FPT Education
FPT UNIVERSITY

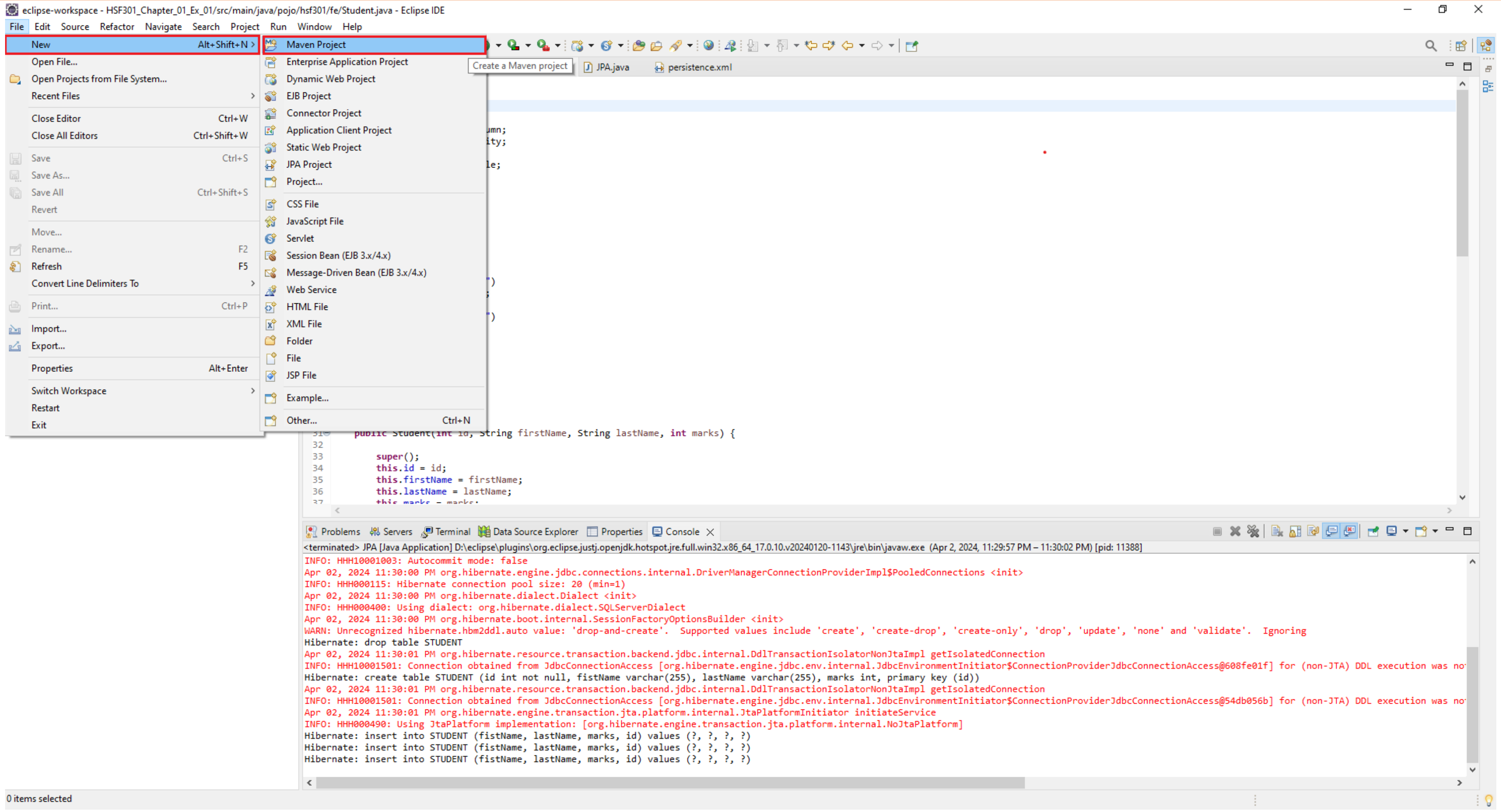
Build a simple application with Data Access using Spring & Spring JDBC, Spring ORM

Objectives

Build Desktop application with Data access

- Create a new Maven project in Eclipse IDE
- Add the necessary dependencies for Spring MVC and data access
- Create the Model - Define the entity classes that represent domain objects
- Define the data access object (DAO) interfaces and their implementations for interacting with the database.
- Create the Controller
- Set Up the Views
- Create the views using JSP, Thymeleaf, or another templating engine
- Implement the Business Logic
- Testing
- Run the Application

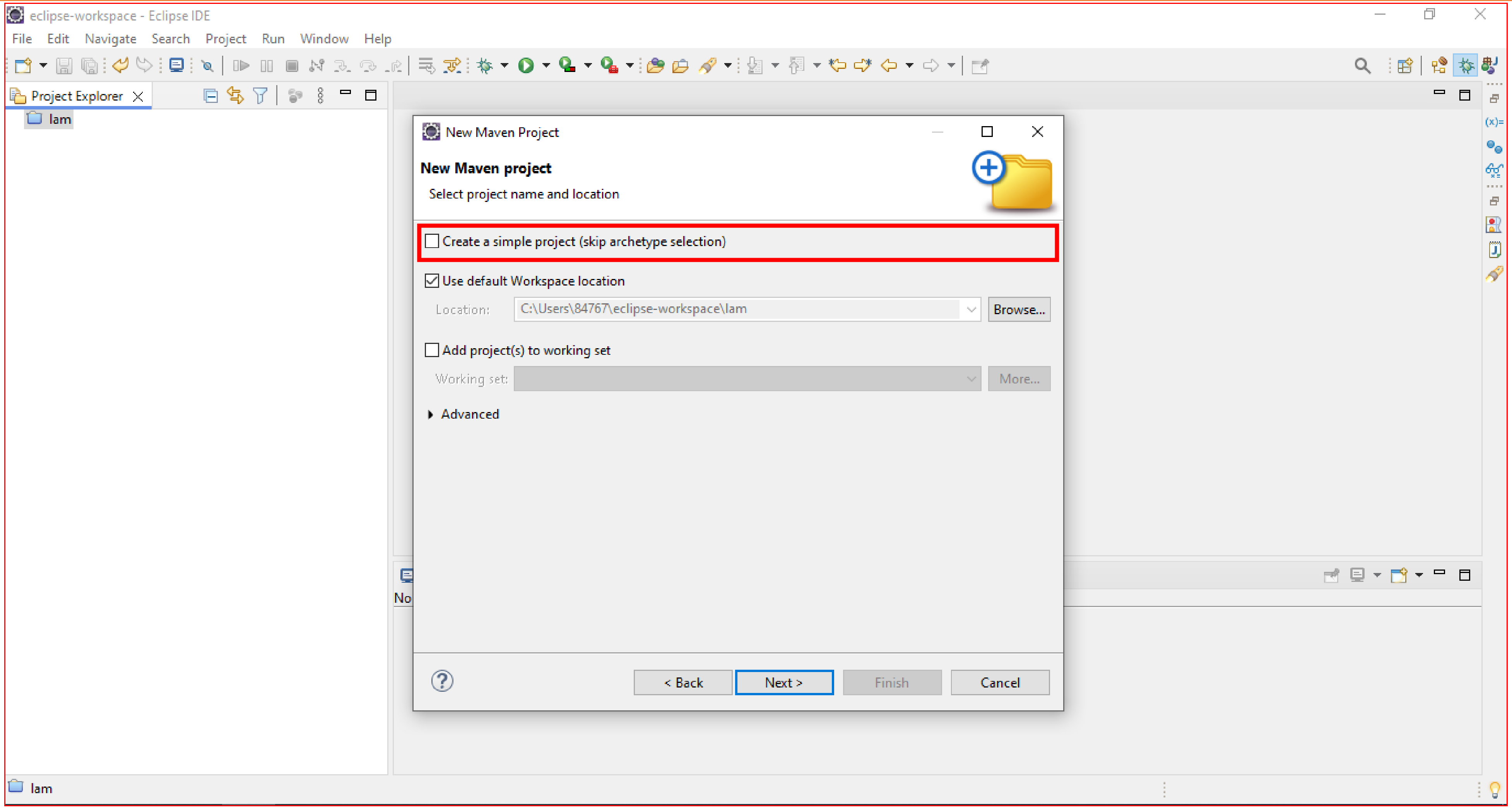
1. Open Eclipse, File | New | Maven Project



```
    public Student(int id, String firstName, String lastName, int marks) {

        super();
        this.id = id;
        this.firstName = firstName;
        this.lastName = lastName;
```

```
<terminated> JPA [Java Application] D:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe (Apr 2, 2024, 11:29:57 PM – 11:30:02 PM) [pid: 11388]
INFO: HHH10001003: Autocommit mode: false
Apr 02, 2024 11:30:00 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <init>
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
Apr 02, 2024 11:30:00 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.SQLServerDialect
Apr 02, 2024 11:30:00 PM org.hibernate.boot.internal.SessionFactoryOptionsBuilder <init>
WARN: Unrecognized hibernate.hbm2ddl.auto value: 'drop-and-create'.  Supported values include 'create', 'create-drop', 'create-only', 'drop', 'update', 'none' and 'validate'.  Ignoring
Hibernate: drop table STUDENT
Apr 02, 2024 11:30:01 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProviderJdbcConnectionAccess@608fe01f] for (non-JTA) DDL execution was no
Hibernate: create table STUDENT (id int not null, fistName varchar(255), lastName varchar(255), marks int, primary key (id))
Apr 02, 2024 11:30:01 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProviderJdbcConnectionAccess@54db056b] for (non-JTA) DDL execution was no
Apr 02, 2024 11:30:01 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
Hibernate: insert into STUDENT (fistName, lastName, marks, id) values (?, ?, ?, ?)
Hibernate: insert into STUDENT (fistName, lastName, marks, id) values (?, ?, ?, ?)
Hibernate: insert into STUDENT (fistName, lastName, marks, id) values (?, ?, ?, ?)
```

2. Check Create a simple project -> Browse Project -> Next



New Maven Project

New Maven project

Select project name and location

☐ Create a simple project (skip archetype selection)

☑ Use default Workspace location

Location: C:\Users\84767\eclipse-workspace\lam

☐ Add project(s) to working set

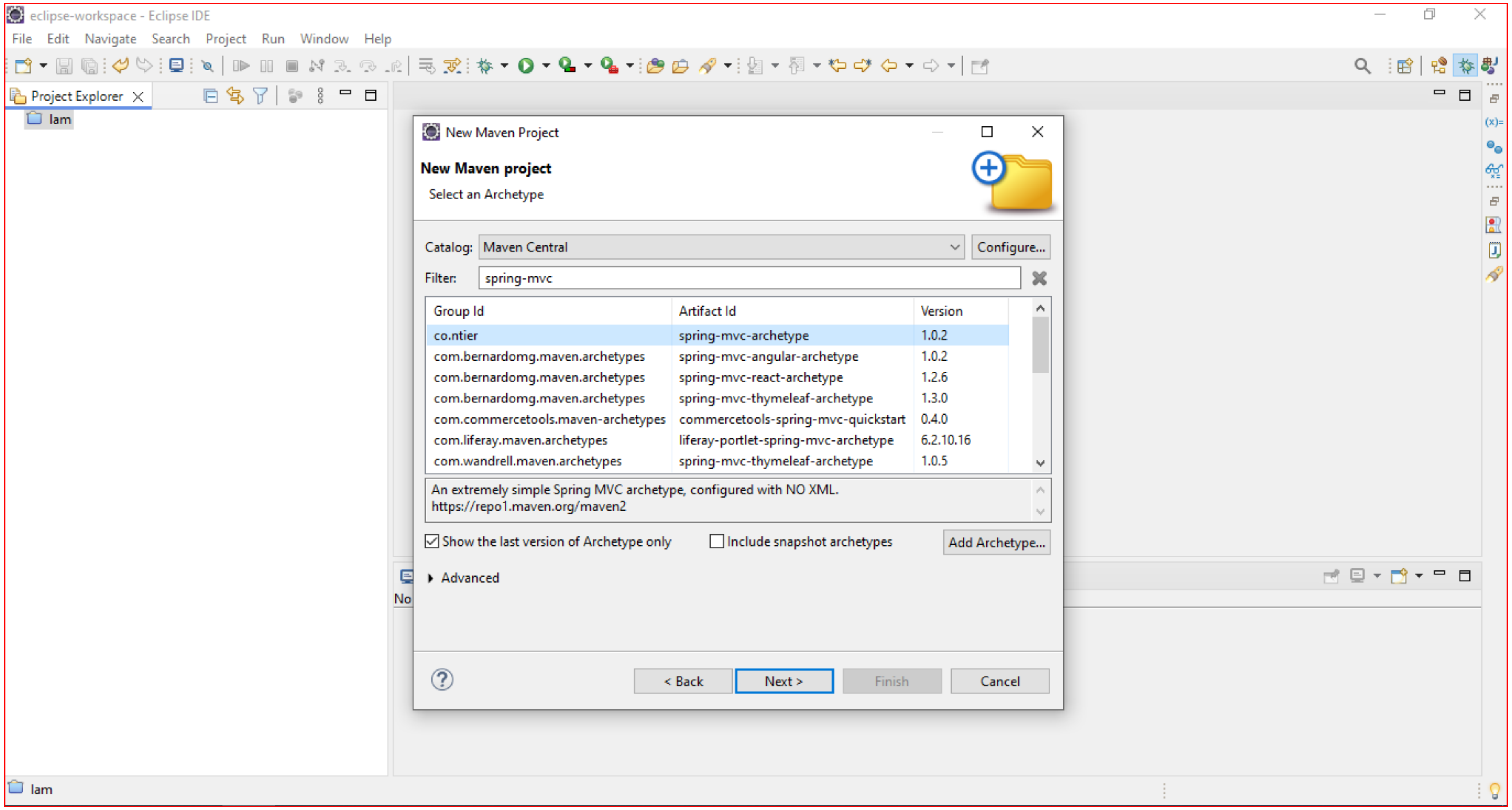
Working set:

Advanced

eclipse-workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Project Explorer
lam
Browse...
More...
< Back
Next >
Finish
Cancel

FPT Education
FPT UNIVERSITY

3. Fill the information Project -> Click Finish


eclipse-workspace - Eclipse IDE

New Maven Project

New Maven project

Select an Archetype

Catalog: Maven Central

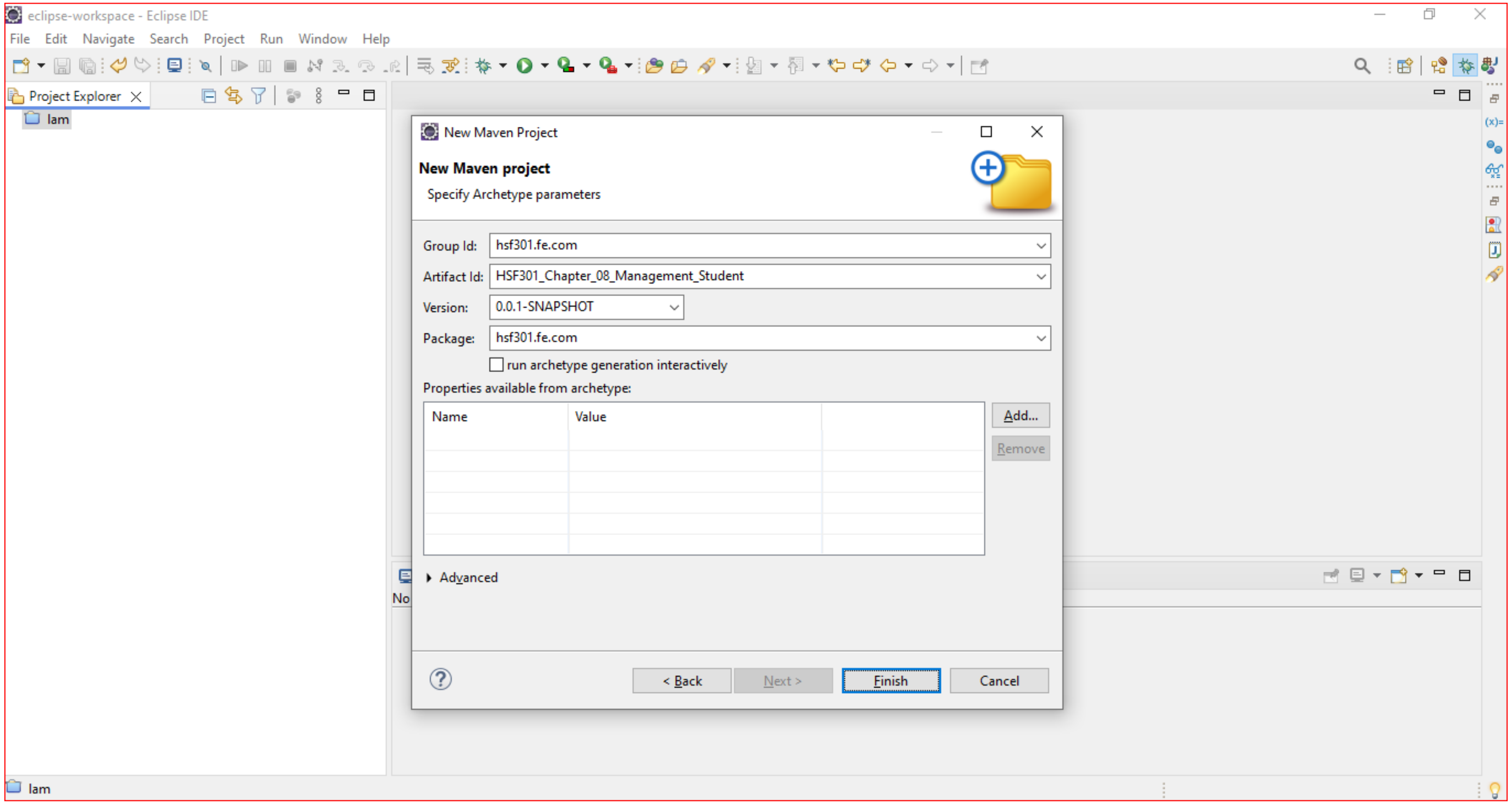
Filter: spring-mvc

Group Id	Artifact Id	Version
co.ntier	spring-mvc-archetype	1.0.2
com.bernardomg.maven.archetypes	spring-mvc-angular-archetype	1.0.2
com.bernardomg.maven.archetypes	spring-mvc-react-archetype	1.2.6
com.bernardomg.maven.archetypes	spring-mvc-thymeleaf-archetype	1.3.0
com.commercetools.maven-archetypes	commercetools-spring-mvc-quickstart	0.4.0
com.liferay.maven.archetypes	liferay-portlet-spring-mvc-archetype	6.2.10.16
com.wandrell.maven.archetypes	spring-mvc-thymeleaf-archetype	1.0.5

An extremely simple Spring MVC archetype, configured with NO XML.
https://repo1.maven.org/maven2

