

Topic: Auction and Bidding system

Description:

A auctioning system with a backend database and a Web front-end.

Item registration, auctioning, searching and bidding (using servlets, JSP for the front-end, and EJBs for the middleware).

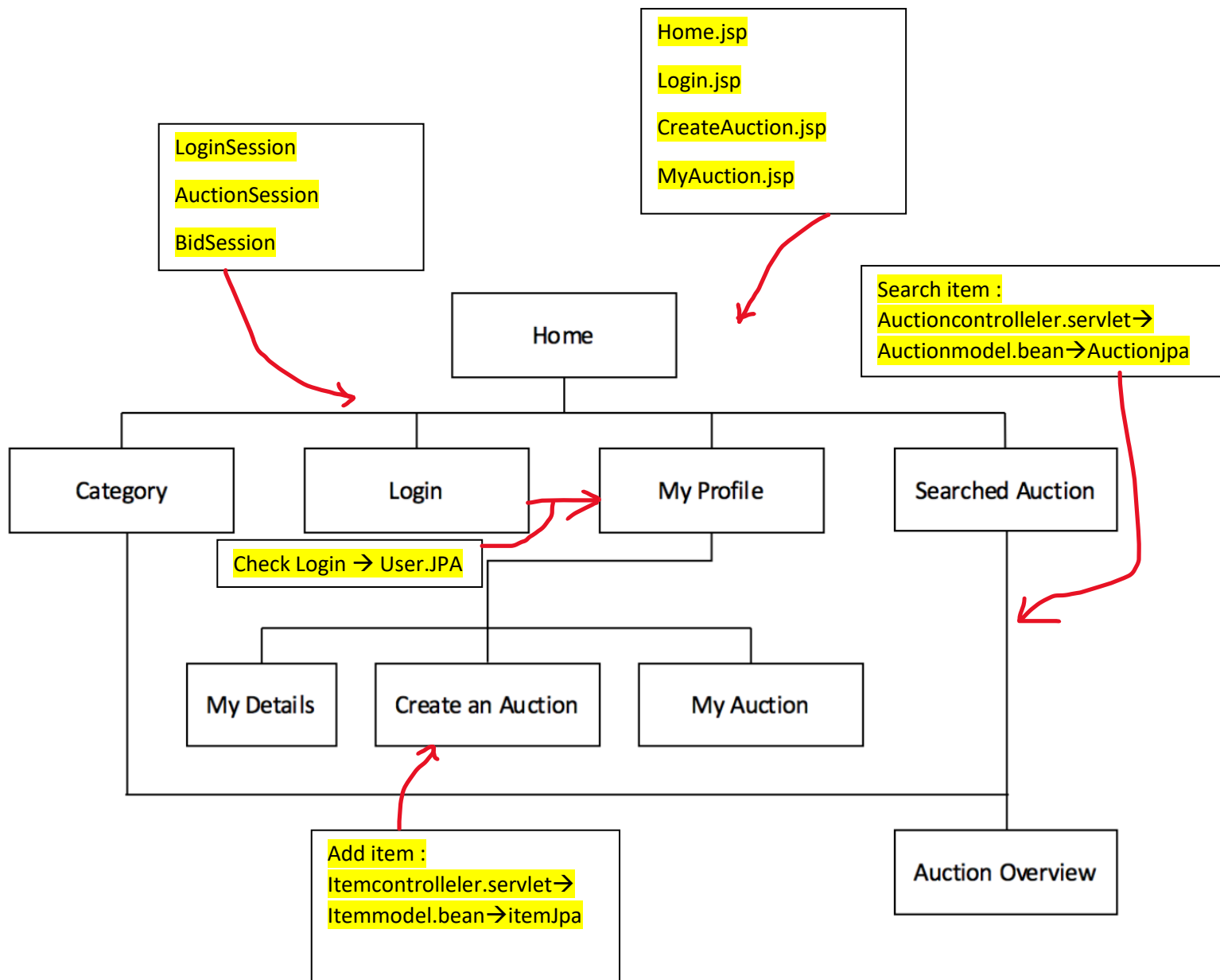
The software architecture implements is MVC (Model-View-Controller).

The user will have the ability to log into the system. The user will have the ability to create an auction on the web application with the description of the item and the start price for the auction. Once submitted, the web application will add the item to the database and automatically start the auction for that specific item. The user can search for auctions in the system The user can make a bid on an auction. The admin can remove an auction from the system.(Admin)

Technologies Used: The aim of this this project is to use the Java2EE framework to make a simple web site for Auction. under it the roof of J2EE several other technologies were entailed, like Glassfish for server application, EJB which is the abbreviation of Enterprise java Beans, Servlets and also JSP's, we have also resorted to JavaScript, CSS and Bootstrap to make a Good-looking web site, and since also we need a database, we used MYSQL database.

Bean state: Singleton session bean using JPA

Database Design: The database that is used is MySQL and has 4 tables for storing the description of auction items, personal information about the bidders, operational information about the auctions and historical information about the bids (i.e., USER, ITEM, AUCTION, BID).



