

University Management System

This system consists of almost all the basic transactions of a typical university. Managing Departments, Professors, Students and Library are some of them.

Following are the functions that can be carried out by using this system.

Departments

1. Add Departments & Department Heads
2. Edit Departments
3. View Departments & Department Heads
4. Delete Departments

Professors

1. Add Professors
2. Edit Professors
3. View Professors
4. Add Course Books
5. Remove Professors

Students

1. Register Students
2. View Students (Allows you to view All Students at once and Graduates, Undergraduates, Non Matriculating separately)
3. Delete Students
4. Assign Lab Sessions (for graduate students)
5. Enroll in courses (for undergraduate students)
6. View Grades

Courses

1. Add Courses, Sections & Lab Sessions
2. View Courses, Sections & Lab Sessions
3. Edit Courses
4. Delete Courses

Library

1. Add Books and Copies
2. View Books
3. Borrow & Return Book Copies

Companies

1. Add Companies, Sessions & Assessments
2. Edit Companies & Company Sessions
3. View Companies & Company Sessions
4. Remove Companies & Company Sessions

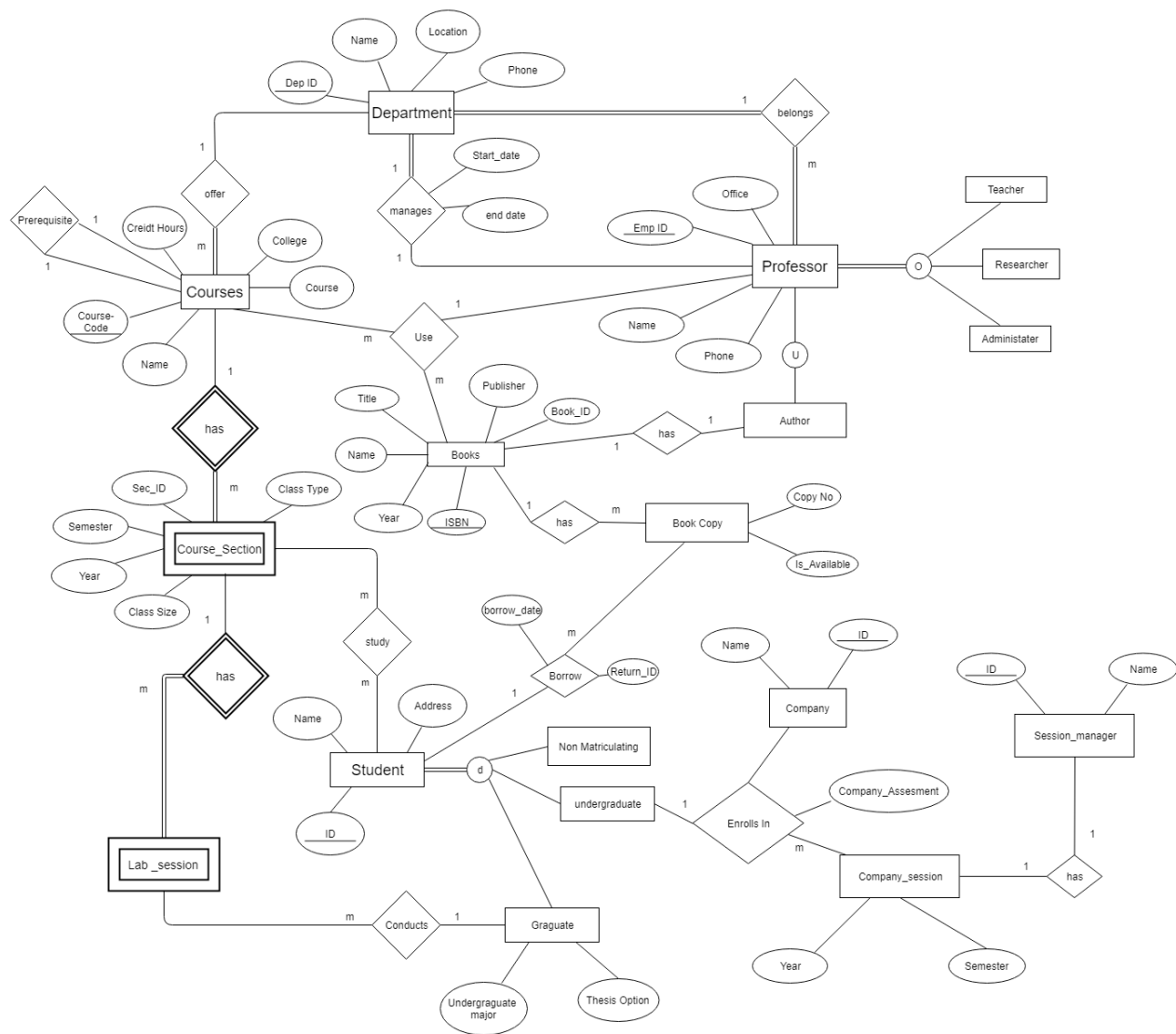
Insert, Update, Delete & View for a given function is allowed only for specific user levels.

User Levels

- | | | |
|--------------------|---|--|
| 1. Admin | : | Allowed to do any operation for any of the functions |
| 2. Student | : | Allowed to view Students, Courses, Sections, Lab Sessions & Company Assessments |
| 3. Librarian | : | Allowed to do view and insert operations in the Library . |
| 4. Company Manager | : | Allowed to do any operation with respect to Company Sessions and Assessments . |

Technologies Used : PHP, HTML, CSS, JavaScript, My SQL

ER Diagram



Assumptions

1. A book has only one author.
2. Session manager is not from the university staff

Mapping

Department (Dep_ID, Name, Location, Phone, Emp_ID)

Professor (Emp_ID, Dep_ID, Name, Office, Phone, Type)

Course (Course_Code, Course, Name, Credit_Hours, College, Prerequisite, Dep_ID)

Book (ISBN, Name, Title, Year, Author, Publisher)

Prof_Book_Course (Course_Code, Emp_ID, ISBN)

Book_Copy (Copy_No, ISBN, Is_Available)

Burrow_Book (Copy_No, ISBN, Student_ID, Burrow_Date, Return_Date)

Course_Section (Sec_ID, Course_Code, Semester, Year, Class_Size, Class_Type, Emp_ID)

Graduate (Student_ID, Name, Address, Undergraduate_Major, Thesis_Option)

Undergraduate (Student_ID, Name, Address)

Non_Matriculating (Student_ID, Name, Address)

Student_Course_Section (Course_Code, Section_ID, Student_ID)

Lab_Session (Lab_Session_ID, Section_ID, Course_Code)

Grad_Lab_Session (Section_ID, Course_Code, Lab_Session_ID, Student_ID)

Company_Session (Year, Semester, Session_Manager)

Company (Company_ID, Name)

Session_Manager (ID, Name)

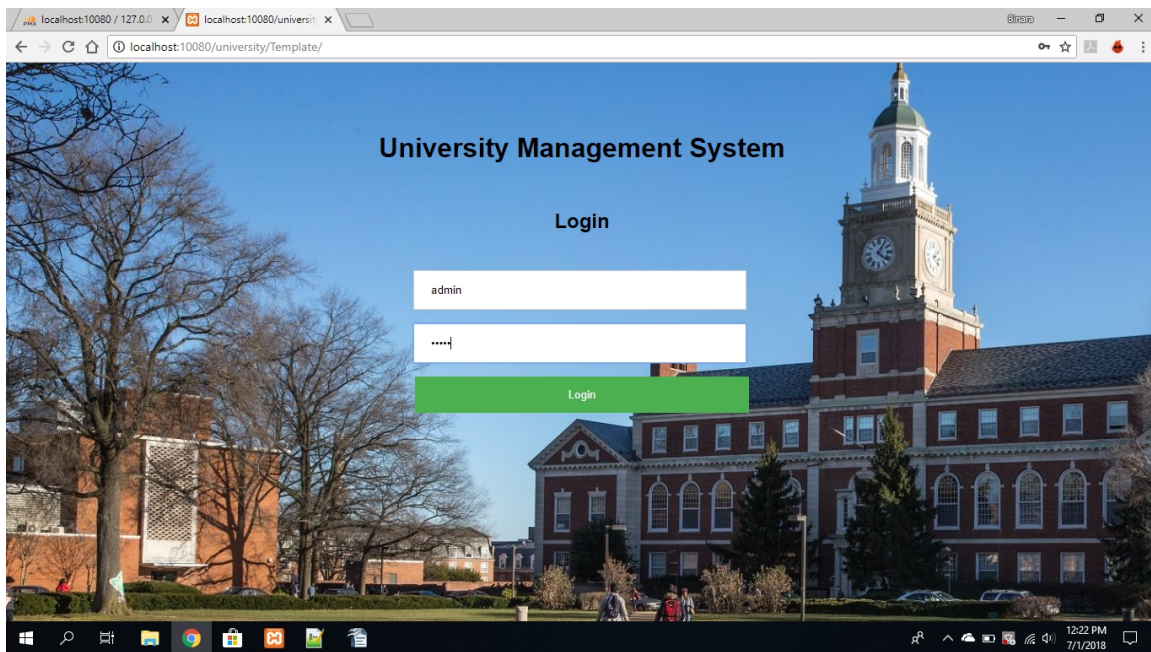
Company_Assessment (Student_ID, Company_ID, Year, Semester, Company_Assessment)

When creating the database....

