Practical 5

Aim: Connecting Ajax with Maria DB

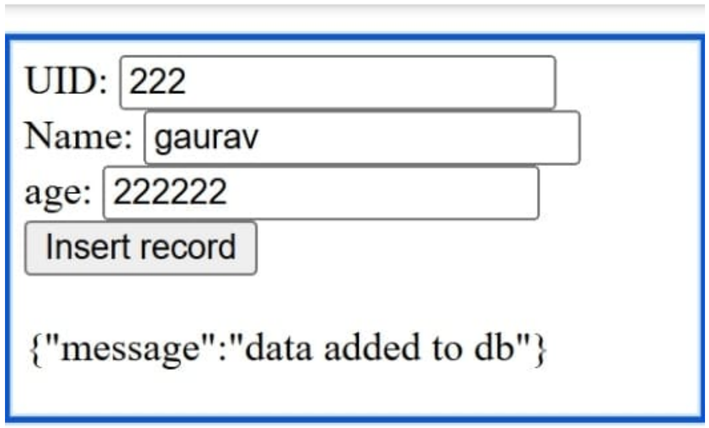
frontend.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>

        <scriptsrc="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>

        <script type="text/javascript">
                function check(){
                        alert("call received");

                        $.ajax({
                                 type:"post",
                                 url: "ajaxprocess.jsp",
        data:"uid="+$("#uid").val()+"&name="+$("#name").val()+"&age="+$("#age").val(),
                                success: function(result){
                                        $("#output").append(result);
                                 },
                                error:function(){
                                        alert("error in alert");
                                }
                        });

                }
        </script>
</head>
<body>
                UID: <input id="uid" type='text' name="uid" value="" /> <br />
                Name: <input id="name" type='text' name="name" value="" /> <br />
                age: <input id="age" type="text" name="age" value="" /> <br />
                <input type="button" onClick="check()" value="Insert record" />
        <br />
        <br />
        <div id="output"></div>
</body>
</html>
```

ajaxprocess.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1" import="java.sql.*, org.json.simple.JSONObject "%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<%
        int uid = Integer.parseInt(request.getParameter("uid"));
        String name = request.getParameter("name");
        int age = Integer.parseInt(request.getParameter("age"));
        JSONObject jsonResponse = new JSONObject();

        try{
                Class.forName("org.mariadb.jdbc.Driver");
                Connection connection =
DriverManager.getConnection("jdbc:mariadb://localhost:3306/db","root","maria");

                PreparedStatement ps = connection.prepareStatement ("insert into students values
        (?,?,?)");
                ps.setInt(1, uid);
                ps.setString(2, name);
                ps.setInt(3, age);

                int rows = ps.executeUpdate();
                if(rows > 0){
                        jsonResponse.put("message","data added to db");
                }else{
                        jsonResponse.put("message","data not added to db");
                        }
                }
                catch (Exception e){
                        System.out.print("Error occureed due to "+e.toString());
                        e.printStackTrace();
                        jsonResponse.put("message","error occured while inserting data");
                }

                response.getWriter().write(jsonResponse.toString());
                out.flush(); // returns to first page
        %>
</body>
</html>
```

Output


UID: 222

Name: gaurav

age: 222222

{"message":"data added to db"}

Algorithm:

1. **Frontend Page Setup (frontend.jsp)**
   - Create an HTML form with three input fields:
     - UID
     - Name
     - Age
   - Add a button with an `onClick()` event that triggers the `check()` JavaScript function.
   - Include jQuery library for AJAX functionality.
2. **AJAX Call (Using jQuery)**
   - When the button is clicked, the `check()` function is called.
   - The function triggers an `alert()` to confirm the click event.
   - AJAX request is sent using the `$.ajax()` method with:
     - HTTP **POST** method
     - URL: `ajaxprocess.jsp`
     - Data: UID, Name, Age (Fetched from input fields)
   - On **Success**: The server response is appended to the `div` with `id="output"`.
   - On **Error**: An alert message is shown.
3. **Backend Processing (ajaxprocess.jsp)**
   - Fetch input values from the request using `request.getParameter()`.
   - Convert the received data into proper data types.
   - Use **JDBC** to connect to the **MariaDB** database.
   - Insert the received data into the `students` table using a **PreparedStatement**.
   - Check if the insertion was successful:
     - If yes → Send JSON response `{ "message": "data added to db" }`
     - If no → Send JSON response `{ "message": "data not added to db" }`
   - Handle exceptions using `try-catch`.
   - Send the JSON response to the frontend using `response.getWriter().write()`.

Learnings:

AJAX: Used to send data to the server without reloading the page.

JSON Response: JSON data format used for server-to-client communication.

JDBC: Database connectivity to **MariaDB**.

PreparedStatement: Prevents **SQL Injection** by parameterized queries.

Exception Handling: Catches and logs exceptions.

jQuery: Simplifies **AJAX** requests and **DOM manipulation**.

HTTP POST Method: Securely sends data to the server.

Response Handling: Appends server response dynamically without refreshing the page.

Error & Debug

AJAX Call Not Working (No Alert Received)

1. **Error Reason:** `onClick` function not triggering.
2. **Debugging Steps:**
    - Check if the **jQuery CDN** is loading correctly.
    - Use `console.log()` inside the `check()` function to verify if the function is called.
    - Add `alert("Button Clicked")` at the start of the function.