[✓] Show the last version of Archetype only
[ ] Include snapshot archetypes

4. Fill the information Project -> Click Finish



eclipse-workspace - Eclipse IDE

New Maven Project

New Maven project

Specify Archetype parameters

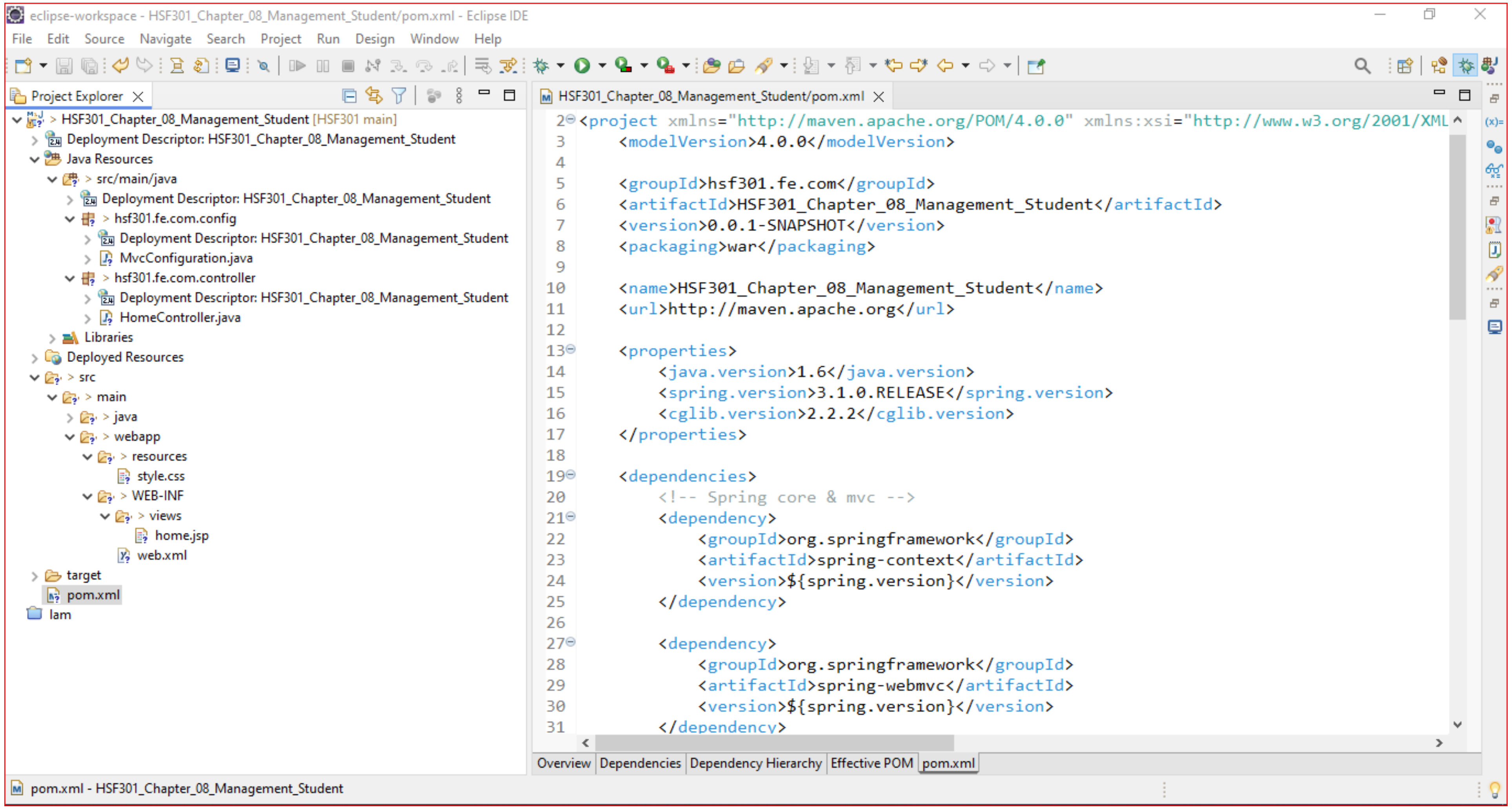
Group Id:	hsf301.fe.com
Artifact Id:	HSF301_Chapter_08_Management_Student
Version:	0.0.1-SNAPSHOT
Package:	hsf301.fe.com

☐ run archetype generation interactively

Properties available from archetype:

Name	Value

5. Structure of Maven Project

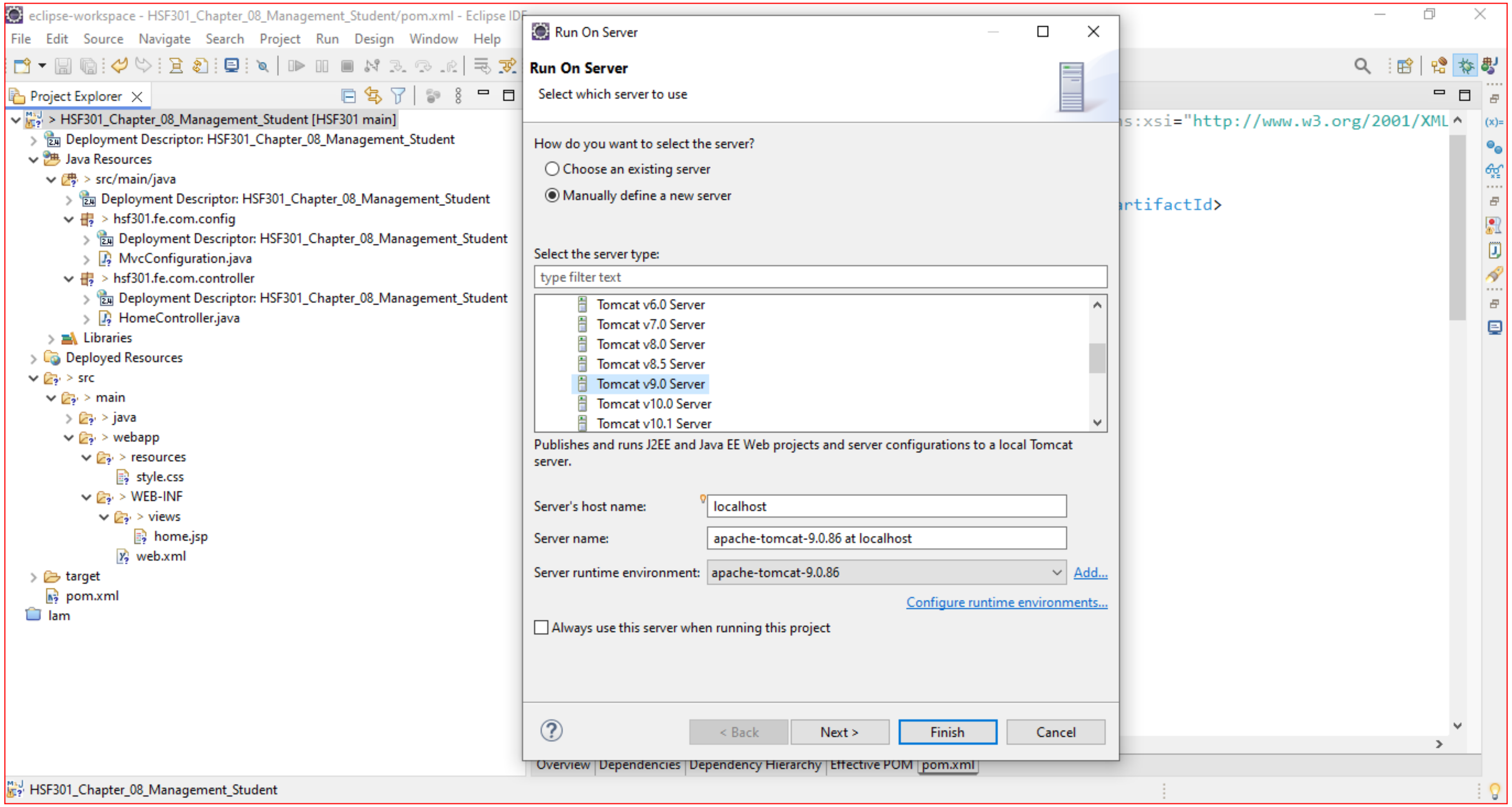

eclipse-workspace - HSF301_Chapter_08_Management_Student/pom.xml - Eclipse IDE
File Edit Source Navigate Search Project Run Design Window Help
Project Explorer
HSF301_Chapter_08_Management_Student/pom.xml
Overview
Dependencies
Dependency Hierarchy
Effective POM
pom.xml
pom.xml - HSF301_Chapter_08_Management_Student

```
> HSF301_Chapter_08_Management_Student [HSF301 main]
    Deployment Descriptor: HSF301_Chapter_08_Management_Student
    Java Resources
        > src/main/java
            Deployment Descriptor: HSF301_Chapter_08_Management_Student
            > hsf301.fe.com.config
                Deployment Descriptor: HSF301_Chapter_08_Management_Student
                MvcConfiguration.java
            > hsf301.fe.com.controller
                Deployment Descriptor: HSF301_Chapter_08_Management_Student
                HomeController.java
        Libraries
    Deployed Resources
    > src
        > main
            > java
            > webapp
                > resources
                    style.css
                > WEB-INF
                    > views
                        home.jsp
                    web.xml
    target
    pom.xml
lam
```

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XML
    <modelVersion>4.0.0</modelVersion>

    <groupId>hsf301.fe.com</groupId>
    <artifactId>HSF301_Chapter_08_Management_Student</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>war</packaging>

    <name>HSF301_Chapter_08_Management_Student</name>
    <url>http://maven.apache.org</url>

    <properties>
        <java.version>1.6</java.version>
        <spring.version>3.1.0.RELEASE</spring.version>
        <cglib.version>2.2.2</cglib.version>
    </properties>

    <dependencies>
        <!-- Spring core & mvc -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>${spring.version}</version>
        </dependency>

        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-webmvc</artifactId>
            <version>${spring.version}</version>
        </dependency>
```

FPT Education
FPT UNIVERSITY

6. Right Click -> Run On Server


eclipse-workspace - HSF301_Chapter_08_Management_Student/pom.xml - Eclipse IDE
File Edit Source Navigate Search Project Run Design Window Help
Project Explorer

```
> HSF301_Chapter_08_Management_Student [HSF301 main]
  Deployment Descriptor: HSF301_Chapter_08_Management_Student
  Java Resources
    > src/main/java
      Deployment Descriptor: HSF301_Chapter_08_Management_Student
      > hsf301.fe.com.config
        Deployment Descriptor: HSF301_Chapter_08_Management_Student
        MvcConfiguration.java
      > hsf301.fe.com.controller
        Deployment Descriptor: HSF301_Chapter_08_Management_Student
        HomeController.java
    Libraries
  Deployed Resources
  > src
    > main
      > java
      > webapp
        > resources
          style.css
        > WEB-INF
          > views
            home.jsp
          web.xml
  target
  pom.xml
lam
```

Run On Server

Run On Server
Select which server to use

How do you want to select the server?
- ( ) Choose an existing server
- (●) Manually define a new server

Select the server type:
type filter text
- Tomcat v6.0 Server
- Tomcat v7.0 Server
- Tomcat v8.0 Server
- Tomcat v8.5 Server
- Tomcat v9.0 Server
- Tomcat v10.0 Server
- Tomcat v10.1 Server

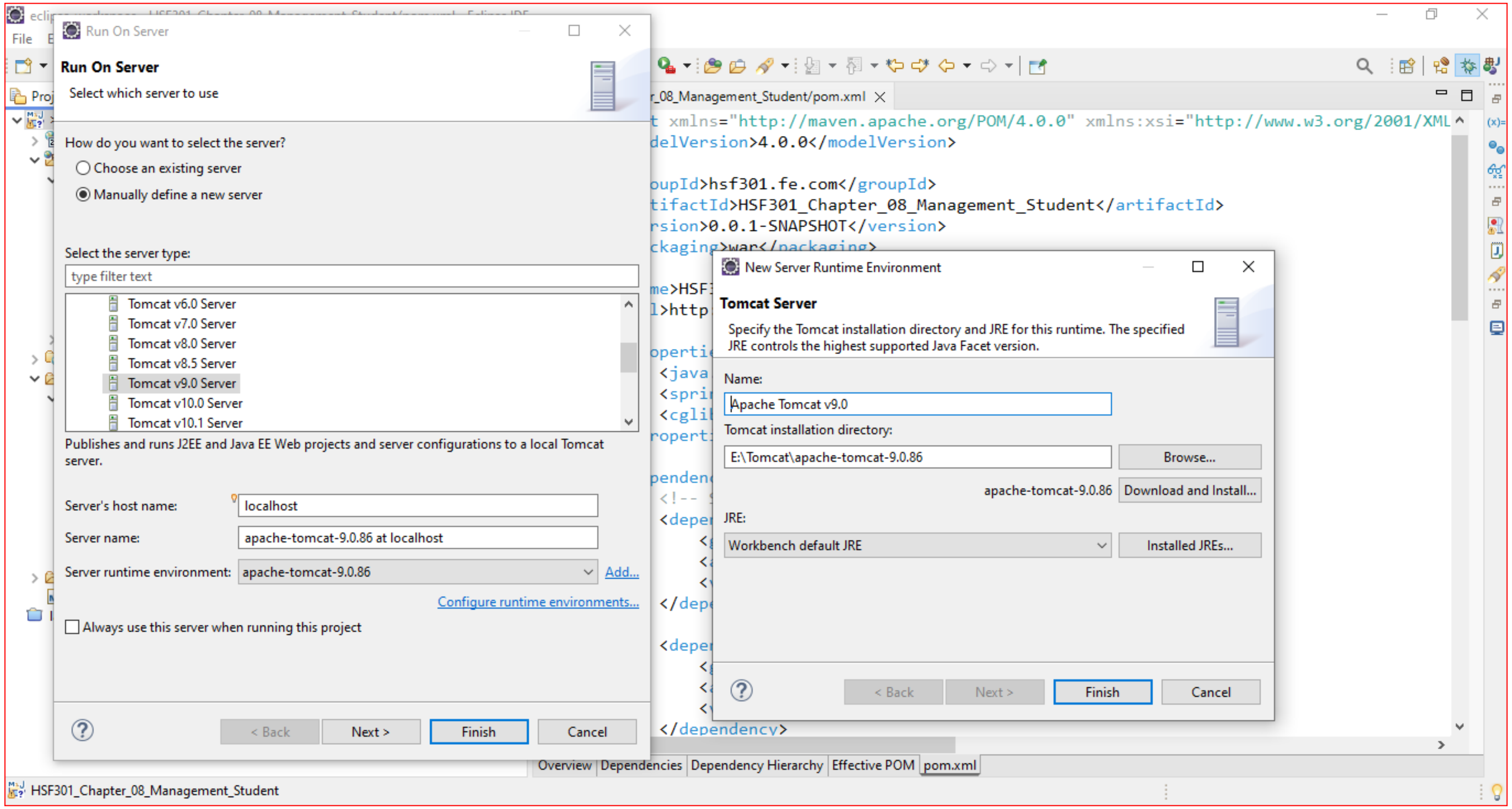
Publishes and runs J2EE and Java EE Web projects and server configurations to a local Tomcat server.

Server's host name: localhost
Server name: apache-tomcat-9.0.86 at localhost
Server runtime environment: apache-tomcat-9.0.86 Add...
Configure runtime environments...
☐ Always use this server when running this project

< Back | Next > | Finish | Cancel

s:xsi="http://www.w3.org/2001/XML
artifactId>
Overview Dependencies Dependency Hierarchy Effective POM pom.xml
HSF301_Chapter_08_Management_Student

7. Browse the Server Location



Run On Server

Run On Server

Select which server to use

How do you want to select the server?

( ) Choose an existing server

(●) Manually define a new server

Select the server type:

type filter text

- Tomcat v6.0 Server
- Tomcat v7.0 Server
- Tomcat v8.0 Server
- Tomcat v8.5 Server
- Tomcat v9.0 Server
- Tomcat v10.0 Server
- Tomcat v10.1 Server

Publishes and runs J2EE and Java EE Web projects and server configurations to a local Tomcat server.

Server's host name: localhost

Server name: apache-tomcat-9.0.86 at localhost

Server runtime environment: apache-tomcat-9.0.86 Add...

Configure runtime environments...

[ ] Always use this server when running this project

< Back | Next > | Finish | Cancel

New Server Runtime Environment

Tomcat Server

Specify the Tomcat installation directory and JRE for this runtime. The specified JRE controls the highest supported Java Facet version.

Name: Apache Tomcat v9.0

Tomcat installation directory: E:\Tomcat\apache-tomcat-9.0.86 Browse...

apache-tomcat-9.0.86 Download and Install...

JRE: Workbench default JRE Installed JREs...

< Back | Next > | Finish | Cancel

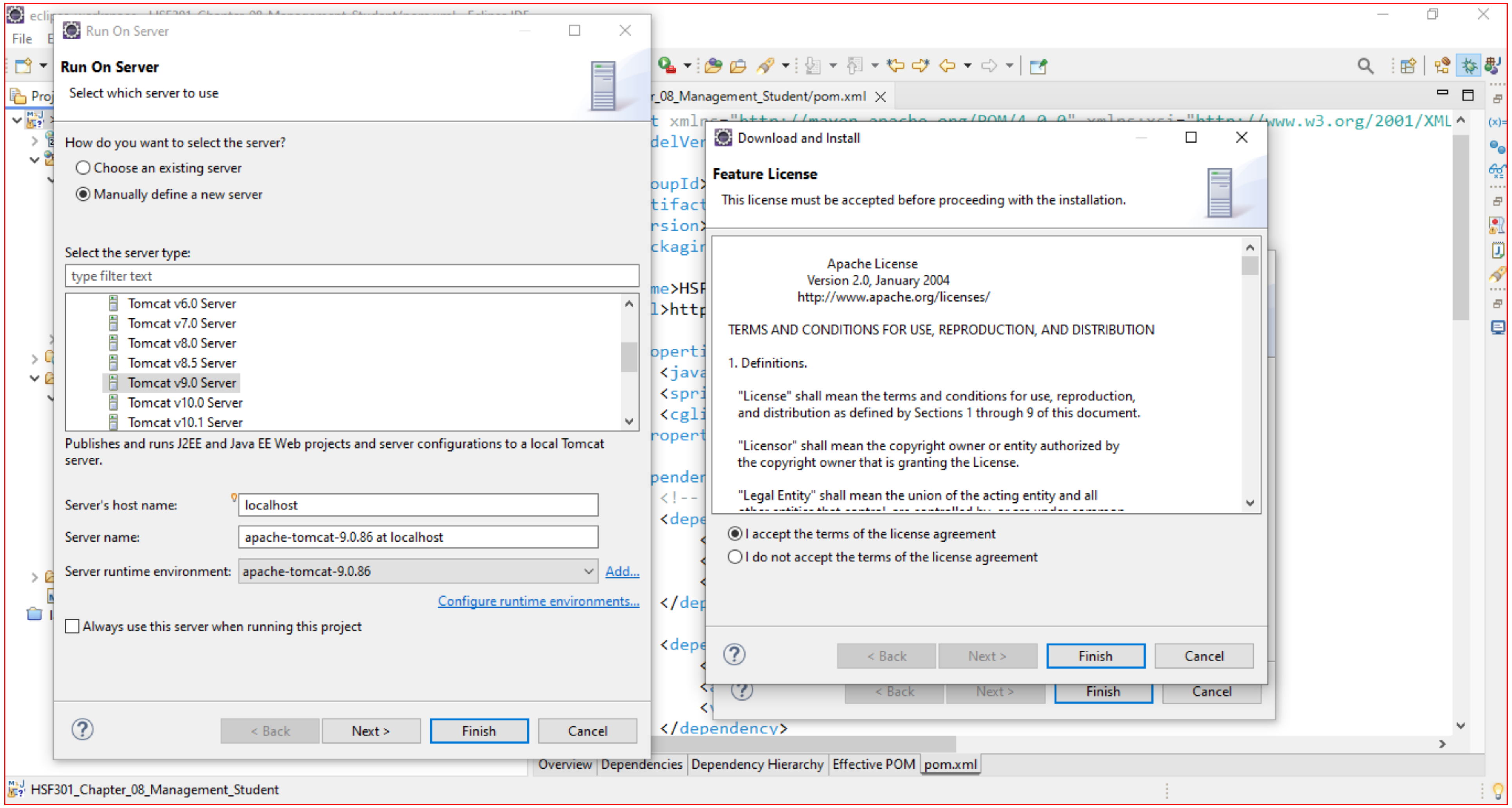
```
t xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XML
delVersion>4.0.0</modelVersion>

