Database Systems
Assignment

Topic: Undergraduate Student Information System

Under the guidance of:
Dr. Sarat Saharia
Professor
Department of CSE
Tezpur University

Submitted by:
Pushpak Jalan (CSB17002)
Nilam Pathak (CSB17041)
B. Tech (5th Semester)
Computer Science and Engineering
Tezpur University

ACKNOWLEDGEMENT

It gives us immense pleasure to be associated with this project, which was a great learning experience. The presentation of the assignment in the way required has been made possible by the contribution of various people.

Firstly, we would like to convey my sincere thanks to Dr. Sarat Saharia, course instructor (Database Systems); whose valuable guidance helped us complete this project successfully. His suggestions and instructions have served as major contributors towards the completion of this project.

Lastly, we would like to convey our heartfelt thanks to our friends for their unfailing support and encouragement.

Thank you

Pushpak Jalan

Nilam Kumar Pathak

Contents:

Introduction:

The following report which is the part of second phase of our database application design on a Student Information System summarizes the work done so far and how it was achieved. Basically, we have designed our relational database structure based on the relational model drafted earlier. It consists of the following tables:

1. Student
2. Department
3. Programme
4. School
5. Performance
6. Academics
7. Attendance
8. Courses
9. Hostel

We have included 12 views in our web application for the user, more of which can be included as and when the need for it is felt. The views included in our database application are:

1. Individual views for all of the above table contents (9 views).
2. View for displaying every student's academic details.
3. View for displaying a particular student's attendance.
4. View for displaying a particular student's performance in previous semesters.

To achieve the above, i.e., create database along with the above tables, insert data to them, set foreign and primary keys and retrieving data from them we have used Structured Query Language. Following description includes the commands and queries used, their syntax and their example usage for our database application.

SQL Queries:

1. Create Table:

   Syntax:

   - CREATE TABLE <table_name> (
     Column1 datatype,
     Column2 datatype,
     Column3 datatype,
     .
     .
     ColumnN datatype
     );

   Example:

   - CREATE TABLE STUDENT (
     ROLL_NO VARCHAR (10) PRIMARY KEY,
     FIRST_NAME VARCHAR (20) NOT NULL,
     LAST_NAME VARCHAR (20) NOT NULL,
     SEMESTER INT (2) NOT NULL,
     EMAIL VARCHAR (50),
     PHONE_NO BIGINT (10),
     SEX VARCHAR (1),
     PROGRAMME_NAME VARCHAR (20),
     DEPARTMENT_ID VARCHAR (5),
     AGE INT (2)
     );

2. Insert Data:

   Syntax:

   - INSERT INTO <table_name> VALUES (<value_1>, <value_2>, ….. , <value_N>);

Example:

- INSERT INTO DEPARTMENT VALUES ('CIVIL', 'CIV', 7896554521, 'civ_department.tezu@ernet.in', 'ENGINEERING');

3. Alter Table (Setting Foreign Key):

   Syntax:

   - ALTER TABLE <table_name> ADD CONSTRAINT <constraint_name> FOREIGN KEY (<referencing_column_name>) REFERENCES <referenced_table> (<referenced_column_name>);

   Example:

   - ALTER TABLE STUDENT ADD CONSTRAINT FK_DEPT_ID FOREIGN KEY (DEPARTMENT_ID) REFERENCES DEPARTMENT (ID);

4. Select Table:

   Syntax:

   - SELECT <column1>, <column2>, …., <columnN> from <table_name1>, <table_name2>, ……, <table_nameN> WHERE <condition(s)> ORDER BY <column1>, <column2>, …., <columnN>;

   Examples:

   - SELECT FIRST_NAME, LAST_NAME, S.ROLL_NO AS ROLL, P.SEMESTER AS SEMESTER, SGPA, CGPA FROM STUDENT S, PERFORMANCE P WHERE S.ROLL_NO = P.ROLL_NO AND S.ROLL_NO = 'CSB17001';

   - SELECT * FROM STUDENT ORDER BY ROLL_NO;

Description of modules:

Below we try to describe in detail each and every aspect of our web database application. For better understanding we have provided the snapshots here itself.

