

```
package classes;
```

```
/**  
 * @author Jordan Williamson & Michelle Pham  
 */
```

```
import java.io.IOException;
```

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

```
//  
//Servlet implementation class AllClientServlet  
//  
@WebServlet("/AllClientServlet")  
public class AllClientServlet extends HttpServlet  
{  
    private static final long serialVersionUID = -7419930036249986454L;  
  
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,  
IOException  
    {  
        DatabaseCon dbQuery = new DatabaseCon();  
        AllClientViewer clients = dbQuery.getAllClients();  
        request.getSession().setAttribute("clientList", clients.AllClient);  
        response.sendRedirect("allclientinfo.jsp");  
    }  
  
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,  
IOException  
    {  
        doGet(request, response);  
    }  
}
```

```
package classes;
```

```
/**  
 * @author Jordan Williamson & Michelle Pham  
 *  
 */
```

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

```
// We will use this list to display all client info in UI.
```

```
public class AllClientViewer {  
  
    List <ClientInformation>AllClient = new ArrayList<ClientInformation>();
```

```
}
```

```
package classes;
```

```
/**
```

```
 * @author Jordan Williamson & Michelle Pham
```

```
 Display values in UI using the model class.
```

```
 */
```

```
//Model Structure
```

```
public class ClientInformation {
```

```
    int clientID;
```

```
    String name;
```

```
    String address;
```

```
    String phone;
```

```
    String email;
```

```
    String city, state;
```

```
    int zip;
```

```
    //                      //
```

```
    //    getters          //
```

```
    //                      //
```

```
    /** * @return the name */
```

```
    public String getName() {return name;}
```

```
    /**      * @return the clientID  */
```

```
    public int getClientID() {return clientID;}
```

```
    /**      * @return the email      */
```

```
    public String getEmail() {return email;}
```

```
    /**      * @return the address  */
```

```
    public String getAddress() {return address;}
```

```
    /**      * @return the phone  */
```

```
    public String getPhone() {return phone;}
```

```
    public String getCity() {return city;}
```

```
    public String getState() {return state;}
```

```
    public int getZip() {return zip; }
```

```
    //                      //
```

```
    //    setters          //
```

```
    //                      //
```

```
    /**      * @param clientID the clientID to set      */
```

```
    public void setClientID(int clientID) {this.clientID = clientID;}
```

```
    /**      * @param name the name to set      */
```

```
    public void setName(String name) {this.name = name;}
```

```
    /**      * @param phone the phone to set      */
```

```
    public void setPhone(String phone) {this.phone = phone;}
```

```
    /**      * @param address the address to set      */
```

```
    public void setAddress(String address) {this.address = address;}
```

```
    /**      * @param email the email to set *      */
```

```
    public void setEmail(String email) {this.email = email;}
```

```
    public void setCity(String city) {this.city = city;};
```

```
    public void setState(String state) {this.state = state;};
```

```
    public void setZip(int zip ) {this.zip = zip;}
```

```

}

/**
 * @author Jordan Williamson & Michelle Pham
 */
package classes;

import java.io.IOException;
import java.sql.SQLException;

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 classes.ClientInformation;

//This portion is currently hardcoded to return the
//values stored in ClientInformation.java,
//as we currently have no database of
//names to pull from.
/**
 * Servlet implementation class clientServlet
 */
@WebServlet("/clientServlet")
public class clientServlet extends HttpServlet {

    private static final long serialVersionUID = -7419930036249986454L;

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

        ClientInformation newCI = new ClientInformation();
        DatabaseCon sqlInsert = new DatabaseCon();

        // read form fields
        // String cid = request.getParameter("cid");
        String name = request.getParameter("name");
        String adr = request.getParameter("adr");
        String city = request.getParameter("city");
        String state = request.getParameter("state");
        String zip = request.getParameter("zip");
        String email = request.getParameter("email");
        String phone = request.getParameter("phone");

        newCI.setAddress(adr);
        // int id = Integer.parseInt(cid);
        // newCI.setClientID(id);
        newCI.setName(name);
        newCI.setEmail(email);
        newCI.setPhone(phone);
    }
}

```

```

        newCI.setCity(city);
        newCI.setState(state);
        int intzip = Integer.parseInt(zip);
        newCI.setZip(intzip);

// do some processing here...
//sets data for JSP file
request.getSession().setAttribute("name", name);
request.getSession().setAttribute("adr", adr);
request.getSession().setAttribute("city", city);
request.getSession().setAttribute("state", state);
request.getSession().setAttribute("zip", zip);
request.getSession().setAttribute("email", email);
request.getSession().setAttribute("phone", phone);

//JDBC Response
//Send ClientInfo obj to DatabaseCon obj function to insert into DB

    try {
        sqlInsert.sendClientSQL(newCI);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    int cid = sqlInsert.getClientID(name);
    request.getSession().setAttribute("cid", cid);
    // return response
    response.sendRedirect("clientinfo.jsp");
}

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

}

package classes;

import java.sql.Connection;
import java.sql.Date;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import org.joda.time.DateTime;

public class DatabaseCon {

