Web Portal for Student Information System

Contents

[1. Introduction 3](#_Toc291588032)

[1.1 Purpose 3](#_Toc291588033)

[1.2 Intended Audience and Reading Suggestions 3](#_Toc291588034)

[1.3 Product Scope 3](#_Toc291588035)

[1.4 Definitions, acronyms, and abbreviations 4](#_Toc291588036)

[1.5 References 4](#_Toc291588037)

[2. System Overview 5](#_Toc291588038)

[3. Design Considerations 6](#_Toc291588039)

[3.1 Assumptions and Dependencies 6](#_Toc291588040)

[3.2 General Constraints 7](#_Toc291588042)

[3.2.1 Testing 7](#_Toc291588043)

[3.2.2 Performance Requirements 9](#_Toc291588044)

[3.2.3 Security Requirements 9](#_Toc291588045)

[3.3 Goals 9](#_Toc291588046)

[4. Development Methods 9](#_Toc291588047)

[4.1 Microsoft SQL Server 9](#_Toc291588048)

[4.2 JAVA 1.6 11](#_Toc291588051)

[5. Architectural Strategies 13](#_Toc291588054)

[5.1 Table Structure 13](#_Toc291588055)

[5.2 ER-DIAGRAM 18](#_Toc291588056)

[5.3 USER INTERFACES 19](#_Toc291588057)

[6. List of Table and Figures 27](#_Toc291588058)

# Introduction

## Purpose

This SDS describes the various functions of our project. The actual design of the various modules and components of **Student Information System** is described in this document which takes care of various functionalities that our project aims to achieve and for determining the operating characteristics of the system.

.

## Intended Audience and Reading Suggestions

This SDS document is intended for any developers, project manager or documentation writer that needs to understand the basic system architecture and its specifications for development.

* **Developer**: The developer who wants to read, change, modify or add new requirements into the existing program, must firstly consult this document and update the requirements with appropriate manner so as to not destroy the actual meaning of them and pass the information correctly to the next phases of the development process.
* **Tester**: The tester needs this document to validate that the initial requirements of this web portal actually corresponds to the executable code correctly.

For each one of the reader types to better understand this document, here is a suggestion to read in this document:

* Overall description
* System Features
* External Interface Requirement
* Non Functional Requirement

## Product Scope

Student Information System (SIS) is a web-based application for students, faculty, academic staff and parents who want to get and retrieve student’s whole information instantly via internet. The major benefit of this web portal is to store the students information at one place (like SERVER) and it can be accessed via online interaction. The SIS web portal is to replace the old and traditional file(paper work) storing process. Instead of tedious paper work, students will be able to submit required information electronically, and the departments will be able to evaluate the submissions with a much quicker turnaround.

## Definitions, acronyms, and abbreviations

* SIS: Student Information System
* SERVER: Refers to the Host machine
* CLIENT: Refers to the user of SIS.
* SQL: Structured Query Language; used to retrieve information from a database.
* SQL Server: A server used to store data in an organized format.
* BOOLEAN: A true/false notation
* UNIQUE KEY: Used to differentiate entries in a database.
* LAYER: Represents a section of the project.
* DATA STORAGE LAYER: the section of the assignment referring to where all data is recorded.

## References

Books:-

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Title** | **Author** | **Edition** |
| 1 | JAVA Complete reference | Herbert Shield | Fourth Edition |
| 2 | J2EE Complete reference | Stephanie Bodoff  Debbie Bode Carson  Ian Evans | Third Edition |
| 3 | Database Management System | Korth | Second Edition |
| 4 | Software Engineering | Roger Pressman | Sixth Edition |

**Figure 1.1 Reference Table**

Websites:-

* www.java.sun.com
* www.javagroupsvc.com

# System Overview

Student Information System (SIS) is a web-based application that tracks current student’s academic information. It maintains academic information for ready access by office staff, students, their faculty advisors, and committee members. Instead of tedious paper work, students will be able to submit required information electronically, and the departments will be able to evaluate the submissions with a much quicker turnaround.

The Student Information System has been modularized into following modules.

**LOGIN MODULE**

The purpose of this module is to provide entry to the system or website. Based on the type of login, the user is provided with various facilities and functionalities. The main function of this module is to allow the user to use SIS. This module provides two types of login —Admin login and Student login.

