**A Project Report on**

**“ATTENDANCE MANAGEMENT SYSTEM”**

**Submitted in partial fulfilment of the requirement for**

**Degree in Bachelor of Engineering (Information Technology)**

**By**

Rony Benny (501750)

Nevin Joseph (501751)

Rachel Angelin (501745)

**Guided by:**

**Prof. Kalpana Wani**



**Department of Information Technology**

**Fr. Conceicao Rodrigues Institute of Technology**

Sector 9A, Vashi, Navi Mumbai – 400703

**University of Mumbai**

**2019-2020**

**CERTIFICATE**

This is to certify that the project entitled

**ATTENDANCE MANAGEMENT SYSTEM**

**Submitted By**

Rony Benny

Nevin Joseph

Rachel Angelin

In partial fulfilment of degree of T.E in **Information Technology** for term work of the project is approved.

**External Examiner Internal Examiner**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Internal Guide**

**Date: -**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Head of Department**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ABSTRACT**

Education is the sole basis on which future of the human race depends. Our quest for knowledge is unending and will continue till planet Earth exists. A man without education is like a building without foundation. There are many reasons why people need education. It helps them to learn new things, find good jobs and to lead a respectable life in the society.

Now in order to receive this education we have to be a part of an institution or a university. So we need to follow all the rules and regulation which is given by the university so that we might to do things in a proper order and receive quality education. One of the compulsory criteria which almost all educational institutes follow is 75% minimum attendance that has to be maintained by the students.

So our website is an online system that students can use in order to know the status of their attendance. The students have to signup first, and then add the subjects for which the attendance has to be calculated and on regular basis update it so that at the end of the week they get overall percentage of how much they have attended.

I

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Topic** | **Page No.** |
| 1  1.1  1.2  1.3  1.4  1.5 | **Introduction**  Problem Statement  ea Project Objective  Project Scope  Advantages  Disadvantages | **1**  2  2  2  3  3 |
| 2  2.1  2.2  2.3 | **Project Plan**  Project Plan  Task Sheet  Project Roles and Responsibility | **4**  5  6  8 |
| 3  3.1  3.2  3.3  3.4  3.5 | **Requirement Analysis**  Functional Requirements  Non Functional Requirements  Hardware Requirements  Software Requirement  System Features | **9**  10  10  11  11  11 |
| 4  4.1  4.2  4.3 | **Design Document**  Design view of web pages’ layout  Code design for Responsive web design  RIA design using Ajax | **12**  13  16  17 |
| 5  5.1  5.2 | **Database Design and Connectivity**  Database Table Design  Database Connectivity | **18**  19  20 |
| 6  6.1 | **Implementation Details**  Snapshots | **21**  22 |
| 7  7.1  7.2 | **Testing**  HTML form validation  website security testing | **27**  28  31 |
| 8 | **Website Launching** | **32** |
| 9 | **Conclusion and Future Scope** | **33** |
| 10 | **Appendix I: Code Sample** | **36** |

III

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Name of the figure** | **Page No.** |
| 4.1.1 | Main page | 13 |
| 4.1.2 | Login and Registration page | 14 |
| 4.1.3 | Buyer Page | 15 |
| 4.1.4 | Product description Page | 15 |
| 4.1.5 | Billing page | 16 |
| 6.1.1 | Homepage | 22 |
| 6.1.2 | Login Page | 23 |
| 6.1.3 | Registration Page | 23 |
| 6.1.4 | Products Page | 24 |
| 6.1.5 | Product Description Page | 24 |
| 6.1.6 | Billing Page | 25 |
| 6.1.7 | Admin Product Insertion Page | 26 |
| 6.1.8 | Iframe | 26 |

IV

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Name of the table** | **Page No.** |
| 2.1.1 | Project Plan | 5 |
| 2.2.1 | Task Sheet | 6 |
| 2.3.1 | Roles And Responsibility | 8 |
| 5.1.1 | Students SignUp Details | 19 |
| 5.1.2 | Subjects Table | 19 |
| 5.1.3 | Time Table | 19 |
| 5.1.4 | Login Table | 19 |
| 5.1.5 | Attendance View Table | 19 |

V

**Chapter 1**

**INTRODUCTION**

**1.1 Problem Statement**

As there is a lot of transition happening in the education system of our society because of the remarkable progress in the field of science and technology. So the requirement of quality education has increased so much that in order to find a well-paying job, higher education must be an important factor in a student’s life. Therefore, educational institutes and university has made the curriculum pattern in such a way that students along with education have an overall development. So attending lectures regularly especially in colleges is very important, which most of the students find it difficult to sit for long hours.

