**NATIONAL UNIVERSITY OF SCIENCES & TECHNOLOGY**

**MILTARY COLLEGE OF SIGNALS, NUST**

****

Object Oriented Programming (OOP)

(CS-212)

* **Submitted by:** MUHAMMAD AHMAD SULTAN
* RANK: NC o CMS ID: 408709
* COURSE: BESE-28 o SECTION: C
* DEPARTMENT:  **COMPUTER SOFTWARE**

**ENGINEERING** (CSE)

* Submitted to: LE Muhammad Asif
* DATED: **13-05-2023**
* **Overview:**
* **INTRODUCTION**

**STUDENT MANAGEMENT SYSTEM (SMS)** is a flagship product of Easy Solution which covers all aspects of Universities, Colleges or Schools. SMS covers every minute aspects of a universities work flow and integrates all processes with user friendly interface. With hundreds of satisfied customers SMS is first choice of several state, governments/semi- government universities and institutions. SMS is an outcome of hard work done by our expert technical team in supervision of several renowned educationists which includes Controller of examination, faculties. SMS is a rare combination of experience and precision. SMS streamline path of information flow in organization by taking care of following departments:

• Fee Department

• Examination Department

• Attendance

• Faculty information portal

• Student information portal

* **Purpose:**

• Drive operational efficiency.

• Self-service systems with simple to use with little or no training.

• Elimination of duplicate data entry processes.

• Integrated with Online Application workflow with unified data model.

• Monitoring and decision support system.

• Automation of all the Academic / Examination / Administration operations.

• Ease and accuracy of reporting.

* **Scope:**

This project deals with the various functioning in College management process. The main idea is to implement a proper process to system. In our existing system contains a many operations registration, student search, fees, attendance, exam records, performance of the student etc. All these activity takeout manually by administrator.

* **REQUIREMENT SPECIFICATIONS**

**Hardware Requirements:**

* Processor Brand : Intel
* Processor Type : Core i3
* Processor Speed : 2 GHz
* Processor Count : 1
* RAM Size : 2 GB
* Memory Technology : DDR3
* Computer Memory Type : DDR3 SDRAM
* Hard Drive Size : 160 GB

**Software Requirements:**

* Operating system : Windows 10
* Application server : JAVA (NetBeans)
* Front end : JAVA
* Connectivity : JDBC Driver
* Database connectivity : WAMP (MYSQL Console)

**TOOL DESCRIPTION**

* **Overview of Front End**

An important issue for the development of a project is the selection of suitable front- end and back-end. When we decided to develop the project we went through an extensive study to determine the most suitable platform that suits the needs of the organization as well as helps in development of the project.

The aspects of our study included the following factors. Front-end selection:

1. It must have a graphical user interface that assists employees that are not from IT background.

2. Scalability and extensibility.

3. Flexibility.

4. Robustness.

5. According to the organization requirement and the culture.

6. Must provide excellent reporting features with good printing support.

7. Platform independent.

8. Easy to debug and maintain.

9. Event driven programming facility.

10. Front end must support some popular back end like MySQL.

According to the above stated features we selected PHP and CSS as the front-end for developing.

* **About Java:**

Java is a general-purpose, class-based, object-oriented programming language designed for having lesser implementation dependencies. It is a computing platform for application development. Java is fast, secure, and reliable, therefore. It is widely used for developing Java applications in laptops, data centers, game consoles, scientific supercomputers, cell phones, etc.

Here are some important Java applications:

• It is used for developing Android Apps

• Helps you to create Enterprise Software

• Wide range of Mobile java Applications

• Scientific Computing Applications

• Use for Big Data Analytics

• Java Programming of Hardware devices

• Used for Server-Side Technologies like Apache, JBoss, GlassFish, etc.

* **Overview of Back End**

Back End Selection:

1. Multiple user support.

2. Efficient data handling.

3. Provide inherent features for security.

4. Efficient data retrieval and maintenance.

5. Stored procedures.

6. Popularity.

7. Operating System compatible.

8. Easy to install.

9. Various drivers must be available.

10. Easy to implant with the Front-end.

According to above stated features we selected MySQL as the backend.

The technical feasibility is frequently the most difficult area encountered at this stage. It is essential that the process of analysis and definition be conducted in parallel with an assessment to technical feasibility. It centers on the existing computer system (hardware, software etc.) and to what extent it can support the proposed system.

* **About SQL:**

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons.

MySQL is released under an open-source license. So you have nothing to pay to use it. MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages. MySQL uses a standard form of the well-known SQL data language. MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.

MySQL works very quickly and works well even with large data sets. MySQL is very friendly to PHP, the most appreciated language for web development. MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

Also, they are using different dialects, such as −

• Oracle using PL/SQL

• SQL is widely popular because it offers the following advantages −

• Allows users to access data in the database management systems.

• Allows users to describe the data.relational

• Allows users to define the data in a database and manipulate that data.

• Allows to embed within other languages using SQL modules, libraries & pre-compilers.

• Allows users to create and drop databases and tables.

