STUDENT ENROLLMENT MANAGEMENT SYSTEM

Final Project Report

ISM6218.001F22.88140 - Advanced Database Management

Submitted by

Manoj Kumar Goud Sowdari	U54666573
Sai Swetha Somi	U04880400
Prathyusha Hari	U43145266
Sai Prakash Madderla	U41306017

Major in

BUSINESS ANALYTICS AND INFORMATION SYSTEM

Under the guidance of

Associate Professor
DONALD BERNDT

MUMA COLLEGE OF BUSINESS

UNIVERSITY OF SOUTH FLORIDA

Table of Contents

1. INTRODUCTION

1.1 Summary:

In the developing countries like India, the education system is not yet modernized as compared to developed countries in the west. In most of the schools and universities, the enrollment of the classes is not flexible to the students and is mostly done by the school management. When given a choice of various subjects, the student can easily enroll as per their interests. The motivation of this project is to create a webpage which can be used by students in these schools and universities for the ease of enrollment of the classes.

Benefits of Online Enrollment Administration Systems

- A web enrollment framework can collect all the data into a central center, making it simple to get to and modify.
- Enlistment data is right away included to the database without the ought to enter the data into a registry by hand.
- This drastically speeds up the enrollment handle, liberating up representatives to do the assignments that must be done by hand.
- You increment exactness by cutting down on the number of data-entry mistakes, such as untrue addresses and moved understudies.
- Give portable enrollment get to to families without a computer or broadband web access.
- Bolster the dialects talked in your community, guaranteeing get to for families whose to begin with dialect isn't English.
- Oversee understudy enrollment, enrollment, and re-registration with an easy-to-use, computerized platform.
- Spare you the time and cash related with a conventional pen-and-paper enrollment prepare.

1.2 Business Requirements:

- Student should be able to create an account and sign in with their credentials.
- They can look for the course details, instructor details, and campus details.
- Student should be able to enroll for the classes and get the status of their registration.

1.3 Entities with Attributes:

CAMPUS

campus_code
campus_name
location
capacity

BUILDING_ADDRESS

location
street
province
country
pincode

STUDENT

Student_id
first_name
last_name
age
gender
email
password
campcode
coursed

COURSE

course_id
course_name
course_duration
credits
instructorid

INSTRUCTOR

Instructor_ID

name

edge

gender

department

INSTRUCTOR_ADDRESS

inst_add

inst_id

street

city

country

pincode

SUBJECT DETAILS

course_ID_

course_name

dept_id_

deptname

DEPARTMENT DETAILS

Dept_id

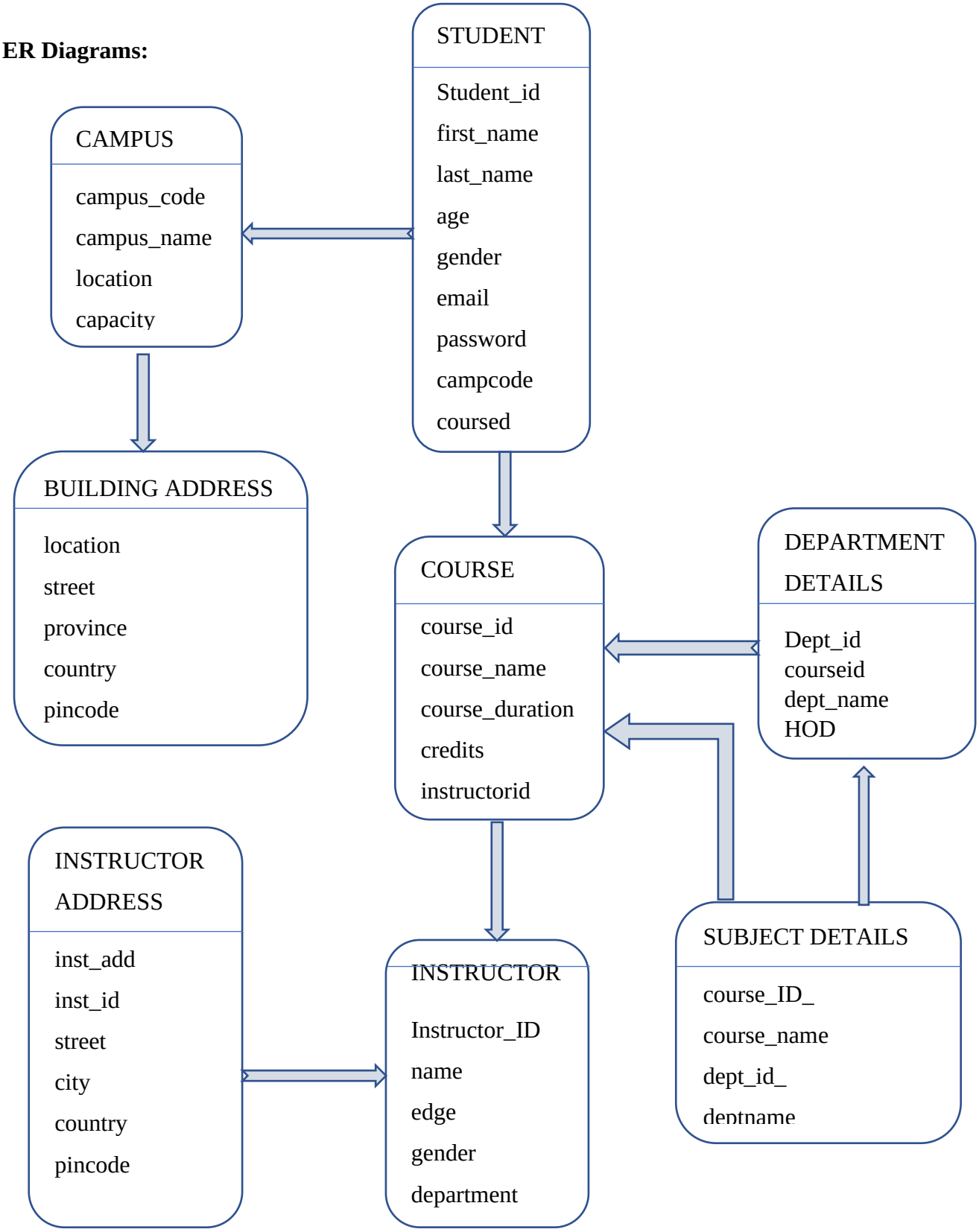
courseid

dept_name

HOD

2. DATABASE DESIGN

2.1 ER Diagrams:



2.2 Data Generation and Loading

For tables with few numbers of rows, data is loaded using INSERT INTO command. For the table STUDENT, the data cannot be entered through INSERT INTO command. For this table, the data is taken from an excel file which contains huge number of rows of student details. We use Text editor software like Sublime Text Editor for generating the sql command.

1. CREATING TABLE FOR student

```
CREATE TABLE student (
  student_id int NOT NULL,
  first_name varchar(45) NOT NULL,
  last_name varchar(45) NOT NULL,
  age int NOT NULL,
  gender varchar(45) NOT NULL,
  email varchar(45) NOT NULL,
  password varchar(45) NOT NULL,
  campcode varchar(45) NOT NULL,
  courseid varchar(45) NOT NULL,
);
```

Auto increment:

```
create sequence student_id minvalue 1 start with 1 cache 10;
```

Loading data for TABLE STUDENT:

Below is the snapshot of the data from the excel file.

student_id	first_name	last_name	age	gender	email	password	campcode	courseid
1101	'Lena'	'Smith'	25	'F'	'lena.smith@gmail.com'	'lena123456'	'uni101'	'CS 01'
1102	'Nicol'	'Green'	24	'M'	'nicol.green@gmail.com'	'nicol123456'	'uni101'	'CS 02'
1103	'Tom'	'Taylor'	26	'M'	'tom.taylor@gmail.com'	'tom123456'	'uni101'	'CS 03'
1104	'Paul'	'Miler'	22	'M'	'paul.miler@gmail.com'	'paul123456'	'uni101'	'CS 04'
1105	'Alen'	'Lee'	26	'F'	'alen.lee@gmail.com'	'alan123456'	'uni101'	'CS 05'
1106	'David'	'King'	23	'M'	'david.king@gmail.com'	'david123456'	'uni101'	'Math 01'
1107	'Omar'	'Wood'	22	'M'	'omar.wood@gmail.com'	'omar123456'	'uni101'	'Math 02'
1108	'Jone'	'Sith'	25	'F'	'jone.smith@gmail.com'	'jone123456'	'uni101'	'Math 03'
1109	'Lucy'	'White'	24	'F'	'lucy.white@gmail.com'	'lucy123456'	'uni101'	'Math 04'
1110	'Eren'	'Forger'	23	'M'	'eren.forger@gmai.com'	'eren123456'	'uni101'	'Math 05'
1111	'tiny'	'rodger'	22	'Male'	'tiny.rodger@gmail.com'	'tiny123456'	'uni101'	'CS 04'
1112	'elli'	'oslen'	22	'Female'	'elli.oslen@gmail.com'	'elli123456'	'uni105'	'Math 04'
1113	'Mark'	'Anderson'	22	'Male'	'mark.anderson@gmail.com'	'mark123456'	'uni104'	'CS 03'
1114	'Ramon'	'Davis'	26	'Male'	'ramon.davis@gmail.com'	'ramon123456'	'uni103'	'CS 05'
1115	'Debra'	'Perry'	29	'Male'	'debra.perry@gmail.com'	'debra123456'	'uni101'	'CS 02'
1116	'Sabrina'	'Moody'	24	'Male'	'sabrina.moody@gmail.com'	'sabrina123456	'uni103'	'CS 01'
1117	'Seth'	'Scott'	21	'Male'	'seth.scott@gmail.com'	'seth123456'	'uni102'	'CS 05'
1118	'Dan'	'Richards'	29	'Female'	'dan.richards@gmail.com'	'dan123456'	'uni101'	'CS 04'
1119	'Chrysta'	'Mitchell'	26	'Female'	'chrysta.mitchell@gmail.com'	'chrysta123456	'uni102'	'CS 02'
1120	'Sabrina'	'Thomas'	28	'Female'	'sabrina.thomas@gmail.com'	'sabrina123456	'uni101'	'Math 04'
1121	'Lucy'	'Nelson'	30	'Male'	'lucy.nelson@gmail.com'	'lucy123456'	'uni104'	'Math 03'
1122	'Tracey'	'Norman'	24	'Male'	'tracey.norman@gmail.com'	'tracey123456'	'uni103'	'CS 03'
1123	'Ivan'	'Walker'	21	'Female'	'ivan.walker@gmail.com'	'ivan123456'	'uni105'	'Math 02'
1124	'Ramon'	'Moody'	28	'Female'	'ramon.moody@gmail.com'	'ramon123456'	'uni104'	'CS 04'
1125	'Ethan'	'Potter'	25	'Female'	'ethan.potter@gmail.com'	'ethan123456'	'uni104'	'CS 03'

Sheet1

Ready Accessibility: Good to go Average: 1351 Count: 503

Step 1: Convert .xlsx to .csv file.

Step 2: Open the .csv file in the Sublime text editor

Step 3: Place multiple cursors in the beginning of every line using CTRL + A then CTRL + SHIFT+L then press HOME button.

Step 4: Type the sql code:

INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(

Similarly place the multiple cursor at the end of each line and enter closing bracket and a paranthesis.

);

The data looks as shown in the fig below:

INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1102, 'Nicol', 'Green', 24, 'M', 'nicol.green@gmail.com', 'nicol123456', 'uni101', 'CS 02');

C:\Users\manoj_nuhq5\Desktop\student details.csv • - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

student details.csv

```csv
1101, 'Lena', 'Smith',25, 'F', 'lena.smith@gmail.com', 'lena123456', 'uni101', 'CS 01'
1102, 'Nicol', 'Green',24, 'M', 'nicol.green@gmail.com', 'nicol123456', 'uni101', 'CS 02'
1103, 'Tom', 'Taylor',26, 'M', 'tom.taylor@gmail.com', 'tom123456', 'uni101', 'CS 03'
1104, 'Paul', 'Miler',22, 'M', 'paul.miler@gmail.com', 'paul123456', 'uni101', 'CS 04'
1105, 'Alen', 'Lee',26, 'F', 'alen.lee@gmail.com', 'alan123456', 'uni101', 'CS 05'
1106, 'David', 'King',23, 'M', 'david.king@gmail.com', 'david123456', 'uni101', 'Math 01'
1107, 'Omar', 'Wood',22, 'M', 'omar.wood@gmail.com', 'omar123456', 'uni101', 'Math 02'
1108, 'Jone', 'Sith',25, 'F', 'jone.smith@gmail.com', 'jone123456', 'uni101', 'Math 03'
1109, 'Lucy', 'White',24, 'F', 'lucy.white@gmail.com', 'lucy123456', 'uni101', 'Math 04'
1110, 'Eren', 'Forger',23, 'M', 'eren.forger@gmai.com', 'eren123456', 'uni101', 'Math 05'
1111, 'tiny', 'rodger',22, 'Male', 'tiny.rodger@gmail.com', 'tiny123456', 'uni101', 'CS 04'
1112, 'elli', 'oslen',22, 'Female', 'elli.oslen@gmail.com', 'elli123456', 'uni105', 'Math 04'
1113, 'Mark', 'Anderson',22, 'Male', 'mark.anderson@gmail.com', 'mark123456', 'uni104', 'CS 03'
1114, 'Ramon', 'Davis',26, 'Male', 'ramon.davis@gmail.com', 'ramon123456', 'uni103', 'CS 05'
1115, 'Debra', 'Perry',29, 'Male', 'debra.perry@gmail.com', 'debra123456', 'uni101', 'CS 02'
1116, 'Sabrina', 'Moody',24, 'Male', 'sabrina.moody@gmail.com', 'sabrina123456', 'uni103', 'CS 01'
1117, 'Seth', 'Scott',21, 'Male', 'seth.scott@gmail.com', 'seth123456', 'uni102', 'CS 05'
1118, 'Dan', 'Richards',29, 'Female', 'dan.richards@gmail.com', 'dan123456', 'uni101', 'CS 04'
1119, 'Chrysta', 'Mitchell',26, 'Female', 'chrysta.mitchell@gmail.com', 'chrysta123456', 'uni102', 'CS 02'
1120, 'Sabrina', 'Thomas',28, 'Female', 'sabrina.thomas@gmail.com', 'sabrina123456', 'uni101', 'Math 04'
1121, 'Lucy', 'Nelson',30, 'Male', 'lucy.nelson@gmail.com', 'lucy123456', 'uni104', 'Math 03'
1122, 'Tracey', 'Norman',24, 'Male', 'tracey.norman@gmail.com', 'tracey123456', 'uni103', 'CS 03'
1123, 'Ivan', 'Walker',21, 'Female', 'ivan.walker@gmail.com', 'ivan123456', 'uni105', 'Math 02'
1124, 'Ramon', 'Moody',28, 'Female', 'ramon.moody@gmail.com', 'ramon123456', 'uni104', 'CS 04'
1125, 'Ethan', 'Potter',25, 'Female', 'ethan.potter@gmail.com', 'ethan123456', 'uni104', 'CS 03'
1126, 'Harvey', 'Fletcher',30, 'Female', 'harvey.fletcher@gmail.com', 'harvey123456', 'uni102', 'CS 01'
1127, 'Isabel', 'Martin',24, 'Female', 'isabel.martin@gmail.com', 'isabel123456', 'uni104', 'Math 01'
1128, 'Claude', 'Miller',21, 'Male', 'claude.miller@gmail.com', 'claude123456', 'uni105', 'Math 02'
1129, 'Jorge', 'Williams',23, 'Male', 'jorge.williams@gmail.com', 'jorge123456', 'uni101', 'Math 03'
1130, 'Beatrice', 'Holland',26, 'Male', 'beatrice.holland@gmail.com', 'beatrice123456', 'uni103', 'CS 03'
1131, 'Dave', 'Miles',21, 'Female', 'dave.miles@gmail.com', 'dave123456', 'uni104', 'CS 03'
1132, 'Chrysta', 'Scott',27, 'Female', 'chrysta.scott@gmail.com', 'chrysta123456', 'uni104', 'CS 02'
1133, 'Angela', 'Scott',22, 'Male', 'angela.scott@gmail.com', 'angela123456', 'uni104', 'CS 02'
1134, 'Ethan', 'Martin',23, 'Female', 'ethan.martin@gmail.com', 'ethan123456', 'uni105', 'CS 05'
1135, 'Nathaniel', 'White',22, 'Male', 'nathaniel.white@gmail.com', 'nathaniel123456', 'uni101', 'CS 02'
1136, 'Jorge', 'Martin',23, 'Female', 'jorge.martin@gmail.com', 'jorge123456', 'uni101', 'CS 05'
1137, 'Lewis', 'Norman',25, 'Male', 'lewis.norman@gmail.com', 'lewis123456', 'uni103', 'CS 03'
1138, 'Chrysta', 'Wheeler',29, 'Male', 'chrysta.wheeler@gmail.com', 'chrysta123456', 'uni103', 'Math 05'
1139, 'Claude', 'Norris',23, 'Female', 'claude.norris@gmail.com', 'claude123456', 'uni102', 'Math 02'
1140, 'Chrysta', 'Shelton',27, 'Female', 'chrysta.shelton@gmail.com', 'chrysta123456', 'uni101', 'CS 01'
```

