VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", BELAGAVI – 590018

A MINI PROJECT REPORT

ON

"STUDENTS NOTES MANAGEMENT SYSTEM"

Submitted in partial fulfillment of requirements for the course
DBMS Laboratory with Mini Project [18CSL58] of Fifth Semester
of Bachelor of Engineering in Computer Science & Engineering
during the academic year 2020-21.

Submitted By

Sreyas P
[4MH18CS107]

Sena Ahammed
[4MH18CS100]

Under the Guidance of

Prof. Santosh E
Assistant Professor,
Dept. of CS&E,

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE
Belawadi, S.R. Patna (T), Mandya (D) – 571477.

2020 - 2021

MAHARAJA INSTITUTE OF TECHNOLOGY MYSORE

Belawadi, S.R. Patna (T), Mandya (D) – 571477.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the mini project work entitled **"STUDENTS NOTES MANAGEMENT SYSTEM"** is a bonafide work carried out by **Sena Ahammed [4MH18CS100]** and **Sreyas P [4MH18CS107]** in partial fulfillment for the **DBMS Laboratory with Mini Project (18CSL58)** prescribed by the Visvesvaraya Technological University, Belagavi during the year 2020-2021 for the fifth semester B.E in Computer Science and Engineering. The mini project report has been approved as it satisfies the academic requirements.

Signature of Guide	Signature of HOD
(Prof.Santosh E)	**(Dr.Shivamurthy R C)**
Assistant Professor, Dept. of CS&E	Professor & Head, Dept. of CS&E
MIT Mysore	MIT Mysore

Name of the Examiners	Signature with date
1.....	….....
2.....	….....

ACKNOWLEDGEMENT

We sincerely owe our gratitude to all the persons who helped and guided us in completing this mini project work.

We are thankful to **Dr. B.G. Naresh Kumar, Principal, Maharaja Institute of Technology Mysore**, for having supported us in our academic endeavors.

We are extremely thankful to **Dr.Shivamurthy R C, Professor & Head, Department of Computer Science and Engineering,** for his valuable support and his timely inquiries into the progress of the work.

We are greatly indebted to our guide **Prof. Santosh E, Assistant Professor, Department of Computer Science and Engineering,** for the consistent co-operation and support.

We are obliged to all **teaching and non-teaching staff members** of **Department of Computer Science and Engineering,** for the valuable information provided by them in their respective field's. We are grateful for their co-operation during the period of our mini project.

Sreyas P (4MH18CS107)

Sena Ahammed (4MH18CS100)

ABSTRACT

Nowadays we students are more dependent on digital notes more than we are on physical notes . The digital notes are also a little difficult to get and even if we get them, we people are not sure if the notes we got is of the right scheme and if it will be sufficient . Sometimes we study the wrong topics or we study the unnecessary explanations , all of these only creating a hundred more problem for us . Our aim is to make it easier for the students to have access on the notes anytime the may want it.

In our project we are creating a Students Notes Management System . Here the students , the teachers and the admin has a separate login option . The students can login anytime they want ,choose their branch and view the notes of their liking . For the teachers we are giving them authority to insert ,update or delete any notes. Finally the admin has the authority to insert , update or delete any teacher or student . We believe this makes it easier for both the students and the teachers , for students it is now easier to access the notes and for the teachers it is now not necessary to circulate the to every WhatsApp group rather all he/she has to do is upload the notes into our database.