**ADMINISTRATOR MODULE**

In this module when the administrator will enter his/her user name and password, then he/she will enter in to the administrator page and this page consists of two following sub modules.

* **Student Addition/ Updation / Deletion:** In SIS each Student is added, updated or deleted according to its branch.
* **Notice/Attendance/Result Generation:** In SIS information about notice, attendance and Internal result is generated.
* **Fee Detail and Schedules:** Fee information detail and schedule detail are managed.

**STUDENT MODULE**

In this module when a user enters his student id and password, then he can visit all the following pages.

* **Profile View:**

When the student clicks on this link he/she will get his/her information like student id, student name, password, father name, date of birth, nationality, city, address, country, phone number, mobile number, email. If he/she wants then he/she can change the profile.

* **Notice View:**

When the student clicks on this link, he can see latest notices released by the administrator.

* **Attendance View**:

When the student clicks on this one, the student can get his overall attendance percentage (present and absent).

* **Internal Results View:**

When the student clicks on this, he/she will get the internals result in all the subjects. How much grade point he/she secure out of 20 he/she can know.

* **Time Table View:**

When the student clicks on this link then he/she get all the information that on which day which room which class will be held at which time.

* **Fee Detail View:**

When the student clicks this link he/she can get all the fees structure semester wise and annual fee.

#### The Student Helpdesk:

This helpdesk is staffed by faculty who are there to help you. You may contact on (faculty phone no.).

# [Design Considerations](http://www.cmcrossroads.com/bradapp/docs/sdd.html#TOC_SEC6)

The design consideration section shows the overall description of all technical aspects which are related to this project including hardware and software.

## [Assumptions and Dependencies](http://www.cmcrossroads.com/bradapp/docs/sdd.html#TOC_SEC7)

Although basic password authentication and role based security mechanisms will be used to protect SIS from unauthorized access; functionality such as inserting, deleting and updating are assumed to be sufficiently protected under the existing security policies applied by the SIS developing team. It describes assumptions or dependencies regarding the software and its use. These may concern such issues as:

Related software:

* + Microsoft SQL Server 2005 to store the database.
  + JAVA 1.6, JSP, SERVLET, HTML to develop the product.
  + Virtual Server Apache Tomcat 6.0.18 and upward .

# Related hardware:

* + Microprocessor : Pentium-4 class processor, 2.2 GHz
  + Ram : 1 GB of RAM
  + Hard Disk : 10 gigabytes (GB) on installation drive,

Operating systems:

* + We use Windows Vista 32 bit Operating System for developing this system.

End-user characteristics

* + Every user must have basic knowledge of English.
  + He should be able to work with computer.
  + All users must have his/her unique login name and password for join web portal.
  + The user should know the details (meaning) of the operation.

## General Constraints

### Testing

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation.

Once source code has been generated, software must be tested to uncover as many errors as possible before delivery to the customer. Our goal is to design a series of test cases that have a high likelihood of finding errors. That’s where software testing techniques come into the picture. These techniques provide systematic guidance for designing tests that exercise the internal logic of software components and exercise the input and output domains of the program to uncover errors in program function, behavior and performance.

**TESTING OBJECTIVES**

A number of rules that can serve as testing objectives are: -

* Testing is a process of executing a program with the intent of finding an error.
* A good test case is one that a high probability of finding an as-yet-undiscovered error.
* A successful test is one that uncovers an as-yet-undiscovered error.

**TESTING TECHNIQUES**

**White Box Testing**

It is also called Glass Box Testing. It is a test case design method that uses control structure of the procedural design to derive test cases. Using White Box Testing methods, the software engineer can derive test cases that-

* Guarantee that all independent paths have been exercised at least once.
* Exercised all routes at their boundaries and within their operational bounds.
* Exercised all logical decision on their true and false sides.
* Exercised internal data structure to assure their validity.

**Black Box Testing**

Black Box Testing focuses on the functional requirements of the software. That is, Black Box Testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black Box Testing is not an alternative to White Box Testing. Rather, it is a complementary that is likely to uncover a different class of errors than White Box Testing. Black Box Testing attempts to find errors on the following categories –

In-correct or missing functions.

Interface errors.

Errors in data structures or external database access.

Performance errors.

Initialization and termination errors.

