**REPORT**

**Project name:-Student Classified And Search**

**Submitted To:- Submitted By:-**

**Mr. Sachin Chirgaiya Shruti Chouksey**

**1.For the user**

**About**

A software application for schools and universities to manage student data. Capabilities of these student databases include attendance, behavioural and medical information, as well as storing assessment information.A student management system (also known as a student information system or SIS) helps a school manage data, communications, and scheduling.A school system generates and uses a large amount of data. This data must be communicated appropriately to students, faculty, and parents. A student management system helps schools to store, manage, and distribute this information.

Some student management systems are designed to serve all of a school’s data management needs. Other student management systems are specialized. These specialized solutions target specific needs, such as school applications or student behavior tracking.

Features:

1. Manage courses within a class environment
2. Support for profile based management: Administrator, Teacher and Student
3. Online teacher and student registration
4. Allow students to self manage class enrollment
5. Allow teachers to manage student attendance
6. Ability for students to submit course evaluations

**2.Technical terms**

Objective Of the Project

* Faster processing time and more accurate data for student requests and reimbursements
* Ability for Students to track authorization and reimbursement request status through the system rather than via phone calls or campus mail
* Major technological upgrades to the current Student system
* Use of IU's standardized, virtual J2EE environments
* Many new features and enhancements

Installation/Configuration Steps

* Download zip file from <https://github.com/Shruti1830/javaproject.git>
* Copy and Paste the unzip files inside the workspace folder

Database Configuration:

* Create a new database named “Student”.
* Import database management.sql file through MySQL

Run/Execute

* Open Your Web Browser
* Put/type inside the web browser : "localhost/Student"

Admin Login

* Open Your Web Browser
* Put/type inside the web browser : "localhost/Student/login.jsp"
* Admin User : Shruti
* Admin Password : 1234567890

**3. The hardware requirements (min) are:**

* 1. CPU (Dual core, 950MHz, CPU)
  2. Memory (512MB RAM)
  3. Video graphics adaptor (16bit VGA)
  4. Network card (1GB Ethernet)
  5. Hard disk (128GB)

**3.1. The Software requirements are:**

1. Eclipse version 3.0 IDE
2. MySQL connectivity for database.
3. Tomcat server 7.0 or 8.0

**3.2. The Technology we use here:**

1. Java
2. Jsp servlet
3. MySQL for JDBC
4. Html, Css, JavaScript, Bootstrap.

**4. Methodology**

* The approach employed in designing the proposed system is the Rational Unified Process (RUP). The RUP
* methodology is based on the fact that the system represents an organized way of gathering business
* Requirements and building the goal of the project. This was employed, because it is an object-oriented and web-enabled program development methodology and also a framework for developing software systems. It also clearly outlines the different roles of the individuals involved in the project, such as the project manager,

**5. System Testing and Component Testing**

System testing is the integration of two or more components of a system as it relates to the functionality of the

system and running an integrated test on the entire system. The software was tested with Web browsers namely

Internet Explorer 8, Google Chrome version 47.6 and Opera 35.0. Windows OS and Mac OS and WAMP server

version 2.5 was also used to test the software.

Component testing also known as module testing is the process of testing the individual components of the

