WAD-Case Study

R.AKHILDEV REDDY
HU22CSEN0100716
SET-4

Registration form using Servlets

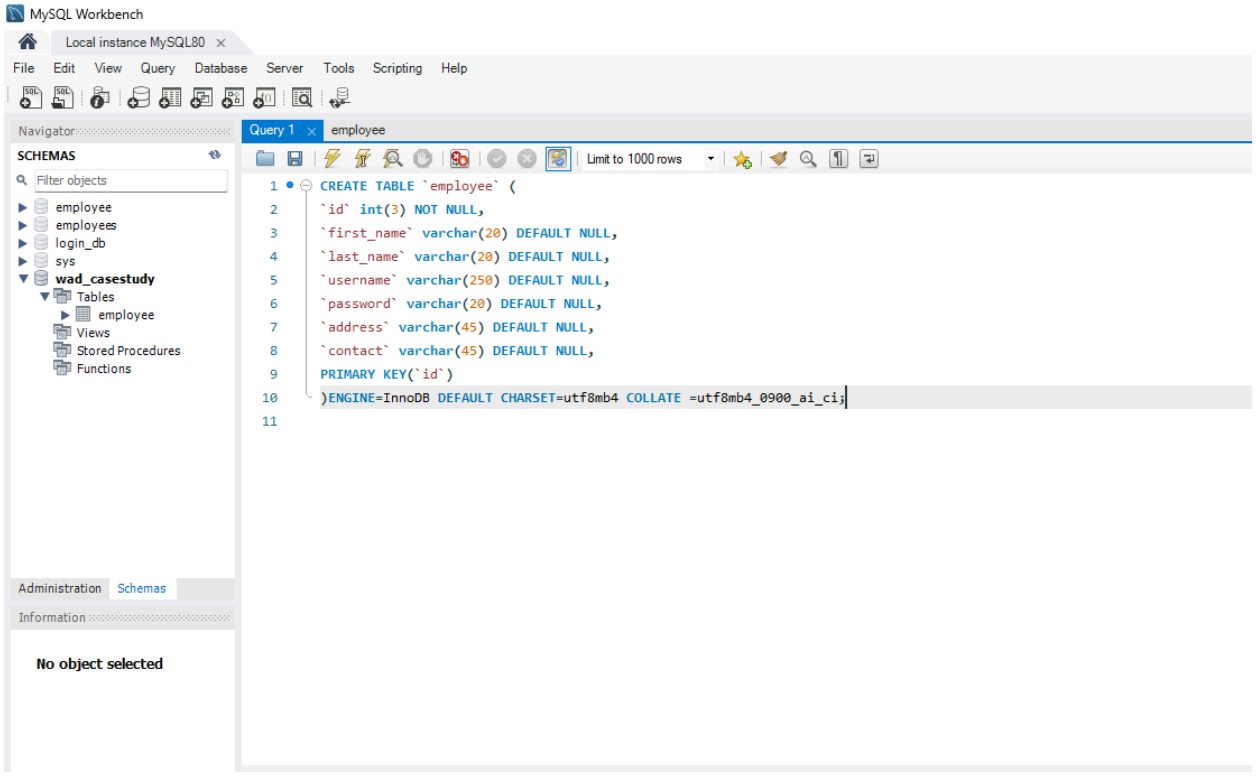
Required tools:

