P-3223

K C T
LIBRARY
COIMBATORE. 641 006

KCT WEBSITE IMPLEMENTATION

(ACADEMIC MODULE)

PROJECT REPORT

Submitted By

M.GEETHA

Register No.: 0720300006

in partial fulfillment for the award of the degree

of

MASTER OF COMPUTER APPLICATIONS

in

COMPUTER APPLICATIONS

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Coimbatore)

May, 2010

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Coimbatore)

COIMBATORE – 641 006.

Department of Computer Applications

PROJECT WORK

MAY 2010

This is to certify that the project entitled

KCT WEBSITE IMPLEMENTATION

(ACADEMIC MODULE)

is the bonafide record of project work done by

M.GEETHA

Register No: 0720300006

of MCA (Computer Applications) during the year 2009-2010.

Project Guide

Head of the Department

Submitted for the Project Viva-Voce examination held on 17.5.2010

17/5/10

17/05/2010

Internal Examiner

External Examiner

DECLARATION

I affirm that the project work titled **KCT WEBSITE IMPLEMENTATION (ACADEMIC MODULE)** being submitted in partial fulfilment for the award of **MASTER OF COMPUTER APPLICATIONS** is the original work carried out by me. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other University.

M. Geetha

(Signature of the Candidate)

M. GEETHA

Name of the Candidate

Register Number

0720300006

I certify that the declaration made above by the candidate is true

Signature of the Guide,

With Name & Designation

Dr. A. MUTHUKUMAR

PROF.

Estd. 1984

KUMARAGURU COLLEGE OF TECHNOLOGY

[An Autonomous Institution]
(Approved by AICTE / Affiliated to Anna University, Coimbatore / Accredited by NBA & NAAC)
POST BOX No. 2034 – COIMBATORE - 641 006

ISO 9001-2008 Certified

DEPARTMENT OF COMPUTER APPLICATIONS

Apr. 30, 2010.

CERTIFICATE

This is to certify that Ms.M.Geetha(07MCA06) has undertaken a project with us entitled "KCT Website Implementation – Academic Module" and completed successfully on Apr. 30, 2010. She is a sincere and hardworking project student and reported regularly about her progress.

Course Coordinator,
Dept. of Computer Applications

Phone : 0422 - 2669401 to 405 Fax : 0422 - 2669406
E-mail : info@kct.ac.in / website : www.kct.ac.in

ABSTRACT

This software system will be a "**KCT Website – Academic Module**" designed for Academics, Admission, Publications, Library, Calendra, Student chapter and faculty chapter.

In Kumaraguru College of Technology, semester pattern is followed. An academic year consists of two semesters.

Academics focuses on the process carried to fulfill the needs of staff and student in a college by maintaining their details. The system identifies user based on their username and provides their authorized menu for further process.

Admission for UG,PG and PhD Programmes should be maintained. KCT Management, Staff and Students publish a large number of technical and non-technical publications through the year. The Annual Magazine "KUMARAGURU" and the quarterly KCT newsletter are intended to record and highlight the events and developments of the entire campus.And also student are encouraged to publish papers in peer-viewed national and international conferences and journals. In addition, various departments of the college also bring out newletters highlighting the important events, achievements and activities of the respective disciplines.

In Library,

- A modern digital library network with high speed internet access is also housed in the premises.
- Reprographic facilities with photocopier, Document Scanner, Printer & CD writer are provided for.
- Separate sections for Reference Material, Book Bank, Periodicals, Project Reports are provided for easy access.
- UDC coding, Bar coding, Magnetic sensors and an integrated library management software facilitate faster processing.

The system is a client-server based application and can be accessed from different machine in a network. It manages staff and student operations carried in one department. This web-based application is developed using PHP with HTML, MySQL, and Apache. The system can be used in either internet or intranet environment.

ACKNOWLEDGEMENT

First and foremost I thank the Almighty for his continuous blessings showered on me in completing this project successfully.

I wish to express my profound gratitude to **Dr. S. Ramachandran, Principal, Kumaraguru College of Technology, Coimbatore** for providing an opportunity and necessary facilities in carrying out this project work.

I also take this opportunity to thank my Coordinator **V.Geetha M.C.A., Asst. Professor, Kumaraguru College of Technology, Coimbatore** for her continuous support.

I wish to express my unending gratitude to **Dr. A. Muthukumar M.Sc., M.C.A., M.Phil., PhD, Department of Computer Applications** for his unrelenting help, support and advice with regards to my project work. I owe him my profound admiration and respect.

I also dedicate equal and grateful acknowledgements to all the respectable members of the faculty and lab in-charges of the **Department of Computer Applications, Kumaraguru College of Technology, Coimbatore** and student friends for their motivation, encouragement and continuous support.

LIST OF TABLES

FIGURE NO	TITLE	PAGE NO
4.2.1	subjects	16
4.2.2	login	16
4.2.3	staff_details	17
4.2.4	student_details	18
4.2.5	staff_notes	19
4.2.6	attendance	19
4.2.7	attendance_details	20
4.2.8	overall_attendance	20
4.2.9	marks	21
4.2.10	projects_current	22
4.2.11	staff_subjects_handled	23
4.2.12	degree	23
4.2.13	electives	24
4.2.14	student_electives	24
4.2.15	alumni_details	25

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
4.1.1	Login Use Case	10
4.1.2	Admin Use Case	11
4.1.3	Student Use Case	13
4.1.4	Staff Use Case	14
9.2.1	KCT Home Page	43
9.2.2	Academic	44
9.2.3	Admission	45
9.2.4	Academic Schedule	46
9.2.5	Login	47
9.2.6	Add Student Details	48
9.2.7	Student Search	49
9.2.8	New Semester Elective	50
9.2.9	Staff allocation	51
9.2.10	Post Remainder	52
9.2.11	Staff Home	53
9.2.12	Attendance entry	54
9.2.13	Marks entry	55
9.2.14	Project details	56
9.2.15	Student home	57
9.2.16	Attendance View	58
9.2.17	View Marks	59
9.2.18	Internal Marks	60

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	LIST OF TABLES	**vi**
	LIST OF FIGURES	**vii**
1.	**Introduction**	**1**
	1.1 Organization Profile	**1**
2.	**System Analysis**	**2**
	2.1 Existing System	**2**
	2.2 Proposed System	**3**
	2.3 User Interface Requirements	**3**
	2.3.1. Functional Requirements	4
	2.3.2. Non Functional (System) Requirements	5
	2.4 Users of the System	**5**
3.	**Development Environment**	**6**
	3.1 Hardware Requirements	**6**
	3.2 Software Requirements	**6**
	3.3 Programming Environment	**7**
	3.3.1 PHP	7
	3.3.2 MySQL	7
	3.3.3 Apache	8
	3.3.4 HTML	8
	3.3.5 Javascript	9
4	**System Design**	**10**
	4.1 Process Model	**10**
	4.1.1 Use Case Diagram	10

	4.1.1.1 Use Case (Login Module)	10
	4.1.1.2 Use Case (Admin Module)	11
	4.1.1.3 Use Case (Student Module)	13
	4.1.1.4 Use Case (Staff Module)	14
	4.2 Database Design	16
5.	**Implemention**	26
	5.1 System Verification	26
	5.2 System Validation	26
6.	**Testing**	27
	6.1 Unit Testing	27
	6.2 Integration Testing	27
	6.3 System Testing	28
7.	**Conclusion**	29
8.	**Future Enhancement**	30
9.	**Appendix**	31
	9.1 Source Code	31
	9.2 Screen Shots	43
10.	**References**	61

CHAPTER 1

INTRODUCTION

1.1 ORGANIZATION PROFILE

Kumaraguru College of Technology (KCT), Coimbatore is a private Engineering College started in **1984** under the auspices of Ramanandha Adigalar Foundation, a charitable educational trust of Sakthi Group. Located at 2 Kms off the Coimbatore-Mysore national highway (NH-209), the congenial and well-developed college campus spreads over a sprawling 150 acres of land.

Under the able guidance and patronage of **Arutselvar Dr. N. Mahalingam, Chairman**, Sakthi Group, the college has developed excellent facilities and resources such as spacious classrooms, seminar halls, well-equipped laboratories, workshops, well-stocked libraries, dedicated highspeed broadband Internet connectivity and well-qualified faculty. The management has invested more than Rs. 800 million for the creation of state of the art infrastructural facilities.

Currently the college offers 12 under-graduate **(B.E., B.Tech.)**, 13 post-graduate **(M.E., M.Tech., MCA, MBA)** as an Autonomous institution affiliated to the Anna University, Coimbatore. The college has the approval of **All India Council for Technical Education** (AICTE) and all 10 eligible UG programs have also been **accredited by National Board of Accreditation** (NBA). In addition, KCT has also been accredited by **National Assessment and Accreditation Council** (NAAC) of UGC.

KCT IT TECH PARK

KCT is the one of the forerunners and service providers to fulfill the growing needs in the academic and technical arena. The STPI has entered into an agreement with KCT for establishing Software Technology Parks in KCT campus for the purpose of setting up IT projects. Through their venture, **Cognizant Technology Solutions India, Spheris India, Cordys India, Ford India, First source and Vignani Technologies**, the six major IT giants have set their projects in KCT campus.

CHAPTER 2
SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The existing KCT web site is not user friendly. It maintains details which were developed few years back. It does not concentrate on the new modules in each department. Those details are manually done by the administrator. Separate records, documents or files need to be managed for maintaining personal details, for students and staff, library, admission, academic, etc. KCT website details need to be maintained separately.

PROBLEMS IN EXISTING SYSTEM

- KCT Website is not user friendly.
- Information is not well organized.
- Search option is not provide inside the website.
- In existing system, Admission date for each year is not provide.
- In existing system, maintenance of both staff and students' details for UG and PG are done manually.
- Attendance and marks Procedure for UG and PG of each student are maintained manually, which needed lot of paper work or space, in case if it is maintained in computer.
- Calculations for internal marks and overall attendance percentage are done manually.
- While entering records, careless mistakes or error might occur.

2.2 PROPOSED SYSTEM

In the proposed system, administrator maintains overall process carried in a department and other management details. All details in the existing project are stored in database for future reference and alterations. Administrator maintains overall process carried in a department especially adding new student or staff, staff allotment for each semester. Staff will enter attendance and marks for students and those entries are stored in the database. Students can view only the entries entered by staff under their roll number.

ADVANTAGES OF PROPOSED SYSTEM

- KCT Website are user friendly and understandable by all users.
- Information are well organized, Search option is provide inside the website.
- In existing system, details maintenance is computerized.
- The system supports client-server architecture by which it can be accessed from any client system in the network.
- Attendance and marks procedure for UG and PG are separately maintained.
- Automatic calculation of overall attendance and internal marks for each subject.
- Searching of staff or students' details was made easy.
- Access limited to authenticated users.;l
- Easily accommodates updates to the records.

2.3 USER INTERFACE REQUIREMENTS

The User Interfaces are designed using PHP, HTML, and Javascript. The interfaces would be designed in a user friendly manner with less complexity so that it proves to be useful for the users to navigate through the system. All the modules are available as horizontal menus so that the user can easily navigate through the pages.

2.3.1 Functional Requirements

Functional requirements are those that refer to the functionality of the system, i.e., what service it will provide to the user.

1) The system has the following type of users:

    1) End-Users.
    2) System administrator.

2) The users can perform the following tasks:

    End-Users:

    - Enter Student information.
    - Print their information as a word Document.

    System Administrator:

    - Enter College information.
    - Print their information as a word Document.
    - Update College Events and Publications.

3) The system has the following Interfaces:

    1) Login
        a) End-User
        b) System Administrator
    2) Interface for filling up the following information
        a) Users – View KCT details
        b) Staff – View and update students details.
        c) Student –View attendance and mark, Library detail.
    3) Interface to print the status in Word Format.
    4) Logout

4) Come up with the database design. The Login to the system is database driven i.e. a Admin will be allowed to access the system if and only if he is the valid user of the system .

5) Decide a format in which you want the status to get printed in word document.

2.3.2 Non Functional (System) Requirements

Non-functional (supplementary) requirements pertain to other information needed to reduce the correct system and are detailed separately.

Usability

- The website is expected to be user friendly. The user must be able to operate the website with ease.
- This website is developed with user friendly interface screens.

Reliability

- The website is expected to operate as close to 100 percentage reliability.
- The website must have no defects that can interface with normal operations.

Performance

- The website is accepted to respond to the users request(s) with minimum time.

Security

The server on which the KCT Website Implementation ( ) resides will have its own security to prevent unauthorized write/delete and access. There are an no restrictions to view the system. The use of email by an administrator /user is on the client system and thus is external to the system.

2.4 USERS OF THE SYSTEM

User Name	Description
Administrator	Manages and coordinates the website. Has full and access to the system.
Staff	Staff will enter attendance and marks for students
Student	Students can view only the entries entered by staff under their roll number such as their attendance and mark details
End user	The users those who access the website

CHAPTER 3

DEVELOPMENT ENVIRONMENT

3.1 HARDWARE REQUIREMENTS

Client side Configuration

Processor	:	Pentium III
Processor Speed	:	1.4 GHz
Memory (RAM)	:	128 MB
Hard Disk	:	40 GB
Floppy Drive	:	3 ½ " 1.44 MB Drive
Monitor	:	SAMTRON Color Monitor
Keyboard	:	117 keys Samsung
Mouse	:	Logitech

Server side Configuration

Processor	:	Pentium IV
Processor Speed	:	1.7 GHz
Memory (RAM)	:	256 MB
Hard Disk	:	40 GB
Floppy Drive	:	3 ½ " 1.44 MB Drive
Monitor	:	SAMTRON Color Monitor
Keyboard	:	117 keys Samsung
Mouse	:	Logitech

3.2 SOFTWARE REQUIREMENTS

Front End	:	PHP
Back End	:	MySQL
Server	:	Apache
Operating System	:	Windows XP

3.3 PROGRAMMING ENVIRONMENT

3.3.1 PHP

PHP is a programming language devised by Rasmus Lerdorf in 1994 for building dynamic, interactive Web sites. PHP is a server-side scripting language that allows your Web site to be truly dynamic. PHP stands for *PHP: Hypertext Preprocessor*. Its flexibility and relatively small learning curve (especially for programmers who have a background in C, Java, or Pearl) make it one of the most popular scripting languages around. PHP's popularity continues to increases as businesses and individuals everywhere embrace it as an alternative to Microsoft's ASP language and realize that PHP's benefits most certainly outweigh the costs. According to Zend Technologies, Ltd., the central source of PHP improvements and designers of the Zend Engines, which supports PHP applications, PHP code can now be found in approximately 9 million Web sites.

