



Pune Vidhyarthi Griha's
College Of Engineering and Technology

**PUNE VIDHYARTHI GRIHA'S
COLLEGE OF ENGINEERING AND TECHNOLOGY, PUNE-09.**

DEPARTMENT OF INFORMATION TECHNOLOGY

**A
PROJECT REPORT
ON**

“STUDENT PLANNER”

**Under the Guidance of
Prof. N.R.SONAWANE**

**Submitted By
Abhishek V. Karanjekar.
Jeevan M. Pawar.
Janhavi J. Wagh.**

Academic Year: 2017 - 2018



Pune Vidhyarthi Griha's
College Of Engineering and Technology

**PUNE VIDHYARTHI GRIHA'S
COLLEGE OF ENGINEERING AND TECHNOLOGY, PUNE-09.**

CERTIFICATE

This is to certify that, the project entitled as

“ Student Planner “

Is a bonafide work done by

Abhishek V. Karanjekar.

Jeevan M. Pawar.

Janhavi J. Wagh.

Submitted in the partial fulfilment for the award of Third Year Engineering in Information Technology as prescribed by Savitribai Phule Pune University, Pune as a record of students own work carried out under the guidance of **Prof. N. R. Sonawane** during academic year 2017-2018.

Prof. N.R.Sonawane

Project Guide

Prof. S.S.Dixit

H.O.D Comp/IT Department

ACKNOWLEDGEMENT

With immense pleasure, we are presenting this project report as a part of the curriculum of Information Technology. We wish to thank all the people who gave us unending support right from the stage the idea was conceived.

We express our profound thanks to our respected Head of Department Prof. S.S.Dixit, whose advice and valuable guidance helped us in making this project interesting and successful one.

We are also grateful to our project guide Prof. N.R.Sonawane, whose valuable guidance helped us in making this project interesting and successful one by providing us with immense support and guidance throughout the project.

We also thank all those who have directly or indirectly guided and helped us in preparation of this project.

Last but not the least we thank our beloved Parents, Friends and our well-wishers who helped us to do this project by their kind help and assistance.

Abhishek V. Karanjekar.

Jeevan M. Pawar.

Janhavi J. Wagh.

ABSTRACT

Student Planner deals **Attendance Management** , **Attendance Reports** , **To Do Lists**, **Reminders**. With Attendance Management a student can maintain his/her own attendance details. It generates the attendance of the student on basis of presence in class. It is maintained on the daily basis of their attendance. Student gets to sign up with separate username & password to create his/her own account to maintain the attendance.

Also, for those busy students who always like to list things to be done. A feature is provided called To Do List. This feature comes with the ability to maintain and easily strike out completed tasks. Another feature called reminders allows student set important reminders. Student gets a pop up of reminder before a week of task. This pop up is shown on user homepage so that user will be easily notified.

The system being developed is Student Planner is using MySQL and PHP. It is a real time developed software for students to keep a track of their attendance. It also helps student to create To Do Lists as well as set important Reminders .It performs Attendance updation as well as generates dynamic reports based on attendance.

INDEX

Sr. No.	Title	Page No.
1.	Introduction	6
2.	Feasibility Study	7
3.	Requirement Analysis	8
4.	Technology Overview	9
5.	Conceptual & Logical Designs	15
6.	Database Normalization	16
7.	Hardware & Software Requirements	18
8.	Graphical User Interface	21
9.	Source Code	22
10.	Testing Document	29
11.	Future Enhancement	34
12.	Conclusion	35
13.	Bibliography	36

Chapter-1

1. INTRODUCTION

Student Planner deals **Attendance Management , Attendance Reports , To Do Lists, Reminders**. With Attendance Management a student can maintain his/her own attendance details. It generates the attendance of the student on basis of presence in class. It is maintained on the daily basis of their attendance. Student gets to sign up with separate username & password to create his/her own account to maintain the attendance. With dynamic chart generation student gets a graphical view of the attendance. To Do List and Reminders help student to monitor tasks and events.

PURPOSE OF THE SYSTEM

A common problem faced by all students is that once teacher takes attendance students don't know their current attendance status. And they have to approach their teacher to know the exact attendance.

There are various systems available in the market for student attendance but none of them allows student to maintain the attendance.

Additionally the system is advantageous to maintain task lists and events at the same place.

BENEFITS OF THIS SYSTEM:-

This system benefits a Student in the following ways :

- Easily Maintain his/her Attendance
- Maintain To-Do List
- Setting Reminders
- Analyse Dynamic Generated Attendance Reports
- Hardware And Software Requirements Are Minimum

Chapter-2

2. Feasibility Study

Before recommending the new system, it is necessary to investigate whether it is possible as well as feasible to develop the new system. The important outcome of this preliminary investigation is the determination of whether the proposed system is feasible for target user or not. Feasibility & Risk analysis are related in many ways. If project risk is great, the feasibility of producing quality software is required.

The feasible analysis was carried out It Involved following steps:

- Identification of servers user need.
- Identification of how different tasks are carried out for keeping record of client user activities.
- Identifying whether the proposed system can meet user's need.
- And finally proving the system technically, financially & operational feasible.

2.1 Economic Feasibility:

It includes an evaluation of development cost weight against the ultimate income or benefit derived from the developed system or product.

