Online College Management System

Project Report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology.

in

Computer Science & Engineering

under the Supervision of

Dr. Yashwant Singh

By

Ankit Sood (111288)

to



Jaypee University of Information and Technology Waknaghat, Solan – 173234, Himachal Pradesh Certificate

This is to certify that project report entitled "Online College Management System", submitted by

Ankit Sood in partial fulfillment for the award of degree of Bachelor of Technology in

Computer Science & Engineering to Jaypee University of Information Technology, Waknaghat,

Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the

award of this or any other degree or diploma.

Date:

Supervisor's Name: Dr. Yashwant Singh

Designation: Asst. Professor

Acknowledgement

Apart from the efforts of me, the success of my project depends largely on the encouragement

and guidelines of many others. I take this opportunity to express my gratitude to the people who

have been instrumental in the successful completion of this project.

I would like to show my greatest appreciation to my project in-charge, **Dr. Yashwant Singh**. I

can't say thank you enough for the tremendous support and help. I feel motivated and

encouraged every time I visit his cabin. Without his encouragement and guidance this project

work would not have materialized.

I'm highly grateful to all of the teachers of JUIT for their thorough guidance right from day 1.

They actually laid the ground for conceptual understanding of technologies used in project

Date:

Name of the student: Ankit Sood

Table of Content

S. No.	Topic	Page No.
1.	Introduction	1-2
1.1	Project Introduction	1
1.2	Project Profile	1
1.3	Project Scope	1
1.4	Project Objectives	2
1.3	Project Outline	2
2.	System Planning	3-8
2.1	Task Dependency Graph	7
2.2	Gantt Chart	8
3.	Proposed System	9-14
3.1	Scope of System	9
3.2	Functional Requirements of System	12
3.3	Non-Functional Requirements of System	12
3.4	Constraints of the System	12
3.5	Expected Advantages of the System	13
3.6	SystemRequirements	13
3.7	Technology Used	14

4.		System Modeling	19-33
	4.1	Use Case Diagrams	19
	4.2	System Sequence Diagram	22
	4.3	Activity Diagrams	23
	4.4	Data Flow Diagram	30
	4.5	Architecture Diagram	32
	4.6	Database Tables	33
5.		Implementation	35-51
	5.1	General	35
	5.2	HTML	35
	5.3	Database	38
	5.4	SQL	40
	5.5	Screenshots	42
6.		Testing and Enhancements	52-58
	5.1	Test Objective	52
	5.2	Test Cases	54
	5.3	Limitations and Enhancements	57
7.		References	59
8.		Appendix	60

List of Figures

Fig No.	Title	Page No.
1.0	Modified Waterfall Model	5
1.1	Task Dependency Graph	7
2.0	Gantt Chart	8
2.1	.NET Architecture	18
3.0	Use Case Diagram for admin	20
3.1	Use Case Diagram for student	20
3.2	Use Case Diagram for faculty	21
3.3	Use Case Diagram for end user	21
3.4	Sequence Diagram	22
3.5-4.1	Activity Diagrams	23-29
4.2	Level 0 Data Flow Diagram	30
4.3	Level 1 Data Flow Diagram	31
4.4	Architecture Diagrams 32	
5.1	Home Page	42
5.2	Admission Page	43
5.3	Academics Page	44
5.4	Campus Life Page	45
5.5	Contact Us Page	46
5.6	Login Page	47
5.7	Admin Home Page	48

5.8	Add Student Details Page	48
5.9	Add Employee Details Page	49
5.10	Add Course Details Page	49
5.11	View Employee Details Page	50
5.12	Students Home Page	50
5.13	Teachers' Home Page	51

Abstract

College management system is a centralized system that can access all the information about the students and teachers. I have been using the web kiosk from 1st year and I have always wondered about the entire working about it and how it can be improved, thus the idea of choosing this as my project came in my mind. I started this project by requirement analysis, for requirement analysis I studied other university websites and our web kiosk. I have always helped by my project guide Dr. Yashwant Singh in this regard. After the requirement analysis, I did the planning for my project like how it will be prepared, how many days it will took, what will be back end, what will be the front end. Third phase of this project development was system modeling. In this phase first I understood the system thoroughly using UML diagrams, then I designed a basic site map which gave me the layout of the system. Since, each university have their own emblem so I designed an emblem for this one too. Next task was to decide a color palette for the system. So I tried various palettes and chose one of them. Next, I decided to design the system keeping only end user in mind and designed the web pages.

CHAPTER 1: INTRODUCTION

1.1 Project Introduction:

Online college management system allows users to access the whole information about the college, its staff and students. This application will also provide a virtual tour of campus. This application is also designed for assisting the students of the institute regarding information on the streams, subjects, classes, assignments, grades and attendance. The system will provide the fee structure, course details and will act as the online e-portal of the college or university.

1.2 Project Profile:

Project Title: Online College Management System

Platform: Microsoft Windows 7 Home Premium

Database: Sql Express 2008

Front End: HTML V4 and V5, CSS

Web Server: Visual Studio Server 2010

Development: Adobe Muse CC, Microsoft VS 2010, Adobe Photoshop

Tools

1.3 Project Scope:

College Information: This facility will help any person to access the information about the college campus such as courses available, admission procedure, placements, faculty etc.

Student attendance status: This facility will provide the attendance status of students. Faculty will update the attendance periodically and can students can view their attendance on the wensite.

Student's performance in the exams: This facility will provide the performance of the student in exam conducted by university or college after each semester.

Online assignments and lectures: This facility will provide the online support for the assignments submission as well as assignment upload. All the students will be notified when the assignment will be uploaded. Online lectures will also be provided along with the study material which students can access from anywhere.

1.4 Objectives:

- To automate all the things in the college like assignment submission, result declaration and to provide all the details of the students to the faculty and to provide a provide a online system where attendance can be uploaded and displayed.
- To provide all the information about the college as well as all the ongoing activities of the college.
- To provide notifications to the user via email or sms whenever the assignment or attendance is uploaded.

1.5 Project Outline:

- Chapter 2 contains system planning and the approach followed for the development of the project.
- Chapter 3 explains the proposed system and the modules in which the system will be developed.
- Chapter 4 contains all the diagrams that explain the modeling of the system.
- Chapter 5 contains the test cases and the information regarding the shortcomings and enhancements that can be made in the system.
- Chapter 6 contains the screenshots of the working system.
- Appendix contains the sample code of important parts of the project.

CHAPTER 2: SYSTEM PLANNING

The system will be developed according Software Development Life Cycle which consists of following phases:

- Requirement Gathering
- Planning
- Coding
- Integration and Testing
- Maintenance

Requirement Gathering:

It's the first and most important part of the development because if the requirements will not be clear to us then the software developed at the end may or may not meet our expectations. For gathering need and functionalities, I studied various other websites online like Oxford University, University of California etc. I also took the help of some very experienced person like Dr. Yashwant Singh.

Planning:

In this phase, it's decided that in how much time the system will be developed, what type of model will be followed, which architecture will be used, what will be the front end and what will be the back end, which language will be used for the logical implementation.

There are mainly 5 types of models by the help of which software is developed. These are:

- Waterfall model
- Prototype Model
- Iterative Incremental Model
- The Spiral Model
- Dynamic System Development model

- 1. Waterfall model: This method suggests a systematic, sequential approach to software development that being at the system level and progress through analysis, design, coding, testing and maintenance. This model is also called "Linear Sequential Model" as one have to follow all the above five stages sequentially. Also called the "Classic Life Cycle" is the oldest and the most widely used paradigm for Software Engineering.
- 2. Prototype Model: The goal of prototyping based development is to counter the limitations of the waterfall model discussed earlier. The basic idea here is that instead of freezing the requirements before a design or coding can proceed, a throwaway prototype is built to understand the requirements. This prototype is developed based on the currently known requirements. Development of the prototype obviously undergoes design, coding and testing. But each of these phases is not done very formally or thoroughly. By using this prototype, the client can get an "actual feel" of the system, since the interactions with prototype can enable the client to better understand the requirements of the desired system.
- 3. The Spiral Model: This is a recent model that has been proposed by Boehm. As the name suggests, the activities in this model can be organized like a spiral. The spiral has many cycles. The radial dimension represents the cumulative cost incurred in accomplishing the steps dome so far and the angular dimension represents the progress made in completing each cycle of the spiral. The structure of the spiral model is shown in the figure given below. Each cycle in the spiral begins with the identification of objectives for that cycle and the different alternatives are possible for achieving the objectives and the imposed constraints
- 4. Incremental Model: The iterative enhancement life cycle model counters the third limitation of the waterfall model and tries to combine the benefits of both prototyping and the waterfall model. The basic idea is that the software should be developed in increments, where each increment adds some functional capability to the system until the full system is implemented. At each step extensions and design modifications can be made. An advantage of this approach is that it can result in better testing, since testing each increment is likely to be easier than testing entire system like in the waterfall model. Furthermore, as

in prototyping, the increments provide feedback to the client which is useful for determining the final requirements of the system.

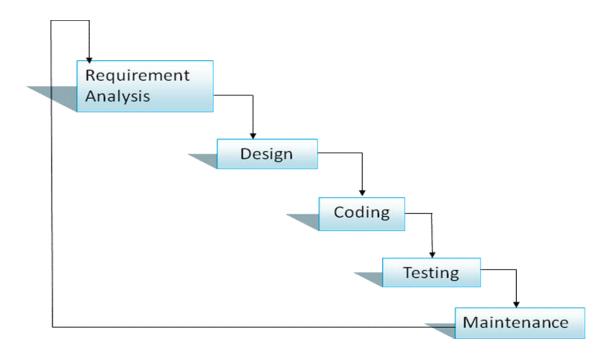


Fig 1.0

Reasons for selecting this model:

- 1. If all the stages mentioned above are followed sequentially then one will be able to make efficient software.
- 2. It provides the flavor of both prototype model and waterfall model so one can develop the system with fewer features and can again repeat the cycle to add some more features or one can develop the system for only one user and can make the system working for more users later on.
- 3. Also, it will be possible to add a service in between the any phase of the life cycle.

Coding:

- Coding of the front end will be done in HTML and CSS.
- Coding of the back end will be done in C#.

