Checks

1. Check database table
2. Check server.xml
3. App.properties
4. Web.xml for security constraints
5. @servletSecurity annotation

Demo

1. Create project
2. Add dependency
3. Create structure like controller, dao, service, exception, model
4. Create ui using basic jsp pages
5. Demonstrate multiple request with single servlet
6. Add security

AdminController.java

```
package com.cybage.controller;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.HttpConstraint;
import javax.servlet.annotation.ServletSecurity;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.cybage.dao.UserDao;
import com.cybage.model.Users;
import com.cybage.service.UserService;



@ServletSecurity(
		value = @HttpConstraint(
				rolesAllowed = {"admin"}
				)
		)
public class AdminController extends HttpServlet {
	UserService us = new UserService();
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
		doGet(request, response);
	}
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
		PrintWriter out = response.getWriter();
		out.print("admin servlet");
		String path =  request.getPathInfo();


		if(path.equals("/add")) {
```

```
                try {
                        Users user = new Users();
                        user.setUsername(request.getParameter("username"));
                        user.setPassword(request.getParameter("password"));
                        user.setRole(request.getParameter("userrole"));
                        us.addUser(user);
                        response.sendRedirect("list");
                } catch (Exception e) {
                        e.printStackTrace();
                }
        }

        if(path.equals("/edit")) {
                try {
                        System.out.println("path: "+ path);
                        Users user = us.getUser(request.getParameter("un"));
                        System.out.println("user details: "+ user);
                        request.setAttribute("user", user);
                        request.getRequestDispatcher("/admin/admin-update-user.jsp").include(request,
response);
                } catch (Exception e) {
                        e.printStackTrace();
                }
        }

        if(path.equals("/update")) {
                try {
                        Users user = new Users();
                        user.setUsername(request.getParameter("username"));
                        user.setPassword(request.getParameter("password"));
                        user.setRole(request.getParameter("userrole"));
                        us.updateUser(user);
                        response.sendRedirect("list");
                } catch (Exception e) {
                        e.printStackTrace();
                }
        }

        if(path.equals("/list")) {
                try {
                        request.setAttribute("users", us.getUsers());
                        request.getRequestDispatcher("/admin/admin-user-management.jsp").forward(request,
response);
                } catch (Exception e) {
                        e.printStackTrace();
                }
        }

        if(path.equals("/delete")) {
                try {
                        us.deleteUser(request.getParameter("un"));
                        response.sendRedirect("list");
                } catch (Exception e) {
                        e.printStackTrace();
```

```
                }
        }
    }
}
```

AppController.java

```
package com.cybage.controller;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.HttpConstraint;
import javax.servlet.annotation.ServletSecurity;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


@ServletSecurity(
        value = @HttpConstraint(
                rolesAllowed = {"admin" ,"member", "manager"}
                )
        )
public class AppController extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {

        if(request.isUserInRole("admin")) {
            request.getRequestDispatcher("/admin/admin-home.jsp").forward(request, response);
        }
        if(request.isUserInRole("member")) {
            request.getRequestDispatcher("/member/member-home.jsp").forward(request, response);
        }
        if(request.isUserInRole("manager")) {
            request.getRequestDispatcher("/manager/manager-home.jsp").forward(request, response);
        }
    }
}
```

ManagerController.java

```
package com.cybage.controller;

import java.io.IOException;
import java.io.PrintWriter;
import java.time.LocalDate;
import java.time.LocalTime;

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

```
import com.cybage.model.Batch;
import com.cybage.model.Plan;
import com.cybage.model.Users;
import com.cybage.service.ManagerService;
import com.cybage.service.MemberService;

@ServletSecurity(
		value = @HttpConstraint(
				rolesAllowed = {"manager"}
				)
		)
public class ManagerController extends HttpServlet {

	ManagerService ms = new ManagerService();
	MemberService memberService = new MemberService();

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

	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
		PrintWriter out = response.getWriter();
		out.print("manager servlet");
		String path =  request.getPathInfo();


		if(path.equals("/addplan")) {
			try {
				Plan plan = new Plan();
				plan.setPlanname(request.getParameter("planname"));
				plan.setSportid(request.getParameter("sportsid"));
				plan.setFees(Double.parseDouble(request.getParameter("fees")));
				plan.setDuration(Integer.parseInt(request.getParameter("duration")));

				ms.addPlan(plan);
				response.sendRedirect("planlist");
			} catch (Exception e) {
				e.printStackTrace();
			}
		}

		if(path.equals("/planlist")) {
			try {
				request.setAttribute("plans", ms.getPlans());
				request.getRequestDispatcher("/manager/plans.jsp").forward(request, response);
			} catch (Exception e) {
				e.printStackTrace();
			}
		}

		if(path.equals("/deleteplan")) {
			try {
```

```
                ms.deletePlan(request.getParameter("planid"));
                response.sendRedirect("planlist");
        } catch (Exception e) {
                e.printStackTrace();
        }
}

if(path.equals("/editplan")) {
        try {
                System.out.println("path: "+ path);
                Plan plan = ms.getPlan(request.getParameter("planid"));
                request.setAttribute("plan", plan);
                request.getRequestDispatcher("/manager/edit-plan.jsp").include(request, response);
        } catch (Exception e) {
                e.printStackTrace();
        }
}

if(path.equals("/updateplan")) {
        try {
                Plan plan = new Plan();
                plan.setPlanid(request.getParameter("planid"));
                plan.setPlanname(request.getParameter("planname"));
                plan.setSportid(request.getParameter("sportsid"));
                plan.setFees(Double.parseDouble(request.getParameter("fees")));
                plan.setDuration(Integer.parseInt(request.getParameter("duration")));
                System.out.println("inside controller: "+ plan);
                ms.updatePlan(plan);
                response.sendRedirect("planlist");
        } catch (Exception e) {
                e.printStackTrace();
        }
}

//batches
if(path.equals("/batchlist")) {
        try {
                request.setAttribute("batches", ms.getBatches());
                request.getRequestDispatcher("/manager/batches.jsp").forward(request, response);
        } catch (Exception e) {
                e.printStackTrace();
        }
}
if(path.equals("/addbatch")) {
        try {
                Batch batch = new Batch();
                batch.setBatchname(request.getParameter("batchname"));
                batch.setSportsid(request.getParameter("sportsid"));
                batch.setBatchStartDate(LocalDate.parse(request.getParameter("batchstartdate")));
                batch.setBatchTime(LocalTime.parse(request.getParameter("batchtime")));
                batch.setBatchDuration(Double.parseDouble(request.getParameter("batchduration")));
                batch.setBatchSize(Integer.parseInt(request.getParameter("batchsize")));
                ms.addBatch(batch);
                response.sendRedirect("batchlist");
```

```
                } catch (Exception e) {
                        e.printStackTrace();
                }
        }

        if(path.equals("/deletebatch")) {
                try {
                        ms.deleteBatch(request.getParameter("batchid"));
                        response.sendRedirect("batchlist");
                } catch (Exception e) {
                        e.printStackTrace();
                }
        }

        if(path.equals("/editbatch")) {
                try {
                        Batch batch = ms.getBatch(request.getParameter("batchid"));
                        request.setAttribute("batch", batch);
                        request.getRequestDispatcher("/manager/edit-batch.jsp").include(request, response);
                } catch (Exception e) {
                        e.printStackTrace();
                }
        }

        if(path.equals("/updatebatch")) {
                try {
                        Batch batch = new Batch();
                        batch.setBatchid(request.getParameter("batchid"));
                        batch.setBatchname(request.getParameter("batchname"));
                        batch.setSportsid(request.getParameter("sportsid"));
                        batch.setBatchStartDate(LocalDate.parse(request.getParameter("batchstartdate")));
                        batch.setBatchTime(LocalTime.parse(request.getParameter("batchtime")));
                        batch.setBatchDuration(Double.parseDouble(request.getParameter("batchduration")));
                        batch.setBatchSize(Integer.parseInt(request.getParameter("batchsize")));