So in order to fulfill the minimum criteria students have to attend the lectures regularly, and to keep a track of overall percentage of the lectures they have attended weekly or monthly is not possible. It becomes very difficult to calculate manually, for each and every subject and everyday how much they have attended.

**1.2 Project Objective**

* To produce a web-based system that allows students to calculate the overall percentage of the lectures they have attended in college on weekly or monthly basis.
* To make the students aware of the number of lectures attended subject wise.

**1.3 Project Scope**

* Since the basic idea behind the project is to manage the attendance online we provide a signup page for the students to register. So that they can use it as a registered user.
* It includes a login page in which user has to login through the username and password which he/she had given during signup to keep a track of each user which uses the website, ensuring authentication, confidentiality and security to user.
* Also, it includes a dashboard where user can add the subjects for which the attendance has to be calculated. After adding we get overall analysis of each subject as we update it every day.
* We also provide the user with the percentage of the lectures attended and how much more lectures they have to attend to fulfill the criteria.

**1.4 Advantages**

1. Reliable and Accurate: By using traditional method for calculating attendance there are chances of making mistake, but since here we already provide the system with all the procedures to calculate, we just have to give the input and the other calculations are done correct and accurately.

2. Improved Efficiency: Keeping the track of the attendance will also help the students to get an idea about their regularity in attending lectures and how much more they have to improve themselves in becoming more punctual and regular.

3. Convenient: It is an advanced system that can be used by the students to analyse their attendance very easily and in a convenient way.

**1.5 Disadvantages**

This project can be used only by students. Because for this we are giving input from only one login we don't have a separate login for the teachers to keep the track of student’s attendance.

**Chapter 2**

**PROJECT PLAN**

**2.1 Table for Project Plan**

**Table 2.1.1 Project Plan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.No** | **Task Name** | **Duration to Complete** | **Start Date** | **End Date** | **Resource Name** |
| 1 | Selection of project Topic | 5 days | 11/7/19 | 15/7/19 | Internet, Group Discussion |
| 2 | Requirement Analysis | 7 days | 18/7/19 | 24/7/19 | Internet, Group discussion |
| 3 | Further Research on the selected topic to check feasibility | 4 days | 25/7/19 | 28/7/19 | Existing similar websites |
| 4 | Role division | 3 days | 1/8/19 | 3/8/19 | Group Discussion |
| 5 | Acquisition of software | 2 days | 4/8/19 | 5/8/19 | Laptops,Internet,Executable files |
| 6 | Detailed design of project-database, webpage layouts | 4 days | 5/8/17 | 8/8/19 | Group Discussion, software utilities, internet |
| 7 | Implementation-coding | 34 days | 8/8/19 | 11/9/19 | Group discussion, software utilities, reference book, internet |
| 8 | Testing | 7 days | 12/9/19 | 18/9/19 | Software utilities |
| 9 | Deployment | 5 days | 19/9/19 | 23/9/19 | Project |