```

```

String testDB = "jdbc:mysql://localhost:3306/cs3320";
String psw = "your_password";
String clientID = "root";//Note that at some point clD will be generated/used from elsewhere, likely for
Assignment 5?
String driver = "com.mysql.cj.jdbc.Driver";

public void sendClientSQL(ClientInformation newCI) throws SQLException
{
    try {
        Class.forName(driver);
    } catch (ClassNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    Connection SQLCon = DriverManager.getConnection(testDB, clientID, psw);
    Statement SQLState = SQLCon.createStatement();

    String name = newCI.getName();
    String email = newCI.getEmail();
    String phone = newCI.getPhone();
    int cid = newCI.getClientID();

    String adr = newCI.getAddress();
    String city = newCI.getCity();
    String state = newCI.getState();
    int zip = newCI.getZip();

    String statement = "INSERT INTO clientInformation (fullName, address, city, state, zipCode, phone,
email) "
        + "VALUES ('" + name + "', '" + adr + "', '" + city + "', '" + state + "', '" + zip + "', '" + phone
+ "'", '" + email + "')";
    long SQLResult = SQLState.executeLargeUpdate(statement);
}

public void sendQuoteToSQL(Quote quote) throws SQLException
{
    try {
        Class.forName(driver);
    } catch (ClassNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    Connection SQLCon = DriverManager.getConnection(testDB, clientID, psw);
    Statement SQLState = SQLCon.createStatement();

    int cid = quote.getClientID();
    String name = quote.getdeliveryContactName();
    String email = quote.getdeliveryContactEmail();
    String phone = quote.getdeliveryContactPhone();
    DateTime reqDate = quote.getRequestDate();
    DateTime delDate = quote.getdeliveryDate();
    String adr = quote.getdeliveryAdr();

```

```

String state = quote.getdeliveryState();
String city = quote.getdeliveryCity();
String zip = quote.getdeliveryZip();
double gals = quote.getgallonsRequested();
double price = quote.getsuggestedPrice();
double total = quote.getTotalAmountDue();

///Currently hardcoded clientID 1, because in order to put in a new quote,
///the ID must already exist in clientinformation table.
///
String insert = "INSERT INTO fuelquote (clientId, gallonsRequested, requestDate, deliveryDate,
deliveryAddress, deliveryCity, deliveryState,"
+ " deliveryZipCode, deliveryContactName, deliveryContactPhone, deliveryContactEmail,
suggestedPrice, totalAmountDue) "
+ "VALUES (" + cid + ", " + gals + ", " + reqDate + ", " + delDate + ", " + adr + ", " + city + ", " + state
+ ", " + zip + ", " + name + ", " + phone + ", " + email + ", " + price + ", " + total + ");";

long SQLResult = SQLState.executeLargeUpdate(insert);
}

public void updateClientSQL(ClientInformation newCI) throws SQLException
{
    try {
        Class.forName(driver);
    } catch (ClassNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    Connection SQLCon = DriverManager.getConnection(testDB, clientId, psw);

    String name = newCI.getName();
    String email = newCI.getEmail();
    String phone = newCI.getPhone();
    int cid = newCI.getClientID();

    String adr = newCI.getAddress();
    String city = newCI.getCity();
    String state = newCI.getState();
    int zip = newCI.getZip();

    String update = "UPDATE clientInformation SET address = " + adr + ", city = "
+ city + ", state = " + state + ", zipCode = " + zip + ", phone = " + phone + ", email
= "
+ email + " " + " WHERE fullName=" + name + " ";

    Statement ps = SQLCon.createStatement();

    // execute the java preparedstatement
    ps.executeUpdate(update);
}

public QuoteViewer getQuoteHistory()
{

```

```
QuoteViewer qv = new QuoteViewer();
String galReq, reqD, delD, delAdr, delZIP,
delST, delName, delPhone, delEmail, sugPrice, TAD = "";
```

```
//JDBC Implementation
```

```
//Connects to database, returns info.
```

```
try {
    try {
        Class.forName(driver);
    }
    catch (ClassNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
//    DriverManager.registerDriver(new Driver()); //Don't think this is needed.
    Connection testCon = DriverManager.getConnection(testDB, clientId, psw);
    Statement testState = testCon.createStatement();
    ResultSet testResult = testState.executeQuery("select * from fuelquote");

    while (testResult.next())
    {
        //Create a new quote object EVERY time
        Quote quote = new Quote();

        //reading data from each row while rows exist.

        galReq = testResult.getString("gallonsRequested");
        reqD = testResult.getString("requestDate");
        delD = testResult.getString("deliveryDate");
        delAdr = testResult.getString("deliveryAddress");
        delST = testResult.getString("deliveryState");
        delZIP = testResult.getString("deliveryZipCode");
        delName = testResult.getString("deliveryContactName");
        delPhone = testResult.getString("deliveryContactPhone");
        delEmail = testResult.getString("deliveryContactEmail");
        sugPrice = testResult.getString("suggestedPrice");
        TAD = testResult.getString("totalAmountDue");
        //set all quote obj data to above strings.

        int intQID = Integer.parseInt(testResult.getString("quoteId"));
        int intcid = Integer.parseInt(testResult.getString("clientId"));
        double dblGals = Double.parseDouble(galReq);
        double dblSugP = Double.parseDouble(sugPrice);
        double dblTAD = Double.parseDouble(TAD);

        quote.setgallonsRequested(dblGals);
        quote.setsuggestedPrice(dblSugP);
        quote.setTotalAmountDue(dblTAD);

        quote.setdeliveryAdr(delAdr);
        quote.setdeliveryState(delST);
        quote.setdeliveryZip(delZIP);

        quote.setQuoteID(intQID);
```

```

        quote.setClientID(intcid);
        quote.setdeliveryContactEmail(delEmail);
        quote.setdeliveryContactName(delName);
        quote.setdeliveryContactPhone(delPhone);

//          Date delDate = Date.parse(delD);
//          Date reqDate = Date.parse(reqD);
//          quote.setdeliveryDate(delDate);
//          quote.setrequestDate(reqDate);

//add new quote obj to list of quotes
//this should be a new quote for every row
        qv.QuoteHistory.add(quote);
//repeat
    }
}
catch (SQLException e)
{
    e.printStackTrace();
}
//return the list of quotes!
return qv;
}

public AllClientViewer getAllClients()
{
    AllClientViewer acv = new AllClientViewer();
    String name, addr, city, state, zip,
    phone, email = "";

//JDBC Implementation
//Connects to database, returns info.
    try {
        try {
            Class.forName(driver);
        }
        catch (ClassNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
//        DriverManager.registerDriver(new Driver()); //Don't think this is needed.
        Connection testCon = DriverManager.getConnection(testDB, clientID, psw);
        Statement testState = testCon.createStatement();
        ResultSet testResult = testState.executeQuery("select * from clientinformation");

        while (testResult.next())
        {
            //Create a new clientinfo object EVERY time
            ClientInformation client = new ClientInformation();

            //reading data from each row while rows exist.

            name = testResult.getString("fullName");
            addr = testResult.getString("address");

