

**LIST OF FIGURES**

<b>Figure No.</b>	<b>Figure Name</b>	<b>Page No.</b>
4.1	ER Diagram	8
4.2	Schema Diagram	9
5.1	Account table description	10
5.2	Student table description	10
5.3	Teacher table description	11
5.4	Attendance_student table description	11
5.5	Attendance_teacher table description	12
5.6	Subject table description	12
5.7	Marks table description	13
5.8	Fee table description	13
9.1	Login Page	24
9.2	Signup Page	24
9.3	Home Page	25
9.4	Student Page	25
9.5	Teacher Page	26
9.6	Subject Page	26
9.7	Fee Page	27

**LIST OF TABLES**

<b>TABLE NO.</b>	<b>DESCRIPTION</b>	<b>PAGE NO.</b>
1.1	Account Table	14
1.2	Student Table	14
1.3	Teacher Table	15
1.4	Attendance_Student Table	15
1.5	Attendance_Teacher Table	16
1.6	Subject Table	16
1.7	Marks Table	17
1.8	Fee Table	17
1.9	Test Cases	23

## CHAPTER 1

## INTRODUCTION

### 1.1 Overview:

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

Fee Department

Examination Department

Attendance

Faculty information portal

Student information portal

### 1.2 Purpose:

Drive operational efficiency.

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

Elimination of duplicate data entry processes.

Integrated with Online Application workflow with unified data model.

Monitoring and decision support system.

Automation of all the Academic / Examination / Administration operations.

Ease and accuracy of reporting.

### **1.3 Scope:**

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

## CHAPTER 2

### REQUIREMENT SPECIFICATIONS

#### 2.1 Hardware Requirements :

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

#### 2.2 Software Requirements :

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

## CHAPTER 3

### TOOL DESCRIPTION

#### 3.1 Overview of Front End

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

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

1. It must have a graphical user interface that assists employees that are not from IT background.
2. Scalability and extensibility.
3. Flexibility.
4. Robustness.
5. According to the organization requirement and the culture.
6. Must provide excellent reporting features with good printing support.
7. Platform independent.
8. Easy to debug and maintain.
9. Event driven programming facility.
10. Front end must support some popular back end like MySQL.

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

##### 3.1.1 About Java:

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

Here are some important Java applications:

- It is used for developing Android Apps
- Helps you to create Enterprise Software
- Wide range of Mobile java Applications
- Scientific Computing Applications
- Use for Big Data Analytics
- Java Programming of Hardware devices
- Used for Server-Side Technologies like Apache, JBoss, GlassFish, etc.

### 3.2 Overview of Back End

Back End Selection:

1. Multiple user support.
2. Efficient data handling.
3. Provide inherent features for security.
4. Efficient data retrieval and maintenance.
5. Stored procedures.
6. Popularity.
7. Operating System compatible.
8. Easy to install.
9. Various drivers must be available.
10. Easy to implant with the Front-end.

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

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

#### 3.2.1 About SQL:

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

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

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons.[1]

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

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

Also, they are using different dialects, such as –

- Oracle using PL/SQL,.
- SQL is widely popular because it offers the following advantages –
- Allows users to access data in the database management systems.
- Allows users to describe the data.relational
- Allows users to define the data in a database and manipulate that data.
- Allows to embed within other languages using SQL modules, libraries & pre-compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database.
- Allows users to set permissions on tables, procedures and views.