- A Department_Manager table is added to store the data regarding department heads
- To identify professors as Teacher, Researcher or Administrator a new column called Type is added. Given that every professor is a Teacher so if he is a Researcher or a Administrator that is recorded in the type column.
- To reduce the number of tables, Undergraduate, Graduate and Non Matriculating students are added to the same table Student having all the columns. Here NULL is set for Undergraduate_Major and Thesis_Option of Undergraduates and Non Matriculating students. This will reduce the errors that can occur due to Primary key (Student_ID)

Use Case Diagrams

Login Page



Here you can login as four user levels.

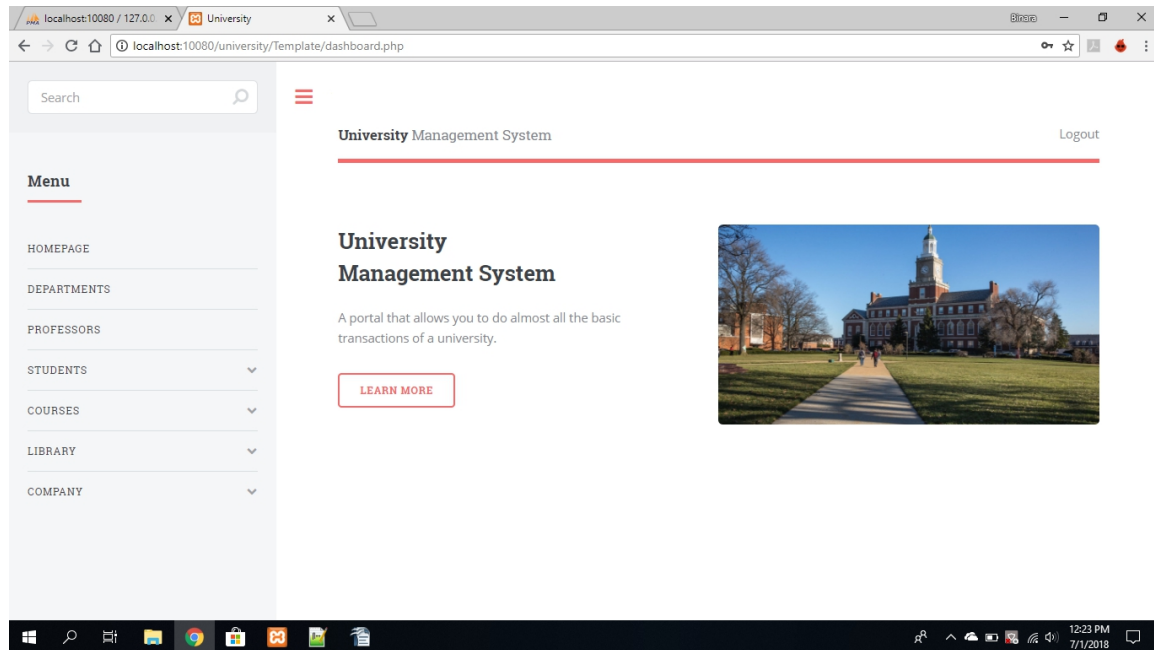
1. Administrator
2. Student
3. Librarian
4. Session Manager

SQL Query

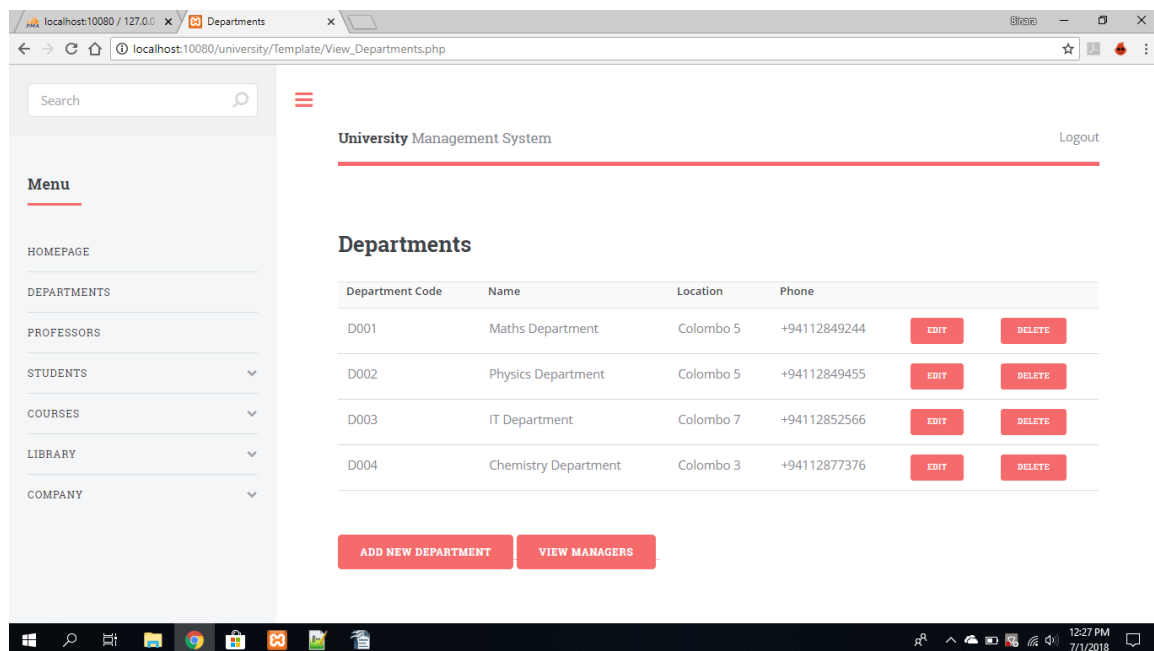
```
SELECT * FROM login WHERE username='".$user.'" AND password='".$pass.'"
```

1. Admin User Level

1.1 Dashboard



1.2 Departments



SQL query

```
SELECT * FROM department
```

Buttons

1. Add New Department : Directs you to the Add New Department page
2. Edit : Directs you to the Edit Department page
3. Delete : Prompts confirmation message
4. View Managers : Directs you to the Managers View

1.2.1 Add New Department

The screenshot shows a web browser window with the URL `localhost:10080/university/Template/Reg_Department.php`. The page is titled "University Management System" and features a sidebar menu with options like HOMEPAGE, DEPARTMENTS, PROFESSORS, STUDENTS, COURSES, LIBRARY, and COMPANY. The main content area is titled "Register Department" and contains four input fields: "Department Code", "Department Name", "Location", and "Phone". Below these fields are three buttons: "BACK", "RESET", and "SUBMIT".

Buttons

- a. Back : Go back to Department View.
- b. Reset : Resets the form. ie. Remove all inserted data.
- c. Submit : Submits the form data to the department table using following query

SQL Query

```
INSERT INTO department(Dep_Code, Name, Location, Phone) VALUES('$code', '$name', '$location', '$phone')
```

1.2.2 Edit Department

First it selects the data to be loaded into the Edit form using the following query

SQL Query

```
SELECT * FROM department WHERE Dep_Code = '$id'
```

localhost:10080 / 127.0.0.1 x Departments

localhost:10080/university/Template/Edit_Department.php?id=D001

Search

Menu

HOME PAGE

DEPARTMENTS

PROFESSORS

STUDENTS

COURSES

LIBRARY

COMPANY

University Management System

Edit Department

Department Code: D001

Department Name: Maths Department

Location: Colombo 5

Phone: +94112849244

BACK RESET CONFIRM

12:44 PM 7/1/2018

Buttons

- Back : Go back to Department View.
- Reset : Resets the form with initial data.
- Confirm : Updates the form data to the department table using following query

SQL Query

```
UPDATE department SET Name=".$name.", Location=".$location.", Phone=".$phone." WHERE Dep_Code=".$code.";
```

1.2.3 Delete Department

localhost:10080 / 127.0.0.1 x Departments

localhost:10080/university/Template/Delete_Department.php?id=D001

Search

Menu

HOME PAGE

DEPARTMENTS

PROFESSORS

STUDENTS

COURSES

LIBRARY

COMPANY

University Management System

Delete Department

Are you sure you want to delete?

CANCEL CONFIRM

12:53 PM 7/1/2018

Buttons

- a. Cancel : Go back to Department View.
- b. Confirm : Deletes the selected record using the following query. If the department is used in another table it won't get deleted.