```



```

city = testResult.getString("city");
state = testResult.getString("state");
zip = testResult.getString("zipCode");
phone = testResult.getString("phone");
email = testResult.getString("email");
//set all quote obj data to above strings.

```

```

int intcid = Integer.parseInt(testResult.getString("clientId"));
int iZip = Integer.parseInt(zip);

```

```

client.setClientID(intcid);
client.setName(name);
client.setPhone(phone);
client.setEmail(email);
client.setAddress(addr);
client.setCity(city);
client.setState(state);
client.setZip(iZip);

```

```

//add new quote obj to list of quotes
//this should be a new quote for every row
acv.AllClient.add(client);

```

```

//repeat
    }
}
catch (SQLException e)
{
    e.printStackTrace();
}

```

```

//return the list of quotes!
return acv;
}

```

```

public int getClientID(String name)
{

```

```

    int intCid = 0;
    //JDBC Implementation
    //Connects to database, returns info.
    try {

```

```

        try {
            Class.forName(driver);
        } catch (ClassNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

```

```

        // DriverManager.registerDriver(new Driver()); //Don't think this is needed.
        Connection testCon = DriverManager.getConnection(testDB, clientId, psw);
        Statement testState = testCon.createStatement();
        ResultSet testResult = testState.executeQuery("select * from clientinformation WHERE

```

```

fullName="" + name + "";"");

```

```

        while (testResult.next())
        {
            //reading data from each row while rows exist.

```

```

        String cid = testResult.getString("clientId");
        intCid = Integer.parseInt(cid);
        //return intCid;
    }
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
return intCid;
}

//
//

}

/**
 * @author Jordan Williamson & Michelle Pham
 */
package classes;

import java.io.IOException;

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

//
//Servlet implementation class GetHistoryServlet
//
@WebServlet("/GetHistoryServlet")
public class GetHistoryServlet extends HttpServlet
{
    private static final long serialVersionUID = 5298036022257976608L;

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException
    {
        DatabaseCon dbQuery = new DatabaseCon();
        QuoteViewer quotes = dbQuery.getQuoteHistory();
        request.getSession().setAttribute("quoteList", quotes.QuoteHistory);
        response.sendRedirect("quotehistory.jsp");
    }

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

package classes;

```

```

/**
 * @author Jordan Williamson & Michelle Pham
 *
 */

import java.sql.Date;
import org.joda.time.DateTime;

public class Quote {

    int clientID;
    int quoteID;
    double gallonsRequested;
    DateTime deliveryDate;
    DateTime requestDate;
    String deliveryZip;
    String deliveryAdr, deliveryState, deliveryCity;
    String deliveryContactName;
    String deliveryContactPhone;
    String deliveryContactEmail;
    double suggestedPrice;
    double totalAmountDue;

    //                //
    //    getters    //
    //                //
    //@return the clientID
    public int getClientID() {return clientID;}
    //@return the gallonsRequested;
    public double getgallonsRequested() {return gallonsRequested;}
    //@return the deliveryDate
    public DateTime getdeliveryDate() {return deliveryDate;}
    //@return the requestDate
    public DateTime getrequestDate() {return requestDate;}
    //@return the del contactName
    public String getdeliveryContactName() {return deliveryContactName;}
    //@return the deliveryLocation
    public String getdeliveryZip() {return deliveryZip;}
    public String getdeliveryAdr() {return deliveryAdr;}
    public String getdeliveryState() {return deliveryState;}
    public String getdeliveryCity() {return deliveryCity;}
    //@return the delContactPhone
    public String getdeliveryContactPhone() {return deliveryContactPhone;}
    //@return the delContactEmail
    public String getdeliveryContactEmail() {return deliveryContactEmail;}
    //@return getSugPrice.
    public double getsuggestedPrice() {return suggestedPrice;}
    //@return totalAmountDue.
    public double getTotalAmountDue() {return totalAmountDue;}
    public int getQuoteID() {return quoteID;}

    //                //