5

**2.2 Table for Task Sheet**

**Table 2.2.1 Task Sheet**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.No** | **Task Name** | **Sub Task** | **Start Date** | **End Date** | **Person Allotted** |
| 1. | Introduction and research on Project related technologies | Understanding and learning HTML, CSS and JAVASCRIPT | 15/7/19 | 29/7/19 | Rachel  Rony  Nevin |
| 2. | Acquisition of software | Review of software versions, Downloads from internet, Distribution of software among project members. | 4/8/19 | 5/8/19 | Rachel  Rony  Nevin |
| 3. | Decisions on design of the project | -Finalizing the web layout  -Finalizing the database tables | 5/8/19 | 8/8/19 | Rony  Nevin |
| 4. | Designing the layout | Includes homepage and various user interactive and responsive web pages | 1/8/19 | 9/8/19 | Rachel  Rony |
| 5. | Coding | Coding various user interface events | 8/8/19 | 4/9/19 | Rachel  Rony  Nevin |
| 6. | Testing and Security | Development of test cases Validation and verification. Includes providing security against SQL injection | 12/9/19 | 18/9/19 | Rony  Nevin |
| 7. | Deployment of application on server | Publishing project on server | 19/9/19 | 23/9/19 | Rony  Nevin |
| 8. | Documentation | Test plan and Test cases, Final Report Submission | 20/7/19 | 26/9/19 | Rachel  Rony  Nevin |

**2.3. Project Roles and Responsibility**

**Table 2.3.1**

|  |  |  |
| --- | --- | --- |
| **Role** | **Responsibility** | **Assigned Person** |
| Full Stack Developer | 1.Homepage(CSS/JavaScript)  2.Login/Registration Page  3.Subject input Page | Rachel |
| Full Stack Developer | 1. Homepage(HTML)  2.Sign-up Page  3.Login/Registration database | Rony |
| Front End Developer | 1. Subject Analysis Page  3. Dashboard Page | Nevin |

**Chapter 3**

**REQUIREMENT ANALYSIS**

**3.1. Functional Requirements**

A software requirements definition is an abstract description of the services which the system should provide and the constraints under which the system must operate.

These requirements are offered in the following points:

* **Login**- We have login for students who want to keep track of their attendance from our website. Here we use MySQL for storing students’ personal details into the database.
* **Registration-** For first time users, we provide with the functionality of registration with us, so that it becomes a hassle free process for the students to monitor their attendance.
* **Data Retrieval**- We retrieve attendance details directly from database.
* **Analysis-** We provide the analysis of the attendance for each subjects in the form of charts.

**3.2. Non-Functional Requirements:**

* **GUI-** For an interactive use of the website, we provide a navigation bar to the user so that he/she can easily access various website sections at a glance.
* **Authentication-** We use login/registration process so that only verified customers are only able to login.
* **Confidentiality:** We use email verification so that no personal details are leaked.

**3.3. Hardware Requirements**

* 4 Gb RAM
* 64-bit OS
* 1 TB ROM
* intel i3 core processor.

**3.4. Software Requirements**

* Operating System: Windows 7 or higher
* Front end: html5, css3
* Database: MySQL
* Tools used: Xampp

**3.5. System Features**

* Allows user to monitor their attendance for each subjects.
* Student details are saved in database to fasten the process of logging in.

