CAMPUS LOCALE

Project Report Submitted By

ANTU JILSON

Reg. No: AJC19MCA003

In Partial fulfillment for the Award of the Degree Of

MASTER OF COMPUTER APPLICATIONS
(MCA)
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE,
Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2019-2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, “**CAMPUS LOCALE”** is the bonafide work of **ANTU JILSON (Reg.No:AJC19MCA003)** in partial fulfillment of the requirements for the award of the Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2019-22.

Ms. Meera Rose Mathew
Internal Guide

Rev.Fr. Dr. Rubin Thottupurathu Jose
Coordinator

Rev. Fr. Dr. Rubin Thottupurathu Jose
Head of the Department

External Examiner

DECLARATION

I hereby declare that the project report **“CAMPUS LOCALE”** is a bonafided work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2019-2022.

Date: **ANTU JILSON**
KANJIRAPPALLY **Reg. No: AJC19MCA003**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish toexpress my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev. Fr. Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Rev. Fr. Dr. Rubin Thottupurathu Jose** and for his valuable suggestions and for overwhelming concern and guidance from thebeginning to the end of the project. I would also like to express sincere gratitude to my guide**, Ms. Meera Rose Mathew** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughoutthe project. I express my thanks to all my friends and classmates for their interest, dedication,and encouragement shown towards the project. I convey my hearty thanks to my family forthe moral support, suggestions, and encouragement to make this venture a success.

ANTU JILSON

ABSTRACT

The Campus Locale is a web-based system which will use as a platform for interaction between student, teachers and parents. While the main objective of this project is to computerize the paperwork in the system and automate the work. The computerization is done so that the storage of all the details regarding students and teachers will be stored in the system which makes system centralized and the chance of duplication of any data is minimized. While by doing automation to the system will reduce the time for storing any data in the system. It will manage all the work in any campus in particular order so that the time requirement and complexity of the system will be reduced, at first it will focus on student related information. As a student gets the admission in the school system will start managing the details regarding the students. It will manage the fee details, and if the full payment has not done, then it will notify about the fee to a staff of the school. Campus Locale will then display the date of the test and when the test completes it will display the results of the students.

CONTENT

List of Abbreviations

IDE	-	Integrated Development Environment
HTML	-	Hyper Text Markup Language.
CSS	-	Cascading Style Sheet
SQL	-	Structured Query Language
UML	-	Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Campus Locale deals with all kind of student details, academic related reports, college details, course details, curriculum, batch details and other resource related details too. It tracks all the details of a student from the day one to the end of his course which can be used for all reporting purpose, tracking of attendance, progress in the course, completed semesters years, coming semester year curriculum details, exam details, project or any other assignment details, final exam result; and all these will be available for future references too.Our program will have the databases of Courses offered by the college under all levels of graduation or main streams, teacher or faculty's details, batch execution details, students' details in all aspects.This program can facilitate us explore all the activities happening in the college, even we can get to know which teacher / faculty is assigned to which batch, the current status of a batch, attendance percentage of a batch and upcoming requirements of a batch. Different reports and Queries can be generated based of vast options related to students, batch, course, teacher / faculty, exams, semesters, certification and even for the entire college.

The College management system is an automated version of manual Student Management System. It can handle all details about a student. The details include college details, subject details, student personnel details, academic details, exam details etc... In case of manual system they need a lot of time, manpower etc.Here almost all work is computerized. So the accuracy is maintained. Maintaining backup is very easy. It can do with in a few minutes. Our system has two type of accessing modes, administrator and user. Student management system is managed by an administrator. It is the job of the administrator to insert update and monitor the whole process. When a user log in to the system. He would only view details of the student. He can't perform any changes .

1.2 PROJECT SPECIFICATION

The proposed system is made to help the customers for an easy and convenient way ofreserving a package. We will also provide users to give feedbacks, payment details, complaint details etc.

The system includes 3 modules. They are:

1. Admin

Admin must have a login into this system. The administrator is responsible for entering the new student and managing the student Accounts. The administrator also manages the faulty accounts like entering a new faculty assigning the faculty to the subjects. The Administrator also updates the college related information about events that occur in the college. The administrator will check all the updates i.e. student updates faculty, updates, exam updates etc. The administrator has the highest level of power in the college management system.

2. Faculty

Faculty can login with username and password. The staff can update the information regarding the students attendance, internal marks of the students and any information regarding the subjects they handle. They can view publicaly available data. That is view all events that approved by venue coordinator. They can also view the student details for better understanding the student performance and improving the efficiency of the student. The staff also gets the updates from the college regarding any events occurring in the college.

3. Student

Student can login with admission number and password. The student is of center focus, because in every college student plays the very important role. Student can access the information of the college, subject details, training. The course details include information regarding branch he is studying, the academic curriculum of the college, year wise subject offered by the branch, the subject details include the syllabus of the subjects, information regarding the staff handling the subjects, the subjects he presently registered for the semester he is presently studying, attendance and internal marks of the subjects, he can also ask any queries to the staff

CHAPTER 2

SYSTEM STUDY

2.1 .INTRODUCTION

System study is a step-by-step process used to identify and develop or acquire the software need to control the processing of specific application. It is a continuing activity the stages of the systems development. It is the process of gathering and interpreting facts, diagnosing problems and using the facts to improve the system. The outputs from the organization are traced through the various processing that the input phases through in the organization. This involves gathering information and using structured tools for analysis.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. Thissystem is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

2.1 EXISTING SYSTEM

The current system for reserving a class room, seminar hall, conference room, or any room in the college was done by through manually writing the request and submitting it into the authorized faculty. Managing and keeping those request are tedious task. Modification and cancellation process are also make issues with the current system. Requested and documented pages need to store in files, shelves, or any other places. By doing this may lead of lost, missing, falsification, or degraded due to water, fire, rustic or any damage to the file.

2.2 PROPOSED SYSTEM

The Automate the existing system by which information on system can be accessed from anywhere just in a mouse click with minimum time consumption. The new system introduces requesting a venue through mouse click and help to identify which days are not booked by the other users. The request of program coordinators are send from Department head to the respected venue coordinator and manage it by them. There is a calendar in the system that will show which events are taking place at those venues. A map will show the event venues to the user. Confirmation of the booking and cancellations are send to each user through Notification

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

3.1.1 Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

The cost of project, CAMPUS LOCALE was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project was developed in a low cost. As it iscompletely developed using open source software.

3.1.2 Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in theterms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project requires High Resolution Scanning device and utilizes Cryptographic techniques. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The System used was also of good performance of Processor Intel i3 core; RAM 4GB and, Hard disk 1TB

3.1.3 Behavioral Feasibility

The proposed system includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible. CAMPUS LOCALE, GUI is simple so that users can easily use it. CAMPUS LOCALE is simple enough so that no training is needed.

3.1 SYSTEM SPECIFICATION

3.1.1 Hardware Specification

Processor: core i5

RAM: 4 GB

Mouse: Standard Mouse

Keyboard: Logitech Keyboard

Processor Speed: 2.20 GH

3.1.2 Software Specification

Operating System: Windows 8.1 or later
Front End Design: HTML5, CSS3, Bootstrap
Front End Development: JavaScript, JQuery, Ajax
Back End Development: PHP
Database: MySQL
Web Server: XAMPP
IDE: Visual Studio Code

3.1 SOFTWARE DESCRIPTION

3.1.1 PHP

PHP is a server side scripting language designed for web development but also used as a general purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page ,it now stands for PHP: Hypertext Preprocessor, a recursive acronym. PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page. PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed onmost web servers and also as a standalone shell on almost every operating system and platform, free of charge.

3.1.2 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

- **MySQL is a database management system.**
  A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play acentral role in computing, as standalone utilities, or as parts of other applications.

- **MySQL databases are relational.**
  A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL92” refers to the standard released in 1992, “SQL: 1999” refers to the standard released in 1999, and “SQL: 2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

- **MySQL software is Open Source.**

  Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

  If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

- **MySQL Server works in client/server or embedded systems.**

  The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product orsystem. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit itsphysical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm thatis used. The system design develops the architectural detail required to build a system orproduct. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object oriented analysis and design. After some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation forth modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service- oriented task. For example, use cases in a product sales environment would include itemordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.

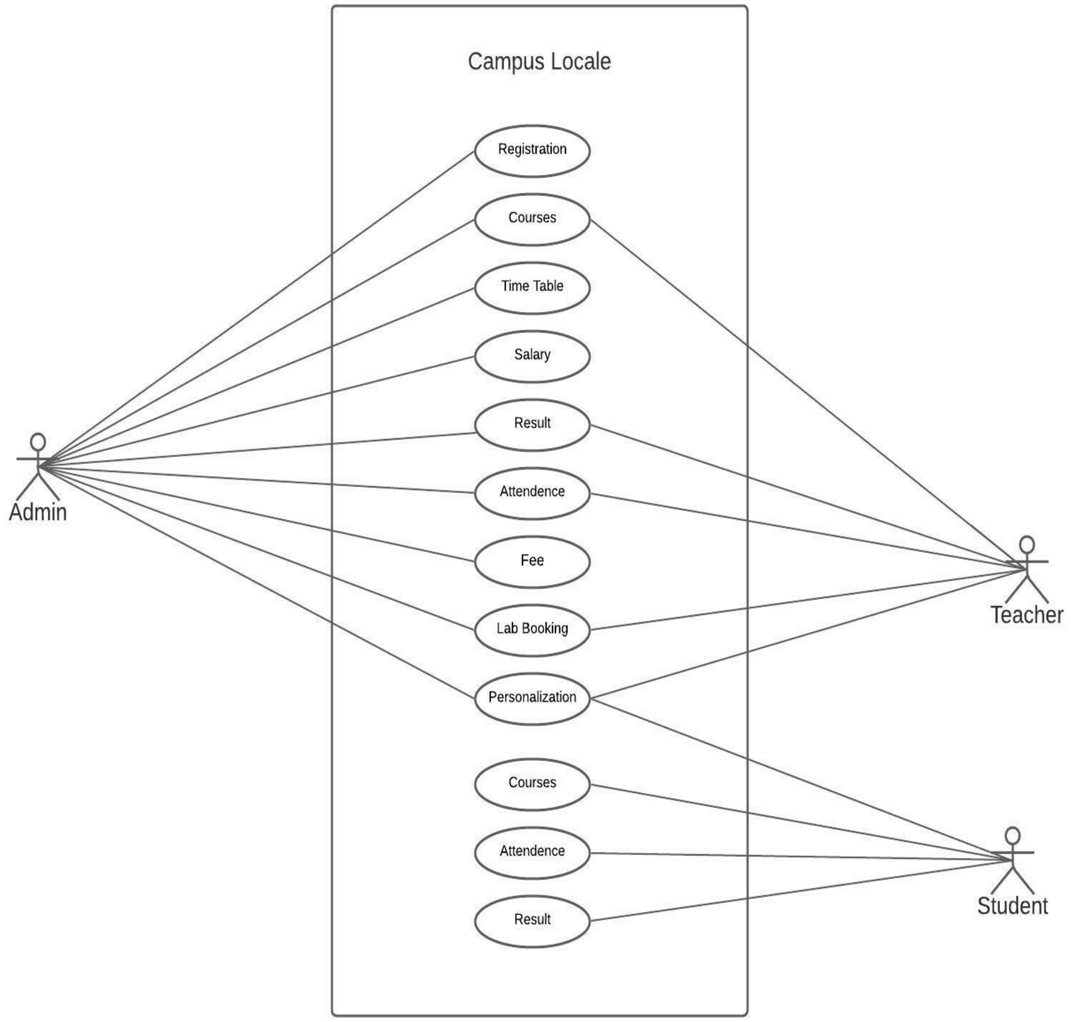
The actors, usually individuals involved with the system defined according to their roles.

- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

Use case diagram



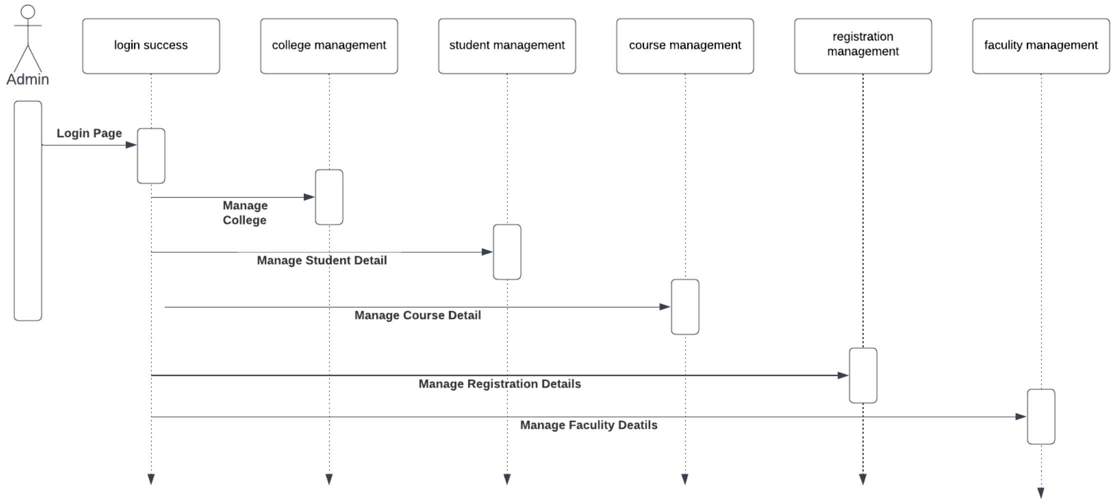
4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Uses of sequence diagrams –

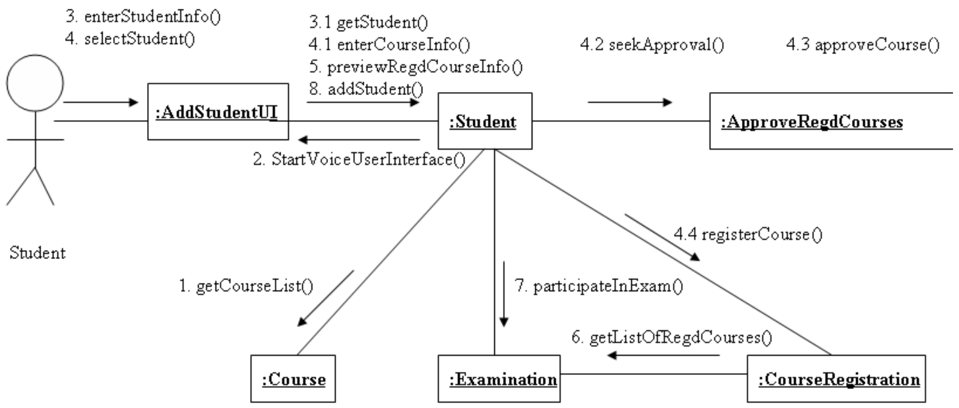
- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualise how messages and tasks move between objects or components in a system.

Sequence diagram



4.2.3 COLLABORATION DIAGRAM

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages,it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram , which is also known as a communication diagram, is used to portray the object's architecture in the system.



4.2.4 STATE CHART DIAGRAM

It describes different states of a component in a system . The states are specific to a Component / object of a system. A Statechart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events. They define different states of an object during its lifetime and these states are changed by events. Statechart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events. Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it Eresort Booking 24 Amal Jyothi College of Engineering Dept. of Computer Applications changes when some event is triggered. The most important purpose of Statechart diagram is to model lifetime of an object from creation to termination. Statechart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system. Following are the main purposes of using Statechart diagrams –

- To model the dynamic aspect of a system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object

Figure of Student

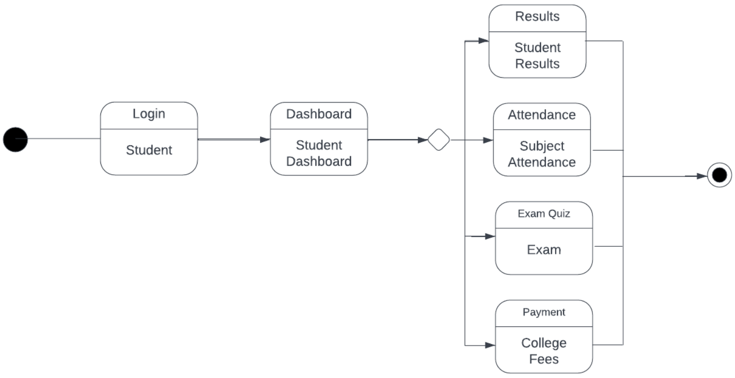
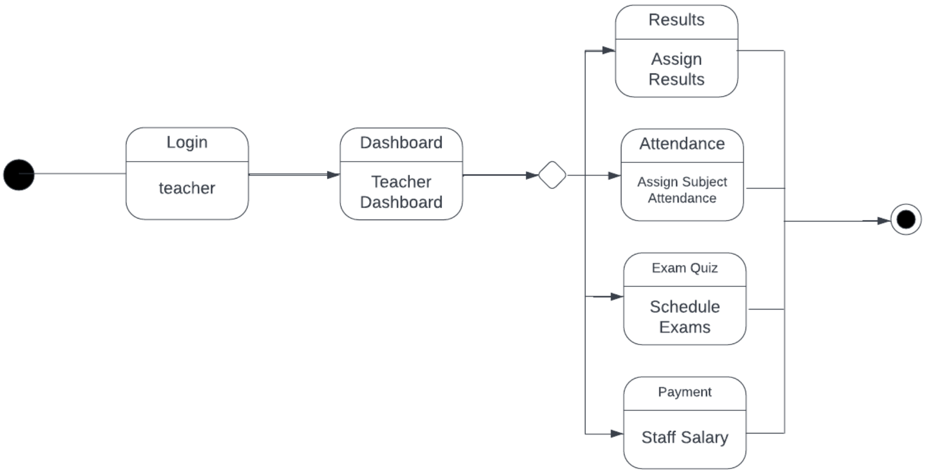
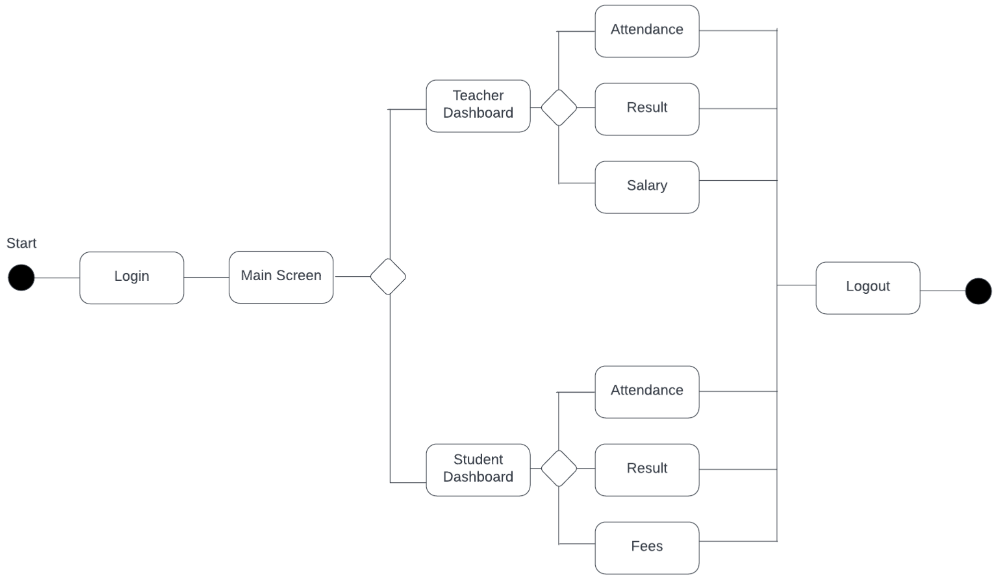


Figure on Teacher



4.2.5 ACTIVITY DIAGRAM

Activity Diagram describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join,etc.Activity diagrams are not only used for visualizing the dynamic nature of a system,but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.It does not show any message flow from one activity to another.Activity diagram is sometimes considered as the flowchart.

