**1.0 PREAMBLE**

* 1. **Introduction to the Project**

**“ONLINE PROGRAMME MANAGEMENT SYSTEM”** is a web application for a charity trust which provides free programmes which can be applied by the applicants and get admitted for the same. The main aim of the project is to manage the working of the organization. The project is being developed using Java JSP as its Front End and MySQL as its Back End.

This software mainly deals with 3 types of users which are **administrator**, **staff, students** who interacts with the system**.** This software are secured from unauthorized access because only users of the system can access the credential data.

**Admission Process**

The organization provides free programmes for a duration of 3 years and the 3 duration is assigned as a batch of the programme. The admin updates the status of each programme, applicant can apply for any programme which is already been started and can only apply for the programme in which he/she satisfies the eligibility criteria.

**Scrutiny Process**

After the submission of the application by the applicant ,the admin screens each and every application for each programmes and either selects or rejects the application based on the eligibility criteria or on the strength of the students allocated for the programmes.

**Admitted Process**

The applicants will be informed about their information and if they are admitted for the programme they will be allowed to view the session allotted for the subjects of

their programme and can view the attendance of the particular sessions and can also view the marks after attempting the exam.

The Staff’s of the organization are provided with a username and password with which they logs into the system and add their details, view the sessions allotted to them, mark the attendance for the students for the allotted sessions and enter the marks of the students for the conducted exams.

The software does not allow to enter the invalid data and if the user tries to enter the invalid data if will be reported by an error message and all the data stored inside the tables are also normalized thus reducing the chance or cause of redundancy and maintain the efficiency .

* 1. **Objectives of the Project**

**“ONLINE PROGRAMME MANAGEMENT SYSTEM”** is a web application for charity trust .

The admin mainly does the addition of different programmes , courses or subjects within each programme ,their duration, status of the courses, makes timetable for the session, assign teachers for the session etc.

The faculty can view their sessions, register their details, mark the attendances of the students and enter the marks attained by the students for the respective exams.

The students can register their details, can apply for a course, view the sessions ,can view their attendance and can also download their certificate after completion of the course they have applied.

The main objectives are:

* The complete information about the operations handled in the organization
* Details of programmes, subjects, sessions etc and maintain their record in database.
* A user friendly interface which can be operated b anyone with a minimum knowledge of the computer system.
* Eliminate the entering of invalid data.
* The administrator have the rights for overall management.
  1. **Scope of the Project**

The “**Online Programme Management System**” is a web application, which is developed with an intention to make the records stored in the database easier.

The “**Online Programme Management System**” is a time-saving and efficient project. The System can use it to efficiently store all the data in a secure database. It is less prone to errors as the program checks the data entered before saving it to database. If it finds any data to be unsatisfactory it shows a warning to the user to correct the error.

It is extremely simple to use and quite powerful at the same time. It takes the load off the staff or admin in the “**Online Programme Management System**”. The system is very flexible and changes can be made without much difficulty. The future extension in the system can be made in such a way that addition of new modules can be done without much difficulty. The reconstruction of the system will increase the flexibility of the system

Our main aim of the project is to make it easy for the students to get a clear idea about their programmes and easily able to apply for the respective programmes.The student can easily check into his attendances and marks for which he/she has been admitted. Enables the staff and administrators to search documents and records by various criteria. The goals that wish to be achieved are:

1. System manages to save all the student and staff records accurately and computerized.

2. System able to avoid the data redundancy of students marks and attendance details .

3. System administrator manages to check the request for application of various programmes by the student.

4. System able to generate certificates for the students after completion of their programmes.

5. System provides an interactive design to view particular reports and status of their application for programmes.

**2.0 SOFTWARE REQUIREMENT SPECIFICATION**

**2.1 System Study**

**2.1.1 Existing System**

The existing system uses manual method for the whole process such manual entry of details such as staffs, students,programmes,courses,timetables,attendance,marks etc. This requires a lot of hard work and time consumption to complete the task. This may include human errors. In the existing system, it is difficult to retrieve some particular information. Also, all the records are stored manually and it is a tedious task. As a result the security of these records is always a challenging task. In spite of all the efforts undertaken, the destruction of data may happen often. Hence, the computerization of the system of record maintenance is the only solution to reduce the shortcomings of the existing system.

There are many disadvantages in manual system:

* **Inaccuracies:**

Since all the data entry is done manually, inaccuracies can occur. Cause of error is more in manual system.

* **Modification:**

The transactions are executed in off-line mode, hence on-line data capture and modification is not possible.

* **Expensive:**

The maintenance of this project can be expensive.

Large number of personal hours is required for each and every part of the manual system. We have to buy lot of books, paper for keeping the information. So the existing system is very expensive.

* **Inefficiency:**

The existing system only provides text-based interface, which is not as user-friendly as Graphical user Interface. Inefficiency in the existing system is also caused by the lack of proper communication between the people of the organization.

* **Time and effort:**

Large amount of time and efforts may require. For searching a particular data, we have to search all the paper documents, its corresponding book and find the data.

* **All the searching are done manually:**

In the existing system, all the work like entering the staff details, student details, programme details etc. is done manually. This is really time consuming.

**2.1.2 Proposed System**

The proposed system is interactive, highly user friendly and designed exclusively for the “**Online Programme Management System**”. The system covers almost all the functional areas of the “**OnlineProgramme Management System**” such as details of staffs, programmes, courses,students, attendance,marks etc.

All the operations and activities related to the “**Online Programme Management System**” can be carried out efficiently. The project maintains well organized database for storing the resources that they are provided by the client. This helps us to eliminate the entering of invalid data. Most problems of manual system can be solved by this system.

The computerization of the system allows the easy maintenance of the details. Large amount of data can be stored easily. Addition and updating other changes can be done easily. The information can be retrieved with high speed and accuracy. The use of GUI oriented software makes the system user friendly. Since all work is computerized, the calculations are effortless and less time consuming. Speed, accuracy, storage capacity, versatility, automation etc. are the advantages of using a computerized system.

The main purpose behind the proposed system is to provide a comprehensive computerized system, which can capture, collate and analyze the data from these wards and evaluate the impact of the program.

The main advantages of the new system are:

* **Security:**

The software used for this “Online Programme Management System” includes the password, so the security is provided. When anyone opens the software it has the provision for entering password. We have to enter the correct password; otherwise we cannot enter into the system. Password is saved in system registry for more security.

* **User Friendly:**

This package is very user friendly because it is easy to maintain and operate. All data entry operations are simple, administrator needs only to enter data and all other operations are performed by the computer.

* **Speed and Accuracy:**

Computerization process increases the speed of all the operations. The manpower is reduced. Instead of doing all operation manually, computer will do it automatically. It also increases the accuracy of all the operations performed.

* **Efficiency and flexibility:**

The flexibility and the efficiency of all the operation in the company is increased because of the computerization. No errors are occurred compared to the manual system

## Automation:

## The proposed system automates each and every activity of the manual system and increases its throughput. Thus the response time of the system is very less and it works very fast.

**2.2 Functional Specification**

* **Login/Registration:**

In this module each user of the application registers his/her details by giving username and password with which they will be able to login to the system. Each user has his/her usertype with which each will be having different functionalities.Each user who logins to the system will be only able to access the responsibilities which they are allowed to.

* **Programme Management:**

In Programme Management the admin adds all the programmes that are made available for the students.The admin also updates the status of the programme like ongoing, completed,enrolled etc.The admin also adds various subjects under each programme which can be viewed by the student and apply for the same.

* **Staff Management:**

In this module the staff enters his/her personal details once entered they can update their details and even delete and view the same.

* **Attendance Management:**

In this module the staff add the attendance of each students corresponding to the sessions and the students can login to the system and can view their attendance for the same.

* **Session Management:**

In this module the admin assigns each session for each staff and generates a timetable for the particular programmes which can be viewed by the students and also the staff can view their sessions assigned.

* **Admission Management:**

In this module the students can login to the system and view all courses that are available to them and they can apply for any course for which they meet the criterias.They can view their status regarding their admission and if they are admitted they will be informed with the sessions and timings.

* **Result Management:**

In this module the student can login to the system and get the marks secured for the exams conducted and at the end of the course they can view the overall result for the programme they secured and they can get the consolidated mark sheet and the completion certificate.

**2.3 User Characteristics**

* **Admin**

The Admin logs into the system and has certain privileges, has full control over the system.Admin can add various programmes ,courses within each programme,status of each programme,can assign sessions for the staffs,generate timetable,view applied student’s details and approve or reject their application,verify the staffs etc.

* **Staff/Faculty**

Staff can login to the system and can add their details,view the sessions assigned to them,add the attendance of the students for the particular sessions,add the marks of the students secured for the exams etc.

* **User/Student**

Student can login to the system,can view the available courses and can apply for the same,after applying they can view their status of their admission.Once they are admitted for the programme they can view the sessions or timetable

of the programme for a specific day,can view their attendance,can view their marks and also they can generate certificate after the completion of the course.

**2.4 System Specification**

**2.4.1 Software Specification for Development, Implementation**

* + - * Development Configuration
        + Machine– Windows 7 or above /Linux distros, MySqL
      * Implementation Configuration
        + Client machine- Windows 7 or above /Linux distros
        + Server Machine- Windows 7 or above/Linux distros, WAMP or XAMP and MySqL

**2.4.2 Hardware Specification for Development, Implementation**

* + - * Development Configuration
        + Machine (*Minimum Requirement*)

Processor –Intel i3, AMD and above version.