SQL Query

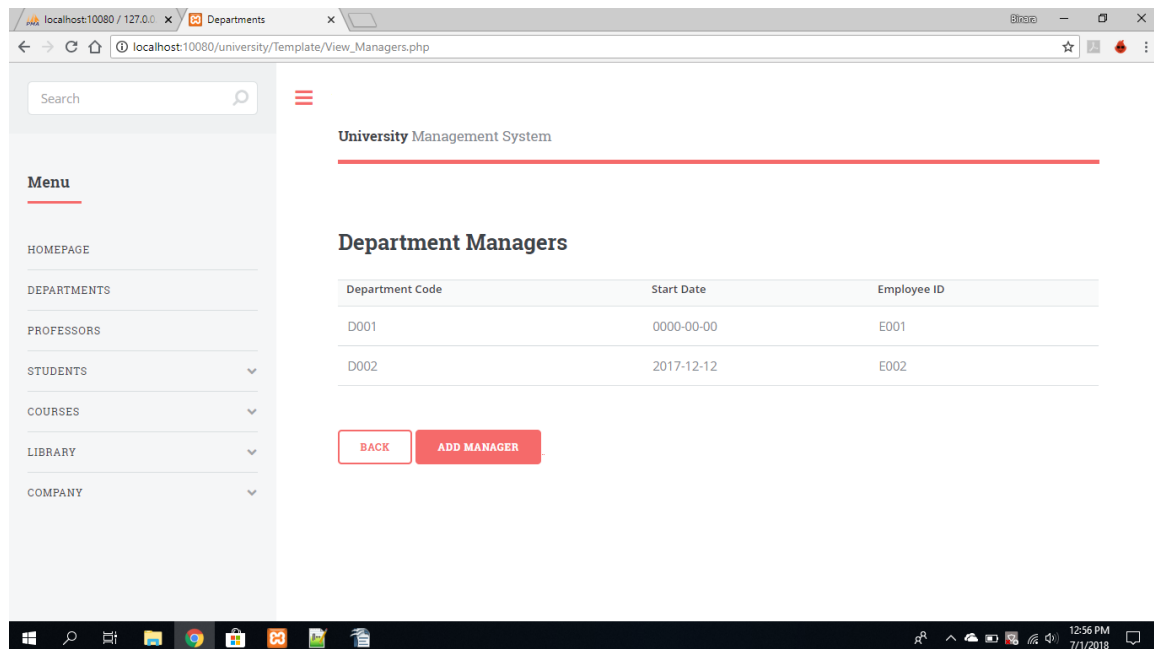
```
Delete FROM department WHERE Dep_Code='".$code.'";
```

1.2.4 View Managers

To view managers following query is used,

SQL Query

```
SELECT * FROM department_manager
```



Buttons

- a. Back : Go back to Department View.
- b. Add Manager : Directs you to the Add Manager Page.

1.2.4.b Add Department Manager

The screenshot shows a web browser window with the URL `localhost:10080/university/Template/Add_Dep_Manager.php`. The page title is "University Management System". On the left is a sidebar menu with options: HOMEPAGE, DEPARTMENTS, PROFESSORS, STUDENTS, COURSES, LIBRARY, and COMPANY. The main content area is titled "Add Department Manager" and contains three input fields: "Department Code", "Start Date", and "Employee ID". Below these fields are three buttons: "BACK", "RESET", and "CONFIRM".

Buttons

- a. Back : Go back to Department Managers View.
- b. Reset : Resets the form. ie. Remove all inserted data.
- c. Confirm : Inserts the form data to the department_manager table using following query

SQL Query

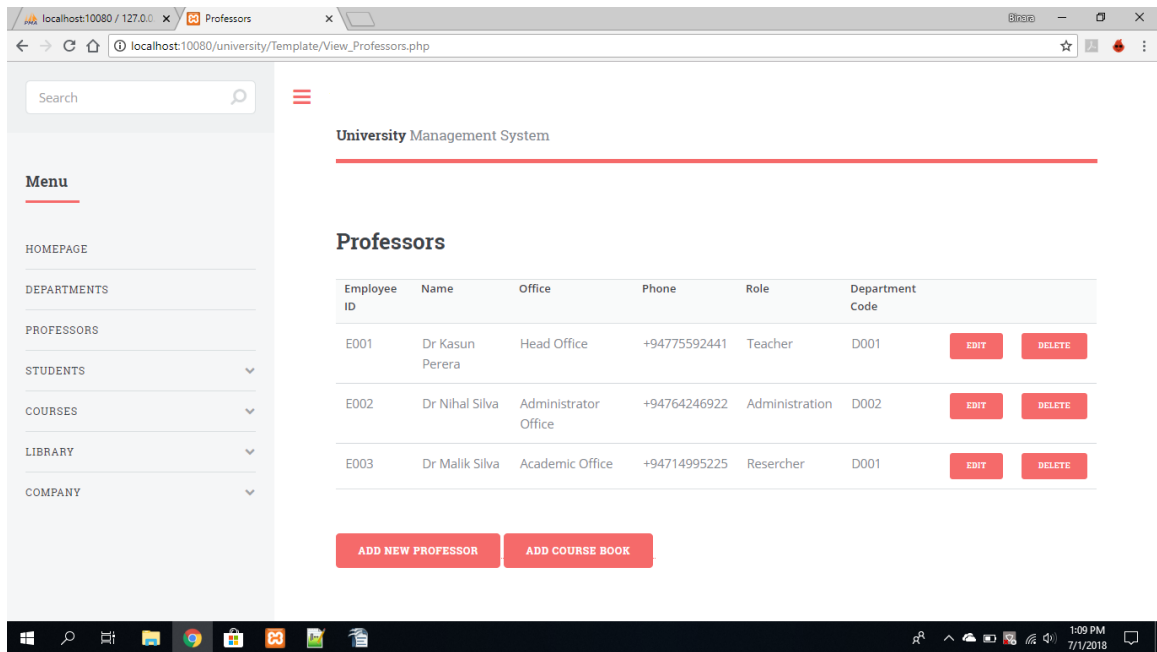
```
INSERT INTO department_manager(Dep_Code, Start_Date, Emp_ID)
VALUES('$dep_code', '$start_date', '$emp_id')
```

1.3 Professors

To view professors following query is used,

SQL query

```
SELECT * FROM professor
```



Buttons

1. Add New Professor : Directs you to the Add New Professor page
2. Edit : Directs you to the Edit Professor page
3. Delete : Prompts confirmation message
4. Add Course Book : Directs you to the Add Course Book page

1.3.1 Add New Professor

The screenshot displays the 'Professor Registration' page of the 'University Management System'. The left sidebar is identical to the previous page. The main content area features a registration form with the following fields:

- Employee ID
- Full Name
- Office
- Phone
- Role (dropdown menu)
- Department Code

Below the form are three buttons: 'BACK', 'RESET', and 'SUBMIT'. The browser's taskbar at the bottom shows the time as 1:34 PM on 7/1/2018.

Buttons

- a. Back : Go back to Professor View.
- b. Reset : Resets the form. ie. Remove all inserted data.
- c. Submit : Submits the form data to the professor table using following query

SQL Query

```
INSERT INTO professor(Emp_ID, Name, Office, Phone, Role, Dep_Code)
VALUES('$code', '$name', '$office', '$phone', '$role', '$dcode')
```

1.3.2 Edit Professor

First it selects the data to be loaded into the Edit form using the following query,

SQL Query

```
SELECT * FROM professor WHERE Emp_ID = '$id'
```

The screenshot shows a web browser window with the URL `localhost:10080/university/Template/Edit_Professor.php?id=E001`. The page title is "University Management System". On the left is a sidebar menu with options: HOME PAGE, DEPARTMENTS, PROFESSORS, STUDENTS, COURSES, LIBRARY, and COMPANY. The main content area is titled "Edit Professor" and contains a form with the following fields: Employee ID (E001), Full Name (Dr Kasun Perera), Office (Head Office), Phone (+94775592441), Role (Teacher), and Department Code (D001). At the bottom of the form are three buttons: BACK, RESET, and CONFIRM. The Windows taskbar at the bottom shows the time as 1:38 PM on 7/1/2018.

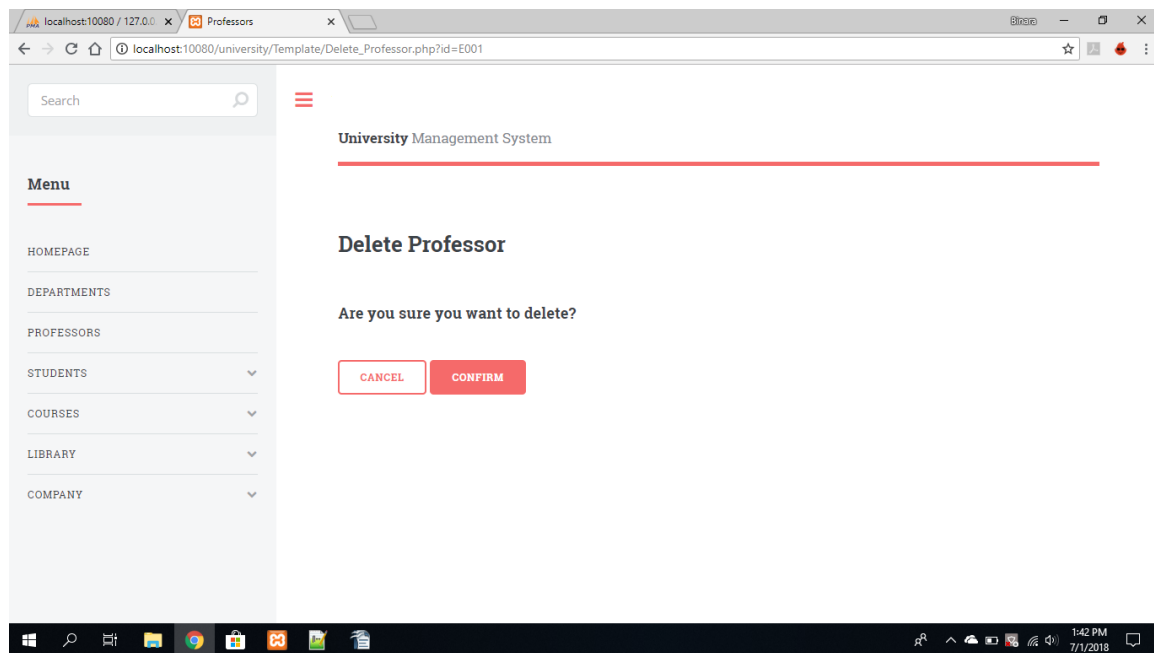
Buttons

- a. Back : Go back to Professor View.
- b. Reset : Resets the form with initial data.
- c. Confirm : Updates the form data to the professor table using following query