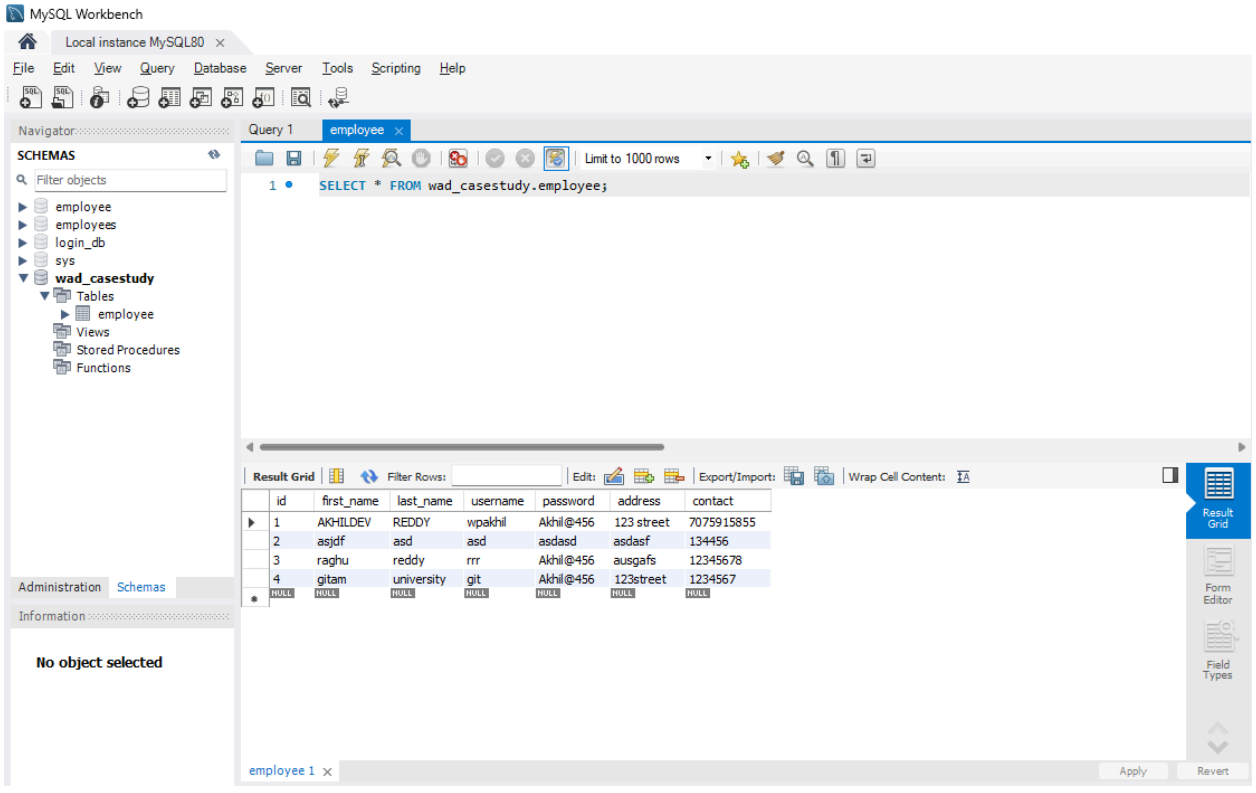
1. JSP 2.2+
2. IDE - Eclipse
3. JDK - 1.8
4. Apache Tomcat - 11.0
5. JSTL - 1.2.1
6. Servlet API - 2.5
7. MySQL - mysql-connector-java-8.0.13.jar

Development Steps:

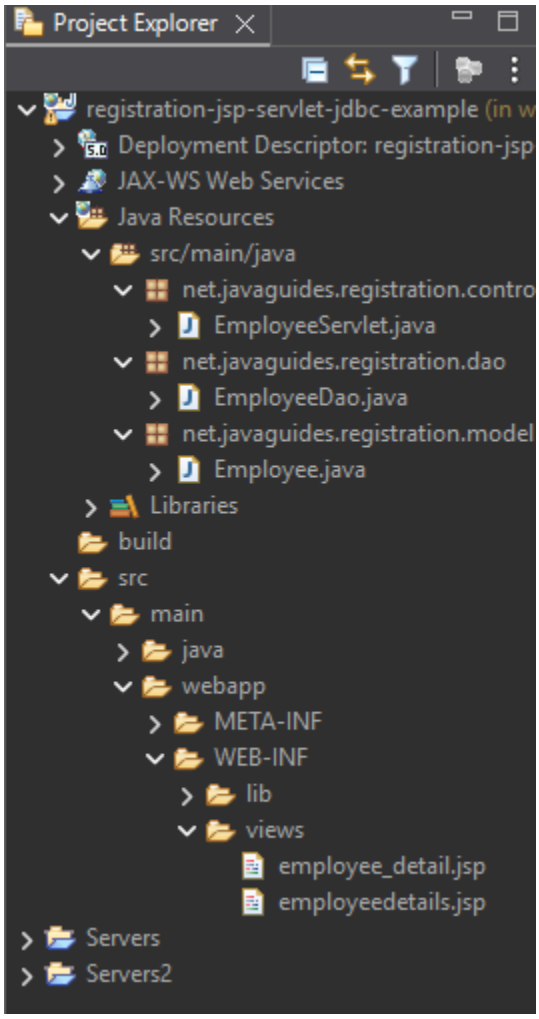
1. Create an Eclipse Dynamic web page
2. Add dependencies
3. Project structure
4. MySQL database structure
5. Create a JavaBean - Employee.java
6. Create a EmployeeDao.java
7. Create a EmployeeServlet.java
8. Create a employeeregister.jsp
9. Create a employeedetail.jsp
10. Demo