502 selection regions

C:\Users\manoj_nuhq5\Desktop\student details.csv • - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

student details.csv

```sql
INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1101, 'Lena', 'Smith',25, 'F',
'lena.smith@gmail.com', 'lena123456', 'uni101', 'CS 01' );
INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1102, 'Nicol', 'Green',24, 'M',
'nicol.green@gmail.com', 'nicol123456', 'uni101', 'CS 02' );
INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1103, 'Tom', 'Taylor',26, 'M',
'tom.taylor@gmail.com', 'tom123456', 'uni101', 'CS 03' );
INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1104, 'Paul', 'Miler',22, 'M',
'paul.miler@gmail.com', 'paul123456', 'uni101', 'CS 04' );
INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1105, 'Alen', 'Lee',26, 'F',
'alen.lee@gmail.com', 'alan123456', 'uni101', 'CS 05' );
INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1106, 'David', 'King',23, 'M',
'david.king@gmail.com', 'david123456', 'uni101', 'Math 01' );
INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1107, 'Omar', 'Wood',22, 'M',
'omar.wood@gmail.com', 'omar123456', 'uni101', 'Math 02' );
INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1108, 'Jone', 'Sith',25, 'F',
'jone.smith@gmail.com', 'jone123456', 'uni101', 'Math 03' );
INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1109, 'Lucy', 'White',24, 'F',
'lucy.white@gmail.com', 'lucy123456', 'uni101', 'Math 04' );
INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1110, 'Eren', 'Forger',23, 'M',
'eren.forger@gmai.com', 'eren123456', 'uni101', 'Math 05' );
INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1111, 'tiny', 'rodger',22, 'Male',
'tiny.rodger@gmail.com', 'tiny123456', 'uni101', 'CS 04' );
INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1112, 'elli', 'oslen',22, 'Female',
'elli.oslen@gmail.com', 'elli123456', 'uni105', 'Math 04' );
INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1113, 'Mark', 'Anderson',22, 'Male',
'mark.anderson@gmail.com', 'mark123456', 'uni104', 'CS 03' );
INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1114, 'Ramon', 'Davis',26, 'Male',
'ramon.davis@gmail.com', 'ramon123456', 'uni103', 'CS 05' );
INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1115, 'Debra', 'Perry',29, 'Male',
'debra.perry@gmail.com', 'debra123456', 'uni101', 'CS 02' );
INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1116, 'Sabrina', 'Moody',24, 'Male',
'sabrina.moody@gmail.com', 'sabrina123456', 'uni103', 'CS 01' );
INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1117, 'Seth', 'Scott',21, 'Male',
'seth.scott@gmail.com', 'seth123456', 'uni102', 'CS 05' );
INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1118, 'Dan', 'Richards',29, 'Female',
'dan.richards@gmail.com', 'dan123456', 'uni101', 'CS 04' );
INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1119, 'Chrysta', 'Mitchell',26, 'Female',
'chrysta.mitchell@gmail.com', 'chrysta123456', 'uni102', 'CS 02' );
INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1120, 'Sabrina', 'Thomas',28, 'Female',
'sabrina.thomas@gmail.com', 'sabrina123456', 'uni101', 'Math 04' );
```

Step 5: Copy all the statements and paste in the oracle sql developer to run the code:

SQL Worksheet History

ManojDB

Worksheet Query Builder

```
683 INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1587, 'Debra', 'Williams', 24, 'Male'
684 INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1588, 'Gilbert', 'Scott', 29, 'Male',
685 INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1589, 'Gilbert', 'Shelton', 25, 'Fema
686 INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1590, 'Harvey', 'Potter', 24, 'Male',
687 INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1591, 'Dan', 'Harris', 25, 'Male', 'c
688 INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1592, 'Chrysta', 'Norris', 24, 'Male'
689 INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1593, 'Sabrina', 'Moody', 24, 'Female
690 INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1594, 'Molly', 'Daniel', 28, 'Male',
691 INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1595, 'Stella', 'Lindsey', 28, 'Femal
692 INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1596, 'Dan', 'Wheeler', 29, 'Female',
693 INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1597, 'Glen', 'Anderson', 26, 'Female
694 INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1598, 'Seth', 'Daniel', 23, 'Female',
695 INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1599, 'Seth', 'Walker', 27, 'Female',
696 INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1600, 'Vicki', 'Wheeler', 25, 'Male',
697 INSERT INTO student (student_id, first_name, last_name, age, gender, email, password, campcode, courseid) VALUES(1601, 'Test', 'case', 24, 'Male', 'te
698 -- ----------------------------------------------------------
699
700 select * from student;
```

select * from student;

Script Output | Query Result

SQL | All Rows Fetched: 500 in 0.355 seconds

	STUDEN..	FIRST_NAME	LAST_NAME	AGE	GENDER	EMAIL	PASSWORD	CAMPCODE	COURSEID
1	1102	Nicol	Green	24	M	nicol.green@gmail.com	nicol123456	uni101	CS 02
2	1103	Tom	Taylor	26	M	tom.taylor@gmail.com	tom123456	uni101	CS 03
3	1104	Paul	Miler	22	M	paul.miler@gmail.com	paul123456	uni101	CS 04
4	1105	Alen	Lee	26	F	alen.lee@gmail.com	alan123456	uni101	CS 05
5	1106	David	King	23	M	david.king@gmail.com	david123456	uni101	Math 01
6	1107	Omar	Wood	22	M	omar.wood@gmail.com	omar123456	uni101	Math 02
7	1108	Jone	Sith	25	F	jone.smith@gmail.com	jone123456	uni101	Math 03
8	1109	Lucy	White	24	F	lucy.white@gmail.com	lucy123456	uni101	Math 04
9	1110	Eren	Forger	23	M	eren.forger@gmai.com	eren123456	uni101	Math 05
10	1111	tiny	rodger	22	Male	tiny.rodger@gmail.com	tiny123456	uni101	CS 04
11	1112	elli	oslen	22	Female	elli.oslen@gmail.com	elli123456	uni105	Math 04
12	1113	Mark	Anderson	22	Male	mark.anderson@gmail.com	mark123456	uni104	CS 03
13	1114	Ramon	Davis	26	Male	ramon.davis@gmail.com	ramon123456	uni103	CS 05
14	1115	Debra	Perry	29	Male	debra.perry@gmail.com	debra123456	uni101	CS 02
15	1116	Sabrina	Moody	24	Male	sabrina.moody@gmail.com	sabrina123456	uni103	CS 01
16	1117	Seth	Scott	21	Male	seth.scott@gmail.com	seth123456	uni102	CS 05
17	1118	Dan	Richards	29	Female	dan.richards@gmail.com	dan123456	uni101	CS 04
18	1119	Chrysta	Mitchell	26	Female	chrysta.mitchell@gmail....	chrysta123456	uni102	CS 02
19	1120	Sabrina	Thomas	28	Female	sabrina.thomas@gmail.com	sabrina123456	uni101	Math 04
20	1121	Lucy	Nelson	30	Male	lucy.nelson@gmail.com	lucy123456	uni104	Math 03
21	1122	Tracey	Norman	24	Male	tracey.norman@gmail.com	tracey123456	uni103	CS 03
22	1123	Ivan	Walker	21	Female	ivan.walker@gmail.com	ivan123456	uni105	Math 02

Data from an excel file is successfully loaded into SQL Developer code.

2. CREATING TABLE FOR building_address:

```
CREATE TABLE building_address (
  location varchar(45) NOT NULL,
  street varchar(45) NOT NULL,
  province varchar(45) NOT NULL,
  country varchar(45) NOT NULL,
  pincode varchar(45) NOT NULL
) ;
```

```
INSERT INTO building_address(location, street, province, country, pincode) VALUES('Austin', 'A1', 'Texas', 'USA', '509123');
INSERT INTO building_address(location, street, province, country, pincode) VALUES('Houston', 'H7', 'Texas', 'USA', '609165');
INSERT INTO building_address(location, street, province, country, pincode) VALUES('Miami', 'M4', 'Florida', 'USA', '778902');
INSERT INTO building_address(location, street, province, country, pincode) VALUES('Orlando', 'O2', 'Florida', 'USA', '445323');
INSERT INTO building_address(location, street, province, country, pincode) VALUES('Tampa', 'T5', 'Florida', 'USA', '665431')
```

```
select * from building_address;
```

Script Output | Query Result

SQL | All Rows Fetched: 5 in 0.033 seconds

	LOCATION	STREET	PROVINCE	COUNTRY	PINCODE
1	Austin	A1	Texas	USA	509123
2	Houston	H7	Texas	USA	609165
3	Miami	M4	Florida	USA	778902
4	Orlando	O2	Florida	USA	445323
5	Tampa	T5	Florida	USA	665431

3. CREATING TABLE FOR TABLE campus:

```
CREATE TABLE campus (
  campus_code varchar(45) NOT NULL,
  campus_name varchar(45) NOT NULL,
  location varchar(45) NOT NULL,
  capacity int NOT NULL
);
```

```
INSERT INTO campus (campus_code, campus_name, location, capacity) VALUES ('uni101', 'Lake', 'Miami', 60000);
INSERT INTO campus (campus_code, campus_name, location, capacity) VALUES ('uni102', 'Wood', 'Tampa', 35000);
INSERT INTO campus (campus_code, campus_name, location, capacity) VALUES ('uni103', 'Sky', 'Orlando', 55000);
INSERT INTO campus (campus_code, campus_name, location, capacity) VALUES ('uni104', 'Red', 'Houston', 48000);
INSERT INTO campus (campus_code, campus_name, location, capacity) VALUES ('uni105', 'Cloud', 'Austin', 50000);
select * from campus;
```

Script Output | Query Result

SQL | All Rows Fetched: 5 in 0.054 seconds

	CAMPUS_CODE	CAMPUS_NAME	LOCATION	CAPACITY
1	uni101	Lake	Miami	60000
2	uni102	Wood	Tampa	35000
3	uni103	Sky	Orlando	55000
4	uni104	Red	Houston	48000
5	uni105	Cloud	Austin	50000

4. CREATING TABLE FOR TABLE course

```
CREATE TABLE course (
  course_id varchar(45) NOT NULL,
  course_name varchar(45) NOT NULL,
  course_duration varchar(45) NOT NULL,
  credits int NOT NULL,
  instructorid int NOT NULL
) ;
INSERT INTO course (course_id, course_name, course_duration, credits, instructorid) VALUES ('CS 01', 'DBMS', '3 Months', 3, 5506);
INSERT INTO course (course_id, course_name, course_duration, credits, instructorid) VALUES ('CS 02', 'Data Mining', '3 Months', 3, 5507);
INSERT INTO course (course_id, course_name, course_duration, credits, instructorid) VALUES ('CS 03', 'Java ', '4 Months', 3, 5508);
INSERT INTO course (course_id, course_name, course_duration, credits, instructorid) VALUES ('CS 04', 'Operating Systems', '6 Months', 2, 5509);
INSERT INTO course (course_id, course_name, course_duration, credits, instructorid) VALUES ('CS 05', 'Data Structures', '1 Year', 4, 5510);
INSERT INTO course (course_id, course_name, course_duration, credits, instructorid) VALUES ('MATH 01', 'Statistics', '6 Months', 4, 5501);
INSERT INTO course (course_id, course_name, course_duration, credits, instructorid) VALUES ('MATH 02', 'Algebra', '4 Months', 3, 5502);
INSERT INTO course (course_id, course_name, course_duration, credits, instructorid) VALUES ('MATH 03', 'Regression Models', '6 Months', 3, 5503);
INSERT INTO course (course_id, course_name, course_duration, credits, instructorid) VALUES ('MATH 04', 'Calculus', '6 Months', 2, 5504);
INSERT INTO course (course_id, course_name, course_duration, credits, instructorid) VALUES ('MATH 05', 'Multivariate Analysis', '1 Year', 4, 5505);

select * from course;
```