                        ms.updateBatch(batch);
                        response.sendRedirect("batchlist");
                } catch (Exception e) {
                        e.printStackTrace();
                }
        }

        if(path.equals("/enrollments")) {
                try {
                        request.setAttribute("enrollments", memberService.getEnrollments());
                        request.getRequestDispatcher("/manager/enrollments.jsp").forward(request, response);

                } catch (Exception e) {
                        e.printStackTrace();
                }
        }
    }
}
```

MemberController.java

```
package com.cybage.controller;

import java.io.IOException;
import java.io.PrintWriter;
import java.time.LocalDate;

import javax.servlet.ServletException;
import javax.servlet.annotation.HttpConstraint;
import javax.servlet.annotation.ServletSecurity;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.cybage.model.Enrollment;
import com.cybage.model.Users;
import com.cybage.service.MemberService;

public class MemberController extends HttpServlet {

	MemberService ms = new MemberService();

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


	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
		PrintWriter out = response.getWriter();
		String path =  request.getPathInfo();
		if(path.equals("/planlist")) {
			try {
				request.setAttribute("plans", ms.getPlans());
				request.getRequestDispatcher("/member/plans.jsp").forward(request, response);
			} catch (Exception e) {
				e.printStackTrace();
			}
		}

		if(path.equals("/enroll")) {
			try {
				request.setAttribute("planid", request.getParameter("planid"));
				request.getRequestDispatcher("/member/enroll.jsp").forward(request, response);
			} catch (Exception e) {
				e.printStackTrace();
			}
		}
		if(path.equals("/makeenrollment")) {
			try {
				Enrollment enroll = new Enrollment();
				enroll.setUsername("user");
```

```
                                enroll.setBatchid(request.getParameter("batchid"));
                                enroll.setPlanid(request.getParameter("planid"));
                                enroll.setStartdate(LocalDate.parse(request.getParameter("startdate")));
                                ms.enroll(enroll);
                                request.getRequestDispatcher("/member/enroll.jsp").forward(request, response);
                        } catch (Exception e) {
                                e.printStackTrace();
                        }
                }
                if(path.equals("/enrollments")) {
                        try {
                                request.setAttribute("enrollments", ms.getEnrollments());
                                request.getRequestDispatcher("/member/enrollments.jsp").forward(request, response);
                        } catch (Exception e) {
                                e.printStackTrace();
                        }
                }

                if(path.equals("/register")) {
                        try {
                                System.out.println("inside registration");
                                Users user = new Users(request.getParameter("username"),
request.getParameter("password"), "member");
                                ms.register(user);
                                out.print("Registration successful...");
                                request.getRequestDispatcher("/index.jsp").include(request, response);

                        } catch (Exception e) {
                                e.printStackTrace();
                        }
                }
        }
}
```

Dbutil.java

```
package com.cybage.dao;

import java.io.FileReader;
import java.io.InputStream;
import java.net.URL;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.Statement;
import java.util.Properties;

import org.apache.commons.dbcp2.BasicDataSource;


public class DbUtil {

        public static String dbURL;
        public static String dbUser ;
        public static String dbPassword ;
        static {
```

```
		try {
			ClassLoader classloader = Thread.currentThread().getContextClassLoader();
			InputStream is = classloader.getResourceAsStream("app.properties");

			Properties props = new Properties();
			props.load(is);
			dbURL = props.getProperty("dbURL");
			dbUser = props.getProperty("dbUser");
			dbPassword = props.getProperty("dbPassword");

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

	//with connection pool
	public static Connection getConnection() throws Exception{
		Connection con = getDataSource().getConnection();
		return con;
	}

	private static BasicDataSource getDataSource()
	{
		BasicDataSource ds = new BasicDataSource();
		ds.setUrl(dbURL);
		ds.setUsername(dbUser);
		ds.setPassword(dbPassword);
		ds.setMinIdle(5);
		ds.setMaxIdle(10);
		ds.setMaxOpenPreparedStatements(100);
		return ds;
	}


	public static Statement getStatement() throws Exception {
		return getConnection().createStatement();
	}
	public static PreparedStatement getPreparedStatement(String sql) throws Exception{
		return getConnection().prepareStatement(sql);
	}
}
```

managerdao.java

```
package com.cybage.dao;

import java.sql.Connection;
import java.sql.Date;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Time;
import java.util.ArrayList;
import java.util.List;

import com.cybage.model.Batch;
```

```
import com.cybage.model.Plan;
import com.cybage.model.Sports;
import com.cybage.model.Users;

public class ManagerDao {

	public List<Plan> getPlans() throws Exception{

		String sql = "select * from plans";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ResultSet rs = ps.executeQuery();

		List<Plan> plans = new ArrayList<>();
		while(rs.next()) {

			plans.add(new Plan(rs.getString(1), rs.getString(2), rs.getString(3), rs.getDouble(4), rs.getInt(5)));

		}
		return plans;
	}

	public List<Sports> getSports() throws Exception{

		String sql = "select * from sports";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ResultSet rs = ps.executeQuery();

		List<Sports> sports = new ArrayList<>();
		while(rs.next()) {
			sports.add(new Sports(rs.getString(1), rs.getString(2)));
		}
		System.out.println(sports);
		return sports;
	}

	public int addPlan(Plan plan) throws Exception{
		String sql = "insert into plans values (?, ?, ?, ?, ?)";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ps.setString(1, plan.getPlanid());
		ps.setString(2, plan.getPlanname());
		ps.setString(3, plan.getSportid());
		ps.setDouble(4, plan.getFees());
		ps.setInt(5, plan.getDuration());
		return ps.executeUpdate();
	}

	public int deletePlan(String planid) throws Exception{
		String sql = "delete from plans where planid=?";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ps.setString(1, planid);
```

```
		return ps.executeUpdate();
	}

	public Plan findPlan(String planid) throws Exception{
		String sql = "select * from plans where planid = ? ";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ps.setString(1, planid);
		ResultSet rs = ps.executeQuery();

		Plan plan = new Plan();
		if(rs.next()) {
			plan = new Plan(rs.getString(1), rs.getString(2), rs.getString(3), rs.getDouble(4), rs.getInt(5));
		}
		System.out.println("inside dao: "+ plan);
		return plan;
	}

	public int updatePlan(Plan plan) throws Exception{
		String sql = "update plans set planname = ?, sportsid = ?, fees = ? , duration = ? where planid = ? ";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ps.setString(1, plan.getPlanname());
		ps.setString(2, plan.getSportid());
		ps.setDouble(3, plan.getFees());
		ps.setInt(4, plan.getDuration());
		ps.setString(5, plan.getPlanid());
		return ps.executeUpdate();
	}

	public List<Batch> getBatches() throws Exception{
		String sql = "select * from batches";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ResultSet rs = ps.executeQuery();

		List<Batch> batches = new ArrayList<>();
		while(rs.next()) {
			batches.add(
			new Batch(rs.getString(1), rs.getString(2), rs.getString(3),
					rs.getDate(4).toLocalDate(), rs.getTime(5).toLocalTime(), rs.getDouble(6),
rs.getInt(7)
					)
			);
		}
		return batches;

	}

	public int addBatch(Batch batch) throws Exception{
		String sql = "insert into batches values (?, ?, ?, ?, ?, ? , ?)";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ps.setString(1, batch.getBatchid());
```

