4.Demonstrate stored procedures and exception handling in JDBC.

Index.html:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>JDBC Statements and Resultsets</title>
</head>
<body>
<a href="DemoJDBC">Product Info</a><br>
</body>
</html>
```

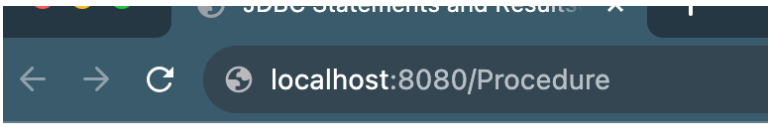
Servlet code

```
package Abc;

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;

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

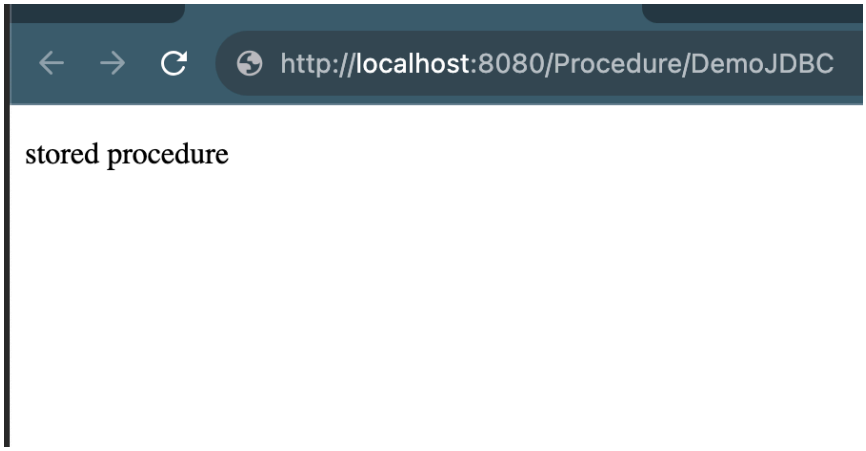
```
   protected void doGet(HttpServletRequest request, HttpServletResponse
response)
          throws ServletException, IOException {

       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();
       }
   }

}
```

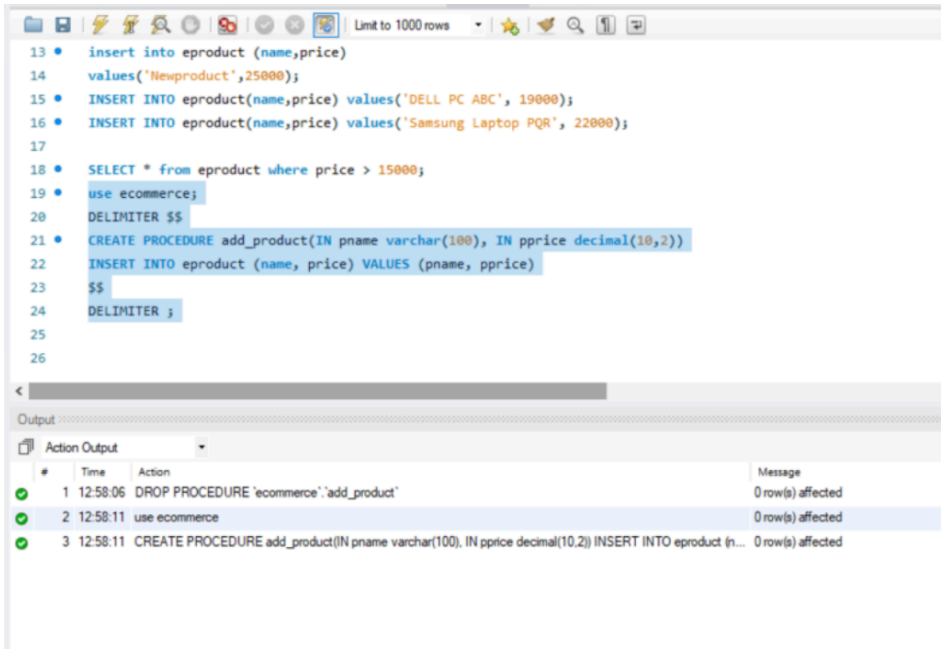
Database connection

```
package Abc;

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();
    }
}
```



Product Info


http://localhost:8080/Procedure/DemoJDBC
stored procedure


Limit to 1000 rows
13 insert into eproduct (name,price)
14 values('Newproduct',25000);
15 INSERT INTO eproduct(name,price) values('DELL PC ABC', 19000);
16 INSERT INTO eproduct(name,price) values('Samsung Laptop PQR', 22000);
17
18 SELECT * from eproduct where price > 15000;
19 use ecommerce;
20 DELIMITER $$
21 CREATE PROCEDURE add_product(IN pname varchar(100), IN pprice decimal(10,2))
22 INSERT INTO eproduct (name, price) VALUES (pname, pprice)
23 $$
24 DELIMITER ;
25
26
Output
Action Output
Time Action Message
1 12:58:06 DROP PROCEDURE `ecommerce`.`add_product` 0 row(s) affected
2 12:58:11 use ecommerce 0 row(s) affected
3 12:58:11 CREATE PROCEDURE add_product(IN pname varchar(100), IN pprice decimal(10,2)) INSERT INTO eproduct (n... 0 row(s) affected