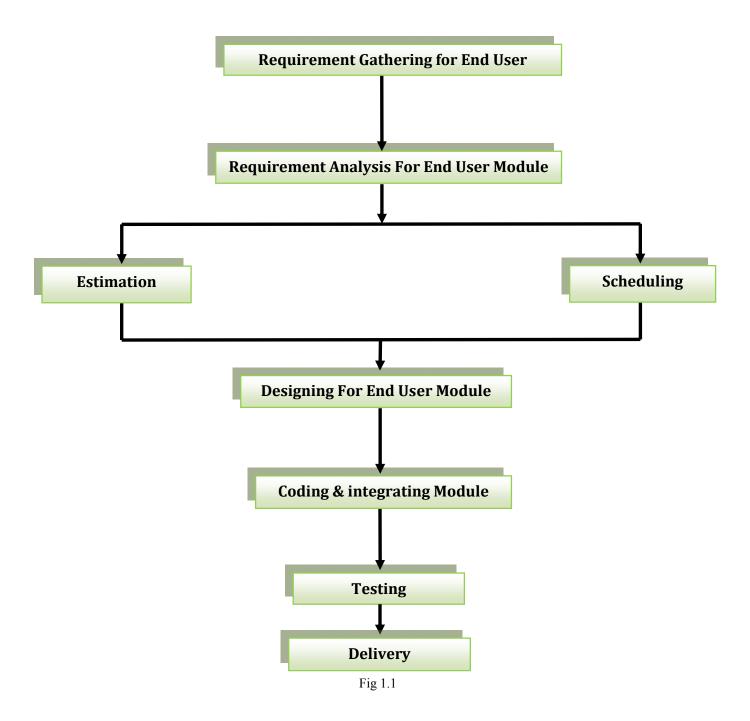
Coding phase will involve following steps:

- 1. Firstly, a MASTER pager will be designed which will ensure the main design of the page. Then all the other pages and forms will be designed.
- 2. Connectivity with the database and the validations in the form will be done with the help of visual studio which will also act as server for the web application.

Testing and Integration:

After the coding and designing all the pages independently all the pages will be tested for validation and working separately. If testing will be successful then all the pages will be integrated and the entire system will be tested for bugs.

2.1 Task Dependency Graph:



Gantt chart:



Fig 2.0

CHAPTER 3: PROPOSED SYSTEM

3.1 Scope of System:

The online college management system is concerned with four types of modules:

- Admin module
- End User Module
- Student Module
- Faculty Module

End User Module:

End User for this system will be the one who will not login into the system and will only view all the college information, courses offered, fee structure, photo gallery etc.

Following web pages will be made in this module:

- <u>Home Page:</u> This page will be the home page of the system which will appear whenever the user will open the system. On this page there will be information about the college as well as an option to view courses, faculty and to login to the system.
- Academics Page: This page will provide all the information regarding the schools of
 university, faculty of the university and academic calendar of the university and for each
 course offered there will be another page displaying the information regarding it.
- <u>Photo Gallery:</u> This page will contain the photos of the main events of the college as well of the college campus.
- <u>Campus Life:</u> This page will be totally devoted to the life inside the campus. Here all of the students will describe their life inside the campus.
- <u>Contacts Us:</u> There will be a contact us page where the contact information will be provided so that user can get in touch with the university administration. A feedback form will also be there where he will rate the website of the college.

- Maps & Directions: A page will be devoted for the maps so that user can easily find out
 where the university is situated and how he can reach there with the means of
 transportation.
- <u>Login/ Sign Up:</u> A page will be developed so that a user can log into the university system.
- Since an end user may or may not be a student, faculty or admin so all of these pages will also be for all of them before they log into the system.

Admin Module:

Admin will be the person who will add the information about the college events, courses etc. He will also be able to view all the details of the faculty as well as of the students. He will also be able to add a new user into the system.

Following web pages will be made in this module:

- <u>Home Screen Page:</u> A home screen page will be developed for the admin which will appear when the admin will be log into the system. All the functionalities available to the admin will be displayed on this homepage like view courses, update news or add user.
- <u>Courses Detail Page:</u> A web page will be made from where not only he can view all the courses information, he can also update or delete them.
- <u>User Page:</u> An add user page will be developed and by the use of this page admin will be able to add a new user, delete a new user or update any of the details of the user.
- News and Events Page: There will be a separate web page for news and events from where admin can view/ update/ delete the news and events of the college.
- <u>E-learning:</u> There will be a page on which admin will add all lecture slides as well as video lectures for the students.

Other than this, there will a faculty attendance page where admin will submit the attendance of the faculty.

Student Module:

This module will be focused on the students of the university, how they will interact with the system and what type of features will be available to the students.

Following web pages will be made in this module:

- <u>Profile page:</u> There will be a page which will be focused on the student's personal details like name, address, contact number etc. Contact information provided by the student on this page will be used for contacting student.
- Attendance pages: There will be a page dedicated to the attendance of the student. Here student can see their attendance subject wise.
- <u>Grades pages:</u> This page will be totally dedicated to the grades of the student in the examination. Here student can see their grades in each subject or they can see their CGPA or SGPA.
- <u>Assignment page:</u> There will be a page on which the students can submit their assignments online.

Other than this, there will be an online learning page where all the materials will be available and all the students will be notified whenever the assignment or attendance is uploaded.

Faculty Module:

This module will be focused on the faculties of the university, how they will interact with the system and what type of services will be available to the faculties.

Following web pages will be made in this module:

- <u>Profile page:</u> There will be a page which will be focused on the personal details of the faculty like name, address, contact number, qualification etc. Contact information provided by the faculty on this page will be used for contacting them.
- <u>Attendance pages:</u> There will be a page dedicated to the attendance of the student. Here faculty can see their attendance date wise. Other than this, there will be page from which the faculty will upload the student's attendance for their subject.

- <u>Grades pages:</u> This page will be totally dedicated to the grades of the student in the examination. Here faculty will upload grades of each student in each subject.
- <u>Assignment page:</u> There will be a page on which the faculty will upload the assignments for the students.

Other than this, there will be an online learning page where all the materials will be uploaded by the teachers of the university.

3.2 Functional Requirements of the System:

- Student can view their own result and attendance.
- It gives facility to staff to view attendance, to upload the attendance and assignments online.
- Students can search semester-wise subjects details and can see the curriculum of the university.
- End user can see all the college details, photographs, faculty details of the system so that they should be able to familiarize themselves with the college environment
- Students should be able to download time-table, assignments and exam schedule.
- System should be web based so that all of the students all around the world should be able to access the information about the university.

3.3 Non-Functional Requirements of the System:

- The system will be designed to be portable across popular Windows OS
- The system should be extensible to add further information and users for more expansion
- The system's code could be reused to add further new features if need to be added in future.

3.4 Constraints of the System:

- Admins' mistake can turn into an improper student result & attendance.
- It is a web based system so internet facility must be continuous.

- The performance of the system will be dependent on the internet speed
- It requires a server which .NET framework 4.0 or above.

3.5 Expected Advantages of the System:

- University will be able to interact with large no of students demographically.
- Student's Exam results will only be visible to him.
- Students will be able to view all the academic details from their home.
- Students can submit their assignments from anywhere in the world.
- Administrator can change or access data easily and at anytime.
- Students will be notified whenever an assignment or the attendance will be uploaded.

3.6 System Requirements:

Client Side Tools	
Processor	PC with Intel Pentium 4 processor or higher
RAM	Min 256 MB DDR1 or higher
Hard Disk	40 MB or higher space available
Operating System	Windows XP or higher
Browser	Google Chrome/ Mozilla firefox 7.0/ Internet Explorer 7.0 or onwards

Server Side Tools	
Processor	PC with Intel C2D processor or higher
RAM	Min 1 GB DDR2 or higher
Hard Disk	1 GB or higher space available
Operating System	Windows XP or higher
Browser	Google Chrome/ Mozilla firefox 7.0/ Internet Explorer 7.0 or onwards
Software	Sql Express 2008, .NET framework 4.0

Development Side Tools	
Processor	PC with Intel Core i5 processor
RAM	4 GB
Hard Disk	1 TB
Operating System	Windows 7
Software	MS Visual Studio 2010, Sql Express 2008, Adobe Muse CC,
	Adobe Photoshop CS5
Browser	Google Chrome

3.7 Technology Used:

- Adobe MUSE CC and Visual Studio 2010 is used for the development of the web pages.
- C#.NET is used for the logical implementation of the website.
- SQL Server is used for managing database at server side.

Microsoft SQL Server 2008:

Microsoft SQL Server 2008 is a full-featured relational database management system (RDBMS) that offers a variety of administrative tools to ease the burdens of database development, maintenance and administration. In this article, we'll cover six of the more frequently used tools: Enterprise Manager, Query Analyzer, SQL Profiler, Service Manager, Data Transformation Services and Books Online.

Components of Microsoft SQL Server 2008

Enterprise Manager: is the main administrative console for SQL Server installations. It
provides you with a graphical "birds-eye" view of all of the SQL Server installations on
your network. You can perform high-level administrative functions that affect one or
more servers, schedule common maintenance tasks or create and modify the structure of
individual databases.

- Query Analyzer: offers a quick and dirty method for performing queries against any of your SQL Server databases. It's a great way to quickly pull information out of a database in response to a user request, test queries before implementing them in other applications, create/modify stored procedures and execute administrative tasks.
- **SQL Profiler:** provides a window into the inner workings of your database. You can monitor many different event types and observe database performance in real time. SQL Profiler allows you to capture and replay system "traces" that log various activities. It's a great tool for optimizing databases with performance issues or troubleshooting particular problems.
- Service Manager: is used to control the MS-SQL Server (the main SQL Server process),
 MSDTC (Microsoft Distributed Transaction Coordinator) and SQL Server Agent processes. An icon for this service normally resides in the system tray of machines running SQL Server. You can use Service Manager to start, stop or pause any one of these services.
- Data Transformation Services (DTS): provide an extremely flexible method for importing and exporting data between a Microsoft SQL Server installation and a large variety of other formats. The most commonly used DTS application is the "Import and Export Data" wizard found in the SQL Server program group.
- Books Online: is an often overlooked resource provided with SQL Server that contains
 answers to a variety of administrative, development and installation issues. It's a great
 resource to consult before turning to the Internet or technical support.

Features of Microsoft SQL Server 2008

• User-Defined Functions: SQL Server 2000 introduces the long-awaited support for user-defined functions. User-defined functions can take zero or more input parameters and return a single value—either a scalar value like the system-defined functions, or a table result. Table-valued functions can be used anywhere table or view expressions can be used in queries, and they can perform more complex logic than is allowed in a view.

• Indexed Views: Views are often used to simplify complex queries, and they can contain joins and aggregate functions. In SQL Server 2000 Enterprise or Developer Edition, you can define indexes on views to improve query performance against the view. When creating an index on a view, the result set of the view is stored and indexed in the database. Existing applications can take advantage of the performance improvements without needing to be modified.

.NET FRAMEWORK

• **DEFINITION:**

The .Net framework can be defined as a language neutral platform designed to provide a number of rich and powerful application development tools and technologies. It has an extensive class library that provides wide-ranging support for data access. .Net framework has an independent code execution and management environment called the Common Language Runtime (CLR) which ensures that code is safe to run, and provides an abstract layer on top of the operating system, which allows the elements of the .Net framework to run on many operating systems and devices.

• THE .NET VISION:

The basic idea behind the development of the .Net framework is that a global broadband network will someday globally connect all devices and the software will become a service provided over this network. The .Net framework provides the foundation and the plumbing on which the Microsoft .Net vision is built. The vision is about having a global network and open standards like XML to make it easier for parties to exchange data and work together. Now devices will use common languages like XML over standardized or shared protocols such as HTTP and these devices will be running a multitude of software on various operating systems. This is not limited to Microsoft but also to other operating systems such as sun, IBM.