oupId>hsf301.fe.com</groupId>
tifactId>HSF301_Chapter_08_Management_Student</artifactId>
rsion>0.0.1-SNAPSHOT</version>
ckaging>war</packaging>
```

Overview | Dependencies | Dependency Hierarchy | Effective POM | pom.xml

HSF301_Chapter_08_Management_Student

FPT Education
FPT UNIVERSITY

8. Click Download and Install



Run On Server

Run On Server

Select which server to use

How do you want to select the server?

( ) Choose an existing server
(●) Manually define a new server

Select the server type:

type filter text

- Tomcat v6.0 Server
- Tomcat v7.0 Server
- Tomcat v8.0 Server
- Tomcat v8.5 Server
- Tomcat v9.0 Server
- Tomcat v10.0 Server
- Tomcat v10.1 Server

Publishes and runs J2EE and Java EE Web projects and server configurations to a local Tomcat server.

Server's host name:	localhost
Server name:	apache-tomcat-9.0.86 at localhost
Server runtime environment:	apache-tomcat-9.0.86

[ ] Always use this server when running this project

Download and Install

Feature License

This license must be accepted before proceeding with the installation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

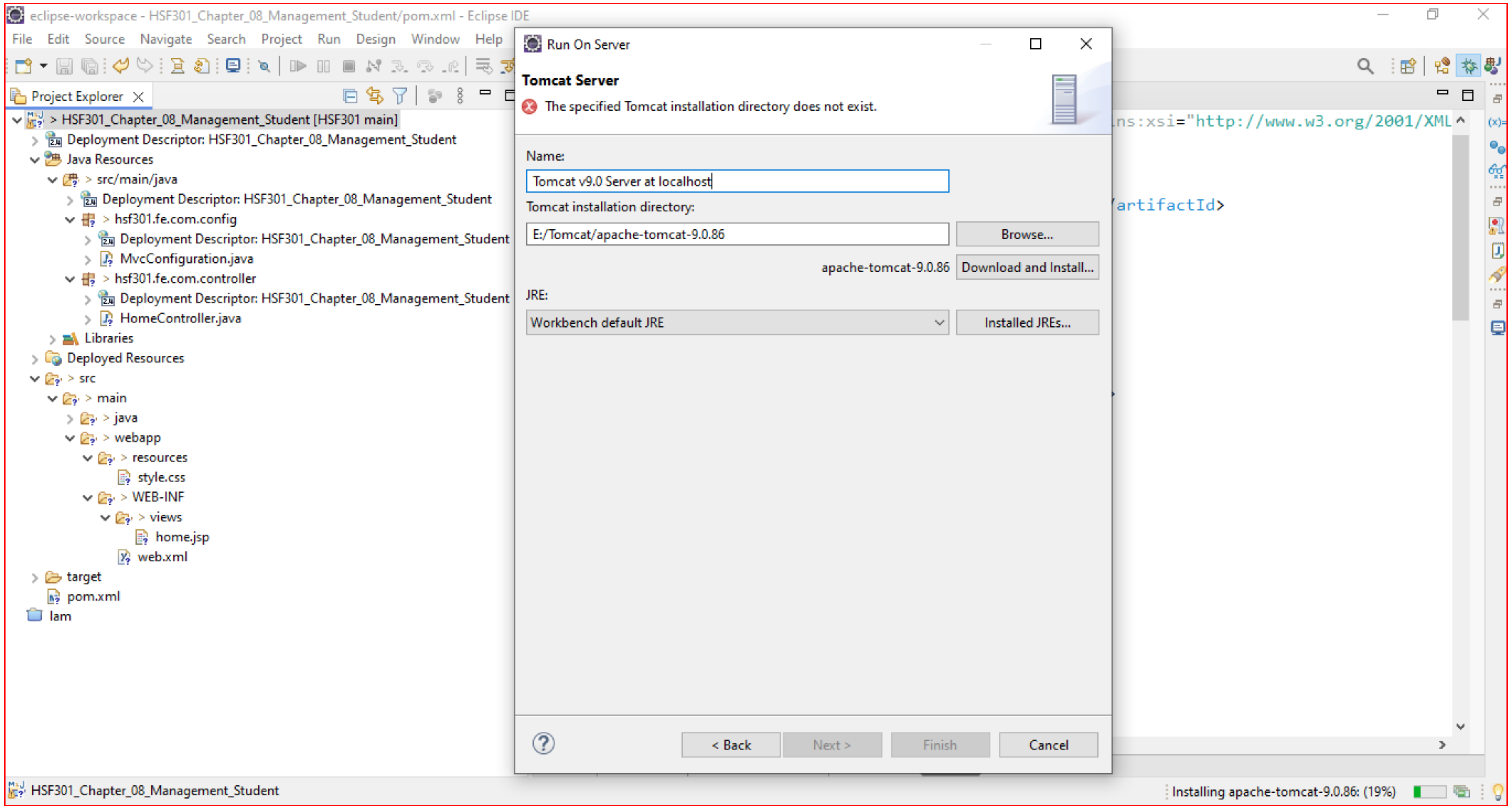
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

(●) I accept the terms of the license agreement
( ) I do not accept the terms of the license agreement

9. Next -> Finish



eclipse-workspace - HSF301_Chapter_08_Management_Student/pom.xml - Eclipse IDE

Run On Server

Tomcat Server

The specified Tomcat installation directory does not exist.

Name: Tomcat v9.0 Server at localhost

Tomcat installation directory: E:/Tomcat/apache-tomcat-9.0.86

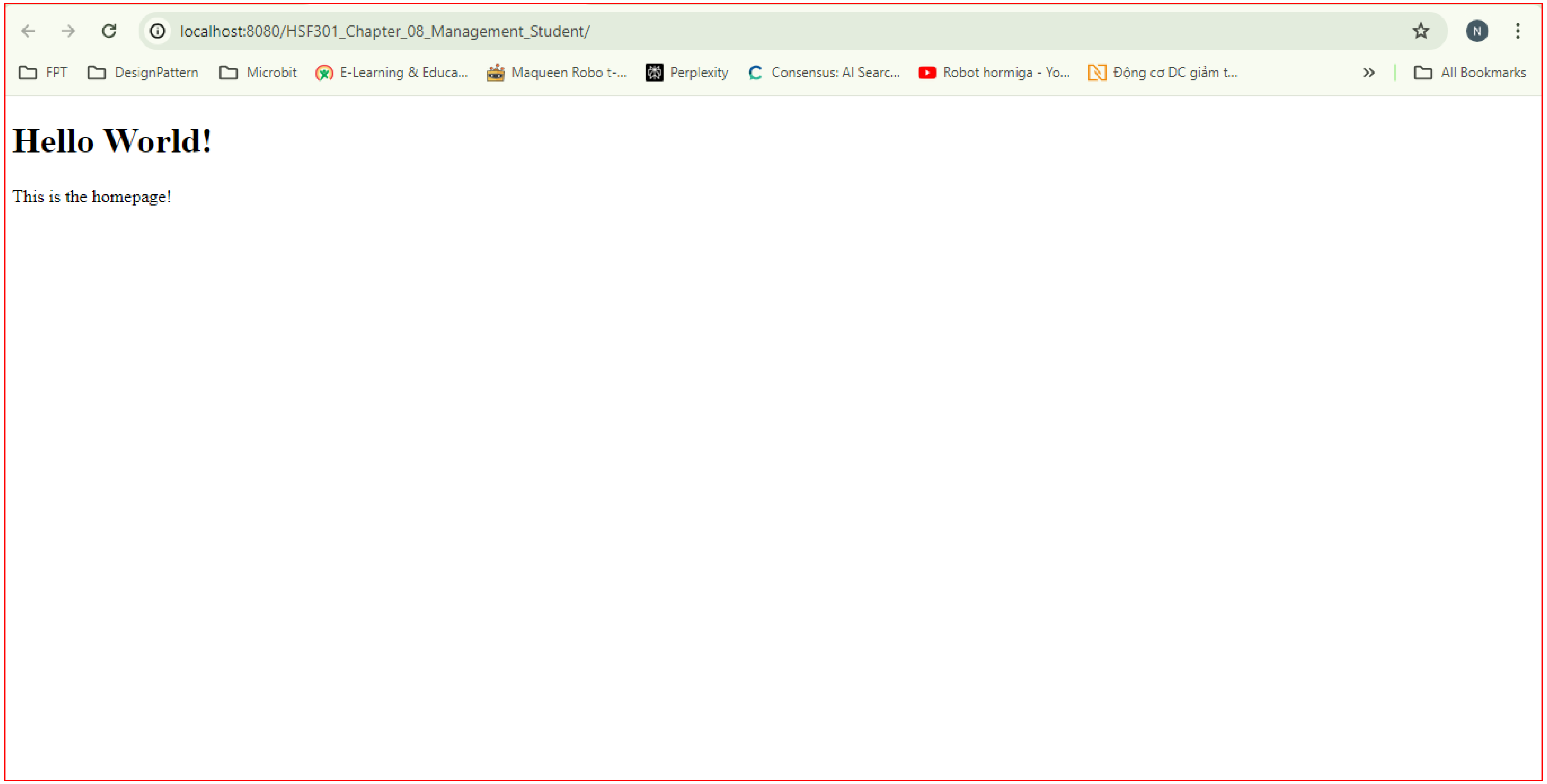
apache-tomcat-9.0.86

JRE: Workbench default JRE

```
HSF301_Chapter_08_Management_Student [HSF301 main]
  Deployment Descriptor: HSF301_Chapter_08_Management_Student
  Java Resources
    src/main/java
      Deployment Descriptor: HSF301_Chapter_08_Management_Student
      hsf301.fe.com.config
        Deployment Descriptor: HSF301_Chapter_08_Management_Student
        MvcConfiguration.java
      hsf301.fe.com.controller
        Deployment Descriptor: HSF301_Chapter_08_Management_Student
        HomeController.java
    Libraries
  Deployed Resources
  src
    main
      java
      webapp
        resources
          style.css
        WEB-INF
          views
            home.jsp
          web.xml
  target
  pom.xml
lam
```

Installing apache-tomcat-9.0.86: (19%)

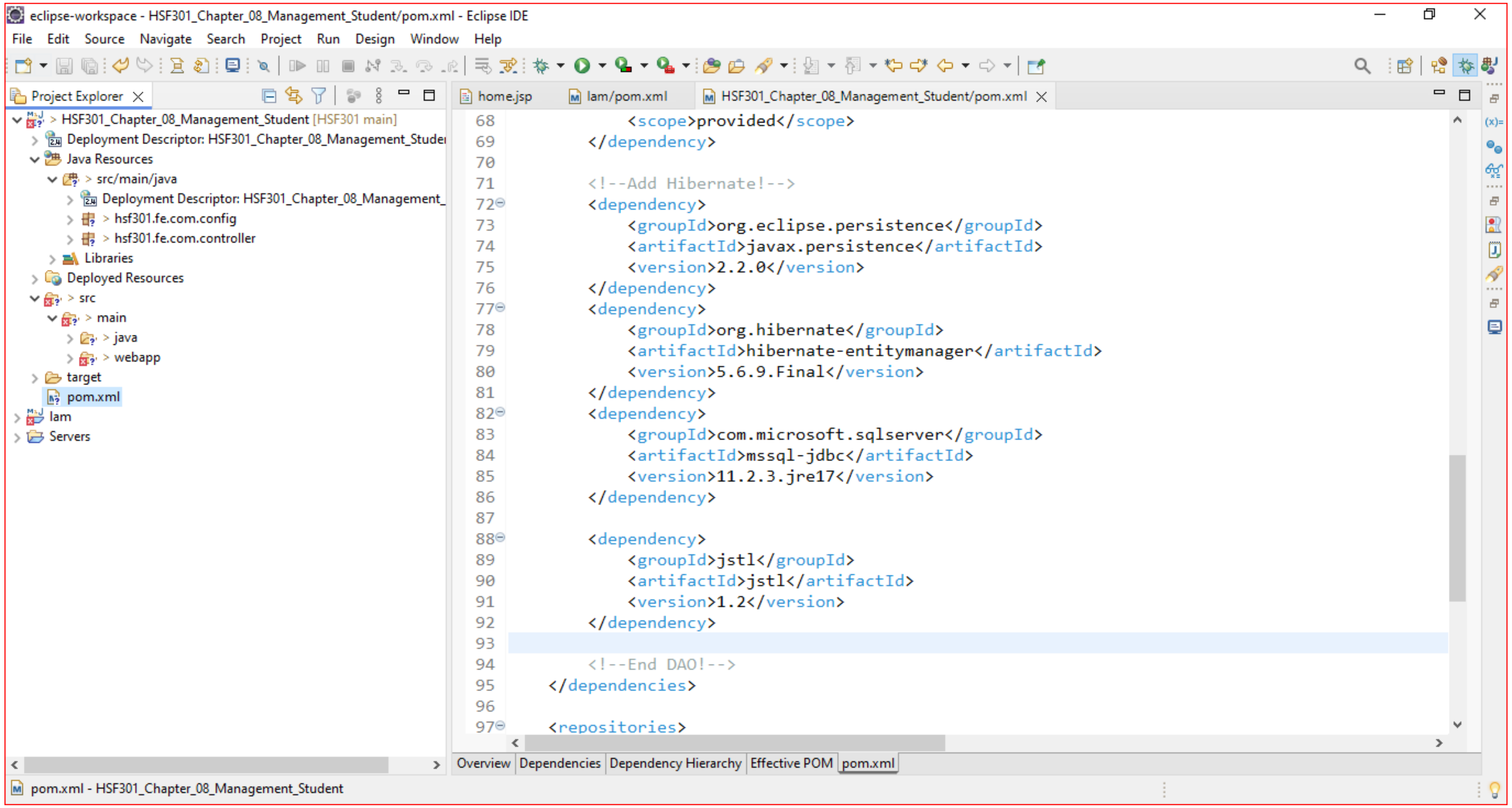
10. Result



```
localhost:8080/HSF301_Chapter_08_Management_Student/
```

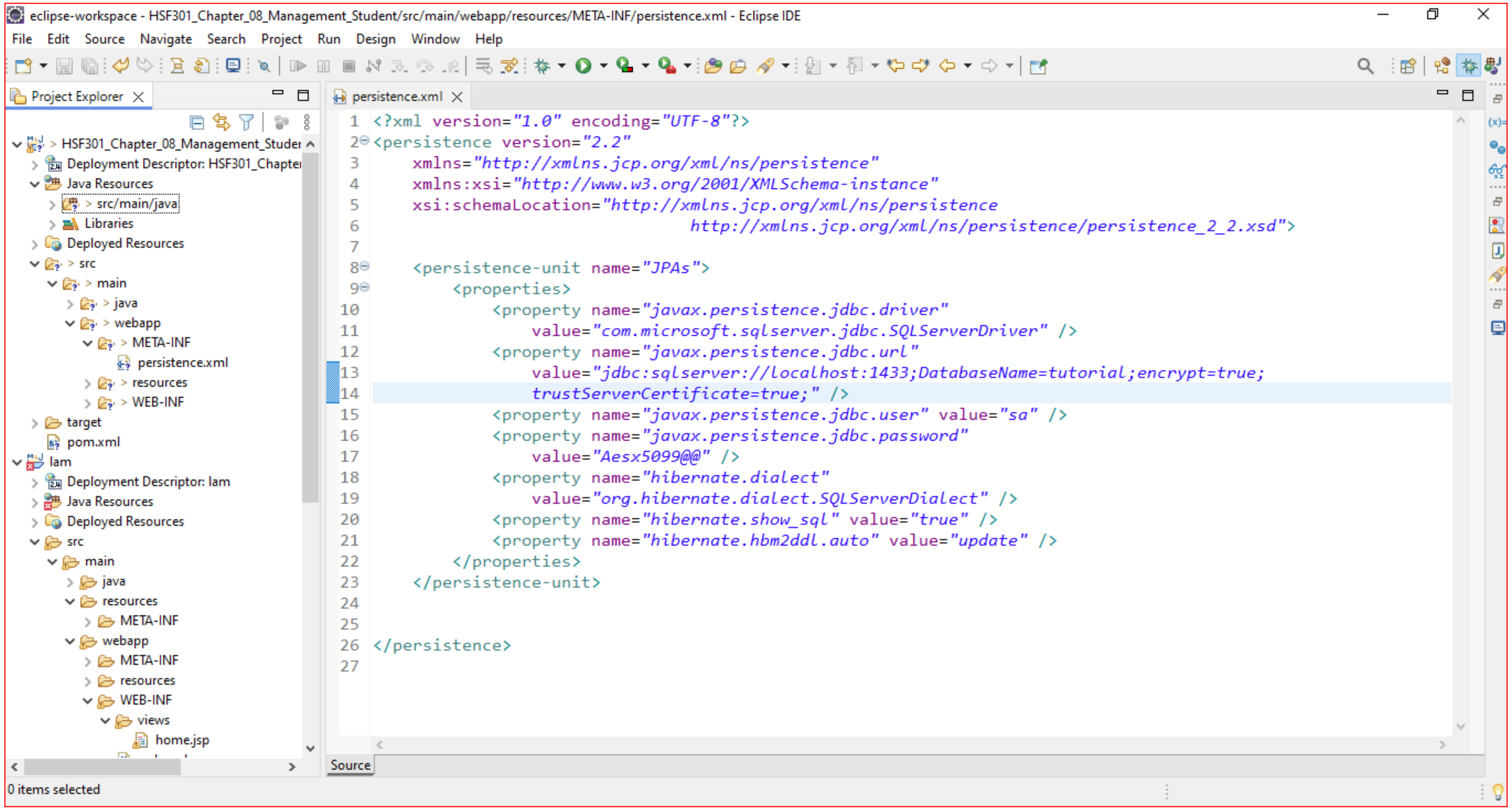
```
Hello World!

