CoachController.java Code

```java
package edu.deakin.sit218.coachwebapp.controller;

import java.util.logging.Level;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;
import javax.validation.Valid;
import org.springframework.mock.web.MockHttpSession;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;

import edu.deakin.sit218.coachwebapp.dao.ClientDAO;
import edu.deakin.sit218.coachwebapp.dao.ClientDAOImpl;
import edu.deakin.sit218.coachwebapp.entity.Client;

@Controller
public class CoachController {

    private String referer1;
    private String referer2;

    @RequestMapping("/workout")
    public String workout(Model model, HttpServletRequest request) {
        HttpSession session = request.getSession();
        String uname = (String) session.getAttribute("user");
        this.referer1 = request.getHeader("referer");

        ClientDAO dao = new ClientDAOImpl();
        Client client = dao.retrieveClient(uname);
        // Retrieve Client object from database
        giveWorkoutToClient(client, model);

        // Return the View
        return "workout";
    }

    private void giveWorkoutToClient(Client client, Model model) {
        if (client.getUsername().equals("Rolando")) {
            model.addAttribute("message", "Rolando never works out");
        } else if (client.getAge() < 40) {
            model.addAttribute("message", "Hey, " + client.getUsername() +
                    " you are still too young, no need to work out!");
        } else {
            model.addAttribute("message", client.getUsername() +
                    ", please, run for 30 min." + System.lineSeparator() +
                    " You have worked out " + client.getWorkouts() + " times.");
            client.addWorkout();
            ClientDAO dao = new ClientDAOImpl();
            dao.updateClient(client);
        }
    }

    @RequestMapping("/change")
    public String changeAge(@RequestParam("age") int age, Model model,
HttpServletRequest request) {
        // Logic to prevent CSRF attacks
        this.referer2 = request.getHeader("referer");
        if (!this.referer1.equals(this.referer2)) {
            HttpSession session = request.getSession(false);
            if (session != null) {
                session.invalidate(); // Invalidate session if CSRF is detected
            }
            model.addAttribute("message", "CSRF attack detected. Session
invalidated.");
            return "error"; // Redirect to custom error page
        }

        ClientDAO dao = new ClientDAOImpl();
        MockHttpSession session = (MockHttpSession) request.getSession();
        String uname = (String) session.getAttribute("user");
        Client client = dao.retrieveClient(uname);
        client.setAge(age);
        dao.updateClient(client);
        model.addAttribute("message", "Your age has been updated");
        return "workout";
    }

    @ExceptionHandler(Exception.class)
    public String handleException(Exception ex, Model model) {

java.util.logging.Logger.getLogger(CoachController.class.getName()).log(Level.SEV
ERE, null, ex);
        model.addAttribute("errorMessage", "A server error occurred. Please try
again.");
```

```
        return "error"; // Custom error page
    }
}
```