**com.subham.fitnessclubautomation**

FitnessclubautomationApplication.java

package com.subham.fitnessclubautomation;

import com.subham.fitnessclubautomation.model.Admin;

import com.subham.fitnessclubautomation.repository.AdminRepository;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

import java.awt.\*;

import java.io.IOException;

import java.net.URI;

import java.net.URISyntaxException;

@SpringBootApplication

public class FitnessclubautomationApplication {

public static void main(String[] args) {

try {

ApplicationContext context = SpringApplication.run(FitnessclubautomationApplication.class, args);

openHomePage();

} catch (Exception ex) {

System.out.println("ERROR - " + ex.getMessage());

}

}

private static void openHomePage() {

try {

URI homepage = new URI("http://localhost:8086/");

if (Desktop.isDesktopSupported()) {

Desktop desktop = Desktop.getDesktop();

try {

desktop.browse(homepage);

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

} else {

Runtime runtime = Runtime.getRuntime();

try {

runtime.exec("xdg-open " + homepage);

} catch (IOException e) {

// TODO Auto-generated catch block

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

}

}

} catch (URISyntaxException e) {

e.printStackTrace();

}

}

}

**Com.subham.fitnessclubautomation.controller**

AdminController.java

package com.subham.fitnessclubautomation.controller;

import com.subham.fitnessclubautomation.model.Admin;

import com.subham.fitnessclubautomation.service.AdminService;

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

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.bind.annotation.RequestParam;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

@Controller

public class AdminController {

@Autowired

AdminService adminService;

@RequestMapping("/")

private String home(ModelMap modelMap) {

modelMap.addAttribute("pagetitle", "Login");

return "login";

}

@RequestMapping("/login")

private String login(ModelMap modelMap,

HttpServletRequest request,

@RequestParam(value = "user\_email", required = true) String user\_email,

@RequestParam(value = "user\_password", required = true) String user\_password) {

modelMap.addAttribute("pagetitle", "Login");

Admin admin = adminService.login(user\_email, user\_password);

if (admin == null) {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", "Invalid Credentials! Please try again.");

return "login";

}

HttpSession session = request.getSession();

session.setAttribute("adminId", admin.getAdminId());

session.setAttribute("adminName", admin.getAdminName());

return "redirect:/subscriber";

}

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

public String logout(ModelMap modelMap, HttpServletRequest request) {

HttpSession session = request.getSession();

session.invalidate();

return "redirect:/";

}

}

PaymentController.java

package com.subham.fitnessclubautomation.controller;

import com.subham.fitnessclubautomation.model.Payment;

import com.subham.fitnessclubautomation.model.Subscriber;

import com.subham.fitnessclubautomation.service.PaymentService;

import com.subham.fitnessclubautomation.service.SubscriberService;

import com.subham.fitnessclubautomation.service.SubscriptionPlanService;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

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

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

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

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

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

@Controller

public class PaymentController {

@Autowired

PaymentService paymentService;

@Autowired

SubscriberService subscriberService;

@Autowired

SubscriptionPlanService subscriptionPlanService;

@RequestMapping(value = "/payment/add-new")

private String addPayment(ModelMap modelMap, HttpServletRequest request) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

String[] payment\_mode = {"Cash", "Credit/Debit Card", "UPI"};

List<Subscriber> subscribers = subscriberService.getAllSubscribers();

if (subscribers.size() > 0) {

modelMap.addAttribute("subscribers", subscribers);

modelMap.addAttribute("payment\_mode", payment\_mode);

} else {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", "No subscriber found for initiating payment.<br>");

}

return "add-payment";

}

@RequestMapping(value = "/payment/add", method = RequestMethod.POST)

private String addPayment(ModelMap modelMap, HttpServletRequest request, @RequestParam(value = "payment\_amount", required = true) int payment\_amount, @RequestParam(value = "payment\_mode", required = true) String payment\_mode, @RequestParam(value = "payment\_subscriber\_id", required = true) Long payment\_subscriber\_id) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

Subscriber s = subscriberService.getSubscriber(payment\_subscriber\_id);

Long planId = s.getSubscriptionPlan().getPlanId();

int planFees = subscriptionPlanService.getSubscriptionPlan(planId).getPlanFees();

int paid = s.getSubscriberFeesPaid();

int total = paid + payment\_amount;

if (total > planFees) {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", "Payment amount exceeding the actual fees limit.<br>");

return "add-payment";

}

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

Date date = new Date();

String payment\_date = sdf.format(date);

Payment payment = new Payment(payment\_amount, payment\_date, payment\_mode, s);

paymentService.addPayment(payment);

s.setSubscriberFeesPaid(total);

subscriberService.addSubscriber(s);

return "redirect:/payment";

}

@RequestMapping(value = "/payment/edit-payment/{payment\_id}")

private String editPayment(ModelMap modelMap, HttpServletRequest request, @PathVariable Long payment\_id) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

Payment payment = new Payment();

try {

payment = paymentService.getPayment(payment\_id);

modelMap.addAttribute("payment", payment);

String[] payment\_mode = {"Cash", "Credit/Debit Card", "UPI"};

modelMap.addAttribute("payment\_mode", payment\_mode);

List<Subscriber> subscribers = subscriberService.getAllSubscribers();

modelMap.addAttribute("subscribers", subscribers);

return "edit-payment";

} catch (Exception ex) {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", ex.getMessage());

return "redirect:/payment-list";

}

}

@RequestMapping(value = "/payment/edit", method = RequestMethod.POST)

private String editPayment(ModelMap modelMap, HttpServletRequest request, @RequestParam(value = "payment\_id", required = true) Long payment\_id, @RequestParam(value = "payment\_amount", required = true) int payment\_amount, @RequestParam(value = "old\_payment\_amount", required = true) int old\_payment\_amount, @RequestParam(value = "payment\_date", required = true) String payment\_date, @RequestParam(value = "payment\_mode", required = true) String payment\_mode, @RequestParam(value = "payment\_subscriber\_id", required = true) Long payment\_subscriber\_id) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

Subscriber subscriber = subscriberService.getSubscriber(payment\_subscriber\_id);

int old\_total = subscriber.getSubscriberFeesPaid() - old\_payment\_amount;

int new\_total = old\_total + payment\_amount;

int planFees = subscriber.getSubscriptionPlan().getPlanFees();

/\*System.out.println("FEES PAID - " + subscriber.getSubscriberFeesPaid());

System.out.println("PAYMENT AMOUNT - " + payment\_amount);

System.out.println("PLAN FEES - " + planFees);

System.out.println("OLD TOTAL - " + old\_total);

System.out.println("NEW TOTAL - " + new\_total);

\*/

if (new\_total > planFees) {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", "Payment amount exceeding the actual fees limit.<br>");

return "edit-payment";

}

Payment payment = new Payment(payment\_id, payment\_amount, payment\_date, payment\_mode, subscriber);

paymentService.addPayment(payment);

subscriber.setSubscriberFeesPaid(new\_total);

subscriberService.addSubscriber(subscriber);

return "redirect:/payment";

}

@RequestMapping(value = "/payment")

private String getAllPayments(ModelMap modelMap, HttpServletRequest request) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

List<Payment> payments = new ArrayList<>();

try {

payments = paymentService.getAllPayments();

modelMap.addAttribute("payments", payments);

modelMap.addAttribute("message", "Total <b>" + payments.size() + "</b> payment records found.");

} catch (Exception ex) {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", ex.getMessage() + "<br>");

}

return "payment-list";

}

@RequestMapping(value = "/payment/{payment\_id}")

private String getPayment(ModelMap modelMap, HttpServletRequest request, @PathVariable Long payment\_id) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

Payment payment = new Payment();

try {

payment = paymentService.getPayment(payment\_id);

modelMap.addAttribute("payment", payment);

return "payment-single";

} catch (Exception ex) {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", ex.getMessage() + "<br>");

return "redirect:/payment-list";

}

}

}

SubscriberController.java

package com.subham.fitnessclubautomation.controller;

import com.subham.fitnessclubautomation.model.Subscriber;

import com.subham.fitnessclubautomation.model.SubscriptionPlan;

import com.subham.fitnessclubautomation.model.Trainer;

import com.subham.fitnessclubautomation.service.SubscriberService;

import com.subham.fitnessclubautomation.service.SubscriptionPlanService;

import com.subham.fitnessclubautomation.service.TrainerService;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

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

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

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

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

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import java.util.ArrayList;

import java.util.List;

@Controller

public class SubscriberController {

@Autowired

SubscriberService subscriberService;

@Autowired

private TrainerService trainerService;

@Autowired

private SubscriptionPlanService subscriptionPlanService;

@RequestMapping(value = "/subscriber/add-new")

private String addSubscriber(ModelMap modelMap, HttpServletRequest request) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

List<Trainer> trainers = trainerService.getAllTrainers();

modelMap.addAttribute("trainers", trainers);

List<SubscriptionPlan> subscriptionPlans = subscriptionPlanService.getAllSubscriptionPlans();

modelMap.addAttribute("subscriptionPlans", subscriptionPlans);

return "add-subscriber";

}

@RequestMapping(value = "/subscriber/add", method = RequestMethod.POST)

private String addSubscriber(HttpServletRequest request,

@RequestParam(value = "subscriber\_name", required = true) String subscriber\_name,

@RequestParam(value = "subscriber\_age", required = true) int subscriber\_age,

@RequestParam(value = "subscriber\_gender", required = true) String subscriber\_gender,

@RequestParam(value = "subscriber\_address", required = true) String subscriber\_address,

@RequestParam(value = "subscriber\_trainer\_id", required = true) Long subscriber\_trainer\_id,

@RequestParam(value = "subscription\_plan\_id", required = true) Long subscription\_plan\_id) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

Trainer trainer = trainerService.getTrainer(subscriber\_trainer\_id);