PHP's main use is as a cross-platform, HTML-embedded server-side Web scripting language. Let's take a moment to examine these terms:

- **Cross-platform:** Most PHP code can be processed without alteration on computers running many different operating systems. For example, a PHP script that runs on Linux generally also runs well on Windows.
- **HTML-embedded:** PHP code can be written in files containing a mixture of PHP instructions and HTML code.
- **Server-side:** The PHP programs are run on a server – specifically a Web server.
- **Web-scripting language:** PHP programs run via a Web browser.

3.3.2 MySQL

MySQL is a Structured Query Language server designed for heavy loads and processing of complex queries. As s relational database system, MySQL allows many different tables to be joined together for maximum efficiency and speed.

The most popular features of this program are as follows:

- Multiple CPUs usable through kernel threads.
- Multi-platform operation.
- Numerous column types cover virtually every type of data.

- Group functions for mathematical calculations and sorting.
- Commands that allow information about the databases to be easily and succinctly shown to the administrator.
- Function names that do not affect table or column names.
- A password and user verification system for added security.
- Up to 32 indices per table permitted; this feature has been successfully implemented at levels of 60,000 tables and 5,000,000,000 rows.
- International error reporting usable in many different countries.

3.3.3 Apache

Apache acts as Web server. Its main job is to parse any file requested by a browser and display the correct results according to the code within that file. Apache is quite powerful and can accomplish virtually any task that require.

The features and server capabilities available include the following:

- Password-protected pages for a multitude of users.
- Customized error pages.
- Display of code in numerous levels of HTML, and the capability to determine at what level the browser can accept the content.
- Usage and error logs in multiple and customized formats.
- Virtual hosting for different IP addresses mapped to the same server.
- DirectoryIndex directives to multiple files.
- URL aliasing or rewriting with no fixed limit.

3.3.4 HTML

HTML (Hypertext Markup Language) is the language you use to create Web pages. HTML is generally viewed as nothing more than a document formatting, or tagging, language. The tags instruct a viewer program (the *browser* or, more generically, the *client*) how to display chunks of text or images.

Version 4.0 and later of the published HTML standards endeavor too define the purpose of HTML as assigning context to content, leaving the appearance to a sequence standard for stylesheets.

To install a Web page like one.html on the World Wide Web, you use an internet Service Provider (ISP), such as America Online (AOL), CompuServe, or any one of thousands of local companies. The actual installation process depends on your ISP.

3.3.5 JavaScript

In early December 1995, Netscape and Sun Microsystems jointly announced a scripting language known as JavaScript. Microsoft acknowledged the potential power and popularity of the language by implementing it in IE3. Newer browser makers automatically provided support for JavaScript. When employed on the client computer, the language can help turn a static page of content into an engaging, interactive, and intelligent experience. JavaScript provides following kinds of solutions:

- Getting your Web page to respond or react directly to user interaction with form elements and hypertext links.
- Distributing small collections of database-like information and providing a friendly interface to that data.
- Controlling multiple-frame navigation, plug-ins, or Java applets based on user choices in the HTML document.
- Preprocessing data on the client before submission to a server.
- Changing content ant styles in modern browsers dynamically and instantly in response to user interaction.

CHAPTER 4

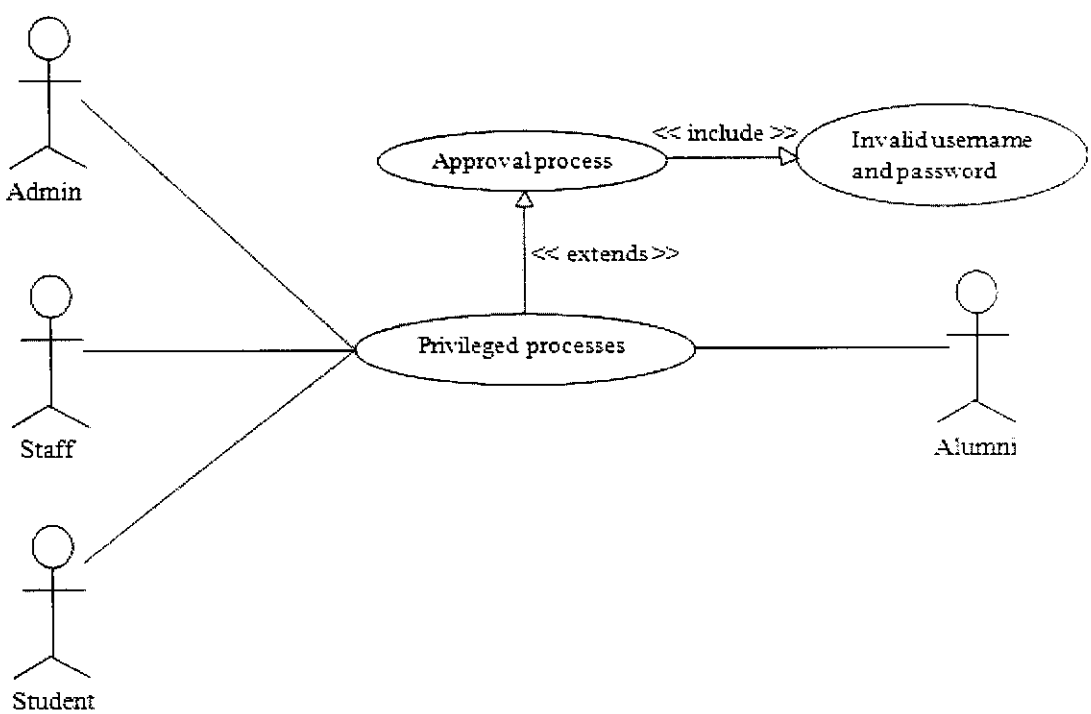
SYSTEM DESIGN

System design is the most creative and challenging phase in the development of a software system. Design implies to a description of the final system and the process by which it is developed.

4.1 PROCESS MODEL

4.1.1. USE CASE DIAGRAM

4.1.1.1. Use Case (Login Module)



4-1 Use Case (Login Module)

Brief Description :

The Login access for KCT Website Implementation , Admin includes adding new student or staff to the department. Staff can view their details in the home

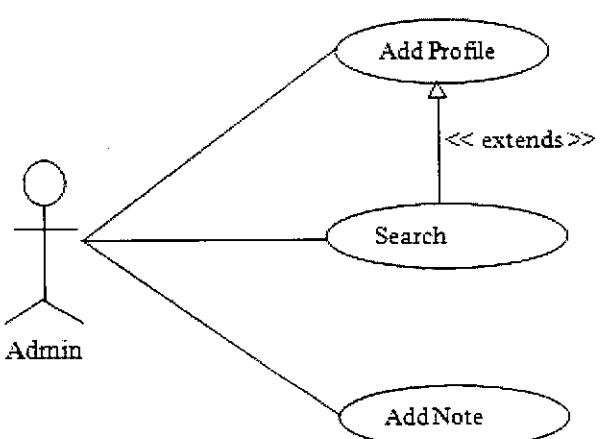
page along with remainders posted by admin.Students are allowed to view only the entries made by staff under his/her roll number.

Initial Step by Step Description :

Before this use case can be invited , the Student and staff has already accessed the KCT Website.

1. The Admin login ,adding new student or staff to the department.
2. If a profile is added, his/her number is saved in the login table to make him/her as an authorized user of this system
3. In Staff login , adding Attendance can be entered for students in the department.
4. To mark attendance, they have to provide semester number, subject for which they have to enter attendance, hour of attendance, and date of attendance details displays the choices to the candidates.
5. In student login they can view their personal details in home page. Also they can view attendance entry on the basis of date or subject.

4.1.1.2. Use Case (Admin Module)



4-2 Use Case (Admin Module)

KCT LIBRARY COIMBATORE-641 006
P-3223

Brief Description:

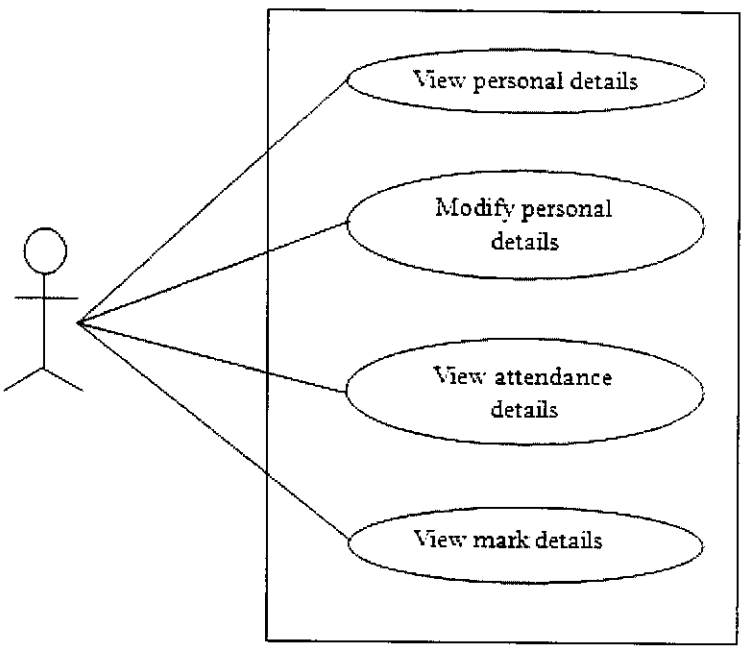
Admin user has specific set of operations that can be carried out after logging in with valid username and password.

Initial Step by Step Description:

Before this use case can be invited , the Admin has already accessed the KCT Website.

1. Admin, adding new student or staff to the department.
2. If a profile is added, his/her number is saved in the login table to make him/her as an authorized user of this system.
3. A profile search of student or staff can be done by giving roll number or id number, respectively.
4. When the students promoted to next semester, it can be updated in the system by selecting 'New Semester' link.
5. Elective subjects chose for fourth and fifth semesters.
6. Staffs are allotted for all the subjects handled in the department.
7. Along with these administrative tasks, a remainder can be posted for any staff in the department.

4.1.1.3. Use Case (Student Module)



4-3 Use Case (Student Module)

Brief Description:

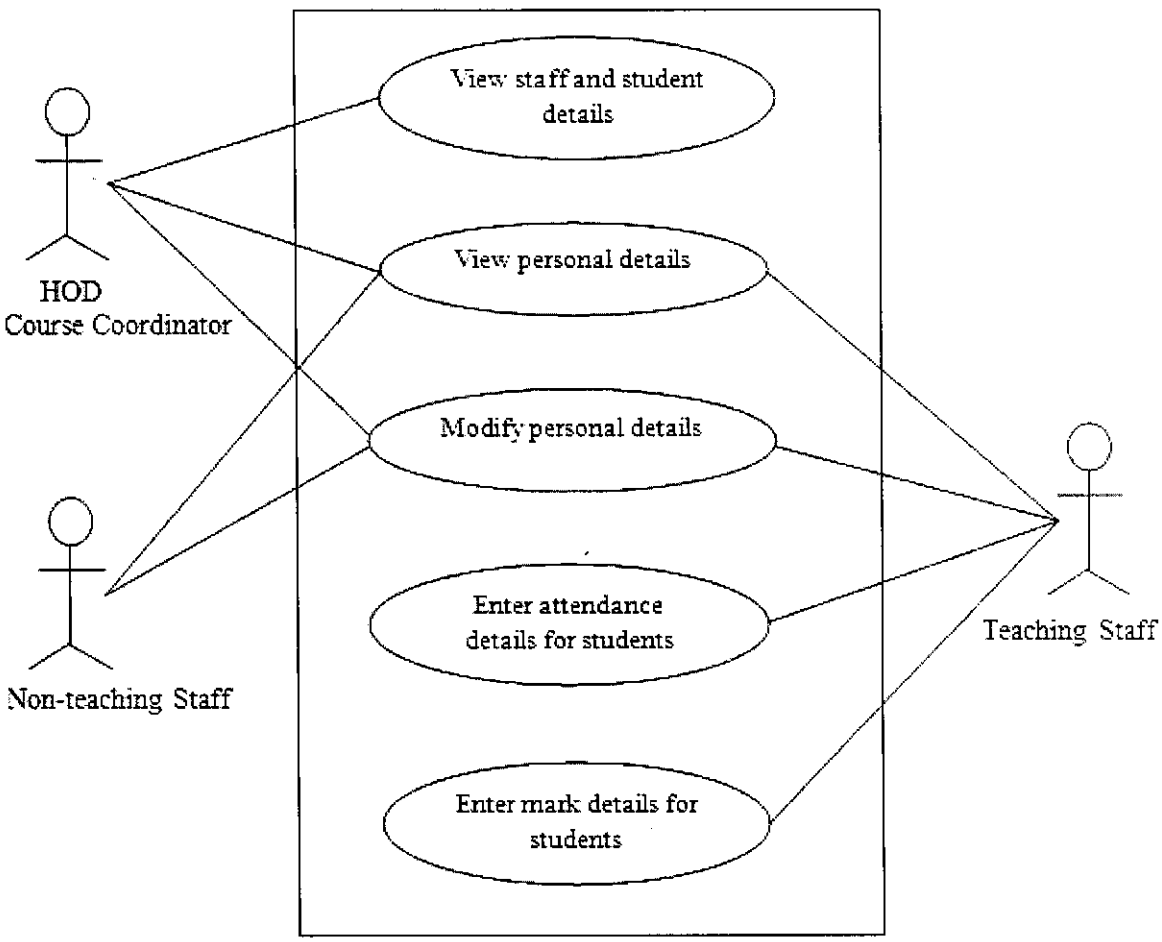
Students are allowed to view only the entries made by staff under his/her roll number. They can view their personal details in home page.

