1 Set Up JDBC Environment

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>JDBC Setup</title>
</head>
<body>
<a href="init">Initialize JDBC</a><br>

</body>
</html>
```

DBConnection

```
package com.ecommerce;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnection {

    private Connection connection;

    public DBConnection(String dbURL, String user, String pwd) throws ClassNotFoundException,
SQLException{

        Class.forName("com.mysql.jdbc.Driver");
        this.connection = DriverManager.getConnection(dbURL, user, pwd);
    }

    public Connection getConnection(){
        return this.connection;
```

```
	}

	public void closeConnection() throws SQLException {
		if (this.connection != null)
			this.connection.close();
	}
}
```

DemoJDBC

```
import java.io.IOException;
import java.io.InputStream;
import java.io.PrintWriter;
import java.math.BigDecimal;
import java.sql.CallableStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Properties;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.ecommerce.DBConnection;

/**
* Servlet implementation class DemoJDBC
*/
@WebServlet("/DemoJDBC")
```

```
public class DemoJDBC extends HttpServlet {
	private static final long serialVersionUID = 1L;

   /**
* @see HttpServlet#HttpServlet()
*/
   public DemoJDBC() {
      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

            try {
                  PrintWriter out = response.getWriter();
                  out.println("<html><body>");

                  InputStream in = getServletContext().getResourceAsStream("/WEB-INF/config.properties");
                  Properties props = new Properties();
                  props.load(in);

                  DBConnection conn = new DBConnection(props.getProperty("url"),
props.getProperty("userid"), props.getProperty("password"));
                  out.println("DB Connection initialized.<br>");

                  conn.closeConnection();
```

```
            out.println("DB Connection closed.<br>");



            out.println("</body></html>");
            conn.closeConnection();

        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }

}
```

2.Set Up JDBC Environment ..

2.1.1. Creating a Dynamic Web Project

1. Open Eclipse IDE.
2. Go to `File -> New -> Dynamic Web Project`.
3. Enter the project name (e.g., `JDBCDemo`) and click "Next."
4. Choose the target runtime (Apache Tomcat) and configure the project as needed. Click "Finish."

2.1.2. Pushing the Code to Your GitHub Repositories

1. If you haven't initialized a Git repository, right-click on the project, select `Team -> Share Project`, and follow the steps to create a Git repository.
2. Commit your changes: right-click on the project, select `Team -> Commit`.
3. Push to GitHub: right-click on the project, select `Team -> Remote -> Push to Upstream`.

2.1.3. Running the Project

1. Open a web browser and navigate to `http://localhost:8080/JDBCDemo/` to ensure the project is running.

2.1.4. Publishing and Starting the Project

1. Right-click on the project and select `Run As -> Run on Server`.
2. Choose the configured Apache Tomcat server and click "Finish."

2.1.5. Building the Project

1. Ensure the project is built by right-clicking on the project and selecting `Build Project`.

2.1.6. Checking for servlet-api.jar

1. Right-click on the project and select `Properties`.
2. In the properties window, navigate to `Java Build Path -> Libraries`.
3. Ensure that `Apache Tomcat` is listed, and `Servlet API` is included. If not, add it by clicking "Add Library" and selecting "Server Runtime."

2.1.7. Configuring web.xml

1. In the `WEB-INF` folder, create a `web.xml` file.
2. Configure the servlet and servlet-mapping in `web.xml`.

```
<web-app>
  <servlet>
    <servlet-name>DemoJDBC</servlet-name>
    <servlet-class>com.example.DemoJDBC</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>DemoJDBC</servlet-name>
    <url-pattern>/DemoJDBC</url-pattern>
```

```
    </servlet-mapping>
</web-app>
```

2.1.8. Creating a DemoJDBC Servlet

1. Create a servlet class (e.g., `DemoJDBC`) in the `src` folder.

```
package com.example;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/DemoJDBC")
public class DemoJDBC extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        // TODO: Implement JDBC operations here
        PrintWriter out = response.getWriter();
        out.println("JDBC Demo Servlet");
    }
}
```

2.1.9. Creating a config.properties File to Store JDBC Credentials

1. Create a `config.properties` file in the `src` folder.

```
jdbc.url=jdbc:mysql://localhost:3306/your_database
```

```
jdbc.username=your_username

jdbc.password=your_password
```

2.1.10. Creating a DBConnection Class to Initiate a JDBC Connection in Code

1. Create a `DBConnection` class to manage JDBC connections.