**Chapter 4**

**DESIGN DOCUMENT**

**4.1 Design view of web pages’ layout**

**4.1.1. Main page:**

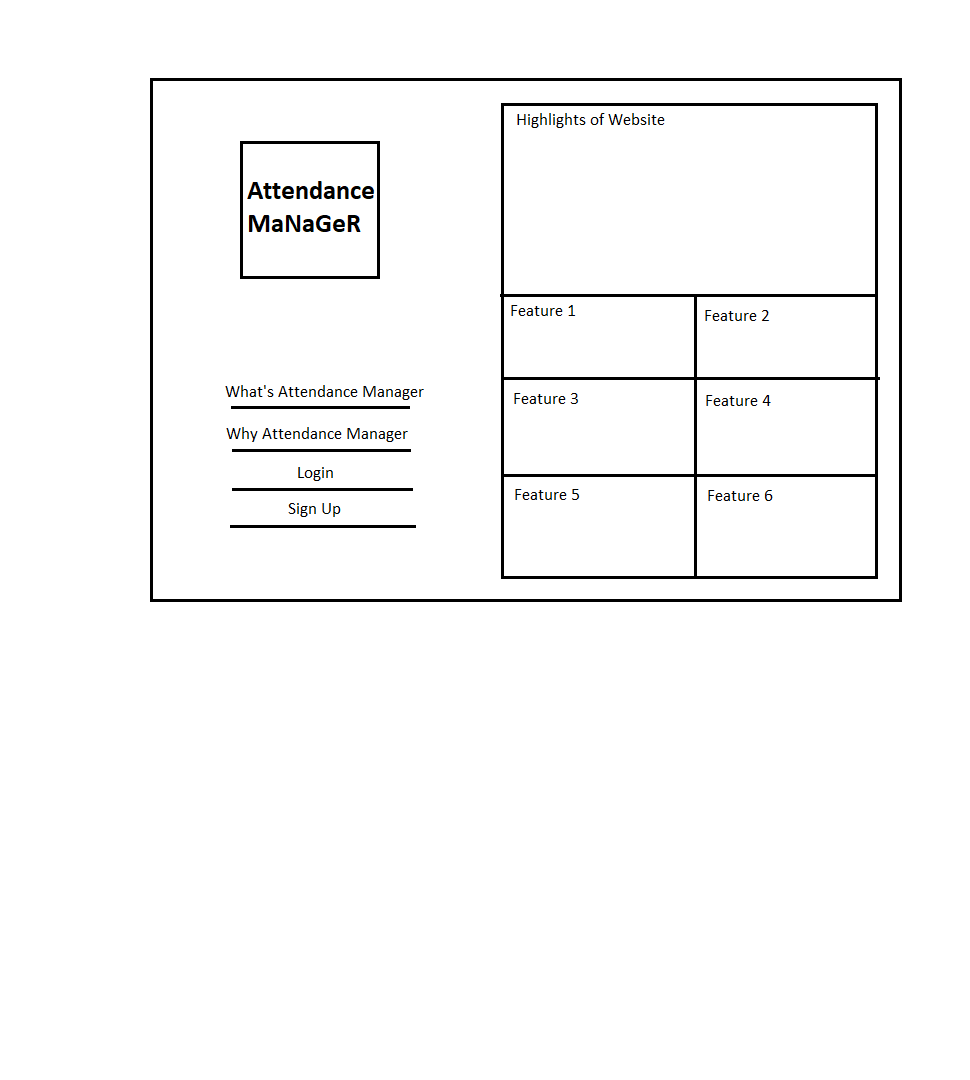


Fig 4.1.1 Main page

**4.1.2 Login and Registration Page**:

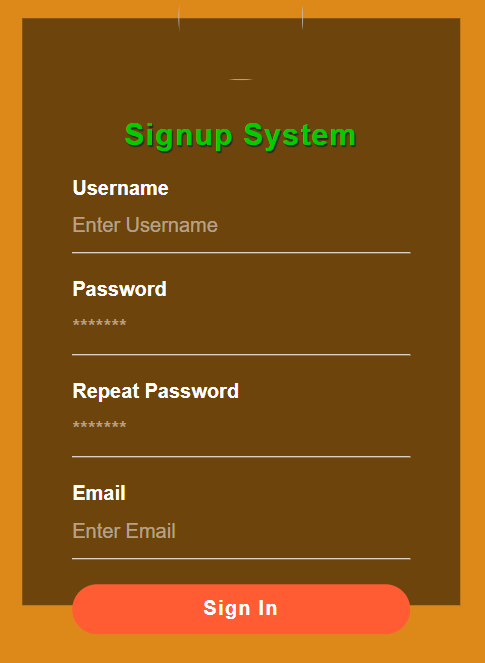
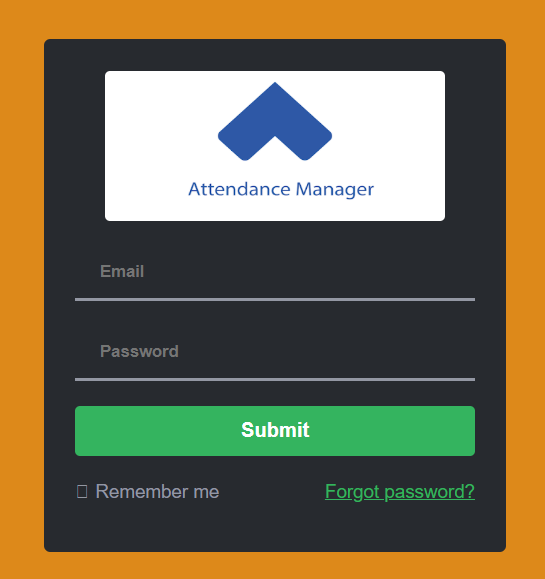