### Performance Requirements

Checking the fact that the system must perform as what every user expects .So in every action-response of the system, there are no immediate delays. In case of opening web forms, of popping error messages and saving the settings or sessions there is delay much below 2 seconds, In case of opening databases, sorting questions and computing there are no delays and the operation is performed in less than 2 seconds for opening, sorting, computing > 95% of the files. Also when connecting to the server the delay is based on the distance of the 2 systems and the configuration between them so there is high probability that there will be or not a successful connection in less than 20 seconds.

### Security Requirements

This program uses object oriented mechanisms to protect its data passed using methods also there is no currently a security schema of this program. Thus the log files that are being created are readable using a simple text reader.

## [Goals](http://www.cmcrossroads.com/bradapp/docs/sdd.html#TOC_SEC9)

* Quick and full information retrieval
* Time saving process
* High security level of data
* Multiuser environment
* Global access of data
* No repetition of data

# [Development Methods](http://www.cmcrossroads.com/bradapp/docs/sdd.html" \l "TOC_SEC10)

The following methods and approaches are used to develop this project.

## Microsoft SQL Server

A database management, or DBMS, gives the user access to their data and helps them transform the data into information. These systems allow users to create, update and extract information from their database.

A database is a structured collection of data. Data refers to the characteristics of people, things and events. SQL Server stores each data item in its own fields. In SQL Server, the fields relating to a particular person, thing or event are bundled together to form a single complete unit of data, called a record (it can also be referred to as raw or an occurrence). Each record is made up of a number of fields. No two fields in a record can have the same field name.

## SQL Server Tables

SQL Server stores records relating to each other in a table. Different tables are created for the various groups of information. Related tables are grouped together to form a database.

## Primary Key

Every table in SQL Server has a field or a combination of fields that uniquely identifies each record in the table. The Unique identifier is called the Primary Key, or simply the Key. The primary key provides the means to distinguish one record from all other in a table. It allows the user and the database system to identify, locate and refer to one particular record in the database.

* **Foreign Key**

When a field is one table matches the primary key of another field is referred to as a foreign key. A foreign key is a field or a group of fields in one table whose values match those of the primary key of another table.

* **Referential Integrity**

Not only does SQL Server allow you to link multiple tables, it also maintains consistency between them. Ensuring that the data among related tables is correctly matched is referred to as maintaining referential integrity.

* **Relational Database**

Sometimes all the information of interest to a business operation can be stored in one table. SQL Server makes it very easy to link the data in multiple tables. Matching an employee to the department in which they work is one example. This is what makes SQL Server a relational database management system, or RDBMS. It stores data in two or more tables and enables you to define relationships between the table and enables you to define relationships between the tables.

* **Data Abstraction**

A major purpose of a database system is to provide users with an abstract view of the data. This system hides certain details of how the data is stored and maintained. Data abstraction is divided into three levels.

* + **Physical level:** This is the lowest level of abstraction at which one describes how the data are actually stored.
  + **Conceptual Level:**  At this level of database abstraction all the attributed and what data are actually stored is described and entries and relationship among them.
  + **View level:**  This is the highest level of abstraction at which one describes only part of the database.

**Advantages of RDBMS**

1. Redundancy can be avoided
2. Inconsistency can be eliminated
3. Data can be Shared
4. Standards can be enforced
5. Security restrictions ca be applied
6. Integrity can be maintained
7. Conflicting requirements can be balanced
8. Data independence can be achieved.

**Disadvantages of RDBMS**

A significant disadvantage of the RDBMS system is cost. In addition to the cost of purchasing of developing the software, the hardware has to be upgraded to allow for the extensive programs and the workspace required for their execution and storage. While centralization reduces duplication, the lack of duplication requires that the database be adequately backed up so that in case of failure the data can be recovered.

## JAVA 1.6

The most important characteristic of Java is that it was designed from the outset to be machine independent. We can run Java programs unchanged on any machine and operating system combination that supports Java. Java programs are intrinsically more portable than programs written in other languages. An application written in Java will only require a single set of source code statements, regardless of the number of different computer platforms on which

it is run, so it is very useful for internet application.