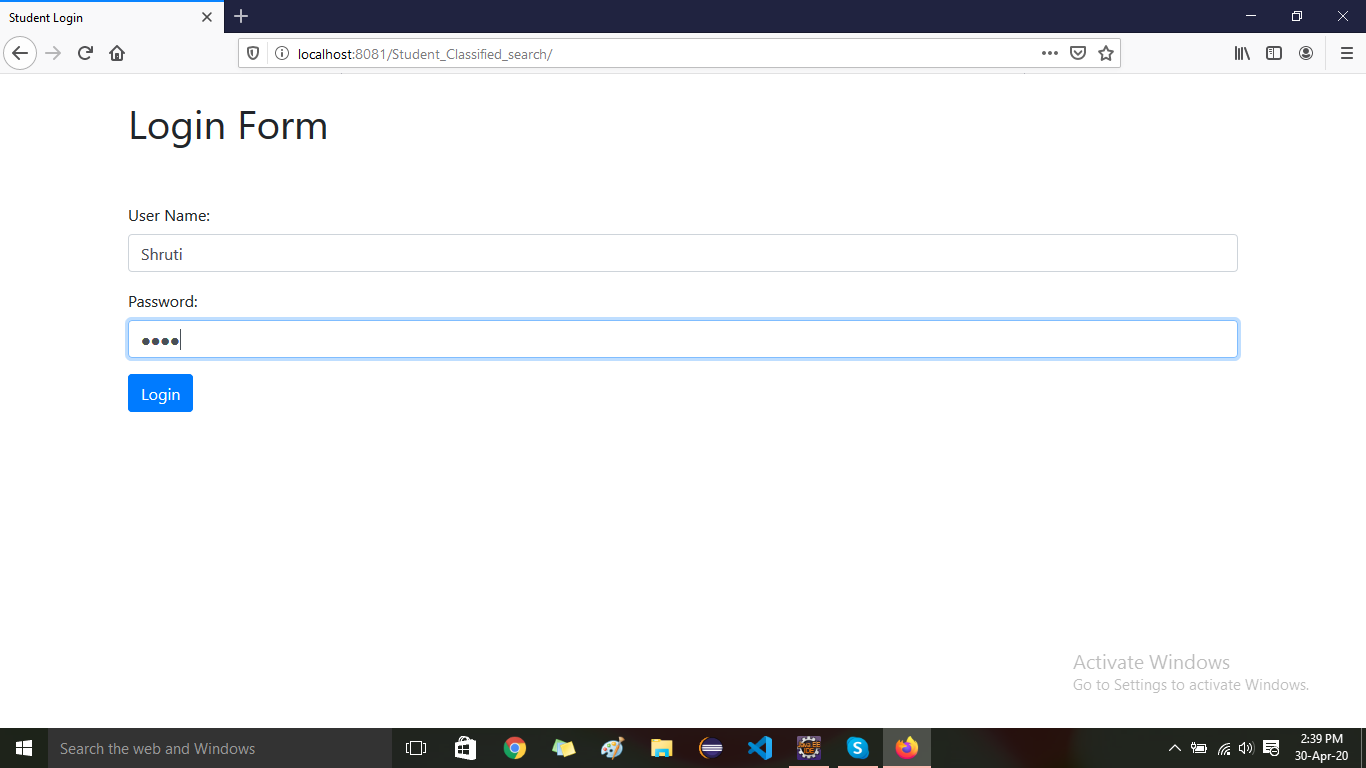
system to determine if its desired functionality is met. It helps to find defect in components or modules of a

system; making sure all faults in the components are exposed for correction. Component testing is essential

because it helps to find bugs before integration testing is carried out.

**6.User Validation**

**Login page**

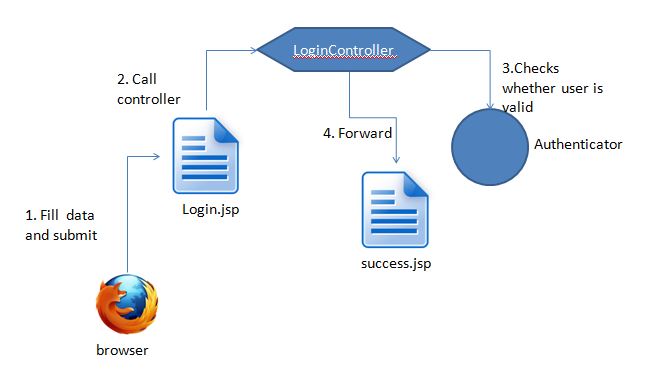


shows the user validation page. In the event that a user who intends to have access to the system but

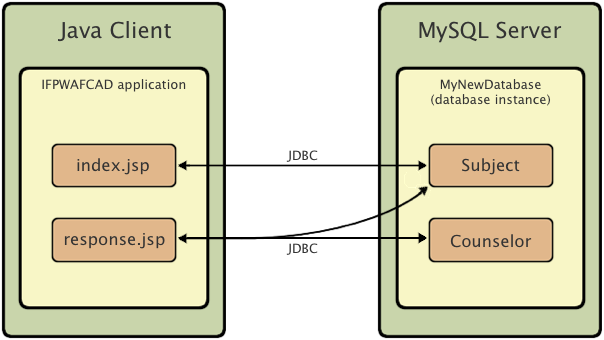
ends up inputting a wrong username or password, the system displays a warning message showing “Invalid

Username or Password”.