Therefore making everyones life easier.

~~~~~ TABLE OF CONTENTS ~~~~~

Chapter 01

INTRODUCTION

1.1 Aim of the Project

As a total manpower based system is currently running for the notes procedures, designing a new system which makes the whole process online, demands a deep knowledge about the existing system. Throughout the project we focus on presenting information and commands in an easy and intelligible manner. The purpose of our DIGITAL NOTES is to provide a leading technological tool for the ease of NOTES functions such as Downloading, Learners efficient notes written by our knowledgeable faculties, and accurate copy of notes etc. It will reduce considerably the difficulties faced on existing system, with minimum error and difficulties

1.2 Overview of the Project

Our project the DIGITAL NOTES provides easy access of notes to the students making their lifes so much easier . Now the notes are always available for them ,anytime anywhere. Our project allows three types of login one for the admin another one for staff/teacher and the last one for the students . The three types of login gives 3 different sets of permissions for the user, each different requires user_id and password. Where the admin can manipulate the Staff , notes and the Student database , while teacher can only manipulate the notes database and finally the students will simply be able to view the notes . The student can select his /hers branch , semester and then get the notes for the subject she needs.

1.3 Outcome of the Project

Our project DIGITAL NOTES now allows teachers to easily circulate the notes among the students . The students now have easier access to all the notes when and where

they want . The manual work of carrying the notes around and keeping them somewhere safe is out of question now. All the knowledge you want is a click away . The admin now can easily decide who can and cannot view the notes.

1.4 Software Requirements

Front End : HTML, CSS, Java Script

Back End : MySQL

Language : php

The system is implemented as 3-tier approach with a backend database handled by the system administrator and a web browser as the front-end client. This document will discuss each of the underlying technologies used to create and implement notes management website. To implement this we have used PHP, which is platform independent and therefore, can be run on all major operating systems. PHP provides support to all major servers like Apache and databases like MySQL. Since it uses its own memory, the loading time is decreased and processing time is increased. Next, we have used HTML, JavaScript, CSS and Bootstrap for front-end implementation. They provide a front-end development framework to create fully responsive web pages and define proper styles and presentation of the document. Lastly, MySQL is used as the back-end database since it is one of the most popular open source databases, and it provides fast data access, easy installation and simplicity.

Chapter 2

DESIGN

2.1 Schema Diagram

We have five different tables department ,faculty, student, subject , notes. The table department stores the details of all the departments. Details such as department_id ,department name and the name of the head of the department . In the table faculty we have faculty_id , faculty name, the department to which the faculty belongs to and also the password they have to use to login . The student table contains details of students such as his/hers USN , name ,their branch(which is understood using the attribute department_id which is referring to the department table) , semester and finally their date of birth to be used as a password for login . We have another table which contains the details all the subject , we also know which branch that subject belongs to using the department_id attribute in the table . Finally we have another table to actually store the notes , this table is linked with subject table using the attribute suject_code from here we get all the subject details of a note.

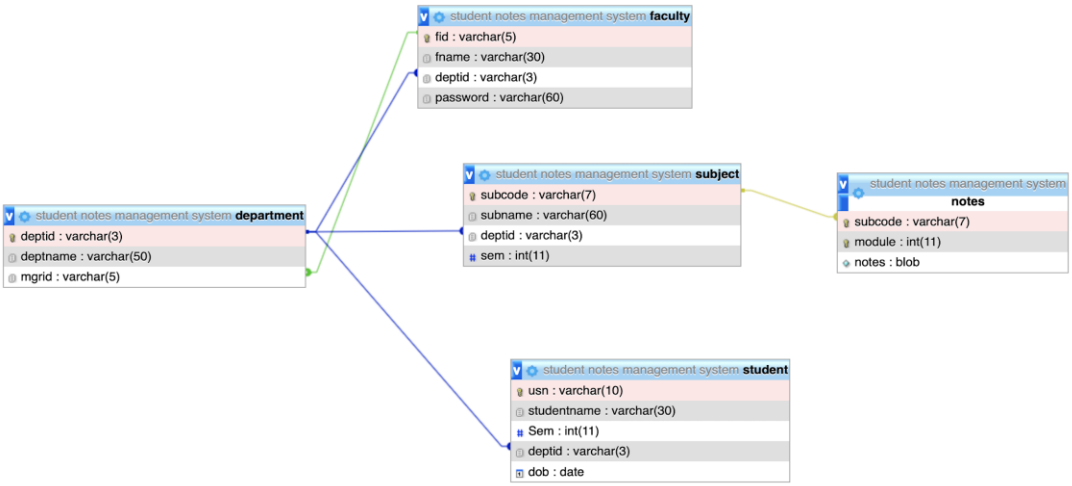


Fig 2.1 :Schema Diagram

2.2 E-R Diagram

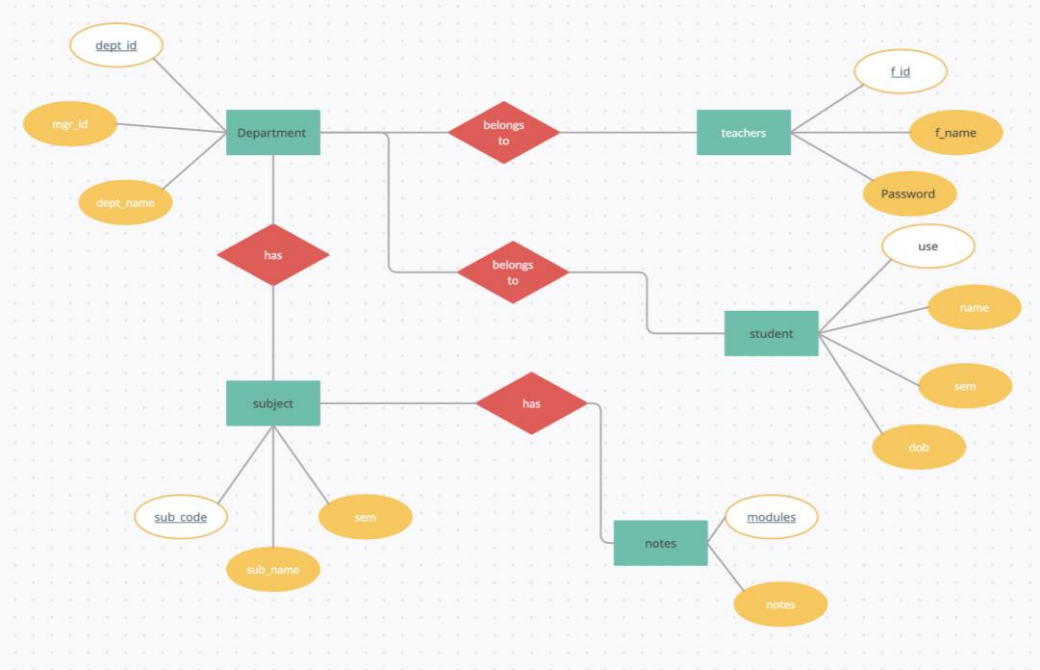


Fig 2.2:E-R Diagram

In our schema diagram we have five tables Department , Teachers , Students , Subject and Notes.The table department ,teachers and student is linked with the foreign key department_id . The table subject and notes is linked with the foreign key subj_code. The department table contains details such as department_id ,department name and the name of the head of the department . In the table faculty we have faculty_id , faculty name, the department to which the faculty belongs to and also the password they have to use to login . The student table contains details of students such as his/hers USN , name ,their branch(which is understood using the attribute department_id which is referring to the department table) , semester and finally their date of birth to be used as a password for login . We have another table which contains the details all the subject , we also know which branch that subject belongs to using the department_id attribute in the table . Finally we have another

table to actually store the notes , this table is linked with subject table using the attribute suject_code from here we get all the subject details of a note.

2.3 Use Case Diagram

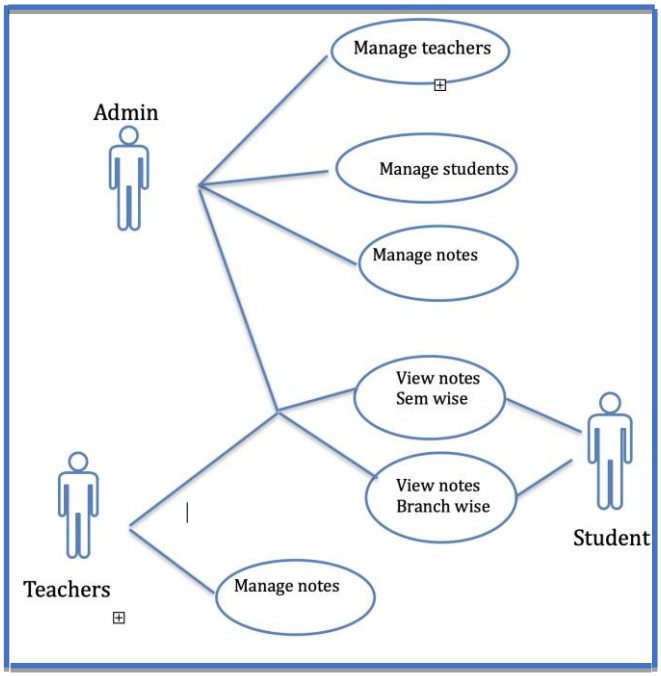


Fig 2.3:Use case diagram

In our project Digital notes we have three types of users , the Admin , the Teacher and the Student. The admin has the ability to add ,remove or change details of any teacher or student and the manager can also view the notes. The teacher on the other hand can only add remove or re-upload notes and the teachers can also view which ever notes she wants to. The final user can access any notes he wants but doest have any other authority.

2.4 Data Flow Diagram

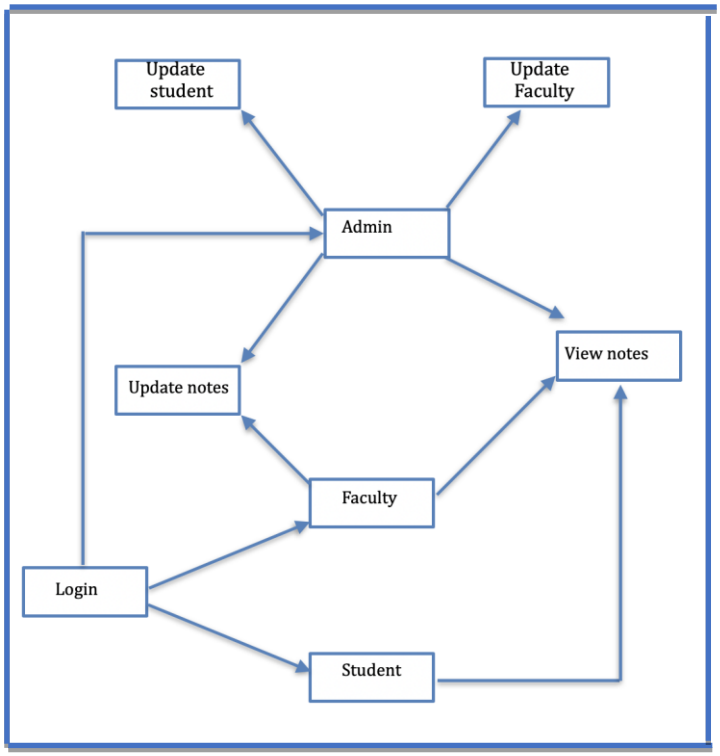


Fig 2.4:Data Flow Diagram

We give three login options in our web based application. One for the admin another for the teacher and then for the student. In our web based application we have given the authority to admin to add , edit or remove any student , teacher or notes. The teacher can add ,edit or remove any notes. Both the teacher and the Admin can view any notes. They student in the other hand can only view the notes .

Chapter 3

IMPLEMENTATION

3.1 Table Description

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	deptid	varchar(3)	utf8mb4_general_ci		No	None			Change Drop More
2	deptname	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
3	mgrid	varchar(5)	utf8mb4_general_ci		No	None			Change Drop More

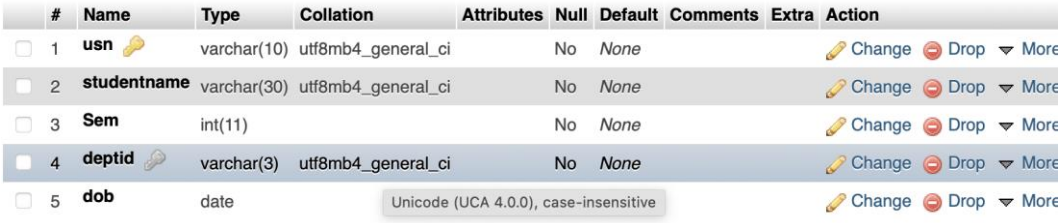
Fig 3.1.1:Department table

- Department table :The table department contains three attributes . Department id (dept_id), department name (dept_name) and then the manager id (mgr_id). The dept_id is given to uniquely identify the department . Then we have the mgr_id which is being referred from the faculty table letting us know when is in charge of that particular department .

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	fid	varchar(5)	utf8mb4_general_ci		No	None			Change Drop More
2	fname	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
3	deptid	varchar(3)	utf8mb4_general_ci		No	None			Change Drop More
4	password	varchar(60)	utf8mb4_general_ci		No	None			Change Drop More

Fig 3.1.2:Faculty table

- Faculty Table :The faculty table has four attributes . Faculty id (Fid), faculty name (fname) , department id (dept_id) and password (password). Fid is a unique id given to each faculty , this is the primary key of this table . We have other attribute such as the fname which stores the name of the teachers . Then we have the dept_id indicating which department the faculty belongs to . Finally we have the password , this attribute is given to the teachers so that they can login using the fid and the password .



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	usn	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
2	studentname	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
3	Sem	int(11)			No	None			Change Drop More
4	deptid	varchar(3)	utf8mb4_general_ci		No	None			Change Drop More
5	dob	date							Change Drop More

Unicode (UCA 4.0.0), case-insensitive

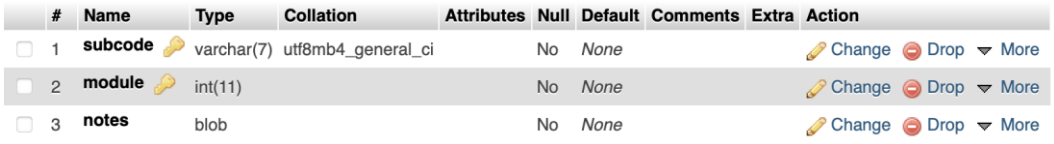
Fig 3.1.3:Student table

- Student Table: The student table contains five attributes which are USN (usn), student name (studentname) , semester (sem) , department id (deptid) , date of birth (dob). Usn is the attribute which uniquely identifies each student . All of these attributes specifies certain features of the student . Date of birth (dob) is used by the student to login .

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	subcode	varchar(7)	utf8mb4_general_ci		No	None			Change Drop More
2	subname	varchar(60)	utf8mb4_general_ci		No	None			Change Drop More
3	deptid	varchar(3)	utf8mb4_general_ci		No	None			Change Drop More
4	sem	int(11)			No	None			Change Drop More

Fig 3.1.4:Subject table

- Subject table:In the table subject we have all the details we need about each subject , such as the subject code (subcode) which uniquely identifies each subject , subject name (subname) ,the department which has thus subject (dept_id) and finally we have details about which semester this subject belongs to .



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	subcode	varchar(7)	utf8mb4_general_ci		No	None			Change Drop More
2	module	int(11)			No	None			Change Drop More
3	notes	blob			No	None			Change Drop More

Fig 3.1.5:Notes table

- Notes table :We are using the the table notes to store the pdf file content(notes) . To understand which subject's notes it is we have the subject code (subcode) and we also have module (module) which will tell us which module notes the file is .

3.2 Constraint on Tables

In the table department we have a primary key (dept_id) constrain and a foreign key (mgr_id) constrain referring to an attribute (fid) in the table faculty . In the table faculty we have a primary key (fid) constrain and a foreign key (dept_id) constrain referring to an attribute (dept_id) in the table department .

In the table students we have a primary key (usn) constrain and a foreign key (dept_id) constrain referring to an attribute (dept_id) in the table department .

In the table subject we have a primary key (subcode) constrain and a foreign key (dept_id) constrain referring to an attribute (dept_id) in the table department.

In the table Notes we have used combinational primary key setting subject code and module number (subcode, module) as the primary key specifying that their combination cannot repeat where they may be repeated separately .

The primary helps us uniquely identify the tuple or the row , the primary can neither be repeated nor kept null where as the foreign key helps us to maintain the relation between two tables . Foreign kelps us access datas of one table from another .

3.3 Back End Implementations

