LAB REPORT

Submitted by

Kalimisetty Sashank [RA2011003010649]

Under the Guidance of

Mr. U. M. Prakash

Assistant Professor, Computing Technologies

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING

with specialization in CSE CORE

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

JUNE 2022

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this lab report titled **"Student Database Management System"** is the bonafide work done by Kalimisetty Sashank (RA2011003010649) who carried out the lab exercises under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

SIGNATURE

Mr. U. M. Prakash

SEPM – Course Faculty

Assistant Professor

Department of Computing Technologies

ABSTRACT

Students form a main part of any institution that concerns with. But the institutions find it difficult to keep details of so many students of the organization just in one stretch. It will involve a lot of pen paper work. Sometimes there will be some huge heap of files bundled up and kept together in some corner of the office. If you want any information regarding the particular student then it can be obtained by just entering the roll number or the name of the student to be searched. This student management system will make the work of storing the data in an organized way. The student management system application will help in managing the student's reports, results and exams will become easier with one such system. It will also help in saving time and effort. The user interface must be user friendly and easy to understand. The information of the particular student will be obtained in just one mouse click. Some of the features that it can include are as follows:

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	iii
	LIST OF FIGURES	v
	LIST OF ABBREVIATIONS	vi
1	PROBLEM STATEMENT	1
2	STAKEHOLDERS & PROCESS MODELS	2
3	IDENTIFYING REQUIREMENTS	3
4	PROJECT PLAN & EFFORT	4
5	WORK BREAKDOWN STRUCTURE & RISK ANALYSIS	7
6	SYSTEM ARCHITECTURE, USE CASE & CLASS DIAGRAM	12
7	ENTITY RELATIONSHIP DIAGRAM	14
8	DATA FLOW DIAGRAM	15
9	SEQUENCE & COLLABORATION DIAGRAM	16
10	DEVELOPMENT OF TESTING FRAMEWORK/USER INTERFACE	19
11	TEST CASES & REPORTING	22
12	ARCHITECTURE/DESIGN/FRAMEWORK/IMPLE -MENTATION	24
	CONCLUSION	28
	REFERENCES	29
	APPENDIX (CODE)	30

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
1	Work breakdown structure	7
2	Timeline chart	9
3	System Architecture	12, 24
4	Class Diagram	13
5	Use Case Diagram	13
6	Entity relationship diagram	14
7	Data Flow Diagram	15
8	Sequence Diagram	16
9	Collaboration Diagram	17
10	Framework	24

LIST OF ABBREVIATIONS

S No	ABBREVIATIONS	FULL FORM
1	WBS	Work Breakdown Sturcture
2	ER Diagram	Entity Relationship Diagram
3	SWOT	Strength Weakness Oppurtunity Threat
4	RMMM	Risk Mitigation, Monitoring and Management
5	UI	User Interface
6	ER	Entity Relationship

CHAPTER 1

PROBLEM STATEMENT

The current problem in the standard schools is that they did not have systematic data arrangement in the student management. When the staff of the administrator wants to record the data of the student, they need to fill it out by using the manual system. In this case data might be lost when several problems occur. Another problem that can happen is hard to search and update the student information and class arrangement. Teachers also have some problems as administrators that use manual systems including to develop the student performance example student discipline, result, attendance and so on that have no systematic record. The systematic requirement is required so that all data is stored into the database for future reference and enhancement. Below is the specific problem statement that occurs in standard school via using a manual system. i) Lack of data arrangement that is recorded by using a manual system and using a lot of paper to record the student information, student result and performance. ii) The manual system is hard to search and update about the student information, result and performance iii) The manual system is not providing the security of the academic information that might be lost. iv) Some information released by the school is not known by the parents or teachers.

CHAPTER 2

STAKEHOLDERS & PROCESS MODELS

Selection of Methodology

Agile Model

- **The meaning of Agile is swift or versatile."Agile process model" refers to a software development approach based on iterative development.**
- **Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning.**
- **The project scope and requirements are laid down at the beginning of the development process.**
- **Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.**

CHAPTER 3

IDENTIFYING REQUIREMENTS

System Requirements

It is an aspect of what the proposed system must do to do this project (STUDENT DATABASE MANGEMENT SYSTEM) all requires

Hardware Requirements:

System: P IV or above.

RAM: 4 GB or above.

Hard Disk: 1 TB or above.

Software Requirements:

Operating System: Windows & or above.

Software platform:

Visual studio 2010, SQL Server,C++.

Functional Requirements

"Any Requirement Which Specifies What The System Should Do."

1. **User Login:**
2. **Adminstration:**
3. **Register NewUser:**

Non-Functional Requirements

"Any Requirement That Specifies How The System Performs A Certain Function."

- Storing as many details as possible.
- Can be able to access the software anywhere they want.
- With individual login and keeping the details secure.
- Less Chances of errors in software.

CHAPTER 4

PROJECT PLAN & EFFORT

Requirements

1. Project Management Plan

Describe the key issues driving the project. **[Min 3 Focus Areas]**

Focus Area	Details
Schedule Management	Define Milestones Schedule Control
Quality Management	Team manager: Ch.Kalyan Gupta Backend Dev:S.Vishnu tejas Frontend Dev:K.Sashank Model Acquired: AGILE model
Resource Management	Estimate and Manage the need People:3 People Skills Required :C++, Python Finance: 10,00,000 Physical: Home,Library

2. Estimation

2.1. Effort and Cost Estimation

Activity Description	Sub-Task	Sub-Task Description	Effort (in hours)	Cost in INR
Design the user screen	E1R1A1T1 (Effort-Requirement-Activity-Task)	Confirm the user requirements (acceptance criteria)	3	1500
	E1R1A1T2	Analysis of details	3	1500
	E1R1A1T3	Focusing and designing on looks and style	5	2500
Identify Data Source for displaying units of Energy Consumption	E1R1A2T1	Go through Interface contract (Application Data Exchange) documents	5	2500
	E1R1A2T2	Document	1	500

Effort (hr)	Cost (INR)
1	500

2.2. Infrastructure/Resource Cost [CapEx]

< OneTime Infra requirements >

Infrastructure Requirement	Qty	Cost per qty	Cost per item
Computers	3	60,000	1,80,000
Hard disks	3	3,500	10,500

2.3 Maintenance and Support Cost [OpEx]

Category	Details	Qty	Cost per qty per annum	Cost per item
People	Network, System, Middleware and DB admin Developer , Support Consultant	3	2,000,000	6,000,000
License	Operating System Database Middleware IDE	10	10000	100,000
Infrastructures	Server, Storage and Network	20	20000	400,000

3. Project Team Formation

3.1. Identification Team members

Name	Role	Responsibilities
Vishnu Tejas	Key Business User (Product Owner)	Provide clear business and user requirements
Kalyan Gupta	Project Manager	Manage the project
Sasank	Business Analyst	Discuss and Document Requirements
Aryan Jordan	Technical Lead	Design the end-to-end architecture
Harihaaran	Frontend Developer	Develop user interface
Ragunath	Backend Developer	Design, Develop and Unit Test Services/API/DB
Haarish	Cloud Operations	Provision required Services
Prakash	Tester	Define Test Cases and Perform Testing

3.2. Responsibility Assignment Matrix

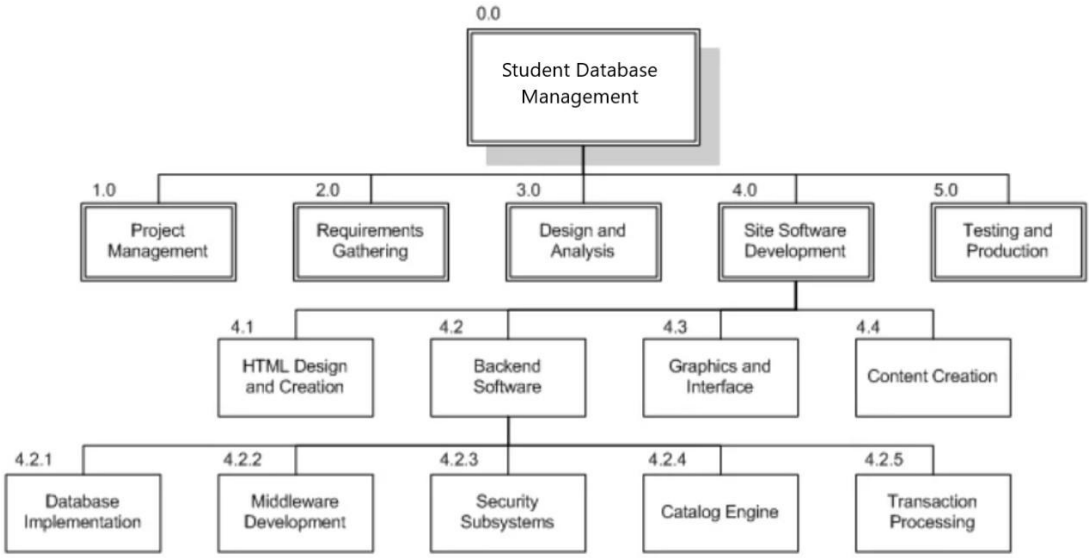
RACI Matrix	Team Members			
Activity	**Name (BA)**	**Name (Developer)**	**Name (Project Manager)**	**Key Business User**
User Requirement Documentation	A	Vishnu Tejas	Kaalyan Gupta	K. Sashank
Analysis	A	C/I	I	R
Design GUI	A	C/I	I	R
Construction	A	C/I	I	R
RTesting	A	C/I	I	R

A	Accountable
R	Responsible
C	Consult
I	Inform

CHAPTER 5

WORK BREAKDOWN STRUCTURE & RISK ANALYSIS

WBS :



Work breakdown structure (WBS) in project management is a method for completing a complex, multi-step project. It's a way to divide and conquer large projects to get things done faster and more efficiently. The goal of a WBS is to make a large project more manageable. Breaking it down into smaller chunks means work can be done simultaneously by different team members, leading to better team productivity and easier project management. WBS for house price prediction using python and machine learning: project stages:

- importing libraries and dataset
- exploring and pre-processing the dataset
- model implementation
- model testing

