**Drug-Drug Interaction Checking Tool**

**Summary:**

The objective of the project is to develop a system which helps doctors to provide better service to their patients with interaction checking for the medication prescribed to patients. In medical terms, It is called as drug-drug interaction where drugs which are prescribed to patients are checked interaction using API and pop up alert to doctor if the interaction occurs for that patient.

The workflow includes physician placing an order against a patient. We will be performing interaction checking while the physician places the order for Drug-Drug interactions. Application will help physician prescribe correct order to avoid adverse drug events. This application will have entities like Doctor, Patient and Pharmacy (Drugs list) and related database tables will be created to support all the entities associated with this application like tables related to doctor profile, patient profile, drugs list and interaction results.

**Functionality Performed:**

Below functionality performed as part of this web application.

1. Register User (Doctor, Receptionist or Pharmacist) in the application.
2. Login functionality for each role type and redirected to respective home page for each user.
3. Storing of drug details by pharmacist in application.
4. Storing patient details by receptionist in application.
5. Searching patient details by doctor user for order summary flow.
6. Searching drug details by doctor user to prescribe the patient. Multiple drug searching is implemented.
7. Checking drug-drug interaction by doctor for the prescribed drug to patient using API implementation.
8. Parsing JSON data from API output to get interaction description for the prescribed drugs.
9. PDF creation for the order summary for the patient.

**Technologies Used:**

* Spring MVC Framework
* Hibernate
* POJOs
* Servlet
* JSP and JSTL
* HQL
* API integration
* HTML and CSS

**Roles of application:**

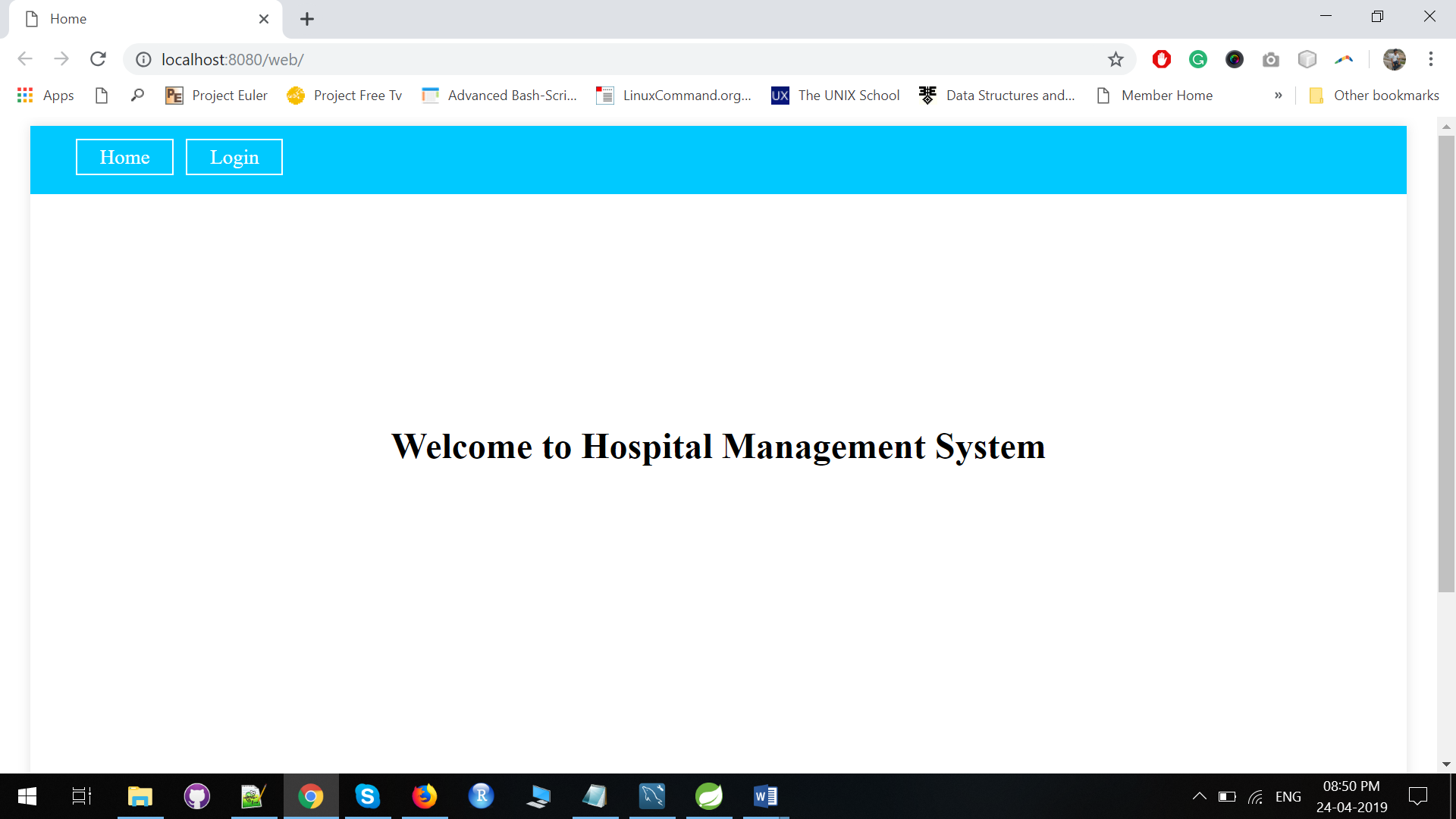
(1) Pharmacist: Pharmacist can add drug with drug id, drug name and drug dosage detail in the database which can be prescribed by the doctor to patient.

(2) Receptionist: Receptionist can register patient detail along with detail like patient name, patient mobile number, patient age and patient address detail to database. This detail can be searched by the doctor when patient goes to doctor for medication or checkup.

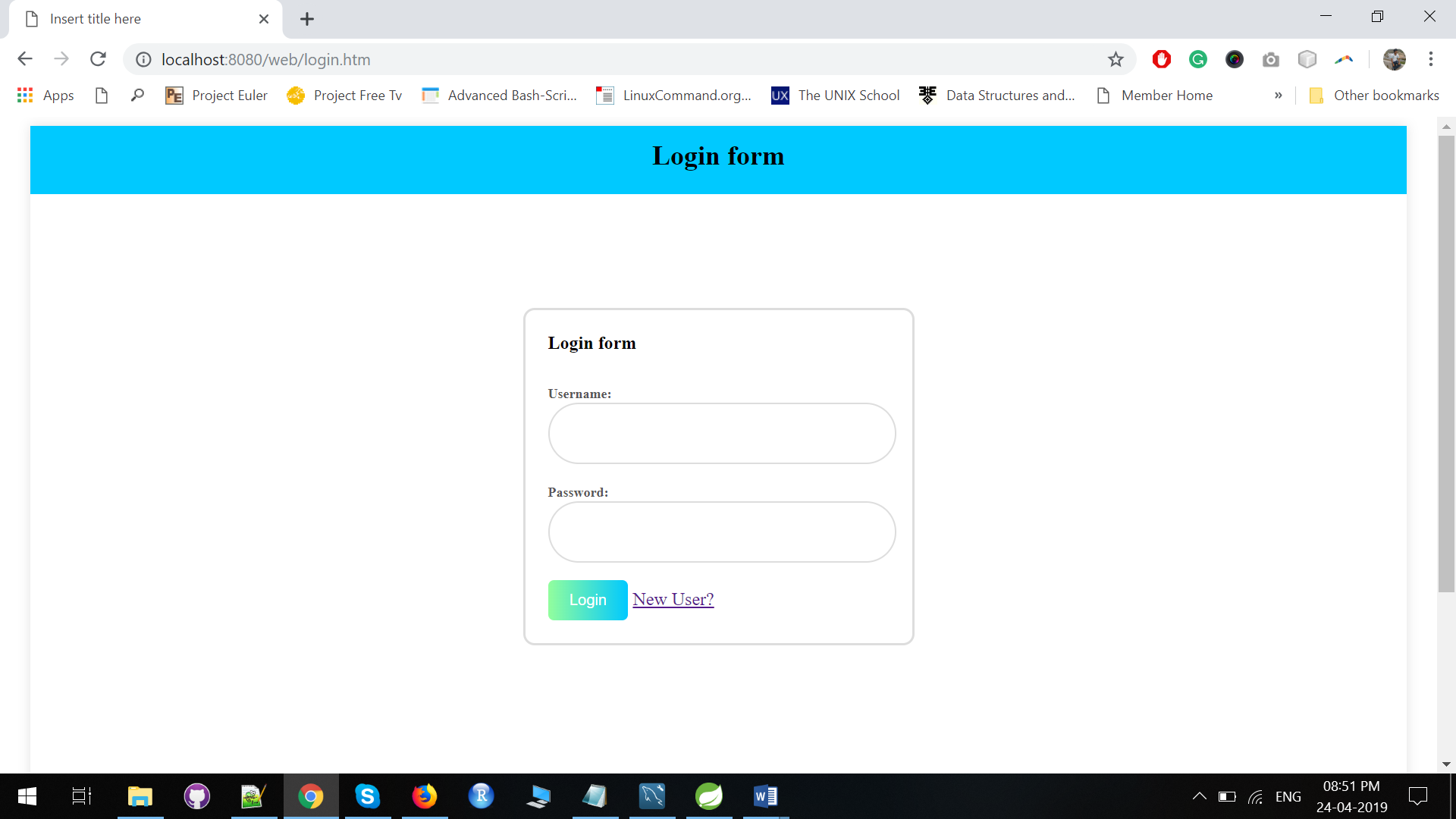
(3) Doctor: Doctor can search patient name from the database to start the checkup for current patient along with drug search functionality to prescribe drugs to patient after the checkup. This functionality can add multiple drugs to order by doctor. After drug prescription, doctor can check interaction for prescribed drug to patient and create pdf report for the order summary.

**Screenshots of application:**

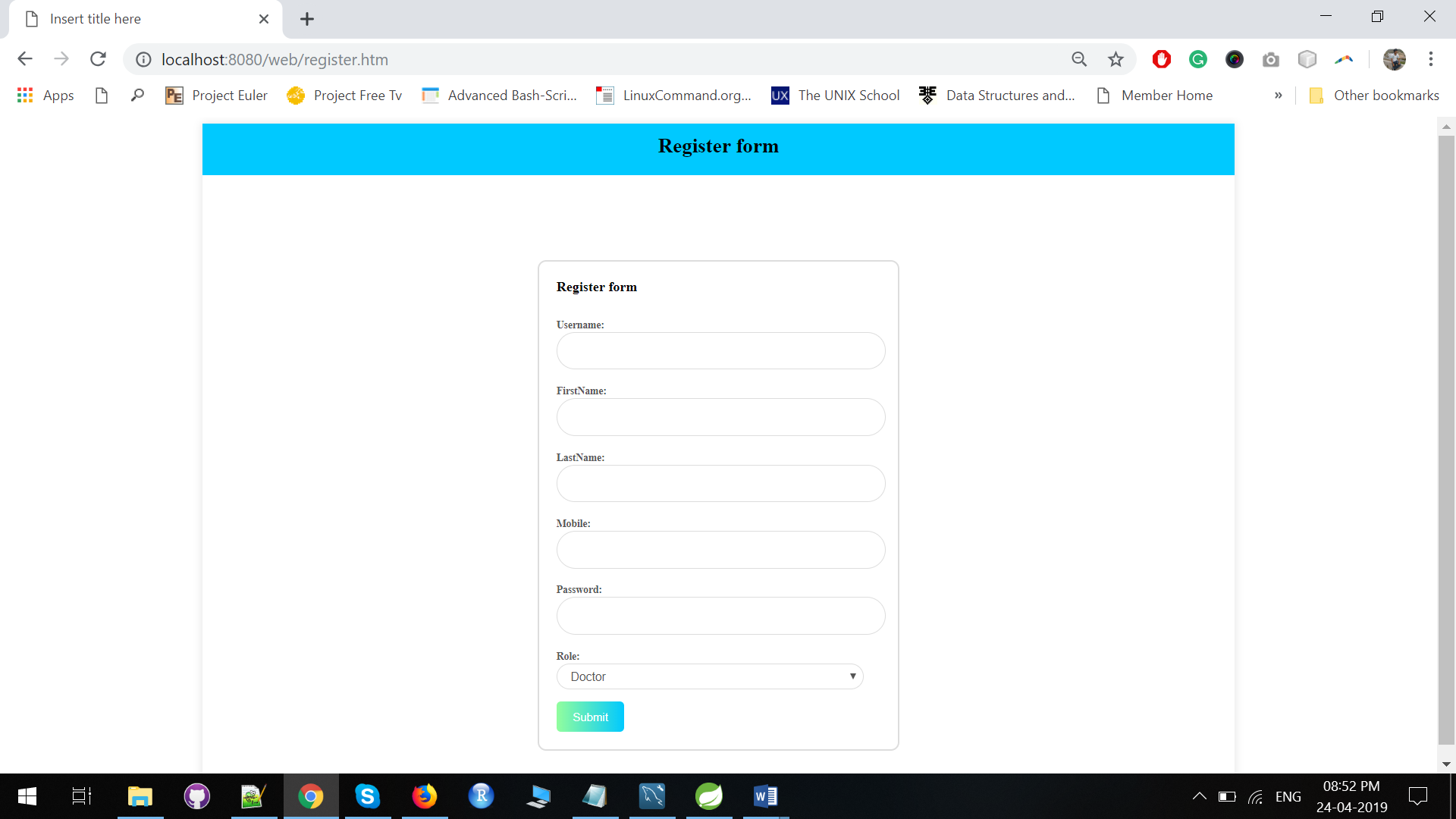
(1) Home page:



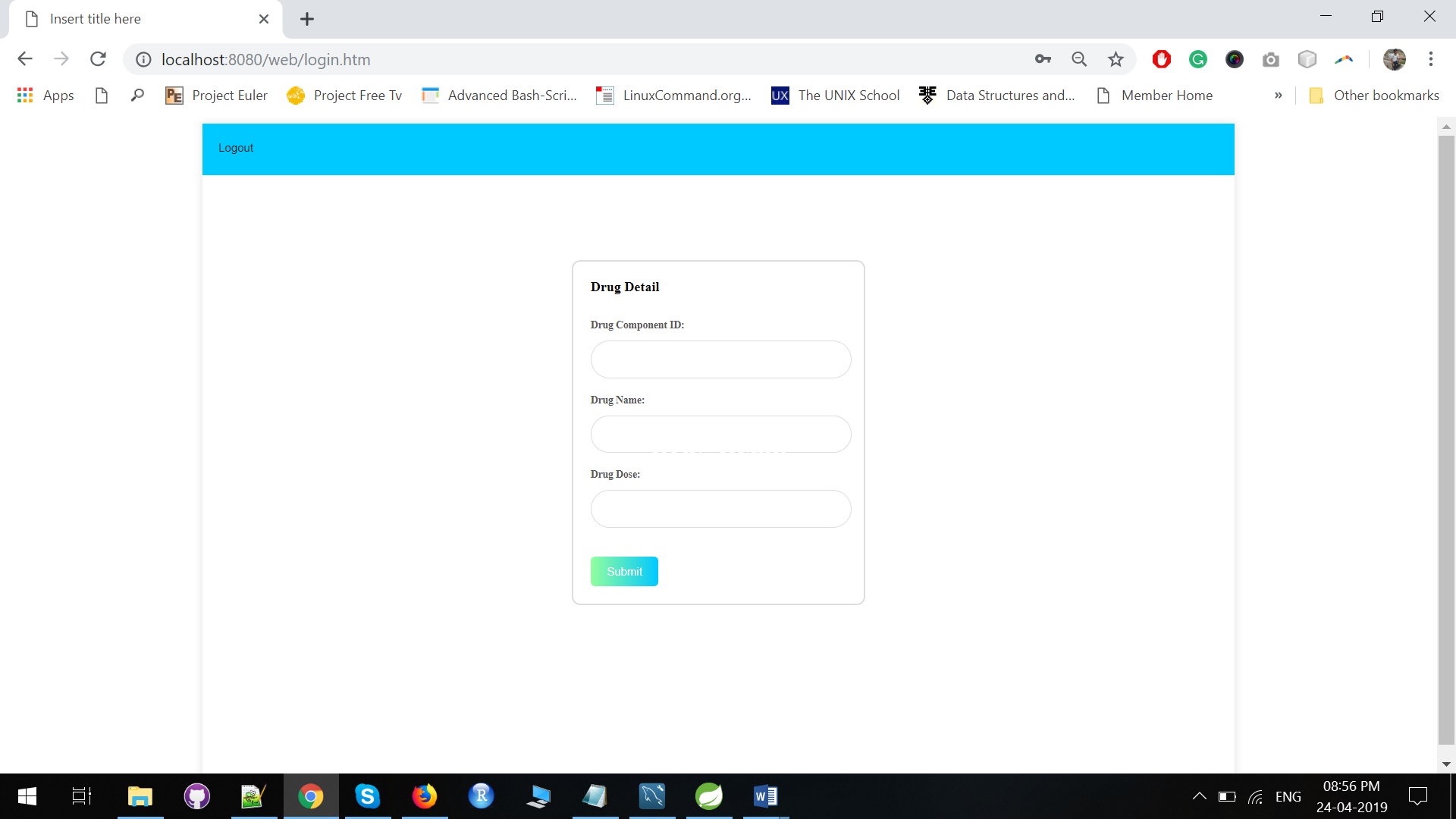
(2) Login page:



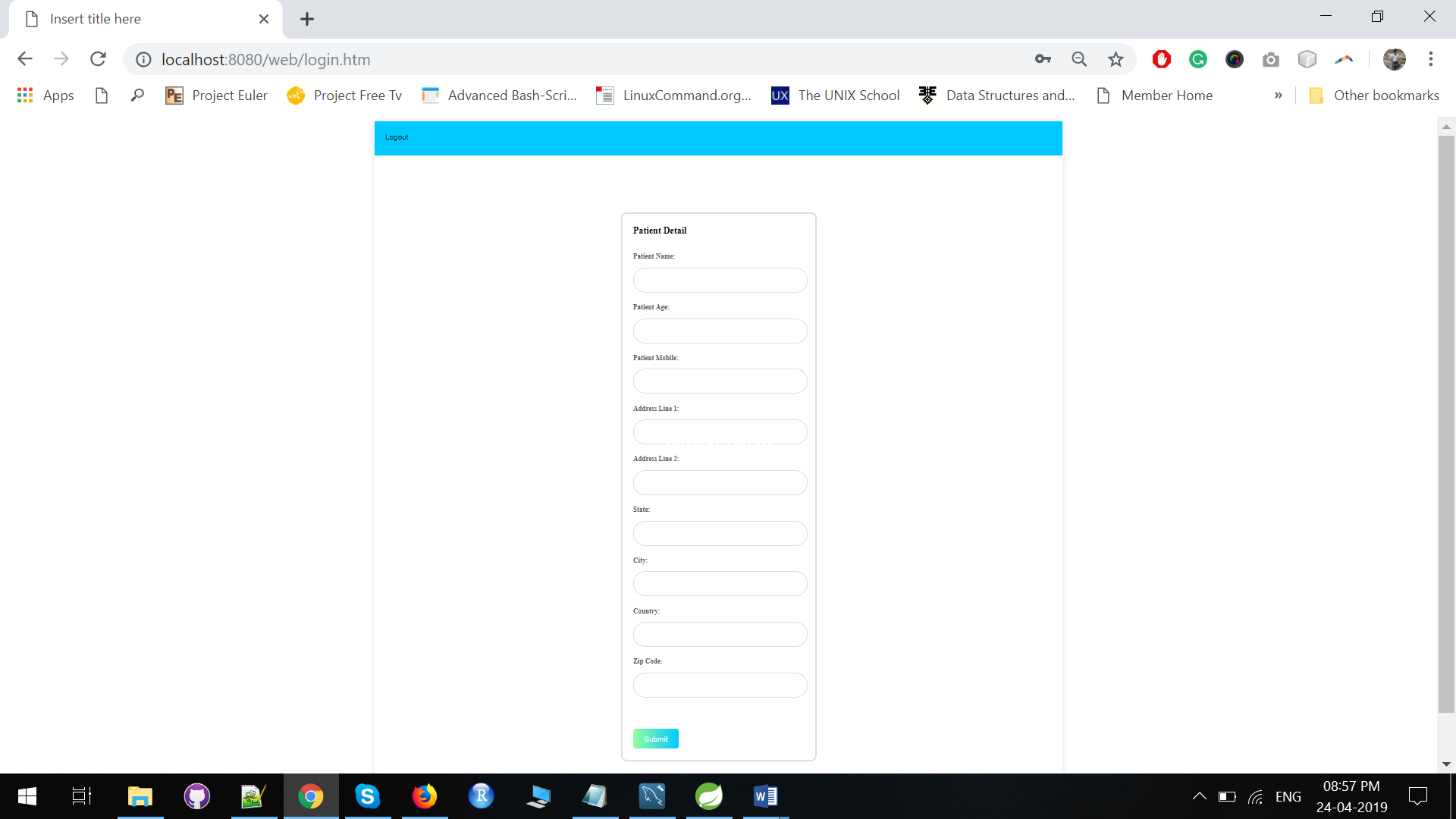
(3) Register User page:



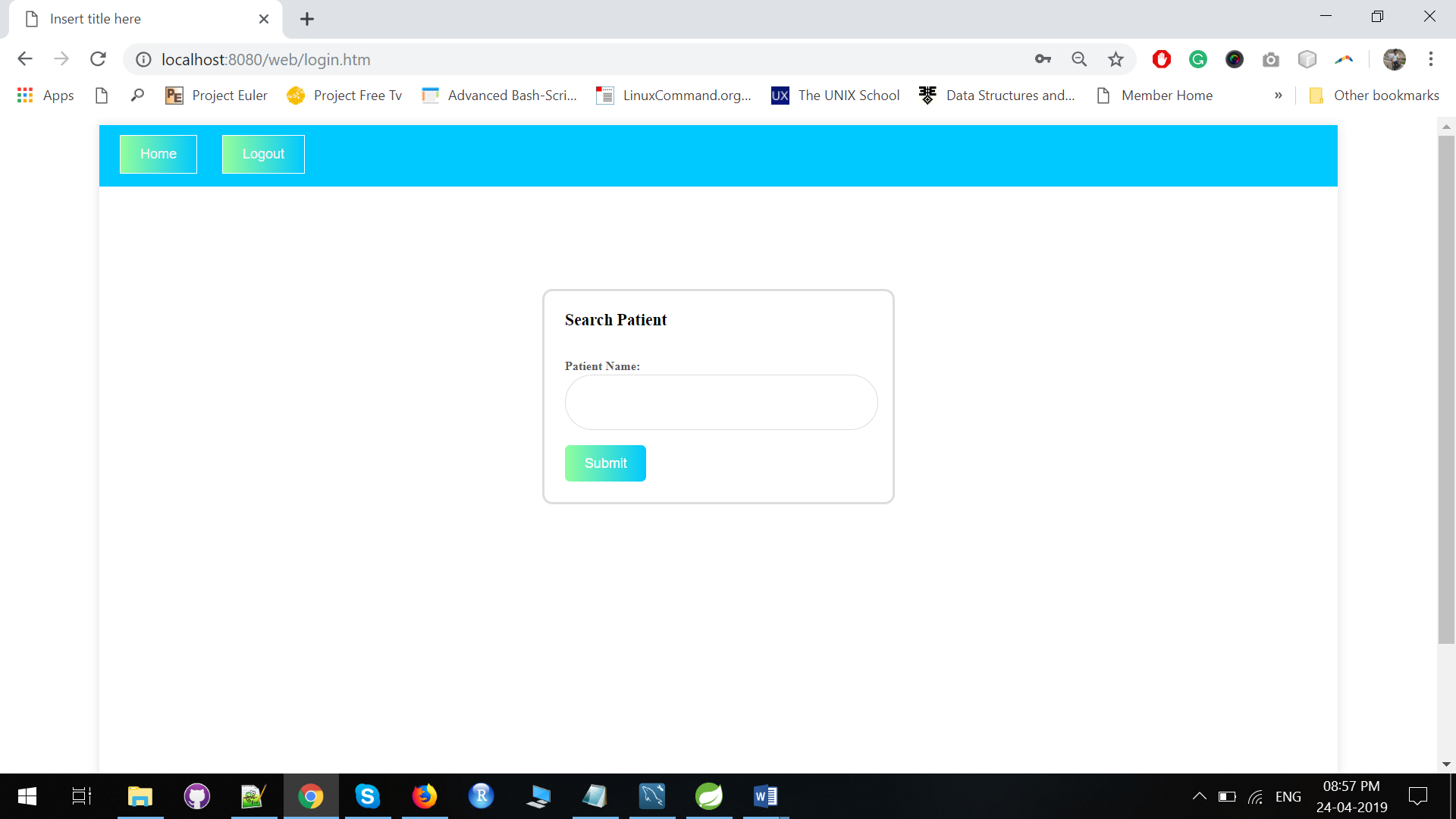
(4) Pharmacist:



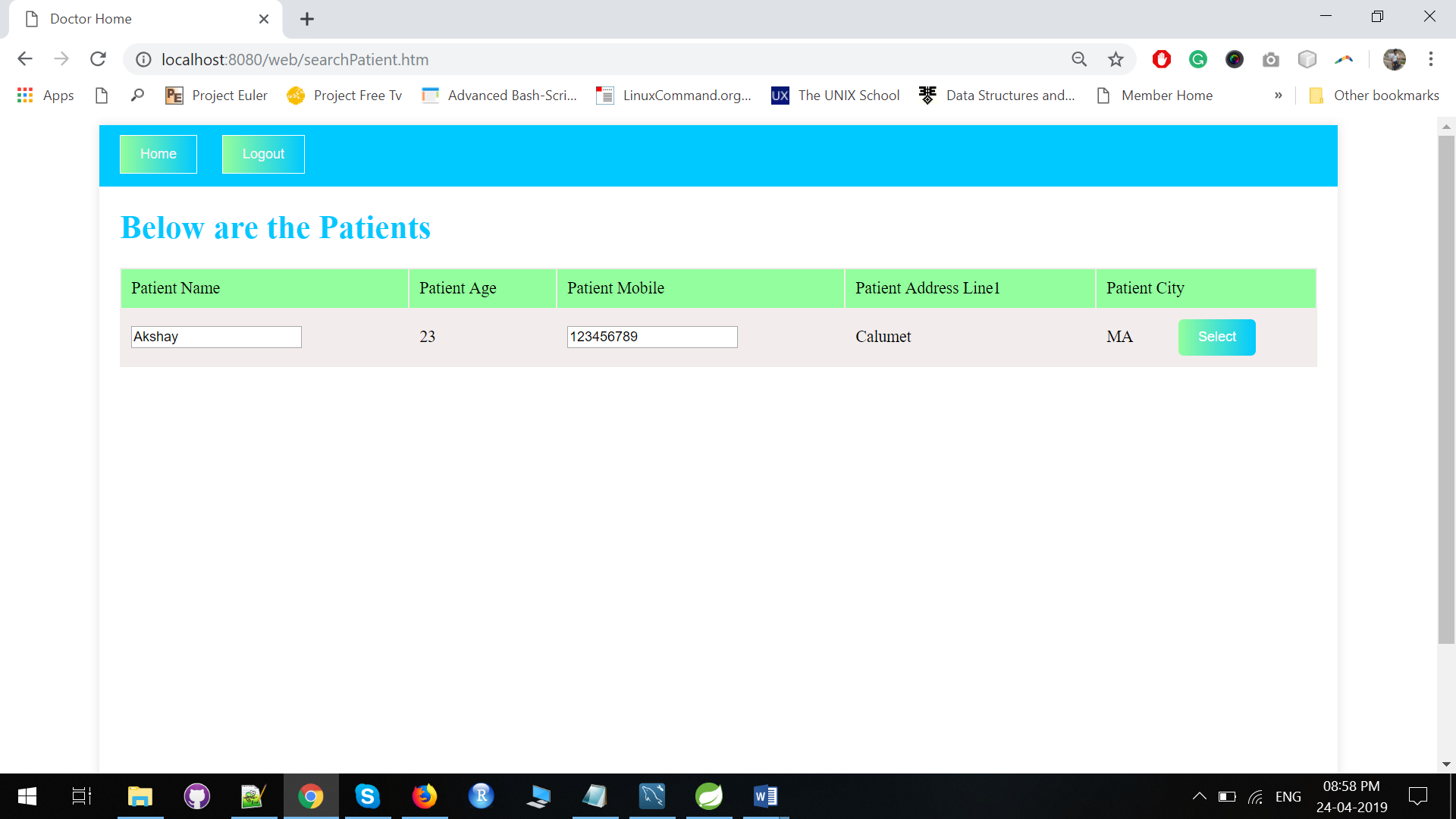
(5) Receptionist:



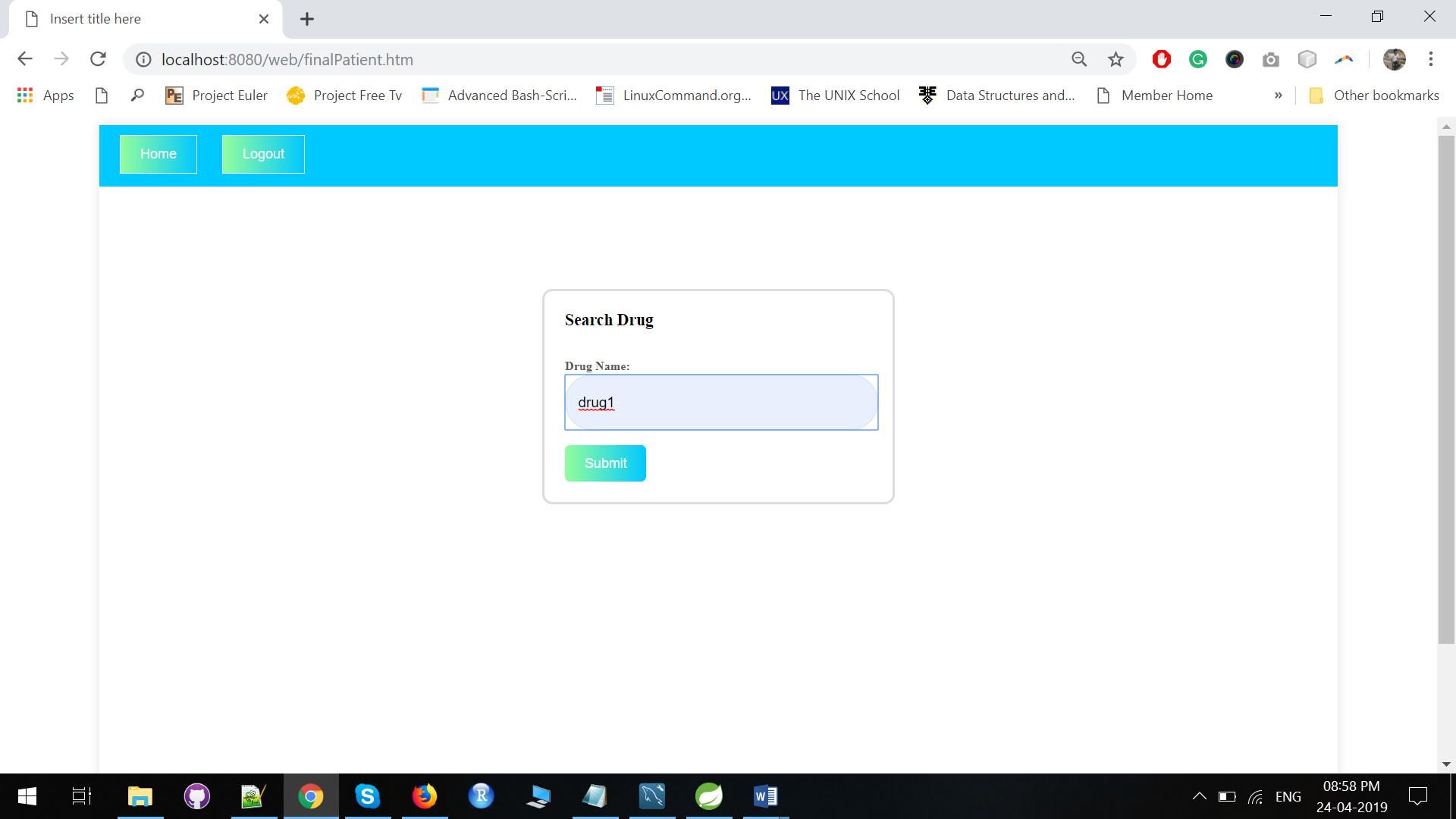
(6) Search Patient:



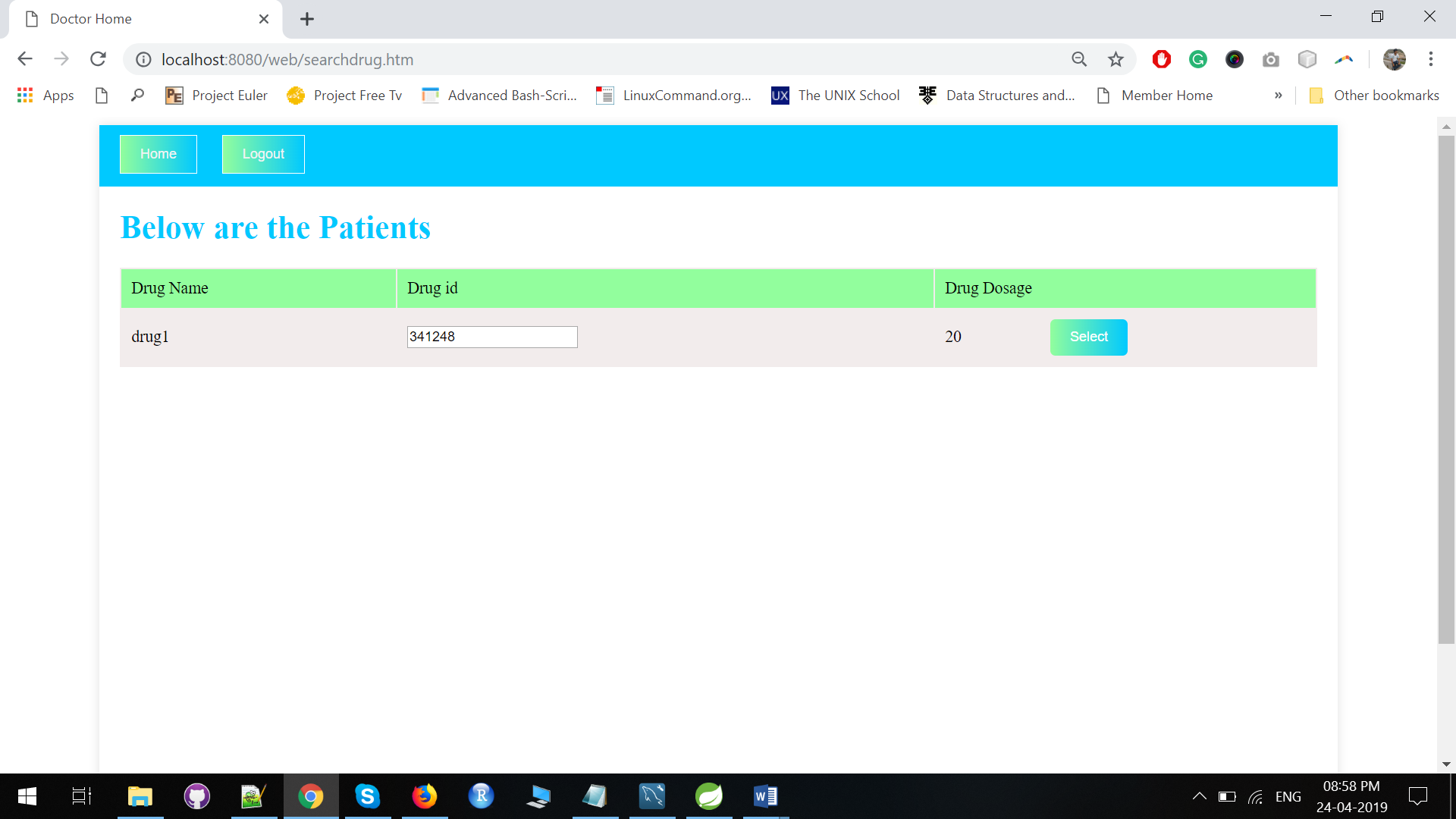
(7) Patient List:



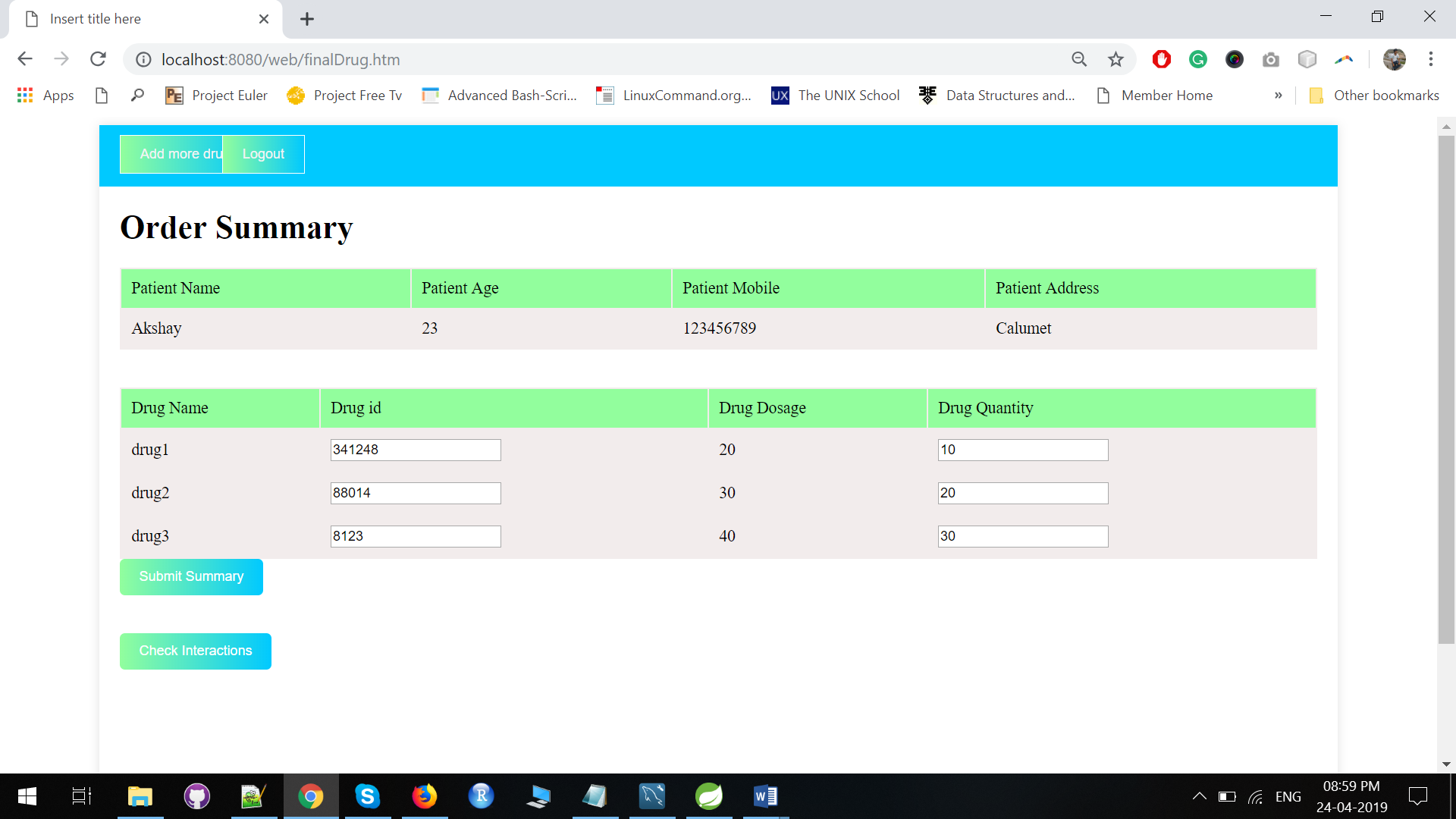
(8) Drug Search:



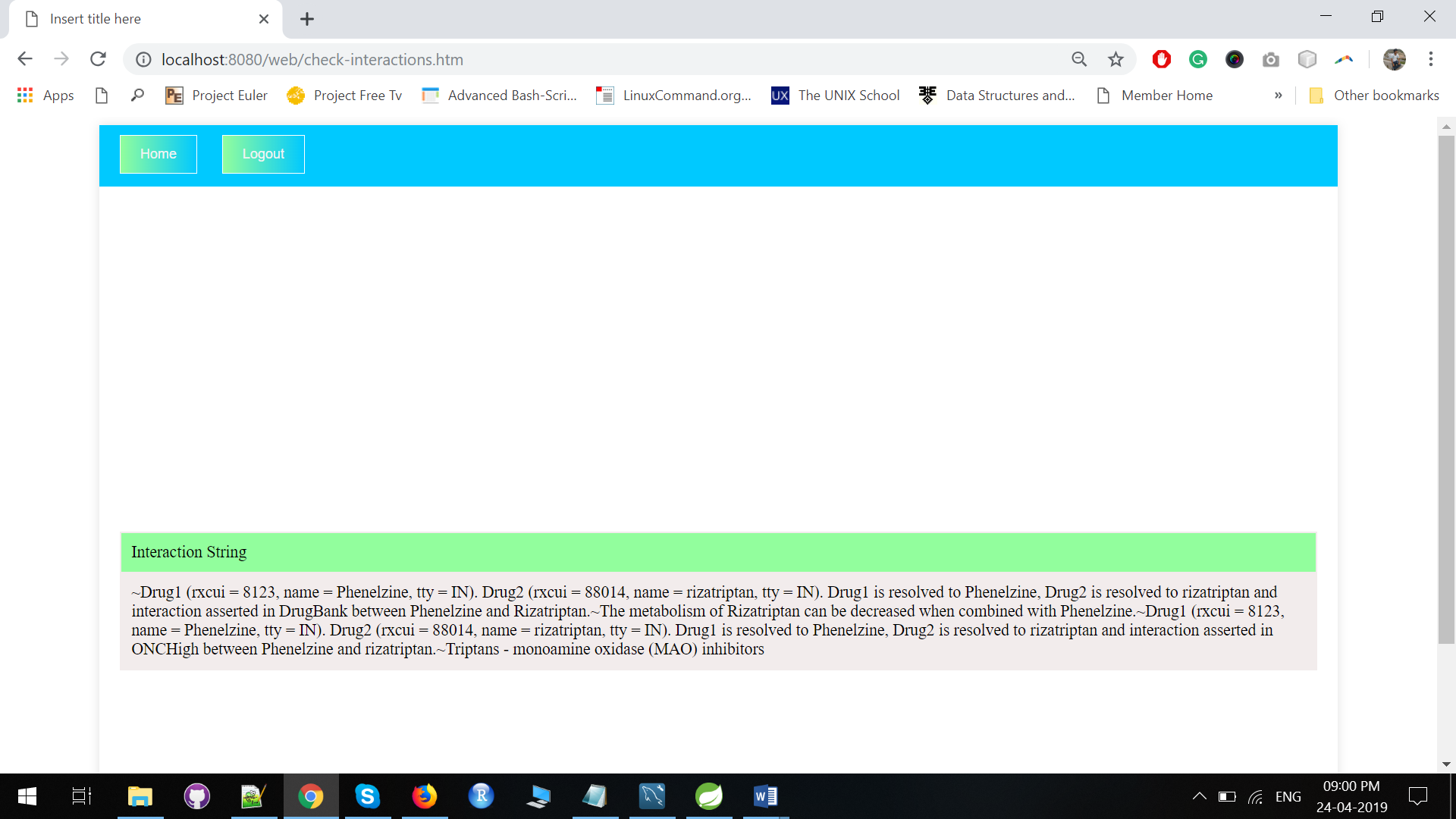
(9) Drug List:



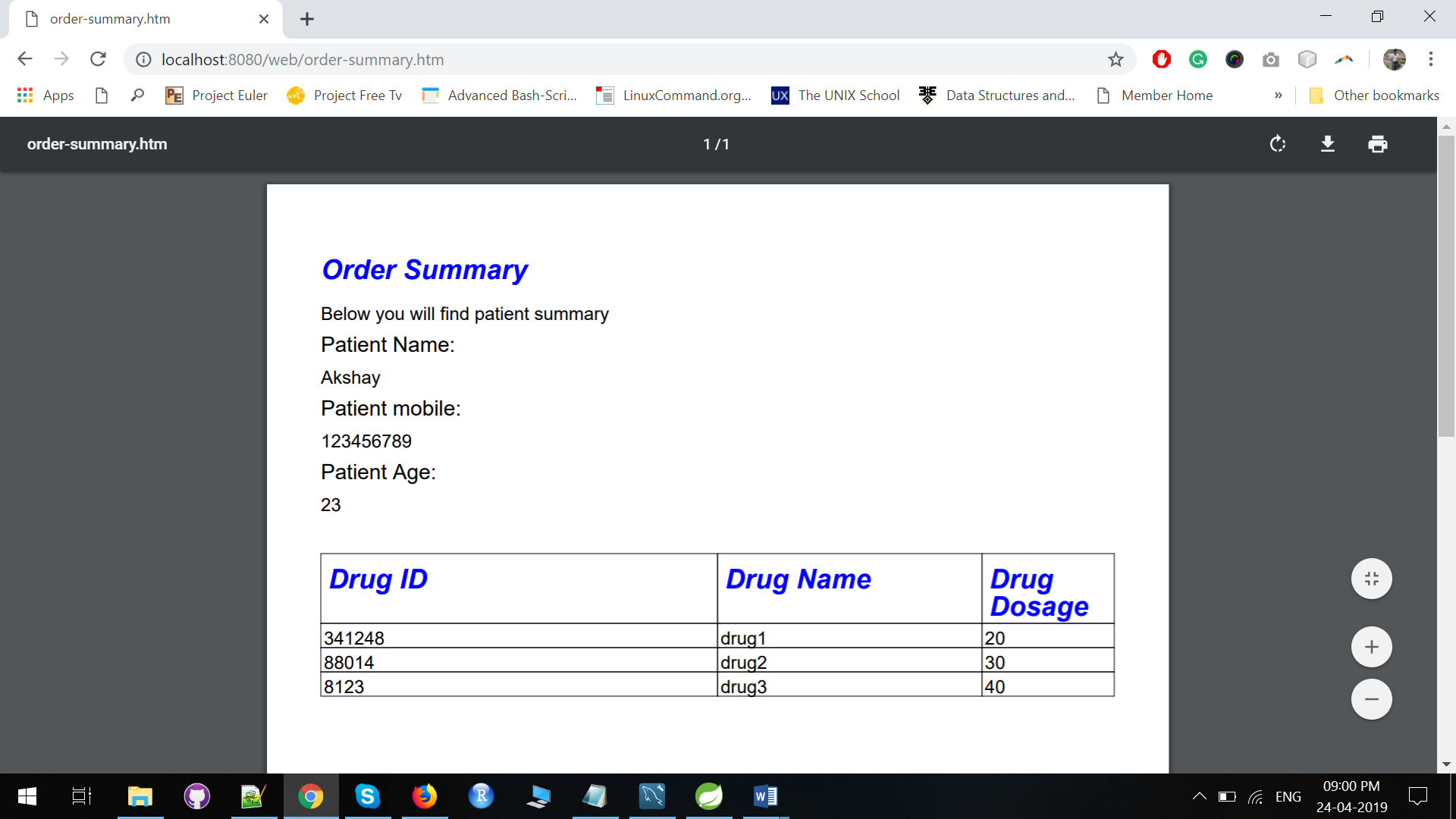
(10) Order Summary:



(11) Interaction View:



(12) PDF view of order summary:



**Controllers:**

package com.neu.web.controller;

import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.beans.factory.annotation.Qualifier;

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

import org.springframework.web.bind.WebDataBinder;

import org.springframework.web.bind.annotation.InitBinder;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.servlet.ModelAndView;

import com.neu.web.pojo.\*;

import com.neu.web.validator.userValidation;

import com.neu.web.dao.\*;

@Controller

public class RegisterUserController {

@Autowired

UserDAO userDao;

@Autowired

@Qualifier("userValidator")

userValidation validator;

@InitBinder

private void initBinder(WebDataBinder binder) {

binder.setValidator(validator);

}

@RequestMapping(value = "/registerUser.htm", method = RequestMethod.POST)

public ModelAndView signUp(@ModelAttribute("user")User user,BindingResult result,HttpSession session) throws Exception{

validator.validate(user, result);

if (result.hasErrors()) {

return new ModelAndView("register", "user", user);

}

User u = userDao.getUser(user.getUsername());

if(u == null)

{

userDao.add(user);

return new ModelAndView("login");

}

else

{

return new ModelAndView("home");

}

}

}

package com.neu.web.controller;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.MalformedURLException;

import java.net.URL;

import java.util.ArrayList;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import org.json.simple.JSONArray;

import org.json.simple.JSONObject;

import org.json.simple.parser.JSONParser;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.expression.ParseException;

import org.springframework.http.HttpRequest;

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

import org.springframework.validation.BindingResult;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.servlet.ModelAndView;

import org.springframework.web.servlet.View;

import com.neu.web.dao.DrugDAO;

import com.neu.web.dao.PatientDAO;

import com.neu.web.dao.UserDAO;

import com.neu.web.pojo.Drug;

import com.neu.web.pojo.Patient;

import com.neu.web.pojo.PdfReport;

import com.neu.web.pojo.User;