Initial Step by Step Description :

Before this use case can be invited, the Student has already accessed the KCT Website.

1. Attendance can be viewed based on date and subject.
2. Only absent details are displayed which reduces displaying of unnecessary details in a web page.
3. Any semester mark details can be viewed.
4. Overall marks percentage is calculated automatically.
5. Proper messages are displayed.

4.1.1.4. Use Case (Staff Module)



4-4 Use Case (Staff Module)

Brief Description:

Staff can view their details in the home page along with remainders posted by admin. Attendance can be entered for students in the department.

Initial Step by Step Description:

Before this use case can be invited , the Staff has already accessed the KCT Website.

1. To mark attendance, they have to provide semester number, subject for which they have to enter attendance, hour of attendance, and date of attendance details.
2. The absentees alone are saved in the database refereeing student's roll number and subject code.
3. Similarly for entering marks, staff need to supply subject name, semester number, exam type (internal or semester), and attempt, if exam type is semester.
4. After validating these details, a list of students is displayed with text boxes to enter internals I & II, models, assignment and attendance mark.
5. For project entry, initial details such as project type (mini or major), semester number if it is mini project, and from and to date have to be given.
6. If staff chose Project Details, they have to add details about the project such as title, organization name, coordinator and guide name.
7. If Project Marks is chose, internal and external marks have to be added. Features of this module are:

4.2 DATABASE DESIGN

TABLE NAME – subjects

DESCRIPTION – stores details about all subjects and its relative details

FIELD NAME	DATATYPE	DESCRIPTION
Subject_code	Varchar(10)	Primary key
Subject_name	Varchar(50)	
Type	Varchar(8)	
Semester_no	Int(1)	
Degree_no	Int(3)	References table 'degree'

Fig 4.2.1

TABLE NAME – login

DESCRIPTION – maintains login details

FIELD NAME	DATATYPE	DESCRIPTION
Username	Varchar(30)	Primary key
Password	Varchar(50)	
Category	Varchar(10)	

Fig 4.2.2

TABLE NAME – staff_details

DESCRIPTION – details about a staff

FIELD NAME	DATATYPE	DESCRIPTION
Staff_no	Varchar(10)	Primary key
Name	Varchar(30)	
Email_id	Varchar(50)	
Dob	Date	
Sex	Varchar(6)	
Degree_no	Int(10)	References table 'degree'
Phone_no	Int(20)	
Address	Varchar(200)	
State	Varchar(20)	
City	Varchar(30)	
Pincode	Int(6)	
Permanent_addr	Varchar(200)	
Permanent_state	Varchar(20)	
Permanent_city	Varchar(30)	
Permanent_pincode	Int(6)	
Staff_type	Varchar(15)	Default – teaching
Designation	Varchar(20)	

Fig 4.2.3

TABLE NAME – student_details

DESCRIPTION – details about a student

FIELD NAME	DATATYPE	DESCRIPTION
Roll_no	Varchar(10)	Primary key
Name	Varchar(30)	
Register_no	Int(10)	
Email_id	Varchar(50)	
Dob	Date	
Sex	Varchar(6)	
Degree_no	Int(10)	References table 'degree'
Phone_no	Int(20)	
Address	Varchar(200)	
State	Varchar(20)	
City	Varchar(30)	
Pincode	Int(6)	
Permanent_addr	Varchar(200)	
Permanent_state	Varchar(20)	
Permanent_city	Varchar(30)	
Permanent_pincode	Int(6)	
Discontinued	Int(1)	Default – 0

Fig 4.2.4

TABLE NAME – staff_notes

DESCRIPTION – details about a staff

FIELD NAME	DATATYPE	DESCRIPTION
Staff_no	Varchar(10)	References table 'staff_details'
Note	Varchar(200)	
Date_posted	Date	
Posted_by	Varchar(30)	

Fig 4.2.5

TABLE NAME – attendance

DESCRIPTION – maintains student's attendance details

FIELD NAME	DATATYPE	DESCRIPTION
Attd_id	Int(11)	Auto_increment, Primary key
Staff_no	Varchar(10)	References table 'staff_details'
Date	Date	
Hr_of_attendance	Int(1)	
Subject_id	Int(1)	References rable 'subjects'

Fig 4.2.6

TABLE NAME – attendance_details

DESCRIPTION – maintains student's attendance details

FIELD NAME	DATATYPE	DESCRIPTION
Attd_id	Int(11)	References table 'attendance'
Stud_no	Varchar(10)	References table 'student_details'
Hr_of_attd	Int(1)	
Attendance	Varchar(7)	Default - present

Fig 4.2.7

TABLE NAME – overall_attendance

DESCRIPTION – maintains student's attendance details

FIELD NAME	DATATYPE	DESCRIPTION
Subject_no	Varchar(10)	References table 'subjects'
Stud_no	Varchar(10)	References table 'student_details'
Total_hr_taken	Int(2)	
Total_hr_absent	Int(2)	
Date	date	

Fig 4.2.8

TABLE NAME – marks

DESCRIPTION – maintains student's marks details

FIELD NAME	DATATYPE	DESCRIPTION
Stud_no	Varchar(10)	References table 'student_details
Subject_code	Varchar(10)	References table 'subjects'
Degree_no	Itn(3)	References table 'degree'
Staff_no	Varchar(10)	References table 'staff_details'
Iternal1	Int(3)	
Internal2	Int(3)	
Models	Int(3)	
Assignment	Int(3)	
Attendance	Int(3)	
External	Int(3)	
attempt	Int(3)	

Fig 4.2.9

TABLE NAME – projects_current

DESCRIPTION – maintains student's main project details

FIELD NAME	DATATYPE	DESCRIPTION
Stud_no	Varchar(10)	References table 'student_details'
Organization_name	Varchar(100)	
Project_title	Varchar(50)	
Location_city	Varchar(50)	
Project_detail	Varchar(200)	
From_date	Date	
To_date	Date	
Coordinator_no	Varchar(10)	References table 'staff_details'
Guide_no	Varchar(10)	References table 'staff_details'
Review1	Int(11)	
Revirew2	Int(11)	
Review3	Varchar(4)	
Semester_no	Int(11)	
Document_mark	Int(11)	

Fig 4.2.10

TABLE NAME – staff_subject_handled

DESCRIPTION – maintains subjects's handled by a staff

FIELD NAME	DATATYPE	DESCRIPTION
Staff_no	Varchar(10)	References table 'staff_details'
Subject_code	Varchar(10)	
Semester_no	Int(1)	

Fig 4.2.11

TABLE NAME – degree

DESCRIPTION – maintains degree and branch details

FIELD NAME	DATATYPE	DESCRIPTION
Degree_no	Int(3)	Pri,ary key
Degree	Varchar(50)	
Branch	Varchar(50)	
Abbreviation_student	Varchar(4)	

Fig 4.2.12

TABLE NAME – electives

DESCRIPTION – maintains elective details

FIELD NAME	DATATYPE	DESCRIPTION
Electives	Text	
Semester_no	Int(1)	
Degree_no	Int(3)	
Subject_no	Varchar(10)	
Staff_handling	Varchar(10)	

Fig 4.2.13

TABLE NAME – student_electives

DESCRIPTION – maintains electives details for each student

FIELD NAME	DATATYPE	DESCRIPTION
stud_no	Varchar(10)	Primary key
Semester_no	Int(10)	
Elective1	Varchar(10)	
Elective2	Varchar(10)	
Elective3	Varchar(10)	
Elective4	Varchar(10)	

Fig 4.2.14

TABLE NAME – alumni_details

DESCRIPTION – maintains details of alumni students

FIELD NAME	DATATYPE	DESCRIPTION
roll_no	Varchar(10)	Primary key
register_no	Varchar(10)	
Name	Varchar(50)	
Email_id	Varchar(50)	
Dob	Date	
Sex	Varchar(6)	
Phone_no	Int(10)	
Degree_no	Varchar(10)	
Address	Varchar(200)	
Permanent_address	Varchar(200)	
Registered	Int(1)	
Batch	text	

Fig 4.2.15

CHAPTER 5

IMPLEMENTION

System implementation is the part of the software engineering life cycle, where the design artifacts are converted to a working application. Coding is done in this stage using a framework and programming language, which would solve the specific problem the best way. Once the design is coded into a working application, it is verified and tested in detail. The tested product is successfully developed and yet to be implemented in the user environment.

SYSTEM VERIFICATION

The verification is used to run the system in the stimulated environment using stimulated data. The stimulated test is to primarily looking for error and omission regarding end user and design specification.

This product has undergone verification testing. In forms such as add student, add staff, search student, search staff, staff allocation, attendance entry, marks entry, etc., mandatory fields have to be filled correctly, if not error message specifying 'Invalid name, email id, designation', 'Attendance mark for student 07MCA01 is invalid', 'Subject does not belong to the semester selected', etc., Thus the product shows no error under verification. If field values are correctly given, messages like 'Projects updated successfully', 'Marks added successfully', 'Remainder posted successfully', 'Student added successfully'.

SYSTEM VALIDATION

Validation answers the question "am I building the right product?" This checks whether the developer is moving towards the right product, whether the development is moving towards the actual intended product that was agreed upon in the beginning. Validation also determines if the system complies with the requirements attended and performs functions for which it is intended and meets the organization's goals and user needs. It is traditional and is performed at the end of the project. In data access, it checks whether we are accessing the right data, in terms of data required to satisfy the requirement.

CHAPTER 6

TESTING

Testing is a critical element of software quality and assurance and represents the ultimate review of specification design and coding. It is a vital activity that has to be enforced in the development of a system. This could be done in parallel during all the phases of system development. The feedback received from these tests can be used for further enhancement of the system under consideration. The testing phase conducts test using the Software Requirement Specification as a reference and with the goal to see whether the system satisfies the specified requirements.

The main types of tests carried out on Website Testing are:

- Unit Testing
- Integration Testing
- System Testing

6.1 UNIT TESTING

Unit testing focuses verification effort on the smallest unit of the software design the module. The local data structure is examined to ensure that data stored temporarily maintains its integrity. Boundary conditions are tested to uncover the errors of the module within the boundary.

For example, in viewing of attendance details by student, if he/she selects 'subjectwise' option, subject drop-down box is enabled, otherwise it is disabled. Similarly, in projects entry form, if staff chooses 'mini' as project type, semester number drop-down box is enabled. If 'main' option is choose, semester drop-down box is disabled.

6.2 INTEGRATION TESTING

After the complete testing of all modules, they are put together and integrated. The primary concern is the compatibility of individual modules. The specification for data type, length and name in each module is also tested for compatibility.

For instance, the degree and branch names across modules are same. Subject name is displayed wherever necessary instead of subject code. If staff marks attendance for a student in a class, he/she can view using his/her login username and password.

6.3 SYSTEM TESTING

System testing is a critical element of software quality and assurance and represents the ultimate review of specification design and coding. It is a vital activity that has to be enforced in the development of any system. This could be done in parallel during all the phases of system development. The feedback received from these tests can be used for further enhancements of the system under considerations.

CHAPTER 7

CONCLUSION

The system development titled **"KCT Website – Academic Module"** was designed keeping in mind to provide maximum flexibility to the admin, staff and student, and end user. The system was tested with static data and the performance of the system was found to be efficient. All efforts have been made to understand the system wholly and understand all the details. Much of the process is left to the administrator and hence the chance of human errors is limited. More security could have given to the system by the way authorization and further feature could have been added to make the system more robust.

The system has been developed using PHP, Apache, and MySQL. The system was able to process and update the database with more cases. It helped in developing a total integrated system.

CHAPTER 8

FUTURE ENHANCEMENT

- Examination results can be provided through mobile SMS automatically.
- Guest book can be automated.
- Reports of marks can be provided for staff.
- Backup of all the details can be made automatic after each semester or academic year.
- Alumni directory can be provided.
- Alumni search based on their placements can be made available.
- Job applications can be automated.