```

```

//      setters      //
//                  //
public void setQuoteID(int quoteID) {this.quoteID = quoteID;}
public void setClientID(int clientID) {this.clientID = clientID;}
public void setgallonsRequested(double gallonsRequested) {this.gallonsRequested = gallonsRequested;}
public void setdeliveryDate(DateTime deliveryDate) {this.deliveryDate = deliveryDate;}
public void setrequestDate(DateTime requestDate) {this.requestDate = requestDate;}
public void setdeliveryZip(String deliveryZip) {this.deliveryZip = deliveryZip;}
public void setdeliveryAdr(String deliveryAdr) {this.deliveryAdr = deliveryAdr;}
public void setdeliveryState(String deliveryState) {this.deliveryState = deliveryState;}
public void setdeliveryCity(String deliveryCity) {this.deliveryCity = deliveryCity;}
public void setdeliveryContactName(String delContactName) {this.deliveryContactName = delContactName;}
public void setdeliveryContactPhone(String delContactPhone) {this.deliveryContactPhone = delContactPhone;}
public void setdeliveryContactEmail(String delContactEmail) {this.deliveryContactEmail = delContactEmail;}
public void setsuggestedPrice(double suggestedPrice) {this.suggestedPrice = suggestedPrice;}
public void setTotalAmountDue(double totalAmountDue) {this.totalAmountDue = totalAmountDue;}
}

/**
 * @author Jordan Williamson & Michelle Pham
 */
package classes;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
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 java.sql.Date;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Locale;
import java.time.LocalDateTime;
import org.joda.time.DateTime;
import org.joda.time.format.*;
import org.joda.time.LocalDate;
import java.time.format.DateTimeFormatter;

//
//Servlet implementation class QuoteServlet
//
@WebServlet("/QuoteServlet")
public class QuoteServlet extends HttpServlet
{
    private static final long serialVersionUID = 5298036022257976608L;

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

```

```

{
    Quote test = new Quote();

    //int cid = request.getParameter("clientId")
    //    SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    //    Date date = dateFormat.parse(request.getParameter("DelDate")); //Date convert error
    //    Converts java.util.Date to java.sql.Date format
    //    java.util.Date utilDate = dateFormat.parse(request.getParameter("DelDate"));
    //    java.sql.Date date = new java.sql.Date(utilDate.getTime());
    //    System.out.println(date);

    /*        //Parse datetime
        DateTime dt = DateTime.parse(request.getParameter("DelDate"));
        System.out.println(dt);

        //Converts dt to the correct format
        DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy-MM-dd'T'HH:mm:ss.SSSXXX");
    //The format dt gets stored in
        DateTimeFormatter dtf2 = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss"); //The
    format MYSQL datetime is stored

        String dateStr = dt.toString();

        LocalDateTime dateTime = LocalDateTime.parse(dateStr, dtf);
        String formattedDateTime = dateTime.format(dtf2);
        System.out.println(formattedDateTime);

        System.out.println(dateTime);
        System.out.println(dateTime.format(dtf2));*/

    String cid = request.getParameter("cid");
    String name = request.getParameter("DelCPN");
    String email = request.getParameter("DelCPE");
    String phone = request.getParameter("DelCPP");
    String adr = request.getParameter("DelAdr");
    String state = request.getParameter("DelState");
    String city = request.getParameter("DelCity");
    String zip = request.getParameter("DelZip");

    double gals = Double.parseDouble(request.getParameter("GalReq"));
    System.out.println(gals);
    double price = Double.parseDouble(request.getParameter("PPG"));
    System.out.println(price);
    double total = price*gals;

    int intCID = Integer.parseInt(cid);
    test.setClientID(intCID);
    //    test.setdeliveryDate(dt);
    test.setdeliveryAdr(adr);
    test.setdeliveryCity(city);
    test.setdeliveryState(state);
    test.setdeliveryZip(zip);
    test.setgallonsRequested(gals);

```

```

test.setdeliveryContactEmail(email);
test.setdeliveryContactName(name);
test.setdeliveryContactPhone(phone);
test.setsuggestedPrice(price);
test.setTotalAmountDue(total);

```

```

//sets data for JSP file
//    request.getSession().setAttribute("DelDate", dt);
//    request.getSession().setAttribute("cid", cid);
//    request.getSession().setAttribute("DelAdr", adr);
//    request.getSession().setAttribute("DelState", state);
//    request.getSession().setAttribute("DelCity", city);
//    request.getSession().setAttribute("DelZip", zip);
//    request.getSession().setAttribute("GalReq", gals);
//    request.getSession().setAttribute("DelCPN", name);
//    request.getSession().setAttribute("DelCPE", email);
//    request.getSession().setAttribute("DelCPP", phone);
//    request.getSession().setAttribute("PPG", price);
//    request.getSession().setAttribute("TotalDue", total);

```

```

//Calls new DatabaseCon class, containing functions for various DB actions.

```

```

    DatabaseCon querydb = new DatabaseCon();

```

```

//Send new Quote Object to the SQL database as an SQL entry.

```

```

    try {

```

```

        querydb.sendQuoteToSQL(test); //Sends quote object to DatabaseCon function

```

```

        //which translates into SQL execution.

