Record & Data Management System(JSP)

717823P352

717823P247

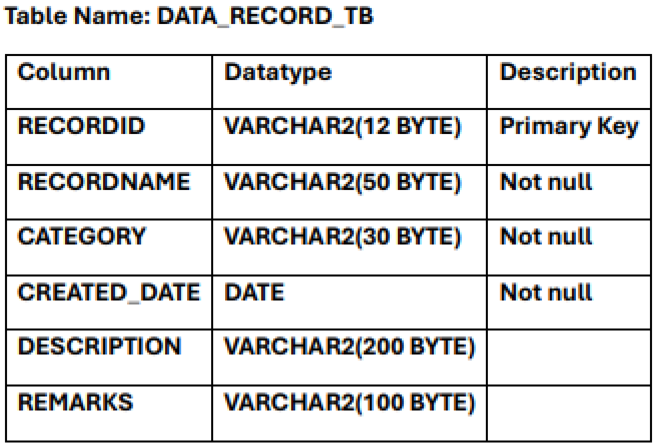
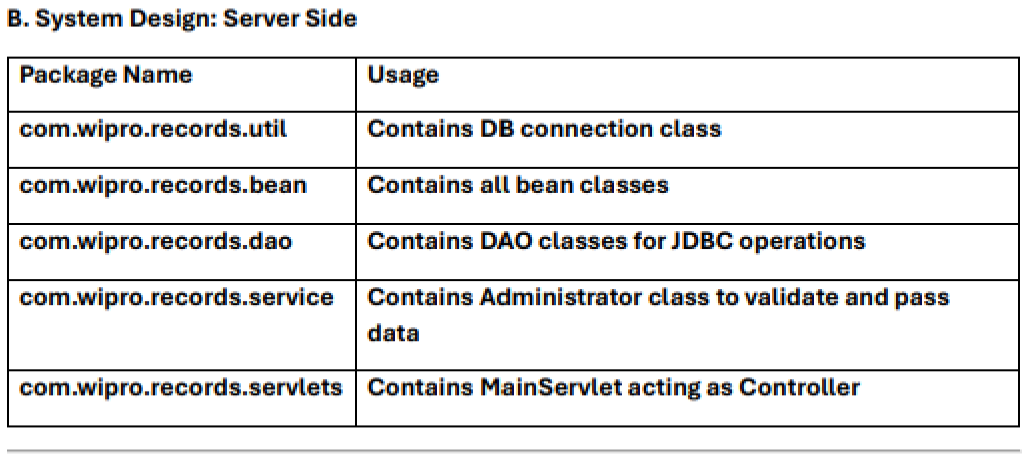


Table Name: DATA_RECORD_TB

Column	Datatype	Description
RECORDID	VARCHAR2(12 BYTE)	Primary Key
RECORDNAME	VARCHAR2(50 BYTE)	Not null
CATEGORY	VARCHAR2(30 BYTE)	Not null
CREATED_DATE	DATE	Not null
DESCRIPTION	VARCHAR2(200 BYTE)	
REMARKS	VARCHAR2(100 BYTE)	

```
CREATE DATABASE records_db;
USE records_db;
CREATE TABLE DATA_RECORD_TB (
  RECORDID VARCHAR(12) PRIMARY KEY,
  RECORDNAME VARCHAR(50) NOT NULL,
  CATEGORY VARCHAR(30) NOT NULL,
  CREATED_DATE DATE NOT NULL,
  DESCRIPTION VARCHAR(200),
  REMARKS VARCHAR(100)
);
CREATE TABLE DATA_RECORD_SEQ (
  SEQ_ID INT AUTO_INCREMENT PRIMARY KEY
);
INSERT INTO DATA_RECORD_SEQ VALUES (9);
INSERT INTO DATA_RECORD_TB VALUES
('20250410RE10', 'Report', 'Finance', '2025-04-10', 'Quarterly financial report',
'Approved'),
('20250411IN11', 'Invoice', 'Accounts', '2025-04-11', 'Vendor invoice', 'Pending'),
('20250412ME12', 'Memo', 'HR', '2025-04-12', 'Internal memo', 'For review');
SELECT * FROM DATA_RECORD_TB;
```