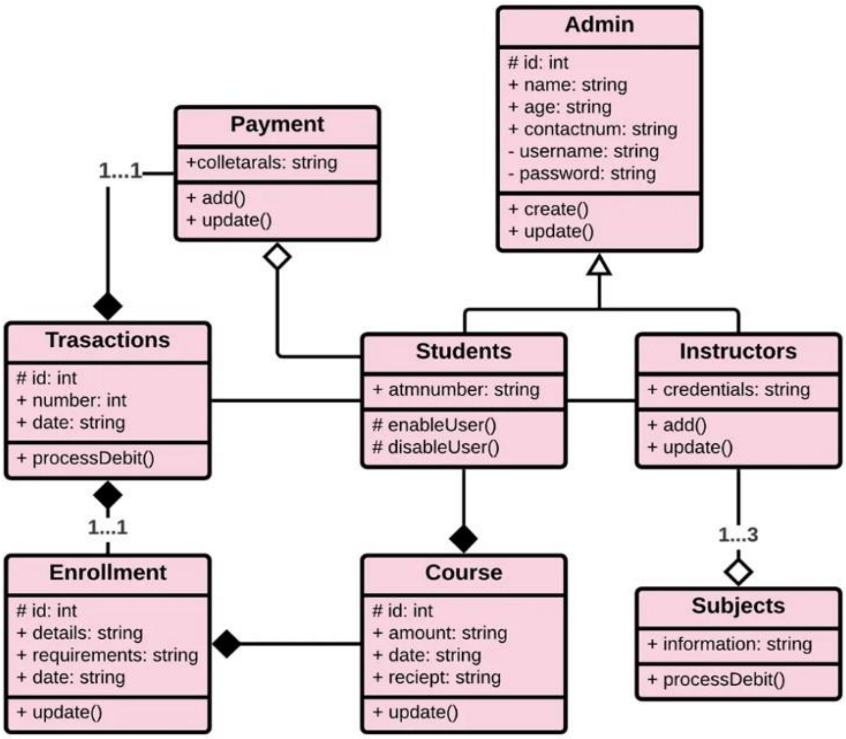


4.2.6 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of software application.Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

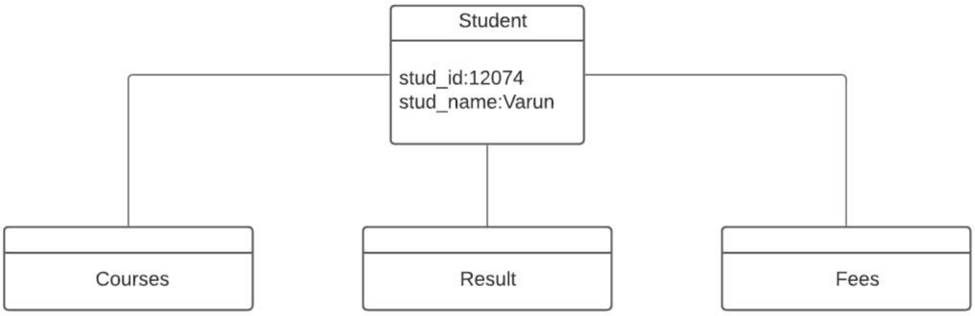
Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram. The purpose of the class diagram can be summarized as –

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering



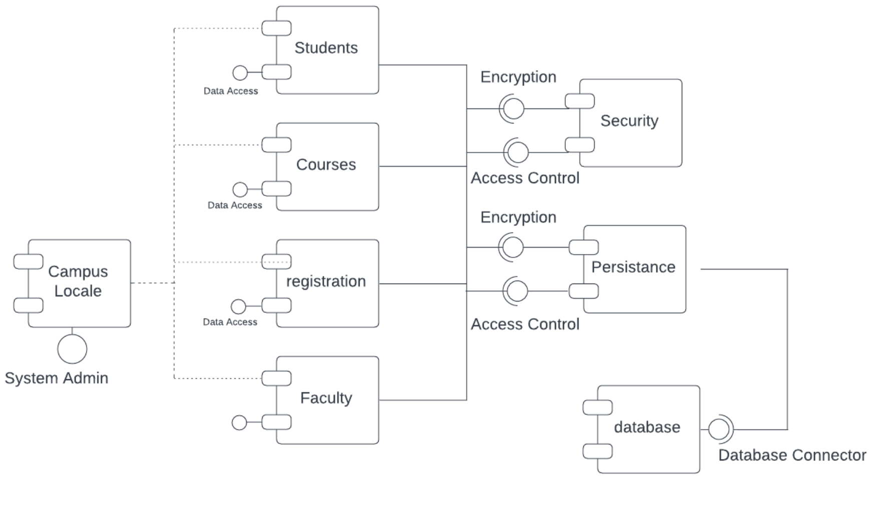
4.2.7 OBJECT DIAGRAM

Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams. Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment. Object diagrams are used to render a set of objects and their relationships as an instance.



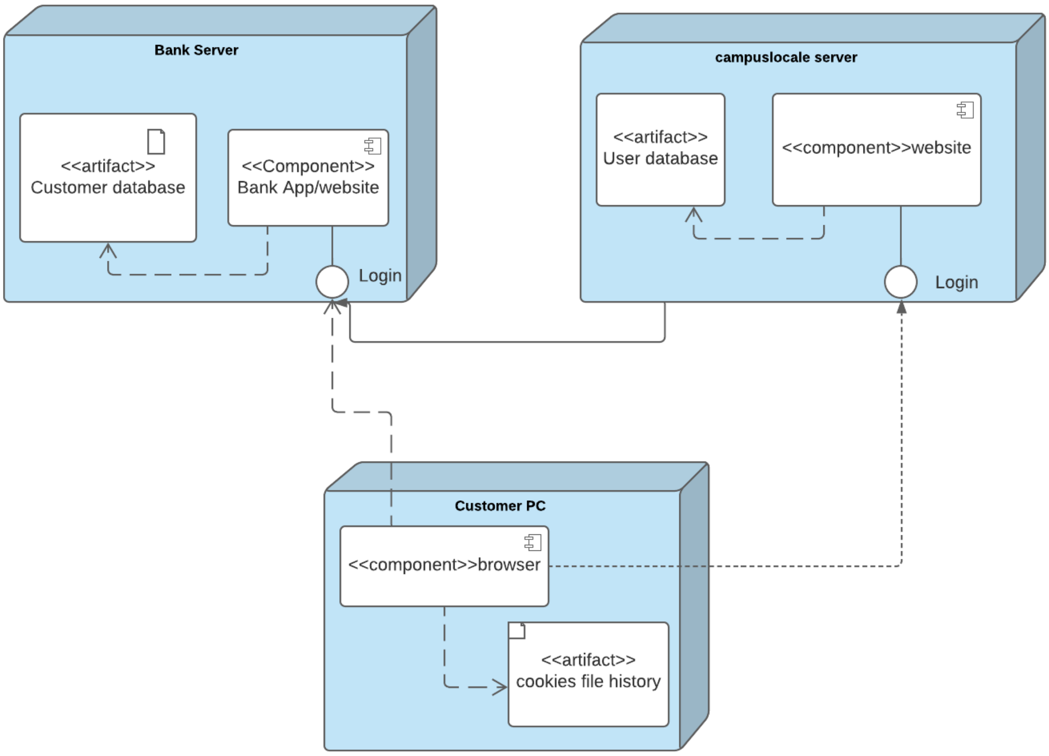
4.2.8 COMPONENT DIAGRAM

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities. Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc. Component diagrams can also be Described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment. A single component diagram cannot represent the entire system but collection of diagrams is used to represent the whole.



4.2.9 DEPLOYMENT DIAGRAM

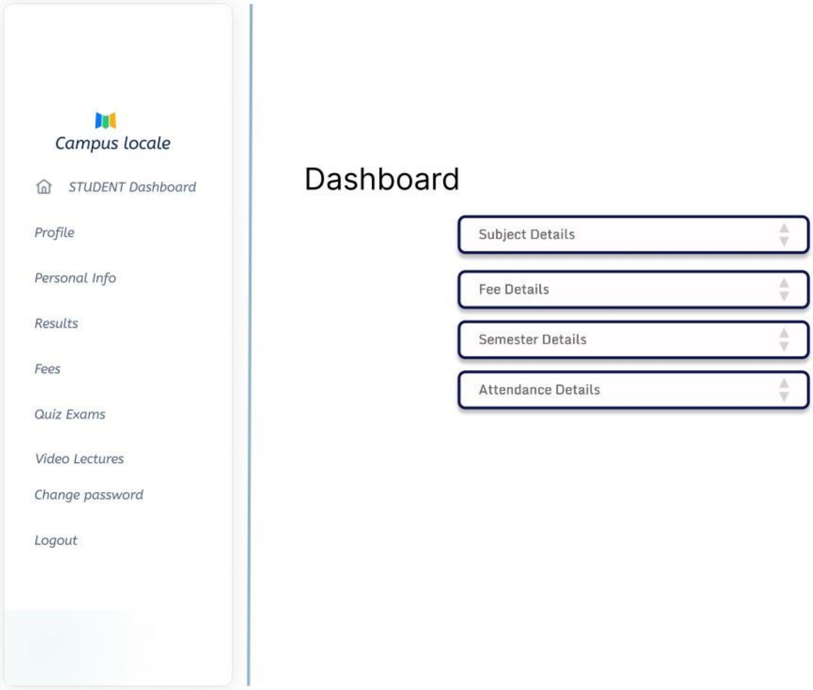
Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships. It ascertains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node.Since it involves many nodes,the relationship is shown by utilizing communication paths.


Bank Server
<<artifact>> Customer database
<<Component>> Bank App/website
Login
campuslocale server
<<artifact>> User database
<<component>>website
Login
Customer PC
<<component>>browser
<<artifact>> cookies file history