0.0 Student Database Management

1.0 Project Management

2.0 Requirements Gathering

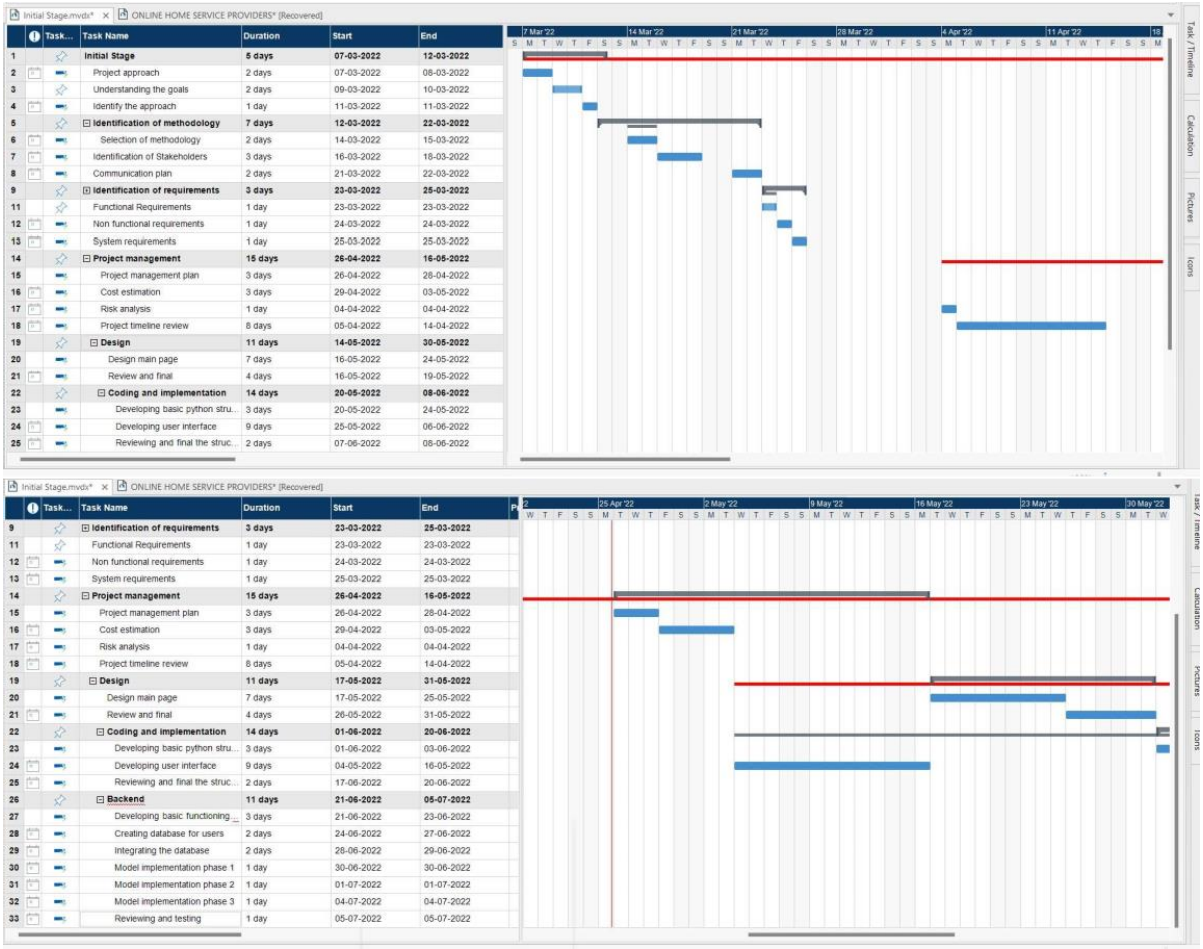
3.0 Analysis & Design

4.0 Site Software Development

- 4.1 HTML Design and Creation
- 4.2 Backend Software
  - 4.2.1 Database Implementation
  - 4.2.2 Middleware Development
  - 4.2.3 Security Subsystems
  - 4.2.4 Catalog Engine
  - 4.2.5 Transaction Processing
- 4.3 Graphics and Interface
- 4.4 Content Creation

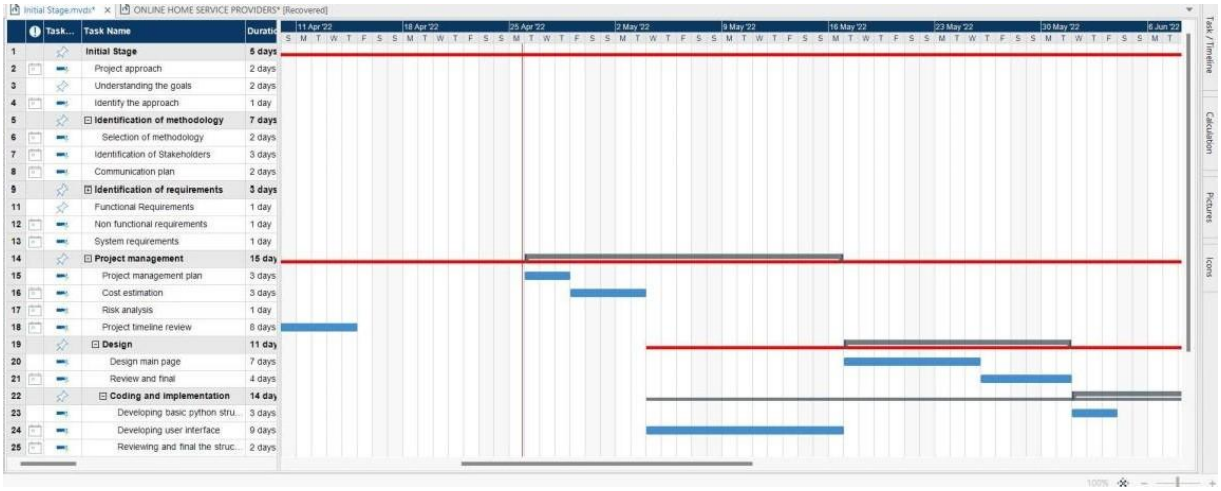
5.0 Testing and Production

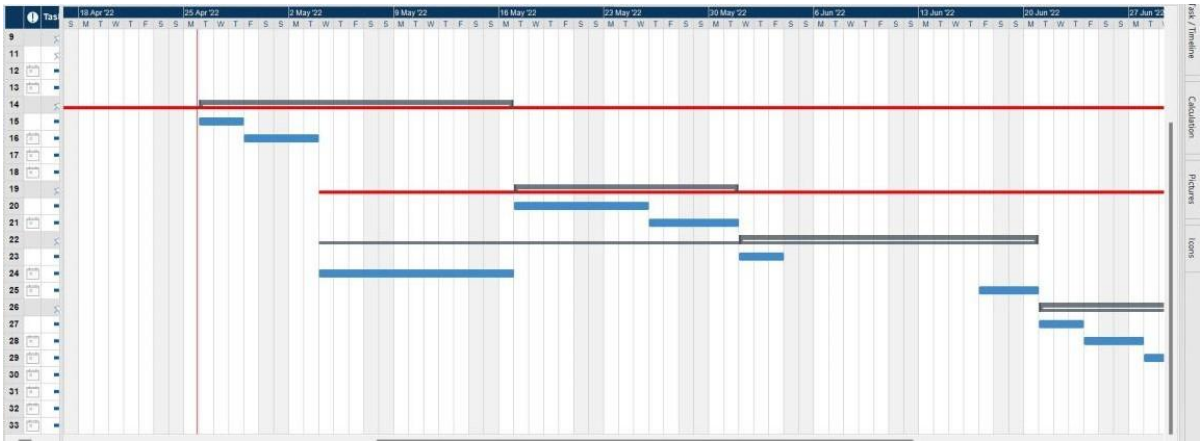
TIMELINE – GANTT CHART



	Task...	Task Name	Duration	Start	End
1		**Initial Stage**	**5 days**	**07-03-2022**	**12-03-2022**
2		Project approach	2 days	07-03-2022	08-03-2022
3		Understanding the goals	2 days	09-03-2022	10-03-2022
4		Identify the approach	1 day	11-03-2022	11-03-2022
5		**Identification of methodology**	**7 days**	**12-03-2022**	**22-03-2022**
6		Selection of methodology	2 days	14-03-2022	15-03-2022
7		Identification of Stakeholders	3 days	16-03-2022	18-03-2022
8		Communication plan	2 days	21-03-2022	22-03-2022
9		**Identification of requirements**	**3 days**	**23-03-2022**	**25-03-2022**
11		Functional Requirements	1 day	23-03-2022	23-03-2022
12		Non functional requirements	1 day	24-03-2022	24-03-2022
13		System requirements	1 day	25-03-2022	25-03-2022
14		**Project management**	**15 days**	**26-04-2022**	**16-05-2022**
15		Project management plan	3 days	26-04-2022	28-04-2022
16		Cost estimation	3 days	29-04-2022	03-05-2022
17		Risk analysis	1 day	04-04-2022	04-04-2022
18		Project timeline review	8 days	05-04-2022	14-04-2022
19		**Design**	**11 days**	**14-05-2022**	**30-05-2022**
20		Design main page	7 days	16-05-2022	24-05-2022
21		Review and final	4 days	16-05-2022	19-05-2022
22		**Coding and implementation**	**14 days**	**20-05-2022**	**08-06-2022**
23		Developing basic python stru...	3 days	20-05-2022	24-05-2022
24		Developing user interface	9 days	25-05-2022	06-06-2022
25		Reviewing and final the struc...	2 days	07-06-2022	08-06-2022