```

```

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

```

```

//////////

```

```

    // return response

```

```

    response.sendRedirect("returnquote.jsp");

```

```

}

```

```

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

```

```

{

```

```

    doGet(request, response);

```

```

}

```

```

}

```

```

package classes;

```

```

/**

```

```

 * @author Jordan Williamson & Michelle Pham

```

```

 *

```

```

 */

```

```

import java.util.ArrayList;

```

```

import java.util.List;

```

```

//    We will use this list to display quote history in UI.

```

```

public class QuoteViewer {

```

```

        List <Quote>QuoteHistory = new ArrayList<Quote>();
    }

package classes;

import java.io.IOException;
import java.sql.SQLException;

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 classes.ClientInformation;

@WebServlet("/UpdateClientServlet")
public class UpdateClientServlet extends HttpServlet {

    private static final long serialVersionUID = -7419930036249986454L;

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

        ClientInformation newCI = new ClientInformation();
        DatabaseCon sqlUpdate = new DatabaseCon();

        // read form fields
        // String cid = request.getParameter("cid");
        String name = request.getParameter("name");
        String adr = request.getParameter("adr");
        String city = request.getParameter("city");
        String state = request.getParameter("state");
        String zip = request.getParameter("zip");
        String email = request.getParameter("email");
        String phone = request.getParameter("phone");

        newCI.setAddress(adr);
        // int id = Integer.parseInt(cid);
        // newCI.setClientID(id);
        newCI.setName(name);
        newCI.setEmail(email);
        newCI.setPhone(phone);

        newCI.setCity(city);
        newCI.setState(state);
        int intzip = Integer.parseInt(zip);
        newCI.setZip(intzip);

        // do some processing here...
        //sets data for JSP file
        request.getSession().setAttribute("name", name);
        request.getSession().setAttribute("adr", adr);
    }
}

```

```

request.getSession().setAttribute("city", city);
request.getSession().setAttribute("state", state);
request.getSession().setAttribute("zip", zip);
request.getSession().setAttribute("email", email);
request.getSession().setAttribute("phone", phone);

```

```

//JDBC Response
//Send ClientInfo obj to DatabaseCon obj function to insert into DB

```

```

try {
    sqlUpdate.updateClientSQL(newCI);
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

int cid = sqlUpdate.getClientID(name);
request.getSession().setAttribute("cid", cid);
// return response
response.sendRedirect("updateclientinfo.jsp");

```

```

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

```

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1" import="classes.ClientInformation,
javax.servlet.jsp.jstl.core.*, classes.AllClientViewer"
    pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<link rel="stylesheet" href="styling.css">
<title>Client Information</title>
</head>
<body>
<header>
<h2>All Clients</h2>
</header>
<section>
<nav>
<ul>
<!-- <li><a href="login.html">Login/Signup</a></li> -->
<li><a href="Index.html">Home</a></li>
<li><a class="active" href="clientinfo.html">Client Information</a></li>
<li><a href="getquote.html">Request a Quote</a></li>

```



```
    <li><a href="history.html">Quote History</a></li>
</ul>
</nav>
```

```
<article>
  <h1>Client Information</h1>
  <table>
    <tr>
      <th>Client ID</th>
      <th>Name</th>
      <th>Phone</th>
      <th>Email</th>
      <th>Address</th>
      <th>City</th>
      <th>State</th>
      <th>Zip</th></tr>
```

```

    <c:forEach items="${clientList}" var="client">
      <tr>
        <td><c:out value="${client.clientID}" /></td>
        <td><c:out value="${client.name}" /></td>
        <td><c:out value="${client.phone}" /></td>
        <td><c:out value="${client.email}" /></td>
        <td><c:out value="${client.address}" /></td>
        <td><c:out value="${client.city}" /></td>
        <td><c:out value="${client.state}" /></td>
        <td><c:out value="${client.zip}" /></td>
      </tr>
    </c:forEach>
```

```

    <form name="AllClients" method="get" action="AllClientServlet">
      <br><input type="submit" value="View ALL Client Information">
    </form>
```

```

  </table>
</article>
</section>
```

```
<footer>
  <p>Copyright © 2018 Jordan Williamson & Michelle Pham. All rights reserved.</p>
</footer>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <title>Client Information</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="styling.css">
</head>
```

```

<body>
<header>
  <h2>Client Information</h2>
</header>

<section>
  <nav>
    <ul>
<!-- <li><a href="login.html">Login/Signup</a></li> -->
      <li><a href="Index.html">Home</a></li>
      <li><a class="active" href="clientinfo.html">Client Information</a></li>
      <li><a href="getquote.html">Request a Quote</a></li>
      <li><a href="history.html">Quote History</a></li>
    </ul>
  </nav>

  <article>
<h1>Enter NEW Client Information Here</h1>
<form name="contactCard" method="post" action="clientServlet">
  <b>Name:</b>
  <br><input type="text" placeholder="Enter Name" name="name" required><br>
  <b>Address:</b>
  <br><input type="text" placeholder="Address" name="adr" required><br>
  <b>City:</b>
  <br><input type="text" placeholder="City" name="city" required><br>
  <b>State:</b>
  <br><input type="text" placeholder="ST" name="state" required><br>
  <b>Zip:</b>
  <br><input type="number" placeholder="Zipcode" name="zip" required><br>
  <b>Phone:</b>
  <br><input type="text" placeholder="Enter Phone #" name="phone" required><br>
  <b>Email:</b>
  <br><input type="email" placeholder="Enter Email" name="email" required><br>