SQL Query

```
UPDATE professor SET Name=".$name.", Office=".$office.", Phone=".$phone.",
Role=".$role.", Dep_Code=".$DID." WHERE Emp_ID=".$EID.";
```

1.3.3 Delete Professor



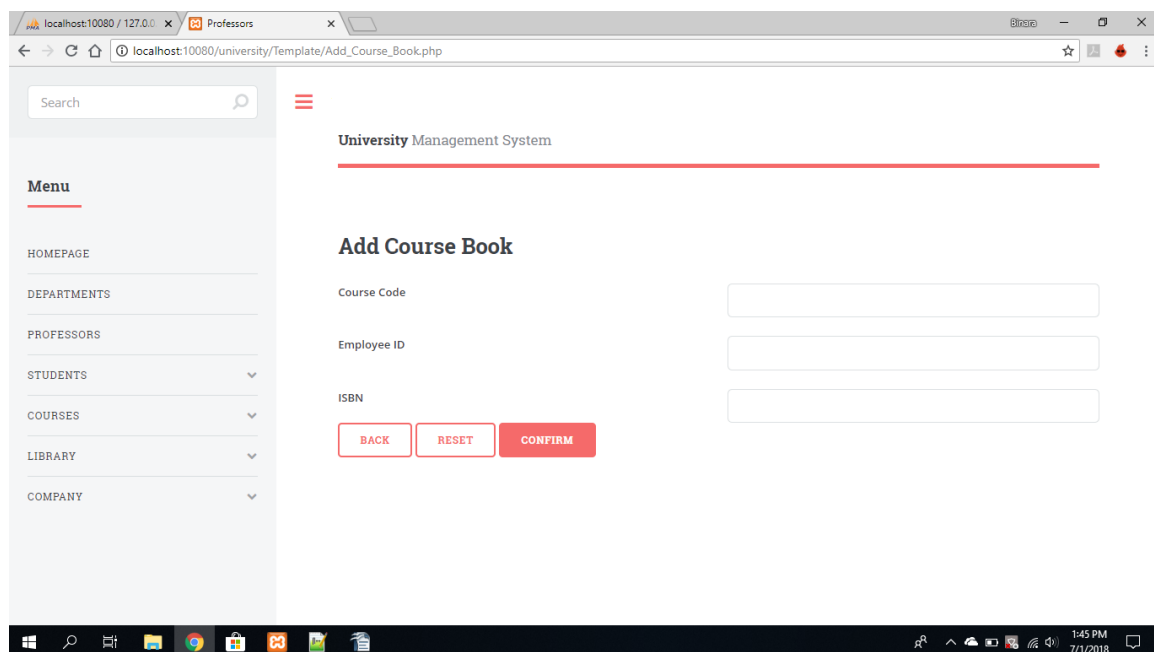
Buttons

- a. Cancel : Go back to Professor View.
- b. Confirm : Deletes the selected record using the following query. If the professor is used in another table it won't get deleted.

SQL Query

```
DELETE FROM professor WHERE Emp_ID="".$EID."";
```

1.3.4 Add Course Book



Buttons

- a. Back : Go back to Professor View.
- b. Reset : Resets the form. ie. Remove all inserted data.
- c. Confirm : Inserts the form data to the prof_book_course table using following query

SQL Query

```
INSERT INTO prof_book_course(Course_Code, Emp_ID, ISBN) VALUES('$c_code', '$emp_id', '$isbn')
```

1.4 Students

View Types

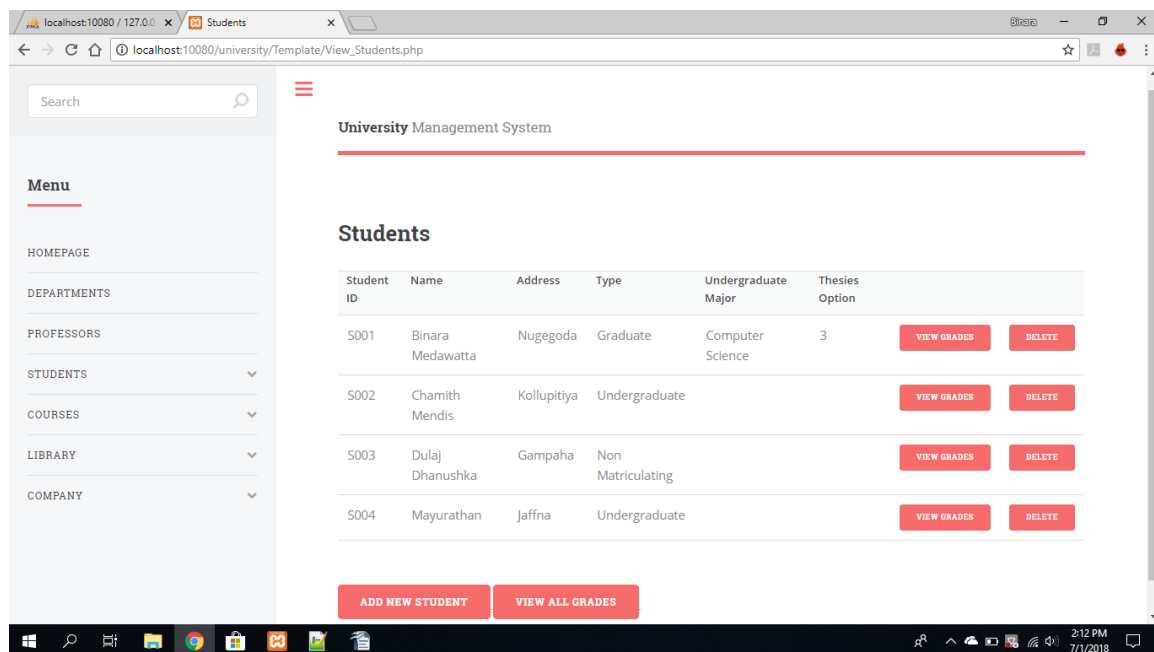
- A) View All
- B) View Graduates
- C) View Undergraduates
- D) View Non Matriculating

A. View All

To view students following query is used,

SQL query

```
SELECT * FROM student
```



Buttons

1. Add New Student : Directs you to the Add New Student page
2. Delete : Prompts confirmation message
3. View Grades : Directs you to the View Student Grade page
4. View All Grades : Directs you to the View All Grades page

1.4.A.1 Add New Student

Similar to Add New Professor page.

1.4.A.2 Delete Student

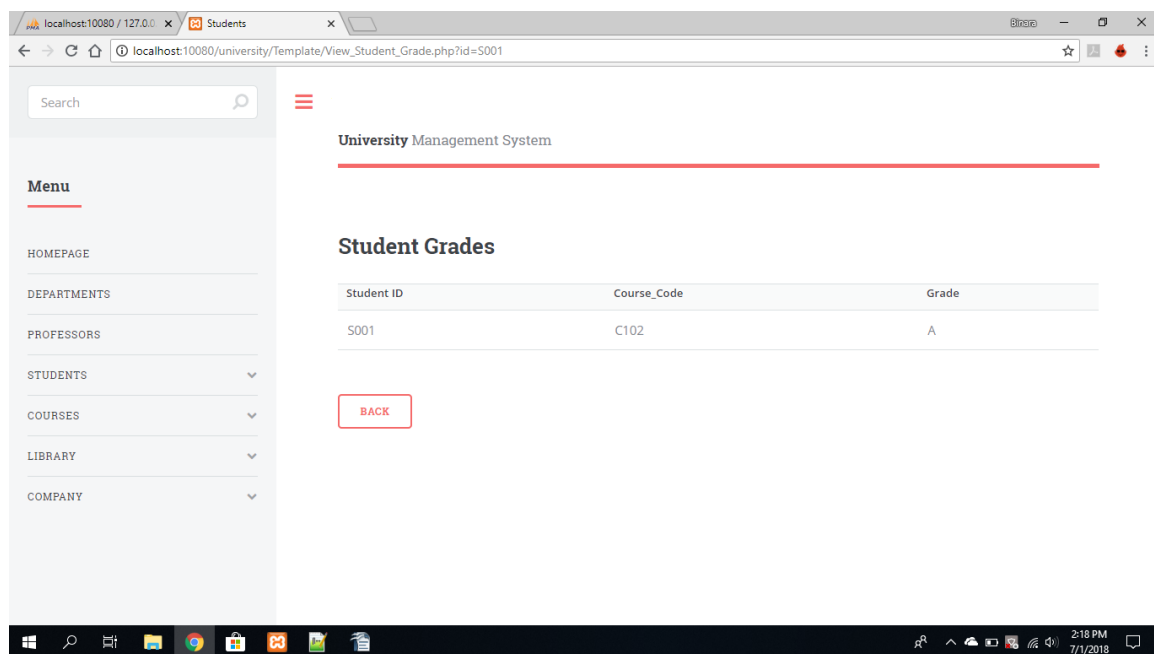
Similar to Delete Professor page.