The .Net framework provides the tools and the technologies needed to write applications that can seamlessly and easily communicate over the Internet using open standards like XML and SOAP. It also aims at solving the various problems that are faced by developers while working on

applications made using the windows DNA. It provides a solution to problems such as registering of components and shutting down applications in order to upgrade them.

Benefits of Using .NET

The windows platform is inherently complex and may pose several problems for programmers. However the .Net platform is used as it provides solutions to many such problems and offers an Internet centric platform.

The following are the benefits of using .Net platform:

1) Multiple platform Support

.Net platform has been designed keeping multiple platform support as a key feature. For version 2.0 this means that the code written using the .Net platform can run on all versions of Windows. i.e. Windows 98,95,NT,2000 and so on.

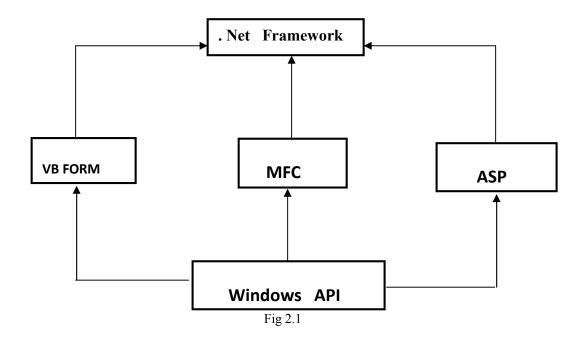
Microsoft has included rich support for all the platforms. Also this code shall also work on any 64-bit processor that may be used by Microsoft later.

It is also expected that .Net shall run on other platforms like UNIX also, however it is not for sure that all functionality will be available for the same.

2) . Net as a clean start

On the windows platform, the technologies available often depend on the programming language that they are written in, so they are very different. Depending on the chosen programming language, these technologies can be restrictive.

The problem with this approach is that it makes the Windows platform harder to understand. Using .Net there is just one simple object oriented way of accessing the functionality of the .Net Framework.



3) Performance

.Net has been designed to provide great performance and scalability. For .Net to succeed, the companies must be able to migrate their applications and not suffer due to the way the CLR works. To ensure this CLR converts the code into native machine code. So the conversion can take step-by-step method and will make use of the given machines resources and processor features.

As newer versions of the CLR are released and technologies like windows forms are released, each release will have a better performance and smaller memory footprints. Also .Net has succeeded in replacing completely the older technologies like COM with better and efficient design equivalents. At the heart of the .Net platform is the CLR.

CHAPTER 4: SYSTEM MODELLING

The main focus of the analysis phase of Software development is on "What needs to be done". The objects discovered during the analysis can serve as the framework or Design. The class's attributes, methods and association identified during analysis must be designed for implementation language. New classes must be introduced to store intermediate results during the program execution.

Emphasis shifts from the application domain of implementation and computer such as user interfaces or view layer and access layer. During analysis, we look at the physical entities or business objects in the system, that is, which players and how they cooperate to do the work of the application. These objects represent tangible elements of the business.

During the Design phase, we elevate the model into logical entities, some of which might relate more to the computer domain as people or employees. Here his goal is to design the classes that we need to implement the system the difference is that, at this level we focus on the view and access classes, such as how to maintain information or the best way o interact with a user or present information.

Design process:

During the design phase the classes identified in object-oriented analysis Must be revisited with a shift focus to their implementation. New classes or attribute and Methods must be an added for implementation purposes and user interfaces.

The following are some of the vies of software design life cycle. They are

- Data Flow Diagrams
- UML Diagrams
- Data Base Design

4.1 Use Case Diagram:

A use case diagram is a diagram that shows a set of use cases and actors and relationships.

Use case Diagrams represent the functionality of the system from a user's point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system .Use cases focus on the behavior of the system from external point of view.

This is the use case diagram for the Admin. It describes that the admin can view courses, photos, college details as well as the news and events. He will also be able to update and delete all the information regarding courses, college or events as well as the personal details.

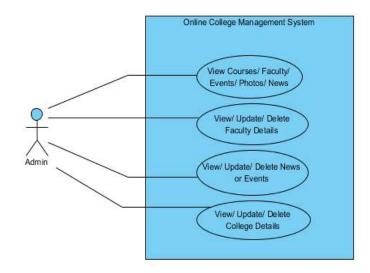


Fig 3.0 Use Case diagram for Admin

This is the use case diagram for the Student. It describes that the student can view courses, events, attendance and result. He will also be able to update his personal information and to upload the assignments online.

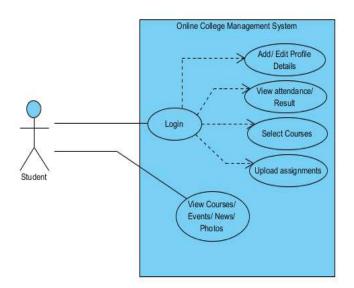


Fig 3.1 Use Case diagram for Student

This is the use case diagram for the Faculty. It describes that the faculty can view result, attendance, college details as well as the personal details. He will also be able to update and delete personal details, assignments and information regarding the results.

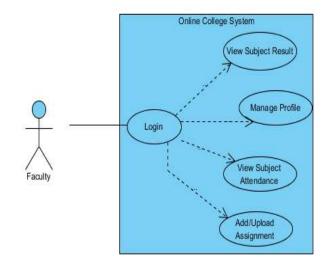


Fig 3.2 Use Case diagram for Faculty

This is the use case diagram for the third user. He can view all the information regarding college, courses it offers and about the faculty of the college along with the fee structure and events of the college.

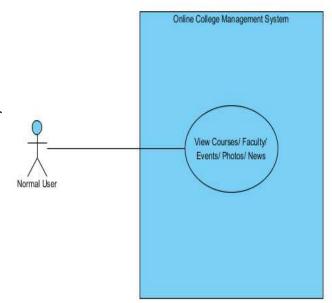
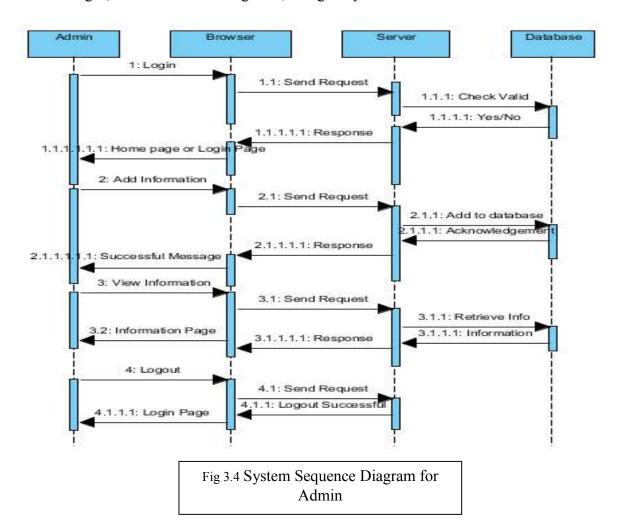


Fig 3.3 Use Case diagram for End User

4.2 System Sequence Diagram:

- A sequence diagram is an interaction diagram that emphasizes the time ordering of messages.
- An interaction diagram shows an interaction, consisting of a set of objects and their relationships, including the messages that may be dispatched among them.
- Graphically, a sequence diagram is a table that shows objects arranged along x-axis and messages, ordered in increasing time, along the y-axis.



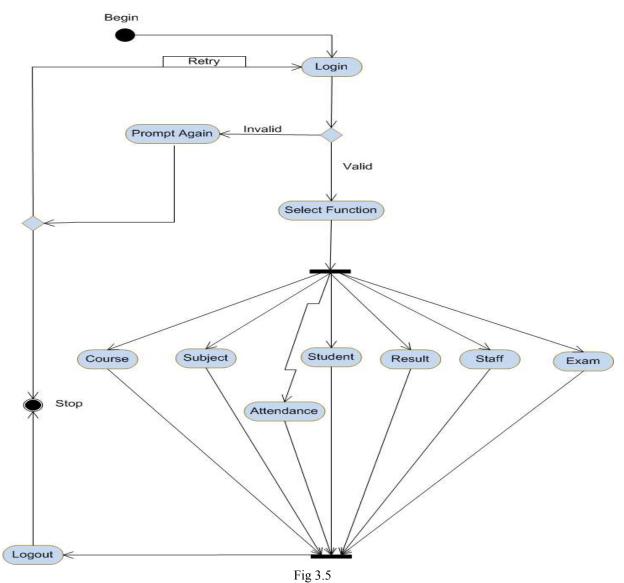
This is the system sequence diagram for the admin. It shows how the various layers of the system will interact with each other whenever user will provide the input to the system and how it will show the output back to the user. It will also show the time line of each layer of the system.

4.3 Activity Diagrams:

- An activity diagram shows the flow from activity to activity. An activity is an ongoing non-atomic execution within a state machine.
- Activities ultimately result in some action, which is made up of executable atomic computations that result in a change in state of the system or the return of a value.