* Platform independence - Java programs can be run on many platforms without modification. This portability is assured by using a Virtual machine1. When a Java program is compiled byte-code is created rather than a standard executable file. Effectively, this is machine code for a virtual machine, which is then interpreted by the Java interpreter. The byte-code can be run on any platform which has a suitable interpreter.
* Security - since Java has always been designed with distributed applications in mind, security has been incorporated right from the start, and if anything this has been seen to be too restrictive.

**Java and the JDK**

The Java Development Kit contains all the necessary tools for the development of Java applications and applets, including a compiler, interpreter, class libraries, applet viewer and debugger. The current version of the JDK, 1.4.2, is freely available from Sun's web she- for machines running Windows 95, Windows NT, Solaris SPARC and Solaris x86, and a version for the Macintosh is expected later this year (version 1,0 is currently available). Ports to other machines may also be available from other sources. It should be noted that browsers may not yet support applets created using features in the latest JDK.

**Characteristics of Java**

* Simple
* Secure
* Portable
* Object-oriented
* Robust
* Multithreaded
* Architecture-neutral
* Interpreted
* High performance
* Distributed
* Dynamic

**J2EE**

The Java-2 Enterprise Edition (J2EE™) provides a component-based approach to the design, development, assembly, and deployment of enterprise applications. The J2EE platform offers a multitiered distributed application model, reusable components, a unified security model, flexible transaction control, and web services support through integrated data interchange on Extensible Markup Language (XML)-based open standards and protocols.

**Components of J2EE**

* **SERVLET**

Java servlets are small, platform-independent Java programs that can be used to extend the functionality of a Web server in a variety of ways. Servlets are to the server what applets are to the client—small Java programs compiled to bytecode that can be loaded dynamically and that extend the capabilities of the host. It is a server side programming language.

* **JSP**

Java Server Pages is the extension of servlet to simplify the programming and coding of servlet. It includes HTML tags to make coding easy. Ultimately it convert into the servlet at the time of calling.

**HTML**

Without HTML, the World Wide Web wouldn’t exist. HTML allow the individual elements on the Web to be brought together and assented as a collection. Text, images, multimedia, and other files can be packed together using HTML. This section explains the basic principles behind the interaction between HTML and the World Wide Web.

# [Architectural Strategies](http://www.cmcrossroads.com/bradapp/docs/sdd.html#TOC_SEC11)

## Table Structure

Here in below tables \* is used for denoting Foreign Key and # for Primary Key.

**Table 1:- Login**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Size(Bytes)** | **Constraint** |
| UserID# | Nchar | 10 | Not Null |
| Password | Varchar | 20 | Not Null |

**Table 2:- Personal Profile**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Size(In Byte)** | **Constraint** |
| College\_ID# | Nchar | 10 | Not Null |
| Stud\_name | Varchar | 50 | Not Null |
| F\_Name | Varchar | 50 | Not Null |
| An\_In\_father | Decimal | 1,0 | Not Applied |
| DOB | DateTime | Std Size | Not Null |
| Nationality | Varchar | 50 | Not Applied |
| Gender | Char | 1 | Not Null |
| Category | Char | 10 | Not Null |
| Marr\_status | Char | 1 | Not Applied |
| Area\_R\_U | Char | 1 | Not Applied |
| PAddress | Varchar | 100 | Not Null |
| City | Varchar | 50 | Not Null |
| Dist | Varchar | 50 | Not Applied |
| Pincode | Decimal | 6,0 | Not Applied |
| State | Varchar | 50 | Not Null |
| Country | Varchar | 50 | Not Null |
| Phone\_No1 | Decimal | 12,0 | Not Applied |
| Email | Varchar | 50 | Not Applied |
| LAddress | Varchar | 100 | Not Applied |
| L\_G\_Address | Varchar | 100 | Not Applied |
| Phone\_No2 | Decimal | 12,0 | Not Applied |
| Nearest\_Station | Varchar | 50 | Not Applied |

**Table 3:- Academic Profile**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Size(Bytes)** | **Constraint** |
| College\_ID\* | Nchar | 10 | Not Null |
| DOJ | Datetime | Std Size | Not Null |
| Current\_sem | Decimal | 1,0 | Not Applied |
| \*Course\_Code | Char | 5 | Not Null |
| Sec\_per | Decimal | (2,2) | Not Null |
| Sr\_sec\_per | Decimal | (2,2) | Not Null |
| Entrance\_exam | Varchar | 10 | Not Null |
| Rank | Decimal | 5,0 | Not Applied |

