Shukla,ketul **Job Portal** Web Tools

I have tried implementing the job portal where in the job seekers can find the jobs for them and the company looking for the candidate can find the person. Thus, this works as an vice versa. The hired jobs might not be available for the new job seeker. But It can be a goof platform where in people can put some role on their part and get the best for them.

Functionalities:

* The jobseeker and the company can register.
* The company can post the job.
* All the posted jobs can be seen by the job seeker. Also, the different criteria can be applied and different jobs could be found out.
* The job seeker has the option to view the type of the job or the job by some company name.
* Once the job seeker applies for the job; the company can see the applicants for position they are looking for.
* Once the company hires an employee the job is no longer available to the other job seeker.
* The job seeker can see his application.
* The job seeker can also see the calls he has for the applications he had made.
* Also, once the job seeker applies to a job he is notified via email about his application.
* When the company selects the person; the person is notified about the job offering to him.
* Thus, this are some of the key features of this application.

Controllers:

**JobApplicationController:**

package com.ketul.jobportal.controller;

import java.io.File;

import javax.servlet.ServletContext;

import javax.servlet.http.HttpServletRequest;

import org.apache.commons.mail.DefaultAuthenticator;

import org.apache.commons.mail.Email;

import org.apache.commons.mail.SimpleEmail;

import org.hibernate.HibernateException;

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.annotation.ModelAttribute;

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

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

import org.springframework.web.multipart.commons.CommonsMultipartFile;

import org.springframework.web.servlet.ModelAndView;

import com.ketul.jobportal.dao.DAO;

import com.ketul.jobportal.dao.JobApplicationDAO;

import com.ketul.jobportal.dao.JobDAO;

import com.ketul.jobportal.pojo.Job;

import com.ketul.jobportal.pojo.JobApplication;

import com.ketul.jobportal.pojo.Person;

import com.ketul.jobportal.pojo.User;

@Controller

public class JobApplicationController {

@Autowired

@Qualifier("jobDao")

JobDAO jobDao;

@Autowired

@Qualifier("jobApplicationDao")

JobApplicationDAO jobApplicationDao;

@Autowired

ServletContext servletContext;

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

protected ModelAndView applypage(HttpServletRequest request) throws Exception{

ModelAndView mv= new ModelAndView();

Job job = jobDao.get(Integer.parseInt(request.getParameter("job")));

User user = (User) request.getSession().getAttribute("user");

int userID = user.getUserID();

JobApplication jobApplication = jobApplicationDao.checkApplication(Integer.parseInt(request.getParameter("job")),userID);

if(jobApplication == null){

mv.addObject("job", job);

mv.addObject("jobApplication", new JobApplication());

mv.setViewName("job-apply");

}

else{

mv.addObject("errorMessage", "You have already applied for this job. Please apply to other job.");

mv.setViewName("error");

}

return mv;

}

@RequestMapping(value="/job/apply",method=RequestMethod.POST)

protected ModelAndView applyJob(HttpServletRequest request, @ModelAttribute("jobApplication") JobApplication jobApplication, BindingResult result) throws Exception{

try{

DAO.begin();

Job job = jobDao.get(Integer.parseInt(request.getParameter("job")));

jobApplication.setJob(job);

Person person = (Person) request.getSession().getAttribute("user");

jobApplication.setPerson(person);

File localFile;

CommonsMultipartFile photoInMemory=jobApplication.getUpload();

String name = photoInMemory.getOriginalFilename();

localFile =new File("C:\\Users\\ketul\\Desktop\\Resume",name);

photoInMemory.transferTo(localFile);

jobApplication.setResume(localFile.getPath());

jobApplication.setStatus("applied");

DAO.getSession().merge(jobApplication);

DAO.commit();

try {

Email email = new SimpleEmail();

email.setHostName("smtp.googlemail.com");//If a server is capable of sending email, then you don't need the authentication. In this case, an email server needs to be running on that machine. Since we are running this application on the localhost and we don't have a email server, we are simply asking gmail to relay this email.

email.setSmtpPort(587);

email.setAuthenticator(new DefaultAuthenticator("portaljobs95@gmail.com", "jobportal95"));

email.setSSLOnConnect(true);

email.setFrom("portaljobs95@gmail.com");//This email will appear in the from field of the sending email. It doesn't have to be a real email address.This could be used for phishing/spoofing!

email.setSubject("Application Success");

email.setMsg("Congratulations! you have successfully applied for"+ job.getJobTitle() + "position at" + job.getCompany().getCompanyName());

email.addTo(person.getEmailID());//Will come from the database

email.send();

} catch (Exception e) {

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

}

return new ModelAndView("apply-success");

}

catch(HibernateException e){

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", e.getMessage());