The approximate cost of this project will be definitely less than any other software with this kind of reliability, ease, functionality & feasibility since the application does not require any extra hardware or any supporting software.

2.2: Technical Feasibility:

Technical analysis begins with an assessment of the technical viability of proposed system. What technologies are required to accomplish system functions and requirements? Which other resources (hardware and software) are available to build the system? The system is developed using MySQL and PHP.

2.3: Operational Feasibility:

In this type of feasibility study, the operational implementation of the system is considered. Checking is done regarding whether it is feasible for the user to use the software or will there be any instance from the user.

Proposed system is beneficial only if they can turn in to information system that will meet user's requirement. That is new in the new developing world with the automatic capturing image and automatic saving the data. The users ease with minimum effort to keep the record hence it is operationally feasible.

Chapter-3

3. Requirement Analysis

System analysis basically deals with determination of system requirements. The determination of requirements means studying the existing system and collection details about it to find out what these parameters are.

There are three main activities that are performed in requirement analysis.

□ **Requirement Anticipation:**

It means the study done by the system analyst. In this case system analyst raised the question and applies methods to solve that question based on his/her past experiences.

□ **Requirement Investigation:**

This method is related with finding and investing more features of the system. We have collected information about our system by using fact finding methods like observation and questioning clients.

□ **Requirement Specification:**

A Requirement Specification is an agreement between the developer and the user. The clients specified their requirements to us and we selected a strategy in which he tries to fulfil their requirements.

Chapter 4

4. TECHNOLOGY OVERVIEW

The technology selected for implementing Employee Management System is PHP/MYSQL. Apache is used as the HTTP server. The development was done in a windows environment using adobe Dreamweaver CS5.

PHP

PHP is a general-purpose scripting language that is especially suited to server-side web development where PHP generally runs on a web server code is embedded into the HTML source document. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content. It can also be used for command-line scripting and client-side GUI applications. PHP can be deployed on many web servers and operating systems, and can be used with many relational database management systems (RDBMS). It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.

MySQL

MySQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. MySQL is a popular choice of database for use in web applications and is an open source product. The process of setting up a MySQL database varies from host to host, however we will end up with a database name, a user name and a password. Before using our database, we must create a table. A table is a section of the database for storing related information. In a table we will set up the different fields which will be used in that table. Creating a table in phpMyAdmin is simple, we just type the name, select the number of fields and click the 'go' button. we will then be taken to a setup screen where you must create the fields for the database. Another way of creating databases and tables in phpMyAdmin is by executing simple SQL statements. We have used this method in order to create our database and tables.

Apache

The Apache HTTP Server is a web server software notable for playing a key role in the initial growth of the World Wide Web. In 2009 it became the first web server software to surpass the 100 million web site milestone. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation. Since April 1996 Apache has been the most popular HTTP server software in use. As of November 2010 Apache served over 59.36% of all websites and over 66.56% of the first one million busiest websites.

LAMP (Linux, Apache, MySQL, PHP)

LAMP is an open source Web development platform that uses Linux as the operating system, Apache as the Web server, MySQL as the relational database management system and PHP as the object-oriented scripting language. (Sometimes Perl or Python is used instead of PHP.) Because the platform has four layers, LAMP is sometimes referred to as a LAMP stack. Stacks can be built on

different operating systems. Developers that use these tools with a Windows operating system instead of Linux are said to be using WAMP; with a Macintosh system, MAMP; and with a Solaris system, SAMP.

Installation

Step 1: Install Apache and Allow in Firewall

The Apache web server is among the most popular web servers in the world. It's well-documented, and has been in wide use for much of the history of the web, which makes it a great default choice for hosting a website.

We can install Apache easily using Ubuntu's package manager, apt. A package manager allows us to install most software pain-free from a repository maintained by Ubuntu.

For our purposes, we can get started by typing these commands:

```
sudo apt-get update
sudo apt-get install apache2
```

Since we are using a sudo command, these operations get executed with root privileges. It will ask you for your regular user's password to verify your intentions.

Once you've entered your password, apt will tell you which packages it plans to install and how much extra disk space they'll take up. Press **Y** and hit **Enter** to continue, and the installation will proceed.



Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*/available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. Calling `/usr/bin/apache2` directly will not work with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under `/var/www`. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

If you see this page, then your web server is now correctly installed and accessible through your firewall. Regardless of the method you use to get your IP address, you can type it into your web browser's address bar to get to your server.

Step 2: Install MySQL

Now that we have our web server up and running, it is time to install MySQL. MySQL is a database management system. Basically, it will organize and provide access to databases where our site can store information.

Again, we can use apt to acquire and install our software. This time, we'll also install some other "helper" packages that will assist us in getting our components to communicate with each other:

```
sudo apt-get install mysql-server
```

Note: In this case, you do not have to run `sudo apt-get update` prior to the command. This is because we recently ran it in the commands above to install Apache. The package index on our computer should already be up-to-date.

Again, you will be shown a list of the packages that will be installed, along with the amount of disk space they'll take up. Enter **Y** to continue.

During the installation, your server will ask you to select and confirm a password for the MySQL "root" user. This is an administrative account in MySQL that has increased privileges. Think of it as being similar to the root account for the server itself (the one you are configuring now is a MySQL-specific account, however). Make sure this is a strong, unique password, and do not leave it blank.