Fig 4.1.2 Login and SignUp Page

**4.1.3 Dashboard Page:**

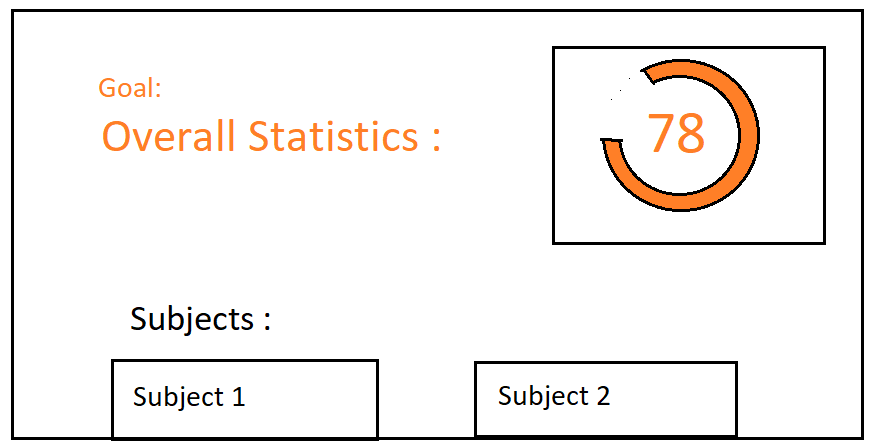


Fig 4.1.3 Dashboard Page

**4.1.4 Subject Analysis Page:**

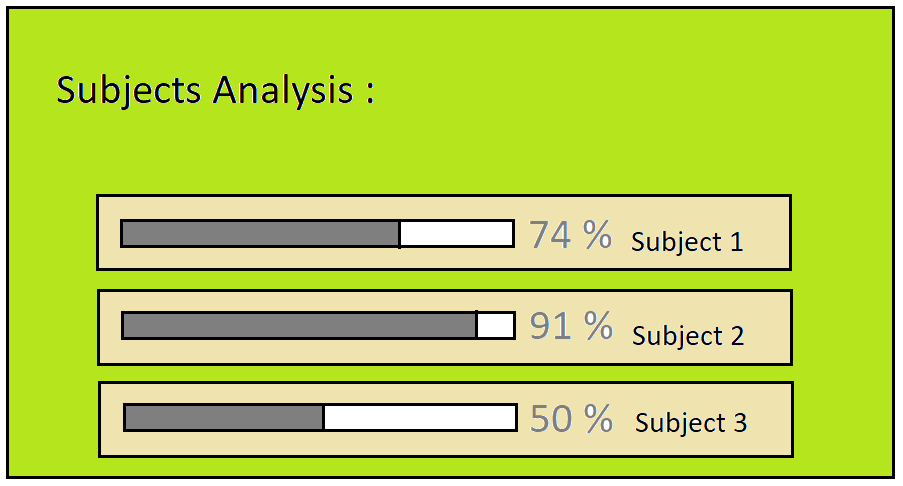
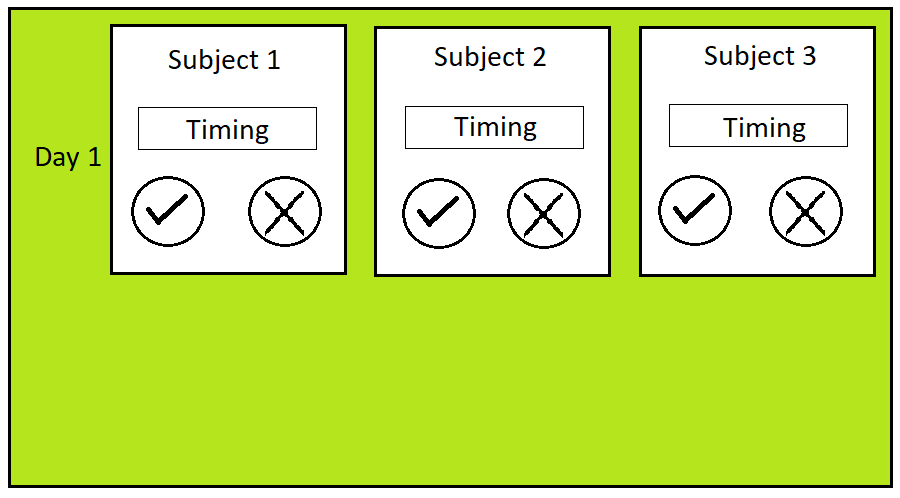