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0.031 seconds

	COURSE_ID	COURSE_NAME	COURSE_DURATION	CREDITS	INSTRUCTORID
1	CS 01	DBMS	3 Months	3	5506
2	CS 02	Data Mining	3 Months	3	5507
3	CS 03	Java	4 Months	3	5508
4	CS 04	Operating Systems	6 Months	2	5509
5	CS 05	Data Structures	1 Year	4	5510
6	MATH 01	Statistics	6 Months	4	5501
7	MATH 02	Algebra	4 Months	3	5502
8	MATH 03	Regression Models	6 Months	3	5503
9	MATH 04	Calculus	6 Months	2	5504
10	MATH 05	Multivariate Analysis	1 Year	4	5505

5. CREATING TABLE FOR TABLE department details

```
CREATE TABLE department_details (
  dept_id varchar(45) NOT NULL,
  dept_name varchar(45) NOT NULL,
  HOD varchar(45) NOT NULL
);

INSERT INTO department_details (dept_id, dept_name, HOD) VALUES
('CS', 'Computer Science', 'Paul Rowell');
INSERT INTO department_details (dept_id, dept_name, HOD) VALUES('Math', 'Mathematics', 'David White');

select * from department_details;
```

Script Output | Query Result

SQL | All Rows Fetched: 2 in 0.033 seconds

	DEPT_ID	DEPT_NAME	HOD
1	CS	Computer Scie...	Paul Rowell
2	Math	Mathematics	David White

6. CREATING TABLE FOR TABLE INSTRUCTOR

```
CREATE TABLE instructor (
  instructor_id int NOT NULL,
  name varchar(45) NOT NULL,
  age varchar(45) NOT NULL,
  gender varchar(45) NOT NULL,
  department varchar(45) NOT NULL
);

INSERT INTO instructor (instructor_id, name, age, gender, department) VALUES (5501, 'John', '55', 'Male', 'MATH');
INSERT INTO instructor (instructor_id, name, age, gender, department) VALUES (5502, 'Mark', '40', 'Male', 'MATH');
INSERT INTO instructor (instructor_id, name, age, gender, department) VALUES (5503, 'Rick', '39', 'Male', 'MATH');
INSERT INTO instructor (instructor_id, name, age, gender, department) VALUES (5504, 'Liu', '52', 'Female', 'MATH');
INSERT INTO instructor (instructor_id, name, age, gender, department) VALUES (5505, 'Sim', '51', 'Female', 'MATH');
INSERT INTO instructor (instructor_id, name, age, gender, department) VALUES (5506, 'Ashley', '48', 'Female', 'CS');
INSERT INTO instructor (instructor_id, name, age, gender, department) VALUES (5507, 'Riley', '49', 'Female', 'CS');
INSERT INTO instructor (instructor_id, name, age, gender, department) VALUES (5508, 'Wade', '43', 'Wade', 'CS');
INSERT INTO instructor (instructor_id, name, age, gender, department) VALUES (5509, 'Ivan', '42', 'Male', 'CS');
INSERT INTO instructor (instructor_id, name, age, gender, department) VALUES (5510, 'Dan', '60', 'Male', 'CS');

select * from instructor;
```

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0.035 seconds

	INSTRUCTOR_ID	NAME	AGE	GENDER	DEPARTMENT
1	5501	John	55	Male	MATH
2	5502	Mark	40	Male	MATH
3	5503	Rick	39	Male	MATH
4	5504	Liu	52	Female	MATH
5	5505	Sim	51	Female	MATH
6	5506	Ashley	48	Female	CS
7	5507	Riley	49	Female	CS
8	5508	Wade	43	Wade	CS
9	5509	Ivan	42	Male	CS
10	5510	Dan	60	Male	CS

7. CREATING TABLE FOR TABLE INSTRUCTOR_ADDRESS

```
CREATE TABLE instructor_address (
  inst_add varchar(45) NOT NULL,
  inst_id int NOT NULL,
  street varchar(45) NOT NULL,
  city varchar(45) NOT NULL,
  state varchar(45) NOT NULL,
  country varchar(45) NOT NULL,
  pincode varchar(45) NOT NULL
);
```

```
INSERT INTO instructor_address (inst_add, inst_id, street, city, state, country, pincode)
VALUES ('Flat No. 002', 5507, 'S3', 'Seattle', 'Washington', 'USA', '468870');
```

INSERT INTO instructor_address (inst_add, inst_id, street, city, state, country, pincode) VALUES ('Flat No. 065', 5506, 'A3', 'Atlanta', 'Georgia', 'USA', '354111');

INSERT INTO instructor_address (inst_add, inst_id, street, city, state, country, pincode) VALUES ('Flat No. 222', 5503, 'C4', 'San Diego', 'California', 'USA', '345234');

INSERT INTO instructor_address (inst_add, inst_id, street, city, state, country, pincode) VALUES ('Flat No. 268', 5501, 'I8', 'Chicago', 'Illinois', 'USA', '456665');

INSERT INTO instructor_address (inst_add, inst_id, street, city, state, country, pincode) VALUES ('Flat No. 356', 5508, 'D7', 'Denver', 'Colorado', 'USA', '887543');

INSERT INTO instructor_address (inst_add, inst_id, street, city, state, country, pincode) VALUES ('Flat No. 456', 5510, 'M9', 'Miami', 'Florida', 'USA', '354567');

INSERT INTO instructor_address (inst_add, inst_id, street, city, state, country, pincode) VALUES ('Flat No. 568', 5504, 'A9', 'Pheonix', 'Arizona', 'USA', '970065');

INSERT INTO instructor_address (inst_add, inst_id, street, city, state, country, pincode) VALUES ('Flat No. 751', 5505, 'B5', 'Boston', 'Massachusetts', 'USA', '345665');

INSERT INTO instructor_address (inst_add, inst_id, street, city, state, country, pincode) VALUES ('Flat No. 850', 5502, 'T7', 'Houston', 'Texas', 'USA', '345332');

INSERT INTO instructor_address (inst_add, inst_id, street, city, state, country, pincode) VALUES ('Flat No. 908', 5509, 'P9', 'Portland', 'Oregon', 'USA', '666532');

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0.035 seconds

	INST_ADD	INST_ID	STREET	CITY	STATE	COUNTRY	PINCODE
1	Flat No. 002	5507	S3	Seattle	Washington	USA	468870
2	Flat No. 065	5506	A3	Atlanta	Georgia	USA	354111
3	Flat No. 222	5503	C4	San Diego	California	USA	345234
4	Flat No. 268	5501	I8	Chicago	Illinois	USA	456665
5	Flat No. 356	5508	D7	Denver	Colorado	USA	887543
6	Flat No. 456	5510	M9	Miami	Florida	USA	354567
7	Flat No. 568	5504	A9	Pheonix	Arizona	USA	970065
8	Flat No. 751	5505	B5	Boston	Massachuse...	USA	345665
9	Flat No. 850	5502	T7	Houston	Texas	USA	345332
10	Flat No. 908	5509	P9	Portland	Oregon	USA	666532

8. CREATING TABLE for table subject details

```
CREATE TABLE subject_details (
  course_id_ varchar(45) NOT NULL,
  course_name varchar(45) NOT NULL,
  dept_id_ varchar(45) NOT NULL,
  deptname varchar(45) NOT NULL
);
```

```
INSERT INTO subject_details (course_id_, course_name, dept_id_, deptname) VALUES ('MATH 02', 'Algebra', 'MATH', 'Mathematics');

INSERT INTO subject_details (course_id_, course_name, dept_id_, deptname) VALUES ('MATH 04', 'Calculus', 'MATH', 'Mathematics');

INSERT INTO subject_details (course_id_, course_name, dept_id_, deptname) VALUES ('CS 02', 'Data Mining', 'CS', 'Computer Science');

INSERT INTO subject_details (course_id_, course_name, dept_id_, deptname) VALUES ('CS 05', 'Data Structures', 'CS', 'Computer Science');

INSERT INTO subject_details (course_id_, course_name, dept_id_, deptname) VALUES ('CS 01', 'DBMS', 'CS', 'Computer Science');

INSERT INTO subject_details (course_id_, course_name, dept_id_, deptname) VALUES ('CS 03', 'Java', 'CS', 'Computer Science');

INSERT INTO subject_details (course_id_, course_name, dept_id_, deptname) VALUES ('MATH 05', 'Multivariate Analysis', 'MATH', 'Mathematics');

INSERT INTO subject_details (course_id_, course_name, dept_id_, deptname) VALUES ('CS 04', 'Operating Systems', 'CS', 'Computer Science');

INSERT INTO subject_details (course_id_, course_name, dept_id_, deptname) VALUES ('MATH 03', 'Regression Models', 'MATH', 'Mathematics');

INSERT INTO subject_details (course_id_, course_name, dept_id_, deptname) VALUES ('MATH 01', 'Statistics', 'MATH', 'Mathematics');
```

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0.035 seconds

	COURSE_ID_	COURSE_NAME	DEPT_ID_	DEPTNAME
1	MATH 02	Algebra	MATH	Mathematics
2	MATH 04	Calculus	MATH	Mathematics
3	CS 02	Data Mining	CS	Computer Science
4	CS 05	Data Structures	CS	Computer Science
5	CS 01	DBMS	CS	Computer Science
6	CS 03	Java	CS	Computer Science
7	MATH 05	Multivariate Analysis	MATH	Mathematics
8	CS 04	Operating Systems	CS	Computer Science
9	MATH 03	Regression Models	MATH	Mathematics
10	MATH 01	Statistics	MATH	Mathematics

2.3 Data integrity:

Data Integrity refers to the consistency and maintenance of the data through the life cycle of the database. In a database, data integrity can be ensured through the implementation of Integrity constraints in a table. Integrity constraints apply business rules to the database tables. The constraints can either be at a column level or a table level. Below is the description of the constraints used in each table:

1. *Student table* holds the data of all the student information, also it gets updated automatically when a new student enrolls in a course. Here we used student_id as the primary key identifier and additionally followed by basic details like first_name,last_name,student_id, first_name, last_name, age, gender, email, password, campcode, courseid. This table also has 2 foreign keys that references course_id column in course table and campus_code column in campus table.

2. *Course table* holds the data of all the course information. Here we used course_id as the primary key identifier and additionally followed by basic details like course_name, course_duration, credits, instructorid. This table has 1 foreign key that references the instructor_id column in the instructor table.

3. *Campus table* holds the data of all the campus information. Here we used campus_code as the primary key identifier and additionally followed by basic details like campus_name,capacity,location. This table has 1 foreign key that references the location column in the building address table.

4. *Building address table* holds the data of all the building location information. Here we used location as the primary key identifier and additionally followed by basic details like street, province, country, pincode.

5. *Instructor table* holds the data of all the instructor information. Here we used instrcutor_id as the primary key identifier and additionally followed by basic details like name, age,gender, department.

6. *Instructor address table* holds the data of all address related information of instructor. Here we used inst_add as the primary key identifier and additionally followed by basic details like inst_id,street, state, city,country,pincode. This table has 1 foreign key that references the instructor_id column in the instructor table.

7. *Department details table* holds the data of all the department information of courses offered. Here we used dept_id as the primary key identifier and additionally followed by basic details like courseid, dept_name,HOD. This table has 1 foreign key that references the course_id column in the course table.

8. *Subject details table* holds the data of all the subject information. Here we used course_name as the primary key identifier and additionally followed by details like course_id_, dept_id_,deptname. This table has 2 foreign keys that references the course_id column in the course table and dept_id column in the department details table.

PRIMARY KEY Constraints:

1. Table building_address:

```
ALTER TABLE building_address
  ADD PRIMARY KEY (location);
```

2. Table campus:

```
ALTER TABLE campus
  ADD PRIMARY KEY (campus_code),
```

3. Table course

```
ALTER TABLE course
  ADD PRIMARY KEY (course_id);
```

4. Table department details

```
ALTER TABLE department details
  ADD PRIMARY KEY (dept_id);
```

5. Table instructor

```
ALTER TABLE instructor
  ADD PRIMARY KEY (instructor_id);
```

6. Table instructor_address:

```
ALTER TABLE instructor_address
  ADD PRIMARY KEY (inst_add);
```

7. Table student

```
ALTER TABLE student
  ADD PRIMARY KEY (student_id);
```

8. Table subject details

```
ALTER TABLE subject details
  ADD PRIMARY KEY (course_name);
```

FOREIGN KEY Constraints:

```
ALTER TABLE campus

    ADD CONSTRAINT location
    FOREIGN KEY (location) REFERENCES building_address (location)
    ON DELETE NO ACTION ON UPDATE NO ACTION;


ALTER TABLE course

    ADD CONSTRAINT instructorid
    FOREIGN KEY (instructorid) REFERENCES instructor (instructor_id)
    ON DELETE NO ACTION ON UPDATE NO ACTION;


ALTER TABLE instructor_address

    ADD CONSTRAINT inst_id
    FOREIGN KEY (inst_id) REFERENCES instructor (instructor_id)
    ON DELETE NO ACTION ON UPDATE NO ACTION;


ALTER TABLE student

    ADD CONSTRAINT campcode
    FOREIGN KEY (campcode) REFERENCES campus (campus_code)
    ON DELETE NO ACTION ON UPDATE NO ACTION,

    ADD CONSTRAINT courseid
    FOREIGN KEY (courseid) REFERENCES course (course_id)
    ON DELETE NO ACTION ON UPDATE NO ACTION;


ALTER TABLE subject details

    ADD CONSTRAINT course_id
    FOREIGN KEY (course_id_) REFERENCES course (course_id)
    ON DELETE NO ACTION ON UPDATE NO ACTION,

    ADD CONSTRAINT dept_id_
    FOREIGN KEY (dept_id_) REFERENCES department_details (dept_id)
    ON DELETE NO ACTION ON UPDATE NO ACTION;

COMMIT;
```

3. Query Writing

Creating Procedures:

For this Experiment we used MySQL Developer platform. This platform is very convenient and precise in creating procedures.

Steps:

Select Schema -> Right click on Stored Procedures -> Create Stored Procedure

After writing the query, click apply and then call the procedure in a new file.