Database:


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
employee
employees
login_db
sys
wad_casestudy
Tables
employee
Views
Stored Procedures
Functions
Administration Schemas
Information
No object selected
Query 1
employee
Limit to 1000 rows
1 CREATE TABLE `employee` (
2 `id` int(3) NOT NULL,
3 `first_name` varchar(20) DEFAULT NULL,
4 `last_name` varchar(20) DEFAULT NULL,
5 `username` varchar(250) DEFAULT NULL,
6 `password` varchar(20) DEFAULT NULL,
7 `address` varchar(45) DEFAULT NULL,
8 `contact` varchar(45) DEFAULT NULL,
9 PRIMARY KEY(`id`)
10)ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE =utf8mb4_0900_ai_ci;
11


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
employee
employees
login_db
sys
wad_casestudy
Tables
employee
Views
Stored Procedures
Functions
Administration Schemas
Information
No object selected
Query 1
employee
Limit to 1000 rows
1 SELECT * FROM wad_casestudy.employee;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
id first_name last_name username password address contact
1 AKHILDEV REDDY wpakhil Akhil@456 123 street 7075915855
2 asjdf asd asd asdasd asdasf 134456
3 raghu reddy rrr Akhil@456 ausgafs 12345678
4 gitam university git Akhil@456 123street 1234567
Result Grid
Form Editor
Field Types
employee 1
Apply
Revert

Project Structure:



Servlet code:

```
package net.javaguides.registration.controller;

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;

import net.javaguides.registration.dao.EmployeeDao;
import net.javaguides.registration.model.Employee;
/**
```

```
 * Servlet implementation class EmployeeServlet
 */
@WebServlet("/register")
public class EmployeeServlet extends HttpServlet {
	private static final long serialVersionUID = 1L;

	private EmployeeDao employeeDao = new EmployeeDao();

   /**
    * Default constructor.
    */
   public EmployeeServlet() {
	super();
      // TODO Auto-generated constructor stub
   }

	/**
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
		// TODO Auto-generated method stub
		response.getWriter().append("Served at: ").append(request.getContextPath());

		jakarta.servlet.RequestDispatcher dispatcher =
request.getRequestDispatcher("/WEB-INF/views/employeedetails.jsp");
		dispatcher.forward(request, response);
	}

	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
		 String firstName = request.getParameter("firstName");
	    String lastName = request.getParameter("lastName");
	    String UserName = request.getParameter("username");
	    String password = request.getParameter("password");
	    String address = request.getParameter("address");
	    String contact = request.getParameter("contact");
```

```
            Employee employee = new Employee();
              employee.setFirstName(firstName);
              employee.setLastName(lastName);
              employee.setUserName(UserName);
              employee.setPassword(password);
              employee.setContact(contact);
              employee.setAddress(address);

              try {
                 employeeDao.registerEmployee(employee);
              } catch (Exception e) {
                 // TODO Auto-generated catch block
                 e.printStackTrace();
              }

              jakarta.servlet.RequestDispatcher dispatcher =
request.getRequestDispatcher("/WEB-INF/views/employee_detail.jsp");
                        dispatcher.forward(request, response);
        }

}
```

EmployeeDao.java code:

```
package net.javaguides.registration.dao;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

import net.javaguides.registration.model.Employee;

public class EmployeeDao {
```

```
   public int registerEmployee(Employee employee) throws ClassNotFoundException {
      String INSERT_USERS_SQL = "INSERT INTO employee" +
         "  (first_name, last_name, username, password, address, contact) VALUES " +
         " (?, ?, ?, ?,?,?);";

      int result = 0;

      Class.forName("com.mysql.jdbc.Driver");

      try (Connection connection = DriverManager
         .getConnection("jdbc:mysql://localhost:3306/wad_casestudy?useSSL=false", "root",
"Akhil@456");

         // Step 2:Create a statement using connection object
         PreparedStatement preparedStatement =
connection.prepareStatement(INSERT_USERS_SQL)) {
         preparedStatement.setString(1, employee.getFirstName());
         preparedStatement.setString(2, employee.getLastName());
         preparedStatement.setString(3, employee.getUserName());
         preparedStatement.setString(4, employee.getPassword());
         preparedStatement.setString(5, employee.getAddress());
         preparedStatement.setString(6, employee.getContact());

         System.out.println(preparedStatement);
         // Step 3: Execute the query or update query
         result = preparedStatement.executeUpdate();

      } catch (SQLException e) {
         e.printStackTrace();
      }
      return result;
   }}
```