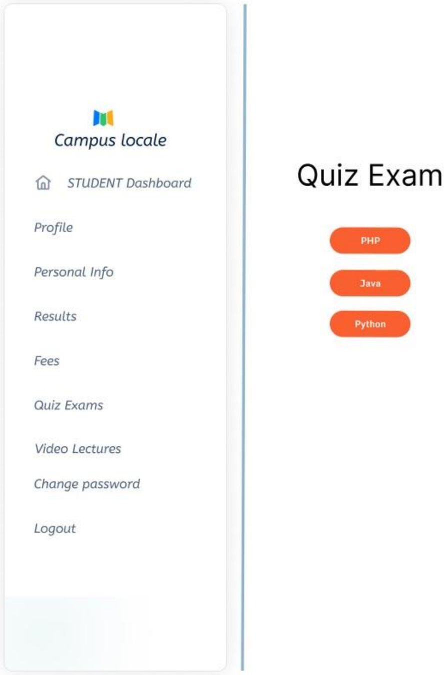
4.3 USER INTERFACE DESIGN USING FIGMA

4.3.1-LOGIN FORM

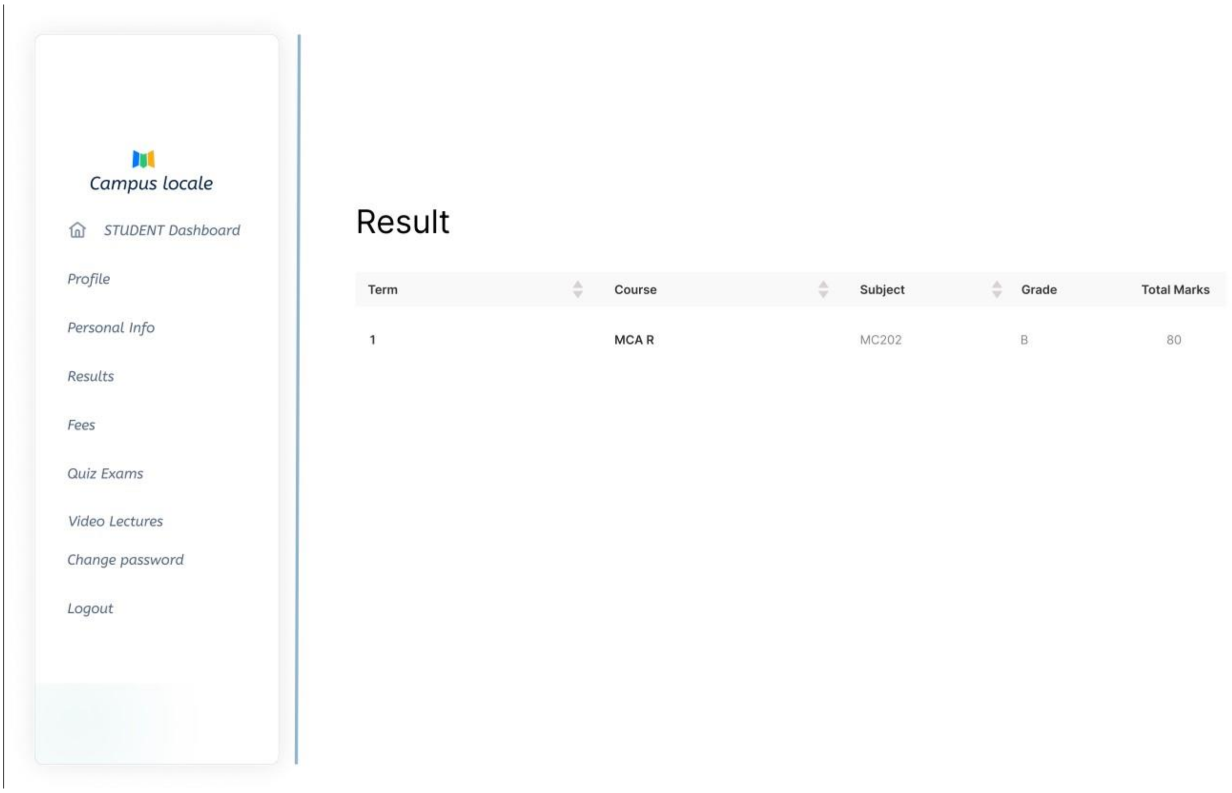
Form Name : Student Dashboard



Form Name : Student Exam



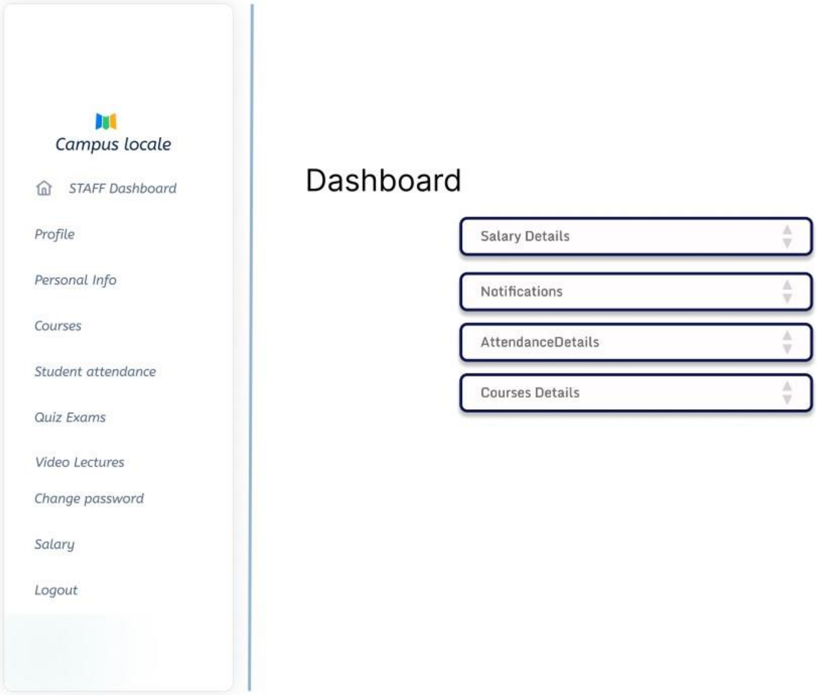
Form Name : Student Result



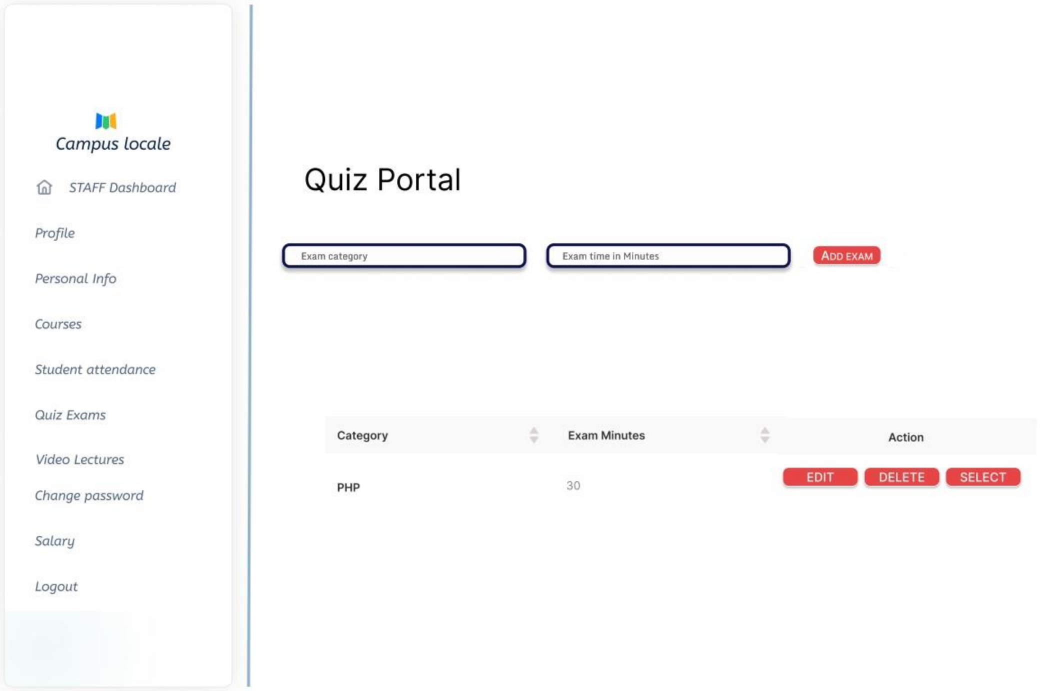
Result

Term	Course	Subject	Grade	Total Marks
1	MCA R	MC202	B	80

Form Name : Teacher Dashboard

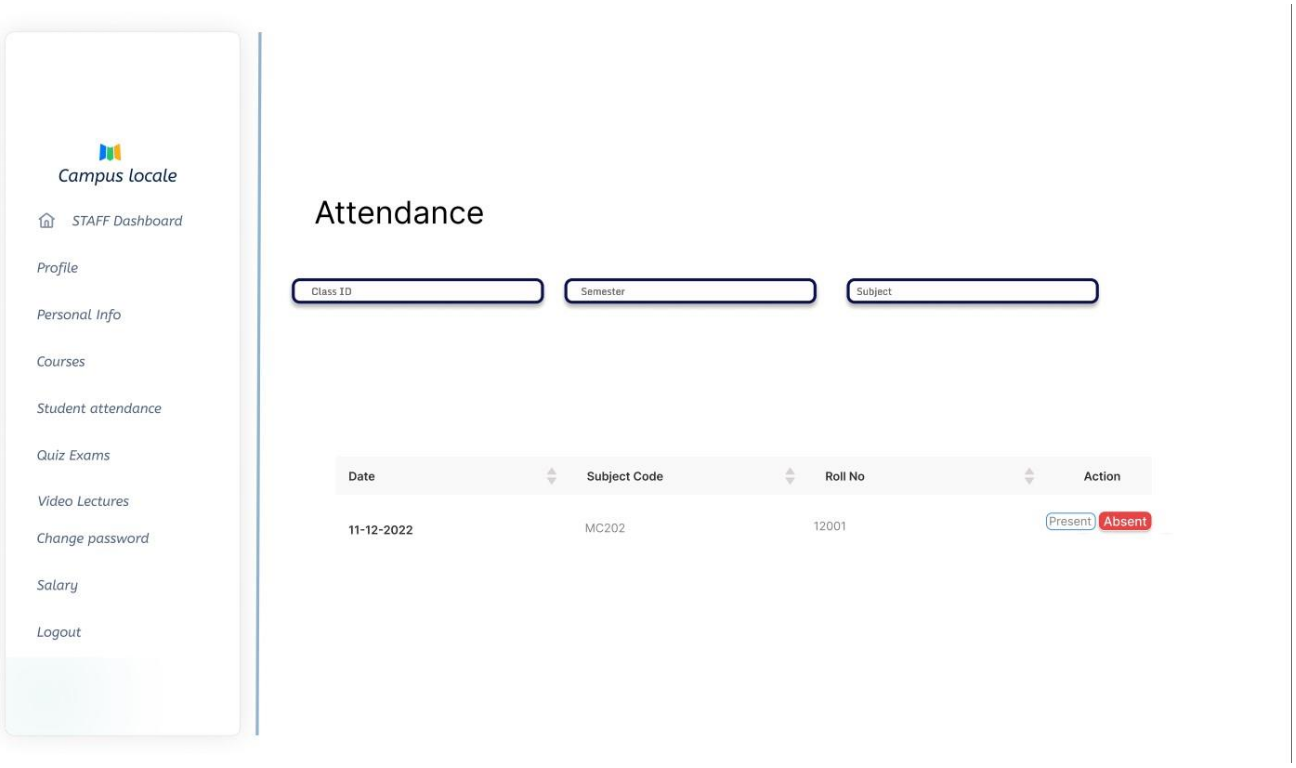


Form Name : Teacher Quiz Portal



Category	Exam Minutes	Action
PHP	30	EDIT DELETE SELECT

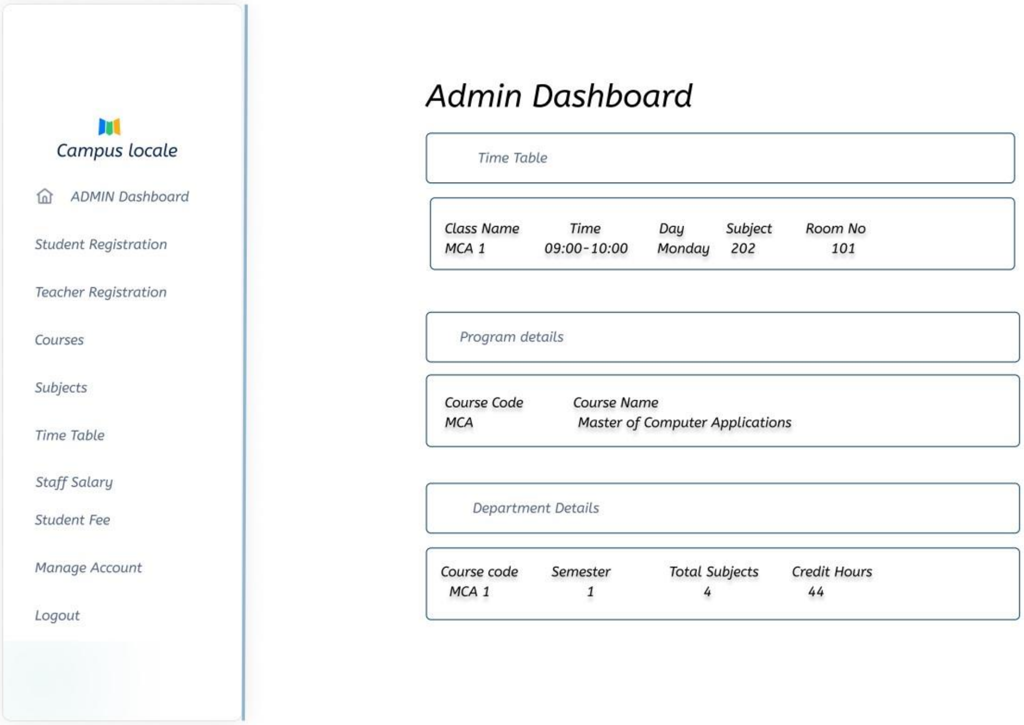
Form Name: Teacher Attendence



Attendance

Date	Subject Code	Roll No	Action
11-12-2022	MC202	12001	Present Absent

Form Name : Admin Dashboard



Admin Dashboard

Time Table

Class Name	Time	Day	Subject	Room No
MCA 1	09:00-10:00	Monday	202	101

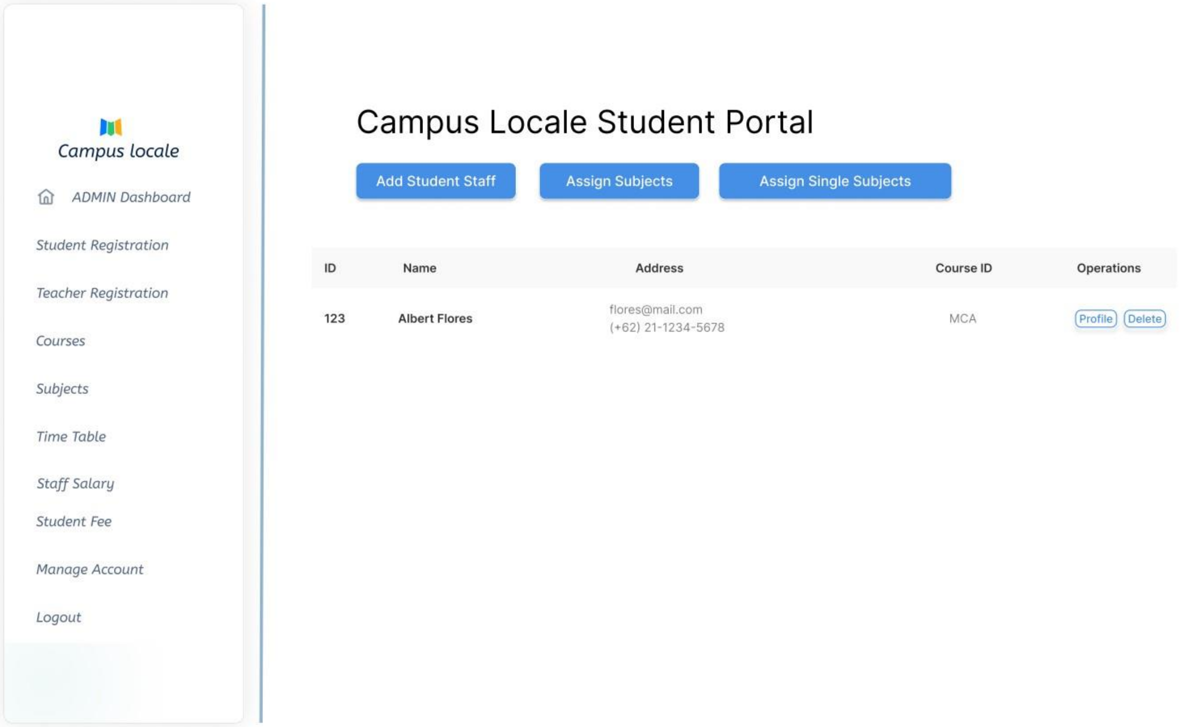
Program details

Course Code	Course Name
MCA	Master of Computer Applications

Department Details

Course code	Semester	Total Subjects	Credit Hours
MCA 1	1	4	44

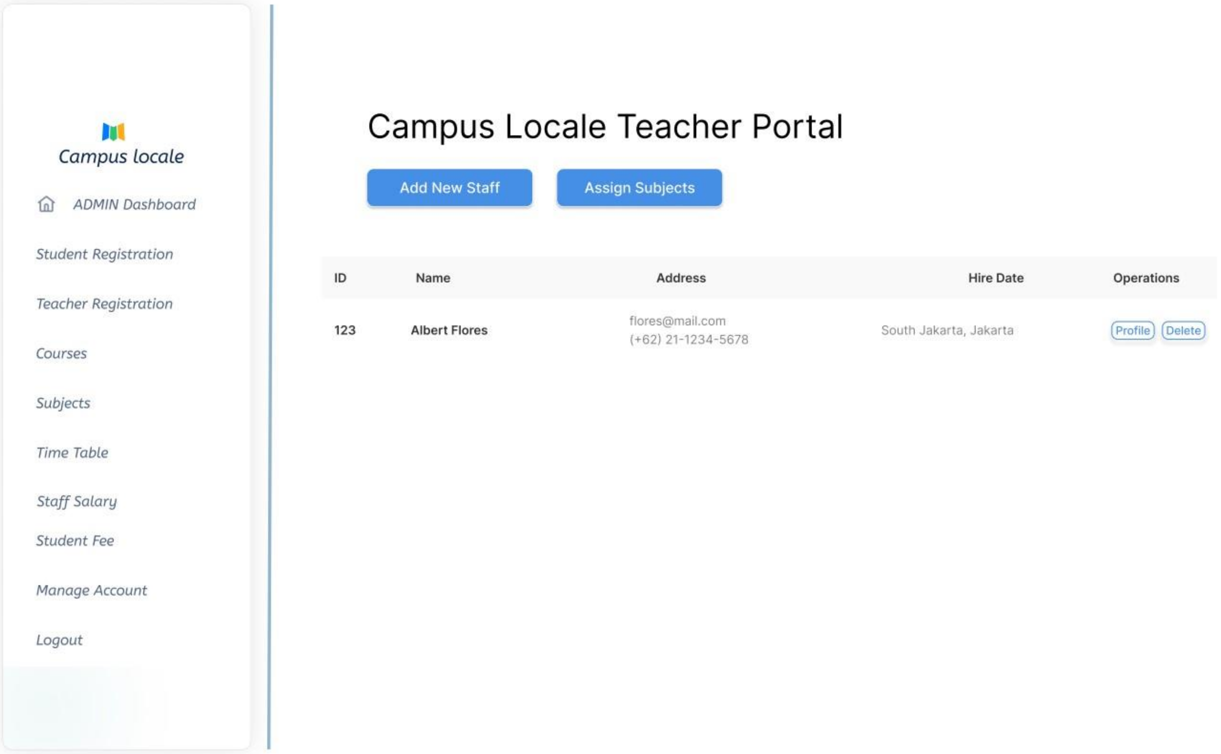
Form Name: Add Student



Campus Locale Student Portal

ID	Name	Address	Course ID	Operations
123	Albert Flores	flores@mail.com (+62) 21-1234-5678	MCA	Profile Delete

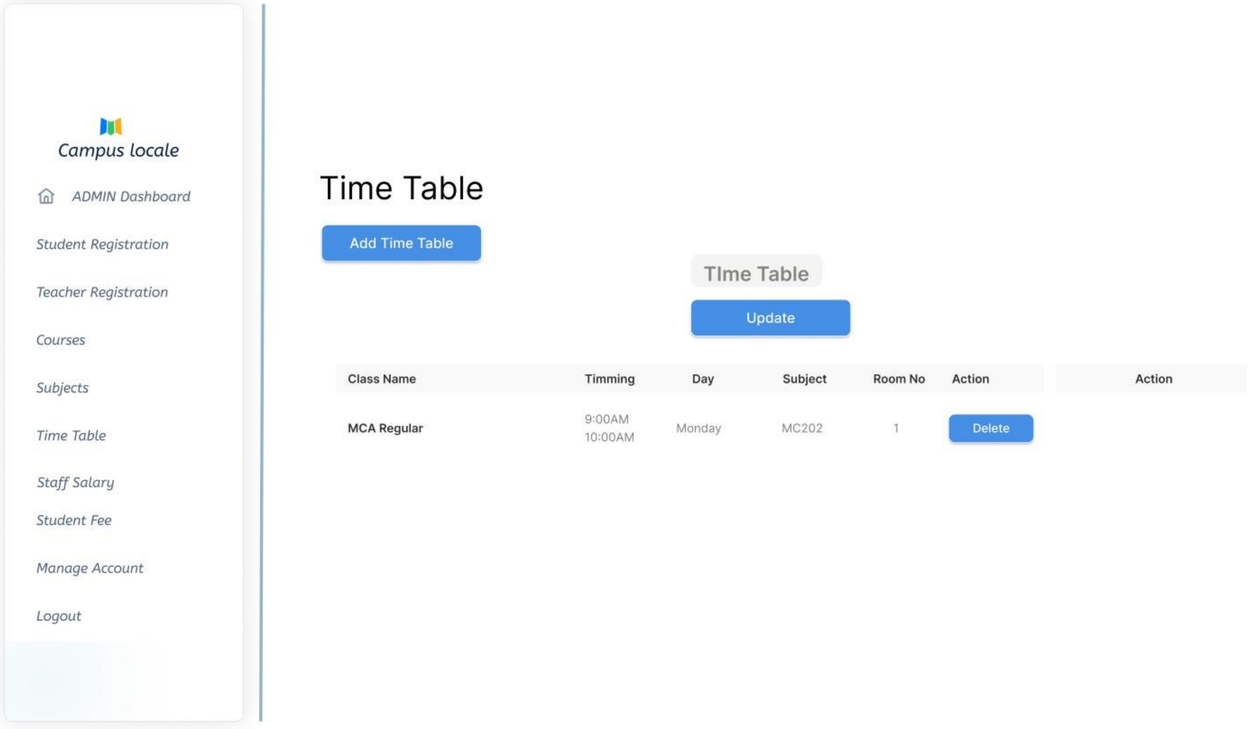
Form Name : Add Teacher



Campus Locale Teacher Portal

ID	Name	Address	Hire Date	Operations
123	Albert Flores	flores@mail.com (+62) 21-1234-5678	South Jakarta, Jakarta	Profile Delete

Form Name : Time table



Class Name	Timming	Day	Subject	Room No	Action	Action
MCA Regular	9:00AM 10:00AM	Monday	MC202	1	Delete	

4.1 TABLE DESIGN

4.4.1 Table Name:Login

Primary Key: id

Foreign Key:

Table info : Login, The table is to store Login information.

Column	Datatype	Key	Description
id	Int(20)	Primary Key	Login id for the registered user
user_id	Varchar(50)	Not null	Email id of registered user
password	Varchar(50)	Not null	Password of registered user
role	Varchar(50)	Not null	Role of registered user

4.4.2 Table Name:Course_subjects

Primary Key: sub_code

Foreign Key:course_code

Table info: Course subjects, The table is to store Subjects information of course .

Column	Datatype	Key	Description
sub_code	varchar(20)	Primary key	Subject code of the course
sub_name	varchar(20)	Not Null	Subject name
course_code	varchar(20)	Foreign Key	Course code
credit_hours	int(10)	Not Null	Total number of hours
semester	int(10)	Not Null	Current semester

4.4.3 Table Name: Class_results
Primary Key: class_id
Foreign Key:course_code
Table info : Class results, The table is to store exam results of the class.

Column	**Datatype**	**Key**	**Descriptions**
class_id	Int(20)	Primary key	Id for the different class
course_code	varchar(50)	Foreign Key	Course code
sub_code	varchar(10)	Not Null	Subject Code
total_mark	Int(50)	Not Null	Total mark
obtain_mark	int(20)	Not Null	Obtain mark
result_date	date	Not Null	Result date

4.4.4 Table Name: Student_courses
Primary Key:stud_course_id
Foreign Key:course_code
Table info : Student Courses, The table is to store Courses information.

Column	**Datatype**	**Key**	**Descriptions**
stud_course_id	int(20)	Primary key	Student course id
course_code	varchar(20)	Foreign Key	Course Code
semester	Int(20)	Not Null	Current semester
roll_no	varchar(10)	Not Null	Roll number
sub_code	varchar(10)	Not Null	Subject Code
assign_code	varchar(10)	Not Null	

4.4.5 Table Name:Student_attendance

Primary Key: attend_id

Foreign Key: course_code

Table info : Attendance, The table is to store student attendance information

.

Column	**Datatype**	**Key**	**Descriptions**
attend_id	int(20)	Primary key	Attendence id of student
course_code	varchar(20)	Foreign Key	Course Code
subject_code	varchar(20)	Not Null	Subject Code
semester	Int(20)	Not Null	Current semester
sub_id	varchar(10)	Not Null	Subject Id
attendence	int(10)	Not Null	Attendence
attend_date	Varchar(20)	Not Null	Attendence date

4.4.5 Table Name:Student_Fee

Primary Key: fee_voucher

Foreign Key:roll_no

Table info :Fees, The table is to store fee details of each student.

Column	**Datatype**	**Key**	**Descriptions**
fee_voucher	int(20)	Primary key	Voucher
roll_no	varchar(10)	Foreign Key	Roll number
Amount	int(10)	Not Null	Fee amount
posting_date	date	Not Null	Current posting date
Status	varchar(10)	Not Null	Paid or due

4.4.6 Table Name:Teacher_course

Primary Key: teacher_course_id

Foreign Key:course_code

Table info :The table is to store and assign courses for each teachers.

Column	**Datatype**	**Key**	**Descriptions**
teacher_course_id	int(20)	Primary key	Course id
course_code	varchar(20)	Not Null	Course Code
semester	Int(20)	Not Null	Current semester
teacher_id	varchar(10)	Not Null	Teacher id
subject_code	varchar(20)	Not Null	Subject Code
assign_date	varchar(20)	Not Null	Date
total_class	int(10)	Not Null	Total class

4.4.7 Table Name:Teacher_info

Primary Key: teacher_id

Foreign Key:

Table info : The table is to store teacher personal profile informations.

Column	**Datatype**	**Key**	**Descriptions**
teacher_id	int(20)	Primary key	Course id
first_name	varchar(20)	Not Null	First name
last_name	varchar(20)	Not Null	Last name
email	varchar(10)	Not Null	Email id

phone_number	int(20)	Not Null	Phone number
profile_img	blob	Not Null	Profile Image
teacher_status	varchar(10)	Not Null	
application_status	varchar(10)	Not Null	
Dob	varchar(10)	Not Null	Date Of birth
Gender	varchar(10)	Not Null	Gender
Address	varchar(50)	Not Null	Address
place_of_birth	varchar(10)	Not Null	Place of birth
matric_certificate	varchar(10)	Not Null	
fa_certificate	varchar(10)	Not Null	
ba_certificate	varchar(10)	Not Null	
last_qualification	varchar(10)	Not Null	Last Qualification
State	varchar(10)	Not Null	State
hire_date	varchar(10)	Not Null	

4.4.8 Table Name:Teacher_salary_report

Primary Key: salary_id
Foreign Key:teacher_id
Table info : The table is to store salary information of teachers**.**

Column	Datatype	Key	Descriptions
salary_id	int(20)	Primary key	Salary id
teacher_id	int(20)	Not Null	Teacher id
total_amount	int(20)	Not Null	Total mark
Status	varchar(10)	Not Null	Attendence date
paid_date	date	Not Null	Paid date

4.4.9 Table Name:Timetable

Primary Key: id
Foreign Key:course_code
Table info : The table is to store time table of each class.

Column	Datatype	Key	Descriptions
id	Int(20)	Primary key	id
course_code	varchar(20)	Not Null	Course Code
semester	int(10)	Not Null	Current semester
timing_from	varchar(20)	Not Null	Course timing
timing_to	varchar(20)	Not Null	Course timing
day	varchar(20)	Not Null	

subject_code	varchar(20)	Not Null	Subject code
room_no	int(10)	Not Null	Room number

4.5 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is calledPhysical Level Design, concerned with the characteristics of the specific DBMS thatwill be used. A database design runs parallel with the system design. The organizationof the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

4.5.1 Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigneda unique name. A row in a tale represents a set of related values.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.