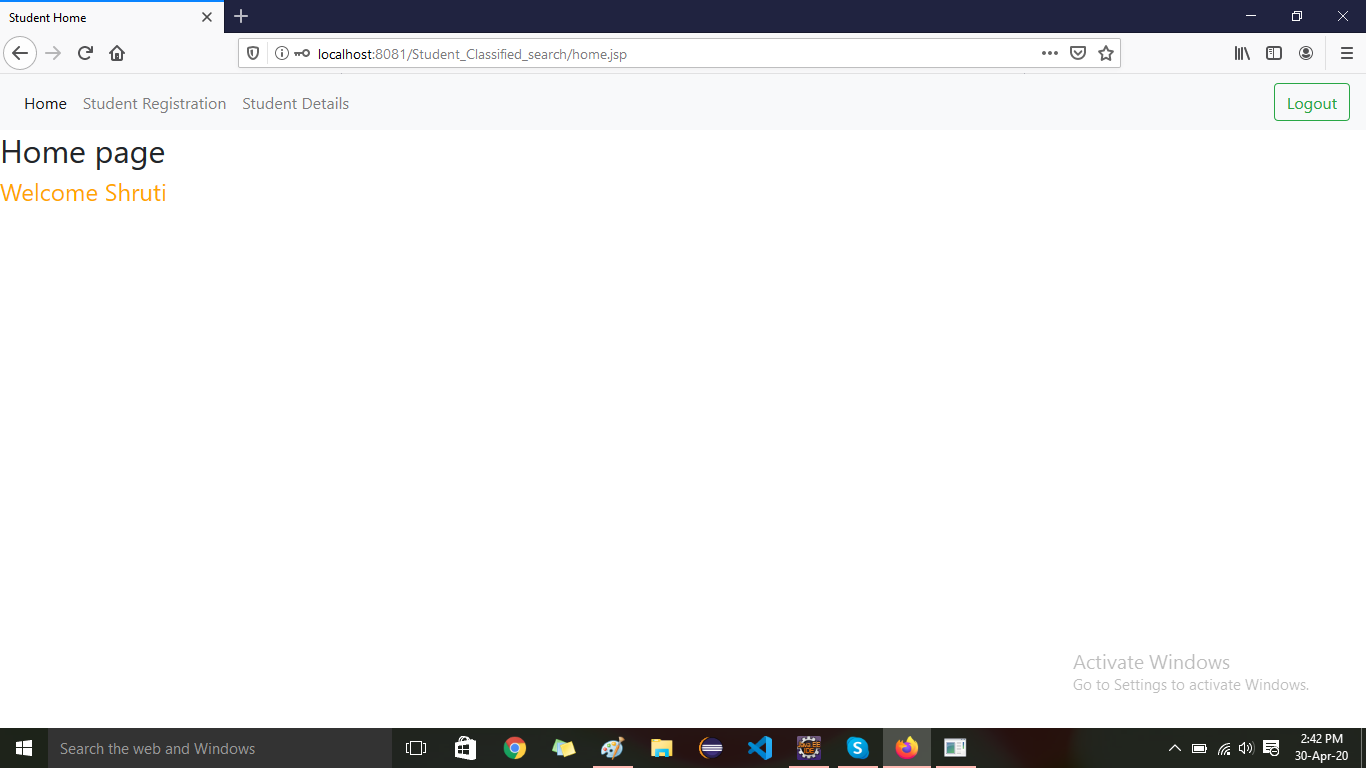
**Login page Validation diagram:-**



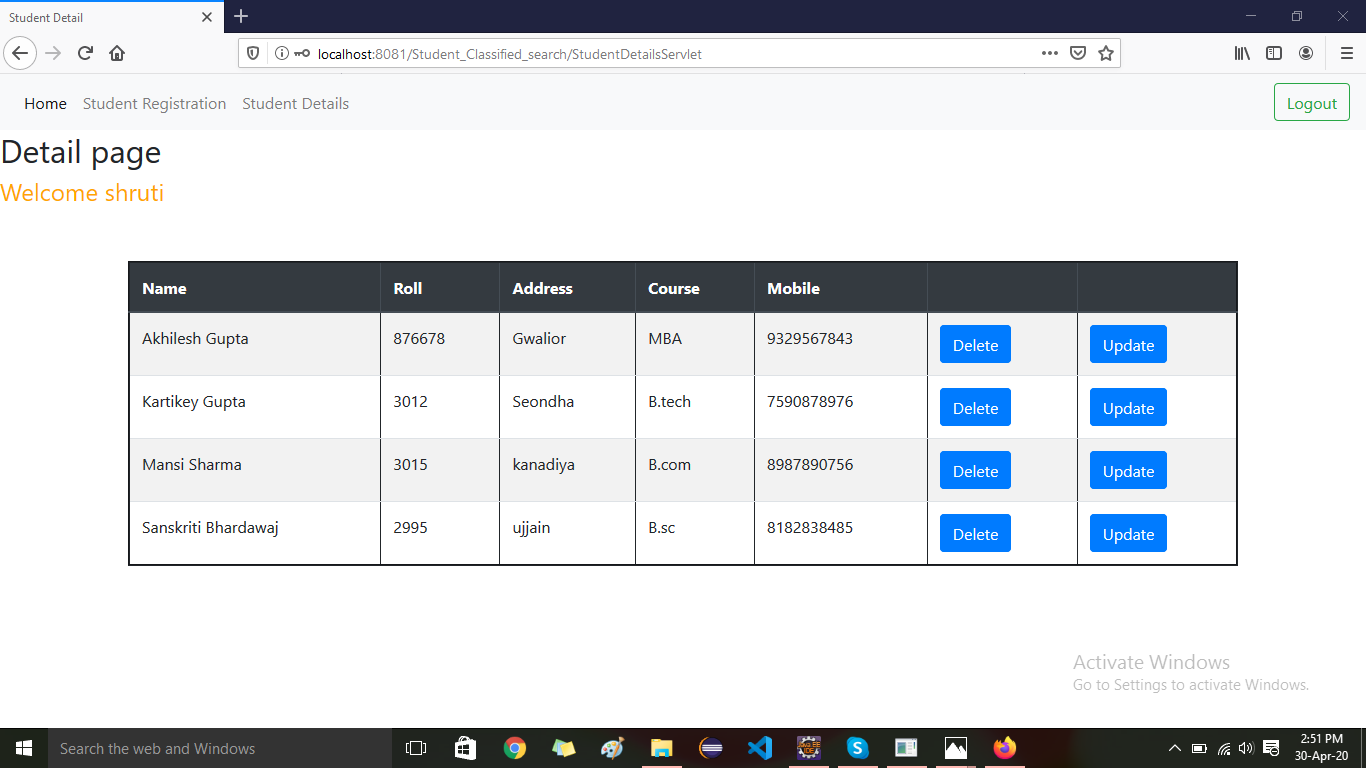
**Servlet connectivity with Database:-**