This is the homepage!
```

11. Edit pom.xml


eclipse-workspace - HSF301_Chapter_08_Management_Student/pom.xml - Eclipse IDE
File Edit Source Navigate Search Project Run Design Window Help
Project Explorer
HSF301_Chapter_08_Management_Student [HSF301 main]
Deployment Descriptor: HSF301_Chapter_08_Management_Student
Java Resources
src/main/java
Deployment Descriptor: HSF301_Chapter_08_Management_
hsf301.fe.com.config
hsf301.fe.com.controller
Libraries
Deployed Resources
src
main
java
webapp
target
pom.xml
lam
Servers
home.jsp
lam/pom.xml
HSF301_Chapter_08_Management_Student/pom.xml
Overview Dependencies Dependency Hierarchy Effective POM pom.xml
pom.xml - HSF301_Chapter_08_Management_Student

```xml
            <scope>provided</scope>
        </dependency>

        <!--Add Hibernate!-->
        <dependency>
            <groupId>org.eclipse.persistence</groupId>
            <artifactId>javax.persistence</artifactId>
            <version>2.2.0</version>
        </dependency>
        <dependency>
            <groupId>org.hibernate</groupId>
            <artifactId>hibernate-entitymanager</artifactId>
            <version>5.6.9.Final</version>
        </dependency>
        <dependency>
            <groupId>com.microsoft.sqlserver</groupId>
            <artifactId>mssql-jdbc</artifactId>
            <version>11.2.3.jre17</version>
        </dependency>

        <dependency>
            <groupId>jstl</groupId>
            <artifactId>jstl</artifactId>
            <version>1.2</version>
        </dependency>

        <!--End DAO!-->
    </dependencies>

    <repositories>
```

12. Create META-INF and persistence.xml File

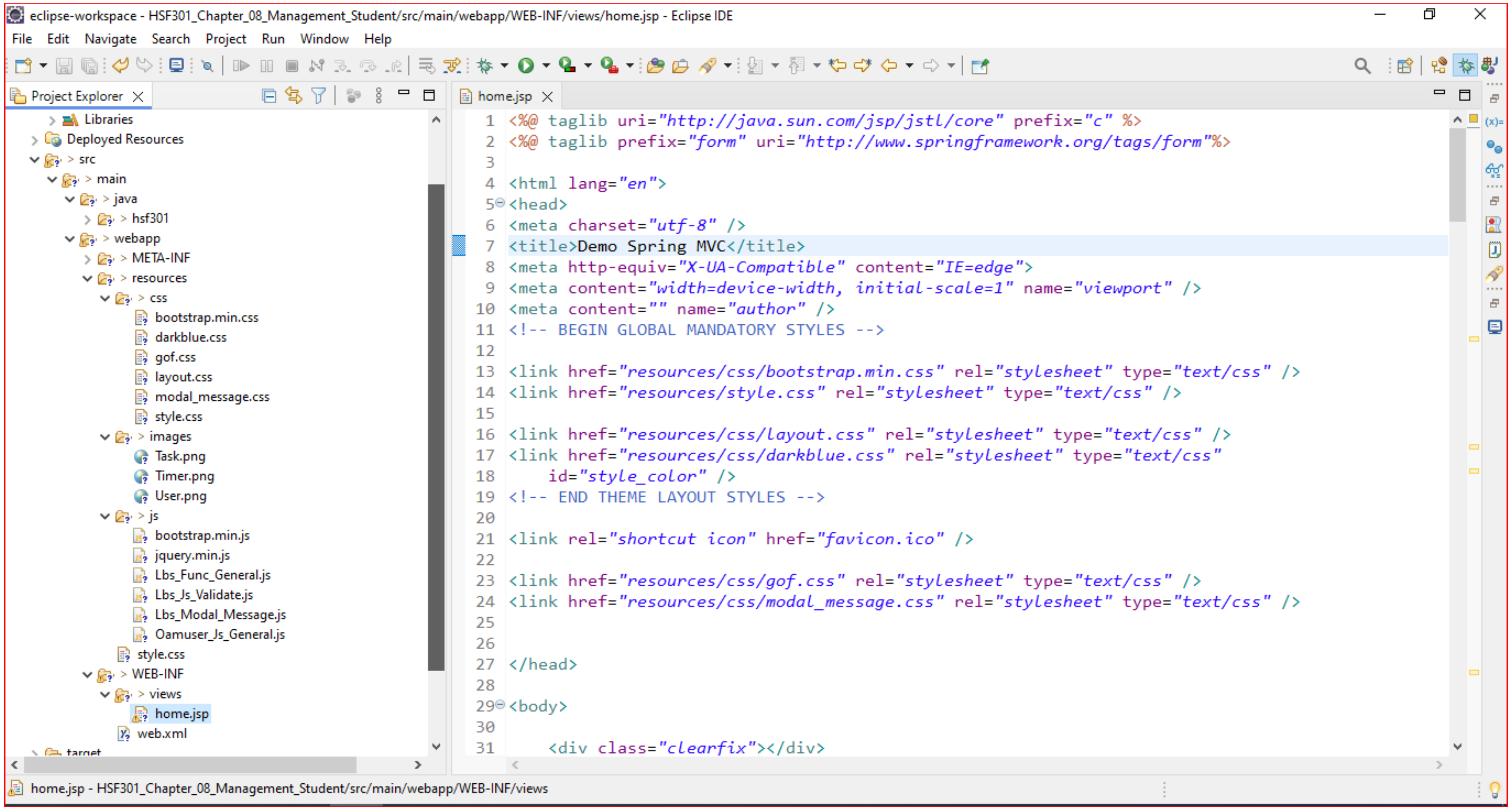


eclipse-workspace - HSF301_Chapter_08_Management_Student/src/main/webapp/resources/META-INF/persistence.xml - Eclipse IDE
File Edit Source Navigate Search Project Run Design Window Help
Project Explorer
persistence.xml
Source
0 items selected

```
HSF301_Chapter_08_Management_Studer
  Deployment Descriptor: HSF301_Chapter
  Java Resources
    src/main/java
    Libraries
  Deployed Resources
  src
    main
      java
      webapp
        META-INF
          persistence.xml
        resources
        WEB-INF
  target
  pom.xml
lam
  Deployment Descriptor: lam
  Java Resources
  Deployed Resources
  src
    main
      java
      resources
        META-INF
      webapp
        META-INF
        resources
        WEB-INF
          views
            home.jsp
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.2"
    xmlns="http://xmlns.jcp.org/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
                        http://xmlns.jcp.org/xml/ns/persistence/persistence_2_2.xsd">

    <persistence-unit name="JPAs">
        <properties>
            <property name="javax.persistence.jdbc.driver"
                value="com.microsoft.sqlserver.jdbc.SQLServerDriver" />
            <property name="javax.persistence.jdbc.url"
                value="jdbc:sqlserver://localhost:1433;DatabaseName=tutorial;encrypt=true;
                trustServerCertificate=true;" />
            <property name="javax.persistence.jdbc.user" value="sa" />
            <property name="javax.persistence.jdbc.password"
                value="Aesx5099@@" />
            <property name="hibernate.dialect"
                value="org.hibernate.dialect.SQLServerDialect" />
            <property name="hibernate.show_sql" value="true" />
            <property name="hibernate.hbm2ddl.auto" value="update" />
        </properties>
    </persistence-unit>


</persistence>

```

13. Copy material into project



eclipse-workspace - HSF301_Chapter_08_Management_Student/src/main/webapp/WEB-INF/views/home.jsp - Eclipse IDE

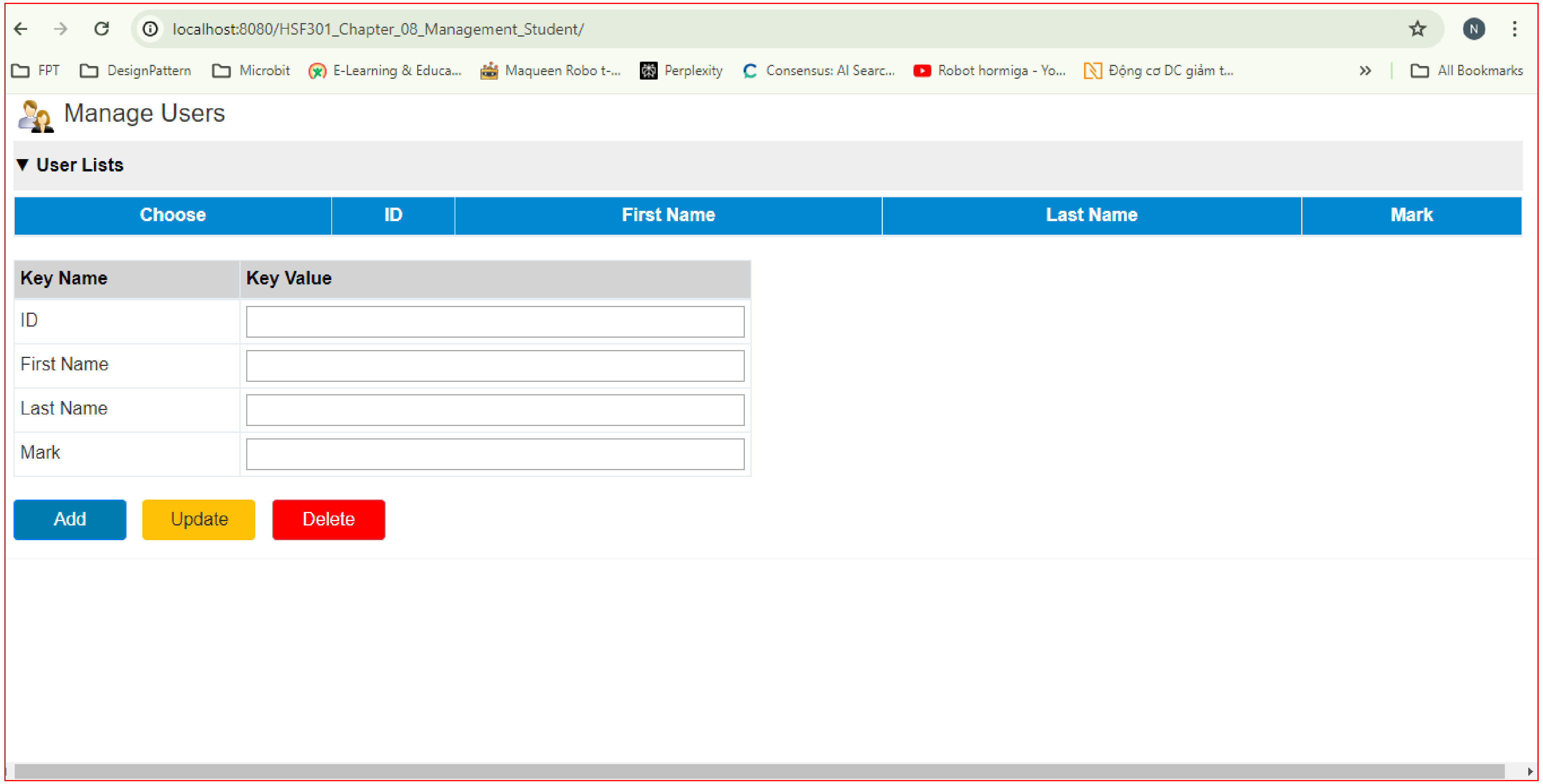
```
> Libraries
> Deployed Resources
v > src
  v > main
    v > java
      > > hsf301
    v > webapp
      > > META-INF
      v > resources
        v > css
            bootstrap.min.css
            darkblue.css
            gof.css
            layout.css
            modal_message.css
            style.css
        v > images
            Task.png
            Timer.png
            User.png
        v > js
            bootstrap.min.js
            jquery.min.js
            Lbs_Func_General.js
            Lbs_Js_Validate.js
            Lbs_Modal_Message.js
            Oamuser_Js_General.js
          style.css
      v > WEB-INF
        v > views
            home.jsp
          web.xml
> target
```

```jsp
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>

<html lang="en">
<head>
<meta charset="utf-8" />
<title>Demo Spring MVC</title>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta content="width=device-width, initial-scale=1" name="viewport" />
<meta content="" name="author" />
<!-- BEGIN GLOBAL MANDATORY STYLES -->

<link href="resources/css/bootstrap.min.css" rel="stylesheet" type="text/css" />
<link href="resources/style.css" rel="stylesheet" type="text/css" />

<link href="resources/css/layout.css" rel="stylesheet" type="text/css" />
<link href="resources/css/darkblue.css" rel="stylesheet" type="text/css"
    id="style_color" />
<!-- END THEME LAYOUT STYLES -->

<link rel="shortcut icon" href="favicon.ico" />

<link href="resources/css/gof.css" rel="stylesheet" type="text/css" />
<link href="resources/css/modal_message.css" rel="stylesheet" type="text/css" />


</head>

<body>

    <div class="clearfix"></div>
```

home.jsp - HSF301_Chapter_08_Management_Student/src/main/webapp/WEB-INF/views

FPT Education
FPT UNIVERSITY

14. Result


localhost:8080/HSF301_Chapter_08_Management_Student/

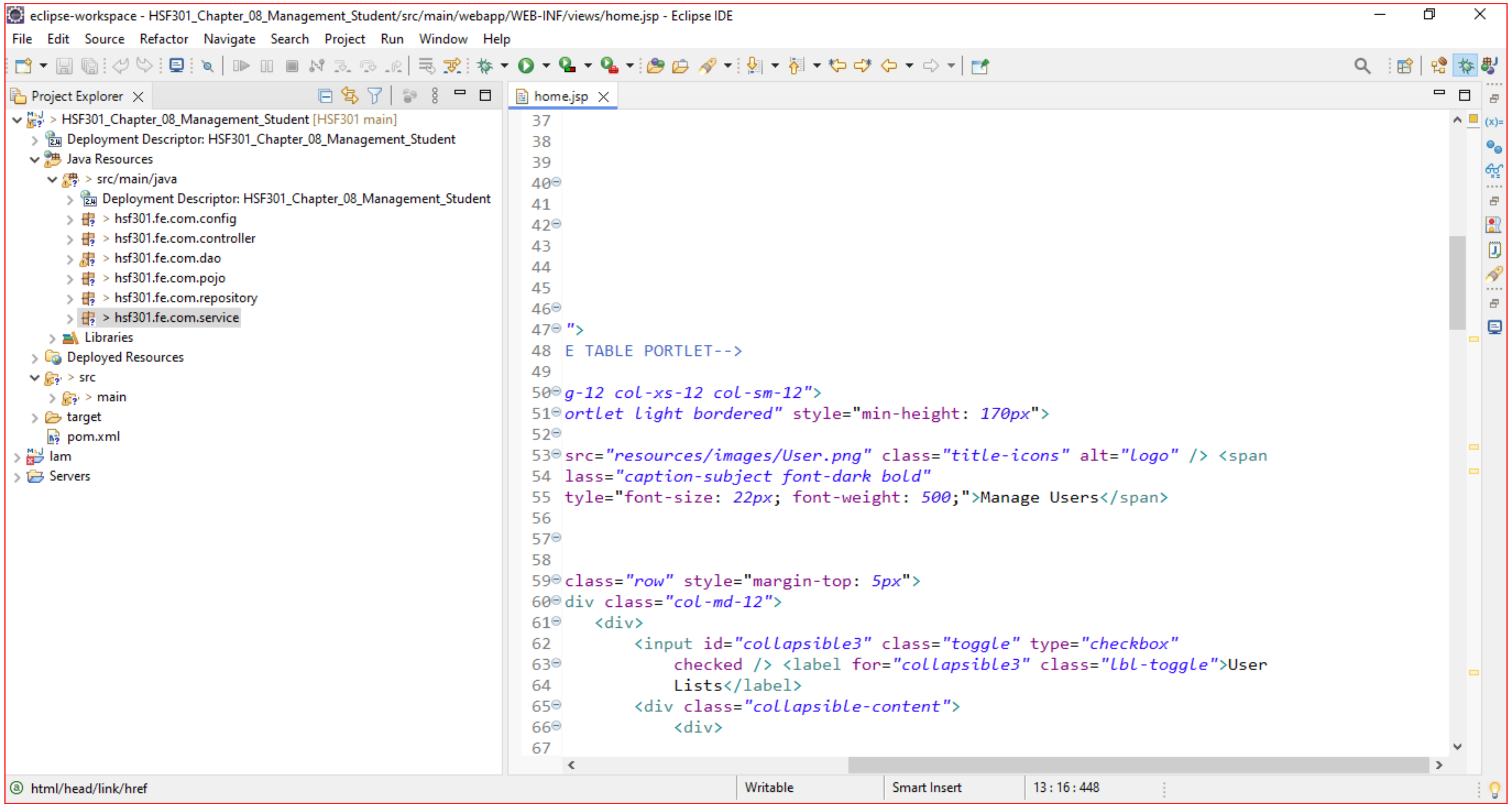
Manage Users

▼ User Lists

Choose	ID	First Name	Last Name	Mark

Key Name	Key Value
ID	
First Name	
Last Name	
Mark	

15. Build the Architecture



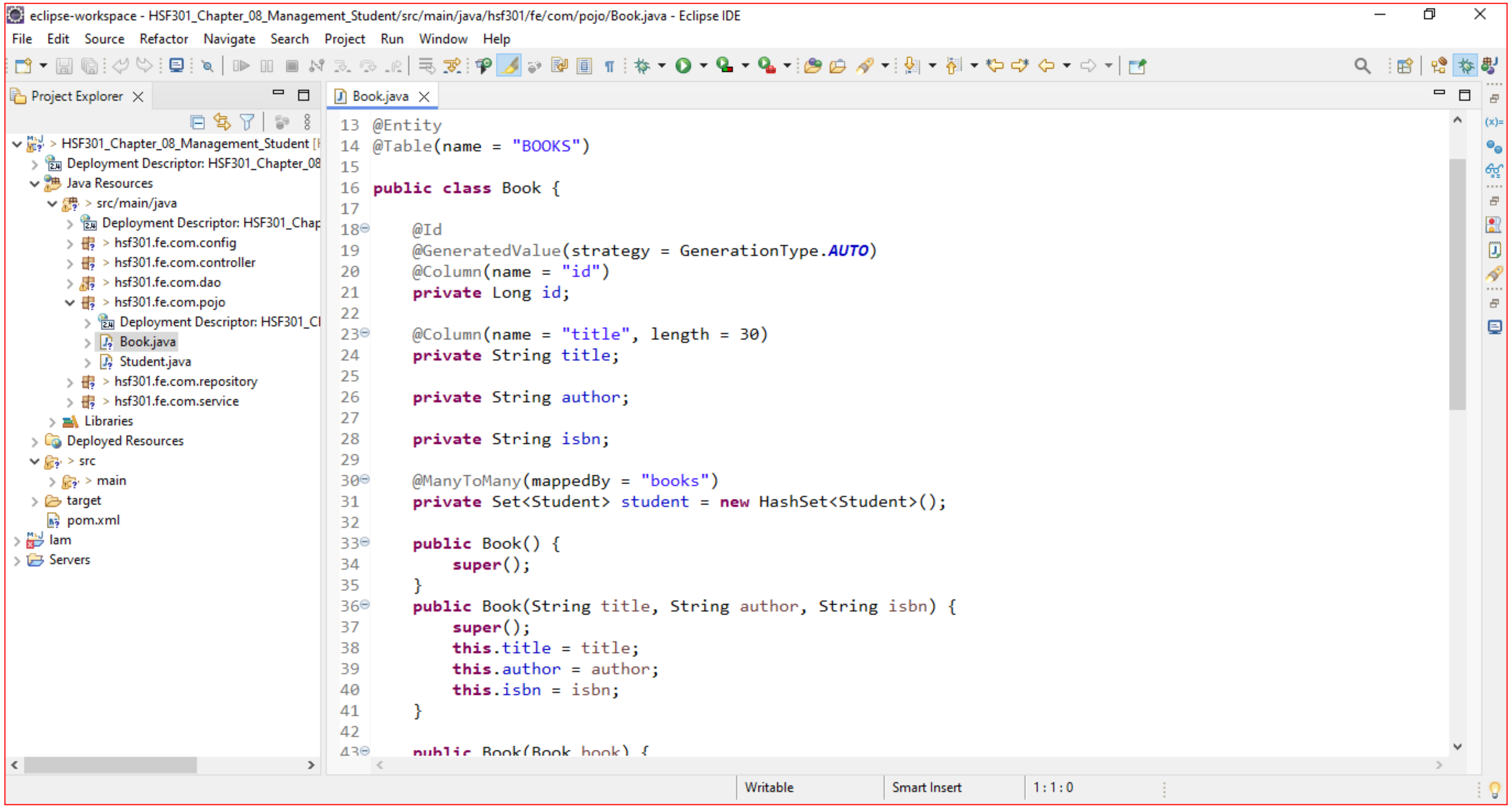
eclipse-workspace - HSF301_Chapter_08_Management_Student/src/main/webapp/WEB-INF/views/home.jsp - Eclipse IDE