SubscriptionPlan subscriptionPlan = subscriptionPlanService.getSubscriptionPlan(subscription\_plan\_id);

Subscriber subscriber = new Subscriber(subscriber\_name, subscriber\_age, subscriber\_gender, subscriber\_address, true, 0, trainer, subscriptionPlan);

if (subscriberService.addSubscriber(subscriber) != null) {

return "redirect:/subscriber";

} else {

return "redirect:/subscriber";

}

}

@RequestMapping(value = "/subscriber/edit-subscriber/{subscriber\_id}")

private String editSubscriber(ModelMap modelMap, HttpServletRequest request,

@PathVariable Long subscriber\_id) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

Subscriber subscriber = subscriberService.getSubscriber(subscriber\_id);

modelMap.addAttribute("subscriber", subscriber);

List<Trainer> trainers = trainerService.getAllTrainers();

modelMap.addAttribute("trainers", trainers);

List<SubscriptionPlan> subscriptionPlans = subscriptionPlanService.getAllSubscriptionPlans();

modelMap.addAttribute("subscriptionPlans", subscriptionPlans);

return "edit-subscriber";

}

@RequestMapping(value = "/subscriber/edit", method = RequestMethod.POST)

private String editSubscriber(ModelMap modelMap, HttpServletRequest request, @RequestParam(value = "subscriber\_id", required = true) Long subscriber\_id,

@RequestParam(value = "subscriber\_name", required = true) String subscriber\_name,

@RequestParam(value = "subscriber\_age", required = true) int subscriber\_age,

@RequestParam(value = "subscriber\_gender", required = true) String subscriber\_gender,

@RequestParam(value = "subscriber\_status", required = true) boolean subscriber\_status,

@RequestParam(value = "subscriber\_address", required = true) String subscriber\_address,

@RequestParam(value = "subscriber\_trainer\_id", required = true) Long subscriber\_trainer\_id,

@RequestParam(value = "subscription\_plan\_id", required = true) Long subscription\_plan\_id) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

if (subscriber\_trainer\_id == 0) {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", "Please select a valid trainer");

return "edit-subscriber";

}

if (subscription\_plan\_id == 0) {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", "Please select a valid subscription plan");

return "edit-subscriber";

}

Trainer trainer = trainerService.getTrainer(subscriber\_trainer\_id);

SubscriptionPlan subscriptionPlan = subscriptionPlanService.getSubscriptionPlan(subscription\_plan\_id);

int paid\_fees = subscriberService.getSubscriber(subscriber\_id).getSubscriberFeesPaid();

Subscriber subscriber = new Subscriber(subscriber\_id, subscriber\_name, subscriber\_age, subscriber\_gender, subscriber\_address, subscriber\_status, paid\_fees, trainer, subscriptionPlan);

subscriber = subscriberService.addSubscriber(subscriber);

return "redirect:/subscriber";

}

@RequestMapping(value = "/subscriber/delete/{subscriber\_id}")

private String deleteSubscriber(@PathVariable Long subscriber\_id, HttpServletRequest request) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

try {

subscriberService.deleteSubscriber(subscriber\_id);

return "redirect:/subscriber";

} catch (Exception ex) {

return "redirect:/subscriber";

}

}

@RequestMapping(value = "/subscriber")

private String getAllSubscribers(ModelMap modelMap, HttpServletRequest request) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

List<Subscriber> subscribers = new ArrayList<>();

try {

subscribers = subscriberService.getAllSubscribers();

modelMap.addAttribute("subscriber\_list", subscribers);

modelMap.addAttribute("message", "Total <b>" + subscribers.size() + "</b> subscribers found.");

return "subscriber-list";

} catch (Exception ex) {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", ex.getMessage() + "");

return "subscriber-list";

}

}

@RequestMapping(value = "/subscriber/{subscriber\_id}")

private String getSubscriber(ModelMap modelMap, HttpServletRequest request, @PathVariable Long subscriber\_id) {

Subscriber subscriber = new Subscriber();

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

try {

subscriber = subscriberService.getSubscriber(subscriber\_id);

modelMap.addAttribute("subscriber", subscriber);

return "subscriber-single";

} catch (Exception ex) {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", ex.getMessage());

return "redirect:/subscriber/{subscriber\_id}";

}

}

}

SubscriptionController.java

package com.subham.fitnessclubautomation.controller;

import com.subham.fitnessclubautomation.model.SubscriptionPlan;

import com.subham.fitnessclubautomation.service.SubscriptionPlanService;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

import org.springframework.web.bind.annotation.\*;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import java.util.ArrayList;

import java.util.List;

@Controller

public class SubscriptionController {

@Autowired

SubscriptionPlanService subscriptionPlanService;

@RequestMapping(value = "/subscriptionplan/add-new")

private String addSubscriptionPlan(ModelMap modelMap, HttpServletRequest request) {

modelMap.addAttribute("pageTitle", "Add new Plan");

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

return "add-plan";

}

@RequestMapping(value = "/subscriptionplan/add", method = RequestMethod.POST)

private String addSubscriptionPlan(ModelMap modelMap, HttpServletRequest request,

@RequestParam(value = "plan\_title", required = true) String plan\_title,

@RequestParam(value = "plan\_duration", required = true) int plan\_duration,

@RequestParam(value = "plan\_fees", required = true) int plan\_fees) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

if (plan\_duration <= 0) {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", "Invalid duration");

return "add-plan";

}

if (plan\_fees <= 0) {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", "Invalid fees");

return "add-plan";

}

SubscriptionPlan subscriptionPlan = new SubscriptionPlan(plan\_title, plan\_duration, plan\_fees);

subscriptionPlanService.addSubscriptionPlan(subscriptionPlan);

modelMap.addAttribute("success", true);

modelMap.addAttribute("message", "Plan added successfully.");

return "redirect:/subscriptionplan";

}

@RequestMapping(value = "/subscriptionplan/edit-plan/{plan\_id}")

private String editSubscriptionPlan(ModelMap modelMap, HttpServletRequest request, @PathVariable Long plan\_id) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

SubscriptionPlan plan = null;

try {

plan = subscriptionPlanService.getSubscriptionPlan(plan\_id);

if (plan != null) {

modelMap.addAttribute("plan", plan);

return "edit-plan";

} else {

return "redirect:/subscriptionplan";

}

} catch (Exception ex) {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", ex.getMessage());

return "redirect:/subscriptionplan";

}

}

@RequestMapping(value = "/subscriptionplan/edit", method = RequestMethod.POST)

private String editSubscriptionPlan(ModelMap modelMap, HttpServletRequest request,

@RequestParam(value = "plan\_id", required = true) Long plan\_id,

@RequestParam(value = "plan\_title", required = true) String plan\_title,

@RequestParam(value = "plan\_duration", required = true) int plan\_duration,

@RequestParam(value = "plan\_fees", required = true) int plan\_fees) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

if (plan\_duration <= 0) {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", "Invalid duration");

return "redirect:/subscriptionplan/edit-plan/" + plan\_id;

}

if (plan\_fees <= 0) {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", "Invalid fees");

return "redirect:/subscriptionplan/edit-plan/" + plan\_id;

}

SubscriptionPlan subscriptionPlan = new SubscriptionPlan(plan\_id, plan\_title, plan\_duration, plan\_fees);

subscriptionPlanService.addSubscriptionPlan(subscriptionPlan);

modelMap.addAttribute("success", true);

modelMap.addAttribute("message", "Plan data saved successfully.");

return "redirect:/subscriptionplan";

}

@RequestMapping(value = "/subscriptionplan/delete/{plan\_id}")

private String deleteSubscriptionPlan(ModelMap modelMap, HttpServletRequest request, @PathVariable Long plan\_id) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

try {

subscriptionPlanService.deleteSubscriptionPlan(plan\_id);

return "redirect:/subscriptionplan";

} catch (Exception ex) {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", ex.getMessage());

return "redirect:/subscriptionplan";

}

}

@RequestMapping(value = "/subscriptionplan")

private String getAllSubscriptionPlans(ModelMap modelMap, HttpServletRequest request) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

List<SubscriptionPlan> subscriptionPlans = new ArrayList<>();

try {

subscriptionPlans = subscriptionPlanService.getAllSubscriptionPlans();

modelMap.addAttribute("subscriptionPlans", subscriptionPlans);

modelMap.addAttribute("success", true);

modelMap.addAttribute("message", "Total <b>" + subscriptionPlans.size() + "</b> subscription plans found.");

return "plan-list";

} catch (Exception ex) {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", ex.getMessage());

return "plan-list";

}

}

@RequestMapping(value = "/subscriptionplan/{plan\_id}")

private String getSubscriptionPlan(ModelMap modelMap, HttpServletRequest request, @PathVariable Long plan\_id) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

SubscriptionPlan subscriptionPlan = new SubscriptionPlan();

try {

subscriptionPlan = subscriptionPlanService.getSubscriptionPlan(plan\_id);

modelMap.addAttribute("plan", subscriptionPlan);

return "plan-single";

} catch (Exception ex) {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", ex.getMessage());

return "redirect:/subscriptionplan";

}

}

}

TrainerController.java

package com.subham.fitnessclubautomation.controller;

import com.subham.fitnessclubautomation.model.Trainer;

import com.subham.fitnessclubautomation.service.TrainerService;

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

import org.springframework.http.HttpRequest;

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

import org.springframework.web.bind.annotation.\*;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import java.util.ArrayList;

import java.util.List;

@Controller

public class TrainerController {

@Autowired

TrainerService trainerService;

@RequestMapping(value = "/trainer/add-new")

private String addTrainer() {

return "add-trainer";

}

@RequestMapping(value = "/trainer/add", method = RequestMethod.POST)

private String addTrainer(ModelMap modelMap, HttpServletRequest request,

@RequestParam(value = "trainer\_name", required = true) String trainer\_name,

@RequestParam(value = "trainer\_age", required = true) int trainer\_age,

@RequestParam(value = "trainer\_gender", required = true) String trainer\_gender,