```
		ps.setString(2, batch.getBatchname());
		ps.setString(3, batch.getSportsid());
		ps.setDate(4, Date.valueOf(batch.getBatchStartDate()));
		ps.setTime(5, Time.valueOf(batch.getBatchTime()));
		ps.setDouble(6, batch.getBatchDuration());
		ps.setInt(7, batch.getBatchSize());
		return ps.executeUpdate();
	}

	public int deleteBatch(String batchid) throws Exception{
		String sql = "delete from batches where batchid=?";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ps.setString(1, batchid);
		return ps.executeUpdate();
	}


	public Batch findBatch(String batchId) throws Exception{
		String sql = "select * from batches where batchid = ? ";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ps.setString(1, batchId);
		ResultSet rs = ps.executeQuery();

		Batch batch = new Batch();
		if(rs.next()) {
			batch = new Batch(rs.getString(1), rs.getString(2), rs.getString(3),
					rs.getDate(4).toLocalDate(), rs.getTime(5).toLocalTime(), rs.getDouble(6),
rs.getInt(7));
		}
		return batch;
	}

	public int updateBatch(Batch batch) throws Exception{
		String sql = "update batches set batchname = ?, sportsid = ?, batchstartdate = ? , "
				+ "batchtime = ?, batchduration = ?, batchsize = ?  where batchid = ? ";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ps.setString(1, batch.getBatchname());
		ps.setString(2, batch.getSportsid());
		ps.setDate(3, Date.valueOf(batch.getBatchStartDate()));
		ps.setTime(4, Time.valueOf(batch.getBatchTime()));
		ps.setDouble(5, batch.getBatchDuration());
		ps.setInt(6, batch.getBatchSize());
		ps.setString(7, batch.getBatchid());
		return ps.executeUpdate();
	}

}
```

memberdao.java

```
package com.cybage.dao;
```

```
import java.sql.Connection;
import java.sql.Date;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Time;
import java.util.ArrayList;
import java.util.List;

import com.cybage.model.Batch;
import com.cybage.model.Enrollment;
import com.cybage.model.Plan;
import com.cybage.model.Sports;
import com.cybage.model.Users;

public class MemberDao {

	public List<Plan> getPlans() throws Exception{

		String sql = "select * from plans";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ResultSet rs = ps.executeQuery();

		List<Plan> plans = new ArrayList<>();
		while(rs.next()) {

			plans.add(new Plan(rs.getString(1), rs.getString(2), rs.getString(3), rs.getDouble(4), rs.getInt(5)));

		}
		return plans;
	}

	public List<Batch> getBatches() throws Exception{
		String sql = "select * from batches";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ResultSet rs = ps.executeQuery();

		List<Batch> batches = new ArrayList<>();
		while(rs.next()) {
			batches.add(
					new Batch(rs.getString(1), rs.getString(2), rs.getString(3),
							rs.getDate(4).toLocalDate(), rs.getTime(5).toLocalTime(),
rs.getDouble(6), rs.getInt(7)
							)
					);
		}
		return batches;

	}

	public Batch findBatch(String batchId) throws Exception{
		String sql = "select * from batches where batchid = ? ";
		Connection con = DbUtil.getConnection();
```

```
		PreparedStatement ps = con.prepareStatement(sql);
		ps.setString(1, batchId);
		ResultSet rs = ps.executeQuery();

		Batch batch = new Batch();
		if(rs.next()) {
			batch = new Batch(rs.getString(1), rs.getString(2), rs.getString(3),
					rs.getDate(4).toLocalDate(), rs.getTime(5).toLocalTime(), rs.getDouble(6),
rs.getInt(7));
		}
		return batch;
	}
	//
	public int updateBatch(Batch batch) throws Exception{
		String sql = "update batches set batchname = ?, sportsid = ?, batchstartdate = ? , "
				+ "batchtime = ?, batchduration = ?, batchsize = ?  where batchid = ? ";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ps.setString(1, batch.getBatchname());
		ps.setString(2, batch.getSportsid());
		ps.setDate(3, Date.valueOf(batch.getBatchStartDate()));
		ps.setTime(4, Time.valueOf(batch.getBatchTime()));
		ps.setDouble(5, batch.getBatchDuration());
		ps.setInt(6, batch.getBatchSize());
		ps.setString(7, batch.getBatchid());
		return ps.executeUpdate();
	}

	public int enroll(Enrollment enroll) throws Exception {
		String sql = "insert into enrollment values (?, ?, ?, ?, ?, ?)";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ps.setString(1, enroll.getEnrollid());
		ps.setString(2, enroll.getUsername());
		ps.setString(3, enroll.getPlanid());
		ps.setString(4, enroll.getBatchid());
		ps.setDate(5, Date.valueOf(enroll.getStartdate()));
		ps.setString(6, enroll.getStatus());
		return ps.executeUpdate();
	}
	public List<Enrollment> getEnrollments() throws Exception{
		String sql = "select * from enrollment";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ResultSet rs = ps.executeQuery();

		List<Enrollment> enrollments = new ArrayList<>();
		while(rs.next()) {
			enrollments.add(
					new Enrollment(rs.getString(1), rs.getString(2), rs.getString(3),rs.getString(4),
							rs.getDate(5).toLocalDate(),  rs.getString(6)
							)
					);
		}
```

```
		return enrollments;

	}

	public int register(Users user) throws Exception{
		String sql = "insert into users values (?, ?, ?)";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ps.setString(1, user.getUsername());
		ps.setString(2, user.getPassword());
		ps.setString(3, user.getRole());
		return ps.executeUpdate();
	}

}
```

userdao.java

```
package com.cybage.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

import com.cybage.model.Users;

public class UserDao {
	public List<Users> findAllUsers() throws Exception{
		String sql = "select * from users";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ResultSet rs = ps.executeQuery();

		List<Users> users = new ArrayList();
		while(rs.next()) {
			users.add(new Users(rs.getString(1), rs.getString(2), rs.getString(3)));
		}
		return users;
	}

	public Users findUser(String username) throws Exception{
		String sql = "select * from users where username = ? ";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ps.setString(1, username);
		ResultSet rs = ps.executeQuery();

		Users user = new Users();
		if(rs.next()) {
			user = new Users(rs.getString(1), rs.getString(2), rs.getString(3));
		}
```

```
		System.out.println("inside dao: "+ user);
		return user;
	}
	public int deleteUser(String username) throws Exception {
		String sql = "delete from users where username=?";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ps.setString(1, username);
		return ps.executeUpdate();
	}
	public int addUser(Users user) throws Exception{
		String sql = "insert into users values (?, ?, ?)";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ps.setString(1, user.getUsername());
		ps.setString(2, user.getPassword());
		ps.setString(3, user.getRole());
		return ps.executeUpdate();
	}

	public int updateUser(Users user) throws Exception{
		String sql = "update users set password = ?, userrole = ? where username = ? ";
		Connection con = DbUtil.getConnection();
		PreparedStatement ps = con.prepareStatement(sql);
		ps.setString(1, user.getPassword());
		ps.setString(2, user.getRole());
		ps.setString(3, user.getUsername());
		return ps.executeUpdate();
	}
}
```

batch.java

```
package com.cybage.model;

import java.time.LocalDate;
import java.time.LocalTime;

public class Batch {
	private String batchid;
	private String batchname;
	private String sportsid;
	private LocalDate batchStartDate;
	private LocalTime batchTime;
	private double batchDuration;
	private int batchSize;
	public Batch() {
		super();
		// TODO Auto-generated constructor stub
	}
	public Batch(String batchid, String batchname, String sportsid, LocalDate batchStartDate, LocalTime batchTime,
			double batchDuration, int batchSize) {
		super();
		this.batchid = batchid;
		this.batchname = batchname;
```