RAM – 512MB and above.

Hardware Device – A Monitor and Keyboard with Mouse.

Hard disk – Min 1 GB.

* + - * Implementation Configuration
        + Client Machine (*Minimum Requirement*)

Hardware Device – A computer with a web browser.

RAM – 512 MB and above (Recommended 1GB).

Hard Disk – Min 60 MB

**2.4.3 About the Software/Tools.**

* **JAVA**

Java is a set [of computer software](https://en.wikipedia.org/wiki/Computer_software) and specifications developed by [Sun](https://en.wikipedia.org/wiki/Sun_Microsystems)

[Microsystems,](https://en.wikipedia.org/wiki/Sun_Microsystems) which was later acquired by the [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation), that provides a system for developing [application software](https://en.wikipedia.org/wiki/Application_software) and deploying it in a [cross-](https://en.wikipedia.org/wiki/Cross-platform) [platform](https://en.wikipedia.org/wiki/Cross-platform) computing environment. Java is used in a wide variety of [computing](https://en.wikipedia.org/wiki/Computing_platform) [platforms](https://en.wikipedia.org/wiki/Computing_platform) from [embedded devices](https://en.wikipedia.org/wiki/Embedded_device) and [mobile phones](https://en.wikipedia.org/wiki/Mobile_phone) to [enterprise](https://en.wikipedia.org/wiki/Enterprise_server) [servers](https://en.wikipedia.org/wiki/Enterprise_server) and [supercomputers.](https://en.wikipedia.org/wiki/Supercomputer) [Java applets](https://en.wikipedia.org/wiki/Java_applet), which are less common than standalone Java applications, run in secure, [sandboxed](https://en.wikipedia.org/wiki/Sandbox_(computer_security)) environments to provide many features of native applications and can be embedded in [HTML](https://en.wikipedia.org/wiki/HTML) pages.

JavaScript often abbreviated as JS, is a [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [dynamic](https://en.wikipedia.org/wiki/Dynamic_programming_language), [weakly](https://en.wikipedia.org/wiki/Weak_typing) [typed,](https://en.wikipedia.org/wiki/Weak_typing) [prototype-based,](https://en.wikipedia.org/wiki/Prototype-based_programming) [multi-paradigm,](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language) and [interpreted](https://en.wikipedia.org/wiki/Interpreted_language) [programming language.](https://en.wikipedia.org/wiki/Programming_language) Alongside [HTML](https://en.wikipedia.org/wiki/HTML) and [CSS](https://en.wikipedia.org/wiki/CSS), JavaScript is one of the three core technologies of [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web) [content production](https://en.wikipedia.org/wiki/World_Wide_Web). It is used to make webpages interactive and provide online programs, including video games

* **JSP**

JSP allows Java code and certain predefined actions to be interleaved with static web markup content, such as HTML, with the resulting page being compiled and executed on the server to deliver a document. The compiled pages, as well as any dependent Java libraries, contain Java bytecode rather than machine code. Like any other Java program, they must be executed within a Java virtual machine (JVM) that interacts with the server's host operating system to provide an abstract, platform-neutral environment.

## Mysql

MySQL is the world’s most popular open source database, enabling the cost effective delivery of reliable, high-performance, and scalable web-based, cloud and embedded database applications, including all five of the top five websites. 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 a central role in computing, as standalone utilities.

**3.0 SYSTEM MODELING**

**3.1 Uml Diagrams**

UML (Unified Modeling Language) 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. It was initially started to capture the behavior of complex software and non-software system and now it has become an OMG standard.

**3.1.1 Use Case Diagram**

The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose. Because other four diagrams (activity, sequence, collaboration and State chart) are also having the same purpose. So, we will investigate some specific purpose which will distinguish it from other four diagrams. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So, when a system is analyzed to gather its functionalities use cases are prepared and actors are identified

**users**

includes

**admin**

**staff**

**student**

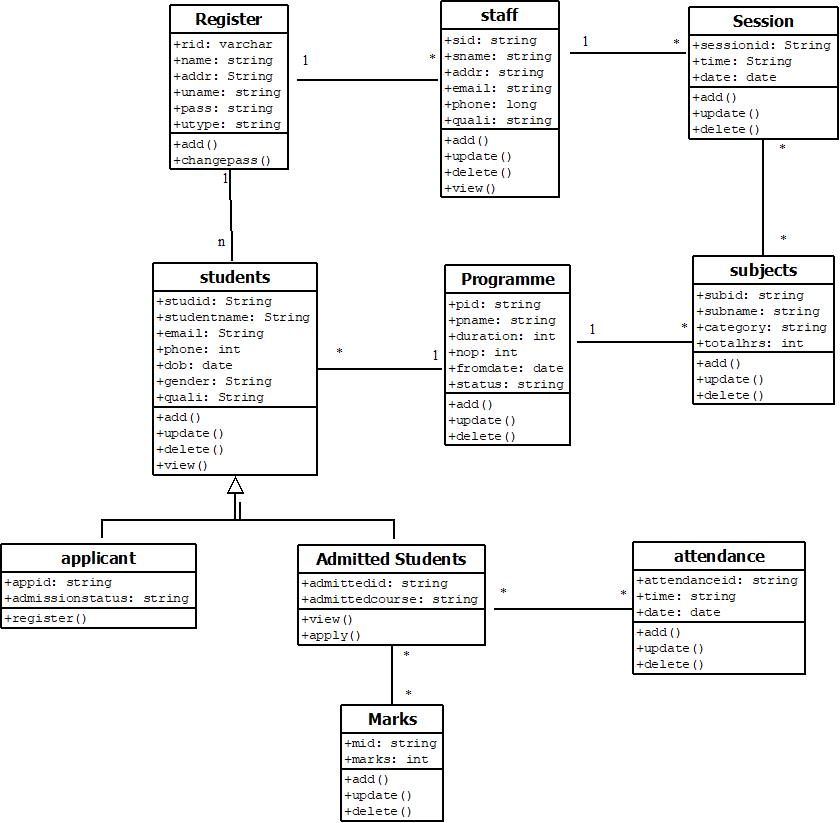
**admitted**

**applicant**

**3.1.2 Class Diagram**

Class diagrams are arguably the most used UML diagram type. It is the main building block of any object-oriented solution. It shows the classes in a system, attributes and operations of each class and the relationship between each class. In most modeling tools, a class has three parts, name at the top, attributes in the middle and operations or methods at the bottom. In large systems with many related classes, classes are grouped together to create class diagrams. Different relationships between classes are shown by different types of arrows.

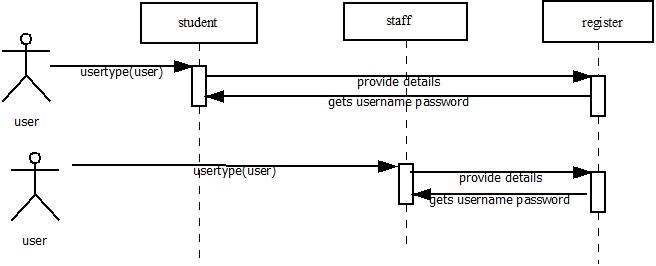
Class diagram shows the information of the classes; the above diagram depicts the operations and the attributes of the classes and the types of attributes which are used in the class diagram. In the figure there are 10 classes.



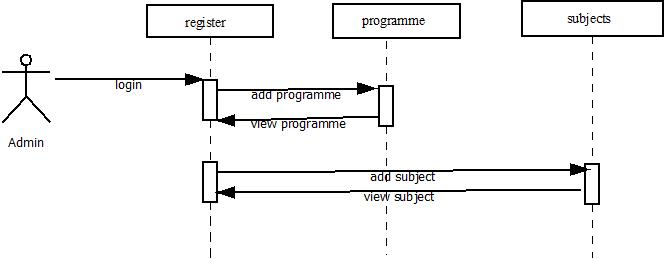
**3.1.3 Sequence Diagram**

Sequence diagram is the most common kind of interaction diagram, which focuses on the message interchange between several lifelines. Sequence diagram describes an interaction by focusing on the sequence of messages that are exchanged, along with their corresponding occurrence specifications on the lifelines. The following nodes and edges are typically drawn in an Uml sequence diagram: lifeline, execution, specification, message, fragment, interaction, state, invariant, continuation, destruction occurrence.