@RequestParam(value = "trainer\_experience", required = true) int trainer\_experience,

@RequestParam(value = "trainer\_address", required = true) String trainer\_address) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

Trainer tr = new Trainer(trainer\_name, trainer\_age, trainer\_gender, trainer\_experience, trainer\_address);

tr = trainerService.addTrainer(tr);

return "redirect:/trainer";

}

@RequestMapping(value = "/trainer/edit-trainer/{trainer\_id}")

private String editTrainer(ModelMap modelMap, HttpServletRequest request,

@PathVariable Long trainer\_id) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

int[] experience = {1, 2, 3, 4, 5, 6, 7, 8, 9};

modelMap.addAttribute("experience", experience);

Trainer trainer = trainerService.getTrainer(trainer\_id);

modelMap.addAttribute("trainer", trainer);

return "edit-trainer";

}

@RequestMapping(value = "/trainer/edit", method = RequestMethod.POST)

private String editTrainer(ModelMap modelMap, HttpServletRequest request,

@RequestParam(value = "trainer\_id", required = true) Long trainer\_id,

@RequestParam(value = "trainer\_name", required = true) String trainer\_name,

@RequestParam(value = "trainer\_age", required = true) int trainer\_age,

@RequestParam(value = "trainer\_gender", required = true) String trainer\_gender,

@RequestParam(value = "trainer\_experience", required = true) int trainer\_experience,

@RequestParam(value = "trainer\_address", required = true) String trainer\_address) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

Trainer tr = new Trainer(trainer\_id, trainer\_name, trainer\_age, trainer\_gender, trainer\_experience, trainer\_address);

trainerService.addTrainer(tr);

return "redirect:/trainer";

}

@RequestMapping(value = "/trainer/delete/{trainer\_id}", method = RequestMethod.GET)

private String deleteTrainer(ModelMap modelMap, HttpServletRequest request,

@PathVariable Long trainer\_id) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

try {

trainerService.deleteTrainer(trainer\_id);

modelMap.addAttribute("success", true);

modelMap.addAttribute("message", "Data deleted successfully.");

return "redirect:/trainer";

} catch (Exception ex) {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", ex.getMessage());

return "redirect:/trainer";

}

}

@RequestMapping(value = "/trainer")

private String getAllTrainers(ModelMap modelMap, HttpServletRequest request) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

List<Trainer> trainers = new ArrayList<>();

try {

trainers = trainerService.getAllTrainers();

if (trainers.size() > 0) {

modelMap.addAttribute("trainers", trainers);

modelMap.addAttribute("message", "Total <b>" + trainers.size() + "</b> trainers found.");

} else {

modelMap.addAttribute("message", "Total <b>" + trainers.size() + "</b> trainers found.");

}

return "trainer-list";

} catch (Exception ex) {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", ex.getMessage());

return "trainer-list";

}

}

@RequestMapping(value = "/trainer/{trainer\_id}")

private String getTrainer(ModelMap modelMap, HttpServletRequest request, @PathVariable Long trainer\_id) {

HttpSession session = request.getSession();

if (session.getAttribute("adminId") == null) {

return "redirect:/";

}

Trainer trainer = new Trainer();

try {

trainer = trainerService.getTrainer(trainer\_id);

modelMap.addAttribute("trainer", trainer);

return "trainer-single";

} catch (Exception ex) {

modelMap.addAttribute("error", true);

modelMap.addAttribute("message", ex.getMessage());

return "redirect:/trainer";

}

}}

**com.subham.fitnessclubautomation.model**

Admin.java

package com.subham.fitnessclubautomation.model;

import javax.persistence.\*;

@Entity

@Table(name = "tbl\_admin")

public class Admin {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "admin\_id", nullable = false)

private Long adminId;

@Column(name = "admin\_name", nullable = false)

private String adminName;

@Column(name = "admin\_email", nullable = false)

private String adminEmail;

@Column(name = "admin\_password", nullable = false)

private String adminPassword;

public Admin() {

}

public Admin(Long adminId, String adminName, String adminEmail, String adminPassword) {

this.adminId = adminId;

this.adminName = adminName;

this.adminEmail = adminEmail;

this.adminPassword = adminPassword;

}

public Admin(String adminName, String adminEmail, String adminPassword) {

this.adminName = adminName;

this.adminEmail = adminEmail;

this.adminPassword = adminPassword;

}

public Long getAdminId() {

return adminId;

}

public void setAdminId(Long adminId) {

this.adminId = adminId;

}

public String getAdminName() {

return adminName;

}

public void setAdminName(String adminName) {

this.adminName = adminName;

}

public String getAdminEmail() {

return adminEmail;

}

public void setAdminEmail(String adminEmail) {

this.adminEmail = adminEmail;

}

public String getAdminPassword() {

return adminPassword;

}

public void setAdminPassword(String adminPassword) {

this.adminPassword = adminPassword;

}

}

Payment.java

package com.subham.fitnessclubautomation.model;

import javax.persistence.\*;

@Entity

@Table(name = "tbl\_subscriberpayments")

public class Payment {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "payment\_id", nullable = false)

private Long paymentId;

@Column(name = "payment\_amount", nullable = false)

private int paidAmount;

@Column(name = "payment\_date", nullable = false)

private String paymentDate;

@Column(name = "payment\_mode", nullable = false)

private String paymentMode;

@ManyToOne

@JoinColumn(name = "payment\_subscriber\_id")

private Subscriber subscriber;

public Subscriber getSubscriber() {

return subscriber;

}

public void setSubscriber(Subscriber subscriber) {

this.subscriber = subscriber;

}

public Payment() {

}

public Payment(Long paymentId, int paidAmount, String paymentDate, String paymentMode, Subscriber subscriber) {

this.paymentId = paymentId;

this.paidAmount = paidAmount;

this.paymentDate = paymentDate;

this.paymentMode = paymentMode;

this.subscriber = subscriber;

}

public Payment(int paidAmount, String paymentDate, String paymentMode, Subscriber subscriber) {

this.paidAmount = paidAmount;

this.paymentDate = paymentDate;

this.paymentMode = paymentMode;

this.subscriber = subscriber;

}

public Long getPaymentId() {

return paymentId;

}

public void setPaymentId(Long paymentId) {

this.paymentId = paymentId;

}

public int getPaidAmount() {

return paidAmount;

}

public void setPaidAmount(int paidAmount) {

this.paidAmount = paidAmount;

}

public String getPaymentDate() {

return paymentDate;

}

public void setPaymentDate(String paymentDate) {

this.paymentDate = paymentDate;

}

public String getPaymentMode() {

return paymentMode;

}

public void setPaymentMode(String paymentMode) {

this.paymentMode = paymentMode;

}

}

Subscriber.java

package com.subham.fitnessclubautomation.model;

import javax.persistence.\*;

@Entity

@Table(name = "tbl\_subscriber")

public class Subscriber {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "subscriber\_id", nullable = false)

private Long subscriberId;

@Column(name = "subscriber\_name", nullable = false)

private String subscriberName;

@Column(name = "subscriber\_age", nullable = false)

private int subscriberAge;

@Column(name = "subscriber\_gender", nullable = false)

private String subscriberGender;

@Column(name = "subscriber\_address", nullable = false)

private String subscriberAddress;

@Column(name = "subscriber\_status", nullable = false)

private boolean subscriberStatus;

@Column(name = "subscriber\_fees\_paid", nullable = false)

private int subscriberFeesPaid;

@ManyToOne

@JoinColumn(name = "subscriber\_trainer\_id")

private Trainer trainer;

@ManyToOne

@JoinColumn(name = "subscription\_plan\_id")

private SubscriptionPlan subscriptionPlan;

public Subscriber() {

}

public Subscriber(Long subscriberId, String subscriberName, int subscriberAge, String subscriberGender, String subscriberAddress, boolean subscriberStatus, int subscriberFeesPaid, Trainer trainer, SubscriptionPlan subscriptionPlan) {

this.subscriberId = subscriberId;

this.subscriberName = subscriberName;

this.subscriberAge = subscriberAge;

this.subscriberGender = subscriberGender;

this.subscriberAddress = subscriberAddress;

this.subscriberStatus = subscriberStatus;

this.subscriberFeesPaid = subscriberFeesPaid;

this.trainer = trainer;

this.subscriptionPlan = subscriptionPlan;

}

public Subscriber(String subscriberName, int subscriberAge, String subscriberGender, String subscriberAddress, boolean subscriberStatus, int subscriberFeesPaid, Trainer trainer, SubscriptionPlan subscriptionPlan) {

this.subscriberName = subscriberName;

this.subscriberAge = subscriberAge;

this.subscriberGender = subscriberGender;

this.subscriberAddress = subscriberAddress;

this.subscriberStatus = subscriberStatus;

this.subscriberFeesPaid = subscriberFeesPaid;

this.trainer = trainer;

this.subscriptionPlan = subscriptionPlan;

}

public Long getSubscriberId() {

return subscriberId;

}

public void setSubscriberId(Long subscriberId) {

this.subscriberId = subscriberId;

}

public String getSubscriberName() {

return subscriberName;

}

public void setSubscriberName(String subscriberName) {

this.subscriberName = subscriberName;

}

public int getSubscriberAge() {

return subscriberAge;

}

public void setSubscriberAge(int subscriberAge) {

this.subscriberAge = subscriberAge;

}

public String getSubscriberGender() {

return subscriberGender;

}

public void setSubscriberGender(String subscriberGender) {

this.subscriberGender = subscriberGender;

}

public String getSubscriberAddress() {

return subscriberAddress;

}

public void setSubscriberAddress(String subscriberAddress) {

this.subscriberAddress = subscriberAddress;

}

public boolean isSubscriberStatus() {

return subscriberStatus;