```
		this.sportsid = sportsid;
		this.batchStartDate = batchStartDate;
		this.batchTime = batchTime;
		this.batchDuration = batchDuration;
		this.batchSize = batchSize;
	}
	public String getBatchid() {
		return batchid;
	}
	public void setBatchid(String batchid) {
		this.batchid = batchid;
	}
	public String getBatchname() {
		return batchname;
	}
	public void setBatchname(String batchname) {
		this.batchname = batchname;
	}
	public String getSportsid() {
		return sportsid;
	}
	public void setSportsid(String sportsid) {
		this.sportsid = sportsid;
	}
	public LocalDate getBatchStartDate() {
		return batchStartDate;
	}
	public void setBatchStartDate(LocalDate batchStartDate) {
		this.batchStartDate = batchStartDate;
	}
	public LocalTime getBatchTime() {
		return batchTime;
	}
	public void setBatchTime(LocalTime batchTime) {
		this.batchTime = batchTime;
	}
	public double getBatchDuration() {
		return batchDuration;
	}
	public void setBatchDuration(double batchDuration) {
		this.batchDuration = batchDuration;
	}
	public int getBatchSize() {
		return batchSize;
	}
	public void setBatchSize(int batchSize) {
		this.batchSize = batchSize;
	}
	@Override
	public String toString() {
		return "Batch [batchid=" + batchid + ", batchname=" + batchname + ", sportsid=" + sportsid + ",
batchStartDate="
				+ batchStartDate + ", batchTime=" + batchTime + ", batchDuration=" + batchDuration + ",
batchSize="
```

```
                                        + batchSize + "]";
        }
}
```

enrollment.java

```
package com.cybage.model;

import java.time.LocalDate;

public class Enrollment {
        private String enrollid;
        private String username;
        private String planid;
        private String batchid;
        private LocalDate startdate;
        private String status;
        public Enrollment() {
                super();
                // TODO Auto-generated constructor stub
        }
        public Enrollment(String enrollid, String username, String planid, String batchid, LocalDate startdate,
                        String status) {
                super();
                this.enrollid = enrollid;
                this.username = username;
                this.planid = planid;
                this.batchid = batchid;
                this.startdate = startdate;
                this.status = status;
        }
        public String getEnrollid() {
                return enrollid;
        }
        public void setEnrollid(String enrollid) {
                this.enrollid = enrollid;
        }
        public String getUsername() {
                return username;
        }
        public void setUsername(String username) {
                this.username = username;
        }
        public String getPlanid() {
                return planid;
        }
        public void setPlanid(String planid) {
                this.planid = planid;
        }
        public String getBatchid() {
                return batchid;
        }
        public void setBatchid(String batchid) {
```

```
		this.batchid = batchid;
	}
	public LocalDate getStartdate() {
		return startdate;
	}
	public void setStartdate(LocalDate startdate) {
		this.startdate = startdate;
	}
	public String getStatus() {
		return status;
	}
	public void setStatus(String status) {
		this.status = status;
	}
	@Override
	public String toString() {
		return "Enrollment [enrollid=" + enrollid + ", username=" + username + ", planid=" + planid + ", batchid="
				+ batchid + ", startdate=" + startdate + ", status=" + status + "]";
	}
}
```

plan.java

```
package com.cybage.model;

public class Plan {
	private String planid;
	private String planname;
	private String sportid;
	private double fees;
	private int duration;
	public Plan() {
		super();
		// TODO Auto-generated constructor stub
	}
	public Plan(String planid, String planname, String sportid, double fees, int duration) {
		super();
		this.planid = planid;
		this.planname = planname;
		this.sportid = sportid;
		this.fees = fees;
		this.duration = duration;
	}
	public String getPlanid() {
		return planid;
	}
	public void setPlanid(String planid) {
		this.planid = planid;
	}
	public String getPlanname() {
		return planname;
	}
	public void setPlanname(String planname) {
		this.planname = planname;
```

```
	}
	public String getSportid() {
		return sportid;
	}
	public void setSportid(String sportid) {
		this.sportid = sportid;
	}
	public double getFees() {
		return fees;
	}
	public void setFees(double fees) {
		this.fees = fees;
	}
	public int getDuration() {
		return duration;
	}
	public void setDuration(int duration) {
		this.duration = duration;
	}
	@Override
	public String toString() {
		return "Plan [planid=" + planid + ", planname=" + planname + ", sportid=" + sportid + ", fees=" + fees
				+ ", duration=" + duration + "]";
	}


}
```

sports.java

```
package com.cybage.model;

public class Sports {
	private String sportsId;
	private String sportsName;


	public Sports() {
		super();
		// TODO Auto-generated constructor stub
	}


	public Sports(String sportsId, String sportsName) {
		super();
		this.sportsId = sportsId;
		this.sportsName = sportsName;
	}


	public String getSportsId() {
		return sportsId;
	}
```

```
	public void setSportsId(String sportsId) {
		this.sportsId = sportsId;
	}


	public String getSportsName() {
		return sportsName;
	}


	public void setSportsName(String sportsName) {
		this.sportsName = sportsName;
	}


	@Override
	public String toString() {
		return "Sports [sportsId=" + sportsId + ", sportsName=" + sportsName + "]";
	}


}
```

users.java

```
package com.cybage.model;

public class Users {
	private String username;
	private String password;
	private String role;
	public Users() {
		super();
		// TODO Auto-generated constructor stub
	}
	public Users(String username, String password, String role) {
		super();
		this.username = username;
		this.password = password;
		this.role = role;
	}
	public String getUsername() {
		return username;
	}
	public void setUsername(String username) {
		this.username = username;
	}
	public String getPassword() {
		return password;
	}
	public void setPassword(String password) {
		this.password = password;
	}
```

```
	public String getRole() {
		return role;
	}
	public void setRole(String role) {
		this.role = role;
	}
	@Override
	public String toString() {
		return "Users [username=" + username + ", role=" + role + "]";
	}

}
```

managerservice.java

```
package com.cybage.service;

import java.util.List;

import com.cybage.dao.ManagerDao;
import com.cybage.model.Batch;
import com.cybage.model.Plan;
import com.cybage.model.Users;

public class ManagerService {
	ManagerDao md = new ManagerDao();

	private String getPlanId() {
		return "plan"+ Math.round(Math.random()*9999);
	}

	private String getBatchId() {
		return "batch"+ Math.round(Math.random()*9999);
	}


	public List<Plan> getPlans() throws Exception{
		return md.getPlans();
	}
	public int addPlan(Plan plan) throws Exception{
		plan.setPlanid(getPlanId());
		return md.addPlan(plan);
	}
	public int deletePlan(String planid) throws Exception{
			return md.deletePlan(planid);
	}
	public Plan getPlan(String planid) throws Exception{
		return md.findPlan(planid);
	}
	public int updatePlan(Plan plan) throws Exception{
		return md.updatePlan(plan);
	}
	public List<Batch> getBatches() throws Exception{
		return md.getBatches();
	}
```

```
	public int addBatch(Batch batch) throws Exception{
		batch.setBatchid(getBatchId());
		return md.addBatch(batch);
	}


	public Batch getBatch(String batchid) throws Exception{
		return md.findBatch(batchid);
	}


	public int deleteBatch(String batchid)  throws Exception{
		return md.deleteBatch(batchid);
	}

	public int updateBatch(Batch batch) throws Exception {
		return md.updateBatch(batch);
	}
}
```

memberservice.java

