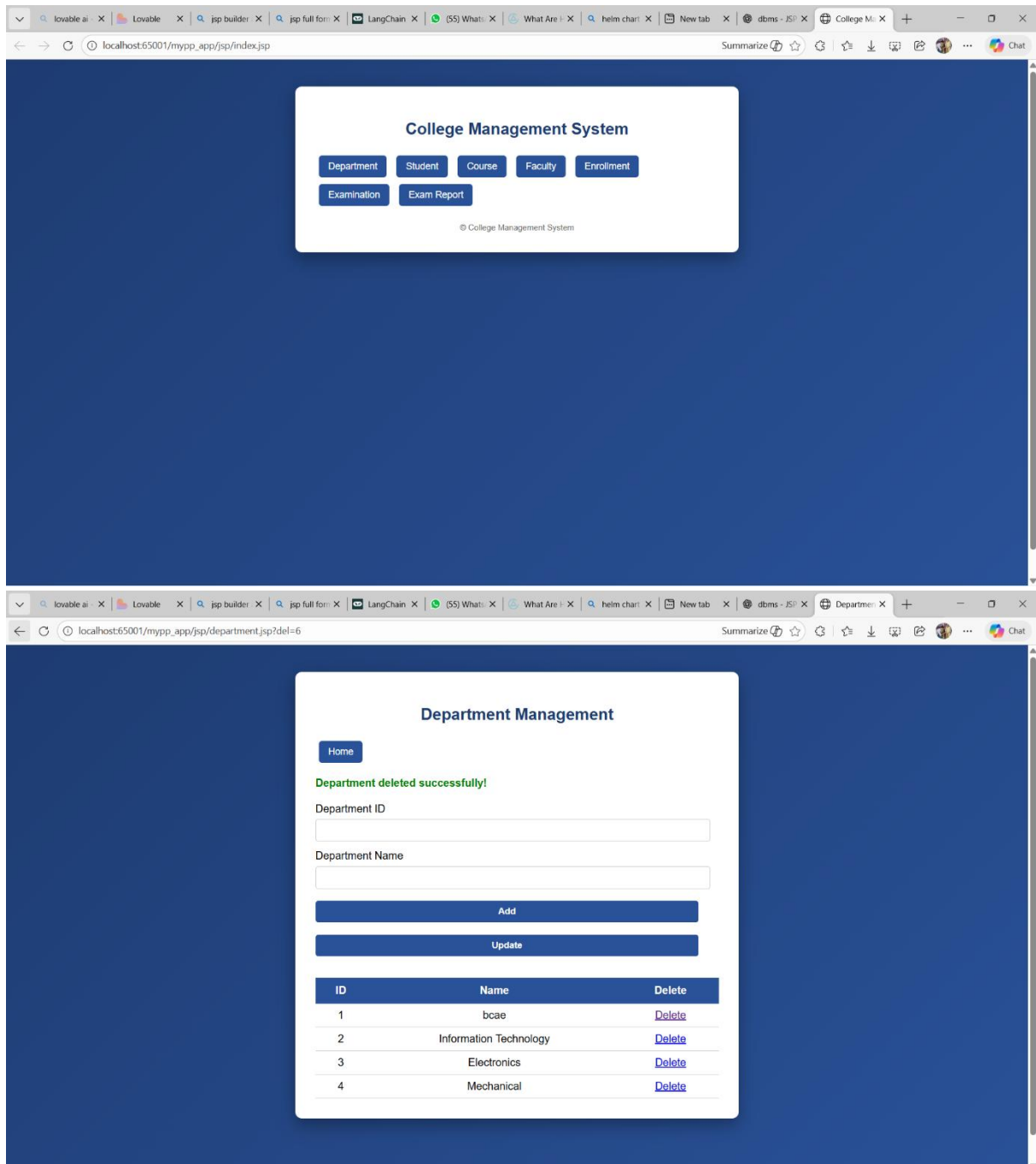DBMS PROJECT→

MADE BY -->   1. Mohammad Maksood(202401100600113)

               2.Parth Pathak(202401100600125)