- The below is a snapshot of our first page which provides the user with the following options:

  1. Connect to the Database.
  2. Insert Data.
  3. Display Data.

  The HTML code for this page is written in index.html file as this page is the first page of our web application and it provides an index to the overall structure of the Student Information System.

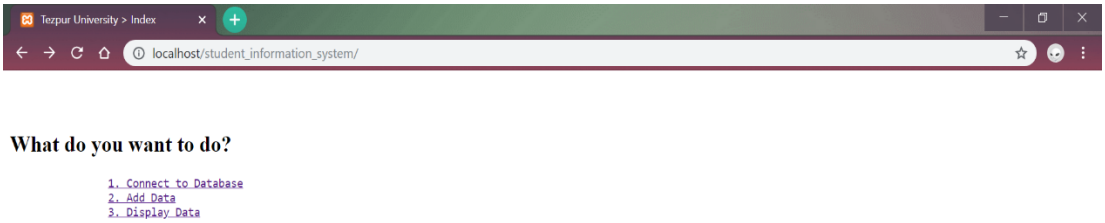


Figure 1: The Start Page.

- The connect module enables the user to connect to the MySQL database using PHP. For this we require data such as the IP address, username, password and remote database name. If all the above requirements are met correctly then the user successfully connects to the database else connection

request fails and application exits. A snapshot on successful connection with the database is as shown:

Figure 2: On successful connection.

- The 'Insert Data' module enables the database administrator to add data to the tables. In the first form, the table to which data is to be added is selected from a drop-down menu after which using PHP the user is redirected to the appropriate form for inserting data to that table. All the data corresponding to the respective fields of that table is entered, which on submission gets added to the respective database table upon meeting the required constraints else in case of some error, an error message is generated. Following are snapshots related to insertion of data:

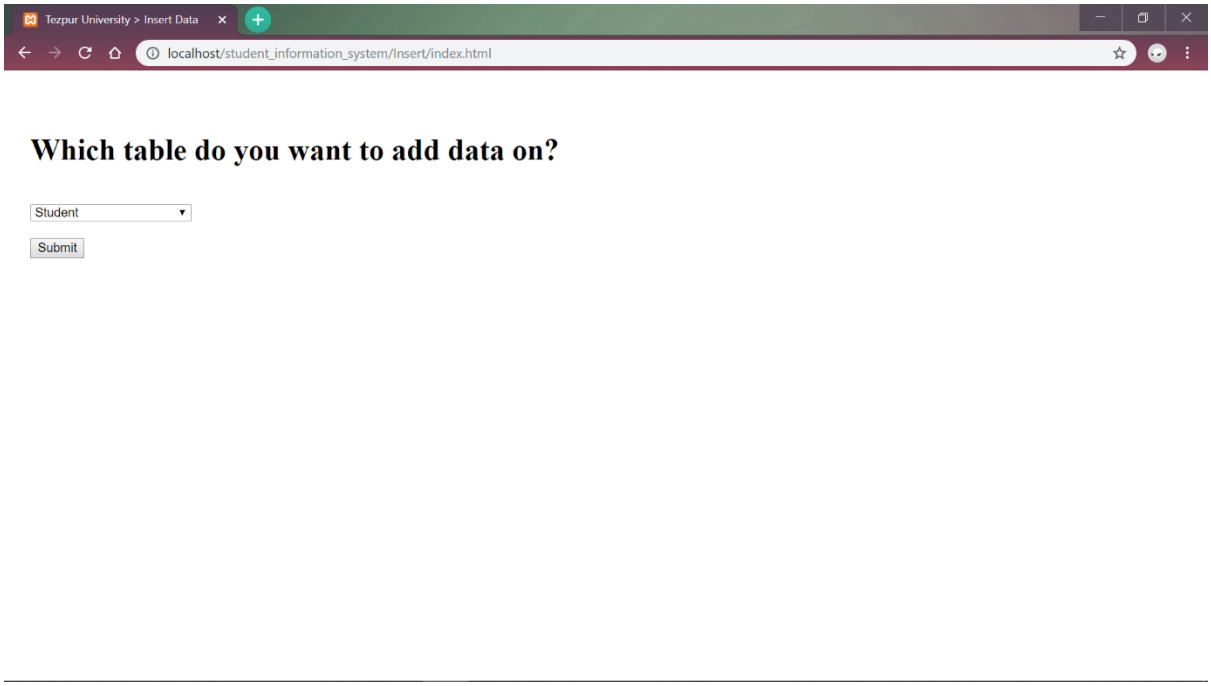


Figure 3: Insert Data > To select the table in which we want to insert data.

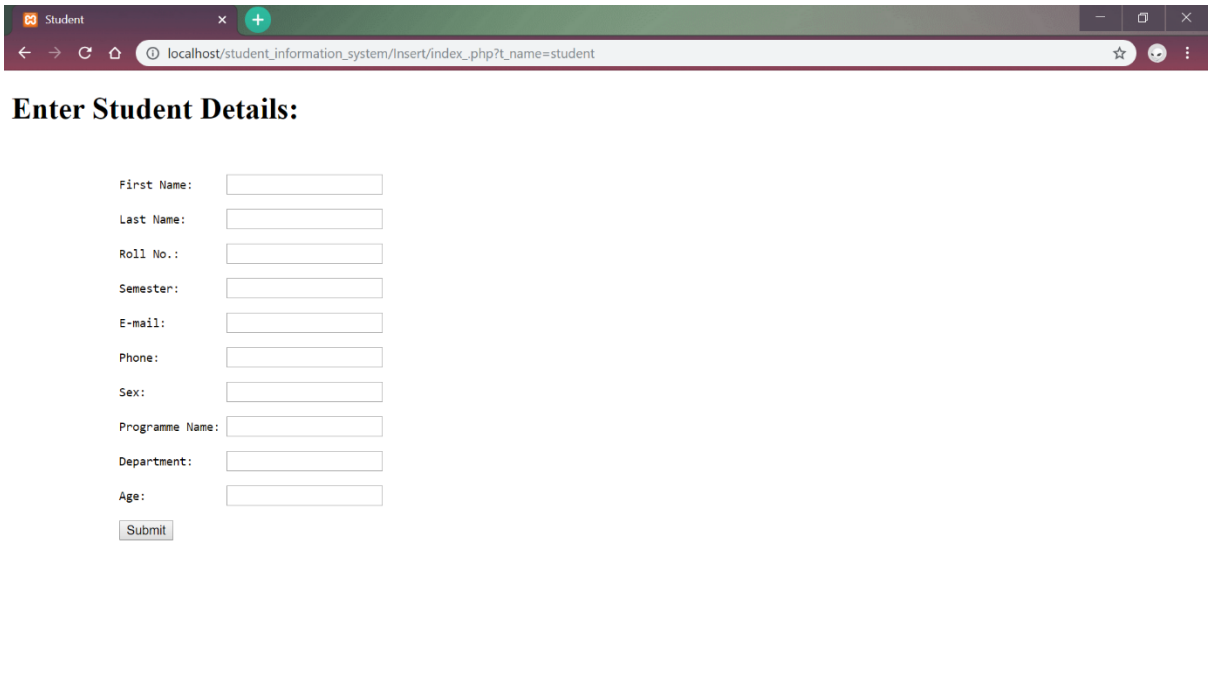


Figure 4: Insert Data > Individual table insert forms.

