW ALL:

Batch 5 2023 - 03_Spring & Hib

Chat | Batch 5 - Spring & Hi

12228 - Nagarjun N S - L1 - Hib

CRUD using Hibernate and Spri

localhost:8081/Hibernate_assessment/index.jsp

Paused

Tender Management System

Dashboard

Insert

Update

Delete

Find

ViewAll

Tender Details

| Tender ID | Tender Name | Tender City | Tender Budget | Tender Description | Tender Owner |
|-----------|---------------------------------|-------------|---------------|--|--------------------|
| 1 | Kalligudi Panchayat Road Schema | Kalligudi | 8967586 | A Tar Road to be held firmly and sewage systems should have a separate way in which the road should not be affected. The tender timeline will be 6 months for all over the panchayat | VIP Constructions |
| 2 | Sewage Cleaning | Madurai | 8785475 | Sewage System should be filled all over the city within the given budget and promptly | WhiteWash Cleaners |

Windows Taskbar

17:21 08-06-2024

DELETION:

BEFORE DELETION:

Batch 5 2023 - 03_Spring & HibBatch 5 - Spring & Hib12228 - Nagarjun N S - L1 - HibCRUD using Hibernate and Spr

localhost:8081/Hibernate_assessment/index.jsp

Paused

Tender Management System

Dashboard

InsertUpdateDeleteFindViewAll

Delete Tender Details by ID

Choose ID:

Select ID ▼Delete

18:0108-06-2024

Batch 5 2023 - 03_Spring & HibBatch 5 - Spring & Hib12228 - Nagarjun N S - L1 - HibCRUD using Hibernate and Spr