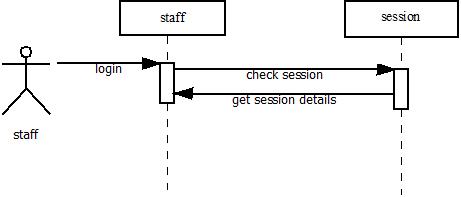
* **Register**

****

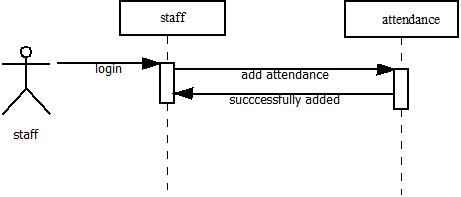
* **Add programmes**



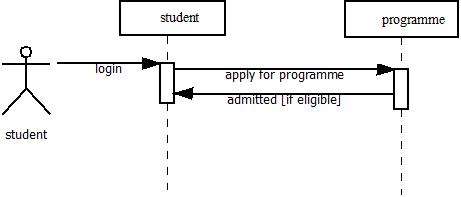
* **View sessions**

****

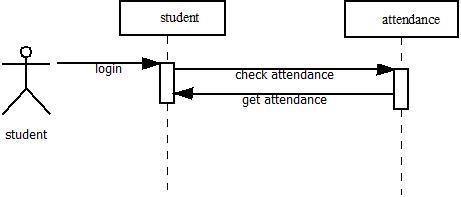
* **Add attendance**

****

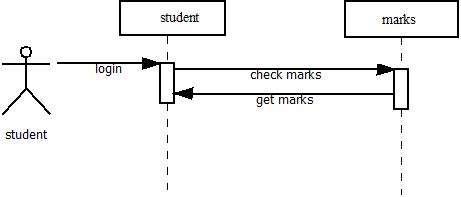
* **Apply Programme**

****

* **View attendance**

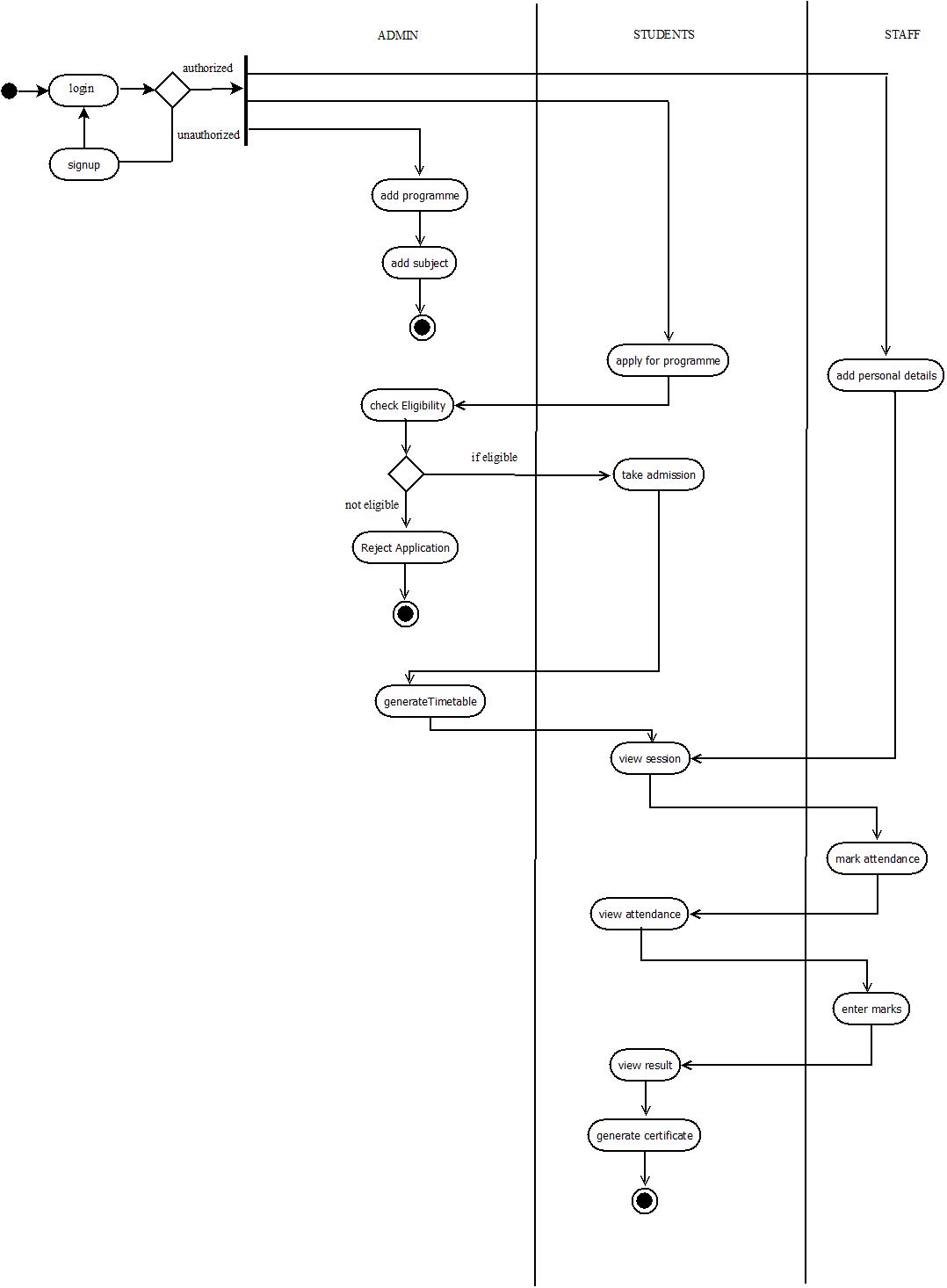
****

* **View session**

****

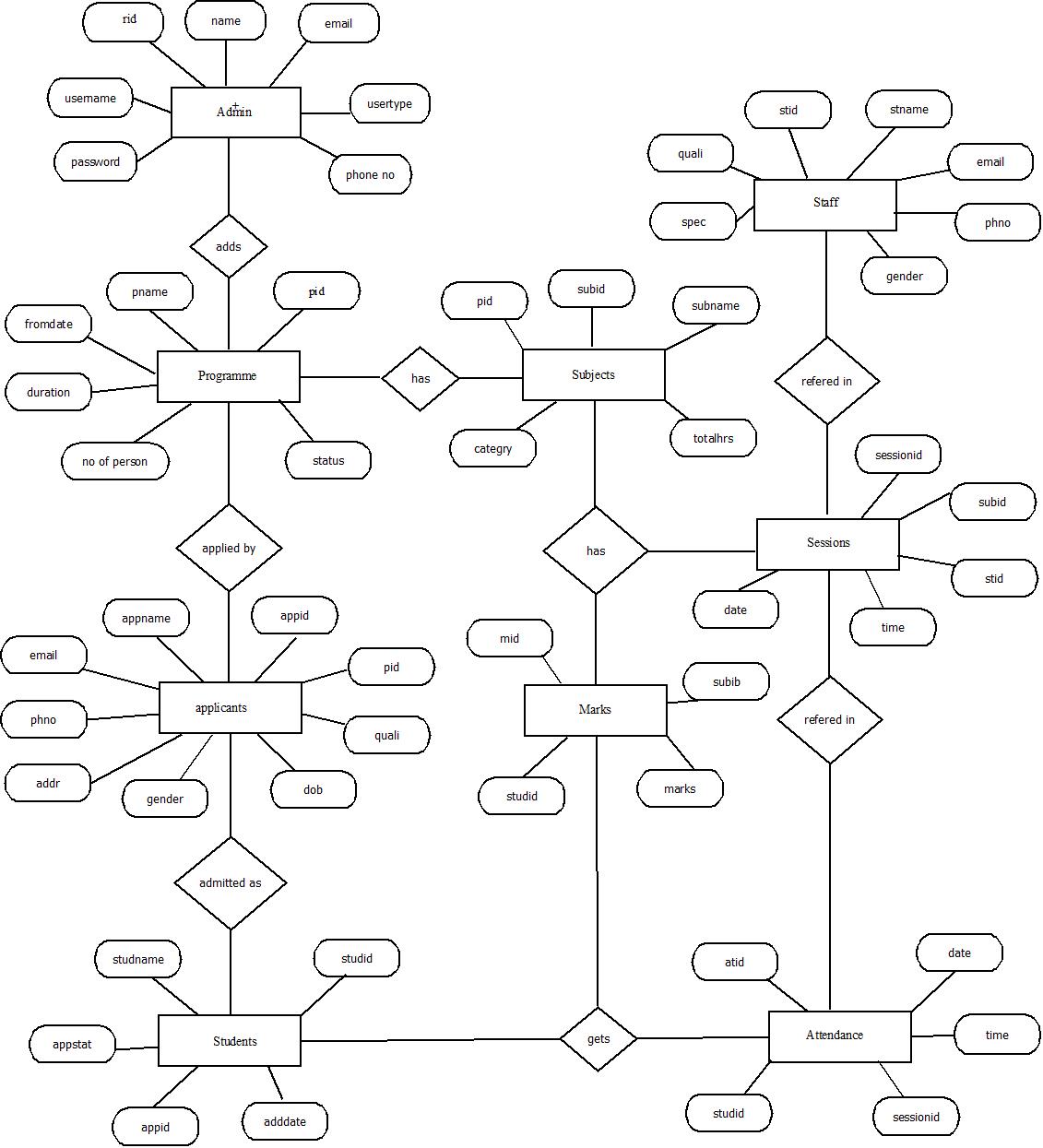
**3.1.4 Activity Diagram**

The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another, but activity diagram is used to show message flow from one activity to another. Activity diagrams are not only used for visualizing 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 activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is some time considered as the flow chart. Although the diagrams look like a flow chart, but it is not. It shows different flow like parallel, branched, concurrent and single.

****

**3.2 ER Diagram**

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a Data base. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases. At first glance an entity relationship diagram looks very much like a flowchart. It is the specialized symbols, and the meanings of those symbols that make it unique.