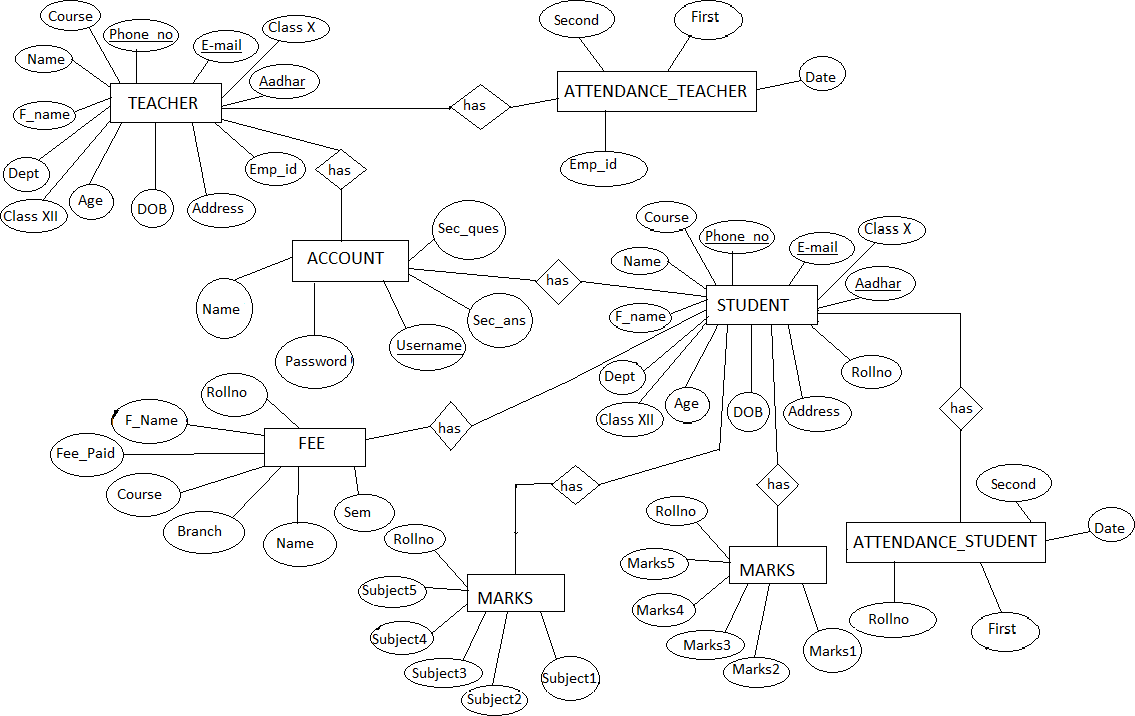
• Allows users to create view, stored procedure, functions in a database.

• Allows users to set permissions on tables, procedures and views.

**REQUIREMENT ANALYSIS**

**E-R DIAGRAM:**

ER Diagram is a high-level conceptual data model diagram. Entity-Relation model is based on the notion of real-world entities and the relationship between them. ER modelling helps you to analyse data requirements systematically to produce a well-designed database.



**SCHEMA DIAGRAM:**

A schema diagram is the skeleton structure that represents the logical view of the entire database. It contains a descriptive detail of the database.

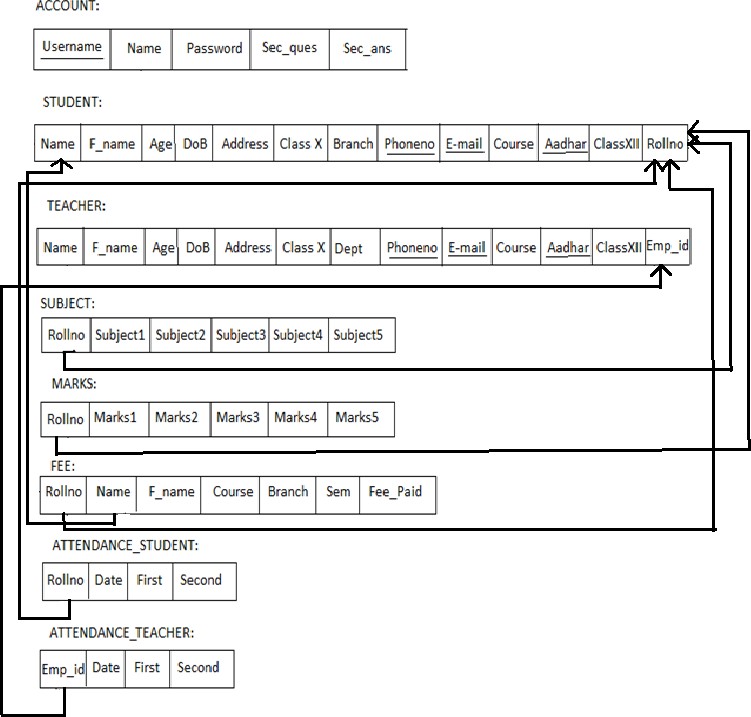


TABLE DESCRIPTION

**Database Queries for COLLEGE MANAGEMENT SYSTEM Project**

**1 - Create database**

create database collegemanagementsystem;

**2 - Use database you just created**

use collegemanagementsystem;

**3 - Create login table**

create table login(username varchar(25), password varchar(25));

**4 - Insert some values in the login table**

insert into login values('mas', 'mas03');

**5 - Create student table**

create table student(name varchar(40), fname varchar(40), rollno varchar(20), dob varchar(40), address varchar(100), phone varchar(20), email varchar(40), class\_x varchar(20), class\_xii varchar(20), cnic varchar(20), course varchar(40), branch varchar(40));

**6 - Create teacher table**

create table teacher(name varchar(40), fname varchar(40), empId varchar(20), dob varchar(40), address varchar(100), phone varchar(20), email varchar(40), class\_x varchar(20), class\_xii varchar(20), cnic varchar(20), education varchar(40), department varchar(40));

**7 - Create student leave table**

create table studentleave(rollno varchar(20), date varchar(50), duration varchar(20));

**8 - Create teacher leave table**

create table teacherleave(empId varchar(20), date varchar(50), duration varchar(20));

**9 - Create table to store subjects**

create table subject(rollno varchar(20), semester varchar(20), subject1 varchar(50), subject2 varchar(50), subject3 varchar(50), subject4 varchar(50), subject5 varchar(50));

**10 - Create table to store marks**

create table marks(rollno varchar(20), semester varchar(20), marks1 varchar(50), marks2 varchar(50), marks3 varchar(50), marks4 varchar(50), marks5 varchar(50));

**11 - Create table for fee structure**

create table fee\_(course varchar(20), semester1 varchar(20), semester2 varchar(20), semester3 varchar(20), semester4 varchar(20), semester5 varchar(20), semester6 varchar(20), semester7 varchar(20), semester8 varchar(20));

**12 - Insert some values in the table**

insert into fee\_ values("Bachelors", "175000", "124000","124000","124000","124000","124000","124000","124000");

insert into fee\_ values("Masters", "140000", "95000","95000","95000","","","","");

insert into fee\_ values("PhD", "135000", "110000","10000","110000","100000","110000","110000","110000");

**13 - Create table to store student fee details**

create table collegefee(rollno varchar(20), course varchar(20), branch varchar(20), semester varchar(20), total varchar(20));

**TESTING**

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is the process of executing the program with the intent of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. The ultimate aim is quality assurance.