B. System Design: Server Side

Package Name	Usage
com.wipro.records.util	Contains DB connection class
com.wipro.records.bean	Contains all bean classes
com.wipro.records.dao	Contains DAO classes for JDBC operations
com.wipro.records.service	Contains Administrator class to validate and pass data
com.wipro.records.servlets	Contains MainServlet acting as Controller

com.wipro.records.util

DBUtil.java

```
package com.wipro.records.util;
public class DBUtil {
   public static Connection getDBConnection() {
      Connection con = null;
      try {
         Class.forName("oracle.jdbc.driver.OracleDriver");
         con = DriverManager.getConnection(
            "jdbc:oracle:thin:@localhost:1521:XE",
            "username",
            "password" );
      } catch (Exception e) {
         e.printStackTrace(); }
      return con;
   }}
```

com.wipro.records.bean

DataRecordBean.java

```
package com.wipro.records.bean;

public class DataRecordBean {
   private String recordId;
   private String recordName;
   private String category;
```

```
    private Date createdDate;
    private String description;
    private String remarks;
            public String getRecordId() {
                    return recordId;}
            public void setRecordId(String recordId) {
                    this.recordId = recordId;}
            public String getRecordName() {
                    return recordName;}
            public void setRecordName(String recordName) {
                    this.recordName = recordName;}
            public String getCategory() {
                    return category;}
            public void setCategory(String category) {
                    this.category = category;}
            public Date getCreatedDate() {
                    return createdDate;}
            public void setCreatedDate(Date createdDate) {
                    this.createdDate = createdDate;}
            public String getDescription() {
                    return description;}
            public void setDescription(String description) {
                    this.description = description;}
            public String getRemarks() {
                    return remarks;}
            public void setRemarks(String remarks) {
                    this.remarks = remarks;}
}
```