Fig 4.1.4 Subject Analysis Page

**4.1.5 Marking Attendance Page:**

  
 Fig 4.1.5 Marking Attendance Page

**4.2 Code design for Responsive web design:**

/\* Responsive layout - when the screen is less than 700px wide, make the two columns stack on top of each other instead of next to each other \*/

@media screen and (max-width: 700px) {

    .row {

      flex-direction: column;

    }

  }

  /\* Responsive layout - when the screen is less than 400px wide, make the navigation links stack on top of each other instead of next to each other \*/

  @media screen and (max-width: 400px) {

    .navbar a {

      float: none;

      width:100%;

    }

  }

**Chapter 5**

**DATABASE DESIGN AND CONNECTIVITY**

**5.1 Database Table Design**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| srno | username | password | rpassword | email |

Table 5.1.1 Students SignUp Details

|  |  |  |
| --- | --- | --- |
| sub\_id | sub\_name | sub\_teacher |

Table 5.1.2 Subjects Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| srno | monday | tuesday | wednesday | thursday | friday | saturday |

Table 5.1.3 Time Table

|  |  |
| --- | --- |
| username | password |

Table 5.1.4 Login Table

|  |  |
| --- | --- |
| attended | total\_conducted |

Table 5.1.3 Attendance View Table

**5.2 Database Connectivity**

<?php

$uname=$\_POST['usr'];

$pword=$\_POST['pswd'];

@session\_start();

$conn=mysqli\_connect("localhost","root","","attendance");

$result=mysqli\_query($conn,"SELECT \* FROM signupDB WHERE username='$uname' && password='$pword'");

$count=mysqli\_num\_rows($result);

if($count==1)

{

echo "Login Success";

$\_SESSION['log']=1;

$\_SESSION['user']=$uname;

header("refresh:2;url=dashboard.php");

}

else

{

echo "Give proper details";

header("refresh:2;url=loginpage.php");