****

**3.3Table Description**

* **Table Name:Register**

**Primary key:rid**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| Rid | Varchar | 10 | Registration id |
| Name | Varchar | 20 | Name |
| Email | Varchar | 20 | Email id |
| Username | Varchar | 20 | Username |
| Password | Varchar | 20 | Password |
| Usertype | Varchar | 20 | Usertype |

* **Table Name:Programme**

**Primary key:pid**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| Pid | Varchar | 10 | Programe id |
| Pname | Varchar | 20 | Programme Name |
| Fdate | Date | 20 | From date |
| Dur | Varchar | 20 | Duration |
| Nop | Int | 20 | No of person |
| Status | Varchar | 20 | Status of Programme |

* **Table Name:subject**

**Primary key:subid**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| Subid | Varchar | 10 | Subject id |
| Subname | Varchar | 20 | Subject Name |
| Thrs | Int | 10 | Total Hours |
| Ctgry | Varchar | 20 | Category |
| Pid | Varchar | 20 | Programme id |
| Substat | Varchar | 20 | Subject status |

* **Table Name:staff**

**Primary key:stid**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| Stid | Varchar | 10 | Staff id |
| Stname | Varchar | 20 | Staff Name |
| Email | Varchar | 20 | Email Address |
| Phone | Varchar | 20 | Phone No |
| Gender | Varchar | 20 | Gender |
| Spec | Varchar | 20 | Specialization |
| Quail | Varchar | 20 | Qualification |

* **Table Name:appliedStudents**

**Primary key:appid**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| Appid | Varchar | 10 | Applicant id |
| Appname | Varchar | 20 | Applicant Name |
| Email | Varchar | 20 | Email Address |
| Phone | Varchar | 20 | Phone No |
| Gender | Varchar | 20 | Gender |
| Addr | Varchar | 20 | Address |
| Quail | Varchar | 20 | Qualification |
| Dob | Date | 10 | Date of Brith |
| Pid | Varchar | 20 | Programme id |

* **Table Name:admittedstudents**

**Primary key:studid**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| Studid | Varchar | 10 | Student id |
| Studname | Varchar | 20 | Student Name |
| Appstatus | Varchar | 20 | Application status |
| Appid | Varchar | 20 | Application id |
| Admsdate | Date | 20 | Admission Date |

* **Table Name:Marks**

**Primary key:mid**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| Mid | Varchar | 10 | Mark id |
| Subid | Varchar | 20 | Subject id |
| Studid | Varchar | 20 | Student id |
| Marks | Int | 20 | Marks Secured |

* **Table Name:Attendance**

**Primary key:atid**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| Atid | Varchar | 10 | Attendance id |
| Studid | Varchar | 20 | Student Id |
| Sessionid | Varchar | 20 | Session id |
| Time | Varchar | 20 | Time |
| Date | Date | 20 | Date |

* **Table Name:session**

**Primary key:sessionid**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| Sessionid | Varchar | 10 | Session Id |
| Subid | Varchar | 20 | Subject id |
| Stid | Varchar | 20 | Staff Id |
| Time | Varchar | 20 | Time |
| Date | Date | 20 | Date |

**4.0 TESTING**

**4.1 Introduction**

Software Testing is the process of executing a program or system with the intent of finding errors. Testing involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. The scope of software testing includes examination of code as well as execution of that code in various environments and conditions as well as examining the quality aspects of code: does it do what it is supposed to do and do what it needs to do. Testing helps not only to uncover errors introduced during coding, but also locates errors committed during the previous phases. Thus the aim of testing is to uncover requirements, design or coding errors in the program.

**Unit Testing**

A unit is the smallest testable part of an application. Unit testing is a method of testing that verifies the individual units of source code are working properly. Rather than initially testing a program as a whole, testing is first focused on the smaller building blocks of the program. Unit testing eases the task of debugging and provide parallelism to program testing process by giving the opportunity to test multiple modules simultaneously. In this system the validity of fields in which data entered in each form and web form are checked. If the entered data are valid, then only further processing will take place.

**Integration Testing**

Data can be lost across an interface; one module can have an adverse effect on the other sub functions, when combined may not produce the desired functions. Integrated testing is the systematic testing to uncover the errors within the interface. This testing is done with simple data and the developed system has run successfully with this simple data.

The need for integrated testing is to find the overall system performance. While developing the system, each module is developed individually and integrated with present system. Modules are integrated by adding the module as a reference in other modules.

**4.2 Test Cases**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Data** | **Cases** | **Expected result** | **Actual result** | **Remark** |
| 1. | Login to system | User enter  \* Invalid username password  \*Valid Password | Error Message: Invalid password  ----- | Error Message: Invalid password  ----- | As expected |
| 2. | Phone number | \*Enter phone number more than 10 digits  \*Valid phone number | Error Message: Incorrect format  ---- | Error Message: Incorrect format  ---- | As expected. |
| 3. | Confirm password and password | \*user enter different value  User enter same value | Error Message: Confirm password and password not same  ----- | Error Message: Confirm password and password not same  ----- | As expected |
| 4 | Email | \*Enter invalid email number for registration  \*Valid email number | Error Message: Incorrect format  ---- | Error Message: Incorrect format  ---- | As expected. |
| 5 | Date selection for session allocation | \*Select already selected date  \*Select new date | Error Message: Already alloted fo this date  ----- | Error Message: Already alloted fo this date  ---- | As expected. |

**5.0 IMPLEMENTATION DETAILS**

**5.1 Introduction**

Implementation is the stage in the project where the theoretical design is turned into a working system. Implementation is the final and important phase. The most critical stage for achieving a successful new system and forgiving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and only if it is found to working according to the specification. This method also offers the high security since it is implemented after identifying and handling all types of transactions while using the new system. Implementation phase include the training that should be provided for the chosen staff.

* 1. **Installation procedure**

The software can be installed in the following simple steps. In implementing machine

* Install mysql/Heidi sql/php My admin
* Install or Enable IIS Manager

**5.3 Implementation plan**

In a direct cut over conversion, the old system is discarded and the new system takes over all at once, it is essentially turning the old system off and turning the new system on. This approach can be the least expensive of the different methods and can occur in the quickest time. A direct cut over conversion may be the only option if the old and new systems cannot co-existing any form. The greatest risk is the impact that errors and failures would have on the organization. The timing of this type of conversion is a key element of its success. The riskiest strategy for new systems installation, the direct cut over conversion can be low cost and the benefits of the new system can be realized without delay.

**6.0 CONCLUSION**

Online Programme Management System is a computerized management system. This management system has been developed to form whole management system including Students,Admin and Staffs. This system will help the students to apply for programmes. This project is web based software that helps in storing, updating and retrieving information through many user-friendly menu driven modules.

In our existing systems all works are done manually. When a students wants to apply for a programme then he/she has to contact the institution directly or through phone and has to fill the application form manually and apply for a programme. And also all other processes done in the organization were done manually. This takes time and effort.

In this computerized system, the above drawbacks are avoided.Admin does all his prcesses using this system such as adding programmes,adding subjects,adding staff,alloting sessions for staffs,preparation of timetable etc.Students can now log on to the system and apply for the programme they want ,can check the status of their application and if admitted for the programme they can view attendance for a particular session,view marks,view timetable and after completion of the programme they can generate certificate.

**7.0 APPENDIX**

**7.1 APPENDIX A**

**Sample Source code/Pseudo code**

**Home Page**

<%@page import="DB.DB"%>

<%@page import="java.sql.\*"%>

<%@page import="java.text.SimpleDateFormat"%>

<%@page import="java.util.Date"%>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="description" content="">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- The above 4 meta tags \*Must\* come first in the head; any other head content must come \*after\* these tags -->

<!-- Title -->

<title>Clever - Education &amp; Courses Template | Home</title>

<!-- Favicon -->

<link rel="icon" href="img/core-img/favicon.ico">

<!-- Stylesheet -->

<link rel="stylesheet" href="style.css">

</head>

<body>

<!-- Preloader -->

<div id="preloader">

<div class="spinner"></div>

</div>

<!-- ##### Header Area Start ##### -->

<header class="header-area">

<!-- Top Header Area -->

<div class="top-header-area d-flex justify-content-between align-items-center">

<!-- Contact Info -->

<div class="contact-info">

<a href="#"><span>Phone:</span> +44 300 303 0266</a>

<a href="#"><span>Email:</span> info@clever.com</a>

</div>

<!-- Follow Us -->

</div>

<!-- Navbar Area -->

<div class="clever-main-menu">

<div class="classy-nav-container breakpoint-off">

<!-- Menu -->

<nav class="classy-navbar justify-content-between" id="cleverNav">

<!-- Logo -->

<a class="nav-brand" href="index.jsp"><img src="img/core-img/logo.png" alt=""></a>

<!-- Navbar Toggler -->

<div class="classy-navbar-toggler">

<span class="navbarToggler"><span></span><span></span><span></span></span>

</div>

<!-- Menu -->

<div class="classy-menu">

<!-- Close Button -->

<div class="classycloseIcon">

<div class="cross-wrap"><span class="top"></span><span class="bottom"></span></div>

</div>

<!-- Nav Start -->

<div class="classynav">

<ul>

<li><a href="index.jsp">Home</a></li>

<li><a href="courses.jsp">Courses</a></li>

<li><a href="instructors.html">Instructors</a></li>

<li><a href="blog.html">Blog</a></li>

<li><a href="contact.html">Contact</a></li>

</ul>

<!-- Register / Login -->

<div class="register-login-area">

<a href="register.jsp" class="btn">Sign Up</a>

<a href="login.jsp" class="btn active">Login</a>

</div>

</div>

<!-- Nav End -->

</div>

</nav>

</div>

</div>

</header>

<!-- ##### Header Area End ##### -->

<!-- ##### Footer Area Start ##### -->

<footer class="footer-area">

<!-- Top Footer Area -->

<div class="top-footer-area">

<div class="container">

<div class="row">

<div class="col-12">

<!-- Footer Logo -->

<div class="footer-logo">

<a href="index.jsp"><img src="img/core-img/logo2.png" alt=""></a>

</div>

<!-- Copywrite -->

<p><a href="#"><!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. -->

Copyright &copy;<script>document.write(new Date().getFullYear());</script> All rights reserved | This template is made with <i class="fa fa-heart-o" aria-hidden="true"></i> by <a href="https://colorlib.com" target="\_blank">Colorlib</a>