com.wipro.records.dao

DataRecordDAO.java

```
package com.wipro.records.dao;

import java.sql.*;
```

```java
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.List;
import com.wipro.records.bean.DataRecordBean;
import com.wipro.records.util.DBUtil;
public class DataRecordDAO {
    public String createRecord(DataRecordBean bean) {
        String result = "FAIL";
        try {
            Connection con = DBUtil.getDBConnection();
            String sql = "INSERT INTO DATA_RECORD_TB VALUES (?,?,?,?,?,?)";
            PreparedStatement ps = con.prepareStatement(sql);
            ps.setString(1, bean.getRecordId());
            ps.setString(2, bean.getRecordName());
            ps.setString(3, bean.getCategory());
            ps.setDate(4, new java.sql.Date(bean.getCreatedDate().getTime()));
            ps.setString(5, bean.getDescription());
            ps.setString(6, bean.getRemarks());
            int rows = ps.executeUpdate();
            if (rows > 0) {
                result = bean.getRecordId();}
        } catch (Exception e) {
            e.printStackTrace();}
        return result;
    }

    public DataRecordBean fetchRecord(String recordName, java.util.Date createdDate) {
        DataRecordBean bean = null;
        try {
            Connection con = DBUtil.getDBConnection();
            String sql = "SELECT * FROM DATA_RECORD_TB WHERE RECORDNAME=?
AND CREATED_DATE=?";
            PreparedStatement ps = con.prepareStatement(sql);
            ps.setString(1, recordName);
            ps.setDate(2, new java.sql.Date(createdDate.getTime()));
            ResultSet rs = ps.executeQuery();
            if (rs.next()) {
                bean = new DataRecordBean();
                bean.setRecordId(rs.getString(1));
                bean.setRecordName(rs.getString(2));
                bean.setCategory(rs.getString(3));
                bean.setCreatedDate(rs.getDate(4));
                bean.setDescription(rs.getString(5));
                bean.setRemarks(rs.getString(6));
            }
        } catch (Exception e) {
            e.printStackTrace();}
        return bean;}
    public boolean recordExists(String recordName, java.util.Date createdDate) {
        boolean exists = false;
        try {
            Connection con = DBUtil.getDBConnection();
            String sql = "SELECT COUNT(*) FROM DATA_RECORD_TB WHERE
RECORDNAME=? AND CREATED_DATE=?";
            PreparedStatement ps = con.prepareStatement(sql);
            ps.setString(1, recordName);
            ps.setDate(2, new java.sql.Date(createdDate.getTime()));
            ResultSet rs = ps.executeQuery();
            if (rs.next() && rs.getInt(1) > 0) {
                exists = true;
            }
        } catch (Exception e) {
            e.printStackTrace();}
        return exists;}

    public List<DataRecordBean> fetchAllRecords() {
        List<DataRecordBean> list = new ArrayList<>();
    try {
      Connection con = DBUtil.getDBConnection();
      Statement st = con.createStatement();
      ResultSet rs = st.executeQuery("SELECT * FROM DATA_RECORD_TB");
      while (rs.next()) {
        DataRecordBean bean = new DataRecordBean();
        bean.setRecordId(rs.getString(1));
        bean.setRecordName(rs.getString(2));
        bean.setCategory(rs.getString(3));
        bean.setCreatedDate(rs.getDate(4));
        bean.setDescription(rs.getString(5));
        bean.setRemarks(rs.getString(6));
        list.add(bean);
      }
    } catch (Exception e) {
      e.printStackTrace();}
    return list;}

  public String generateRecordID(String recordName, java.util.Date createdDate) {
    String recordId = "";
    try {
      Connection con = DBUtil.getDBConnection();
      Statement st = con.createStatement();
      ResultSet rs = st.executeQuery("SELECT DATA_RECORD_SEQ.NEXTVAL FROM
DUAL");
      int seq = 0;
      if (rs.next()) {
        seq = rs.getInt(1);}
      DateFormat df = new SimpleDateFormat("yyyyMMdd");
      String datePart = df.format(createdDate);
      String namePart = recordName.substring(0, 2).toUpperCase();
      recordId = datePart + namePart + seq;
    } catch (Exception e) {
      e.printStackTrace();}
```

```
        return recordId;}
}
```

