Instructions:

1. Follow the handout step by step. Take pictures of your work.
2. Create a folder named Lab 7 on your server for this lab
3. Create a page for each section.
4. Link all pages properly.
5. User friendly

Task I. Database Design(10%)

Design a MySQL database "Registrations" that contains tables:students, Classes, Schedule as what follows:

```
+-------------------------+
| Tables_in_registrations |
+-------------------------+
| Classes                 |
| Schedule                |
| students                |
+-------------------------+
```

Students table(3%):

```
+-------+-------------+------+-----+---------+----------------+
| Field | Type        | Null | Key | Default | Extra          |
+-------+-------------+------+-----+---------+----------------+
| SID   | int(11)     | NO   | PRI | NULL    | auto_increment |
| Name  | varchar(40) | NO   |     | NULL    |                |
+-------+-------------+------+-----+---------+----------------+
```

Classes table(3%):

```
+-------+-------------+------+-----+---------+----------------+
| Field | Type        | Null | Key | Default | Extra          |
+-------+-------------+------+-----+---------+----------------+
| CID   | int(11)     | NO   | PRI | NULL    | auto_increment |
| Name  | varchar(40) | NO   |     | NULL    |                |
+-------+-------------+------+-----+---------+----------------+
```

Schedule table(4%):

```
+--------+-------------+------+-----+---------+----------------+
| Field  | Type        | Null | Key | Default | Extra          |
+--------+-------------+------+-----+---------+----------------+
| EID    | int(11)     | NO   | PRI | NULL    | auto_increment |
| SNAME  | varchar(40) | NO   |     | NULL    |                |
| Class1 | varchar(40) | NO   |     | NULL    |                |
| Class2 | varchar(40) | YES  |     | NULL    |                |
| Class3 | varchar(40) | YES  |     | NULL    |                |
+--------+-------------+------+-----+---------+----------------+
```

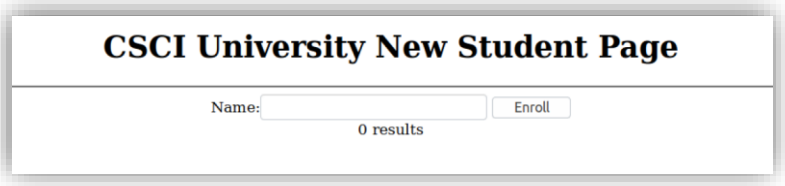
Task 2. UI Design(40%)

Create a CSCI332 university website that allows students to enroll classes through the website. Typically, only system administrations are able to add students and classes to the University's enrollment database, but to simplify your tasks, we are not going to implement authentication mechanisms to the website.

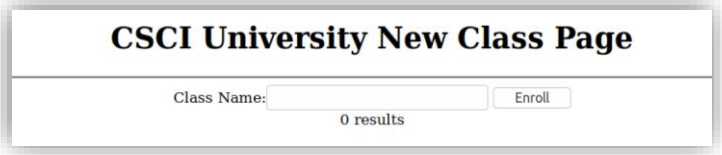
In this website, you are going to offer the following pages that will:

1. Front page as what follows:

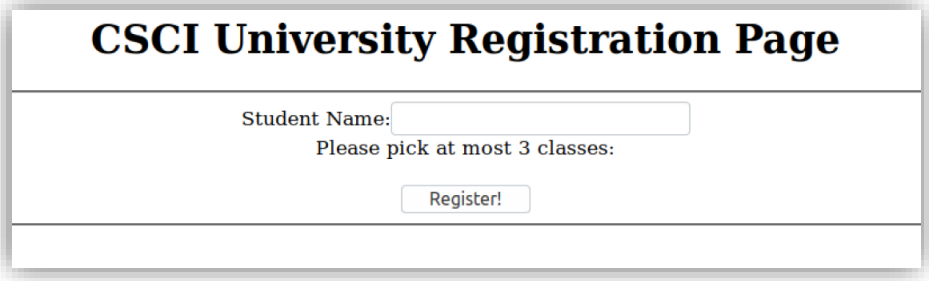
2. New Student Page: enroll new students that looks like what follows:



3. New Class Page: enroll new classes that looks like what follows.



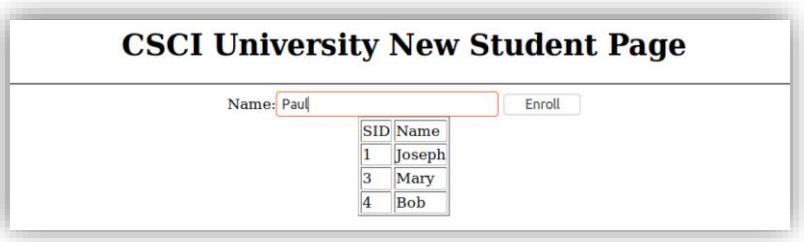
4. Registration Page: Enroll students to classes that looks like what follows.