```
package com.cybage.service;

import java.util.List;

import com.cybage.dao.ManagerDao;
import com.cybage.dao.MemberDao;
import com.cybage.model.Batch;
import com.cybage.model.Enrollment;
import com.cybage.model.Plan;
import com.cybage.model.Users;

public class MemberService {
	MemberDao md = new MemberDao();
	private String getEnrollId() {
		return "enroll"+ Math.round(Math.random()*9999);
	}

	public List<Plan> getPlans() throws Exception{
		return md.getPlans();
	}

	public int enroll(Enrollment enroll) throws Exception{
		enroll.setEnrollid(getEnrollId());
		Batch batch = md.findBatch(enroll.getBatchid());
		int batchSize = batch.getBatchSize();
		if(batchSize > 1) {
			batch.setBatchSize(batchSize-1);
			enroll.setStatus("Enrolled");
			md.updateBatch(batch);
			return md.enroll(enroll);
		}else {
			enroll.setStatus("Rejected");
			return md.enroll(enroll);
```

```
		}
	}
	public List<Enrollment> getEnrollments() throws Exception{
		return md.getEnrollments();
	}

	public int register(Users user) throws Exception{
		return md.register(user);
	}
}
```

userservice.java

```
package com.cybage.service;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.List;

import com.cybage.dao.DbUtil;
import com.cybage.dao.UserDao;
import com.cybage.model.Users;

public class UserService {

	UserDao ud = new UserDao();

	public List<Users> getUsers() throws Exception{
		return ud.findAllUsers();
	}
	public int deleteUser(String username) throws Exception{
		return ud.deleteUser(username);
	}
	public int addUser(Users user) throws Exception{
		return ud.addUser(user);
	}
	public Users getUser(String username) throws Exception{
		return ud.findUser(username);
	}
	public int updateUser(Users user) throws Exception{
		return ud.updateUser(user);
	}
}
```

app.properties

```
dbURL=jdbc:mysql://localhost:3306/cybage
dbUser=root
dbPassword=admin123
```

UI layer

error.jsp

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

```
	pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
	<h2 style="color:red">Invalid user name / password</h2>
	<jsp:include page="login.jsp"></jsp:include>
</body>
</html>
```

index.jsp

```
<html>
<body>
	<h1>Welcome...</h1>
	<br>

	<a href="<%=request.getContextPath()%>/AppController">Login</a>
	<br>
	<a href="<%=request.getContextPath()%>/admin/admin-home.jsp">Admin Login</a>
	<br>
	<a href="<%=request.getContextPath()%>/manager/manager-home.jsp">Manager Login</a>
	<br>
	<a href="<%=request.getContextPath()%>/member/member-home.jsp">Member Login</a>
	<br>
	<a href="<%=request.getContextPath()%>/register.jsp">Register</a>

</body>
</html>
```

login.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
	pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
	<form method="POST" action="j_security_check">
		<table border="0">
			<tr>
				<td>Login</td>
				<td><input type="text" name="j_username"></td>
			</tr>
			<tr>
				<td>Password</td>
				<td><input type="password" name="j_password"></td>
			</tr>
		</table>
```

```
		<input type="submit" value="Login!">

	</form>
</body>
</html>
```

logout.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
	pageEncoding="ISO-8859-1"%>
<%@ page session="true"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
	<h2 style="color:green">
		User '<%=request.getRemoteUser()%>' has been logged out.
	</h2>
	<%
		session.invalidate();
	%>
	<jsp:include page="index.jsp"></jsp:include>
	<br />
	<br />
</body>
</html>
```

register.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
	pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
	<form action="<%=request.getContextPath()%>/MemberController/register" method="post">
		<table border="0">
			<tr>
				<td>Login</td>
				<td><input type="text" name="username"></td>
			</tr>
			<tr>
				<td>Password</td>
				<td><input type="password" name="password"></td>
			</tr>
			<tr>
				<td>Confirm Password</td>
				<td><input type="password" name="cpassword"></td>
			</tr>
		</table>
```

```
                <input type="submit" value="Register">

        </form>
</body>
</html>
```

admin-add-user.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
        <form action="<%=request.getContextPath()%>/AdminController/add" method="post">
                Username: <input type="text" name="username"> <br>
                Password: <input type="password" name="password"> <br>
                userrole:
                <select name="userrole">
                        <option >Select role</option>
                        <option value="admin">Admin</option>
                        <option value="manager">Manager</option>
                        <option value="member">Member</option>
                </select>
                 <br>
                 <input  type="submit" value="Register">
        </form>
</body>
</html>
```

admin-footer.jsp

```
<div>This is footer</div>
</body>
</html>
```

admin-head.jsp

```
 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<nav>
        <ul>
                <li><a href="<%=request.getContextPath()%>/admin/admin-home.jsp">Home</a></li>
                <li><a href="<%=request.getContextPath()%>/AdminController/list">User Management</a></li>
```

```
                <li><a href="<%=request.getContextPath()%>/logout.jsp">Logout</a></li>

        </ul>
</nav>
```

admin-home.jsp

```
<%@include file="admin-head.jsp" %>
        Welcome admin:
        <%
        out.print("user name is :" + request.getRemoteUser());
%>


<%@include file="admin-footer.jsp" %>
```

admin-update-user.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1"%>
<%@ page isELIgnored="false"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
        <form action="<%=request.getContextPath()%>/AdminController/update"
                method="post">
                Username: <input type="text" name="username" value="${user.username }" readonly="readonly">
                <br>
                 Password: <input type="password" name="password" value="${user.username }">
                        <br> userrole: <input
                        type="text" name="userrole" value="${user.role }"> <br>
                <input type="submit" value="Update">
        </form>
</body>
</html>
```

admin-user-management.jsp

```
<%@include file="admin-head.jsp"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ page isELIgnored = "false" %>

<a href="<%=request.getContextPath()%>/admin/admin-add-user.jsp">Add user</a>
<table border="1">
        <tr>
                <th>Username</th>
                <th>User role</th>
        </tr>
        <c:forEach var="table" items="${users}">
                <tr>
                        <td><c:out value="${table.username}" /></td>
                        <td><c:out value="${table.role}" /></td>
```

```
                        <td><a
href="<%=request.getContextPath()%>/AdminController/delete?un=${table.username}">delete</a></td>
                        <td><a
href="<%=request.getContextPath()%>/AdminController/edit?un=${table.username}">update</a></td>
                    </tr>
                </c:forEach>
</table>

<%@include file="admin-footer.jsp"%>
```

manager ui
add-batch.jsp

```
<%@page import="com.cybage.dao.ManagerDao"%>
<%@page import="com.cybage.model.Sports"%>
<%@page import="java.util.List"%>
<%@include file="manager-head.jsp"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ page isELIgnored="false"%>
<form action="<%=request.getContextPath()%>/ManagerController/addbatch"
        method="post">

        Batchname: <input type="text" name="batchname">
        <br>

        Sport:
                <%
                        List<Sports> sports = new ManagerDao().getSports();
                %>
        <select name="sportsid">
                <% for(Sports sport : sports){ %>
                        <option value="<%= sport.getSportsId()%>"><%= sport.getSportsName() %></option>
                <% } %>
        </select>

        <br>
        Batch Start Date: <input type="date" name="batchstartdate">
        <br>
        Batch Time: <input type="time" name="batchtime">
        <br>

        Duration: <input type="text" name="batchduration">

        <br>

        Batch Size:  <input type="text" name="batchsize">
        <input   type="submit" value="Add Batch">
</form>

<%@include file="manager-footer.jsp"%>
```