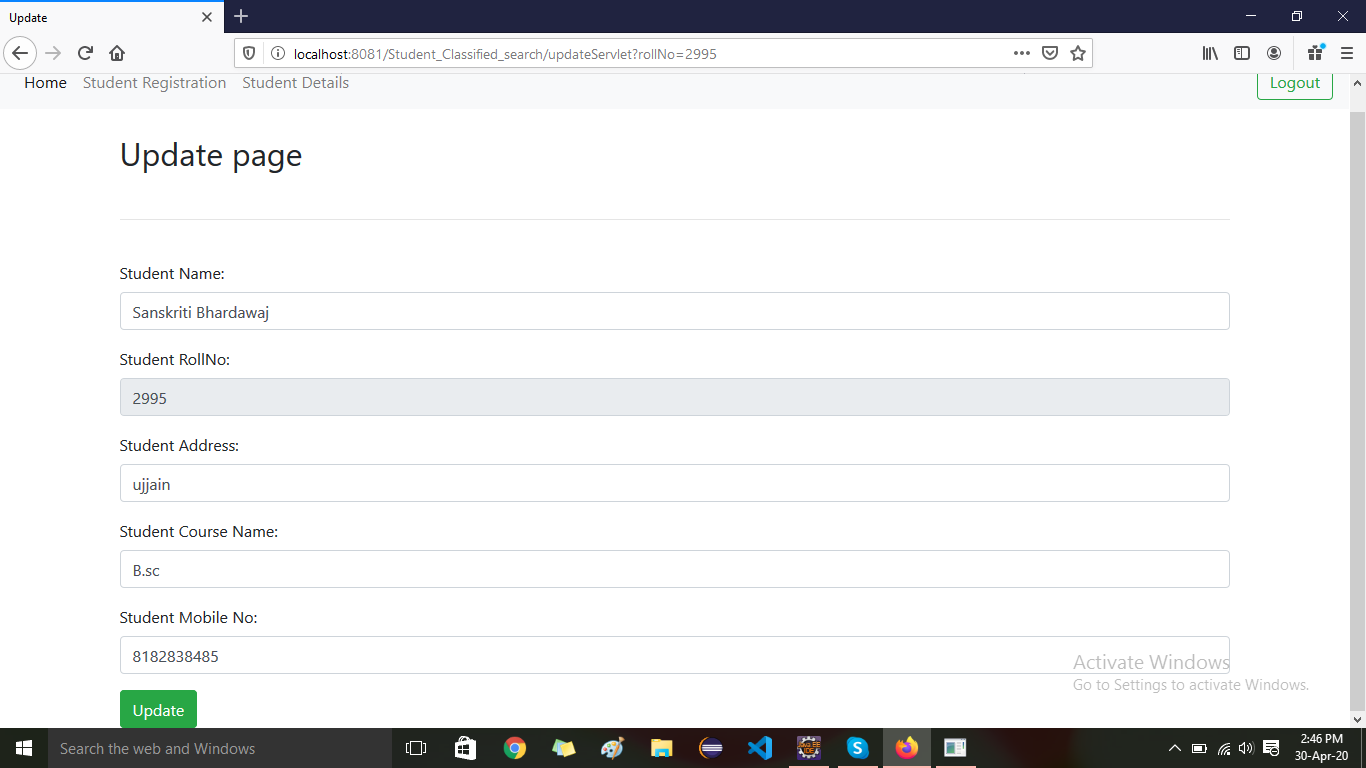
**First Page after successful login:-**

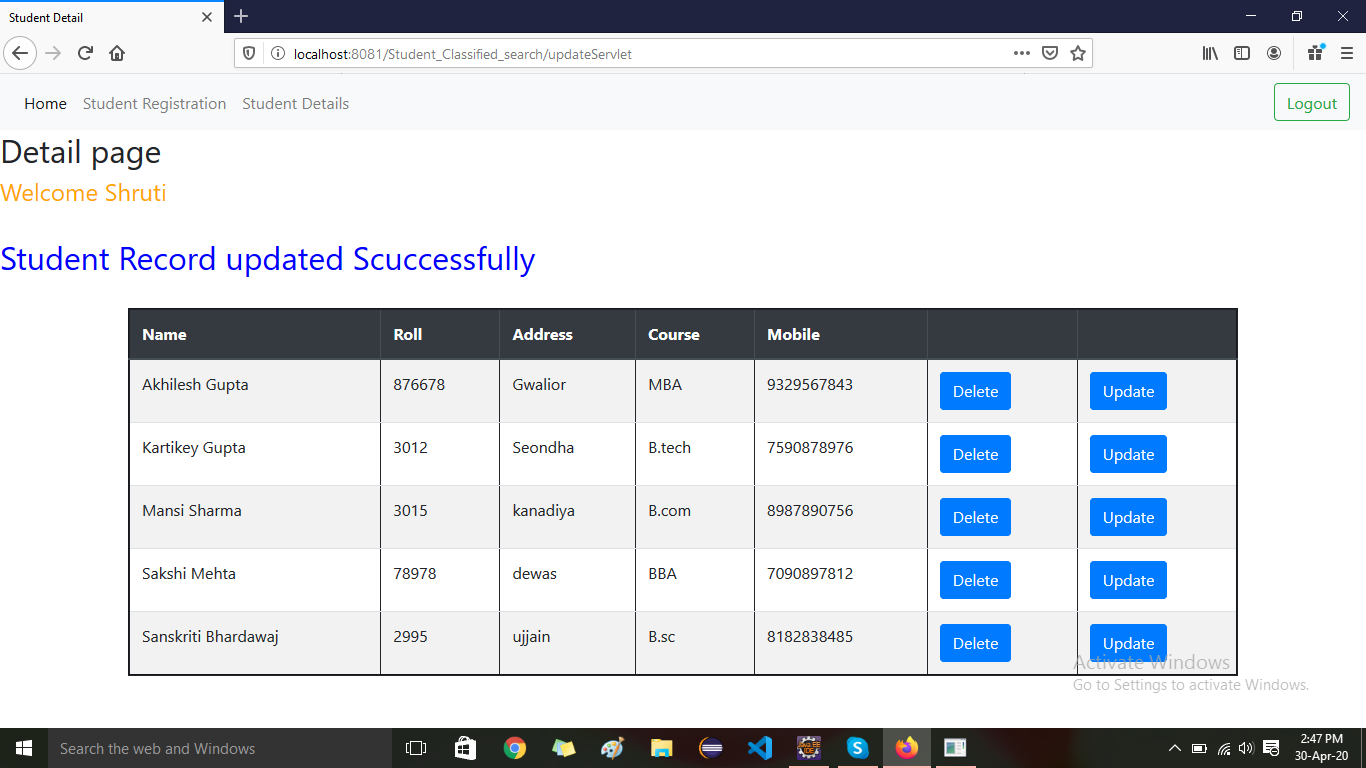