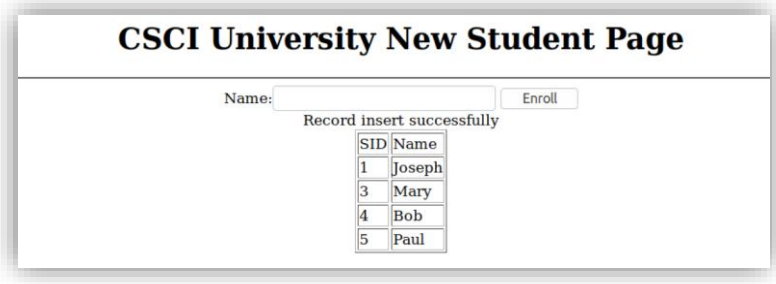
Task 3. Server-Side Programming(50%)

Create server-side PHP code for each page that is able to do what describes below.

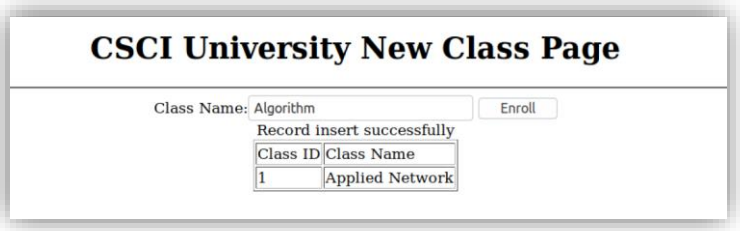
1. (15%)Allow users to add new students to students table and list all students as what follows(We assume no duplicate names will be inserted into the database):
   Before insert a new student:



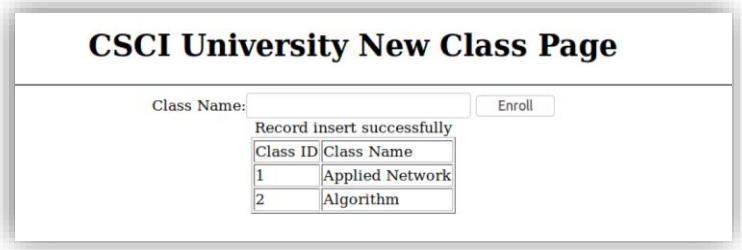
   After the new student is inserted:



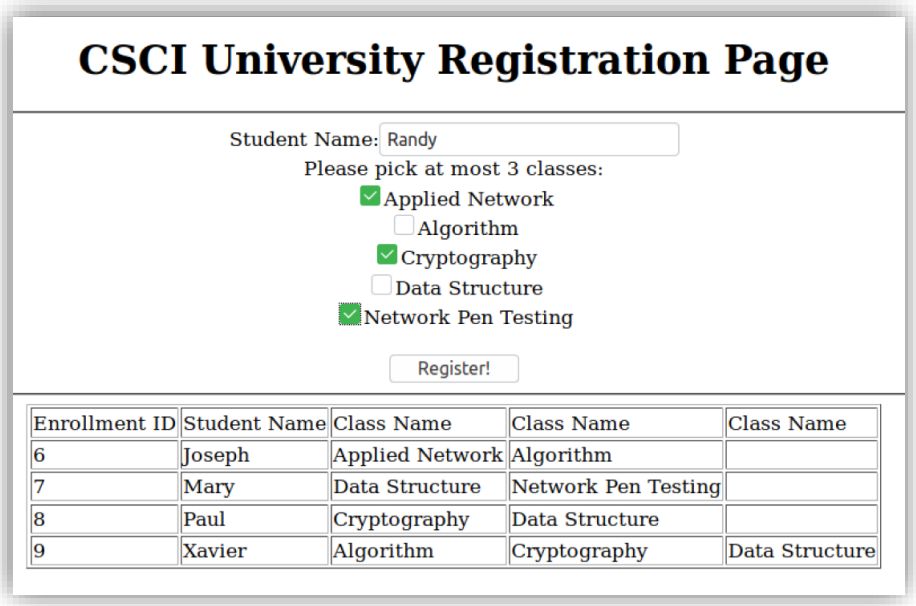
2. (15%)Allow users to add new Classes to Classes table and list all Classess as what follows(We assume no duplicate classes will be inserted into the database):
   Before insert a new class:



   After the new class is inserted:

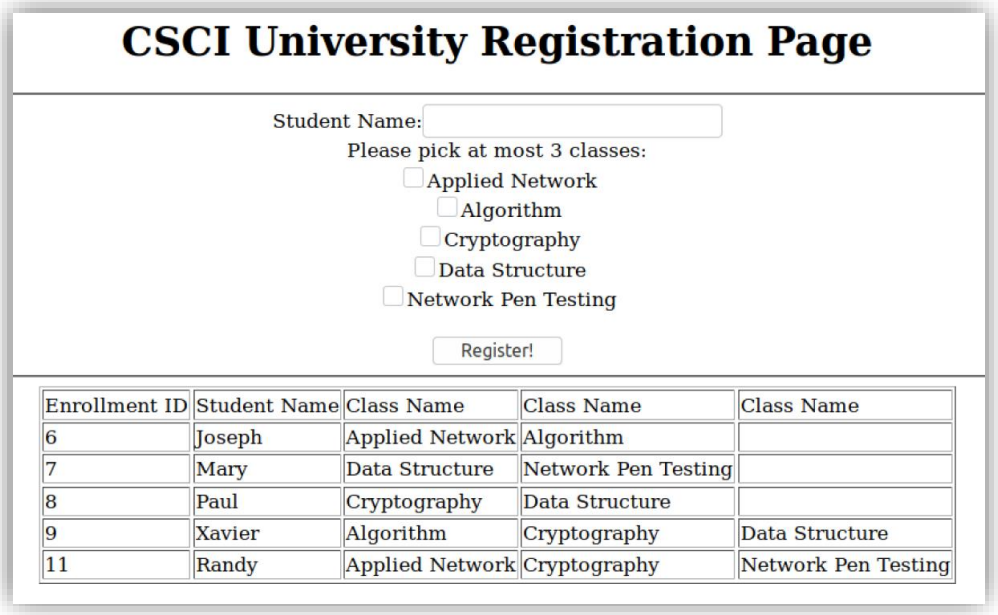


3. (20%)Allow users to register for Classes and list all classes registered for each student as what follows. When the same student changed his/her schedule, your code will reflect the changes. The list of classes depends on classes added in class page.:
Before a student register for a class:



Enrollment ID	Student Name	Class Name	Class Name	Class Name
6	Joseph	Applied Network	Algorithm	
7	Mary	Data Structure	Network Pen Testing	
8	Paul	Cryptography	Data Structure	
9	Xavier	Algorithm	Cryptography	Data Structure

After enrollment:

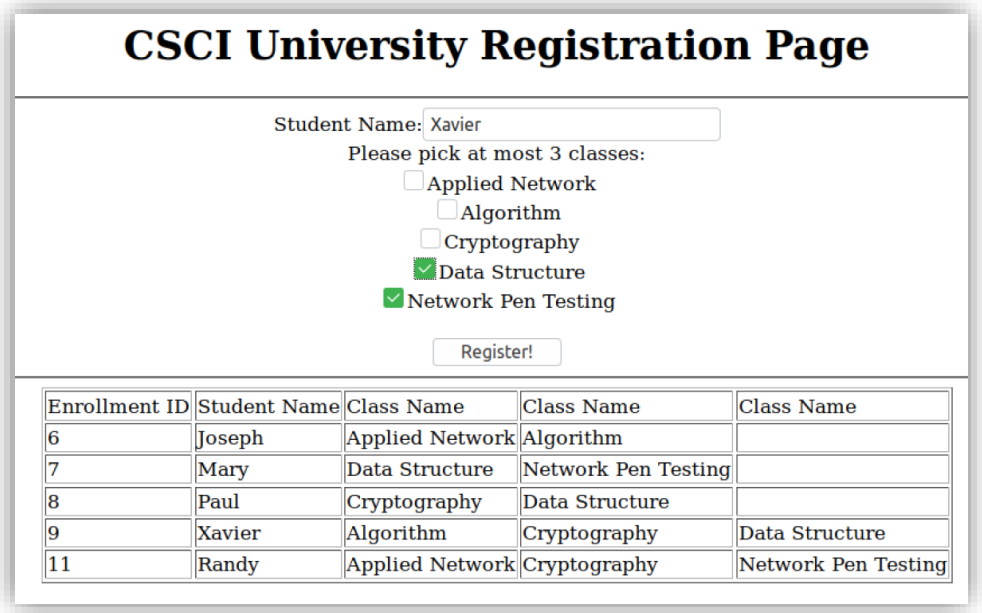


Enrollment ID	Student Name	Class Name	Class Name	Class Name
6	Joseph	Applied Network	Algorithm	
7	Mary	Data Structure	Network Pen Testing	
8	Paul	Cryptography	Data Structure	
9	Xavier	Algorithm	Cryptography	Data Structure
11	Randy	Applied Network	Cryptography	Network Pen Testing

The Enrollment ID will automatically increase. If there is a record deleted from the table, there will be a skip in ID.

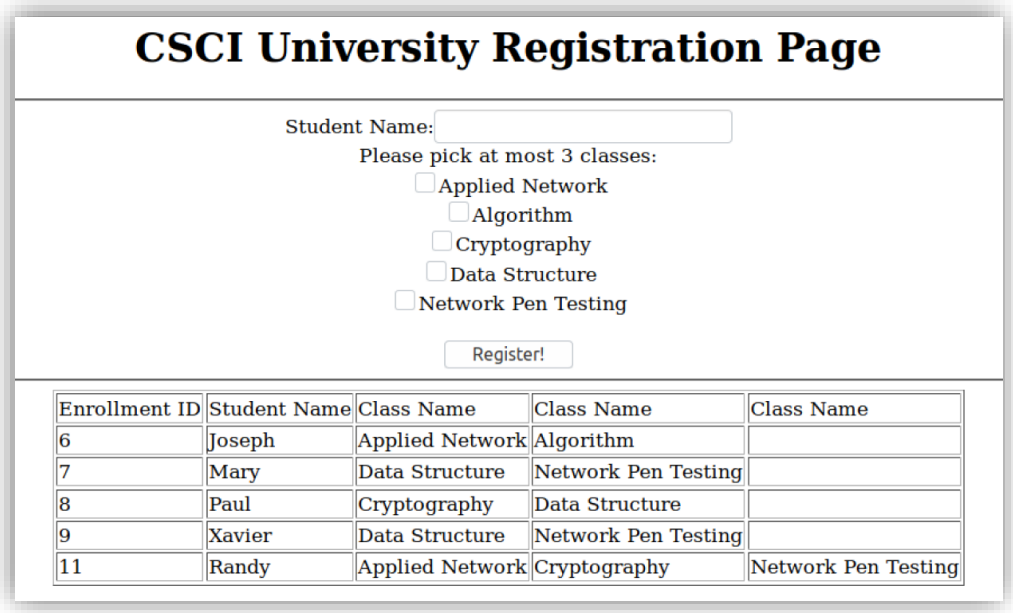
If an existing record is entered, your server-side code will UPDATE the record instead of creating a duplicate record.

Before Update:



Enrollment ID	Student Name	Class Name	Class Name	Class Name
6	Joseph	Applied Network	Algorithm	
7	Mary	Data Structure	Network Pen Testing	
8	Paul	Cryptography	Data Structure	
9	Xavier	Algorithm	Cryptography	Data Structure
11	Randy	Applied Network	Cryptography	Network Pen Testing

After update:



CSCI University Registration Page

Student Name:

Please pick at most 3 classes:

- Applied Network
- Algorithm
- Cryptography
- Data Structure
- Network Pen Testing

Register!

Enrollment ID	Student Name	Class Name	Class Name	Class Name
6	Joseph	Applied Network	Algorithm	
7	Mary	Data Structure	Network Pen Testing	
8	Paul	Cryptography	Data Structure	
9	Xavier	Data Structure	Network Pen Testing	
11	Randy	Applied Network	Cryptography	Network Pen Testing

What to deliver?

1. (10%) Demonstrate using a command prompt to show databases, tables and columns.
2. (40%) Demonstrate UI of each page.
3. (50%) Demonstrate inserting records to each page.