**PROJECT REPORT**

***Developing a Backend Admin for Learner’s Academy***

**DEVELOPER – SURA KUSALA**

**Project Description:**

**Project objective**:

As a Full Stack Developer, design and develop a backend administrative portal for the Learner’s Academy. Use the GitHub repository to manage the project artefacts.

**Background of the problem statement:**

Learner’s Academy is a school that has an online management system. The system keeps track of its classes, subjects, students, and teachers. It has a back-office application with a single administrator login.

**The administrator can**:

* Set up a master list of all the subjects for all the classes
* Set up a master list of all the teachers
* Set up a master list of all the classes
* Assign classes for subjects from the master list
* Assign teachers to a class for a subject (A teacher can be assigned to different classes for different subjects)
* Get a master list of students (Each student must be assigned to a single class)

There will be an option to view a Class Report which will show all the information about the class, such as the list of students, subjects, and teachers.  
       
The goal of the company is to deliver a high-end quality product as early as possible.

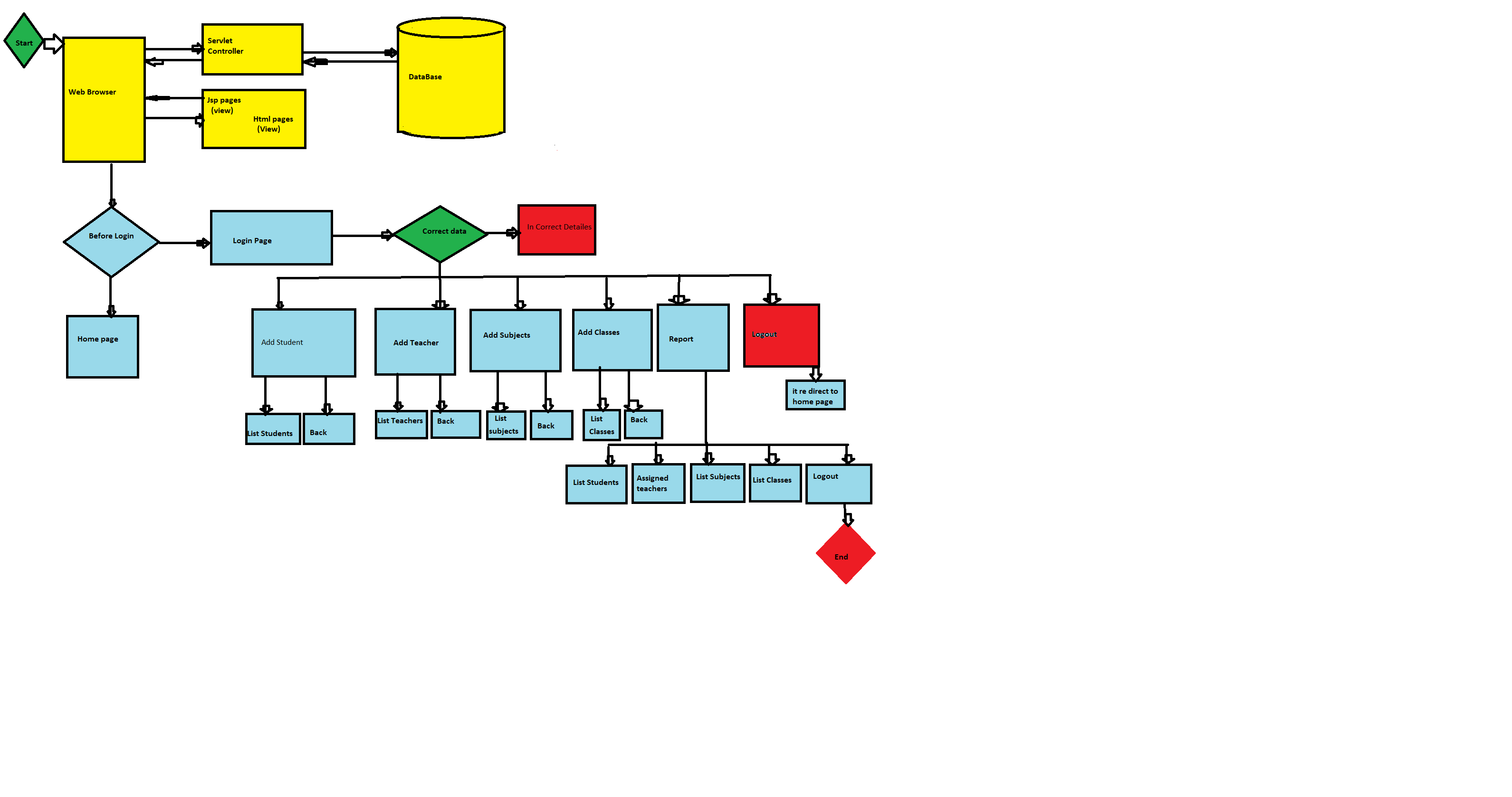
**Technologies and tools Used:**

* Servlet: to do the business logic and works a controller for the project.
* JSP: to handle the presentation view.
* SQL: to create and manage the database.
* JDBC: to make operations on the database for the project.
* CSS: to format the contents.
* phpMyAdmin: to administrate and manage the database manually.
* Eclipse: to write and run the code.
* Tomcat: to run and deploy servlet application.

**Core concepts used in the project:**

* Object-Oriented: used to create and model objects for users and their credentials.
* Databases: used to store and retrieve data.
* Data Sources: used to define a set of properties required to identify and access the database.
* Collections: used some collections such array list to store collection of data.
* Exception Handling: used to catch problems that arises in the code especially in I/O blocks.
* Cookies: to store log-in data on the client browser.

**FLOW CHART:**



**PROJECT USER STORIES (Agile and Scrum):**

The project is planned to be completed in 3 sprints. Tasks assumed to be completed in the sprint are:

* Creating the flow of the application.
* Initializing git repository to track changes as development progresses.
* Writing the Java program to fulfill the requirements of the project.
* Pushing code to GitHub.

**Sprint 1**

1. As a Developer, I want to design and develop a backend administrative portal for the Learner’s Academy.
2. As a Developer, I want to set up a master list of all the subjects for all the classes.
3. As a Developer, I want to set up a master list of all the teachers.

**Sprint 2**

1. As a Dev, I want to set up a master list of all the classes.
2. As a Dev, I want to assign classes for subjects from the master list.
3. As a Dev, I want to assign teachers to a class for a subject.

**Sprint 3**

1. As a user I don’t want the process to be terminated when entering an invalid input.
2. As a Dev, I want to get a master list of students.
3. As a Dev, I want to create an admin login who can access the portal.

**PROJECT GIT REPOSITORY**

Link : <https://github.com/KUSALA-sura/Learners_Academy_Phase-2_Project-1/tree/master/Student_Mngt_System_1>

Clone git : https://github.com/KUSALA-sura/Learners\_Academy\_Phase-2\_Project-1.git