**Student Detail page:-**

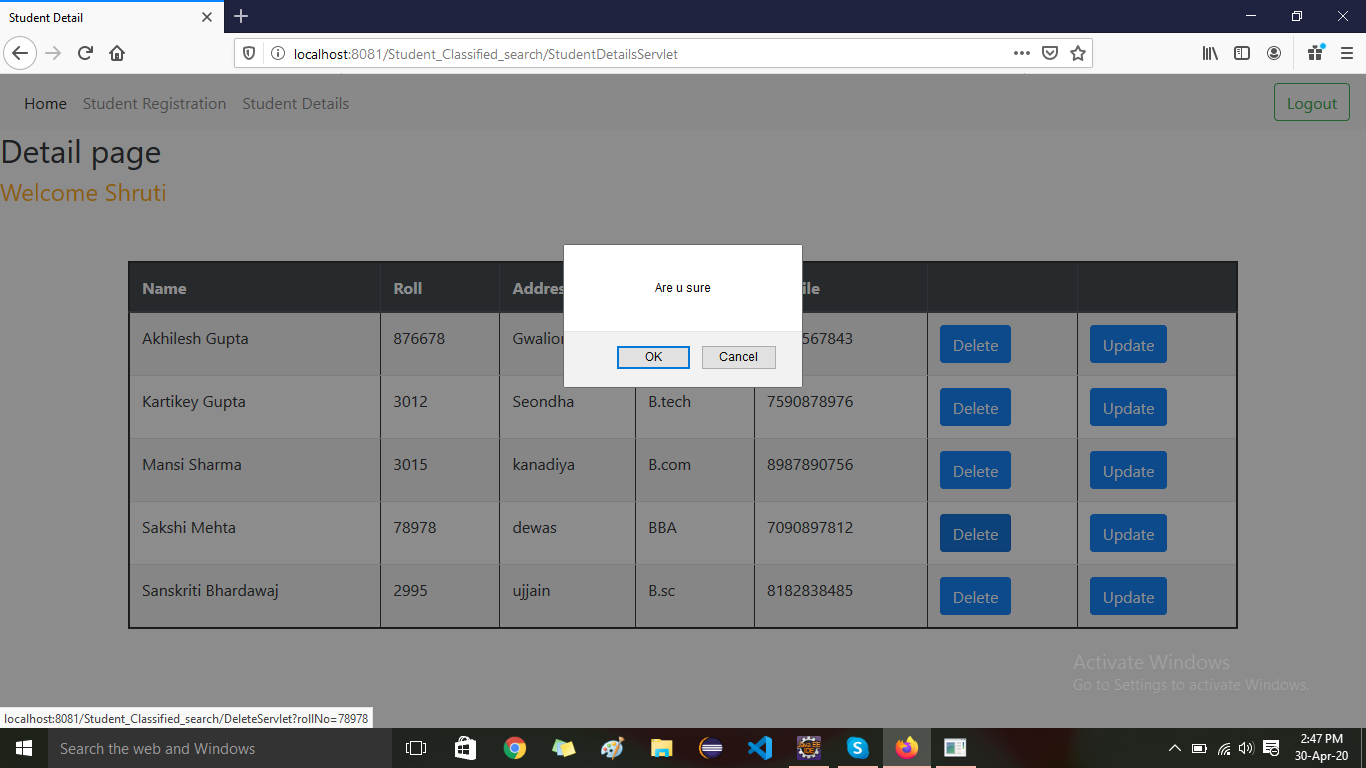
****

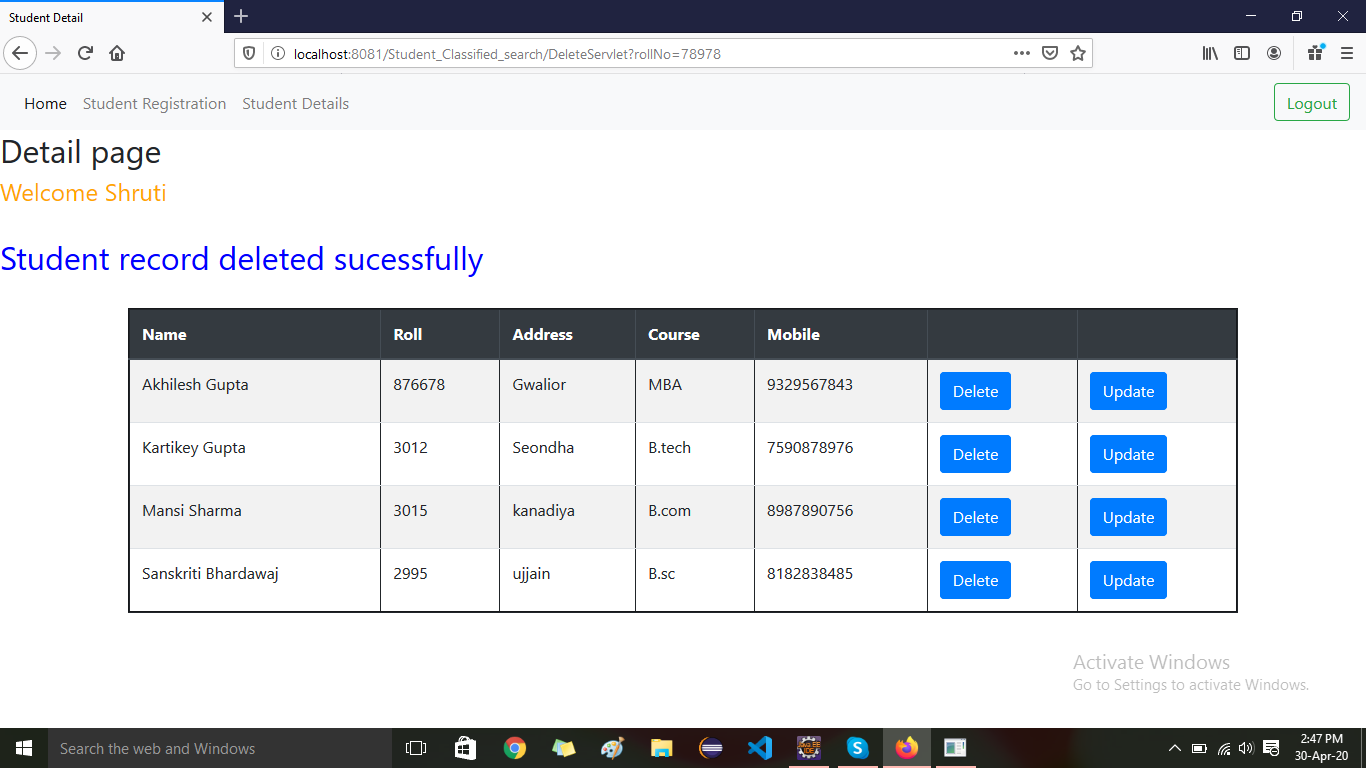