1.4.A.3 View Grades

First it selects the data to be loaded into the Grades table using the following query,

SQL Query

```
SELECT * FROM stud_course_grade WHERE Student_ID = '$id'
```



Buttons

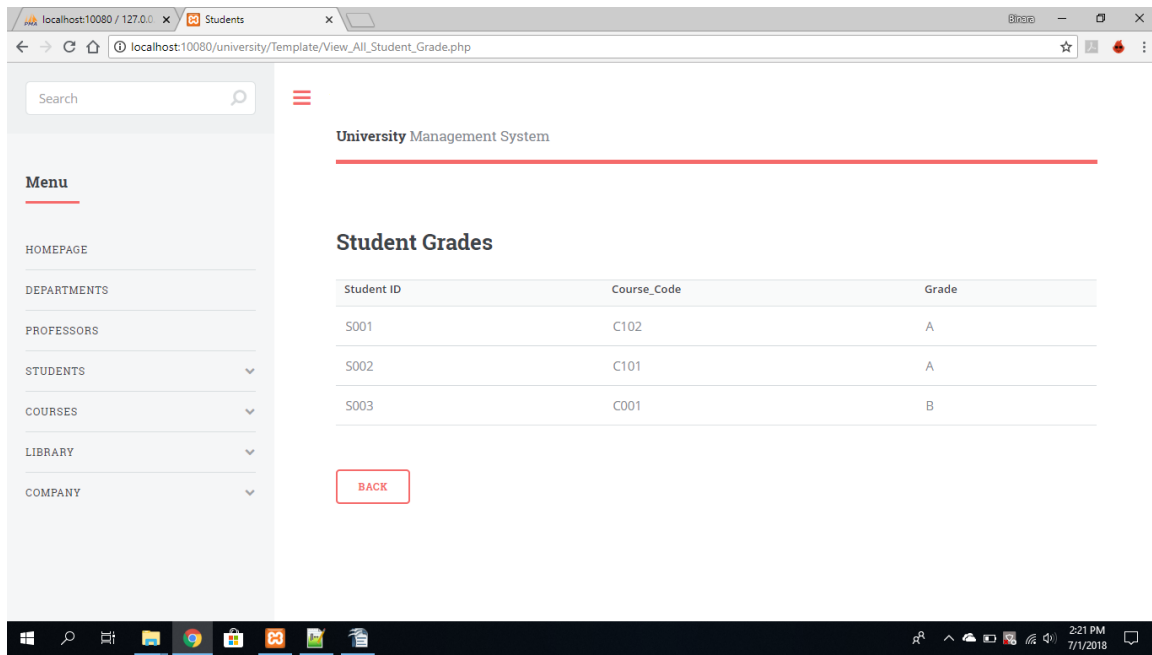
- a. Back : Go back to Student View.

1.4.A.4 View All Grades

First it selects all the data from the Grades table using the following query,

SQL Query

```
SELECT * FROM stud_course_grade
```



Buttons

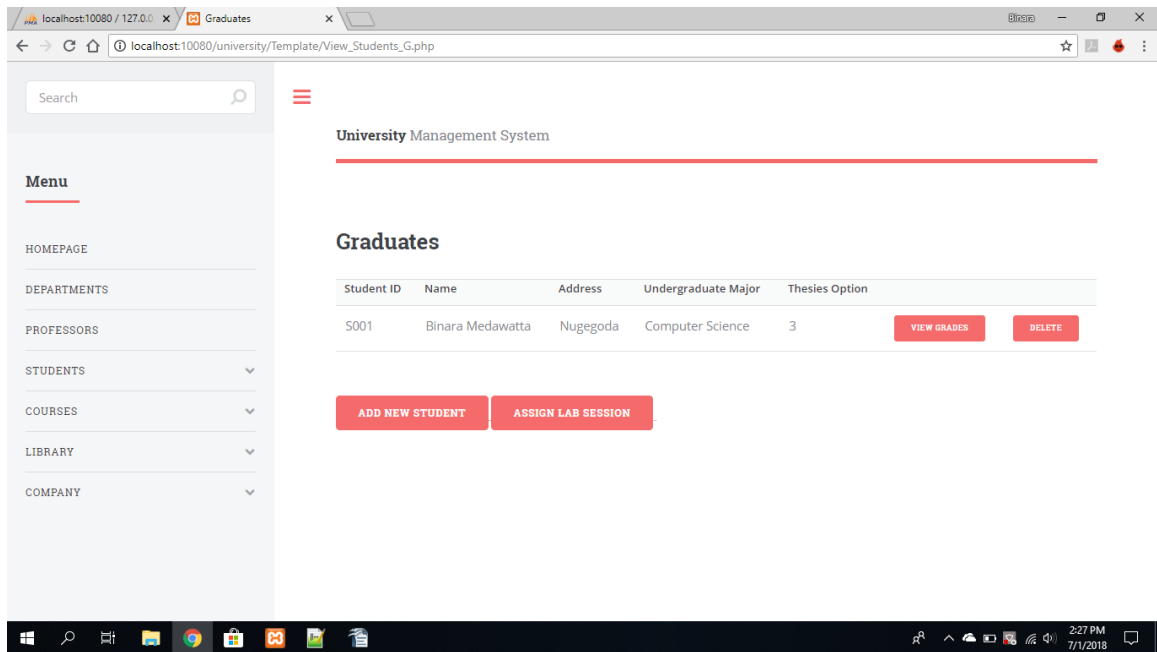
- a. Back : Go back to Student View.

B. View Graduates

To view graduates following query is used,

SQL query

```
SELECT * FROM student WHERE Type='Graduate'
```

Buttons

1. Add New Student : Directs you to the Add New Student page
2. Delete : Prompts confirmation message
3. View Grades : Directs you to the View Student Grade page
4. Assign Lab Sessions : Directs you to the Assign Lab Sessions page

1.4.B.1 Add New Student

Similar to Add New Professor page.

1.4.B.2 Delete Student

Similar to Delete Professor page.

1.4.B.3 Assign Lab Sessions

Buttons

- a. Back : Go back to Graduate View.
- b. Reset : Resets the form. ie. Remove all inserted data.
- c. Confirm : Inserts the form data to the grad_lab_session table using following query

SQL Query

```
INSERT INTO grad_lab_session(Section_ID, Course_Code, Lab_Session_ID, Student_ID)
VALUES('$section_number', '$course_code', '$lab_session_id', '$s_id')
```

B. View Undergraduates

Works the same as Student View.

C. View Non Matriculating Students

Works the same as Student View.

1.5 Courses

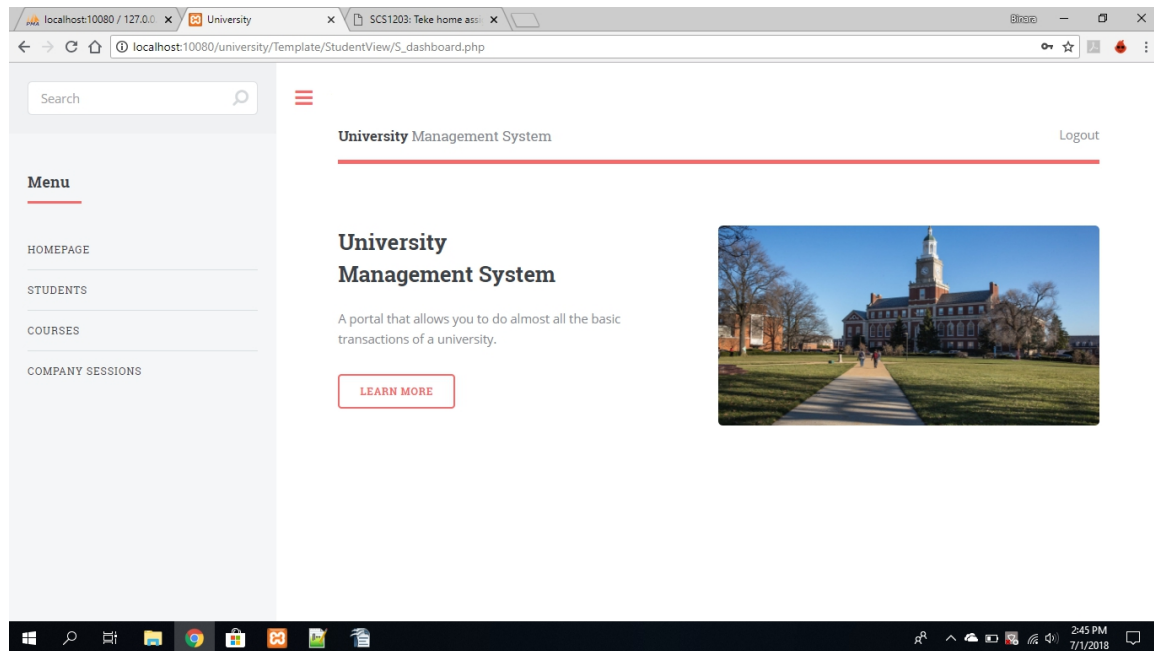
Works the same as Professor View.