add-plan.jsp

```
<%@page import="com.cybage.dao.ManagerDao"%>
```

```
<%@page import="com.cybage.model.Sports"%>
<%@page import="java.util.List"%>
<%@include file="manager-head.jsp"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ page isELIgnored="false"%>
<form action="<%=request.getContextPath()%>/ManagerController/addplan"
	method="post">
	Planname: <input type="text" name="planname"> <br> Sport:

		<%
			List<Sports> sports = new ManagerDao().getSports();
		%>
	<select name="sportsid">
		<% for(Sports sport : sports){ %>
			<option value="<%= sport.getSportsId()%>"><%= sport.getSportsName() %></option>
		<% } %>
	</select>

	<br> Fees: <input type="text" name="fees"> <br>
	Duration: <input type="text" name="duration"> <input
		type="submit" value="Add Plan">
</form>

<%@include file="manager-footer.jsp"%>
```

batches.jsp

```
<%@include file="manager-head.jsp"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ page isELIgnored="false"%>

<a href="<%=request.getContextPath()%>/manager/add-batch.jsp">Add
	Plan</a>
<table border="1">
	<tr>
		<th>Batchid </th>
		<th>Batchname</th>
		<th>Sportid </th>
		<th>Batch Start DAte</th>
		<th>Batch time</th>
		<th>Batch duration</th>
		<th>Batch size</th>

	</tr>
	<c:forEach var="batches" items="${batches}">
		<tr>
			<td><c:out value="${batches.batchid}" /></td>
			<td><c:out value="${batches.batchname}" /></td>
			<td><c:out value="${batches.sportsid}" /></td>
			<td><c:out value="${batches.batchStartDate}" /></td>
			<td><c:out value="${batches.batchTime}" /></td>
			<td><c:out value="${batches.batchDuration}" /></td>
			<td><c:out value="${batches.batchSize}" /></td>
			<td><a
```

```
	href="<%=request.getContextPath()%>/ManagerController/deletebatch?batchid=${batches.batchid}">delete</a></td>
				<td><a

	href="<%=request.getContextPath()%>/ManagerController/editbatch?batchid=${batches.batchid}">update</a></td>
		</tr>
	</c:forEach>
</table>

<%@include file="manager-footer.jsp"%>
```

edit-batch.jsp

```
<%@page import="com.cybage.dao.ManagerDao"%>
<%@page import="com.cybage.model.Sports"%>
<%@page import="java.util.List"%>
<%@include file="manager-head.jsp"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ page isELIgnored="false"%>
<form action="<%=request.getContextPath()%>/ManagerController/updatebatch"
	method="post">
	Batchid: <input type="text" name="batchid" value="${batch.batchid}">
	<br>
	Batchname: <input type="text" name="batchname" value="${batch.batchname}">
	<br>

	Sport:
		<%
			List<Sports> sports = new ManagerDao().getSports();
		%>
	<select name="sportsid">
		<% for(Sports sport : sports){ %>
			<option value="<%= sport.getSportsId()%>"><%= sport.getSportsName() %></option>
		<% } %>
	</select>

	<br>
	Batch Start Date: <input type="date" name="batchstartdate" value="${batch.batchStartDate }">
	<br>
	Batch Time: <input type="time" name="batchtime" value="${batch.batchTime }">
	<br>

	Duration: <input type="text" name="batchduration" value="${batch.batchDuration }">

	<br>

	Batch Size:  <input type="text" name="batchsize" value="${batch.batchSize }">
	<input   type="submit" value="Update Batch">
</form>

<%@include file="manager-footer.jsp"%>
```

edit-plan.jsp

```
<%@page import="com.cybage.dao.ManagerDao"%>
<%@page import="com.cybage.model.Sports"%>
<%@page import="java.util.List"%>
<%@include file="manager-head.jsp"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ page isELIgnored="false"%>
<form action="<%=request.getContextPath()%>/ManagerController/updateplan"
	method="post">
	Planid :  <input type="text" name="planid" value="${plan.planid }"> <br>
	Planname: <input type="text" name="planname" value="${plan.planname }"> <br>

	Sport:
		<%
			List<Sports> sports = new ManagerDao().getSports();
		%>
	<select name="sportsid">
		<% for(Sports sport : sports){ %>
			<option value="<%= sport.getSportsId()%>"><%= sport.getSportsName() %></option>
		<% } %>
	</select>

	<br>
	Fees: <input type="text" name="fees" value="${plan.fees }">
	<br>
	Duration: <input type="text" name="duration" value="${plan.duration }">
	<input   type="submit" value="Edit Plan">
</form>

<%@include file="manager-footer.jsp"%>
```

enrollment.jsp

```
<%@include file="manager-head.jsp" %>


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


<hr>
	Welcome user
	<%
	out.print("user name is :" + request.getRemoteUser());
%>

<table border="1">
	<tr>
		<th>Enrollment id</th>
		<th>Plan id</th>
		<th>Batch id </th>
		<th>Start Date</th>
		<th>Status</th>

	</tr>
```

```
		<c:forEach var="en" items="${enrollments}">
			<tr>
				<td><c:out value="${en.enrollid}" /></td>
				<td><c:out value="${en.planid}" /></td>
				<td><c:out value="${en.batchid}" /></td>
				<td><c:out value="${en.startdate}" /></td>
				<td><c:out value="${en.status}" /></td>

			</tr>
		</c:forEach>
</table>

<hr>
<%@include file="manager-footer.jsp" %>
```

manager-footer.jsp

```
<div>this is footer</div>
</body>
</html>
```

manager-head.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
		pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
		Welcome manager
		<%
		out.print("user name is :" + request.getRemoteUser());
%>

		<ul>
			<li><a href="<%=request.getContextPath()%>/ManagerController/planlist">Plans</a></li>
			<li><a href="<%=request.getContextPath()%>/ManagerController/batchlist">Batches</a></li>
			<li><a href="<%=request.getContextPath()%>/ManagerController/enrollments">Enrollments</a></li>
			<li><a href="<%=request.getContextPath()%>/logout.jsp">logout</a></li>
		</ul>
```

manager-home.jsp

```
<%@include file="manager-head.jsp" %>

<%@include file="manager-footer.jsp" %>
```

plans.jsp

```
<%@include file="manager-head.jsp"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ page isELIgnored="false"%>
```

```
<a href="<%=request.getContextPath()%>/manager/add-plan.jsp">Add
	Plan</a>
<table border="1">
	<tr>
		<th>Plan id</th>
		<th>Plan name</th>
		<th>Sports</th>
		<th>Fees</th>
		<th>Duration</th>

	</tr>
	<c:forEach var="table" items="${plans}">
		<tr>
			<td><c:out value="${table.planid}" /></td>
			<td><c:out value="${table.planname}" /></td>
			<td><c:out value="${table.sportid}" /></td>
			<td><c:out value="${table.fees}" /></td>
			<td><c:out value="${table.duration}" /></td>
			<td><a

	href="<%=request.getContextPath()%>/ManagerController/deleteplan?planid=${table.planid}">delete</a></td>
			<td><a

	href="<%=request.getContextPath()%>/ManagerController/editplan?planid=${table.planid}">update</a></td>
		</tr>
	</c:forEach>
</table>

<%@include file="manager-footer.jsp"%>
```

Member ui

enroll.jsp

```
<%@page import="java.util.ArrayList"%>
<%@page import="com.cybage.model.Batch"%>
<%@page import="java.util.List"%>
<%@page import="com.cybage.dao.MemberDao"%>
<%@include file="member-head.jsp" %>


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


<hr>
	Welcome user: <%	out.print("user name is :" + request.getRemoteUser());	%>
	<h1>Enrollment</h1>
	<form action="<%=request.getContextPath()%>/MemberController/makeenrollment" method="post">
	Plan id: <input type="text" name = "planid" value="${planid }" readonly="readonly">

	<%
		MemberDao md = new MemberDao();
		List<Batch> batches =  md.getBatches();
```