Employee.java code:

```
package net.javaguides.registration.model;

public class Employee {
         private String firstName;
         private String lastName;
```

```
private String userName;
private String password;
private String address;
private String contact;
public String getFirstName() {
	return firstName;
}
public void setFirstName(String firstName) {
	this.firstName = firstName;
}
public String getLastName() {
	return lastName;
}
public void setLastName(String lastName) {
	this.lastName = lastName;
}
public String getUserName() {
	return userName;
}
public void setUserName(String userName) {
	this.userName = userName;
}
public String getPassword() {
	return password;
}
public void setPassword(String password) {
	this.password = password;
}
public String getAddress() {
	return address;
}
public void setAddress(String address) {
	this.address = address;
}
public String getContact() {
	return contact;
}
public void setContact(String contact) {
	this.contact = contact;
}}
```

Employeedetails.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>
</head>
<body>
 <div align="center">
  <h1>Employee Register Form</h1>
  <form action="<%= request.getContextPath() %>/register" method="post">
   <table style="with: 80%">
    <tr>
     <td>First Name</td>
     <td><input type="text" name="firstName" /></td>
    </tr>
    <tr>
     <td>Last Name</td>
     <td><input type="text" name="lastName" /></td>
    </tr>
    <tr>
     <td>UserName</td>
     <td><input type="text" name="username" /></td>
    </tr>
    <tr>
     <td>Password</td>
     <td><input type="password" name="password" /></td>
    </tr>
    <tr>
     <td>Address</td>
     <td><input type="text" name="address" /></td>
    </tr>
    <tr>
     <td>Contact No</td>
     <td><input type="text" name="contact" /></td>
    </tr>
```

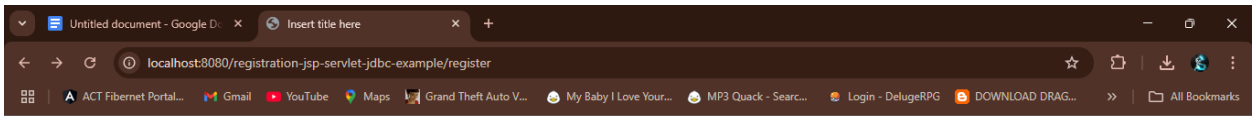
```
  </table>
  <input type="submit" value="Submit" />
 </form>
</div>
</body>
</html>
```

Employee_details.jsp:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<h1>Employee successfully registered</h1>

</body>
</html>
```

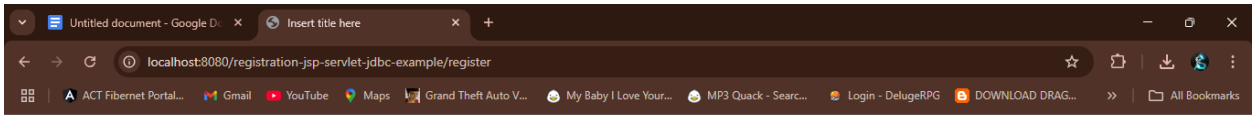
Output:



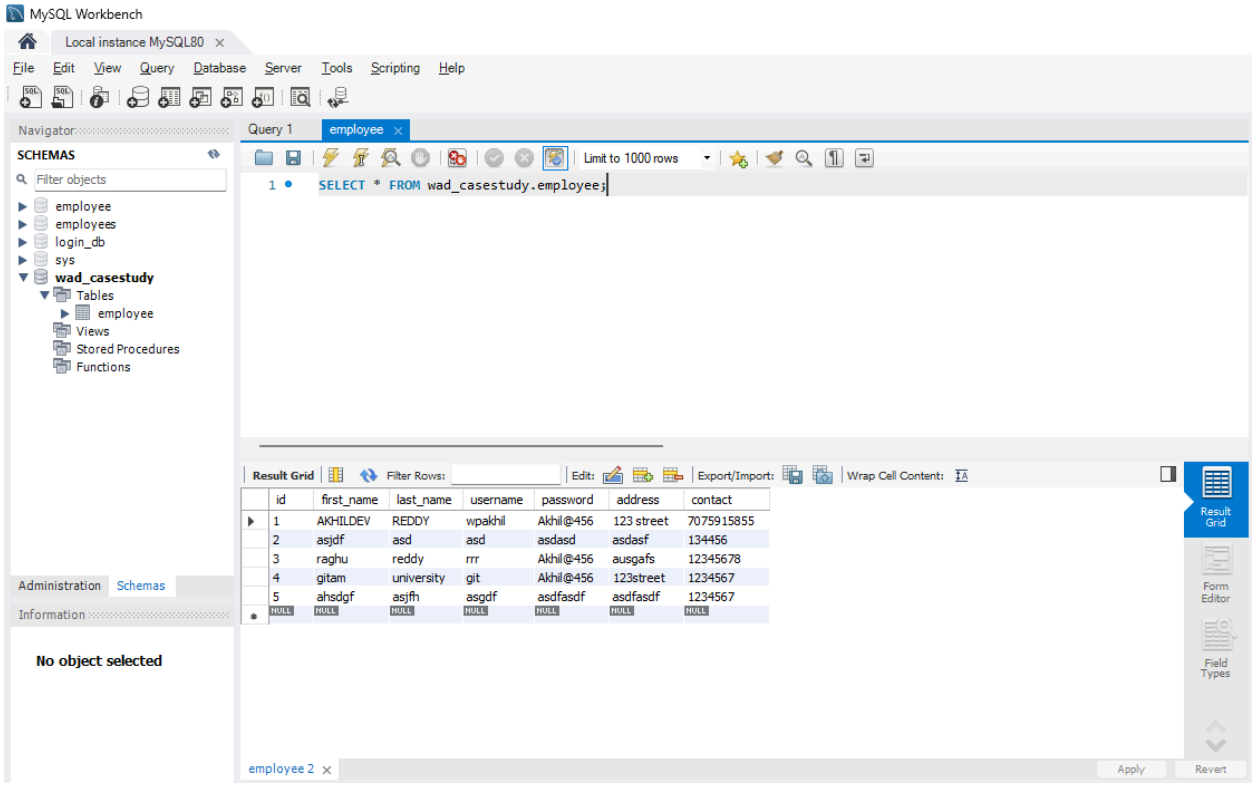
Employee Register Form

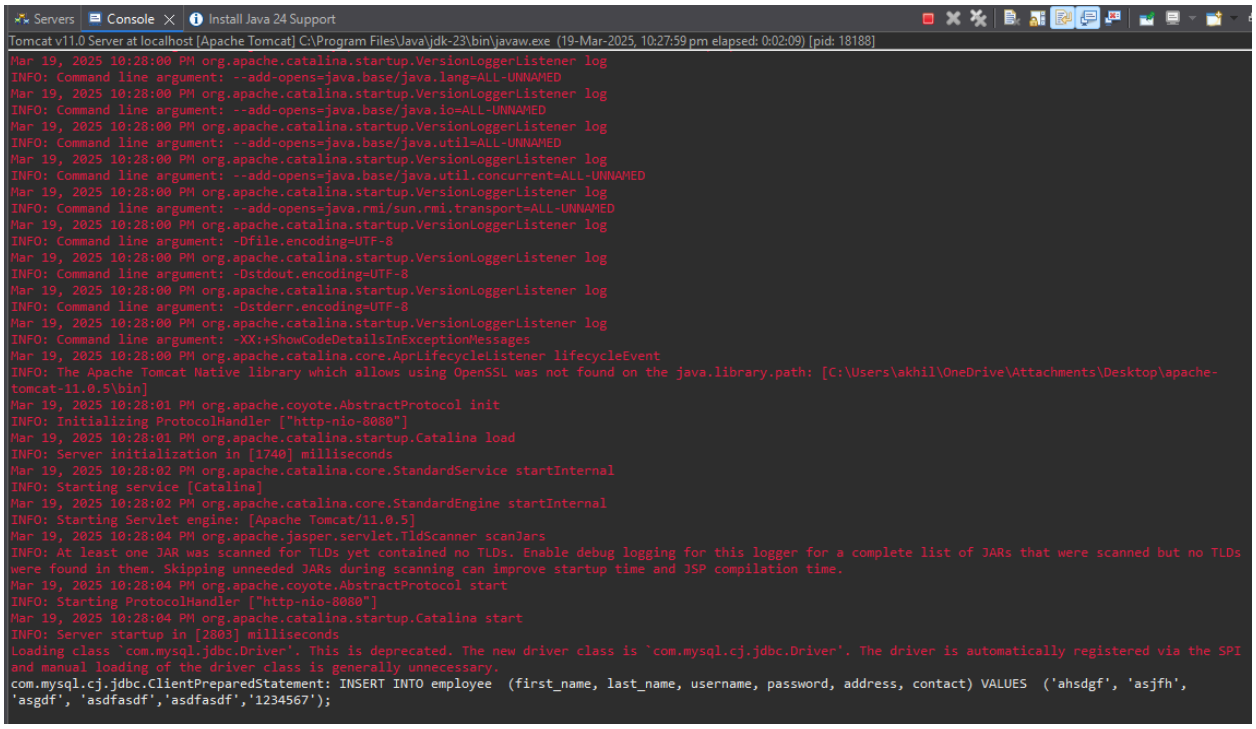
First Name	ahsdgf
Last Name	asjfh
UserName	asgdf
Password	••••••
Address	asdfasdf
Contact No	1234567