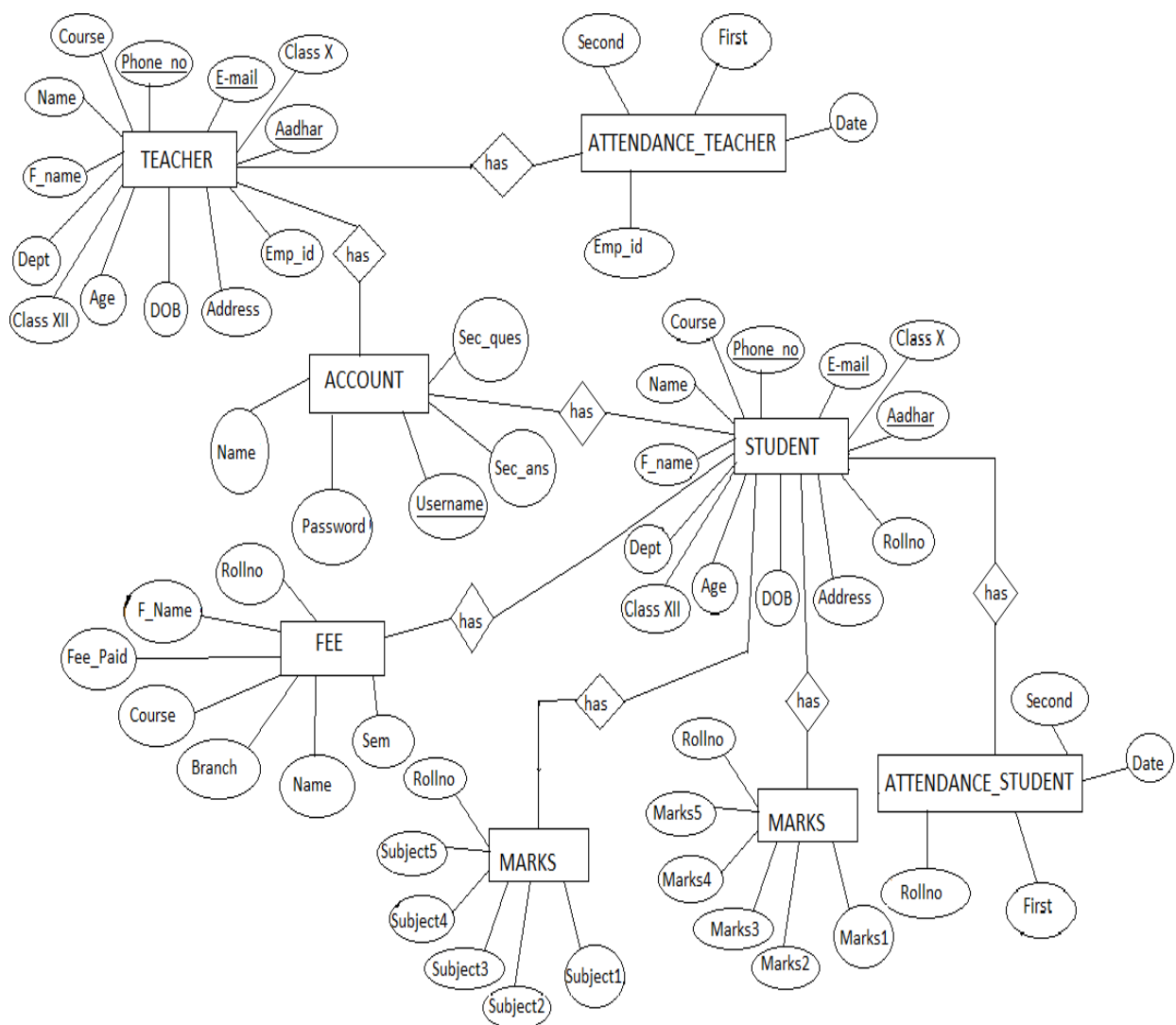


## CHAPTER 4

### REQUIREMENT ANALYSIS

#### 4.1 E-R DIAGRAM:

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



**Figure 4.1: ER Diagram for Smartphone Management Arena**

## 4.2 SCHEMA DIAGRAM:

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

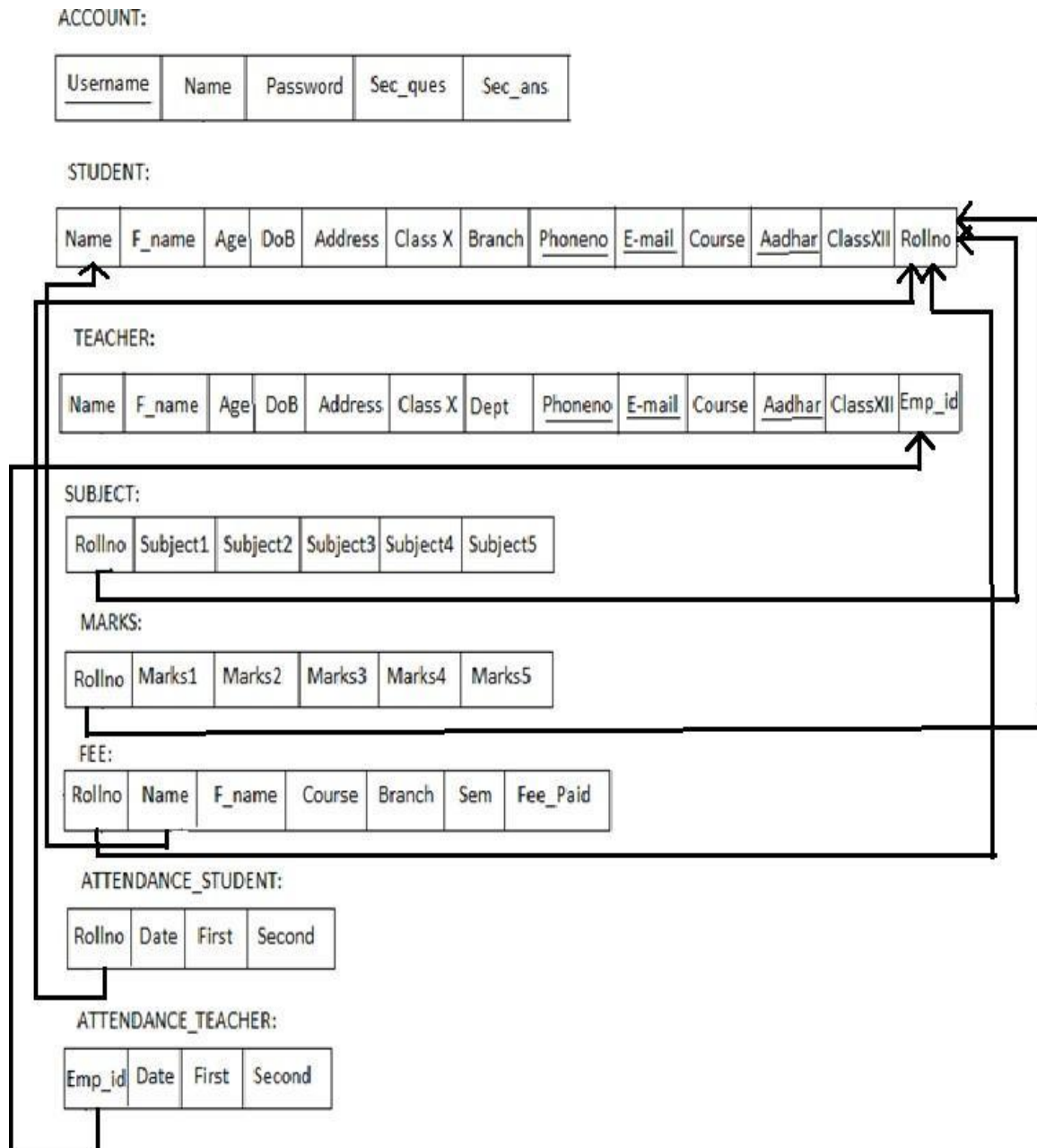


Figure 4.2: Schema Diagram for Smartphone Management System

## CHAPTER 5

### TABLE DESCRIPTION

#### 5.1 Database Design

##### ACCOUNT TABLE

**Account Table:** Account table consists of five attributes which are Username, Name, Password, Sec\_ques, Sec\_ans. Username is used as Primary key.