```
HSF301_Chapter_08_Management_Student [HSF301 main]
  Deployment Descriptor: HSF301_Chapter_08_Management_Student
  Java Resources
    src/main/java
      Deployment Descriptor: HSF301_Chapter_08_Management_Student
      hsf301.fe.com.config
      hsf301.fe.com.controller
      hsf301.fe.com.dao
      hsf301.fe.com.pojo
      hsf301.fe.com.repository
      hsf301.fe.com.service
    Libraries
  Deployed Resources
  src
    main
  target
  pom.xml
lam
Servers
```

```
">
E TABLE PORTLET-->

g-12 col-xs-12 col-sm-12">
ortlet light bordered" style="min-height: 170px">

src="resources/images/User.png" class="title-icons" alt="logo" /> <span
lass="caption-subject font-dark bold"
tyle="font-size: 22px; font-weight: 500;">Manage Users</span>



class="row" style="margin-top: 5px">
div class="col-md-12">
    <div>
        <input id="collapsible3" class="toggle" type="checkbox"
            checked /> <label for="collapsible3" class="lbl-toggle">User
            Lists</label>
        <div class="collapsible-content">
            <div>
```

html/head/link/href | Writable | Smart Insert | 13 : 16 : 448

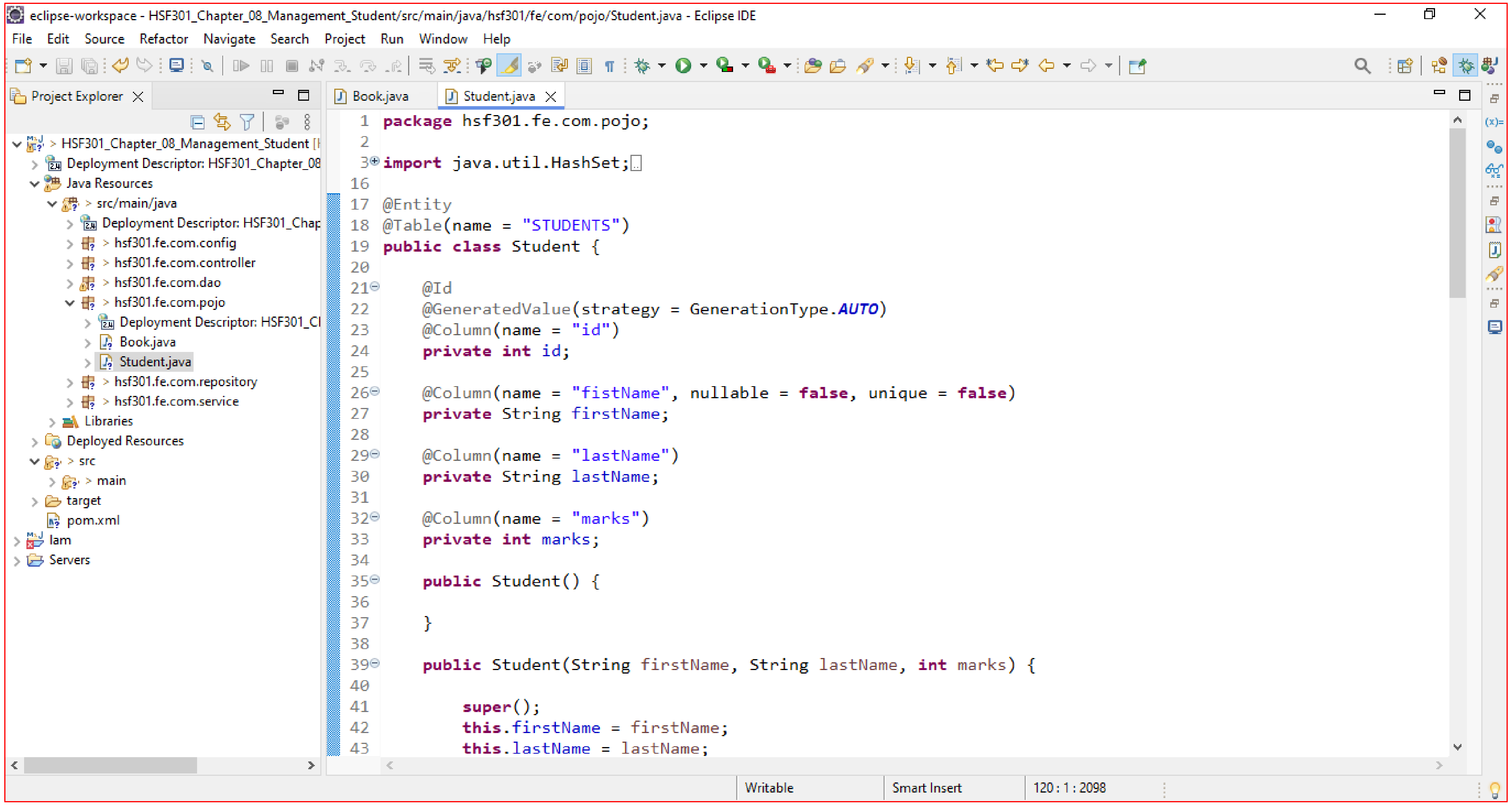
16. Create Books in Pojo



eclipse-workspace - HSF301_Chapter_08_Management_Student/src/main/java/hsf301/fe/com/pojo/Book.java - Eclipse IDE

```java
@Entity
@Table(name = "BOOKS")

public class Book {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id")
    private Long id;

    @Column(name = "title", length = 30)
    private String title;

    private String author;

    private String isbn;

    @ManyToMany(mappedBy = "books")
    private Set<Student> student = new HashSet<Student>();

    public Book() {
        super();
    }
    public Book(String title, String author, String isbn) {
        super();
        this.title = title;
        this.author = author;
        this.isbn = isbn;
    }

    public Book(Book book) {
```

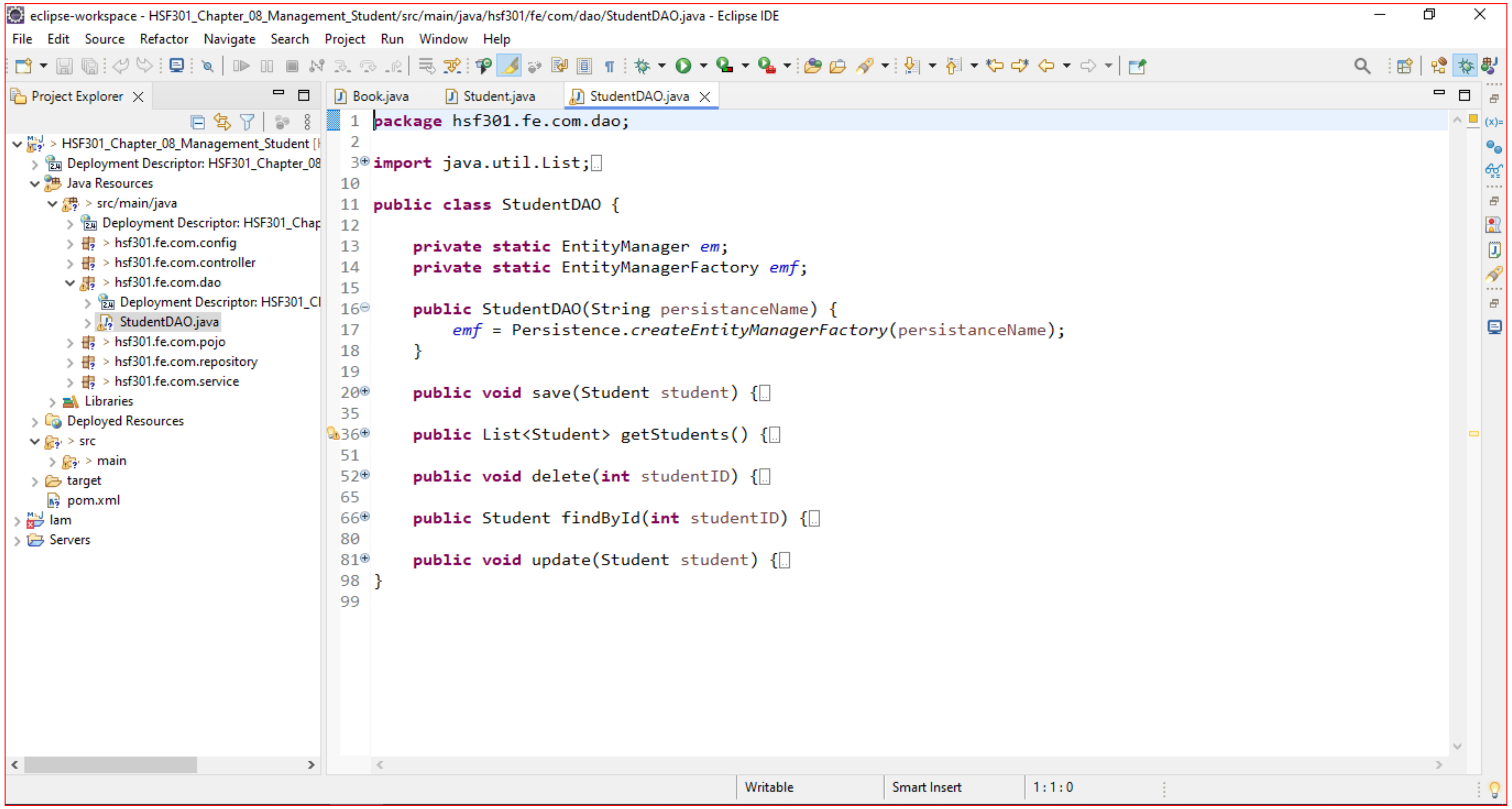
17. Create Students in Pojo



eclipse-workspace - HSF301_Chapter_08_Management_Student/src/main/java/hsf301/fe/com/pojo/Student.java - Eclipse IDE

```java
package hsf301.fe.com.pojo;

import java.util.HashSet;

@Entity
@Table(name = "STUDENTS")
public class Student {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id")
    private int id;

    @Column(name = "fistName", nullable = false, unique = false)
    private String firstName;

    @Column(name = "lastName")
    private String lastName;

    @Column(name = "marks")
    private int marks;

    public Student() {

    }

    public Student(String firstName, String lastName, int marks) {

        super();
        this.firstName = firstName;
        this.lastName = lastName;
```

18. Create StudentDAO



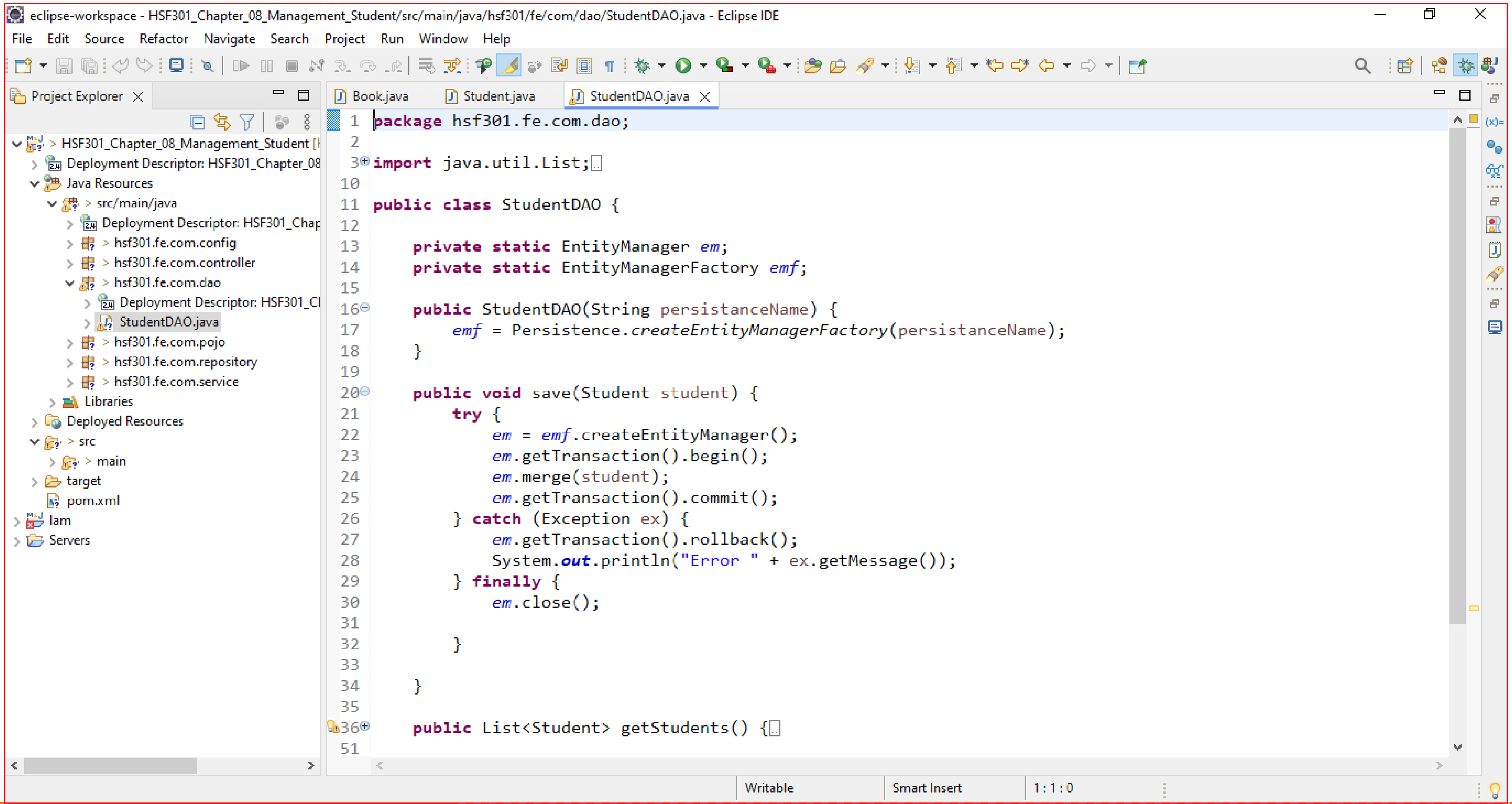
eclipse-workspace - HSF301_Chapter_08_Management_Student/src/main/java/hsf301/fe/com/dao/StudentDAO.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

```java
package hsf301.fe.com.dao;

import java.util.List;

public class StudentDAO {

    private static EntityManager em;
    private static EntityManagerFactory emf;

    public StudentDAO(String persistanceName) {
        emf = Persistence.createEntityManagerFactory(persistanceName);
    }

    public void save(Student student) {

    public List<Student> getStudents() {

    public void delete(int studentID) {

    public Student findById(int studentID) {

    public void update(Student student) {
}
```

FPT Education
FPT UNIVERSITY

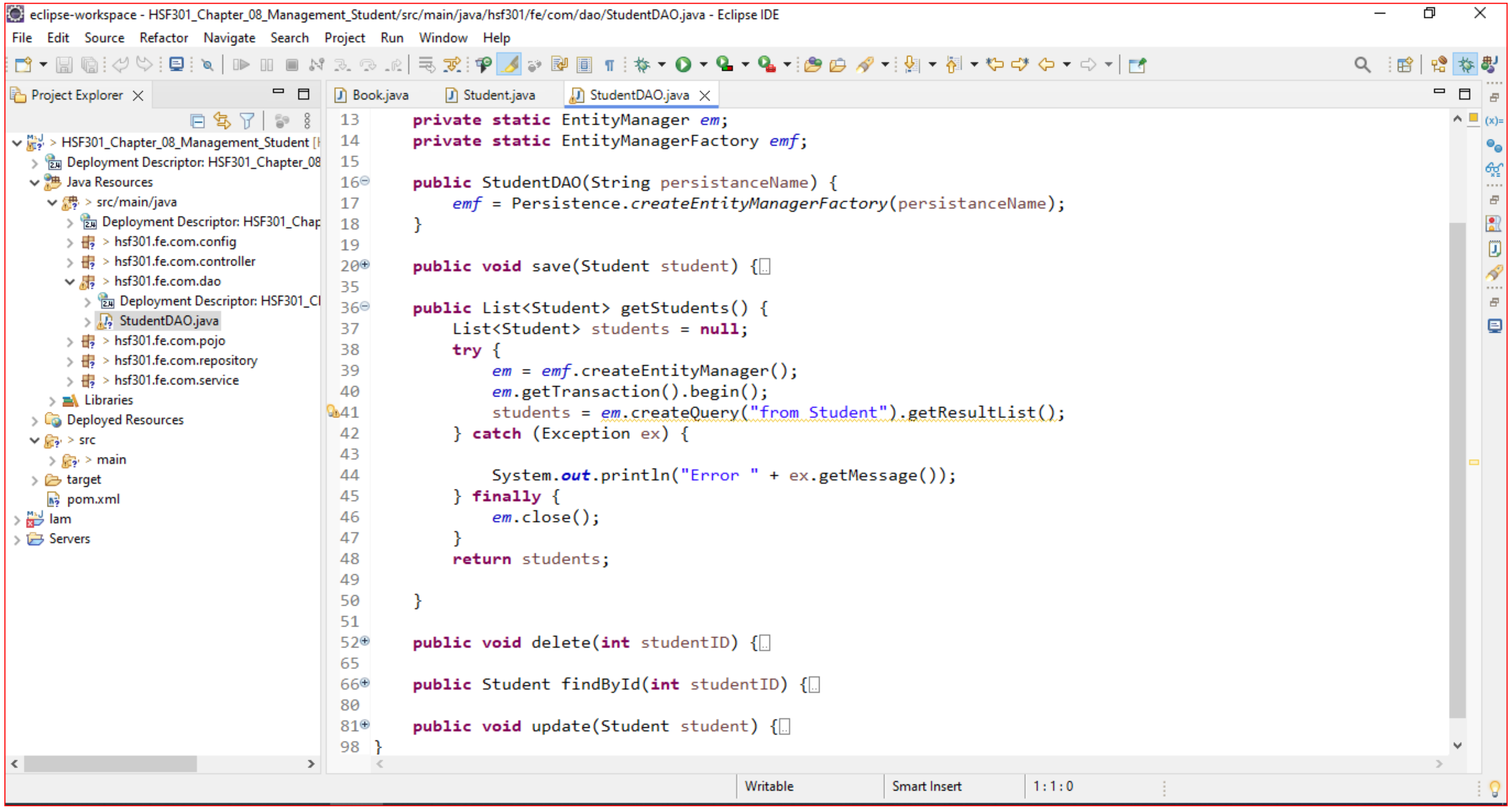
19. Save Student in StudentDAO



eclipse-workspace - HSF301_Chapter_08_Management_Student/src/main/java/hsf301/fe/com/dao/StudentDAO.java - Eclipse IDE