**Update page:-**

****

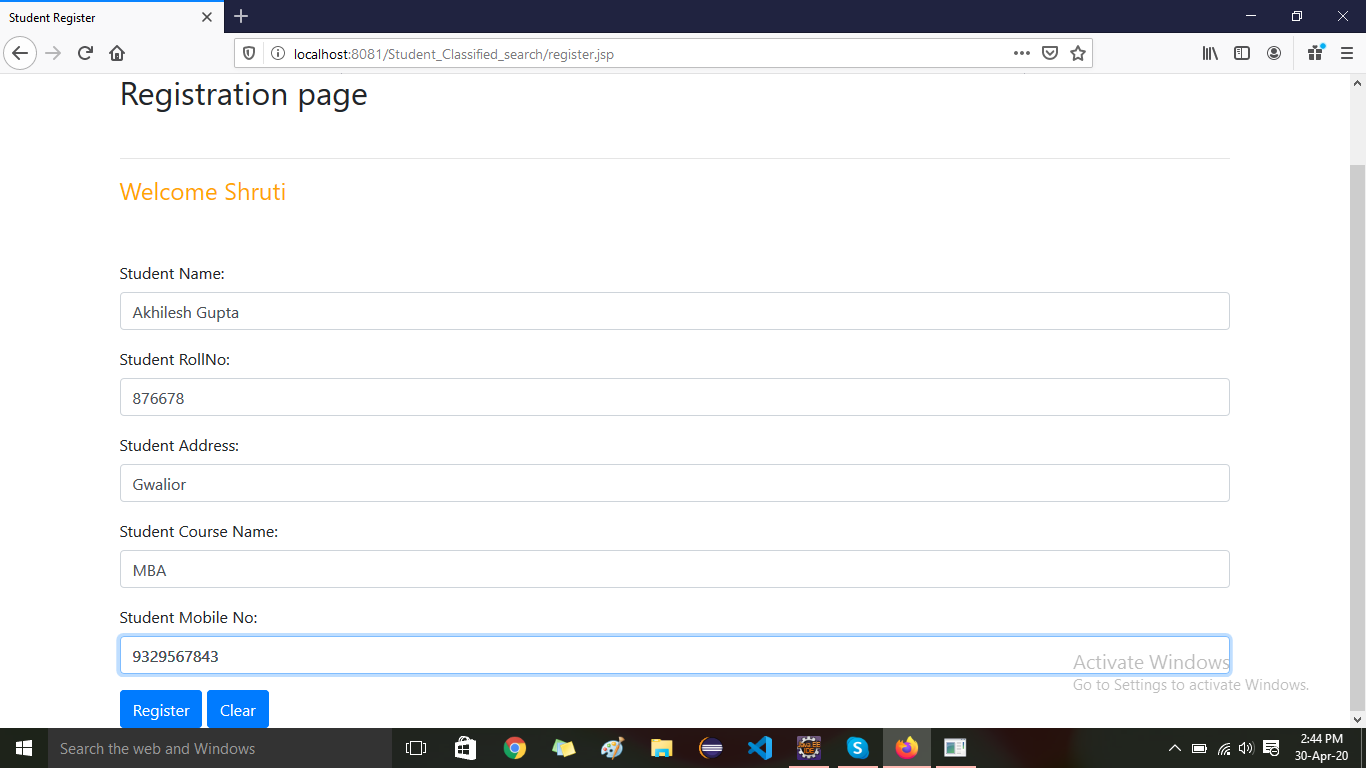


**Delete Student Record page:-**

****

****

**7.Registration Page:-**

****

shows the registration verification after a user has successfully filled in the boxes with the appropriate