  <input type="submit" name="submitcc">
</form>

<form name="AllClients" method="post" action="AllClientServlet">
<br><b>Click Here To View All Client Information:</b><br>
<br><input type="submit" value="View ALL Client Information"></form>

<form name="UpdateClients" method="post" action="updateclientinfo.html">
<br><b>Click Here To Update A Client's Information:</b><br>
<br><input type="submit" value="Update A Client's Info"></form>

</article>
</section>

<footer>

```

<p>Copyright © 2018 Jordan Williamson & Michelle Pham. All rights reserved.</p>

</footer>

</body>

</html>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<link rel="stylesheet" href="styling.css">

<title>Client Information</title>

</head>

<body>

<header>

<h2>Client Information</h2>

</header>

<section>

<nav>

<!-- Login/Signup -->

Home

Client Information

Request a Quote

Quote History

</nav>

<article>

<h1>ClientID: <%= session.getAttribute("cid") %></h1>

<p>Name: <%= session.getAttribute("name") %>

Address: <%= session.getAttribute("adr") %>

Email: <%= session.getAttribute("email") %>

Phone: <%= session.getAttribute("phone") %> </p>

</article>

</section>

<footer>

<p>Copyright © 2018 Jordan Williamson & Michelle Pham. All rights reserved.</p>

</footer>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<title>Request Quote</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="styling.css">

</head>

<body>

```
<header>
  <h2>Request a Quote</h2>
</header>
<section>
  <nav>
    <ul>
<!-- <li><a href="login.html">Login/Signup</a></li> -->
      <li><a href="Index.html">Home</a></li>
      <li><a href="clientinfo.html">Client Information</a></li>
      <li><a class="active" href="getquote.html">Request a Quote</a></li>
      <li><a href="history.html">Quote History</a></li>
    </ul>
  </nav>

  <article>
    <h1>Order Form</h1>
    <form name="OrderForm" method="get" action="QuoteServlet">
      <b>Client ID:</b>
      <br><input type="number" name="cid" placeholder="Only Existing Client ID's are Valid" required><br>
      <b>Gallons Requested:</b>
      <br><input type="number" name="GalReq" placeholder="Enter Gallons" required><br>
      <b>Delivery Date:</b>
      <br><input type="datetime-local" step="1" name="DelDate" placeholder="YYYY-MM-DDTHH:MM:SS" required><br>
      <b>Delivery Address:</b>
      <br><input type="text" name="DelAdr" placeholder="Address" required><br>
      <b>Delivery City:</b>
      <br><input type="text" name="DelCity" placeholder="City" required><br>
      <b>Delivery State:</b>
      <br><input type="text" name="DelState" placeholder="State" required><br>
      <b>Delivery Zip:</b>
      <br><input type="text" name="DelZip" placeholder="Zipcode" required><br>
      <b>Delivery Contact Person Name:</b>
      <br><input type="text" name="DelCPN" pattern="[A-Z a-z]{3,45}" placeholder="Enter Contact Name" required><br>
      <b>Delivery Contact Person Phone:</b>
      <br><input type="tel" name="DelCPP" pattern="[0-9]{3}[0-9]{3}[0-9]{4}" placeholder="5129992020" required><br>
      <b>Delivery Contact Person Email:</b>
      <br><input type="email" name="DelCPE" placeholder="Enter Contact Email" required><br>
      <b>Suggested Price:</b>
      <br><input type="number" step="0.01" name="PPG" placeholder="Enter Suggested Price" required><br>
      <b>Total Amount Due: Submit to Calculate</b><br><br>
      <br><input type="submit" value="Submit Request">
    </form>
  </article>
</section>

<footer>
  <p>Copyright © 2018 Jordan Williamson & Michelle Pham. All rights reserved.</p>
</footer>

</body>
</html>

<!DOCTYPE html>
```

```

<html lang="en">

<head>
<title>Quote History</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="styling.css">
<script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</head>
<body>
<header>
  <h2>Quote History</h2>
</header>

<section>
<nav>
  <ul>
    <!-- <li><a href="login.html">Login/Signup</a></li> -->
    <li><a href="Index.html">Home</a></li>
    <li><a href="clientinfo.html">Client Information</a></li>
    <li><a href="getquote.html">Request a Quote</a></li>
    <li><a class="active" href="history.html">Quote History</a></li>
  </ul>
</nav>

<article>
  <h1>Quote History</h1>
  <table id="quotehistory">
    <th>Client ID</th>
    <th>Contact Name</th>
    <th>Contact Email</th>
    <th>Contact Phone</th>
    <th>Gallons Requested</th>
    <th>Request Date</th>
    <th>Date of Delivery</th>
    <th>Delivery Address</th>
    <th>Delivery State</th>
    <th>Delivery ZIP</th>
    <th>Total Due</th></tr>
  </table>
  <form name="HistoryRequest" method="get" action="GetHistoryServlet" id="getQuotes">
    <br><input type="submit" value="Get Quote History">
  </form>
</article>
</section>

<footer>
  <p>Copyright © 2018 Jordan Williamson & Michelle Pham. All rights reserved.</p>
</footer>