Every value in a relation is atomic, that is not decomposable.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

4.5.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- ✓ Normalize the data.
- ✓ Choose proper names for the tables and columns.
- ✓ Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal formif and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code where removed and ensured that all modules are working, and gives the expected result.

Code:

```
 package newtest;

 import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class Testclass {
public static void main(String[] args) throws InterruptedException {
//Setting system properties of ChromeDriver
System.setProperty("webdriver.chrome.driver",

"C:\\Users\\HP\\eclipse-
workspace\\ProjectTesting\\Lavorotest\\chromedriver.exe");
//Creating an object of ChromeDriver
WebDriver driver = new ChromeDriver();
driver.manage().window().maximize();
//launching the specified URL
driver.get("http://localhost/campuslocale/login/login.php");
driver.findElement(By.name("email")).sendKeys("antujilsonparayil@gmail.com");
driver.findElement(By.name("password")).sendKeys("Student123*");
driver.findElement(By.name("checkLoginButton")).click();
Thread.sleep(5000);
driver.quit();
```

5.2.1.1 Test Case

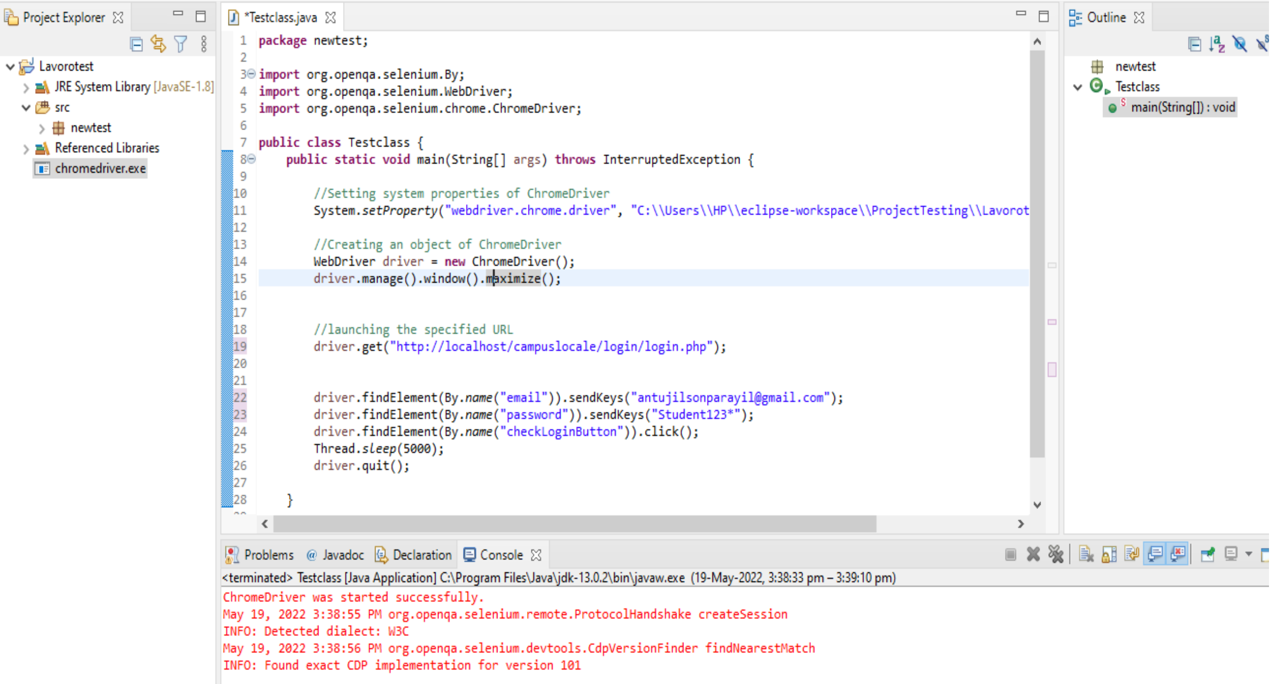
Test Case 1

Project Name:Campus Locale	
Login Test Case	
Test Case ID: Fun_1	**Test Designed By: Antu Jilson**
Test Priority(Low/Medium/High): High	**Test Designed Date:** 19-05-2022
Module Name: Login Screen	**Test Executed By : Ms.Meera Rose Mathew**
Test Title : Verify login with validemail and password	**Test Execution Date:** 19-05-2022
Description: Test the Login Page	
Pre-Condition : User has valid email id and password	

Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/Fail)**
1	Navigation toLogin Page		Login Page should be displayed	Login page displayed	Pass
2	Provide Valid Email Id	User Name: antujilsonparayil@gmail.com	User should beable to Login	User Logged inand navigated to Subadmin Dashboard with records	Pass
3	Provide Valid Password	Password: Student123*			
4	Click on Sign In button				
5	Provide Invalid Email Id or password	Email Id: user@gmail.Com Password: User1234	User shouldnot	Message for enter valid email id or	Pass

6	Provide Null Email Id or Password	Email Id: null Password: null	be able to Login	password displayed	
7	Click on Sign In button				
Post-Condition: User is validated with database and successfully login into account.The Account session details are logged in database					

SCREEN SHOTS



```java
package newtest;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class Testclass {
    public static void main(String[] args) throws InterruptedException {

        //Setting system properties of ChromeDriver
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\HP\\eclipse-workspace\\ProjectTesting\\Lavorot

        //Creating an object of ChromeDriver
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();


        //launching the specified URL
        driver.get("http://localhost/campuslocale/login/login.php");


        driver.findElement(By.name("email")).sendKeys("antujilsonparayil@gmail.com");
        driver.findElement(By.name("password")).sendKeys("Student123*");
        driver.findElement(By.name("checkLoginButton")).click();
        Thread.sleep(5000);
        driver.quit();

    }
```

```
<terminated> Testclass [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (19-May-2022, 3:38:33 pm – 3:39:10 pm)
ChromeDriver was started successfully.
May 19, 2022 3:38:55 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
May 19, 2022 3:38:56 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 101
```

Test Case 2

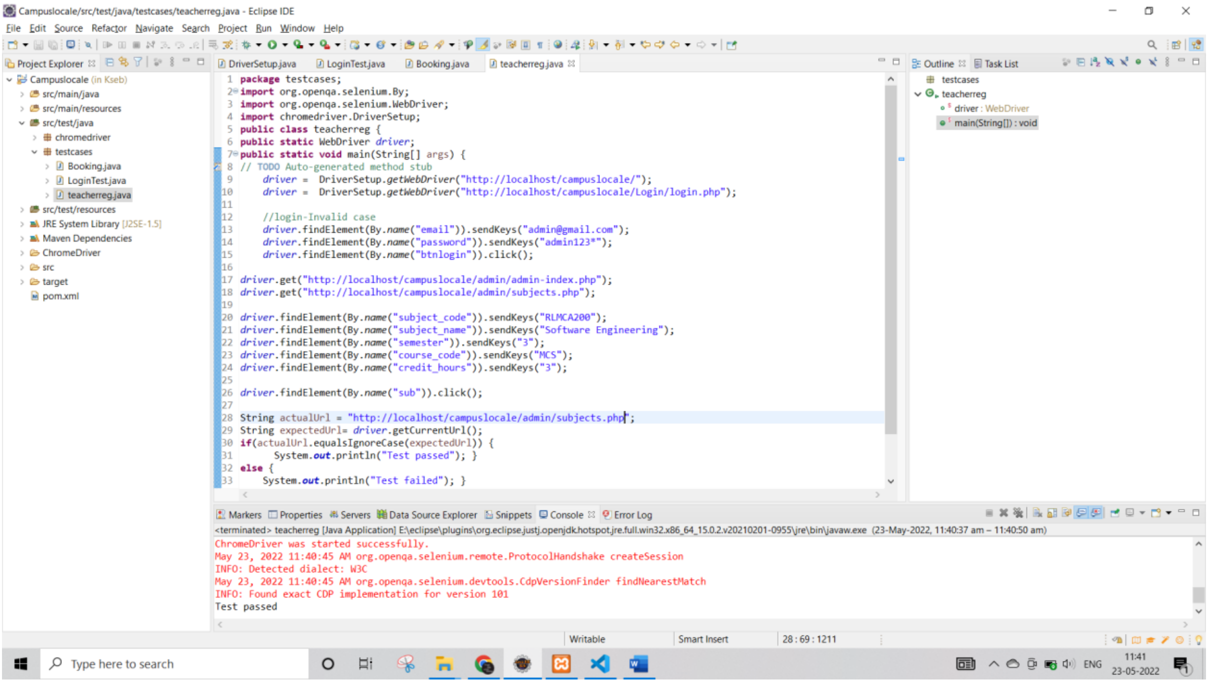
Project Name:Campus Locale	
Adding Subjects Test Case	
Test Case ID: Fun_1	**Test Designed By: Antu Jilson**
Test Priority(Low/Medium/High): High	**Test Designed Date:** 19-05-2022
Module Name: Add Subjects	**Test Executed By : Ms.Meera Rose Mathew**
Test Title : Verify login with validemail and password	**Test Execution Date:** 19-05-2022
Description: Test the Subject adding	
Pre-Condition : User has valid email id and password	

Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/Fail)**
1	Add new subject		Add new Subject	New subject added	Pass
2	Provide Valid Email Id	User Name: admin@gmail.com	User should beable to Login	User Logged inand navigated to Add Subjects with records	Pass
3	Provide Valid Password	Password: Admin123*			
4	Click on Sign In button				
5	Provide Invalid Email Id or password	Email Id: user@gmail.Com Password: User1234	User shouldnot be able to Login	Message for enter valid email id or password displayed	Pass
6	Provide Null Email Id or Password	Email Id: null Password: null			
7	Click on Sign In button				

Post-Condition: User is validated with database and successfully login into account.The Account session details are logged in database

Subjects.java

```
  package testcases;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import chromedriver.DriverSetup;
public class teacherreg {
public static WebDriver driver;
public static void main(String[] args) {
// TODO Auto-generated method stub
  driver =  DriverSetup.getWebDriver("http://localhost/campuslocale/");
  driver =  DriverSetup.getWebDriver("http://localhost/campuslocale/Login/login.php");

  //login-Invalid case
  driver.findElement(By.name("email")).sendKeys("admin@gmail.com");
  driver.findElement(By.name("password")).sendKeys("admin123*");
  driver.findElement(By.name("btnlogin")).click();

driver.get("http://localhost/campuslocale/admin/admin-index.php");
driver.get("http://localhost/campuslocale/admin/subjects.php");

driver.findElement(By.name("subject_code")).sendKeys("RLMCA200");
driver.findElement(By.name("subject_name")).sendKeys("Software Engineering");
driver.findElement(By.name("semester")).sendKeys("3");
driver.findElement(By.name("course_code")).sendKeys("MCS");
driver.findElement(By.name("credit_hours")).sendKeys("3");

driver.findElement(By.name("sub")).click();

String actualUrl = "http://localhost/campuslocale/admin/subjects.php";
String expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl)) {
   System.out.println("Test passed"); }
else {
  System.out.println("Test failed"); }
driver.quit();
}

}
```



```java
package testcases;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import chromedriver.DriverSetup;
public class teacherreg {
public static WebDriver driver;
public static void main(String[] args) {
// TODO Auto-generated method stub
    driver =  DriverSetup.getWebDriver("http://localhost/campuslocale/");
    driver =  DriverSetup.getWebDriver("http://localhost/campuslocale/Login/login.php");

    //login-Invalid case
    driver.findElement(By.name("email")).sendKeys("admin@gmail.com");
    driver.findElement(By.name("password")).sendKeys("admin123*");
    driver.findElement(By.name("btnlogin")).click();

driver.get("http://localhost/campuslocale/admin/admin-index.php");
driver.get("http://localhost/campuslocale/admin/subjects.php");

driver.findElement(By.name("subject_code")).sendKeys("RLMCA200");
driver.findElement(By.name("subject_name")).sendKeys("Software Engineering");
driver.findElement(By.name("semester")).sendKeys("3");
driver.findElement(By.name("course_code")).sendKeys("MCS");
driver.findElement(By.name("credit_hours")).sendKeys("3");

driver.findElement(By.name("sub")).click();

String actualUrl = "http://localhost/campuslocale/admin/subjects.php";
String expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl)) {
       System.out.println("Test passed"); }
else {
    System.out.println("Test failed"); }
```

```
<terminated> teacherreg [Java Application] E:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.2.v20210201-0955\jre\bin\javaw.exe  (23-May-2022, 11:40:37 am – 11:40:50 am)
ChromeDriver was started successfully.
May 23, 2022 11:40:45 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
May 23, 2022 11:40:45 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 101
Test passed
```

5.2.1 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

5.2.2 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as BlackBox testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.1 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm’s need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs,
- Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

□ Careful planning.
□ Investigation of system and constraints.
□ Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won’t take place.

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy

6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The project as Campus Locale is the system that deals with the issue related manual college management system. This project is successfully implemented with all the features required for college. The system is user-friendly, highly interactive and flexible for further enhancement. The system generates the reports as when required. The coding is done in a simplified and understandable manner. The application provides appropriate information to user according to the chosen activity. The project is designed keeping in view the day to day problem faced by a manual college management system. Deployment of our College Management System help the college to reduce unnecessary wastage of time in doing work using manual college management system. Finally the system is implemented and tested according to test cases

7.2 FUTURE SCOPE

- Additional functionality can be added like, online multi chat application for better scope.
- .Can see the entire system more interactive and also be able to give statistics data.
- Run the entire system in Android platform for more ease of use and mobility.
- Make online exams more effective, efficient and more dynamic so that it helps to

  Get good support from the student and teachers.
- Adding an Parent module.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Gary B. Shelly, Harry J. Rosenblatt, "*System Analysis and Design*", 2009.
- Roger S Pressman, "*Software Engineering*", 1994.
- PankajJalote, "So*ftware engineering*: a precise approach", 2006.
- James lee and Brent ware Addison, "Open source web development with LAMP", 2003
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

WEBSITES:

- www.w3schools.com
- www.jquery.com
- https://app.diagrams.net
- http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf
- www.agilemodeling.com/artifacts/useCaseDiagram.html