}

public void setSubscriberStatus(boolean subscriberStatus) {

this.subscriberStatus = subscriberStatus;

}

public int getSubscriberFeesPaid() {

return subscriberFeesPaid;

}

public void setSubscriberFeesPaid(int subscriberFeesPaid) {

this.subscriberFeesPaid = subscriberFeesPaid;

}

public Trainer getTrainer() {

return trainer;

}

public void setTrainer(Trainer trainer) {

this.trainer = trainer;

}

public SubscriptionPlan getSubscriptionPlan() {

return subscriptionPlan;

}

public void setSubscriptionPlan(SubscriptionPlan subscriptionPlan) {

this.subscriptionPlan = subscriptionPlan;

}

}

SubscriptionPlan.java

package com.subham.fitnessclubautomation.model;

import javax.persistence.\*;

import java.util.List;

@Entity

@Table(name = "tbl\_subscriptionplan")

public class SubscriptionPlan {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "plan\_id", nullable = false)

private Long planId;

@Column(name = "plan\_title", nullable = false)

private String planTitle;

@Column(name = "plan\_duration", nullable = false)

private int planDuration;

@Column(name = "plan\_fees", nullable = false)

private int planFees;

public SubscriptionPlan() {

}

public SubscriptionPlan(Long planId, String planTitle, int planDuration, int planFees) {

this.planId = planId;

this.planTitle = planTitle;

this.planDuration = planDuration;

this.planFees = planFees;

}

public SubscriptionPlan(String planTitle, int planDuration, int planFees) {

this.planTitle = planTitle;

this.planDuration = planDuration;

this.planFees = planFees;

}

public Long getPlanId() {

return planId;

}

public void setPlanId(Long planId) {

this.planId = planId;

}

public String getPlanTitle() {

return planTitle;

}

public void setPlanTitle(String planTitle) {

this.planTitle = planTitle;

}

public int getPlanDuration() {

return planDuration;

}

public void setPlanDuration(int planDuration) {

this.planDuration = planDuration;

}

public int getPlanFees() {

return planFees;

}

public void setPlanFees(int planFees) {

this.planFees = planFees;

}

}

Trainer.java

package com.subham.fitnessclubautomation.model;

import javax.persistence.\*;

@Entity

@Table(name = "tbl\_trainer")

public class Trainer {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "trainer\_id", nullable = false)

private Long trainerId;

@Column(name = "trainer\_name", nullable = false)

private String trainerName;

@Column(name = "trainer\_age", nullable = false)

private int trainerAge;

@Column(name = "trainer\_gender", nullable = false)

private String trainerGender;

@Column(name = "trainer\_experience", nullable = false)

private int trainerExperience;

@Column(name = "trainer\_address", nullable = false)

private String trainerAddress;

public Trainer() {

}

public Trainer(Long trainerId, String trainerName, int trainerAge, String trainerGender, int trainerExperience, String trainerAddress) {

this.trainerId = trainerId;

this.trainerName = trainerName;

this.trainerAge = trainerAge;

this.trainerGender = trainerGender;

this.trainerExperience = trainerExperience;

this.trainerAddress = trainerAddress;

}

public Trainer(String trainerName, int trainerAge, String trainerGender, int trainerExperience, String trainerAddress) {

this.trainerName = trainerName;

this.trainerAge = trainerAge;

this.trainerGender = trainerGender;

this.trainerExperience = trainerExperience;

this.trainerAddress = trainerAddress;

}

public Long getTrainerId() {

return trainerId;

}

public void setTrainerId(Long trainerId) {

this.trainerId = trainerId;

}

public String getTrainerName() {

return trainerName;

}

public void setTrainerName(String trainerName) {

this.trainerName = trainerName;

}

public int getTrainerAge() {

return trainerAge;

}

public void setTrainerAge(int trainerAge) {

this.trainerAge = trainerAge;

}

public String getTrainerGender() {

return trainerGender;

}

public void setTrainerGender(String trainerGender) {

this.trainerGender = trainerGender;

}

public int getTrainerExperience() {

return trainerExperience;

}

public void setTrainerExperience(int trainerExperience) {

this.trainerExperience = trainerExperience;

}

public String getTrainerAddress() {

return trainerAddress;

}

public void setTrainerAddress(String trainerAddress) {

this.trainerAddress = trainerAddress;

}

}

**com.subham.fitnessclubautomation.repository**

AdminRepository.java

package com.subham.fitnessclubautomation.repository;

import com.subham.fitnessclubautomation.model.Admin;

import org.springframework.data.jpa.repository.Query;

import org.springframework.data.repository.CrudRepository;

import org.springframework.data.repository.query.Param;

public interface AdminRepository extends CrudRepository<Admin, Long> {

@Query("select a from Admin a where a.adminEmail = ?1 and a.adminPassword = ?2")

Admin login(String adminEmail, String adminPassword);

}

PaymentRepository.java

package com.subham.fitnessclubautomation.repository;

import com.subham.fitnessclubautomation.model.Payment;

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;

@Repository

public interface PaymentRepository extends CrudRepository<Payment, Long> {

}

SubscriberRepository.java

package com.subham.fitnessclubautomation.repository;

import com.subham.fitnessclubautomation.model.Subscriber;

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;

@Repository

public interface SubscriberRepository extends CrudRepository<Subscriber, Long> {

}

SubscriptionPlanRepository.java

package com.subham.fitnessclubautomation.repository;

import com.subham.fitnessclubautomation.model.SubscriptionPlan;

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;

@Repository

public interface SubscriptionPlanRepository extends CrudRepository<SubscriptionPlan, Long> {

}

TrainerRepository.java

package com.subham.fitnessclubautomation.repository;

import com.subham.fitnessclubautomation.model.Trainer;

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;

@Repository

public interface TrainerRepository extends CrudRepository<Trainer, Long> {

}

**com.subham.fitnessclubautomation.service**

AdminService.java

package com.subham.fitnessclubautomation.service;

import com.subham.fitnessclubautomation.model.Admin;

import com.subham.fitnessclubautomation.repository.AdminRepository;

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

import org.springframework.stereotype.Service;

@Service

public class AdminService {

@Autowired

AdminRepository adminRepository;

public Admin getAdmin(Long adminId) {

return adminRepository.findById(adminId).get();

}

public Admin login(String adminEmail, String adminPassword) {

Admin admin = adminRepository.login(adminEmail, adminPassword);

return admin;

}

}

PaymentService.java

package com.subham.fitnessclubautomation.service;

import com.subham.fitnessclubautomation.model.Payment;

import com.subham.fitnessclubautomation.repository.PaymentRepository;

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

import org.springframework.stereotype.Service;

import java.util.ArrayList;

import java.util.List;

@Service

public class PaymentService {

@Autowired

PaymentRepository paymentRepository;

public Payment addPayment(Payment p) {

return paymentRepository.save(p);

}

public void deletePayment(Long payment\_id) {

paymentRepository.deleteById(payment\_id);

}

public List<Payment> getAllPayments() {

List<Payment> payments = new ArrayList<>();

paymentRepository.findAll().forEach(payments::add);

return payments;

}

public Payment getPayment(Long payment\_id) {

return paymentRepository.findById(payment\_id).get();

}

}

SubscriberService.java

package com.subham.fitnessclubautomation.service;

import com.subham.fitnessclubautomation.model.Subscriber;

import com.subham.fitnessclubautomation.repository.SubscriberRepository;

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

import org.springframework.stereotype.Service;

import java.util.ArrayList;

import java.util.List;

@Service

public class SubscriberService {

@Autowired

SubscriberRepository subscriberRepository;

public Subscriber addSubscriber(Subscriber subscriber) {

return subscriberRepository.save(subscriber);

}

public void deleteSubscriber(Long subscriber\_id) {

subscriberRepository.deleteById(subscriber\_id);

}

public List<Subscriber> getAllSubscribers() {

List<Subscriber> subscribers = new ArrayList<>();

subscriberRepository.findAll().forEach(subscribers::add);

return subscribers;

}

public Subscriber getSubscriber(Long subscriber\_id) {

return subscriberRepository.findById(subscriber\_id).get();

}

}

SubscriptionPlanService.java

package com.subham.fitnessclubautomation.service;

import com.subham.fitnessclubautomation.model.SubscriptionPlan;

import com.subham.fitnessclubautomation.repository.SubscriptionPlanRepository;

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

import org.springframework.stereotype.Service;

import java.util.ArrayList;

import java.util.List;

@Service

public class SubscriptionPlanService {

@Autowired

SubscriptionPlanRepository subscriptionPlanRepository;

public SubscriptionPlan addSubscriptionPlan(SubscriptionPlan subscriptionPlan) {

return subscriptionPlanRepository.save(subscriptionPlan);

}

public void deleteSubscriptionPlan(Long subscriptionPlan\_id) {

subscriptionPlanRepository.deleteById(subscriptionPlan\_id);

}

public List<SubscriptionPlan> getAllSubscriptionPlans() {

List<SubscriptionPlan> subscriptionPlans = new ArrayList<>();

subscriptionPlanRepository.findAll().forEach(subscriptionPlans::add);

return subscriptionPlans;

}

public SubscriptionPlan getSubscriptionPlan(Long subscriptionPlan\_id) {

return subscriptionPlanRepository.findById(subscriptionPlan\_id).get(); }}

TrainerService.java

package com.subham.fitnessclubautomation.service;

import com.subham.fitnessclubautomation.model.Trainer;

import com.subham.fitnessclubautomation.repository.TrainerRepository;

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

import org.springframework.stereotype.Service;

import java.util.ArrayList;

import java.util.List;

@Service

public class TrainerService {

@Autowired

TrainerRepository trainerRepository;

public Trainer addTrainer(Trainer tr) {

return trainerRepository.save(tr);

}

public void deleteTrainer(Long trainer\_id) {

trainerRepository.deleteById(trainer\_id);