Answer **y** for yes, or anything else to continue without enabling.

For the rest of the questions, you should press **Y** and hit the **Enter** key at each prompt. This will remove some anonymous users and the test database, disable remote root logins, and load these new rules so that MySQL immediately respects the changes we have made. At this point, your database system is now set up and we can move on.

Step 3: Install PHP

PHP is the component of our setup that will process code to display dynamic content. It can run scripts, connect to our MySQL databases to get information, and hand the processed content over to our web server to display.

We can once again leverage the apt system to install our components. We're going to include some helper packages as well, so that PHP code can run under the Apache server and talk to our MySQL database:

```
sudo apt-get install php libapache2-mod-php php-mcrypt php-mysql
```

This should install PHP without any problems. We'll test this in a moment.

In most cases, we'll want to modify the way that Apache serves files when a directory is requested. Currently, if a user requests a directory from the server, Apache will first look for a file called `index.html`. We want to tell our web server to prefer PHP files, so we'll make Apache look for an `index.php` file first.

At this point, your LAMP stack is installed and configured. We should still test out our PHP though.

Step 4: Test PHP Processing on your Web Server

In order to test that our system is configured properly for PHP, we can create a very basic PHP script.

We will call this script `info.php`. In order for Apache to find the file and serve it correctly, it must be saved to a very specific directory, which is called the "web root".

In Ubuntu 16.04, this directory is located at `/var/www/html/`. We can create the file at that location by typing:

```
sudo nano /var/www/html/info.php
```

This will open a blank file. We want to put the following text, which is valid PHP code, inside the file:

```
info.php
<?php
phpinfo();
?>
```



When you are finished, save and close the file.

Now we can test whether our web server can correctly display content generated by a PHP script. To try this out, we just have to visit this page in our web browser. You'll need your server's public IP address again.

The address you want to visit will be:

```
http://your_server_IP_address/info.php
```

The page that you come to should look something like this:

PHP Version 7.0.4-7ubuntu1	
	
System	Linux ubuntu-16-lamp 4.4.0-12-generic #28-Ubuntu SMP Wed Mar 9 00:33:55 UTC 2016 x86_64
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.0/apache2/conf.d/10-opcache.ini, /etc/php/7.0/apache2/conf.d/10-pdo.ini, /etc/php/7.0/apache2/conf.d/20-calendar.ini, /etc/php/7.0/apache2/conf.d/20-ctype.ini, /etc/php/7.0/apache2/conf.d/20-exif.ini, /etc/php/7.0/apache2/conf.d/20-fileinfo.ini, /etc/php/7.0/apache2/conf.d/20-ftp.ini, /etc/php/7.0/apache2/conf.d/20-gettext.ini, /etc/php/7.0/apache2/conf.d/20-iconv.ini, /etc/php/7.0/apache2/conf.d/20-json.ini, /etc/php/7.0/apache2/conf.d/20-mcrypt.ini, /etc/php/7.0/apache2/conf.d/20-mysql.ini, /etc/php/7.0/apache2/conf.d/20-phar.ini, /etc/php/7.0/apache2/conf.d/20-posix.ini, /etc/php/7.0/apache2/conf.d/20-readline.ini, /etc/php/7.0/apache2/conf.d/20-shmop.ini, /etc/php/7.0/apache2/conf.d/20-sockets.ini, /etc/php/7.0/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.0/apache2/conf.d/20-sysvsem.ini, /etc/php/7.0/apache2/conf.d/20-sysvshm.ini, /etc/php/7.0/apache2/conf.d/20-tokenizer.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012.NTS
PHP Extension Build	API20151012.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	enabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*, mcrypt.*, mdecrypt.*
This program makes use of the Zend Scripting Language Engine: Zend Engine v3.0.0, Copyright (c) 1998-2016 Zend Technologies with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2016, by Zend Technologies	
	

This page basically gives you information about your server from the perspective of PHP. It is useful for debugging and to ensure that your settings are being applied correctly.

If this was successful, then your PHP is working as expected.

You probably want to remove this file after this test because it could actually give information about your server to unauthorized users. To do this, you can type this:

```
sudo rm /var/www/html/info.php
```

You can always recreate this page if you need to access the information again later.

Conclusion

Now that you have a LAMP stack installed, you have many choices for what to do next. Basically, you've installed a platform that will allow you to install most kinds of websites and web software on your server.

SOFTWARE LIFE CYCLE ADOPTED:-

In order to develop the project “Student Planner” we have adopted the iterative enhancement model. This model removes the shortcoming of waterfall model. Since many facts of this system are already known. It is not a new concept and hence no research is required.

A working version can be easily created and hence the system can start working. Rest of the functionalities can be implemented in the next iteration and can be delivered later. As the

requirement analysis is also not required. It not being a new technology risk involved is also less. So, one need not perform detailed risk analysis.