</body>
</html>

```

```

<!DOCTYPE html>
<html lang="en">

<head>
<title>Assignment 3 Index</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="styling.css">
</head>
<body>
<header>
  <h2>Index</h2>
</header>

<section>
  <nav>
    <ul>
<!-- <li><a href="login.html">Login/Signup</a></li> -->
      <li><a class="active" href="Index.html">Home</a></li>
      <li><a href="clientinfo.html">Client Information</a></li>
      <li><a href="getquote.html">Request a Quote</a></li>
      <li><a href="history.html">Quote History</a></li>
    </ul>
  </nav>

  <article>
    <p>This software application will predict the rate of the fuel based on the following criteria:<br><br>
      <ul>
        <li>Location<br>
        <li>Competitors rate<br>
        <li>Client rate history<br>
        <li>Client rate negotiation history<br>
        <li>Gallons requested<br>
        <li>Company profit margin (%)<br>
        <li>Seasonal rate fluctuation (%)<br>
      </ul>
    </p>
  </article>
</section>

<footer>
  <p>Copyright © 2018 Jordan Williamson & Michelle Pham. All rights reserved.</p>
</footer>

</body>
</html>

```

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1" import="classes.Quote,
javax.servlet.jsp.jstl.core.*, classes.QuoteViewer"
    pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

```

```

<link rel="stylesheet" href="styling.css">
<title>Quote Information</title>
</head>
<body>
<header>
  <h2>Quote History</h2>
</header>

<section>
  <nav>
    <ul>
<!-- <li><a href="login.html">Login/Signup</a></li> -->
      <li><a href="Index.html">Home</a></li>
      <li><a href="clientinfo.html">Client Information</a></li>
      <li><a href="getquote.html">Request a Quote</a></li>
      <li><a class="active" href="history.html">Quote History</a></li>
    </ul>
  </nav>
<article>
  <h1>Quote History</h1>
  <table>
    <tr>
      <th>Client ID</th>
      <th>Contact Name</th>
      <th>Contact Email</th>
      <th>Contact Phone</th>
      <th>Gallons Requested</th>
      <th>Request Date</th>
      <th>Date of Delivery</th>
      <th>Delivery Address</th>
      <th>Delivery State</th>
      <th>Delivery ZIP</th>
      <th>Total Due</th></tr>

    <c:forEach items="{quoteList}" var="Quote">
      <tr>
        <td><c:out value="{Quote.clientID}" /></td>
        <td><c:out value="{Quote.deliveryContactName}" /></td>
        <td><c:out value="{Quote.deliveryContactEmail}" /></td>
        <td><c:out value="{Quote.deliveryContactPhone}" /></td>
        <td><c:out value="{Quote.gallonsRequested}" /></td>
        <td><c:out value="{Quote.requestDate}" /></td>
        <td><c:out value="{Quote.deliveryDate}" /></td>
        <td><c:out value="{Quote.deliveryAdr}" /></td>
        <td><c:out value="{Quote.deliveryState}" /></td>
        <td><c:out value="{Quote.deliveryZip}" /></td>
        <td><c:out value="{Quote.totalAmountDue}" /></td>
      </tr>
    </c:forEach>
  </table>
  <form name="HistoryRequest" method="get" action="GetHistoryServlet">
    <br><input type="submit" value="Get NEW Quotes">
  </form>

```

```
</article>
</section>
```

```
<footer>
  <p>Copyright © 2018 Jordan Williamson & Michelle Pham. All rights reserved.</p>
</footer>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<link rel="stylesheet" href="styling.css">
<title>Quote Information</title>
</head>
<body>
<header>
  <h2>Requested Quote</h2>
</header>
<section>
  <nav>
    <ul>
<!-- <li><a href="login.html">Login/Signup</a></li> -->
      <li><a href="Index.html">Home</a></li>
      <li><a href="clientinfo.html">Client Information</a></li>
      <li><a class="active" href="getquote.html">Request a Quote</a></li>
      <li><a href="history.html">Quote History</a></li>
    </ul>
  </nav>

  <article>
    <h1>Quote Offer</h1>
    <p><b>Delivery Date: </b><%= session.getAttribute("DelDate") %> <br><br>
      <b>Delivery Location: </b><%= session.getAttribute("DelAdr") %><br><br>
      <b>Number of Gallons: </b><%= session.getAttribute("GalReq") %> <br><br>
      <b>Contact Name: </b><%= session.getAttribute("DelCPN") %> <br><br>
      <b>Contact Email: </b><%= session.getAttribute("DelCPE") %> <br><br>
      <b>Contact Phone: </b><%= session.getAttribute("DelCPP") %> <br><br>
      <b>Suggested Price: </b><%= session.getAttribute("PPG") %> <br><br>
      <b>Total Due: </b><%= session.getAttribute("TotalDue") %> </p>
  </article>
</section>

<footer>
  <p>Copyright © 2018 Jordan Williamson & Michelle Pham. All rights reserved.</p>
</footer>
</body>
</html>

<style>
```



```
{
  box-sizing: border-box;
}

body {
  font-family: Arial, Helvetica, sans-serif;
}

/* Style the header */
header {
  background-color: #7371FC;
  padding: 30px;
  text-align: center;
  font-size: 35px;
  color: white;
}

/* Container for flexboxes */
section {
  display: -webkit-flex;
  display: flex;
}

/* Style the navigation menu */
nav {
  -webkit-flex: 1;
  -ms-flex: 1;
  flex: 1;
  background: #CDC1FF;
  padding: 20px;
}

/* Style the list inside the menu */
nav ul {
  list-style-type: none;
  padding: 0;
}

nav li a {
  display: block;
  color: #000;
  padding: 8px 16px;
  text-decoration: none;
}

nav li a.active {
  background-color: #A594F9;
}

nav li a:hover:not(.active) {
  background-color: #E5D9F2;
}
```

```

/* Style the content */
article {
    -webkit-flex: 3;
    -ms-flex: 3;
    flex: 3;
    background-color: #F5EFFF;
    padding: 10px;
}

/* Style the footer */
footer {
    background-color: #7371FC;
    padding: 10px;
    text-align: center;
    color: white;
}

/* Style for table*/
table, th, td {
    border: 1px solid black;
    border-collapse: collapse;
    empty-cells: show;
}

th {
    height: 50px;
}

/* Style for form */
input[type=text], input[type=password], input[type=email], input[type=number],
input[type=tel], input[type=date] , input[type=datetime-local]
{
    width: 100%;
    padding: 10px;
    margin: 5px 0 15px 0;
    display: inline-block;
    border: none;
    background: #f1f1f1;
}

input[type=text]:focus, input[type=password]:focus, input[type=email]:focus,
input[type=number]:focus, input[type=tel]:focus, input[type=date]:focus, input[type=datetime-local]:focus
{
    background-color: #ddd;
    outline: none;
}

input[type=submit]{
    background-color: #7371FC;
    color: white;
    padding: 16px 20px;
    margin: 8px 0;
    border: none;
    cursor: pointer;
}