}

public List<Trainer> getAllTrainers() {

List<Trainer> trainers = new ArrayList<>();

trainerRepository.findAll().forEach(trainers::add);

return trainers;

}

public Trainer getTrainer(Long trainer\_id) {

return trainerRepository.findById(trainer\_id).get();

}

}

**src/main/resources**

application.properties

## JPA Properties

#spring.jpa.database=jdbc:mysql://localhost/sampledb

spring.jpa.hibernate.ddl-auto=update

spring.datasource.url=jdbc:mysql://localhost:3306/productdb

spring.datasource.username=root

spring.datasource.password=Subham@123

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.show-sql=true

#Server Configuration

server.port=8086

spring.mvc.view.prefix=/views/

spring.mvc.view.suffix=.jsp

Add-payment.jsp

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

pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

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

<title>Add Payments</title>

<style type="text/css">

td{

border: 1px solid black;

border-collapse: collapse;

padding:24px;

text-align: center;

}

.formdiv{

padding: 8px;

border: 1px solid black;

display: inline-block;

margin-top: 24px;

}

.input{

height: 20px;

width: 300px;

padding: 8px;

}

</style>

</head>

<body>

<jsp:include page="header.jsp"></jsp:include>

<h2>Payments</h2>

<c:if test="${error==true || success==true}">

${message}

</c:if>

<div class="formdiv">

<form action="add" method="POST">

<table>

<thead>

<tr>

<th colspan="2"><h3>Record a New Payment</h3></th>

</tr>

</thead>

<tbody>

<tr>

<td>

<label for="payment\_subscriber\_id">Payment By</label>

</td>

<td>

<select id="payment\_subscriber\_id" name="payment\_subscriber\_id">

<c:forEach items="${subscribers}" var="subscriber">

<option value=${subscriber.subscriberId}>${subscriber.subscriberName}</option>

</c:forEach>

</select>

</td>

</tr>

<tr>

<td>

<label for="payment\_amount">Paid Amount</label>

</td>

<td>

<input type="number" id="payment\_amount" class="input" name="payment\_amount">

</td>

</tr>

<tr>

<td>

<label for="payment\_mode">Payment Mode</label>

</td>

<td>

<select id="payment\_mode" name="payment\_mode">

<c:forEach items="${payment\_mode}" var="mode">

<option value=${mode}>${mode}</option>

</c:forEach>

</select>

</td>

</tr>

<tr>

<td colspan="2"><button type="submit">Submit</button></td>

</tr>

</tbody>

</table>

</form>

</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" %>

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

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

<title>Add Plan</title>

<style type="text/css">

/\* Reset some default styles for consistency \*/

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

/\* Body Styles \*/

body {

font-family: Arial, sans-serif;

background-color: #f2f2f2;

margin: 0;

padding: 0;

}

/\* Form Container Styles \*/

.form-container {

background-color: #fff;

border: 1px solid #000000;

border-radius: 10px;

padding: 20px;

width: 80%;

margin: 24px auto;

text-align: center;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

}

/\* Form Heading Styles \*/

.form-heading {

font-size: 24px;

margin-bottom: 20px;

}

/\* Form Input Styles \*/

.input {

height: 40px;

width: 100%;

padding: 10px;

margin: 10px 0;

border: 1px solid #ccc;

border-radius: 5px;

}

/\* Submit Button Styles \*/

.submit-button {

background-color: #0066cc;

color: #fff;

border: none;

padding: 12px 20px;

cursor: pointer;

border-radius: 5px;

}

.submit-button:hover {

background-color: #0052a3;

}

</style>

</head>

<body>

<jsp:include page="header.jsp"></jsp:include>

<div class="form-container">

<h2 class="form-heading">Add Subscription Plan</h2>

<c:if test="${error==true || success==true}">

<p>${message}</p>

</c:if>

<form action="add" method="POST">

<label for="plan\_title">Plan Title</label>

<input type="text" id="plan\_title" class="input" name="plan\_title">

<label for="plan\_duration">Plan Duration (Months)</label>

<input type="number" id="plan\_duration" class="input" name="plan\_duration">

<label for="plan\_fees">Plan Fees (INR)</label>

<input type="number" id="plan\_fees" class="input" name="plan\_fees">

<button class="submit-button" type="submit">Submit</button>

</form>

</div>

</body>

</html>

add-plan

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

pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

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

<title>Add Plan</title>

<style type="text/css">

/\* Reset some default styles for consistency \*/

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

/\* Body Styles \*/

body {

font-family: Arial, sans-serif;

background-color: #f2f2f2;

margin: 0;

padding: 0;

}

/\* Form Container Styles \*/

.form-container {

background-color: #fff;

border: 1px solid #000000;

border-radius: 10px;

padding: 20px;

width: 80%;

margin: 24px auto;

text-align: center;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

}

/\* Form Heading Styles \*/

.form-heading {

font-size: 24px;

margin-bottom: 20px;

}

/\* Form Input Styles \*/

.input {

height: 40px;

width: 100%;

padding: 10px;

margin: 10px 0;

border: 1px solid #ccc;

border-radius: 5px;

}

/\* Submit Button Styles \*/

.submit-button {

background-color: #0066cc;

color: #fff;

border: none;

padding: 12px 20px;

cursor: pointer;

border-radius: 5px;

}

.submit-button:hover {

background-color: #0052a3;

}

</style>

</head>

<body>

<jsp:include page="header.jsp"></jsp:include>

<div class="form-container">

<h2 class="form-heading">Add Subscription Plan</h2>

<c:if test="${error==true || success==true}">

<p>${message}</p>

</c:if>

<form action="add" method="POST">

<label for="plan\_title">Plan Title</label>

<input type="text" id="plan\_title" class="input" name="plan\_title">

<label for="plan\_duration">Plan Duration (Months)</label>

<input type="number" id="plan\_duration" class="input" name="plan\_duration">

<label for="plan\_fees">Plan Fees (INR)</label>

<input type="number" id="plan\_fees" class="input" name="plan\_fees">

<button class="submit-button" type="submit">Submit</button>

</form>

</div>

</body>

</html>

Add-subscriber.jsp

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

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

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

<title>Add Subscriber</title>

<style type="text/css">

/\* Reset some default styles for consistency \*/

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

/\* Body Styles \*/

body {

font-family: Arial, sans-serif;

background-color: #f2f2f2;

margin: 0;

padding: 0;

}

/\* Form Container Styles \*/

.form-container {

background-color: #fff;

border: 1px solid #000000;

border-radius: 10px;

padding: 20px;

width: 80%;

margin: 24px auto;

text-align: center;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

}

/\* Form Heading Styles \*/

.form-heading {

font-size: 24px;

margin-bottom: 20px;

}

/\* Form Input Styles \*/

.input {

height: 40px;

width: 100%;

padding: 10px;

margin: 10px 0;

border: 1px solid #ccc;

border-radius: 5px;

}

/\* Radio Button Styles \*/

.radio-label {

display: inline-block;

margin-right: 10px;

}

/\* Select Styles \*/

select {

height: 40px;

width: 100%;

padding: 10px;

margin: 10px 0;

border: 1px solid #ccc;

border-radius: 5px;

}

/\* Submit Button Styles \*/

.submit-button {

background-color: #0066cc;

color: #fff;

border: none;

padding: 12px 20px;

cursor: pointer;

border-radius: 5px;

}

.submit-button:hover {

background-color: #0052a3;

}

</style>

</head>

<body>

<jsp:include page="header.jsp"></jsp:include>

<div class="form-container">

<h2 class="form-heading">Add Subscriber</h2>

<c:if test="${error==true || success==true}">

<p>${message}</p>

</c:if>

<form action="add" method="POST">

<label for="subscriber\_name">Subscriber Name</label>

<input type="text" id="subscriber\_name" class="input" name="subscriber\_name" required>

<label for="subscriber\_age">Subscriber Age</label>

<input type="text" id="subscriber\_age" class="input" name="subscriber\_age" required>

<label for="subscriber\_gender">Subscriber Gender</label>

<label class="radio-label" for="subscriber\_gender\_male">

<input type="radio" id="subscriber\_gender\_male" name="subscriber\_gender" value="Male"> Male

</label>

<label class="radio-label" for="subscriber\_gender\_female">

<input type="radio" id="subscriber\_gender\_female" name="subscriber\_gender" value="Female"> Female

</label>

<label for="subscriber\_address">Subscriber Address</label>

<textarea id="subscriber\_address" class="input" style="height: 100px" required name="subscriber\_address"></textarea>

<label for="subscriber\_trainer\_id">Select Trainer</label>

<select id="subscriber\_trainer\_id" name="subscriber\_trainer\_id" class="input" required>

<option value="">Select Trainer</option>

<c:forEach items="${trainers}" var="trainer">

<option value="${trainer.trainerId}">${trainer.trainerName}</option>

</c:forEach>

</select>

<label for="subscription\_plan\_id">Subscription Plan</label>

<select id="subscription\_plan\_id" name="subscription\_plan\_id" class="input" required>

<c:forEach items="${subscriptionPlans}" var="plan">

<option value="${plan.planId}">${plan.planTitle}</option>

</c:forEach>

</select>

<button class="submit-button" type="submit">Submit</button>

</form>

</div>

</body>

</html>

Add-trainer.jsp

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

pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

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

<title>Add Trainer</title>

<style type="text/css">

td{

border: 1px solid black;

border-collapse: collapse;

padding:24px;

text-align: center;

}

.formdiv{

padding: 8px;

border: 1px solid black;

display: inline-block;

margin-top: 24px;

}

.input{

height: 20px;

width: 300px;