@Controller

public class DoctorController {

@Autowired

PatientDAO patientDao;

@Autowired

UserDAO userDao;

@Autowired

DrugDAO drugDao;

@RequestMapping(value = "/searchPatient.htm", method = RequestMethod.POST)

public ModelAndView signUp(HttpServletRequest request,HttpSession session) throws Exception{

User tempuser = (User)session.getAttribute("user");

if(tempuser == null)

System.out.println("logout user trying");

ModelAndView mv = new ModelAndView();

System.out.println(request.getParameter("pname"));

List<Patient> plist = patientDao.getAllPatient(request.getParameter("pname"));

mv.addObject("patients", plist);

System.out.println(plist);

if (plist.isEmpty())

System.out.println("The returned list is empty");

for (Patient patient : plist) {

System.out.println("The matching patient are : " + patient.getPatientName());

}

mv.setViewName("doctor");

return mv;

}

@RequestMapping(value = "/finalPatient.htm", method = RequestMethod.POST)

protected ModelAndView addPatient(HttpServletRequest request,HttpSession session) throws Exception {

try {

ModelAndView mv = new ModelAndView();

List<Drug> dlist = new ArrayList<Drug>() ;

session.setAttribute("dlist", dlist);

//HttpSession session = (HttpSession) request.getSession();

User u = (User) session.getAttribute("user");

String pmobile = request.getParameter("pmobile");

Patient current\_patient = patientDao.getPatient(pmobile);

session.setAttribute("current\_patient", current\_patient);

//mv.addObject("jobs", jobsDao.listCompany(u.getCompany()));

//mv.addObject("joblist", job);

mv.setViewName("search-drug");

return mv;

} catch (Exception e) {

System.out.println("Exception: " + e.getMessage());

return new ModelAndView("error", "errorMessage", "error while adding job");

}

}

@RequestMapping(value = "/searchdrug.htm", method = RequestMethod.POST)

public ModelAndView searchDrug(HttpServletRequest request,HttpSession session) throws Exception{

ModelAndView mv = new ModelAndView();

System.out.println(request.getParameter("dname"));

List<Drug> dlist = drugDao.getAllDrug(request.getParameter("dname"));

mv.addObject("drugList", dlist);

System.out.println(dlist);

if (dlist.isEmpty())

System.out.println("The returned list is empty");

for (Drug drug : dlist) {

System.out.println("The matching patient are : " + drug.getDrugName());

}

mv.setViewName("drug-inter");

return mv;

}

@RequestMapping(value = "/finalDrug.htm", method = RequestMethod.POST)

protected ModelAndView adddrug(HttpServletRequest request, HttpSession session) throws Exception {

ModelAndView mv = new ModelAndView();

System.out.println(request.getParameter("drugid"));

Drug drug = drugDao.getDrug(request.getParameter("drugid"));

//session.setAttribute("dlist", dlist);

if(drug != null)

{

List<Drug> drugList = (List<Drug>) session.getAttribute("dlist");

drugList.add(drug);

session.setAttribute("dlist", drugList);

System.out.println("Drug Added");

} else {

return new ModelAndView("error", "errorMessage", "error while adding job");

}

mv.addObject("finalPatient",session.getAttribute("current\_patient"));

mv.addObject("drugList",session.getAttribute("dlist"));

mv.setViewName("patient-order");

return mv;

}

@RequestMapping(value = "/add-drug.htm", method = RequestMethod.POST)

protected ModelAndView moreDrug(HttpServletRequest request, HttpSession session) throws Exception {

return new ModelAndView("search-drug");

}

@RequestMapping(value = "/check-interactions.htm", method = RequestMethod.POST)

protected ModelAndView checkInteractions(HttpServletRequest request, HttpSession session) throws Exception {

ModelAndView mv = new ModelAndView();

String interStr = "";

List<Drug> dlist = (List<Drug>)session.getAttribute("dlist");

System.out.println("size"+dlist.size());

for (int i=0;i<dlist.size();i++) {

Drug drug = (Drug)dlist.get(i);

interStr += "+" + drug.getDrugcomid();

System.out.println(i+ " "+interStr);

}

try

{

interStr = interStr.substring(1);

}catch(StringIndexOutOfBoundsException e){

return new ModelAndView("error", "errorMessage", "error while checking interactions");

}

System.out.println("done and dusted");

String url = "https://rxnav.nlm.nih.gov/REST/interaction/list.json?rxcuis="+interStr;

//String url = "https://rxnav.nlm.nih.gov/REST/interaction/list.json?rxcuis=123+345";

String finalInteractions = "";

URL obj;

try {

obj = new URL(url);

HttpURLConnection con = (HttpURLConnection) obj.openConnection();

con.setRequestMethod("GET");

int responseCode = con.getResponseCode();

System.out.println(responseCode);

BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));

String inputLine;

StringBuffer response = new StringBuffer();

while ((inputLine = in.readLine()) != null) {

response.append(inputLine);

}

in.close();

//print result

//System.out.println(response.toString());

Object jobj = new JSONParser().parse(response.toString());

JSONObject jo = (JSONObject) jobj;

//JSONObject job = (JSONObject) jo.get("nlmDisclaimer");

JSONArray address = (JSONArray) jo.get("fullInteractionTypeGroup");

for (Object o : address) {

JSONObject objt = (JSONObject) o;

String strTemp = (String) objt.get("sourceDisclaimer");

//System.out.println(strTemp);

JSONArray add2 = (JSONArray) objt.get("fullInteractionType");

for (Object obj2 : add2) {

JSONObject objTemp = (JSONObject) obj2;

String strTemp2 = (String) objTemp.get("comment");

finalInteractions = finalInteractions + "~" + strTemp2;

//System.out.println(strTemp2);

JSONArray add3 = (JSONArray) objTemp.get("interactionPair");

for (Object obj3 : add3) {

JSONObject objTemp3 = (JSONObject) obj3;

String strTemp3 = (String) objTemp3.get("description");

finalInteractions = finalInteractions + "~" + strTemp3;

if (!strTemp3.isEmpty()) {

break;

}

//System.out.println(strTemp3);

}

}

}

} catch (MalformedURLException ex) {

System.out.println("first");

return new ModelAndView("error", "errorMessage", "Please enter proper drug ID");

} catch (IOException ex) {

System.out.println("second");

return new ModelAndView("error", "errorMessage", "Connection Time-Out");

} catch (ParseException ex) {

System.out.println("third");

return new ModelAndView("error", "errorMessage", "API Output is not proper.");

} catch (NullPointerException ex) {

System.out.println(finalInteractions);

System.out.println("fourth");

return new ModelAndView("error", "errorMessage", "API did not give any interactions.");

}

Boolean interactionCheck;

if (finalInteractions.isEmpty()) {

interactionCheck = Boolean.FALSE;

}

else

{

interactionCheck = Boolean.TRUE;

//int cnt = patient.getInteractionCount();

//patient.setInteractionCount(cnt+1);

}

if(interactionCheck)

{

mv.addObject("interaction",finalInteractions);

mv.setViewName("show-interactions");

}else {

mv.setViewName("patient-order");

}

return mv;

}

// @RequestMapping(value = "/go-home.htm", method = RequestMethod.POST)

// protected ModelAndView goHome(HttpServletRequest request, HttpSession session) throws Exception {

// return new ModelAndView("search-drug");

// }

@RequestMapping(value = "/logout.htm", method = RequestMethod.POST)

protected ModelAndView goLogout(HttpServletRequest request, HttpSession session) throws Exception {

System.out.println("Session invalidated");

session.setAttribute("user", null);

session.invalidate();

return new ModelAndView("login");

}

@RequestMapping(value = "/doctor-home.htm", method = RequestMethod.POST)

protected ModelAndView goHomePage(HttpServletRequest request, HttpSession session) throws Exception {

return new ModelAndView("searchPatient");

}

@RequestMapping(value = "/order-summary.htm", method = RequestMethod.POST)

public ModelAndView showPdfReport(@ModelAttribute("patient") Patient patient,ModelMap model,BindingResult result,HttpServletRequest request,HttpSession session) throws Exception

{

model.addAttribute("finalPatient",session.getAttribute("current\_patient"));

model.addAttribute("drugList",session.getAttribute("dlist"));

View v = new PdfReport();

return new ModelAndView(v);

}

}

package com.neu.web.controller;

import java.text.DateFormat;

import java.util.Date;

import java.util.Locale;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

/\*\*

\* Handles requests for the application home page.

\*/

@Controller

public class HomeController {

private static final Logger logger = LoggerFactory.getLogger(HomeController.class);

/\*\*

\* Simply selects the home view to render by returning its name.

\*/

@RequestMapping(value = "/", method = RequestMethod.GET)

public String home(Locale locale, Model model) {

logger.info("Welcome home! The client locale is {}.", locale);

Date date = new Date();

DateFormat dateFormat = DateFormat.getDateTimeInstance(DateFormat.LONG, DateFormat.LONG, locale);

String formattedDate = dateFormat.format(date);

model.addAttribute("serverTime", formattedDate );

return "home";

}

}

package com.neu.web.controller;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.servlet.ModelAndView;

import com.neu.web.dao.UserDAO;

import com.neu.web.pojo.\*;

@Controller

public class LoginUserController {

@Autowired

UserDAO userDao;

@RequestMapping(value="/login.htm", method=RequestMethod.POST)

public String userLogin(HttpServletRequest request,HttpSession session, Drug drug, Patient patient) throws Exception {

System.out.println((String)request.getParameter("username"));

User u = userDao.checkUser((String)request.getParameter("username"), (String)request.getParameter("password"));

if(u == null)

{

System.out.println("null user");

return "login";

}

else

{

session.setAttribute("user", u);

String role = u.getRoleType();

System.out.println(role);

if("D".equals(role)) {

return "searchPatient";

}

else if("P".equals(role)) {

System.out.println("inside pharmacy");

return "pharmacy";

}

else if("R".equals(role)) {

return "patient";

}

else {return "home";}

}

}

@RequestMapping(value = "/go-home.htm", method = RequestMethod.POST)

protected ModelAndView goHomePage(HttpServletRequest request, HttpSession session) throws Exception {

return new ModelAndView("login");

}

// @RequestMapping(value = "/logout.htm", method = RequestMethod.POST)

// protected ModelAndView goLogout(HttpServletRequest request, HttpSession session) throws Exception {

// session.invalidate();

// return new ModelAndView("login");

// }

}

package com.neu.web.controller;

import javax.servlet.http.HttpServletRequest;

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.servlet.ModelAndView;

import com.neu.web.pojo.\*;

@Controller

public class MainPageController {

@RequestMapping(value = "/login.htm", method = RequestMethod.GET)

public ModelAndView showForm(HttpServletRequest request, ModelMap model) {

//command object

//model.addAttribute("applicant", applicant);

request.setAttribute("action", "save");

return new ModelAndView("login");

}

@RequestMapping(value = "/register.htm", method = RequestMethod.GET)

public ModelAndView register(HttpServletRequest request, ModelMap model, User user) {

return new ModelAndView("register");

}

}

package com.neu.web.controller;

import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.beans.factory.annotation.Qualifier;

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

import org.springframework.web.bind.WebDataBinder;

import org.springframework.web.bind.annotation.InitBinder;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.servlet.ModelAndView;

import com.neu.web.pojo.\*;

import com.neu.web.validator.DrugValidation;

import com.neu.web.dao.\*;

@Controller

public class PharmacyController {

@Autowired

DrugDAO drugDao;

@Autowired

@Qualifier("DrugValidation")

DrugValidation validator;

@InitBinder

private void initBinder(WebDataBinder binder) {

binder.setValidator(validator);

}

@RequestMapping(value = "/pharmacy.htm", method = RequestMethod.POST)

public ModelAndView drugAdd(@ModelAttribute("drug")Drug drug,BindingResult result,HttpSession session) throws Exception{

System.out.println("In Pharma controller");

//Drug d = null;

validator.validate(drug, result);

if (result.hasErrors()) {

return new ModelAndView("pharmacy", "drug", drug);

}

Drug d = drugDao.getDrug(drug.getDrugcomid());

if(d == null)

{

System.out.println("In Drug add");

drugDao.add(drug);

return new ModelAndView("drug-added");

}

else

return new ModelAndView("pharmacy");

}

}

package com.neu.web.controller;

import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.beans.factory.annotation.Qualifier;

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

import org.springframework.web.bind.WebDataBinder;

import org.springframework.web.bind.annotation.InitBinder;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.servlet.ModelAndView;

import com.neu.web.dao.PatientDAO;

import com.neu.web.pojo.Patient;

import com.neu.web.validator.PatientValidation;

@Controller

public class RegisterPatientController {

@Autowired

PatientDAO patientDao;

@Autowired

@Qualifier("PatientValidation")

PatientValidation validator;

@InitBinder

private void initBinder(WebDataBinder binder) {

binder.setValidator(validator);

}

@RequestMapping(value = "/patient.htm", method = RequestMethod.POST)

public ModelAndView signUp(@ModelAttribute("patient")Patient patient,BindingResult result,HttpSession session) throws Exception{

validator.validate(patient, result);

if (result.hasErrors()) {

return new ModelAndView("patient", "patient", patient);

}

Patient p = patientDao.getPatient(patient.getPatientMobile());

if(p == null)

{

patientDao.add(patient);

return new ModelAndView("patient-added");

}

else

{

return new ModelAndView("patient");

}

}

}

**DAOs:**

package com.neu.web.dao;

import java.util.logging.Level;

import java.util.logging.Logger;

import org.hibernate.HibernateException;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

import com.neu.web.pojo.User;

//package com.me.book.dao;

import java.util.List;

public class DAO {

private static final Logger log = Logger.getAnonymousLogger();

private static final ThreadLocal sessionThread = new ThreadLocal();

public static final SessionFactory sessionFactory = new Configuration().configure("hibernate.cfg.xml").buildSessionFactory();

public static Session getSession() {

Session session = (Session) sessionThread.get();

if (session == null) {

session = sessionFactory.openSession();

sessionThread.set(session);

}

return session;

}

protected void begin() {

getSession().beginTransaction();

}

protected void commit() {

getSession().getTransaction().commit();

}

protected void rollback() {

try {

getSession().getTransaction().rollback();

} catch (HibernateException e) {

log.log(Level.WARNING, "Cannot rollback", e);

}

try {

getSession().close();

} catch (HibernateException e) {

log.log(Level.WARNING, "Cannot close", e);

}

sessionThread.set(null);

}

public static void close() {

getSession().close();

sessionThread.set(null);

}

}

package com.neu.web.dao;

import java.util.logging.Level;

import java.util.logging.Logger;

import org.hibernate.HibernateException;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

import com.neu.web.pojo.User;

//package com.me.book.dao;

import java.util.List;

public class DAO {

private static final Logger log = Logger.getAnonymousLogger();

private static final ThreadLocal sessionThread = new ThreadLocal();

public static final SessionFactory sessionFactory = new Configuration().configure("hibernate.cfg.xml").buildSessionFactory();

public static Session getSession() {

Session session = (Session) sessionThread.get();

if (session == null) {

session = sessionFactory.openSession();

sessionThread.set(session);

}

return session;

}

protected void begin() {

getSession().beginTransaction();

}

protected void commit() {

getSession().getTransaction().commit();

}

protected void rollback() {

try {

getSession().getTransaction().rollback();

} catch (HibernateException e) {

log.log(Level.WARNING, "Cannot rollback", e);

}

try {

getSession().close();

} catch (HibernateException e) {

log.log(Level.WARNING, "Cannot close", e);

}

sessionThread.set(null);

}

public static void close() {

getSession().close();

sessionThread.set(null);

}

}

package com.neu.web.dao;

import java.util.List;

import org.hibernate.HibernateException;

import org.hibernate.Query;

import com.neu.web.pojo.Patient;

public class PatientDAO extends DAO {

public void add(Patient patient) {

begin();

getSession().save(patient);

commit();

close();

}

public void edit(Patient patient) {

begin();

getSession().update(patient);

this.getSession().flush();

commit();

close();

}

public void delete(Patient patient) {

begin();

getSession().delete(patient);

commit();

close();

}

public Patient getPatient(String mobile) throws Exception {

System.out.println("Inside get Patient ");

System.out.println("Patient Number : " + mobile);

Patient patient = null;

try

{

begin();

Query q = getSession().createQuery("from Patient where patientMobile = :mobile");

q.setString("mobile", mobile);

System.out.println(""+q);

patient = (Patient) q.setMaxResults(1).uniqueResult();

if (patient == null)

System.out.println("PatientDAO : patient is not present in database");

else

System.out.println("PatientDAO : patient is available in database");

commit();

return patient;