1.6 Library

1.7 Company

2. Student User Level

2.1 Dashboard

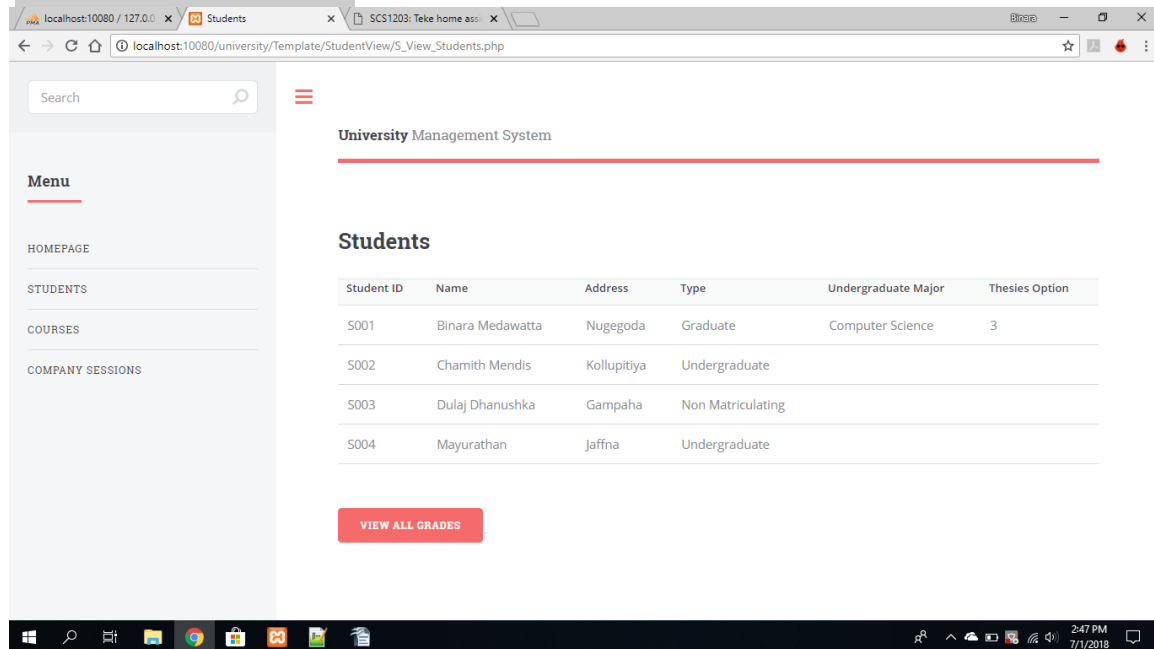


2.2 Students

To view students following query is used,

SQL query

```
SELECT * FROM student
```



Buttons

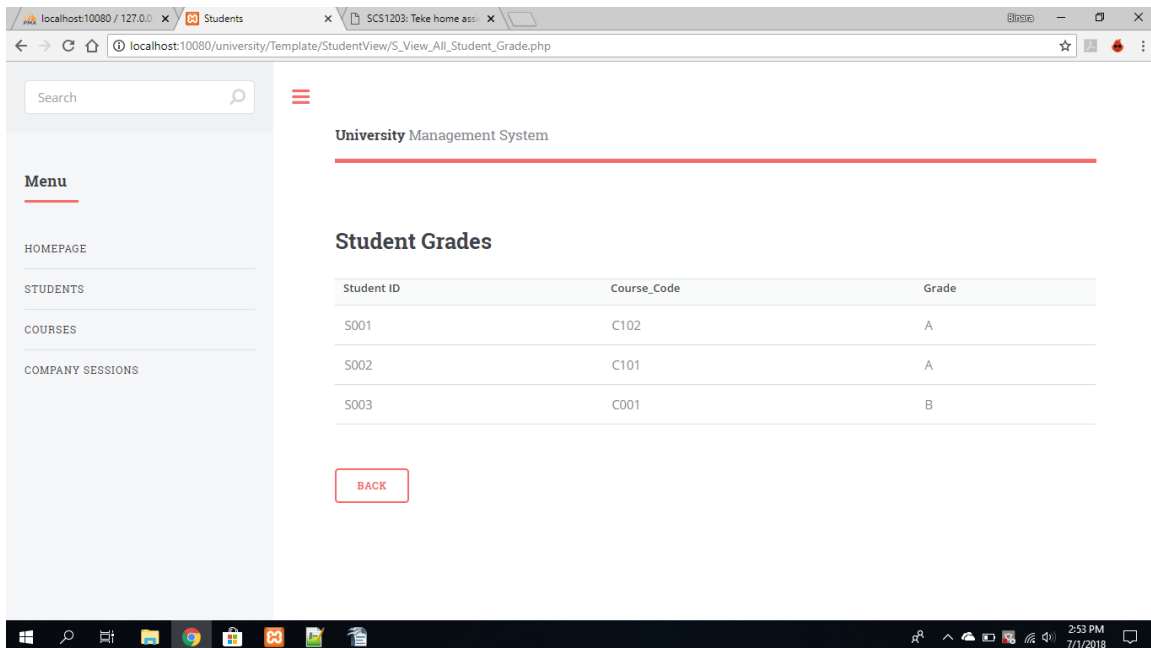
- a. View All Grades

2.2.1 View All Grades

First it selects all the data from the Grades table using the following query,

SQL Query

```
SELECT * FROM stud_course_grade
```



Buttons

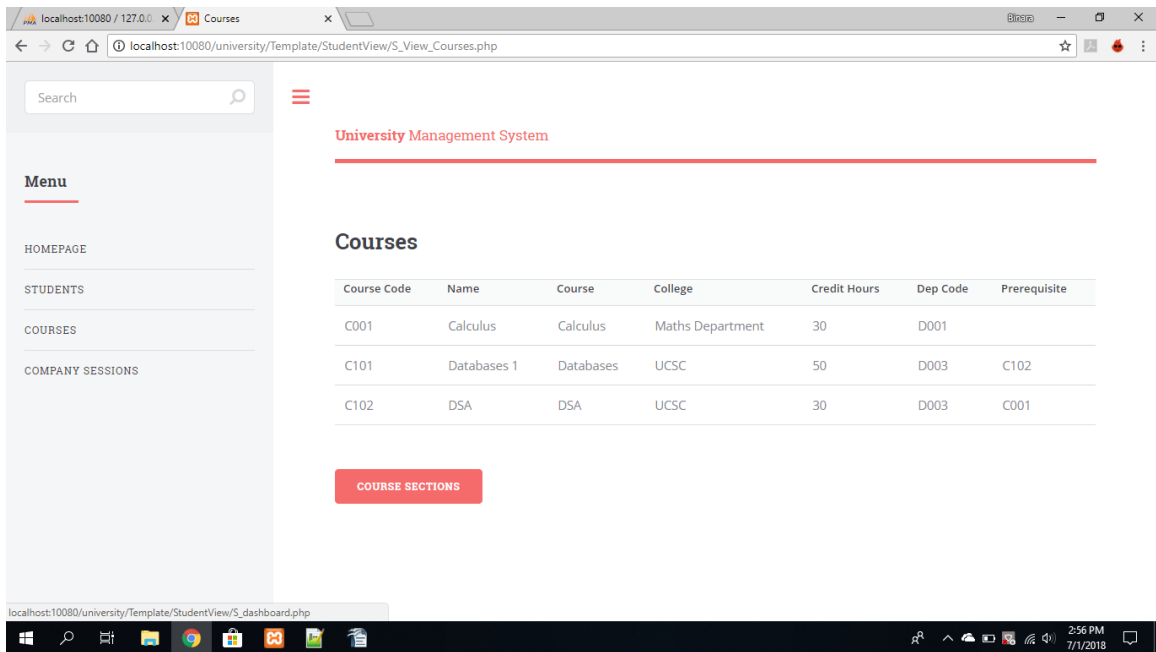
- a. Back : Go back to Student View.