com.wipro.records.service

Administrator.java

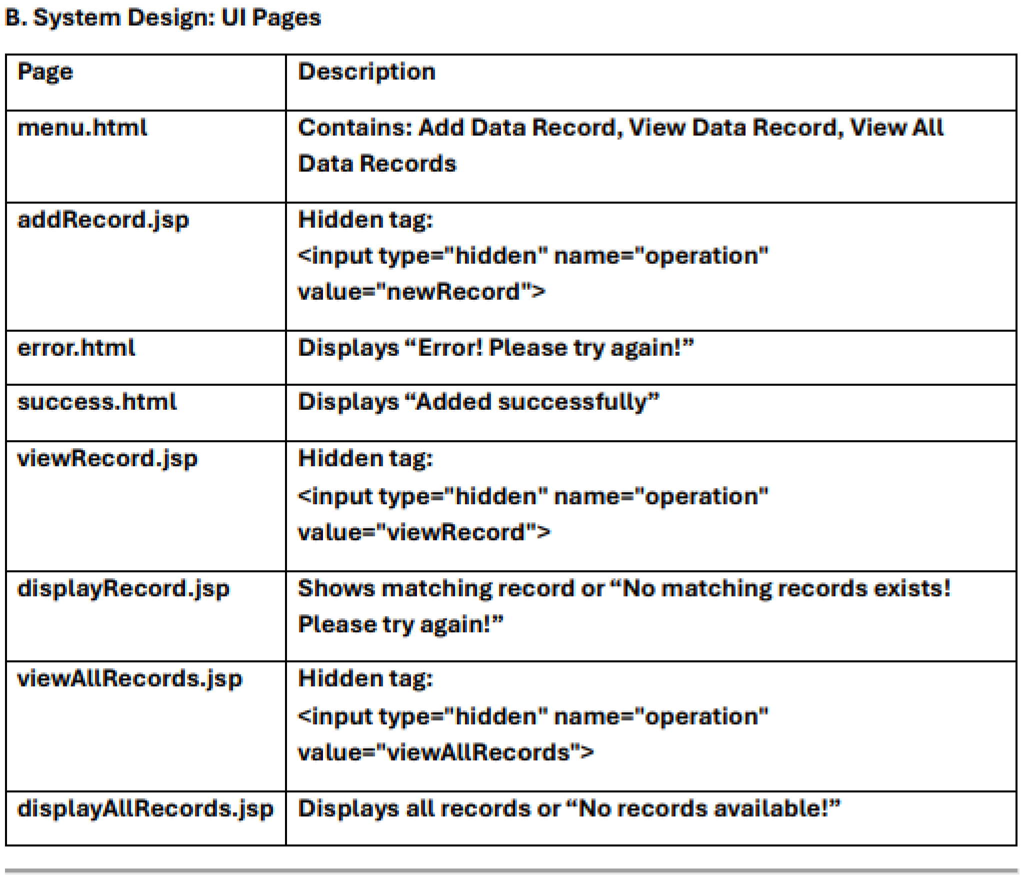
```
package com.wipro.records.service;

import java.util.Date;
import java.util.List;
import com.wipro.records.bean.DataRecordBean;
import com.wipro.records.dao.DataRecordDAO;
import com.wipro.records.util.InvalidInputException;
public class Administrator {
    DataRecordDAO dao = new DataRecordDAO();
    public String addRecord(DataRecordBean bean) {
        try {
            if (bean == null || bean.getRecordName() == null ||
                bean.getCategory() == null || bean.getCreatedDate() == null) {
                throw new InvalidInputException();}
            if (bean.getRecordName().length() < 2)
                return "INVALID RECORD NAME";
            if (bean.getCategory().length() < 2)
                return "INVALID CATEGORY";
            if (dao.recordExists(bean.getRecordName(), bean.getCreatedDate()))
                return "ALREADY EXISTS";
            String recordId = dao.generateRecordID(
                    bean.getRecordName(), bean.getCreatedDate());
            bean.setRecordId(recordId);
            return dao.createRecord(bean);
        } catch (InvalidInputException e) {
            return "INVALID INPUT";
        }}
    public DataRecordBean viewRecord(String recordName, Date createdDate) {
        return dao.fetchRecord(recordName, createdDate);
    }
    public List<DataRecordBean> viewAllRecords() {
        return dao.fetchAllRecords();
    }}
```

com.wipro.records.servlets

MainServlet.java

```
package com.wipro.records.servlets;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;
import javax.servlet.*;
import javax.servlet.http.*;
import com.wipro.records.bean.DataRecordBean;
import com.wipro.records.service.Administrator;
public class MainServlet extends HttpServlet {
                @Override
                protected void doGet(HttpServletRequest req, HttpServletResponse resp)
                        throws ServletException, IOException {
                    doPost(req, resp);
                }
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
            throws ServletException, IOException {
        String operation = req.getParameter("operation");
        Administrator admin = new Administrator();
        try {
            if ("newRecord".equals(operation)) {
                DataRecordBean bean = new DataRecordBean();
                bean.setRecordName(req.getParameter("recordName"));
                bean.setCategory(req.getParameter("category"));
                Date date = new SimpleDateFormat("yyyy-MM-dd")
                        .parse(req.getParameter("createdDate"));
                bean.setCreatedDate(date);
                bean.setDescription(req.getParameter("description"));
                bean.setRemarks(req.getParameter("remarks"));
                String result = admin.addRecord(bean);
                if ("FAIL".equals(result) || "INVALID INPUT".equals(result)) {
                    resp.sendRedirect("error.html");
                } else {
                    resp.sendRedirect("success.html");
                }
            }
            else if ("viewRecord".equals(operation)) {
                String name = req.getParameter("recordName");
                Date date = new SimpleDateFormat("yyyy-MM-dd")
                        .parse(req.getParameter("createdDate"));
                DataRecordBean bean = admin.viewRecord(name, date);
                req.setAttribute("bean", bean);
                RequestDispatcher rd =
                        req.getRequestDispatcher("displayRecord.jsp");
                rd.forward(req, resp);
            }
            else if ("viewAllRecords".equals(operation)) {
                List<DataRecordBean> list = admin.viewAllRecords();
                req.setAttribute("list", list);
                RequestDispatcher rd =
                        req.getRequestDispatcher("displayAllRecords.jsp");
                rd.forward(req, resp);
            }
        } catch (Exception e) {
            e.printStackTrace();
            resp.sendRedirect("error.html");
        }}}
```