Github Repo Link (of this Project)→ parth1234560/dbms_oracle_end_sem

1. Frontend→

Course

Home

ID

Name

Add

Update

ID	Name	Delete
101	DBMS	Delete
102	Operating Systems	Delete
103	Data Structures	Delete

Faculty

Home

Faculty ID

Name

Department ID

Course ID

Add Faculty
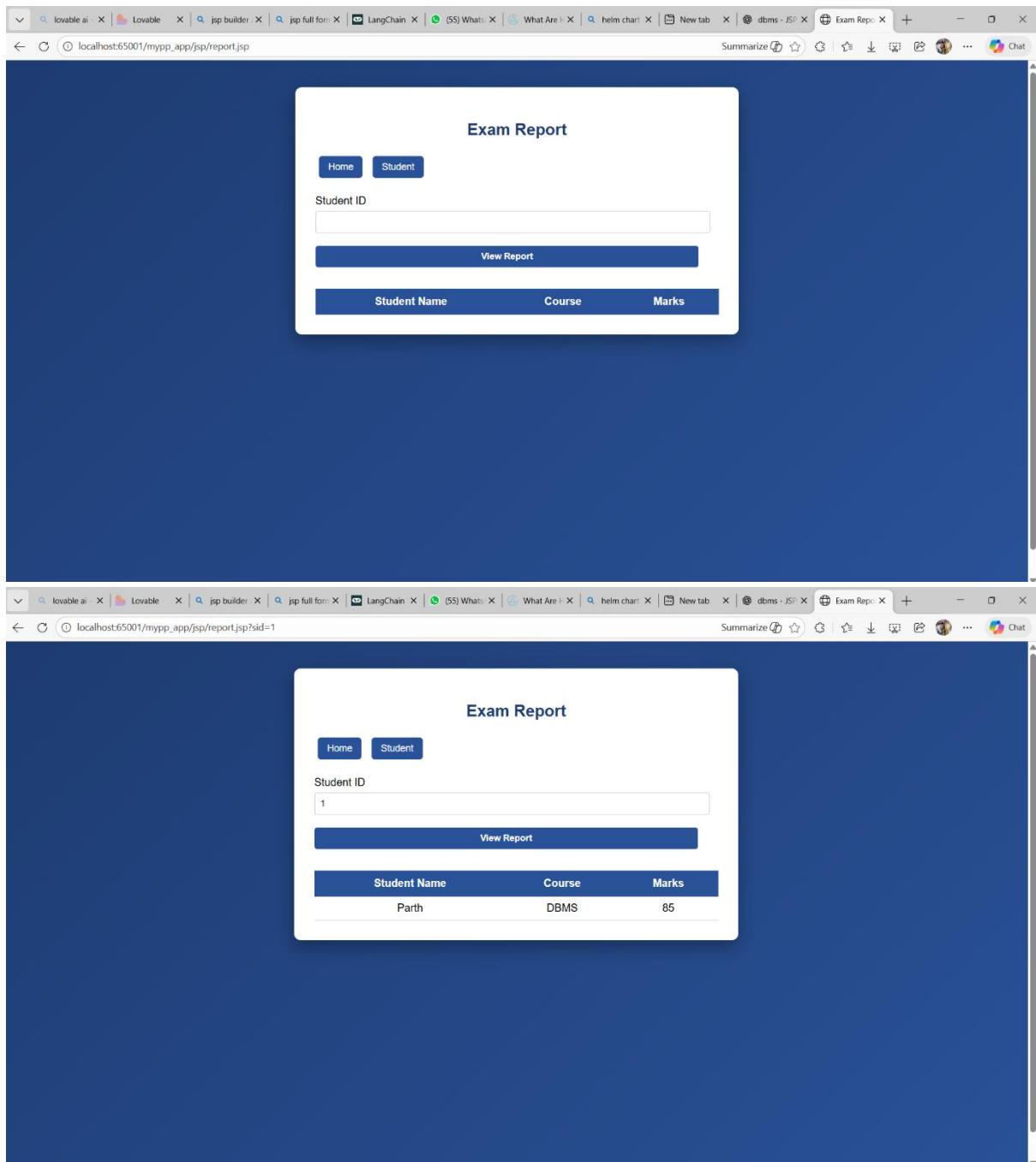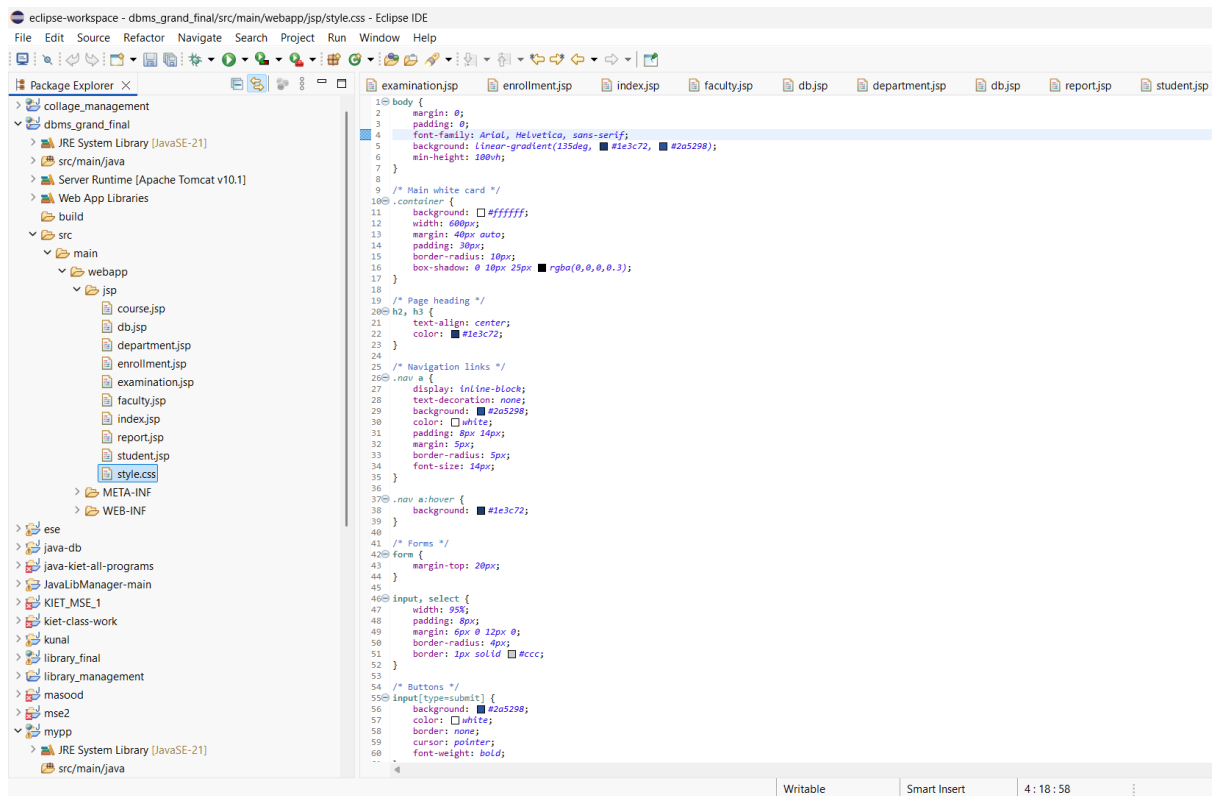
Enrollment