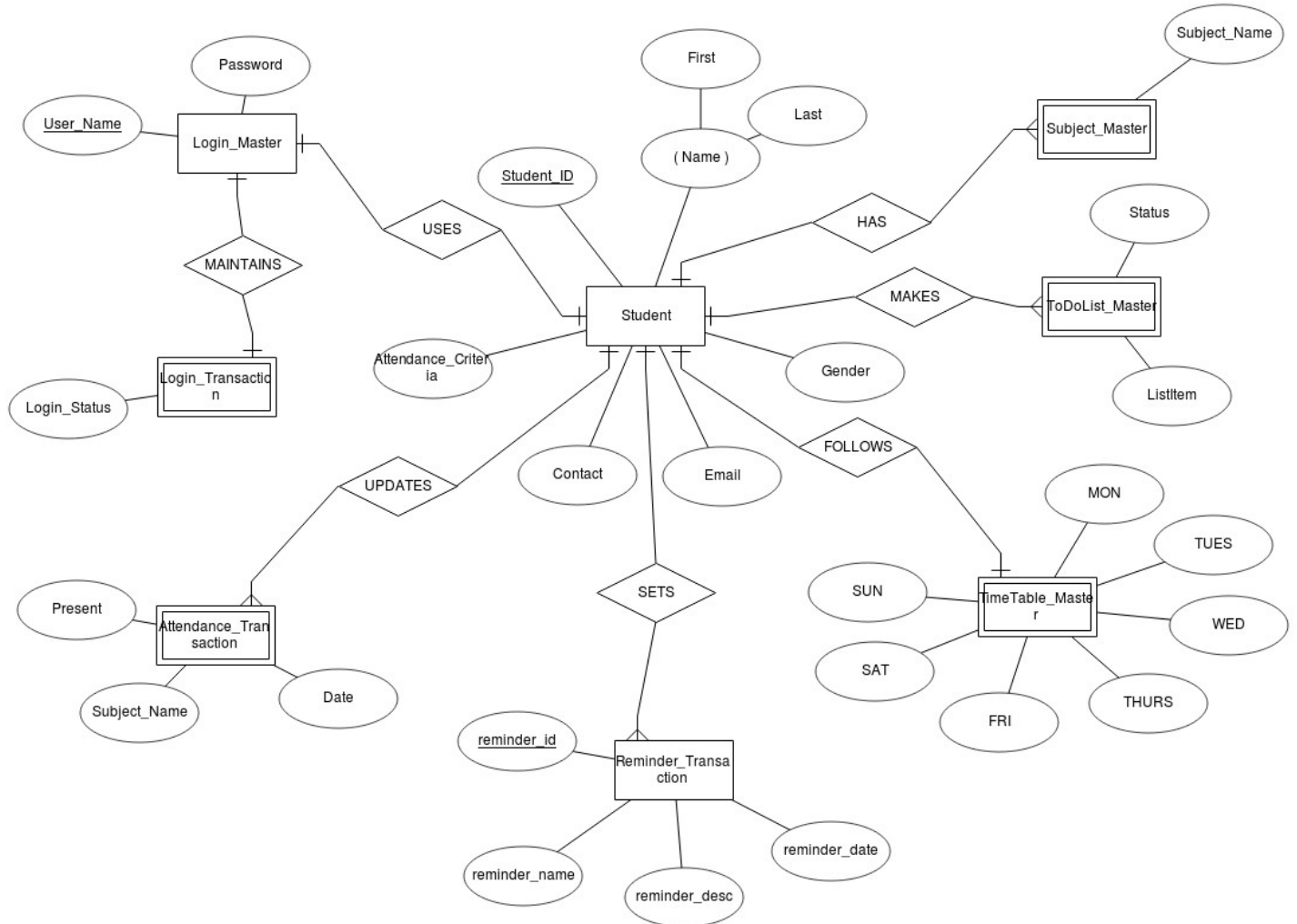
If redevelopment staff is less than development can be started with less number of people and in next increments others can be involved. As this model combines the advantage of waterfall model and prototyping, clients are always aware of the product being delivered and can always suggest changes and enhancements and can get them implemented.

As less amount of customer communication is required one need not apply spiral model in which all types of analysis is done in detail. As the deadline is affordable one need not to for Rapid Application Development model. Iterative enhancement model is useful when less manpower is available for software development and the release deadlines are specified.

It is best suited for in house product development, where it is ensured that the user has something to start with. The complete product is divided into releases and the developer delivers the product release by release.

Chapter-5

5. Conceptual and Logical Designs



Chapter-6

6. Database Normalization

First Normal Form (1NF)

As per **First Normal Form**, no two Rows of data must contain repeating group of information i.e each set of column must have a unique value, such that multiple columns cannot be used to fetch the same row. Each table should be organized into rows, and each row should have a primary key that distinguishes it as unique.

The **Primary key** is usually a single column, but sometimes more than one column can be combined to create a single primary key. For example consider a table which is not in First normal form

Student Table :

Student	Age	Subject
Adam	15	Biology, Maths
Alex	14	Maths
Stuart	17	Maths

In First Normal Form, any row must not have a column in which more than one value is saved, like separated with commas. Rather than that, we must separate such data into multiple rows.

Student Table following 1NF will be :

Student	Age	Subject
Adam	15	Biology
Adam	15	Maths
Alex	14	Maths
Stuart	17	Maths

Using the First Normal Form, data redundancy increases, as there will be many columns with same data in multiple rows but each row as a whole will be unique.

Second Normal Form (2NF)

As per the Second Normal Form there must not be any partial dependency of any column on primary key. It means that for a table that has concatenated primary key, each column in the table that is not part of the primary key must depend upon the entire concatenated key for its existence. If any column depends only on one part of the concatenated key, then the table fails **Second normal form**.