Desc account;

```
mysql> desc account;
```

Field	Type	Null	Key	Default	Extra
username	varchar(30)	NO	PRI	NULL	
name	varchar(40)	YES		NULL	
password	varchar(30)	YES		NULL	
sec_ques	varchar(100)	YES		NULL	
sec_ans	varchar(50)	YES		NULL	

5 rows in set (0.00 sec)

Fig 5.1 Account table description

##### STUDENT TABLE

**Student table :** Student table is used to add the details of new student like Name, phoneno., DoB, course, Branch etc... Phoneno. , E-mail and Aadhar are used as Primary key.

Desc student;

```
mysql> desc student;
```

Field	Type	Null	Key	Default	Extra
name	varchar(20)	YES		NULL	
fathers_name	varchar(20)	YES		NULL	
age	varchar(5)	YES		NULL	
dob	varchar(20)	YES		NULL	
address	varchar(30)	YES		NULL	
phone	varchar(15)	NO	PRI	NULL	
email	varchar(25)	NO	PRI	NULL	
class_x	varchar(10)	YES		NULL	
class_xii	varchar(10)	YES		NULL	
aadhar	varchar(15)	NO	PRI	NULL	
rollno	varchar(15)	YES		NULL	
course	varchar(10)	YES		NULL	
branch	varchar(20)	YES		NULL	

13 rows in set (0.00 sec)

Fig 5.2 Student table description.

## TEACHER TABLE

**Teacher table:** Teacher table is used to add the details of new student like Name, phoneno.,DoB, course,Branch etc...Phoneno. ,E-mail and Aadhar are used as Primary key.

Desc teacher;

```
mysql> desc teacher;
```

Field	Type	Null	Key	Default	Extra
name	varchar(20)	YES		NULL	
fathers_name	varchar(20)	YES		NULL	
age	varchar(5)	YES		NULL	
dob	varchar(20)	YES		NULL	
address	varchar(30)	YES		NULL	
phone	varchar(15)	NO	PRI	NULL	
email	varchar(25)	NO	PRI	NULL	
class_x	varchar(10)	YES		NULL	
class_xii	varchar(10)	YES		NULL	
aadhar	varchar(15)	NO	PRI	NULL	
course	varchar(10)	YES		NULL	
emp_id	varchar(15)	YES		NULL	
dept	varchar(20)	YES		NULL	

13 rows in set (0.00 sec)

Fig 5.3 Teacher table description

## ATTENDANCE\_STUDENT TABLE

**Attendance\_Student Table:** Attendance\_Student table is used to mark the attendance of the student day to day which as attributes like rollno,name,first and second half.

Desc attendance\_student;

```
mysql> desc attendance_student;
```

Field	Type	Null	Key	Default	Extra
rollno	varchar(20)	YES		NULL	
Date	varchar(30)	YES		NULL	
first	varchar(10)	YES		NULL	
second	varchar(10)	YES		NULL	

4 rows in set (0.04 sec)

Fig 5.4 Attendance\_Student table description.

### ATTENDANCE\_TEACHER TABLE

**Attendance\_Teachertable** :Attendance\_Teacher table is used to mark the attendance of the teacher day to day which as attributes like emp\_id,name,first and second half.

Desc attendance\_teacher;

```
mysql> desc attendance_teacher;
```

Field	Type	Null	Key	Default	Extra
emp_id	varchar(20)	YES		NULL	
Date	varchar(30)	YES		NULL	
first	varchar(10)	YES		NULL	
second	varchar(10)	YES		NULL	

4 rows in set (0.00 sec)

Fig 5.5 Attendance\_Teacher table description.

### SUBJECT TABLE

**Subject table** :Subject table is used to add the subjects of the student in that particular sem with the attributes like rollno and five subjects.

Desc Subject;

```
mysql> desc subject;
```

Field	Type	Null	Key	Default	Extra
rollno	varchar(25)	YES		NULL	
subject1	varchar(30)	YES		NULL	
subject2	varchar(30)	YES		NULL	
subject3	varchar(30)	YES		NULL	
subject4	varchar(30)	YES		NULL	
subject5	varchar(30)	YES		NULL	

6 rows in set (0.02 sec)

Fig 5.6 Subject table description.

### MARKS TABLE

**Marks table :** Marks table is used to add the marks of the particular subjects of the student in a particular sem and the attributes used are rollno and five subject marks.

Desc Marks;

```
mysql> desc marks;
```

Field	Type	Null	Key	Default	Extra
rollno	varchar(15)	YES		NULL	
marks1	varchar(20)	YES		NULL	
marks2	varchar(20)	YES		NULL	
marks3	varchar(20)	YES		NULL	
marks4	varchar(20)	YES		NULL	
marks5	varchar(20)	YES		NULL	

6 rows in set (0.03 sec)

Fig 5.7 Marks table description.

### FEE TABLE

**Fee table:** fee table is used to pay the fee dues of the student for that particular sem and the attributes used like rollno, name, fathersname, course, branch, sem and fee\_paid.

Desc Fee;

```
mysql> desc fee;
```

Field	Type	Null	Key	Default	Extra
rollno	varchar(20)	YES		NULL	
name	varchar(25)	YES		NULL	
fathers_name	varchar(25)	YES		NULL	
course	varchar(10)	YES		NULL	
branch	varchar(20)	YES		NULL	
semester	varchar(10)	YES		NULL	
fee_paid	varchar(15)	YES		NULL	

7 rows in set (0.02 sec)

Fig 5.8 Fee table description.

## CHAPTER 6

### TABLE WITH VALUES

#### 6.1 Output design:

**Account table :** Account table consists of five attributes which will be retrived from user when the user signup/logs in.