Home

Enroll ID

Student ID

Course ID

Enroll

ID	Student	Course	Delete
5001	1	101	Delete
5002	2	101	Delete
5003	3	102	Delete

Examination

Home

Exam ID

Marks

Course ID

Student ID

Add

Update

ID	Marks	Delete
9001	85	Delete
9002	78	Delete
9003	89	Delete

Frontend Code→

File   Edit   Source   Refactor   Navigate   Search   Project   Run   Window   Help

Package Explorer

- collage_management
- dbms_grand_final
  - JRE System Library [JavaSE-21]
  - src/main/java
  - Server Runtime [Apache Tomcat v10.1]
  - Web App Libraries
  - build
  - src
    - main
      - webapp
        - jsp
          - course.jsp
          - db.jsp
          - department.jsp
          - enrollment.jsp
          - examination.jsp
          - faculty.jsp
          - index.jsp
          - report.jsp
          - student.jsp
          - style.css
        - META-INF
        - WEB-INF
- ese
- java-db
- java-kiet-all-programs
- JavaLibManager-main
- KIET_MSE_1
- kiet-class-work
- kunal
- library_final
- library_management
- masood
- mse2
- mypp
  - JRE System Library [JavaSE-21]
  - src/main/java

examination.jsp   enrollment.jsp   index.jsp   faculty.jsp   db.jsp   department.jsp   db.jsp   report.jsp   student.jsp

```css
1  body {
2      margin: 0;
3      padding: 0;
4      font-family: Arial, Helvetica, sans-serif;
5      background: linear-gradient(135deg, #1e3c72, #2a5298);
6      min-height: 100vh;
7  }
8
9  /* Main white card */
10 .container {
11     background: #ffffff;
12     width: 600px;
13     margin: 40px auto;
14     padding: 30px;
15     border-radius: 10px;
16     box-shadow: 0 10px 25px rgba(0,0,0,0.3);
17 }
18
19 /* Page heading */
20 h2, h3 {
21     text-align: center;
22     color: #1e3c72;
23 }
24
25 /* Navigation links */
26 .nav a {
27     display: inline-block;
28     text-decoration: none;
29     background: #2a5298;
30     color: white;
31     padding: 8px 14px;
32     margin: 5px;
33     border-radius: 5px;
34     font-size: 14px;
35 }
36
37 .nav a:hover {
38     background: #1e3c72;
39 }
40
41 /* Forms */
42 form {
43     margin-top: 20px;
44 }
45
46 input, select {
47     width: 95%;
48     padding: 8px;
49     margin: 6px 0 12px 0;
50     border-radius: 4px;
51     border: 1px solid #ccc;
52 }
53
54 /* Buttons */
55 input[type=submit] {
56     background: #2a5298;
57     color: white;
58     border: none;
59     cursor: pointer;
60     font-weight: bold;
```

Writable   Smart Insert   4 : 18 : 58

Backend Code→

```html
1   <!DOCTYPE html>
2   <html>
3   <head>
4       <title>College Management</title>
5       <link rel="stylesheet" href="style.css">
6   </head>
7
8   <body>
9
10  <div class="container">
11      <h2>College Management System</h2>
12
13      <div class="nav">
14          <a href="department.jsp">Department</a>
15          <a href="student.jsp">Student</a>
16          <a href="course.jsp">Course</a>
17          <a href="faculty.jsp">Faculty</a>
18          <a href="enrollment.jsp">Enrollment</a>
19          <a href="examination.jsp">Examination</a>
20          <a href="report.jsp">Exam Report</a>
21      </div>
22
23      <div class="footer">
24          © College Management System
25      </div>
26  </div>
27
28  </body>
29  </html>
30
```

Writable   Smart Insert   24 : 35 : 625

```jsp
1   <%@ page import="java.sql.*" %>
2   <%
3   Connection con = null;
4
5   try {
6       Class.forName("oracle.jdbc.driver.OracleDriver");
7       con = DriverManager.getConnection(
8           "jdbc:oracle:thin:@localhost:1521:XE",
9           "parth",
10          "Parth@123"
11      );
12  } catch (Exception e) {
13      out.println("<p style='color:red'>Database connection failed</p>");
14  }
15  %>
16
17
```