}

}

}

**JobController:**

package com.ketul.jobportal.controller;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import org.apache.commons.mail.DefaultAuthenticator;

import org.apache.commons.mail.Email;

import org.apache.commons.mail.SimpleEmail;

import org.hibernate.HibernateException;

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.annotation.ModelAttribute;

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

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

import org.springframework.web.servlet.ModelAndView;

import com.ketul.jobportal.dao.DAO;

import com.ketul.jobportal.dao.JobApplicationDAO;

import com.ketul.jobportal.dao.JobDAO;

import com.ketul.jobportal.pojo.Company;

import com.ketul.jobportal.pojo.Job;

import com.ketul.jobportal.pojo.JobApplication;

import com.ketul.jobportal.pojo.Person;

import com.ketul.jobportal.pojo.User;

@Controller

public class JobController {

@Autowired

@Qualifier("jobDao")

JobDAO jobDao;

@Autowired

@Qualifier("jobApplicationDao")

JobApplicationDAO jobApplicationDao;

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

protected ModelAndView jobPost() throws Exception{

return new ModelAndView("post-job", "job", new Job());

}

@RequestMapping(value="/job/post", method=RequestMethod.POST)

protected ModelAndView jobAdd(HttpServletRequest request, @ModelAttribute("job") Job job, BindingResult result) throws Exception{

try{

DAO.begin();

job.setJobTitle(request.getParameter("jobTitle"));

job.setJobType(request.getParameter("jobType"));

job.setJobField(request.getParameter("jobField"));

job.setContactEmailID(request.getParameter("contactEmailID"));

job.setDescription(request.getParameter("jobDescription"));

job.setStatus("hiring");

job.setCompany((Company) request.getSession().getAttribute("user"));

DAO.getSession().save(job);

DAO.commit();

return new ModelAndView("post-success", "job", job);

}

catch(HibernateException e){

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", e.getMessage());

}

}

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

protected ModelAndView viewJob(HttpServletRequest request) throws Exception {

try {

List<Job> jobs = jobDao.list();

if(jobs.isEmpty()){

return new ModelAndView("error","errorMessage","Sorry! There are no jobs to display.");

}else{

return new ModelAndView("job-list", "jobs", jobs);

}

} catch (HibernateException e) {

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", e.getMessage());

}

}

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

protected ModelAndView updateJobData(HttpServletRequest request) throws Exception{

try{

JobApplication jobApplication = jobApplicationDao.getHired(Integer.parseInt(request.getParameter("job")));

try {

Email email = new SimpleEmail();

email.setHostName("smtp.googlemail.com");//If a server is capable of sending email, then you don't need the authentication. In this case, an email server needs to be running on that machine. Since we are running this application on the localhost and we don't have a email server, we are simply asking gmail to relay this email.

email.setSmtpPort(587);

email.setAuthenticator(new DefaultAuthenticator("portaljobs95@gmail.com", "jobportal95"));

email.setSSLOnConnect(true);

email.setFrom("portaljobs95@gmail.com");//This email will appear in the from field of the sending email. It doesn't have to be a real email address.This could be used for phishing/spoofing!

email.setSubject("Update on your Application");

email.setMsg("Congratulations! you have been hired for"+ jobApplication.getJob().getJobTitle() + "position at" + jobApplication.getJob().getCompany().getCompanyName());

email.addTo(jobApplication.getPerson().getEmailID());//Will come from the database

email.send();

} catch (Exception e) {

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

}

int updateJobApplication = jobApplicationDao.updateJobApplication(Integer.parseInt(request.getParameter("job")));

int updateJob = jobDao.updateJob(Integer.parseInt(request.getParameter("job")));

User user = (User) request.getSession().getAttribute("user");

int userID = user.getUserID();

List<Job> jobs= jobDao.jobList(userID);

if(jobs.isEmpty()){

return new ModelAndView("error","errorMessage","All the job hirings are completed");

}

else{

return new ModelAndView("applications-view","jobs", jobs);

}

}

catch(HibernateException e){

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", e.getMessage());

}

}

}

**JobSeekerController:**

package com.ketul.jobportal.controller;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import org.apache.commons.mail.DefaultAuthenticator;

import org.apache.commons.mail.Email;

import org.apache.commons.mail.SimpleEmail;

import org.hibernate.HibernateException;

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.annotation.ModelAttribute;

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

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

import org.springframework.web.servlet.ModelAndView;

import com.ketul.jobportal.dao.DAO;

import com.ketul.jobportal.dao.JobApplicationDAO;

import com.ketul.jobportal.dao.JobDAO;

import com.ketul.jobportal.pojo.Company;

import com.ketul.jobportal.pojo.Job;

import com.ketul.jobportal.pojo.JobApplication;

import com.ketul.jobportal.pojo.Person;

import com.ketul.jobportal.pojo.User;

@Controller

public class JobController {

@Autowired

@Qualifier("jobDao")

JobDAO jobDao;