```java
package hsf301.fe.com.dao;

import java.util.List;

public class StudentDAO {

    private static EntityManager em;
    private static EntityManagerFactory emf;

    public StudentDAO(String persistanceName) {
        emf = Persistence.createEntityManagerFactory(persistanceName);
    }

    public void save(Student student) {
        try {
            em = emf.createEntityManager();
            em.getTransaction().begin();
            em.merge(student);
            em.getTransaction().commit();
        } catch (Exception ex) {
            em.getTransaction().rollback();
            System.out.println("Error " + ex.getMessage());
        } finally {
            em.close();

        }

    }

    public List<Student> getStudents() {
```

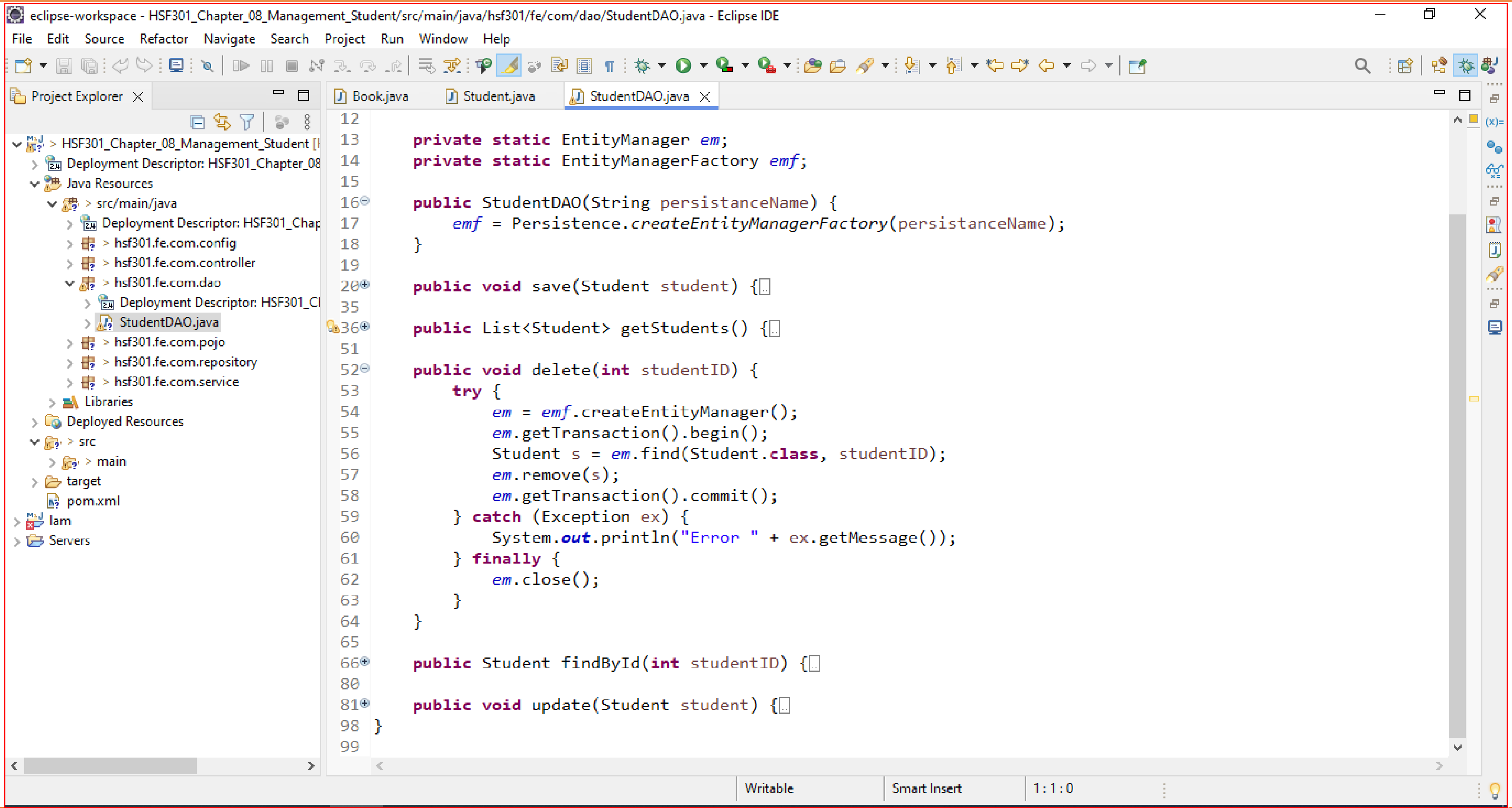
20. Get All Student in StudentDAO



eclipse-workspace - HSF301_Chapter_08_Management_Student/src/main/java/hsf301/fe/com/dao/StudentDAO.java - Eclipse IDE

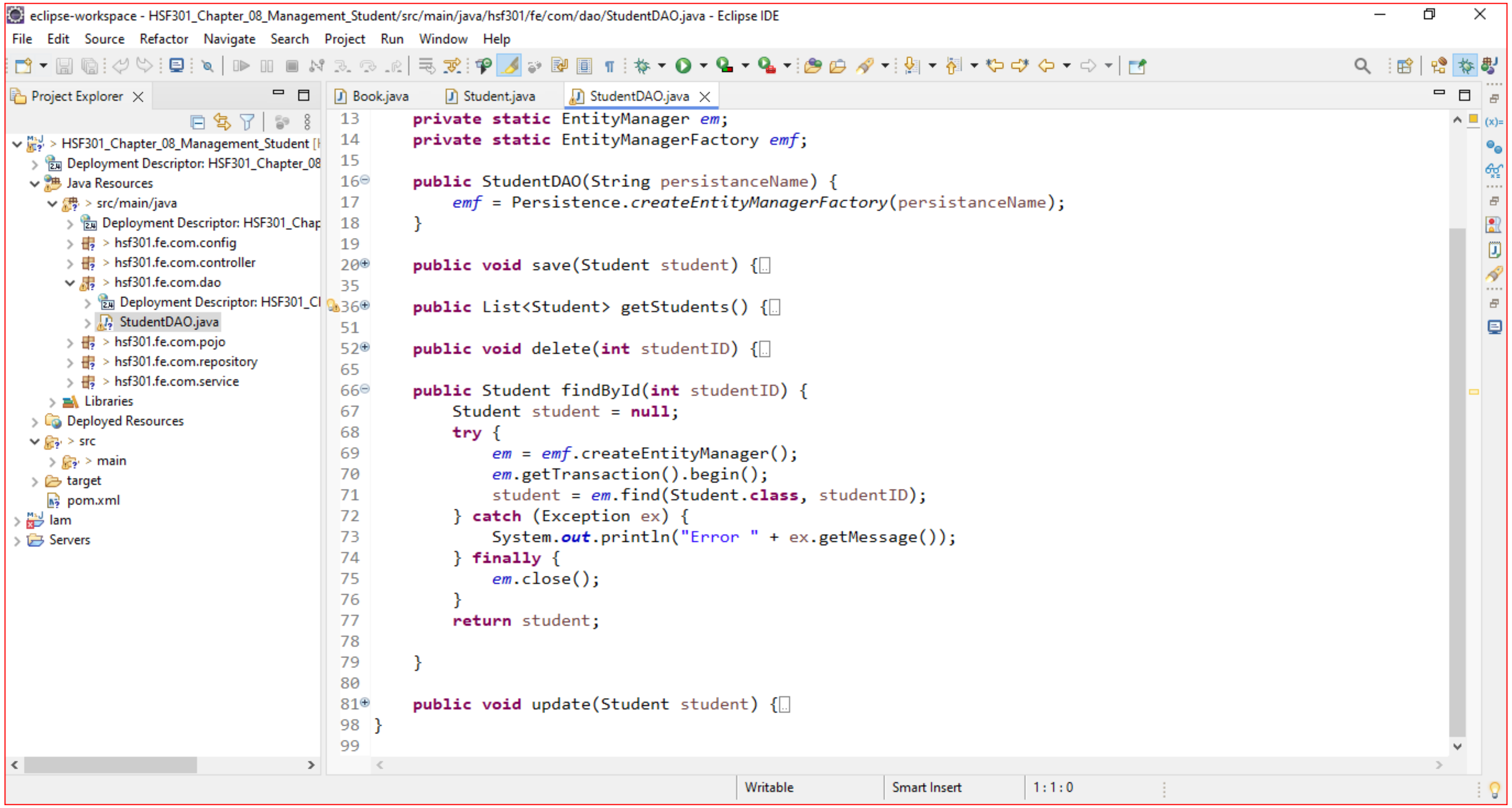
```java
    private static EntityManager em;
    private static EntityManagerFactory emf;

    public StudentDAO(String persistanceName) {
        emf = Persistence.createEntityManagerFactory(persistanceName);
    }

    public void save(Student student) {

    public List<Student> getStudents() {
        List<Student> students = null;
        try {
            em = emf.createEntityManager();
            em.getTransaction().begin();
            students = em.createQuery("from Student").getResultList();
        } catch (Exception ex) {

            System.out.println("Error " + ex.getMessage());
        } finally {
            em.close();
        }
        return students;

    }

    public void delete(int studentID) {

    public Student findById(int studentID) {

    public void update(Student student) {
}
```

21. Delete Student in StudentDAO



```java

    private static EntityManager em;
    private static EntityManagerFactory emf;

    public StudentDAO(String persistanceName) {
        emf = Persistence.createEntityManagerFactory(persistanceName);
    }

    public void save(Student student) {

    public List<Student> getStudents() {

    public void delete(int studentID) {
        try {
            em = emf.createEntityManager();
            em.getTransaction().begin();
            Student s = em.find(Student.class, studentID);
            em.remove(s);
            em.getTransaction().commit();
        } catch (Exception ex) {
            System.out.println("Error " + ex.getMessage());
        } finally {
            em.close();
        }
    }

    public Student findById(int studentID) {

    public void update(Student student) {
}
```

22. Find A Student in StudentDAO

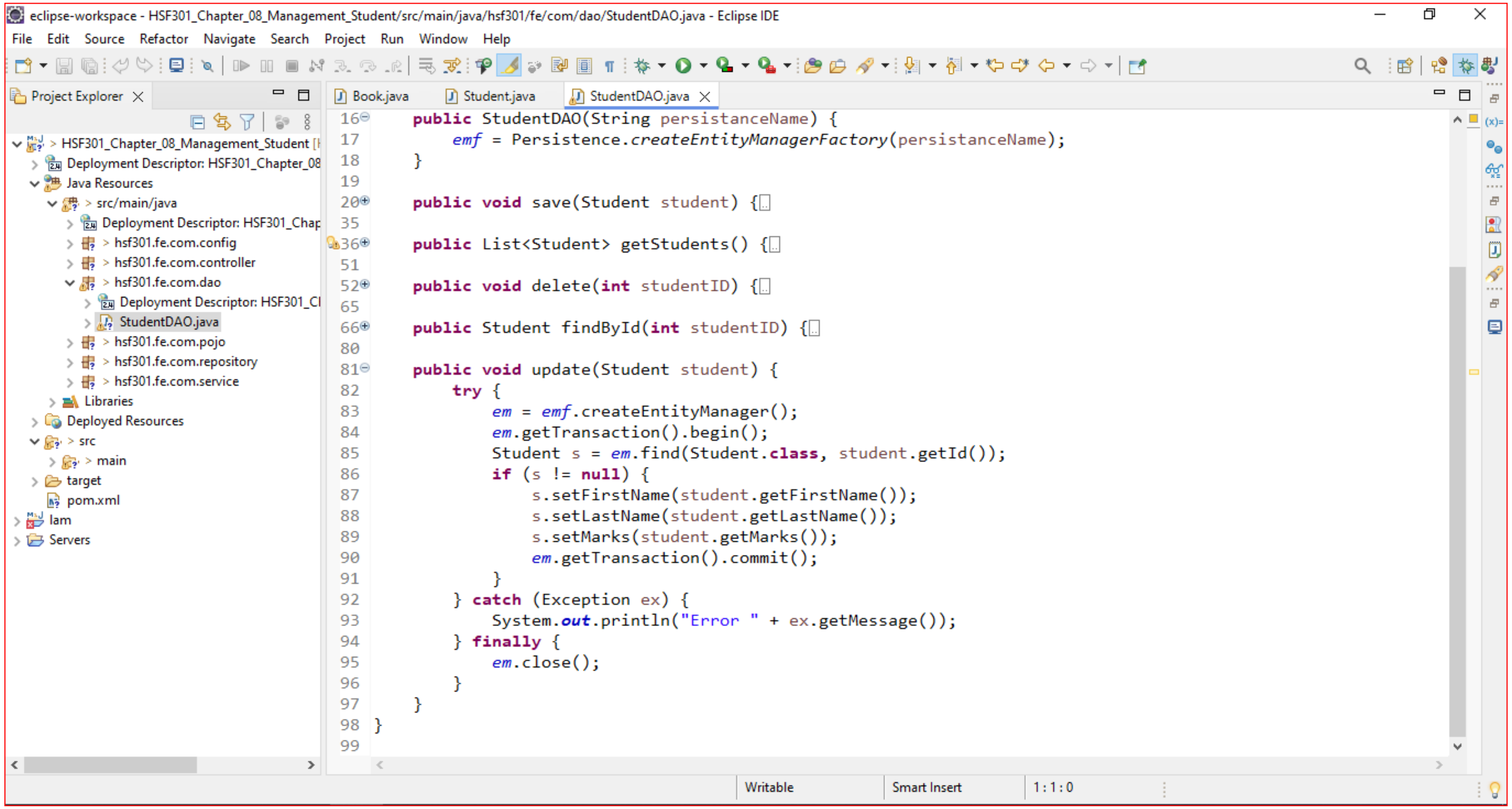


eclipse-workspace - HSF301_Chapter_08_Management_Student/src/main/java/hsf301/fe/com/dao/StudentDAO.java - Eclipse IDE

```java
    private static EntityManager em;
    private static EntityManagerFactory emf;

    public StudentDAO(String persistanceName) {
        emf = Persistence.createEntityManagerFactory(persistanceName);
    }

    public void save(Student student) {

    public List<Student> getStudents() {

    public void delete(int studentID) {

    public Student findById(int studentID) {
        Student student = null;
        try {
            em = emf.createEntityManager();
            em.getTransaction().begin();
            student = em.find(Student.class, studentID);
        } catch (Exception ex) {
            System.out.println("Error " + ex.getMessage());
        } finally {
            em.close();
        }
        return student;

    }

    public void update(Student student) {
}
```

FPT Education
FPT UNIVERSITY

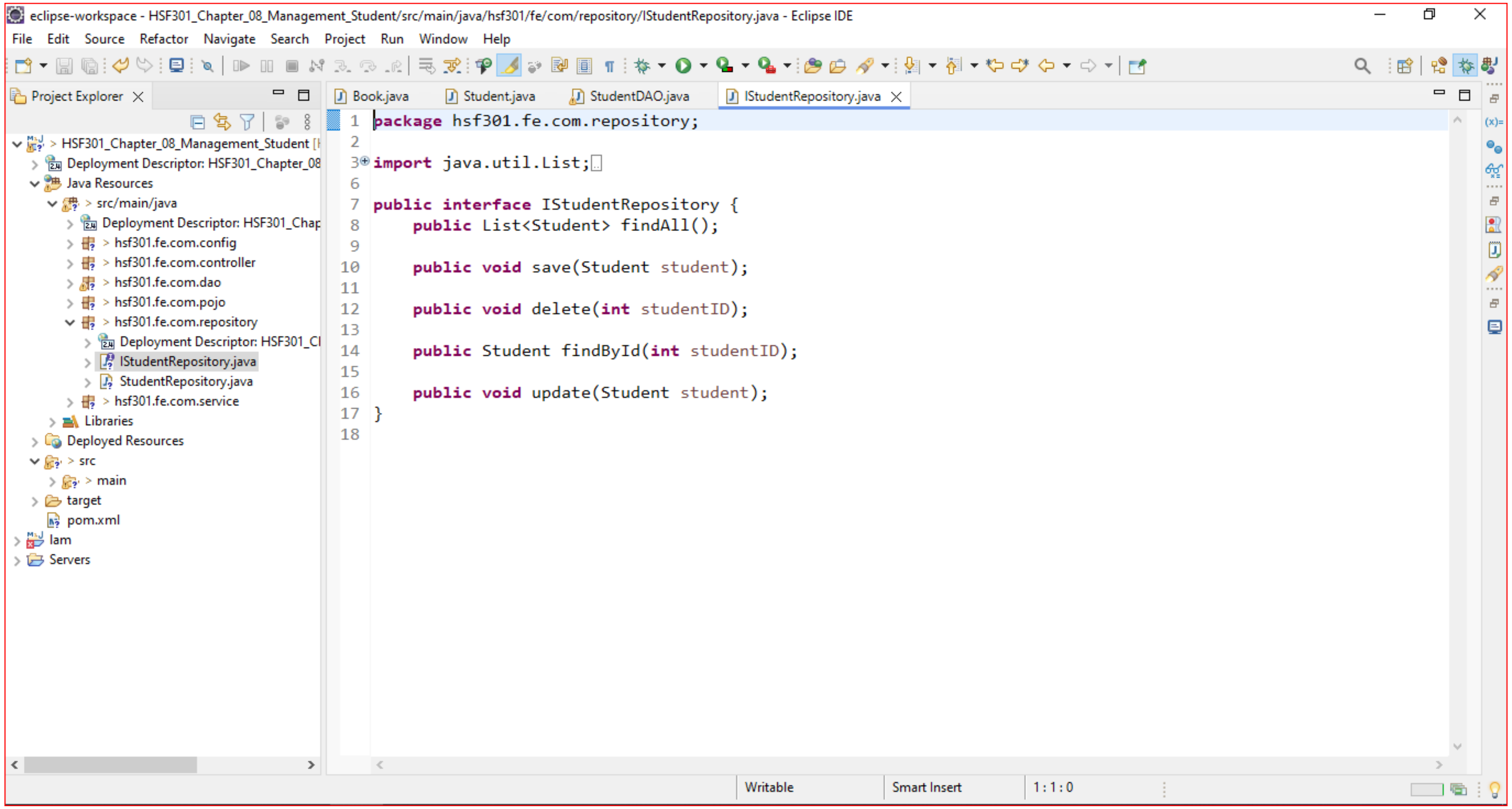
23. Update a Student in StudentDAO



eclipse-workspace - HSF301_Chapter_08_Management_Student/src/main/java/hsf301/fe/com/dao/StudentDAO.java - Eclipse IDE