CODE:

WAR Files:

Servlets

Auctioncontroller.java

```
package controller.auction;

import java.io.IOException;
import java.util.List;

import javax.ejb.EJB;
import javax.servlet.RequestDispatcher;
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 javax.servlet.http.HttpSession;

import model.auction.Auction;
import model.auction.AuctionItem;
import model.auction.AuctionModel;
import model.bid.BidHistory;
import model.bid.BidModel;
import model.category.Category;
import model.category.CategoryModel;
import model.user.User;
import webAuctionException.WebAuctionException;

/**
 * Servlet implementation class AuctionController
 */
@WebServlet(name = "AuctionController", urlPatterns = {
    "/AuctionController", "/MyAuction", "/RemoveAuction", "/SearchAuction", "/AuctionOverview", "/CreateAuction"})
public class AuctionController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    @EJB
    private AuctionModel model;

    @EJB
    private BidModel bidModel;

    @EJB
    private CategoryModel catModel;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public AuctionController() {
        super();
        // TODO Auto-generated constructor stub
    }

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

```

        */
        protected void doGet(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
            processRequest(request, response);
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse
         *      response)
         */
        protected void doPost(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
            processRequest(request, response);
        }

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

            String path = request.getServletPath();

            if (path.equals("/MyAuction")) {

                HttpSession session = request.getSession(false);
                String userid = (String)
session.getAttribute("userid");

                try {
                    Category[] cat = catModel.getCategoryList();
                    request.setAttribute("category", cat);
                    AuctionItem[] auction =
model.getUserAuction(userid);
                    request.setAttribute("auction", auction);
                    RequestDispatcher dispatcher =
request.getRequestDispatcher("myAuction.jsp");
                    dispatcher.forward(request, response);
                } catch (WebAuctionException ex) {

                    request.setAttribute("message", ex.getMessage());
                    RequestDispatcher dispatcher =
request.getRequestDispatcher("indexItemController");
                    dispatcher.forward(request, response);
                };
            } else if (path.equals("/RemoveAuction")) {
                String id = request.getParameter("id");
                System.out.println("BBBBBBBBBBBBB: " + id);
                try {
                    model.deleteAuction(id);
                    RequestDispatcher dispatcher =
request.getRequestDispatcher("MyAuction");
                    dispatcher.forward(request, response);
                } catch (WebAuctionException e) {
                    e.printStackTrace();
                }
            } else if (path.equals("/SearchAuction")) {
                String name = request.getParameter("name");
                System.out.println("XXXXXXXXXX: " + name);
            }
        }
    }
}

```

```

        try {
            Category[] cat = catModel.getCategoryList();
            request.setAttribute("category", cat);
            AuctionItem[] auction =
model.searchAuction(name);
            request.setAttribute("auction", auction);
            RequestDispatcher dispatcher =
request.getRequestDispatcher("searchedAuctionList.jsp");
            dispatcher.forward(request, response);
        } catch (WebAuctionException e) {
            e.printStackTrace();
        }
    }else if(path.equals("/AuctionOverview")){
        String id = request.getParameter("id");
        try {
            Category[] cat = catModel.getCategoryList();
            request.setAttribute("category", cat);
            AuctionItem[] auction =
model.getSearchedAuction(id);
            BidHistory[] bidHistory =
bidModel.getBidHistory(id);
            request.setAttribute("auction", auction);
            request.setAttribute("bidHistory", bidHistory);
            RequestDispatcher dispatcher =
request.getRequestDispatcher("auctionOverview.jsp");
            dispatcher.forward(request, response);
        } catch (WebAuctionException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }else if(path.equals("/CreateAuction")){
        try {
            Category[] cat = catModel.getCategoryList();
            request.setAttribute("category", cat);
            RequestDispatcher dispatcher =
request.getRequestDispatcher("createAuction.jsp");
            dispatcher.forward(request, response);
        } catch (WebAuctionException e) {
            e.printStackTrace();
        }
    }
}
}
}

```

Bidcontroller.java

```

package controller.bid;

import java.io.IOException;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;

import javax.ejb.EJB;
import javax.servlet.RequestDispatcher;
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 javax.servlet.http.HttpSession;

import model.auction.Auction;
import model.auction.AuctionModel;
import model.bid.Bid;
import model.bid.BidModel;
import webAuctionException.WebAuctionException;

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

    @EJB
    private BidModel bidModel;

    @EJB
    private AuctionModel auctionModel;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public BidController() {
        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
        processRequest(request, response);
    }

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

    /**
     * TO BE WRITTEN BY THE STUDENTS
     */
    protected void processRequest(HttpServletRequest request,
    HttpServletResponse response)
        throws ServletException, IOException {

```

```

        String path = request.getServletPath();

        HttpSession session = request.getSession(false);

        if(path.equals("/PlaceBid")){
            if(session.getAttribute("userid") == null){
                RequestDispatcher dispatcher =
request.getRequestDispatcher("LoginBuffer");
                dispatcher.forward(request, response);
            }else{
                try {

                    String id =
request.getParameter("id");
                    Auction auc =
auctionModel.getAuction(id);