```
package com.example;


import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.util.Properties;


public class DBConnection {

    private static Connection connection = null;


    public static Connection getConnection() {

        if (connection != null) {

            return connection;

        } else {

            try {

                Properties properties = new Properties();


properties.load(DBConnection.class.getClassLoader().getResourceAsStream("config.properties"));


                String url = properties.getProperty("jdbc.url");

                String username = properties.getProperty("jdbc.username");

                String password = properties.getProperty("jdbc.password");


                connection = DriverManager.getConnection(url, username, password);

            } catch (SQLException | IOException e) {

                e.printStackTrace();
```

```
        }
        return connection;
      }
    }
}
```

2.1.11. Creating an HTML Page index.html

1. Create an `index.html` file in the `WebContent` folder.

```
<!DOCTYPE html>
<html>
<head>
  <title>JDBC Demo</title>
</head>
<body>
  <h1>JDBC Demo</h1>
  <form action="DemoJDBC" method="get">
    <input type="submit" value="Run JDBC Demo">
  </form>
</body>
</html>
```

2.1.12. Adding the JAR Files for MySQL Connection for Java

1. Download the MySQL Connector/J JAR file (e.g., `mysql-connector-java-8.0.16.jar`).
2. Copy the JAR file to the `WEB-INF/lib` folder of your project.

3.JDBC Connections, Statements, and ResultSets

```
<!DOCTYPE html>
<html>
<head>
```

```
<meta charset="UTF-8">
<title>JDBC Statements and Resultsets</title>
</head>
<body>
<a href="list">Product Info</a><br>

</body>
</html>
```

DBConnection

```
package com.ecommerce;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnection {

    private Connection connection;

    public DBConnection(String dbURL, String user, String pwd) throws ClassNotFoundException,
SQLException{

        Class.forName("com.mysql.jdbc.Driver");
        this.connection = DriverManager.getConnection(dbURL, user, pwd);
    }

    public Connection getConnection(){
        return this.connection;
    }

    public void closeConnection() throws SQLException {
        if (this.connection != null)
```

```
                this.connection.close();
        }
}
```

ProductDetails

```
import java.io.IOException;
import java.io.InputStream;
import java.io.PrintWriter;
import java.math.BigDecimal;
import java.sql.CallableStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Properties;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.ecommerce.DBConnection;

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

    /**
* @see HttpServlet#HttpServlet()
```

```
*/
   public ProductDetails() {
      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

            try {
                  PrintWriter out = response.getWriter();
                  out.println("<html><body>");

                  InputStream in = getServletContext().getResourceAsStream("/WEB-INF/config.properties");
                  Properties props = new Properties();
                  props.load(in);

                  DBConnection conn = new DBConnection(props.getProperty("url"),
props.getProperty("userid"), props.getProperty("password"));
                  Statement stmt =
conn.getConnection().createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
                  stmt.executeUpdate("insert into eproduct (name, price, date_added) values ('New
Product', 17800.00, now())");
                  ResultSet rst = stmt.executeQuery("select * from eproduct");

                  while (rst.next()) {
                        out.println(rst.getInt("ID") + ", " + rst.getString("name") + "<Br>");
```

```
                }

                stmt.close();

                out.println("</body></html>");
                conn.closeConnection();

            } catch (ClassNotFoundException e) {
                e.printStackTrace();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
            // TODO Auto-generated method stub
            doGet(request, response);
        }

}
```

4 Stored Procedures with Exceptions

```
<!DOCTYPE html>
<html>
```

```
<head>
<meta charset="UTF-8">
<title>JDBC Stored Procedures</title>
</head>
<body>
<a href="list">Product Info</a><br>

</body>
</html>
```

DBConnection

```
package com.ecommerce;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnection {

    private Connection connection;

    public DBConnection(String dbURL, String user, String pwd) throws ClassNotFoundException,
SQLException{

        Class.forName("com.mysql.jdbc.Driver");
        this.connection = DriverManager.getConnection(dbURL, user, pwd);
    }
```

```
	public Connection getConnection(){
		return this.connection;
	}

	public void closeConnection() throws SQLException {
		if (this.connection != null)
			this.connection.close();
	}
}
```

ProductDetails

```
import java.io.IOException;
import java.io.InputStream;
import java.io.PrintWriter;
import java.math.BigDecimal;
import java.sql.CallableStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Properties;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.ecommerce.DBConnection;

/**
* Servlet implementation class ProductDetails
*/
```

```
@WebServlet("/ProductDetails")
public class ProductDetails extends HttpServlet {
	private static final long serialVersionUID = 1L;

  /**
* @see HttpServlet#HttpServlet()
*/
  public ProductDetails() {
    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

        try {
            PrintWriter out = response.getWriter();
            out.println("<html><body>");

            InputStream in = getServletContext().getResourceAsStream("/WEB-INF/config.properties");
            Properties props = new Properties();
            props.load(in);

            DBConnection conn = new DBConnection(props.getProperty("url"), props.getProperty("userid"), props.getProperty("password"));
            CallableStatement stmt = conn.getConnection().prepareCall("{call add_product(?, ?)}");
```