* **Unit Testing**

The software units in a system are modules and routines that are assembled and integrated to perform a specific function. Unit testing focuses first on modules, independently of one another, to locate errors. This enables, to detect errors in coding and logic that are contained within each module. This testing includes entering data and ascertaining if the value matches to the type and size supported by java. The various controls are tested to ensure that each performs its action as required.

* **Integration Testing**

Data can be lost across any interface, one module can have an adverse effect on another, sub functions when combined, may not produce the desired major functions. Integration testing is a systematic testing to discover errors associated within the interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. Here the Server module and Client module options are integrated and tested. This testing provides the assurance that the application is well integrated functional unit with smooth transition of data.

* **User Acceptance**

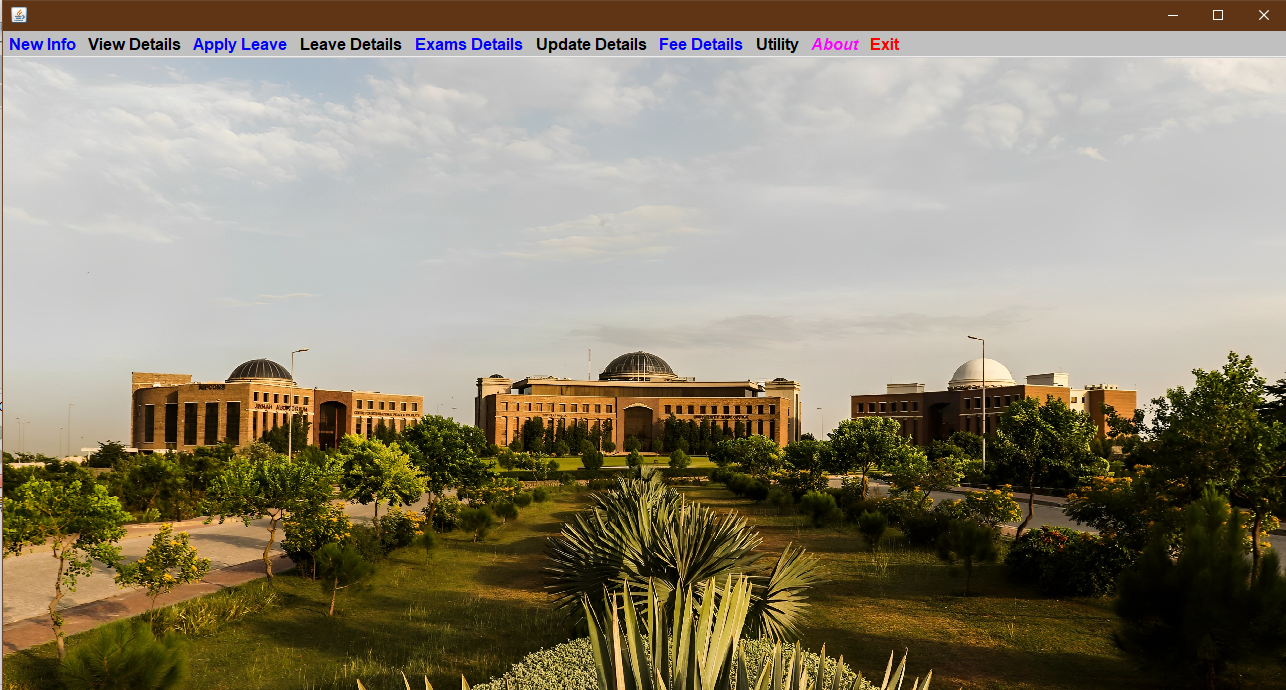
Testing User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the system users at time of developing and making changes whenever required.

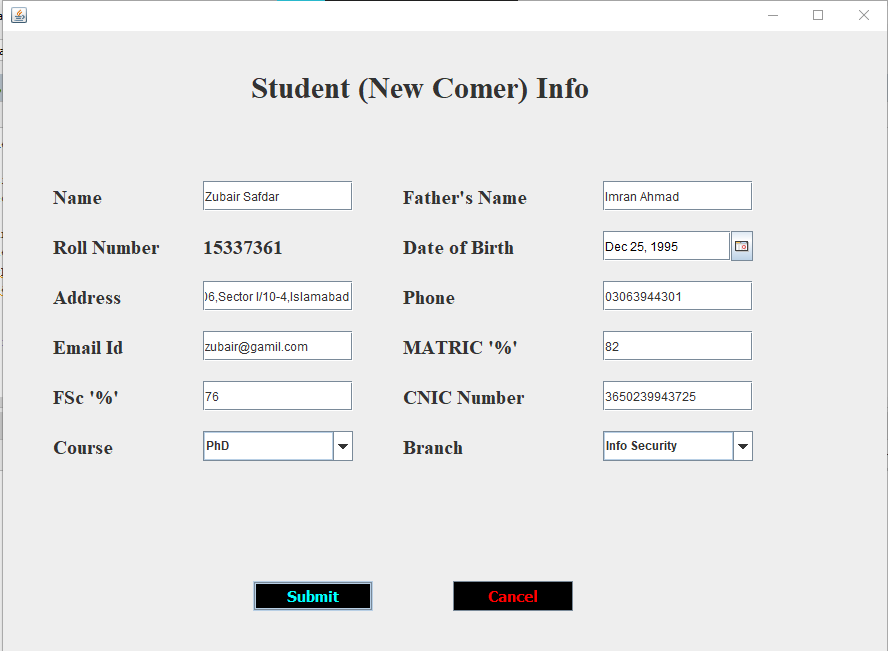
**TEST CASES**

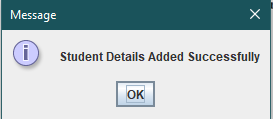
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Test Name** | **input** | **Actual output** | **Expected output** | **Status** |
| 1 | Login | Username and password | User is  successfully Authenticated | User is  successfully Authenticated | Pass |
| 2 | Login | Wrong username and  password | Invalid username or  password | Invalid username or  password | Pass |
| 4 | Student | Details of the  student required. | Student  inserted successfully | Student  inserted successfully | Pass |
| 5 | Teacher | Details of the  teacher required | Teacher  inserted successfully | Teacher  inserted successfully | Pass |
| 6 | Subject | Enter the subject names and marks along with  rollno | Subjects entered successfully | Subjects entered successfully | Pass |
| 7 | Fee | Details and fee\_paid | Paid successfully | Paid successfully | Pass |
| 8 | Remove Student | Enter rollno and click on  remove | Removed successfully | Removed successfully | Pass |
| 9 | Remove Teacher | Enter emp\_id and click on  remove | Teacher removed  successfully | Teacher removed  successfully | Pass |
| 10 | Exit | Click on Exit | Logout successfully | Logout successfully | Pass |