                    String bid =
request.getParameter("bid");
                    String webBid =
request.getParameter("webBid");
                    String dbBid = auc.getCurrentbid();

                    System.out.println("Wesbite Bid
Amount: " + webBid);
                    System.out.println("Database Bid
Amount: " + dbBid);
                    System.out.println("Your Proposed
Bid: " + bid);

                    double IntBid =
Double.parseDouble(bid);
                    double IntWebBid =
Double.parseDouble(dbBid);
                    double IntDbBid =
Double.parseDouble(webBid);

                    if(IntDbBid != IntWebBid){

                        System.out.println("UNSYCHRONISED!!!!!!");
                        request.setAttribute("message",
"A user has just bidded. Bid quick!");
                    }else if (IntDbBid == IntWebBid &&
IntWebBid < IntBid){

                        String userid = (String)
session.getAttribute("userid");

                        DateFormat df = new
SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
                        Date today =
Calendar.getInstance().getTime();
                        String reportDate =
df.format(today);

                        bidModel.addBid(new
Bid("0",id,userid,bid,reportDate));

```



```

    @EJB private CategoryModel catModel;

    @EJB
    private AuctionModel model;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public CategoryController() {
        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
        processRequest(request, response);
    }

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

    /**
     * TO BE WRITTEN BY THE STUDENTS
     */
    protected void processRequest(HttpServletRequest request,
    HttpServletResponse response)
        throws ServletException, IOException {

        String path = request.getServletPath();
        HttpSession session = request.getSession();

        if(path.equals("/indexItemController")){
            try {
                Category[] cat =
catModel.getCategoryList();
                request.setAttribute("category", cat);
                RequestDispatcher dispatcher =
request.getRequestDispatcher("home.jsp");
                dispatcher.forward(request, response);
            } catch (WebAuctionException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

        }else if(path.equals("/CategoryController")){

```

```

        String catid = request.getParameter("catid");
        System.out.println("DDDDDDDD: " + catid);
        try {

            String sess = "";

            if(session == null){
                sess="AAAAAAAAAA: NO SESSION";
            }else{
                sess="BBBBBBBBB: YESSSS SESSION";
            }

            Category[] cat = catModel.getCategoryList();
            request.setAttribute("category", cat);
            AuctionItem[] auction =
model.getCategorisedAuction(catid);
            request.setAttribute("auction", auction);
            request.setAttribute("sess", sess);
            RequestDispatcher dispatcher =
request.getRequestDispatcher("categorisedAuction.jsp");
            dispatcher.forward(request, response);
        } catch (WebAuctionException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

    }

}

}

}

```

Home.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@page import="controller.category.*"%>
<%@page import="model.category.*"%>
<!DOCTYPE html>
<html lang="en">
<head>
    <title>APA Web Auction</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.c
ss">
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></
script>

```

```

<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
></script>
</head>
<body>
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header ">
      <a class="navbar-brand" href="index.jsp">APA Web Auction</a>
    </div>
    <ul class="nav navbar-nav">
      <li class="active"><a href="indexItemController">Home</a></li>
      <li class="dropdown"><a class="dropdown-toggle" data-
toggle="dropdown" href="#">Category <span class="caret"></span></a>
      <ul class="dropdown-menu">
        <% Category[] cat = (Category[])
request.getAttribute("category"); for (int i = 0; i < cat.length; i++) {
Category a = cat[i]; %>
          <li><a
href="CategoryController?catid=<%out.print(a.getCatid());%>"><%
out.print(a.getCatName()); %></a></li>
          <% } %>
        </ul>
      </li>

    </ul>
    <div class="col-sm-3 col-md-3 pull-right">
      <form class="navbar-form" role="search" action="SearchAuction"
method="GET">
        <div class="input-group">
          <input type="text" class="form-control" placeholder="Search"
name="name" >
          <div class="input-group-btn">
            <button class="btn btn-default" type="submit"><i
class="glyphicon glyphicon-search"></i></button>
          </div>
        </div>
      </form>
    </div>
    <ul class="nav navbar-nav navbar-right">
      <% String userlayout = "";
      if(session.getAttribute("userid") == null) {
        userlayout = "<li><a href='LoginBuffer'><span
class='glyphicon glyphicon-log-in'></span> Login</a></li>";
      }else{
        userlayout = "<li class='dropdown'><a class='dropdown-
toggle' data-toggle='dropdown' href='MyDetails'>My Profile <span
class='caret'></span></a><ul class='dropdown-menu'><li><a
href='MyDetails'>My Details</a></li><li><a href='CreateAuction'>Create an
Auction</a></li><li><a href='MyAuction'>My Auctions</a></li><li><a
href='LogOut'>Log Out</a></li></ul></li>";
      }
      %>

      <%out.print(userlayout); %>

    </ul>

  </div>
</nav>

```

```

<div class="container">
  <h3>Welcome to APA Web Auction!</h3>
  <p>Browse many auctions based on it's category and searched criteria.
Remember, you must be logged in to start your own auction and to bid.</p>
</div>

</body>
</html>