CHAPTER 9

APPENDIX

9.1 SampleCode

STUDENT.PHP

```
<!---------------- Session starts form here ----------------------->
 <?php
 session_start();
 if (!$_SESSION["LoginAdmin"])
 {
 header('location:../login/login.php');
 }
 require_once "../connection/connection.php";
 $_SESSION["LoginStudent"]="";?>
 <!---------------- Session Ends form here ------------------------>
 <!--PHP code starts from here for data insertion into database  -->
 <?php  if (isset($_POST['btn_save'])) {
 $roll_no= $_POST["roll_no"];
 $first_name=$_POST["first_name"];
 $middle_name=$_POST["middle_name"];
 $last_name=$_POST["last_name"];
 $father_name=$_POST["father_name"];
 $email=$_POST["email"];
 $mobile_no=$_POST["mobile_no"];
 $course_code=$_POST['course_code'];
 $session=$_POST['session'];
 $prospectus_issued=$_POST["prospectus_issued"];
 $prospectus_amount=$_POST["prospectus_amount"];
 $form_b=$_POST["form_b"];
 $applicant_status=$_POST["applicant_status"];
 $application_status=$_POST["application_status"];
 $cnic=$_POST["cnic"];
 $dob=$_POST["dob"];
 $gender=$_POST["gender"];
 $permanent_address=$_POST["permanent_address"];
 $current_address=$_POST["current_address"];
 $place_of_birth=$_POST["place_of_birth"];
 $matric_complition_date=$_POST["matric_complition_date"];
 $matric_awarded_date=$_POST["matric_awarded_date"];
 $fa_complition_date=$_POST["fa_complition_date"];
 $fa_awarded_date=$_POST["fa_awarded_date"];
 $ba_complition_date=$_POST["ba_complition_date"];
 $ba_awarded_date=$_POST["ba_awarded_date"];
 $password=$_POST['password'];
 $role=$_POST['role'];
 // Images upload code starts here
 $profile_image = $_FILES['profile_image']['name'];$tmp_name=$_FILES['profile_image']
 ['tmp_name'];$path = "../Student/Images".$profile_image;move_uploaded_file($tmp_name, $path);
 $matric_certificate =
 $_FILES['matric_certificate']['name'];$tmp_name=$_FILES['matric_certificate']['tmp_name'];
$path = "images/".$matric_certificate;move_uploaded_file($tmp_name, $path);
$fa_certificate = $_FILES['fa_certificate']['name'];
$tmp_name=$_FILES['fa_certificate']['tmp_name'];
$path = "images/".$fa_certificate;move_uploaded_file($tmp_name, $path);
$ba_certificate = $_FILES['ba_certificate']['name'];
$tmp_name=$_FILES['ba_certificate']['tmp_name'];
$path = "images/".$ba_certificate;move_uploaded_file($tmp_name, $path);
 // Images upload code end here
 $query="Insert into student_info(roll_no,first_name,middle_name,last_name,father_name,email,
 mobile_no,course_code,session,profile_image,prospectus_issued,prospectus_amount,form_b,
 applicant_status,application_status,cnic,dob,gender,permanent_address,current_address,
 place_of_birth,matric_complition_date,matric_awarded_date,matric_certificate,fa_complition_date,
 fa_awarded_date,fa_certificate,ba_complition_date,ba_awarded_date,ba_certificate)
 values('$roll_no','$first_name','$middle_name','$last_name','$father_name',
 '$email','$mobile_no','$course_code','$session','$profile_image','$prospectus_issued',
 '$prospectus_amount','$form_b','$applicant_status','$applicationstatus','$cnic','$dob','$gender',
 '$permanent_address','$current_address','$place_of_birth','$matric_complition_date',
 '$matric_awarded_date','$matric_certificate','$fa_complition_date','$fa_awarded_date',
 '$fa_certificate','$ba_complition_date','$ba_awarded_date','$ba_certificate')";
  $run=mysqli_query($con, $query);
 if ($run) {
 echo "Your Data has been submitted";}
  else {
  echo "Your Data has not been submitted";}
  $query2="insert into login(user_id,Password,Role)values('$email','$password','$role')";
  $run2=mysqli_query($con, $query2);
  if ($run2) {
  echo "And submitted into login";}
  else {
  echo "Your Data has not been submitted into login";}
  }?>
 <?php
 if (isset($_POST['btn_save2'])) {
 $course_code=$_POST['course_code'];
 $semester=$_POST['semester'];
 $roll_no=$_POST['roll_no'];
 $subject_code=$_POST['subject_code'];
 $date=date("d-m-y");
 $query3="insert into
 student_courses(course_code,semester,roll_no,subject_code,assign_date)values('$course_code',
 '$semester','$roll_no','$subject_code','$date')";
 $run3=mysqli_query($con,$query3);
 if ($run3) {
 echo "Your Data has been submitted";
 }else
 {echo "Your Data has not been submitted";}}?>
 <!-- PHP code end from here for data insertion into database  -->
 <!doctype html>
 <html lang="en">
 <head>
 <title> Campus-Locale Admin - Register Student</title>
 </head>
 <body>
 <?php include('../common/common-header.php') ?>
 <?php include('../common/admin-sidebar.php') ?>
 <main role="main" class="col-xl-10 col-lg-9 col-md-8 ml-sm-auto px-md-4 w-100">
 <div class="sub-main">
 <div class="text-center d-flex flex-wrap flex-md-nowrap pt-3 pb-2 mb-3 text-white
```

```
admin-dashboard pl-3">
<div class="d-flex">
<h4 class="mr-5">Student Management System </h4>
<button type="button" class="btn btn-primary ml-5" data-toggle="modal"
data-target=".bd-example-modal-lg">Add Student</button></div>
</div><div class="col-md-2 pt-3 w-100">
<!-- Large modal -->
<div class="modal fade bd-example-modal-lg" tabindex="-1" role="dialog" aria-
labelledby="myLargeModalLabel" aria-hidden="true">
<div class="modal-dialog modal-lg">
<div class="modal-content">
<div class="modal-header bg-info text-white"> <h4 class="modal-title text-center">
Add New Student</h4>
</div>
<div class="modal-body">
 <form action="student.php" method="POST" enctype="multipart/form-data">
<div class="row mt-3">
<div class="col-md-4">
<div class="form-group">
<label for="exampleInputEmail1">Applicant First Name:*</label>
<input type="text" name="first_name" class="form-control" required>
</div></div>
<div class="col-md-4">
<div class="form-group">
<label for="exampleInputPassword1">Applicant Middle Name:</label>
<input type="text" name="middle_name" class="form-control">
</div>
</div>
<div class="col-md-4">
<div class="form-group">
<label for="exampleInputPassword1" required>Applicant Last Name:*</label>
<input type="text" name="last_name" class="form-control"> </div>
</div>
 </div>
 <div class="row">
 <div class="col-md-4"
 <div class="form-group">
 <label for="exampleInputEmail1">Father Name:*</label>
  <input type="text" name="father_name" class="form-control" required>
  </div>
  </div>
<div class="col-md-4">
 <div class="form-group">
 <label for="exampleInputPassword1">Student Roll No:</label>
<input type="text" name="roll_no" class="form-control">
</div>
</div>
<div class="col-md-4">
<div class="form-group"><label for="exampleInputPassword1">Applicant Email:*</label>
<input type="email" name="email" class="form-control" pattern="[a-z0-9._%+-]+@[a-z0-9.-]
+\.[a-z]{2,}$" required>
</div>
</div>
</div>
<div class="row">
<div class="col-md-4">
```

```
<div class="form-group">
<label for="exampleInputEmail1">Course which you want?: </label>
<select class="browser-default custom-select" name="course_code">
<option >Select Course</option>
<?php
$query="select course_code from courses";
$run=mysqli_query($con,$query);
while($row=mysqli_fetch_array($run)) {
echo     "<option value=".$row['course_code'].">".$row['course_code']."</option>";
                        }?>
</select>
</div></div>
<div class="col-md-4">
<div class="form-group">
<label for="exampleInputPassword1">Select Session:</label>
 <select class="browser-default custom-select" name="session">
 <option >Select Session</option><?php
 $query="select session from sessions";
$run=mysqli_query($con,$query);
while($row=mysqli_fetch_array($run)) {
echo     "<option value=".$row['session'].">".$row['session']."</option>";
}?></select>
 </div></div><div class="col-md-4">
<div class="form-group">
<label for="exampleInputPassword1">Your Profile Image:</label>
<input type="file" name="profile_image" placeholder="Student Age" class="form-control">
 </div></div>
</div>
<div class="row">
<div class="col-md-4">


<div class="form-group">
<label for="exampleInputEmail1">Prospectus Issude: </label>
<select class="browser-default custom-select" name="prospectus_issued">
<option>Select Option</option>
 <option value="Yes">Yes</option>
  <option value="No">No</option>
</select>
</div>
</div>
<div class="col-md-4">
<div class="form-group">
<label for="exampleInputPassword1">Prospectus Amount Recvd:</label>
<select class="browser-default custom-select" name="prospectus_amount">
<option>Select Option</option>
<option value="Yes">Yes</option>
<option value="No">No</option>
</select>
</div></div>
<div class="col-md-4">
<div class="form-group">
  <label for="exampleInputPassword1">Form B:</label>
<input type="text" name="form_b" class="form-control">
  </div></div>
</div>
```

```
<div class="row">
<div class="col-md-4">
<div class="form-group">
 <label for="exampleInputEmail1">Applicant Status: </label>
<select class="browser-default custom-select" name="applicant_status">
<option>Select Option</option>
<option value="Admitted">Admitted</option>
<option value="Not Admitted">Not Admitted</option>
</select>
</div>
</div>
<div class="col-md-4">
<div class="form-group">
<label for="exampleInputPassword1">Application Status:</label>
<select class="browser-default custom-select" name="application_status">
<option>Select Option</option>
 <option value="Approved">Approved</option>
<option value="Not Approved">Not Approved</option></select>
</div>
</div>
<div class="col-md-4">
<label for="exampleInputPassword1">CNIC No:</label><input type="text" name="cnic" data-
inputmask="'mask': '99999-9999999-9'" placeholder="XXXXX-XXXXXXX-X" class="form-control">
</div></div>
</div>
<div class="row">
<div class="col-md-4">
<div class="form-group">
<label for="exampleInputEmail1">Date of Birth: </label>


<input type="date" name="dob" class="form-control">
</div>
</div>
<div class="col-md-4">
<div class="form-group">
<label for="exampleInputPassword1">Mobile No:*</label>
<input type="number" name="mobile_no" class="form-control" required>
</div>
</div>
<div class="col-md-4">
<div class="form-group">
<label for="exampleInputPassword1">Gender:</label>
<select class="browser-default custom-select" name="gender">
<option>Select Gender</option>
<option value="Male">Male</option>
<option value="Female">Female</option>
</select>
</div>
</div>
</div>
<div class="row">
<div class="col-md-4">
<div class="form-group">
 <label for="exampleInputEmail1">Permanent Address: </label>
<input type="text" name="permanent_address" class="form-control">
```

```
</div>
</div>
<div class="col-md-4">
<div class="form-group">
<label for="exampleInputPassword1">Current Address:</label>
<input type="text" name="current_address" class="form-control">
</div>
 </div>
<div class="col-md-4">
<div class="form-group">
 <label for="exampleInputPassword1">Place of Birth:</label>
 <input type="text" name="place_of_birth" class="form-control">
 </div>
</div>
</div>
<div class="row">
<div class="col-md-4">
<div class="form-group">
<label for="exampleInputEmail1">Matric/OLevel Complition Date: </label>
<input type="date" name="matric_complition_date" class="form-control">
</div>
</div>
<div class="col-md-4"><div class="form-group">
<label for="exampleInputPassword1">Matric/OLevel Awarded Date:</label>
<input type="date" name="matric_awarded_date" class="form-control">
</div>
</div>
<div class="col-md-4">
<div class="form-group">

 <label for="exampleInputPassword1">Upload Matric/OLevel Certificate:</label>
 <input type="file" name="matric_certificate" class="form-control" value="there is no image"> </div>
</div>
</div>
<div class="row">
<div class="col-md-4">
<div class="form-group">
<label for="exampleInputEmail1">FA/ALevel Complition Date: </label>
<input type="date" name="fa_complition_date" class="form-control">
</div></div>
<div class="col-md-4">
<div class="form-group">
<label for="exampleInputPassword1">FA/ALevel Awarded Date:</label>
<input type="date" name="fa_awarded_date" class="form-control"></div>
</div>
<div class="col-md-4">
<div class="form-group"><label for="exampleInputPassword1">Upload FA/ALevel Certificate:
</label>
<input type="file" name="fa_certificate" class="form-control" value="there is no image" >
</div>
</div></div><div class="row">
<div class="col-md-4">
<div class="form-group">
<label for="exampleInputEmail1">BA Complition Date: </label>
<input type="date" name="ba_complition_date" class="form-control" value="0">
</div>
```

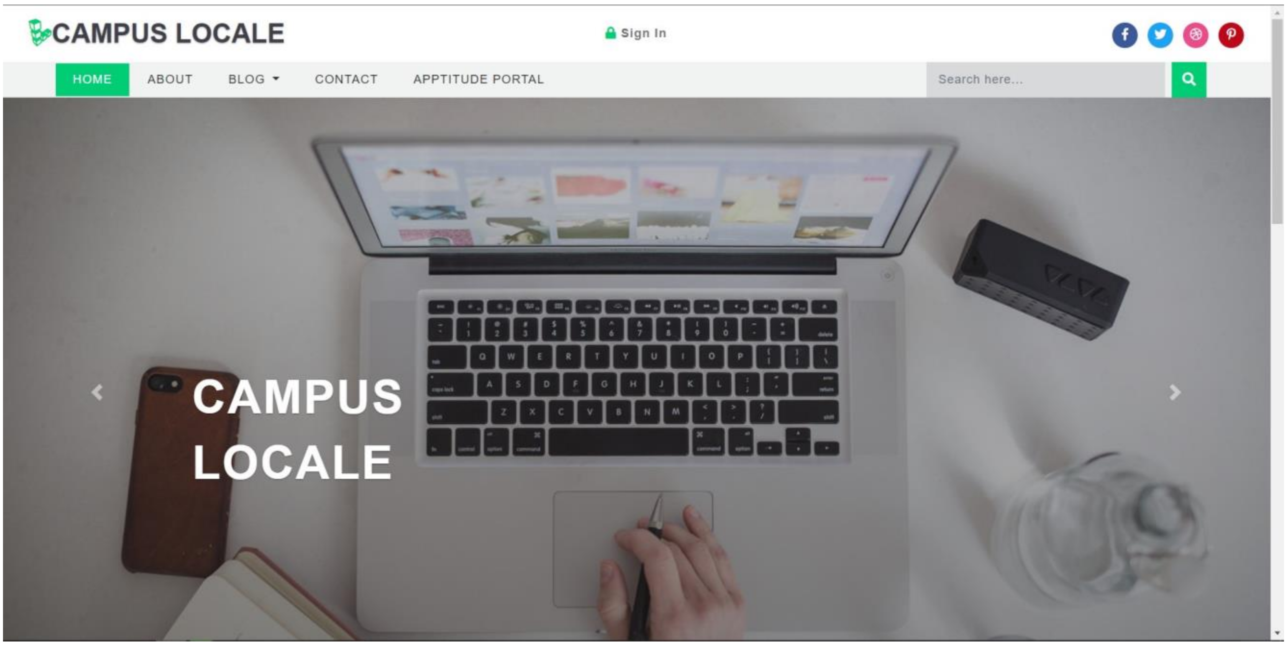
```
</div>
<div class="col-md-4">
<div class="form-group">
<label for="exampleInputPassword1">BA Awarded Date:</label>
<input type="date" name="ba_awarded_date" class="form-control">
 </div>
</div>
<div class="col-md-4"><div class="form-group">
<label for="exampleInputPassword1">Upload BA Certificate:</label>
<input type="file" value="C:/xampp/htdocs/Imperial University/Images/no-image-available.jpg"
name="ba_certificate" class="form-control" >
</div>
</div>
</div>
<div>
<input type="hidden" name="password" value="student123*">

<input type="hidden" name="role" value="Student">
</div>
<div class="modal-footer">
<input type="submit" class="btn btn-primary" name="btn_save">
<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button></div>
</form></div></div>
</div></div></div>
<div class="row w-100">
<div class="col-md-12 ml-2">
<section class="mt-3">
<div class="row">
<div class="col-md-6">
<form action="search_student.php" method="post">
<div class="form-group">
<label for="exampleInputPassword1"><h5>Search:</h5></label>
<div class="d-flex">
<input type="text" name="search" id="search" class="form form-control" placeholder="Enter I'd">
<input class="btn btn-primary px-4 ml-4" type="submit" name="btnSearch" value="Search">
</div>
</div>
</form></div>
<div class="col-md-12 pt-5 mb-2">


<!-- Large modal -->

<button type="button" class="btn btn-primary ml-auto" data-toggle="modal"
 data-target=".bd-example-modal-lg1">Assign Subjects</button>
<a class="btn btn-success" href="asign-single-student-subjects.php">
 Assign Single Student Subject</a>
<div class="modal fade bd-example-modal-lg1" tabindex="-1" role="dialog" aria-
labelledby="myLargeModalLabel" aria-hidden="true">
<div class="modal-dialog modal-lg">
<div class="modal-content">
<div class="modal-header bg-info text-white">
<h4 class="modal-title text-center">Assign Subjects To Student</h4>
</div>
<div class="modal-body">
<form action="student.php" method="POST" enctype="multipart/form-data">
```

```
<div class="row mt-3">
<div class="col-md-6">
<div class="form-group">
<label for="exampleInputEmail1">Select Course:*</label>
<select class="browser-default custom-select" name="course_code" required="">
<option >Select Course</option><?php
$query="select course_code from courses";
$run=mysqli_query($con,$query);
while($row=mysqli_fetch_array($run)) {
echo"<option value=".$row['course_code'].">".$row['course_code']."</option>";}?>
</select>
</div>
</div>
<div class="col-md-6">
<div class="form-group">
<label for="exampleInputPassword1" required>Enter Semester:*</label>
<input type="text" name="semester" class="form-control">
</div</div</div>
<div class="row">
<div class="col-md-6">
<div class="form-group">
<label for="exampleInputPassword1">Enter Roll No:*</label>
<input type="text" name="roll_no" class="form-control">
</div>
</div>
<div class="col-md-6">
<div class="form-group">
<label for="exampleInputPassword1">Select Subject:*</label>
<select class="browser-default custom-select" name="subject_code" required="">
<option >Select Subject</option>
<?php
$query="select subject_code from course_subjects";
$run=mysqli_query($con,$query);
while($row=mysqli_fetch_array($run)) {
echo"<option value=".$row['subject_code'].">".$row['subject_code']."</option>";}?
</select>
</div>
</div>
</div>
<div class="modal-footer">
<input type="submit" class="btn btn-primary" name="btn_save2">
<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
<table class="w-100 table-elements mb-5 table-three-tr text-center" cellpadding="10">
<tr class="table-tr-head table-three text-white">
<th>Roll.No</th>
<th>Student Name</th>
<th>Current Address</th>
<th>Session</th>
```

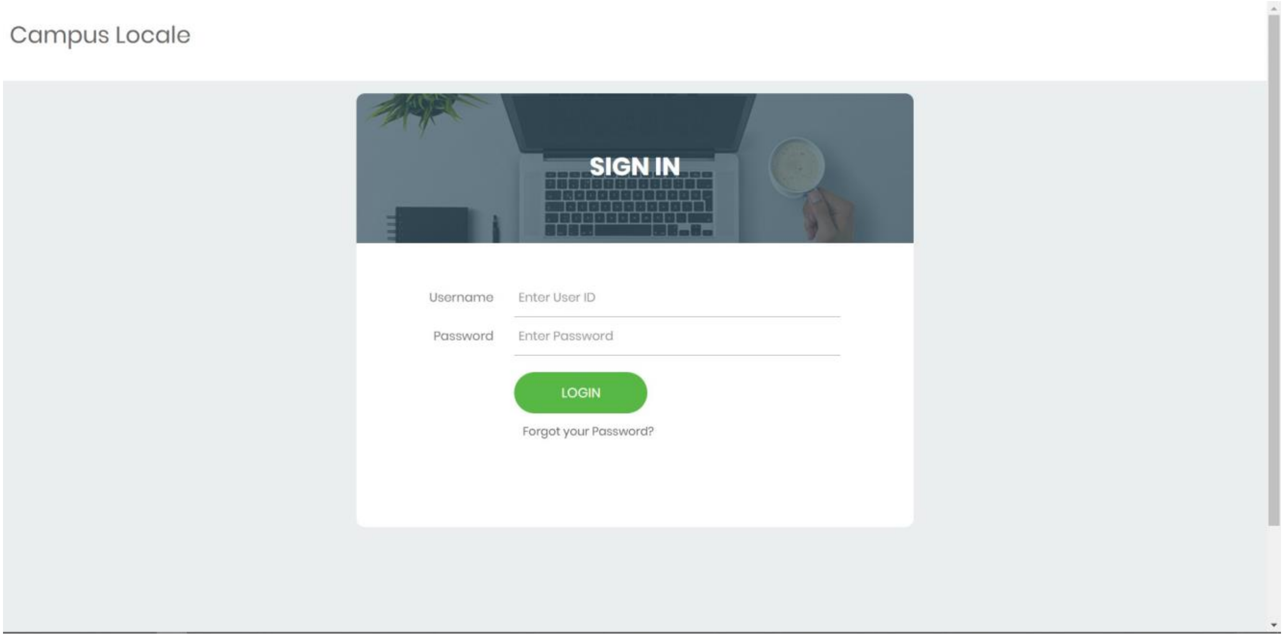
```
<th>Course ID</th>

<th>Admission</th>
<th>Profile</th>
<th colspan="1">Operations</th></tr>
<?php
$query="select first_name,middle_name,admission_date,last_name,current_address,
session,roll_no,form_b ,profile_image,course_code from student_info";
$run=mysqli_query($con,$query);
while($row=mysqli_fetch_array($run)) {?>
<tr>
<td><?php echo $row["roll_no"] ?></td>
<td><?php echo $row["first_name"]." ".$row["middle_name"]." ".$row["last_name"] ?></td>
<td><?php echo $row["current_address"] ?></td>
<td><?php echo $row["session"] ?></td>
<td><?php echo $row["course_code"] ?></td>
<td><?php echo date("Y-M-d",strtotime($row["admission_date"])); ?></td>
<td><?php  $profile_image= $row["profile_image"] ?>
<img height='50px' width='50px' src=<?php echo "images/$profile_image"  ?> >
</td>
<td width='170'>
<?php
echo "<a class='btn btn-primary' href=display-student.php?roll_no=".$row['roll_no'].">Profile</a>
<a class='btn btn-danger' href=delete-function.php?roll_no=".$row['roll_no'].">Delete</a> "?>
</td>
</tr>
<?php }
?>
</table>
</section>
</div>
</div>
</div>
</main>
<script type="text/javascript" src="../bootstrap/js/jquery.min.js"></script>
<script type="text/javascript" src="../bootstrap/js/bootstrap.min.js"></script>
</body>
</html>
```