} catch (HibernateException e) {

rollback();

throw new Exception("Could not get user " + mobile, e);

}

}

//

// public User checkUser(String username, String password) throws Exception {

// System.out.println("Checking Login");

// System.out.println("User in database : " + username);

// System.out.println("User in database : " + password);

//

// User user = null;

// try

// {

// begin();

// Query q = getSession().createQuery("from User where username = :username and password=:password");

// q.setString("username", username);

// q.setString("password", password);

// System.out.println(""+q);

// user = (User) q.setMaxResults(1).uniqueResult();

// if (user == null)

// System.out.println("UserDao : user login failed");

// else

// System.out.println("UserDao : user login successful");

// commit();

// return user;

// } catch (HibernateException e) {

// rollback();

// throw new Exception("Could not get user " + username, e);

// }

//

// }

public List<Patient> getAllPatient(String name) throws HibernateException{

try

{

begin();

Query q = getSession().createQuery("from Patient where patientName=:name");

q.setString("name", name);

List<Patient> list = q.list();

commit();

return list;

}

catch (HibernateException e) {

rollback();

throw new HibernateException("Could not list the Patient", e);

}

}

}

package com.neu.web.dao;

import java.util.List;

import org.hibernate.HibernateException;

import org.hibernate.Query;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.omg.CORBA.UserException;

import org.springframework.beans.factory.annotation.Autowired;

import com.neu.web.dao.DAO;

import com.neu.web.pojo.User;

public class UserDAO extends DAO {

public void add(User user) {

begin();

getSession().save(user);

commit();

close();

}

public void edit(User user) {

begin();

getSession().update(user);

this.getSession().flush();

commit();

close();

}

public void delete(User user) {

begin();

getSession().delete(user);

commit();

close();

}

public User getUser(String username) throws Exception {

System.out.println("Inside get User ");

System.out.println("UserDao : " + username);

User user = null;

try

{

begin();

Query q = getSession().createQuery("from User where username = :username");

q.setString("username", username);

System.out.println(""+q);

user = (User) q.setMaxResults(1).uniqueResult();

if (user == null)

System.out.println("UserDao : user is not present in database");

else

System.out.println("UserDao : user is available in database");

commit();

return user;

} catch (HibernateException e) {

rollback();

throw new Exception("Could not get user " + username, e);

}

}

public User checkUser(String username, String password) throws Exception {

System.out.println("Checking Login");

System.out.println("User in database : " + username);

System.out.println("User in database : " + password);

User user = null;

try

{

begin();

Query q = getSession().createQuery("from User where username = :username and password=:password");

q.setString("username", username);

q.setString("password", password);

System.out.println(""+q);

user = (User) q.setMaxResults(1).uniqueResult();

if (user == null)

System.out.println("UserDao : user login failed");

else

System.out.println("UserDao : user login successful");

commit();

return user;

} catch (HibernateException e) {

rollback();

throw new Exception("Could not get user " + username, e);

}

}

public List getAllUser() {

return getSession().createQuery("from User").list();

}

}

package com.neu.web.pojo;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.OneToOne;

import javax.persistence.PrimaryKeyJoinColumn;

import org.hibernate.annotations.GenericGenerator;

import org.hibernate.annotations.Parameter;

@Entity

public class Address {

@Id

@GeneratedValue(generator = "generator")

@GenericGenerator(name = "generator", strategy = "foreign", parameters={@Parameter(name="property", value="patient")})

@Column(name = "addressID", unique = true, nullable = false)

private int addressID;

@Column

private String addressLine1;

@Column

private String addressLine2;

@Column

private String state;

@Column

private String city;

@Column

private String country;

@Column

private String zipcode;

@OneToOne

@PrimaryKeyJoinColumn

private Patient patient;

public int getAddressID() {

return addressID;

}

public void setAddressID(int addressID) {

this.addressID = addressID;

}

public String getAddressLine1() {

return addressLine1;

}

public void setAddressLine1(String addressLine1) {

this.addressLine1 = addressLine1;

}

public String getAddressLine2() {

return addressLine2;

}

public void setAddressLine2(String addressLine2) {

this.addressLine2 = addressLine2;

}

public String getState() {

return state;

}

public void setState(String state) {

this.state = state;

}

public String getCity() {

return city;

}

public void setCity(String city) {

this.city = city;

}

public String getCountry() {

return country;

}

public void setCountry(String country) {

this.country = country;

}

public String getZipcode() {

return zipcode;

}

public void setZipcode(String zipcode) {

this.zipcode = zipcode;

}

public Patient getPatient() {

return patient;

}

public void setPatient(Patient patient) {

this.patient = patient;

}

}

package com.neu.web.pojo;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Drug {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name="drugId",unique=true,nullable=false)

private long drugId;

@Column

private String drugcomid;

@Column

private String drugName;

@Column

private String dosageValue;

public Drug() {

}

public String getDrugcomid() {

return drugcomid;

}

public void setDrugcomid(String drugcomid) {

this.drugcomid = drugcomid;

}

public long getDrugId() {

return drugId;

}

public void setDrugId(int drugId) {

this.drugId = drugId;

}

public String getDrugName() {

return drugName;

}

public void setDrugName(String drugName) {

this.drugName = drugName;

}

public String getDosageValue() {

return dosageValue;

}

public void setDosageValue(String dosageValue) {

this.dosageValue = dosageValue;

}

}

package com.neu.web.pojo;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class OrderSummary {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name="orderId",unique=true,nullable=false)

private long orderId;

}

package com.neu.web.pojo;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Patient {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name="patientId",unique=true,nullable=false)

private int patientId;

@Column

private String patientName;

@Column

private String patientAge;

@Column

private String patientMobile;

@Column

private String addressLine1;

@Column

private String addressLine2;

@Column

private String state;

@Column

private String city;

@Column

private String country;

@Column

private String zipcode;

public String getAddressLine1() {

return addressLine1;

}

public void setAddressLine1(String addressLine1) {

this.addressLine1 = addressLine1;

}

public String getAddressLine2() {

return addressLine2;

}

public void setAddressLine2(String addressLine2) {

this.addressLine2 = addressLine2;

}

public String getState() {

return state;

}

public void setState(String state) {

this.state = state;

}

public String getCity() {

return city;

}

public void setCity(String city) {

this.city = city;

}

public String getCountry() {

return country;

}

public void setCountry(String country) {

this.country = country;

}

public String getZipcode() {

return zipcode;

}

public void setZipcode(String zipcode) {

this.zipcode = zipcode;

}

// @OneToOne(mappedBy="patient")

// @Cascade(value=org.hibernate.annotations.CascadeType.SAVE\_UPDATE)

// private Address address;

public int getPatientId() {

return patientId;

}

public void setPatientId(int patientId) {

this.patientId = patientId;

}

public String getPatientName() {

return patientName;

}

public void setPatientName(String patientName) {

this.patientName = patientName;

}

public String getPatientAge() {

return patientAge;

}

public void setPatientAge(String patientAge) {

this.patientAge = patientAge;

}

public String getPatientMobile() {

return patientMobile;

}

public void setPatientMobile(String patientMobile) {

this.patientMobile = patientMobile;

}

// public Address getAddress() {

// return address;

// }

// public void setAddress(Address address) {

// this.address = address;

// }

//

}

package com.neu.web.pojo;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Patient {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name="patientId",unique=true,nullable=false)

private int patientId;

@Column

private String patientName;

@Column

private String patientAge;

@Column

private String patientMobile;

@Column

private String addressLine1;

@Column

private String addressLine2;

@Column

private String state;

@Column

private String city;

@Column

private String country;

@Column

private String zipcode;

public String getAddressLine1() {

return addressLine1;

}

public void setAddressLine1(String addressLine1) {

this.addressLine1 = addressLine1;

}

public String getAddressLine2() {

return addressLine2;

}

public void setAddressLine2(String addressLine2) {

this.addressLine2 = addressLine2;

}

public String getState() {

return state;

}

public void setState(String state) {

this.state = state;

}

public String getCity() {

return city;

}

public void setCity(String city) {

this.city = city;

}

public String getCountry() {

return country;

}

public void setCountry(String country) {

this.country = country;

}

public String getZipcode() {

return zipcode;

}

public void setZipcode(String zipcode) {

this.zipcode = zipcode;

}

// @OneToOne(mappedBy="patient")

// @Cascade(value=org.hibernate.annotations.CascadeType.SAVE\_UPDATE)

// private Address address;

public int getPatientId() {

return patientId;

}

public void setPatientId(int patientId) {

this.patientId = patientId;

}

public String getPatientName() {

return patientName;

}

public void setPatientName(String patientName) {

this.patientName = patientName;

}

public String getPatientAge() {

return patientAge;

}

public void setPatientAge(String patientAge) {

this.patientAge = patientAge;

}

public String getPatientMobile() {

return patientMobile;

}

public void setPatientMobile(String patientMobile) {

this.patientMobile = patientMobile;

}

// public Address getAddress() {

// return address;

// }

// public void setAddress(Address address) {

// this.address = address;

// }

//

}

package com.neu.web.pojo;

import javax.persistence.Column;

import javax.persistence.Id;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

@Entity

public class User {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name="userId",unique=true,nullable=false)

private long userId;

@Column

private String username;

@Column

private String firstName;

@Column

private String lastName;

@Column

private String mobileNumber;

@Column

private String password;

@Column

private String roleType;

public User() {

}

public long getUserId() {

return userId;

}

public void setUserId(long userId) {

this.userId = userId;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getFirstName() {

return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

public String getMobileNumber() {

return mobileNumber;

}

public void setMobileNumber(String mobileNumber) {

this.mobileNumber = mobileNumber;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getRoleType() {

return roleType;

}

public void setRoleType(String roleType) {

this.roleType = roleType;

}

}

**Validator:**

package com.neu.web.validator;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import org.springframework.validation.Errors;

import org.springframework.validation.ValidationUtils;

import org.springframework.validation.Validator;

import com.neu.web.pojo.Drug;

public class DrugValidation implements Validator {

public boolean supports(Class dClass) {

return dClass.equals(Drug.class);

}

private Pattern pattern;

private Matcher matcher;

private static final

String STRING\_PATTERN2 = "[a-zA-Z]+";

String STRING\_PATTERN3 = "[0-9]+";

//String ZIP\_PATTERN3 = "^[0-9]\\d{5}$";

public void validate(Object obj, Errors errors) {

Drug drug = (Drug) obj;

ValidationUtils.rejectIfEmptyOrWhitespace(errors, "drugcomid", "error.invalid.drug", "Drug ID Required");

if (!(drug.getDrugcomid() != null && drug.getDrugcomid().isEmpty())) {

pattern = Pattern.compile(STRING\_PATTERN3);

matcher = pattern.matcher(drug.getDrugcomid());

if (!matcher.matches()) {

errors.rejectValue("drugcomid", "drugcomid.containNonChar", "Enter a valid Drug ID");

}

}

ValidationUtils.rejectIfEmptyOrWhitespace(errors, "drugName", "error.invalid.drug", "Drug Name Required");

if (!(drug.getDrugName() != null && drug.getDrugName().isEmpty())) {

pattern = Pattern.compile(STRING\_PATTERN2);

matcher = pattern.matcher(drug.getDrugName());

if (!matcher.matches()) {

errors.rejectValue("drugName", "drugName.containNonChar", "Enter a valid Drug name");

}

}

ValidationUtils.rejectIfEmptyOrWhitespace(errors, "dosageValue", "error.invalid.drug", "Drug Dosage Required");

if (!(drug.getDosageValue() != null && drug.getDosageValue().isEmpty())) {

pattern = Pattern.compile(STRING\_PATTERN3);

matcher = pattern.matcher(drug.getDosageValue());

if (!matcher.matches()) {

errors.rejectValue("dosageValue", "dosageValue.containNonChar", "Enter a valid Drug dosage");

}

}

}

}

package com.neu.web.validator;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import org.springframework.validation.Errors;

import org.springframework.validation.ValidationUtils;

import org.springframework.validation.Validator;

import com.neu.web.pojo.Patient;

public class PatientValidation implements Validator {

public boolean supports(Class pClass) {

return pClass.equals(Patient.class);

}

private Pattern pattern;

private Matcher matcher;

private static final

String STRING\_PATTERN1 = "[a-zA-Z]+";

String STRING\_PATTERN2 = "[a-zA-Z0-9-\\.]+";

String STRING\_PATTERN3 = "[0-9]+";

//String ZIP\_PATTERN3 = "^[0-9]\\d{5}$";

String ZIP\_PATTERN3 = "^[0-9]{5}(?:-[0-9]{4})?$";

String STRING\_PATTERN4 = "^[789]\\d{9}$";

public void validate(Object obj, Errors errors) {

Patient patient = (Patient) obj;

ValidationUtils.rejectIfEmptyOrWhitespace(errors, "patientName", "error.invalid.patient", "Patient Name Required");

if (!(patient.getPatientName() != null && patient.getPatientName().isEmpty())) {

pattern = Pattern.compile(STRING\_PATTERN1);

matcher = pattern.matcher(patient.getPatientName());

if (!matcher.matches()) {

errors.rejectValue("patientName", "patientName.containNonChar", "Enter a valid Patient Name");

}

}

ValidationUtils.rejectIfEmptyOrWhitespace(errors, "patientAge", "error.invalid.patient", "Patient Age Required");

if (!(patient.getPatientAge() != null && patient.getPatientAge().isEmpty())) {

pattern = Pattern.compile(STRING\_PATTERN3);

matcher = pattern.matcher(patient.getPatientAge());

if (!matcher.matches()) {

errors.rejectValue("patientAge", "patientAge.containNonChar", "Enter a valid Patient Age");

} else if (Integer.parseInt(patient.getPatientAge())>160 || Integer.parseInt(patient.getPatientAge())< 1) {

errors.rejectValue("patientAge", "patientAge.containMoreAge", "Enter a valid Patient Age under 160 years or above 1 year");

}

}

ValidationUtils.rejectIfEmptyOrWhitespace(errors, "patientMobile", "error.invalid.patient", "Patient Name Required");

if (!(patient.getPatientMobile() != null && patient.getPatientMobile().isEmpty())) {

pattern = Pattern.compile(STRING\_PATTERN4);

matcher = pattern.matcher(patient.getPatientMobile());

if (!matcher.matches()) {

errors.rejectValue("patientMobile", "patientMobile.containNonChar", "Enter a valid Patient Mobile");

}

}

ValidationUtils.rejectIfEmptyOrWhitespace(errors, "city", "error.invalid.patient", "City Required");

if (!(patient.getCity() != null && patient.getCity().isEmpty())) {

pattern = Pattern.compile(STRING\_PATTERN1);

matcher = pattern.matcher(patient.getCity());

if (!matcher.matches()) {

errors.rejectValue("city", "city.containNonChar", "Enter a valid City Name");

}

}

ValidationUtils.rejectIfEmptyOrWhitespace(errors, "zipcode", "error.invalid.patient", "Patient Name Required");

if (!(patient.getZipcode() != null && patient.getZipcode().isEmpty())) {

pattern = Pattern.compile(ZIP\_PATTERN3);

matcher = pattern.matcher(patient.getZipcode());

if (!matcher.matches()) {

errors.rejectValue("zipcode", "zipcode.containNonChar", "Enter a valid Zip Code");

}

}

}

}

package com.neu.web.validator;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import org.springframework.validation.Errors;

import org.springframework.validation.ValidationUtils;

import org.springframework.validation.Validator;

import com.neu.web.pojo.User;

public class userValidation implements Validator {

public boolean supports(Class uClass) {

return uClass.equals(User.class);

}

private Pattern pattern;

private Matcher matcher;

private static final

String STRING\_PATTERN = "[a-zA-Z]+";

String STRING\_PATTERN2 = "[a-zA-Z0-9-\\.]+";

//String STRING\_PATTERN3 = "\\(?\\d+\\)?[-.\\s]?\\d+[-.\\s]?\\d+";

String STRING\_PATTERN3 = "^[789]\\d{9}$";

public void validate(Object obj, Errors errors) {

User user = (User) obj;

ValidationUtils.rejectIfEmptyOrWhitespace(errors, "firstName", "error.invalid.user", "First Name Required");

if (!(user.getFirstName() != null && user.getFirstName().isEmpty())) {

pattern = Pattern.compile(STRING\_PATTERN);

matcher = pattern.matcher(user.getFirstName());

if (!matcher.matches()) {

errors.rejectValue("firstName", "firstName.containNonChar",

"Enter a valid first name");