@Autowired

@Qualifier("jobApplicationDao")

JobApplicationDAO jobApplicationDao;

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

protected ModelAndView jobPost() throws Exception{

return new ModelAndView("post-job", "job", new Job());

}

@RequestMapping(value="/job/post", method=RequestMethod.POST)

protected ModelAndView jobAdd(HttpServletRequest request, @ModelAttribute("job") Job job, BindingResult result) throws Exception{

try{

DAO.begin();

job.setJobTitle(request.getParameter("jobTitle"));

job.setJobType(request.getParameter("jobType"));

job.setJobField(request.getParameter("jobField"));

job.setContactEmailID(request.getParameter("contactEmailID"));

job.setDescription(request.getParameter("jobDescription"));

job.setStatus("hiring");

job.setCompany((Company) request.getSession().getAttribute("user"));

DAO.getSession().save(job);

DAO.commit();

return new ModelAndView("post-success", "job", job);

}

catch(HibernateException e){

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", e.getMessage());

}

}

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

protected ModelAndView viewJob(HttpServletRequest request) throws Exception {

try {

List<Job> jobs = jobDao.list();

if(jobs.isEmpty()){

return new ModelAndView("error","errorMessage","Sorry! There are no jobs to display.");

}else{

return new ModelAndView("job-list", "jobs", jobs);

}

} catch (HibernateException e) {

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", e.getMessage());

}

}

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

protected ModelAndView updateJobData(HttpServletRequest request) throws Exception{

try{

JobApplication jobApplication = jobApplicationDao.getHired(Integer.parseInt(request.getParameter("job")));

try {

Email email = new SimpleEmail();

email.setHostName("smtp.googlemail.com");//If a server is capable of sending email, then you don't need the authentication. In this case, an email server needs to be running on that machine. Since we are running this application on the localhost and we don't have a email server, we are simply asking gmail to relay this email.

email.setSmtpPort(587);

email.setAuthenticator(new DefaultAuthenticator("portaljobs95@gmail.com", "jobportal95"));

email.setSSLOnConnect(true);

email.setFrom("portaljobs95@gmail.com");//This email will appear in the from field of the sending email. It doesn't have to be a real email address.This could be used for phishing/spoofing!

email.setSubject("Update on your Application");

email.setMsg("Congratulations! you have been hired for"+ jobApplication.getJob().getJobTitle() + "position at" + jobApplication.getJob().getCompany().getCompanyName());

email.addTo(jobApplication.getPerson().getEmailID());//Will come from the database

email.send();

} catch (Exception e) {

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

}

int updateJobApplication = jobApplicationDao.updateJobApplication(Integer.parseInt(request.getParameter("job")));

int updateJob = jobDao.updateJob(Integer.parseInt(request.getParameter("job")));

User user = (User) request.getSession().getAttribute("user");

int userID = user.getUserID();

List<Job> jobs= jobDao.jobList(userID);

if(jobs.isEmpty()){

return new ModelAndView("error","errorMessage","All the job hirings are completed");

}

else{

return new ModelAndView("applications-view","jobs", jobs);

}

}

catch(HibernateException e){

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", e.getMessage());

}

}

}

**RegisterUserController:**

package com.ketul.jobportal.controller;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import org.apache.commons.mail.DefaultAuthenticator;

import org.apache.commons.mail.Email;

import org.apache.commons.mail.SimpleEmail;

import org.hibernate.HibernateException;

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

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

import org.springframework.mail.MailSender;

import org.springframework.mail.SimpleMailMessage;

import org.springframework.stereotype.Controller;

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 com.ketul.jobportal.dao.DAO;

import com.ketul.jobportal.dao.UserDAO;

import com.ketul.jobportal.pojo.Company;

import com.ketul.jobportal.pojo.Person;

import com.ketul.jobportal.pojo.User;

@Controller

@RequestMapping(value = "user/\*")

public class RegisterUserController {

@Autowired

@Qualifier("userDao")

UserDAO userDao;

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

protected String loginUser(HttpServletRequest request){

HttpSession session = request.getSession();

try{

User user = userDao.getUser(request.getParameter("emailID"), request.getParameter("password"));

if(user == null){

session.setAttribute("errorMessage", "UserName or Password does not exist");

return "error";

}

session.setAttribute("user", user);

return "home";

}

catch(HibernateException e){

session.setAttribute("errorMessage", e.getMessage());

return "error";

}

}

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

protected ModelAndView registerNewUser() throws Exception{

return new ModelAndView("register-user", "person" , new Person());

}

@RequestMapping(value = "/user/register", method = RequestMethod.POST)

protected ModelAndView register(HttpServletRequest request, @ModelAttribute("person") Person person, BindingResult result) throws Exception {

try{

User user = userDao.getUser(request.getParameter("emailID"), request.getParameter("password"));

if(user == null)