}

?>

**Chapter 6**

**IMPLEMENTATION DETAILS**

**6.1 Snapshots**

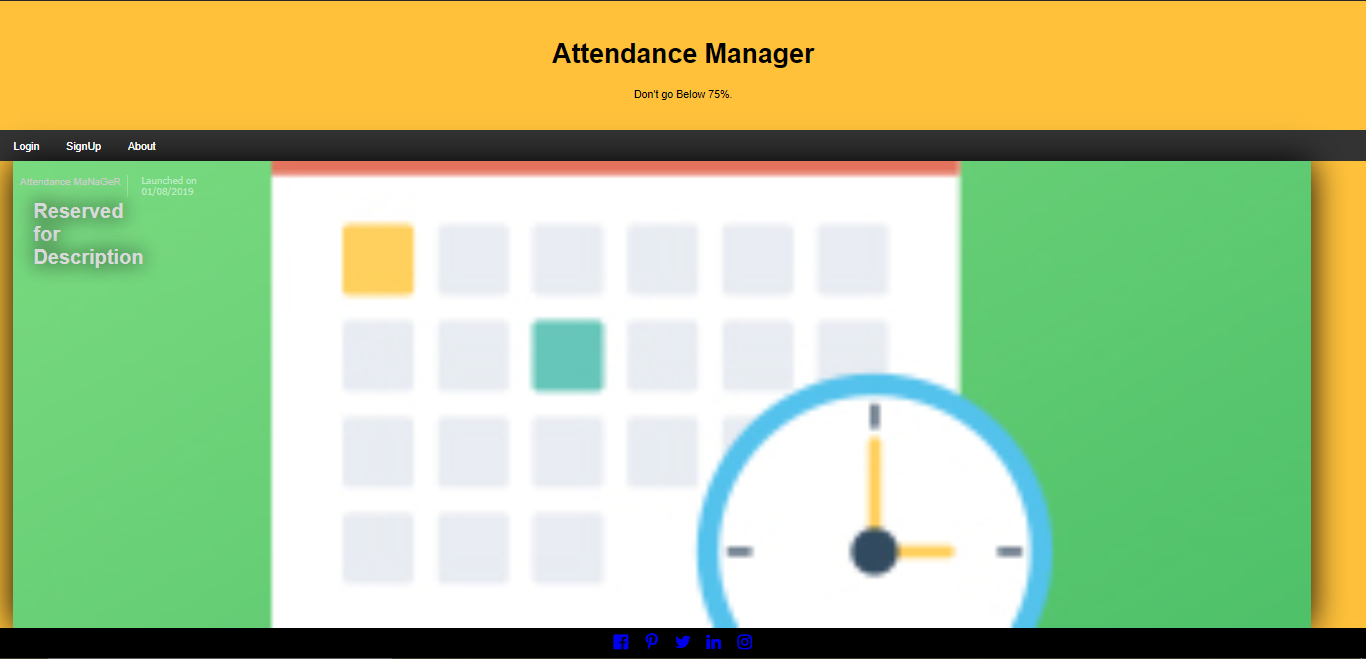


Fig 6.1.1Homepage

Responsive Homepage containing details about Attendance Manager on display with Login & SignUp button.