```java
    public StudentDAO(String persistanceName) {
        emf = Persistence.createEntityManagerFactory(persistanceName);
    }

    public void save(Student student) {

    public List<Student> getStudents() {

    public void delete(int studentID) {

    public Student findById(int studentID) {

    public void update(Student student) {
        try {
            em = emf.createEntityManager();
            em.getTransaction().begin();
            Student s = em.find(Student.class, student.getId());
            if (s != null) {
                s.setFirstName(student.getFirstName());
                s.setLastName(student.getLastName());
                s.setMarks(student.getMarks());
                em.getTransaction().commit();
            }
        } catch (Exception ex) {
            System.out.println("Error " + ex.getMessage());
        } finally {
            em.close();
        }
    }
}
```

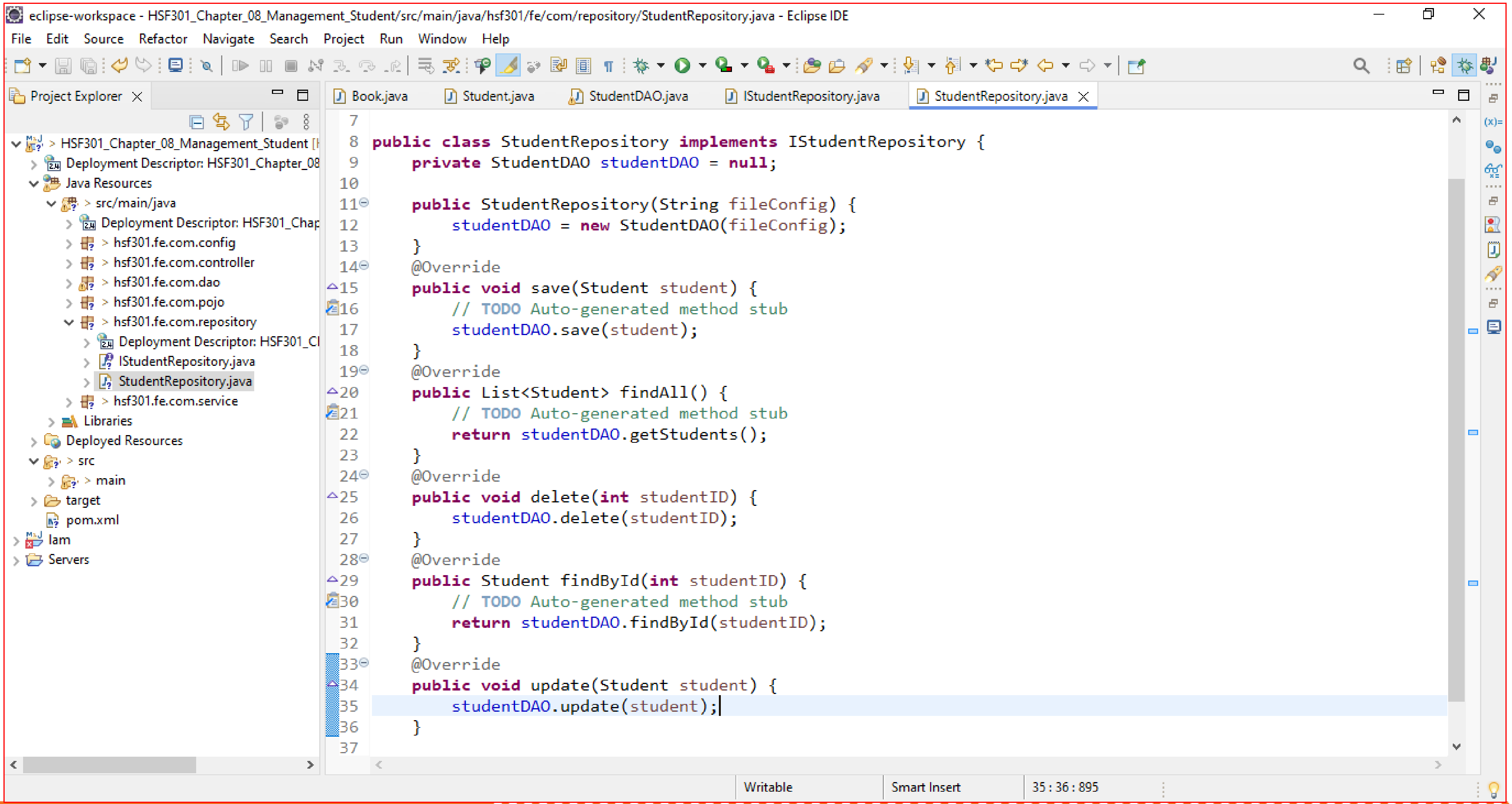
24. Create IStudentRepository



eclipse-workspace - HSF301_Chapter_08_Management_Student/src/main/java/hsf301/fe/com/repository/IStudentRepository.java - Eclipse IDE

```java
package hsf301.fe.com.repository;

import java.util.List;

public interface IStudentRepository {
    public List<Student> findAll();

    public void save(Student student);

    public void delete(int studentID);

    public Student findById(int studentID);

    public void update(Student student);
}
```

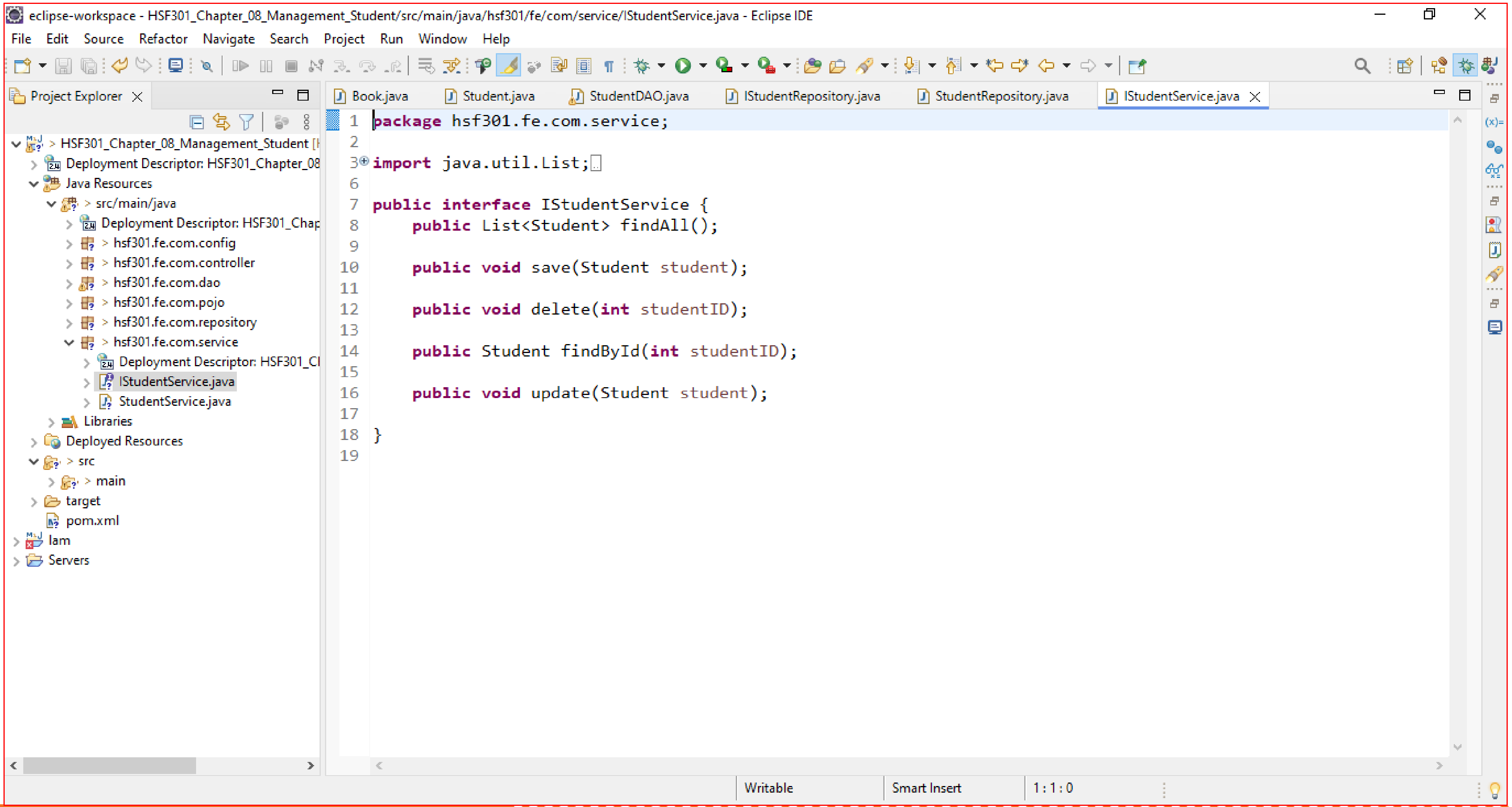
25. Create StudentRepository


eclipse-workspace - HSF301_Chapter_08_Management_Student/src/main/java/hsf301/fe/com/repository/StudentRepository.java - Eclipse IDE

```java

public class StudentRepository implements IStudentRepository {
    private StudentDAO studentDAO = null;

    public StudentRepository(String fileConfig) {
        studentDAO = new StudentDAO(fileConfig);
    }
    @Override
    public void save(Student student) {
        // TODO Auto-generated method stub
        studentDAO.save(student);
    }
    @Override
    public List<Student> findAll() {
        // TODO Auto-generated method stub
        return studentDAO.getStudents();
    }
    @Override
    public void delete(int studentID) {
        studentDAO.delete(studentID);
    }
    @Override
    public Student findById(int studentID) {
        // TODO Auto-generated method stub
        return studentDAO.findById(studentID);
    }
    @Override
    public void update(Student student) {
        studentDAO.update(student);
    }

```

FPT Education
FPT UNIVERSITY

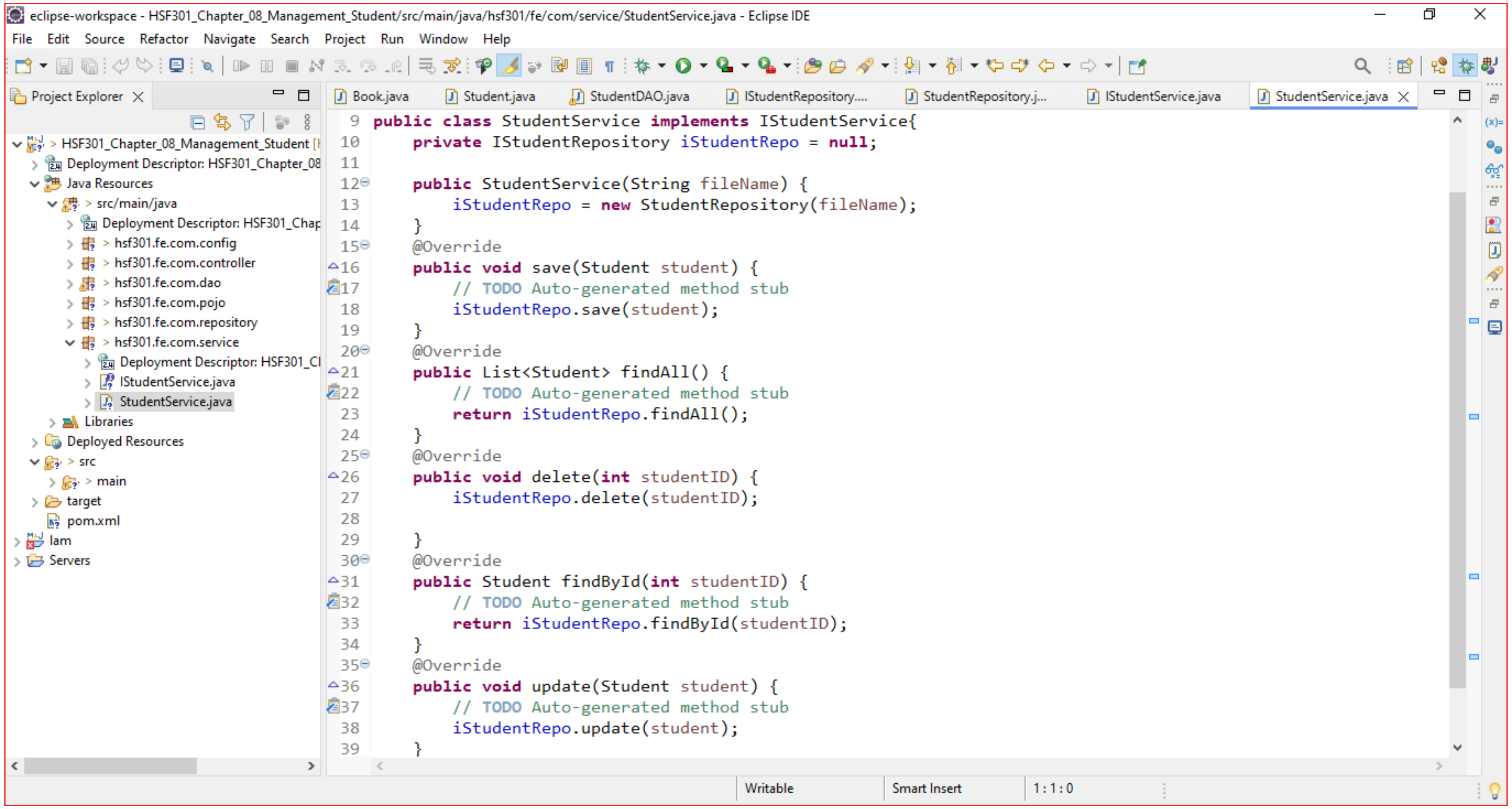
26. Create IStudentService



eclipse-workspace - HSF301_Chapter_08_Management_Student/src/main/java/hsf301/fe/com/service/IStudentService.java - Eclipse IDE

```java
package hsf301.fe.com.service;

import java.util.List;

public interface IStudentService {
    public List<Student> findAll();

    public void save(Student student);

    public void delete(int studentID);

    public Student findById(int studentID);

    public void update(Student student);

}
```

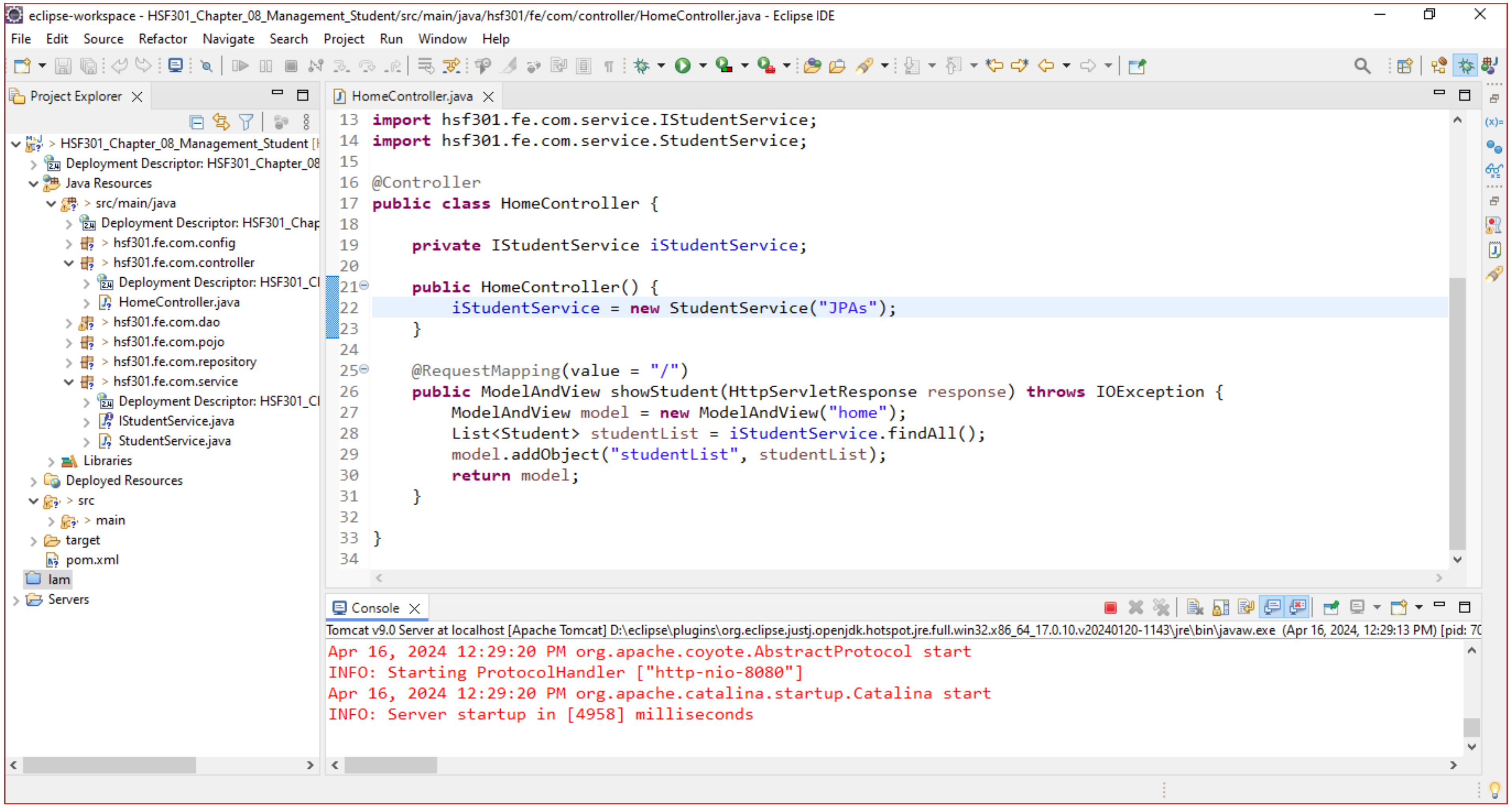
27. Create StudentService



```java
public class StudentService implements IStudentService{
    private IStudentRepository iStudentRepo = null;

    public StudentService(String fileName) {
        iStudentRepo = new StudentRepository(fileName);
    }
    @Override
    public void save(Student student) {
        // TODO Auto-generated method stub
        iStudentRepo.save(student);
    }
    @Override
    public List<Student> findAll() {
        // TODO Auto-generated method stub
        return iStudentRepo.findAll();
    }
    @Override
    public void delete(int studentID) {
        iStudentRepo.delete(studentID);