Writable   Smart Insert   10 : 19 : 249

department.jsp

```jsp
<%@ include file="db.jsp" %>

<%
String msg = "";

/* ADD */
if(request.getParameter("add") != null){
    try {
        int id = Integer.parseInt(request.getParameter("id"));
        String name = request.getParameter("name");

        // Check if department already exists
        PreparedStatement check = con.prepareStatement(
            "SELECT COUNT(*) FROM Department WHERE Dept_ID=?");
        check.setInt(1, id);
        ResultSet crs = check.executeQuery();
        crs.next();

        if(crs.getInt(1) > 0){
            msg = "Department ID already exists!";
        } else {
            PreparedStatement ps = con.prepareStatement(
                "INSERT INTO Department VALUES (?,?)");
            ps.setInt(1, id);
            ps.setString(2, name);
            ps.executeUpdate();
            msg = "Department added successfully!";
        }

    } catch(Exception e){
        msg = "Error while adding department!";
    }
}

/* UPDATE */
if(request.getParameter("update") != null){
    try {
        PreparedStatement ps = con.prepareStatement(
            "UPDATE Department SET Dept_Name=? WHERE Dept_ID=?");
        ps.setString(1, request.getParameter("name"));
        ps.setInt(2, Integer.parseInt(request.getParameter("id")));
        int rows = ps.executeUpdate();

        if(rows == 0){
            msg = "Department ID not found!";
        } else {
            msg = "Department updated successfully!";
        }
    } catch(Exception e){
        msg = "Error while updating department!";
    }
}

/* DELETE */
if(request.getParameter("del") != null){
    try {