B. System Design: UI Pages

Page	Description
menu.html	Contains: Add Data Record, View Data Record, View All Data Records
addRecord.jsp	Hidden tag: <input type="hidden" name="operation" value="newRecord">
error.html	Displays "Error! Please try again!"
success.html	Displays "Added successfully"
viewRecord.jsp	Hidden tag: <input type="hidden" name="operation" value="viewRecord">
displayRecord.jsp	Shows matching record or "No matching records exists! Please try again!"
viewAllRecords.jsp	Hidden tag: <input type="hidden" name="operation" value="viewAllRecords">
displayAllRecords.jsp	Displays all records or "No records available!"

menu.html

```
<!DOCTYPE html>
<html>
<head>
   <title>Menu</title>
</head>
<body>
<h2>Record Management System</h2>
<ul>
   <li><a href="addRecord.jsp">Add Data Record</a></li>
   <li><a href="viewRecord.jsp">View Data Record</a></li>
   <li><a href="viewAllRecords.jsp">View All Data Records</a></li>
</ul>
</body>
</html>
```

addRecord.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"%>
<!DOCTYPE html>
<html>
<head>
   <title>Add Record</title>
</head>
<body>
<h2>Add Data Record</h2>
<form action="MainServlet" method="post">
   <input type="hidden" name="operation" value="newRecord">
   Record Name:
   <input type="text" name="recordName"><br><br>
   Category:
   <input type="text" name="category"><br><br>
   Created Date:
   <input type="date" name="createdDate"><br><br>
   Description:
   <input type="text" name="description"><br><br>
   Remarks:
   <input type="text" name="remarks"><br><br>
   <input type="submit" value="Save Record">
</form>
</body>
</html>
```

error.html

```
<!DOCTYPE html>
<html>
<head>
   <title>Error</title>
</head>
<body>
```

```
<h2>Error! Please try again!</h2>
<a href="menu.html">Back to Menu</a>
</body>
</html>
```

success.html

```
<!DOCTYPE html>
<html>
<head>
   <title>Success</title>
</head>
<body>
<h2>Added successfully</h2>
<a href="menu.html">Back to Menu</a>
</body>
</html>
```

viewRecord.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"%>
<!DOCTYPE html>
<html>
<head>
   <title>View Record</title>
</head>
<body>
<h2>View Data Record</h2>
<form action="MainServlet" method="post">
   <input type="hidden" name="operation" value="viewRecord">
   Enter Record Name:
   <input type="text" name="recordname"><br><br>
   <input type="submit" value="View Record">
</form>
</body>
</html>
```

displayRecord.jsp

```
<%@ page import="com.wipro.records.bean.DataRecordBean"%>
<!DOCTYPE html>
<html>
<head>
   <title>Display Record</title>
</head>
<body>
<h2>Record Details</h2>
<%
   DataRecordBean record = (DataRecordBean) request.getAttribute("record");
   if (record == null) {
%>
   <p>No matching records exists! Please try again!</p>
<%
   } else {
%>
   <p>Record ID: <%= record.getRecordId() %></p>
   <p>Record Name: <%= record.getRecordName() %></p>
   <p>Category: <%= record.getCategory() %></p>
   <p>Created Date: <%= record.getCreatedDate() %></p>
   <p>Description: <%= record.getDescription() %></p>
   <p>Remarks: <%= record.getRemarks() %></p>
<%
   }
%>
<a href="menu.html">Back to Menu</a>
</body>
</html>
```