Select \* from account;

**Table 1.1 Account table**

```
mysql> select * from account;
```

username	name	password	sec_ques	sec_ans
raja	RAJA	12345	Your Lucky Number?	9900
gopi	Gopi	gopi123	Your NickName?	gopi
vikas	VIKAS	sai12	Your child SuperHero?	ntr
mohan	MOHAN	mogan	Your childhood Name ?	mogan
akash	AKASH	67890	Your Lucky Number?	9

5 rows in set (0.00 sec)

**Student table:** Student table is used to add the details of new student like Name,phoneno.,DoB,course,Branch etc...Phoneno.E-mail and Aadhar are used as Primary key.

Select \* from student;

**Table 1.2 Student table**

```
mysql> select * from student;
```

name	fathers_name	age	dob	address	phone	email	class_x	class_xii	aadhar	rollno	course	branch
Vikas	Sai	22	02/03/1998	Bangalore	9869869576	vikasvicky11@gmail.com	84	77	229876589745	15331807	M.Tech	Electronics
Raja	Srinu	21	29/05/1999	Bangalore	9897969984	raja123@gmail.com	88	82	676476486745	15335115	M.Tech	Mechanical
Gopi	Krishna	20	03/10/2000	Kolar	7869687696	gopi11@gmail.com	82	78	885787588758	1533842	B.Tech	Computer Science
Akash	Kumar	20	22/08/2000	Mangalore	7879696896	akash1122@gmail.com	84	81	906895709687	15339828	B.Tech	Civil
Mohan	Mogesh	19	18/02/2001	Bangalore	7869869665	mogan11@gmail.com	82	79	987689786988	15333481	B.Com	Professional Deg

5 rows in set (0.00 sec)

**Teacher table** :Teacher table is used to add the details of new student like Name,phoneno.,DoB,course,Branch etc...Phoneno. ,E-mail and Aadhar are used as Primary key.

Select \* from teacher;

**Table 1.3 Teacher table**

```
mysql> select * from teacher;
```

name	fathers_name	age	dob	address	phone	email	class_x	class_xii	aadhar	course	emp_id	dept
Lakshmi	Venkatesh	45	04/05/1975	Bangalore	7897658656	lakshmi12@gmail.com	83	78	756876487594	Msc	1016569	Computer Science
Prakash	Kumarswamy	54	21/03/1966	Bangalore	9867976976	prakash11@gmail.com	84	81	979477658798	M.Tech	1013079	Mechanical
Naveen.B.M	Bhaskar	38	26/11/1982	Bangalore	8978987687	naveen123@gmail.com	87	77	896596796798	MBA	1012340	Others
Mahesh.G	Ganesh	41	16/09/1979	Mangalore	7897869876	maheshg11@gmail.com	78	68	456736753857	MCA	1014233	Others
Rakesh	Chandrasekhar	36	11/06/1984	Mysore	8876659766	rakesh121@gmail.com	88	87	337659876007	BCom	1012307	Professional Degree

5 rows in set (0.00 sec)

**Attendance\_Student table** :Attendance\_Student table is used to mark the attendance of the student day to day which as attributes like rollno,name,first and second half.

Select \* from attendance\_student;

**Table 1.4 Attendance\_student table**

```
mysql> select * from attendance_student;
```

rollno	Date	first	second
15331807	Thu Jan 14 16:12:03 IST 2021	Present	Present
15335115	Thu Jan 14 16:12:15 IST 2021	Present	Absent
1533842	Thu Jan 14 16:12:27 IST 2021	Absent	Present
15339828	Thu Jan 14 16:12:41 IST 2021	Absent	Absent
15333481	Thu Jan 14 16:13:00 IST 2021	Leave	Leave

5 rows in set (0.00 sec)



**Attendance\_Teacher table :** Attendance\_Teachertable is used to mark the attendance of the teacher day to day which as attributes like emp\_id,name,first and second half.

Select \* from attendance\_teacher;

**Table 1.5 Attendance\_teacher table**

```
mysql> select * from attendance_teacher;
```

emp_id	Date	first	second
1016569	Thu Jan 14 15:45:45 IST 2021	Present	Present
1013079	Thu Jan 14 15:46:00 IST 2021	Absent	Present
1012340	Thu Jan 14 15:46:15 IST 2021	Present	Absent
1014233	Thu Jan 14 15:46:32 IST 2021	Absent	Absent
1012307	Thu Jan 14 15:46:47 IST 2021	Leave	Leave

5 rows in set (0.00 sec)

**Subject table :** Subject table is used to add the subjects of the student in that particular sem with the attributes like rollno and five subjects.

Select \* from Subject;

**Table 1.6 Subject table**

```
mysql> select * from subject;
```

rollno	subject1	subject2	subject3	subject4	subject5
15331807	Devices	Signals	System	Numericals	Circuits
15335115	Mathematics	Statics and Dynamics	Solid mechanics	Material engineering	Composites
1533842	Computer networks	Database management	Python	Unix	ATC
15339828	Building materials	Strength of materials	Structures	Contuction project	Steel design
15333481	Accounts	Economics	Statistics	Management	Finance

5 rows in set (0.00 sec)

**Marks table :** Markstable is used to add the marks of the particular subjects of the student in a particular sem and the attributes used are rollno and five subject marks.

Select \* from Marks;

**Table 1.7 Marks table**

```
mysql> select * from marks;
```

rollno	marks1	marks2	marks3	marks4	marks5
15331807	78	82	79	76	85
15335115	78	83	88	79	80
1533842	77	68	76	68	70
15339828	60	68	65	73	75
15333481	78	72	70	69	74

5 rows in set (0.00 sec)

**Fee table :** fee table is used to pay the fee dues of the student for that particular sem and the attributes used like rollno, name, fathersname, course, branch, sem and fee\_paid.