- The 'Display Data' module allows the database user to view the data directly stored in the tables or joined data from 2 or more tables. Data from the MySQL database is fetched and processed into tables for displaying using

PHP. As of now, we have allowed the user to only display data as per the 12 options presented to him/her on using the database application. In a later stage more views can be added as and when the need arises. The following offers a sample screenshot for displaying data from the web application:

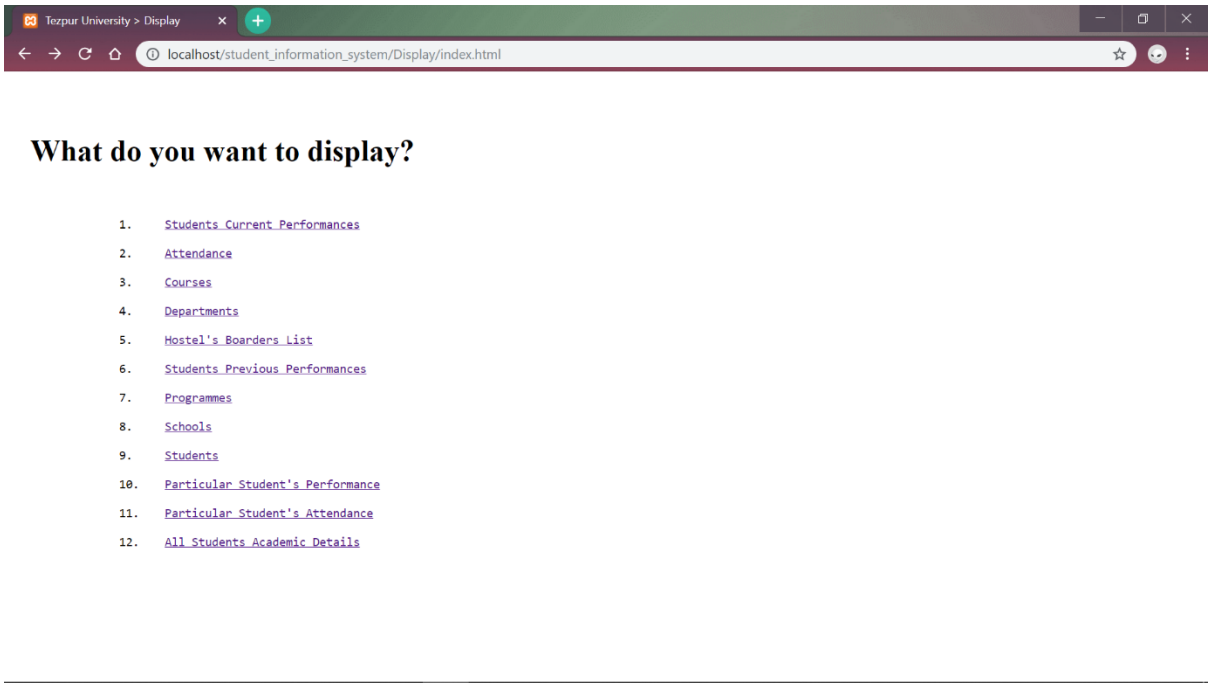


Figure 5: Display Data > Form for selecting item to display.