Below are few examples:

1. **Creating a procedure to find details of all the students who enrolled in MATH courses.**

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `nproc_mathdetails`()
BEGIN
SELECT * FROM student where courseid LIKE "MATH%";
END

call nproc_mathdetails;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	first_name	last_name	age	gender	email	password	campcode	courseid
1106	David	King	23	M	david.king@gmail.com	david123456	uni101	Math 01
1107	Omar	Wood	22	M	omar.wood@gmail.com	omar123456	uni101	Math 02
1108	Jone	Sith	25	F	jone.smith@gmail.com	jone123456	uni101	Math 03
1109	Lucy	White	24	F	lucy.white@gmail.com	lucy123456	uni101	Math 04
1110	Eren	Forger	23	M	eren.forger@gmai.com	eren123456	uni101	Math 05
1112	elli	oslen	22	Female	elli.oslen@gmail.com	elli123456	uni105	Math 04
1120	Sabrina	Thomas	28	Female	sabrina.thomas@gmail.com	sabrina123456	uni101	Math 04
1121	Lucy	Nelson	30	Male	lucy.nelson@gmail.com	lucy123456	uni104	Math 03
1123	Ivan	Walker	21	Female	ivan.walker@gmail.com	ivan123456	uni105	Math 02
1127	Isabel	Martin	24	Female	isabel.martin@gmail.com	isabel123456	uni104	Math 01
1128	Claude	Miller	21	Male	claude.miller@gmail.com	claude123456	uni105	Math 02
1129	Jorge	Williams	23	Male	jorge.williams@gmail.com	jorge123456	uni101	Math 03
1138	Chrysta	Wheeler	29	Male	chrysta.wheeler@gmail.com	chrysta123456	uni103	Math 05
1139	Claude	Norris	23	Female	claude.norris@gmail.com	claude123456	uni102	Math 02
1141	Lucy	Francis	24	Female	lucy.francis@gmail.com	lucy123456	uni103	Math 04
1146	Lauren	Robinson	25	Female	lauren.robinson@gmail.com	lauren123456	uni105	Math 01
1153	Chrysta	Erickson	20	Male	chrysta.erickson@gmail.com	chrysta123456	uni105	Math 03
1160	Dan	Smith	27	Female	dan.smith@gmail.com	dan123456	uni105	Math 04
1162	Daisy	Miles	27	Male	daisy.miles@gmail.com	daisy123456	uni103	Math 03
1166	Daisy	Richards	27	Male	daisy.richards@gmail.com	daisy123456	uni104	Math 01
1172	Tracey	Guzman	20	Male	tracey.guzman@gmail.com	tracey123456	uni103	Math 05
1177	Chrysta	Stanley	21	Female	chrysta.stanley@gmail.com	chrysta123456	uni101	Math 04
1179	Debra	Anderson	20	Female	debra.anderson@gmail.com	debra123456	uni104	Math 05
1186	Lauren	Newton	24	Male	lauren.newton@gmail.com	lauren123456	uni104	Math 02
1190	Vera	Holland	23	Female	vera.holland@gmail.com	vera123456	uni105	Math 02
1191	Jorge	Williams	28	Male	jorge.williams@gmail.com	jorge123456	uni104	Math 03
1192	Roberto	Francis	25	Female	roberto.francis@gmail.com	roberto123456	uni101	Math 01
1193	Milton	Clark	21	Female	milton.clark@gmail.com	milton123456	uni103	Math 02
1194	Ethan	Miles	28	Male	ethan.miles@gmail.com	ethan123456	uni105	Math 03
1195	Loretta	Cooper	22	Male	loretta.cooper@gmail.com	loretta123456	uni102	Math 05
1219	Riley	Guzman	27	Female	riley.guzman@gmail.com	riley123456	uni104	Math 03

2. Writing a procedure to find details of students and course who enrolled in the MATH 01 course.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `nproc_studcourse`()
BEGIN
select student_id,email,course_name,credits,instructorid from student s LEFT
JOIN course co ON s.courseid=co.course_id where s.courseid="MATH 01";
END
```

call nproc_studcourse;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	email	course_name	credits	instructorid
1106	david.king@gmail.com	Statistics	4	5501
1127	isabel.martin@gmail.com	Statistics	4	5501
1146	lauren.robinson@gmail.com	Statistics	4	5501
1166	daisy.richards@gmail.com	Statistics	4	5501
1192	roberto.francis@gmail.com	Statistics	4	5501
1278	beatrice.moody@gmail.com	Statistics	4	5501
1295	christina.fletcher@gmail.com	Statistics	4	5501
1298	isabel.robinson@gmail.com	Statistics	4	5501
1305	roberto.guzman@gmail.com	Statistics	4	5501
1315	ivan.white@gmail.com	Statistics	4	5501
1346	seth.newton@gmail.com	Statistics	4	5501
1369	brian.anderson@gmail.com	Statistics	4	5501
1373	nathaniel.moody@gmail.com	Statistics	4	5501
1374	liam.miles@gmail.com	Statistics	4	5501
1375	isabel.thomas@gmail.com	Statistics	4	5501
1388	isabel.scott@gmail.com	Statistics	4	5501
1403	melody.daniel@gmail.com	Statistics	4	5501
1425	dan.davis@gmail.com	Statistics	4	5501
1438	sabrina.potter@gmail.com	Statistics	4	5501
1445	riley.stanley@gmail.com	Statistics	4	5501
1459	lucy.moody@gmail.com	Statistics	4	5501
1462	joshua.clark@gmail.com	Statistics	4	5501
1466	beatrice.norris@gmail.com	Statistics	4	5501
1498	milton.stanley@gmail.com	Statistics	4	5501
1499	liam.harris@gmail.com	Statistics	4	5501
1516	nathaniel.mitchell@gmail.com	Statistics	4	5501
1517	deborah.holland@gmail.com	Statistics	4	5501
1538	glen.stanley@gmail.com	Statistics	4	5501
1551	isabel.perry@gmail.com	Statistics	4	5501
1566	melody.robinson@gmail.com	Statistics	4	5501

3. Writing a procedure to find details of the campus with the highest capacity.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `nproc_campcapacity`()
BEGIN
select * from campus
order by capacity DESC LIMIT 1;

END
```

call nproc_campcapacity;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

campus_code	campus_name	location	capacity
uni101	Lake	Miami	60000

4. Creating a procedure to list out all the distinct states to which instructors belong.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `nproc_inststate`()
BEGIN
SELECT DISTINCT state FROM enrolldata.`instructor address` ;
END
```

call nproc_inststate;

Result Grid | Filter Rows: | Export:

state
Washington
Georgia
California
Illinois
Colorado
Florida
Arizona
Massachusetts
Texas
Oregon

Below are few more Queries using various actions

5. Query to find details of instructors in the age group 40 to 50

SELECT * FROM enrolldata.instructor i inner join enrolldata.`instructor address` ia on i.instructor_id = ia.inst_id where i.age between 40 and 50;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

instructor_id	name	age	gender	department	inst_add	inst_id	street	city	state	country	pincode
5507	Riley	49	Female	CS	Flat No. 002	5507	S3	Seattle	Washington	USA	468870
5506	Ashley	48	Female	CS	Flat No. 065	5506	A3	Atlanta	Georgia	USA	354111
5508	Wade	43	Wade	CS	Flat No. 356	5508	D7	Denver	Colorado	USA	887543
5502	Mark	40	Male	MATH	Flat No. 850	5502	T7	Houston	Texas	USA	345332
5509	Ivan	42	Male	CS	Flat No. 908	5509	P9	Portland	Oregon	USA	666532

6. Query to find the campus whose student count is more than 100?

SELECT *

FROM student

GROUP BY campcode

HAVING COUNT(student_id) >100;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	first_name	last_name	age	gender	email	password	campcode	courseid
1101	Lena	Smith	25	F	lena.smith@gmail.com	lena123456	uni101	CS 01
1113	Mark	Anderson	22	Male	mark.anderson@gmail.com	mark123456	uni104	CS 03
1112	elli	oslen	22	Female	elli.oslen@gmail.com	elli123456	uni105	Math 04

7. Query to find details of male students whose age is above 25.

select student_id,first_name,last_name,courseid,course_name from student s inner join course co on s.courseid=co.course_id where s.age>25 and s.gender="Male";

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	first_name	last_name	courseid	course_name
1142	Tracey	Cooper	CS 01	DBMS
1174	Janet	Clark	CS 01	DBMS
1198	Isabel	Miles	CS 01	DBMS
1237	Lucy	Parks	CS 01	DBMS
1283	Sabrina	Norman	CS 01	DBMS
1348	Nathaniel	Miller	CS 01	DBMS
1359	Claude	Lucas	CS 01	DBMS
1387	Nathaniel	Terry	CS 01	DBMS
1414	Loretta	Parks	CS 01	DBMS
1419	Lucy	Newton	CS 01	DBMS
1546	Chrysta	Daniel	CS 01	DBMS
1547	Beverly	Miles	CS 01	DBMS
1115	Debra	Perry	CS 02	Data Mining
1184	Sabrina	Walker	CS 02	Data Mining
1213	Dan	Walker	CS 02	Data Mining
1227	Ramon	Terry	CS 02	Data Mining
1280	Ethan	Daniel	CS 02	Data Mining
1334	Roberto	Shelton	CS 02	Data Mining
1357	Deborah	Holland	CS 02	Data Mining
1393	Lucy	Warburton	CS 02	Data Mining
1415	Christina	Parks	CS 02	Data Mining
1452	Miles	Clark	CS 02	Data Mining
1470	Liam	Nelson	CS 02	Data Mining
1520	Janet	Cooper	CS 02	Data Mining
1522	Daisy	Wheeler	CS 02	Data Mining
1530	Miles	Morgan	CS 02	Data Mining
1594	Molly	Daniel	CS 02	Data Mining
1130	Beatrice	Holland	CS 03	Java
1303	Dave	Martin	CS 03	Java
1408	Isabel	Francis	CS 03	Java
1421	Chrysta	Terry	CS 03	Java

8. Query to find details of campus located in FLORIDA

SELECT * FROM enrolldata.campus cm INNER JOIN enrolldata.`building address` ba on cm.location=ba.location where ba.province="Florida";

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

campus_code	campus_name	location	capacity	location	street	province	country	pincode
uni101	Lake	Miami	60000	Miami	M4	Florida	USA	778902
uni102	Wood	Tampa	35000	Tampa	T5	Florida	USA	665431
uni103	Sky	Orlando	55000	Orlando	O2	Florida	USA	445323

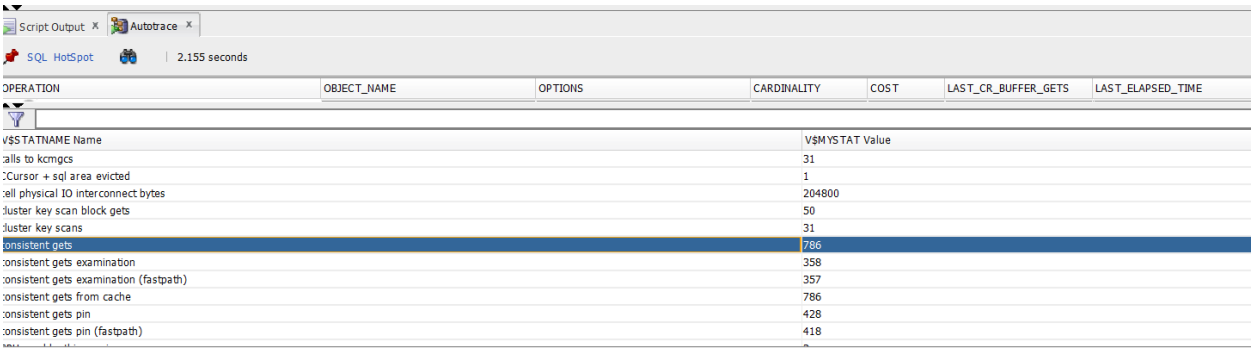
4. PERFORMANCE TUNING

4.1 Indexing:

An index is a schema object that enables quick, direct access to rows by storing entries for each value that appears in the table or cluster's indexed columns.

Without Index:

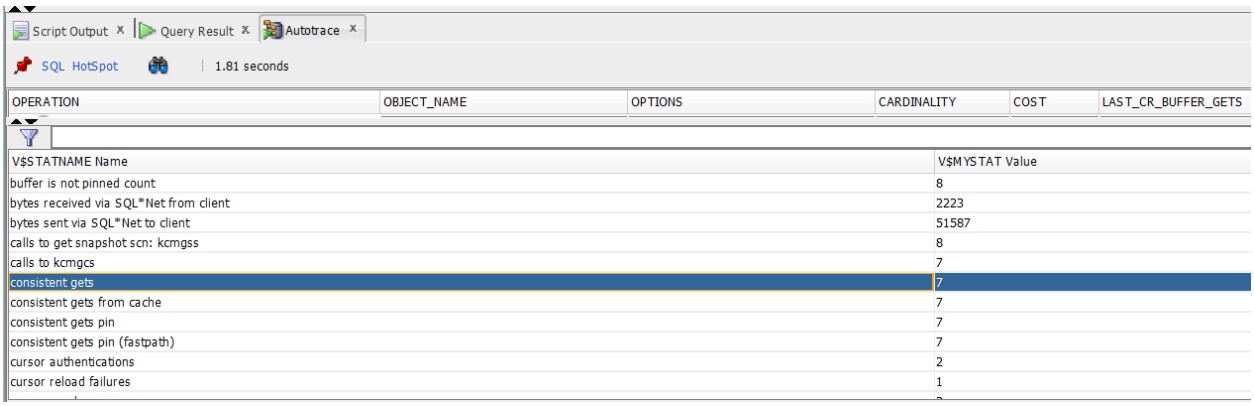
```
CREATE TABLE campus_withoutindex (
  campus_code varchar(45) NOT NULL,
  campus_name varchar(45) NOT NULL,
  location varchar(45) NOT NULL,
  capacity int NOT NULL
);
INSERT INTO campus_withoutindex  (campus_code, campus_name, location, capacity)
VALUES ('uni101', 'Lake', 'Miami', 60000);
INSERT INTO campus_withoutindex  (campus_code, campus_name, location, capacity)
VALUES ('uni102', 'Wood', 'Tampa', 35000);
INSERT INTO campus_withoutindex  (campus_code, campus_name, location, capacity)
VALUES ('uni103', 'Sky', 'Orlando', 55000);
INSERT INTO campus_withoutindex  (campus_code, campus_name, location, capacity)
VALUES ('uni104', 'Red', 'Houston', 48000);
INSERT INTO campus_withoutindex  (campus_code, campus_name, location, capacity)
VALUES ('uni105', 'Cloud', 'Austin', 50000);
select * from campus_withoutindex; #Autotrace
```



CREATE TABLE campus_withindex AS SELECT * FROM campus;

CREATE INDEX campus_code_index ON campus_withindex(campus_code);

select * from campus_withindex; #Autotrace



Script Output | Query Result | Autotrace

SQL HotSpot | 1.81 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST	LAST_CR_BUFFER_GETS

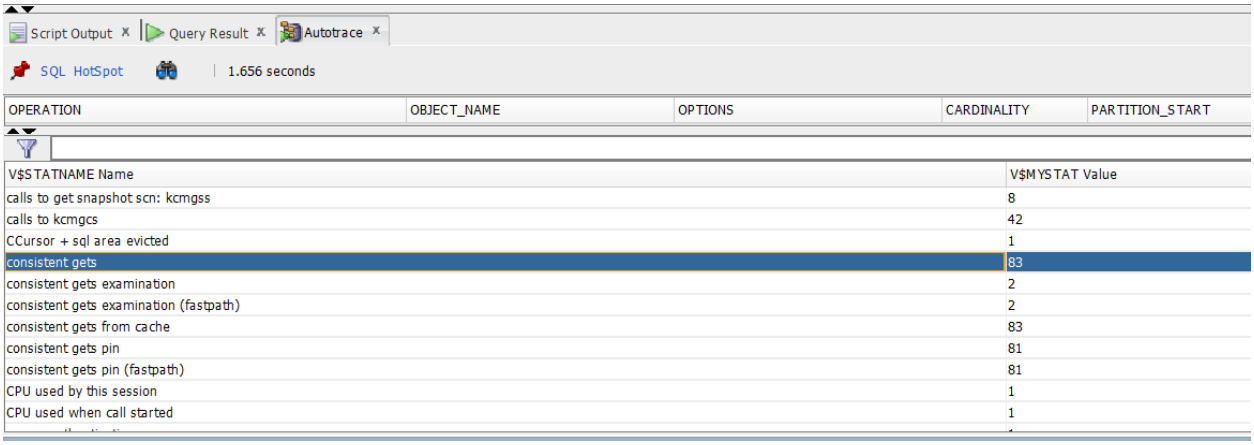
V$STATNAME Name	V$MYSTAT Value
buffer is not pinned count	8
bytes received via SQL*Net from client	2223
bytes sent via SQL*Net to client	51587
calls to get snapshot scn: kcmgss	8
calls to kcmgcs	7
consistent gets	7
consistent gets from cache	7
consistent gets pin	7
consistent gets pin (fastpath)	7
cursor authentications	2
cursor reload failures	1

As we can see from the above two results that the performance of the system is increased effectively after indexing is used.

4.2 Table Partitioning:

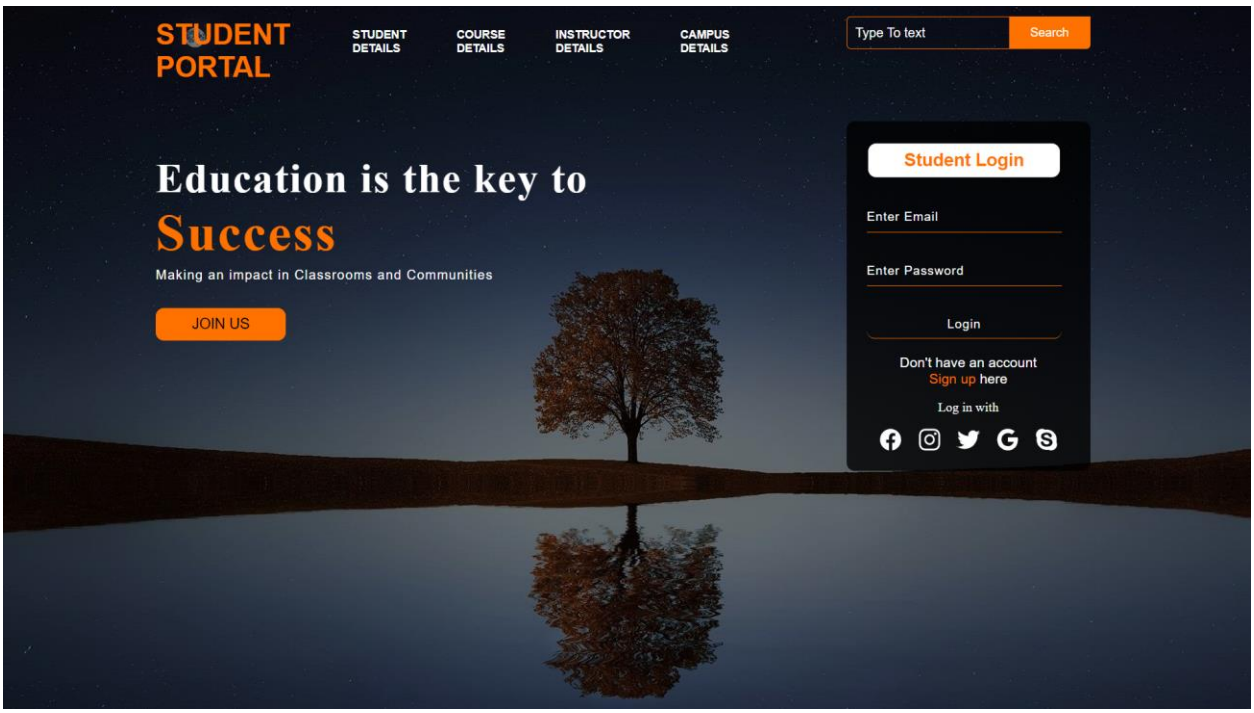
```
CREATE TABLE student_part (
  student_id int NOT NULL PRIMARY KEY ,
  first_name varchar(45) NOT NULL,
  last_name varchar(45) NOT NULL,
  age int NOT NULL,
  gender varchar(45) NOT NULL,
  email varchar(45) NOT NULL,
  password varchar(45) NOT NULL,
  campcode varchar(45) NOT NULL,
  courseid varchar(45) NOT NULL,
  FOREIGN KEY(courseid) REFERENCES course(course_id)
  )
```