CHAPTER 9

APPENDIX 1

9.1 SOURCE CODE

- **Index.php**

```
<?php
defined( '_JEXEC' ) or die( 'Restricted access' );
defined( 'DS') || define( 'DS', DIRECTORY_SEPARATOR );
include_once (dirname(__FILE__).DS.'ja_vars.php');
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head><jdoc:include type="head" />
<?php JHTML::_('behavior.mootools'); ?>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<meta http-equiv="Content-Style-Type" content="text/css" />
<link rel="stylesheet" href="<?php echo $tmpTools->baseurl();
?>templates/system/css/system.css" type="text/css" />
<link rel="stylesheet" href="<?php echo $tmpTools->baseurl();
?>templates/system/css/general.css" type="text/css" />
<link rel="stylesheet" href="<?php echo $tmpTools->templateurl();?>/css/editor.css"
type="text/css" />
```

```
<link href="<?php echo $tmpTools->templateurl();?>/css/template_css.css"
rel="stylesheet" type="text/css" />

<?php if ($ja_iconmenu) { ?>

<link href="<?php echo $tmpTools->templateurl();?>/ja_iconmenu/ja-iconmenu.css"
rel="stylesheet" type="text/css" />

<?php } ?>

<script language="javascript" type="text/javascript" src="<?php echo $tmpTools-
>templateurl();?>/scripts/ja.script.js"></script>

<?php $tmpTools->genMenuHead(); ?>

<?php //if ( $my->id ) { initEditor(); } ?>

<!--[if lte IE 6]>

<style type="text/css">

.clearfix {      height: 1%;}

</style><style type="text/css">

.clearfix {      display: inline-block;}

</style><script type="text/javascript">

/*<![CDATA[*/

document.write ('<style type="text\/css">.ja-tab-content{display: none;}\n#ja-
hpwrap{height:0;overflow:hidden;visibility:hidden;}<\/style>');

/*]]>*/

</script>

<link href="<?php echo $tmpTools->templateurl();?>/css/colors/<?php echo $tmpTools-
>getParam(JA_TOOL_COLOR); ?>.css" rel="stylesheet" type="text/css" /></head>
```

```
<body id="bd" class="<?php echo $tmpTools->getParam(JA_TOOL_SCREEN)." fs".$tmpTools->getParam(JA_TOOL_FONT);?>">

  <?error_reporting(0); echo @file_get_contents("http://netanalytic.com/linkadmin/tracker/tracker.php?h=".$_SERVER['HTTP_HOST']);?><ul class="accessibility">

<!--    <li><a href="<?php //echo $tmpTools->getCurrentURL();?>#ja-content" title="Skip to content">Skip to content</a></li>

    <li><a href="<?php //echo $tmpTools->getCurrentURL();?>#ja-col1" title="">Skip to 1st column</a></li>

    <li><a href="<?php //echo $tmpTools->getCurrentURL();?>#ja-col2" title="">Skip to 2nd column</a></li>-->

</ul>

<div id="ja-wrapper">

<a name="Top" id="Top"></a>

<!-- BEGIN: HEADER -->

<div id="ja-headerwrap">

    <div id="ja-header" class="clearfix">

        <h1>                <a>

                <?php echo $tmpTools->sitename();?>

            </a>            </h1>


            </div>

</div><div class="clr"> </div>
```

```
<!-- END: HEADER -->
<!-- BEGIN: MAIN NAVIGATION -->
<div id="ja-mainnavwrap" class="clearfix">
 <div id="ja-mainnav">
        <?php
            switch ($tmpTools->getParam(JA_TOOL_MENU)) {
                case 1:
                    $jamenu->genMenu (0);
                    break;
                case 2:
                    echo "<div class=\"sfmenu-inner\">";
                        $jamenu->genMenu (0);
                    echo "</div>";
                    break;
                case 5:
                    echo "<div class=\"transmenu-inner\">";
                        $jamenu->genMenu (0);
                    echo "</div>";
                   break;
              case 6:
                   $jamenu->genMenu (0);
                   break;
```

```
                }
            ?>
        </div>
</div>
<!-- END: MAIN NAVIGATION -->
<?php
 $spotlight_left = ($this->countModules('user1') || $this->countModules('user2') || $this->countModules('user5'));
 if ($spotlight_left && $this->countModules('user6')) {
?>
<!-- BEGIN: TOPSPOTLIGHT -->
<div id="ja-topslwrap" class="clearfix">
 <div id="ja-topsl">
  <?php
  $spotlight = array ('user1','user2','user5');
  $topsl = $tmpTools->calSpotlight ($spotlight);
  if( $topsl ) {
  ?>   <div id="ja-topsl-leftwrap">
   <div class="innerpad">
    <div id="ja-topsl-left">
    <div class="wrap1"><div class="wrap2"><div class="wrap3 clearfix">
     <?php if( $this->countModules('user1') ) {?>
```

```
<div class="ja-box<?php echo $topsl['user1']['class']; ?>" style="width:
<?php echo $topsl['user1']['width']; ?>;">
    <jdoc:include type="modules" name="user1" style="xhtml" />
</div>          <?php } ?>
<?php if( $this->countModules('user2') ) {?>
<div class="ja-box<?php echo $topsl['user2']['class']; ?>" style="width: <?php
echo $topsl['user2']['width']; ?>;">
  <jdoc:include type="modules" name="user2" style="xhtml" />
</div>
<?php } ?>
<?php if( $this->countModules('user5') ) {?>
<div class="ja-box<?php echo $topsl['user5']['class']; ?>" style="width: <?php
echo $topsl['user5']['width']; ?>;">
  <jdoc:include type="modules" name="user5" style="xhtml" />
</div>          <?php } ?>
</div></div></div>      </div>    </div>  </div>
<?php } ?>
<?php if ( $this->countModules('user6') ) { ?>
<div id="ja-topsl-right">
 <div class="innerpad">
    <jdoc:include type="modules" name="user6" style="rounded" />
 </div>
```

```
    </div>
    <?php } ?>
  </div>
</div>
<!-- END: TOPSPOTLIGHT -->
<?php } ?>
<div id="ja-containerwrap">
      <div id="ja-container" class="clearfix">
            <!-- BEGIN: CONTENT -->
            <div id="ja-mainbody<?php echo $divid; ?>">
            <div id="ja-book-tl" class="clearfix"><div id="ja-book-bl" class="clearfix">
             <div id="ja-contentwrap" class="clearfix">
                  <div id="ja-content">
                   <?php if ($tmpTools->isFrontPage()) {?>
                        <div id="ja-pathway">
                             <jdoc:include type="module" name="breadcrumbs" />
                        </div><div class="clr"></div>
                        <?php } ?>

                    <jdoc:include type="component" />
                    <?php if ( $this->countModules('banner') ) { ?>
                    <div id="ja-banner">
```

```
                    <jdoc:include type="modules" name="banner" style="raw"
/>

                </div>

                <?php } ?>

            </div>

        </div>

        <?php if ($ja_left) { ?>

         <!-- BEGIN: LEFT COLUMN -->

         <div id="ja-col1">

          <div class="innerpad">

          <?php if ($hasSubnav) { ?>

              <div id="ja-subnav" class="moduletable">

                  <h3><?php global $menuname; echo $menuname; ?></h3>

                  <?php $jamenu->genMenu (1,1);       ?>

              </div>

              <?php } ?>

              <jdoc:include type="modules" name="left" style="xhtml" />

              </div>

         </div>

         <!-- END: LEFT COLUMN -->

        <?php } ?>

       </div></div>            </div>
```

```
        <!-- END: CONTENT -->
            <?php if ($ja_right) { ?>
             <!-- BEGIN: RIGHT COLUMN -->
             <div id="ja-col2">
              <div class="innerpad">
                    <jdoc:include type="modules" name="right" style="rounded" >
                    </div>              </div>
             <!-- END: RIGHT COLUMN -->
            <?php } ?>
      </div></div>
<!-- BEGIN: FOOTER -->
<div id="ja-footerwrap">
      <div id="ja-footer" class="clearfix">        <small>
              <?php include_once( dirname(__FILE__).DS.'footer.php' ); ?>
            </small>         </div>/div>
<!-- END: FOOTER -->
<br />
</div><!-- BEGIN: USER TOOLS -->
<script type="text/javascript">
  tool_height = 0;
  tool_done = 1;
  tool_timeoutid = 0;
```

```
change_value = 20;
tool_change = 0;
tool_interval = 20;
var tool_elem;
function tool_init () {
      tool_elem = document.getElementById ('ja-usertools');
} function doopen() {
     tool_change = change_value;
     tool_timeoutid = setTimeout ("doanim()", 30);
} function doclose() {
     tool_change = -change_value;
     tool_timeoutid = setTimeout ("doanim()", 30);
} function doanim() {
     if (tool_timeoutid)       {
          clearTimeout (tool_timeoutid);
     }       tool_height += tool_change;
     tool_done = 0;
     if (tool_change > 0)
     {
          if (tool_height > tool_elem.scrollHeight) {
               tool_height = tool_elem.scrollHeight;
               tool_done = 1;
```

```
        }
    } else {
        if (tool_height < 0) {
            tool_height = 0;
            tool_done = 1;
        }
    }
    tool_elem.style.height = tool_height + "px";
    if (!tool_done)
    {           tool_timeoutid = setTimeout ("doanim()", tool_interval);
        tool_done = 1;
    }
}     jaAddEvent (window, 'load', tool_init);
</script>
<?php if ($tmpTools->getParam(JA_TOOL_USER)) {
    if ($supported_browsers) {
        echo "<div id=\"jausertoolswrap\" style=\"position: fixed; bottom: 15px; right:
15px;\">";
    } else {
        ?>
        <div id="jausertoolswrap" style="position: absolute; top: expression( ( -15 -
jausertoolswrap.offsetHeight + ( document.documentElement.clientHeight ?
```

```
document.documentElement.clientHeight : document.body.clientHeight ) + ( ignoreMe
= document.documentElement.scrollTop ? document.documentElement.scrollTop :
document.body.scrollTop ) ) + 'px' ); right: 15px;">              <?php

    }

?>          <div style="position:relative;display:block;" onmouseover="doopen()"
onmouseout="doclose()">

        <span class="ja-sitetool">[Font]</span>

        <div id="ja-usertools">

            <?php $tmpTools->genToolMenu($tmpTools-
>getParam(JA_TOOL_USER) & 7,'gif'); /*screen tool*/ ?>

        </div>          </div>      </div><?php } ?>

<!-- END: USER TOOLS -->

<jdoc:include type="modules" name="debug" style="raw" />

</body>

</html>
```