        PreparedStatement ps = con.prepareStatement(
            "DELETE FROM Department WHERE Dept_ID=?");
        ps.setInt(1, Integer.parseInt(request.getParameter("del")));
        ps.executeUpdate();
```
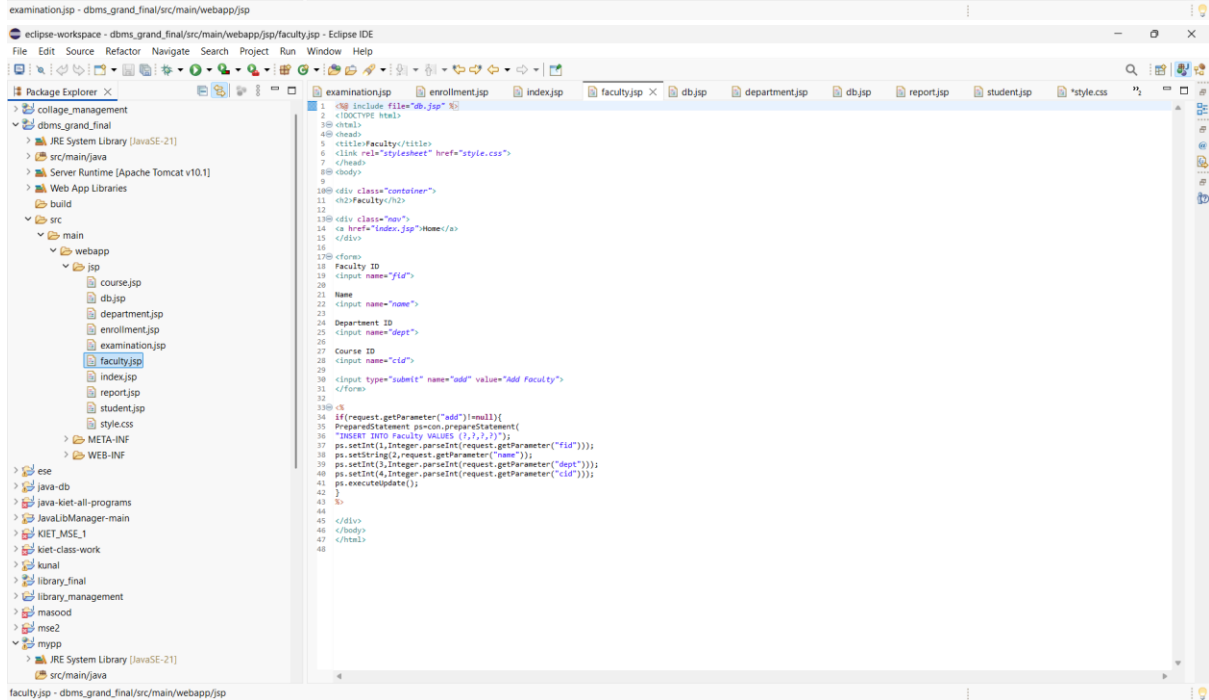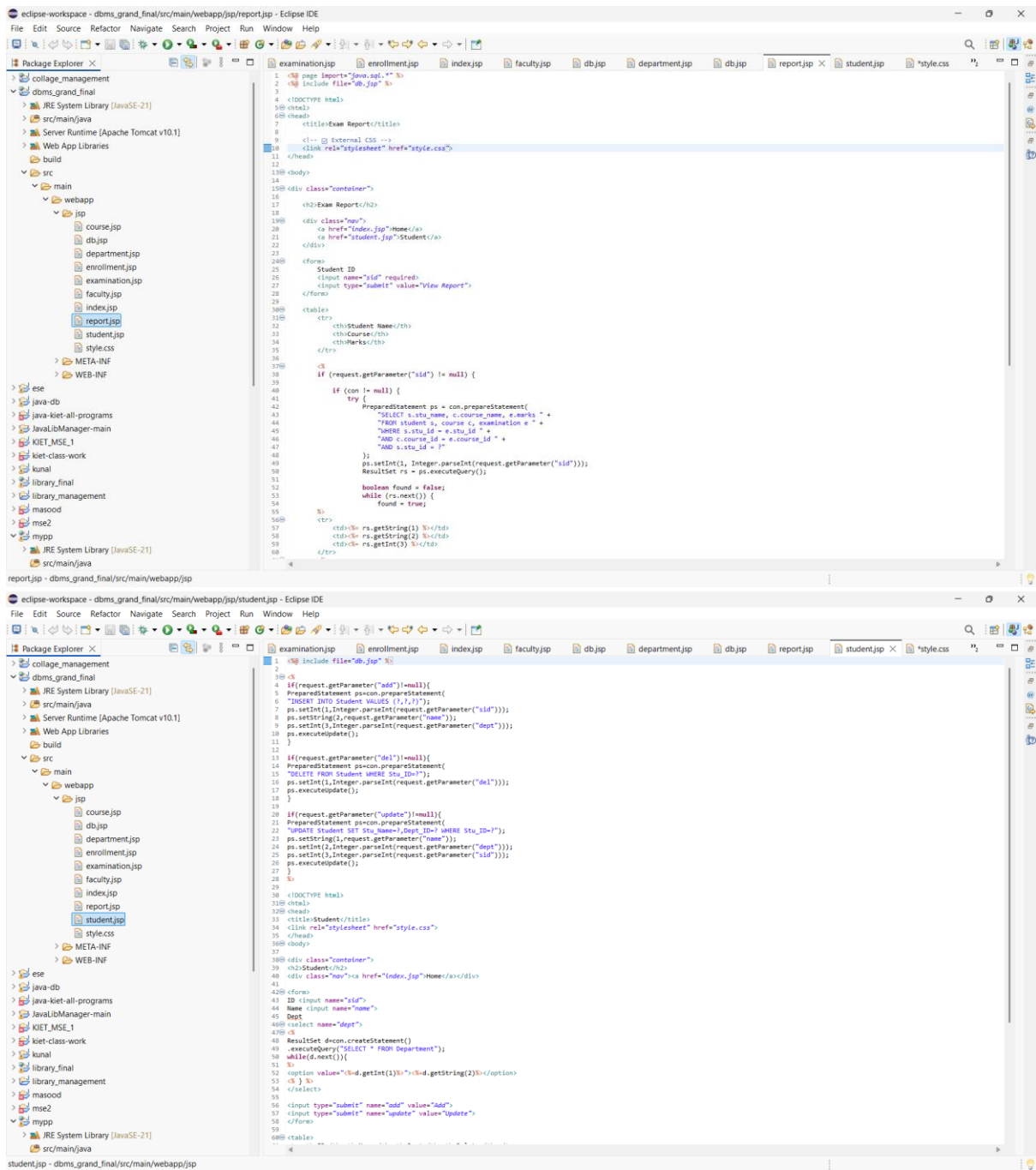
enrollment.jsp

```jsp
<%@ include file="db.jsp" %>

<%
if(request.getParameter("add")!=null){
    PreparedStatement ps=con.prepareStatement(
    "INSERT INTO Enrollment VALUES (?,?,?)");
    ps.setInt(1,Integer.parseInt(request.getParameter("eid")));
    ps.setInt(2,Integer.parseInt(request.getParameter("sid")));
    ps.setInt(3,Integer.parseInt(request.getParameter("cid")));
    ps.executeUpdate();
}

if(request.getParameter("del")!=null){
    PreparedStatement ps=con.prepareStatement(
    "DELETE FROM Enrollment WHERE Enrollment_ID=?");
    ps.setInt(1,Integer.parseInt(request.getParameter("del")));
    ps.executeUpdate();
}
%>

<!DOCTYPE html>
<html>
<head>
<title>Enrollment</title>
<link rel="stylesheet" href="style.css">
</head>
<body>

<div class="container">
<h2>Enrollment</h2>
<div class="nav"><a href="index.jsp">Home</a></div>

<form>
Enroll ID <input name="eid">
Student ID <input name="sid">
Course ID <input name="cid">
<input type="submit" name="add" value="Enroll">
</form>

<table>
<tr><th>ID</th><th>Student</th><th>Course</th><th>Delete</th></tr>
<%
ResultSet rs=con.createStatement()
.executeQuery("SELECT * FROM Enrollment");
while(rs.next()){
%>
<tr>
<td><%=rs.getInt(1)%></td>
<td><%=rs.getInt(2)%></td>
<td><%=rs.getInt(3)%></td>
<td><a href="enrollment.jsp?del=<%=rs.getInt(1)%>">Delete</a></td>
</tr>
<% } %>
</table>
</div>

</body>
</html>
```