```

login.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@page import="controller.category.*"%>
<%@page import="model.category.*"%>
<!DOCTYPE html>
<html lang="en">
<head>
  <title>APA Web Auction</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.c
ss">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></
script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
></script>
  <link href="styles/signin.css" rel="stylesheet">
</head>
<body>
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header ">
      <a class="navbar-brand" href="index.jsp">APA Web Auction</a>
    </div>
    <ul class="nav navbar-nav">
      <li class="active"><a href="indexItemController">Home</a></li>
      <li class="dropdown"><a class="dropdown-toggle" data-
toggle="dropdown" href="#">Category <span class="caret"></span></a>
      <ul class="dropdown-menu">
        <% Category[] cat = (Category[])
request.getAttribute("category"); for (int i = 0; i < cat.length; i++) {
Category a = cat[i]; %>
          <li><a
href="CategoryController?catid=<%out.print(a.getCatid());%>"><%
out.print(a.getCatName()); %></a></li>
          <% } %>
        </ul>
      </li>

    </ul>
  <div class="col-sm-3 col-md-3 pull-right">

```

```

        <form class="navbar-form" role="search" action="SearchAuction"
method="GET">
        <div class="input-group">
            <input type="text" class="form-control" placeholder="Search"
name="name" >
            <div class="input-group-btn">
                <button class="btn btn-default" type="submit"><i
class="glyphicon glyphicon-search"></i></button>
            </div>
        </div>
    </form>
</div>
<ul class="nav navbar-nav navbar-right">
    <% String userlayout = "";
        if(session.getAttribute("userid") == null) {
            userlayout = "<li><a href='LoginBuffer'><span
class='glyphicon glyphicon-log-in'></span> Login</a></li>";
        }else{
            userlayout = "<li class='dropdown'><a class='dropdown-
toggle' data-toggle='dropdown' href='MyDetails'>My Profile <span
class='caret'></span></a><ul class='dropdown-menu'><li><a
href='MyDetails'>My Details</a></li><li><a href='CreateAuction'>Create an
Auction</a></li><li><a href='MyAuction'>My Auctions</a></li><li><a
href='LogOut'>Log Out</a></li></ul></li>";
        }
    <%>

        <%out.print(userlayout); %>

    </ul>

</div>
</nav>

<div class="container">

<center>

    <%
        String message = (String) request.getAttribute("message");
        if (message != null) {
            out.println("<h3 align='center' class='text-danger'>" +
message + "</h3>");
        }
    <%>

</center>

</div>

<div class="container">
    <form class="form-signin" action="LoginController" method="POST">
        <h2 class="form-signin-heading" align="center">Log in</h2>
        <p align="center" class="text-warning">Note: You must be logged
in before you can place a bid on any auction.</p>
        <label for="input" class="sr-only">Username</label>
        <input type="text" name="username" class="form-control"
placeholder="Username" required="true" autofocus="">
        <label for="inputPassword" class="sr-only">Password</label>
        <input type="password" name="password" class="form-control"
placeholder="Password" required="true">
        <button class="btn btn-lg btn-primary btn-block"
type="submit">Sign in</button>
    </form>

```



```

        <button class="btn btn-default" type="submit"><i
class="glyphicon glyphicon-search"></i></button>
    </div>
</div>
</form>
</div>
<ul class="nav navbar-nav navbar-right">
    <% String userlayout = "";
        if(session.getAttribute("userid") == null) {
            userlayout = "<li><a href='LoginBuffer'><span
class='glyphicon glyphicon-log-in'></span> Login</a></li>";
        }else{
            userlayout = "<li class='dropdown'><a class='dropdown-
toggle' data-toggle='dropdown' href='MyDetails'>My Profile <span
class='caret'></span></a><ul class='dropdown-menu'><li><a
href='MyDetails'>My Details</a></li><li><a href='CreateAuction'>Create an
Auction</a></li><li><a href='MyAuction'>My Auctions</a></li><li><a
href='LogOut'>Log Out</a></li></ul></li>";
        }
    <%>

    <%out.print(userlayout); %>

</ul>

</div>
</nav>

<div class="container">
    <h2>My Auctions</h2>
    <p>Below are your auctions that are currently live.</p>
    <table class="table table-hover">
        <thead>
            <tr>
                <th>Item Id</th>
                <th>Auction Id</th>
                <th>Item Name</th>
                <th>Category</th>
                <th>Start Price</th>
                <th>Current Bid</th>
                <th>Action</th>
            </tr>
        </thead>
        <tbody>
            <% AuctionItem[] auc = (AuctionItem[])
request.getAttribute("auction"); for (int i = 0; i < auc.length; i++) {
AuctionItem a = auc[i]; %>
            <tr>
                <td><% out.print(a.getItemid()); %></td>
                <td><% out.print(a.getAuctionid()); %></td>
                <td><% out.print(a.getItemname()); %></td>
                <td><% out.print(a.getCatName()); %></td>
                <td><% out.print(a.getStartprice()); %></td>
                <td><% out.print(a.getCurrentbid()); %></td>
                <td><form action="RemoveAuction" method="post"> <input
type="hidden" type="text" name="id" value=<% out.print(a.getItemid()); %>
/> <input type="submit" value="Remove"></form></td>
            </tr>
        </tbody>
        <!-- close the loop -->
    <% } %>

```