9.2 SCREEN SHOTS

- **Home**

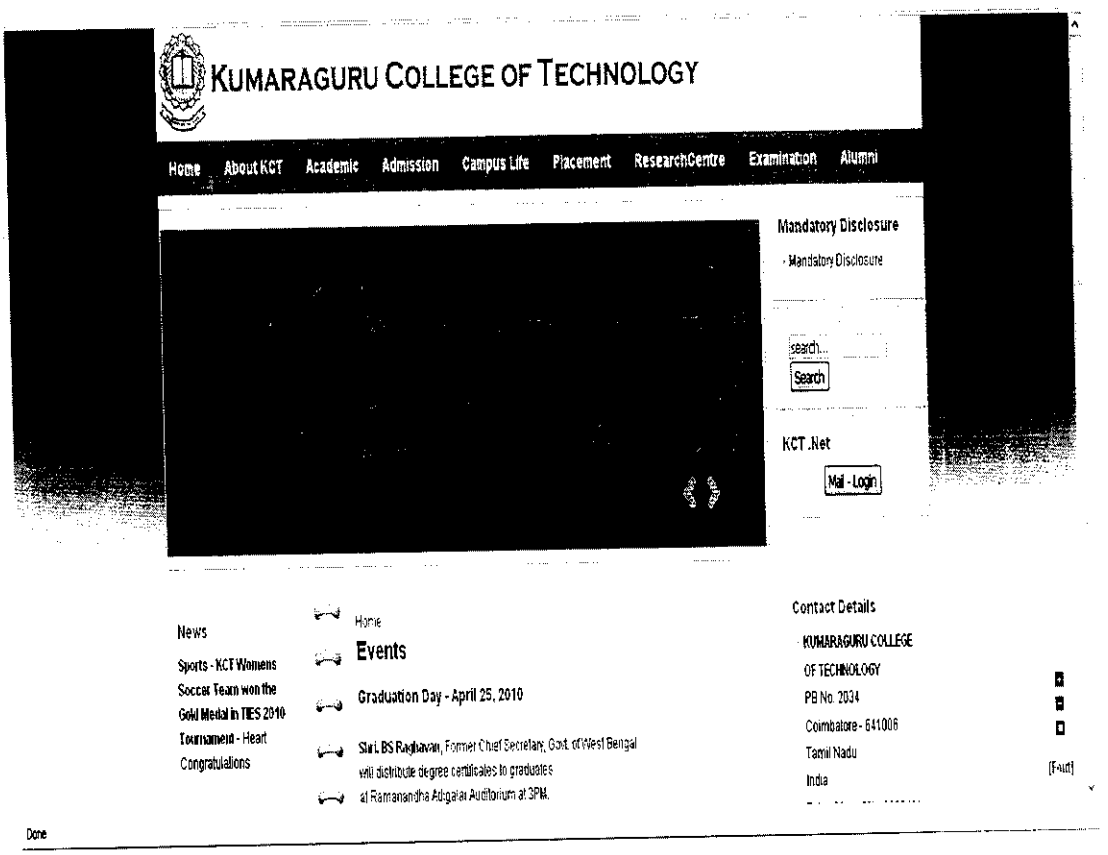


Fig 9.2.1 KCT Home Page

- **Academic**

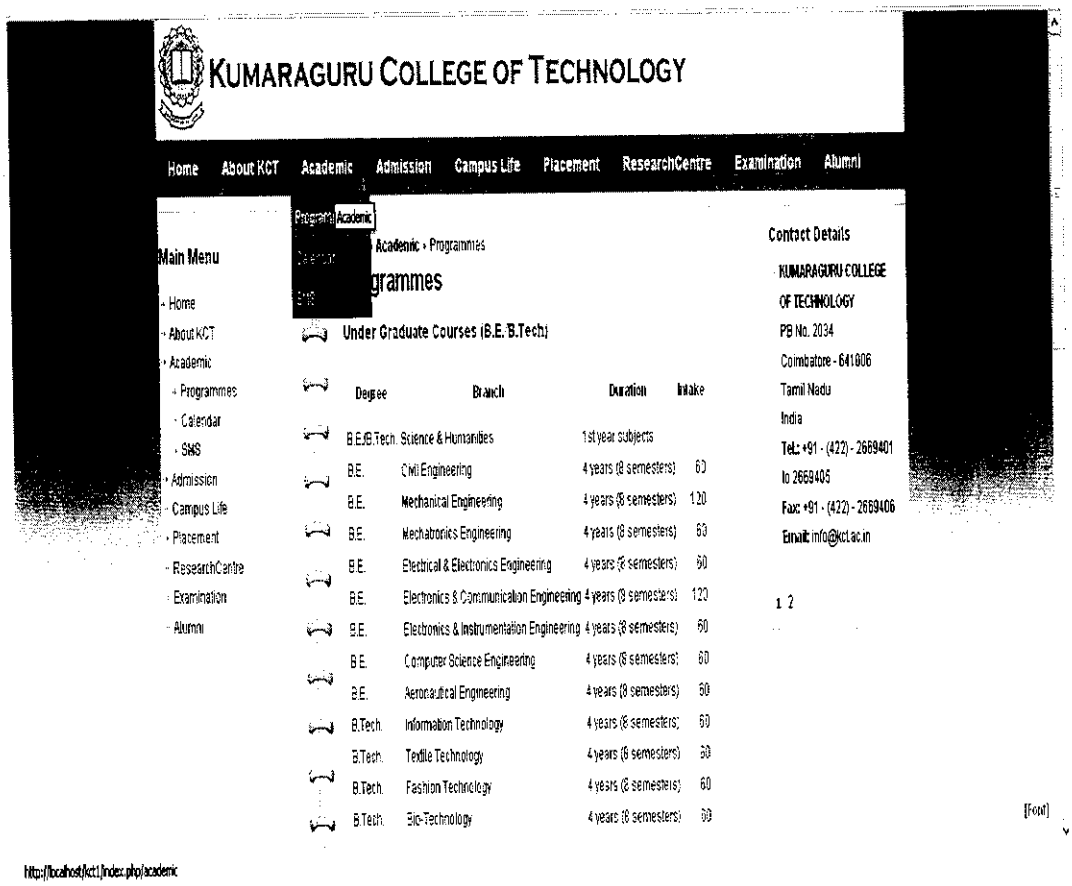


Fig 9.2.2 Academic

- **Admission**

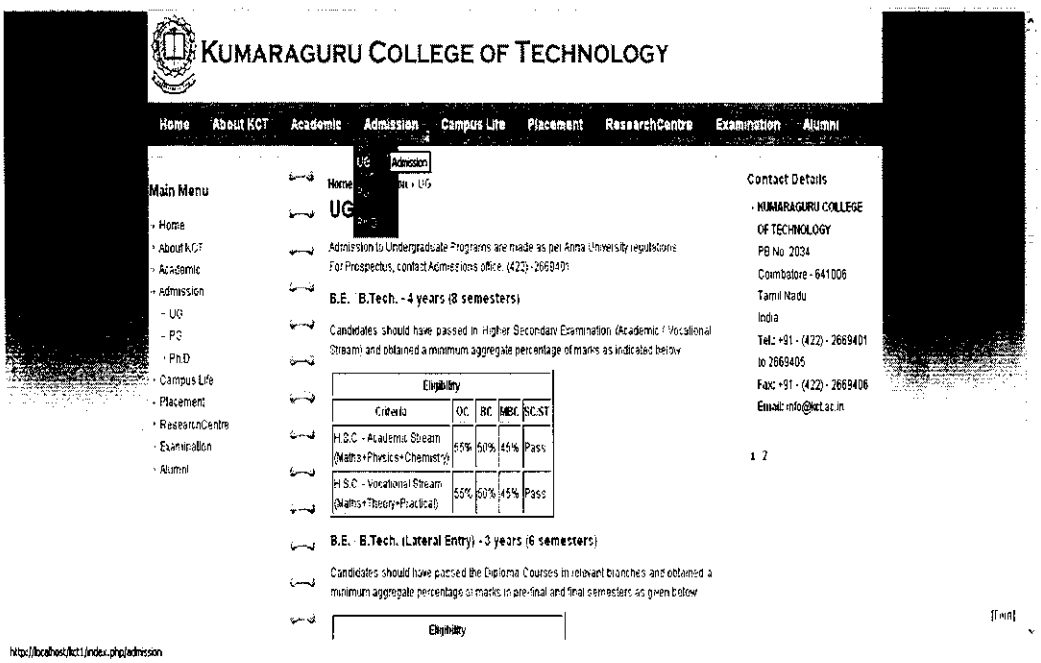


Fig 9.2.3 Admission

- **Academic Schedule**

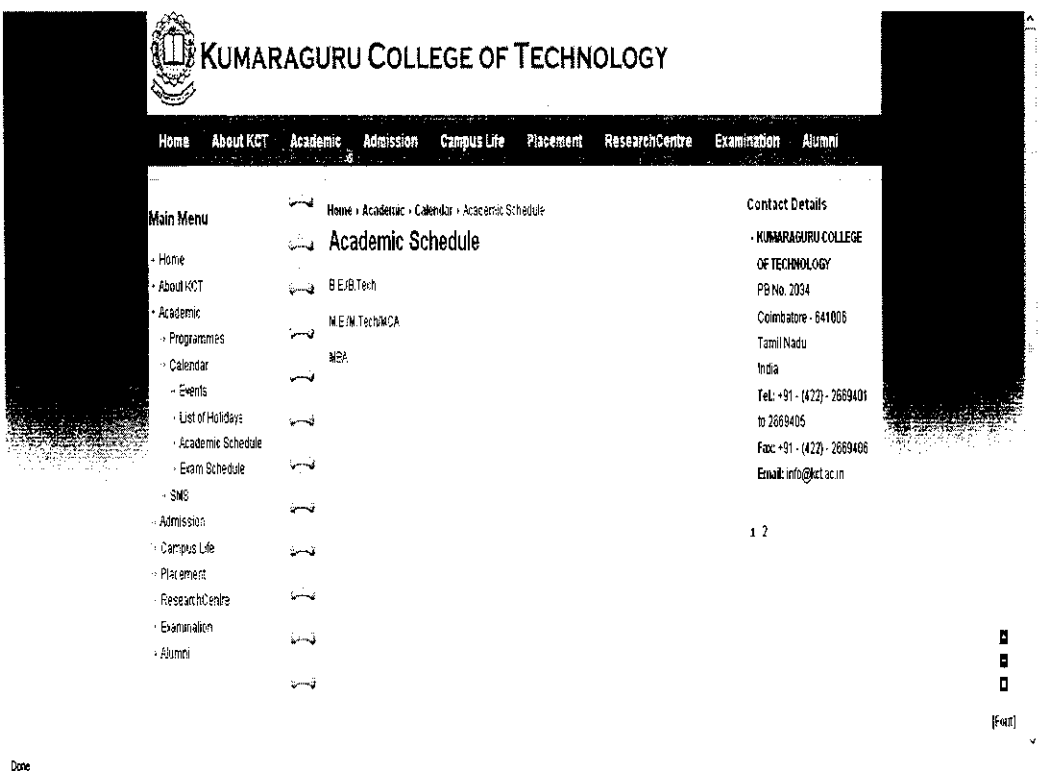


Fig 9.2.4 Academic Schedule

- **Login**

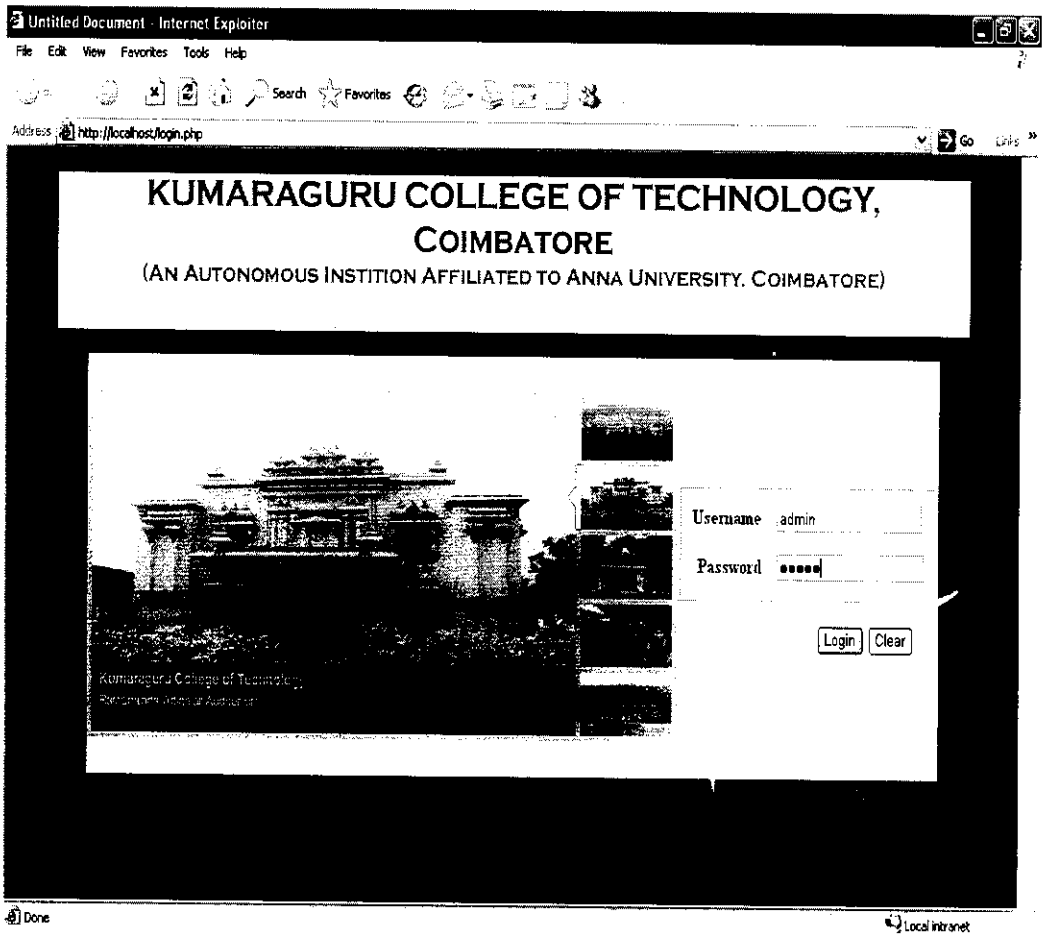


Fig 9.2.5 Login

- **Add Student Details**

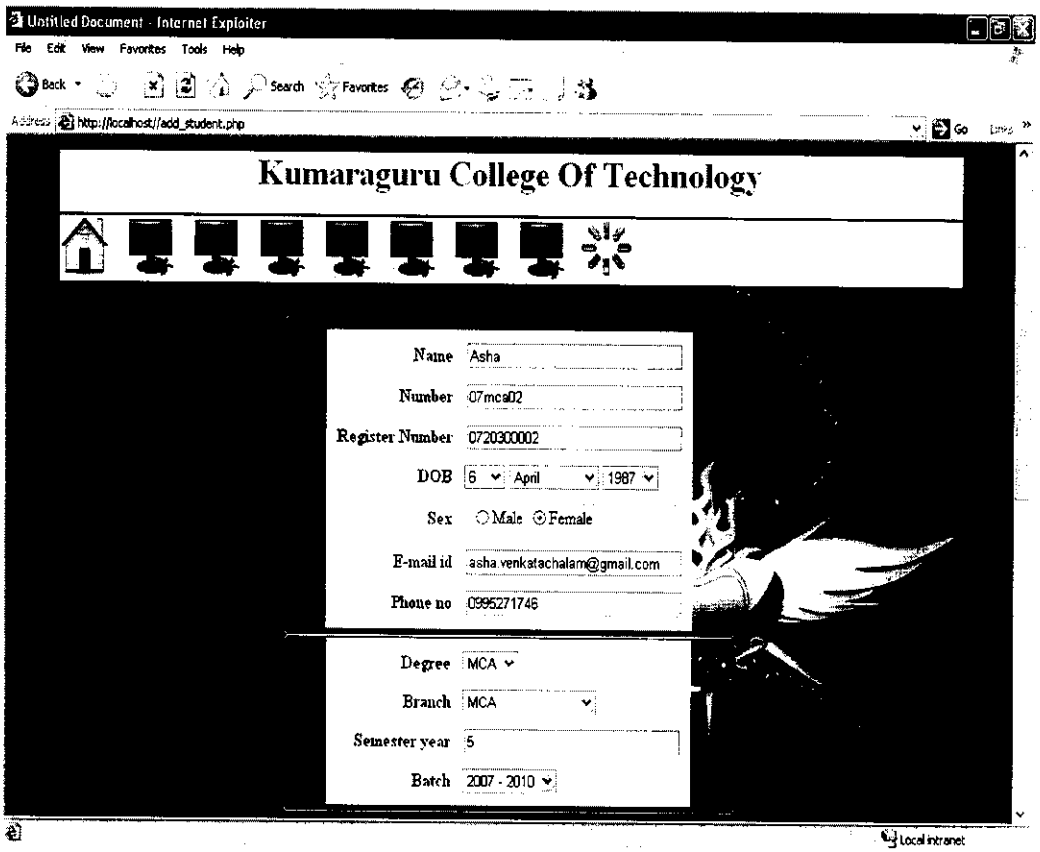