	Task...	Task Name	Duration	Start	End
9		**Identification of requirements**	**3 days**	**23-03-2022**	**25-03-2022**
11		Functional Requirements	1 day	23-03-2022	23-03-2022
12		Non functional requirements	1 day	24-03-2022	24-03-2022
13		System requirements	1 day	25-03-2022	25-03-2022
14		**Project management**	**15 days**	**26-04-2022**	**16-05-2022**
15		Project management plan	3 days	26-04-2022	28-04-2022
16		Cost estimation	3 days	29-04-2022	03-05-2022
17		Risk analysis	1 day	04-04-2022	04-04-2022
18		Project timeline review	8 days	05-04-2022	14-04-2022
19		**Design**	**11 days**	**17-05-2022**	**31-05-2022**
20		Design main page	7 days	17-05-2022	25-05-2022
21		Review and final	4 days	26-05-2022	31-05-2022
22		**Coding and implementation**	**14 days**	**01-06-2022**	**20-06-2022**
23		Developing basic python stru...	3 days	01-06-2022	03-06-2022
24		Developing user interface	9 days	04-05-2022	16-05-2022
25		Reviewing and final the struc...	2 days	17-06-2022	20-06-2022
26		**Backend**	**11 days**	**21-06-2022**	**05-07-2022**
27		Developing basic functioning ...	3 days	21-06-2022	23-06-2022
28		Creating database for users	2 days	24-06-2022	27-06-2022
29		Integrating the database	2 days	28-06-2022	29-06-2022
30		Model implementation phase 1	1 day	30-06-2022	30-06-2022
31		Model implementation phase 2	1 day	01-07-2022	01-07-2022
32		Model implementation phase 3	1 day	04-07-2022	04-07-2022
33		Reviewing and testing	1 day	05-07-2022	05-07-2022





RISK ANALYSIS – SWOT & RMMM

RISK ANAYLSIS:

Risk Management is the system of identifying addressing and eliminating these problems before they can damage the project.house has both consumption and investment properties (demands), and it is very hard to illustrate the consistency between models when financial theories are embedded into economic theory. Risk element cannot be directly introduced into the standard housing model, especially the housing life-cycle model because of the underlying assumption of certainty. To be more realistic,expected housing capital return is redefined to be uncertain and thenhouseholds maximize the expectation of their uncertain future lifetime utility

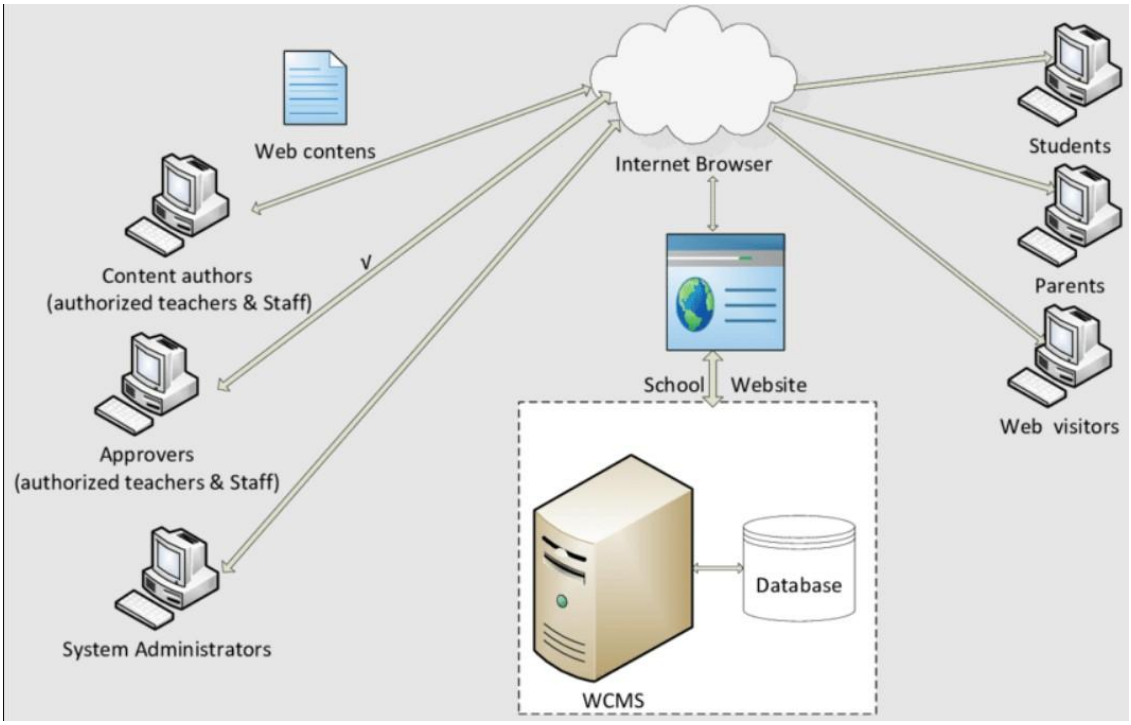
Strengths	**Weakness**
• Prompt services • Save time and money • Price range	• High tech equipments • Cannot show emotions • Limited availability of data
Opportunities	**Threats**
• New rules on the environment • Changes in demographics • Improve Lead Generation • Scaling up success	• Government Regulations • Expensive • Lack of long-term supplier contracts

Response	Strategy	Examples
Avoid	1. Updating information database on students and faculty. 2. Updating the system.	1. Extending the schedule 2. Reducing/reducing scope 3. Change the extension strategy
Transfer	1. Backup of database in cloud. 2. Giving a condition on the website to get a good internet connection.	1. Transferring database. 2. Transferring faculty user data.
Mitigate	1. Equip the server with enough resources to prevent possible or unexpected crashes. 2. Backup database.	1. Better services and resources mean the system can run smoothly and not interrupt anything for the user. 2. Databases may have a risk of getting corrupted at times, backups can help in patching this up.
Accept	1. Hostfixes/patch-ups. 2. Workload for the servers.	1. Contingency reserve budgets 2. Management schedule float 3. Event contingency

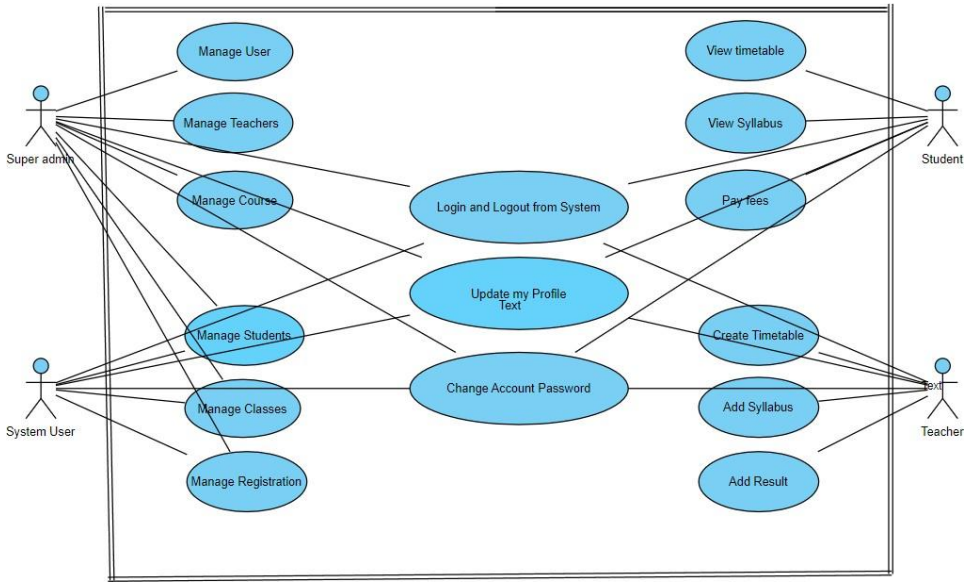
CHAPTER 6

SYSTEM ARCHITECTURE, USE CASE & CLASS DIAGRAM

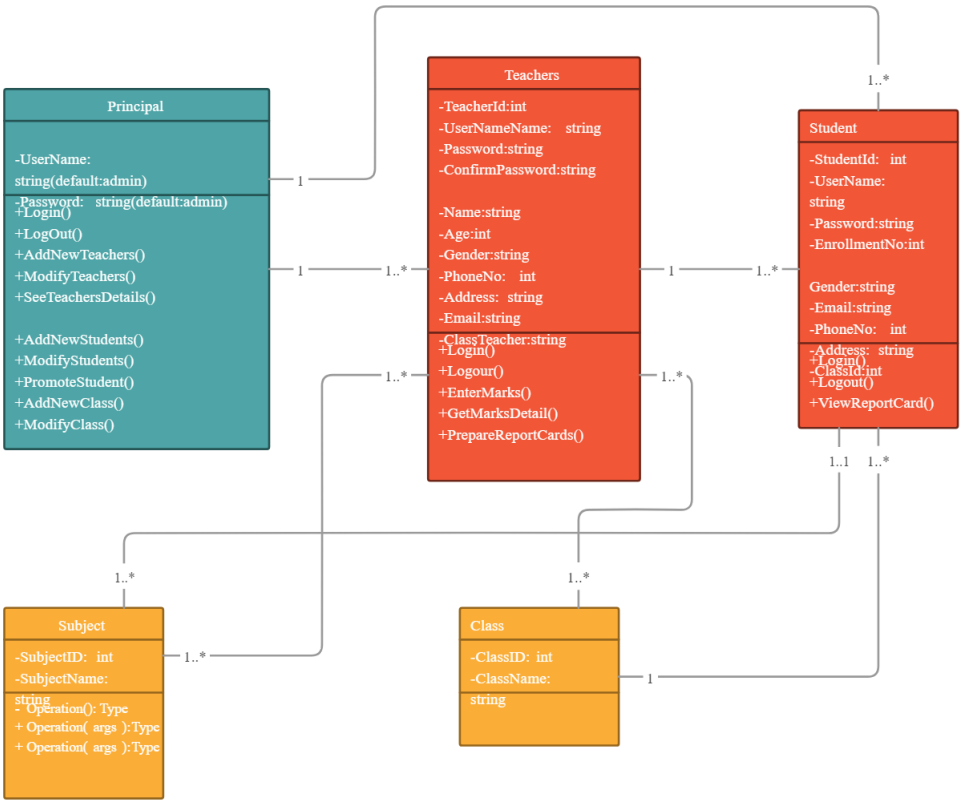
SYSTEM ARCHITECTURE :



USE CASE DIAGRAM :



CLASS DIAGRAM :