**How to run Project**

**Method of implementation to running project**

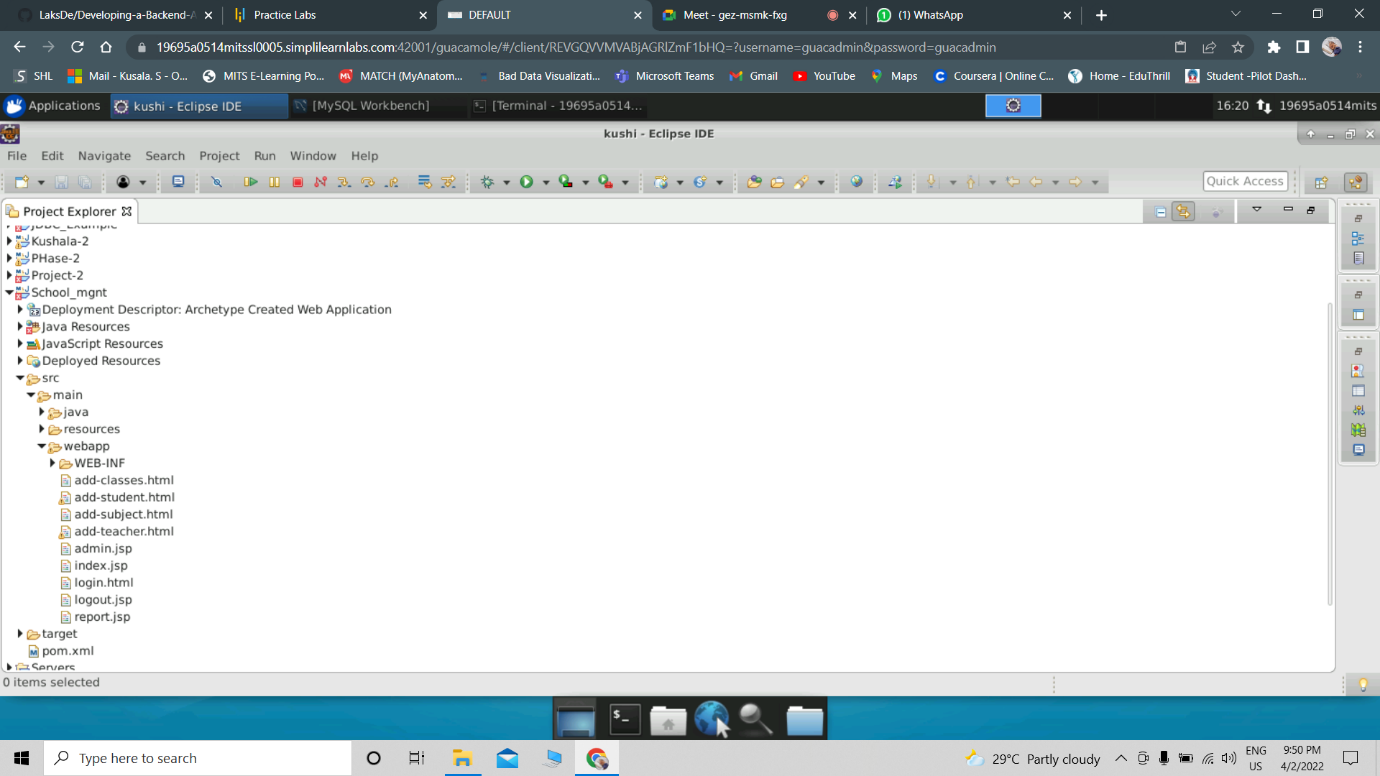
Clone project clone git :

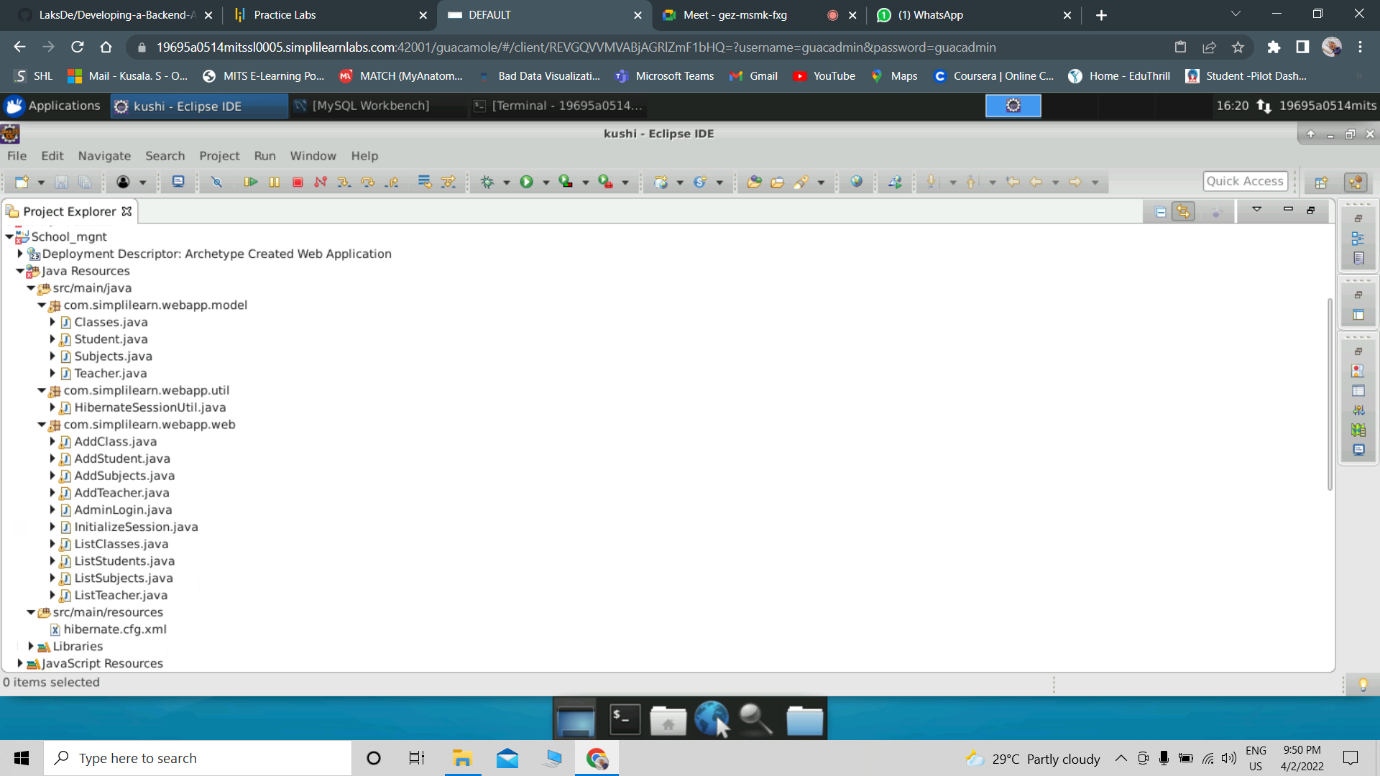
<https://github.com/KUSALA-sura/Learners_Academy_Phase-2_Project-1.git>

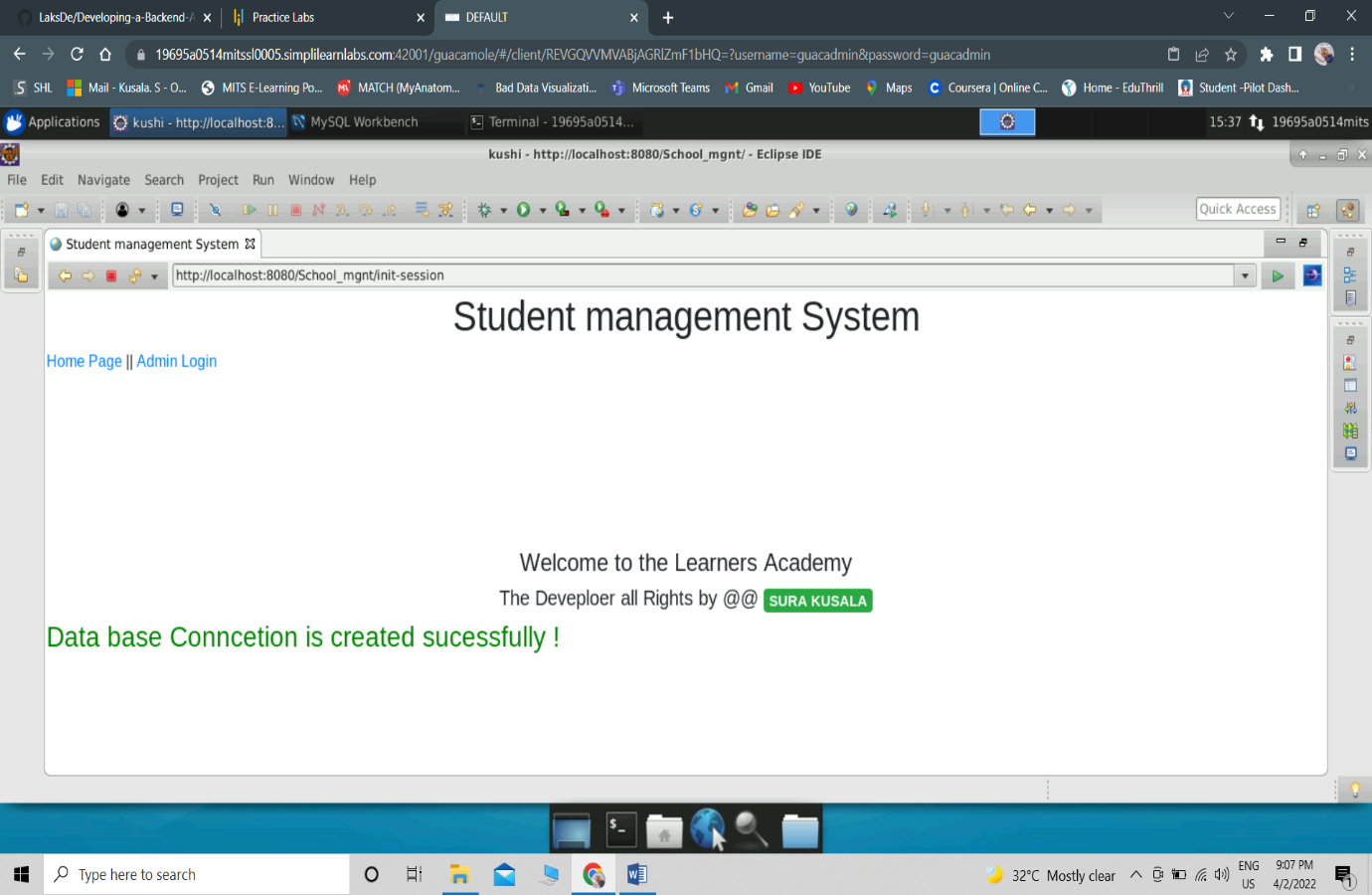
Open :