In example of First Normal Form there are two rows for Adam, to include multiple subjects that he has opted for. While this is searchable, and follows First normal form, it is an inefficient use of space.

Also in the above Table in First Normal Form, while the candidate key is {**Student, Subject**}, **Age** of Student only depends on Student column, which is incorrect as per Second Normal Form. To achieve second normal form, it would be helpful to split out the subjects into an independent table, and match them up using the student names as foreign keys.

New Student Table following 2NF will be :

Student	Age
Adam	15
Alex	14
Stuart	17

In Student Table the candidate key will be **Student** column, because all other column i.e **Age** is dependent on it.

New Subject Table introduced for 2NF will be :

Student	Subject
Adam	Biology
Adam	Maths
Alex	Maths
Stuart	Maths

In Subject Table the candidate key will be {**Student, Subject**} column. Now, both the above tables qualifies for Second Normal Form and will never suffer from Update Anomalies. Although there are a few complex cases in which table in Second Normal Form suffers Update Anomalies, and to handle those scenarios Third Normal Form is there.

Chapter-7

7. Hardware and Software Requirements

6.1 Hardware Requirements:

- Processor: Pentium III and above
- Memory(RAM): 500MB RAM(minimum)
- Input Devices: Keyboard and Mouse
- Hard Disk: Minimum 20GB

6.2 Software Requirements:

- Ubuntu 14 and above
- Apache2 Server
- MySQL Server
- PHP

Chapter-8

8. Graphical User Interface

This chapter documents the current user interface and the various elements needed to fulfil the user requirements. The images of the user interface pages are included to demonstrate the application's look and feel.

Description of Complete User Interface Specification:

The Login page:

The main screen will contain the login page (See Figure). The login page consists of Username, password and the login button. It consists of two text boxes. There is also a submit button with the value login.

The Home page/Attendance Page:

This is the main page of the System. Here the user will be able to display blocks of all the subject he has provided. Inside each block, the attendance percentage for that subject is displayed along with total number of lectures conducted and attended.

There is a button to update attendance which will take user to the page to update the attendance.

Update Attendance Page:

On this page, the user will first select the day whose attendance is to be modified.

On clicking find day, the timetable of that day is displayed.

The user has to select the radio button to update the attendance.

Here the user will also find out if any previous attendance has been marked for the selected day.

On clicking done, the attendance will be calculated and modified. Display is done again on Homepage.

Report Page:

Here the report generation is done.

4 different kinds of reports can be generated as per the click of respective button.

These reports change as soon as there is a modification in attendance.