```
CREATE TABLE `department` (
  `deptid` varchar(3) NOT NULL,
  `deptname` varchar(50) NOT NULL,
  `mgrid` varchar(6) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
--
-- Dumping data for table `department`
--
INSERT INTO `department` (`deptid`, `deptname`, `mgrid`) VALUES
('CS', 'Computer Science & Eng.', 'CS100'),
('CV', 'Civil Eng.', 'CV200'),
('EC', 'Electronics & Communicational Eng.', 'EC300'),
('FY', 'First Year', 'FY600'),
```

```
('IS', 'Information Science & Eng.', 'IS400'),

('MAT', 'Mathematics', 'MAT700'),

('ME', 'Mechanical Eng.', 'ME500');

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

--

-- Table structure for table `faculty`

--

CREATE TABLE `faculty` (

  `fid` varchar(6) NOT NULL,

  `fname` varchar(30) NOT NULL,

  `deptid` varchar(3) NOT NULL,

  `password` varchar(60) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `faculty`

--

INSERT INTO `faculty` (`fid`, `fname`, `deptid`, `password`) VALUES

('CS100', 'Shivamurthy', 'CS', 'shivamurthy'),

('CS101', 'Deepu R', 'CS', 'deepu'),

('CS102', 'Santosh Kumar ', 'CS', 'santosh'),

('CV200', 'Ramakrishna Gowda', 'CV', 'ramakrishna'),

('EC300', 'Mahesh  Rao', 'EC', 'mahesh'),

('FY600', 'Manjunath', 'FY', 'manjunath'),

('IS400', 'Sharath Kumar', 'IS', 'sharath'),

('MAT700', 'Shrinivas', 'MAT', 'shrinivas'),
```

```
('ME500', 'Mohammed Khaiser', 'ME', 'mohammed');

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

--

-- Table structure for table `notes`

--

CREATE TABLE `notes` (

  `subcode` varchar(7) NOT NULL,

  `module` int(11) NOT NULL,

  `notes` varchar(500) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `notes`

--

INSERT INTO `notes` (`subcode`, `module`, `notes`) VALUES

('18CS53', 1, 'DBMS MODULE 1.pdf'),

('18CS53', 2, 'DBMS MODULE 2.pdf'),

('18CS53', 3, 'DBMS MODULE 3.pdf'),

('18CS53', 4, 'DBMS MODULE 4.pdf'),

('18CS53', 5, '502238.docx');