Submit



Employee successfully registered


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
employee
employees
login_db
sys
wad_casestudy
Tables
employee
Views
Stored Procedures
Functions
Administration Schemas
Information
No object selected
Query 1
employee
Limit to 1000 rows
1 SELECT * FROM wad_casestudy.employee;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
id first_name last_name username password address contact
1 AKHILDEV REDDY wpakhil Akhil@456 123 street 7075915855
2 asjdf asd asd asdasd asdasf 134456
3 raghu reddy rrr Akhil@456 ausgafs 12345678
4 gitam university git Akhil@456 123street 1234567
5 ahsdgf asjfh asgdf asdfasdf asdfasdf 1234567
Result Grid
Form Editor
Field Types
employee 2
Apply
Revert


Servers
Console
Install Java 24 Support
Tomcat v11.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jdk-23\bin\javaw.exe (19-Mar-2025, 10:27:59 pm elapsed: 0:02:09) [pid: 18188]
Mar 19, 2025 10:28:00 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: Command line argument: --add-opens=java.base/java.lang=ALL-UNNAMED
Mar 19, 2025 10:28:00 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: Command line argument: --add-opens=java.base/java.io=ALL-UNNAMED
Mar 19, 2025 10:28:00 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: Command line argument: --add-opens=java.base/java.util=ALL-UNNAMED
Mar 19, 2025 10:28:00 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: Command line argument: --add-opens=java.base/java.util.concurrent=ALL-UNNAMED
Mar 19, 2025 10:28:00 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: Command line argument: --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
Mar 19, 2025 10:28:00 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: Command line argument: -Dfile.encoding=UTF-8
Mar 19, 2025 10:28:00 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: Command line argument: -Dstdout.encoding=UTF-8
Mar 19, 2025 10:28:00 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: Command line argument: -Dstderr.encoding=UTF-8
Mar 19, 2025 10:28:00 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: Command line argument: -XX:+ShowCodeDetailsInExceptionMessages
Mar 19, 2025 10:28:00 PM org.apache.catalina.core.AprLifecycleListener lifecycleEvent
INFO: The Apache Tomcat Native library which allows using OpenSSL was not found on the java.library.path: [C:\Users\akhil\OneDrive\Attachments\Desktop\apache-tomcat-11.0.5\bin]
Mar 19, 2025 10:28:01 PM org.apache.coyote.AbstractProtocol init
INFO: Initializing ProtocolHandler ["http-nio-8080"]
Mar 19, 2025 10:28:01 PM org.apache.catalina.startup.Catalina load
INFO: Server initialization in [1740] milliseconds
Mar 19, 2025 10:28:02 PM org.apache.catalina.core.StandardService startInternal
INFO: Starting service [Catalina]
Mar 19, 2025 10:28:02 PM org.apache.catalina.core.StandardEngine startInternal
INFO: Starting Servlet engine: [Apache Tomcat/11.0.5]
Mar 19, 2025 10:28:04 PM org.apache.jasper.servlet.TldScanner scanJars
INFO: At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug logging for this logger for a complete list of JARs that were scanned but no TLDs were found in them. Skipping unneeded JARs during scanning can improve startup time and JSP compilation time.
Mar 19, 2025 10:28:04 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-nio-8080"]
Mar 19, 2025 10:28:04 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in [2803] milliseconds
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The driver is automatically registered via the SPI and manual loading of the driver class is generally unnecessary.
com.mysql.cj.jdbc.ClientPreparedStatement: INSERT INTO employee (first_name, last_name, username, password, address, contact) VALUES ('ahsdgf', 'asjfh', 'asgdf', 'asdfasdf','asdfasdf','1234567');