viewAllRecords.jsp

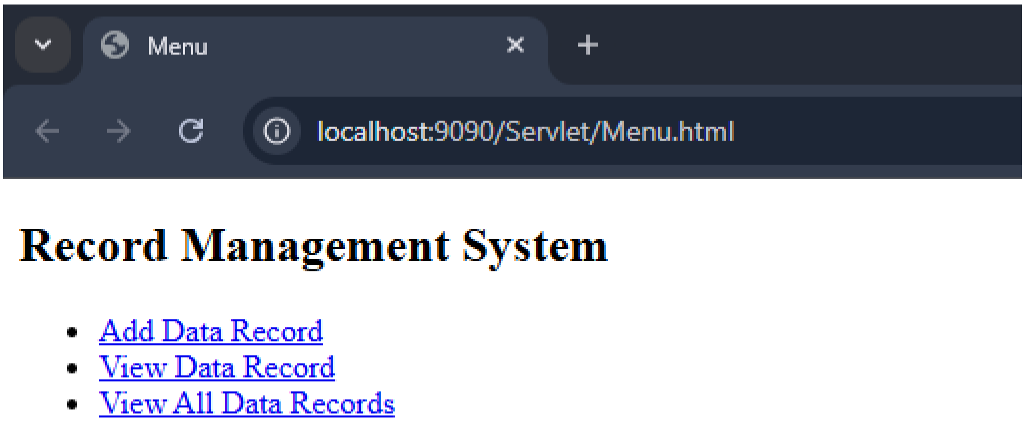
```
<%@ page language="java" contentType="text/html; charset=UTF-8"%>
<!DOCTYPE html>
<html>
```

```
<head>
    <title>View All Records</title>
</head>
<body>
<h2>View All Data Records</h2>
<form action="MainServlet" method="post">
    <input type="hidden" name="operation" value="viewAllRecords">
    <input type="submit" value="View All Records">
</form>
</body>
</html>
```