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

--

-- Table structure for table `student`

--

CREATE TABLE `student` (

  `usn` varchar(10) NOT NULL,
```

```
  `studentname` varchar(30) NOT NULL,
  `deptid` varchar(3) NOT NULL,
  `dob` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
--
-- Dumping data for table `student`
--
INSERT INTO `student` (`usn`, `studentname`, `deptid`, `dob`) VALUES
('4MH18CS077', 'Preethan H N', 'CS', '2000-02-29'),
('4MH18CS099', 'SATHWIK NIDHI Y T', 'CS', '1999-12-26'),
('4MH18CS100', 'SENA AHAMMEED', 'CS', '1998-11-22'),
('4MH18CS107', 'SREYAS P', 'CS', '2000-08-21'),
('4mh18cs117', 'Thejaswini S', 'CS', '2021-01-01');
-- --------------------------------------------------------
--
-- Table structure for table `subject`
--
CREATE TABLE `subject` (
  `subcode` varchar(8) NOT NULL,
  `subname` varchar(60) NOT NULL,
  `deptid` varchar(3) NOT NULL,
  `sem` int(1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
--
-- Dumping data for table `subject`
```

```
--

INSERT INTO `subject` (`subcode`, `subname`, `deptid`, `sem`) VALUES

('18CHE22', 'Chemistry', 'FY', 2),

('18CIV14', 'Civil', 'FY', 1),

('18CPS23', 'C Programming', 'FY', 2),

('18CS32', 'DS', 'CS', 3),

('18CS33', 'ADE', 'CS', 3),

('18CS34', 'CO', 'CS', 3),

('18CS35', 'SE', 'CS', 3),

('18CS36', 'DMS', 'CS', 3),

('18CS42', 'DAA', 'CS', 4),

('18CS43', 'OS', 'CS', 4),

('18CS44', 'MCES', 'CS', 4),

('18CS45', 'OOC', 'CS', 4),

('18CS46', 'DC', 'CS', 4),

('18CS51', 'MEIT', 'CS', 5),

('18CS52', 'CNS', 'CS', 5),

('18CS53', 'DBMS', 'CS', 5),

('18CS54', 'ATC', 'CS', 5),

('18CS55', 'Python', 'CS', 5),

('18CS56', 'UNIX', 'CS', 5),

('18CS61', 'Data Mining', 'CS', 6),

('18CS62', 'Big Data ', 'CS', 6),

('18CS71', 'ACA', 'CS', 7),

('18CS72', 'ML', 'CS', 7),
```

('18CS81', 'IOT', 'CS', 8),

('18CS82', 'UID', 'CS', 8),

('18EC31', 'Network Theory', 'EC', 3),

('18EC32', 'Electronic Devices', 'EC', 3),

('18EC41', 'Analog Circuits', 'EC', 4),

('18EC42', 'Microcontroller', 'EC', 4),

('18EC51', 'Digital Signal Processing', 'EC', 5),

('18EC52', 'Electromagnetic Waves', 'EC', 5),

('18EC61', 'Digital Communication', 'EC', 6),

('18EC62', 'Microwaves and Antennas', 'EC', 6),

('18EC71', 'Computer Networks', 'EC', 7),

('18EC72', 'VLSI Design', 'EC', 7),

('18EC81', 'Network Security', 'EC', 8),

('18EC82', 'Radar Engineering', 'EC', 8),

('18EGDL16', 'Eng, Graphics', 'FY', 1),

('18ELE13', 'Electrical', 'FY', 1),

('18ELN24', 'Electronics', 'FY', 2),

('18MAT11', 'Mathematics 1', 'MAT', 1),

('18MAT12', 'Mathematics 2', 'MAT', 2),

('18MAT31', 'Mathematics 3', 'MAT', 3),

('18MAT41', 'Mathematics 4', 'MAT', 4),

('18ME25', 'Mechanical', 'FY', 2),

('18PHY12', 'Physics', 'FY', 1);

--

-- Indexes for dumped tables

```
--

--

-- Indexes for table `department`

--

ALTER TABLE `department`

  ADD PRIMARY KEY (`deptid`),

  ADD KEY `department_ibfk_1` (`mgrid`);

--

-- Indexes for table `faculty`

--

ALTER TABLE `faculty`

  ADD PRIMARY KEY (`fid`),

  ADD KEY `faculty_ibfk_1` (`deptid`);

--

-- Indexes for table `notes`

--

ALTER TABLE `notes`

  ADD PRIMARY KEY (`subcode`,`module`);

--

-- Indexes for table `student`

--

ALTER TABLE `student`

  ADD PRIMARY KEY (`usn`),

  ADD KEY `student_ibfk_1` (`deptid`);

--
```

```
-- Indexes for table `subject`
--
ALTER TABLE `subject`
  ADD PRIMARY KEY (`subcode`),
  ADD KEY `subject_ibfk_1` (`deptid`);
--
-- Constraints for dumped tables
--
-- Constraints for table `department`
--
ALTER TABLE `department`
   ADD CONSTRAINT `department_ibfk_1` FOREIGN KEY (`mgrid`) REFERENCES `faculty` (`fid`) ON DELETE CASCADE ON UPDATE CASCADE;
--
-- Constraints for table `faculty`
--
ALTER TABLE `faculty`
   ADD CONSTRAINT `faculty_ibfk_1` FOREIGN KEY (`deptid`) REFERENCES `department` (`deptid`) ON DELETE CASCADE ON UPDATE CASCADE;
--
-- Constraints for table `notes`
--
ALTER TABLE `notes`
  ADD CONSTRAINT `notes_ibfk_1` FOREIGN KEY (`subcode`) REFERENCES `subject` (`subcode`) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
--
-- Constraints for table `student`
--
ALTER TABLE `student`
  ADD CONSTRAINT `student_ibfk_1` FOREIGN KEY (`deptid`) REFERENCES `department` (`deptid`) ON DELETE CASCADE ON UPDATE CASCADE;
--
-- Constraints for table `subject`
--
ALTER TABLE `subject`
  ADD CONSTRAINT `subject_ibfk_1` FOREIGN KEY (`deptid`) REFERENCES `department` (`deptid`) ON DELETE CASCADE ON UPDATE CASCADE;
COMMIT;
```

3.4 Front End Implementations

- Login page

```
<?php
 // define database related variables
   $db_name = 'student notes management system';
   $host = 'localhost';
   $user = 'root';
   $password = '';
    $db = mysqli_connect($host, $user, $password, $db_name);
    if(mysqli_connect_errno()) {
        die("Failed to connect with MySQL: ". mysqli_connect_error());   }
session_start();
```

```
$error="";
if($_SERVER["REQUEST_METHOD"] == "POST") {
   if(isset($_POST['action']) && $_POST['action'] == "admin"){
        $myusername = mysqli_real_escape_string($db,$_POST['username']);
        $mypassword = mysqli_real_escape_string($db,$_POST['psw']);
         $sql = " SELECT mgrid,password from department ,faculty where mgrid=fid and fid =
'$myusername' and password = '$mypassword'";
        $result = mysqli_query($db,$sql) ;//or die( mysqli_error($db));
        $row = mysqli_fetch_array($result,MYSQLI_ASSOC);
        $count = mysqli_num_rows($result);
        if($count == 1) {
           header("location: admin.php");
        }else {
           $error = "Invalid Username or Password";}
      }
    elseif(isset($_POST['action']) && $_POST['action'] == "student"){
        $myusername = mysqli_real_escape_string($db,$_POST['username']);
        $mypassword = mysqli_real_escape_string($db,$_POST['psw']);
          $sql = " SELECT usn,dob FROM student WHERE usn = '$myusername' and dob =
'$mypassword'";
        $result = mysqli_query($db,$sql) ;//or die( mysqli_error($db));
        $row = mysqli_fetch_array($result,MYSQLI_ASSOC);
        $count = mysqli_num_rows($result);
        if($count == 1) {
           header("location: branch.php");
```