    }
    @Override
    public Student findById(int studentID) {
        // TODO Auto-generated method stub
        return iStudentRepo.findById(studentID);
    }
    @Override
    public void update(Student student) {
        // TODO Auto-generated method stub
        iStudentRepo.update(student);
    }
```

FPT Education
FPT UNIVERSITY

28. Edit HomeController.java



eclipse-workspace - HSF301_Chapter_08_Management_Student/src/main/java/hsf301/fe/com/controller/HomeController.java - Eclipse IDE

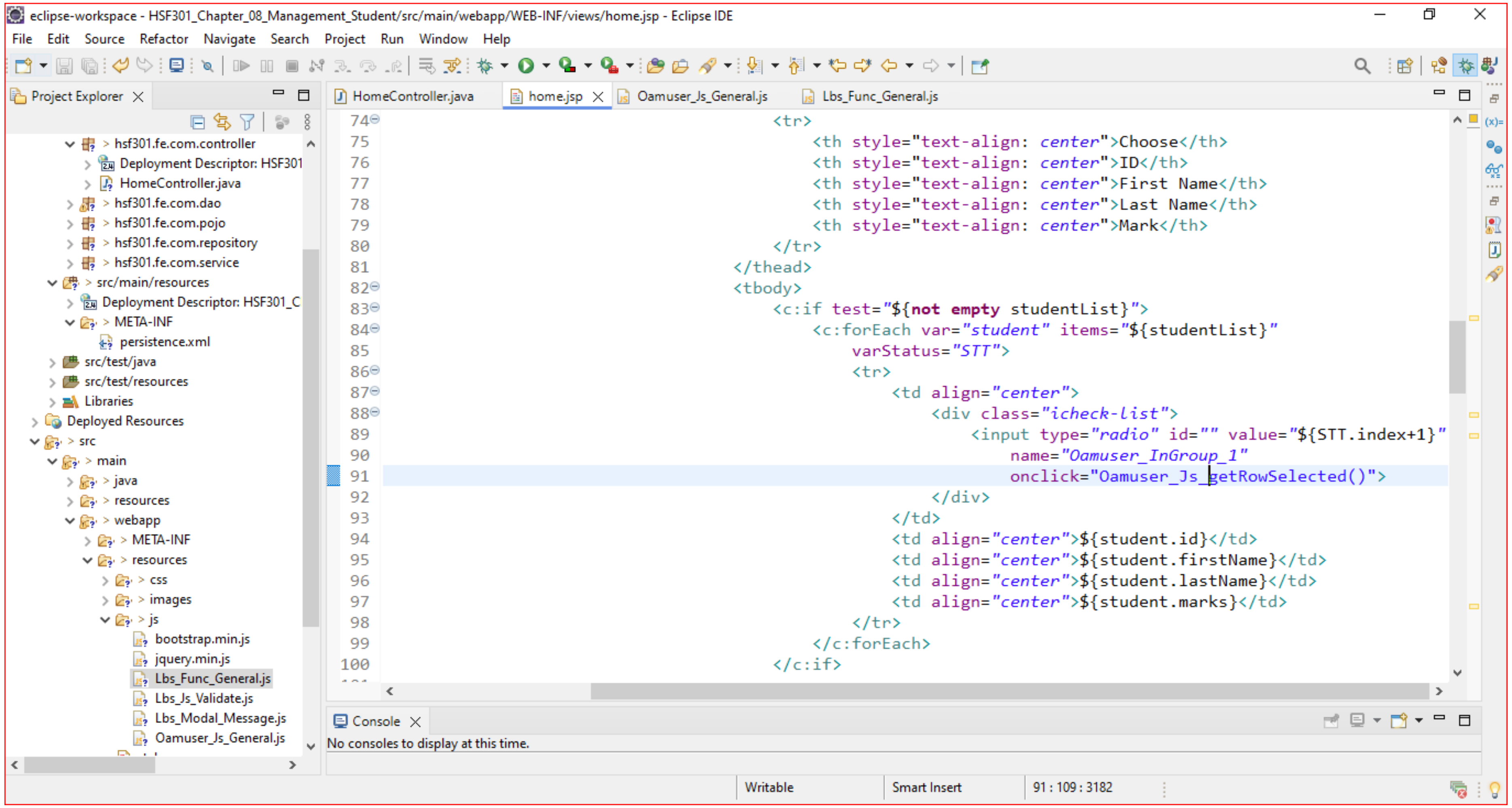
```java
import hsf301.fe.com.service.IStudentService;
import hsf301.fe.com.service.StudentService;

@Controller
public class HomeController {

    private IStudentService iStudentService;

    public HomeController() {
        iStudentService = new StudentService("JPAs");
    }

    @RequestMapping(value = "/")
    public ModelAndView showStudent(HttpServletResponse response) throws IOException {
        ModelAndView model = new ModelAndView("home");
        List<Student> studentList = iStudentService.findAll();
        model.addObject("studentList", studentList);
        return model;
    }

}
```

Tomcat v9.0 Server at localhost [Apache Tomcat] D:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe (Apr 16, 2024, 12:29:13 PM) [pid: 70

```
Apr 16, 2024 12:29:20 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-nio-8080"]
Apr 16, 2024 12:29:20 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in [4958] milliseconds
```

FPT UNIVERSITY

28. Edit Home.jsp



eclipse-workspace - HSF301_Chapter_08_Management_Student/src/main/webapp/WEB-INF/views/home.jsp - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

- hsf301.fe.com.controller
  - Deployment Descriptor: HSF301
  - HomeController.java
- hsf301.fe.com.dao
- hsf301.fe.com.pojo
- hsf301.fe.com.repository
- hsf301.fe.com.service
- src/main/resources
  - Deployment Descriptor: HSF301_C
  - META-INF
    - persistence.xml
- src/test/java
- src/test/resources
- Libraries
- Deployed Resources
- src
  - main
    - java
    - resources
    - webapp
      - META-INF
      - resources
        - css
        - images
        - js
          - bootstrap.min.js
          - jquery.min.js
          - Lbs_Func_General.js
          - Lbs_Js_Validate.js
          - Lbs_Modal_Message.js
          - Oamuser_Js_General.js

HomeController.java | home.jsp | Oamuser_Js_General.js | Lbs_Func_General.js

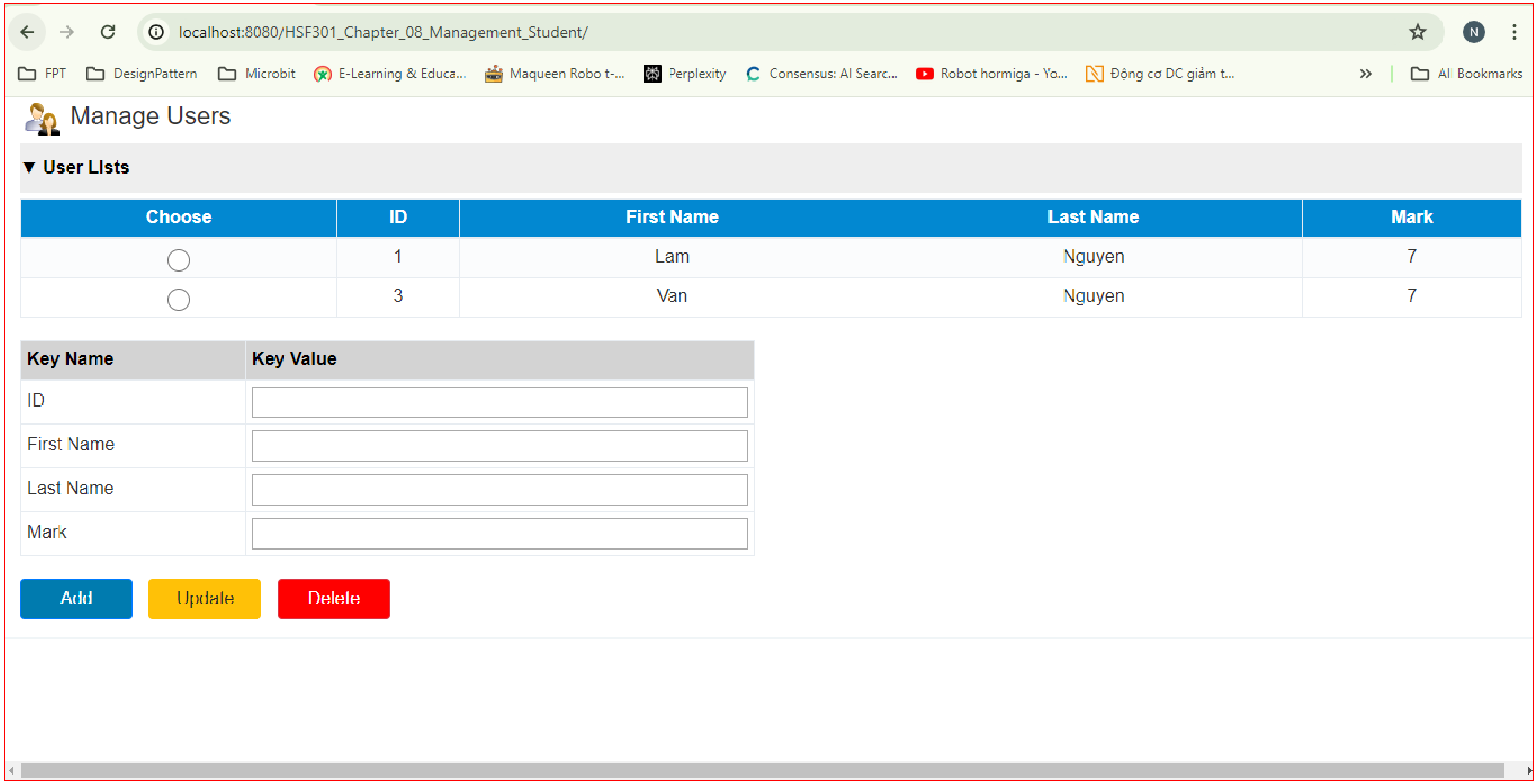
```
    <tr>
      <th style="text-align: center">Choose</th>
      <th style="text-align: center">ID</th>
      <th style="text-align: center">First Name</th>
      <th style="text-align: center">Last Name</th>
      <th style="text-align: center">Mark</th>
    </tr>
  </thead>
  <tbody>
    <c:if test="${not empty studentList}">
      <c:forEach var="student" items="${studentList}"
        varStatus="STT">
        <tr>
          <td align="center">
            <div class="icheck-list">
              <input type="radio" id="" value="${STT.index+1}"
                name="Oamuser_InGroup_1"
                onclick="Oamuser_Js_getRowSelected()">
            </div>
          </td>
          <td align="center">${student.id}</td>
          <td align="center">${student.firstName}</td>
          <td align="center">${student.lastName}</td>
          <td align="center">${student.marks}</td>
        </tr>
      </c:forEach>
    </c:if>
```

Console

No consoles to display at this time.

Writable | Smart Insert | 91 : 109 : 3182

19. Result



localhost:8080/HSF301_Chapter_08_Management_Student/

Manage Users

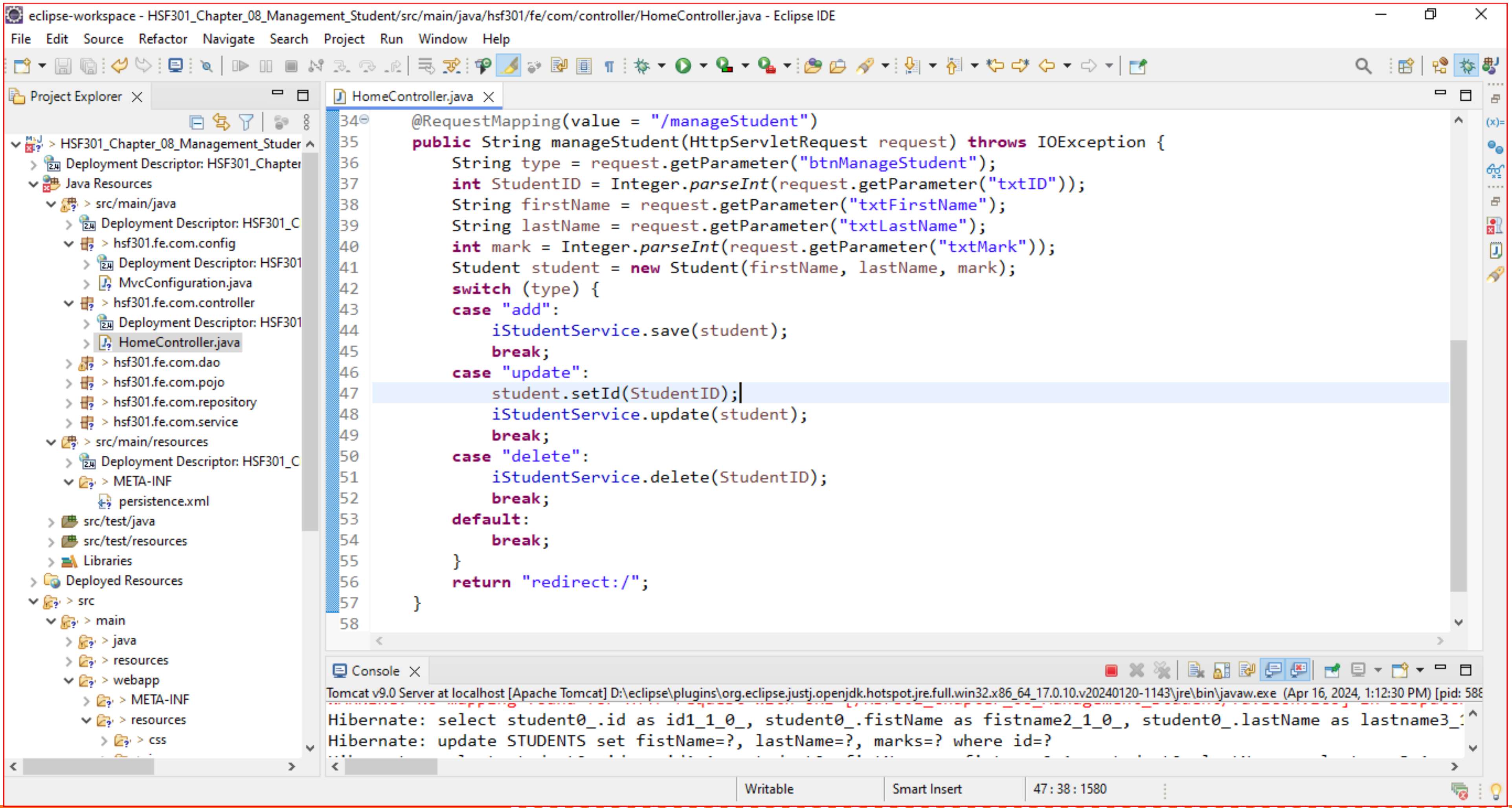
▼ User Lists

Choose	ID	First Name	Last Name	Mark
○	1	Lam	Nguyen	7
○	3	Van	Nguyen	7

Key Name	Key Value
ID	
First Name	
Last Name	
Mark	

FPT Education
FPT UNIVERSITY

20. Edit the HomeController.java

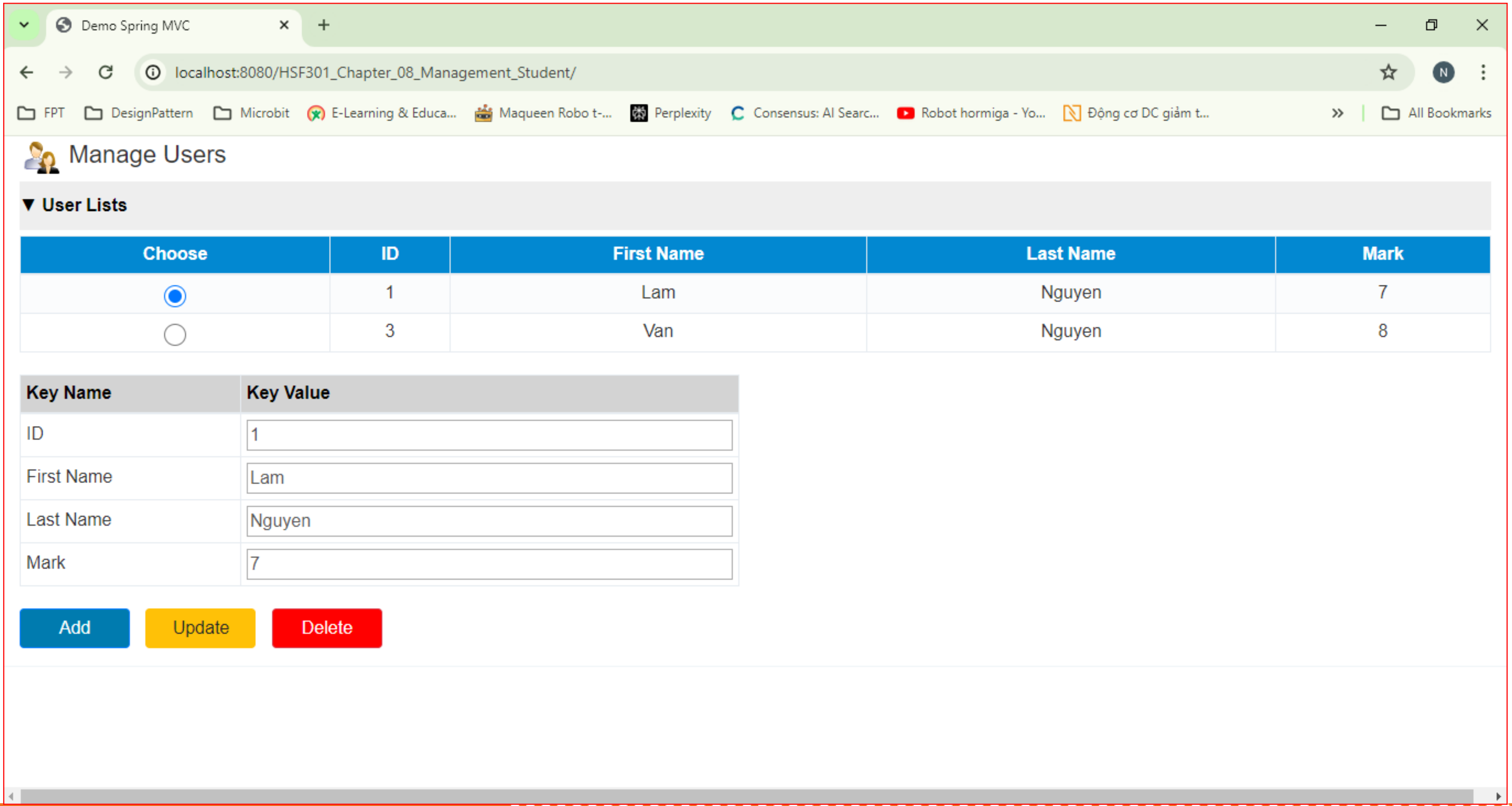

eclipse-workspace - HSF301_Chapter_08_Management_Student/src/main/java/hsf301/fe/com/controller/HomeController.java - Eclipse IDE

```java
	@RequestMapping(value = "/manageStudent")
	public String manageStudent(HttpServletRequest request) throws IOException {
		String type = request.getParameter("btnManageStudent");
		int StudentID = Integer.parseInt(request.getParameter("txtID"));
		String firstName = request.getParameter("txtFirstName");
		String lastName = request.getParameter("txtLastName");
		int mark = Integer.parseInt(request.getParameter("txtMark"));
		Student student = new Student(firstName, lastName, mark);
		switch (type) {
		case "add":
			iStudentService.save(student);
			break;
		case "update":
			student.setId(StudentID);
			iStudentService.update(student);
			break;
		case "delete":
			iStudentService.delete(StudentID);
			break;
		default:
			break;
		}
		return "redirect:/";
	}
```

```
Tomcat v9.0 Server at localhost [Apache Tomcat] D:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe  (Apr 16, 2024, 1:12:30 PM) [pid: 588
Hibernate: select student0_.id as id1_1_0_, student0_.fistName as fistname2_1_0_, student0_.lastName as lastname3_1
Hibernate: update STUDENTS set fistName=?, lastName=?, marks=? where id=?
```

FPT Education
FPT UNIVERSITY

21. Result



Manage Users

▼ User Lists

Choose	ID	First Name	Last Name	Mark
●	1	Lam	Nguyen	7
○	3	Van	Nguyen	8

Key Name	Key Value
ID	1
First Name	Lam
Last Name	Nguyen
Mark	7

Summary

The concepts are introduced:

- Create a new Maven project in Eclipse IDE
- Add the necessary dependencies for Spring MVC and data access
- Create the Model - Define the entity classes that represent domain objects
- Define the data access object (DAO) interfaces and their implementations for interacting with the database.
- Create the Controller
- Set Up the Views
- Create the views using JSP, Thymeleaf, or another templating engine
- Implement the Business Logic
- Testing
- Run the Application