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

examination.jsp enrollment.jsp index.jsp faculty.jsp db.jsp department.jsp db.jsp report.jsp student.jsp *style.css

```jsp
<%@ include file="db.jsp" %>

<%
if(request.getParameter("add")!=null){
PreparedStatement ps=con.prepareStatement(
"INSERT INTO Examination VALUES (?,?,?,?)");
ps.setInt(1,Integer.parseInt(request.getParameter("eid")));
ps.setInt(2,Integer.parseInt(request.getParameter("marks")));
ps.setInt(3,Integer.parseInt(request.getParameter("cid")));
ps.setInt(4,Integer.parseInt(request.getParameter("sid")));
ps.executeUpdate();
}

if(request.getParameter("del")!=null){
PreparedStatement ps=con.prepareStatement(
"DELETE FROM Examination WHERE Exam_ID=?");
ps.setInt(1,Integer.parseInt(request.getParameter("del")));
ps.executeUpdate();
}

if(request.getParameter("update")!=null){
PreparedStatement ps=con.prepareStatement(
"UPDATE Examination SET MARKS=? WHERE Exam_ID=?");
ps.setInt(1,Integer.parseInt(request.getParameter("marks")));
ps.setInt(2,Integer.parseInt(request.getParameter("eid")));
ps.executeUpdate();
}
%>

<!DOCTYPE html>
<html>
<head>
<title>Examination</title>
<link rel="stylesheet" href="style.css">
</head>
<body>

<div class="container">
<h2>Examination</h2>
<div class="nav"><a href="index.jsp">Home</a></div>

<form>
Exam ID <input name="eid">
Marks <input name="marks">
Course ID <input name="cid">
Student ID <input name="sid">

<input type="submit" name="add" value="Add">
<input type="submit" name="update" value="Update">
</form>

<table>
<tr><th>ID</th><th>Marks</th><th>Delete</th></tr>
<%
ResultSet rs=con.createStatement()
.executeQuery("SELECT * FROM Examination");
while(rs.next()){
%>
<tr>
<td><%=rs.getInt(1)%></td>
```