```
        }else {
          $error = "Invalid Username or Password";}
      }
    elseif(isset($_POST['action']) && $_POST['action'] == "faculty")
      {
        $myusername = mysqli_real_escape_string($db,$_POST['uname']);
        $mypassword = mysqli_real_escape_string($db,$_POST['psw']);
          $sql = " SELECT fid,password FROM faculty WHERE fid = '$myusername' and
password = '$mypassword'";
        $result = mysqli_query($db,$sql) ;
        $row = mysqli_fetch_array($result,MYSQLI_ASSOC);
        $count = mysqli_num_rows($result);
        if($count == 1) {
          header("location: faculty.php");
        }else {
          $error = "Invalid Username or Password";}
      }
  }
?>
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<div>
<body>
```

```
<div class="header">
  <center><img src="images/mitlogo.png" alt="logo"></center>
</div>
<h2 style="color:green; font-size:50px;">Save papers! Save trees!</h2>
<div class="login-block">
        <button onclick="document.getElementById('id01').style.display='block'"
style="width:auto;">Admin Login</button>
 <div id="id01" class="modal">
          <form class="modal-content animate" action="<?php echo
$_SERVER['SCRIPT_NAME']; ?>" method="POST">
    <input type="hidden" name="action" value="admin">
   <div class="imgcontainer">
      <span onclick="document.getElementById('id01').style.display='none'" class="close"
title="Close Modal">&times;</span>
    <img src="images/logo.png" alt="Avatar" class="avatar">
   </div>
   <div class="container">
    <h2 style="color:green;">Admin</h2>
    <label for="username"><b>Admin id</b></label>
    <input type="text" placeholder="Enter Your Faculty id" name="username" required>
    <label for="psw"><b>Password</b></label>
    <input type="password" placeholder="Enter Password" name="psw" required>
    <button type="submit" value="adminLogin" name="adminLogin">Login</button>
   </div>
```

```
        <div style = "font-size:20px; color:#cc0000; margin-top:10px"><?php echo $error;
?></div>
   <div class="container" style="background-color:#f1f1f1">
      <button type="button" onclick="document.getElementById('id01').style.display='none'"
class="cancelbtn">Cancel</button>
   </div>
   </form>
 </div>
</div>
<div class=“login-block">
<buttononclick="document.getElementById('id02').style.display='block'"
style="width:auto;">Student Login</button>
 <div id="id02" class="modal">
              <form class="modal-content animate" action="<?php echo
$_SERVER['SCRIPT_NAME']; ?>" method="POST">
 <input type="hidden" name="action" value="student">
<div class=“imgcontainer">
<span onclick="document.getElementById('id02').style.display='none'" class="close"
        title="Close Modal">&times;</span>
<img src="images/logo.png" alt="Avatar" class="avatar">
  </div>
   <div class="container">
     <h2 style="color:green;">Student</h2>
    <label for="username"><b>USN</b></label>
    <input type="text" placeholder="Enter USN" name="username" required>
```

```
    <label for="psw"><b>Password(dob)</b></label>
      <input type="password" placeholder="Enter Password (yyyy-mm-dd)" name="psw" required>
    <button type="submit" value="studentLogin" name="studentLogin">Login</button>
  </div>
      <div style = "font-size:20px; color:#cc0000; margin-top:10px"><?php echo $error; ?></div>
  <div class="container" style="background-color:#f1f1f1">
     <button type="button" onclick="document.getElementById('id02').style.display='none'" class="cancelbtn">Cancel</button>
  </div>
  </form>
 </div>
</div>
<div class="login-block">
          <button onclick="document.getElementById('id03').style.display='block'" style="width:auto;">Faculty Login</button>
 <div id="id03" class="modal">
            <form class="modal-content animate" action="<?php echo $_SERVER['SCRIPT_NAME']; ?>" method="POST">
    <input type="hidden" name="action" value="faculty">
  <div class="imgcontainer">
      <span onclick="document.getElementById('id03').style.display='none'" class="close" title="Close Modal">&times;</span>
    <img src="images/logo.png" alt="Avatar" class="avatar">
```

```
    </div>
    <div class="container">
        <h2 style="color:green;">Faculty</h2>
      <label for="uname"><b>Faculty id</b></label>
      <input type="text" placeholder="Enter Your Faculty id" name="uname" required>
      <label for="psw"><b>Password</b></label>
      <input type="password" placeholder="Enter Password" name="psw" required>
      <button type="submit">Login</button>
    </div>
     <div style = "font-size:20px; color:#cc0000; margin-top:10px"><?php echo $error; ?></div>
    <div class="container" style="background-color:#f1f1f1">
       <button type="button" onclick="document.getElementById('id03').style.display='none'" class="cancelbtn">Cancel</button>
    </div>
    </form>
  </div>
</div>
<script>
// Get the modal
var modal = document.getElementById('id01');
var modal1 = document.getElementById('id02');
var modal2 = document.getElementById('id03');
// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
```

```
    if (event.target == modal) {
        modal.style.display = "none";
    }
    if (event.target == modal1) {
        modal1.style.display = "none";
    }
    if (event.target == modal2) {
        modal2.style.display = "none";
    }
}
</script>
<body>
</body>
</body>
</html>
```

- Faculty page

```
<?php
 // define database related variables
   $db_name = 'student notes management system';
   $host = 'localhost';
   $user = 'root';
   $password = '';
    $db = mysqli_connect($host, $user, $password, $db_name);
    if(mysqli_connect_errno()) {
        die("Failed to connect with MySQL: ". mysqli_connect_error());
```

```
    }
session_start();
$error="";
        if (isset($_POST['uploaded'])==1) {
            $file = $_FILES['notes'];
            $fileName = $_FILES['notes']['name'];
            $fileTmpName = $_FILES['notes']['tmp_name'];
            $fileSize = $_FILES['notes']['size'];
            $fileError = $_FILES['notes']['error'];
            $fileType = $_FILES['notes']['type'];
            $fileExt = pathinfo($fileName, PATHINFO_EXTENSION);
            $allowed = array('pdf', 'txt', 'doc', 'docx', 'png', 'jpg', 'jpeg');
            if (in_array($fileExt, $allowed)) {
              if ($fileError === 0) {
                if ($fileSize < 104857601) {
                  $q = "SELECT * FROM notes WHERE notes='$fileName'";
                  if (mysqli_num_rows(mysqli_query($db, $q)) == 0) {
                              $fileDestination = 'C:\\xampp\\htdocs\\student notes management system\\files\\'.$fileName;
                    move_uploaded_file($fileTmpName, $fileDestination)
                    $created = @date('Y-m-d H:i:s');
                    //$description = mysqli_real_escape_string($db, $_POST['description'])
                          $sql = "INSERT INTO notes (subcode, module, notes) VALUES ('$_POST[subcode]','$_POST[module]','$fileName')";
                        mysqli_query($db, $sql);
```

```
                echo "<p class='alert alert-success'>File uploaded successfully</p><br>";
              }
              else{
                    echo "<p class='alert alert-warning'>File already exixts. Check it out OR Change your filename and try again...</p><br>";
              }
            }
            else{
              echo "<p class='alert alert-warning'>File too large</p><br>";}
          }
          else{
            echo "<p class='alert alert-danger'>Error uploading file</p><br>";
          }
        }
        else{
          echo "<p class='alert alert-danger'>Invalid file type</p><br>";
        }
    }
  elseif(isset($_POST['ok']))
  {
      $file = $_FILES['notes'];
        $fileName = $_FILES['notes']['name'];
        $fileTmpName = $_FILES['notes']['tmp_name'];
        $fileSize = $_FILES['notes']['size'];
        $fileError = $_FILES['notes']['error'];
```