```
 PARTITION BY RANGE (student_id)
 (PARTITION p1200 VALUES LESS THAN (1200),
  PARTITION p1300 VALUES LESS THAN (1300),
  PARTITION p1400 VALUES LESS THAN (1400)
);

INSERT INTO student_part (student_id,
first_name,
last_name,
age,
gender,
email,
password,
campcode,
courseid) select * from student;

select * from STUDENT_PART;
```



5. Interface Design

Student Portal: Its is the main page which provides several functionalities such as reading backend data, checking student credentials and allowing to login and redirecting to sign up page.



Reading Data : Data from student,course,campus, instructor tables can be fetched from database server and read in the frontend UI.

To do this, click on the STUDENT DETAILS icon on top of the page. Similarly to view course,campus and instructor data select the respective details icon.

Data will be displayed as shown in below screenshots

STUDENT DEATILS

STUDENT ID	FIRST NAME	LAST NAME	GENDER	AGE	MAIL ID	CAMPUS CODE	COURSE ID
1101	Lena	Smith	F	25	lena.smith@gmail.com	uni101	CS 01
1102	Nicol	Green	M	24	nicol.green@gmail.com	uni101	CS 02
1103	Tom	Taylor	M	26	tom.taylor@gmail.com	uni101	CS 03
1104	Paul	Miler	M	22	paul.miler@gmail.com	uni101	CS 04
1105	Alen	Lee	F	26	alen.lee@gmail.com	uni101	CS 05
1106	David	King	M	23	david.king@gmail.com	uni101	Math 01
1107	Omar	Wood	M	22	omar.wood@gmail.com	uni101	Math 02
1108	Jone	Sith	F	25	jone.smith@gmail.com	uni101	Math 03
1109	Lucy	White	F	24	lucy.white@gmail.com	uni101	Math 04
1110	Eren	Forger	M	23	eren.forger@gmai.com	uni101	Math 05
1111	tiny	rodger	Male	22	tiny.rodger@gmail.com	uni101	CS 04
1112	elli	oslen	Female	22	elli.oslen@gmail.com	uni105	Math 04
1113	Mark	Anderson	Male	22	mark.anderson@gmail.com	uni104	CS 03
1114	Ramon	Davis	Male	26	ramon.davis@gmail.com	uni103	MATH 05

COURSE DEATILS

COURSE ID	COURSE NAME	COURSE DURATION	CREDITS	INSTRUCTOR ID
CS 01	DBMS	3 Months	3	5506
CS 02	Data Mining	3 Months	3	5507
CS 03	Java	4 Months	3	5508
CS 04	Operating Systems	6 Months	2	5509
CS 05	Data Structures	1 Year	4	5510
MATH 01	Statistics	6 Months	4	5501
MATH 02	Algebra	4 Months	3	5502
MATH 03	Regression Models	6 Months	3	5503
MATH 04	Calculus	6 Months	2	5504
MATH 05	Multivariate Analysis	1 Year	4	5505

CAMPUS DEATILS

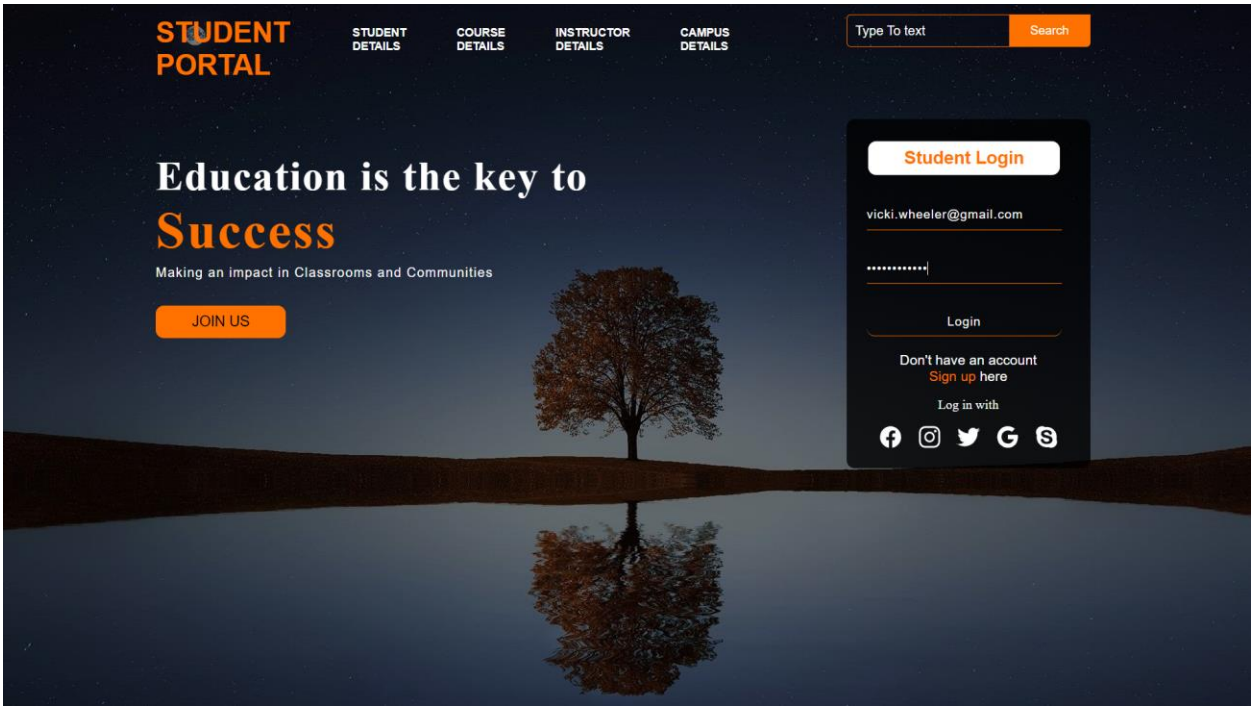
CAMPUS_CODE	CAMPUS_NAME	LOCATION	CAPACITY
uni101	Lake	Miami	60000
uni102	Wood	Tampa	35000
uni103	Sky	Orlando	55000
uni104	Red	Houston	48000
uni105	Cloud	Austin	50000

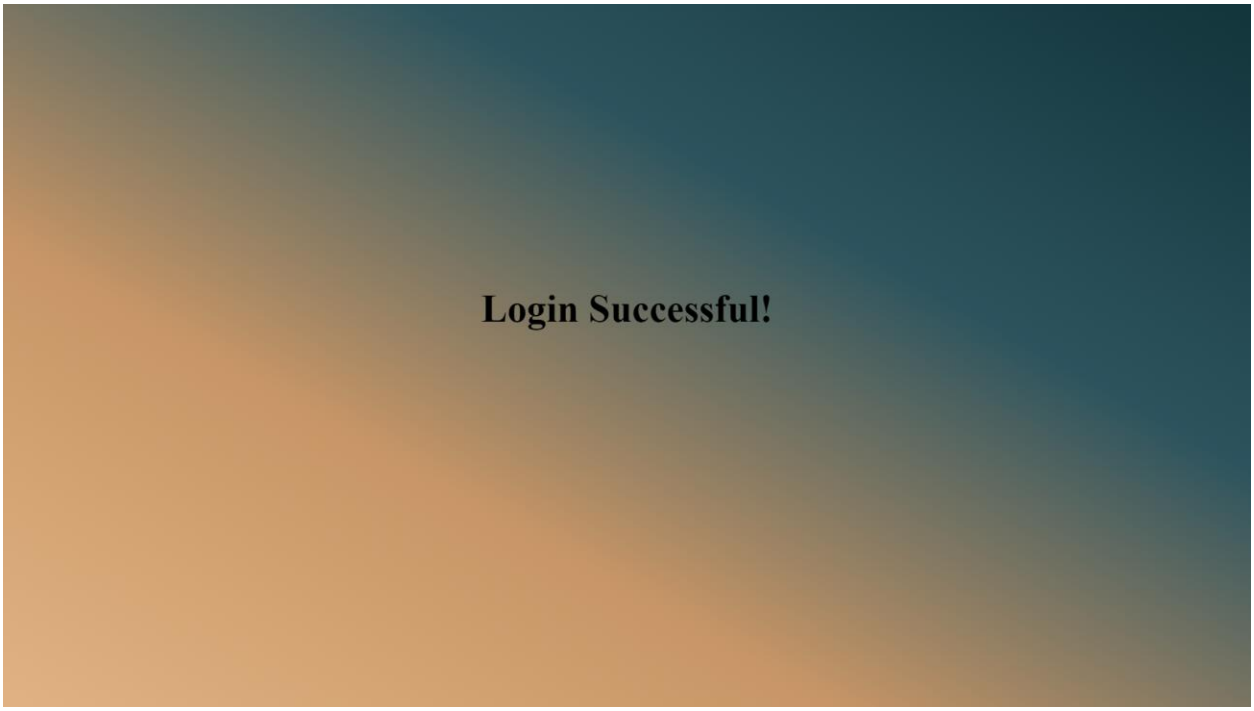
INSTRUCTOR DEATILS

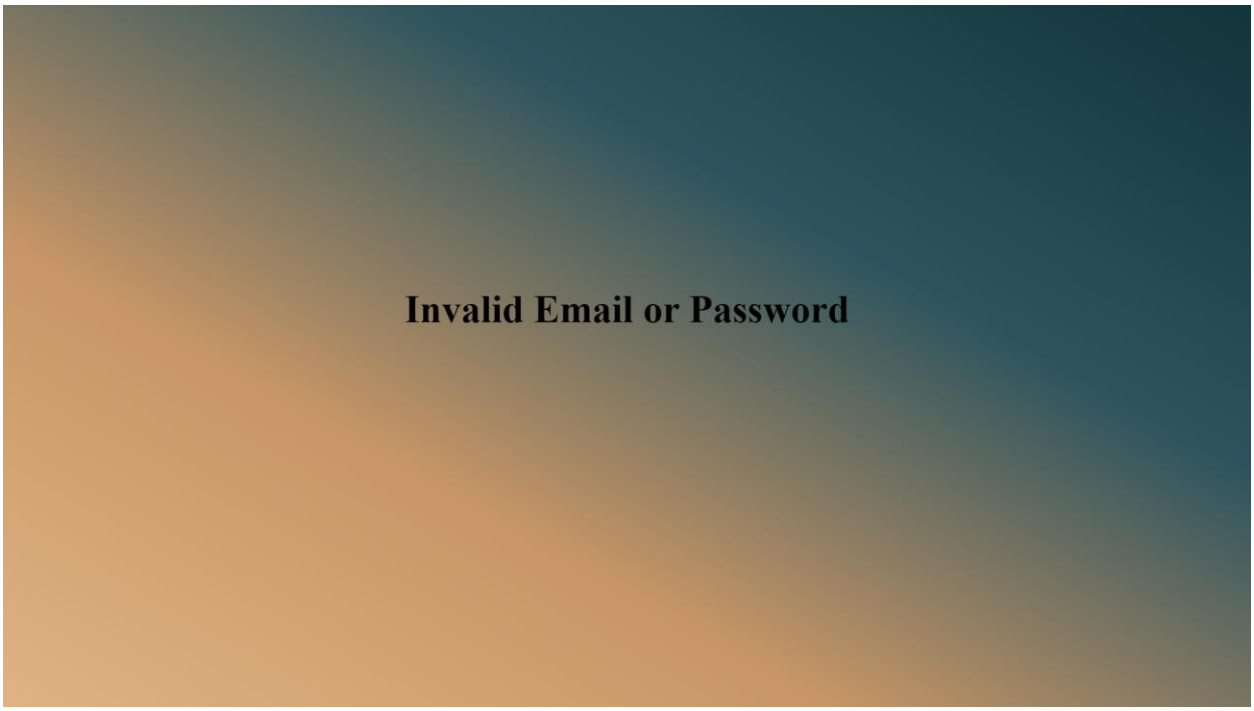
INSTRUCTOR ID	NAME	AGE	GENDER	DEPARTMENT
5501	John	55	Male	MATH
5502	Mark	40	Male	MATH
5503	Rick	39	Male	MATH
5504	Liu	52	Female	MATH
5505	Sim	51	Female	MATH
5506	Ashley	48	Female	CS
5507	Riley	49	Female	CS
5508	Wade	43	Wade	CS
5509	Ivan	42	Male	CS
5510	Dan	60	Male	CS

User Login Functionality : To Sign In , enter email and password, if student is already enrolled in any course then status will change to SUCCESSFUL LOGIN!

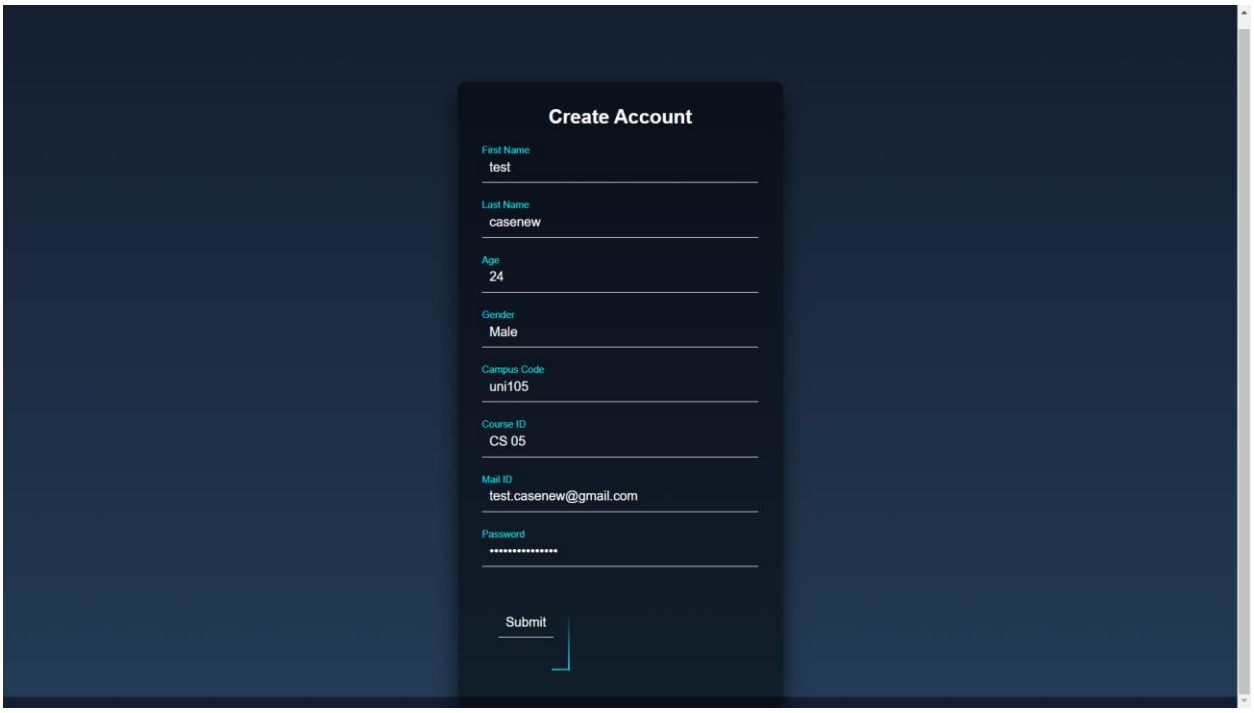
If student is not enrolled , then a prompt will be displayed stating INVALID USERNAME OR PASSWORD then student must click on sign up to access registration form

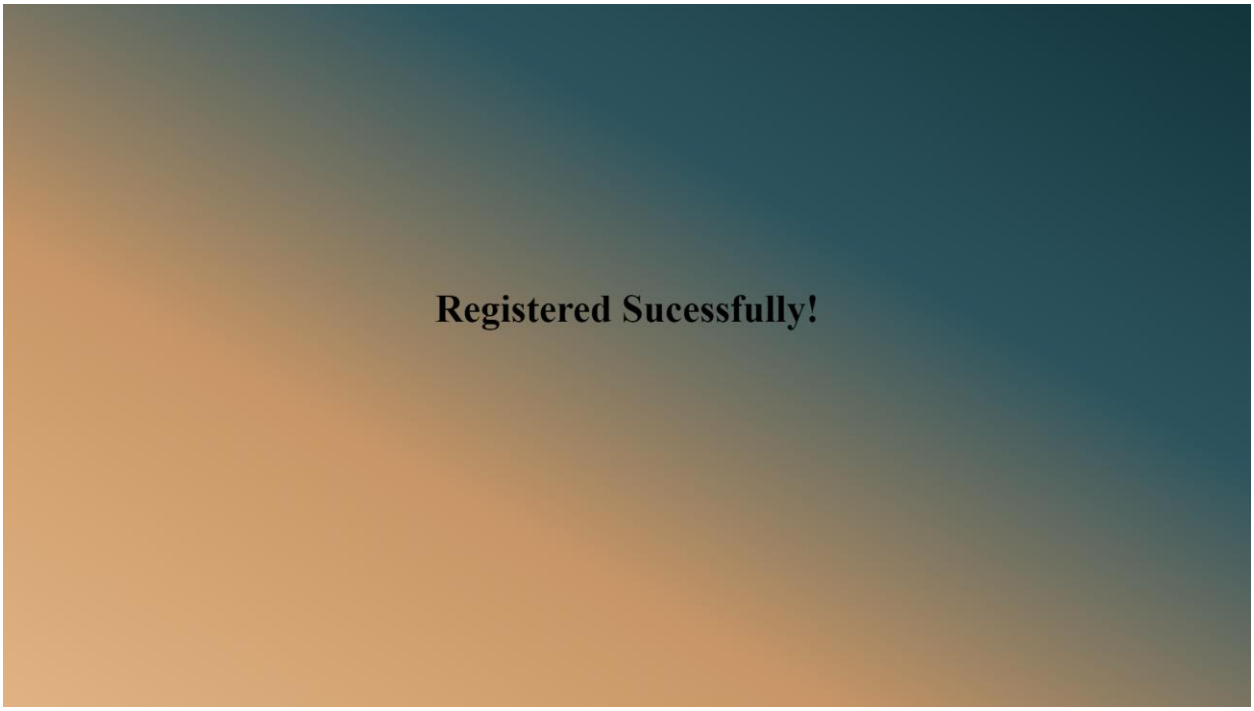


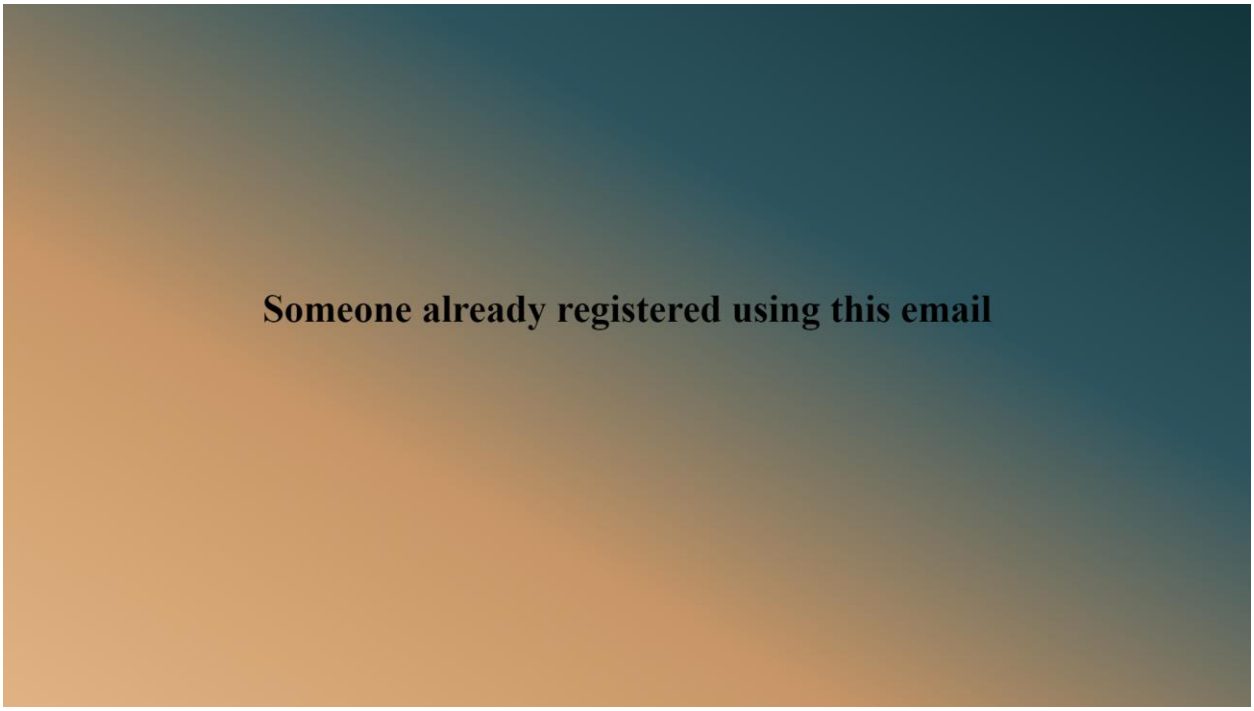




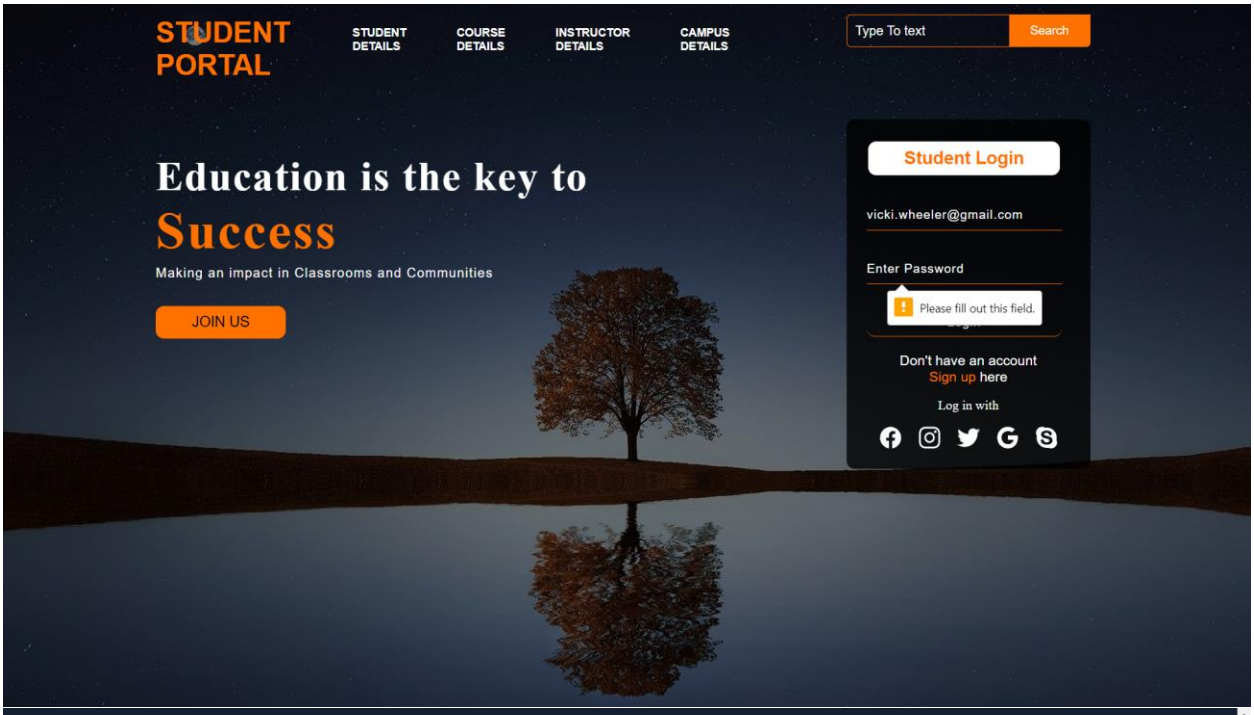
Registration Form : To enroll in any course student must fill all details in student form and click SUBMIT. When a new student is trying to enroll in any course his email ID must be unique. If a mail ID already exists then it shows that SOMEONE ALREADY REGISTERED USING THIS MAIL ID. If email id is unique then Registration will be successful.

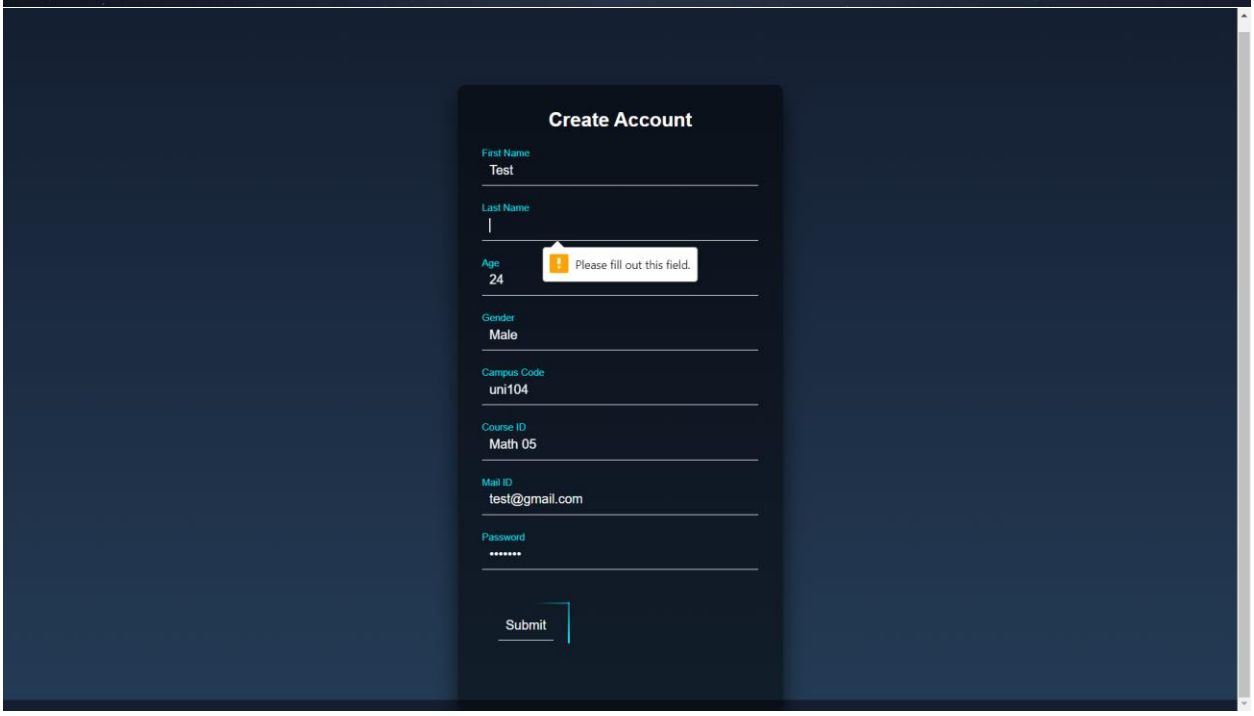