Activity diagrams commonly contain

- Activity states and action states
- Transitions
- Object



Page 24

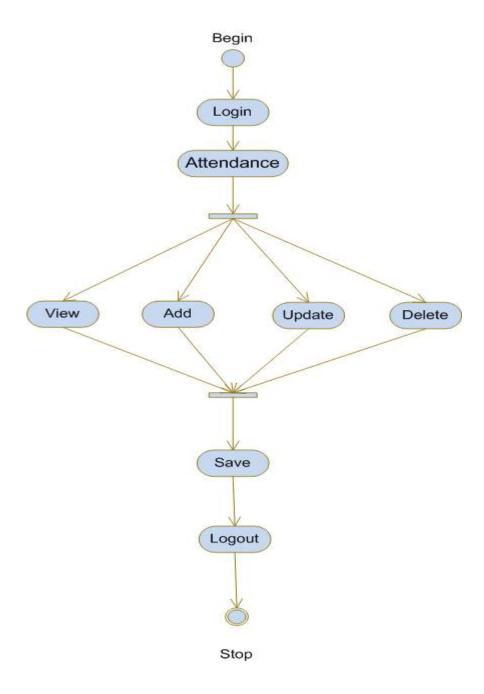


Fig 3.6

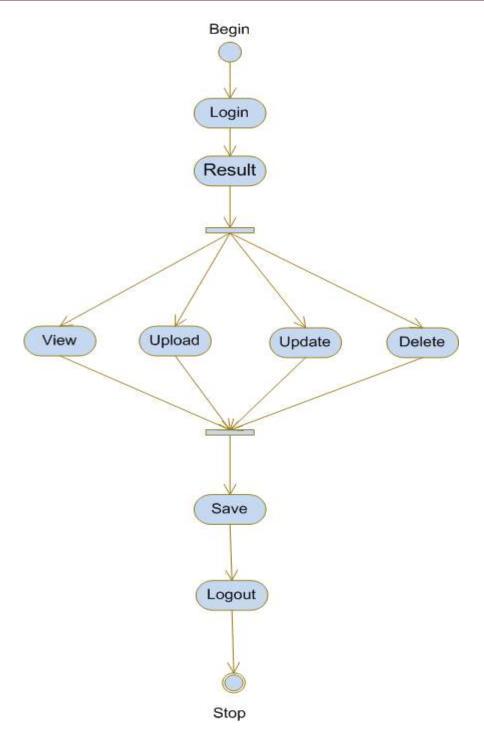


Fig 3.7

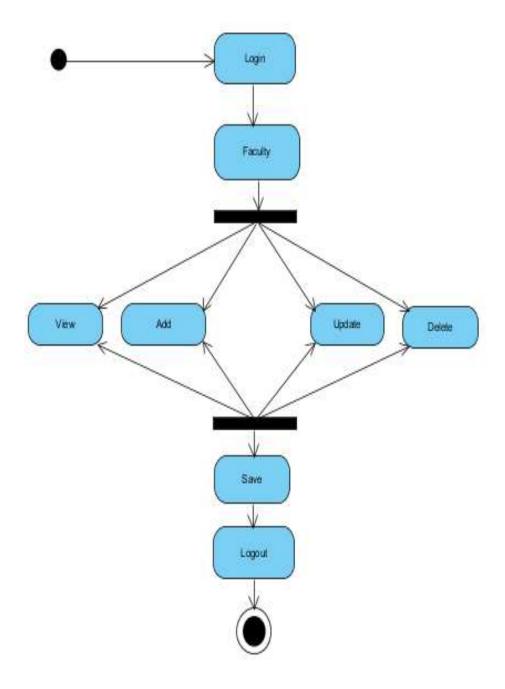


Fig 3.8

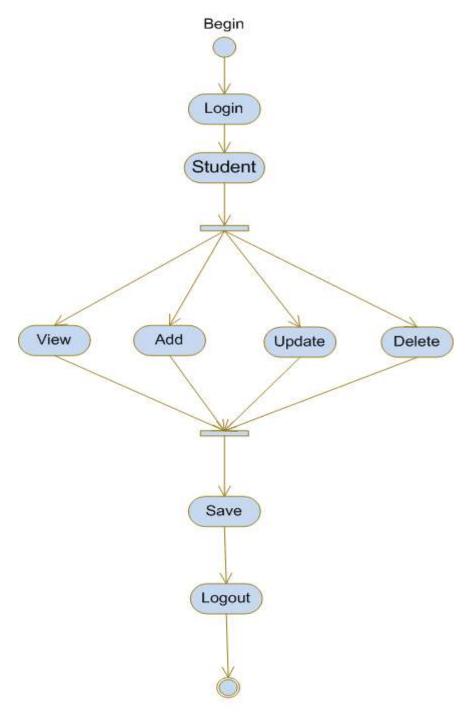


Fig 3.9

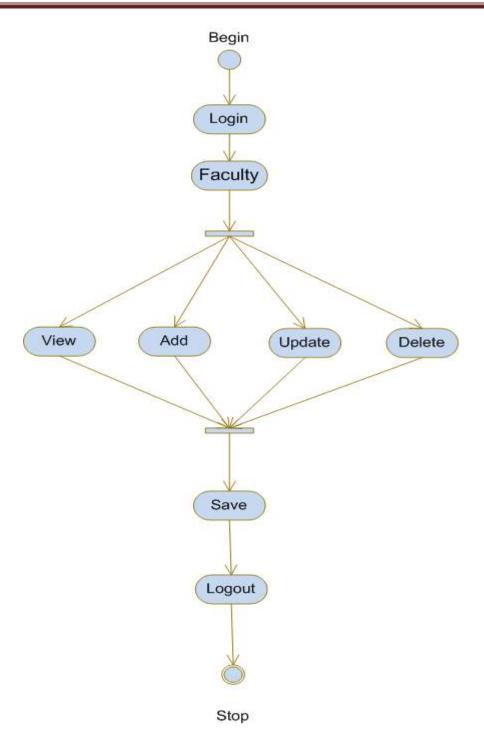


Fig 4.0

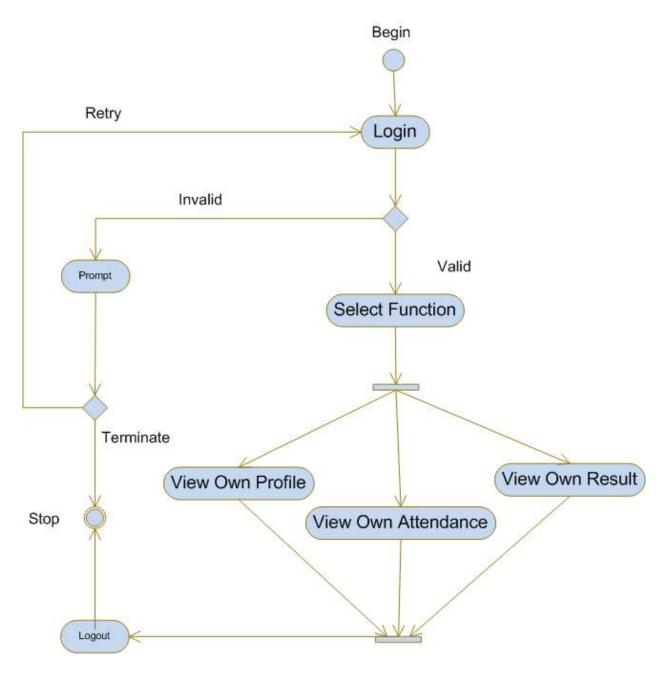


Fig 4.1

4.4 Data Flow Diagram:

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system. It can also be used for the visualization of data processing (structured design).

There are two types of DFDs. They are:

- Context Level DFD
- Top Level DFD

Context Level DFD

In the Context Level the whole system is shown as a single process.

- No data stores are shown.
- Inputs to the overall system are shown together with data sources (as External entities).
- Outputs from the overall system are shown together with their destinations (as External entities).

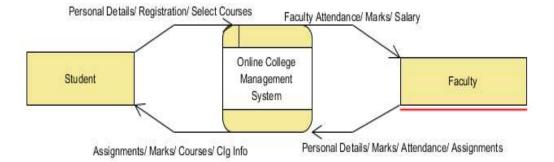


Fig 4.2 Contextual Diagram or Level 0 DFD

Top Level DFD

The Top Level DFD gives the overview of the whole system identifying the major system processes and data flow. This level focuses on the single process that is drawn in the context diagram by 'Zooming in' on its contents and illustrates what it does in more detail.

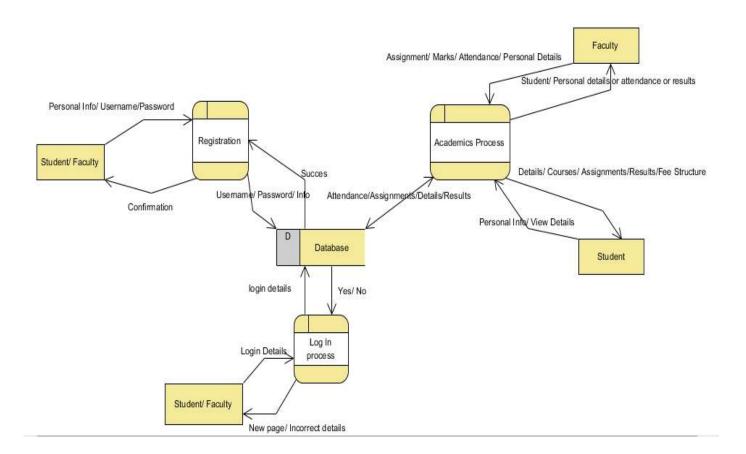


Fig 4.3 Level 1 Data Flow Diagram

4.5 Architecture Diagram:

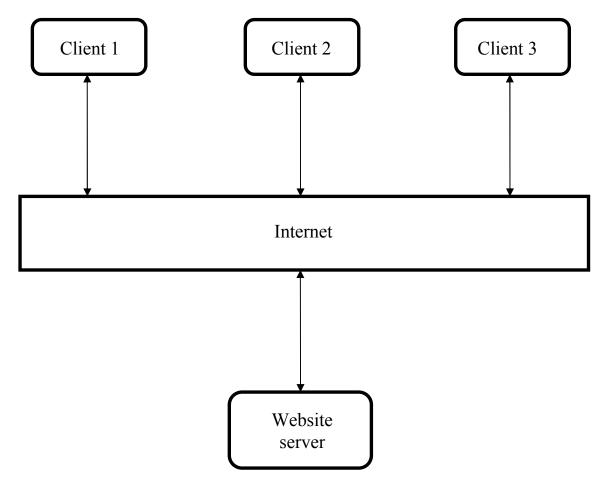


Fig 4.4

This is the architecture diagram of the system and it shows that client-server architecture is used for the development of the system. The communication between client and the server is done with the help of the internet. There can be multiple servers so that they can handle all the clients.

4.6 Database Tables Used:

- Student Table
- Courses table
- Stream Table
- Result Table
- Attendance Table
- Download Table
- News and Events Table
- Assignment Table

Student Table (tbstudent):

stu_id	Int	Primary Key	Student_id
fname	Varchar(50)	Not Null	First Name
lname	Varchar(50)	Not Null	Last Name
Cno	Varchar(50)		Contact Number
Sex	Int	Not Null	Gender
adrs	Varchar(100)	Not Null	Address
email	Varchar(50)	Not Null	Email Address
Dob	Varchar(50)	Not Null	Date of Birth

Courses Table (tbcourse):

c_id	Varchar(10)	Primary Key	Course id
c_name	Varchar(50)	Not Null	Name of the course
s_name	Varchar(10)	Foreign Key	Name of the stream
semno	Int	Not Null and in	Semester Number
		between 1 and 8	

Stream Table (tbstream):

Sid	Int	Primary key and auto	Stream id
		increment	
s_name	Varchar(10)	Not Null	Name of the stream

Result Table (tbresult):

c_id	Varchar(10)	Foreign key	Course id
c_name	Varchar(50)	Foreign key	Name of the course
stu_id	Int	Foreign key	Student_id
semno	Int	Not Null and in	Semester no for the
		between 1 and 8	subject
grades	Char	Not Null	Grade of the student

Attendance Table (tbattendance):

stu_id	Int	Foreign key	Student_id
fname	Varchar(50)	Foreign key	First Name
lname	Varchar(50)	Foreign key	Last Name
c_id	Varchar(10)	Foreign key	Course id
c_name	Varchar(50)	Foreign key	Name of the course
status	Char	Not Null	Present or Absent
timing	Time	Not Null and will always be in between university working hours	Class timing

Download Table:

d_id	int	Primary Key and auto	Id of the file
		increment	
path	Varchar(1000)	Not Null and Unique	Location of the file
Size	Double	Not Null	Size of the file

CHAPTER 5: IMPLEMENTATION

5.1 General

A programming tool or software tool is a program or application that software developers use to create, debug, maintain, or otherwise support other programs and applications. The term usually refers to relatively simple programs that can be combined together to accomplish a task. The Chapter describes about the software tool that is used in our project.