Select \* from Fee;

**Table 1.8 Fee table**

```
mysql> select * from fee;
```

rollno	name	fathers_name	course	branch	semester	fee_paid
15331807	Vikas	Sai	M.Tech	Electronics	2nd	30000
15335115	Raja	Srinu	M.Tech	Mechanical	1st	40000
1533842	Gopi	Krishna	B.Tech	CSE	5th	51000
15339828	Akash	Kumar	B.Tech	Civil	6th	28000
15333481	Mohan	Mogesh	B.com	Other	3rd	30000

5 rows in set (0.00 sec)

**CHAPTER 7****IMPLEMENTATION****Sample code :**

Package

```
institution.management.system;
```

```
import java.awt.*;
```

```
import javax.swing.*;
```

```
import java.awt.event.*;
```

```
import java.sql.*;
```

```
import institution.management.system.Signup;
```

```
public class Login extends JFrame implements ActionListener{
```

```
    private JPanel panel;
```

```
    private JTextField textField;
```

```
    private JPasswordField passwordField;
```

```
    private JButton b1,b2,b3;
```

```
    public Login() {
```

```
        setBackground(new Color(169, 169, 169));
```

```
        setBounds(600, 300, 600, 400);
```

```
        panel = new JPanel();
```

```
        panel.setBackground(new Color(176, 224, 230));
```

```
        setContentPane(panel);
```

```
        panel.setLayout(null);
```

```
        JLabel l1 = new JLabel("Username : ");
```

```
l1.setBounds(124, 89, 95, 24);
```

```
panel.add(l1);

JLabel l2 = new JLabel("Password : ");

l2.setBounds(124, 124, 95, 24);

panel.add(l2);

textField = new JTextField();

textField.setBounds(210, 93, 157, 20);

panel.add(textField);

passwordField = new JPasswordField();

passwordField.setBounds(210, 128, 157, 20);

panel.add(passwordField);

JLabel l3 = new JLabel("");

l3.setBounds(377, 79, 46, 34);

panel.add(l3);

JLabel l4 = new JLabel("");

l4.setBounds(377, 124, 46, 34);

panel.add(l3);

b1 = new JButton("Login");

b1.addActionListener(this);

b1.setForeground(new Color(46, 139, 87));

b1.setBackground(new Color(250, 250, 210));

b1.setBounds(149, 181, 113, 39);

panel.add(b1);

b2 = new JButton("SignUp");

b2.addActionListener(this);
```

```
b2.setForeground(new Color(139, 69, 19));

b2.setBackground(new Color(255, 235, 205));

b2.setBounds(289, 181, 113, 39);

panel.add(b2);

b3 = new JButton("Forgot Password");

b3.addActionListener(this);

b3.setForeground(new Color(205, 92, 92));

b3.setBackground(new Color(253, 245, 230));

b3.setBounds(199, 231, 179, 39);

panel.add(b3);

JLabel l5 = new JLabel("Trouble in Login?");

l5.setFont(new Font("Tahoma", Font.PLAIN, 15));

l5.setForeground(new Color(255, 0, 0));

l5.setBounds(70, 240, 130, 20);

panel.add(l5);

JPanel panel2 = new JPanel();

panel2.setBackground(new Color(176, 224, 230));

panel2.setBounds(24, 40, 434, 263);

panel.add(panel2);

}

public void actionPerformed(ActionEvent ae){

    if(ae.getSource() == b1){

        Boolean status = false;

        try {
```

```
conn con = new conn();

        String sql = "select * from account where username=? and password=?";

PreparedStatementst = con.c.prepareStatement(sql);

st.setString(1, textField.getText());

st.setString(2, passwordField.getText());

ResultSetrs = st.executeQuery();

if (rs.next()) {

this.setVisible(false);

new Loading().setVisible(true);

        } else

                JOptionPane.showMessageDialog(null, "Invalid Login...!");

        } catch (Exception e2)

{ e2.printStackTrace();} }

if(ae.getSource() == b2){

setVisible(false);

        Signup su = new Signup();

        su.setVisible(true);}

if(ae.getSource() == b3){

setVisible(false);

        ForgotPassword forgot = new ForgotPassword();

        forgot.setVisible(true);}

}

public static void main(String[] args) {

        new Login().setVisible(true);    }    }
```

## **CHAPTER 8**

### **TESTING**

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

#### **8.1 Unit Testing**

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

#### **8.2 Integration Testing**

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

#### **8.3 User Acceptance**

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



**8.4 Test Cases:****Table 1.9 Test cases**

<b>Test No.</b>	<b>Test Name</b>	<b>input</b>	<b>Actual output</b>	<b>Expected output</b>	<b>Status</b>
1	Login	Username and password	User is successfully Authenticated	User is successfully Authenticated	Pass
2	Login	Wrong username and password	Invalid username or password	Invalid username or password	Pass
3	Signup	User details and password	Account successfully created	Account successfully created	Pass
4	Student	Details of the student required	Student inserted successfully	Student inserted successfully	Pass
5	Teacher	Details of the teacher required	Teacher inserted successfully	Teacher inserted successfully	Pass
6	Subject	Enter the subject names and marks along with rollno	Subjects entered successfully	Subjects entered successfully	Pass
7	Fee	Details and fee_paid	Paid successfully	Paid successfully	Pass
8	Remove Student	Enter rollno and click on remove	Removed successfully	Removed successfully	Pass
9	Remove Teacher	Enter emp_id and click on remove	Teacher removed successfully	Teacher removed successfully	Pass
10	Exit	Click on Exit	Logout successfully	Logout successfully	Pass

## CHAPTER 9

### SNAPSHOTS

1. **Login form:** This page represents the first thing about our website. It leads on to the login point for its personnel; it takes up the username,password and signup.