Create Account
First Name
test
Last Name
casenew
Age
24
Gender
Male
Campus Code
uni105
Course ID
CS 05
Mail ID
test.casenew@gmail.com
Password
Submit


Registered Sucessfully!



While a LOGIN or Registration student must enter all the fields, if failed to do so, an error message is displayed to fill out the field details.





CODE :

main.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Webpage Design</title>
    <link rel="stylesheet" href="main.css">
</head>
<body>

    <div class="main">
        <div class="navbar">
            <div class="icon">
                <h2 class="logo">STUDENT PORTAL</h2>
            </div>

            <div class="menu">
                <ul>
                    <li><a href="student.php">STUDENT DETAILS</a></li>
                    <li><a href="course.php">Course Details</a><br></li>
                    <li><a href="instructor.php">Instructor Details</a><br></li>
                    <li><a href="campus.php">Campus Details</a></li>

                </ul>
            </div>

            <div class="search">
                <input class="srch" type="search" name="" placeholder="Type To text">
                <a href="#"> <button class="btn">Search</button></a>
            </div>
```

```
</div>
<div class="content">
  <h1>Education is the key to <br><span>Success</span> <br></h1>
   <p class="par">Making an impact in Classrooms and Communities</p>

      <button class="cn"><a href="#">JOIN US</a></button>

      <div class="form">
        <form action="login.php" method="post">
          <h2>Student Login</h2>
          <div class="form-group">
            <input type="email" id="email" class="form-control" name="email" placeholder="Enter Email"  required />
          </div>
          <div class="form-group">
            <input type="password" id="password" class="form-control" name="password" placeholder="Enter Password " required />
          </div>
          <input type="submit" class="btnn" value="Login" name="">
        </form>


        <p class="link">Don't have an account<br>
        <a href="form.html">Sign up </a> here</a></p>
        <p class="liw">Log in with</p>

        <div class="icons">
          <a href="https://www.facebook.com/login/"><ion-icon name="logo-facebook"></ion-icon></a>
          <a href="https://www.instagram.com/accounts/login/"><ion-icon name="logo-instagram"></ion-icon></a>
```

```
                <a href="https://twitter.com/i/flow/login"><ion-icon name="logo-twitter"></ion-icon></a>

                <a href="https://accounts.google.com/v3/signin/identifier?dsh=S-1139991351%3A1667892009997462&continue=https%3A%2F%2Fmail.google.com%2Fmail%2F&rip=1&sacu=1&service=mail&flowName=GlifWebSignIn&flowEntry=ServiceLogin&ifkv=ARgdvAtDAhXRNOIiKN9tKayXRXBGtPmDMJMSVdDlaNsov7K-nHAevizYpBT0q5kw1dSgQLaW2wY"><ion-icon name="logo-google"></ion-icon></a>

                <a href="https://login.live.com/login.srf?wa=wsignin1.0&rpsnv=13&ct=1667892045&rver=7.1.6819.0&wp=MBI_SSL&wreply=https%3A%2F%2Flw.skype.com%2Flogin%2Foauth%2Fproxy%3Fclient_id%3D360605%26redirect_uri%3Dhttps%253A%252F%252Fsecure.skype.com%252Fportal%252Flogin%253Freturn_url%253Dhttps%25253A%25252F%25252Fsecure.skype.com%25252Fportal%25252Foverview%26response_type%3Dpostgrant%26state%3Dc9a3f048cf8d4083e502ab87&lc=1033&id=293290&mkt=en-US&psi=skype&lw=1&cobrandid=2befc4b5-19e3-46e8-8347-77317a16a5a5&client_flight=ReservedFlight33%2CReservedFlight67"><ion-icon name="logo-skype"></ion-icon></a>

            </div>


        </div>

            </div>

        </div>

    </div>

  </div>

  <script src="https://unpkg.com/ionicons@5.4.0/dist/ionicons.js"></script>

</body>

</html>
```