```
<!-- close table -->
</table>
</div>
```

```
</body>
</html>
```

EJB files:

Singleton sessionbeans

Auctionmodelimpl.java

```
package model.auction;

import java.util.ArrayList;
import java.util.List;

import javax.ejb.EJB;
import javax.ejb.Local;
import javax.ejb.Singleton;
import javax.persistence.EntityExistsException;
import javax.persistence.EntityManager;
import javax.persistence.OptimisticLockException;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;

import model.user.UserModel;
import webAuctionException.WebAuctionException;

@Local @Singleton
public class AuctionModelImpl implements AuctionModel{

    @PersistenceContext
    private EntityManager emgr;

    public AuctionModelImpl() {

    }

    @Override
    public void addAuction(Auction auction) throws WebAuctionException {
        System.out.println(auction);
        try {
            emgr.persist(auction);
        } catch (EntityExistsException ex) {
            throw new WebAuctionException("Duplicate Id : ");
        }
    }
}
```



```

    }

    }

    @Override
    public Auction getAuction(String id) throws WebAuctionException {
        Auction auc = emgr.find(Auction.class, id);

        if (auc != null) {
            emgr.refresh(auc);
            return auc;
        } else {
            throw new WebAuctionException("Auction : " + id + " not
found");
        }
    }

    @Override
    public void updateAuction(String id, String currentbid) throws
WebAuctionException {
        int query = emgr.createNativeQuery("UPDATE Auction SET
currentBid='" + currentbid + "' WHERE auctionId='" + id +
"'").executeUpdate();
        System.out.println(query);
    }

    @SuppressWarnings("unchecked")
    public AuctionItem[] getUserAuction(String userid) throws
WebAuctionException {
        /* REPLACE THE LIST OF STUDENTMARK WITH THE RESULT OF SQL QUERY */
        Query query = emgr.createNativeQuery("SELECT Item.itemid,
Auction.auctionId, Item.itemname, Category.catName,
Auction.startprice,Auction.currentbid FROM Item INNER JOIN Category on
Item.catid = Category.catid INNER JOIN Auction on Item.itemid =
Auction.auctionId where Item.sellerid = '" + userid + "'ORDER BY
Auction.auctionId DESC",AuctionItem.class);
        List<AuctionItem> auctions = new ArrayList<AuctionItem>();
        auctions = (List<AuctionItem>) query.getResultList();
        System.out.println("PPPPPPPPPPPP: " + auctions);
        return (AuctionItem[]) auctions.toArray(new AuctionItem[0]);
    }

    public void deleteAuction(String itemid) throws WebAuctionException {
        /* REPLACE DEL OPERATION WITH JPA VERSION*/
        int query = emgr.createNativeQuery("DELETE FROM auction WHERE
`auction`.`itemid` = '" + itemid + "'").executeUpdate();
        System.out.println("AAAAAA: " + query);
    }

    @SuppressWarnings("unchecked")
    public AuctionItem[] searchAuction(String name) throws
WebAuctionException {
        /* REPLACE THE LIST OF STUDENTMARK WITH THE RESULT OF SQL QUERY */
        Query query = emgr.createNativeQuery("SELECT Item.itemid,
Auction.auctionId, Item.itemname, Category.catName,
Auction.startprice,Auction.currentbid FROM Item INNER JOIN Category on
Item.catid = Category.catid INNER JOIN Auction on Item.itemid =

```

```

Auction.auctionId where Item.itemname LIKE
'%" + name + "%', AuctionItem.class);
        List<AuctionItem> auctions = new ArrayList<AuctionItem>();
        auctions = (List<AuctionItem>) query.getResultList();
        System.out.println("PPPPPPPPPPPP: " + auctions);
        return (AuctionItem[]) auctions.toArray(new AuctionItem[0]);
    }

    @SuppressWarnings("unchecked")
    public AuctionItem[] getSearchedAuction(String id) throws
WebAuctionException {
        /* REPLACE THE LIST OF STUDENTMARK WITH THE RESULT OF SQL QUERY */
        Query query = emgr.createNativeQuery("SELECT Item.itemid,
Auction.auctionId, Item.itemname, Category.catName,
Auction.startprice,Auction.currentbid FROM Item INNER JOIN Category on
Item.catid = Category.catid INNER JOIN Auction on Item.itemid =
Auction.auctionId where Item.itemid ='" + id + "'",AuctionItem.class);
        List<AuctionItem> auctions = new ArrayList<AuctionItem>();
        auctions = (List<AuctionItem>) query.getResultList();
        System.out.println("PPPPPPPPPPPP: " + auctions);
        return (AuctionItem[]) auctions.toArray(new AuctionItem[0]);
    }