<!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. --></p>

</div>

</div>

</div>

</div>

<!-- Bottom Footer Area -->

<div class="bottom-footer-area d-flex justify-content-between align-items-center">

<!-- Contact Info -->

<div class="contact-info">

<a href="#"><span>Phone:</span> +44 300 303 0266</a>

<a href="#"><span>Email:</span> info@clever.com</a>

</div>

<!-- Follow Us -->

</div>

</footer>

<!-- ##### Footer Area End ##### -->

<!-- ##### All Javascript Script ##### -->

<!-- jQuery-2.2.4 js -->

<script src="js/jquery/jquery-2.2.4.min.js"></script>

<!-- Popper js -->

<script src="js/bootstrap/popper.min.js"></script>

<!-- Bootstrap js -->

<script src="js/bootstrap/bootstrap.min.js"></script>

<!-- All Plugins js -->

<script src="js/plugins/plugins.js"></script>

<!-- Active js -->

<script src="js/active.js"></script>

</body>

</html>

**Login page**

<%--

Document : insreg

Created on : Dec 27, 2019, 9:41:58 PM

Author : Sreevardhan

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@page import="DB.DB" %>

<%@ page import ="java.sql.\*" %>

<%

try{

String username = request.getParameter("username");

String password = request.getParameter("pass");

//Class.forName("com.mysql.jdbc.Driver"); // MySQL database connection

//Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/progammedb?" + "user=root&password=");

//PreparedStatement pst = conn.prepareStatement("Select \* from register where uname=? and pass=?");

//pst.setString(1, username);

//pst.setString(2, password);

ResultSet rs = DB.executequery("select \* from register1 where uname='"+username+"'and pass='"+password+"'");

if(rs.next())

{

if(rs.getString("utype").equals("admin"))

{

session.setAttribute("rid", rs.getString("rid"));

session.setAttribute("name", rs.getString("name"));

response.sendRedirect("adminlogin.jsp");

}

else if(rs.getString("utype").equals("student"))

{

session.setAttribute("rid", rs.getString("rid"));

session.setAttribute("name", rs.getString("name"));

response.sendRedirect("userlogin.jsp");

}

else if(rs.getString("utype").equals("astudent"))

{

session.setAttribute("rid", rs.getString("rid"));

session.setAttribute("name", rs.getString("name"));

response.sendRedirect("auserlogin.jsp");

}

else if(rs.getString("utype").equals("staff"))

{

session.setAttribute("rid", rs.getString("rid"));

session.setAttribute("name", rs.getString("name"));

response.sendRedirect("stafflogin.jsp");

}

}

else

{

out.println("<script type=\"text/javascript\">");

out.println("alert('Incorrect Username or Password');");

out.println("location='login.jsp';");

out.println("</script>");

}

}

catch(Exception e){

out.println("Something went wrong !! Please try again");

}

%>

**Database connection code:**

package DB;

import java.sql.\*;

public class DB {

static Connection con=null;

static Statement stmt=null;

static ResultSet rs=null;

public static Connection getConnection(){

try {

Class.forName("com.mysql.jdbc.Driver");

con= DriverManager.getConnection("jdbc:mysql://localhost:3306/progammedb","root","");

}catch(Exception e){

System.out.println("Exception" + e);

}

return con;

}

public static boolean executeUpdate(String query){

boolean f=false;

try {

con=getConnection();

stmt=con.createStatement();

int b=stmt.executeUpdate(query);

if(b>0){

f=true;

}

else{

f=false;

}

}

catch(Exception e)

{

System.out.println("Exception" + e);

}

return f;

}

public static ResultSet executequery(String sql)

{

try

{

con=getConnection();

stmt=con.createStatement();

rs=stmt.executeQuery(sql);

return rs;

}

catch(Exception e)

{

System.out.println("Exception" + e);

return null;

}

}

public static void close()

{

try{

con.close();

stmt.close();

rs.close();

}

catch(Exception e){

System.out.println("Exception" + e);

}

}

}

**Programmme adding page**

<%@page import="java.text.SimpleDateFormat"%>

<%@page import="java.util.Date"%>

<!DOCTYPE html>

<html lang="en">

<head>

</head>

<body>

<script type="text/javascript" src="http://code.jquery.com/jquery-2.1.4.min.js"></script>

<script src="//cdn.jsdelivr.net/webshim/1.14.5/polyfiller.js"></script>

<script>

webshims.setOptions('forms-ext', {types: 'date'});

webshims.polyfill('forms forms-ext');

$.webshims.formcfg = {

en: {

dFormat: '-',

dateSigns: '-',

patterns: {

d: "yy-mm-dd"

}

}

};

</script>

<!-- Preloader -->

<div id="preloader">

<div class="spinner"></div>

</div>

<!-- ##### Header Area Start ##### -->

<header class="header-area">

<!-- Top Header Area -->

<div class="top-header-area d-flex justify-content-between align-items-center">

<!-- Contact Info -->

<div class="contact-info">

<a href="#"><span>Phone:</span> +44 300 303 0266</a>

<a href="#"><span>Email:</span> info@clever.com</a>

</div>

<!-- Follow Us -->

</div>

<!-- Navbar Area -->

<div class="clever-main-menu">

<div class="classy-nav-container breakpoint-off">

<!-- Menu -->

<nav class="classy-navbar justify-content-between" id="cleverNav">

<!-- Logo -->

<a class="nav-brand" href="index.jsp"><img src="img/core-img/logo.png" alt=""></a>

<!-- Navbar Toggler -->

<div class="classy-navbar-toggler">

<span class="navbarToggler"><span></span><span></span><span></span></span>

</div>

<!-- Menu -->

<div class="classy-menu">

<!-- Close Button -->

<div class="classycloseIcon">

<div class="cross-wrap"><span class="top"></span><span class="bottom"></span></div>

</div>

<!-- Nav Start -->

<div class="classynav">

<div class="classynav">

<ul>

<li><a href="adminlogin.jsp">Home</a></li>

<li><a href="#">Programmes</a>

<ul class="dropdown">

<li><a href="prgrmadd.jsp">Add Programmes</a></li>

<li><a href="courseadd.jsp">Add Courses</a></li>

</ul>

</li>

<li><a href="#">Sessions</a>

<ul class="dropdown">

<li><a href="allotsession.jsp">Allot Sessions</a></li>

</ul>

</li>

<li><a href="#">Students</a>

<ul class="dropdown">

<li><a href="admitstud.jsp">Admit Students</a></li>

<li><a href="currentstudents.jsp">Admitted\_Students</a></li>

<li><a href="#">View Attendance</a></li>

<li><a href="aviewmark.jsp">View Marks</a></li>

</ul>

</li>

<li><a href="#">View</a>

<ul class="dropdown">

<li><a href="adminstaff.jsp">View Staff</a></li>

<li><a href="#">View Programme</a>

<ul class="dropdown">

<li><a href="viewprog.jsp">Current</a></li>

<li><a href="viewpastprog.jsp">Past</a></li>

</ul>

</li>

</ul>

</ul>

<!-- Search Button -->

<!-- Register / Login -->

<div class="login-state d-flex align-items-center">

<div class="user-name mr-30">

<div class="dropdown">

<a class="dropdown-toggle" href="#" role="button" id="userName" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false"><%=session.getAttribute("name")%></a>

<div class="dropdown-menu dropdown-menu-right" aria-labelledby="userName">

<a class="dropdown-item" href="index.jsp">Logout</a>

</div>

</div>

</div>

<div class="userthumb">

<img src="img/bg-img/t1.png" alt="">

</div>

</div>

</div>

<!-- Nav End -->

</div>

</nav>

</div>

</div>

</header>

<!-- ##### Header Area End ##### -->

<!-- ##### Hero Area Start ##### -->

<section class="blog-area section-padding-100-0">

<div class="container h-100">

<div class="row h-100 align-items-center">

<div class="col-12">

<center>

<div class="wrap-login100">

<div class="login100-form-title">

<span class="login100-form-title-1">

PROGRAMME DETAILS

</span>

</div>

<form class="login100-form validate-form" action="inprgrm.jsp" method="post">

<div class="wrap-input100 validate-input m-b-26">

<span class="label-input100">Name</span>

<input class="input100" type="text" id="name" name="name" required>

<span class="focus-input100"></span>

</div>

<div class="wrap-input100 validate-input m-b-26">

<span class="label-input100">From Date</span>

<input class="input100" type="date" id="frmdate" name="frmdate" required>

<span class="focus-input100"></span>

</div>

<div class="wrap-input100 validate-input m-b-26">

<span class="label-input100">Duration</span>

<input class="input100" type="text" id="dur" name="dur" required>

<span class="focus-input100"></span>

</div>

<div class="wrap-input100 validate-input m-b-26">

<span class="label-input100">Max no of Students</span>

<input class="input100" type="text" id="mnos" name="mnos" required>

<span class="focus-input100"></span>

</div>

<div class="wrap-input100 validate-input m-b-18" >

<span class="label-input100">Status</span>

<select class="input100" name="type" required>

<option class="input100" value="Ongoing">Ongoing</option>

<option class="input100" value="Enrolled">Enrolled</option>

<option class="input100" value="Completed">Completed</option>

<option class="input100" value="Discontinued">Discontinued</option>

</select >

<span class="focus-input100"></span>

</div>

<div class="wrap-input100 validate-input m-b-26">

<span class="label-input100">Registration End\_Date</span>

<input class="input100" type="date" id="frmdate" name="regdate" required>

<span class="focus-input100"></span>

</div>

<div class="container-login100-form-btn">

&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp;