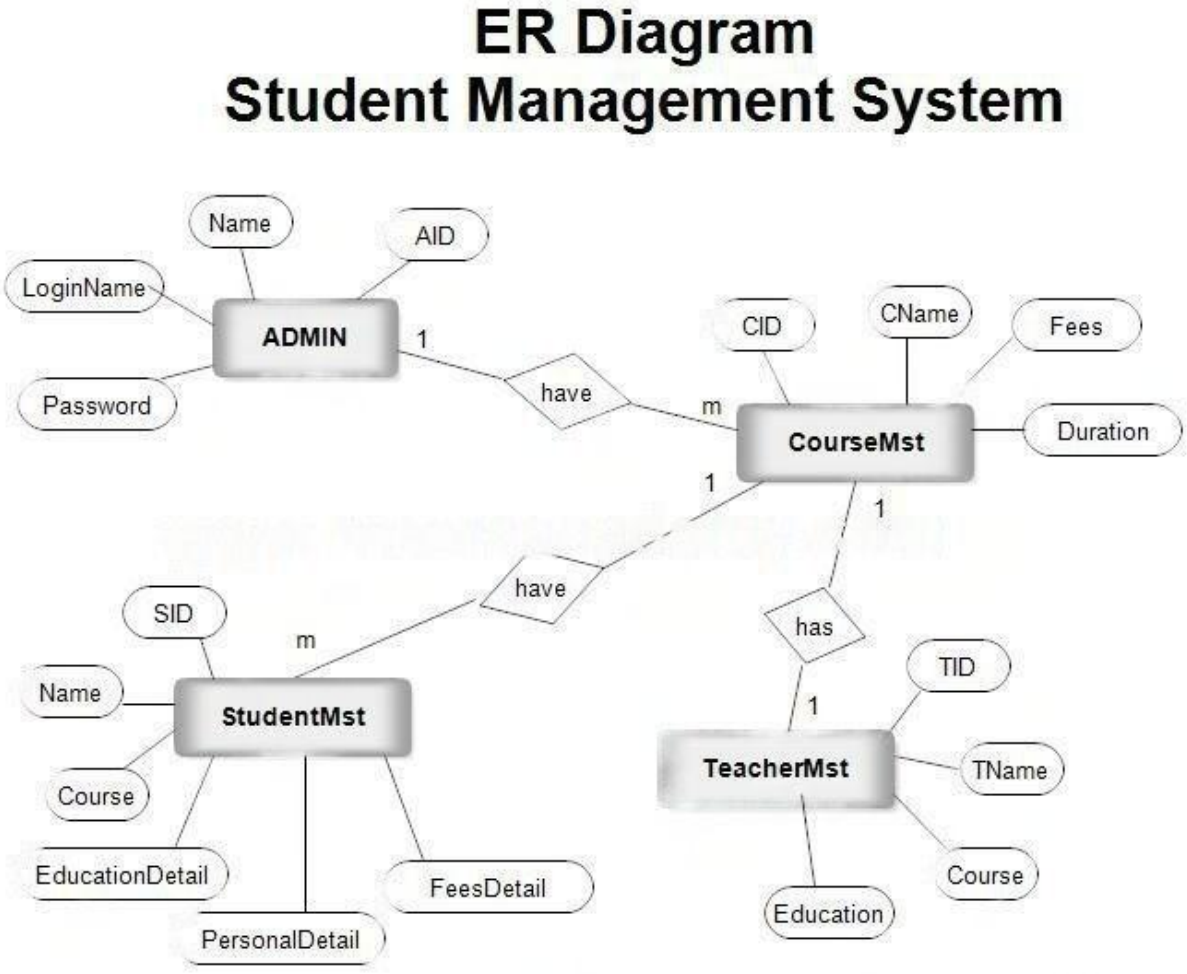


CHAPTER 7

ENTITY RELATIONSHIP DIAGRAM

ER Diagram :

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system.



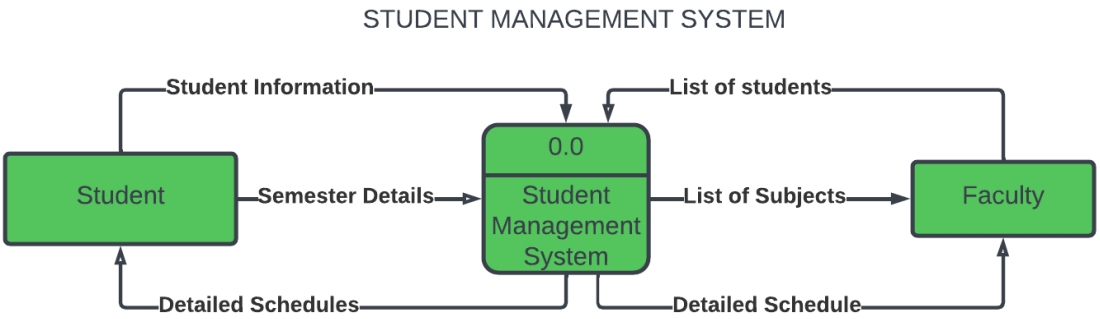
CHAPTER 8

DATA FLOW DIAGRAM

Data FLow Diagram :

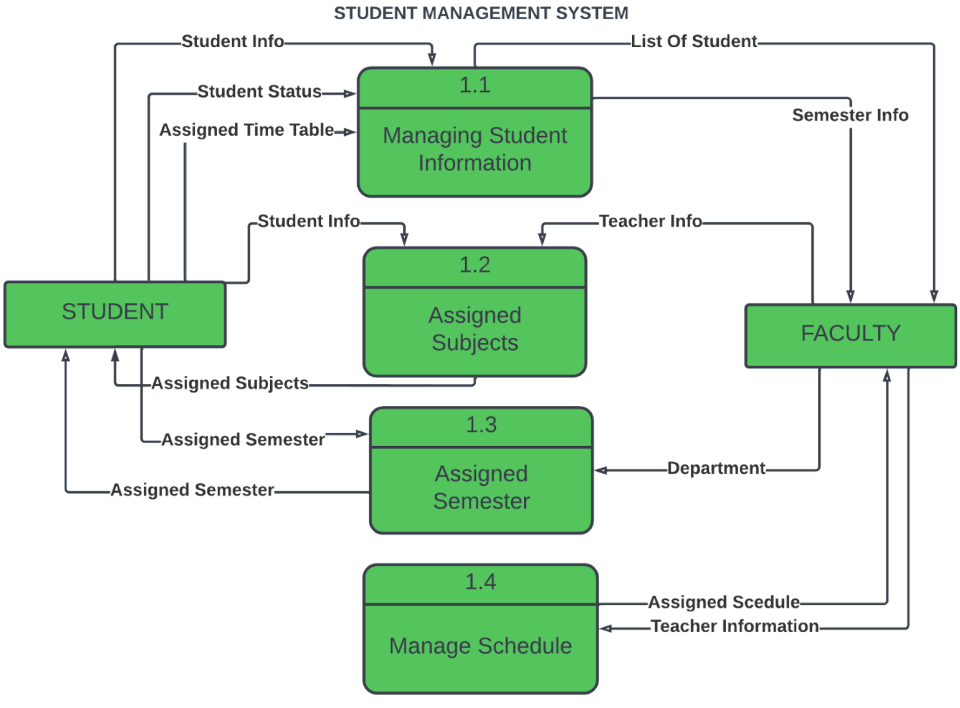
A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled.

LEVEL 0:



DATA FLOW DIAGRAM LEVEL 0

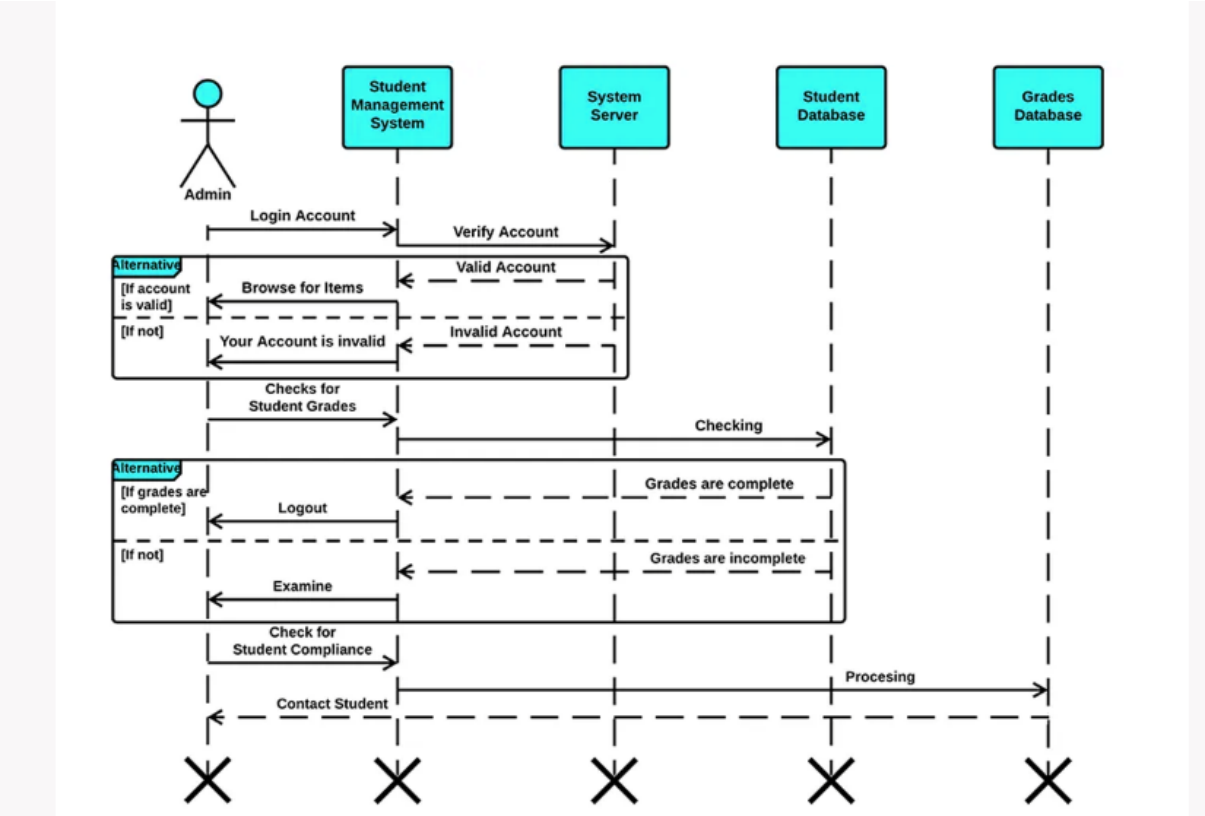
LEVEL 1:



DATA FLOW DIAGRAM LEVEL 1

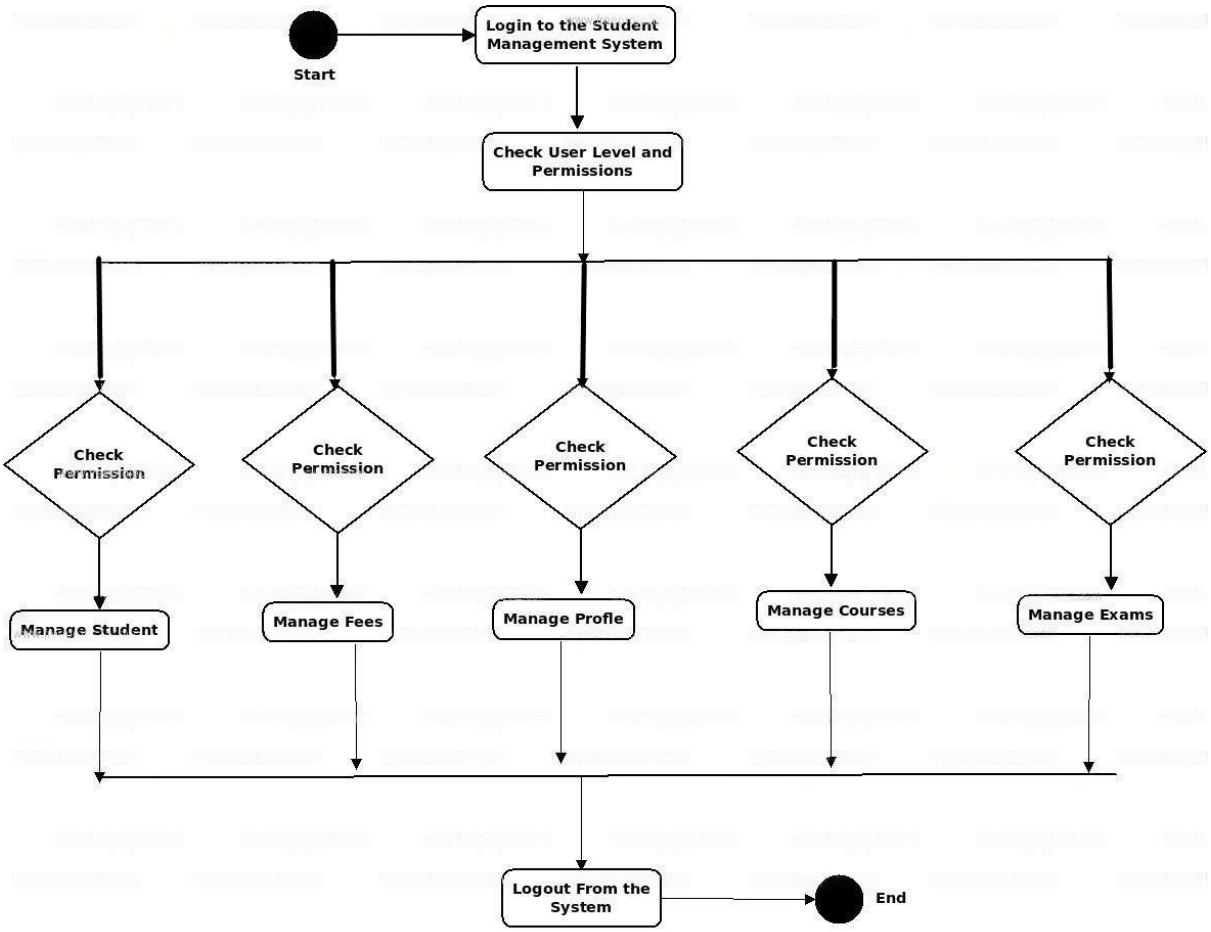
CHAPTER 9
SEQUENCE & COLLABORATION DIAGRAM

Sequence Diagram :



The arrangements of the process when dealing with Student management system is defined by sequence diagram.

Collaboration Diagram :



- A Collaboration is a collection of named objects and actors with links connecting them. They collaborate in performing some task.
- Unlike a sequence diagram, a collaboration diagram shows the relationships among the objects.
- Sequence diagrams and collaboration diagrams express similar information, but show it in different ways.

TESTING

This section contains the test plans and test suites carried out on some of the features in the application. The features tested were the login page, home page, donation page and contact us page. There are many types of test to carry out on a web application such as

- Performance
- Functionality
- Database loading time
- Response time
- Server time handling

Browser check

(Firefox , Microsoft edge and Google chrome)

- Load the website on above browsers
- Registration on the website using the above browsers and registration confirmation by email

Functionality check
The topmost priority for the success of a website is that its functionality across web pages, forms, database connection, call-to-action forms should all work seamlessly

Test plans

Sr. No.	Functionality	Description
1	Login	Method to login based on email address and password
2	Login	Method to login based on the mobile number
3	Register Page	Method to register a donor based on the email address, password, full name

CHAPTER-10
DEVELOPMENT OF TESTING FRAMEWORK/USER INTERFACE

Types of testing

1. Unit testing

It focuses on the smallest unit of software design.

2. Integration testing

The objective is to take unit-tested components and build a program structure that has been directed by design. Integration testing is testing in which a group of components is combined to produce output. Top-down and bottom-up.

3. Regression Testing

Every time a new module is added leads to changes in the program. This type of testing makes sure that the whole component works properly even after adding components to the complete program.

4. Alpha Testing

This is a type of validation testing, it is a type of acceptance testing which is done before the product is released to customers.

5. Beta Testing

The beta test is conducted at one or more customer sites by the end-user of the software.

AUTOMATED TESTING TOOLS

Selenium

- Selenium is available as an open-source tool and is a publicly accessible test automation framework.
- The ease of this tool is its flexibility to support languages such as Ruby, Perl, Java, Python, JavaScript, C#, wherein scripts can be written and this tool converts them into selenium testing compatible codes.
- The significant advantage of this tool is that it supports all web browsers such as Chrome, Firefox, Internet Explorer, Opera, and Edge browsers.
- It enables to record and playback for testing web applications and can run multiple scripts across various browsers.

- In addition, this framework is highly useful for developers as they can analyze the code due to its screenshot property.
- The tests are simple and easy to maintain and it is also easy to repair the test suites of the application using this framework.
- Any automated testing is taken up to ensure the overall testing time is saved and equivalent manual effort is saved.
- With the help of the selenium grid, QA personnel can execute multiple tests in parallel, thus reducing the overall test execution time.

NON FUNCTIONAL TESTING

1. Security

The parameter defines how a system is safeguarded against deliberate and sudden attacks from internal and external sources. This is tested via Security Testing.

2. Reliability

The extent to which any software system continuously performs the specified functions without failure. This is tested by Reliability Testing

3. Survivability

The parameter checks that the software system continues to function and recovers itself in case of system failure. This is checked by Recovery Testing

4. Availability

The parameter determines the degree to which user can depend on the system during its operation. This is checked by Stability Testing.

5. Usability

The ease with which the user can learn, operate, prepare inputs and outputs through interaction with a system. This is checked by Usability Testing

6. Scalability

The term refers to the degree in which any software application can expand its processing capacity to meet an increase in demand. This is tested by Scalability Testing

8. Efficiency

The extent to which any software system can handles capacity, quantity and response time.

9. Flexibility

The term refers to the ease with which the application can work in different hardware and software configurations. Like minimum RAM, CPU requirements.

10. Portability

The flexibility of software to transfer from its current hardware or software environment.

11. Reusability

It refers to a portion of the software system that can be converted for use in another application.

Types of Testing, Methodology, Tools

Category	Methodology	Tools Required
Functional Requirements	Unit Testing	Unit test frameworks like jest and pythons unit testing framework
Functional Requirements	Integration Testing	Tools like git and github are required to merge.
Functional Requirements	System Testing	Testing the whole system using automation.
Non Functional Requirements	load Testing	Use web tools in chrome to see if the site the loads.
Non Functional Requirements	Stress Testing	Use selenium to simulate users and see if people can use it.

CHAPTER-11
TEST CASES & REPORTS

What is Test case?

A test case is a document, which has a set of test data, preconditions, expected results and postconditions, developed for a particular test scenario in order to verify compliance against a specific requirement.

Test Case acts as the starting point for the test execution, and after applying a set of input values, the application has a definitive outcome and leaves the system at some end point or also known as execution postcondition.

Test Case

Test cases are basically two types:

1)FUNCTIONAL TEST CASES

2)NON-FUNCTIONAL TEST CASES

Functional Test Cases

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
t1v1	Verify User Registration	Store it in the server	1. User clicks on User Register 2. Enter the Username and Password 3. Click Register button	User should be taken to the next page for entering more user details	It redirects to his login page	Pass / Failure	success
t2v1	Verify User Registration	Don't Accept any other username other than the system provided				pass/Failure	Success

t3m 1	Verify Details	Details store in server	1)open login 2)Click your profile at right upper corner 3)Go to profile	Student or user should able to see his details	It shows his details in that	Pass/Failure	success

Non-Functional Test Cases

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
t3s1	Precision and Synchronization of data	Check the data is stored and correctly coming in the user login	1)Entering marks by the teacher or principal of the student. 2) And cross-verifying the data	User or Student should get his details and marks correctly	pass	pass/Failure	success

Seek Help From Stakeholders To Remove Obstacles:

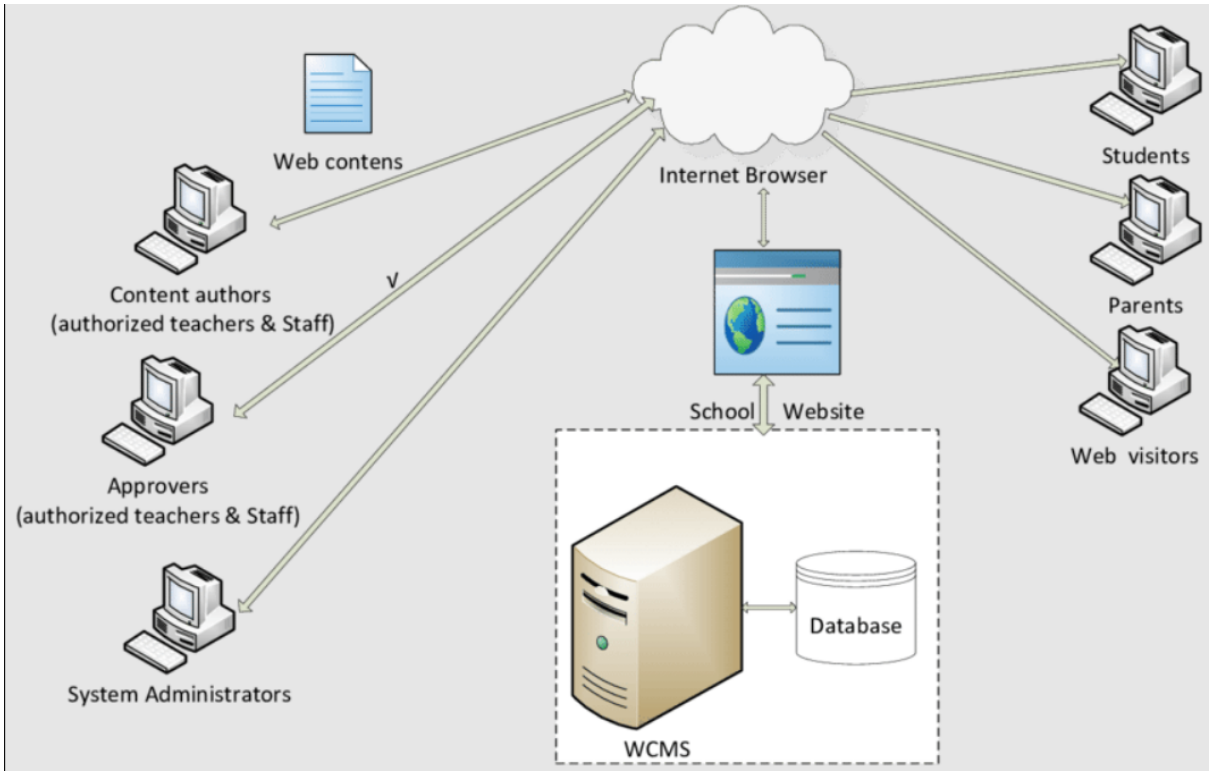
Seek stakeholders to understand the differences of opinions and make them transparent, carefully leading individuals and groups to find common ground. This is more than consensus.

Category	Progress Against Plan	Status
Functional Testing	Green	Completed
Non-Functional Testing	Green	Completed

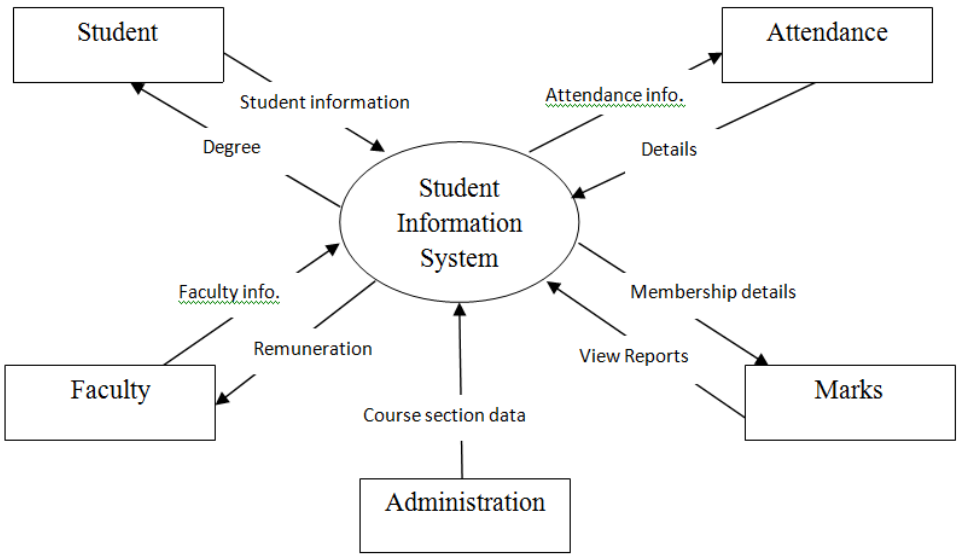
Functional	Test Case Coverage (%)	Status
Logins	50%	Completed
Details	25%	Completed
Synchronization	25%	Completed

CHAPTER 12
ARCHITECTURE/DESIGN/FRAMEWORK/IMPLEMENTATION

SYSTEM ARCHITECTURE :

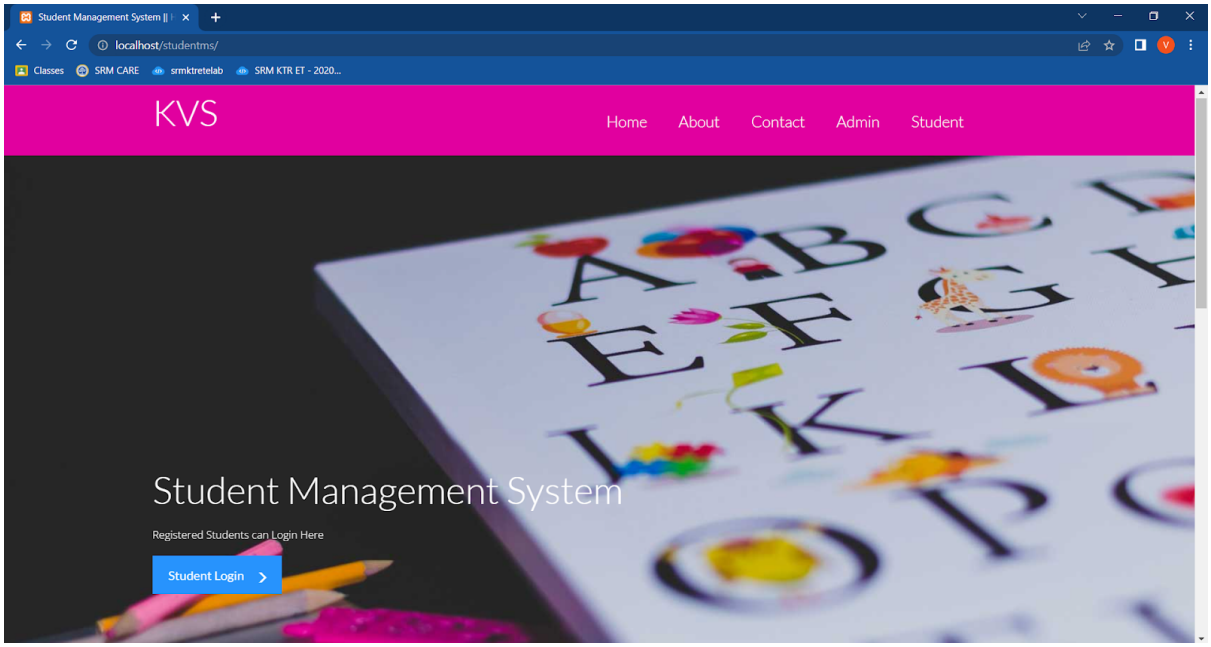


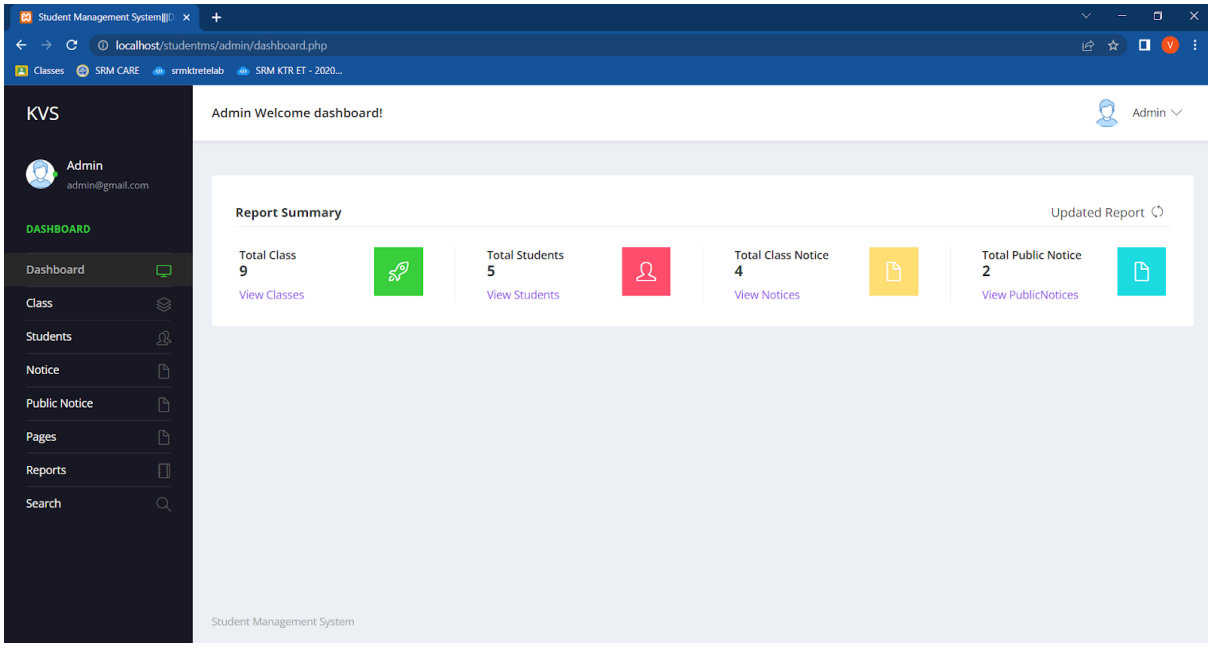
FRAMEWORK :