}

}

ValidationUtils.rejectIfEmptyOrWhitespace(errors, "lastName", "error.invalid.user", "Last Name Required");

if (!(user.getLastName() != null && user.getLastName().isEmpty())) {

pattern = Pattern.compile(STRING\_PATTERN);

matcher = pattern.matcher(user.getLastName());

if (!matcher.matches()) {

errors.rejectValue("lastName", "lastName.containNonChar", "Enter a valid last name");

}

}

ValidationUtils.rejectIfEmptyOrWhitespace(errors, "username", "error.invalid.user", "User Name Required");

if (!(user.getUsername() != null && user.getUsername().isEmpty())) {

pattern = Pattern.compile(STRING\_PATTERN2);

matcher = pattern.matcher(user.getUsername());

if (!matcher.matches()) {

errors.rejectValue("username", "username.containNonChar",

"Enter a valid username");

}

}

ValidationUtils.rejectIfEmptyOrWhitespace(errors, "password", "error.invalid.password", "Password Required");

if (!(user.getPassword() != null && user.getPassword().isEmpty())) {

pattern = Pattern.compile(STRING\_PATTERN2);

matcher = pattern.matcher(user.getPassword());

if (!matcher.matches()) {

errors.rejectValue("password", "password.containNonChar",

"Enter a valid password of strings");

}

}

ValidationUtils.rejectIfEmptyOrWhitespace(errors, "mobileNumber", "error.invalid.mobileNumber", "Mobile Number Required");

if (!(user.getMobileNumber() != null && user.getMobileNumber().isEmpty())) {

pattern = Pattern.compile(STRING\_PATTERN3);

matcher = pattern.matcher(user.getMobileNumber());

if (!matcher.matches()) {

errors.rejectValue("mobileNumber", "mobileNumber.containNonChar",

"Enter a valid password of strings");

}

}

// ValidationUtils.rejectIfEmptyOrWhitespace(errors, "email.emailAddress", "error.invalid.email.emailAddress",

// "Email Required");

// if("NONE".equals(user.get())){

// errors.rejectValue("usertype", "error.invalid.usertype","User type is required");

// }

// check if user exists

}

}

**Configuration Files:**

**Hibernate.cfg.xml:**

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

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name=*"hibernate.connection.driver\_class"*>com.mysql.jdbc.Driver</property>

<property name=*"hibernate.connection.password"*>root</property>

<property name=*"hibernate.connection.url"*>jdbc:mysql://localhost:3306/hospital</property>

<property name=*"hibernate.connection.username"*>root</property>

<property name=*"hibernate.hbm2ddl.auto"*>update</property>

<property name=*"hibernate.dialect"*>org.hibernate.dialect.MySQL5Dialect</property>

<property name=*"hibernate.show\_sql"*>true</property>

<mapping class=*"com.neu.web.pojo.User"* />

<mapping class=*"com.neu.web.pojo.Drug"* />

<mapping class=*"com.neu.web.pojo.Patient"* />

<mapping class=*"com.neu.web.pojo.Address"* />

</session-factory>

</hibernate-configuration>

**Root-context.xml**

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

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans https://www.springframework.org/schema/beans/spring-beans.xsd"*>

<!-- Root Context: defines shared resources visible to all other web components -->

<bean id=*"userValidator"* class=*"com.neu.web.validator.userValidation"*></bean>

<bean id=*"DrugValidation"* class=*"com.neu.web.validator.DrugValidation"*></bean>

<bean id=*"PatientValidation"* class=*"com.neu.web.validator.PatientValidation"*></bean>

<bean id=*"userDAO"* class=*"com.neu.web.dao.UserDAO"* />

<bean id=*"drugDAO"* class=*"com.neu.web.dao.DrugDAO"* />

<bean id=*"patientDAO"* class=*"com.neu.web.dao.PatientDAO"* />

</beans>

**Views:**

<%@taglib uri=*"http://www.springframework.org/tags/form"* prefix=*"form"*%>

<%@taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"*%>

<%@taglib uri=*"http://java.sun.com/jsp/jstl/functions"* prefix=*"fn"*%>

<%@page contentType=*"text/html"* pageEncoding=*"UTF-8"*%>

<!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=UTF-8"*>

<title>Doctor Home</title>

<style>

*.main-wrapper*{

width: *1170px*;

margin: *0 auto*;

padding: *0px 20px*;

box-shadow: *0 0 10px #e3e3e3*;

height: *100%*;

min-height: *768px*;

}

*.header-top* {

background: *#00c9ff*;

padding: *10px 0*;

margin-left: *-20px*;

margin-right: *-20px*;

display: *block*;

height: *40px*;

}

*.banner-section*{

background: *url(slider-1.jpg) no-repeat*;

height: *300px*;

background-size: *cover*;

margin-left: *-20px*;

margin-right: *-20px*;

position: *relative*;

background-position: *0 37%*;

}

*.banner-over*{

position: *absolute*;

text-align: *center*;

transform: *translate(-50% , -50%)*;

left: *50%*;

top: *50%*;

}

**banner-over**{

font-size: *43px*;

}

*.header-top* **form** {

width: *100px*;

float: *left*;

}

*.header-top* **form** **input** {

border: *0*;

color: *red*;

/ background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*; /

background: *transparent*;

border: *0*;

padding: *10px 19px*;

color: *#fff*;

color: *#f4f4f4*;

cursor: *pointer*;

border: *1px solid #fff*;

border-radius: *0*;

margin-left: *20px*;

}

**table**{

background: *#f2ecec*;

}

**table** **tr** **td**{

border: *0*;

}

**table** *.heading-row* **td**{

padding: *10px*;

border: *0*;

}

**table** *.heading-row*{

background: *#92fe9d*;

}

**table** **tr** **td**{

padding: *10px*;

/\* border: 1px solid #c9c9c9;\*/

}

*.btn-success*{

background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*;

padding: *10px 19px*;

border: *0*;

border-radius: *5px*;

color: *#fff*;

}

</style>

</head>

<body>

<div class=*"main-wrapper"*>

<c:set var=*"contextPath"* value=*"*${pageContext.request.contextPath}*"* />

<c:set var=*"patientslist"* value=*"*${patients}*"* />

<div class=*"header-top"*>

<form action=*"doctor-home.htm"* method=*"post"*>

<input class=*"btn btn-success"* type=*"submit"* name=*"action"* value=*"Home"*>

</form>

<form action=*"logout.htm"* method=*"post"*>

<input class=*"btn btn-success"* type=*"submit"* name=*"action"* value=*"Logout"*>

</form>

</div>

<h1 style="color: *#00c9ff*;">Below are the Patients </h1>

<table width=*"100%"*>

<tr class=*"heading-row"*>

<td id=*"td1"*> Patient Name</td>

<td id=*"td1"*> Patient Age </td>

<td id=*"td1"*> Patient Mobile</td>

<td id=*"td1"*> Patient Address Line1</td>

<td id=*"td1"* colspan=*"2"*> Patient City </td>

</tr>

<c:forEach items=*"*${patientslist}*"* var=*"patient"*>

<form:form action=*"finalPatient.htm"* commandName=*"patientslist"* method=*"post"*>

<tr>

<td><input type=*"text"* name=*"pname"* value=*"*${patient.patientName}*"* readonly></td>

<td>${patient.patientAge}</td>

<td><input type=*"text"* name=*"pmobile"* value=*"*${patient.patientMobile}*"* readonly></td>

<td>${patient.addressLine1}</td>

<td>${patient.state}</td>

<td><input class=*"btn btn-success"* type=*"submit"* name=*"action"* value=*"Select"*></td>

</tr>

</form:form>

</c:forEach>

</table>

</div>

</body>

</html>

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

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

<style>

*.header-top* **form** {

width: *100px*;

float: *left*;

}

*.header-top* **form** **input**{

border: *0*;

color: *red*;

/\* background: rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0;\*/

background: *transparent*;

border: *0*;

padding: *10px 19px*;

color: *#fff*;

color: *#121212*;

cursor: *pointer*;

}

</style>

</head>

<body>

<div class=*"header-top "*>

<form action=*"logout.htm"* method=*"post"*>

<input class=*"btn btn-success"* type=*"submit"* name=*"action"* value=*"Logout"*>

</form>

</div></br></br>

<h1>Drug Added Successfully</h1></br></br>

</body>

</html>

<%@taglib uri=*"http://www.springframework.org/tags/form"* prefix=*"form"*%>

<%@taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"*%>

<%@taglib uri=*"http://java.sun.com/jsp/jstl/functions"* prefix=*"fn"*%>

<%@page contentType=*"text/html"* pageEncoding=*"UTF-8"*%>

<!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=UTF-8"*>

<title>Doctor Home</title>

<style>

*.main-wrapper*{

width: *1170px*;

margin: *0 auto*;

padding: *0px 20px*;

box-shadow: *0 0 10px #e3e3e3*;

height: *100%*;

min-height: *768px*;

}

*.header-top* {

background: *#00c9ff*;

padding: *10px 0*;

margin-left: *-20px*;

margin-right: *-20px*;

display: *block*;

height: *40px*;

}

*.banner-section*{

background: *url(slider-1.jpg) no-repeat*;

height: *300px*;

background-size: *cover*;

margin-left: *-20px*;

margin-right: *-20px*;

position: *relative*;

background-position: *0 37%*;

}

*.banner-over*{

position: *absolute*;

color: *#fff*;

text-align: *center*;

transform: *translate(-50% , -50%)*;

left: *50%*;

top: *50%*;

}

**banner-over**{

font-size: *43px*;

}

*.header-top* **form** {

width: *100px*;

float: *left*;

}

*.header-top* **form** **input** {

border: *0*;

color: *red*;

/ background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*; /

background: *transparent*;

border: *0*;

padding: *10px 19px*;

color: *#fff*;

color: *#f4f4f4*;

cursor: *pointer*;

border: *1px solid #fff*;

border-radius: *0*;

margin-left: *20px*;

}

**table**{

background: *#f2ecec*;

}

**table** **tr** **td**{

border: *0*;

}

**table** *.heading-row* **td**{

padding: *10px*;

border: *0*;

}

**table** *.heading-row*{

background: *#92fe9d*;

}

**table** **tr** **td**{

padding: *10px*;

/\* border: 1px solid #c9c9c9;\*/

}

*.btn-success*{

background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*;

padding: *10px 19px*;

border: *0*;

border-radius: *5px*;

color: *#fff*;

}

</style>

</head>

<body>

<div class=*"main-wrapper"*>

<c:set var=*"contextPath"* value=*"*${pageContext.request.contextPath}*"* />

<c:set var=*"patientslist"* value=*"*${patients}*"* />

<div class=*"header-top"*>

<form action=*"doctor-home.htm"* method=*"post"*>

<input class = *"btn btn-success"* type=*"submit"* name=*"action"* value=*"Home"*>

</form>

<form action=*"logout.htm"* method=*"post"*>

<input class=*"btn btn-success"* type=*"submit"* name=*"action"* value=*"Logout"*>

</form>

</div>

<h1 style="color: *#00c9ff*;">Below are the Drugs </h1>

<table width=*"100%"*>

<tr class=*"heading-row"*>

<td id=*"td1"*> Drug Name</td>

<td id=*"td1"*> Drug id </td>

<td id=*"td1"* colspan=*"2"*> Drug Dosage</td>

</tr>

<c:forEach items=*"*${drugList}*"* var=*"drug"*>

<form:form action=*"finalDrug.htm"* commandName=*"druglist"* method=*"post"*>

<tr>

<td>${drug.drugName}</td>

<td><input type=*"text"* name=*"drugid"* value=*"*${drug.drugcomid}*"* readonly></td>

<td>${drug.dosageValue}</td>

<td><input class = *"btn btn-success"* type=*"submit"* name=*"action"* value=*"Select"*></td>

</tr>

</form:form>

</c:forEach>

</table>

</div>

</body>

</html>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@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=UTF-8"*>

<title>Error Page</title>

<style>

*.main-wrapper*{

width: *1170px*;

margin: *0 auto*;

padding: *0px 20px*;

box-shadow: *0 0 10px #e3e3e3*;

height: *100%*;

min-height: *768px*;

}

*.header-top* {

background: *#00c9ff*;

padding: *10px 0*;

margin-left: *-20px*;

margin-right: *-20px*;

display: *block*;

height: *40px*;

}

*.banner-section*{

background: *url(slider-1.jpg) no-repeat*;

height: *300px*;

background-size: *cover*;

margin-left: *-20px*;

margin-right: *-20px*;

position: *relative*;

background-position: *0 37%*;

}

*.banner-over*{

position: *absolute*;

color: *#fff*;

text-align: *center*;

transform: *translate(-50% , -50%)*;

left: *50%*;

top: *50%*;

}

**banner-over**{

font-size: *43px*;

}

*.header-top* **form** {

width: *100px*;

float: *left*;

}

*.header-top* **form** **input** {

border: *0*;

color: *red*;

/ background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*; /

background: *transparent*;

border: *0*;

padding: *10px 19px*;

color: *#fff*;

color: *#f4f4f4*;

cursor: *pointer*;

border: *1px solid #fff*;

border-radius: *0*;

margin-left: *20px*;

}

**table**{

background: *#f2ecec*;

}

**table** **tr** **td**{

border: *0*;

}

**table** *.heading-row* **td**{

padding: *10px*;

border: *0*;

}

**table** *.heading-row*{

background: *#92fe9d*;

}

**table** **tr** **td**{

padding: *10px*;

/\* border: 1px solid #c9c9c9;\*/

}

*.btn-success*{

background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*;

padding: *10px 19px*;

border: *0*;

border-radius: *5px*;

color: *#fff*;

}

</style>

</head>

<body>

<div class=*"main-wrapper"*>

<div class=*"header-top"*>

<form action=*"go-home.htm"* method=*"post"*>

<input class = *"btn btn-success"* type=*"submit"* name=*"action"* value=*"Home"*>

</form>

</div>

</div>

</body>

</html>

<%@ taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"* %>

<%@ page session=*"false"* %>

<html>

<head>

<title>Home</title>

<style>

*.main-wrapper*{

width: *1170px*;

margin: *0 auto*;

padding: *0px 20px*;

box-shadow: *0 0 10px #e3e3e3*;

height: *100%*;

min-height: *768px*;

}

*.header-top* {

background: *#00c9ff*;

padding: *10px 0*;

margin-left: *-20px*;

margin-right: *-20px*;

display: *block*;

height: *40px*;

}

*.banner-section*{

height: *300px*;

background-size: *cover*;

margin-left: *-20px*;

margin-right: *-20px*;

position: *relative*;

background-position: *0 37%*;

}

*.banner-over*{

position: *absolute*;

text-align: *center*;

transform: *translate(-50% , -50%)*;

left: *50%*;

top: *50%*;

}

**banner-over**{

font-size: *43px*;

}

*.header-top* **ul**{

margin: *0*;

margin-top: *7px*;

}

*.header-top* **ul** **li**{

list-style: *none*;

float: *left*;

}

*.header-top* **ul** **li** **a** {

text-decoration: *none*;

color: *#fff*;

padding: *5px 20px*;

margin-right: *11px*;

font-size: *18px*;

border: *1px solid #fff*;

}

*.header-top* **form** {

width: *100px*;

float: *left*;

}

*.header-top* **form** **input** {

border: *0*;

color: *red*;

/ background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*; /

background: *transparent*;

border: *0*;

padding: *10px 19px*;

color: *#fff*;

color: *#f4f4f4*;

cursor: *pointer*;

border: *1px solid #fff*;

border-radius: *0*;

margin-left: *20px*;

}

**table**{

background: *#f2ecec*;

}

**table** **tr** **td**{

border: *0*;

}

**table** *.heading-row* **td**{

padding: *10px*;

border: *0*;

}

**table** *.heading-row*{

background: *#92fe9d*;

}

**table** **tr** **td**{

padding: *10px*;

/\* border: 1px solid #c9c9c9;\*/

}

*.btn-success*{

background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*;

padding: *10px 19px*;

border: *0*;

border-radius: *5px*;

color: *#fff*;

}

</style>

</head>

<body>

<div class=*"main-wrapper"*>

<div class=*"header-top"*>

<ul>

<li><a href=*"#"*>Home</a></li>

<li><a href=*"login.htm"*>Login</a></li>

</ul>

</div>

<div class=*"banner-over"*>

<h1>Welcome to Hospital Management System</h1>

</div>

</div>

</body>

</html>

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

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

<style>

*.main-wrapper*{

width: *1170px*;

margin: *0 auto*;

padding: *0px 20px*;

box-shadow: *0 0 10px #e3e3e3*;

height: *100%*;

min-height: *768px*;

}

*.header-top* {

background: *#00c9ff*;

padding: *10px 0*;

margin-left: *-20px*;

margin-right: *-20px*;

display: *block*;

height: *40px*;