9.2 Screen Shots

9.2.1 Homepage



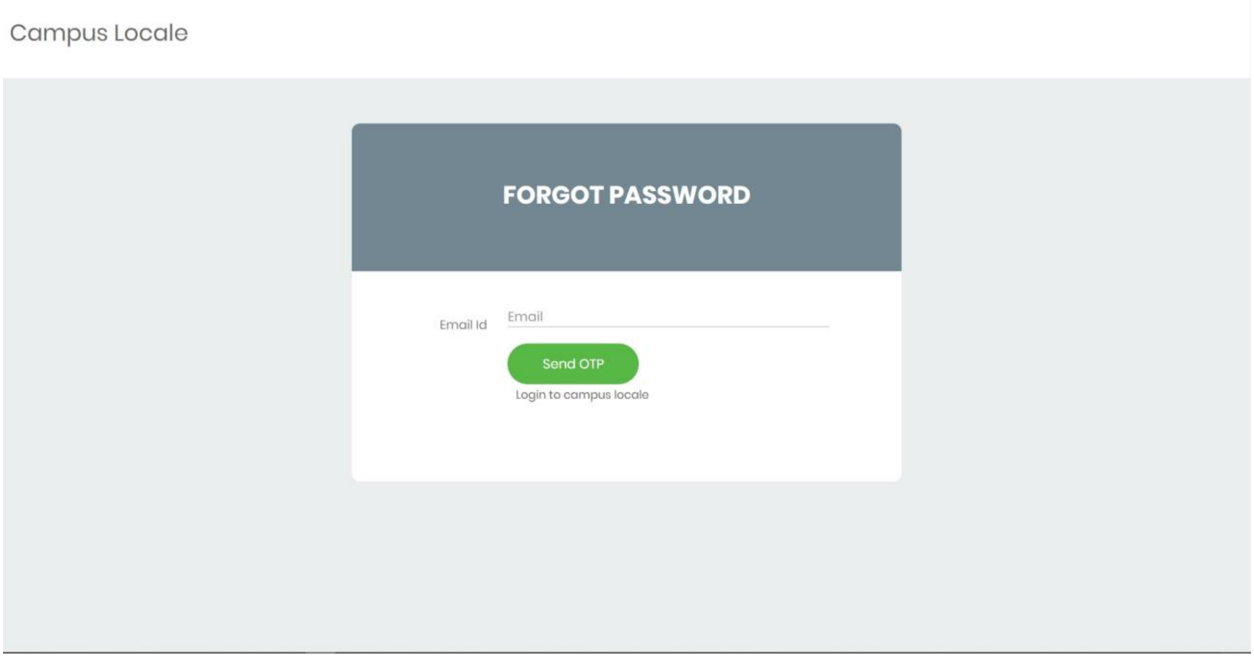
9.2.2. Login Page



Username Enter User ID

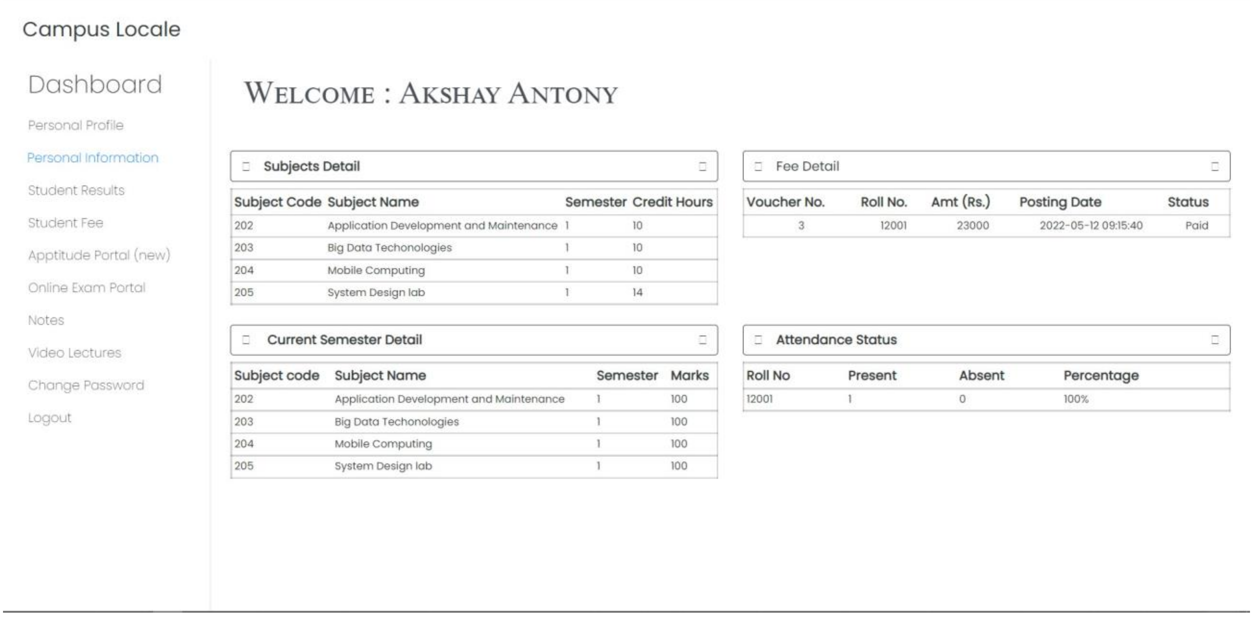
Password Enter Password

9.2.3. Forgot Password



STUDENT PAGES

9.2.4. Student Dashboard



WELCOME : AKSHAY ANTONY

Subjects Detail

Subject Code	Subject Name	Semester	Credit Hours
202	Application Development and Maintenance	1	10
203	Big Data Techonologies	1	10
204	Mobile Computing	1	10
205	System Design lab	1	14

Fee Detail

Voucher No.	Roll No.	Amt (Rs.)	Posting Date	Status
3	12001	23000	2022-05-12 09:15:40	Paid

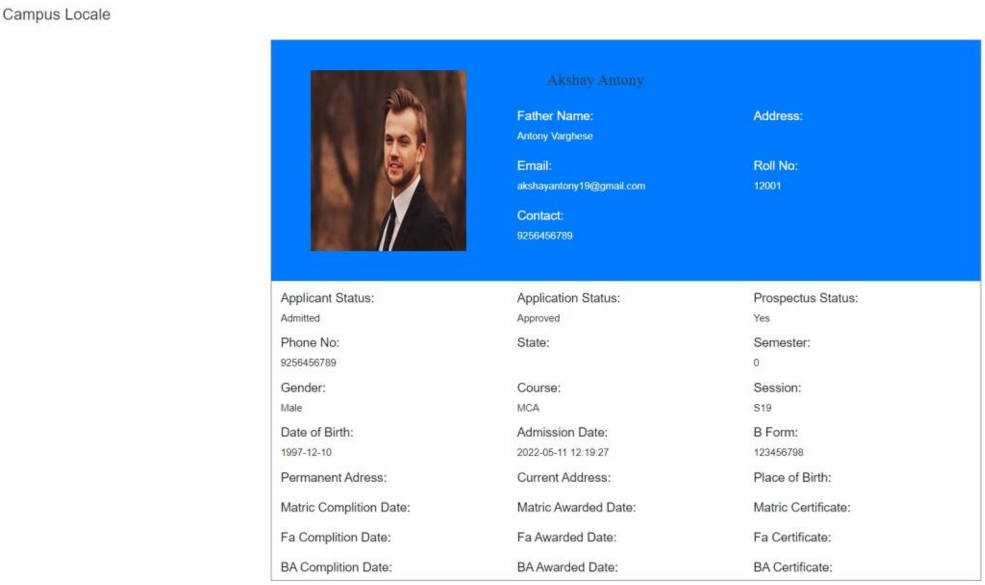
Current Semester Detail

Subject code	Subject Name	Semester	Marks
202	Application Development and Maintenance	1	100
203	Big Data Techonologies	1	100
204	Mobile Computing	1	100
205	System Design lab	1	100

Attendance Status

Roll No	Present	Absent	Percentage
12001	1	0	100%

9.2.5 Student Profile



Akshay Antony

Father Name:
Antony Varghese

Address:

Email:
akshayantony19@gmail.com

Roll No:
12001

Contact:
9256456789

Applicant Status:
Admitted

Application Status:
Approved

Prospectus Status:
Yes

Phone No:
9256456789

State:

Semester:
0

Gender:
Male

Course:
MCA

Session:
S19

Date of Birth:
1997-12-10

Admission Date:
2022-05-11 12:19:27

B Form:
123456798

Permanent Adress:

Current Address:

Place of Birth:

Matric Complition Date:

Matric Awarded Date:

Matric Certificate:

Fa Complition Date:

Fa Awarded Date:

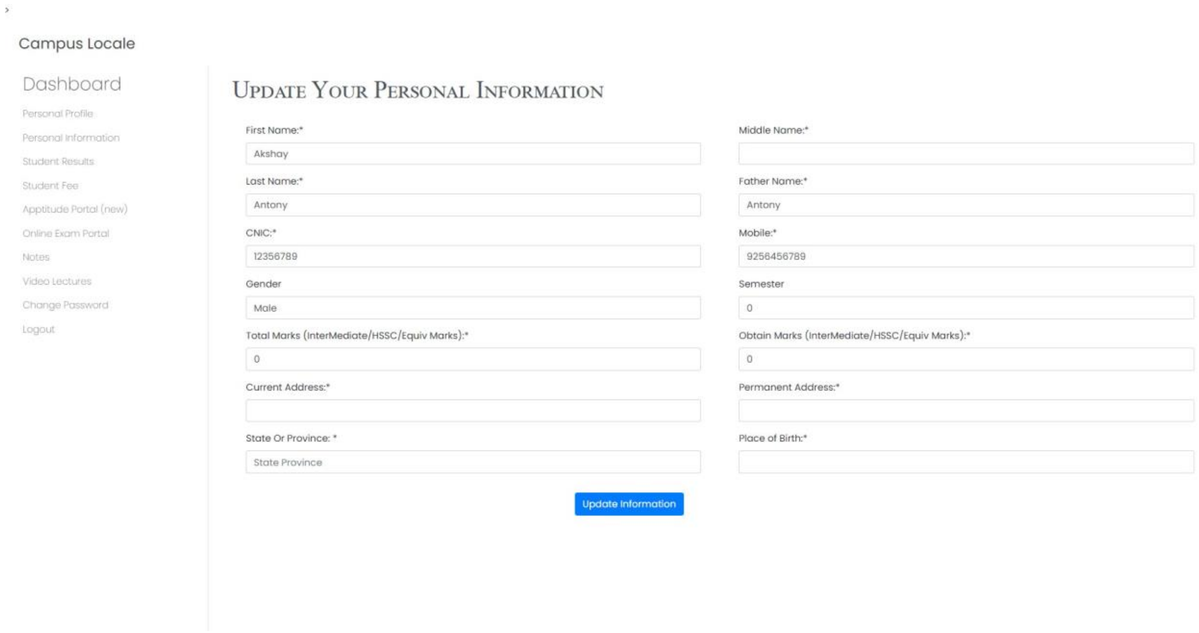
Fa Certificate:

BA Complition Date:

BA Awarded Date:

BA Certificate:

9.2.6 Update Profile



UPDATE YOUR PERSONAL INFORMATION

First Name:*
Akshay

Middle Name:*

Last Name:*
Antony

Father Name:*
Antony

CNIC:*
12356789

Mobile:*
9256456789

Gender
Male

Semester
0

Total Marks (InterMediate/HSSC/Equiv Marks):*
0

Obtain Marks (InterMediate/HSSC/Equiv Marks):*
0

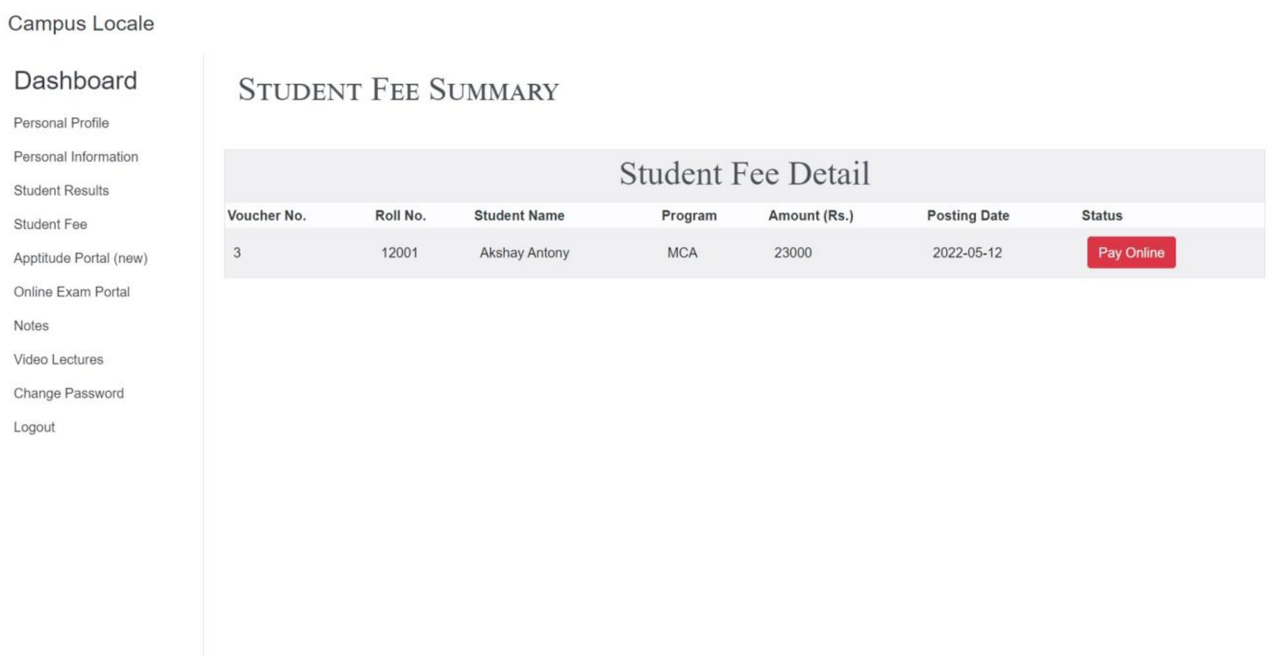
Current Address:*

Permanent Address:*

State Or Province: *
State Province

Place of Birth:*

9.2.7 Student Fees

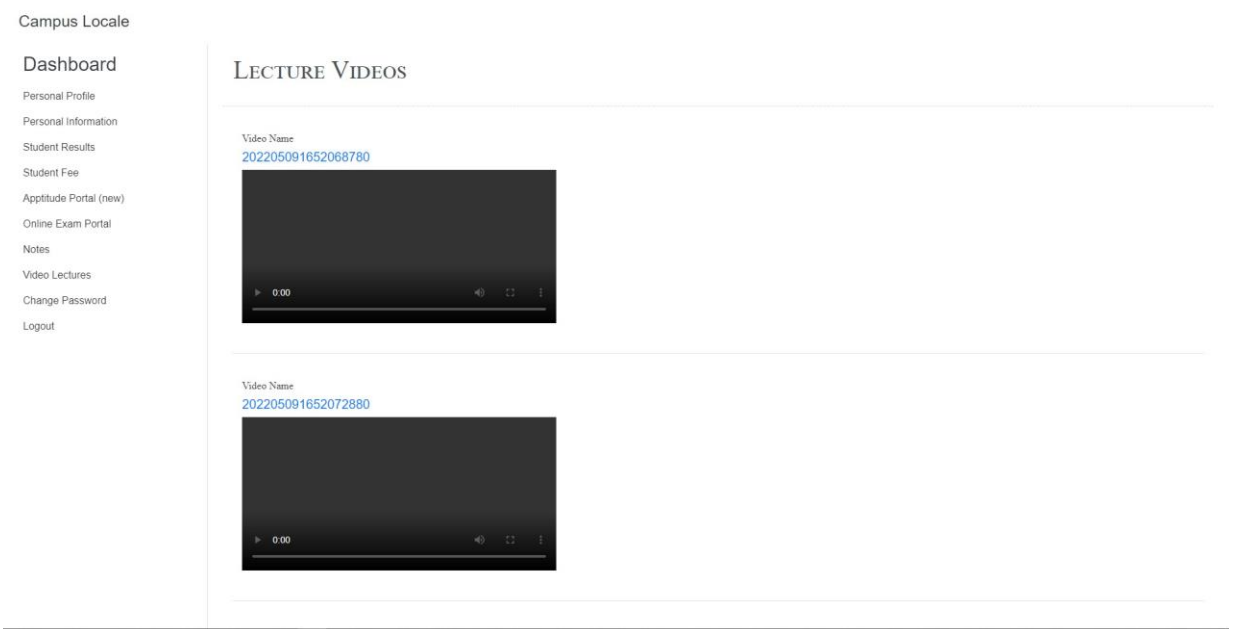


STUDENT FEE SUMMARY

Student Fee Detail

Voucher No.	Roll No.	Student Name	Program	Amount (Rs.)	Posting Date	Status
3	12001	Akshay Antony	MCA	23000	2022-05-12	Pay Online

9.2.8 Lecture Videos



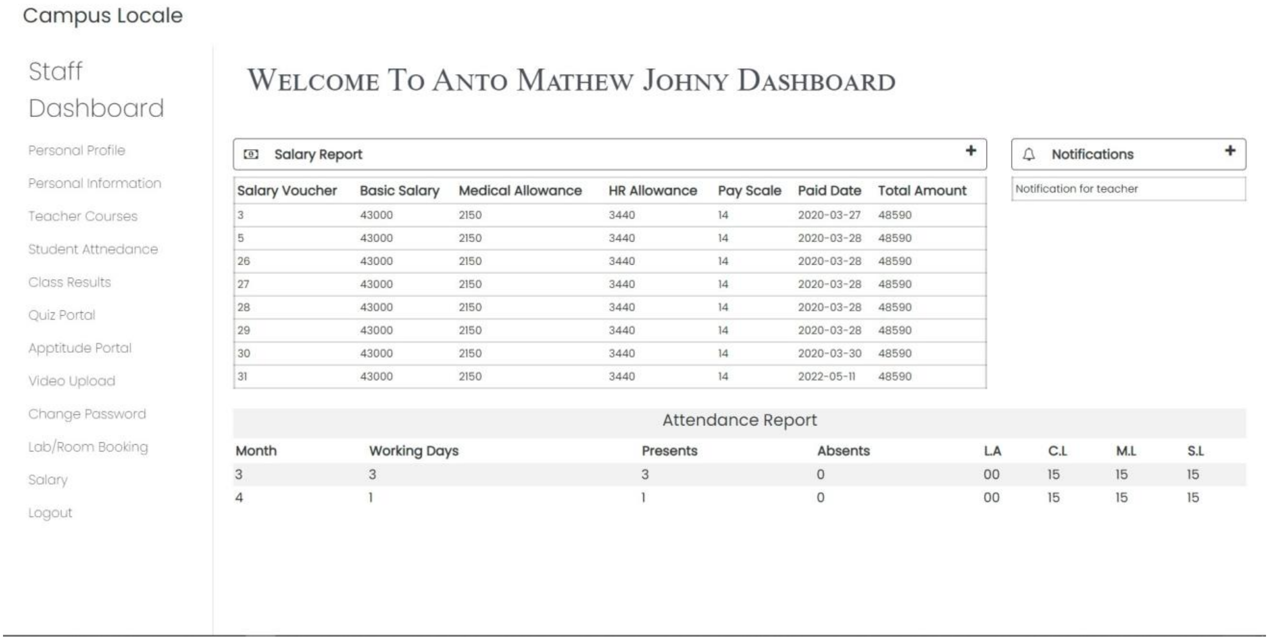
LECTURE VIDEOS

Video Name
202205091652068780

Video Name
202205091652072880

TEACHER PAGES

9.3.1 Home Page



WELCOME TO ANTO MATHEW JOHNY DASHBOARD

Salary Report +

Salary Voucher	Basic Salary	Medical Allowance	HR Allowance	Pay Scale	Paid Date	Total Amount
3	43000	2150	3440	14	2020-03-27	48590
5	43000	2150	3440	14	2020-03-28	48590
26	43000	2150	3440	14	2020-03-28	48590
27	43000	2150	3440	14	2020-03-28	48590
28	43000	2150	3440	14	2020-03-28	48590
29	43000	2150	3440	14	2020-03-28	48590
30	43000	2150	3440	14	2020-03-30	48590
31	43000	2150	3440	14	2022-05-11	48590