* **OUTPUT (ScreenShot) for Main Window**

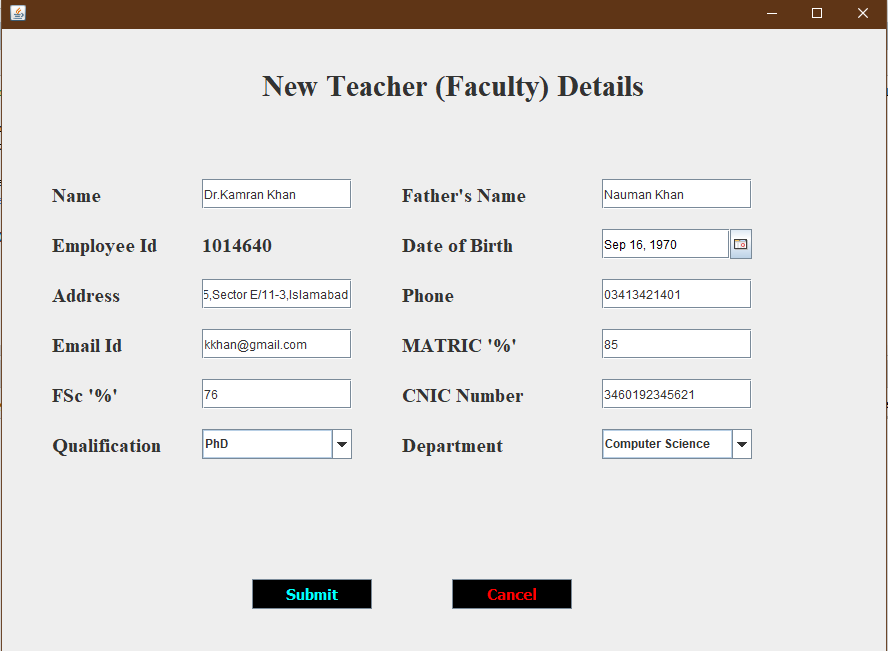


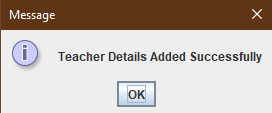
* **OUTPUT (ScreenShot) for LOGIN Window**
* **OUTPUT (ScreenShot) for LOGIN Window**
* **OUTPUT (ScreenShot) for ADD NEW STUDENT Window**



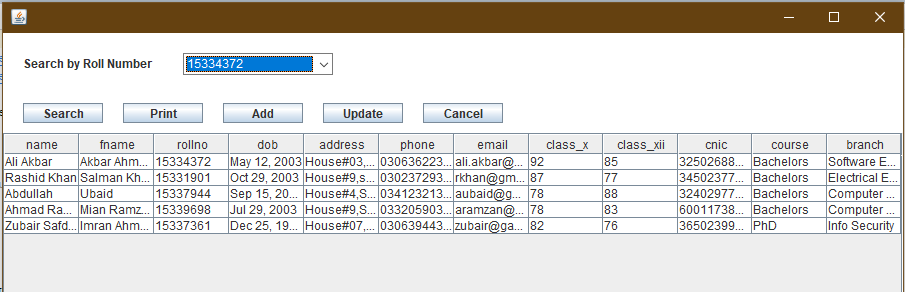


* **OUTPUT (ScreenShot) for ADD NEW TEACHER Window**

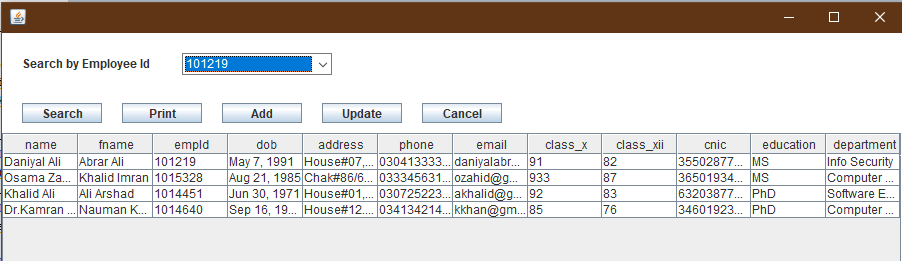




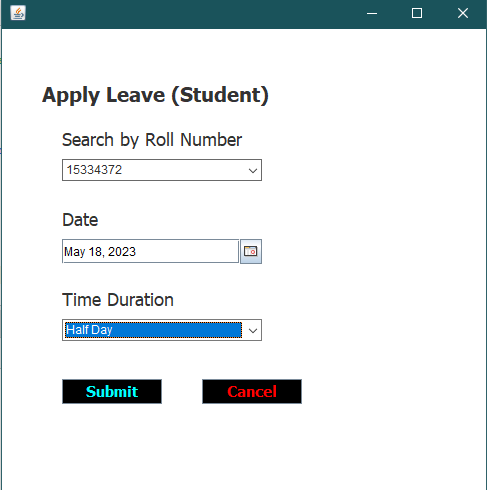
* **OUTPUT (ScreenShot) for STUDENT DETAILS Window**

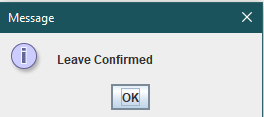


* **OUTPUT (ScreenShot) for TEACHER DETAILS Window**

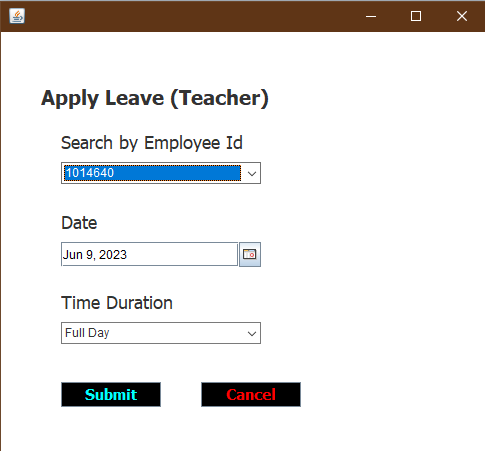


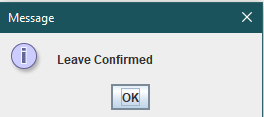
* **OUTPUT (ScreenShot) for STUDENT LEAVE Window**