}

*.form-area* {

background-color: *#fff*;

border: *2px solid #ddd*;

border-radius: *10px*;

width: *320px*;

margin: *0 auto*;

margin-top: *100px*;

padding-right:*20px*;

}

*.form-padding* {

padding: *20px*;

}

*.form-padding* **h4** {

margin: *0 0 25px*;

font-weight: *700*;

}

*.form-group* {

margin-bottom: *15px*;

}

*.form-area* **label** {

font-weight: *600*;

font-size: *12px*;

color: *#575757*;

}

*.form-control* {

display: *block*;

width: *100%*;

height: *34px*;

padding: *6px 12px*;

font-size: *14px*;

line-height: *1.42857143*;

color: *#555*;

background-color: *#fff*;

background-image: *none*;

border: *1px solid #ccc*;

border-radius: *4px*;

-webkit-box-shadow: *inset 0 1px 1px rgba(0,0,0,.075)*;

box-shadow: *inset 0 1px 1px rgba(0,0,0,.075)*;

-webkit-transition: *border-color ease-in-out .15s,-webkit-box-shadow ease-in-out .15s*;

-o-transition: *border-color ease-in-out .15s,box-shadow ease-in-out .15s*;

transition: *border-color ease-in-out .15s,box-shadow ease-in-out .15s*;

}

*.form-area* *.form-control* {

height: *40px*;

border-radius: *30px*;

box-shadow: *none*;

border: *1px solid #ddd*;

}

*.btn*{

background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*;

padding: *10px 19px*;

border: *0*;

border-radius: *5px*;

color: *#fff*;

cursor: *pointer*;

}

</style>

</head>

<body>

<div class=*"main-wrapper"*>

<div class=*"header-top "*>

<h2 style=" margin-top: *2px*; text-align: *center*;">Login form</h2>

</div>

<div id=*"form"* class=*"form-area"*>

<form action=*"login.htm"* method=*"post"*>

<div class=*"form-padding"*>

<h4>Login form</h4>

<div class=*"form-group"*>

<label for=*""*>Username:</label>

<input type=*"text"* name=*"username"* class=*"form-control"*/>

</div>

<div class=*"form-group"*>

<label for=*""*>Password:</label>

<input type=*"password"* name=*"password"* class=*"form-control"*/>

</div>

<input type=*"submit"* value=*"Login"* class=*"btn"*/> <a href=*"register.htm"*>New User?</a>

</div>

</form>

</div>

</div>

</body>

</html>

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

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

<style>

*.header-top* **form** {

width: *100px*;

float: *left*;

}

*.header-top* **form** **input**{

border: *0*;

color: *red*;

/\* background: rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0;\*/

background: *transparent*;

border: *0*;

padding: *10px 19px*;

color: *#fff*;

color: *#121212*;

cursor: *pointer*;

}

</style>

</head>

<body>

<div class=*"header-top "*>

<form action=*"logout.htm"* method=*"post"*>

<input class=*"btn btn-success"* type=*"submit"* name=*"action"* value=*"Logout"*>

</form>

</div></br></br>

<h1>Patient Added Successfully</h1></br></br>

</body>

</html>

<%@taglib uri=*"http://www.springframework.org/tags/form"* prefix=*"form"*%>

<%@taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"*%>

<%@taglib uri=*"http://java.sun.com/jsp/jstl/functions"* prefix=*"fn"*%>

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

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

<style>

*.main-wrapper*{

width: *1170px*;

margin: *0 auto*;

padding: *0px 20px*;

box-shadow: *0 0 10px #e3e3e3*;

height: *100%*;

min-height: *768px*;

}

*.header-top* {

background: *#00c9ff*;

padding: *10px 0*;

margin-left: *-20px*;

margin-right: *-20px*;

display: *block*;

height: *40px*;

}

*.banner-section*{

background: *url(slider-1.jpg) no-repeat*;

height: *300px*;

background-size: *cover*;

margin-left: *-20px*;

margin-right: *-20px*;

position: *relative*;

background-position: *0 37%*;

}

*.banner-over*{

position: *absolute*;

color: *#fff*;

text-align: *center*;

transform: *translate(-50% , -50%)*;

left: *50%*;

top: *50%*;

}

**banner-over**{

font-size: *43px*;

}

*.header-top* **ul**{

margin: *0*;

margin-top: *7px*;

}

*.header-top* **ul** **li**{

list-style: *none*;

float: *left*;

}

*.header-top* **ul** **li** **a** {

text-decoration: *none*;

color: *#fff*;

padding: *5px 20px*;

margin-right: *11px*;

font-size: *18px*;

border: *1px solid #fff*;

}

*.header-top* **form** {

width: *100px*;

float: *left*;

}

*.header-top* **form** **input** {

border: *0*;

color: *red*;

/ background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*; /

background: *transparent*;

border: *0*;

padding: *10px 19px*;

color: *#fff*;

color: *#f4f4f4*;

cursor: *pointer*;

border: *1px solid #fff*;

border-radius: *0*;

margin-left: *20px*;

}

**table**{

background: *#f2ecec*;

}

**table** **tr** **td**{

border: *0*;

}

**table** *.heading-row* **td**{

padding: *10px*;

border: *0*;

}

**table** *.heading-row*{

background: *#92fe9d*;

}

**table** **tr** **td**{

padding: *10px*;

/\* border: 1px solid #c9c9c9;\*/

}

*.btn-success*{

background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*;

padding: *10px 19px*;

border: *0*;

border-radius: *5px*;

color: *#fff*;

cursor: *pointer*;

}

</style>

</head>

<body>

<c:set var=*"contextPath"* value=*"*${pageContext.request.contextPath}*"*/>

<div class=*"main-wrapper"*>

<div class=*"header-top"*>

<form action=*"add-drug.htm"* method=*"post"*>

<input class = *"btn btn-success"* type=*"submit"* name=*"action"* value=*"Add more drug"*>

</form>

<form action=*"logout.htm"* method=*"post"*>

<input class=*"btn btn-success"* type=*"submit"* name=*"action"* value=*"Logout"*>

</form>

</div>

<h1>Order Summary </h1>

<form:form action=*"order-summary.htm"* CommandName=*"patient"* method=*"post"*>

<table width=*"100%"*>

<tr class=*"heading-row"*>

<td id=*"td1"*> Patient Name</td>

<td id=*"td1"*> Patient Age </td>

<td id=*"td1"*> Patient Mobile</td>

<td id=*"td1"*> Patient Address</td>

</tr>

<tr>

<td>${finalPatient.patientName}</td>

<td>${finalPatient.patientAge}</td>

<td>${finalPatient.patientMobile}</td>

<td>${finalPatient.addressLine1}</td>

</tr>

</table>

<br>

<br>

<table width=*"100%"*>

<tr class=*"heading-row"*>

<td id=*"td1"*> Drug Name</td>

<td id=*"td1"*> Drug id </td>

<td id=*"td1"*> Drug Dosage</td>

<td id=*"td1"*> Drug Quantity</td>

</tr>

<c:forEach items=*"*${drugList}*"* var=*"drug"*>

<tr>

<td>${drug.drugName}</td>

<td><input type=*"text"* name=*"drugid"* value=*"*${drug.drugcomid}*"* readonly></td>

<td>${drug.dosageValue}</td>

<td><input type=*"text"* name=*"qty"*/></td>

</tr>

</c:forEach>

</table>

<input class = *"btn btn-success"* type=*"submit"* name=*"action"* value=*"Submit Summary"*>

</form:form> </br></br>

<form:form action=*"check-interactions.htm"* method=*"post"*>

<input class = *"btn btn-success"* type=*"submit"* name=*"action"* value=*"Check Interactions"*>

</form:form>

</div>

</body>

</html>

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

pageEncoding=*"ISO-8859-1"*%>

<%@taglib uri=*"http://www.springframework.org/tags/form"* prefix=*"form"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

<style>

*.main-wrapper*{

width: *1170px*;

margin: *0 auto*;

padding: *0px 20px*;

box-shadow: *0 0 10px #e3e3e3*;

min-height: *768px*;

padding-bottom: *100px*;

}

*.header-top* {

background: *#00c9ff*;

padding: *10px 0*;

margin-left: *-20px*;

margin-right: *-20px*;

display: *block*;

height: *40px*;

}

*.banner-section*{

background: *url("/resources/slider-1.jpg") no-repeat*;

height: *300px*;

background-size: *cover*;

margin-left: *-20px*;

margin-right: *-20px*;

position: *relative*;

background-position: *0 37%*;

}

*.banner-over*{

position: *absolute*;

color: *#fff*;

text-align: *center*;

transform: *translate(-50% , -50%)*;

left: *50%*;

top: *50%*;

}

**banner-over**{

font-size: *43px*;

}

*.header-top* **ul**{

margin: *0*;

margin-top: *7px*;

}

*.header-top* **ul** **li**{

list-style: *none*;

float: *left*;

}

*.header-top* **ul** **li** **a**{

text-decoration: *none*;

color: *#000*;

padding: *10px 20px*;

margin-right: *11px*;

font-size: *18px*;

}

*.header-top* **form** {

width: *100px*;

float: *left*;

}

*.header-top* **form** **input**{

border: *0*;

color: *red*;

/\* background: rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0;\*/

background: *transparent*;

border: *0*;

padding: *10px 19px*;

color: *#fff*;

color: *#121212*;

cursor: *pointer*;

}

**table**{

background: *#f2ecec*;

}

**table** **tr** **td**{

border: *0*;

}

**table** *.heading-row* **td**{

padding: *10px*;

border: *0*;

}

**table** *.heading-row*{

background: *#92fe9d*;

}

**table** **tr** **td**{

padding: *10px*;

/\* border: 1px solid #c9c9c9;\*/

}

*.btn-success*{

background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*;

padding: *10px 19px*;

border: *0*;

border-radius: *5px*;

color: *#fff*;

}

*.form-area* {

background-color: *#fff*;

border: *2px solid #ddd*;

border-radius: *10px*;

width: *320px*;

margin: *0 auto*;

margin-top: *100px*;

padding-right: *20px*;

}

*.form-padding* {

padding: *20px*;

}

*.form-padding* **h4** {

margin: *0 0 25px*;

font-weight: *700*;

}

*.form-group* {

margin-bottom: *15px*;

}

*.form-area* **label** {

font-weight: *600*;

font-size: *12px*;

color: *#575757*;

}

*.form-control* {

display: *block*;

width: *100%*;

height: *34px*;

padding: *6px 12px*;

font-size: *14px*;

line-height: *1.42857143*;

color: *#555*;

background-color: *#fff*;

background-image: *none*;

border: *1px solid #ccc*;

border-radius: *4px*;

-webkit-box-shadow: *inset 0 1px 1px rgba(0,0,0,.075)*;

box-shadow: *inset 0 1px 1px rgba(0,0,0,.075)*;

-webkit-transition: *border-color ease-in-out .15s,-webkit-box-shadow ease-in-out .15s*;

-o-transition: *border-color ease-in-out .15s,box-shadow ease-in-out .15s*;

transition: *border-color ease-in-out .15s,box-shadow ease-in-out .15s*;

}

*.form-area* *.form-control* {

height: *30px*;

border-radius: *30px*;

box-shadow: *none*;

border: *1px solid #ddd*;

margin-top: *10px*;

}

*.btn*{

background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*;

padding: *10px 19px*;

border: *0*;

border-radius: *5px*;

color: *#fff*;

cursor: *pointer*;

}

</style>

</head>

<body>

<div class=*"main-wrapper"*>

<div class=*"header-top "*>

<form action=*"logout.htm"* method=*"post"*>

<input class=*"btn btn-success"* type=*"submit"* name=*"action"* value=*"Logout"*>

</form>

</div>

<div class=*"banner-over"*>

<h1>Patient Detail</h1>

</div>

<div id=*"form"* class=*"form-area"*>

<form:form commandName=*"patient"* action=*"patient.htm"* method=*"post"*>

<div class=*"form-padding"*>

<h4>Patient Detail</h4>

<div class=*"form-group"*>

<label for=*""*>Patient Name: </label>

<form:input type=*"text"* class=*"form-control"* id=*"patientName"* path=*"patientName"* required=*"required"* /><form:errors path=*"patientName"* />

</div>

<div class=*"form-group"*>

<label for=*""*> Patient Age:</label>

<form:input type=*"text"* class=*"form-control"* id=*"patientAge"* path=*"patientAge"* required=*"required"* /><form:errors path=*"patientAge"* />

</div>

<div class=*"form-group"*>

<label for=*""*>Patient Mobile: </label>

<form:input type=*"text"* class=*"form-control"* id=*"patientMobile"* path=*"patientMobile"* required=*"required"* /><form:errors path=*"patientMobile"* />

</div>

<div class=*"form-group"*>

<label for=*""*>Address Line 1:</label>

<form:input type=*"text"* class=*"form-control"* id=*"addressLine1"* path=*"addressLine1"* />

</div>

<div class=*"form-group"*>

<label for=*""*>Address Line 2: </label>

<form:input type=*"text"* class=*"form-control"* id=*"addressLine2"* path=*"addressLine2"* />

</div>

<div class=*"form-group"*>

<label for=*""*>State: </label>

<form:input type=*"text"* class=*"form-control"* id=*"state"* path=*"state"* />

</div>

<div class=*"form-group"*>

<label for=*""*>City:</label>

<form:input type=*"text"* class=*"form-control"* id=*"city"* path=*"city"* required=*"required"* /><form:errors path=*"city"* />

</div>

<div class=*"form-group"*>

<label for=*""*>Country:</label>

<form:input type=*"text"* class=*"form-control"* id=*"country"* path=*"country"* />

</div>

<div class=*"form-group"*>

<label for=*""*>Zip Code:</label>

<form:input type=*"text"* class=*"form-control"* id=*"zipcode"* path=*"zipcode"* required=*"required"* /><form:errors path=*"zipcode"*/></br>

</div>

</br>

<input type=*"submit"* name=*"Add\_Patient"* class=*"btn"* /></br>

</div>

</form:form>

</div>

</div>

</body>

</html>

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

pageEncoding=*"ISO-8859-1"*%>

<%@taglib uri=*"http://www.springframework.org/tags/form"* prefix=*"form"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

<style>

*.main-wrapper*{

width: *1170px*;

margin: *0 auto*;

padding: *0px 20px*;

box-shadow: *0 0 10px #e3e3e3*;

height: *100%*;

min-height: *768px*;

}

*.header-top* {

background: *#00c9ff*;

padding: *10px 0*;

margin-left: *-20px*;

margin-right: *-20px*;

display: *block*;

height: *40px*;

}

*.banner-section*{

background: *url(slider-1.jpg) no-repeat*;

height: *300px*;

background-size: *cover*;

margin-left: *-20px*;

margin-right: *-20px*;

position: *relative*;

background-position: *0 37%*;

}

*.banner-over*{

position: *absolute*;

color: *#fff*;

text-align: *center*;

transform: *translate(-50% , -50%)*;

left: *50%*;

top: *50%*;

}

**banner-over**{

font-size: *43px*;

}

*.header-top* **ul**{

margin: *0*;

margin-top: *7px*;

}

*.header-top* **ul** **li**{

list-style: *none*;

float: *left*;

}

*.header-top* **ul** **li** **a**{

text-decoration: *none*;

color: *#000*;

padding: *10px 20px*;

margin-right: *11px*;

font-size: *18px*;

}

*.header-top* **form** {

width: *100px*;

float: *left*;

}

*.header-top* **form** **input**{

border: *0*;

color: *red*;

/\* background: rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0;\*/

background: *transparent*;

border: *0*;

padding: *10px 19px*;

color: *#fff*;

color: *#121212*;

cursor: *pointer*;

}

**table**{

background: *#f2ecec*;

}

**table** **tr** **td**{

border: *0*;

}

**table** *.heading-row* **td**{

padding: *10px*;

border: *0*;

}

**table** *.heading-row*{

background: *#92fe9d*;

}

**table** **tr** **td**{

padding: *10px*;

/\* border: 1px solid #c9c9c9;\*/

}

*.btn-success*{

background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*;

padding: *10px 19px*;

border: *0*;

border-radius: *5px*;

color: *#fff*;

}

*.form-area* {

background-color: *#fff*;

border: *2px solid #ddd*;

border-radius: *10px*;

width: *320px*;

margin: *0 auto*;

margin-top: *100px*;

padding-right: *20px*;

}

*.form-padding* {

padding: *20px*;

}

*.form-padding* **h4** {

margin: *0 0 25px*;

font-weight: *700*;

}

*.form-group* {

margin-bottom: *15px*;

}

*.form-area* **label** {

font-weight: *600*;

font-size: *12px*;

color: *#575757*;