information. The system returns a message to the user showing the left blank boxes to fill and a “Registration

successful, proceed to login” message if all boxes are filled correctly. This information is stored in the database

and in turn this information is used by the user to login to the system.

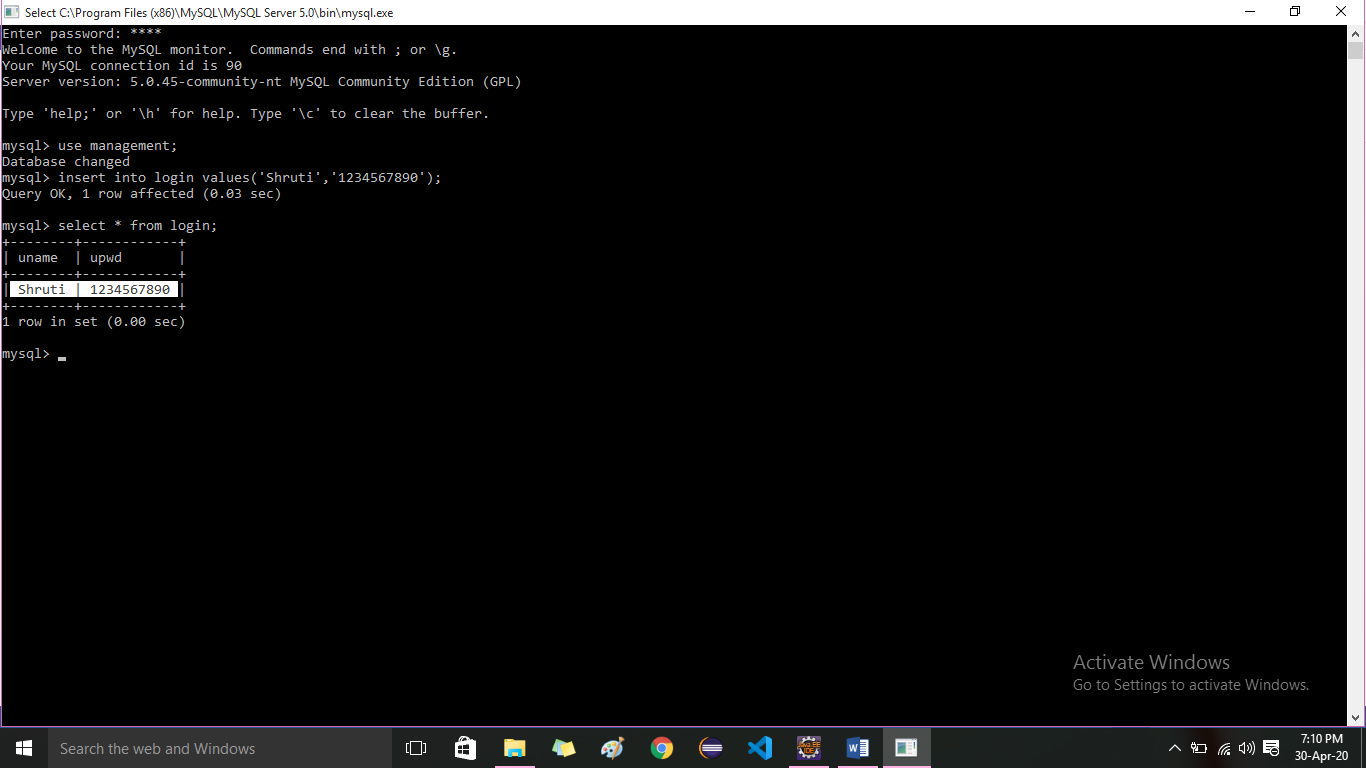
**8. Database Testing For Login page:-**

Database testing is essential because it helps to obtain errors which might affect the system performance,

reliability, consistency and security. It also helps to validate the system against the requirement specified by the

user . It is essential one performs database testing to obtain a database system which satisfies the acid

properties (Atomicity, consistency, isolation and durability) of a database management system.



**Database connectivity diagram:-**