    @SuppressWarnings("unchecked")
    public AuctionItem[] getCategorisedAuction(String id) throws
WebAuctionException {
        /* REPLACE THE LIST OF STUDENTMARK WITH THE RESULT OF SQL QUERY */
        Query query = emgr.createNativeQuery("SELECT Item.itemid,
Auction.auctionId, Item.itemname, Category.catName,
Auction.startprice,Auction.currentbid FROM Item INNER JOIN Category on
Item.catid = Category.catid INNER JOIN Auction on Item.itemid =
Auction.auctionId where Item.catid ='" + id + "'",AuctionItem.class);
        List<AuctionItem> auctions = new ArrayList<AuctionItem>();
        auctions = (List<AuctionItem>) query.getResultList();
        System.out.println("PPPPPPPPPPPP: " + auctions);
        return (AuctionItem[]) auctions.toArray(new AuctionItem[0]);
    }
}

```

Itemmodelimpl.java

```

package model.item;

import java.util.ArrayList;
import java.util.List;

import javax.ejb.Local;
import javax.ejb.Singleton;
import javax.persistence.EntityExistsException;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;

import model.auction.Auction;
import webAuctionException.WebAuctionException;

@Local @Singleton

```

```

public class ItemModelImpl implements ItemModel {

    @PersistenceContext
    private EntityManager emgr;

    public ItemModelImpl() {

    }

    @Override
    public void addItem(Item item) throws WebAuctionException {
        System.out.println(item);
        System.out.println(" ----- " + emgr);
        try {
            emgr.persist(item);
        } catch (EntityExistsException ex) {
            throw new WebAuctionException("Duplicate Id : ");
        }
    }

    @SuppressWarnings("unchecked")
    public Item[] getLastUserItemId(String userid) throws WebAuctionException
    {
        /* REPLACE THE LIST OF STUDENTMARK WITH THE RESULT OF SQL QUERY */
        Query query = emgr.createNativeQuery("SELECT itemid FROM Item Where
sellerid = '"+userid+"' ORDER BY itemid DESC LIMIT 0,1", Item.class);
        List<Item> items = new ArrayList<Item>();
        items = (List<Item>) query.getResultList();
        System.out.println("PPPPPPPPPPPP: " + items);
        if (items.size() == 0) {
            throw new WebAuctionException("You have no items.");
        } else {
            return (Item[]) items.toArray(new Item[0]);
        }
    }
}

```

Usermodelimpl.java

```

package model.user;

import java.util.ArrayList;
import java.util.List;

import javax.ejb.Local;
import javax.ejb.Singleton;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;

import webAuctionException.WebAuctionException;

@Local @Singleton
public class UserModelImpl implements UserModel {

```

```

@PersistenceContext
private EntityManager emgr;

    private static UserModel instance = new UserModelImpl();

    public static UserModel getInstance() {
        return instance;
    }

public UserModelImpl() {

}

    @SuppressWarnings("unchecked")
    public User[] checkLogin(String username, String password)
throws WebAuctionException {
        System.out.println(" ---- " + emgr);
        System.out.println("A: " + username + " : " + password);
        Query query = emgr.createNativeQuery("SELECT * FROM User
WHERE username = '"+username+"' AND password = '"+password+"'",
        User.class);
        List<User> user = new ArrayList<User>();
        user = (List<User>) query.getResultList();
        System.out.println("PPPPPPPPPPPP: " + user);
        if (user.size() == 0) {
            throw new WebAuctionException("Invalid
Credential");
        } else {
            return (User[]) user.toArray(new User[0]);
        }
    }