main.css

```
*{

  margin: 0;

  padding: 0;

}
```

```
.main{
    width: 100%;
    background: linear-gradient(to top, rgba(0,0,0,0.5)50%,rgba(0,0,0,0.5)50%),
url(https://i.postimg.cc/gdS1PGkm/1.jpg);
    background-position: center;
    background-size: cover;
    height: 100vh;
}

.navbar{
    width: 1200px;
    height: 75px;
    margin: auto;
}

.icon{
    width: 200px;
    float: left;
    height: 70px;
}

.logo{
    color: #ff7200;
    font-size: 35px;
    font-family: Arial;
    padding-left: 20px;
    float: left;
    padding-top: 10px;
    margin-top: 5px
}
```

```
.menu{
    width: 400px;
    float: left;
    height: 70px;
}

ul{
    float: left;
    display: flex;
    justify-content: center;
    align-items: center;
}

ul li{
    list-style: none;
    margin-left: 62px;
    margin-top: 27px;
    font-size: 14px;
}

ul li a{
    text-decoration: none;
    color: #fff;
    font-family: Arial;
    font-weight: bold;
    transition: 0.4s ease-in-out;
    text-transform: uppercase;
}
```

```
ul li a:hover{
    color: #ff7200;
}

.search{
    width: 330px;
    float: left;
    margin-left: 270px;
}

.srch{
    font-family: 'Times New Roman';
    width: 200px;
    height: 40px;
    background: transparent;
    border: 1px solid #ff7200;
    margin-top: 13px;
    color: #fff;
    border-right: none;
    font-size: 16px;
    float: left;
    padding: 10px;
    border-bottom-left-radius: 5px;
    border-top-left-radius: 5px;
}

.btn{
    width: 100px;
    height: 40px;
    background: #ff7200;
```

```
    border: 2px solid #ff7200;
    margin-top: 13px;
    color: #fff;
    font-size: 15px;
    border-bottom-right-radius: 5px;
    border-bottom-right-radius: 5px;
    transition: 0.2s ease;
    cursor: pointer;
}
.btn:hover{
    color: #000;
}

.btn:focus{
    outline: none;
}

.srch:focus{
    outline: none;
}

.content{
    width: 1200px;
    height: auto;
    margin: auto;
    color: #fff;
    position: relative;
}

.content .par{
```

```
    padding-left: 20px;
    padding-bottom: 25px;
    font-family: Arial;
    letter-spacing: 1.2px;
    line-height: 30px;
}

.content h1{
    font-family: 'Times New Roman';
    font-size: 50px;
    padding-left: 20px;
    margin-top: 9%;
    letter-spacing: 2px;
}

.content .cn{
    width: 160px;
    height: 40px;
    background: #ff7200;
    border: none;
    margin-bottom: 10px;
    margin-left: 20px;
    font-size: 18px;
    border-radius: 10px;
    cursor: pointer;
    transition: .4s ease;

}

.content .cn a{
```

```
    text-decoration: none;
    color: #000;
    transition: .3s ease;
}

.cn:hover{
    background-color: #fff;
}

.content span{
    color: #ff7200;
    font-size: 65px
}

.form{
    width: 250px;
    height: 380px;
    background: linear-gradient(to top, rgba(0,0,0,0.8)50%,rgba(0,0,0,0.8)50%);
    position: absolute;
    top: -20px;
    left: 870px;
    transform: translate(0%,-5%);
    border-radius: 10px;
    padding: 25px;
}

.form h2{
    width: 220px;
    font-family: sans-serif;
    text-align: center;
```

```
    color: #ff7200;
    font-size: 22px;
    background-color: #fff;
    border-radius: 10px;
    margin: 2px;
    padding: 8px;
}

.form input{
    width: 240px;
    height: 35px;
    background: transparent;
    border-bottom: 1px solid #ff7200;
    border-top: none;
    border-right: none;
    border-left: none;
    color: #fff;
    font-size: 15px;
    letter-spacing: 1px;
    margin-top: 30px;
    font-family: sans-serif;
}

.form input:focus{
    outline: none;
}

::placeholder{
    color: #fff;
    font-family: Arial;
```

```
}

.btnn{
    width: 240px;
    height: 40px;
    background: #ff7200;
    border: none;
    margin-top: 30px;
    font-size: 18px;
    border-radius: 10px;
    cursor: pointer;
    color: #fff;
    transition: 0.4s ease;
}
.btnn:hover{
    background: #fff;
    color: #ff7200;
}
.btnn a{
    text-decoration: none;
    color: #000;
    font-weight: bold;
}
.form .link{
    font-family: Arial, Helvetica, sans-serif;
    font-size: 17px;
    padding-top: 20px;
    text-align: center;
}
.form .link a{
```

```
    text-decoration: none;
    color: #ff7200;
}
.liw{
    padding-top: 15px;
    padding-bottom: 10px;
    text-align: center;
}
.icons a{
    text-decoration: none;
    color: #fff;
}
.icons ion-icon{
    color: #fff;
    font-size: 30px;
    padding-left: 14px;
    padding-top: 5px;
    transition: 0.3s ease;
}
.icons ion-icon:hover{
    color: #ff7200;
}
```

STUDENT.PHP

```
<html>
<head>
        <style type="text/css">
                body{
```

```
background: rgb(233,76,161);
background: -moz-linear-gradient(90deg, rgba(233,76,161,1) 0%, rgba(199,74,233,1) 100%);
background: -webkit-linear-gradient(90deg, rgba(233,76,161,1) 0%, rgba(199,74,233,1) 100%);
background: linear-gradient(90deg, rgba(233,76,161,1) 0%, rgba(199,74,233,1) 100%);
filter:
progid:DXImageTransform.Microsoft.gradient(startColorstr="#e94ca1",endColorstr="#c74ae9",Gradient
Type=1);
opacity: .9999;
}

table{
 position: absolute;
 z-index: 2;
 left: 13%;
 top: 15%;
 border-collapse: collapse;
 border-spacing: 0;
 box-shadow: 0 2px 15px rgba(64,64,64,.7);
 border-radius: 12px 12px 0 0;
 overflow: hidden;

}

td , th{
    border: none;
 padding: 15px 20px;
 text-align: center;



}
th{
```

```
 background-color: #ba68c8;
 color: #fafafa;
 font-family: 'Open Sans',Sans-serif;
 font-weight: 200;
 text-transform: uppercase;

}
tr{
 width: 100%;
 background-color: #fafafa;
 font-family: 'Montserrat', sans-serif;
}
tr:nth-child(even){
 background-color: #eeeeee;
}

h1 {  color: #fafafa;
 font-family: 'Open Sans',Sans-serif;
 font-weight: 200; font-size: 30px;
   line-height: 72px; margin: 0 0 24px; text-align: center; text-transform: uppercase; }

            </style>

            <title>Student Deatils</title>

</head>
<body>

<H1>Student Deatils</H1>
```

```
<table border="1">
        <tr>
                <th>student id</th>
                <th>first name</th>
                <th>last name</th>
                <th>gender</th>
                <th>age</th>
                <th>Mail ID</th>
                <th>Campus Code</th>
                <th>Course id</th>



        </tr>




</body>
</html>



<?php
        $connect=mysqli_connect("localhost","root","","enrolldata") or die("Connection Failed");
        $query="SELECT * from student";
        $result=mysqli_query($connect,$query);
        while($row=mysqli_fetch_assoc($result))
        {
                ?>
                <tr>
                        <td><?php echo $row['student_id']?></td>
                        <td><?php echo $row['first_name']?></td>
```

```
                <td><?php echo $row['last_name']?></td>
                <td><?php echo $row['gender']?></td>
                <td><?php echo $row['age']?></td>
                <td><?php echo $row['email']?></td>
                <td><?php echo $row['campcode']?></td>
                <td><?php echo $row['courseid']?></td>
            </tr>
            <?php
        }
?>
```