```
		%>
		<br>
		Choose Batch:
		<select name="batchid">
		<% for(Batch batch :batches){ %>
			<option value="<%= batch.getBatchid()%>"><%=batch.getBatchname()+ " - time "+ batch.getBatchTime()
%></option>
		<%} %>
		</select>
		<br>
		Start Date: <input type="date" name="startdate">
		<br>
		<input type="submit" value="Enroll">
		</form>
<hr>
<%@include file="member-footer.jsp" %>
```

enrollment.jsp

```
<%@include file="member-head.jsp" %>


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


<hr>
		Welcome user
		<%
		out.print("user name is :" + request.getRemoteUser());
%>

<table border="1">
		<tr>
			<th>Enrollment id</th>
			<th>Plan id</th>
			<th>Batch id </th>
			<th>Start Date</th>
			<th>Status</th>

		</tr>
		<c:forEach var="en" items="${enrollments}">
			<tr>
				<td><c:out value="${en.enrollid}" /></td>
				<td><c:out value="${en.planid}" /></td>
				<td><c:out value="${en.batchid}" /></td>
				<td><c:out value="${en.startdate}" /></td>
				<td><c:out value="${en.status}" /></td>

			</tr>
		</c:forEach>
</table>

<hr>
<%@include file="member-footer.jsp" %>
```

member-footer.jsp

```
<div>this is footer</div>
</body>
</html>
```

member-head.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
	pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

	<ul>
		<li><a href="<%=request.getContextPath()%>/MemberController/planlist">Plans</a></li>
		<li><a href="<%=request.getContextPath()%>/MemberController/enrollments">Enrollment</a></li>
		<li><a href="<%=request.getContextPath()%>/logout.jsp">logout</a></li>
	</ul>
```

member-home.jsp

```
<%@include file="member-head.jsp" %>
<hr>
	Welcome user
	<%
	out.print("user name is :" + request.getRemoteUser());
%>


<hr>
<%@include file="member-footer.jsp" %>
```

plans.jsp

```
<%@include file="member-head.jsp" %>


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


<hr>
	Welcome user
	<%
	out.print("user name is :" + request.getRemoteUser());
%>

<table border="1">
	<tr>
		<th>Plan id</th>
		<th>Plan name</th>
```

```
                <th>Sports</th>
                <th>Fees</th>
                <th>Duration</th>

        </tr>
        <c:forEach var="plan" items="${plans}">
                <tr>
                        <td><c:out value="${plan.planid}" /></td>
                        <td><c:out value="${plan.planname}" /></td>
                        <td><c:out value="${plan.sportid}" /></td>
                        <td><c:out value="${plan.fees}" /></td>
                        <td><c:out value="${plan.duration}" /></td>
                        <td><a

        href="<%=request.getContextPath()%>/MemberController/enroll?planid=${plan.planid}">Enroll</a></td>
                </tr>
        </c:forEach>
</table>

<hr>
<%@include file="member-footer.jsp" %>
```

web.xml

```
<!DOCTYPE web-app PUBLIC
 "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
 "http://java.sun.com/dtd/web-app_2_3.dtd" >

<web-app>

        <display-name>Archetype Created Web Application</display-name>
        <servlet>
                <servlet-name>AdminController</servlet-name>
                <display-name>AdminController</display-name>
                <description></description>
                <servlet-class>com.cybage.controller.AdminController</servlet-class>
        </servlet>

        <servlet>
                <servlet-name>ManagerController</servlet-name>
                <display-name>ManagerController</display-name>
                <description></description>
                <servlet-class>com.cybage.controller.ManagerController</servlet-class>
        </servlet>
        <servlet>
                <servlet-name>AppController</servlet-name>
                <display-name>AppController</display-name>
                <description></description>
                <servlet-class>com.cybage.controller.AppController</servlet-class>
        </servlet>
        <servlet>
                <servlet-name>MemberController</servlet-name>
                <display-name>MemberController</display-name>
                <description></description>
                <servlet-class>com.cybage.controller.MemberController</servlet-class>
        </servlet>
```

```
<servlet-mapping>
	<servlet-name>AdminController</servlet-name>
	<url-pattern>/AdminController/*</url-pattern>
</servlet-mapping>

<servlet-mapping>
	<servlet-name>ManagerController</servlet-name>
	<url-pattern>/ManagerController/*</url-pattern>
</servlet-mapping>
<servlet-mapping>
	<servlet-name>AppController</servlet-name>
	<url-pattern>/AppController/*</url-pattern>
</servlet-mapping>
<servlet-mapping>
	<servlet-name>MemberController</servlet-name>
	<url-pattern>/MemberController/*</url-pattern>
</servlet-mapping>

<error-page>
	<exception-type>java.lang.Exception</exception-type>
	<location>/error.jsp</location>
</error-page>


<security-constraint>
	<web-resource-collection>
		<web-resource-name>SecuredBookSite</web-resource-name>
		<url-pattern>/admin/*</url-pattern>
	</web-resource-collection>
	<auth-constraint>
		<description>Let only admin use this app</description>
		<role-name>admin</role-name>
	</auth-constraint>
</security-constraint>
<security-constraint>
	<web-resource-collection>
		<web-resource-name>SecuredBookSite</web-resource-name>
		<url-pattern>/member/*</url-pattern>
	</web-resource-collection>
	<auth-constraint>
		<description>Let only member use this app</description>
		<role-name>member</role-name>
	</auth-constraint>
</security-constraint>

<security-constraint>
	<web-resource-collection>
		<web-resource-name>SecuredBookSite</web-resource-name>
		<url-pattern>/manager/*</url-pattern>
	</web-resource-collection>
	<auth-constraint>
		<description>Let only managers use this app</description>
		<role-name>manager</role-name>
	</auth-constraint>
```

```
	</security-constraint>


	<login-config>
		<auth-method>FORM</auth-method>
		<form-login-config>
			<form-login-page>/login.jsp</form-login-page>
			<form-error-page>/error.jsp</form-error-page>
		</form-login-config>
	</login-config>

</web-app>
```

pom.xml

```
<project 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 http://maven.apache.org/maven-v4_0_0.xsd">
	<modelVersion>4.0.0</modelVersion>
	<groupId>com.cybage</groupId>
	<artifactId>case-study</artifactId>
	<packaging>war</packaging>
	<version>1.1</version>
	<name>case-study Maven Webapp</name>
	<url>http://maven.apache.org</url>
	<dependencies>
		<!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->
		<dependency>
			<groupId>javax.servlet</groupId>
			<artifactId>javax.servlet-api</artifactId>
			<version>4.0.1</version>
			<scope>provided</scope>
		</dependency>
		<!-- https://mvnrepository.com/artifact/javax.servlet.jsp.jstl/jstl -->
		<dependency>
			<groupId>javax.servlet</groupId>
			<artifactId>jstl</artifactId>
			<version>1.2</version>
		</dependency>
		<!-- https://mvnrepository.com/artifact/javax.servlet.jsp/javax.servlet.jsp-api -->
		<dependency>
			<groupId>javax.servlet.jsp</groupId>
			<artifactId>javax.servlet.jsp-api</artifactId>
			<version>2.3.3</version>
			<scope>provided</scope>
		</dependency>
		<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
		<dependency>
			<groupId>mysql</groupId>
			<artifactId>mysql-connector-java</artifactId>
			<version>8.0.22</version>
		</dependency>
		<!-- https://mvnrepository.com/artifact/org.apache.commons/commons-dbcp2 -->
		<dependency>
			<groupId>org.apache.commons</groupId>
			<artifactId>commons-dbcp2</artifactId>
```

```
                <version>2.8.0</version>
            </dependency>
            <!-- https://mvnrepository.com/artifact/org.apache.commons/commons-pool2 -->
            <dependency>
                <groupId>org.apache.commons</groupId>
                <artifactId>commons-pool2</artifactId>
                <version>2.8.0</version>
            </dependency>
            <!-- https://mvnrepository.com/artifact/commons-logging/commons-logging -->
            <dependency>
                <groupId>commons-logging</groupId>
                <artifactId>commons-logging</artifactId>
                <version>1.2</version>
            </dependency>

        </dependencies>
        <build>
            <finalName>case-study</finalName>
        </build>
</project>
```

server.xml

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


<Server port="8888" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener"/>
  <!-- Security listener. Documentation at /docs/config/listeners.html
  <Listener className="org.apache.catalina.security.SecurityListener" />
  -->
  <!--APR library loader. Documentation at /docs/apr.html -->
  <Listener SSLEngine="on" className="org.apache.catalina.core.AprLifecycleListener"/>
  <!-- Prevent memory leaks due to use of particular java/javax APIs-->
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener"/>
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener"/>
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener"/>