**Table 4:- Course Code**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Size(Bytes)** | **Constraint** |
| Course\_code# | Char | 5 | Not Null |
| Course\_name | Varchar | 50 | Not Null |
| Duration | Decimal | 1,0 | Not Null |

**Table 5:- Attendance**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Size(Bytes)** | **Constraint** |
| Collage\_ID\* | Nchar | 10 | Not Null |
| Sub1 | Char | 5 | Not Null |
| Att1 | decimal | 2,0 | Not Null |
| Sub2 | Char | 5 | Not Null |
| Att2 | decimal | 2,0 | Not Null |
| Sub3 | Char | 5 | Not Null |
| Att3 | decimal | 2,0 | Not Null |
| Sub4 | Char | 5 | Not Null |
| Att4 | decimal | 2,0 | Not Null |
| Sub5 | Char | 5 | Not Null |
| Att5 | decimal | 2,0 | Not Null |
| Sub6 | Char | 5 | Not Null |
| Att6 | Decimal | 2,0 | Not Null |

**Table 6:- Internal Marks**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Size(Bytes)** | **Constraint** |
| College\_ID\* | Nchar | 10 | Not Null |
| Sub1 | Char | 5 | Not Null |
| Mark1 | decimal | 2,0 | Not Null |
| Sub2 | Char | 5 | Not Null |
| Mark2 | decimal | 2,0 | Not Null |
| Sub3 | Char | 5 | Not Null |
| Mark3 | decimal | 2,0 | Not Null |
| Sub4 | Char | 5 | Not Null |
| Mark4 | decimal | 2,0 | Not Null |
| Sub5 | Char | 5 | Not Null |
| Mark5 | Decimal | 2,0 | Not Null |
| Sub6 | Char | 5 | Not Null |
| Mark6 | decimal | 2,0 | Not Null |

**Table 7:- Fee**

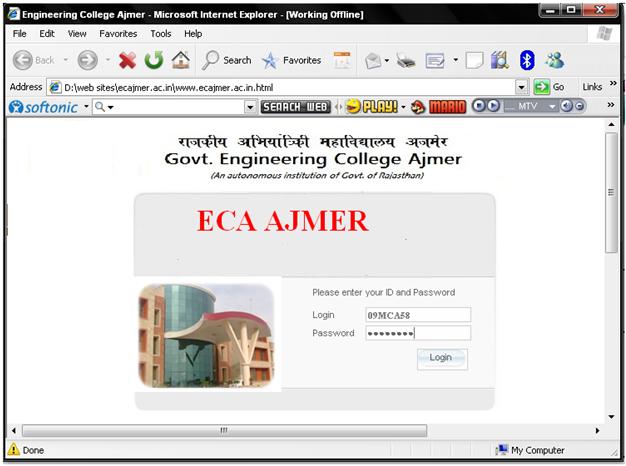
|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Size(Bytes)** | **Constraint** |
| College\_ID\* | Nchar | 10 | Not Null |
| H\_Fee | decimal | 5,0 | Not Null |
| T\_Fee | Decimal | 5,0 | Not Null |
| H\_Date | Datetime | Std Size | Not Null |
| T\_Date | Datetime | Std Size | Not Null |

## ER-DIAGRAM

**Figure: 5.1 E-R Diagram**

## USER INTERFACES

LOGIN SCREEN



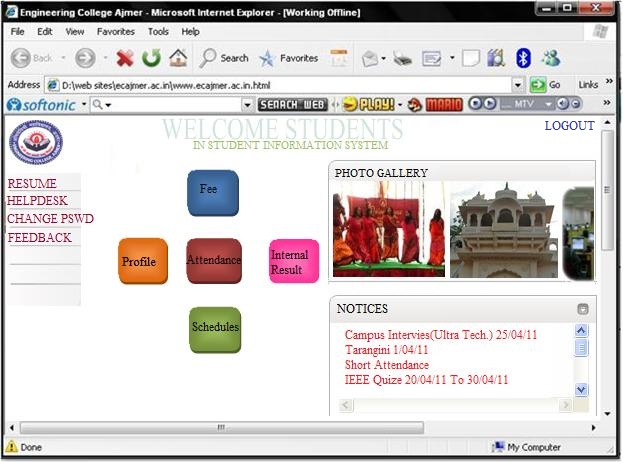
**Figure: 5.2 Login Screen**

**Description:**

a. Enter user ID in Caps for student/Admin/Staff login (eg. 09MCA11/Admin/Staff) .

b. Enter password provided by admin for student’s login and for staff or admin’s login.