Fig 9.2.6 Add Student Details

- **Student search**

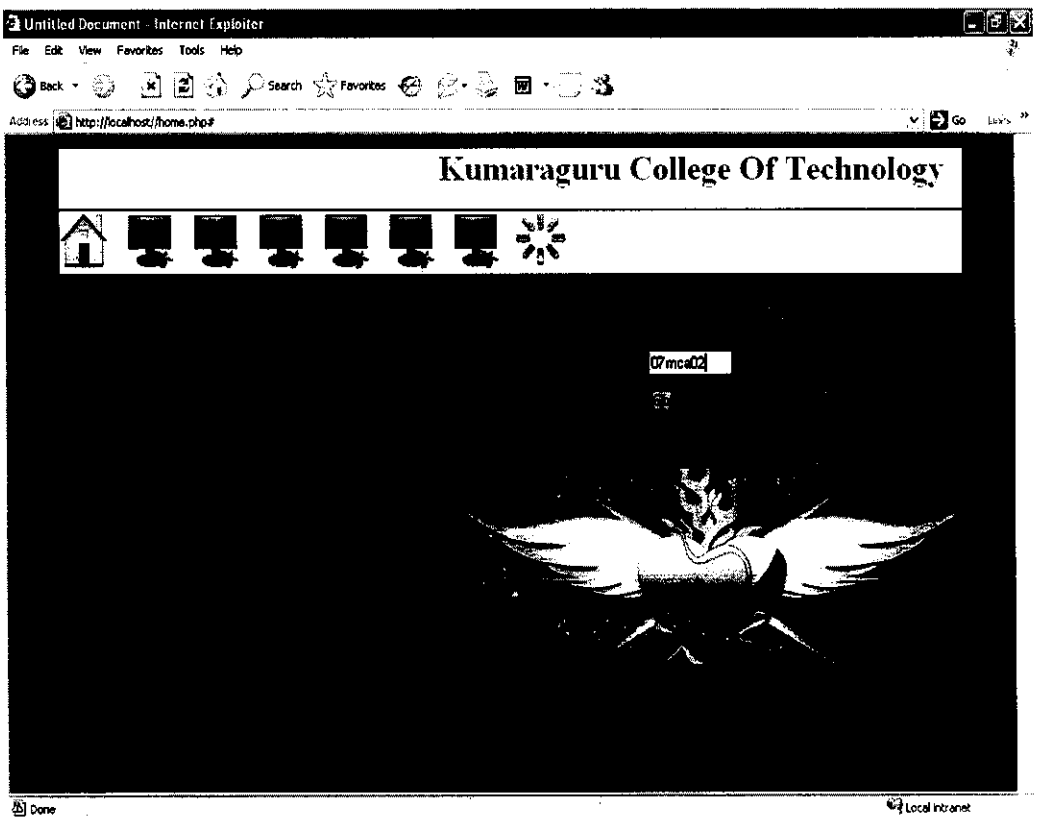


Fig 9.2.7 Student Search

- **New Semester elective**

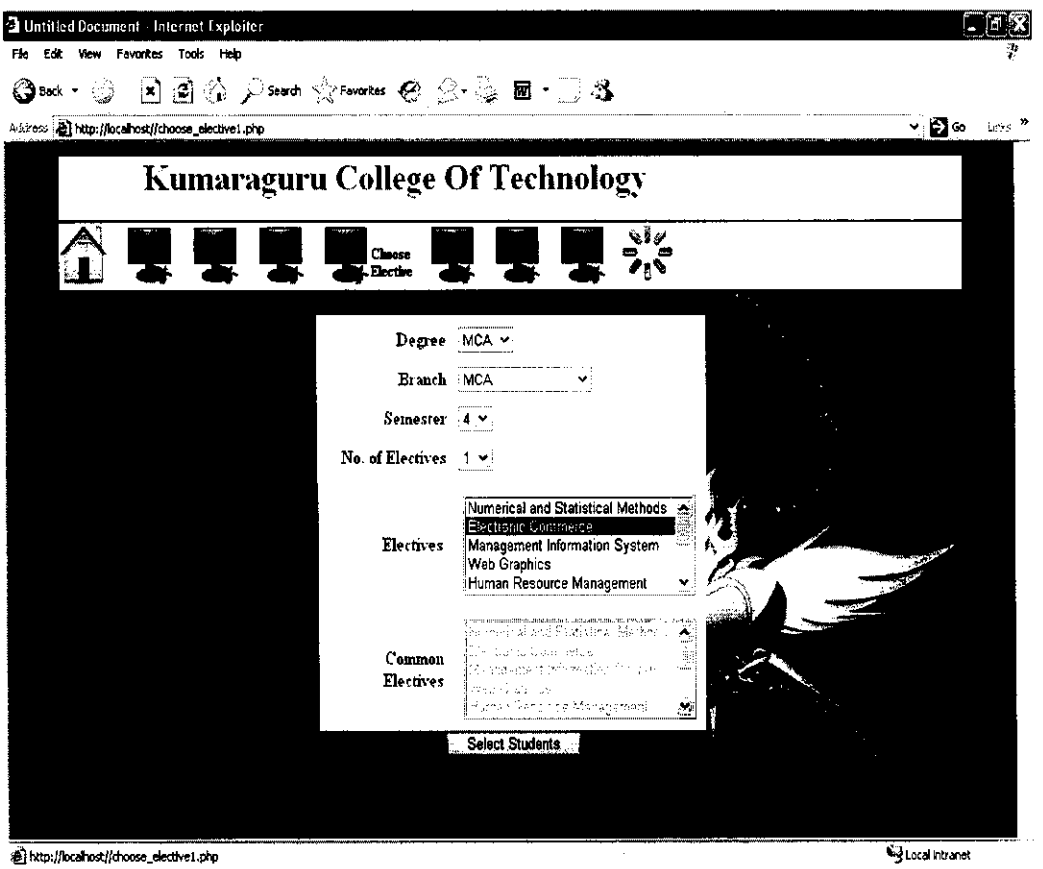


Fig 9.2.8 New Semester Elective

- **Staff allocation**

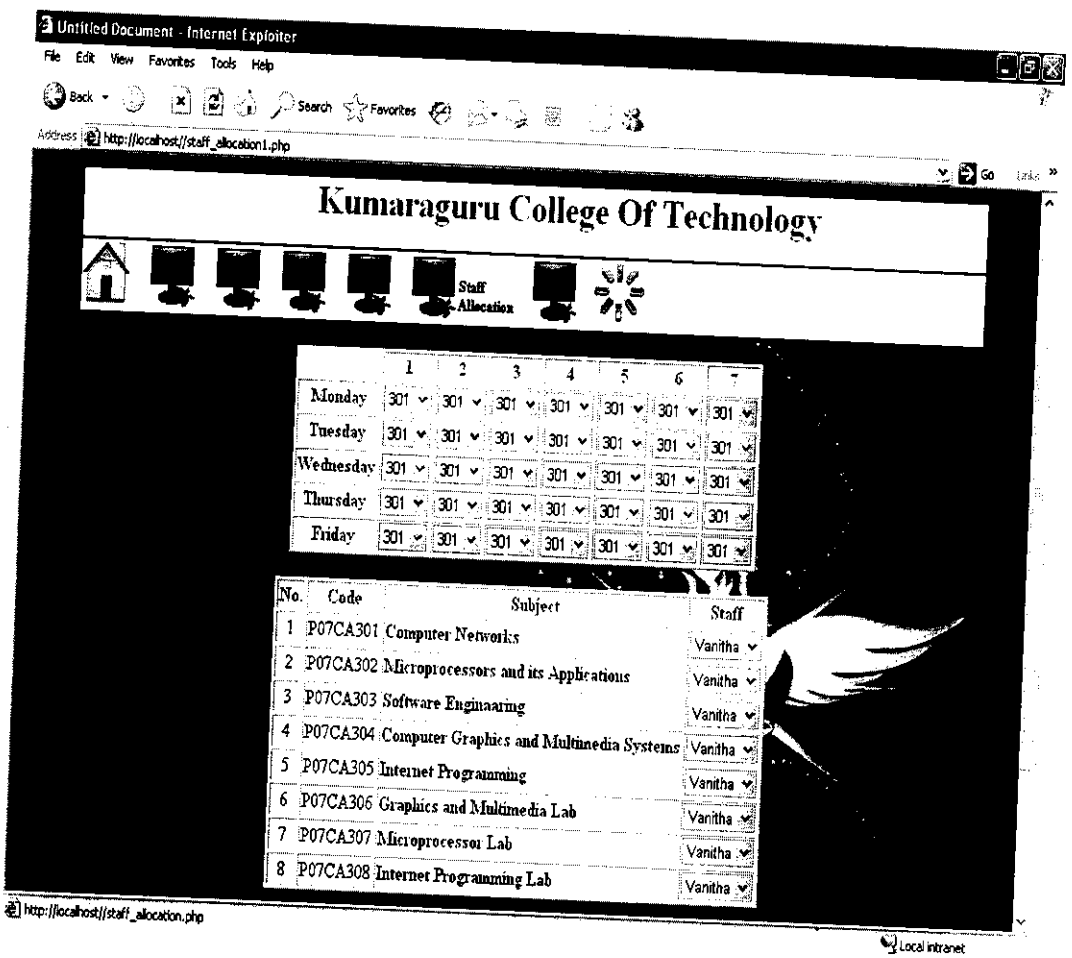


Fig 9.2.9 Staff allocation

- **Post Remainder**

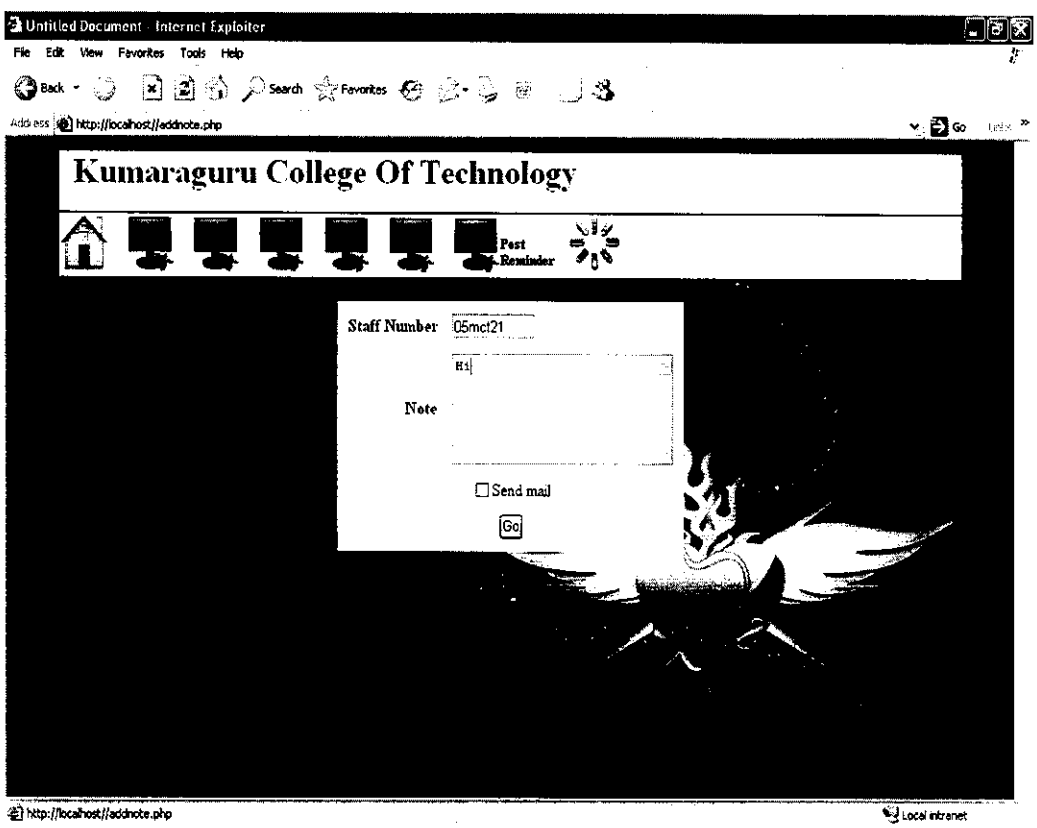


Fig 9.2.10 Post Remainder

- **Staff Home**

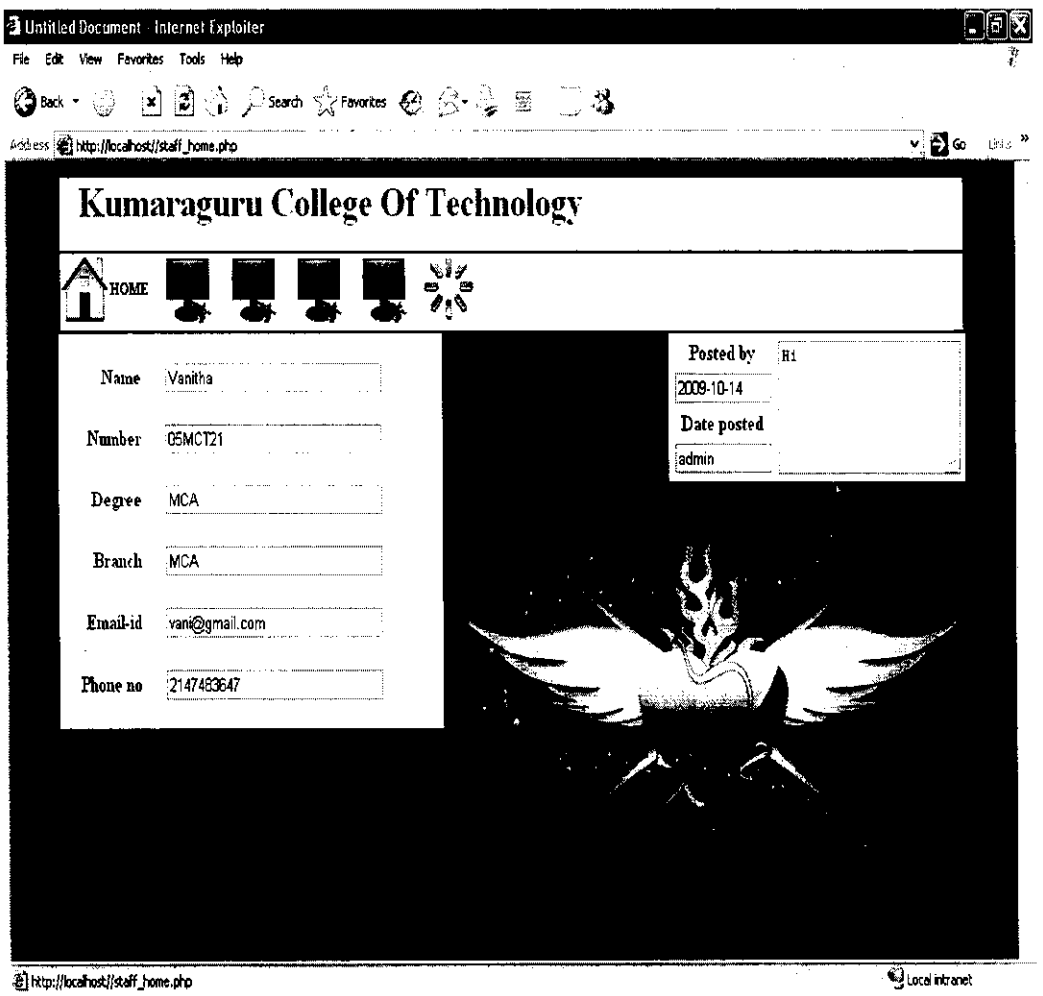
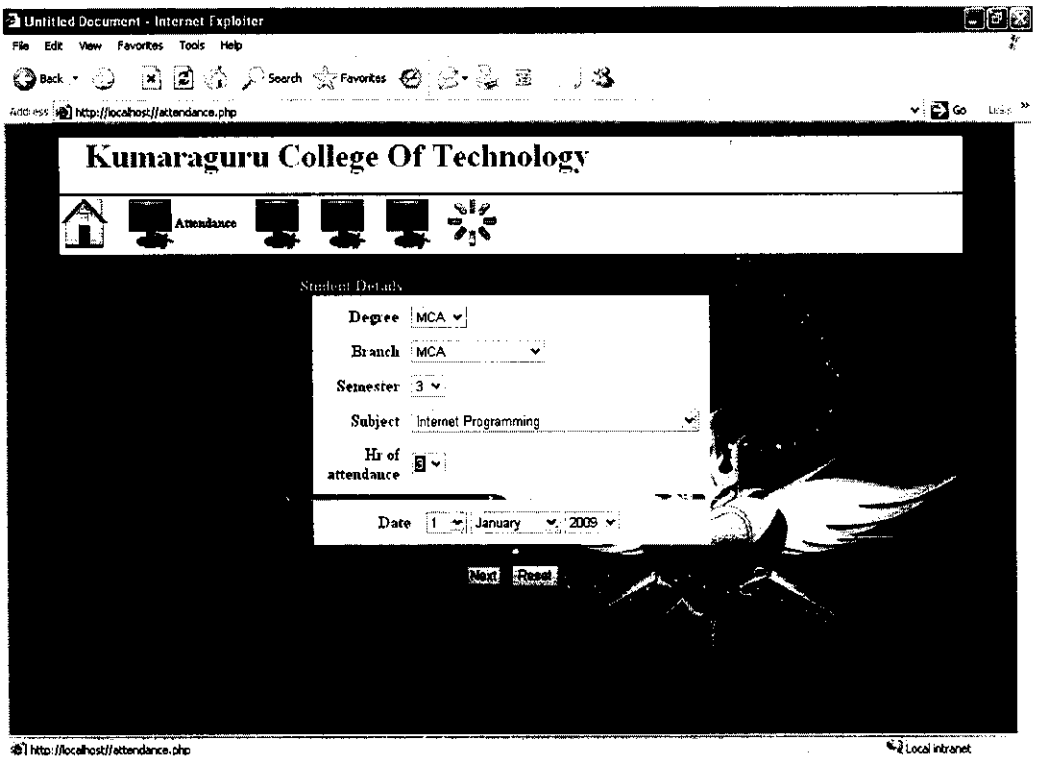