}

</style>

</head>

<body>

<jsp:include page="header.jsp" ></jsp:include>

<h2>Trainer Information</h2>

<c:if test="${error==true || success==true}">

${message}

</c:if>

<div class="formdiv">

<form action="add" method="POST">

<table>

<thead>

<tr>

<th colspan="2"><h3>Add new Trainer</h3></th>

</tr>

</thead>

<tbody>

<tr>

<td>

<label for="trainer\_name">Trainer Name</label>

</td>

<td>

<input type="text" id="trainer\_name" class="input" name="trainer\_name">

</td>

</tr>

<tr>

<td>

<label for="trainer\_age">Trainer Age</label>

</td>

<td>

<input type="text" id="trainer\_age" class="input" name="trainer\_age">

</td>

</tr>

<tr>

<td>

<label for="trainer\_gender">Trainer Gender</label>

</td>

<td>

<input type="text" id="trainer\_gender" class="input" name="trainer\_gender">

</td>

</tr>

<tr>

<td>

<label for="trainer\_experience">Trainer Experience</label>

</td>

<td>

<select id="trainer\_experience" name="trainer\_experience" class="input">

<option value="">Select Experience</option>

<option value="1">1</option>

<option value="1">1</option>

<option value="2">2</option>

<option value="3">3</option>

<option value="4">4</option>

<option value="5">5</option>

<option value="6">6</option>

<option value="7">7</option>

<option value="8">8</option>

<option value="9">9</option>

</select>

</td>

</tr>

<tr>

<td>

<label for="trainer\_address">Trainer Address</label>

</td>

<td>

<textarea type="text" id="trainer\_address" class="input" name="trainer\_address"></textarea>

</td>

</tr>

<tr>

<td colspan="2"><button type="submit">Submit</button></td>

</tr>

</tbody>

</table>

</form>

</div>

</body>

</html>

Centerwise\_citizens.jsp

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

pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<html>

<head>

<title>${pageTitle}</title>

<style type="text/css">

table,th,td{

border: 1px solid black;

border-collapse: collapse;

padding:8px;

text-align: center;

}

</style>

</head>

<body>

<jsp:include page="header.jsp"></jsp:include>

<h2>Citizens</h2>

<c:if test="${error==true || success==true}">

${message}

</c:if>

<table>

<thead>

<tr>

<th>ID</th>

<th>Name</th>

<th>City</th>

<th>No. Of Doses</th>

<th>Vaccination Status</th>

<th>Vaccination Center</th>

</tr>

</thead>

<tbody>

<c:forEach items="${citizens}" var="citizen">

<tr>

<td>${citizen.citizenId}</td>

<td>${citizen.citizenName}</td>

<td>${citizen.citizenCity}</td>

<td>${citizen.citizenDoses}</td>

<td>

<c:if test="${citizen.citizenDoses == 0}">

Not Vaccinated

</c:if>

<c:if test="${citizen.citizenDoses == 1}">

Partially Vaccinated

</c:if>

<c:if test="${citizen.citizenDoses == 2}">

Fully Vaccinated

</c:if>

</td>

<td><a href="../../vaccinationcenter/${citizen.citizenCenterId}">${citizen.citizenCenter}</a></td>

</tr>

</c:forEach>

</tbody>

<tfoot>

<tr>

<td colspan="6">${message}</td>

</tr>

</tfoot>

</table>

</body>

</html>

Edit-payment.jsp

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

pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

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

<title>Edit Payment :: ${payment.subscriber.subscriberName}</title>

<style type="text/css">

/\* Reset some default styles for consistency \*/

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

/\* Body Styles \*/

body {

font-family: Arial, sans-serif;

background-color: #f2f2f2;

margin: 0;

padding: 0;

}

/\* Form Container Styles \*/

.form-container {

background-color: #fff;

border: 1px solid #000000;

border-radius: 10px;

padding: 20px;

width: 80%;

margin: 24px auto;

text-align: center;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

}

/\* Form Heading Styles \*/

.form-heading {

font-size: 24px;

margin-bottom: 20px;

}

/\* Form Input Styles \*/

.input {

height: 40px;

width: 100%;

padding: 10px;

margin: 10px 0;

border: 1px solid #ccc;

border-radius: 5px;

}

/\* Submit Button Styles \*/

.submit-button {

background-color: #0066cc;

color: #fff;

border: none;

padding: 12px 20px;

cursor: pointer;

border-radius: 5px;

}

.submit-button:hover {

background-color: #0052a3;

}

</style>

</head>

<body>

<jsp:include page="header.jsp"></jsp:include>

<div class="form-container">

<h2 class="form-heading">Edit Payment</h2>

<c:if test="${error==true || success==true}">

<p>${message}</p>

</c:if>

<form action="/payment/edit" method="POST">

<label for="payment\_subscriber\_id">Payment By</label>

<input type="hidden" value="${payment\_id}" name="payment\_id">

<input type="hidden" value="${payment.subscriber.subscriberId}" name="payment\_subscriber\_id">

<label>${payment.subscriber.subscriberName}</label>

<label for="payment\_amount">Paid Amount</label>

<input type="hidden" id="old\_payment\_amount" class="input" name="old\_payment\_amount" value="${payment.paidAmount}">

<input type="number" id="payment\_amount" class="input" name="payment\_amount" value="${payment.paidAmount}">

<br><label>Out of INR. ${payment.subscriber.subscriptionPlan.planFees}</label>

<label for="payment\_date">Payment Date</label>

<input type="date" id="payment\_date" class="input" name="payment\_date" value="${payment.paymentDate}">

<label for="payment\_mode">Payment Mode</label>

<select id="payment\_mode" name="payment\_mode">

<c:forEach items="${payment\_mode}" var="mode">

<option value="${mode}" <c:if test="${mode == payment.paymentMode}">selected</c:if>>

${mode}

</option>

</c:forEach>

</select>

<button class="submit-button" type="submit">Submit</button>

</form>

</div>

</body>

</html>

Edit-plan.jsp

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

pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

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

<title>Edit Plan :: ${plan.planTitle}</title>

<style type="text/css">

td{

border: 1px solid black;

border-collapse: collapse;

padding:24px;

text-align: center;

}

.formdiv{

padding: 8px;

border: 1px solid black;

display: inline-block;

margin-top: 24px;

}

.input{

height: 20px;

width: 300px;

padding: 8px;

}

</style>

</head>

<body>

<jsp:include page="header.jsp"></jsp:include>

<h2>Plan Information</h2>

<c:if test="${error==true || success==true}">

${message}

</c:if>

<div class="formdiv">

<form action="/subscriptionplan/edit" method="POST">

<table>

<thead>

<tr>

<th colspan="2"><h3>Edit Plan - ${plan.planTitle}</h3></th>

</tr>

</thead>

<tbody>

<tr>

<td>

<label for="plan\_title">Plan Title</label>

</td>

<td>

<input type="hidden" id="plan\_id" class="hidden" name="plan\_id" value="${plan.planId}">

<input type="text" id="plan\_title" class="input" name="plan\_title" value="${plan.planTitle}">

</td>

</tr>

<tr>

<td>

<label for="plan\_duration">Plan Duration (Months)</label>

</td>

<td>

<input type="number" id="plan\_duration" class="input" name="plan\_duration" value="${plan.planDuration}">

</td>

</tr>

<tr>

<td>

<label for="plan\_fees">Plan Fees (INR)</label>

</td>

<td>

<input type="number" id="plan\_fees" class="input" name="plan\_fees" value="${plan.planFees}">

</td>

</tr>

<tr>

<td colspan="2"><button type="submit">Submit</button></td>

</tr>

</tbody>

</table>

</form>

</div>

</body>

</html>

Edit-sibscriber.jsp

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

pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

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

<title>Edit Subscriber :: ${subscriber.subscriberName}</title>

<style type="text/css">

td{

border: 1px solid black;

border-collapse: collapse;

padding:24px;

text-align: center;

}

.formdiv{

padding: 8px;

border: 1px solid black;

display: inline-block;

margin-top: 24px;

}

.input{

height: 20px;

width: 300px;

padding: 8px;

}

</style>

</head>

<body>

<jsp:include page="header.jsp" ></jsp:include>

<h2>Subscriber Information</h2>

<c:if test="${error==true || success==true}">

${message}

</c:if>

<div class="formdiv">

<form action="/subscriber/edit" method="POST">

<table>

<thead>

<tr>

<th colspan="2"><h3>Edit Subscriber</h3></th>

</tr>

</thead>

<tbody>

<tr>

<td>

<label for="subscriber\_name">Subscriber Name</label>

</td>

<td>

<input type="hidden" id="subscriber\_id" class="input" name="subscriber\_id" value="${subscriber.subscriberId}">

<input type="text" required id="subscriber\_name" class="input" name="subscriber\_name" value="${subscriber.subscriberName}">

</td>

</tr>

<tr>

<td>

<label for="subscriber\_age">Subscriber Age</label>

</td>

<td>

<input type="text" required id="subscriber\_age" class="input" name="subscriber\_age" value="${subscriber.subscriberAge}">

</td>

</tr>

<tr>

<td>

<label for="subscriber\_address">Subscriber Address</label>

</td>

<td>

<textarea id="subscriber\_address" required class="input" name="subscriber\_address" value="${subscriber.subscriberAge}">${subscriber.subscriberAddress}</textarea>

</td>

</tr>

<tr>

<td>

<label for="subscriber\_gender">Subscriber Gender</label>

</td>

<td>

<input type="radio" required id="subscriber\_gender\_male" name="subscriber\_gender" value="Male" <c:if test="${subscriber.subscriberGender == \"Male\"}">checked</c:if>>

<label for="subscriber\_gender\_male">Male</label>

<input type="radio" required id="subscriber\_gender\_female" name="subscriber\_gender" value="Female" <c:if test="${subscriber.subscriberGender == \"Female\"}">checked</c:if>>

<label for="subscriber\_gender\_female">Female</label>

</td>

</tr>

<tr>

<td>

<label for="subscriber\_trainer\_id">Subscriber Trainer</label>

</td>

<td>

<select id="subscriber\_trainer" name="subscriber\_trainer\_id">

<option value="0">Select Trainer</option>

<c:forEach items="${trainers}" var="trainer">

<option value="${trainer.trainerId}" <c:if test="${trainer.trainerId == subscriber.trainer.trainerId}">selected</c:if>>