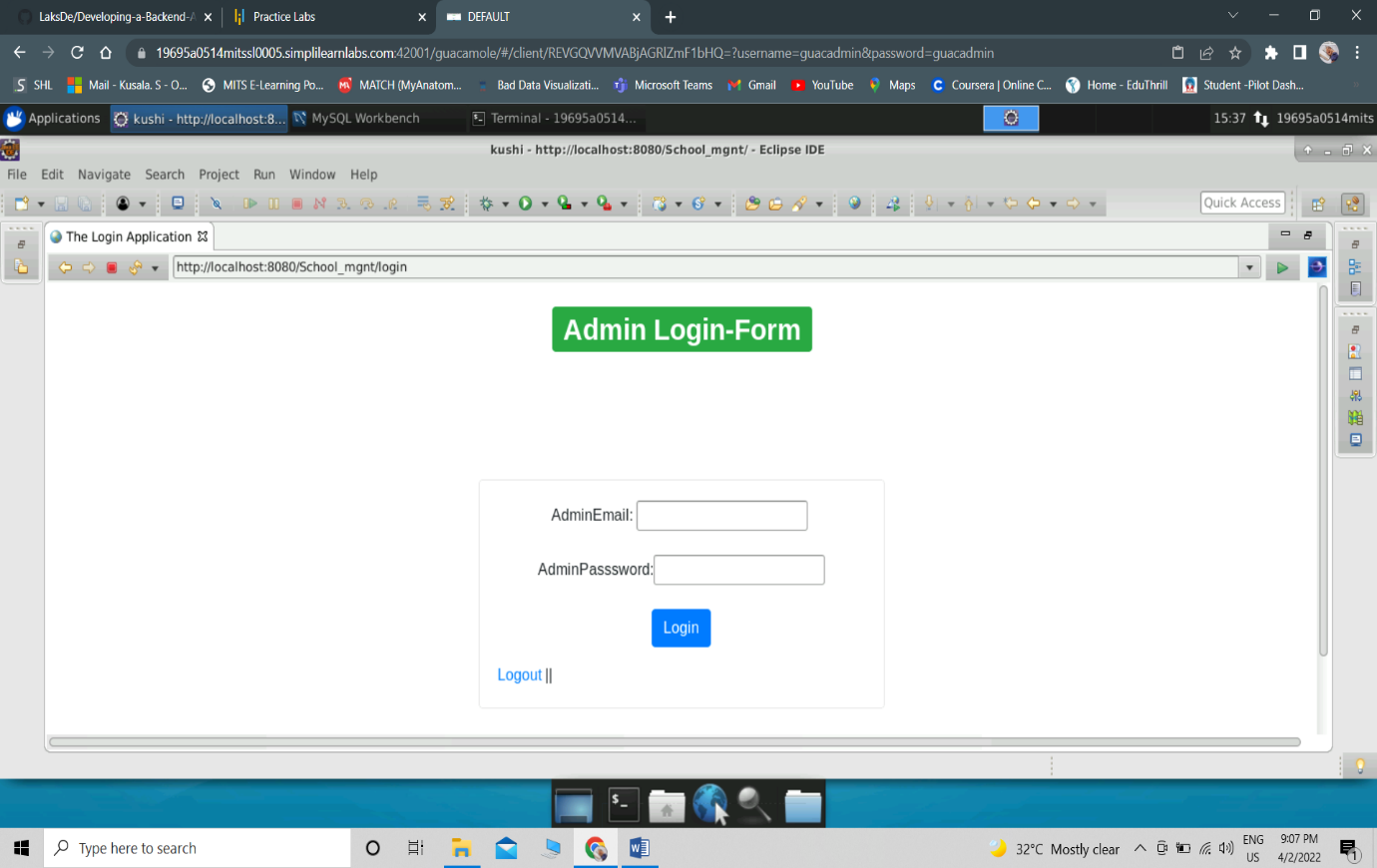
Learners\_Academy\_Phase2\_Project1/Student\_Mngt\_System\_1/src/main/java/com/simplilearn/webapp/util/HibernateSessionUtil.java