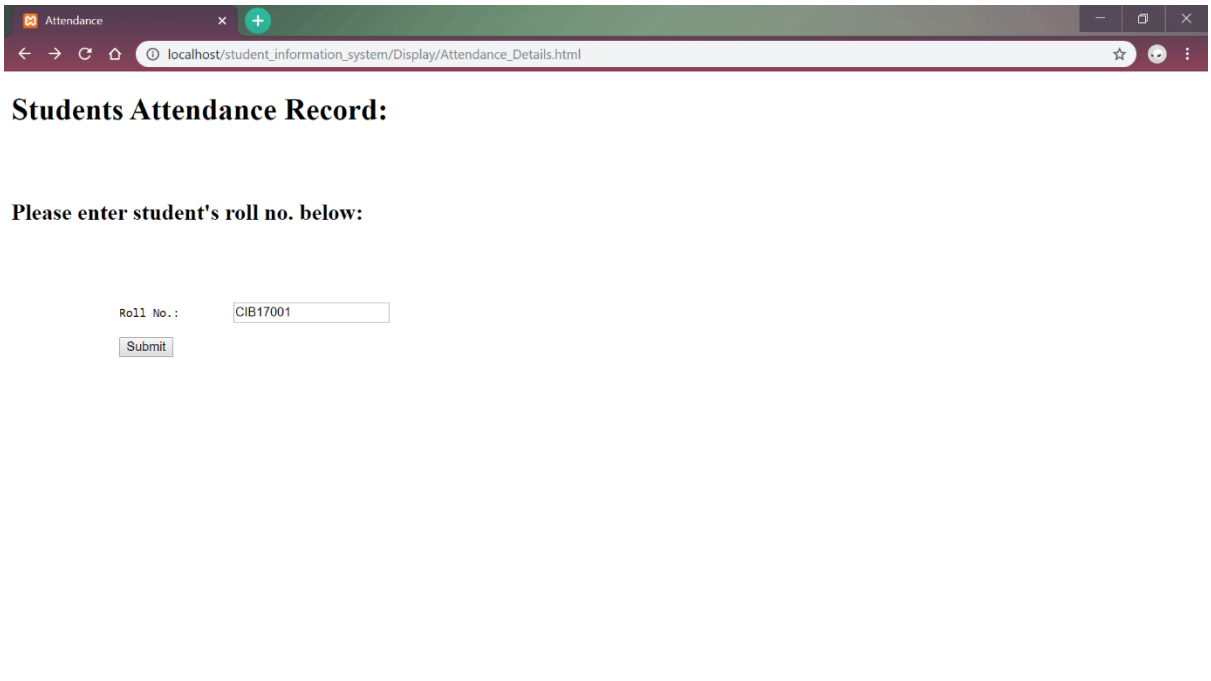
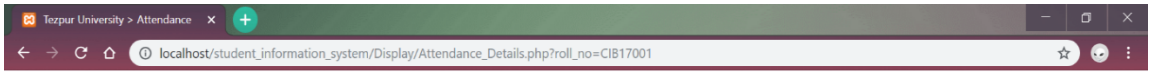


Figure 6: Display Data > Forms for displaying specific information from whole table.



Class Attendance:

Name	Roll No.	Course Code	Classes Present	Total Classes	Percentage
DEEPANKAR ACHARYA	CIB17001	BL301	21	30	70%

Figure 7: Display Data > Sample Output 1

Departments:

Name	ID	School	Contact No.	E-mail
CIVIL	CIV	ENGINEERING	7896554521	civ_departmet.tezu@ernet.in
COMPUTER SCIENCE	CSE	ENGINEERING	8011323212	cse_departmet.tezu@ernet.in
ELECTRONICS AND COMM	ECE	ENGINEERING	8790213421	ece_departmet.tezu@ernet.in
FOOD ENGINEERING	FET	ENGINEERING	9435052955	fet_engineering@tezu.ernet.in
MATHEMATICS	MATH	SCIENCE	7678901212	math_departmet.tezu@ernet.in
PHYSICS	PHY	SCIENCE	9957216754	phy_departmet.tezu@ernet.in

Figure 8: Display Data > Sample Output 2

Conclusion:

This project provided us with immense knowledge about HTML, PHP and SQL. With deeper insights gained we are now able to grasp complicated database concepts easily and effectively. Also, now as we have learned the basic database concepts using a practical approach by designing a web interface for our Student Information System, we can extend our ideas to develop this project into a more sophisticated and useful one so that it could be implemented for use in the universities. With the knowledge gained we can also implement other such useful systems with database at its back for reliable and efficient data storage and retrieval for use in day to day lives.