<input type="submit" class="login100-form-btn" value="ADD">

<!--&nbsp&nbsp<input type="submit" class="login100-form-btn" value="UPDATE/DELETE" onClick="checkvali();">-->

</div>

</form>

</div>

</center>

</div>

</div>

</div>

</section>

<!-- ##### Hero Area End ##### -->

<!-- ##### Cool Facts Area Start ##### -->

<section class="cool-facts-area section-padding-100-0">

</section>

<!-- ##### Cool Facts Area End ##### -->

<!-- ##### Popular Courses Start ##### -->

<section class="cool-facts-area section-padding-100-0">

</section>

<!-- ##### Popular Courses End ##### -->

<!-- ##### Best Tutors Start ##### -->

<section class="cool-facts-area section-padding-100-0">

</section>

<!-- ##### Best Tutors End ##### -->

<!-- ##### Register Now Start ##### -->

<section class="cool-facts-area section-padding-100-0">

</section>

<!-- ##### Upcoming Events End ##### -->

<!-- ##### Blog Area Start ##### -->

<section class="cool-facts-area section-padding-100-0">

</section>

<!-- ##### Blog Area End ##### -->

<!-- ##### Footer Area Start ##### -->

<footer class="footer-area">

<!-- Top Footer Area -->

<div class="top-footer-area">

<div class="container">

<div class="row">

<div class="col-12">

<!-- Footer Logo -->

<div class="footer-logo">

<a href="index.jsp"><img src="img/core-img/logo2.png" alt=""></a>

</div>

<!-- Copywrite -->

<p><a href="#"><!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. -->

Copyright &copy;<script>document.write(new Date().getFullYear());</script> All rights reserved | This template is made with <i class="fa fa-heart-o" aria-hidden="true"></i> by <a href="https://colorlib.com" target="\_blank">Colorlib</a>

<!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. --></p>

</div>

</div>

</div>

</div>

<!-- Bottom Footer Area -->

<div class="bottom-footer-area d-flex justify-content-between align-items-center">

<!-- Contact Info -->

<div class="contact-info">

<a href="#"><span>Phone:</span> +44 300 303 0266</a>

<a href="#"><span>Email:</span> info@clever.com</a>

</div>

<!-- Follow Us -->

</div>

</footer>

<!-- ##### Footer Area End ##### -->

<!-- ##### All Javascript Script ##### -->

<!-- jQuery-2.2.4 js -->

<script src="js/jquery/jquery-2.2.4.min.js"></script>

<!-- Popper js -->

<script src="js/bootstrap/popper.min.js"></script>

<!-- Bootstrap js -->

<script src="js/bootstrap/bootstrap.min.js"></script>

<!-- All Plugins js -->

<script src="js/plugins/plugins.js"></script>

<!-- Active js -->

<script src="js/active.js"></script>

</body>

</html>

**Apply programme page**

<%@page import="java.text.SimpleDateFormat"%>

<%@page import="java.util.Date"%>

<%@page import="DB.DB"%>

<%@page import="java.sql.\*"%>

<!DOCTYPE html>

<html lang="en">

<head>

<script type="text/javascript">

function checkvali()

{

var ph=document.getElementById("ph").value;

if(isNaN(ph))

{

alert("Phone Number should only contain digits!!");

return false;

}

else if(ph.length!==10)

{

alert("Phone Number should be of 10 digits!!");

return false;

}

}

</script>

</head>

<body>

<!-- Preloader -->

<div id="preloader">

<div class="spinner"></div>

</div>

<!-- ##### Header Area Start ##### -->

<header class="header-area">

<!-- Top Header Area -->

<div class="top-header-area d-flex justify-content-between align-items-center">

<!-- Contact Info -->

<div class="contact-info">

<a href="#"><span>Phone:</span> +44 300 303 0266</a>

<a href="#"><span>Email:</span> info@clever.com</a>

</div>

<!-- Follow Us -->

</div>

<!-- Navbar Area -->

<div class="clever-main-menu">

<div class="classy-nav-container breakpoint-off">

<!-- Menu -->

<nav class="classy-navbar justify-content-between" id="cleverNav">

<!-- Logo -->

<a class="nav-brand" href="index.jsp"><img src="img/core-img/logo.png" alt=""></a>

<!-- Navbar Toggler -->

<div class="classy-navbar-toggler">

<span class="navbarToggler"><span></span><span></span><span></span></span>

</div>

<!-- Menu -->

<div class="classy-menu">

<!-- Close Button -->

<div class="classycloseIcon">

<div class="cross-wrap"><span class="top"></span><span class="bottom"></span></div>

</div>

<!-- Nav Start -->

<div class="classynav">

<ul>

<li><a href="userlogin.jsp">Home</a></li>

<li><a href="ucourses.jsp">View Programmes</a></li>

<%

ResultSet rs11 = DB.executequery("select \* from applicant where rid="+session.getAttribute("rid"));

if(rs11.next())

{%>

<li><a href="viewdetails.jsp">View Status</a></li>

<%}

else

{ Date d = new Date();

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

String date=sdf.format(d);

ResultSet rs1 = DB.executequery("select \* from programme");

if (rs1.next())

{

if(date.compareTo(rs1.getString("regdate"))==-1)

{

%>

<li><a href="iapply.jsp">Apply for Programme</a></li>

<%}}}%>

</ul>

<!-- Search Button -->

<!-- Register / Login -->

<div class="login-state d-flex align-items-center">

<div class="user-name mr-30">

<div class="dropdown">

<a class="dropdown-toggle" href="#" role="button" id="userName" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false"><%=session.getAttribute("name")%></a>

<div class="dropdown-menu dropdown-menu-right" aria-labelledby="userName">

<a class="dropdown-item" href="#">Profile</a>

<a class="dropdown-item" href="index.jsp">Logout</a>

</div>

</div>

</div>

<div class="userthumb">

<img src="img/bg-img/t1.png" alt="">

</div>

</div>

</div>

<!-- Nav End -->

</div>

</nav>

</div>

</div>

</header>

<!-- ##### Header Area End ##### -->

<!-- ##### Hero Area Start ##### -->

<section class="blog-area section-padding-100-0">

<div class="container h-100">

<div class="row h-100 align-items-center">

<div class="col-12">

<center>

<div class="wrap-login100">

<div class="login100-form-title">

<span class="login100-form-title-1">

APPLY HERE

</span>

</div>

<form class="login100-form validate-form" onsubmit="return checkvali()" action="inapply.jsp" method="post">

<div class="wrap-input100 validate-input m-b-26">

<span class="label-input100">Applicant Name</span>

<input class="input100" type="text" id="name" name="name" required>

<span class="focus-input100"></span>

</div>

<div class="wrap-input100 validate-input m-b-26">

<span class="label-input100">Address</span>

<input class="input100" type="text" id="addr" name="addr" required>

<span class="focus-input100"></span>

</div>

<div class="wrap-input100 validate-input m-b-26">

<span class="label-input100">Email</span>

<input class="input100" type="text" id="em" name="email" required>

<span class="focus-input100"></span>

</div>

<div class="wrap-input100 validate-input m-b-26">

<span class="label-input100">Phone no</span>

<input class="input100" type="text" id="ph" name="phno" required>

<span class="focus-input100"></span>

</div>

<div class="wrap-input100 validate-input m-b-26">

<label class="label-input100">Gender</label>

<input type="radio" id="gen" name="gender" value="male" checked="1" required>Male

&nbsp&nbsp&nbsp<input type="radio" id="gen" name="gender" value="female" required>Female

</div>

<div class="wrap-input100 validate-input m-b-26">

<span class="label-input100">DOB</span>

<input class="input100" type="date" id="dob" name="dob" required>

<span class="focus-input100"></span>

</div>

<div class="wrap-input100 validate-input m-b-18" >

<span class="label-input100">Qualification</span>

<select class="input100" name="type" required>

<option class="input100" value="pg">PostGraduation</option>

<option class="input100" value="ug">Graduation</option>

<option class="input100" value="twelve">XII</option>

<option class="input100" value="ten">X</option>

</select >

<span class="focus-input100"></span>

</div>

<div class="wrap-input100 validate-input m-b-18" >

<span class="label-input100">Programme</span>

<select class="input100" name="prgrm" required>

<%

ResultSet rs = DB.executequery("select \* from programme where pid="+request.getParameter("id"));

while (rs.next()) {

%>

<option class="input100" value="<%= rs.getString("pname")%>"><%= rs.getString("pname")%></option>

<%}%>

</select >

<span class="focus-input100"></span>

</div>

<div class="container-login100-form-btn">

&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp;

<input type="submit" class="login100-form-btn" value="APPLY" onClick="checkvali();">