5.2. HTML

HTML, an initialism of Hypertext Markup Language, is the predominant markup language for web pages. It provides a means to describe the structure of text-based information in a document — by denoting certain text as headings, paragraphs, lists, and so on — and to supplement that text with interactive forms, embedded images, and other objects. HTML is written in the form of labels (known as tags), surrounded by angle brackets. HTML can also describe, to some degree, the appearance and semantics of a document, and can include embedded scripting language code which can affect the behavior of web browsers and other HTML processors.

HTML is also often used to refer to content of the MIME type text/html or even more broadly as a generic term for HTML whether in its XML-descended form (such as XHTML 1.0 and later) or its form descended directly from SGML

Hyper Text Markup Language

Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produces Web pages that include text, graphics and pointer to other Web pages (Hyperlinks).

HTML is not a programming language but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preference. A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be

displayed. Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.

HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop.

HTML provides tags (special codes) to make the document look attractive. HTML tags are not case-sensitive. Using graphics, fonts, different sizes, color, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

Basic HTML Tags:

<! --> specifies comments <A>.... Creates hypertext links Formats text as bold <BIG>.....</BIG> Formats text in large font. <BODY>...</BODY> Contains all tags and text in the HTML document <CENTER>...</CENTER> Creates text <DD>...</DD> Definition of a term <DL>...</DL> Creates definition list ... Formats text with a particular font <FORM>...</FORM> Encloses a fill-out form <FRAME>...</FRAME> Defines a particular frame in a set of frames <H#>...</H#> Creates headings of different levels (1-6)<HEAD>...</HEAD> Contains tags that specify information about a document <HR>...</HR> Creates a horizontal rule <HTML>...</HTML> Contains all other HTML tags <META>...</META> Provides meta-information about a document <SCRIPT>...</SCRIPT> Contains client-side or server-side script <TABLE>...</TABLE> Creates a table <TD>...</TD> Indicates table data in a table

<TR>...</TR>

Designates a table row

<TH>...</TH>

Creates a heading in a table

5.2.1 Attributes

The attributes of an element are name-value pairs, separated by "=", and written within the start label of an element, after the element's name. The value should be enclosed in single or double quotes, although values consisting of certain characters can be left unquoted in HTML (but not XHTML). Leaving attribute values unquoted is considered unsafe.

Most elements take any of several common attributes: id, class, style and title. Most also take language-related attributes: lang and dir.

The id attribute provides a document-wide unique identifier for an element. This can be used by style sheets to provide presentational properties, by browsers to focus attention on the specific element or by scripts to alter the contents or presentation of an element. The class attribute provides a way of classifying similar elements for presentation purposes. For example, an HTML document (or a set of documents) may use the designation class="notation" to indicate that all elements with this class value are all subordinate to the main text of the document (or documents). Such notation classes of elements might be gathered together and presented as footnotes on a page, rather than appearing in the place where they appear in the source HTML.

An author may use the style non-attributable codes presentational properties to a particular element. It is considered better practice to use an element's son- id page and select the element with a style sheet, though sometimes this can be too cumbersome for a simple ad hoc application of styled properties. The title is used to attach sub textual explanation to an element. In most browsers this title attribute is displayed as what is often referred to as a tooltip. The generic inline span element can be used to demonstrate these various non-attributes.

The preceding displays as HTML (pointing the cursor at the abbreviation should display the title text in most browsers).

5.2.2 Advantages

- A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.
- HTML is platform independent and case insensitive.

5.3 Database

A database management system (DBMS) is computer software designed for the purpose of managing databases, a large set of structured data, and run operations on the data requested by numerous users. Typical examples of DBMSs include Oracle, DB2, Microsoft Access, Microsoft SQL Server, Firebird, PostgreSQL, MySQL, SQLite, FileMaker and Sybase Adaptive Server Enterprise. DBMSs are typically used by Database administrators in the creation of Database systems. Typical examples of DBMS use include accounting, human resources and customer support systems.

Originally found only in large companies with the computer hardware needed to support large data sets, DBMSs have more recently emerged as a fairly standard part of any company back office.

5.3.1 Description

A DBMS is a complex set of software programs that controls the organization, storage, management, and retrieval of data in a database. A DBMS includes:

- ✓ A modeling language to define the schema of each database hosted in the DBMS, according to the DBMS data model.
 - The four most common types of organizations are the hierarchical, network, relational and object models. Inverted lists and other methods are also used. A given database management system may provide one or more of the four models. The optimal structure depends on the natural organization of the application's data, and on the application's requirements (which include transaction rate (speed), reliability, maintainability, scalability, and cost).
 - The dominant model in use today is the ad hoc one embedded in SQL, despite the objections of purists who believe this model is a corruption of the relational model, since it violates several of its fundamental principles for the sake of practicality and performance. Many DBMSs also support the Open Database Connectivity API that supports a standard way for programmers to access the DBMS.
 - ✓ Data structures (fields, records, files and objects) optimized to deal with very large amounts of data stored on a permanent data storage device (which implies relatively slow access compared to volatile main memory).

- ✓ A database query language and report writer to allow users to interactively interrogate the database, analyze its data and update it according to the users privileges on data.
 - It also controls the security of the database.
 - Data security prevents unauthorized users from viewing or updating the database. Using passwords, users are allowed access to the entire database or subsets of it called sub schemas. For example, an employee database can contain all the data about an individual employee, but one group of users may be authorized to view only payroll data, while others are allowed access to only work history and medical data.
 - If the DBMS provides a way to interactively enter and update the database, as well as
 interrogate it, this capability allows for managing personal databases. However, it
 may not leave an audit trail of actions or provide the kinds of controls necessary in a
 multi-user organization. These controls are only available when a set of application
 programs are customized for each data entry and updating function.
- ✓ A transaction mechanism, that ideally would guarantee the ACID properties, in order to ensure data integrity, despite concurrent user accesses (concurrency control), and faults (fault tolerance).
 - It also maintains the integrity of the data in the database.
 - The DBMS can maintain the integrity of the database by not allowing more than one user to update the same record at the same time. The DBMS can help prevent duplicate records via unique index constraints; for example, no two customers with the same customer numbers (key fields) can be entered into the database. See ACID properties for more information (Redundancy avoidance).

The DBMS accepts requests for data from the application program and instructs the operating system to transfer the appropriate data.

When a DBMS is used, information systems can be changed much more easily as the organization's information requirements change. New categories of data can be added to the database without disruption to the existing system.

Organizations may use one kind of DBMS for daily transaction processing and then move the detail onto another computer that uses another DBMS better suited for random inquiries and

analysis. Overall systems design decisions are performed by data administrators and systems analysts. Detailed database design is performed by database administrators.

Database servers are specially designed computers that hold the actual databases and run only the DBMS and related software. Database servers are usually multiprocessor computers, with RAID disk arrays used for stable storage. Connected to one or more servers via a high-speed channel, hardware database accelerators are also used in large volume transaction processing environments.

DBMSs are found at the heart of most database applications. Sometimes DBMSs are built around a private multitasking kernel with built-in networking support although nowadays these functions are left to the operating system.

5.4 SQL

Structured Query Language (SQL) is the language used to manipulate relational databases. SQL is tied very closely with the relational model.

In the relational model, data is stored in structures called relations or tables.

SQL statements are issued for the purpose of:

Data definition: Defining tables and structures in the database (DDL used to create, alter and drop schema objects such as tables and indexes).

Data manipulation: Used to manipulate the data within those schema objects (DML Inserting, Updating, Deleting the data, and Querying the Database).

A schema is a collection of database objects that can include: tables, views, indexes and sequences

List of SQL statements that can be issued against an Oracle database schema are:

- ALTER Change an existing table, view or index definition (DDL)
- **AUDIT** Track the changes made to a table (DDL)

- **COMMENT** Add a comment to a table or column in a table (DDL)
- **COMMIT** Make all recent changes permanent (DML transactional)
- **CREATE** Create new database objects such as tables or views (DDL)
- **DELETE** Delete rows from a database table (DML)
- **DROP** Drop a database object such as a table, view or index (DDL)
- **GRANT** Allow another user to access database objects such as tables or views (DDL)
- **INSERT** Insert new data into a database table (DML)
- **No AUDIT** Turn off the auditing function (DDL)
- **REVOKE** Disallow a user access to database objects such as tables and views (DDL)
- **ROLLBACK** Undo any recent changes to the database (DML Transactional)
- **SELECT** Retrieve data from a database table (DML)
- TRUNCATE Delete all rows from a database table (can not be rolled back) (DML)
- **UPDATE** Change the values of some data items in a database table (DML)

5.5 Screenshots:

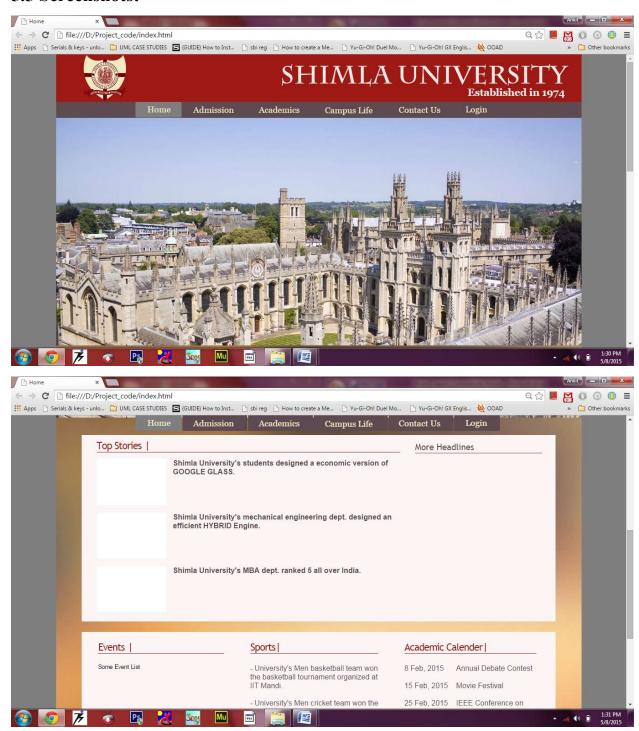
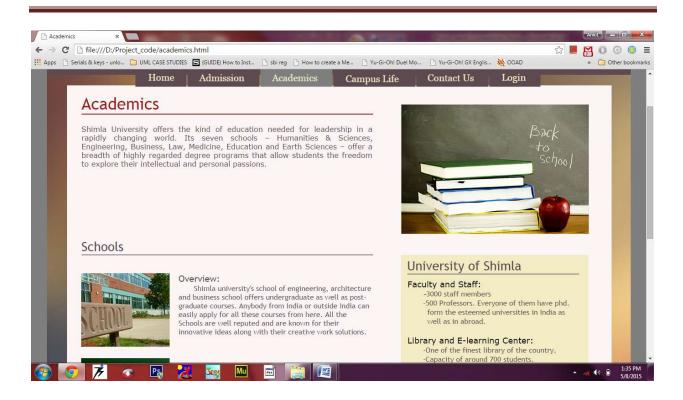