${trainer.trainerName}

</option>

</c:forEach>

</select>

</td>

</tr>

<tr>

<td>

<label for="subscription\_plan\_id">Subscription Plan</label>

</td>

<td>

<select id="subscription\_plan\_id" name="subscription\_plan\_id">

<option value="0">Select Subscription Plan</option>

<c:forEach items="${subscriptionPlans}" var="plan">

<option value="${plan.planId}" <c:if test="${plan.planId == subscriber.subscriptionPlan.planId}">selected</c:if>>

${plan.planTitle}

</option>

</c:forEach>

</select>

</td>

</tr>

<tr>

<input type="hidden" value="${subscriber.subscriberStatus}" name="subscriber\_status">

<td colspan="2"><button type="submit">Submit</button></td>

</tr>

</tbody>

</table>

</form>

</div>

</body>

</html>

Edit-trainer.jsp

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

pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

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

<title>Edit Trainer :: ${trainer.trainerName}</title>

<style type="text/css">

td{

border: 1px solid black;

border-collapse: collapse;

padding:24px;

text-align: center;

}

.formdiv{

padding: 8px;

border: 1px solid black;

display: inline-block;

margin-top: 24px;

}

.input{

height: 20px;

width: 300px;

}

</style>

</head>

<body>

<jsp:include page="header.jsp" ></jsp:include>

<h2>Trainer Information</h2>

<c:if test="${error==true || success==true}">

${message}

</c:if>

<div class="formdiv">

<form action="/trainer/edit" method="POST">

<table>

<thead>

<tr>

<th colspan="2"><h3>Edit - ${trainer.trainerName}</h3></th>

</tr>

</thead>

<tbody>

<tr>

<td>

<label for="trainer\_name">Trainer Name</label>

</td>

<td>

<input type="hidden" id="trainer\_id" class="input" value="${trainer.trainerId}" name="trainer\_id">

<input type="text" id="trainer\_name" class="input" value="${trainer.trainerName}" name="trainer\_name">

</td>

</tr>

<tr>

<td>

<label for="trainer\_age">Trainer Age</label>

</td>

<td>

<input type="text" id="trainer\_age" class="input" value="${trainer.trainerAge}" name="trainer\_age">

</td>

</tr>

<tr>

<td>

<label for="trainer\_gender">Trainer Gender</label>

</td>

<td>

<input type="radio" id="trainer\_gender\_male" name="trainer\_gender" value="Male" <c:if test="${subscriber.subscriberGender == \"Male\"}">checked</c:if>>

<label for="trainer\_gender\_male">Male</label>

<input type="radio" id="trainer\_gender\_female" name="trainer\_gender" value="Female" <c:if test="${subscriber.subscriberGender == \"Female\"}">checked</c:if>>

<label for="trainer\_gender\_female">Female</label>

</td>

</tr>

<tr>

<td>

<label for="trainer\_experience">Trainer Experience</label>

</td>

<td>

<select id="trainer\_experience" name="trainer\_experience" class="input">

<option value="">Select Experience</option>

<c:forEach items="${experience}" var="ex">

<option value="${ex}" <c:if test="${ex == trainer.trainerExperience}">selected</c:if>>

${ex}

</option>

</c:forEach>

</select>

</td>

</tr>

<tr>

<td>

<label for="trainer\_address">Trainer Address</label>

</td>

<td>

<input type="text" id="trainer\_address" class="input" value="${trainer.trainerAddress}" name="trainer\_address">

</td>

</tr>

<tr>

<td colspan="2"><button type="submit">Submit</button></td>

</tr>

</tbody>

</table>

</form>

</div>

</body>

</html>

Header.jsp

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

pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Header</title>

<style type="text/css">

.formdiv{

padding: 8px;

border: 1px solid black;

display: inline-block;

margin: auto;

/\*width: 50%;\*/

}

table,th{

border: 1px solid black;

border-collapse: collapse;

padding:24px;

text-align: center;

}

.input{

height: 20px;

width: 300px;

}

</style>

</head>

<body>

<table>

<tbody>

<tr>

<th><a href="/subscriber">Subscribers</a></th>

<th><a href="/trainer">Trainers</a></th>

<th><a href="/subscriptionplan">Subscription Plans</a></th>

<th><a href="/payment">Payments</a></th>

<th><a href="/logout">Log Out</a></th>

<th colspan="5"><u style="text-align:right">Welcome, ${adminName}</u></th>

</tr>

</tbody>

</table>

</body>

</html>

Login.jsp

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

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

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

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

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

<title>${pagetitle}</title>

<style type="text/css">

/\* Reset some default styles for consistency \*/

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

/\* Body Styles \*/

body {

font-family: Arial, sans-serif;

background-color: #f2f2f2;

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;

margin: 0;

padding: 0;

}

/\* Form Container Styles \*/

.form-container {

background-color: #ffffff;

border: 1px solid #000000;

border-radius: 10px;

padding: 20px;

width: 400px;

text-align: center;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

}

/\* Form Heading Styles \*/

.form-heading {

font-size: 24px;

margin-bottom: 20px;

}

/\* Form Input Styles \*/

.input {

height: 40px;

width: 100%;

padding: 10px;

margin: 10px 0;

border: 1px solid #ccc;

border-radius: 5px;

}

/\* Login Button Styles \*/

.login-button {

background-color: #0066cc;

color: #fff;

border: none;

padding: 12px 20px;

cursor: pointer;

border-radius: 5px;

}

.login-button:hover {

background-color: #0052a3;

}

</style>

</head>

<body>

<div class="form-container">

<h3 class="form-heading">Login</h3>

<c:if test="${error==true || success==true}">

<p>${message}</p>

</c:if>

<form action="login" method="POST">

<label for="user\_email">Username</label>

<input type="email" id="user\_email" class="input" name="user\_email">

<label for="user\_password">Password</label>

<input type="password" id="user\_password" class="input" name="user\_password">

<button class="login-button" type="submit">Login</button>

<!-- Include a registration link if needed -->

<!-- <a href="/registration">Registration</a> -->

</form>

</div>

</body>

</html>

Payment-list.jsp

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

pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<html>

<head>

<title>Payments</title>

<style type="text/css">

table,th,td{

border: 1px solid black;

border-collapse: collapse;

padding:8px;

text-align: center;

}

</style>

</head>

<body>

<jsp:include page="header.jsp" ></jsp:include>

<c:if test="${error==true || success==true}">

${message}

</c:if>

<h2>Payments</h2>

<table>

<thead>

<tr>

<td colspan="6"><a href="payment/add-new" style="float:right"><button>Add New Payment</button></a></td>

</tr>

<tr>

<th>ID</th>

<th>Amount (INR)</th>

<th>Date</th>

<th>Mode Of Payment</th>

<th>Paid By</th>

<th>Action</th>

</tr>

</thead>

<tbody>

<c:forEach items="${payments}" var="payment">

<tr>

<td>${payment.paymentId}</td>

<td>${payment.paidAmount}</td>

<td>${payment.paymentDate}</td>

<td>${payment.paymentMode}</td>

<td>${payment.subscriber.subscriberName}</td>

<td>

<a href="payment/${payment.paymentId}"><button>View</button></a>

<a href="payment/edit-payment/${payment.paymentId}"><button>Edit</button></a>

</td>

</tr>

</c:forEach>

</tbody>

<tfoot>

<tr>

<td colspan="6">${message}</td>

</tr>

</tfoot>

</table>

</body>

</html>

Payment-single.jsp

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

pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<html>

<head>

<title>Payment :: ${payment.subscriber.subscriberName}</title>

<style type="text/css">

table,th,td{

border: 1px solid black;

border-collapse: collapse;

padding:8px;

text-align: center;

}

</style>

</head>

<body>

<jsp:include page="header.jsp" ></jsp:include>

<h2>Subscription payment Information</h2>

<c:if test="${error} eq true">

<h3>${error}</h3>

</c:if>

<table>

<thead>

<tr>

<th colspan="2"><h2>Payment By - ${payment.subscriber.subscriberName}</h2></th>

</tr>

</thead>

<tbody>

<tr>

<td>ID</td>

<td>${payment.paymentId}</td>

</tr>

<tr>

<td>Paid Amount</td>

<td>INR. ${payment.paidAmount}</td>

</tr>

<tr>

<td>Payment Date</td>

<td>${payment.paymentDate}</td>

</tr>

<tr>

<td>Payment Mode</td>

<td>${payment.paymentMode}</td>

</tr>

<tr>

<td>Paid By</td>

<td><a href="/subscriber/${payment.subscriber.subscriberId}">${payment.subscriber.subscriberName}</a></td>

</tr>

</tbody>

</table>

</body>

</html>

Plan-list.jsp

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

pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<html>

<head>

<title>Subscription Plans</title>

<style type="text/css">

table,th,td{

border: 1px solid black;

border-collapse: collapse;

padding:8px;

text-align: trainer;

}

</style>

</head>

<body>

<jsp:include page="header.jsp" ></jsp:include>

<c:if test="${error==true || success==true}">

${message}

</c:if>

<h2>Subscription Plans</h2>

<table>

<thead>

<tr>

<td colspan="5"><a href="subscriptionplan/add-new" style="float:right"><button>Add New SubscriptionPlans</button></a></td>

</tr>

<tr>

<th>ID</th>

<th>Title</th>

<th>Duration(Months)</th>

<th>Fees</th>

<th>Action</th>

</tr>

</thead>

<tbody>

<c:forEach items="${subscriptionPlans}" var="subscriptionPlan">

<tr>

<td>${subscriptionPlan.planId}</td>

<td>${subscriptionPlan.planTitle}</td>

<td>${subscriptionPlan.planDuration}</td>

<td>INR. ${subscriptionPlan.planFees}</td>

<td>

<a href="subscriptionplan/${subscriptionPlan.planId}"><button>View</button></a>

<a href="subscriptionplan/edit-plan/${subscriptionPlan.planId}"><button>Edit</button></a>