COURSE.PHP

```
<html>
<head>

    <title>Student Deatils</title>
     <link rel="stylesheet" href="style.css">
</head>
<body>
<h1>Course Deatils</h1>

<table border="1">
    <tr>
            <th>course id</th>
            <th>course name</th>
            <th>course duration</th>
            <th>credits</th>
```

```
                <th>instructor id</th>



        </tr>




</body>
</html>



<?php
        $connect=mysqli_connect("localhost","root","","enrolldata") or die("Connection Failed");
        $query="SELECT * from course";
        $result=mysqli_query($connect,$query);
        while($row=mysqli_fetch_assoc($result))
        {
                ?>
                <tr>
                        <td><?php echo $row['course_id']?></td>
                        <td><?php echo $row['course_name']?></td>
                        <td><?php echo $row['course_duration']?></td>
                        <td><?php echo $row['credits']?></td>
                        <td><?php echo $row['instructorid']?></td>



                </tr>
                <?php
        }
?>
```

INSTRUCTOR.PHP

```
<html>
<head>

        <title>Student Deatils</title>
         <link rel="stylesheet" href="style.css">
</head>
<body>
<h1>Instructor Deatils</h1>
<table border="1">
        <tr>
                <th>instructor id</th>
                <th>name</th>
                <th>age</th>
                <th>gender</th>
                <th>department</th>



        </tr>


</body>
</html>
```

```
<?php
	$connect=mysqli_connect("localhost","root","","enrolldata") or die("Connection Failed");
	$query="SELECT * from instructor";
	$result=mysqli_query($connect,$query);
	while($row=mysqli_fetch_assoc($result))
	{
		?>
		<tr>
			<td><?php echo $row['instructor_id']?></td>
			<td><?php echo $row['name']?></td>
			<td><?php echo $row['age']?></td>
			<td><?php echo $row['gender']?></td>
			<td><?php echo $row['department']?></td>
		</tr>
		<?php
	}
?>
```

CAMPUS.PHP

```
<html>
<head>
	<style type="text/css">
		table{
 position: absolute;
 z-index: 2;
 left: 15%;
 top: 15%;
```

```
 border-collapse: collapse;
 border-spacing: 0;
 box-shadow: 0 2px 15px rgba(64,64,64,.7);
 border-radius: 12px 12px 0 0;
 overflow: hidden;

}
        </style>

        <title>Student Deatils</title>
         <link rel="stylesheet" href="style.css">
</head>
<body>
<h1>Campus Deatils</h1>
<table border="1">
        <tr>
                <th>campus_code</th>
                <th>campus_name</th>
                <th>location</th>
                <th>capacity</th>



        </tr>

</body>
</html>




<?php
```

```
        $connect=mysqli_connect("localhost","root","","enrolldata") or die("Connection Failed");
        $query="SELECT * from campus";
        $result=mysqli_query($connect,$query);
        while($row=mysqli_fetch_assoc($result))
        {
                ?>
                <tr>
                        <td><?php echo $row['campus_code']?></td>
                        <td><?php echo $row['campus_name']?></td>
                        <td><?php echo $row['location']?></td>
                        <td><?php echo $row['capacity']?></td>

                </tr>
                <?php
        }
?>
```

STYLE.CSS

```
@import url('https://fonts.googleapis.com/css?family=Montserrat|Open+Sans|Roboto');
*{
 margin:0;
 padding: 0;
 outline: 0;
}

.filter{
 position: absolute;
 left: 0;
 top: 0;
```

```
 bottom: 0;
 right: 0;
 z-index: 1;
 background: rgb(233,76,161);
background: -moz-linear-gradient(90deg, rgba(233,76,161,1) 0%, rgba(199,74,233,1) 100%);
background: -webkit-linear-gradient(90deg, rgba(233,76,161,1) 0%, rgba(199,74,233,1) 100%);
background: linear-gradient(90deg, rgba(233,76,161,1) 0%, rgba(199,74,233,1) 100%);
filter:
progid:DXImageTransform.Microsoft.gradient(startColorstr="#e94ca1",endColorstr="#c74ae9",Gradient
Type=1);
opacity: .9999;


}
body{
background: rgb(233,76,161);
background: -moz-linear-gradient(90deg, rgba(233,76,161,1) 0%, rgba(199,74,233,1) 100%);
background: -webkit-linear-gradient(90deg, rgba(233,76,161,1) 0%, rgba(199,74,233,1) 100%);
background: linear-gradient(90deg, rgba(233,76,161,1) 0%, rgba(199,74,233,1) 100%);
filter:
progid:DXImageTransform.Microsoft.gradient(startColorstr="#e94ca1",endColorstr="#c74ae9",Gradient
Type=1);
opacity: .9999;
}
table{
 position: absolute;
 z-index: 2;
 left: 15%;
 top: 15%;
 border-collapse: collapse;
 border-spacing: 0;
 box-shadow: 0 2px 15px rgba(64,64,64,.7);
```

```
 border-radius: 12px 12px 0 0;
 overflow: hidden;

}

td , th{
    border: none;
 padding: 15px 20px;
 text-align: center;



}
th{
 background-color: #ba68c8;
 color: #fafafa;
 font-family: 'Open Sans',Sans-serif;
 font-weight: 200;
 text-transform: uppercase;

}
tr{
 width: 100%;
 background-color: #fafafa;
 font-family: 'Montserrat', sans-serif;
}
tr:nth-child(even){
 background-color: #eeeeee;
}

h1 {  color: #fafafa;
```

```
 font-family: 'Open Sans',Sans-serif;
 font-weight: 200; font-size: 30px;
   line-height: 72px; margin: 0 0 24px; text-align: center; text-transform: uppercase; }
```

LOGIN.PHP

```
<html>
<head>
    <style type="text/css">
        h2 {
                position: relative;
            top: 40%;
            font-size: 3em;
            text-align: center;


    }
    body{
      background: linear-gradient(to top right, #e1b382,#c89666,#2d545e,#12343b);
    }
    </style>
    <title></title>

</head>
<body>

</body>
</html>
```

```
<?php


    $email = $_POST['email'];
    $password = $_POST['password'];
    $con = new mysqli("localhost","root","","enrolldata");
    if($con->connect_error)
    {
        die("Failed to connect : ".$conn->connect_error);

    }
    else
    {

        $stmt = $con->prepare("SELECT * from student where email = ?");
        $stmt->bind_param("s", $email);
        $stmt->execute();
        $stmt_result=$stmt->get_result();
        if($stmt_result->num_rows > 0)
        {

            $data = $stmt_result -> fetch_assoc();
            if($data["password"] === $password)
            {
                echo "<h2>Login Successful!<h2>";
```

```
            } else{
                echo "<h2>Invalid Email or Password<h2>";
            }
        }
        else
        {
            echo "<h2>Invalid Email or Password<h2>";
        }
        }
?>
```

FOR.HTML

```
<html>
<head>
  <style type="text/css">

  </style>

  <title></title>
   <link rel="stylesheet" href="form2.css">
</head>
<body>

<div class="login-box">
  <h2>Create Account</h2>
  <form  action="form.php" method="POST" >
    <div class="user-box">
      <input type="text" name="first_name" required>
```

```
  <label>First Name</label>
</div>
<div class="user-box">
  <input type="last_name" name="last_name" required>
  <label>Last Name</label>
</div>
    <div class="user-box">
  <input type="age" name="age" required></td>

  <label>Age</label>
</div>
<div class="user-box">
  <input type="text" name="gender" required></td>
  <label>Gender</label>
</div>
<div class="user-box">
  <input type="text" name="campcode" required></td>
  <label>Campus Code</label>
</div>
<div class="user-box">
  <input type="text" name="courseid" required></td>
  <label>Course ID</label>
</div>
<div class="user-box">
<input type="email" name="email" required>
  <label>Mail ID</label>
</div>
<div class="user-box">
<input type="password" name="password" required>
```

```
      <label>Password</label>
    </div>
  <div class="user-box">
      <a href="#">
      <span></span>
      <span></span>
      <span></span>
      <span></span>
      <input type="submit" value="Submit" name="submit">
      </a>
  </div>

  </form>
</div>
</body>
</html>
```

FORM.CSS

```
 html {
  height: 100%;
}
body {
  margin:0;
  padding:20;
  font-family: sans-serif;
  background: linear-gradient(#141e30, #243b55);
}
.login-box {
  position: absolute;
```

```
  top: 57%;
  left: 50%;
  width: 400px;
  padding: 30px;
  transform: translate(-50%, -50%);
  background: rgba(0,0,0,.5);
  box-sizing: border-box;
  box-shadow: 0 15px 25px rgba(0,0,0,.6);
  border-radius: 10px;
}
.login-box h2 {
  margin: 0 0 30px;
  padding: 0;
  color: #fff;
  text-align: center;
}

.login-box .user-box {
  position: relative;
}
.login-box .user-box input {
  width: 100%;
  padding: 9px ;
  font-size: 16px;
  color: #fff;
  margin-bottom: 30px;
  border: none;
  border-bottom: 1px solid #fff;
  outline: none;
  background: transparent;
```

```
}

.login-box .user-box label {
  position: absolute;
  top:0;
  left: 0;
  padding: 10px 0;
  font-size: 16px;
  color: #fff;
  pointer-events: none;
  transition: .5s;
}

.login-box .user-box input:focus ~ label,
.login-box .user-box input:valid ~ label {
  top: -20px;
  left: 0;
  color: #03e9f4;
  font-size: 12px;
}

.login-box form a {
  position: relative;
  display: inline-block;
  padding: 10px 20px;
  color: #03e9f4;
  font-size: 16px;
  text-decoration: none;
  text-transform: uppercase;
  overflow: hidden;
```

```
  transition: .5s;
  margin-top: 10px;
  letter-spacing: 4px
}

.login-box a:hover {
  background: #03e9f4;
  color: #fff;
  border-radius: 5px;
  box-shadow: 0 0 5px #03e9f4,
              0 0 25px #03e9f4,
              0 0 50px #03e9f4,
              0 0 100px #03e9f4;
}

.login-box a span {
  position: absolute;
  display: block;
}

.login-box a span:nth-child(1) {
  top: 0;
  left: -100%;
  width: 100%;
  height: 2px;
  background: linear-gradient(90deg, transparent, #03e9f4);
  animation: btn-anim1 1s linear infinite;
}

@keyframes btn-anim1 {
```

```
  0% {
    left: -100%;
  }
  50%,100% {
    left: 100%;
  }
}

.login-box a span:nth-child(2) {
  top: -100%;
  right: 0;
  width: 2px;
  height: 100%;
  background: linear-gradient(180deg, transparent, #03e9f4);
  animation: btn-anim2 1s linear infinite;
  animation-delay: .25s
}

@keyframes btn-anim2 {
  0% {
    top: -100%;
  }
  50%,100% {
    top: 100%;
  }
}

.login-box a span:nth-child(3) {
  bottom: 0;
  right: -100%;
```

```
  width: 100%;
  height: 2px;
  background: linear-gradient(270deg, transparent, #03e9f4);
  animation: btn-anim3 1s linear infinite;
  animation-delay: .5s
}

@keyframes btn-anim3 {
  0% {
    right: -100%;
  }
  50%,100% {
    right: 100%;
  }
}

.login-box a span:nth-child(4) {
  bottom: -100%;
  left: 0;
  width: 2px;
  height: 100%;
  background: linear-gradient(360deg, transparent, #03e9f4);
  animation: btn-anim4 1s linear infinite;
  animation-delay: .75s
}

@keyframes btn-anim4 {
  0% {
    bottom: -100%;
  }
```

```
  50%,100% {
    bottom: 100%;
  }
}
```

FORM.PHP

```
<html>
<head>
        <style type="text/css">
        h2{
            position: relative;
            top: 40%;
            font-size: 3em;
            text-align: center;



    }

    body{
      background: linear-gradient(to top right, #e1b382,#c89666,#2d545e,#12343b);

    }
    </style>




</head>
<body>

</body>
</html>

<?php
if (isset($_POST['submit'])) {
```

```
if ( isset($_POST['first_name']) && isset($_POST['last_name'])
    && isset($_POST['age']) && isset($_POST['gender']) && isset($_POST['campcode']) && isset($_POST['courseid'])
    && isset($_POST['email']) && isset($_POST['password'])) {



    $first_name = $_POST['first_name'];
    $last_name = $_POST['last_name'];
    $age = $_POST['age'];
    $gender = $_POST['gender'];
    $campcode = $_POST['campcode'];
    $courseid = $_POST['courseid'];
    $email = $_POST['email'];
    $password = $_POST['password'];



    $host = "localhost";
    $dbUsername = "root";
    $dbPassword = "";
    $dbName = "enrolldata";

    $conn = new mysqli($host, $dbUsername, $dbPassword, $dbName);

    if ($conn->connect_error) {
        die('Could not connect to the database.');
    }
    else {
        $Select = "SELECT email FROM student WHERE email = ? LIMIT 1";
        $Insert = "INSERT INTO student( first_name, last_name, age, gender, campcode, courseid, email, password) values( ?, ?, ?, ?, ?, ?, ?, ?)";
```

```
            $stmt = $conn->prepare($Select);
            $stmt->bind_param("s", $email);
            $stmt->execute();
            $stmt->bind_result($resultEmail);
            $stmt->store_result();
            $stmt->fetch();
            $rnum = $stmt->num_rows;

            if ($rnum == 0) {
                $stmt->close();

                $stmt = $conn->prepare($Insert);
                $stmt->bind_param("ssisssss",  $first_name, $last_name, $age, $gender, $campcode, $courseid,
$email, $password);
                if ($stmt->execute()) {
                    echo "<h2>Registered Sucessfully!</h2>";
                }
                else {
                    echo $stmt->error;
                }
            }
            else {
                echo "<h2>Someone already registered using this email</h2>";
            }
            $stmt->close();
            $conn->close();
        }
    }
    else {
        echo "<h2>All field are required.</h2>";
```

```
        die();
    }
}
else {
    echo "<h2>Submit button is not set</h2>";
}
?>
```