Fig 5.1 (Home page)





Fig 5.2 (Admission page)



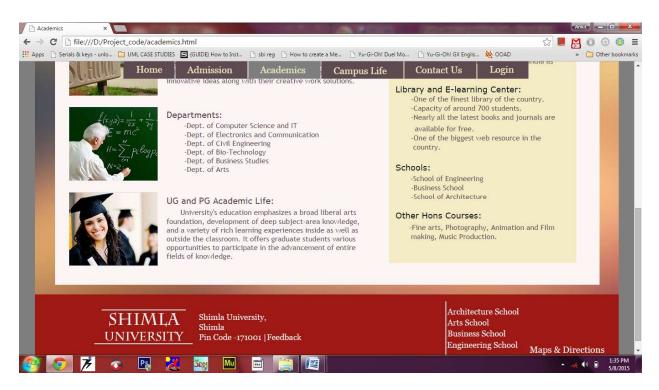
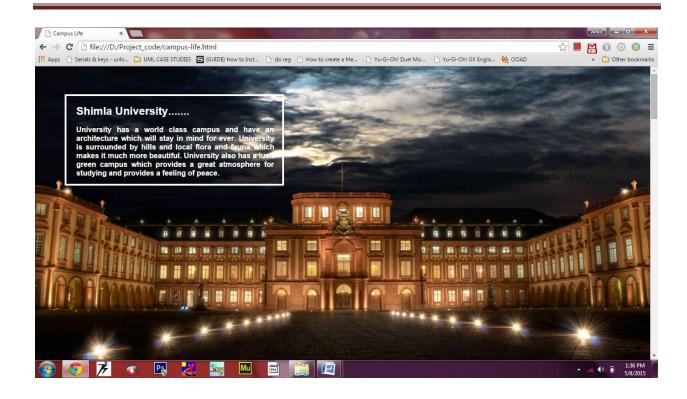


Fig 5.3 (Academics page)



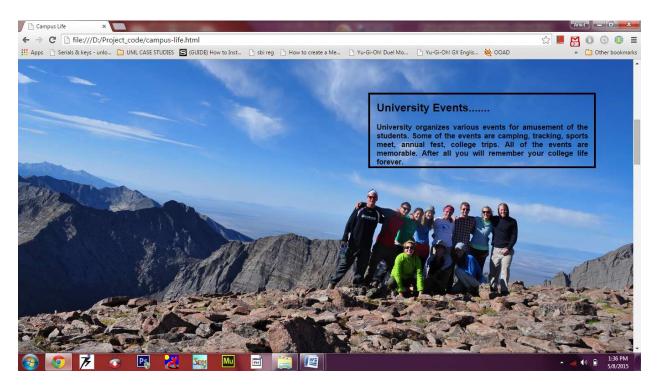
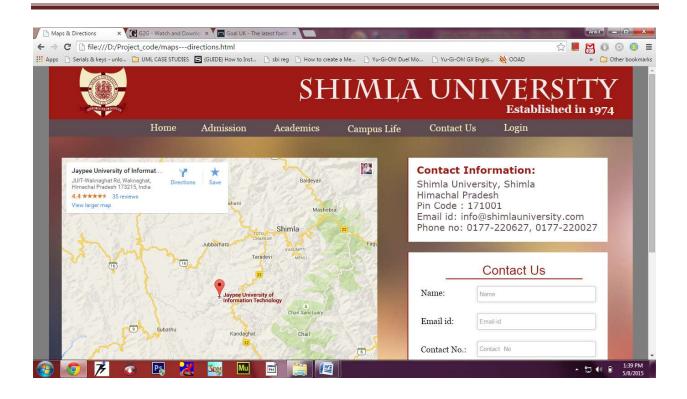


Fig 5.4 (Campus Life page)



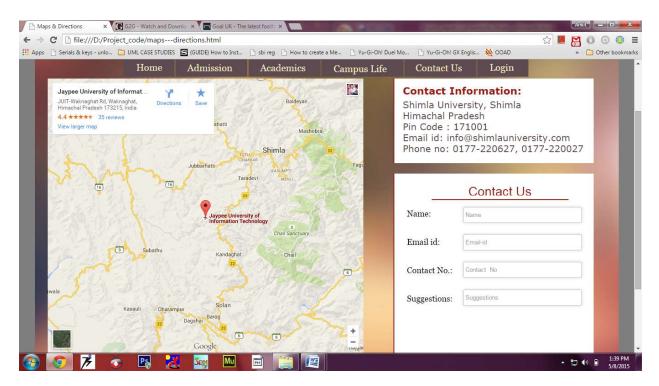
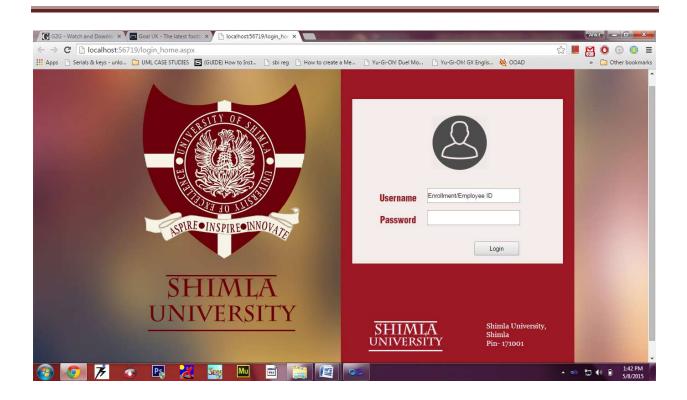


Fig 5.5 (Contact Us or Maps & Directions page)



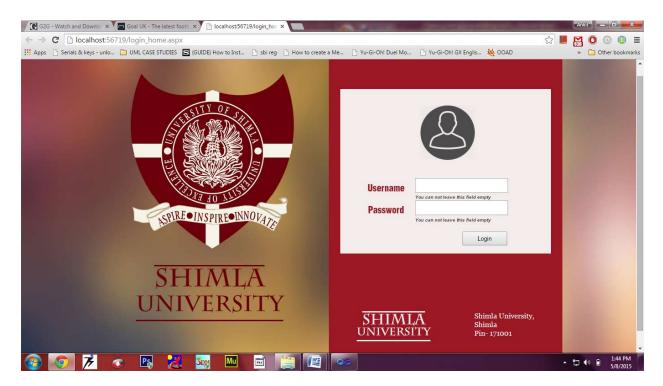


Fig 5.6 (Login page)



Fig 5.7 (Admin Home page)

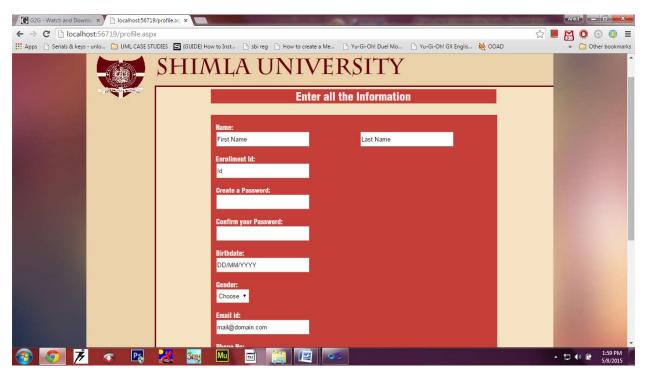


Fig 5.8 (Add Student Details page)



Fig 5.9 (Add Employee Details page)

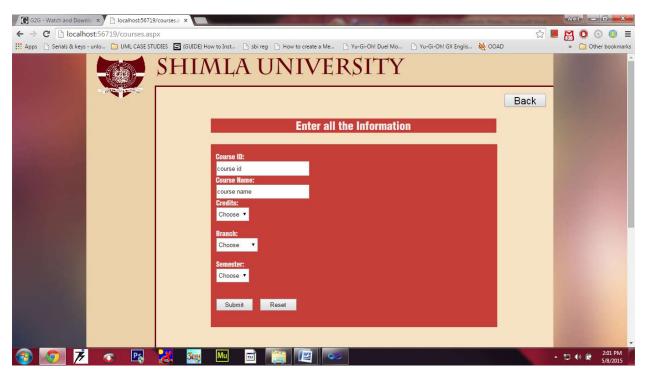


Fig 5.10 (Add Courses Details page)



Fig 5.11 (View Employee Details Page)



Fig 5.12(Students Home Page)



Fig 5.13(Teachers' Home Page)

CHAPTER 6: TESTING AND ENHANCEMENTS

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing is an exposure of a system to trial input to see whether software meets correct output. Testing cannot be determined whether software meets user's needs, only whether it appears to confirm to requirements. Testing can show that a system is free of errors, only that it contains error. Testing finds errors, it does not correct errors. Software success is a quality product, on time and within cost. Through testing can reveal critical mistakes. Testing should therefore,

- ⇒ Validate performance
- □ Detect Errors
- ⇒ Identify inconsistencies

5.1 Test Objective:

There is strong evidence that effective requirement management leads to overall project cost savings. The three primary reasons for this are,

- 1. Requirement errors typically cost well over 10 times more to repair than other errors.
- 2. Requirement errors typically comprise over 40% of all errors in a software project.
- 3. Small reduction in the number of requirement errors pays big dividend in avoided rework costs and schedule delays.

System are not designed as entire systems nor are they tested as single systems the analyst must perform both unit and system testing. For this different levels of testing are used.

5.1.1 Unit Testing:

- In unit testing Module is tested separately and the programmer simultaneously along with the coding of the module performs it.
- In unit testing the analyst tests the programs making up a system. For this reason, unit testing is sometime called program testing. Unit testing gives stress on modules independently of one another, to find errors. This helps the tester in detecting errors in

- coding and logic that are contained within that module alone. The errors resulting from the interaction between modules are initially avoided.
- Unit testing can be performed from the bottom up, Starting with smallest and lowest-level modules and proceeding one at a time., for each module in Bottom-up testing a short program is used to execute the module and provides the needed data, so that the module is asked to perform the way it will when embedded within the larger system.

5.1.2 System Testing:

- This is performed after the system is put together. The system is tested against the system requirement to check if all the requirements are met and if the system performs of specify by the requirements.
- Testing is an important function to the success of the system. System testing makes a
 logical assumption that if all the parts of the system are correct, the goal will be
 successfully activated. Another reason for system testing is its utility as a user-oriented
 vehicle before implementation.