<a href="subscriptionplan/delete/${subscriptionPlan.planId}"><button>Delete</button></a>

</td>

</tr>

</c:forEach>

</tbody>

<tfoot>

<tr>

<td colspan="5">${message}</td>

</tr>

</tfoot>

</table>

</body>

</html>

Plan-single.jsp

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

pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<html>

<head>

<title>Subscription Plan :: ${plan.planTitle}</title>

<style type="text/css">

table,th,td{

border: 1px solid black;

border-collapse: collapse;

padding:8px;

text-align: center;

}

</style>

</head>

<body>

<jsp:include page="header.jsp" ></jsp:include>

<h2>Subscription Plan Information</h2>

<c:if test="${error} eq true">

<h3>${error}</h3>

</c:if>

<table>

<thead>

<tr>

<th colspan="2"><h2>${plan.planTitle}</h2></th>

</tr>

</thead>

<tbody>

<tr>

<td>ID</td>

<td>${plan.planId}</td>

</tr>

<tr>

<td>Duration(Months)</td>

<td>${plan.planDuration}</td>

</tr>

<tr>

<td>Fees</td>

<td>INR. ${plan.planFees}</td>

</tr>

</tbody>

</table>

</body>

</html>

Register.jsp

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

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

<title>Registration</title>

<style type="text/css">

table,td{

border: 1px solid black;

border-collapse: collapse;

padding:24px;

text-align: center;

}

.formdiv{

padding: 8px;

border: 1px solid black;

display: inline-block;

margin-top: 24px;

}

.input{

height: 20px;

width: 300px;

}

</style>

</head>

<body>

<h2>Admin Registration</h2>

<div class="formdiv">

<form action="register" method="POST">

<table>

<thead>

<tr>

<th colspan="2"><h3>Registration</h3></th>

</tr>

</thead>

<tbody>

<tr>

<td>

<label for="user\_name">Name</label>

</td>

<td>

<input type="text" id="user\_name" class="input" name="user\_name">

</td>

</tr>

<tr>

<td>

<label for="user\_email">Email</label>

</td>

<td>

<input type="text" id="user\_email" class="input" name="user\_email">

</td>

</tr>

<tr>

<td>

<label for="user\_password">Password</label>

</td>

<td>

<input type="password" id="user\_password" class="input" name="user\_password">

</td>

</tr>

<tr>

<td colspan="2"><button type="submit">Register</button></td>

</tr>

</tbody>

</table>

</form>

</div>

</body>

</html>

Subscriber-list.jsp

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

pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<html>

<head>

<title>Subscribers</title>

<style type="text/css">

table,th,td{

border: 1px solid black;

border-collapse: collapse;

padding:8px;

text-align: center;

}

</style>

</head>

<body>

<jsp:include page="header.jsp"></jsp:include>

<h2>Subscribers</h2>

<c:if test="${error==true || success==true}">

${message}

</c:if>

<table>

<thead>

<tr>

<td colspan="8"><a href="subscriber/add-new" style="float:right"><button>Add New Subscriber</button></a></td>

</tr>

<tr>

<th>ID</th>

<th>Name</th>

<th>Age</th>

<th>Gender</th>

<th>Address</th>

<th>Status</th>

<th>Fees Paid</th>

<th>Action</th>

</tr>

</thead>

<tbody>

<c:forEach items="${subscriber\_list}" var="subscriber">

<tr>

<td>${subscriber.subscriberId}</td>

<td>${subscriber.subscriberName}</td>

<td>${subscriber.subscriberAge}</td>

<td>${subscriber.subscriberGender}</td>

<td>${subscriber.subscriberAddress}</td>

<td>

<c:if test="${subscriber.subscriberStatus == false}">

InActive

</c:if>

<c:if test="${subscriber.subscriberStatus == true}">

Active

</c:if>

</td>

<td>${subscriber.subscriberFeesPaid}</td>

<td>

<a href="subscriber/${subscriber.subscriberId}"><button>View</button></a>

<a href="subscriber/edit-subscriber/${subscriber.subscriberId}"><button>Edit</button></a>

<a href="subscriber/delete/${subscriber.subscriberId}"><button>Delete</button></a>

</td>

</tr>

</c:forEach>

</tbody>

<tfoot>

<tr>

<td colspan="8">${message}</td>

</tr>

</tfoot>

</table>

</body>

</html>

Subscriber-single.jsp

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

pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<html>

<head>

<title>Subscriber :: ${subscriber.subscriberName}</title>

<style type="text/css">

table,th,td{

border: 1px solid black;

border-collapse: collapse;

padding:8px;

text-align: center;

}

</style>

</head>

<body>

<jsp:include page="header.jsp"></jsp:include>

<h2>Subscriber Information</h2>

<c:if test="${error==true || success==true}">

${message}

</c:if>

<table>

<thead>

<tr>

<th colspan="2"><h3>${subscriber.subscriberName}</h3></th>

</tr>

</thead>

<tbody>

<tr>

<td>ID</td>

<td>${subscriber.subscriberId}</td>

</tr>

<tr>

<td>Gender</td>

<td>${subscriber.subscriberGender}</td>

</tr>

<tr>

<td>Age(Years)</td>

<td>${subscriber.subscriberAge}</td>

</tr>

<tr>

<td>Address</td>

<td>${subscriber.subscriberAddress}</td>

</tr>

<tr>

<td>Paid Fees</td>

<td>${subscriber.subscriberFeesPaid}</td>

</tr>

<tr>

<td>Allocated Trainer</td>

<td><a href="/trainer/${subscriber.trainer.trainerId}">${subscriber.trainer.trainerName}</a></td>

</tr>

<tr>

<td>Subscription Plan</td>

<td><a href="/subscriptionplan/${subscriber.subscriptionPlan.planId}">${subscriber.subscriptionPlan.planTitle}</a></td>

</tr>

<tr>

<td>Subscriber Status</td>

<td>

<c:if test="${subscriber.subscriberStatus == false}">

Inactive

</c:if>

<c:if test="${subscriber.subscriberStatus == true}">

Active

</c:if>

</td>

</tr>

</tbody>

</table>

</body>

</html>

Trainer-list.jsp

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

pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<html>

<head>

<title>Trainers</title>

<style type="text/css">

table,th,td{

border: 1px solid black;

border-collapse: collapse;

padding:8px;

text-align: trainer;

}

</style>

</head>

<body>

<jsp:include page="header.jsp" ></jsp:include>

<c:if test="${error==true || success==true}">

${message}

</c:if>

<h2>Trainers</h2>

<table>

<thead>

<tr>

<td colspan="7"><a href="trainer/add-new" style="float:right"><button>Add New Trainer</button></a></td>

</tr>

<tr>

<th>ID</th>

<th>Name</th>

<th>Age</th>

<th>Gender</th>

<th>Experience (Years)</th>

<th>Address</th>

<th>Action</th>

</tr>

</thead>

<tbody>

<c:forEach items="${trainers}" var="trainer">

<tr>

<td>${trainer.trainerId}</td>

<td>${trainer.trainerName}</td>

<td>${trainer.trainerAge}</td>

<td>${trainer.trainerGender}</td>

<td>${trainer.trainerExperience}</td>

<td>${trainer.trainerAddress}</td>

<td>

<a href="trainer/${trainer.trainerId}"><button>View</button></a>

<a href="trainer/edit-trainer/${trainer.trainerId}"><button>Edit</button></a>

<a href="trainer/delete/${trainer.trainerId}"><button>Delete</button></a>

</td>

</tr>

</c:forEach>

</tbody>

<tfoot>

<tr>

<td colspan="7">${message}</td>

</tr>

</tfoot>

</table>

</body>

</html>

Trainer-single.jsp

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

pageEncoding="UTF-8"%>

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

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

<%@ page isELIgnored="false" %>

<html>

<head>

<title>Trainer :: ${trainer.trainerName}</title>

<style type="text/css">

table,th,td{

border: 1px solid black;

border-collapse: collapse;

padding:8px;

text-align: center;

}

</style>

</head>

<body>

<jsp:include page="header.jsp" ></jsp:include>

<h2>Trainer Information</h2>

<c:if test="${error} eq true">

<h3>${error}</h3>

</c:if>

<table>

<thead>

<tr>

<th colspan="2"><h2>${trainer.trainerName}</h2></th>

</tr>

</thead>

<tbody>

<tr>

<td>ID</td>

<td>${trainer.trainerId}</td>

</tr>

<tr>

<td>Age</td>

<td>${trainer.trainerAge} Years</td>

</tr>

<tr>

<td>Gender</td>

<td>${trainer.trainerGender}</td>

</tr>

<tr>

<td>Experience</td>

<td>${trainer.trainerExperience} Years</td>

</tr>

<tr>

<td>Address</td>

<td>${trainer.trainerAddress}</td>

</tr>

</tbody>

</table>

</body>

</html>

pom.xml

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

<subham xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.7.5</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.subham</groupId>

<artifactId>fitnessclubautomation</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>fitnessclubautomation</name>

<description>Fitness club automation subham for automating the tasks of fitness club using Spring Boot

</description>

<properties>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-actuator</artifactId>

</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.tomcat.embed/tomcat-embed-jasper -->

<dependency>

<groupId>org.apache.tomcat.embed</groupId>

<artifactId>tomcat-embed-jasper</artifactId>

</dependency>

<!-- https://mvnrepository.com/artifact/javax.servlet.jsp.jstl/jstl -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>servlet-api</artifactId>

<version>2.5</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>com.mysql</groupId>

<artifactId>mysql-connector-j</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</subham>