```
                stmt.setString(1, "new product");
                stmt.setBigDecimal(2, new BigDecimal(1900.50));
                stmt.executeUpdate();

                out.println("Stored procedure has been executed.<Br>");
                stmt.close();



                out.println("</body></html>");
                conn.closeConnection();

            } catch (ClassNotFoundException e) {
                e.printStackTrace();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
            // TODO Auto-generated method stub
            doGet(request, response);
        }

}
```

5 Create, Select, and Drop a Database

```
<!DOCTYPE html>
```

```
<html>
<head>
<meta charset="UTF-8">
<title>JDBC Datbase Operations</title>
</head>
<body>
<a href="dboperations">Database Operations</a><br>

</body>
</html>
```

DBConnection

```
package com.ecommerce;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnection {

    private Connection connection;

    public DBConnection(String dbURL, String user, String pwd) throws ClassNotFoundException,
SQLException{

        Class.forName("com.mysql.jdbc.Driver");
        this.connection = DriverManager.getConnection(dbURL, user, pwd);
    }

    public Connection getConnection(){
        return this.connection;
    }
```

```
	public void closeConnection() throws SQLException {
		if (this.connection != null)
			this.connection.close();
	}
}
```

DBOperations

```
import java.io.IOException;
import java.io.InputStream;
import java.io.PrintWriter;
import java.math.BigDecimal;
import java.sql.CallableStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Properties;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.ecommerce.DBConnection;

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

```
   /**
* @see HttpServlet#HttpServlet()
*/
   public DBOperations() {
      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


            try {
                  PrintWriter out = response.getWriter();
                  out.println("<html><body>");


                  InputStream in = getServletContext().getResourceAsStream("/WEB-INF/config.properties");
                  Properties props = new Properties();
                  props.load(in);



                  DBConnection conn = new DBConnection(props.getProperty("url"),
props.getProperty("userid"), props.getProperty("password"));
                  Statement stmt = conn.getConnection().createStatement();
                  stmt.executeUpdate("create database mydatabase");
                  out.println("Created database: mydatabase<br>");
                  stmt.executeUpdate("use mydatabase");
                  out.println("Selected database: mydatabase<br>");
```

```
                stmt.executeUpdate("drop database mydatabase");
                stmt.close();
                out.println("Dropped database: mydatabase<br>");









                conn.closeConnection();



                out.println("</body></html>");
                conn.closeConnection();

            } catch (ClassNotFoundException e) {
                e.printStackTrace();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
            // TODO Auto-generated method stub
            doGet(request, response);
        }

}
```

6 Insertion, Updation, and Deletion of Database Records

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>JDBC Insert, Update, Delete</title>
</head>
<body>
<a href="list">Product Info</a><br>


</body>
</html>
```

DBConnection

```
package com.ecommerce;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnection {

    private Connection connection;

    public DBConnection(String dbURL, String user, String pwd) throws ClassNotFoundException,
SQLException{

        Class.forName("com.mysql.jdbc.Driver");
        this.connection = DriverManager.getConnection(dbURL, user, pwd);
    }
```

```
	public Connection getConnection(){
		return this.connection;
	}

	public void closeConnection() throws SQLException {
		if (this.connection != null)
			this.connection.close();
	}
}
```

ProductDetails

```
import java.io.IOException;
import java.io.InputStream;
import java.io.PrintWriter;
import java.math.BigDecimal;
import java.sql.CallableStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Properties;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.ecommerce.DBConnection;

/**
* Servlet implementation class ProductDetails
*/
```

```
@WebServlet("/ProductDetails")
public class ProductDetails extends HttpServlet {
	private static final long serialVersionUID = 1L;

  /**
* @see HttpServlet#HttpServlet()
*/
  public ProductDetails() {
    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

        try {
            PrintWriter out = response.getWriter();
            out.println("<html><body>");

            InputStream in = getServletContext().getResourceAsStream("/WEB-INF/config.properties");
            Properties props = new Properties();
            props.load(in);

            DBConnection conn = new DBConnection(props.getProperty("url"), props.getProperty("userid"), props.getProperty("password"));
```

```
            Statement stmt =
conn.getConnection().createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
            stmt.executeUpdate("insert into eproduct (name, price, date_added) values ('New
Product', 17800.00, now())");
            out.println("Executed an insert operation<br>");

            stmt.executeUpdate("update eproduct set price=2000 where name = 'New Product'");
            out.println("Executed an update operation<br>");

            stmt.executeUpdate("delete from eproduct where name = 'New Product'");
            out.println("Executed a delete operation<br>");

            stmt.close();

            conn.closeConnection();



            out.println("</body></html>");
            conn.closeConnection();

        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
```

```
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
		// TODO Auto-generated method stub
		doGet(request, response);
	}


}
```