examination.jsp - dbms_grand_final/src/main/webapp/jsp

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

examination.jsp enrollment.jsp index.jsp faculty.jsp db.jsp department.jsp db.jsp report.jsp student.jsp *style.css

```jsp
<%@ include file="db.jsp" %>
<!DOCTYPE html>
<html>
<head>
<title>Faculty</title>
<link rel="stylesheet" href="style.css">
</head>
<body>

<div class="container">
<h2>Faculty</h2>

<div class="nav">
<a href="index.jsp">Home</a>
</div>

<form>
Faculty ID
<input name="fid">

Name
<input name="name">

Department ID
<input name="dept">

Course ID
<input name="cid">

<input type="submit" name="add" value="Add Faculty">
</form>

<%
if(request.getParameter("add")!=null){
PreparedStatement ps=con.prepareStatement(
"INSERT INTO Faculty VALUES (?,?,?,?)");
ps.setInt(1,Integer.parseInt(request.getParameter("fid")));
ps.setString(2,request.getParameter("name"));
ps.setInt(3,Integer.parseInt(request.getParameter("dept")));
ps.setInt(4,Integer.parseInt(request.getParameter("cid")));
ps.executeUpdate();
}
%>

</div>
</body>
</html>
```

faculty.jsp - dbms_grand_final/src/main/webapp/jsp

Backend Database implementation on oracle→

Project Based Learning (PBL) Description

Project Title→

College Management System using JSP and Oracle Database

Problem Statement

In many colleges, academic data such as departments, students, courses, enrollments, and examination records are managed manually or using disconnected systems. This leads to data redundancy, errors, difficulty in maintaining relationships between entities, and lack of centralized access. There is a need for a simple, web-based system that can efficiently manage college-related data in a structured and reliable manner.

Objective of the Project

The objective of this project is to design and develop a **web-based College Management System** that provides a centralized platform to manage academic information using **JSP** and **Oracle Database**. The system aims to simplify data entry, ensure data integrity using relational constraints, and provide easy access to student and examination records.

Scope of the Project

The scope of the project includes:

- Managing departments, students, courses, faculty, enrollments, and examinations

- Performing **CRUD operations** (Create, Read, Update, Delete) on all entities

- Maintaining relationships between tables using **primary and foreign keys**

- Generating examination reports for students

- Providing a user-friendly interface using JSP and CSS

The project is designed for academic use and does not include advanced features such as authentication or role-based access.

Technologies Used

- **Frontend:** JSP (Java Server Pages), HTML, CSS

- **Backend:** Java (JDBC)

- **Database:** Oracle 10g

- **Web Server:** Apache Tomcat

- **Tools:** Eclipse IDE, GitHub

System Modules

1. **Department Module**

   o Add, update, delete, and view departments

2. **Student Module**

   o Add, update, delete students

   o Assign students to departments

3. **Course Module**

   o Manage course details

4. **Faculty Module**

   o Store faculty information and department/course mapping

5. **Enrollment Module**

   o Enroll students into courses

6. **Examination Module**

   o Store student marks for courses

7. **Report Module**

   o Generate student examination reports using SQL joins

Database Design

The database follows a **relational model** with proper normalization.
Primary keys uniquely identify records, while foreign keys maintain relationships between tables such as Student–Department, Enrollment–Student, and Examination–Course.

Oracle constraints are used to ensure:

- Data integrity

- Prevention of invalid deletions

- Consistency across related tables

Key Features

- Centralized academic data management

- Real-time database updates

- Referential integrity using Oracle constraints

- Safe handling of delete operations to avoid data inconsistency

- Simple and clean user interface

- External CSS for consistent UI across pages

Learning Outcomes

Through this project, the following concepts were learned:

- JSP page lifecycle and request handling

- JDBC connectivity with Oracle database

- SQL queries including joins and constraints

- Handling database exceptions and integrity violations

- Importance of separation of concerns (UI, logic, database)

- Secure handling of configuration and environment variables

- Version control and secret protection using GitHub

Conclusion

The College Management System successfully demonstrates how a web-based application can manage academic data efficiently using JSP and Oracle Database. The project provides hands-on experience with database design, backend integration, and frontend development. It serves as a strong foundation for understanding enterprise-level web applications and database-driven systems.