}

*.form-control* {

display: *block*;

width: *100%*;

height: *34px*;

padding: *6px 12px*;

font-size: *14px*;

line-height: *1.42857143*;

color: *#555*;

background-color: *#fff*;

background-image: *none*;

border: *1px solid #ccc*;

border-radius: *4px*;

-webkit-box-shadow: *inset 0 1px 1px rgba(0,0,0,.075)*;

box-shadow: *inset 0 1px 1px rgba(0,0,0,.075)*;

-webkit-transition: *border-color ease-in-out .15s,-webkit-box-shadow ease-in-out .15s*;

-o-transition: *border-color ease-in-out .15s,box-shadow ease-in-out .15s*;

transition: *border-color ease-in-out .15s,box-shadow ease-in-out .15s*;

}

*.form-area* *.form-control* {

height: *30px*;

border-radius: *30px*;

box-shadow: *none*;

border: *1px solid #ddd*;

margin-top: *10px*;

}

*.btn*{

background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*;

padding: *10px 19px*;

border: *0*;

border-radius: *5px*;

color: *#fff*;

cursor: *pointer*;

}

</style>

</head>

<body>

<div class=*"main-wrapper"*>

<div class=*"header-top"*>

<form action=*"logout.htm"* method=*"post"*>

<input class=*"btn btn-success"* type=*"submit"* name=*"action"* value=*"Logout"*>

</form>

</div>

<div class=*"banner-over"*>

<h1>Drug Detail</h1>

</div>

<div id=*"form"* class=*"form-area"*>

<form:form commandName=*"drug"* action=*"pharmacy.htm"* method=*"post"*>

<div class=*"form-padding"*>

<h4>Drug Detail</h4>

<div class=*"form-group"*>

<label for=*""*>Drug Component ID: </label>

<form:input type=*"text"* class=*"form-control"* id=*"drugcomid"* path=*"drugcomid"* required=*"required"* />

</div>

<div class=*"form-group"*>

<label for=*""*> Drug Name:</label>

<form:input type=*"text"* class=*"form-control"* id=*"drugName"* path=*"drugName"* required=*"required"* />

</div>

<div class=*"form-group"*>

<label for=*""*> Drug Dose: </label>

<form:input type=*"text"* class=*"form-control"* id=*"dosageValue"* path=*"dosageValue"* required=*"required"* />

</div>

<br>

<input type=*"submit"* name=*"Add\_Drug"* class=*"btn"* /></br>

</div>

</form:form>

</div>

</div>

</body>

</html>

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

pageEncoding=*"ISO-8859-1"*%>

<%@taglib uri=*"http://www.springframework.org/tags/form"* prefix=*"form"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

<style>

*.main-wrapper*{

width: *1170px*;

margin: *0 auto*;

padding: *0px 20px*;

box-shadow: *0 0 10px #e3e3e3*;

height: *100%*;

min-height: *768px*;

}

*.header-top* {

background: *#00c9ff*;

padding: *10px 0*;

margin-left: *-20px*;

margin-right: *-20px*;

display: *block*;

height: *40px*;

}

*.form-area* {

background-color: *#fff*;

border: *2px solid #ddd*;

border-radius: *10px*;

width: *400px*;

margin: *0 auto*;

margin-top: *100px*;

padding-right: *20px*;

}

*.form-padding* {

padding: *20px*;

}

*.form-padding* **h4** {

margin: *0 0 25px*;

font-weight: *700*;

}

*.form-group* {

margin-bottom: *15px*;

}

*.form-area* **label** {

font-weight: *600*;

font-size: *12px*;

color: *#575757*;

}

*.form-control* {

display: *block*;

width: *100%*;

height: *34px*;

padding: *6px 12px*;

font-size: *14px*;

line-height: *1.42857143*;

color: *#555*;

background-color: *#fff*;

background-image: *none*;

border: *1px solid #ccc*;

border-radius: *4px*;

-webkit-box-shadow: *inset 0 1px 1px rgba(0,0,0,.075)*;

box-shadow: *inset 0 1px 1px rgba(0,0,0,.075)*;

-webkit-transition: *border-color ease-in-out .15s,-webkit-box-shadow ease-in-out .15s*;

-o-transition: *border-color ease-in-out .15s,box-shadow ease-in-out .15s*;

transition: *border-color ease-in-out .15s,box-shadow ease-in-out .15s*;

}

*.form-area* *.form-control* {

height: *30px*;

border-radius: *30px*;

box-shadow: *none*;

border: *1px solid #ddd*;

}

*.btn*{

background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*;

padding: *10px 19px*;

border: *0*;

border-radius: *5px*;

color: *#fff*;

cursor: *pointer*;

}

</style>

</head>

<body>

<div class=*"main-wrapper"*>

<div class=*"header-top "*>

<h2 style=" margin-top: *2px*; text-align: *center*;">Register form</h2>

</div>

<div id=*"form"* class=*"form-area"*>

<form:form commandName=*"user"* action=*"registerUser.htm"* method=*"post"*>

<div class=*"form-padding"*>

<h4>Register form</h4>

<div class=*"form-group"*>

<label for=*""*>Username:</label>

<form:input type=*"text"* class=*"form-control"* id=*"username"* path=*"username"* required=*"required"* />

<form:errors path=*"username"* />

</div>

<div class=*"form-group"*>

<label for=*""*>FirstName:</label>

<form:input type=*"text"* class=*"form-control"* id=*"firstname"* path=*"firstName"* required=*"required"* />

<form:errors path=*"firstName"* />

</div>

<div class=*"form-group"*>

<label for=*""*>LastName:</label>

<form:input type=*"text"* class=*"form-control"* id=*"lastname"* path=*"lastName"* required=*"required"* />

<form:errors path=*"lastName"* />

</div>

<div class=*"form-group"*>

<label for=*""*>Mobile:</label>

<form:input type=*"text"* class=*"form-control"* id=*"mobile"* path=*"mobileNumber"* required=*"required"* />

<form:errors path=*"mobileNumber"* />

</div>

<div class=*"form-group"*>

<label for=*""*>Password:</label>

<form:input type=*"password"* class=*"form-control"* id=*"password"* path=*"password"* required=*"required"* />

<form:errors path=*"password"* />

</div>

<div class=*"form-group"*>

<label for=*""*>Role:</label>

<form:select path=*"roleType"* class=*"form-control"* required=*"required"*>

<form:option value=*"D"* label=*"Doctor"* />

<form:option value=*"R"* label=*"Receptionist"* />

<form:option value=*"P"* label=*"Pharmacist"* />

</form:select>

</div>

<input type=*"submit"* name=*"register"* class=*"btn"*/>

</div>

</form:form>

</div>

</div>

</body>

</html>

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

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

<style>

*.main-wrapper*{

width: *1170px*;

margin: *0 auto*;

padding: *0px 20px*;

box-shadow: *0 0 10px #e3e3e3*;

height: *100%*;

min-height: *768px*;

}

*.header-top* {

background: *#00c9ff*;

padding: *10px 0*;

margin-left: *-20px*;

margin-right: *-20px*;

display: *block*;

height: *40px*;

}

*.form-area* {

background-color: *#fff*;

border: *2px solid #ddd*;

border-radius: *10px*;

width: *320px*;

margin: *0 auto*;

margin-top: *100px*;

padding-right:*20px*;

}

*.form-padding* {

padding: *20px*;

}

*.form-padding* **h4** {

margin: *0 0 25px*;

font-weight: *700*;

}

*.form-group* {

margin-bottom: *15px*;

}

*.form-area* **label** {

font-weight: *600*;

font-size: *12px*;

color: *#575757*;

}

*.form-control* {

display: *block*;

width: *100%*;

height: *34px*;

padding: *6px 12px*;

font-size: *14px*;

line-height: *1.42857143*;

color: *#555*;

background-color: *#fff*;

background-image: *none*;

border: *1px solid #ccc*;

border-radius: *4px*;

-webkit-box-shadow: *inset 0 1px 1px rgba(0,0,0,.075)*;

box-shadow: *inset 0 1px 1px rgba(0,0,0,.075)*;

-webkit-transition: *border-color ease-in-out .15s,-webkit-box-shadow ease-in-out .15s*;

-o-transition: *border-color ease-in-out .15s,box-shadow ease-in-out .15s*;

transition: *border-color ease-in-out .15s,box-shadow ease-in-out .15s*;

}

*.form-area* *.form-control* {

height: *40px*;

border-radius: *30px*;

box-shadow: *none*;

border: *1px solid #ddd*;

}

*.btn*{

background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*;

padding: *10px 19px*;

border: *0*;

border-radius: *5px*;

color: *#fff*;

cursor: *pointer*;

}

*.header-top* **form** {

width: *100px*;

float: *left*;

}

*.header-top* **form** **input** {

border: *0*;

color: *red*;

/ background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*; /

background: *transparent*;

border: *0*;

padding: *10px 19px*;

color: *#fff*;

color: *#f4f4f4*;

cursor: *pointer*;

border: *1px solid #fff*;

border-radius: *0*;

margin-left: *20px*;

}

</style>

</head>

<body>

<div class=*"main-wrapper"*>

<div class=*"header-top"*>

<form action=*"doctor-home.htm"* method=*"post"*>

<input class = *"btn btn-success"* type=*"submit"* name=*"action"* value=*"Home"*>

</form>

<form action=*"logout.htm"* method=*"post"*>

<input class=*"btn btn-success"* type=*"submit"* name=*"action"* value=*"Logout"*>

</form>

</div>

<div id=*"form"* class=*"form-area"*>

<form action=*"searchdrug.htm"* method=*"post"*>

<div class=*"form-padding"*>

<h4>Search Drug</h4>

<div class=*"form-group"*>

<label for=*""*>Drug Name:</label>

<input type=*"text"* name=*"dname"* class=*"form-control"*/>

</div>

<input type=*"submit"* name=*"Search Drug"* class=*"btn"*/></br>

</div>

</form>

</div>

</div>

</body>

</html>

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

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

<style>

*.main-wrapper*{

width: *1170px*;

margin: *0 auto*;

padding: *0px 20px*;

box-shadow: *0 0 10px #e3e3e3*;

height: *100%*;

min-height: *768px*;

}

*.header-top* {

background: *#00c9ff*;

padding: *10px 0*;

margin-left: *-20px*;

margin-right: *-20px*;

display: *block*;

height: *40px*;

}

*.form-area* {

background-color: *#fff*;

border: *2px solid #ddd*;

border-radius: *10px*;

width: *320px*;

margin: *0 auto*;

margin-top: *100px*;

padding-right:*20px*;

}

*.form-padding* {

padding: *20px*;

}

*.form-padding* **h4** {

margin: *0 0 25px*;

font-weight: *700*;

}

*.form-group* {

margin-bottom: *15px*;

}

*.form-area* **label** {

font-weight: *600*;

font-size: *12px*;

color: *#575757*;

}

*.form-control* {

display: *block*;

width: *100%*;

height: *34px*;

padding: *6px 12px*;

font-size: *14px*;

line-height: *1.42857143*;

color: *#555*;

background-color: *#fff*;

background-image: *none*;

border: *1px solid #ccc*;

border-radius: *4px*;

-webkit-box-shadow: *inset 0 1px 1px rgba(0,0,0,.075)*;

box-shadow: *inset 0 1px 1px rgba(0,0,0,.075)*;

-webkit-transition: *border-color ease-in-out .15s,-webkit-box-shadow ease-in-out .15s*;

-o-transition: *border-color ease-in-out .15s,box-shadow ease-in-out .15s*;

transition: *border-color ease-in-out .15s,box-shadow ease-in-out .15s*;

}

*.form-area* *.form-control* {

height: *40px*;

border-radius: *30px*;

box-shadow: *none*;

border: *1px solid #ddd*;

}

*.btn*{

background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*;

padding: *10px 19px*;

border: *0*;

border-radius: *5px*;

color: *#fff*;

cursor: *pointer*;

}

*.header-top* **form** {

width: *100px*;

float: *left*;

}

*.header-top* **form** **input** {

border: *0*;

color: *red*;

/ background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*; /

background: *transparent*;

border: *0*;

padding: *10px 19px*;

color: *#fff*;

color: *#f4f4f4*;

cursor: *pointer*;

border: *1px solid #fff*;

border-radius: *0*;

margin-left: *20px*;

}

</style>

</head>

<body>

<div class=*"main-wrapper"*>

<div class=*"header-top"*>

<form action=*"doctor-home.htm"* method=*"post"*>

<input class = *"btn btn-success"* type=*"submit"* name=*"action"* value=*"Home"*>

</form>

<form action=*"logout.htm"* method=*"post"*>

<input class=*"btn btn-success"* type=*"submit"* name=*"action"* value=*"Logout"*>

</form>

</div>

<div id=*"form"* class=*"form-area"*>

<form action=*"searchPatient.htm"* method=*"post"*>

<div class=*"form-padding"*>

<h4>Search Patient</h4>

<div class=*"form-group"*>

<label for=*""*>Patient Name:</label>

<input type=*"text"* name=*"pname"* class=*"form-control"*/>

</div>

<input type=*"submit"* name=*"Search Patient"* class=*"btn"*/></br>

</div>

</form>

</div>

</div>

</body>

</html>

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

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

<style>

*.main-wrapper*{

width: *1170px*;

margin: *0 auto*;

padding: *0px 20px*;

box-shadow: *0 0 10px #e3e3e3*;

height: *100%*;

min-height: *768px*;

}

*.header-top* {

background: *#00c9ff*;

padding: *10px 0*;

margin-left: *-20px*;

margin-right: *-20px*;

display: *block*;

height: *40px*;

}

*.banner-section*{

background: *url(slider-1.jpg) no-repeat*;

height: *300px*;

background-size: *cover*;

margin-left: *-20px*;

margin-right: *-20px*;

position: *relative*;

background-position: *0 37%*;

}

*.banner-over*{

position: *absolute*;

color: *#fff*;

text-align: *center*;

transform: *translate(-50% , -50%)*;

left: *50%*;

top: *50%*;

}

**banner-over**{

font-size: *43px*;

}

*.header-top* **form** {

width: *100px*;

float: *left*;

}

*.header-top* **form** **input** {

border: *0*;

color: *red*;

/ background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*; /

background: *transparent*;

border: *0*;

padding: *10px 19px*;

color: *#fff*;

color: *#f4f4f4*;

cursor: *pointer*;

border: *1px solid #fff*;

border-radius: *0*;

margin-left: *20px*;

}

**table**{

background: *#f2ecec*;

}

**table** **tr** **td**{

border: *0*;

}

**table** *.heading-row* **td**{

padding: *10px*;

border: *0*;

}

**table** *.heading-row*{

background: *#92fe9d*;

}

**table** **tr** **td**{

padding: *10px*;

/\* border: 1px solid #c9c9c9;\*/

}

*.btn-success*{

background: *rgba(0, 0, 0, 0) linear-gradient(to right, #92fe9d, #00c9ff) repeat scroll 0 0*;

padding: *10px 19px*;

border: *0*;

border-radius: *5px*;

color: *#fff*;

}

</style>

</head>

<body>

<div class=*"main-wrapper"*>

<div class=*"header-top"*>

<form action=*"doctor-home.htm"* method=*"post"*>

<input class = *"btn btn-success"* type=*"submit"* name=*"action"* value=*"Home"*>

</form>

<form action=*"logout.htm"* method=*"post"*>

<input class=*"btn btn-success"* type=*"submit"* name=*"action"* value=*"Logout"*>

</form>

</div>

<div class=*"banner-over"*>

<h1>Show Interactions</h1>

</div>

<br>

<br>

<table width=*"100%"*>

<tr class=*"heading-row"*>

<td id=*"td1"*> Interaction String</td>

</tr>

<tr>

<td>${interaction}</td>

</tr>

</table>

</div>

</body>

</html>

# **Appendix**

**Appendix A**

* DoctorController.java
* HomeController.java
* LoginUserController.java
* MainPageController.java
* PharmacyController.java
* RegisterPatientController.java
* RegisterUserController.java

**Appendix B**

* DAO.java
* DrugDAO.java
* PatientDAO.java
* UserDAO.java

**Appendix C**

* Address.java
* Drug.java
* OrderSummary.java
* Patient.java
* PdfReport.java
* User.java

**Appendix D**

* DrugValidation.java
* PatientValidation.java
* userValidation.java

**Appendix E**

* hibernate.cfg.xml
* root-context.xml

**Appendix F**

* doctor.jsp
* drug-added.jsp
* drug-inter.jsp
* error.jsp
* home.jsp
* login.jsp
* patient-added.jsp
* patient-order.jsp
* patient.jsp
* pharmacy.jsp
* register.jsp
* search-drug.jsp
* searchPatient.jsp
* show-interactions.jsp

Appendix F