HOME PAGE

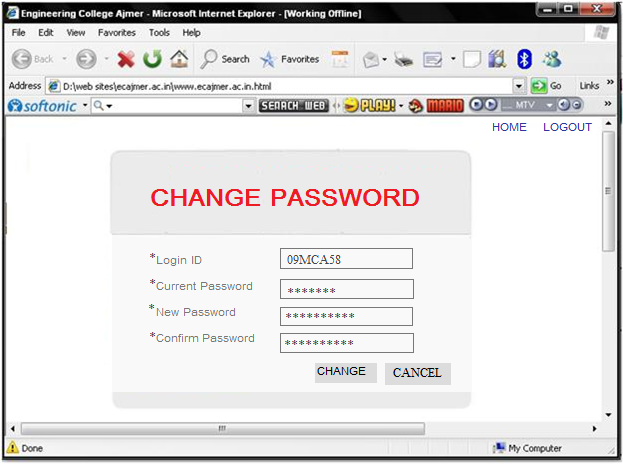


**Figure: 5.3 Home Page**

**Description:**

This is the home page of Student Information System.

PASSWORD CHANGE FORM

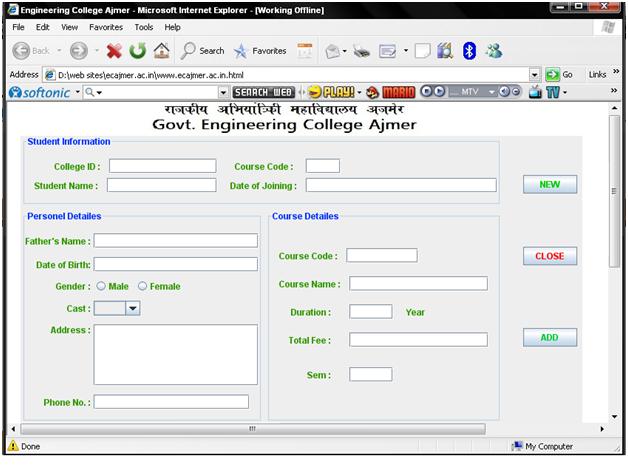


**Figure: 5.4 Change Password Screen**

**Description:**

After login student can change his password by selecting the change password option in main menu of home page.

PROFILE CREATION FORM

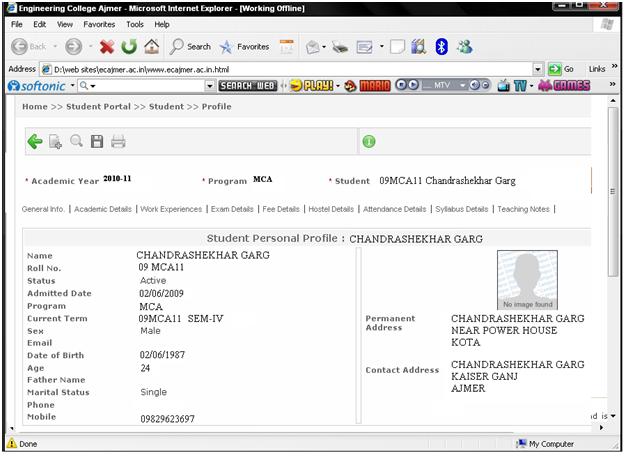


**Figure: 5.5 Profile Creation**

**Description:**

On clicking the profile button first choice is for creating the profile or for updating the existing profile. In this form student upload his personal and academic information.