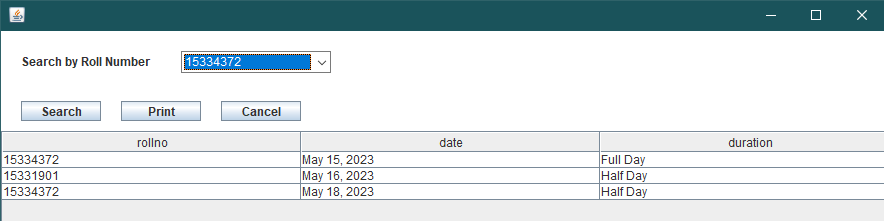


* **OUTPUT (ScreenShot) for TEACHER LEAVE Window**

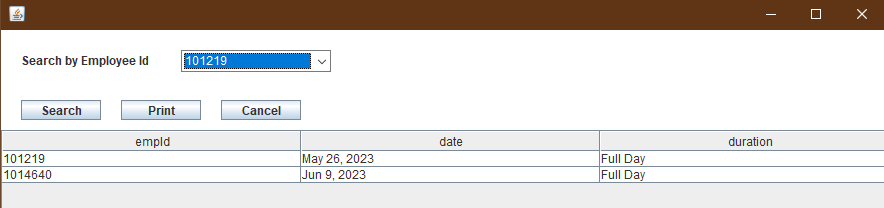




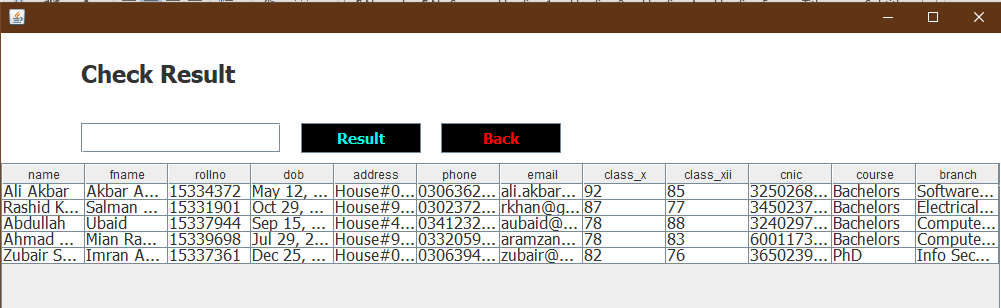
* **OUTPUT (ScreenShot) for STUDENT LEAVE DETAILS Window**

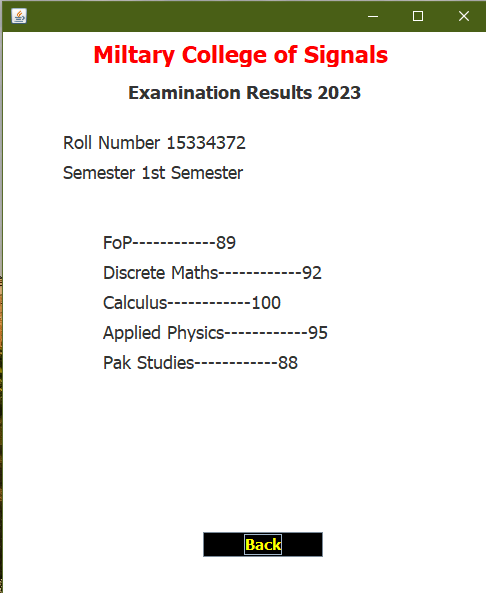


* **OUTPUT (ScreenShot) for TEACHER LEAVE DETAILS Window**

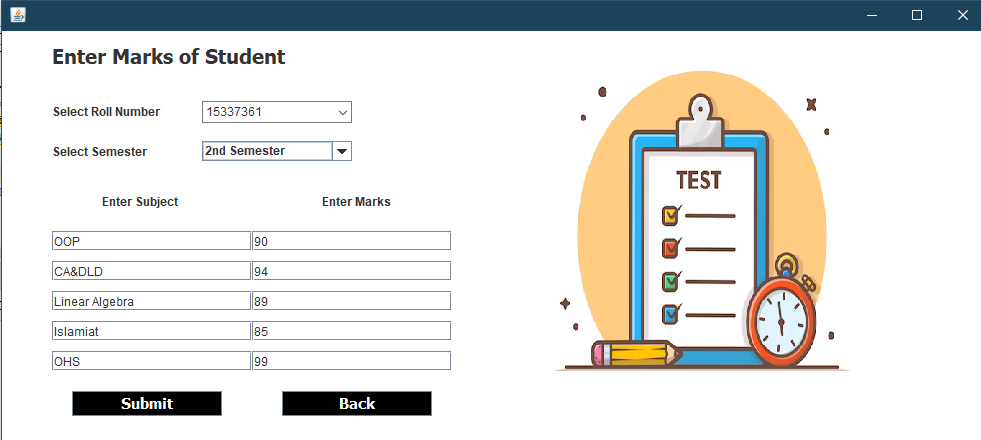


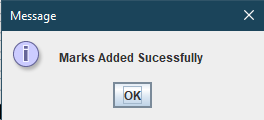
* **OUTPUT (ScreenShot) for EXAMINATION DETAILS Window**