Login

localhost/SMS/login.php

Search

Username

Username

Password

Login

Forgot password

Sign Up

MyAccount

localhost/SMS/userhome.php

Search

Attendance Attendance Report To Do List Reminders Vedika Barde | Sign Out

Update Attendance

Total : 5 DBMS Attended : 3

Your Current Attendance : 60.00

Total : 5 HCI Attended : 3

Your Current Attendance : 60.00

Timetable

localhost/SMS/update_att.php

Search

Attendance

Attendance Report

To Do List

Reminders

Vedika Barde | Sign Out

Update Your Attendance Here

DATE TO BE UPDATED

8

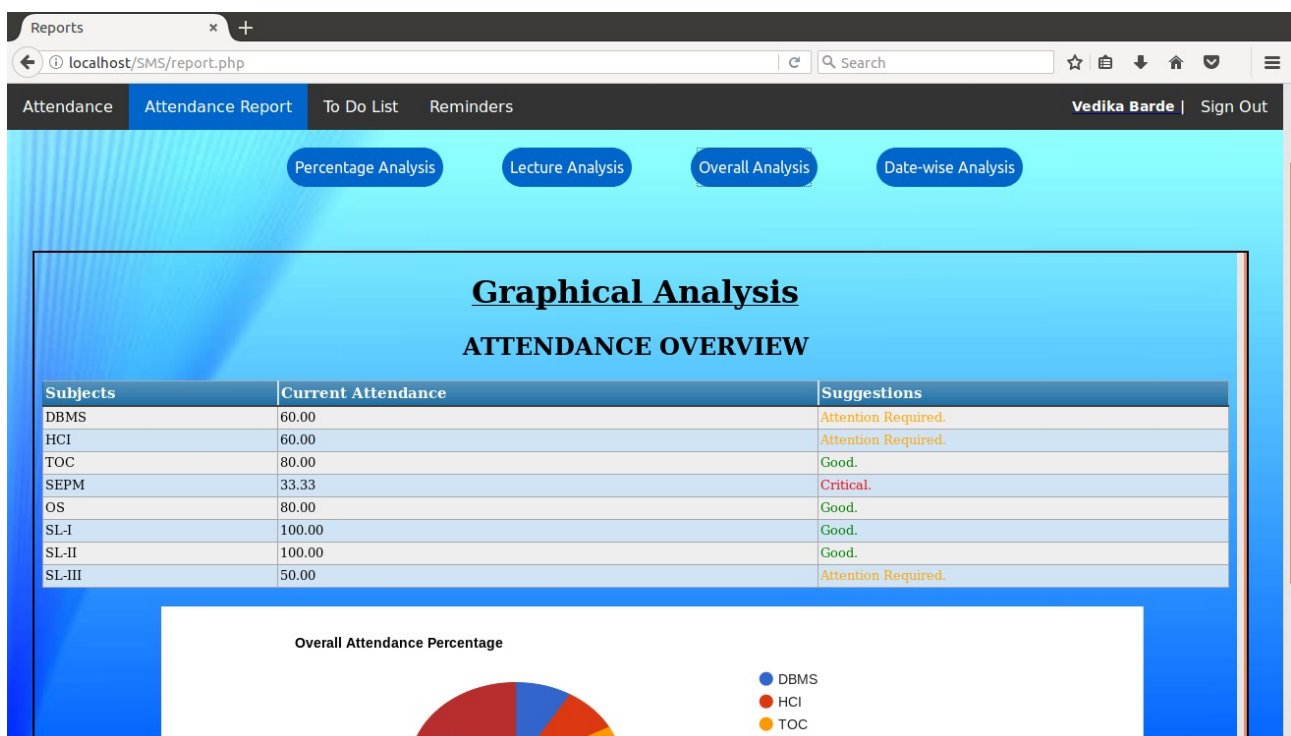
SEPT

2017

Find Day

Timetable on Friday 08-09-2017

Subject Name	Present	Absent	Lecture Off
HCI	<div></div>	<div></div>	<div></div>



9. Source Code

signUpPage.php

```
<?php
// define variables and set to empty values

$firstNameErr = $lastNameErr = $genderErr = $usernameErr =
$passwordErr = $confirmPassErr = $emailErr = $contactErr =
"";
$firstName = $lastName = $gender = $username = $password =
$confirmPass = $email = $contact = "";

if ($_SERVER["REQUEST_METHOD"] == "POST")
{

    if (empty($_POST["FN"]))
    {
        $firstNameErr = "First Name Required.";
    }

    else
    {
        $firstName = test_input($_POST["FN"]);

        // check if name only contains letters and whitespace
        if (!preg_match("/^[a-zA-Z]*$/", $firstName))
        {
            $firstNameErr = "Only Alphabets And White Spaces
Allowed.";
        }

    }

    if (empty($_POST["LN"]))
    {
        $lastNameErr = "Last Name Required.";
    }

    else
    {
        $lastName = test_input($_POST["LN"]);

        // check if name only contains letters and whitespace
        if (!preg_match("/^[a-zA-Z]*$/", $lastName))
        {
```

```

        $lastNameErr = "Only Alphabets And White Spaces
Allowed.";
    }

}

if (empty($_POST["Gender"]))
{
    $genderErr = "Gender Field Required.";
}

else
{
    $gender = test_input($_POST["Gender"]);
}

if (empty($_POST["UN"]))
{
    $usernameErr = "Username Required.";
}

else
{
    $username = test_input($_POST["UN"]);

    // check if name only contains letters and whitespace
    if (!preg_match("/^[a-zA-Z0-9!@#$$%^&*.]*/", $username))
    {
        $usernameErr = "Only Alphabets, Special Symbols And
Numbers Allowed.";
    }

}

if (empty($_POST["PSWD"]))
{
    $passwordErr = "Password Required.";
}

else
{
    $password = $_POST["PSWD"];
}

if (empty($_POST["CPSWD"]))

```

```

{
    $confirmPassErr = "Confirm Your Password.";
}

else
{
    $confirmPass = $_POST["CPSWD"];

    if($confirmPass != $password)
    {
        $confirmPassErr = "Passwords Don't Match !";
    }
}

if (empty($_POST["EMAIL"]))
{
    $emailErr = "E-mail Field Required.";
}

else
{
    $email = test_input($_POST["EMAIL"]);

    // check if e-mail address is well-formed
    if (!filter_var($email, FILTER_VALIDATE_EMAIL))
    {
        $emailErr = "Invalid E-mail.";
    }
}

if (empty($_POST["CON"]))
{
    $contactErr = "Contact Field Required.";
}

else
{
    $contact = test_input($_POST["CON"]);

    // check if contact number is well-formed
    if (!preg_match('/^[0-9]{10}+$/ ', $contact))
    {
        $contactErr = "Invalid Contact Number. (10-Digits
Only)";
    }
}

```



```

    }

}

function test_input($data)
{
    $data = trim($data);           //Trim extra white-
spaces, lines etc.
    $data = stripslashes($data);   //Strips off backslashes
in the input
    $data = htmlspecialchars($data); //Converts special chars
into HTML entities for security reasons
    return $data;
}

?>

<html>

<head>
    <link rel="stylesheet" href="SignUp_style.css">
    <title>Sign Up</title>
</head>

<body>

    <br><br>
    <div class="signupbox">
    
    <h2 style = "text-align:center"><b>Sign Up</b></h2>
    <br>
    <form method="post" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]);?>" >

        <label>FIRST NAME </label> &nbsp; <span
class="error">* <?php echo $firstNameErr; ?></span><br>
        <input type="text" name="FN" value="<?php echo
$firstName; ?>" placeholder=" First Name">
        <br><br>

        <label>LAST NAME </label> &nbsp; <span
class="error">* <?php echo $lastNameErr; ?></span><br>