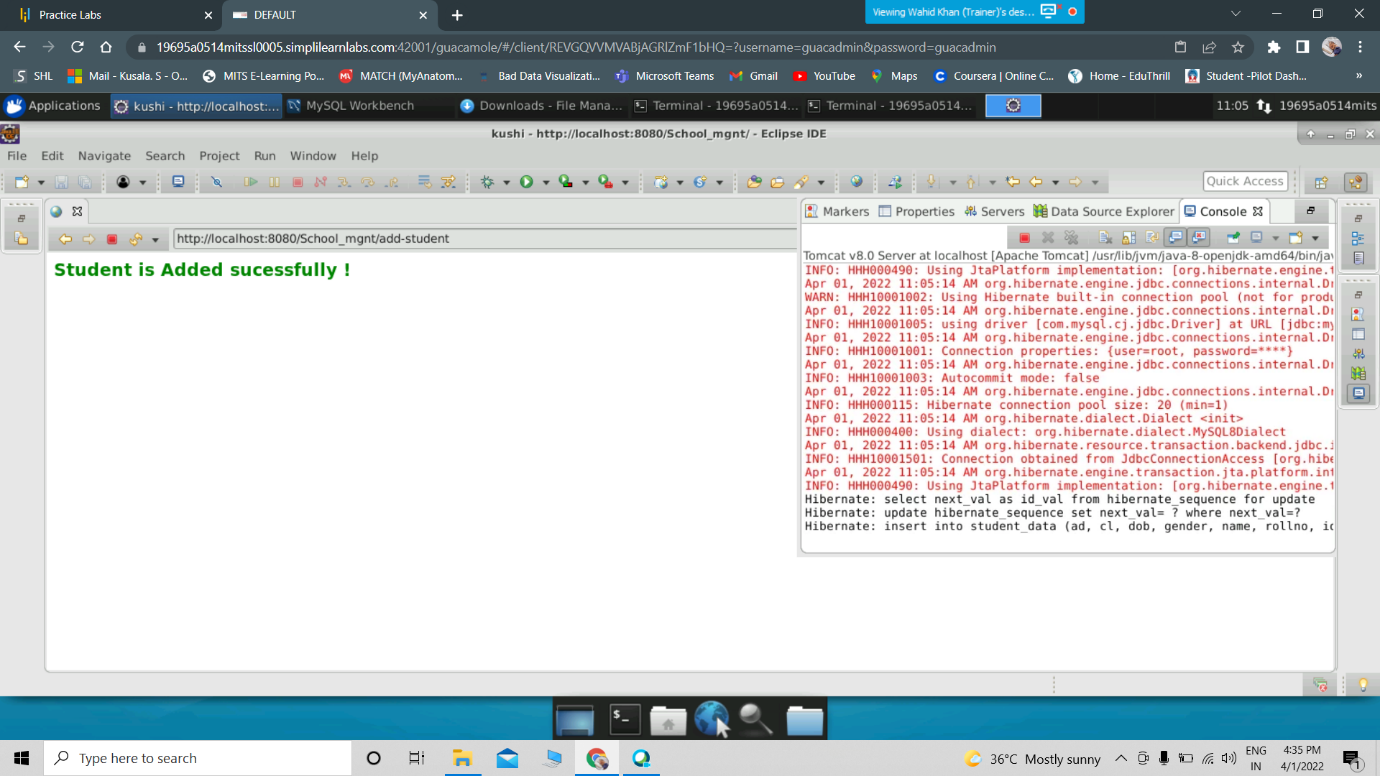
**Screenshots output results:**

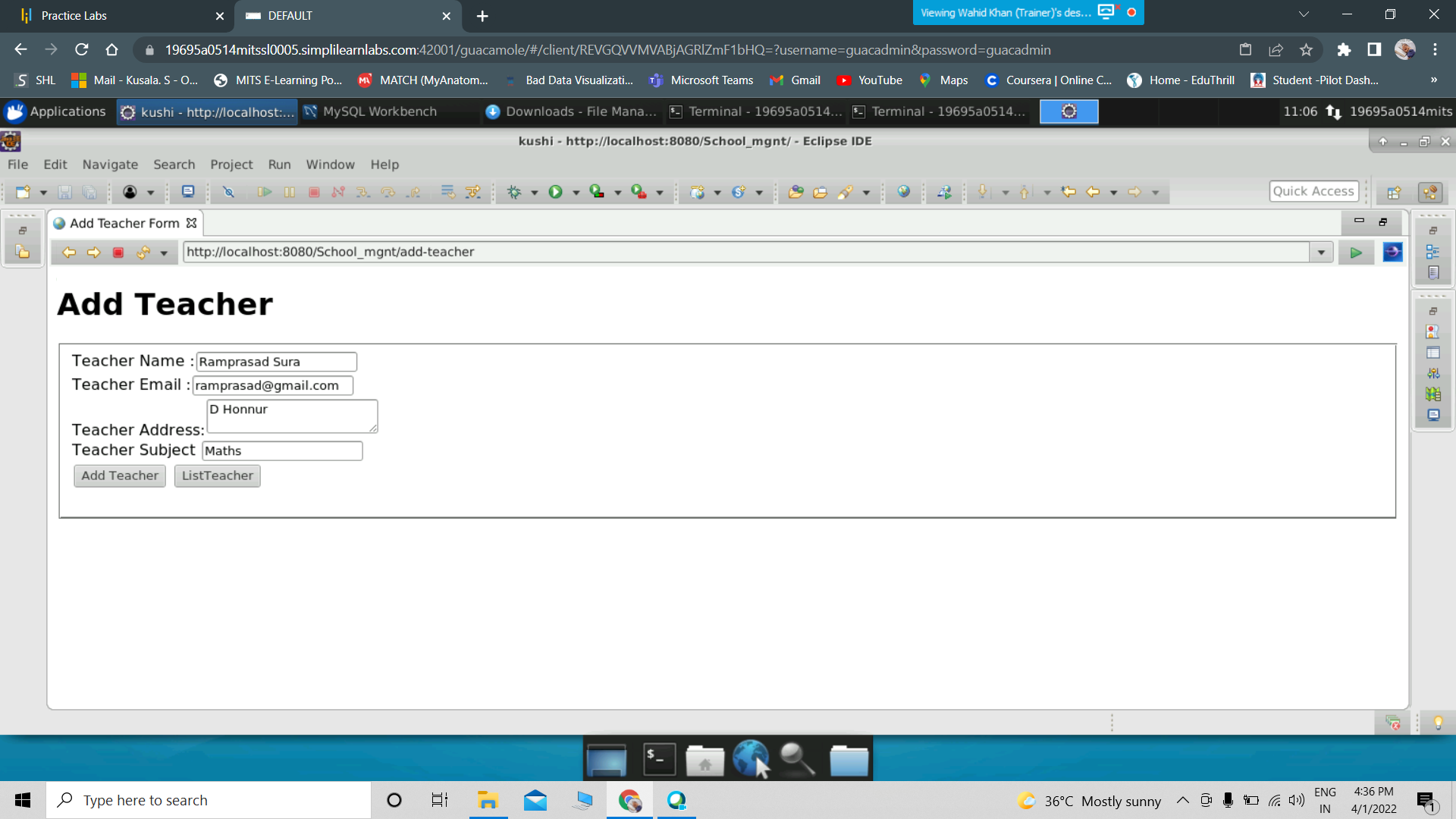