2.3 Courses

To view courses following query is used,

SQL query

```
SELECT * FROM course
```



Buttons

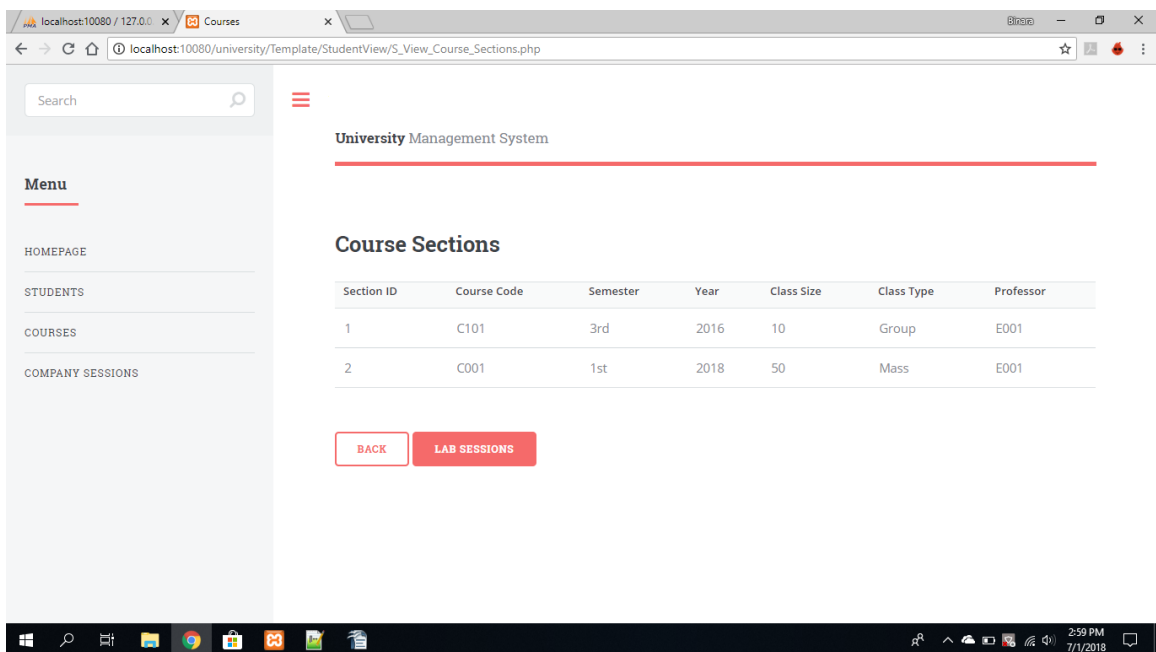
- Course Sections : Directs you to the Course Sections page.

2.3.1 Course Sections

To view courses sections following query is used,

SQL query

```
SELECT * FROM course_section
```



Buttons

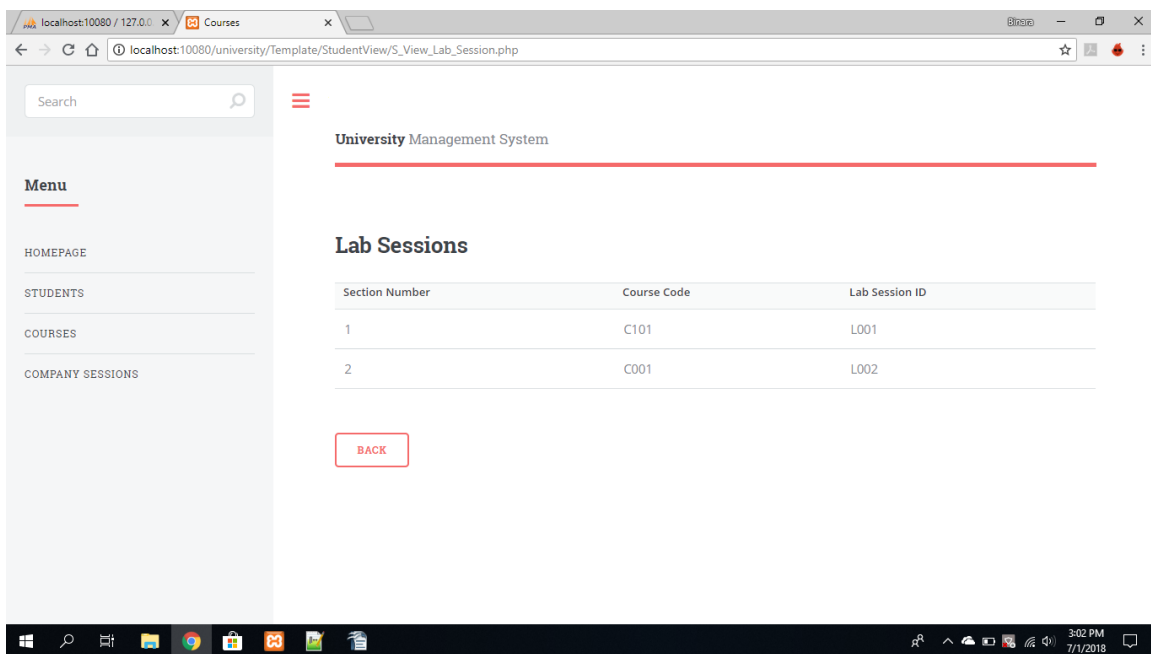
- a. Back : Go back to Course View.
- b. Lab Sessions : Directs you to the Lab Sessions page.

2.3.1.1 Lab Sessions

To view lab sessions following query is used,

SQL query

```
SELECT * FROM lab_session
```



Buttons

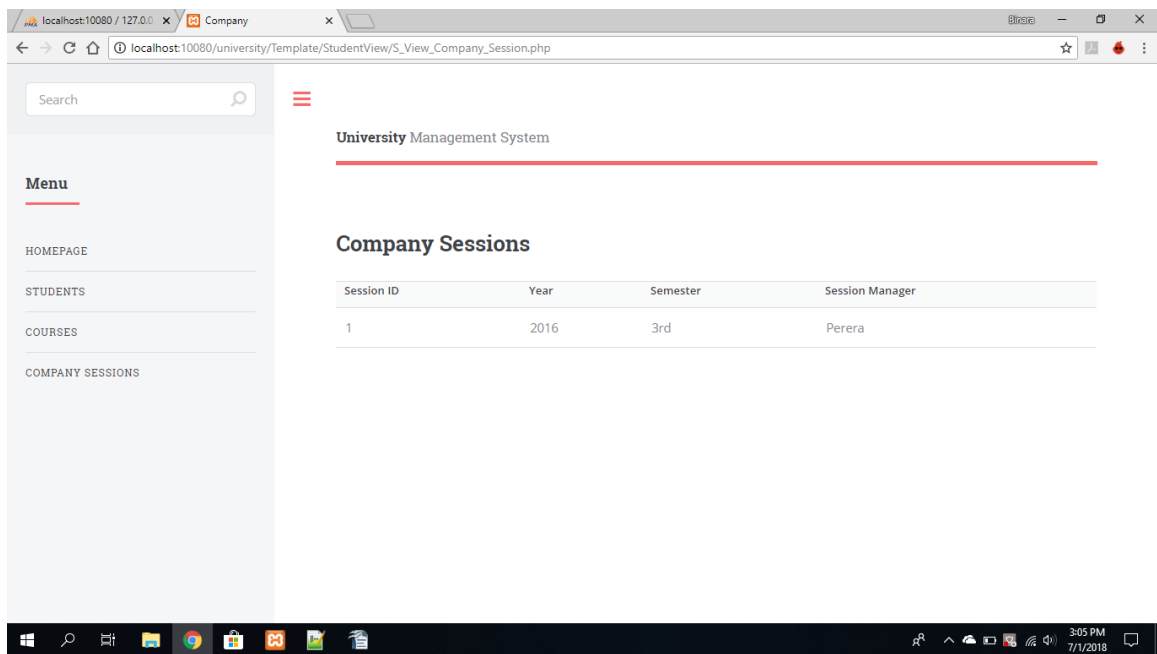
- Back : Go back to Course Sections View.

2.4 Company Sessions

To view Company Sessions following query is used,

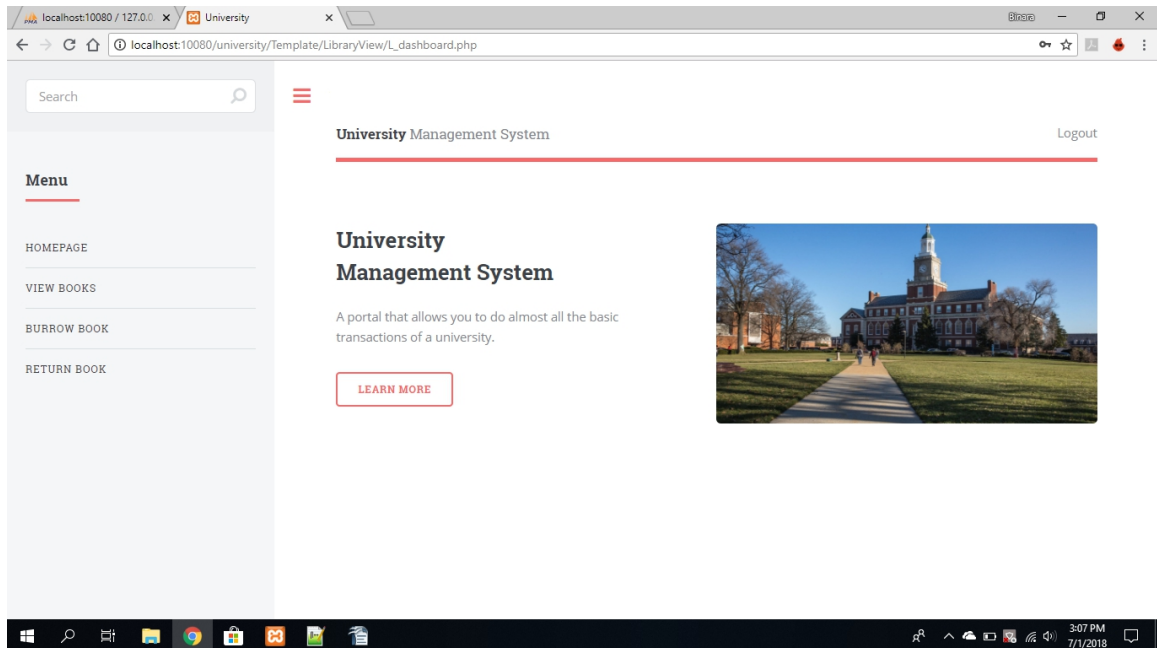
SQL query

```
SELECT * FROM company_session
```