<!--&nbsp&nbsp<input type="submit" class="login100-form-btn" value="UPDATE/DELETE" onClick="checkvali();">-->

</div>

</form>

</div>

</center>

</div>

</div>

</div>

</section>

<!-- ##### Hero Area End ##### -->

<!-- ##### Cool Facts Area Start ##### -->

<section class="cool-facts-area section-padding-100-0">

</section>

<!-- ##### Cool Facts Area End ##### -->

<!-- ##### Popular Courses Start ##### -->

<section class="cool-facts-area section-padding-100-0">

</section>

<!-- ##### Popular Courses End ##### -->

<!-- ##### Best Tutors Start ##### -->

<section class="cool-facts-area section-padding-100-0">

</section>

<!-- ##### Best Tutors End ##### -->

<!-- ##### Register Now Start ##### -->

<section class="cool-facts-area section-padding-100-0">

</section>

<!-- ##### Upcoming Events End ##### -->

<!-- ##### Blog Area Start ##### -->

<section class="cool-facts-area section-padding-100-0">

</section>

<!-- ##### Blog Area End ##### -->

<!-- ##### Footer Area Start ##### -->

<footer class="footer-area">

<!-- Top Footer Area -->

<div class="top-footer-area">

<div class="container">

<div class="row">

<div class="col-12">

<!-- Footer Logo -->

<div class="footer-logo">

<a href="index.jsp"><img src="img/core-img/logo2.png" alt=""></a>

</div>

</p>

</div>

</div>

</div>

</div>

<!-- Bottom Footer Area -->

<div class="bottom-footer-area d-flex justify-content-between align-items-center">