localhost:8081/Hibernate_assessment/index.jsp

Paused

Tender Management System

Dashboard

InsertUpdateDeleteFindViewAll

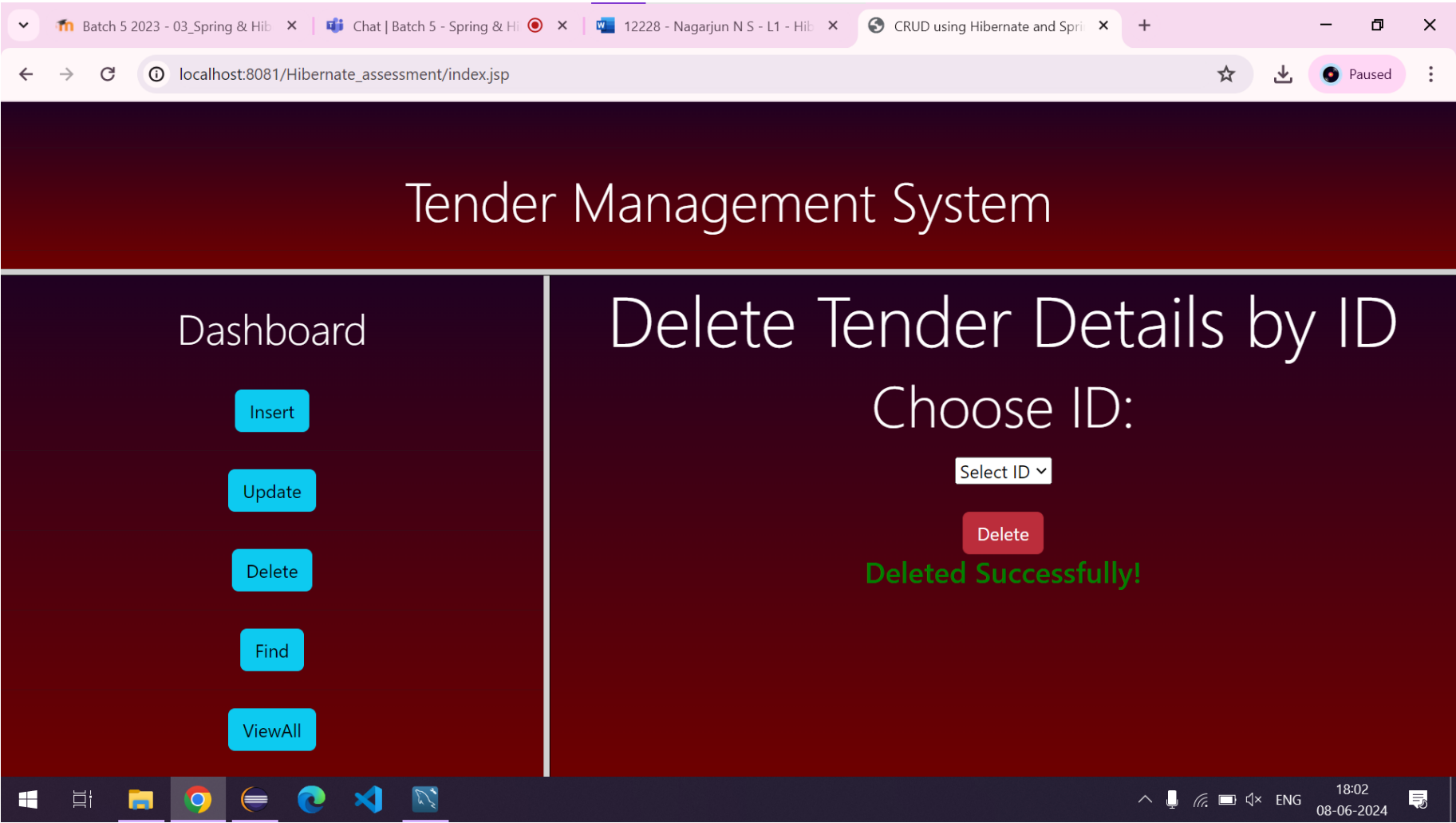
Delete Tender Details by ID

Choose ID:

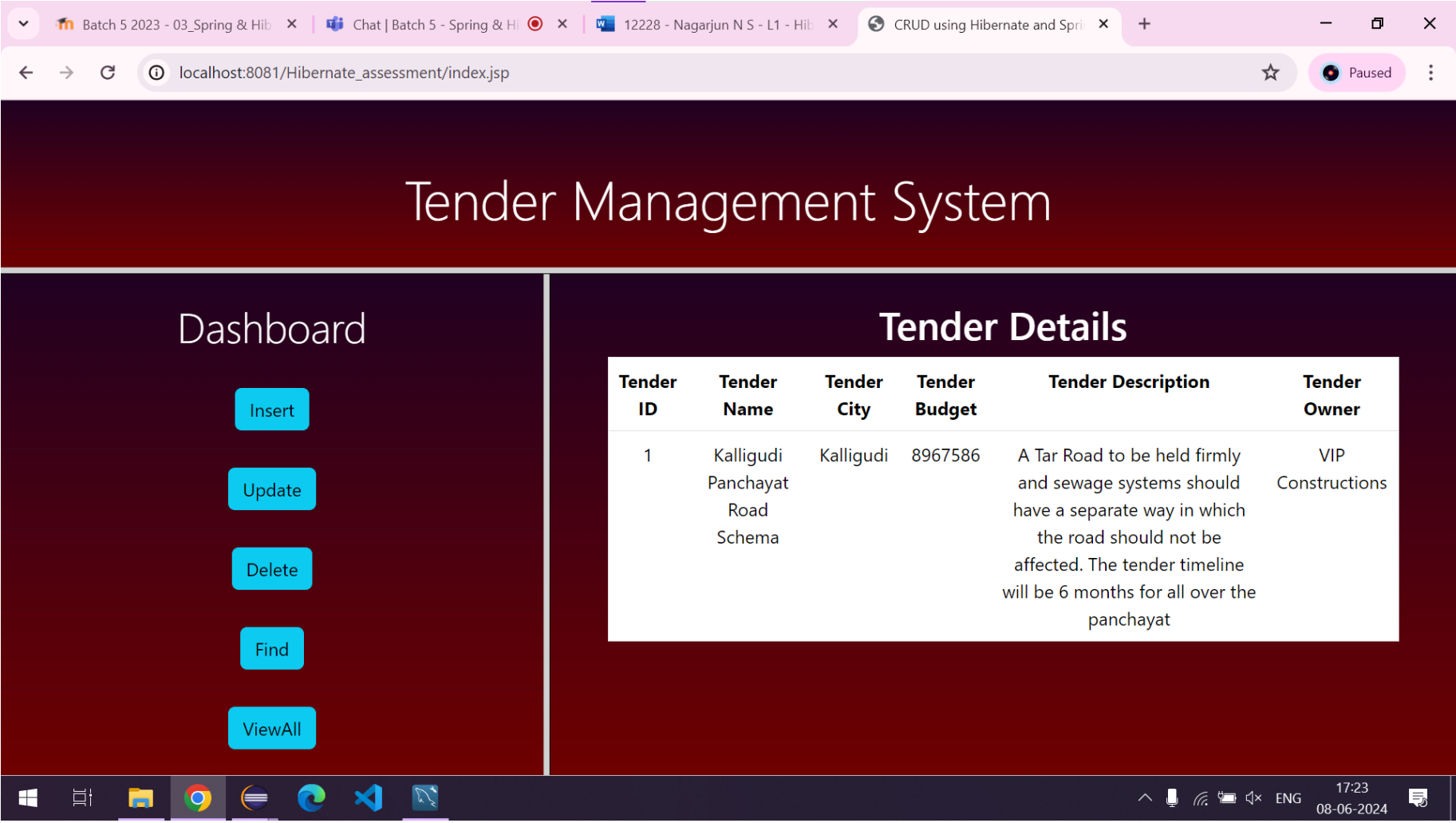
Select ID ▼

Select ID13

18:0108-06-2024



AFTER DELETION VIEW ALL:



DB TABLE STRUCTURE:

MySQL Workbench

Local instance MySQL80

FileEditViewQueryDatabaseServerToolsScriptingHelp

Limit to 1000 rows

Navigator

SCHEMAS

Filter objects

Functions

subqueries

sys

tender_db

Tables

tender_tbl

Columns

Indexes

Foreign Keys

Triggers

Views

Stored Procedures

Functions

AdministrationSchemas

Information

Table: tender_tbl

Columns:

tidint AI P

Tender_Budgetint

Cityvarchar(255)

Tender_Descriptionvarchar(255)

Tender_Namevarchar(255)

Tender_Ownervarchar(255)

models_manyone

2 • use tender_db;

3 • show tables;

4 • truncate tender_tbl;

5 • select * from tender_tbl;

6 • truncate stu_tbl;

7 • desc tender_tbl;

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

| Field | Type | Null | Key | Default | Extra |
|--------------------|--------------|------|-----|-----------------|----------------|
| tid | int | NO | PRI | <div>NULL</div> | auto_increment |
| Tender_Budget | int | YES | | <div>NULL</div> | |
| City | varchar(255) | YES | | <div>NULL</div> | |
| Tender_Description | varchar(255) | YES | | <div>NULL</div> | |
| Tender_Name | varchar(255) | YES | | <div>NULL</div> | |
| Tender_Owner | varchar(255) | YES | | <div>NULL</div> | |

Result 24

Read Only

Output

Action Output

| # | Time | Action | Message | Duration / Fetch |
|----|----------|--|-------------------|-----------------------|
| 35 | 17:02:10 | select * from tender_tbl LIMIT 0, 1000 | 0 row(s) returned | 0.000 sec / 0.000 sec |
| 36 | 17:06:54 | select * from tender_tbl LIMIT 0, 1000 | 1 row(s) returned | 0.000 sec / 0.000 sec |
| 37 | 17:07:05 | desc tender_tbl | 6 row(s) returned | 0.016 sec / 0.000 sec |

Object Info

Session

Windows Taskbar

System Tray