```
$fileType = $_FILES['notes']['type'];
$fileExt = pathinfo($fileName, PATHINFO_EXTENSION);
$allowed = array('pdf', 'txt', 'doc', 'docx', 'png', 'jpg', 'jpeg');
if (in_array($fileExt, $allowed)) {
  if ($fileError === 0) {
    if ($fileSize < 104857601) {
      $q = "SELECT * FROM notes WHERE notes='$fileName'";
      if (mysqli_num_rows(mysqli_query($db, $q)) == 0) {

                $fileDestination = 'C:\\xampp\\htdocs\\student notes management system\\files\\'.$fileName;
        move_uploaded_file($fileTmpName, $fileDestination);
        $created = @date('Y-m-d H:i:s');
        //$description = mysqli_real_escape_string($db, $_POST['description']);
         $sql = "Update notes set notes='$fileName' where subcode='$_POST[subcode]' and module='$_POST[module]'";
            mysqli_query($db, $sql);
        echo "<p class='alert alert-success'>File uploaded successfully</p><br>";
       }
       else{
        echo "<p class='alert alert-warning'>Unable to Update. Check it out OR Change your filename and try again...</p><br>";
       }
      }
      else{
```

```
                echo "<p class='alert alert-warning'>File too large</p><br>";
              }
             }
             else{
               echo "<p class='alert alert-danger'>Error uploading file</p><br>";
             }
            }
            else{
             echo "<p class='alert alert-danger'>Invalid file type</p><br>";}
  }
elseif(isset($_POST['Delete']))
  {
     if(empty($_POST['subcode']) || empty($_POST['module']) )
     {
        echo ' Please Fill in the Blanks ';
     }
     else
     {
        $subcode = $_POST['subcode'];
        $module = $_POST['module'];
        $query = " delete from notes where subcode='$subcode' and module='$module' ";
        $result = mysqli_query($db,$query);
        if($result)
        {
          echo "File Successfully Deleted";
```

```
        }
        else
        {
            echo 'Something Went Wrong';    }
      }
   }
?>
<!DOCTYPE html>
<html>
   <body bgcolor="white">
      <meta name="viewport" content="width=device-width, initial-scale=1">
                  </head>
                  <body>
         <ul>
        <li><a class="tablinks" onclick="openCity(event, 'view')" id="defaultOpen">$<a><li>
           <li><a class="tablinks" onclick="openCity(event, 'Insert')">Insert Notes</a></li>
           <li><a class="tablinks" onclick="openCity(event, 'Update')">Update Notes</a><li>
           <li><a class="tablinks" onclick="openCity(event, 'Delete')">Delete Notes</a></li>
           <li><a class="tablinks" href="branch.php" >Notes</a></li>
           <li style="float:right"><a class="active" href="login.php">Logout</a></li>
         </ul>
      <div id="view" class="tabcontent">
        <img src="images/college.gif" alt="college"  style=" width:1247px ">
      </div>
      <div id="Insert" class="tabcontent">
```

```
                    <h3 style="color:green; font-size:45px; ">Insert Notes</h3>
        <form method="POST" enctype="multipart/form-data">
                    <p>
          <label style="font-size:30px">Subject Code    :</label>
            <input type="text" name="subcode" placeholder="Enter SubCode" class="form-
control mb-2" required>
        </p>
        <p>
            <label  style="font-size:30px">Module           
  :</label>
              <input type="text" name="module" placeholder="Enter Module Number"
class="form-control mb-2" required>
        </p>
        <div class="form-group">
         <div class="input-group">
          <label for="file" style="font-size:30px">Upload Notes    :</label>
          <span class="input-group-addon">
                <i class="fa fa-info-circle" data-toggle="tooltip" data-placement="top"
title="Supported File Format: pdf, txt, doc, docx, png, jpg, jpeg" aria-hidden="true"></i>
          </span>
            <input type="file" class="btn btn-default" style="width: 225px" id="notes"
name="notes" required>
         </div>
        </div>
        <input type="submit" class="btn btn-success" name="uploaded" value="Upload">
```

```
<img src="images/in.png" alt="in" style="float:right;margin:-150px 100px">
</form>
                </div>
<div id="Update" class="tabcontent">
        <h3 style="color:green; font-size:45px; ">Update Notes</h3>
<form method="POST" enctype="multipart/form-data">
                <p>
        <label style="font-size:30px">Subject Code    :</label>
        <input type="text" name="subcode" placeholder="Enter SubCode" class="form-control mb-2" required>
</p>
<p>
        <label style="font-size:30px">Module              :</label>
        <input type="text" name="module" placeholder="Enter Module Number" class="form-control mb-2" required>
</p>
<div class="form-group">
   <div class="input-group">
        <label for="file" style="font-size:30px">Upload Notes    :</label>
        <span class="input-group-addon">
                <i class="fa fa-info-circle" data-toggle="tooltip" data-placement="top" title="Supported File Format: pdf, txt, doc, docx, png, jpg, jpeg" aria-hidden="true"></i>
        </span>
```

```
                    <input type="file" class="btn btn-default" style="width: 225px" id="notes"
name="notes" required>
                </div>
              </div>
               <input type="submit" class="btn btn-success" name="ok" value="Upload">
            <img src="images/up.png" alt="up" style="float:right;margin:-150px 100px">
           </form>
        </div>
                              <div id="Delete" class="tabcontent">
 <h3 style="color:green; font-size:45px; ">Delete Notes</h3>
  <form method="POST">
  <p>
          <label style="font-size:30px">Subject Code    :</label>
              <input type="text" name="subcode" placeholder="Enter SubCode" class="form-
control mb-2" required>
   </p>
   <p>
            <label style="font-size:30px">Module             
:</label>
                  <input type="text" name="module" placeholder="Enter Module Number"
class="form-control mb-2" required>
          </p>
          <input type="submit" value="Delete" name="Delete">
          </form>
          <img src="images/dlt.png" alt="dlt" style="float:right;margin:-150px 100px">
```