PROFILE VIEW FORM

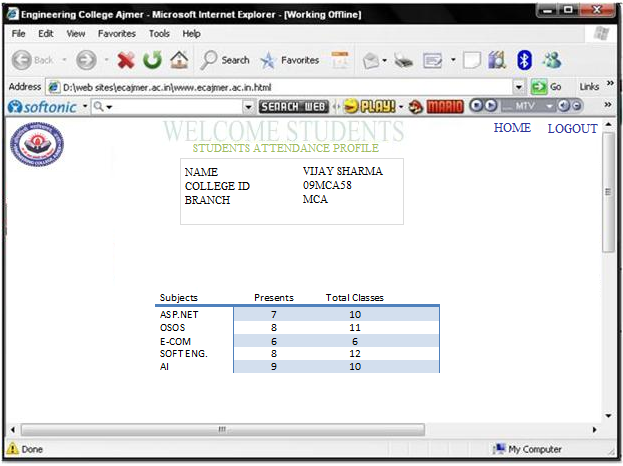


**Figure: 5.6 Profile View Screen**

**Description:**

This is the second choice of profile button. This form shows the overall details of student including personal and academic details.After filling the above form this form is generated.

ATTENDANCE DETAIL FORM

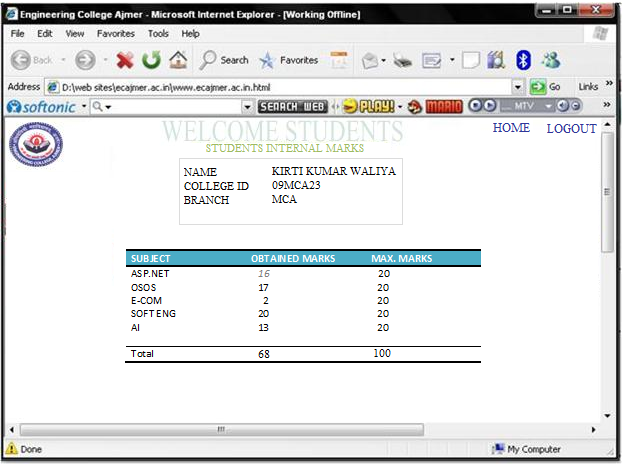


**Figure: 5.7 Attendance Detail Screen**

**Description:**

On clicking the attendance button this form is viewed. This form shows the subject wise student attendance.

INTERNAL MARKS DETAIL FORM

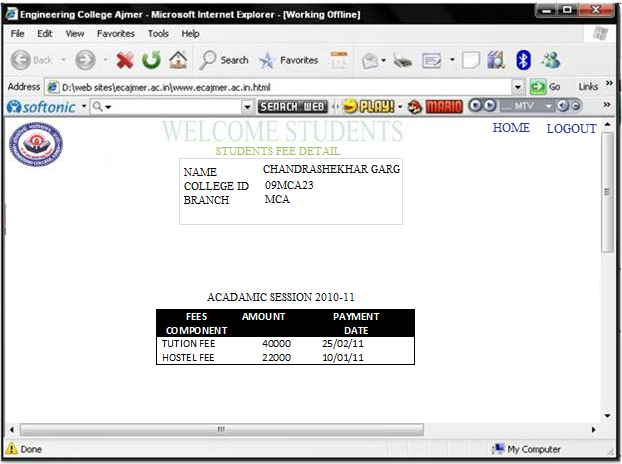


**Figure: 5.8 Internal Marks Screen**

**Description:**

On clicking Internal marks button student viewed this form. This from shows subject wise marks of the student.

FEE DETAIL FORM



**Figure: 5.9 Fee Detail Screen**

**Description:**

On clicking fee button this form is appeared. This form shows current session fees detail of student.

# List of Table and Figures

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Name** | **Figure No.** | **Page** |
| 1 | Reference Table | 1.1 | 4 |
| 2 | Login Table | 1 | 13 |
| 3 | Personal Profile Table | 2 | 14 |
| 4 | Academic Profile Table | 3 | 15 |
| 5 | Course Code Table | 4 | 15 |
| 6 | Attendance Table | 5 | 15 |
| 7 | Internal Marks Table | 6 | 16 |
| 8 | Fee Detail Table | 7 | 16 |
| 9 | ER Diagram | 5.1 | 18 |
| 10 | Login Screen | 5.2 | 19 |
| 11 | Home Page Screen | 5.3 | 20 |
| 12 | Change Password | 5.4 | 21 |
| 13 | Profile Creation | 5.5 | 22 |
| 14 | Profile View | 5.6 | 23 |
| 15 | Attendance Detail Screen | 5.7 | 24 |
| 16 | Internal Marks | 5.8 | 25 |
| 17 | Fee Detail Screen | 5.9 | 26 |