**# Project 3**

**## Setup Instructions**

**1. \*\*Prerequisites:\*\***

- Apache Tomcat 10.1

- MySQL Database

- Java JDK (preferably version 8 or higher)

**2. \*\*Database Setup:\*\***

- Create a MySQL database named `student\_db`.

- Run the following SQL commands to create the `students` table:

```sql

CREATE DATABASE student\_db;

USE student\_db;

CREATE TABLE students (

id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(100) NOT NULL,

email VARCHAR(100) NOT NULL

);

```

**3. \*\*Project Deployment:\*\***

- Copy the `project3` directory to the `webapps` directory of your Tomcat installation (if not already present).

- Ensure the `mysql-connector-j-8.4.0.jar` and `servlet-api.jar` are placed in the `WEB-INF/lib` directory.

- Configure the `database.properties` file located in `WEB-INF/classes` with the appropriate database connection details:

```

jdbc.driverClassName=com.mysql.cj.jdbc.Driver

jdbc.url=jdbc:mysql://localhost:3306/student\_db

jdbc.username=root

jdbc.password=yourpassword

```

- Start the Tomcat server.

**4. \*\*Access the Application:\*\***

- Open a web browser and navigate to `http://localhost:{portnumber}/project3`.

**## Assumptions and Design Decisions**

- The database connection details are stored in a `database.properties` file for easy configuration.

- The application follows the MVC architecture to separate concerns and improve maintainability.

- JSP pages are used for the presentation layer, while Servlets handle the business logic and database interactions.

**## Challenges Faced and Solutions**

- \*\*Editing Student Data:\*\* Initially, the edit function was embedded within the `viewStudents.jsp` page. This caused issues when trying to delete or edit student data, leading to access errors. The solution was to separate the edit functionality into a new option in `index.jsp` and create a dedicated `editStudentForm.jsp` page for editing students.

- \*\*Backend Data Access Issues:\*\* After entering data through the frontend, accessing it from the backend was problematic and frequently resulted in errors. These issues were resolved by debugging and fixing the `StudentDAO.java` file, ensuring proper database interactions.

**## Sample Execution**

**1. \*\*Add a New Student:\*\***

- Navigate to `http://localhost:{portnumber}/project3/index.jsp`.

- Click “Add New Student”

A white background with black text

Description automatically generated



- Fill in the student details and click Add Student.

A screenshot of a computer

Description automatically generated

The new student is now added

A screenshot of a computer

Description automatically generated



**2. \*\*View All Students:\*\***

- Navigate to `http://localhost: {portnumber}/project3/viewStudents.jsp`.

- View the list of all students. You can delete a student from this page.

A screenshot of a computer

Description automatically generated

Lets try deleting the student with ID: 8 Name: Naruto Uzumaki Email: [naruto@hokage.com](mailto:naruto@hokage.com)

A screenshot of a computer

Description automatically generated



After Deleting, this is the result:

A list of email and other information

Description automatically generated with medium confidence

**3. \*\*Edit a Student:\*\***

- Navigate to `http://localhost: {portnumber}/project3/EditStudentServlet`.

- Select a student to edit from the list, modify the details, and click Update Student.

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

After updating, this is the result

A screenshot of a computer

Description automatically generated

**Project Structure**  
project3/

├── README.docx

├── WEB-INF/

│ ├── classes/

│ │ ├── com/

│ │ │ └── example/

│ │ │ ├── AddStudentServlet.class

│ │ │ ├── DatabaseUtil.class

│ │ │ ├── EditStudentServlet.class

│ │ │ ├── Student.class

│ │ │ ├── StudentDAO.class

│ │ │ ├── ViewStudentsServlet.class

│ │ │ └── (other compiled .class files)

│ │ └── database.properties

│ ├── lib/

│ │ ├── mysql-connector-j-8.4.0.jar

│ │ └── servlet-api.jar

│ └── src/

│ ├── com/

│ │ └── example/

│ │ ├── AddStudentServlet.java

│ │ ├── DatabaseUtil.java

│ │ ├── EditStudentServlet.java

│ │ ├── Student.java

│ │ ├── StudentDAO.java

│ │ ├── ViewStudentsServlet.java

│ │ └── (other .java source files)

├── addStudent.jsp

├── editStudentForm.jsp

├── index.jsp

├── viewStudents.jsp

└── (other .jsp files)