```
                    </div>
<script>
function openCity(evt, cityName) {
  var i, tabcontent, tablinks;
  tabcontent = document.getElementsByClassName("tabcontent");
  for (i = 0; i < tabcontent.length; i++) {
    tabcontent[i].style.display = "none";
  }
  tablinks = document.getElementsByClassName("tablinks");
  for (i = 0; i < tablinks.length; i++) {
    tablinks[i].className = tablinks[i].className.replace(" active", "");
  }
  document.getElementById(cityName).style.display = "block";
  evt.currentTarget.className += " active";
}
// Get the element with id="defaultOpen" and click on it
document.getElementById("defaultOpen").click();
</script>
</html>
```

Chapter 4

RESULT ANALYSIS

4.1 Snap Shots

Fig 4.1.1 : Home page 1.

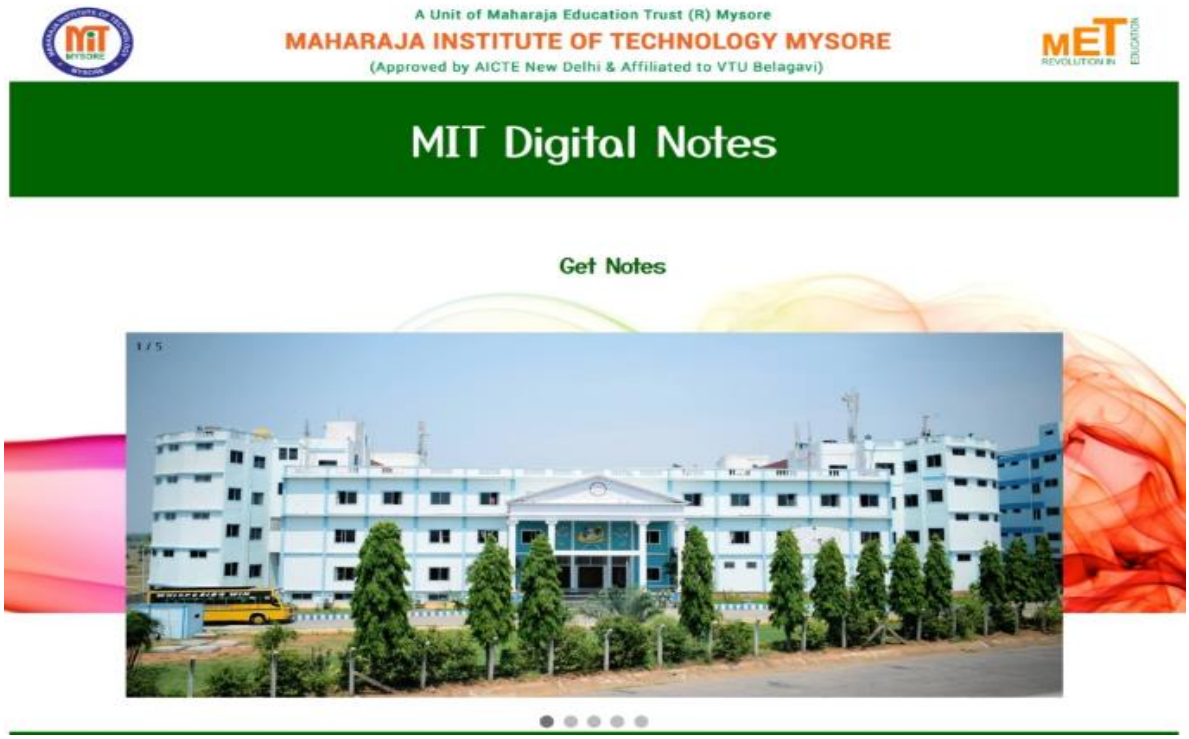


Fig 4.1.2 : Home page 2.

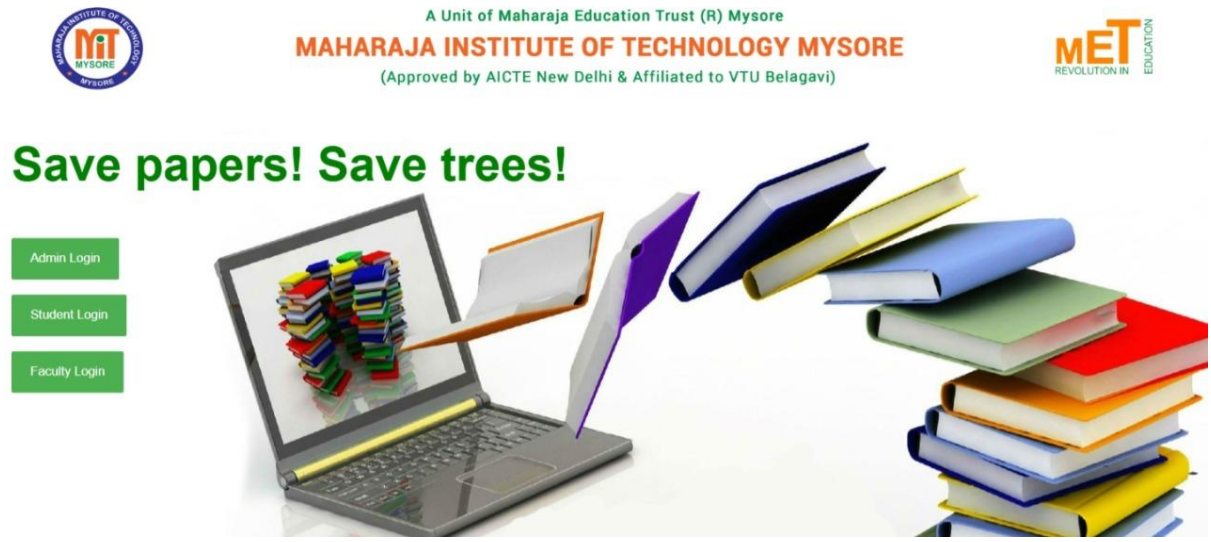


Fig 4.1.3 : Login Option.

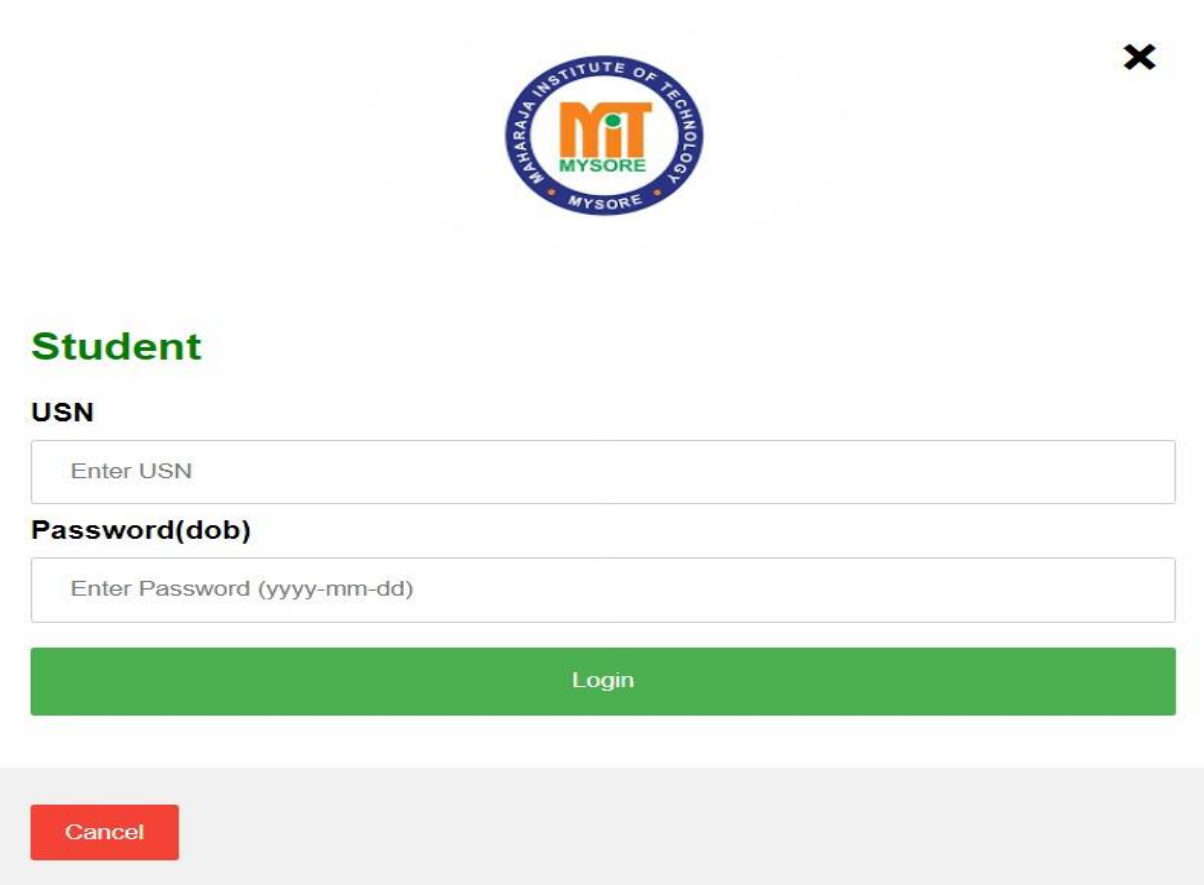


Fig 4.1.4 : Login.

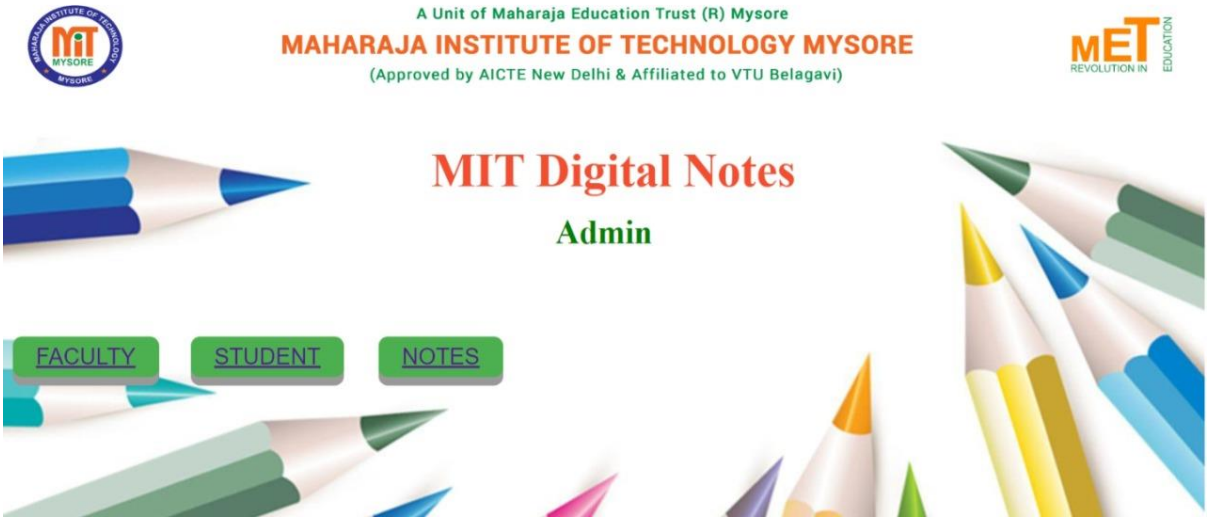


Fig 4.1.5 : Admin authority.

Fig 4.1.6 : Select branch.

Fig 4.1.7 : Manipulate details.

Fig 4.1.8 : Select semister.

Fig 4.1.9 : Select subject.

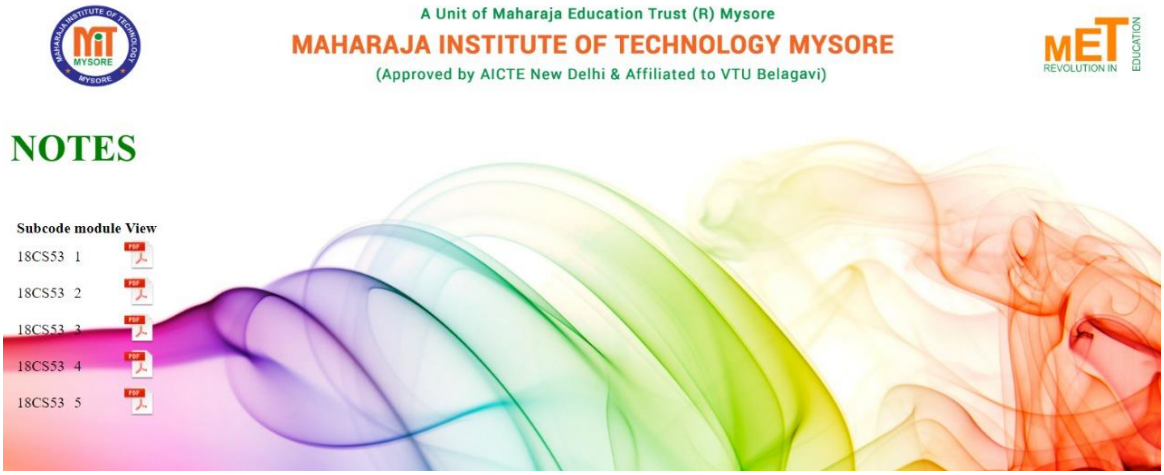


Fig 4.1.10 : Select notes.

4.2 Discussion

- Get Notes

MIT digital notes page points us towards ‘get notes’ where we can see the soft copies of our notes but in our webpage it points to different respected aspects .thus access is simplified.

- Admin , Student, Faculty login

If we press get notes it takes us to the next page where we can see ‘admin login’, student login ,faculty login this page helps all the key aspects such as admin ,student ,faculty to point towards their concerned page.

- Admin login

In this page we see two columns showing faculty id and password where only a management level faculty manages the admin page is given access.

- Student login

Student login page is given access to all students by entering their student id and password where we have used date of birth just to make easy access for the students

- Faculty login

In this page we see two columns showing faculty id and password where all faculty are given access to the faculty page. After accessing into their allotted pages they are given a set of own privileges such as admin has access to both faculty and student where he can make changes or can insert delete or update as he requires.

- Insert Delete or Update

Here both faculty and students are given access to these functions where one can insert notes and delete notes if he/she finds inappropriate or update the notes as per the syllabus. as we have given access to both because by giving access we are increasing the database which is helpful for each one of them.

- Insert , Update or Delete faculty

This page helps us to create faculty id and passwords where we can insert or update faculty id , faculty name, department id , password for both the pages.