****

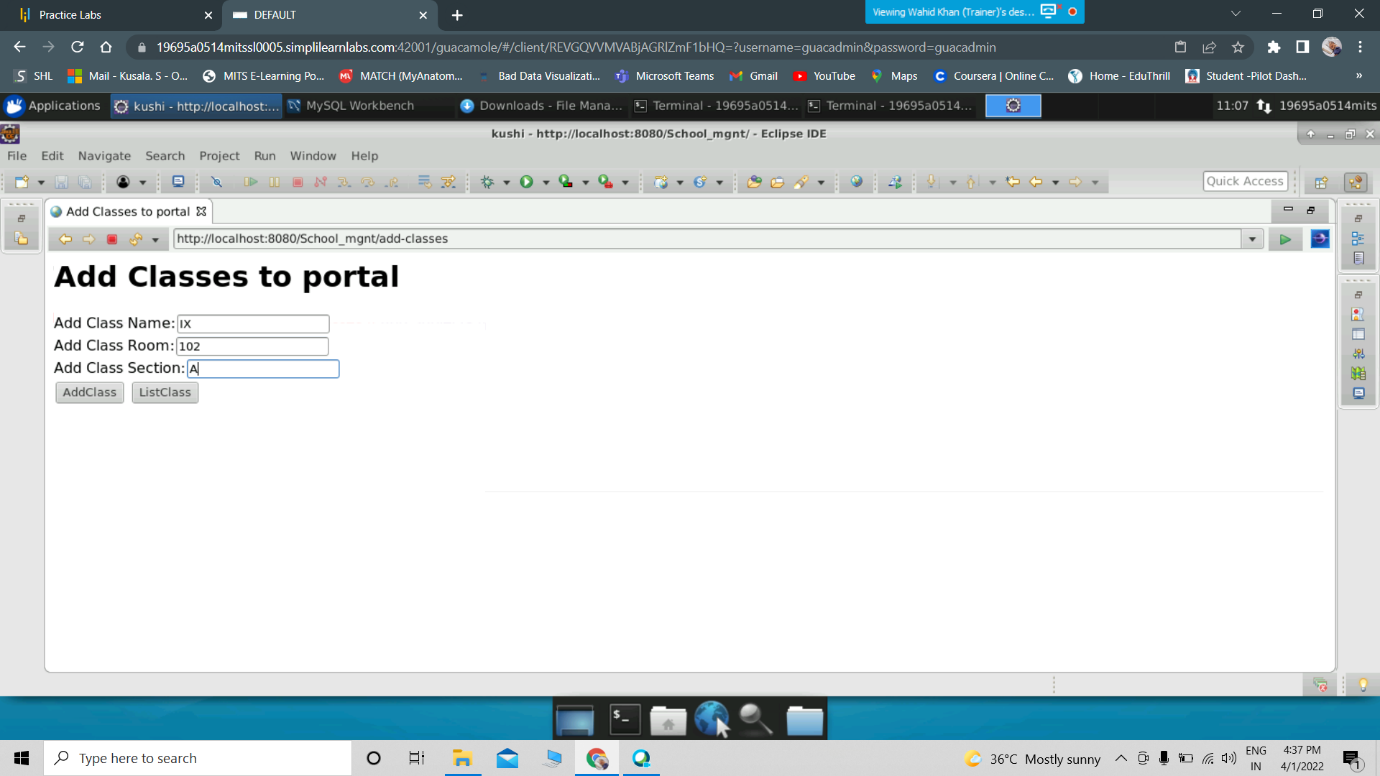
****

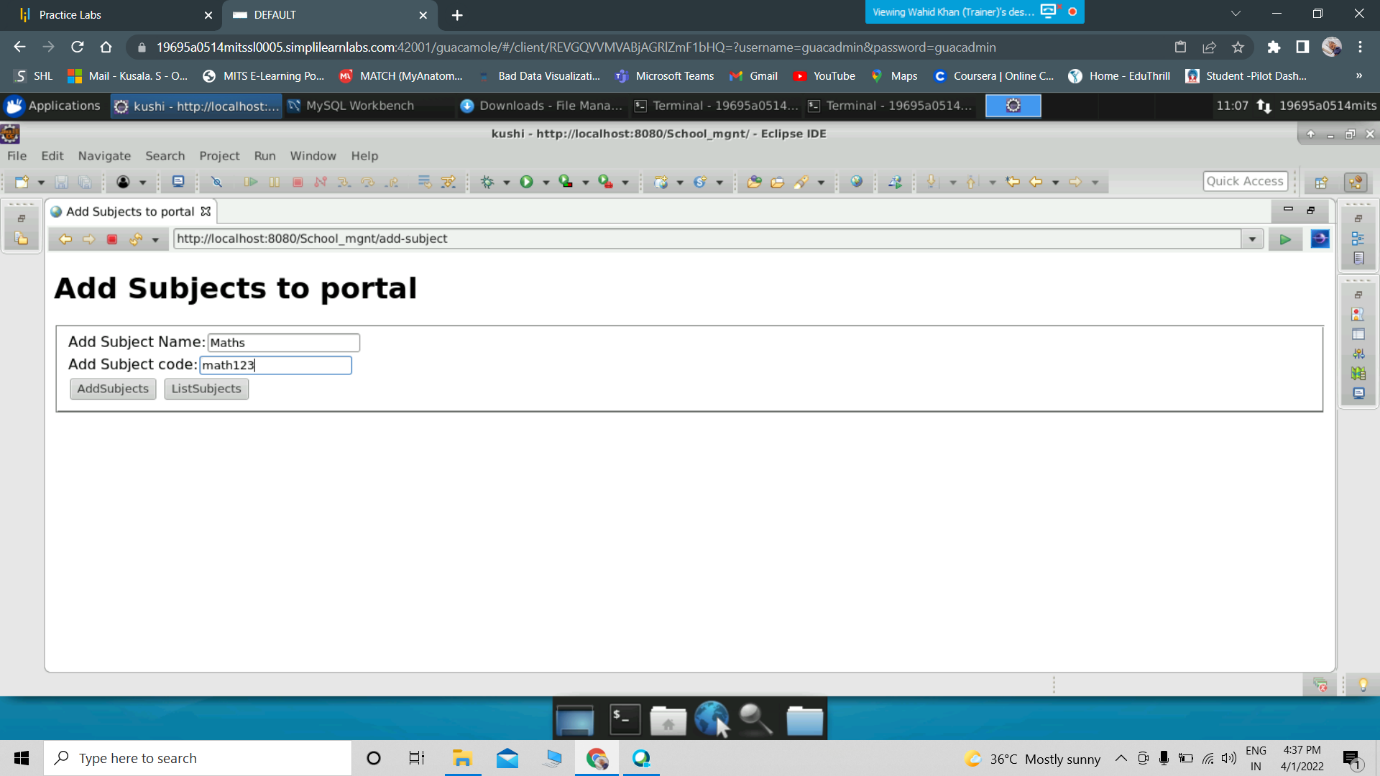
****

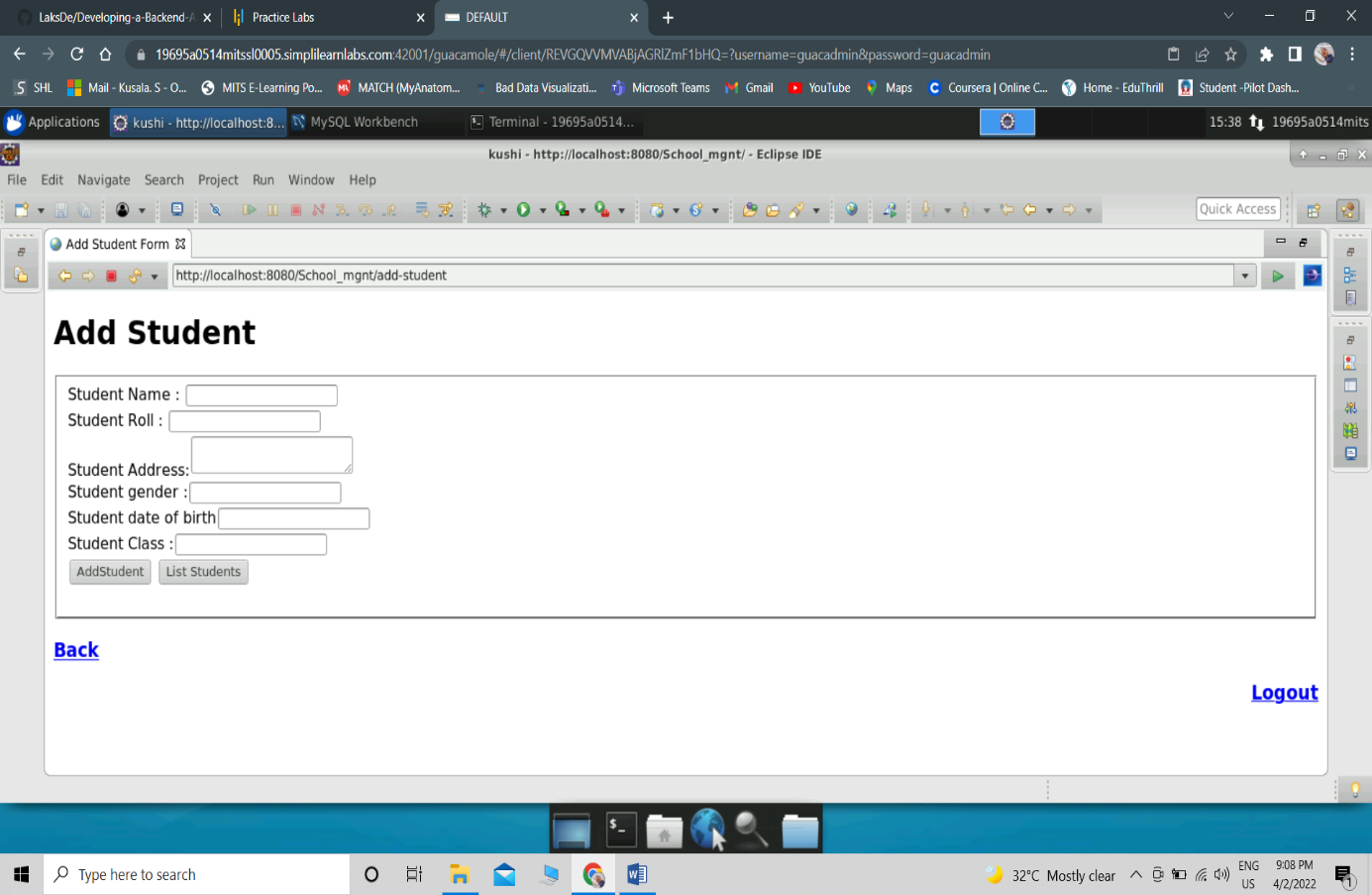
