



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## Experiment 3.3

**Name:** Kushal

**Branch:** BE-CSE

**Semester:** 6<sup>th</sup>

**Subject Name:** Java

**UID:** 21BCS5845

**Section/Group:** 620-A

**Date** 04/04/2024

**Subject Code:** 21CSH-319

**Aim:** Create an application for online auction using Servlet and JSP.

**Objective:** To learn about concept of Servlets in java and Apache Server.

### Steps:

Below is a simplified version of an online auction application:

- **User authentication:** Users can register, login, and logout.
- **Item listing:** Registered users can list items for auction.
- **Bidding functionality:** Registered users can place bids on listed items.

### Code:

**User.java** (POJO representing a user):

```
public class User {  
    private String username;  
    private String password;  
    // Constructor, getters, and setters  
}
```

**Item.java** (POJO representing an auction item):

```
public class Item {  
    private int id;  
    private String name;  
    private String description;  
    private double currentBid;  
    private String seller;  
    // Constructor, getters, and setters }  
}
```

**UserDAO.java** (Data Access Object for managing user-related operations):

```
import java.util.List;  
public interface UserDAO {
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.IOException;
import java.util.List;
public class ListItemsServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        // Fetch all items using ItemDAO
        // Forward to JSP to display items
    }
}
```

**BidServlet.java** (Servlet for placing bids):

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.IOException;
public class BidServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        int itemId = Integer.parseInt(request.getParameter("itemId"));
        double amount = Double.parseDouble(request.getParameter("amount"));
        String bidder = request.getParameter("bidder");
        // Place bid using ItemDAO
        // Redirect to appropriate page
    }
}
```

**login.jsp, list\_items.jsp, bid.jsp** (JSP files for user interface).



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## Output:

tion-jsp-servlet-jdbc-example/register

ng Boot Tutorial Java EE 7: Building... Creating a Blog Sys... JavaScript ES6 Class... Spring Boot 2 + An... A guide to prototyp.

### Employee Register Form

First Name

Last Name

UserName

Password

Address

Contact No

localhost:8080/registration-jsp-servlet-jdbc-example/register

Apps Java Platform, Enter... Spring Boot Tutorial Java EE 7: Building... Creating a Blog

## Employee successfully registered !

### Learning Outcome:

To solve using methods, to create a separate method for all input data.

To take input by user in run time and solve the problem.

Learn about get and post requests.