- Search

We can further search a particular faculty/student by just giving a minimum information such as name , id or department can be further classified.

- Select your branch /semester

As we encounter different branches or department such as EC, CS, Mech, Civil, IS in engineering we further helped the user to get easy access for there required data, thus entering your own department or branch we have divided them into semesters for the user to get into there semester to fetch notes.

4.2 Testing

System testing is actually a series of different whose primary purpose is to fully exercise the computer based system. Software testing is critical element of software quality assurance and represent the ultimate review of the specification, design and coding. System testing makes a logical assumption that all the part of the system is correct; the goal will be successfully achieved. Testing is the final verification and validation activity within the organisation itself. During testing the major activities are concerned on the examinations and modification of the source code.

SLNO	Description	Input	Output	Result
1	Login	Correct username Correct password	Login Successful	Pass
2	Login	Wrong username Wrong password	Login Unsuccessful	Pass
3	Login	Correct username Wrong password	Login successful	Fail
4	Login	Wrong username Wrong password	Login Successful	Fail

Chapter 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

The application is designed in such a way that any further enhancements can be done with ease. The system has the capability for easy integration with other systems. New modules can be added to the existing system with less effort. The system has six classes. Each of these classes has various procedures and functions. In future a new function or procedure can be easily added in the system through these classes. Or even a new class can be added. The system generates only a limited number of reports. If more detailed reports are required the system can be directed. Even though the system has well communication facility, it's not enough. The mail service can be enhanced with features bcc, cc etc. the system has full security but the account information for the customer credit information. Thus by adding this module the system transaction will be improved.

5.2 Future Enhancement

This application can be easily implemented under various situations. We can add new features as and when we require. Reusability is possible as and when require in this application. There is flexibility in all the modules.

Chapter 6

REFERENCES

- Learn HTML5 ,CSS3 ,PHP ,JAVA Script from W3Schools an online web tutorials.
  - Website link : https://www.w3schools.com
- Refered some web designing video tutorials from YouTube.
  - https://www.youtube.com/watch?v=wHFflWvii3M&t=140s
  - https://www.youtube.com/watch?v=lBfshkPlMW8&t=1974s