****

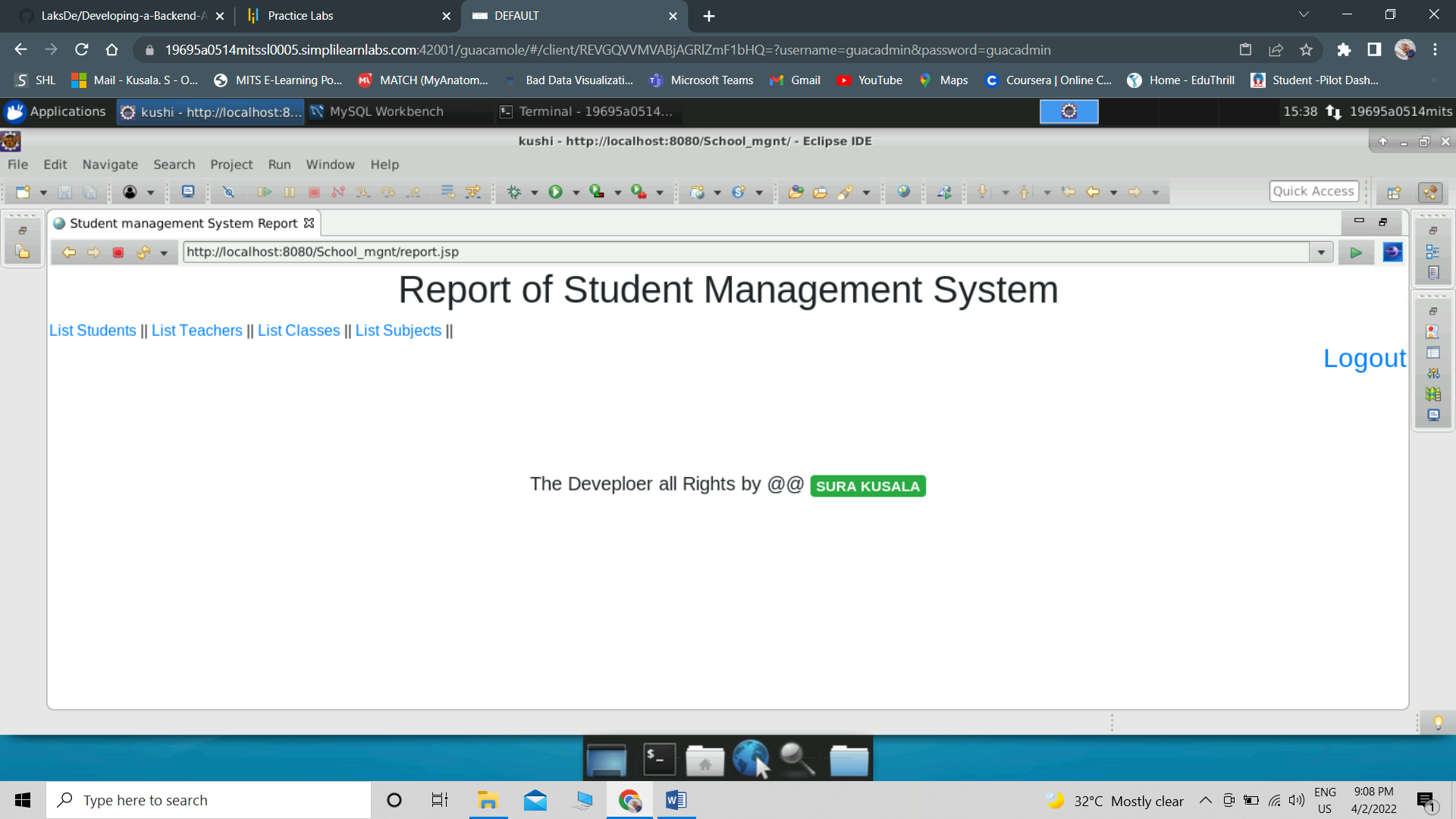
****

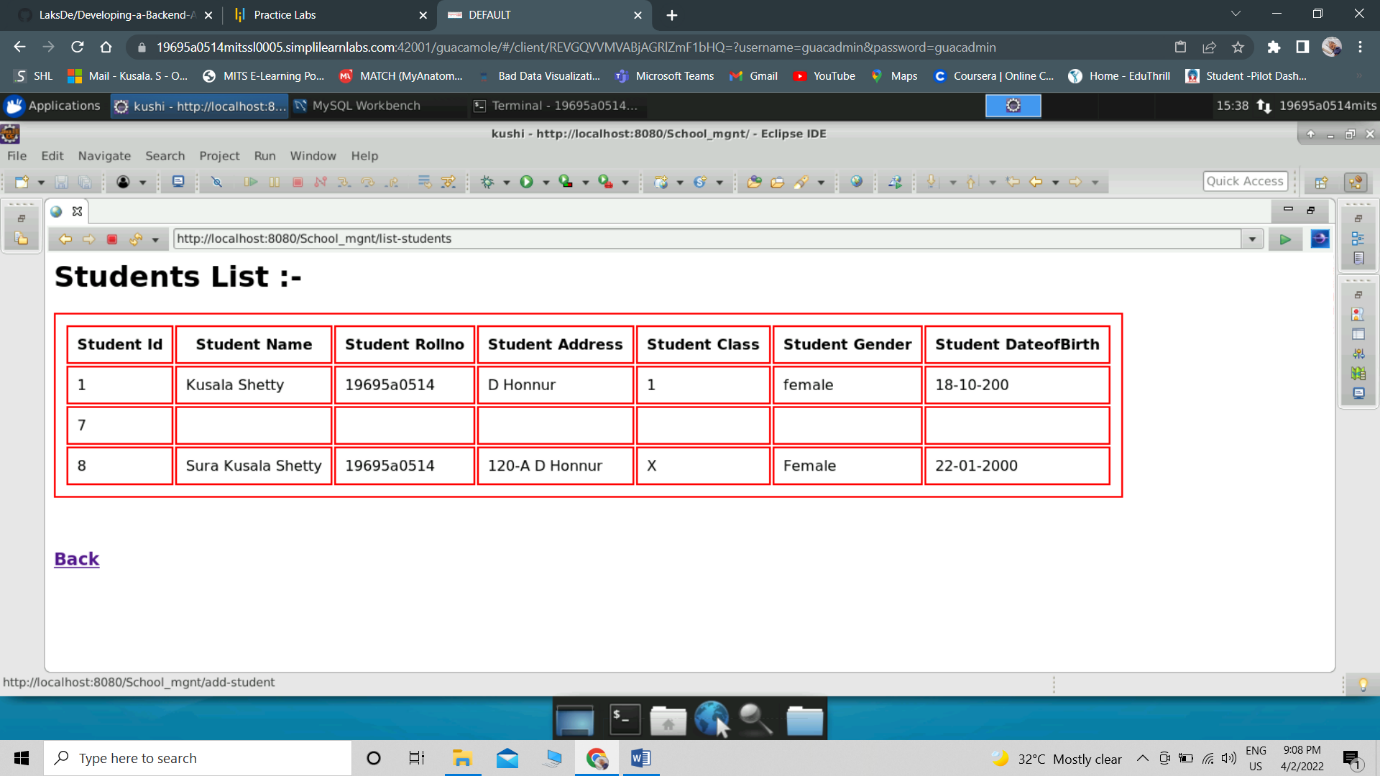
****