```

```

        <input type="text" name="LN" value="<?php echo
$lastName; ?>" placeholder=" Last Name">
        <br><br>

        <label> GENDER : </label> &nbsp; <span
class="error">* <?php echo $genderErr; ?></span><br>
        <input type="radio" name="Gender" <?php if
(isset($gender) && $gender=="M") echo "checked";?>
value="M">MALE <br>
        <input type="radio" name="Gender" <?php if
(isset($gender) && $gender=="F") echo "checked";?>
value="F">FEMALE
        <br><br>

        <label> USER NAME </label> &nbsp; <span
class="error">* <?php echo $usernameErr; ?></span><br>
        <input type="text" name="UN" value="<?php echo
$username; ?>" placeholder=" Username">
        <br><br>

        <label> PASSWORD : </label> &nbsp; <span
class="error">* <?php echo $passwordErr; ?></span><br>
        <input type="password" name="PSWD" value="<?php
echo $password; ?>" placeholder="*****">
        <br><br>

        <label> CONFIRM<br>PASSWORD : </label> &nbsp;
<span class="error">* <?php echo $confirmPassErr;
?></span><br>
        <input type="password" name="CPSWD" value="<?
php echo $confirmPass; ?>" placeholder="*****">
        <br><br>

        <label> EMAIL ID : </label> &nbsp; <span
class="error">* <?php echo $emailErr; ?></span><br>
        <input type="text" name="EMAIL" value="<?php
echo $email; ?>" placeholder=" abc@xyz.com">
        <br><br>

        <label> CONTACT : </label> &nbsp; <span
class="error">* <?php echo $contactErr; ?></span><br>
        <input type="text" name="CON" value="<?php echo
$contact; ?>" placeholder="10-Digit Contact Number">
        <br><br>

```

```
                <input type="submit" name="Next-pg"
value="Next">
                <input type="reset" name="Reset-All"
value="Reset">
```

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

```
</body>
</html>
```

```
<?php
```

```
    if($firstNameErr == "" && $lastNameErr == "" && $genderErr ==
"" && $usernameErr == "" && $passwordErr == "" &&
$confirmPassErr == "" && $emailErr == "" && $contactErr == ""
&& $_SERVER["REQUEST_METHOD"] == "POST")
    {
        $db = mysqli_connect('localhost','root','root','SMS')
        or die('Error connecting to MySQL server.');
```

```
        echo $row[0];

        if(isset($_POST['Next-pg']))
        {
            $insertQuery1 = ("INSERT INTO
Student_Master(First_Name,Last_Name,Gender,Email,Contact,Atten
dance_Criteria) VALUES('".$_POST["FN"]."', '".$_
$_POST["LN"]."', '".$_POST["Gender"]."', '".$_
$_POST["EMAIL"]."', '".$_POST["CON"]."', '0')");
            mysqli_query($db, $insertQuery1);

            $query = "SELECT MAX(Student_ID) from Student_Master";
            $result = mysqli_query($db,$query);
            $row = mysqli_fetch_array($result);

            $insertQuery2 = ("INSERT INTO
Login_Master(User_Name>Password,StudentID) VALUES('".$_
$_POST["UN"]."', '".$_POST["PSWD"]."', $row[0])");
            mysqli_query($db, $insertQuery2);

        }

        mysqli_close($db);
        header("location:subject_page.php");
```

}
?>

Chapter-10

10. Testing

10.1 Introduction:

The Testing phase forms an important part of the software development life cycle. Any software product has to be tested thoroughly before it is delivered to the end customer. Well tested software with limited features is certainly better than the one having many features with only a few of them working. This document provides a general overview of the testing strategy adopted for testing our product.

10.2 Goals and Objectives:

The software testing involves verification and validation of the software produced. Testing is a process of executing of program with the intent of finding an error. A good test is one where there is high probability of finding an error. A successful test is one which uncovers and as yet undiscovered errors. Our objective of testing is systematically uncovering different classes of errors and to do so with minimum efforts and amount of time and effort. The data collected in the test provides a good indication of the software reliability and some indication of software quality as a whole. The results of testing will not only help to know which parts of system are working below average but also helps to make the system more user friendly. Testing is considered as an unavoidable part of any responsible effort to develop a software system.

10.3 Test Plan:

Testing Strategy

A good testing strategy is one using which a lot of errors can be easily found. The testing which is to be carried out is divided into number of modules for a proper judgment of the quality of the software. The testing strategy mainly carried out was Module Testing. Test cases are plotted considering the above categories and correct functionality of various parts of code is ensured.

Unit Testing

This involves testing of individual modules. Here we have tested Individual modules written for various operations,

Module for Sign Up

Module for Login

Module for Home Page

Module for Updating Attendance

Module for Reports Generation

Module for Reminders.

Module for To Do List.

Integration Testing

The system as a whole is tested here. The system is said to be operating correctly if it passes these tests. After the different modules have been developed and tackled, all the modules are integrated and tested during the Integration testing.