displayAllRecords.jsp

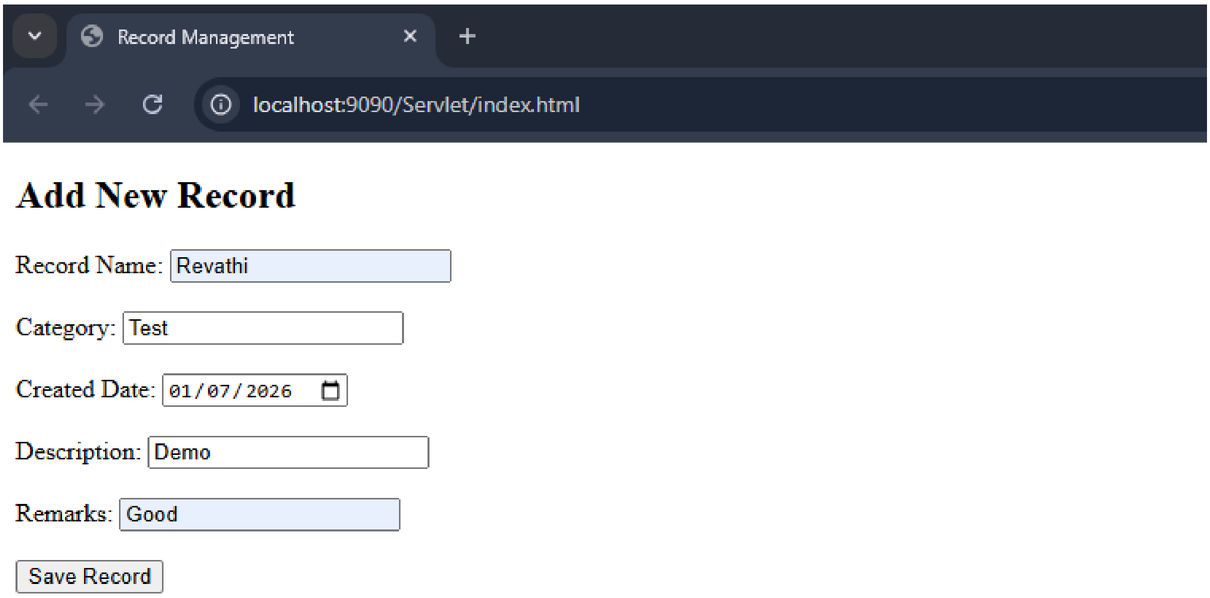
```
<%@ page import="java.util.List"%>
<%@ page import="com.wipro.records.bean.DataRecordBean"%>
<!DOCTYPE html>
<html>
<head>
    <title>All Records</title>
</head>
<body>
<h2>All Records</h2>
<%
    List<DataRecordBean> list = (List<DataRecordBean>) request.getAttribute("recordList");
    if (list == null || list.isEmpty()) {
%>
    <p>No records available!</p>
<%
    } else {
%>
<table border="1">
    <tr>
        <th>ID</th>
        <th>Name</th>
        <th>Category</th>
        <th>Date</th>
        <th>Description</th>
        <th>Remarks</th>
    </tr>
<%
    for (DataRecordBean r : list) {
%>
    <tr>
        <td><%= r.getRecordId() %></td>
        <td><%= r.getRecordName() %></td>
        <td><%= r.getCategory() %></td>
        <td><%= r.getCreatedDate() %></td>
        <td><%= r.getDescription() %></td>
        <td><%= r.getRemarks() %></td>
    </tr>
<%
    }
%>
</table>
<%
    }
%>
<a href="menu.html">Back to Menu</a>
</body>
</html>
```

Output Screenshots:



Record Management System

- Add Data Record
- View Data Record
- View All Data Records



Add New Record

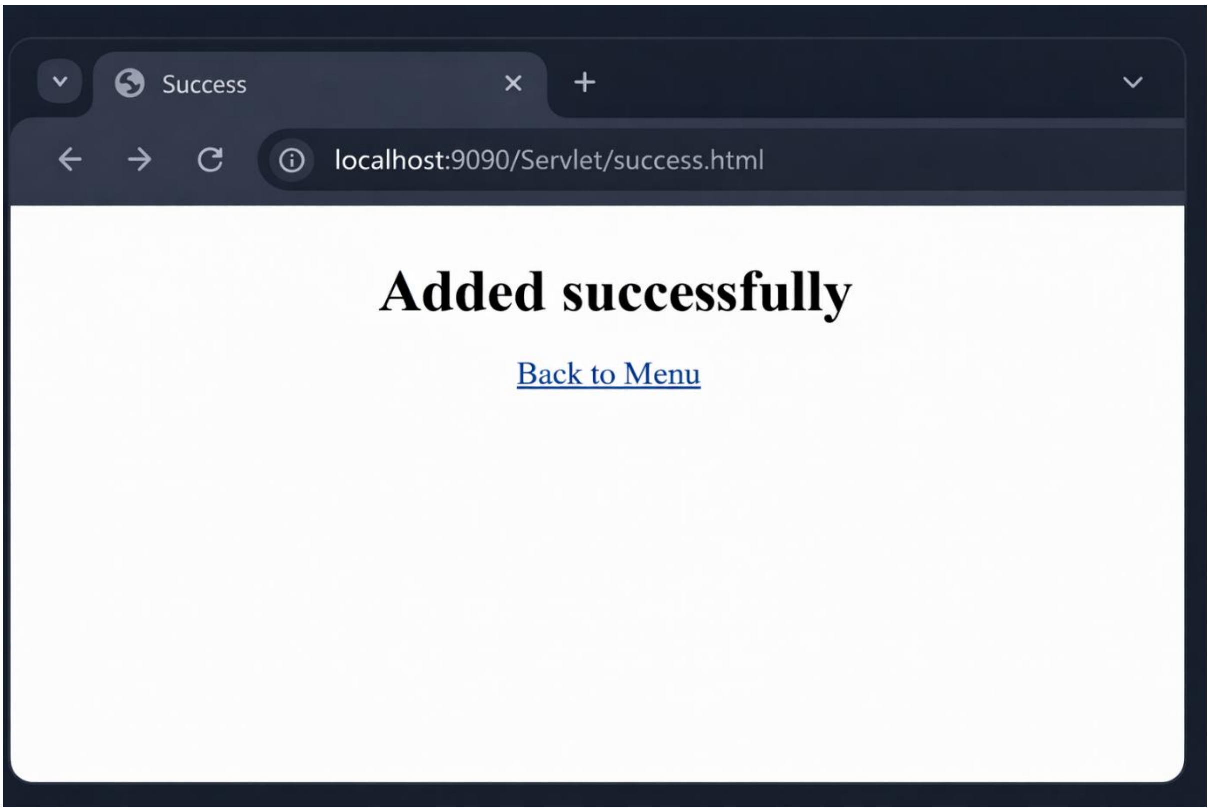
Record Name: Revathi

Category: Test

Created Date: 01/07/2026

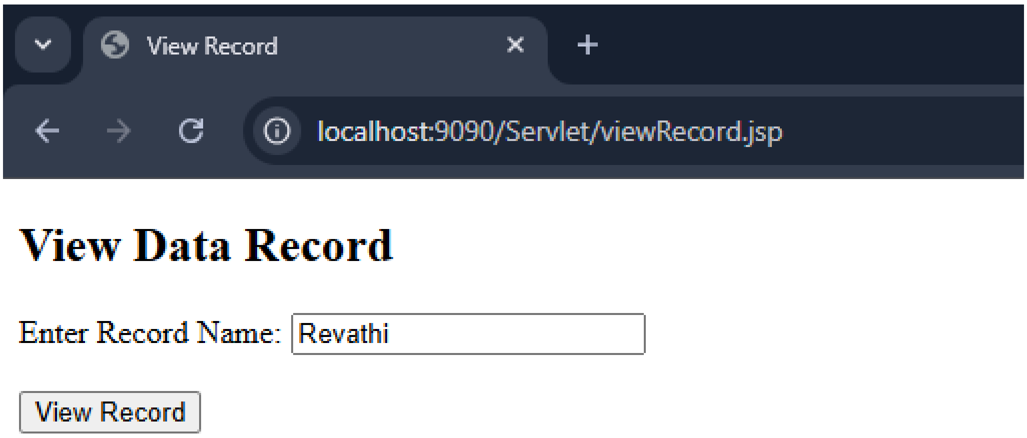
Description: Demo

Remarks: Good



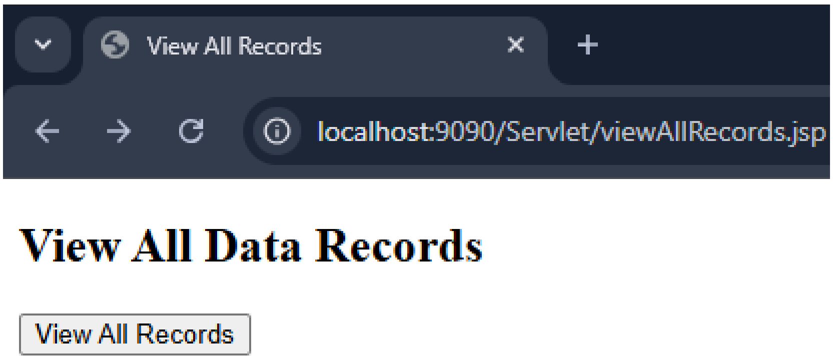
Added successfully

Back to Menu



View Data Record

Enter Record Name: Revathi



View All Data Records