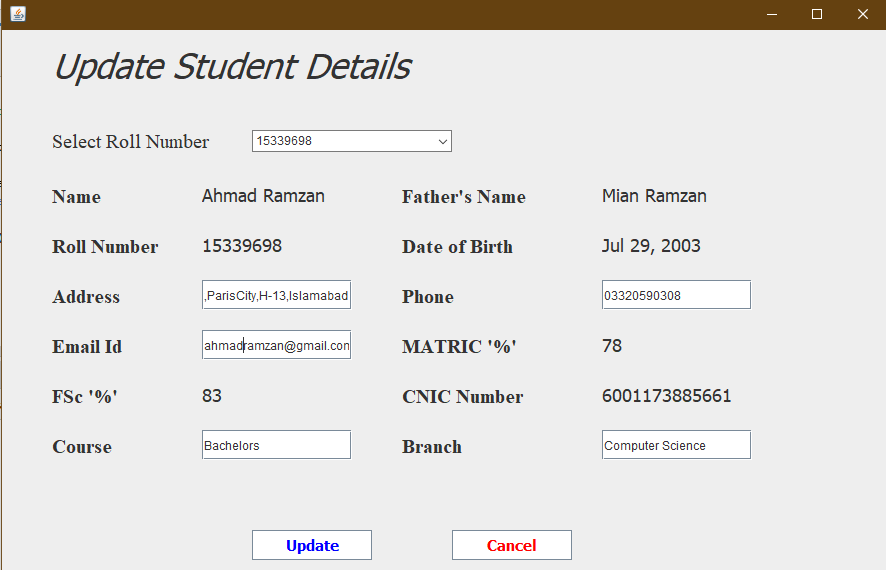


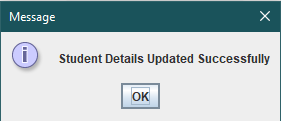
* **OUTPUT (ScreenShot) for ENTER MARKS Window**



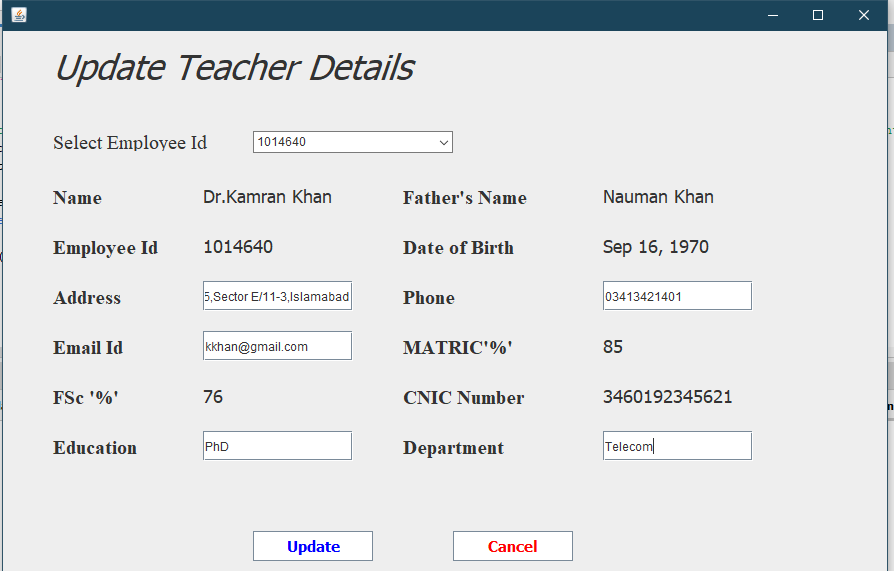


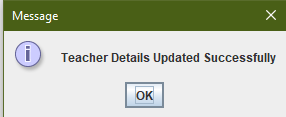
* **OUTPUT (ScreenShot) for UPDATE STUDENT DETAILS Window**