1. Sign Up Form - User Details - Test 1

Test Case Summary	Filling of all the necessary fields in Sign Up Form. Required fields are specified by * symbol.
Prerequisites	No prerequisites as new user will be using this form.
Test Data	Skipping required fields.
Expected Result	Error message should be displayed in front of the field.
Actual Result	Error message has been displayed in front of fields.
Status	PASS

2. Sign Up Form - User Details - Test 2

Test Case Summary	Validations on various fields: 1.No digits allowed in First, Last Name. 2.Password and Confirm Password fields should match. 3.Valid format in email address field. 4.Contact Number of 10 digits only.
Prerequisites	No prerequisites as new user will be using this form.
Test Data	Input of invalid data w.r.t. above requirements.
Expected Result	Error message should be displayed according to the invalid input type.

Actual Result	Error message has been displayed in front of fields.
Status	PASS

3. Sign Up Form- Accept Subjects - Test 1

Test Case Summary	The 'Add Subject' button should not work once the limit on number of subject has reached.
Prerequisites	No prerequisites as new user will be using this form.
Test Data	Trying to add subject after limit on number of subject has reached.
Expected Result	Add button should get disabled
Actual Result	Add button got Disabled
Status	PASS

4. Sign Up Form - Timetable page - Test 1

Test Case Summary	List of subjects has to be according to the entered Subjects
Prerequisites	No prerequisites as new user will be using this form.
Test Data	Checking the subject List
Expected Result	List should get displayed
Actual Result	List has been displayed
Status	PASS

5. Login Page - Test 1

Test Case Summary	Filling of User name, Password fields in Login Form. Required fields are specified by * symbol.
Prerequisites	Authentic User
Test Data	Skipping required fields.
Expected Result	Error message should be displayed in front of the field.
Actual Result	Error message has been displayed in front of fields.
Status	PASS

6. Login Page - Test 2

Test Case Summary	Filling of correct User name and Password fields in Login Form.
Prerequisites	Authentic User

Test Data	Wrong User name or Password
Expected Result	Error message should be displayed according to the invalid input type.
Actual Result	Error message has been displayed in front of fields.
Status	PASS

7. User Home page

Test Case Summary	List of subjects blocks has to be according to the entered Subjects
Prerequisites	Authentic User
Test Data	Checking the subject List
Expected Result	List should get displayed
Actual Result	List has been displayed
Status	PASS

8. User Home page - Test 2

Test Case Summary	To check the display of correct attendance percentage for a subject when the attendance is updated
Prerequisites	Authentic User
Test Data	Updating of attendance of any Subject of user
Expected Result	Correct Attendance should be displayed
Actual Result	Correct Attendance has been displayed
Status	PASS

9. Updating/Calculation of attendance

Test Case Summary	To check the correct calculation of attendance via backend procedures on Updating of attendance.
Prerequisites	Authentic User
Test Data	Updating of attendance of any Subject of user
Expected Result	Correct Attendance should be calculated
Actual Result	Correct Attendance has been calculated.
Status	PASS

10. Report Generation

Test Case Summary	To check the generation of reports as soon the attendance is updated.
Prerequisites	Authentic User
Test Data	Updating of attendance of any Subject of user
Expected Result	Correct report should be generated as per the modification in attendance.
Actual Result	Correct report has been generated.
Status	PASS

11. Reminders - Test 1

Test Case Summary	Filling of all the necessary fields. Required fields are specified by * symbol.
Prerequisites	Authentic User
Test Data	Skipping required fields.
Expected Result	Error message should be displayed in front of the field.
Actual Result	Error message has been displayed in front of fields.
Status	PASS

12. Reminders - Test 2

Test Case Summary	Setting the alert date for reminder as a past date w.r.t. current date.
Prerequisites	Authentic User
Test Data	Past date w.r.t. today
Expected Result	Error message should be displayed
Actual Result	Error message has been displayed
Status	PASS

13. To Do List -

Test Case Summary	Filling of all the necessary fields. Required fields are specified by * symbol.
Prerequisites	Authentic User
Test Data	Skipping required fields.
Expected Result	Error message should be displayed in front of the field.
Actual Result	Error message has been displayed in front of fields.
Status	PASS

Chapter-11

11. Future Enhancement

The following Future Enhancements can be provided to the User:

1. Multiple options for Sign Up

The User will have a choice to sign up using Google/Facebook accounts. In this case, the timetable and subjects will be taken in a later stage.

2. Allow upadting attendance of multiple days

The user can update attendance of a week/month without actually going through every day one by one.

3. Sending out Email/SMS for the following

1. Attendance suggestion for subject whose attendance is critical.
2. Reminders which have nearby alert date.
3. Reports which user may request.
4. Linking of To Do List with Reminders. The To Do list task can be linked with the date for reminders.

4. Improvement in UI and Performance:

This will facilitate user satisfaction and liking for our system.

5. Documentation of Reports:

Feature to download the reports in required formats i.e. PDF, DOC etc.

Chapter-12

12. Conclusion

As, considering our system, we conclude that using our system, we can maintain the attendance for the subjects provided by the user. The instantaneous updating and calculation in attendance is done for the user. The system can be effectively used by students or professionals with a need to maintain their personal attendance in their field.

When our system starts, it performs so many different types of tasks and operations accessing the database defined by the creator and various database operations easily and effectively reflect it.

Chapter-13

13. References

BOOKS

- Introduction to HTML, PHP, JavaScript
- Cookbook of MySQL
- Software Engineering by Roger Pressman

WEBSITES

- www.w3schools.com
- www.stackoverflow.com
- www.mysql.com