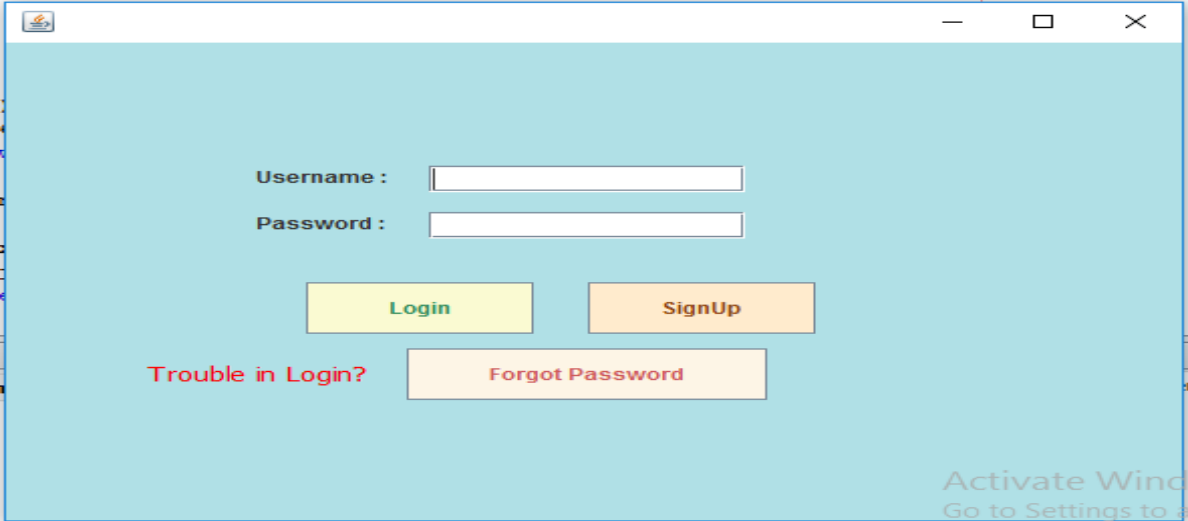
A screenshot of a web browser window displaying a login form. The form is centered on a light blue background. It includes two input fields: 'Username :' and 'Password :'. Below these are two buttons: 'Login' (yellow) and 'SignUp' (orange). To the left of the 'Forgot Password' button is the text 'Trouble in Login?'. The 'Forgot Password' button is orange. In the bottom right corner, there is a faint watermark that says 'Activate Windows Go to Settings to activate Windows'.

Figure 9.1: Login form

2. **Signup page:** This page represents signing up to website. It leads to registering to website making username and password, it takes the up username, name, password and security question. These information are mandatory.

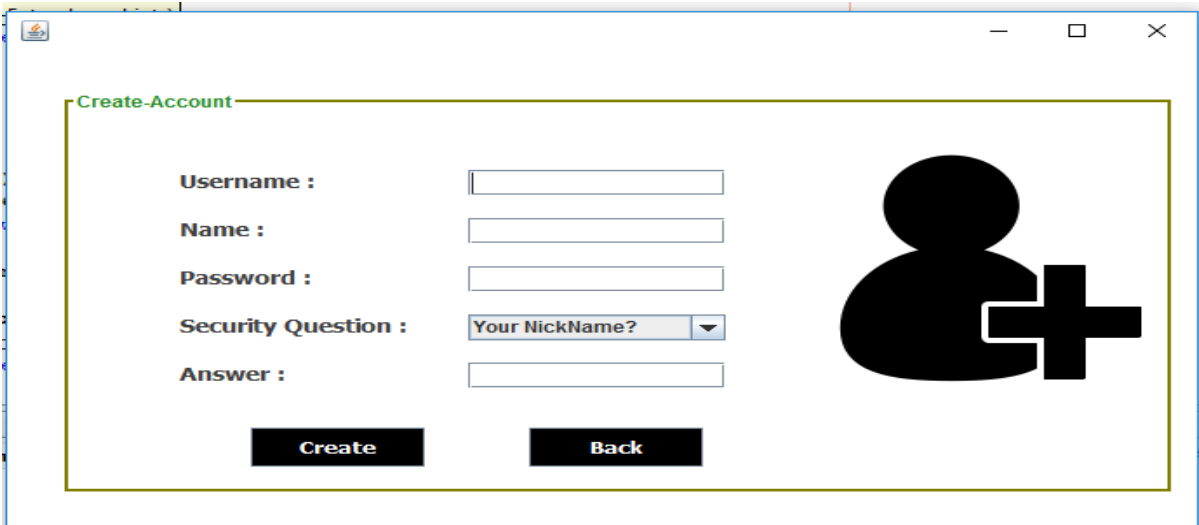
A screenshot of a web browser window displaying a signup page. The page has a white background. A green-bordered box contains the form fields. The fields are: 'Username :', 'Name :', 'Password :', 'Security Question :', and 'Answer :'. The 'Security Question :' field has a dropdown menu with 'Your NickName?' selected. Below the fields are two buttons: 'Create' (black) and 'Back' (black). To the right of the form is a large black icon of a person with a plus sign.

Figure 9.2: Signup page

3. **Home page user :** This page shows us what user can see and access. He can add, remove, update and upload the data. He can logout from the website in homepage.