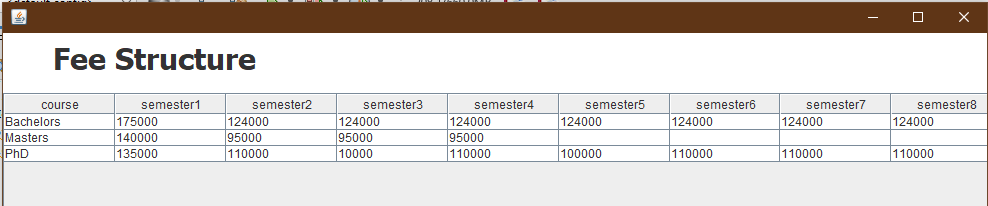


* **OUTPUT (ScreenShot) for UPDATE TEACHER Window**

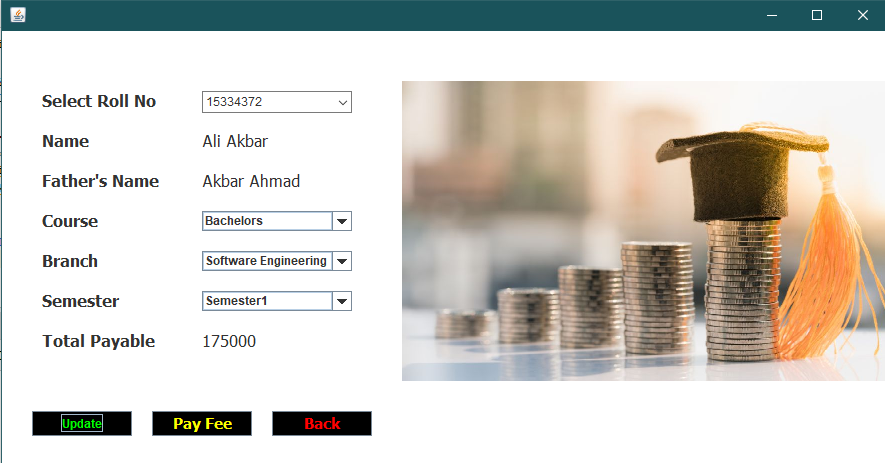




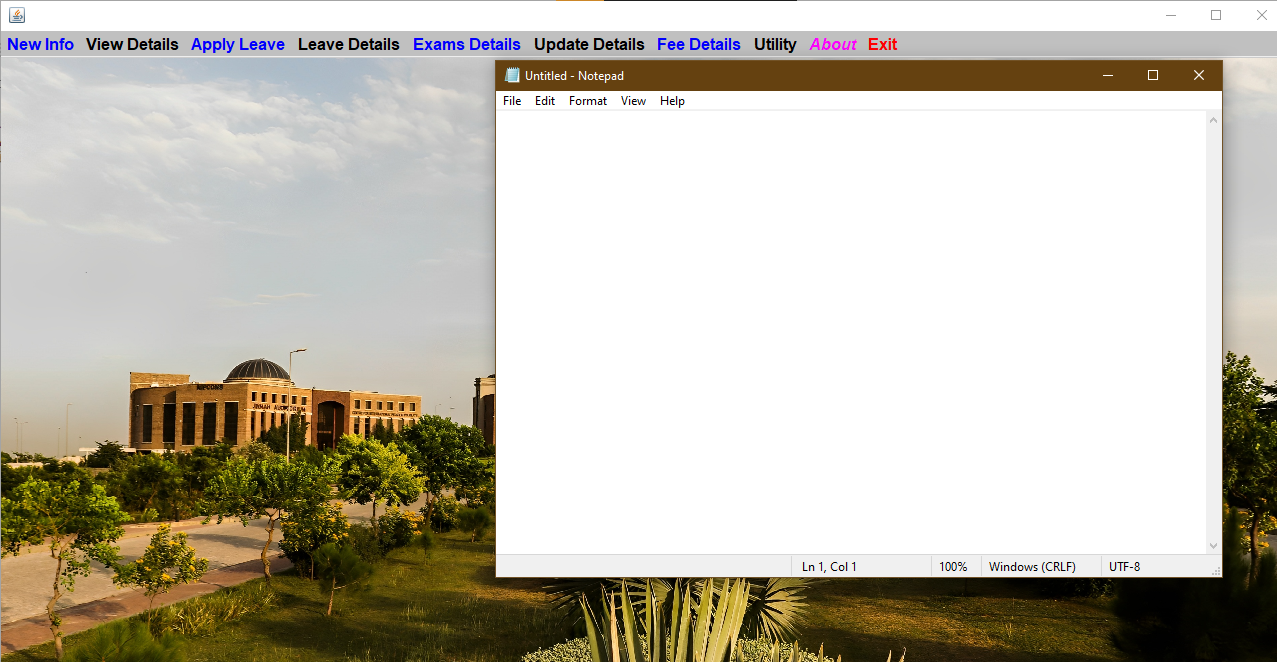
* **OUTPUT (ScreenShot) for FEE STRUCTURE Window**

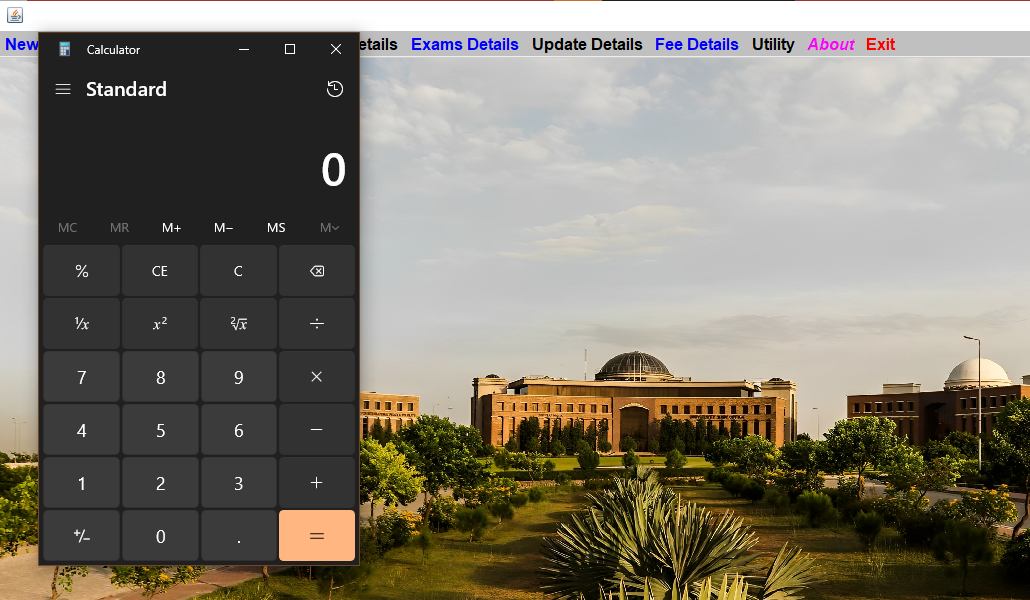


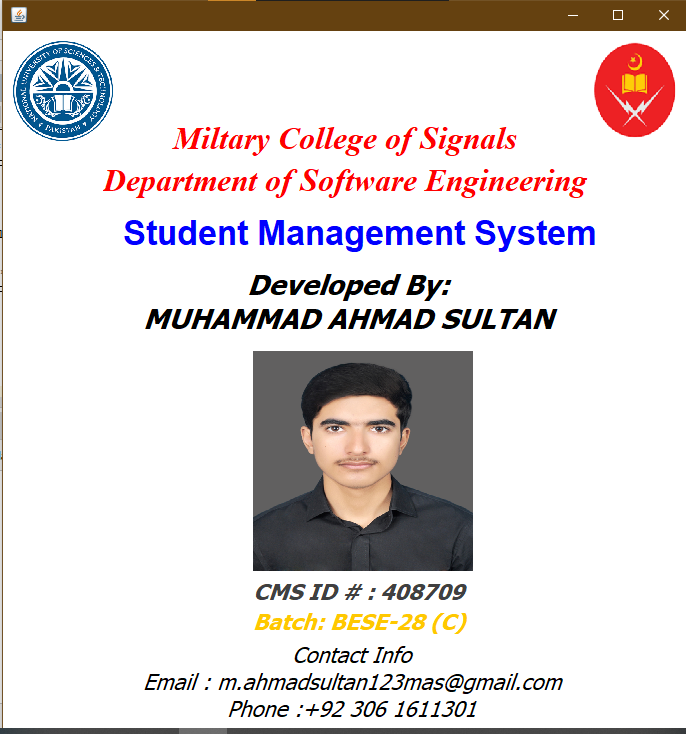
* **OUTPUT (ScreenShot) for STUDENT FEE FORM Window**