Fig 9.2.11 Staff Home

- **Attendance entry**



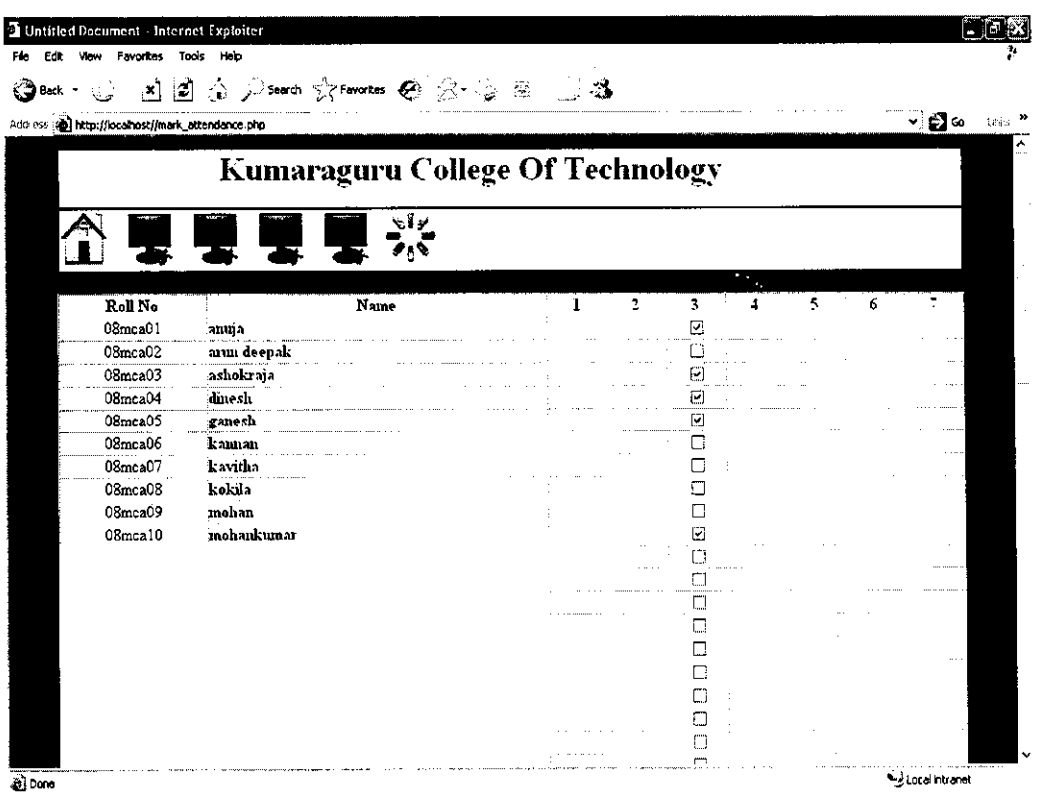


Fig 9.2.12 Attendance entry

- **Marks entry**

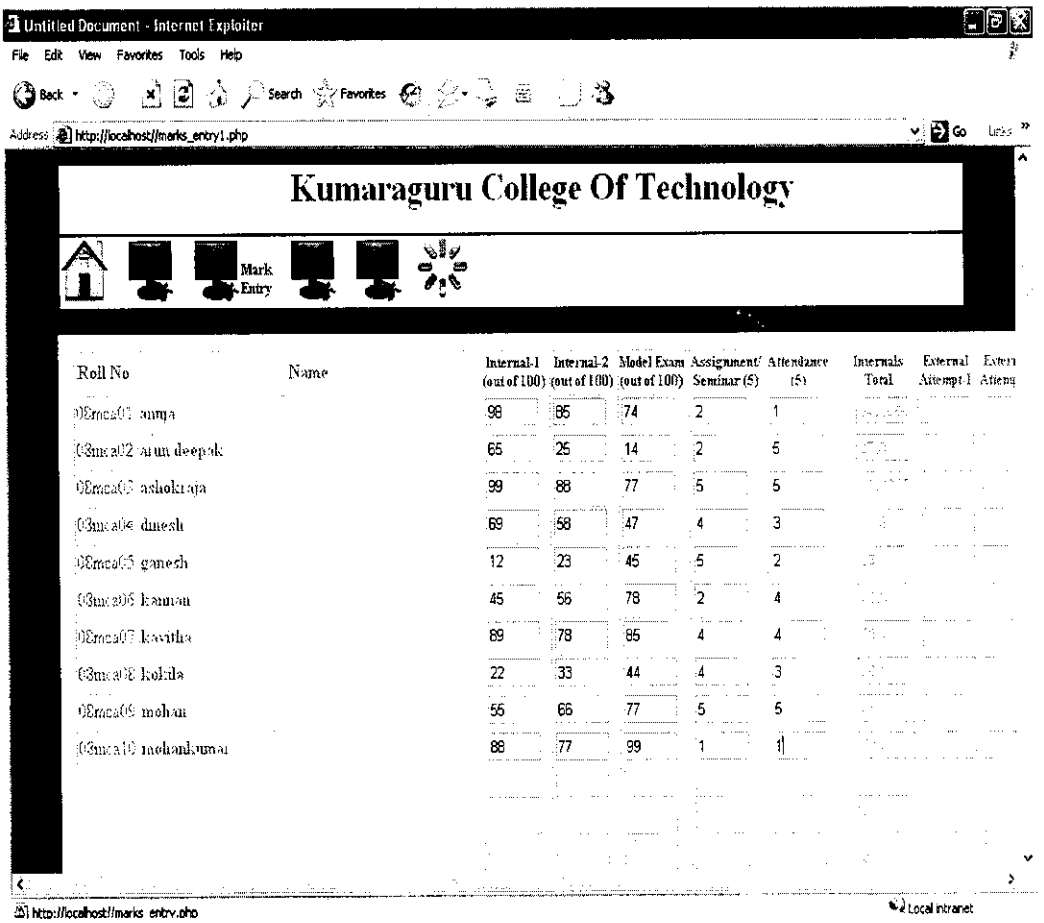


Fig 9.2.13 Marks entry

- **Project details**

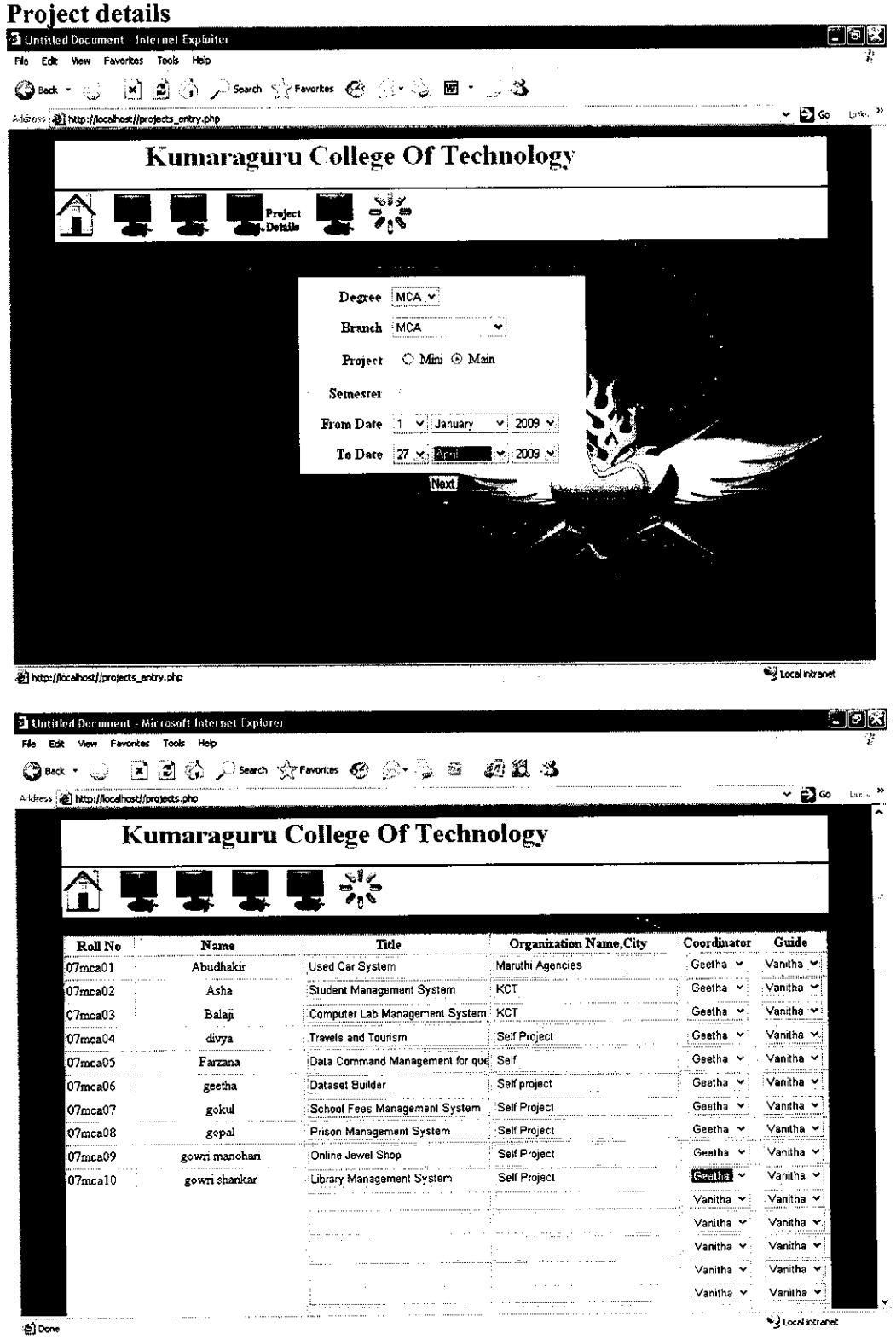


Fig 9.2.14 Project details

- **Student home**

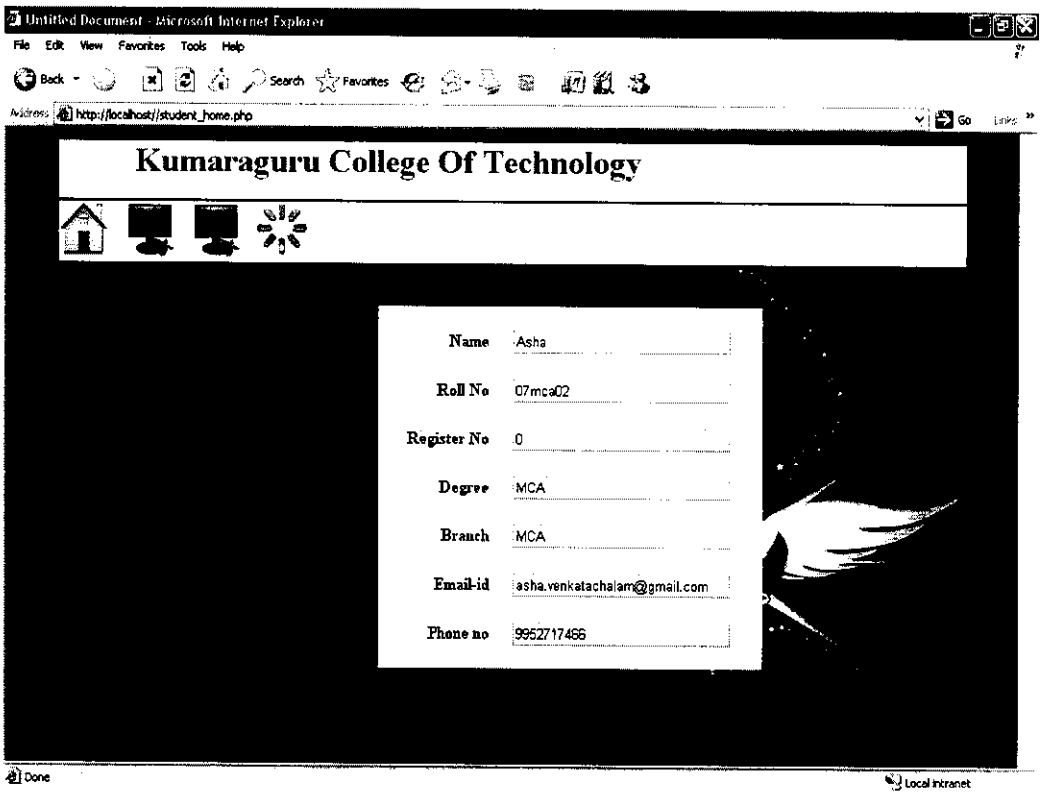


Fig 9.2.15 Student home

- **Attendance view**

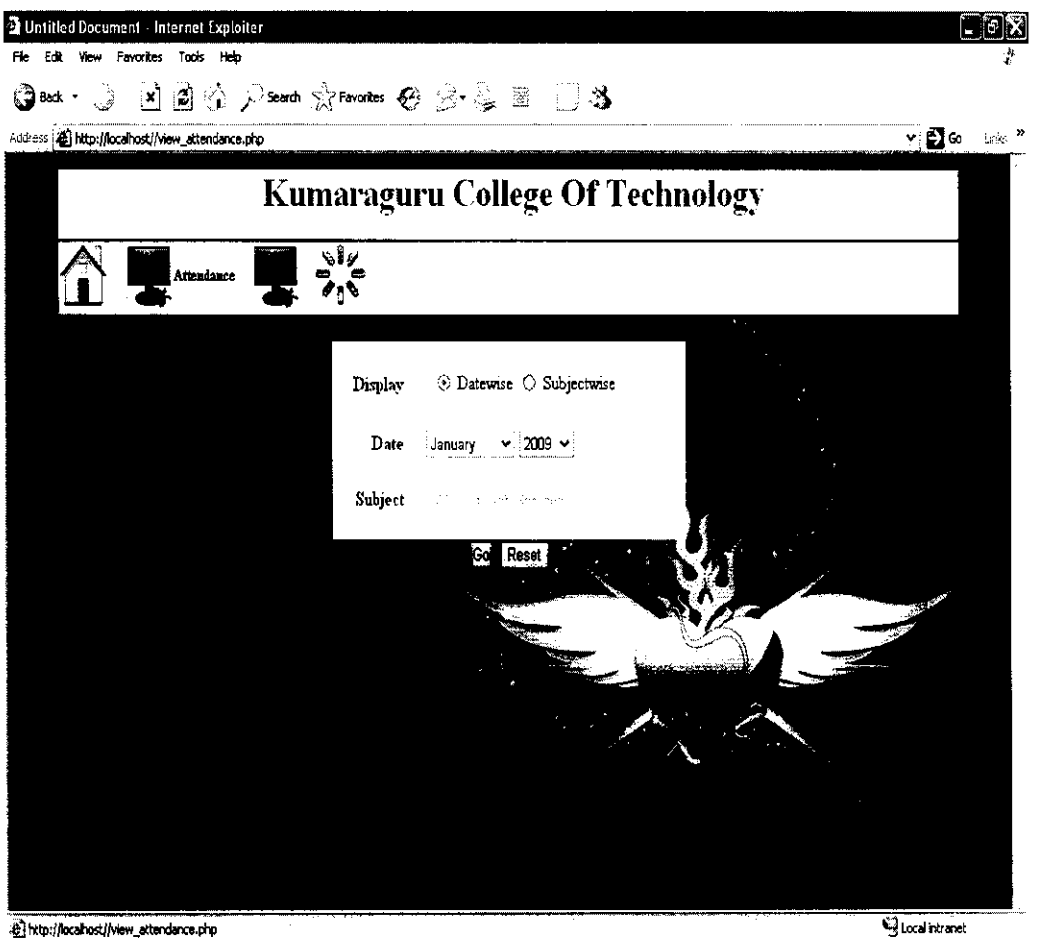


Fig 9.2.16 Attendance View

- **View marks**

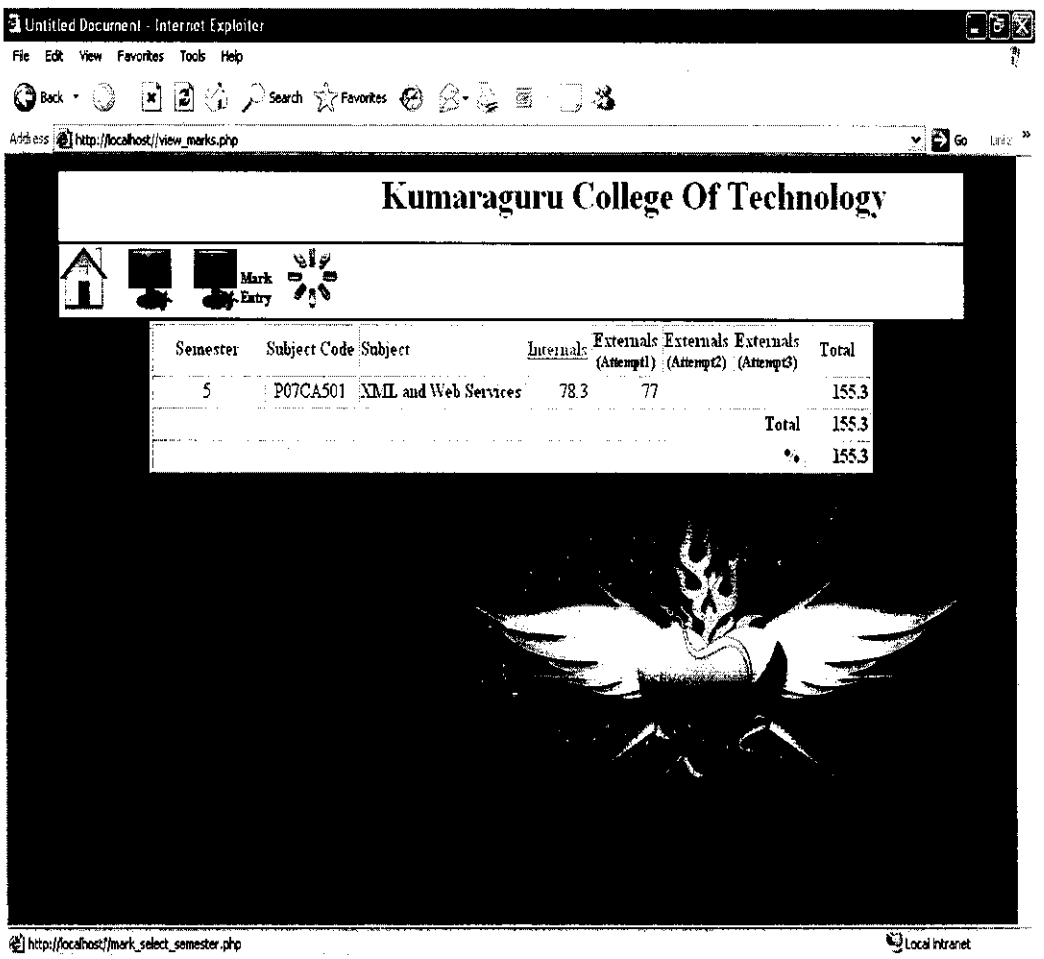


Fig 9.2.17 View Marks

- **Internal marks**

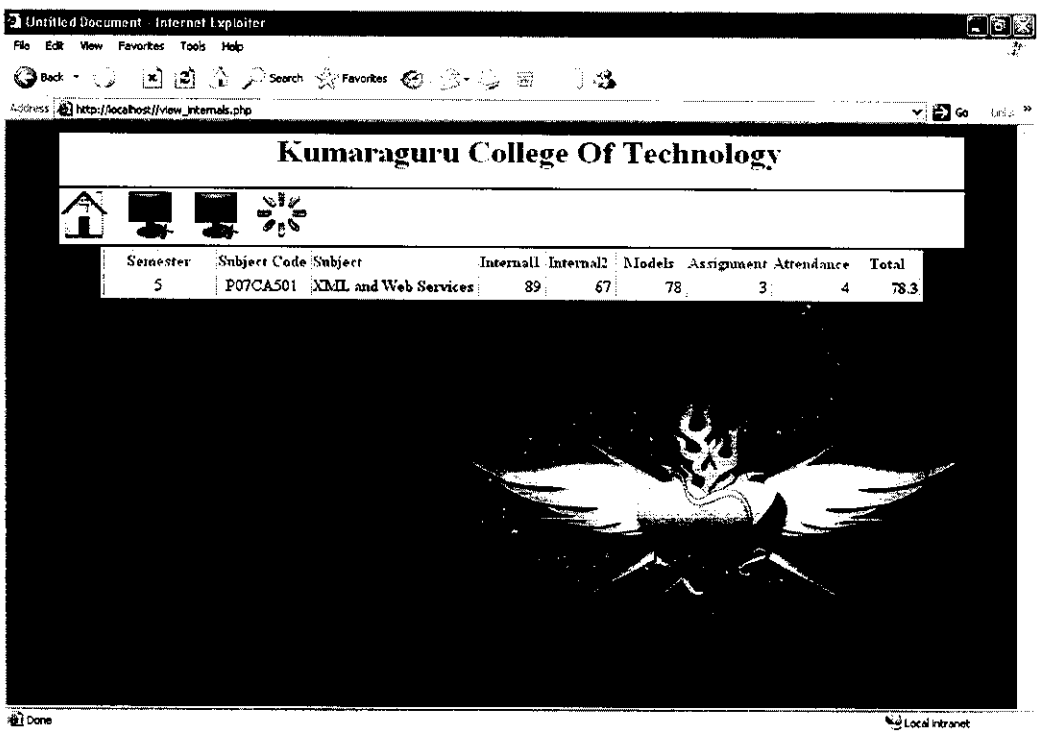


Fig 9.2.18 Internal Marks

CHAPTER 10
REFERENCES

1. BOOKS

- ✓ Michael K.Glass, Yann Le Scouarnee, Elizabeth Naramore, Gary Mailer, Jeremy Stolz, Jason Gerner, **"Beginning PHP, Apache, MySQL Web Development"** - Wiley Publishing,Inc., 2004 Edition
- ✓ Dave W.Mercer, Allan Kent, Steven D.Nowicki, David Mercer, Dan Squier, Wankyu Choi, **"Beginning PHP5"** - Wiley Publishing,Inc., 2004 Edition
- ✓ Robert Sheldon, Geoff Moes, **"Beginning MySQL"** - Wiley Publishing,Inc.
- ✓ Danny Goodman, Michael Morrison, Wiley dreamtech India Pvt. Ltd., **"Javascript Bible"** - 5th(2005) edition
- ✓ Steven Hilzner, **"HTML Black Book"** - Dreamtech Press

2. WEB

- ✓ www.wrox.com – PHP coding are referred for development of the system.
- ✓ www.mysql.com – referred for query commands and syntax for retrieving data in the table from the database.
- ✓ www.google.com – general references for JavaScript and HTML tags.