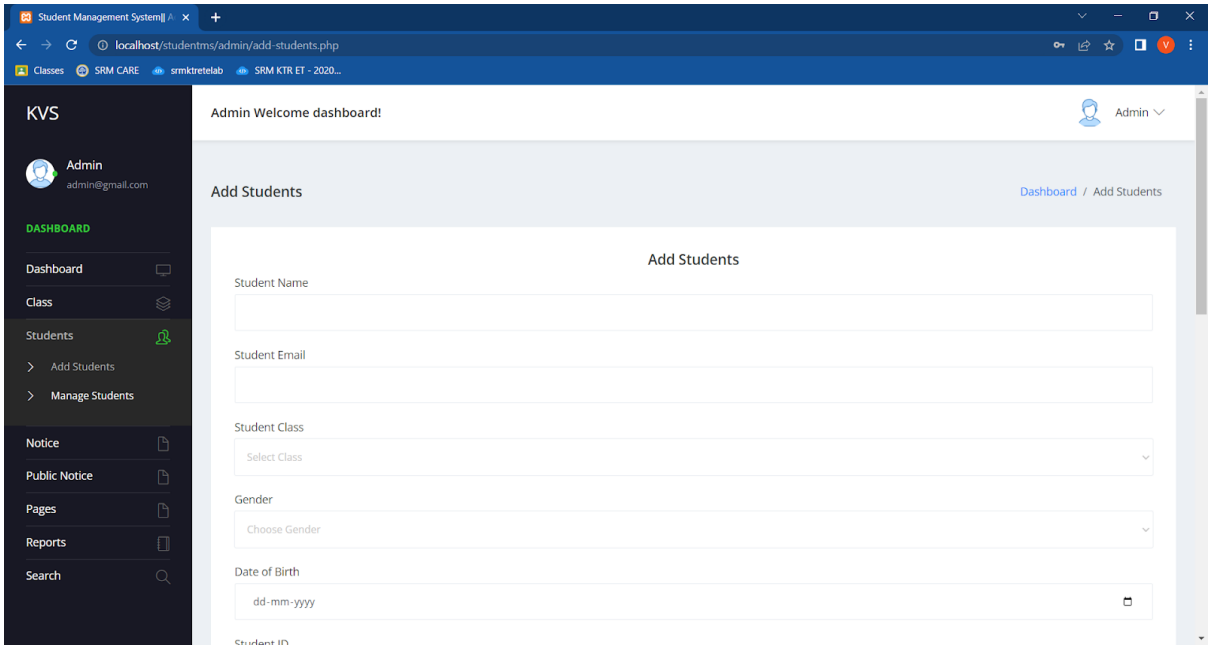


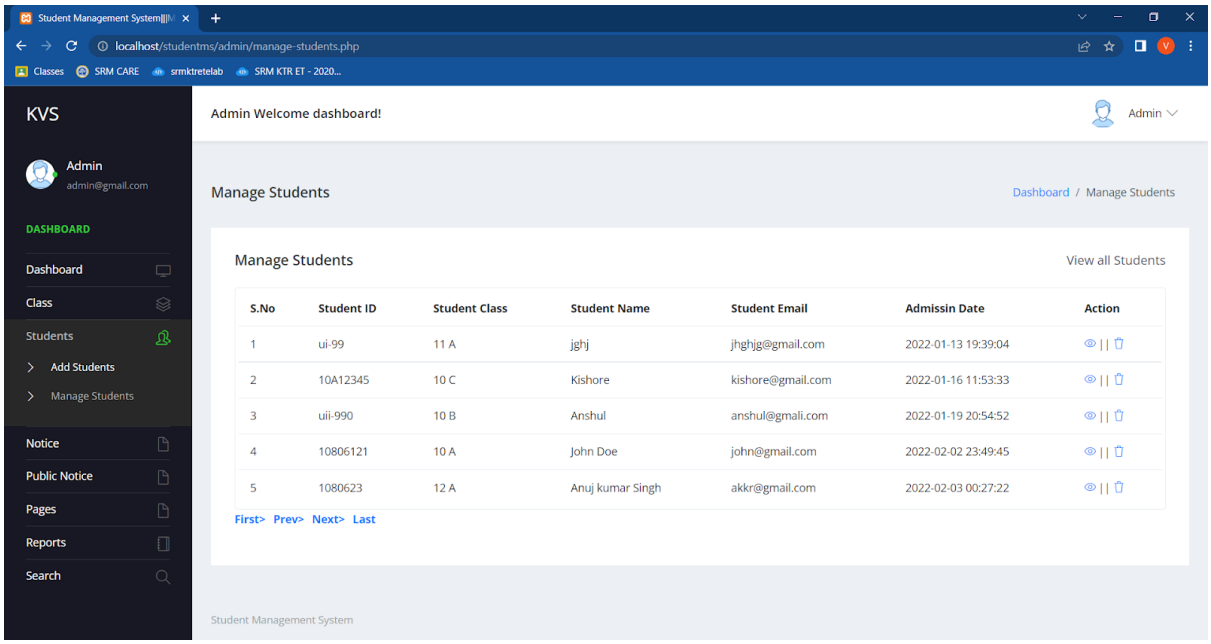
IMPLEMENTATION :

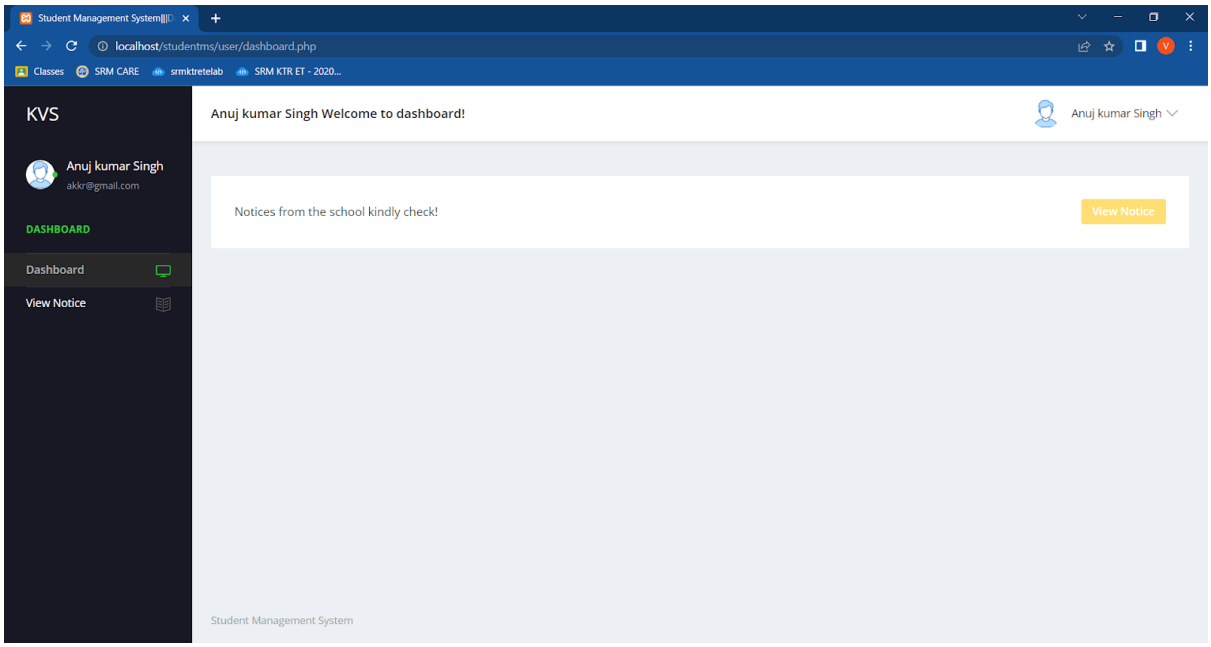
SCREENSHOTS OF WEB PAGES :






Student Management System
localhost/studentms/admin/add-students.php
Classes
SRM CARE
srmktretelab
SRM KTR ET - 2020...
KVS
Admin Welcome dashboard!
Admin
Admin
admin@gmail.com
DASHBOARD
Dashboard
Class
Students
Add Students
Manage Students
Notice
Public Notice
Pages
Reports
Search
Add Students
Dashboard / Add Students
Add Students
Student Name
Student Email
Student Class
Select Class
Gender
Choose Gender
Date of Birth
dd-mm-yyyy


Student Management System
localhost/studentms/admin/manage-students.php
Classes
SRM CARE
srmktretelab
SRM KTR ET - 2020...
KVS
Admin Welcome dashboard!
Admin
Admin
admin@gmail.com
DASHBOARD
Dashboard
Class
Students
Add Students
Manage Students
Notice
Public Notice
Pages
Reports
Search
Manage Students
Dashboard / Manage Students
Manage Students
View all Students
S.No
Student ID
Student Class
Student Name
Student Email
Admissin Date
Action
1 ui-99 11 A jghj jhghjg@gmail.com 2022-01-13 19:39:04
2 10A12345 10 C Kishore kishore@gmail.com 2022-01-16 11:53:33
3 uii-990 10 B Anshul anshul@gmali.com 2022-01-19 20:54:52
4 10806121 10 A John Doe john@gmail.com 2022-02-02 23:49:45
5 1080623 12 A Anuj kumar Singh akkr@gmail.com 2022-02-03 00:27:22
First> Prev> Next> Last
Student Management System


Student Management System
localhost/studentms/user/dashboard.php
Classes
SRM CARE
srmktretelab
SRM KTR ET - 2020...
KVS
Anuj kumar Singh
akkr@gmail.com
DASHBOARD
Dashboard
View Notice
Anuj kumar Singh Welcome to dashboard!
Anuj kumar Singh
Notices from the school kindly check!
View Notice
Student Management System

CONCLUSION

- This project is developed to provide easier way to stode and maneg student database.
- Although the student database management module is not fully integrated to the system and used on real time, the system prototype demonstrates easy navigation and data are stored in a systematic way.
- Overall, efficiency has improved and work processes simplified.
- Although all the objectives have been met, the system still has room for improvement.
- The system is robust and flexible enough for future upgrade using advanced technology and devices.