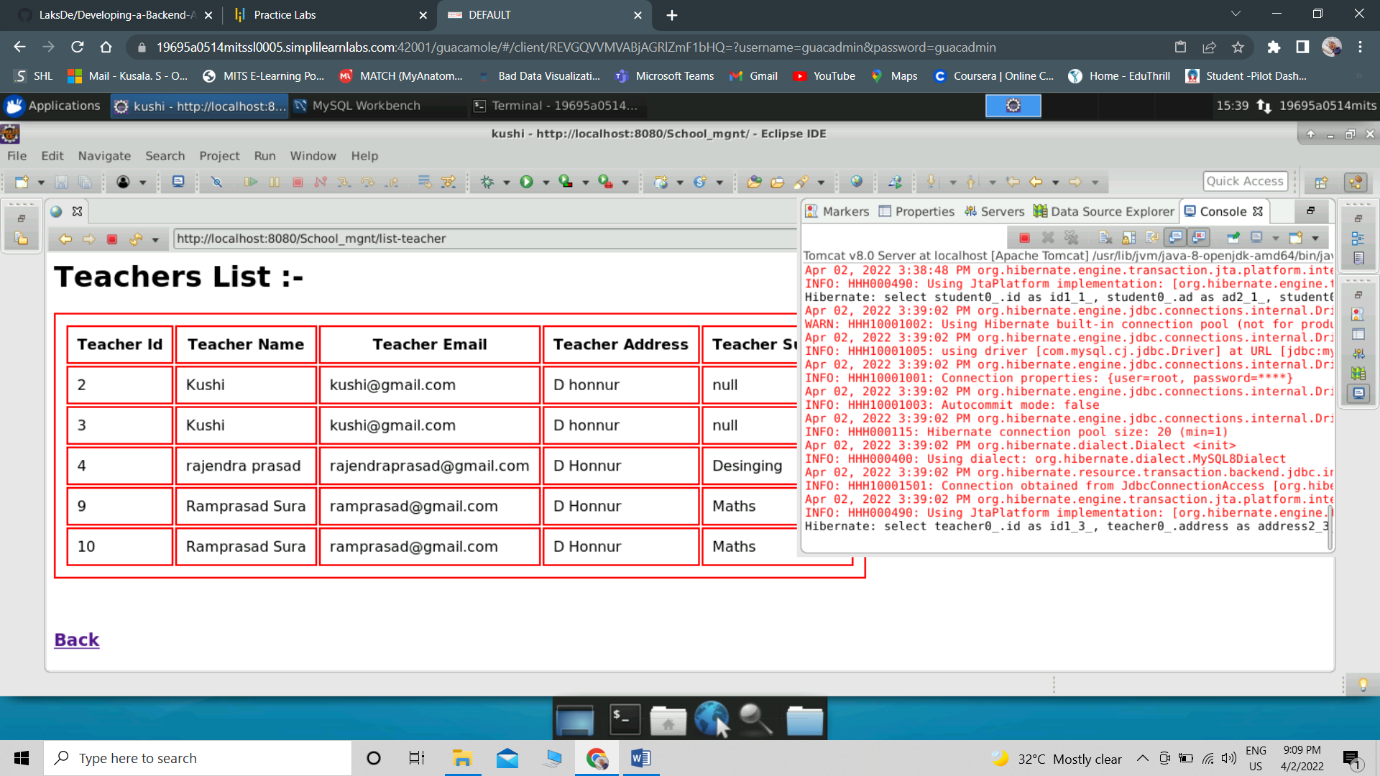
****

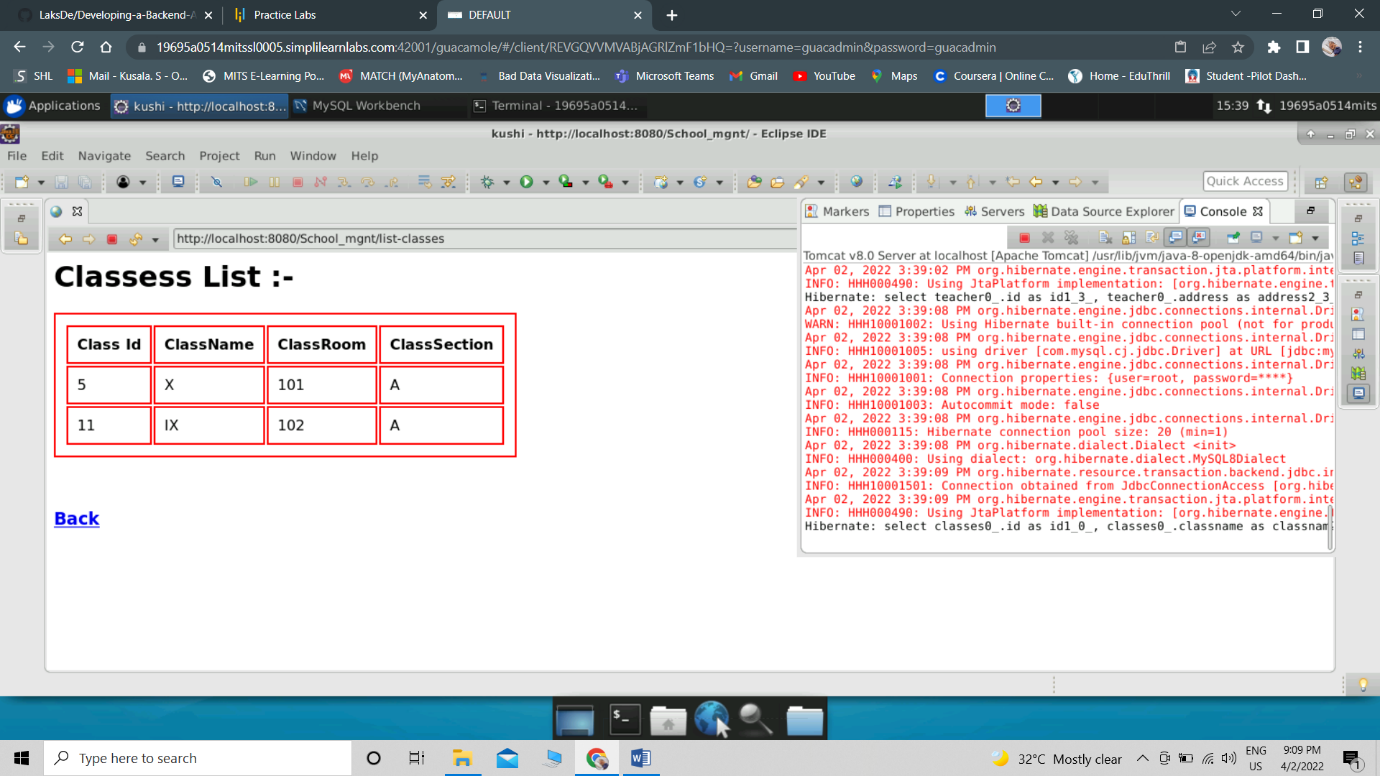
****

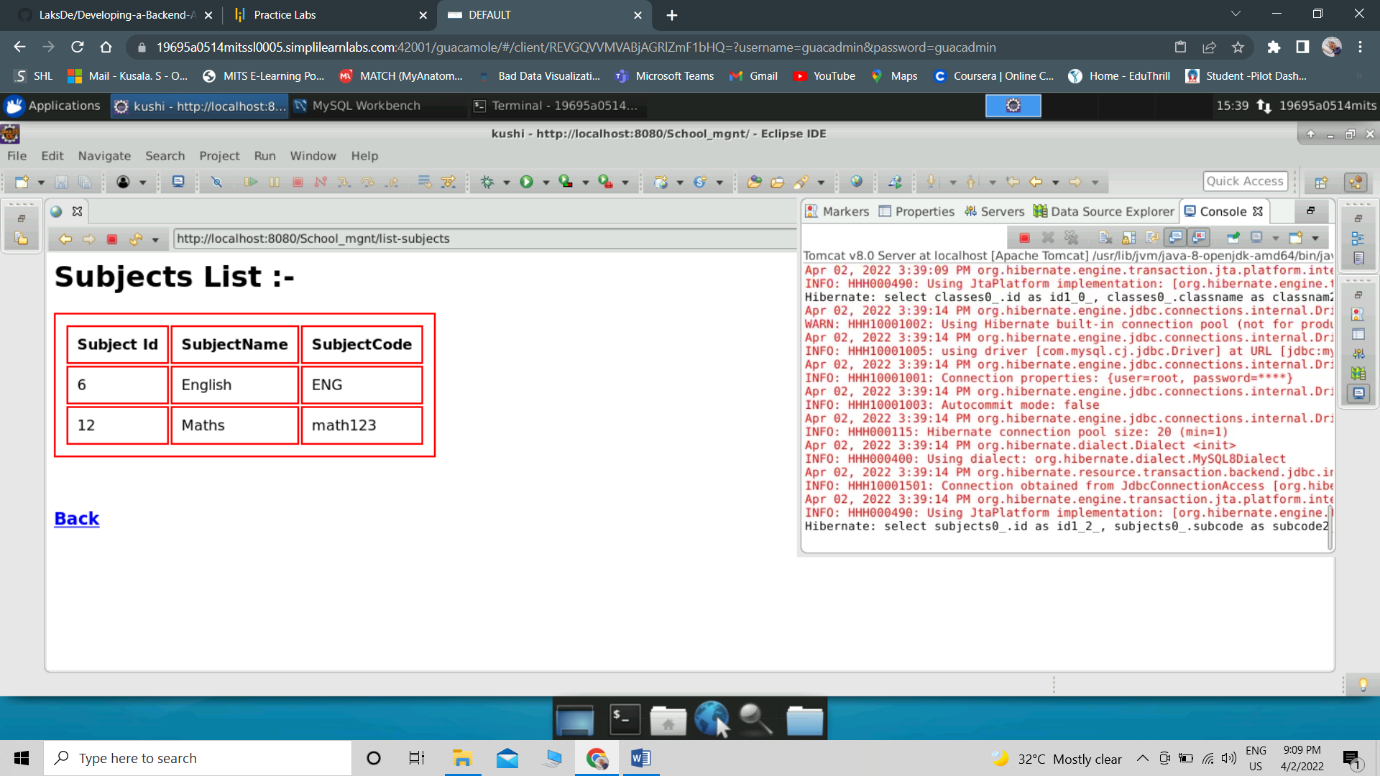