<!-- Contact Info -->

<div class="contact-info">

<a href="#"><span>Phone:</span> +44 300 303 0266</a>

<a href="#"><span>Email:</span> info@clever.com</a>

</div>

<!-- Follow Us -->

</div>

</footer>

<!-- ##### Footer Area End ##### -->

<!-- ##### All Javascript Script ##### -->

<!-- jQuery-2.2.4 js -->

<script src="js/jquery/jquery-2.2.4.min.js"></script>

<!-- Popper js -->

<script src="js/bootstrap/popper.min.js"></script>

<!-- Bootstrap js -->

<script src="js/bootstrap/bootstrap.min.js"></script>

<!-- All Plugins js -->

<script src="js/plugins/plugins.js"></script>

<!-- Active js -->

<script src="js/active.js"></script>

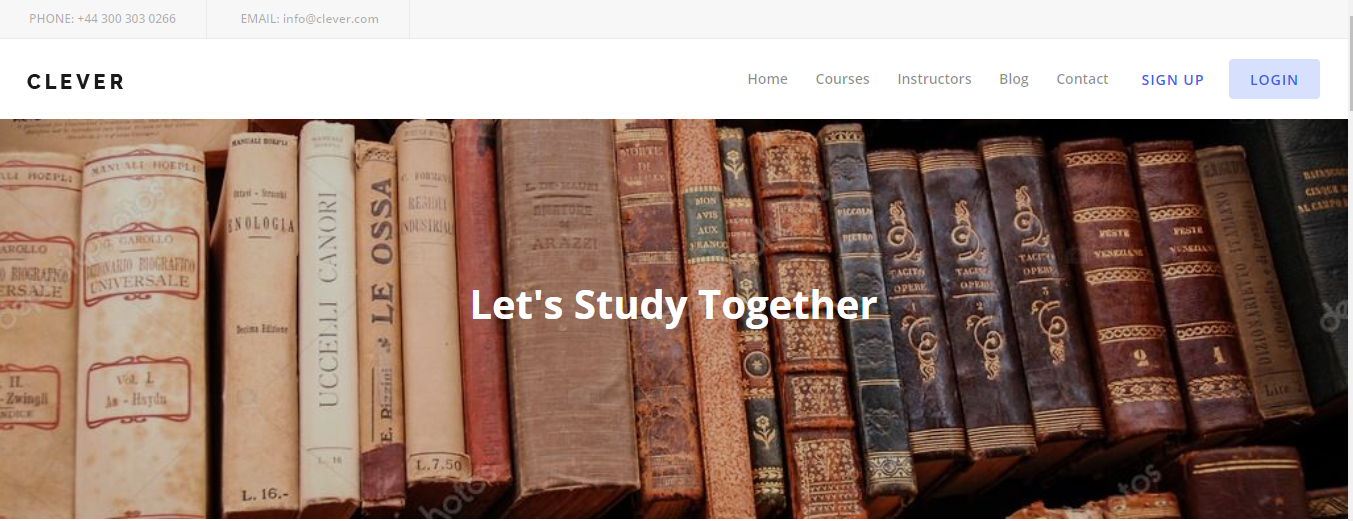
</body>

</html>

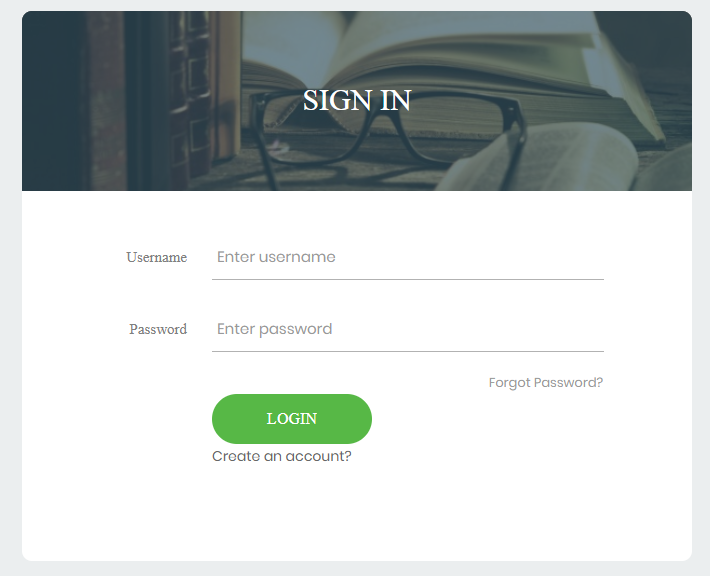
**7.2 APPENDIX B**

**7.2.1 Screenshots**

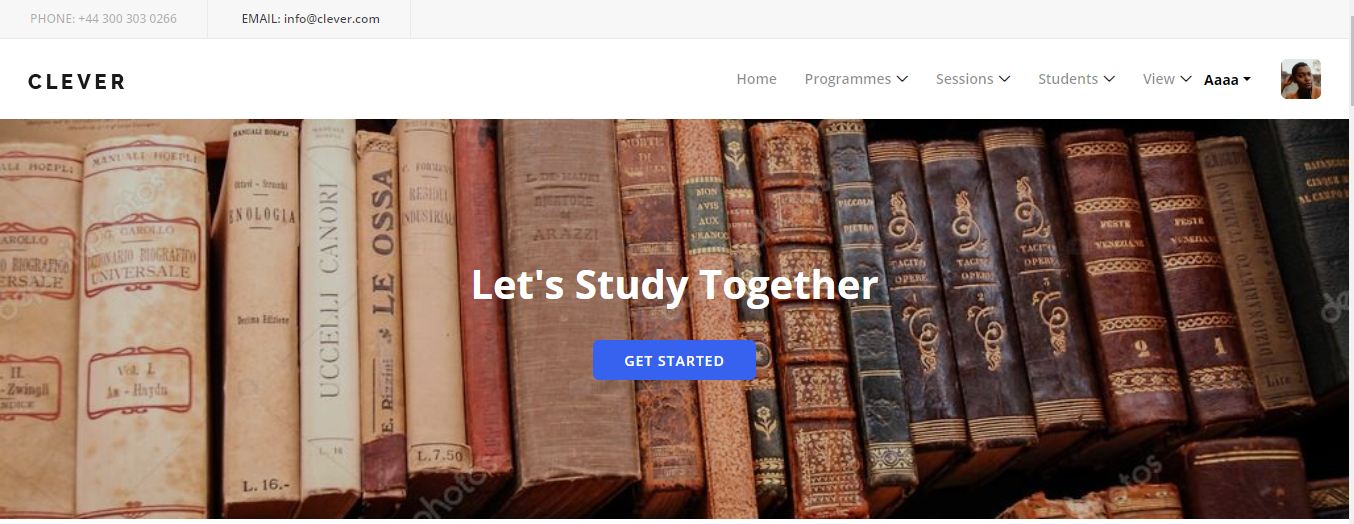
**Home Page**



**login page**



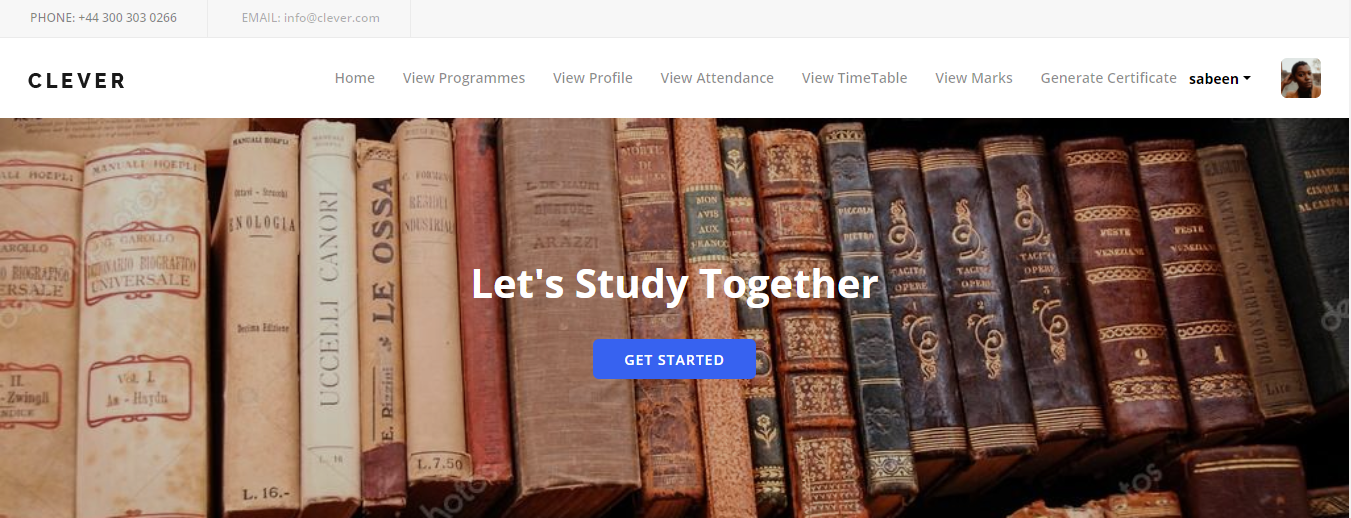
**Admin page**



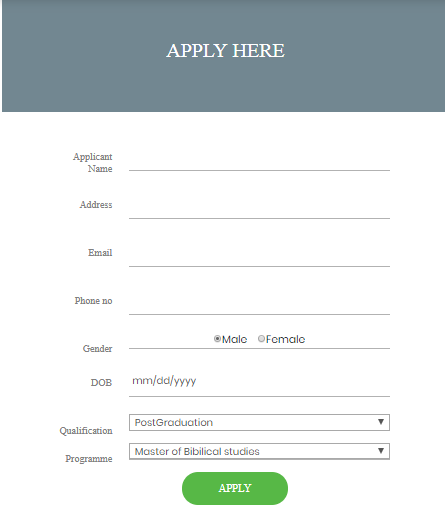
**add programme**



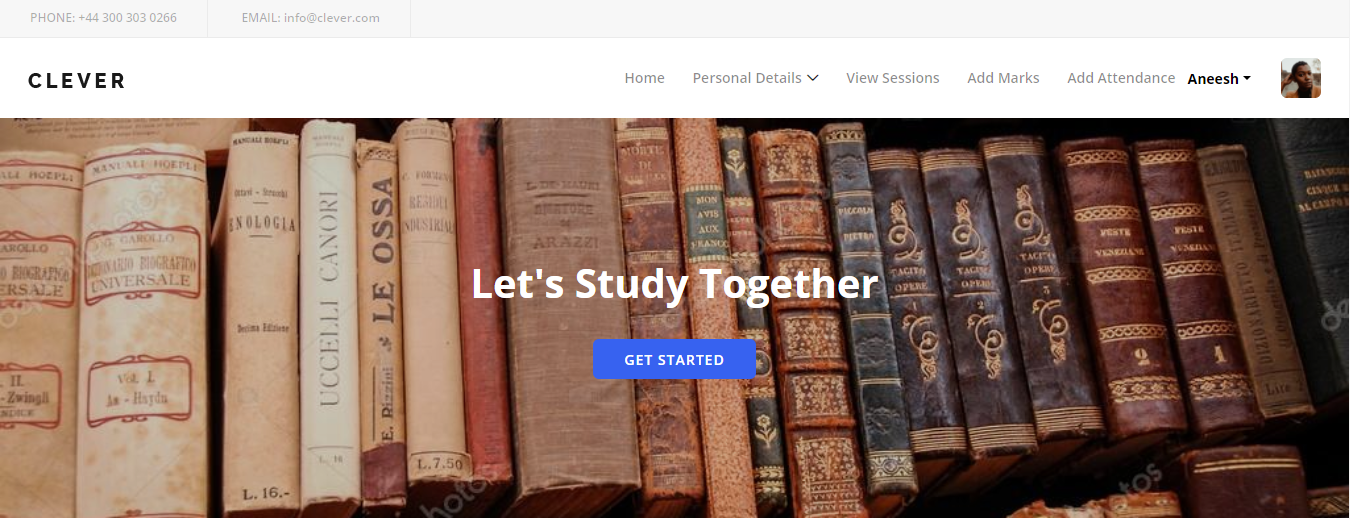
**student page**

****

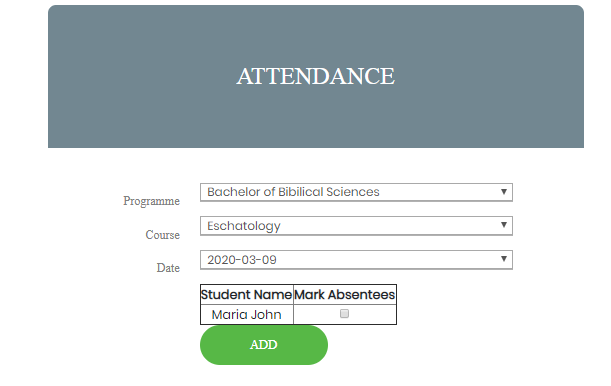
**apply programme**

****

**staff page**

**~~~~**

**add attendance**

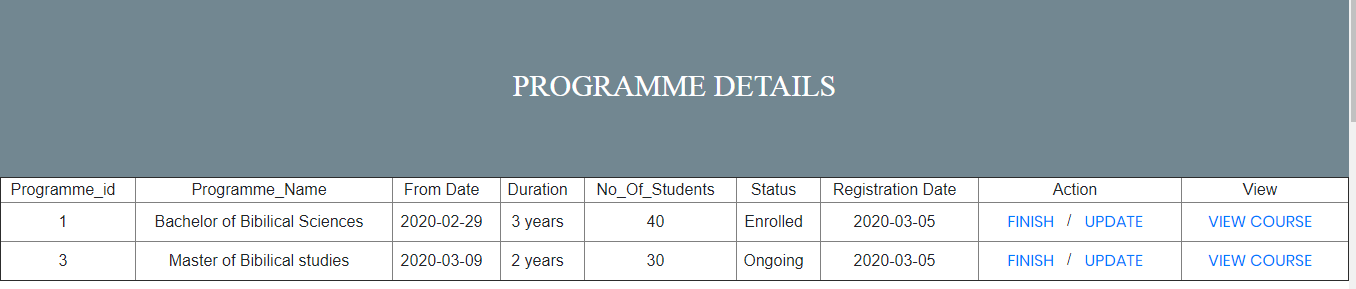
**~~~~**

**7.2.2 Printed Reports**

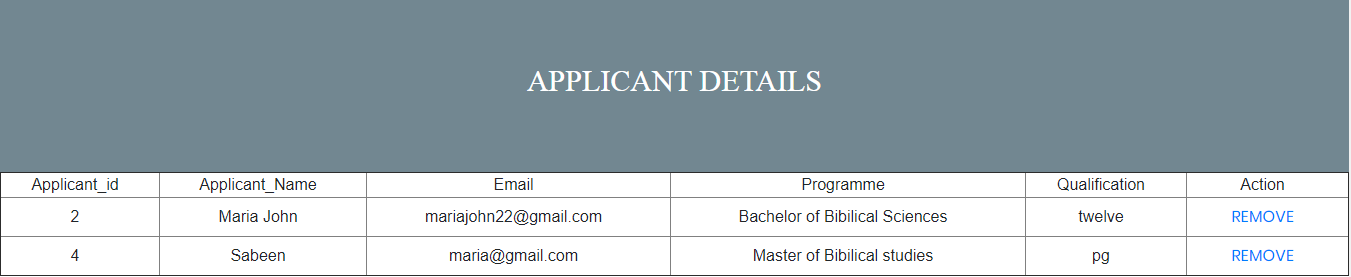
**staffs report**

****

**current coursereport**

****

**admitted students report**

****

**7.3 APPENDIX C**

**7.3.1 Bibiliography**

Books Referred

* Elias M. Award.(2005)System Analysis and Design. Galgotia. Second Edition.
* Roger S. Pressman.(2001)Software Engineering. McGraw Hill. Fifth Edition.
* Craig Larman.(2004) Applying UML and Patterns: An Introduction to Object oriented Analysis and Design And the Unified Process Pearson Education. Third Edition.
* Web Development with JavaServer Pagesby Duane K. Fields and Mark A. Kolb (Manning Publications, 2000)
* Core Servlets and JavaServer Pagesby Marty Hall (Prentice Hall, 2000)
* Pure JSP: Java Server Pagesby James Goodwill (Sams, 2000)
* JavaServer Pagesby Larne Pekowsky (Addison-Wesley, 2000)

Internet Site

* “Retrieved from Software Developing Life Cycles”(March 2012). http://www.sdlc.ws.
* “ObjectModelling Technique”(June 1999).http://www.omt.org/technology/uml.
* “SoftwareTestingDictionary”(2014). <http://www.tutorialspoint.com/sofware_testing/implementation>
* “Oriented\_Modeling\_And\_Design”

<https://www.researchgate.net/publication/2324983_Notes_On_Object\>

* “WebLogic jsp Reference”

https://docs.oracle.com/cd/E12839\_01/web.1111/e13712/reference.html