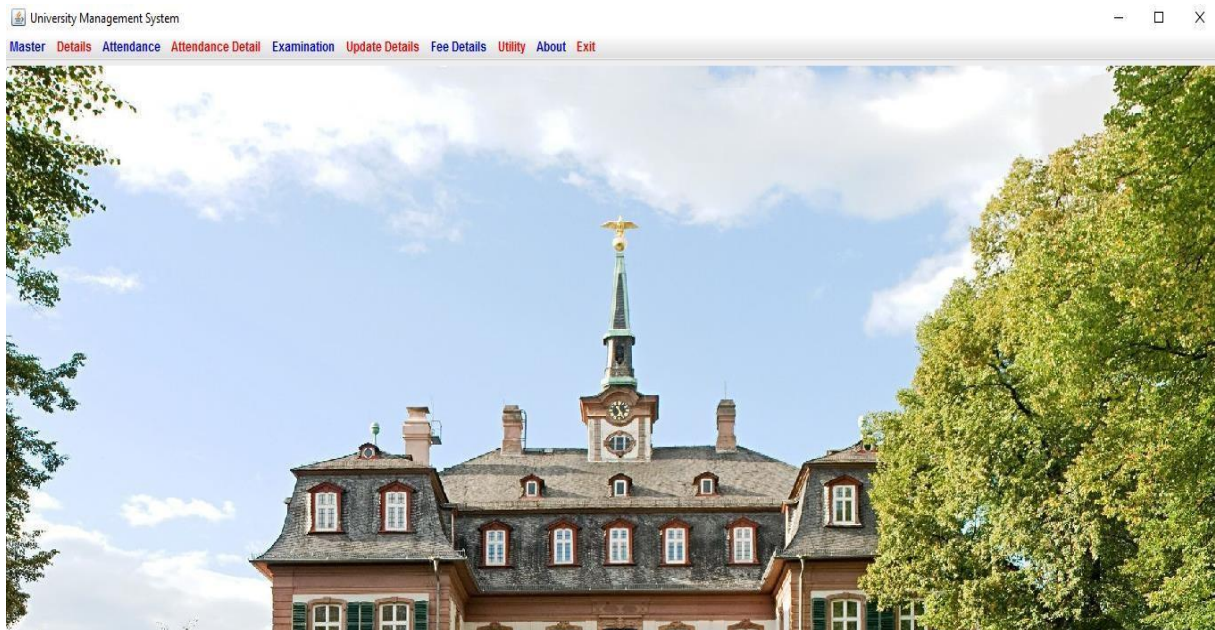


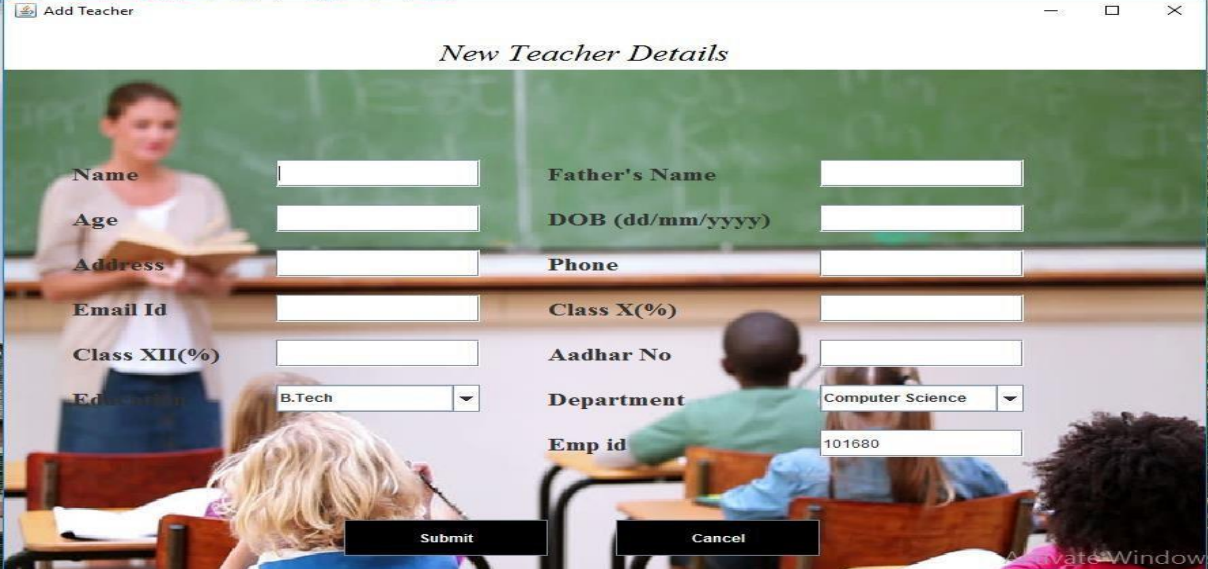
Figure 9.3: Home page user

4. **Student form :** In this we can add the new student details which will be stored in back end of user. This details further can be updated in the update page.

A screenshot of a web browser window titled "Add Student". The form is titled "New Student Details" and is set against a background image of a smiling student. The form contains two columns of input fields. The left column includes fields for Name, Age, Address, Email Id, Class XII(%), Roll No (pre-filled with "15331641"), and Branch (a dropdown menu currently showing "Computer Science"). The right column includes fields for Father's Name, DOB (dd/mm/yyyy), Phone, Class X(%), Aadhar No, and Course (a dropdown menu currently showing "B.Tech"). At the bottom of the form are two black buttons labeled "Submit" and "Cancel".

Figure 9.4: Student form

5. **Teacher form:** In this we can add the new teacher details which will be stored in back end of user. This details further can be updated in the update page.