```

```
width: 100%;
opacity: 0.9;
}
```

```
input[type=submit]:hover{
    opacity: 1;
}
```

```
/* Responsive layout - makes the menu and the content (inside the section) sit on top of each other instead of next to each other */
```

```
@media (max-width: 600px) {
    section {
        -webkit-flex-direction: column;
        flex-direction: column;
    }
}
</style>
```

```
<!DOCTYPE html>
<html lang="en">

<head>
<title>Update Client Information</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="styling.css">
</head>
```

```
<body>
<header>
    <h2>Update Client Information</h2>
</header>
```

```
<section>
    <nav>
        <ul>
<!-- <li><a href="login.html">Login/Signup</a></li> -->
        <li><a href="Index.html">Home</a></li>
        <li><a class="active" href="clientinfo.html">Client Information</a></li>
        <li><a href="getquote.html">Request a Quote</a></li>
        <li><a href="history.html">Quote History</a></li>
        </ul>
    </nav>
```

```
<article>
<h1>Enter The Client Information You Want To Update Here</h1>
<form name="contactCard" method="post" action="UpdateClientServlet">
    <b>Enter Name Of The Client You Want To Update:</b><br><br>
    <b>Name:</b>
    <br><input type="text" placeholder="Enter Name" name="name" required><br>
    <b>Enter The Old & New Information Of The Client:</b><br><br>
```

```
<b>Address:</b>
<br><input type="text" placeholder="Address" name="adr" required><br>
<b>City:</b>
<br><input type="text" placeholder="City" name="city" required><br>
<b>State:</b>
<br><input type="text" placeholder="ST" name="state" required><br>
<b>Zip:</b>
<br><input type="number" placeholder="Zipcode" name="zip" required><br>
<b>Phone:</b>
<br><input type="text" placeholder="5125550000" name="phone" required><br>
<b>Email:</b>
<br><input type="email" placeholder="Enter Email" name="email" required><br>

<input type="submit" name="Update Info">
```

```
</form>
</article>
</section>
```

```
<footer>
  <p>Copyright © 2018 Jordan Williamson & Michelle Pham. All rights reserved.</p>
</footer>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<link rel="stylesheet" href="styling.css">
<title>Client Information</title>
</head>
<body>
<header>
  <h2>Client Information</h2>
</header>
<section>
  <nav>
    <ul>
<!-- <li><a href="login.html">Login/Signup</a></li> -->
      <li><a href="Index.html">Home</a></li>
      <li><a class="active" href="clientinfo.html">Client Information</a></li>
      <li><a href="getquote.html">Request a Quote</a></li>
      <li><a href="history.html">Quote History</a></li>
    </ul>
  </nav>

  <article>
    <h1>Updated Client's Information:</h1>
    <p><b>ClientID: <%= session.getAttribute("cid") %></b></p>
    <p><b>Name:</b> <%= session.getAttribute("name") %> <br><br>
```

```

        <b>Address:</b> <%= session.getAttribute("adr") %> <br><br>
        <b>Email:</b> <%= session.getAttribute("email") %> <br><br>
        <b>Phone:</b> <%= session.getAttribute("phone") %> </p>
    </article>
</section>
<footer>
    <p>Copyright © 2018 Jordan Williamson & Michelle Pham. All rights reserved.</p>
</footer>
</body>
</html>

<?xml version="1.0" encoding="UTF-8"?>
<web-app

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
version="3.1">

    <display-name>ISWD-Project</display-name>

    <welcome-file-list>
        <welcome-file>Index.html</welcome-file>
    </welcome-file-list>

</web-app>

```