REFERENCES

1. https://www.slideshare.net/GyanraoPhysics/student-database-management-system-project

2. https://projectabstracts.com/6933/student-database-management-system.html

Textbooks :

1. Steven Holzner, “HTML Black Book: The Programmer’s Complete HTML Reference Book”

2. Robin Nixon, “Learning PHP, MySQL, and JavaScript”, 4 th edition

APPENDIX (CODE)

CODE :
STUDENT DATABASE MANAGEMENT SYSTEM :

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS
*/;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `sturecdb`
--

-- --------------------------------------------------------

--
-- Table structure for table `tbladmin`
--

CREATE TABLE `tbladmin` (
  `ID` int(10) NOT NULL,
  `AdminName` varchar(120) DEFAULT NULL,
  `UserName` varchar(120) DEFAULT NULL,
  `MobileNumber` bigint(10) DEFAULT NULL,
```

```
  `Email` varchar(200) DEFAULT NULL,
  `Password` varchar(200) DEFAULT NULL,
  `AdminRegdate` timestamp NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `tbladmin`
--

INSERT INTO `tbladmin` (`ID`, `AdminName`, `UserName`, `MobileNumber`, `Email`,
`Password`, `AdminRegdate`) VALUES
(1, 'Admin', 'admin', 8979555558, 'admin@gmail.com', 'f925916e2754e5e03f75dd58a5733251',
'2019-10-11 04:36:52');

-- --------------------------------------------------------

--
-- Table structure for table `tblclass`
--

CREATE TABLE `tblclass` (
  `ID` int(5) NOT NULL,
  `ClassName` varchar(50) DEFAULT NULL,
  `Section` varchar(20) DEFAULT NULL,
  `CreationDate` timestamp NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `tblclass`
--
```

```
INSERT INTO `tblclass` (`ID`, `ClassName`, `Section`, `CreationDate`) VALUES
(1, '10', 'A', '2022-01-13 10:42:14'),
(2, '10', 'B', '2022-01-13 10:42:35'),
(3, '10', 'C', '2022-01-13 10:42:41'),
(4, '11', 'A', '2022-01-13 10:42:47'),
(5, '11', 'B', '2022-01-13 10:42:52'),
(6, '11', 'C', '2022-01-13 10:42:57'),
(7, '11', 'D', '2022-01-13 10:43:04'),
(8, '12', 'A', '2022-01-13 10:43:10'),
(9, '12', 'C', '2022-01-13 10:43:15');

-- --------------------------------------------------------

--
-- Table structure for table `tblnotice`
--

CREATE TABLE `tblnotice` (
  `ID` int(5) NOT NULL,
  `NoticeTitle` mediumtext DEFAULT NULL,
  `ClassId` int(10) DEFAULT NULL,
  `NoticeMsg` mediumtext DEFAULT NULL,
  `CreationDate` timestamp NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `tblnotice`
--

INSERT INTO `tblnotice` (`ID`, `NoticeTitle`, `ClassId`, `NoticeMsg`, `CreationDate`)
VALUES
```

```
(2, 'Marks of Unit Test.', 3, 'Meet your class teacher for seeing copies of unit test', '2022-01-19 06:35:58'),
(3, 'Marks of Unit Test.', 2, 'Meet your class teacher for seeing copies of unit test', '2022-01-19 06:35:58'),
(4, 'Test', 3, 'This is for testing.', '2022-02-02 18:17:03'),
(5, 'Test Notice', 8, 'This is for Testing.', '2022-02-02 19:03:43');

-- --------------------------------------------------------

--
-- Table structure for table `tblpage`
--

CREATE TABLE `tblpage` (
  `ID` int(10) NOT NULL,
  `PageType` varchar(200) DEFAULT NULL,
  `PageTitle` mediumtext DEFAULT NULL,
  `PageDescription` mediumtext DEFAULT NULL,
  `Email` varchar(200) DEFAULT NULL,
  `MobileNumber` bigint(10) DEFAULT NULL,
  `UpdationDate` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `tblpage`
--

INSERT INTO `tblpage` (`ID`, `PageType`, `PageTitle`, `PageDescription`, `Email`, `MobileNumber`, `UpdationDate`) VALUES
(1, 'aboutus', 'About Us', '<div style=\"text-align: start;\"><font color=\"#7b8898\" face=\"Mercury SSm A, Mercury SSm B, Georgia, Times, Times New Roman, Microsoft YaHei
```

New, Microsoft Yahei, ????, ??, SimSun, STXihei, ????, serif\"><span style=\"font-size: 26px;\">Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
 ', NULL, NULL, NULL),
(2, 'contactus', 'Contact Us', '890,Sector 62, Gyan Sarovar, GAIL Noida(Delhi/NCR)', 'infodata@gmail.com', 7896541236, NULL);

-- --

--
-- Table structure for table `tblpublicnotice`
--

CREATE TABLE `tblpublicnotice` (
  `ID` int(5) NOT NULL,
  `NoticeTitle` varchar(200) DEFAULT NULL,
  `NoticeMessage` mediumtext DEFAULT NULL,
  `CreationDate` timestamp NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `tblpublicnotice`
--

INSERT INTO `tblpublicnotice` (`ID`, `NoticeTitle`, `NoticeMessage`, `CreationDate`) VALUES
(1, 'School will re-open', 'Consult your class teacher.', '2022-01-20 09:11:57'),
(2, 'Test Public Notice', 'This is for Testing\r\n', '2022-02-02 19:04:10');

```
-- --------------------------------------------------------

--
-- Table structure for table `tblstudent`
--

CREATE TABLE `tblstudent` (
  `ID` int(10) NOT NULL,
  `StudentName` varchar(200) DEFAULT NULL,
  `StudentEmail` varchar(200) DEFAULT NULL,
  `StudentClass` varchar(100) DEFAULT NULL,
  `Gender` varchar(50) DEFAULT NULL,
  `DOB` date DEFAULT NULL,
  `StuID` varchar(200) DEFAULT NULL,
  `FatherName` mediumtext DEFAULT NULL,
  `MotherName` mediumtext DEFAULT NULL,
  `ContactNumber` bigint(10) DEFAULT NULL,
  `AltenateNumber` bigint(10) DEFAULT NULL,
  `Address` mediumtext DEFAULT NULL,
  `UserName` varchar(200) DEFAULT NULL,
  `Password` varchar(200) DEFAULT NULL,
  `Image` varchar(200) DEFAULT NULL,
  `DateofAdmission` timestamp NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `tblstudent`
--
```

```
INSERT INTO `tblstudent` (`ID`, `StudentName`, `StudentEmail`, `StudentClass`, `Gender`, `DOB`, `StuID`, `FatherName`, `MotherName`, `ContactNumber`, `AltenateNumber`, `Address`, `UserName`, `Password`, `Image`, `DateofAdmission`) VALUES
(1, 'jghj', 'jhghjg@gmail.com', '4', 'Male', '2022-01-13', 'ui-99', 'bbmnb', 'mnbmb', 5465454645, 4646546565, 'J-908, Hariram Nagra New Delhi', 'kjhkjh', '202cb962ac59075b964b07152d234b70', 'ebcd036a0db50db993ae98ce380f64191642082944.png', '2022-01-13 14:09:04'),
(2, 'Kishore Sharma', 'kishore@gmail.com', '3', 'Male', '2019-01-05', '10A12345', 'Janak Sharma', 'Indra Devi', 7879879879, 7987979879, 'kjhkhjkhdkshfiludzshfiu\r\nfjedh\r\nk;jk', 'kishore2019', '202cb962ac59075b964b07152d234b70', '5bede9f47102611b4df6241c718af7fc1642314213.jpg', '2022-01-16 06:23:33'),
(3, 'Anshul', 'anshul@gmali.com', '2', 'Female', '1986-01-05', 'uii-990', 'Kailesg', 'jakinnm', 4646546546, 6546598798, 'jlkjkljoiujiouoil', 'anshul1986', '202cb962ac59075b964b07152d234b70', '4f0691cfe48c8f74fe413c7b92391ff41642605892.jpg', '2022-01-19 15:24:52'),
(4, 'John Doe', 'john@gmail.com', '1', 'Female', '2002-02-10', '10806121', 'Anuj Kumar', 'Garima Singh', 1234698741, 1234567890, 'New Delhi', 'john12', 'f925916e2754e5e03f75dd58a5733251', 'ebcd036a0db50db993ae98ce380f64191643825985.png', '2022-02-02 18:19:45'),
(5, 'Anuj kumar Singh', 'akkr@gmail.com', '8', 'Male', '2001-05-30', '1080623', 'Bijendra Singh', 'Kamlesh Devi', 1472589630, 1236987450, 'New Delhi', 'anujk3', 'f925916e2754e5e03f75dd58a5733251', '2f413c4becfa2db4bc4fc2ccead84f651643828242.png', '2022-02-02 18:57:22');

--
-- Indexes for dumped tables
--


--
-- Indexes for table `tbladmin`
--
ALTER TABLE `tbladmin`
```

```
  ADD PRIMARY KEY (`ID`);

--
-- Indexes for table `tblclass`
--
ALTER TABLE `tblclass`
  ADD PRIMARY KEY (`ID`);
--
-- Indexes for table `tblnotice`
--
ALTER TABLE `tblnotice`
  ADD PRIMARY KEY (`ID`);

--
-- Indexes for table `tblpage`
--
ALTER TABLE `tblpage`
  ADD PRIMARY KEY (`ID`);

--
-- Indexes for table `tblpublicnotice`
--
ALTER TABLE `tblpublicnotice`
  ADD PRIMARY KEY (`ID`);

--
-- Indexes for table `tblstudent`
--
ALTER TABLE `tblstudent`
  ADD PRIMARY KEY (`ID`);
```

```
--
-- AUTO_INCREMENT for dumped tables
--
--
-- AUTO_INCREMENT for table `tbladmin`
--
ALTER TABLE `tbladmin`
  MODIFY `ID` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

--
-- AUTO_INCREMENT for table `tblclass`
--
ALTER TABLE `tblclass`
  MODIFY `ID` int(5) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=12;

--
-- AUTO_INCREMENT for table `tblnotice`
--
ALTER TABLE `tblnotice`
  MODIFY `ID` int(5) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

--
-- AUTO_INCREMENT for table `tblpage`
--
ALTER TABLE `tblpage`
  MODIFY `ID` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;
-- AUTO_INCREMENT for table `tblpublicnotice`
--
ALTER TABLE `tblpublicnotice`
  MODIFY `ID` int(5) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;
-- AUTO_INCREMENT for table `tblstudent`
```

```
--
ALTER TABLE `tblstudent`
  MODIFY `ID` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;
COMMIT;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```