5.1.3 System Testing consists of following steps:

- 1. <u>Program Testing:</u> A program represents the logical elements of a system. For a program to run satisfactorily, it must compile and test data correctly and tie in properly with other programs. It is the responsibility of a programmer to have an error free program. At the time of testing the system, there exist two types of errors that should be checked. These errors are Syntax errors and Logical errors.
 - ⇒ A syntax error is a program statement that violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted key words are common syntax errors. These errors are shown through error messages generated by the Computer.
 - ⇒ A logic error, on the other hand, deals with incorrect data fields out of range items, and invalid combinations. Since the logical errors are not detected by compile, the programmer must examine the output carefully to detect them.

- ➡ When program is tested, the actual output is compared with the expected output. When there is a discrepancy, the sequence of the instructions, must be traced to determine the problem. Breaking the program down into self- contained portion, each of which can be checked at certain key points, facilitates the process.
- 2. <u>String Testing:</u> Programs are invariably related to one another and interact in total system. Each program is tested to see whether it confirms to related programs in the system. Each part of the system is tested against the entire module with both test and live data before the whole system is ready to be tested.
- 3. System Testing: System testing is designed to uncover weaknesses that were not found in earlier tests. This includes forced system failure and validation of total system, as its user in the operational environment will implement it. Under this testing, generally we take low volumes of transactions based on live data. This volume is increased until the maximum level for each transaction type is related. The total system is also tested for recovery and fall back after various major failures to ensure that no data are lost during the emergency.

5.2 Test Cases:

Login Process:

Field	Value	Valid / Invalid	Result
Username	Null	Invalid	Invalid username and
Password	Null	Invalid	password
Username	111288	Valid	You can't leave this
Password	Null	Invalid	empty
Username	111288	Valid	Redirect to next page
Password	Sharma	Valid	
Username	71	Valid	Password should have
Password	123	Invalid	at least 6 characters

Uploading Module:

Field	Value	Valid / Invalid	Result
Upload	Null	Invalid	The field is empty
Upload	Null	Invalid	Assignment Uploaded

Student Record:

Field	Value	Valid / Invalid	Result
First name	Null	Invalid	The field is empty
Last Name	Null	Invalid	The field is empty
Enrollment Id	234	Invalid	Length should be of 6
Password	Sood12	Valid	Password Doesn't
Confirm Password	sood	Invalid	Match
Birth date	sdsdfg	Invalid	Invalid Format
Gender	Male	Valid	
Email id	sood	Invalid	Invalid Format
Phone No.	981660002112	Invalid	Length should be of
			10 characters
Address	VPO Arki	Valid	

Field	Value	Valid / Invalid	Result
First name	Ankit	Valid	
Last Name	Sood	Valid	
Enrollment Id	111288	Valid	
Password	Sood12	Valid	
Confirm Password	Sood12	Valid	
Birth date	21/07/1993	Valid	
Gender	Male	Valid	
Email id	soodankit1993@gmail.com	Valid	

Phone No.	9816600021	Valid	
Address	VPO, Arki	Valid	Details Added to the
			database

Employee Records:

Field	Value	Valid / Invalid	Result
First name	Null	Invalid	The field is empty
Last Name	Null	Invalid	The field is empty
Employee Id	1111	Invalid	Length should be of 3
Password	Sood12	Valid	Password Doesn't
Confirm Password	sood	Invalid	Match
Birth date	sdsdfg	Invalid	Invalid Format
Gender	Male	Valid	
Qualification	Choose	Invalid	No value is selected
Email id	rajeshkumar	Invalid	Not a valid format
Phone No	23213	Invalid	It should be of 10 char
Address	Raja Mohalla	Valid	
Salary	200000	Valid	

Field	Value	Valid / Invalid	Result
First name	Rachit	Valid	
Last Name	Jain	Valid	
Enrollment Id	111	Valid	
Password	Sood12	Valid	
Confirm Password	Sood12	Valid	
Birth date	21/07/1993	Valid	
Gender	Male	Valid	
Qualification	B.tech	Valid	

Email id	soodankit1993@gmail.com	Valid	
Phone No.	9816600021	Valid	
Address	VPO, Arki	Valid	
Salary	200000	Valid	Details Added to the
			database

Courses Records:

Field	Value	Valid / Invalid	Result
Course Id	17B11CI203	Valid	
Course Name	C Programming	Valid	
Credits	Choose	Invalid	Nothing is selected
Branch	Choose	Invalid	Nothing is selected
Semester	Choose	Invalid	Nothing is selected

Field	Value	Valid / Invalid	Result
Course Id	17B11CI203	Valid	
Course Name	C Programming	Valid	
Credits	2	Valid	
Branch	CSE	Valid	
Semester	5	Valid	Course Info added

5.3 Limitations and Enhancements:

Each and every system has some limitations that can restrict then to work on a particular environment. Through our system is provided high amount of accuracy and quick data generation there is still some limitations that led us to in greater deficiency to work on that system. Some of the limitations are following:

- ✓ This system requires knowledgeable person to use this system.
- ✓ The system is not cover the online fees collection.

- ✓ This system doesn't provide library management.
- ✓ The system is web-base application (website) so internet facility must be need.
- ✓ This system is a web-base system so it may slow when more users accessing same interface at same time.
- ✓ For data storage large number of memory covered by system on web.
- User does not view topper of that college.
- ✓ This system does not cover online admission.
- ✓ This system does not cover to pay staff salary.

Enhancements:

- ✓ In future system can include online fees collection, online admission procedure, and good backup facility.
- ✓ The management system can have an android app as the generation of smart phones is here.
- ✓ The management system website can also be designed for the mobile phones as the screen size and resolution varies from mobile to mobile as well as desktop to mobile.
- ✓ The modules like library management system can also be added.
- ✓ The era of big data is coming, So university can implement the big data solutions in order to have cheap and secure backup of the system. The university can also check the previous trends using this and can accordingly design it's courses according to the student's interest and market needs.

References:

- http://fearlessflyer.com/all-you-need-to-know-about-parallax-scrolling-in-webdesign/ accessed on 12/09/2014
- http://www.umsl.edu/~sauterv/analysis/488_f01_papers/quillin.htm accessed on 17/09/2014
- http://www.tutorialspoint.com/uml/ accessed on 25/9/2014
- http://www.ox.ac.uk/ accessed on 9/10/2014
- http://www.stanford.edu accessed on 23/10/2014
- http://msdn.microsoft.com/en-us/library/ms973829.aspx accessed on 12/11/2014
- http://freelinksdirectory.net
- http://www.freesmscode.biz
- http://www.php.net
- http://www.w3schools.com
- http://www.ithacaforward.org
- http://www.skoobe.biz
- http://www.dmegs.com
- Software Engineering(7th edition) by IAN SOMMERVILLE, Pearson Publications
- Software Engineering (5th Edition) by ROGER S. PRESSMAN, TMH publications
- Visual Paradigm UML user guide by Microsoft
- Youtube Channel nemed TerryWhite(official Adobe Developer)

Appendix:

Web.Config:

```
<?xml version="1.0"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
 http://go.microsoft.com/fwlink/?LinkId=169433
  -->
<configuration>
    <connectionStrings>
        <add name="cn" connectionString="Data Source=devil_inside-</pre>
pc\sqlexpress1;Initial Catalog=dbclass;Integrated Security=True;Pooling=False"
            providerName="System.Data.SqlClient" />
    </connectionStrings>
    <system.web>
        <compilation debug="true" targetFramework="4.0" />
    </system.web>
</configuration>
App.master:
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="app.Master.cs"</pre>
Inherits="clg app.Site1" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"</pre>
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
    <title></title>
     <link rel="stylesheet" type="text/css" href="app_master.css" />
    <asp:ContentPlaceHolder ID="head" runat="server">
    </asp:ContentPlaceHolder>
    <style type="text/css">
        .style1
        {
            width: 120px;
            height: 100px;
        }
        .style2
            width: 100%;
            height: 69px;
        }
        .style3
        {
            width: 558px;
            height: 65px;
        }
```

```
.style5
      {
          height: 19px;
      .style6
      {
      }
   </style>
</head>
<body id="bod">
     <div id="wrapperin">
           <div id="nav-wrapper">
                <div id="navigation">
                      <center>
                <img alt="Shimla University" class="style1"</pre>
src="Images/main_logo34.jpg" /><br />
             </center>
                </div>
           </div>
           <div id="section">
                <div id="head-wrapper">
                      <div id="header">
                         <font size="13" id="title">Shimla
<asp:ContentPlaceHolder ID="ContentPlaceHolder3"</pre>
runat="server">
                              </asp:ContentPlaceHolder>
                          </div>
                </div>
                <div id="content">
          <form id="form2" runat="server">
                   <asp:ContentPlaceHolder ID="ContentPlaceHolder2"</pre>
runat="server">
             </asp:ContentPlaceHolder>
          </form>
                </div>
```

```
<div id="footer">
                           <center>Copyrights of Shimla University</center>
                    </div>
             </div>
      </div>
</body>
</html>
App_master.css:
#bod{
             background-image: url("Images/Pepsized_Blur_new2.jpg");
             background-repeat: no-repeat;
        top: 0;
             left:0;
             margin-top: 0;
             margin-bottom: 0;
      }
      #wrapperin{
             width:1024px;
             height: 1070px;
             margin: 0 auto 0;
      }
      #nav-wrapper{
             position: fixed;
             float: left;
             height:695px;
             width:150px;
      }
      #navigation{
             height:100%;
             width:100%;
             background-color: #ebd1a9;
             float:left;
      }
      #section{
             height:1070px;
             width:874px;
             float: right;
      }
      #head-wrapper{
             position: fixed;
             height:70px;
             width:874px;
      }
```

```
#header{
             height:100%;
             width:100%;
             background-color: #ebd1a9;
             font-family: Charlemagne Std;
             color: #6c010b;
             border-bottom-style: solid;
             border-color: #6c010b;
      }
      #content{
             margin-top: 70px;
             height:970px;
             width:871px;
             background-color: #f6e1c2;
             border-left-style: solid;
             border-color: #6c010b;
      }
      #footer{
             height:30px;
             width:870px;
             background-color: #ebd1a9;
             border-style: solid;
             border-color: #6c010b;
      }
.myButton {
      -moz-box-shadow:inset Opx 1px Opx Opx #ffffff;
      -webkit-box-shadow:inset Opx 1px Opx Opx #ffffff;
      box-shadow:inset Opx 1px Opx Opx #ffffff;
      background:-webkit-gradient(linear, left top, left bottom, color-stop(0.05,
#fcfcfc), color-stop(1, #e8e3e8));
      background:-moz-linear-gradient(top, #fcfcfc 5%, #e8e3e8 100%);
      background:-webkit-linear-gradient(top, #fcfcfc 5%, #e8e3e8 100%);
      background:-o-linear-gradient(top, #fcfcfc 5%, #e8e3e8 100%);
      background:-ms-linear-gradient(top, #fcfcfc 5%, #e8e3e8 100%);
      background:linear-gradient(to bottom, #fcfcfc 5%, #e8e3e8 100%);
      filter:progid:DXImageTransform.Microsoft.gradient(startColorstr='#fcfcfc',
endColorstr='#e8e3e8',GradientType=0);
      background-color:#fcfcfc;
      -moz-border-radius:15px;
      -webkit-border-radius:15px;
      border-radius:15px;
      border:1px solid #dcdcdc;
      display:inline-block;
      cursor:pointer;
      color:#666666;
      font-family:Arial;
      font-size:15px;
      font-weight:bold;
      padding:6px 24px;
      text-decoration:none;
      text-shadow:0px 1px 0px #ffffff;
```

```
.myButton:hover {
      background:-webkit-gradient(linear, left top, left bottom, color-stop(0.05,
#e8e3e8), color-stop(1, #fcfcfc));
      background:-moz-linear-gradient(top, #e8e3e8 5%, #fcfcfc 100%);
      background:-webkit-linear-gradient(top, #e8e3e8 5%, #fcfcfc 100%);
      background:-o-linear-gradient(top, #e8e3e8 5%, #fcfcfc 100%);
      background:-ms-linear-gradient(top, #e8e3e8 5%, #fcfcfc 100%);
      background:linear-gradient(to bottom, #e8e3e8 5%, #fcfcfc 100%);
      filter:progid:DXImageTransform.Microsoft.gradient(startColorstr='#e8e3e8',
endColorstr='#fcfcfc',GradientType=0);
      background-color:#e8e3e8;
}
.myButton:active {
      position:relative;
      top:1px;
}
.myButton1 {
      -moz-box-shadow:inset Opx Opx 15px 3px #23395e;
      -webkit-box-shadow:inset Opx Opx 15px 3px #23395e;
      box-shadow:inset 0px 0px 15px 3px #23395e;
      background:-webkit-gradient(linear, left top, left bottom, color-stop(0.05,
#2e466e), color-stop(1, #415989));
      background:-moz-linear-gradient(top, #2e466e 5%, #415989 100%);
      background:-webkit-linear-gradient(top, #2e466e 5%, #415989 100%);
      background:-o-linear-gradient(top, #2e466e 5%, #415989 100%);
      background:-ms-linear-gradient(top, #2e466e 5%, #415989 100%);
      background:linear-gradient(to bottom, #2e466e 5%, #415989 100%);
      filter:progid:DXImageTransform.Microsoft.gradient(startColorstr='#2e466e',
endColorstr='#415989',GradientType=0);
      background-color:#2e466e;
      -moz-border-radius:17px;
      -webkit-border-radius:17px;
      border-radius:17px;
      border:1px solid #1f2f47;
      display:inline-block;
      cursor:pointer;
      color:#ffffff;
      font-family:Arial;
      font-size:15px;
      padding:6px 13px;
      text-decoration:none;
      text-shadow:0px 1px 0px #263666;
.myButton1:hover {
      background:-webkit-gradient(linear, left top, left bottom, color-stop(0.05,
#415989), color-stop(1, #2e466e));
      background:-moz-linear-gradient(top, #415989 5%, #2e466e 100%);
      background:-webkit-linear-gradient(top, #415989 5%, #2e466e 100%);
      background:-o-linear-gradient(top, #415989 5%, #2e466e 100%);
      background:-ms-linear-gradient(top, #415989 5%, #2e466e 100%);
      background:linear-gradient(to bottom, #415989 5%, #2e466e 100%);
      filter:progid:DXImageTransform.Microsoft.gradient(startColorstr='#415989',
endColorstr='#2e466e',GradientType=0);
```

```
background-color:#415989;
}
.myButton1:active {
      position:relative;
      top:1px;}
Stu home.aspx:
<%@ Page Title="" Language="C#" MasterPageFile="~/app.Master" AutoEventWireup="true"</pre>
CodeBehind="stu_home.aspx.cs" Inherits="clg_app.WebForm6" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
<style type="text/css">
        .p_style
            margin-left:120px;
            margin-right:120px;
            text-align: center;
            font-family:helvetica;
            font-size: 25px;
            background-color: #C73D38;
            color:#F4EBEB;
        width: 626px;
            font-weight: 700;
        }
         #form style
            margin-bottom:120px;
            margin-left:120px;
            margin-right:120px;
            background-color: #C73D38;
            color:#F4EBEB;
            width: 628px;
    }
        .style4
            width: 547px;
            font-family: "Helvetica Condensed Black";
            font-size: large;
        }
        .style5
        {
            font-size: xx-large;
        }
    </style>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder3" runat="server">
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolder2" runat="server">
   <br />
```

&

```
;            
                                                                                                                                                                                                                                                                                                                                                     
nbsp;                                                                                                                                                                                                                                                                                                                                                    &
bsp;                
sp;            
p;            
anbsp;                        
                                                                                                                                                                                                                                                                                                                                                     
nbsp;  
    <asp:Button ID="LinkButton1" runat="server" Font-Size="22px"</pre>
        onclick="LinkButton1_Click" Height="39px" Text="Logout" Width="104px"
        CssClass="myButton1"/>
    <br /><br />
    <span class="style5">Welcome !!! </span><br />
    <span class="style4">&nbsp;Choose From the following:</span><br /><br />
         
        <asp:Button ID="Button1" runat="server" Text="Edit Profile" Width="164px"</pre>
Height="36px" CssClass="myButton" />
       
        <asp:Button ID="Button3" runat="server" Height="36px" Text="Pre-Registration"</pre>
            Width="164px" CssClass="myButton" />
           
        <asp:Button ID="Button6" runat="server" Height="36px" Text="Assignment Sub"</pre>
            Width="164px" CssClass="myButton" />
      
    <br /><br />
         
        <asp:Button ID="Button4" runat="server" Text="View Attendance" Width="164px"</pre>
            Height="36px" CssClass="myButton" />
           
        <asp:Button ID="Button5" runat="server" Text="View Results" Width="164px"</pre>
            Height="36px" CssClass="myButton" />
           
        <asp:Button ID="Button7" runat="server" Height="36px" Text="View Time-Table"</pre>
            Width="164px" CssClass="myButton" />
    <br /><br />
    <br />
    </asp:Content>
Emp home.aspx:
<%@ Page Title="" Language="C#" MasterPageFile="~/app.Master" AutoEventWireup="true"</pre>
CodeBehind="emp home.aspx.cs" Inherits="clg app.WebForm7" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
```

```
<style type="text/css">
        .p_style
            margin-left:120px;
            margin-right:120px;
            text-align: center;
            font-family:helvetica;
            font-size: 25px;
            background-color: #C73D38;
            color:#F4EBEB;
        width: 626px;
            font-weight: 700;
        }
         #form style
            margin-bottom:120px;
            margin-left:120px;
            margin-right:120px;
            background-color: #C73D38;
            color:#F4EBEB;
            width: 628px;
   }
        .style4
        {
            width: 547px;
            font-family: "Helvetica Condensed Black";
            font-size: large;
        }
        .stvle5
        {
            font-size: xx-large;
        }
    </style>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder3" runat="server">
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolder2" runat="server">
    <br />
```

&

```
<asp:Button ID="LinkButton1" runat="server" Font-Size="22px"</pre>
onclick="LinkButton1 Click" Height="29px" Text="Logout" Width="91px"/>
   <br /><br />
   <span class="style5">Welcome !!! </span><br />
   <span class="style4">Choose From the following:</span><br /><br />
        
      <asp:Button ID="Button1" runat="server" Text="Add Attendance Record"</pre>
Width="164px" Height="36px" />
       
      <asp:Button ID="Button3" runat="server" Height="36px" Text="Add Results"</pre>
Width="164px" />
           
      <asp:Button ID="Button6" runat="server" Height="36px" Text="Add Assignments"</pre>
Width="164px" />
     
   <br /><br />
        
      <asp:Button ID="Button4" runat="server" Text="View Attendance" Width="164px"
Height="36px" />
          
      <asp:Button ID="Button5" runat="server" Text="View Assignments" Width="164px"</pre>
Height="36px" />
          
      <asp:Button ID="Button7" runat="server" Height="36px" Text="Add Course</pre>
Material" Width="164px" />
   <br /><br />
   <br />
   </asp:Content>
WebForm4.aspx:
<%@ Page Title="" Language="C#" MasterPageFile="~/app.Master" AutoEventWireup="true"</pre>
CodeBehind="WebForm4.aspx.cs" Inherits="clg_app.WebForm4" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
   <style type="text/css">
      .p_style
         margin-left:120px;
         margin-right:120px;
          text-align: center;
          font-family:helvetica;
          font-size: 25px;
          background-color: #C73D38;
```

```
color: #F4EBEB:
          width: 626px;
               font-weight: 700;
          }
           #form style
               margin-bottom:120px;
               margin-left:120px;
               margin-right:120px;
               background-color: #C73D38;
                color:#F4EBEB;
               width: 628px;
     }
          .style4
          {
               width: 547px;
               font-family: "Helvetica Condensed Black";
               font-size: large;
          }
          .style5
          {
               font-size: xx-large;
          }
     </style>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder3" runat="server">
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolder2" runat="server">
     <br />
              
nbsp;                                                                                                                                                                                                                                                                                                                                                    &
bsp;                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       &nbs
sp;             
p;            
;            
                                                                                                                                                                                                                                                                                                                                                     
nbsp;                                                                                                                                                                                                                                                                                                                                                    &
bsp;             
sp;             
p;            
;            
             
nbsp; 
     <asp:Button ID="LinkButton1" runat="server" Font-Size="22px"</pre>
          onclick="LinkButton1_Click" Height="40px" Text="Logout" Width="103px"
          CssClass="myButton1"/>
     <br /><br />
     <span class="style5">Welcome !!! Admin</span><br />
     <span class="style4">Choose From the following:</span><br /><br />
```

```
   
       <asp:Button ID="Button1" runat="server" Text="Add Student Rec" Width="185px"</pre>
           Height="36px" onclick="Button1_Click" CssClass="myButton" />
          
       <asp:Button ID="Button3" runat="server" Height="36px"</pre>
           Text="Add Employee Rec" Width="185px" onclick="Button3 Click"
           CssClass="myButton" />
          
       <asp:Button ID="Button6" runat="server" Height="36px"</pre>
           Text="Add Courses" Width="185px" onclick="Button6_Click"
           CssClass="myButton" />
     
   <br /><br />
          
       <asp:Button ID="Button4" runat="server" Text="View Student Rec" Width="185px"</pre>
          Height="36px" onclick="Button1_Click" CssClass="myButton" />
      
       <asp:Button ID="Button5" runat="server" Text="View Employee Rec"</pre>
Width="185px"
          Height="36px" onclick="Button5 Click" CssClass="myButton" />
            <asp:Button ID="Button7" runat="server"</pre>
Height="36px"
           Text="Add Reg Details" Width="185px" onclick="Button7_Click"
          CssClass="myButton" />
   <br /><br />
   <br />
   </asp:Content>
```