  <!-- Global JNDI resources
       Documentation at /docs/jndi-resources-howto.html
  -->
  <GlobalNamingResources>
    <!-- Editable user database that can also be used by
         UserDatabaseRealm to authenticate users
    -->
    <Resource auth="Container" description="User database that can be updated and saved"
factory="org.apache.catalina.users.MemoryUserDatabaseFactory" name="UserDatabase" pathname="conf/tomcat-
users.xml" type="org.apache.catalina.UserDatabase"/>
  </GlobalNamingResources>

  <!-- A "Service" is a collection of one or more "Connectors" that share
       a single "Container" Note:  A "Service" is not itself a "Container",
       so you may not define subcomponents such as "Valves" at this level.
       Documentation at /docs/config/service.html
   -->
```

```
<Service name="Catalina">

  <!--The connectors can use a shared executor, you can define one or more named thread pools-->
  <!--
  <Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
    maxThreads="150" minSpareThreads="4"/>
  -->


  <!-- A "Connector" represents an endpoint by which requests are received
       and responses are returned. Documentation at :
       Java HTTP Connector: /docs/config/http.html
       Java AJP  Connector: /docs/config/ajp.html
       APR (HTTP/AJP) Connector: /docs/apr.html
       Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
  -->
  <Connector connectionTimeout="20000" port="8081" protocol="HTTP/1.1" redirectPort="8443"/>
  <!-- A "Connector" using the shared thread pool-->
  <!--
  <Connector executor="tomcatThreadPool"
           port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
  -->
  <!-- Define an SSL/TLS HTTP/1.1 Connector on port 8443
       This connector uses the NIO implementation. The default
       SSLImplementation will depend on the presence of the APR/native
       library and the useOpenSSL attribute of the
       AprLifecycleListener.
       Either JSSE or OpenSSL style configuration may be used regardless of
       the SSLImplementation selected. JSSE style configuration is used below.
  -->
  <!--
  <Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
           maxThreads="150" SSLEnabled="true">
    <SSLHostConfig>
      <Certificate certificateKeystoreFile="conf/localhost-rsa.jks"
                   type="RSA" />
    </SSLHostConfig>
  </Connector>
  -->
  <!-- Define an SSL/TLS HTTP/1.1 Connector on port 8443 with HTTP/2
       This connector uses the APR/native implementation which always uses
       OpenSSL for TLS.
       Either JSSE or OpenSSL style configuration may be used. OpenSSL style
       configuration is used below.
  -->
  <!--
  <Connector port="8443" protocol="org.apache.coyote.http11.Http11AprProtocol"
           maxThreads="150" SSLEnabled="true" >
    <UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />
    <SSLHostConfig>
      <Certificate certificateKeyFile="conf/localhost-rsa-key.pem"
                   certificateFile="conf/localhost-rsa-cert.pem"
```

```
                certificateChainFile="conf/localhost-rsa-chain.pem"
                type="RSA" />
        </SSLHostConfig>
    </Connector>
    -->

    <!-- Define an AJP 1.3 Connector on port 8009 -->
    <!--
    <Connector protocol="AJP/1.3"
            address="::1"
            port="8009"
            redirectPort="8443" />
    -->

    <!-- An Engine represents the entry point (within Catalina) that processes
         every request.  The Engine implementation for Tomcat stand alone
         analyzes the HTTP headers included with the request, and passes them
         on to the appropriate Host (virtual host).
         Documentation at /docs/config/engine.html -->

    <!-- You should set jvmRoute to support load-balancing via AJP ie :
    <Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
    -->
    <Engine defaultHost="localhost" name="Catalina">

      <!--For clustering, please take a look at documentation at:
          /docs/cluster-howto.html  (simple how to)
          /docs/config/cluster.html (reference documentation) -->
      <!--
      <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
      -->

      <!-- Use the LockOutRealm to prevent attempts to guess user passwords
           via a brute-force attack -->
      <Realm className="org.apache.catalina.realm.LockOutRealm">
        <!-- This Realm uses the UserDatabase configured in the global JNDI
             resources under the key "UserDatabase".  Any edits
             that are performed against this UserDatabase are immediately
             available for use by the Realm.  -->
        <Realm className="org.apache.catalina.realm.UserDatabaseRealm" resourceName="UserDatabase"/>
      </Realm>

<Realm className="org.apache.catalina.realm.JDBCRealm" connectionName="root" connectionPassword="admin123"
connectionURL="jdbc:mysql://localhost:3306/cybage" driverName="com.mysql.jdbc.Driver" roleNameCol="userrole"
userCredCol="password" userNameCol="username" userRoleTable="users" userTable="users"/>

      <Host appBase="webapps" autoDeploy="true" name="localhost" unpackWARs="true">

        <!-- SingleSignOn valve, share authentication between web applications
             Documentation at: /docs/config/valve.html -->
        <!--
        <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
        -->
```

```
      <!-- Access log processes all example.
           Documentation at: /docs/config/valve.html
           Note: The pattern used is equivalent to using pattern="common" -->
      <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs" pattern="%h %l %u %t
&amp;quot;%r&amp;quot; %s %b" prefix="localhost_access_log" suffix=".txt"/>

    <Context docBase="case-study2" path="/case-study" reloadable="true" source="org.eclipse.jst.j2ee.server:case-
study2"/></Host>
   </Engine>
  </Service>
</Server>
```

---

sql.sql

```
use cybage;

create table sports (sportsid varchar(20), sportsname varchar(20), primary key(sportsid));
insert into sports values ('sport101', 'cricket');
insert into sports values ('sport102', 'football');
insert into sports values ('sport103', 'tennis');
insert into sports values ('sport104', 'yoga');
insert into sports values ('sport105', 'badmintan');


create table users (username varchar(20), password varchar(20), userrole varchar(20), primary key (username));
insert into users values('manager',  'manager', 'manager');
insert into users values('user',  'user', 'user');
insert into users values('admin',  'admin', 'admin');




create table plans(planid varchar(20), planname varchar(20), sportsid varchar(20), fees double, duration int, primary
key(planid), FOREIGN KEY (sportsid) REFERENCES sports(sportsid));



create table batches(batchid varchar(20), batchname varchar(20), sportsid varchar(20), batchstartdate date, batchtime time,
batchduration double, batchsize int, primary key (batchid),
 FOREIGN KEY (sportsid) REFERENCES sports(sportsid));



create table enrollment (enrollid varchar(20), username varchar(20), planid varchar(20), batchid varchar(20), startdate date,
status varchar(20),
          primary key(enrollid),
   FOREIGN KEY (username) REFERENCES users(username),
   FOREIGN KEY (planid) REFERENCES plans(planid),
   FOREIGN KEY (batchid) REFERENCES batches(batchid)
)
```