    @SuppressWarnings("unchecked")
    public User[] getUserDetails(String id) throws
WebAuctionException {
        Query query = emgr.createNativeQuery("SELECT * FROM User
WHERE userid = '"+id+"'", User.class);
        List<User> user = new ArrayList<User>();
        user = (List<User>) query.getResultList();
        System.out.println("PPPPPPPPPPPP: " + user);
        if (user.size() == 0) {
            throw new WebAuctionException("Invalid
Credential");
        } else {
            return (User[]) user.toArray(new User[0]);
        }
    }
}

```

Output:

APA Web Auction

Home

Category ▾

🔑 Login

Search

🔍

Vehicle

Mobile Phone

Clothing

Toys

Web Auction!

Browse by category and searched criteria. Remember, you must be logged in to start your own auction and to bid.

APA Web Auction

Home

Category ▾

🔑 Login

Search

🔍

Category: Vehicle

Item Id	Auction Id	Item Name	Category	Start Price	Current Bid	Action
1	1	BMW 116i	Vehicle	3500.0	3510.0	<button>Open</button>
2	2	Audi A3	Vehicle	5000.0	5550.0	<button>Open</button>
10	10	Jaguar XJ	Vehicle	35000.0	35001.0	<button>Open</button>
11	11	Ford Focus	Vehicle	1200.0	1200.0	<button>Open</button>
21	21	Rain coat	Vehicle	1000.0	1000.0	<button>Open</button>
28	28	Tesla car	Vehicle	1500.0	1500.0	<button>Open</button>
29	29	Tesla car	Vehicle	1.0E7	1.0E7	<button>Open</button>



Category: Mobile Phone

Item Id	Auction Id	Item Name	Category	Start Price	Current Bid	Action
3	3	Iphone 6	Mobile Phone	500.0	510.0	<button>Open</button>
4	4	Iphone 7s	Mobile Phone	850.0	850.0	<button>Open</button>
12	12	Samsung Galaxy S5	Mobile Phone	600.0	600.0	<button>Open</button>
13	13	Blackberry	Mobile Phone	300.0	510.0	<button>Open</button>
14	14	Google Pixel	Mobile Phone	700.0	700.0	<button>Open</button>

Category: Clothing

Item Id	Auction Id	Item Name	Category	Start Price	Current Bid	Action
15	15	Primark Vest	Clothing	4.0	4.0	<button>Open</button>
16	16	Gucci Belt	Clothing	50.0	50.0	<button>Open</button>
17	17	Winter Coat	Clothing	20.0	20.0	<button>Open</button>
24	24	Jump suit	Clothing	555.0	555.0	<button>Open</button>
25	25	Rain coat	Clothing	3.0E7	3.0E7	<button>Open</button>

Category: Toys

Item Id	Auction Id	Item Name	Category	Start Price	Current Bid	Action
19	19	RC Car	Toys	70.0	70.0	<button>Open</button>
20	20	Mini Pram	Toys	12.0	12.0	<button>Open</button>
30	30	robot	Toys	900.0	900.0	<button>Open</button>

Log in

Note: You must be logged in before you can place a bid on any auction.

prasanna25

Sign in

Welcome to APA Web Auction!

Browse many auctions based on it's category and searched criteria. Remember, you must be logged in to start your own auction s

- My Details
- Create an Auction
- My Auctions
- Log Out

My Auctions

Below are your auctions that are currently live.

Item Id	Auction Id	Item Name	Category	Start Price	Current Bid	Action
29	29	Tesla car	Vehicle	1.0E7	1.0E7	<button>Remove</button>
28	28	Tesla car	Vehicle	1500.0	1500.0	<button>Remove</button>
25	25	Rain coat	Clothing	3.0E7	3.0E7	<button>Remove</button>
24	24	Jump suit	Clothing	555.0	555.0	<button>Remove</button>
21	21	Rain coat	Vehicle	1000.0	1000.0	<button>Remove</button>
4	4	Iphone 7s	Mobile Phone	850.0	850.0	<button>Remove</button>
3	3	Iphone 6	Mobile Phone	500.0	510.0	<button>Remove</button>
2	2	Audi A3	Vehicle	5000.0	5550.0	<button>Remove</button>
1	1	BMW 116i	Vehicle	3500.0	3510.0	<button>Remove</button>

Create Auction

Fill out the form and click start to begin the auction.

Item Name

Rain coat

Category

Clothing

Start Price

900

Start Auction

My Details

ID: 1

Name: Prasannapathi S

Username: prasanna25

phpMyAdmin

Recent Favorites

- New
- cat2
- crud_je
- employee
- hotel_reservation_system
 - New
 - reservation
 - room
- information_schema
- jcaoqs
- mysql
- oqs
- performance_schema
- phpmyadmin
- quiz
- sample
- sample1
- schooladmission
- test
- webauction
 - New
 - auction
 - bid
 - category
 - item
 - user

Server: 127.0.0.1 » Database: webauction » Table: auction

Browse Structure SQL Search Insert Export Import Privileges Operations

		auctionId	itemId	startPrice	currentBid
<input type="checkbox"/>	Edit Copy Delete	1	1	3500	3510
<input type="checkbox"/>	Edit Copy Delete	2	2	5000	5550
<input type="checkbox"/>	Edit Copy Delete	3	3	500	510
<input type="checkbox"/>	Edit Copy Delete	4	4	850	850
<input type="checkbox"/>	Edit Copy Delete	10	10	35000	35001
<input type="checkbox"/>	Edit Copy Delete	11	11	1200	1200
<input type="checkbox"/>	Edit Copy Delete	12	12	600	600
<input type="checkbox"/>	Edit Copy Delete	13	13	300	510
<input type="checkbox"/>	Edit Copy Delete	14	14	700	700
<input type="checkbox"/>	Edit Copy Delete	15	15	4	4
<input type="checkbox"/>	Edit Copy Delete	16	16	50	50
<input type="checkbox"/>	Edit Copy Delete	17	17	20	20