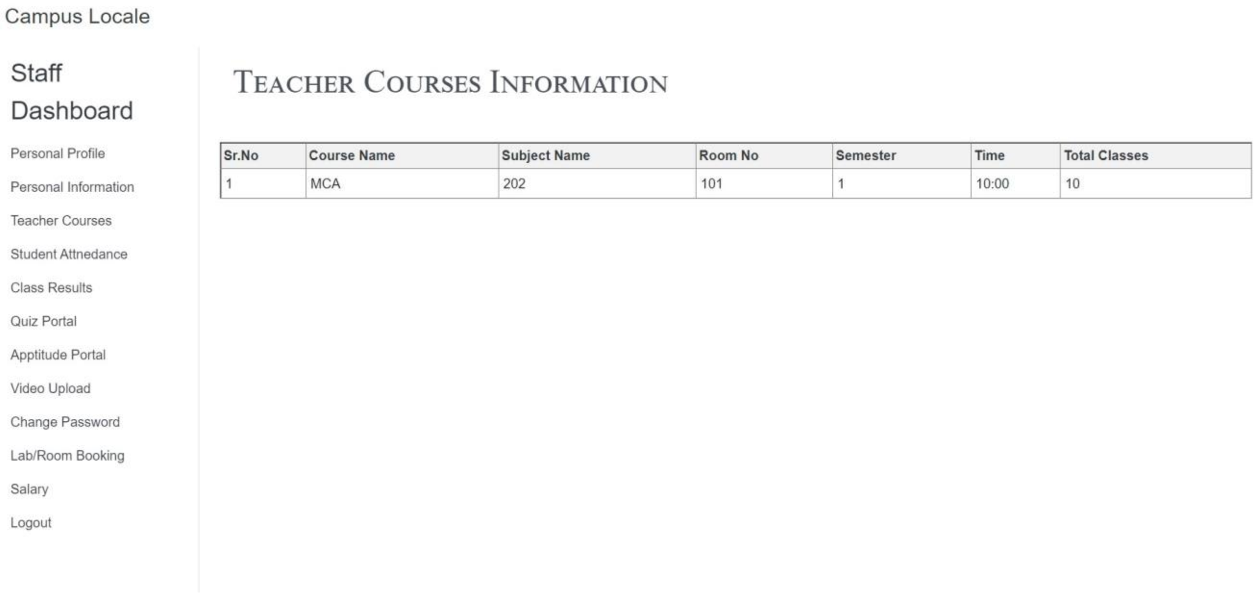
Notifications +

Notification for teacher

Attendance Report

Month	Working Days	Presents	Absents	L.A	C.L	M.L	S.L
3	3	3	0	00	15	15	15
4	1	1	0	00	15	15	15

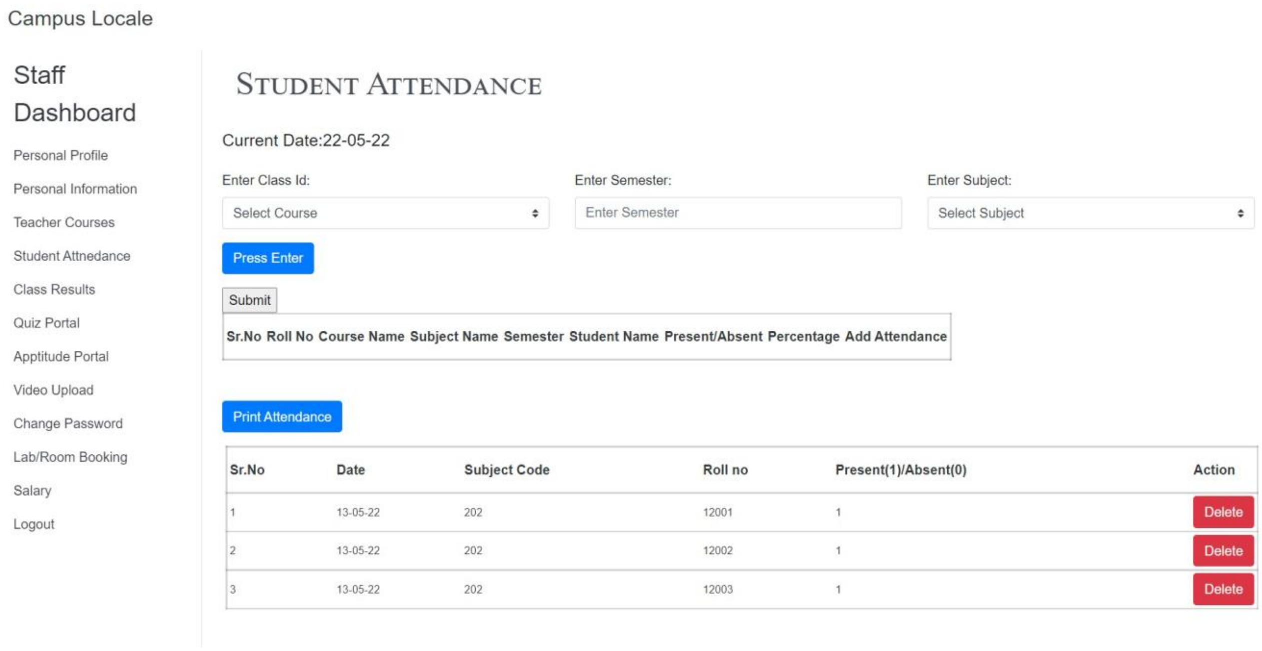
9.3.2 Course Page



TEACHER COURSES INFORMATION

Sr.No	Course Name	Subject Name	Room No	Semester	Time	Total Classes
1	MCA	202	101	1	10:00	10

9.3.3 Attendance Page



Campus Locale

Staff Dashboard

STUDENT ATTENDANCE

Current Date:22-05-22

Enter Class Id: Select Course

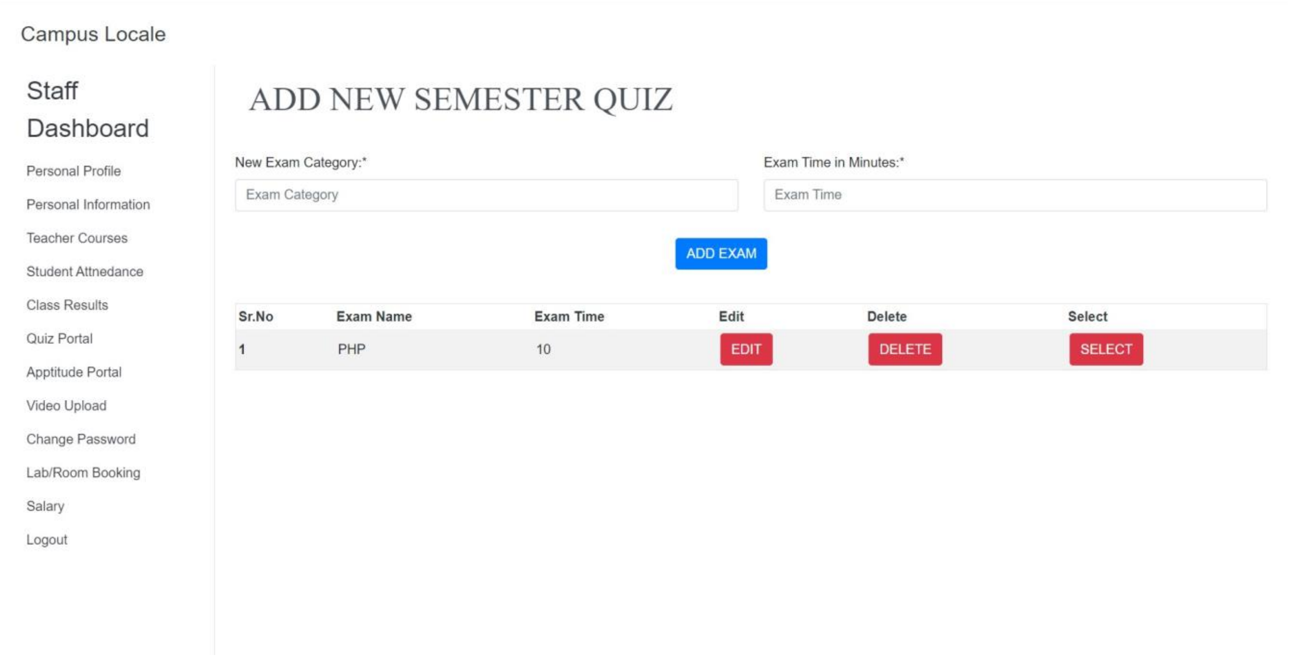
Enter Semester: Enter Semester

Enter Subject: Select Subject

Sr.No	Roll No	Course Name	Subject Name	Semester	Student Name	Present/Absent	Percentage	Add Attendance

Sr.No	Date	Subject Code	Roll no	Present(1)/Absent(0)	Action
1	13-05-22	202	12001	1	Delete
2	13-05-22	202	12002	1	Delete
3	13-05-22	202	12003	1	Delete

9.3.4 Quiz Adding



Campus Locale

Staff Dashboard

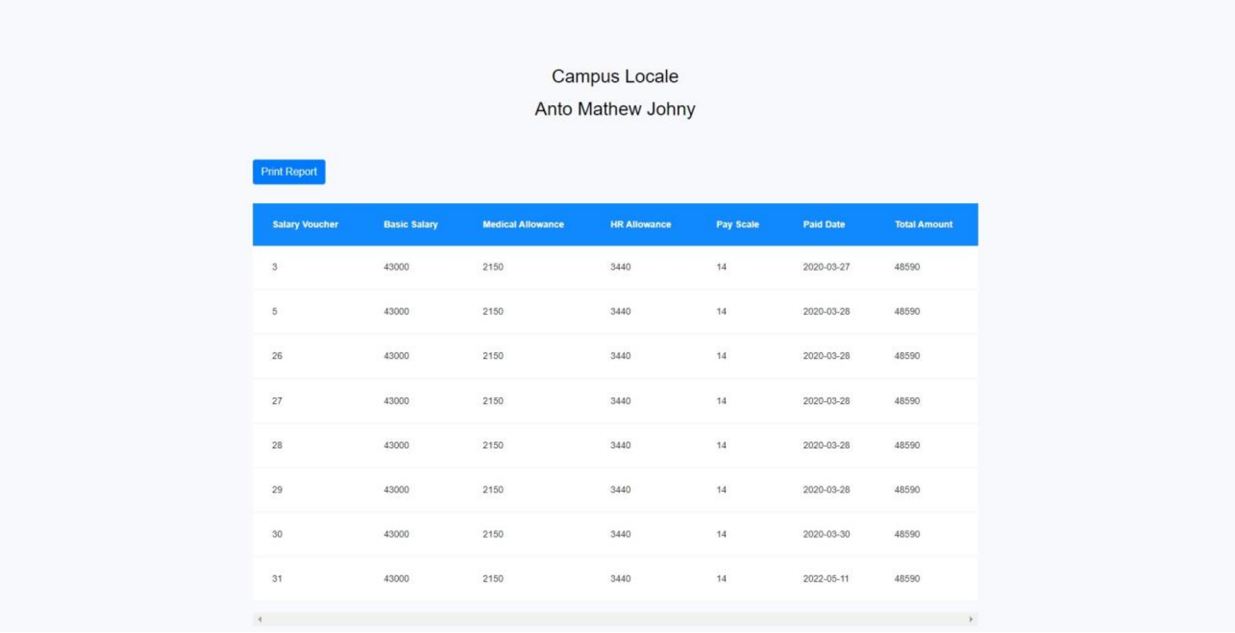
ADD NEW SEMESTER QUIZ

New Exam Category:* Exam Category

Exam Time in Minutes:* Exam Time

Sr.No	Exam Name	Exam Time	Edit	Delete	Select
1	PHP	10	EDIT	DELETE	SELECT

9.3.5 Salary report

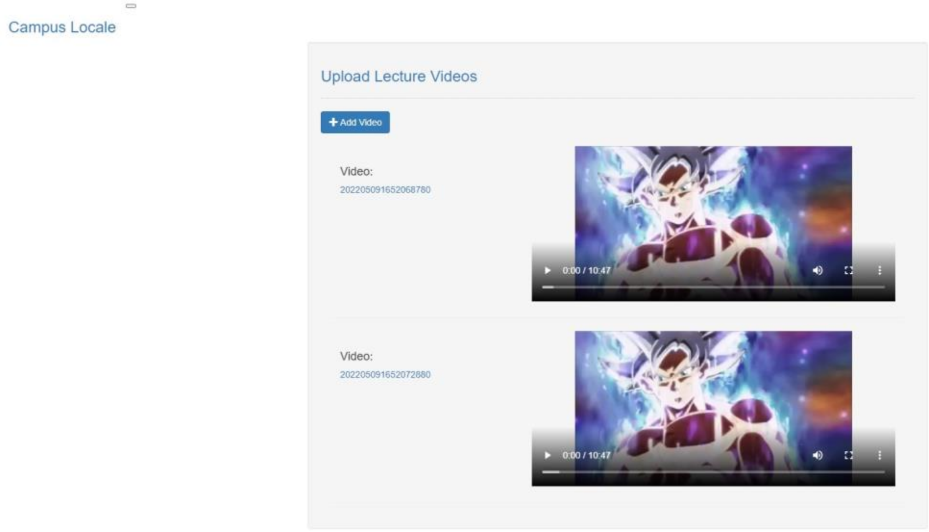


Campus Locale

Anto Mathew Johny

Salary Voucher	Basic Salary	Medical Allowance	HR Allowance	Pay Scale	Paid Date	Total Amount
3	43000	2150	3440	14	2020-03-27	48590
5	43000	2150	3440	14	2020-03-28	48590
26	43000	2150	3440	14	2020-03-28	48590
27	43000	2150	3440	14	2020-03-28	48590
28	43000	2150	3440	14	2020-03-28	48590
29	43000	2150	3440	14	2020-03-28	48590
30	43000	2150	3440	14	2020-03-30	48590
31	43000	2150	3440	14	2022-05-11	48590

9.3.6 Video Upload

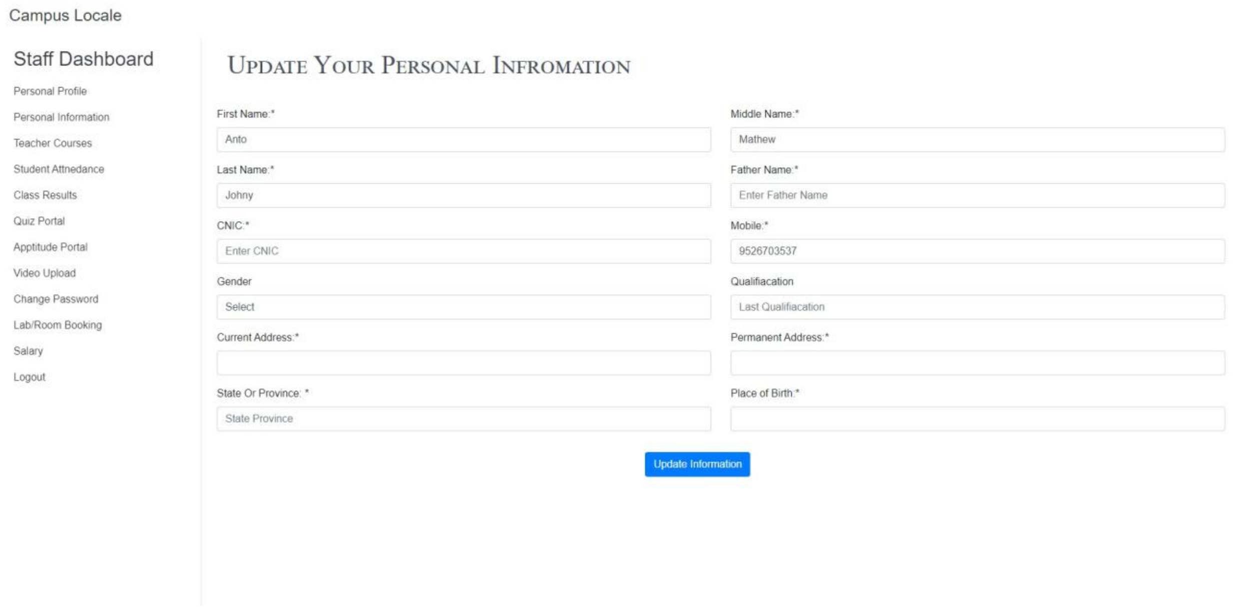


Upload Lecture Videos

Video:
202205091652068780

Video:
202205091652072880

9.3.7 Update Info

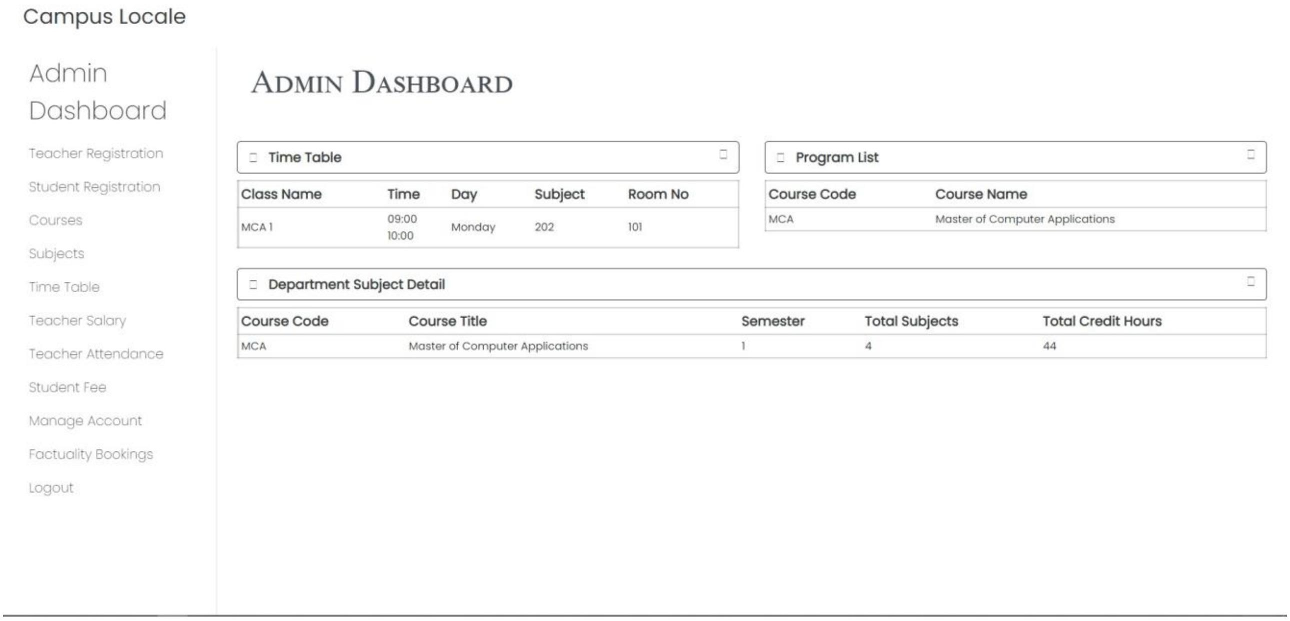


UPDATE YOUR PERSONAL INFROMATION

First Name:* Anto	Middle Name:* Mathew
Last Name:* Johny	Father Name:* Enter Father Name
CNIC:* Enter CNIC	Mobile:* 9526703537
Gender Select	Qualifiacation Last Qualifiacation
Current Address:*	Permanent Address:*
State Or Province: * State Province	Place of Birth:*