3. Librarian User Level

3.1 Dashboard

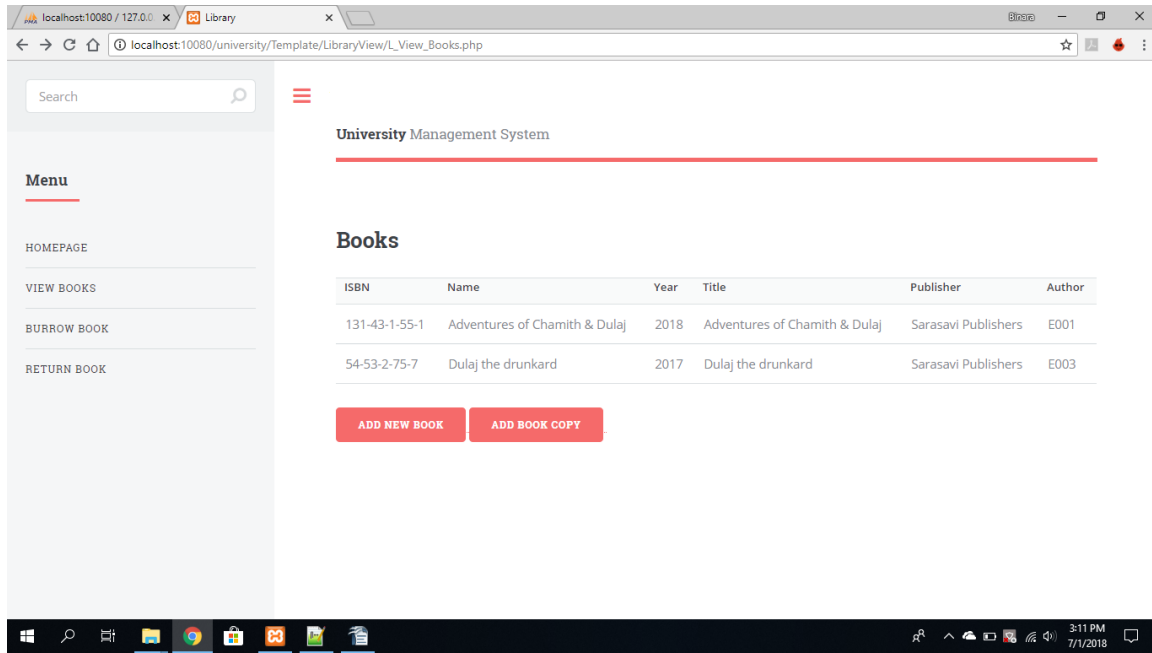


3.2 View Books

To view Books following query is used,

SQL query

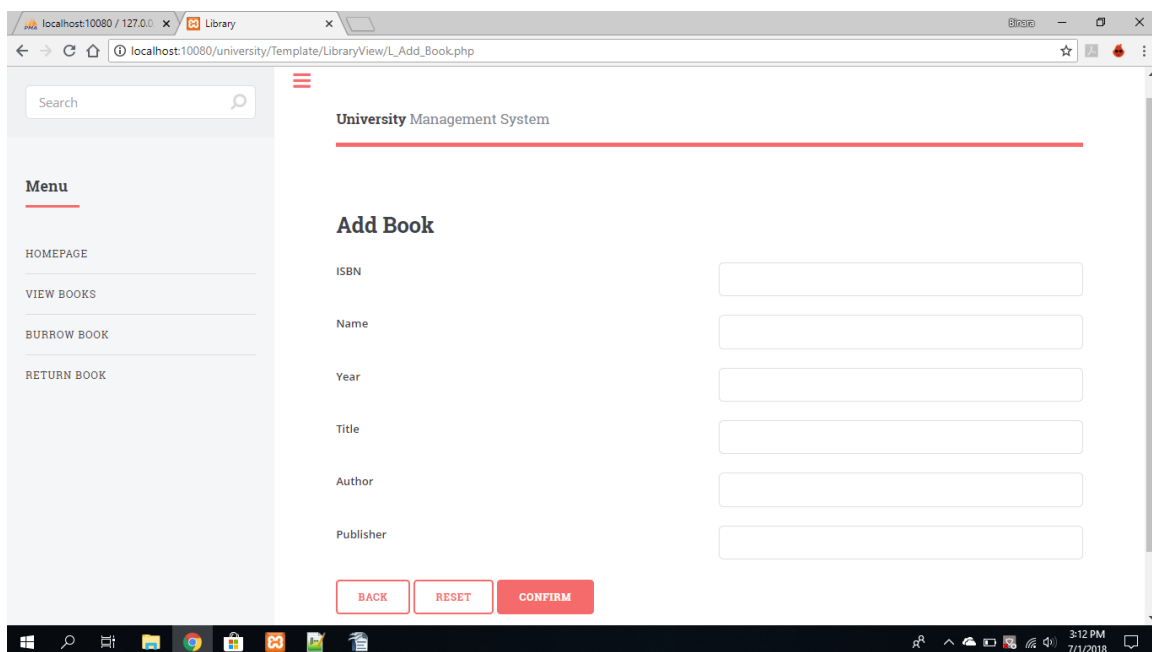
```
SELECT * FROM book
```



Buttons

- a. Add New Book : Directs you to the Add New Book page.
- b. Add Book Copy : Directs you to the Add Book Copy page.

3.2.1 Add New Book



Buttons

- a. Back : Go back to Books View.
- b. Reset : Resets the form. ie. Remove all inserted data.
- c. Confirm : Inserts the form data to the book table using following query

SQL Query

```
INSERT INTO book(ISBN, Name, Year, Title, Publisher, Author) VALUES('$isbn', '$name', '$year', '$title', '$publisher', '$author')
```

3.2.2 Add Book Copy

The screenshot shows a web browser window with the URL `localhost:10080/university/Template/LibraryView/L_Add_Book_Copy.php`. The page is titled "University Management System" and "Add Book Copy". It features a sidebar menu with options: HOME PAGE, VIEW BOOKS, BURROW BOOK, and RETURN BOOK. The main content area contains three input fields: "Copy_No", "ISBN", and "Is_Available". Below these fields are three buttons: "BACK", "RESET", and "CONFIRM".

Buttons

- a. Back : Go back to Books View.
- b. Reset : Resets the form. ie. Remove all inserted data.
- c. Confirm : Inserts the form data to the book_copy table using following query

SQL Query

```
INSERT INTO book_copy(Copy_NO, ISBN, Is_Available) VALUES('$copy_no', '$isbn', '$is_available')
```

3.3 Burrow Book

The screenshot shows a web browser window with the URL `localhost:10080/university/Template/LibraryView/L_Borrow_Book.php`. The page title is "University Management System". On the left is a sidebar menu with options: HOME PAGE, VIEW BOOKS, BURROW BOOK (selected), and RETURN BOOK. The main content area is titled "Burrow Book" and contains a form with the following fields: "Copy No", "ISBN", "Student_ID", and "Borrow Date" (pre-filled with "2018-07-01"). At the bottom of the form are two red buttons: "RESET" and "BURROW".

Buttons

- a. Reset : Resets the form. ie. Remove all inserted data.
- b. Burrow : Inserts the form data to the book_burrow table using the following query

SQL Query

```
INSERT INTO book_burrow(Copy_No, ISBN, Student_ID, Borrow_Date)
VALUES('$copy_no', '$isbn', '$student_id', '$borrow_date')
```

3.4 Return Book

The screenshot shows a web browser window with the URL `localhost:10080/university/Template/LibraryView/L_Return_Book.php`. The page title is "University Management System". On the left is a sidebar menu with options: HOME PAGE, VIEW BOOKS, BURROW BOOK, and RETURN BOOK (selected). The main content area is titled "Return Book" and contains a form with the following fields: "Enter Copy Number", "Enter ISBN", "Copy No" (pre-filled with "2"), "ISBN" (pre-filled with "54-53-2-75-7"), "Student ID" (pre-filled with "E001"), "Borrow Date" (pre-filled with "0000-00-00"), and "Return Date". At the bottom of the form are two red buttons: "RESET" and "RETURN".

Buttons

- a. Search : Searches for the borrowed book in the book_burrow table using query 1
- b. Reset : Resets the form with initial data
- c. Burrow : Inserts the form data to the book_burrow table using query 2

SQL Query 1

```
SELECT * FROM book_burrow WHERE Copy_No = '$copy_no' AND ISBN = '$isbn'
```

SQL Query 2

```
UPDATE book_burrow SET Return_Date=".$return_date." WHERE Copy_No=".$n_copy_no." AND ISBN=".$isbn." AND Student_ID=".$student_id."
```

Group Details

Group No : 04

| | |
|--|--|
| | |
| | |
| | |
| | |
| | |