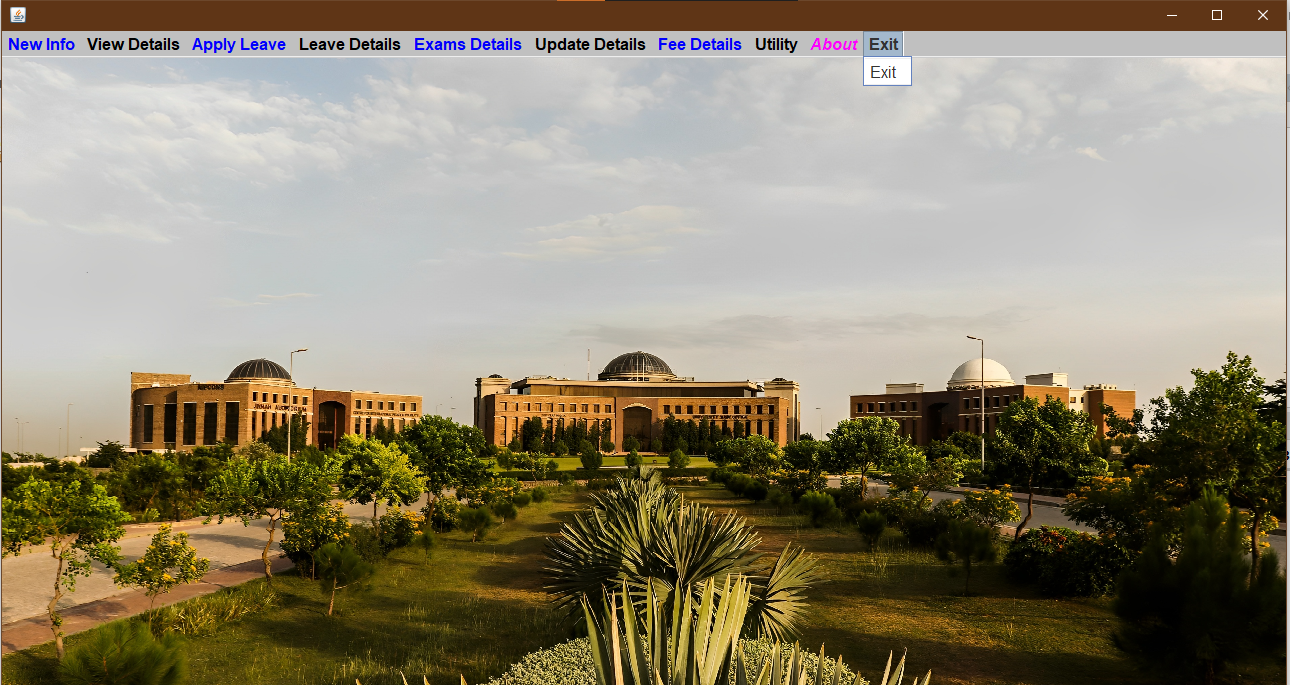
* **OUTPUT (ScreenShot) for NOTEPAD Window**

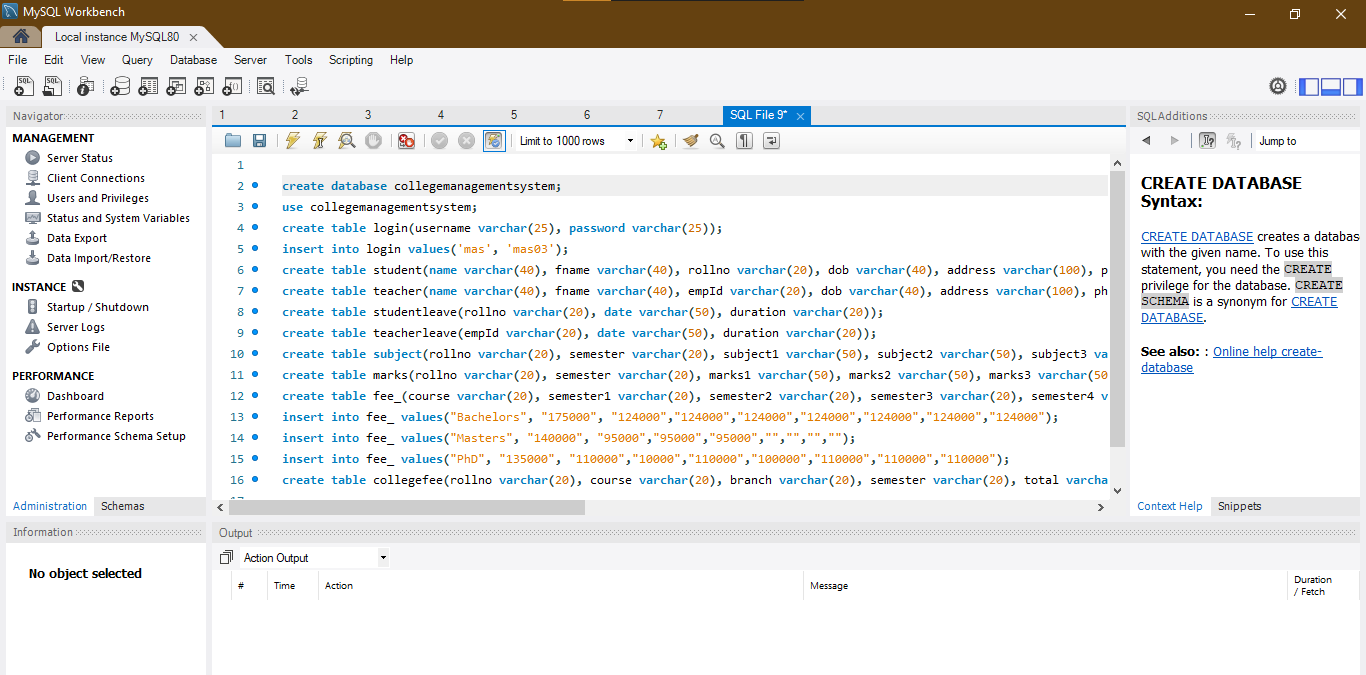


* **OUTPUT (ScreenShot) for CALCULATOR Window**
* **OUTPUT (ScreenShot) for ABOUT DEV Window**



* **OUTPUT (ScreenShot) for EXIT Window**



* **OUTPUT (ScreenShot) for DATABASE mySQL CONNECTIVITY Window**

**CONCLUSION**

The project entitled as Institution Management System is the system that deals with the issues related to a particular institution.

This project is successfully implemented with all the features mentioned in system requirements specification.

The application provides appropriate information to users according to the chosen service.

The project is designed keeping in view the day to day problems faced by a college.

Deployment of our application will certainly help the college to reduce unnecessary wastage of time in personally going to each department for some information.

Awareness and right information about any college is essential for both the development of student as well as faculty. So this serves the right purpose in achieving the desired requirements of both the communities.

* Signature of the student:
* Name of the Instructor: LE Muhammad Asif
* Signature of the Instructor :

**THE END**