{

DAO.begin();

person.setEmailID(request.getParameter("emailID"));

person.setPassword(request.getParameter("password"));

person.setRole("employee");

person.setFirstName(request.getParameter("firstName"));

person.setLastName(request.getParameter("lastName"));

person.setCountry(request.getParameter("country"));

person.setState(request.getParameter("state"));

person.setZipCode(Integer.parseInt(request.getParameter("zipCode")));

person.setCurrentCareerLevel(request.getParameter("careerLevel"));

person.setEducationLevel(request.getParameter("educationLevel"));

DAO.getSession().save(person);

DAO.commit();

request.getSession().setAttribute("user", person);

return new ModelAndView("home", "user", person);

}

else{

return new ModelAndView("error","errorMessage","Seems you are already registered with "+ user.getEmailID() +" email id. Please try using a different id." );

}

}

catch(HibernateException e){

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", e.getMessage());

}

}

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

protected ModelAndView registerNewCompany() throws Exception{

return new ModelAndView("register-company","company", new Company());

}

@RequestMapping(value = "/user/company", method = RequestMethod.POST)

protected ModelAndView registerCompany(HttpServletRequest request, @ModelAttribute("company") Company company, BindingResult result) throws Exception{

try{

User user = userDao.getUser(request.getParameter("emailID"), request.getParameter("password"));

if(user == null)

{

DAO.begin();

company.setCompanyName(request.getParameter("companyName"));

company.setStreet(request.getParameter("street"));

company.setCity(request.getParameter("city"));

company.setState(request.getParameter("state"));

company.setCountry(request.getParameter("country"));

company.setRole("company");

company.setUrl(request.getParameter("companyUrl"));

company.setZipCode(Integer.parseInt(request.getParameter("zipCode")));

company.setEmailID(request.getParameter("emailID"));

company.setPassword(request.getParameter("password"));

DAO.getSession().save(company);

DAO.commit();

request.getSession().setAttribute("user", company);;

return new ModelAndView("home", "user", company);

}

else{

return new ModelAndView("error","errorMessage","Seems you are already registered with "+ user.getEmailID() +" email id. Please try using a different id." );

}

}

catch(HibernateException e){

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", e.getMessage());

}

}

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

protected ModelAndView goToUserHome(HttpServletRequest request) throws Exception {

User user = (User) request.getSession().getAttribute("user");

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

}

@RequestMapping(value="/user/logout")

public String logout(HttpServletRequest request){

request.getSession().invalidate();

return "redirect: /jobportal/index.jsp";

}

}

**ViewApplicaionsController:**

package com.ketul.jobportal.controller;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import org.hibernate.HibernateException;

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

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

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.ketul.jobportal.dao.JobApplicationDAO;

import com.ketul.jobportal.dao.JobDAO;

import com.ketul.jobportal.pojo.Job;

import com.ketul.jobportal.pojo.JobApplication;

import com.ketul.jobportal.pojo.User;

@Controller

public class ViewApplicationsController {

@Autowired

@Qualifier("jobDao")

JobDAO jobDao;

@Autowired

@Qualifier("jobApplicationDao")

JobApplicationDAO jobApplicationDao;

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

protected ModelAndView viewApplications(HttpServletRequest request)throws Exception {

try{

User user = (User) request.getSession().getAttribute("user");

int userID = user.getUserID();

List<Job> jobs= jobDao.jobList(userID);

if(jobs.isEmpty()){

return new ModelAndView("error","errorMessage","All the job hirings are completed");

}

else

{

return new ModelAndView("applications-view","jobs", jobs);

}

}

catch(HibernateException e){

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", e.getMessage());

}

}

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

protected ModelAndView viewApplicants(HttpServletRequest request) throws Exception{

try{

List<JobApplication> jobApplication=jobApplicationDao.getApplicants(Integer.parseInt(request.getParameter("job")));

if(jobApplication.isEmpty()){

return new ModelAndView("error", "errorMessage","We don't have any applications for now");

}

else{

return new ModelAndView("applicants-view","jobApplication",jobApplication);

}

}

catch(HibernateException e){

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", e.getMessage());

}

}

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

protected ModelAndView selectedApplicants(HttpServletRequest request) throws Exception{

try{

User user = (User) request.getSession().getAttribute("user");

int userID = user.getUserID();

System.out.println("1");

List<JobApplication> jobApplication=(List<JobApplication>) jobApplicationDao.getSelectedApplicants(userID);

System.out.println("2");

if(jobApplication == null){

return new ModelAndView("error", "errorMessage", "Sorry you don't have any hired applicants right now");

}

else{

return new ModelAndView("hired-applicants", "jobApplication", jobApplication );

}

}

catch(HibernateException e){

System.out.println(e.getMessage());

return new ModelAndView("error", "errorMessage", e.getMessage());

}

}

}