****

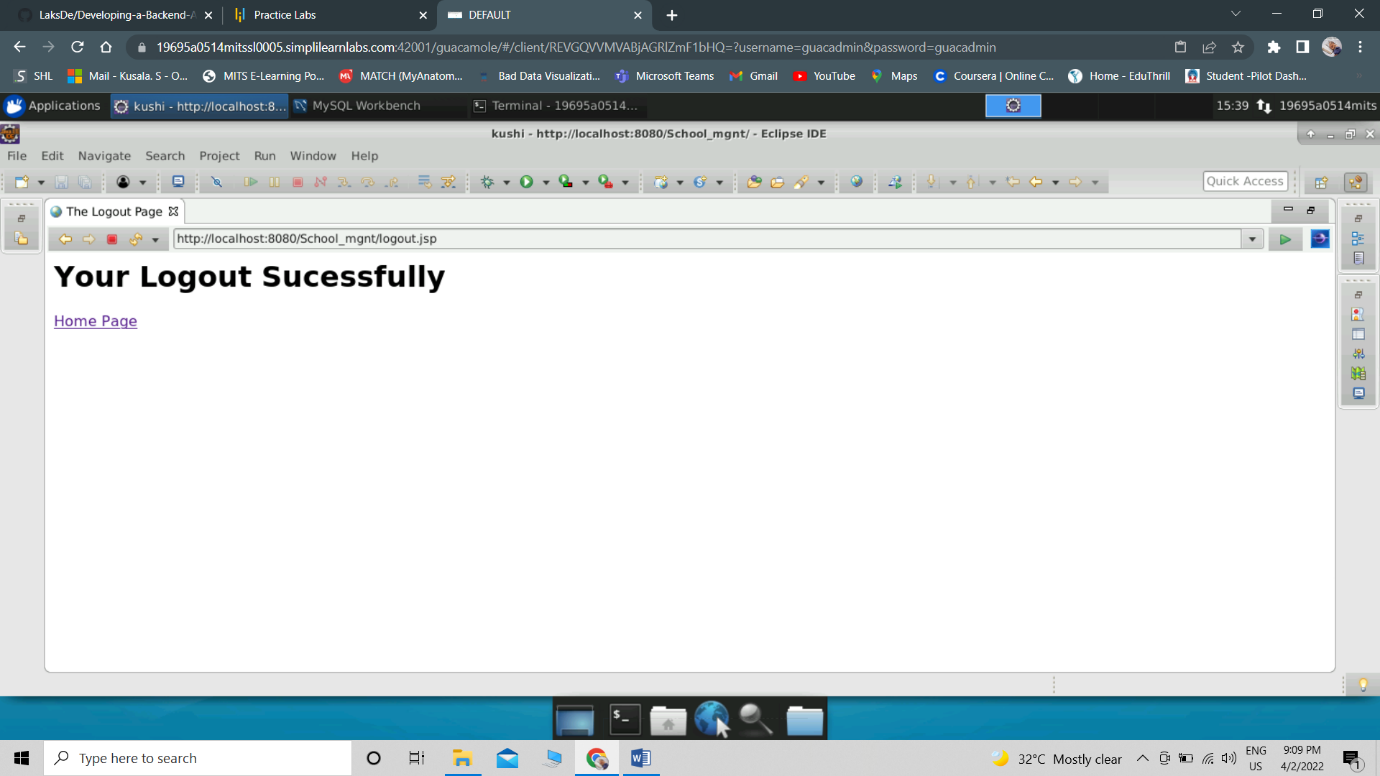
****

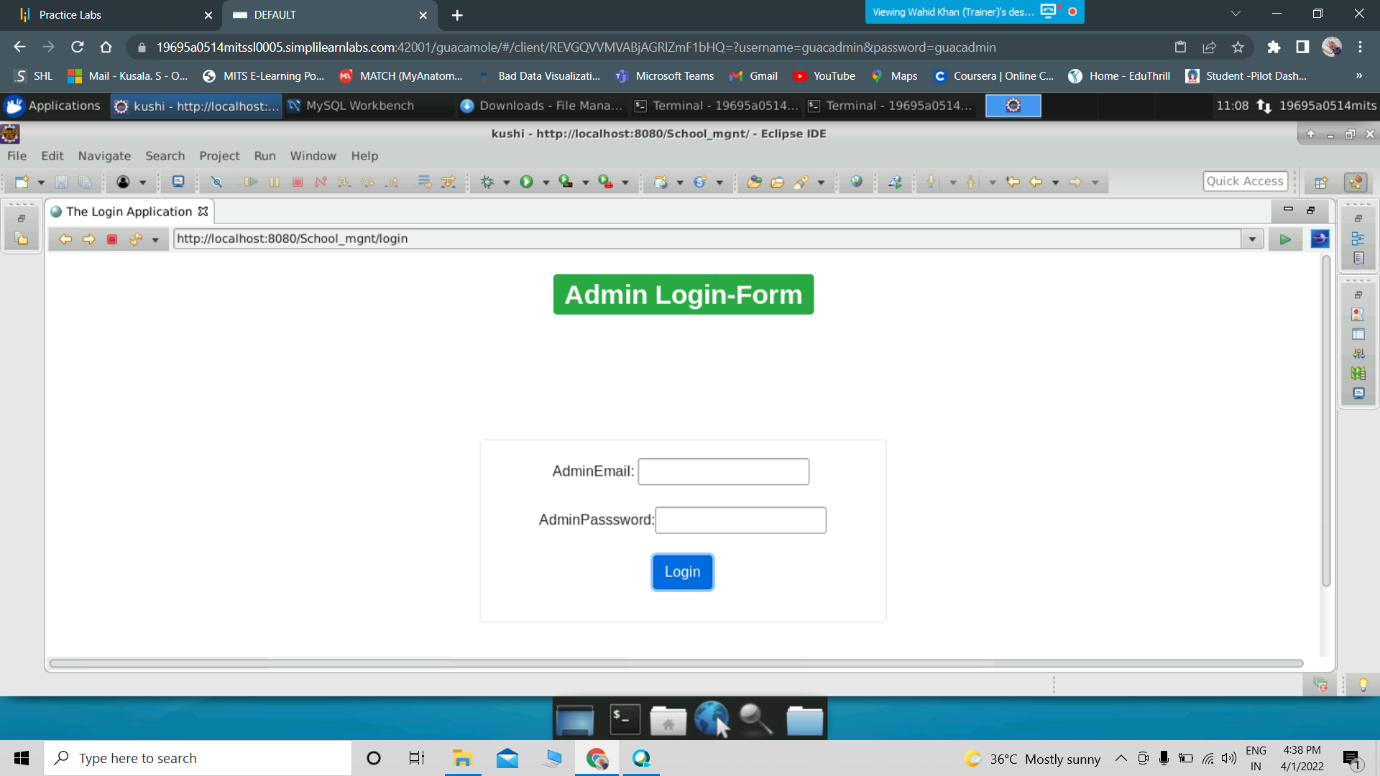
****

****

****

****

****

****