<input type="checkbox"/>	Edit Copy Delete	19	19	70	70
<input type="checkbox"/>	Edit Copy Delete	20	20	12	12
<input type="checkbox"/>	Edit Copy Delete	21	23	1000	1000
<input type="checkbox"/>	Edit Copy Delete	24	26	555	555
<input type="checkbox"/>	Edit Copy Delete	25	27	30000000	30000000
<input type="checkbox"/>	Edit Copy Delete	29	31	10000000	10000000
<input type="checkbox"/>	Edit Copy Delete	30	32	900	900

phpMyAdmin

Recent Favorites

- New
- cat2
- crud_je
- employee
- hotel_reservation_system
 - New
 - reservation
 - room
- information_schema
- jcaoqs
- mysql
- oqs
- performance_schema
- phpmyadmin
- quiz
- sample
- sample1
- schooladmission
- test
- webauction
 - New
 - auction

Server: 127.0.0.1 » Database: webauction » Table: bid

Browse Structure SQL Search Insert Export Import Privileges Operations

Showing rows 0 - 5 (6 total, Query took 0.0008 seconds.)

SELECT * FROM `bid`

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

		bidId	auctionId	bidderId	bidAmount	time
<input type="checkbox"/>	Edit Copy Delete	1	3	3	510	13/11/2022 20:18:36
<input type="checkbox"/>	Edit Copy Delete	2	1	3	3510	14/11/2022 21:33:42
<input type="checkbox"/>	Edit Copy Delete	3	2	3	5550	14/11/2022 21:41:13
<input type="checkbox"/>	Edit Copy Delete	4	13	3	310	14/11/2022 22:04:28
<input type="checkbox"/>	Edit Copy Delete	5	13	3	510	14/11/2022 22:23:41
<input type="checkbox"/>	Edit Copy Delete	6	10	1	35001	14/11/2022 23:08:46

☐ Check all | With selected: Edit Copy Delete Export

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Recent Favorites

New
cat2
crud_je
employee
hotel_reservation_system
New
reservation
room
information_schema
jcaoqs
mysql
oqs
performance_schema
phpmyadmin
quiz
sample
sample1
schooladmission
test

Showing rows 0 - 3 (4 total, Query took 0.0006 seconds.)

`SELECT * FROM `category``

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

	catid	catName
<input type="checkbox"/> Edit Copy Delete	1	Vehicle
<input type="checkbox"/> Edit Copy Delete	2	Mobile Phone
<input type="checkbox"/> Edit Copy Delete	3	Clothing
<input type="checkbox"/> Edit Copy Delete	4	Toys

☐ Check all With selected: Edit Copy Delete

☐ Show all | Number of rows: 25 | Filter rows: Search this table

phpMyAdmin

Recent Favorites

New
cat2
crud_je
employee
hotel_reservation_system
New
reservation
room
information_schema
jcaoqs
mysql
oqs
performance_schema
phpmyadmin
quiz
sample
sample1
schooladmission
test
webauction
New
auction
bid
category
item
user

Server: 127.0.0.1 » Database: webauction » Table: item

Browse Structure SQL Search Insert Export Import Privileges

Showing rows 0 - 24 (32 total, Query took 0.0004 seconds.)

`SELECT * FROM `item``

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 > >> | ☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

	itemid	itemname	catid	sellerid
<input type="checkbox"/> Edit Copy Delete	1	BMW 116i	1	1
<input type="checkbox"/> Edit Copy Delete	2	Audi A3	1	1
<input type="checkbox"/> Edit Copy Delete	3	Iphone 6	2	1
<input type="checkbox"/> Edit Copy Delete	4	Iphone 7s	2	1
<input type="checkbox"/> Edit Copy Delete	5	Plain T-Shirt	3	1
<input type="checkbox"/> Edit Copy Delete	6	Black Boots	3	1
<input type="checkbox"/> Edit Copy Delete	7	Yellow Flat Cap	3	1
<input type="checkbox"/> Edit Copy Delete	8	Lego Bricks	4	1
<input type="checkbox"/> Edit Copy Delete	9	Fireman Sam Doll	4	1
<input type="checkbox"/> Edit Copy Delete	10	Jaguar XJ	1	2
<input type="checkbox"/> Edit Copy Delete	11	Ford Focus	1	2
<input type="checkbox"/> Edit Copy Delete	12	Samsung Galaxy S5	2	2
<input type="checkbox"/> Edit Copy Delete	13	Blackberry	2	2

Console

phpMyAdmin

Recent

Favorites

New

cat2

crud_je

employee

hotel_reservation_system

New

reservation

room

information_schema

jaoqs

mysql

oqs

performance_schema

phpmyadmin

quiz

sample

sample1

schooladmission

test

webauction

New

auction

bid

category

item

user

Server: 127.0.0.1 » Database: webauction » Table: user

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Operations

T

Showing rows 0 - 2 (3 total, Query took 0.0007 seconds.)

SELECT * FROM `user`

Profiling

Edit inline

Edit

Explain SQL

Create PHP code

Refresh

Show all

Number of rows: 25

Filter rows: Search this table

Sort by key: None

Extra options

userid

name

username

password

☐

Edit

Copy

Delete

1

Prasannapathi S

prasanna25

prasanna25

☐

Edit

Copy

Delete

2

Adhava Perumal S

adhava03

adhava03

☐

Edit

Copy

Delete

3

Avinash R

avinash06

avinash06

Check all

With selected:

Edit

Copy

Delete

Export

Show all

Number of rows: 25

Filter rows: Search this table

Sort by key: None

Query results operations

Print

Copy to clipboard

Export

Display chart

Create view

Bookmark this SQL query

Label:

☐ Let every user access this bookmark