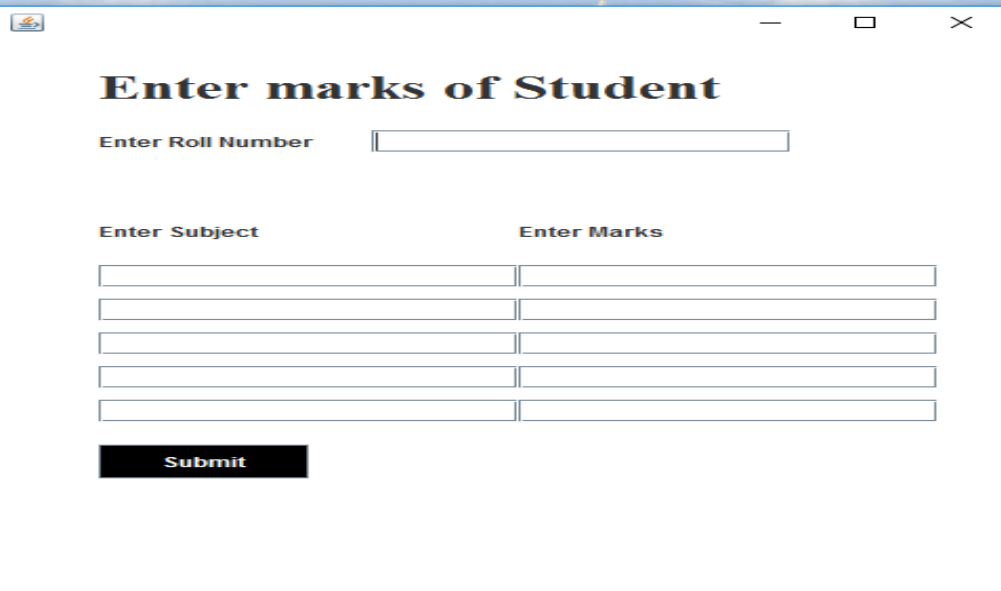
The screenshot shows a web application window titled "Add Teacher" with a subtitle "New Teacher Details". The background image depicts a teacher in a classroom. The form contains the following fields:

Name	<input type="text"/>	Father's Name	<input type="text"/>
Age	<input type="text"/>	DOB (dd/mm/yyyy)	<input type="text"/>
Address	<input type="text"/>	Phone	<input type="text"/>
Email Id	<input type="text"/>	Class X(%)	<input type="text"/>
Class XII(%)	<input type="text"/>	Aadhar No	<input type="text"/>
Education	<input type="text" value="B.Tech"/>	Department	<input type="text" value="Computer Science"/>
		Emp id	<input type="text" value="101680"/>

At the bottom, there are two buttons: "Submit" and "Cancel".

Figure 9.5: Teacher form

6. **Marks and Subject page :** In this page we can enter the subjects and marks scored in that particular subject along the rollno.



The screenshot shows a web application window titled "Enter marks of Student". The form contains the following fields:

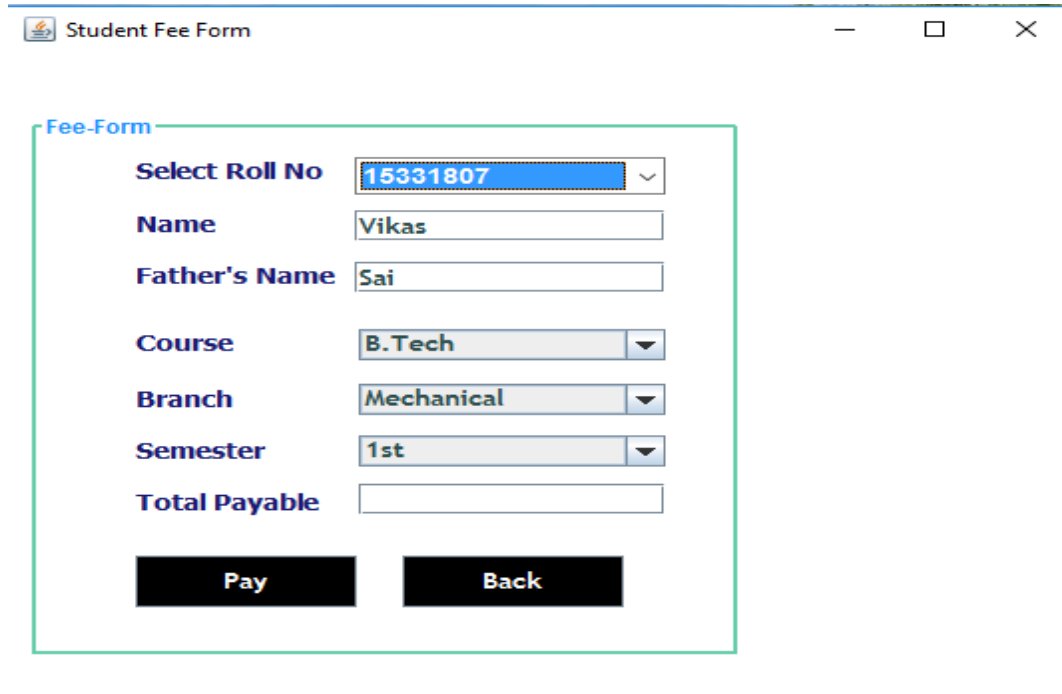
Enter Roll Number

Enter Subject	Enter Marks
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

At the bottom, there is a "Submit" button.

Figure 9.6: Marks and Subject page

7. **Fee payment page :** In this page we can pay the fee dues of the particular student which uses rollno, course, branch and sem to pay the fee.



The screenshot shows a window titled "Student Fee Form" with a standard Windows title bar (minimize, maximize, close buttons). Inside the window, there is a sub-form titled "Fee-Form" with a green border. The "Fee-Form" contains the following fields and controls:

- Select Roll No**: A dropdown menu with "15331807" selected.
- Name**: A text input field containing "Vikas".
- Father's Name**: A text input field containing "Sai".
- Course**: A dropdown menu with "B.Tech" selected.
- Branch**: A dropdown menu with "Mechanical" selected.
- Semester**: A dropdown menu with "1st" selected.
- Total Payable**: An empty text input field.
- Buttons**: Two black buttons labeled "Pay" and "Back" are positioned at the bottom of the form.

Figure 9.7: Fee payment page

## CONCLUSION

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

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

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

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

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

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

## REFERENCES

### Books and Websites:

- Internet & World Wide Web: How to Program Deitel, PJ Deitel.
- Code for Interview YouTube Channel.
- Database System Concepts, by Silberschatz, Sudarshan, and Korth.
- Fundamentals of Database Systems, RamezElmasri and Shamkant B. Navathe, 7th Edition. 2017, Pearson...