ADMIN PAGES

9.4.1 Admin Dashboard Page



ADMIN DASHBOARD

Time Table

Class Name	Time	Day	Subject	Room No
MCA 1	09:00 10:00	Monday	202	101

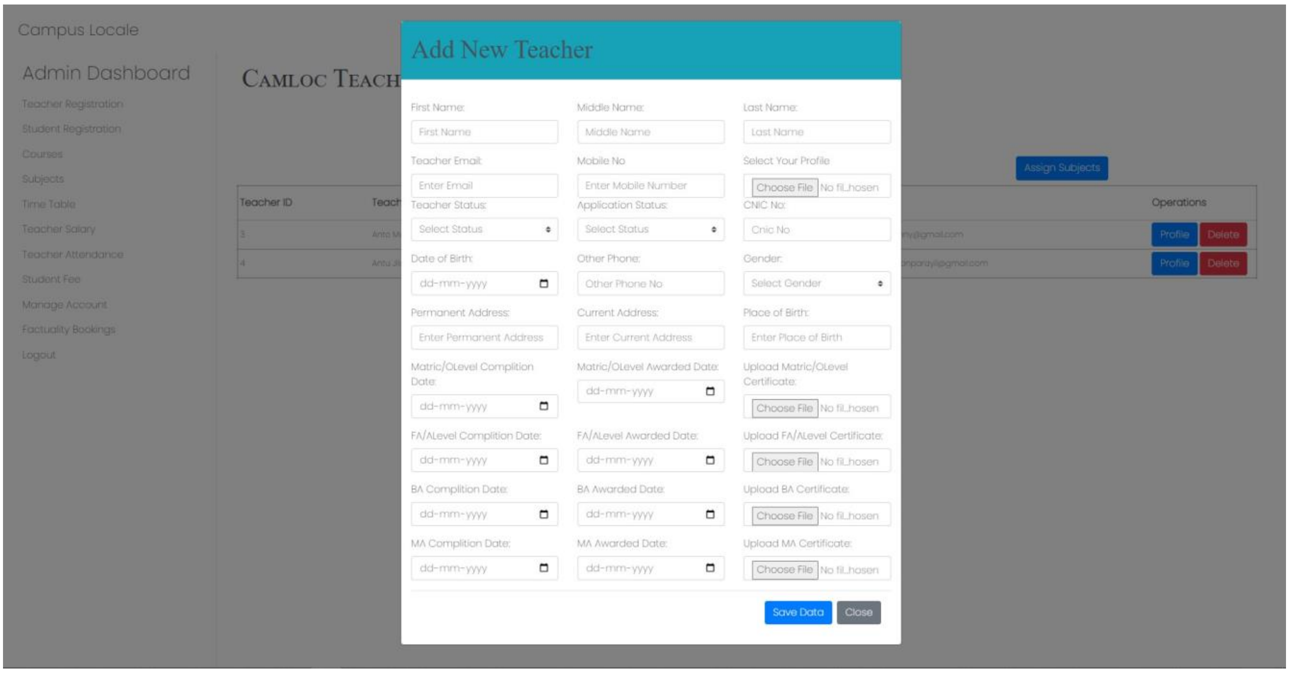
Program List

Course Code	Course Name
MCA	Master of Computer Applications

Department Subject Detail

Course Code	Course Title	Semester	Total Subjects	Total Credit Hours
MCA	Master of Computer Applications	1	4	44

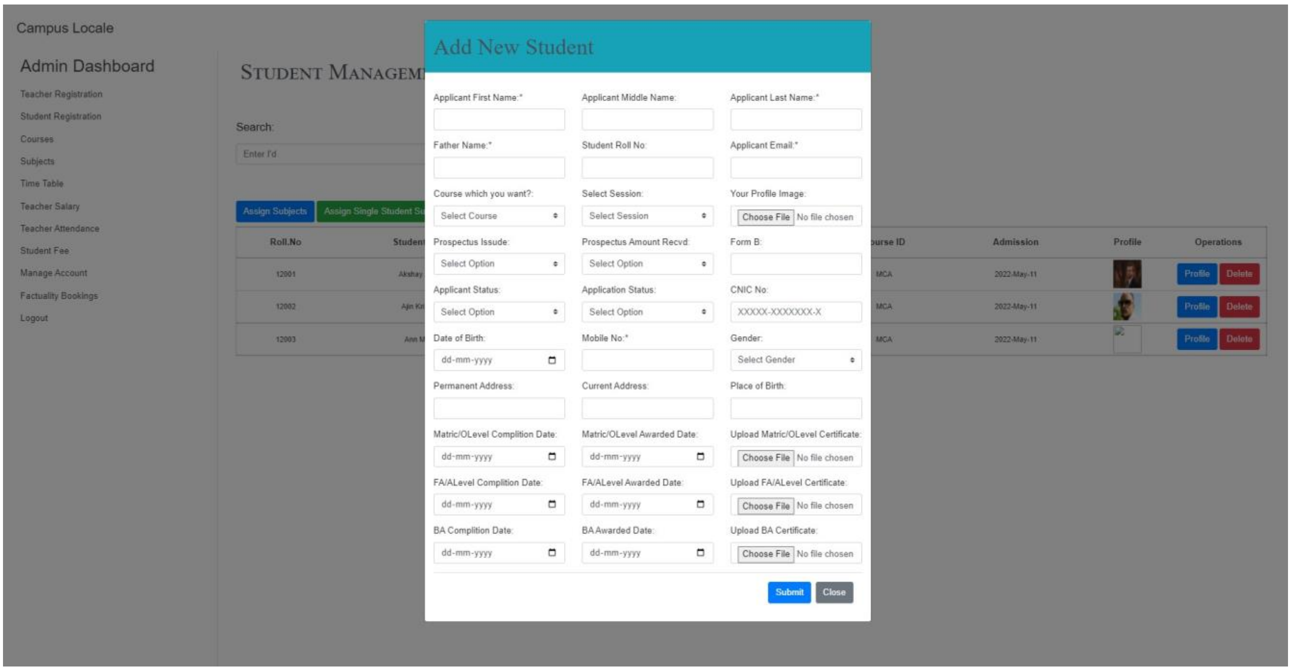
9.4.2 Add Teacher



Add New Teacher

First Name: First Name
Middle Name: Middle Name
Last Name: Last Name
Teacher Email: Enter Email
Mobile No: Enter Mobile Number
Select Your Profile: Choose File No fil..hosen
Teacher Status: Select Status
Application Status: Select Status
CNIC No: Cnic No
Date of Birth: dd-mm-yyyy
Other Phone: Other Phone No
Gender: Select Gender
Permanent Address: Enter Permanent Address
Current Address: Enter Current Address
Place of Birth: Enter Place of Birth
Matric/OLevel Complition Date: dd-mm-yyyy
Matric/OLevel Awarded Date: dd-mm-yyyy
Upload Matric/OLevel Certificate: Choose File No fil..hosen
FA/ALevel Complition Date: dd-mm-yyyy
FA/ALevel Awarded Date: dd-mm-yyyy
Upload FA/ALevel Certificate: Choose File No fil..hosen
BA Complition Date: dd-mm-yyyy
BA Awarded Date: dd-mm-yyyy
Upload BA Certificate: Choose File No fil..hosen
MA Complition Date: dd-mm-yyyy
MA Awarded Date: dd-mm-yyyy
Upload MA Certificate: Choose File No fil..hosen

9.4.3 Add Student

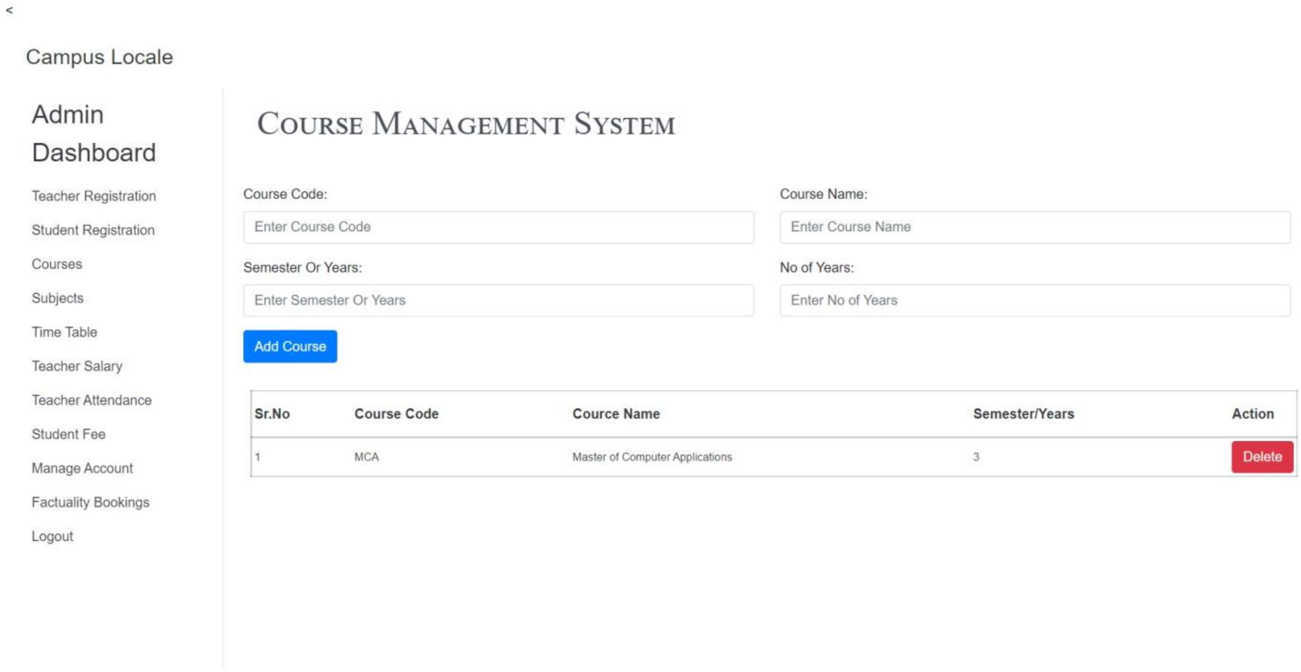


Add New Student

Roll.No	Student	ourse ID	Admission	Profile	Operations
12001	Akshay	MCA	2022-May-11		Profile Delete
12002	Ajin Kr	MCA	2022-May-11		Profile Delete
12003	Ann M	MCA	2022-May-11		Profile Delete

Applicant First Name:*
Applicant Middle Name:
Applicant Last Name:*
Father Name:*
Student Roll No:
Applicant Email:*
Course which you want?: Select Course
Select Session: Select Session
Your Profile Image: Choose File No file chosen
Prospectus Issude: Select Option
Prospectus Amount Recvd: Select Option
Form B:
Applicant Status: Select Option
Application Status: Select Option
CNIC No: XXXXX-XXXXXXX-X
Date of Birth: dd-mm-yyyy
Mobile No:*
Gender: Select Gender
Permanent Address:
Current Address:
Place of Birth:
Matric/OLevel Complition Date: dd-mm-yyyy
Matric/OLevel Awarded Date: dd-mm-yyyy
Upload Matric/OLevel Certificate: Choose File No file chosen
FA/ALevel Complition Date: dd-mm-yyyy
FA/ALevel Awarded Date: dd-mm-yyyy
Upload FA/ALevel Certificate: Choose File No file chosen
BA Complition Date: dd-mm-yyyy
BA Awarded Date: dd-mm-yyyy
Upload BA Certificate: Choose File No file chosen

9.4.4 Add Course



COURSE MANAGEMENT SYSTEM

Course Code: Enter Course Code

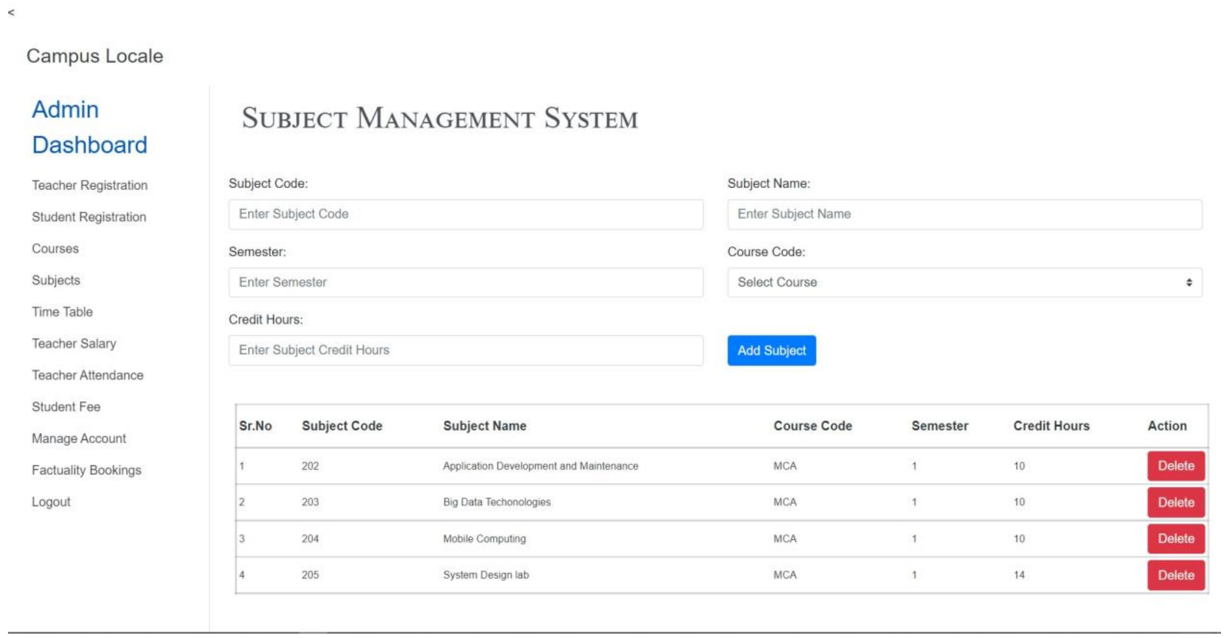
Course Name: Enter Course Name

Semester Or Years: Enter Semester Or Years

No of Years: Enter No of Years

Sr.No	Course Code	Cource Name	Semester/Years	Action
1	MCA	Master of Computer Applications	3	Delete

9.4.5 Add Subjects



SUBJECT MANAGEMENT SYSTEM

Subject Code: Enter Subject Code

Subject Name: Enter Subject Name

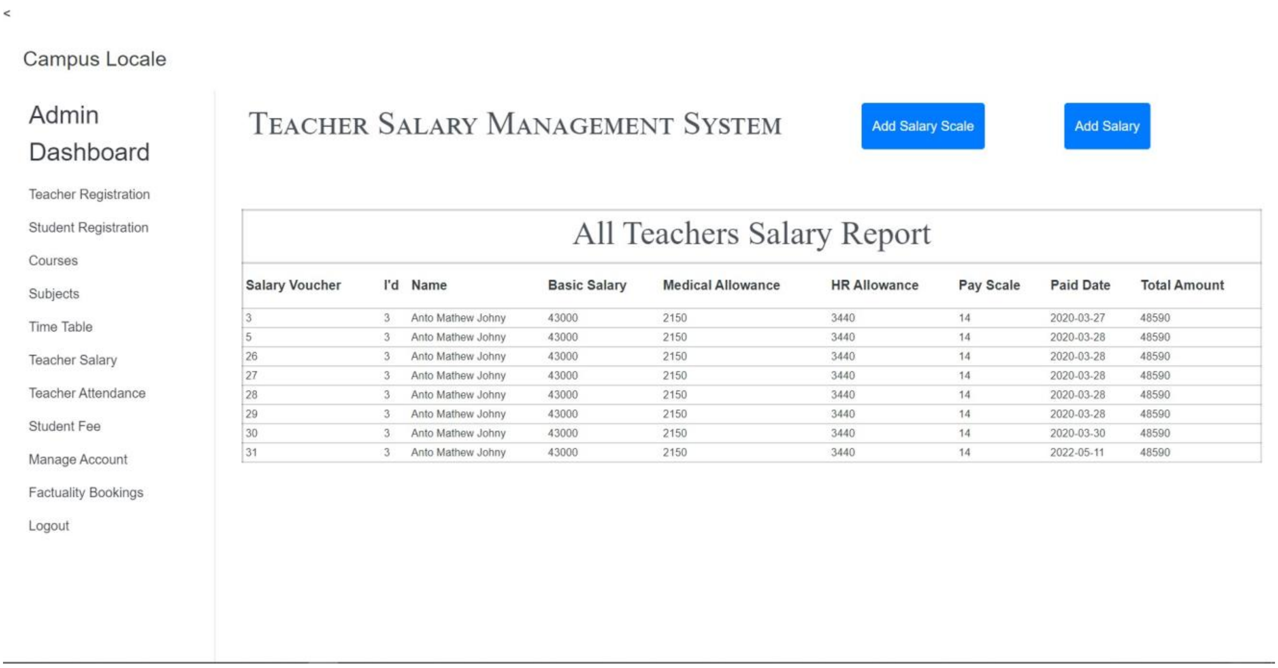
Semester: Enter Semester

Course Code: Select Course

Credit Hours: Enter Subject Credit Hours

Sr.No	Subject Code	Subject Name	Course Code	Semester	Credit Hours	Action
1	202	Application Development and Maintenance	MCA	1	10	Delete
2	203	Big Data Techonologies	MCA	1	10	Delete
3	204	Mobile Computing	MCA	1	10	Delete
4	205	System Design lab	MCA	1	14	Delete

9.4.5 Add Salary

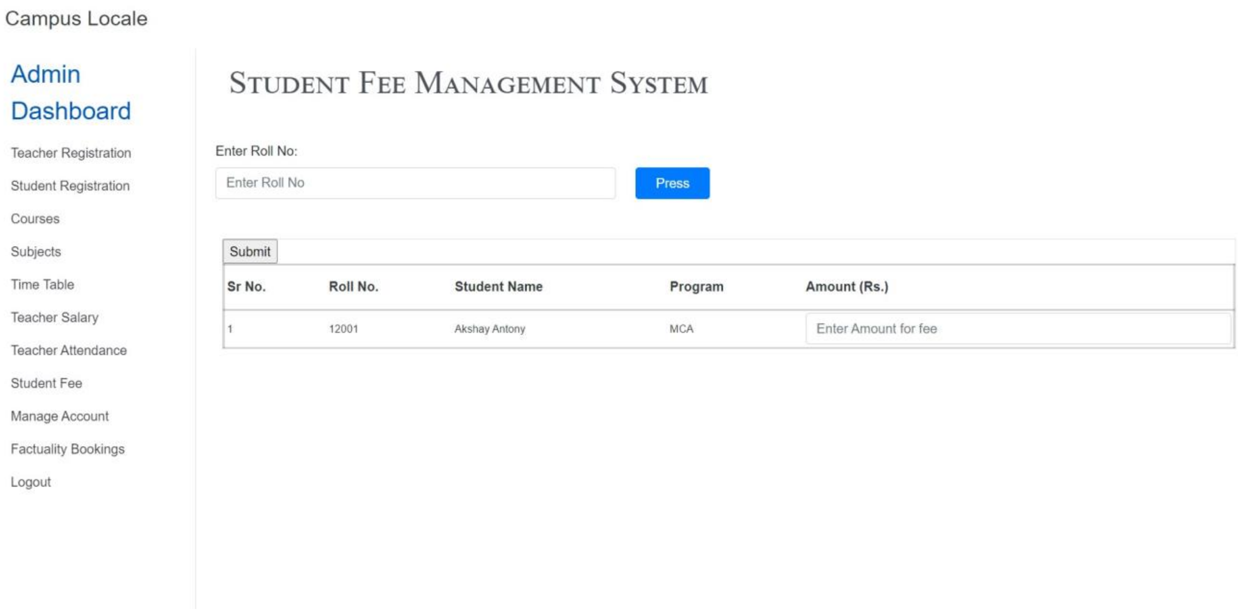


TEACHER SALARY MANAGEMENT SYSTEM

All Teachers Salary Report

Salary Voucher	I'd	Name	Basic Salary	Medical Allowance	HR Allowance	Pay Scale	Paid Date	Total Amount
3	3	Anto Mathew Johny	43000	2150	3440	14	2020-03-27	48590
5	3	Anto Mathew Johny	43000	2150	3440	14	2020-03-28	48590
26	3	Anto Mathew Johny	43000	2150	3440	14	2020-03-28	48590
27	3	Anto Mathew Johny	43000	2150	3440	14	2020-03-28	48590
28	3	Anto Mathew Johny	43000	2150	3440	14	2020-03-28	48590
29	3	Anto Mathew Johny	43000	2150	3440	14	2020-03-28	48590
30	3	Anto Mathew Johny	43000	2150	3440	14	2020-03-30	48590
31	3	Anto Mathew Johny	43000	2150	3440	14	2022-05-11	48590

9.4.6 Fees Message



STUDENT FEE MANAGEMENT SYSTEM

Enter Roll No: Enter Roll No

Sr No.	Roll No.	Student Name	Program	Amount (Rs.)
1	12001	Akshay Antony	MCA	Enter Amount for fee