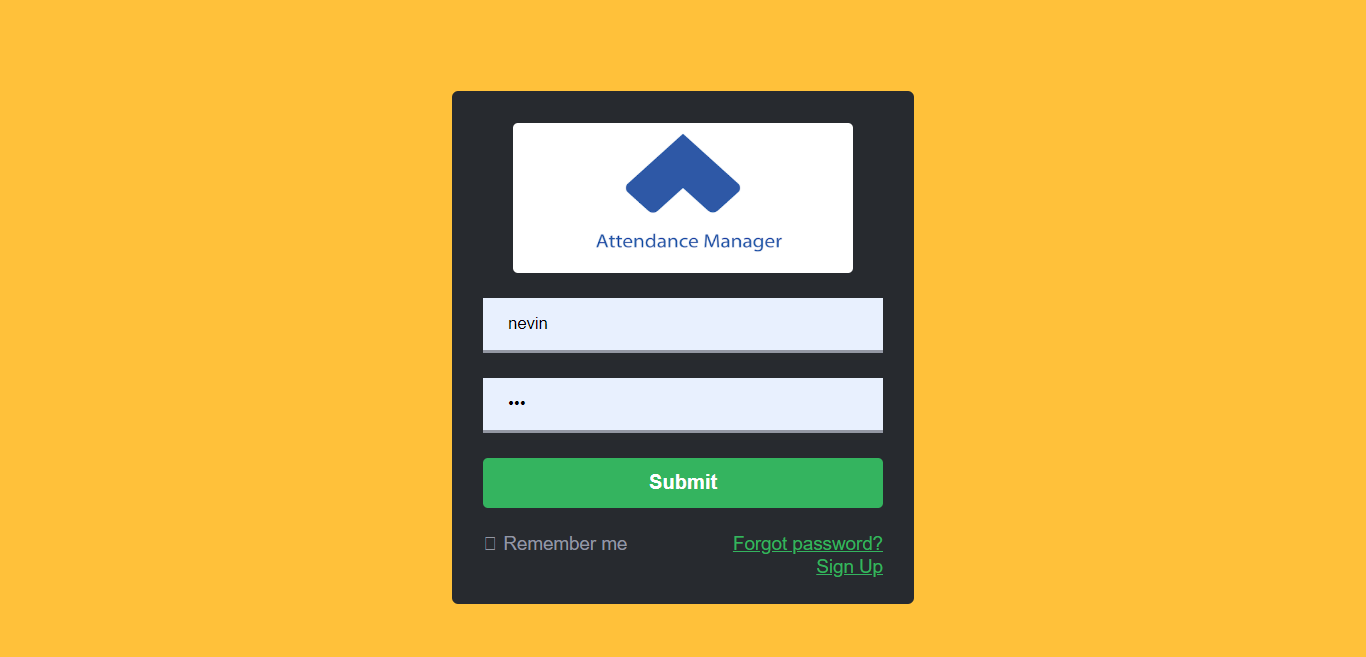


Fig 6.1.2 Login Page

Registered user can login by entering their username and password with a forgot password option. If not registered, the user can register using the “Sign\_Up” tab.

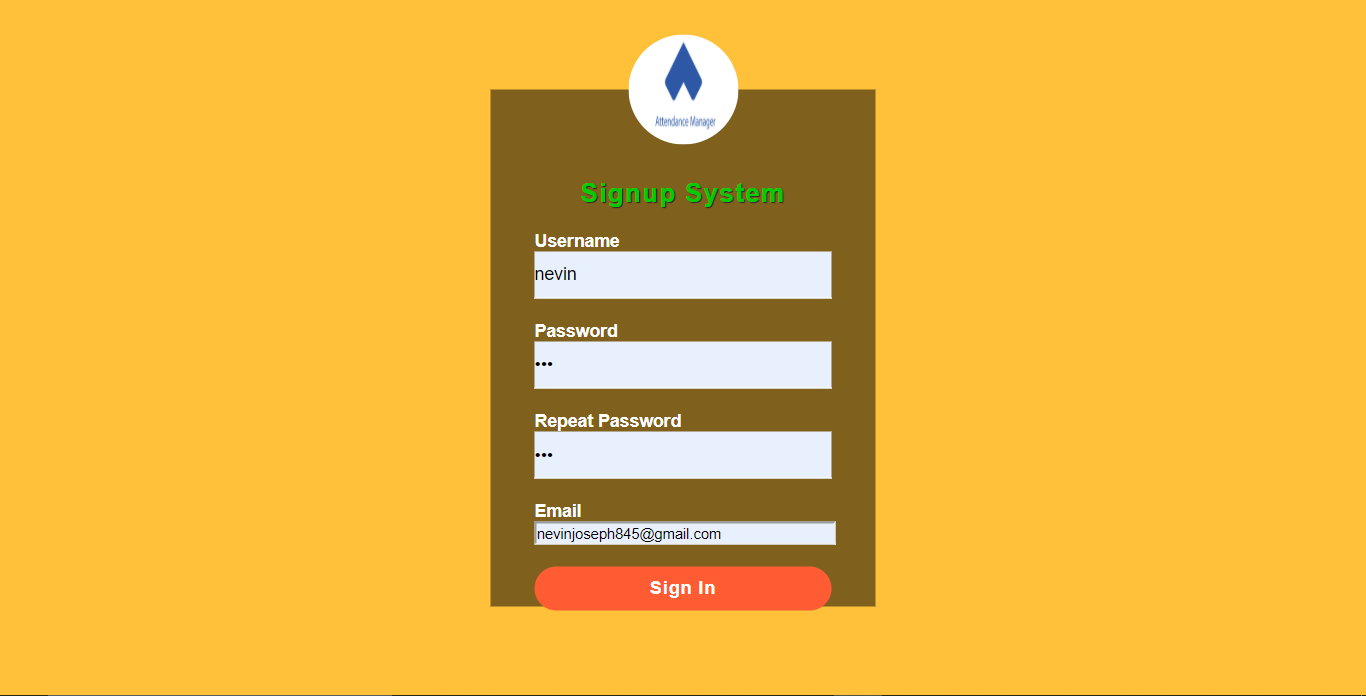


Fig 6.1.3 Registration Page

New users can register themselves by creating username, password and entering valid email address as the website verifies for the syntax of the email address entered.

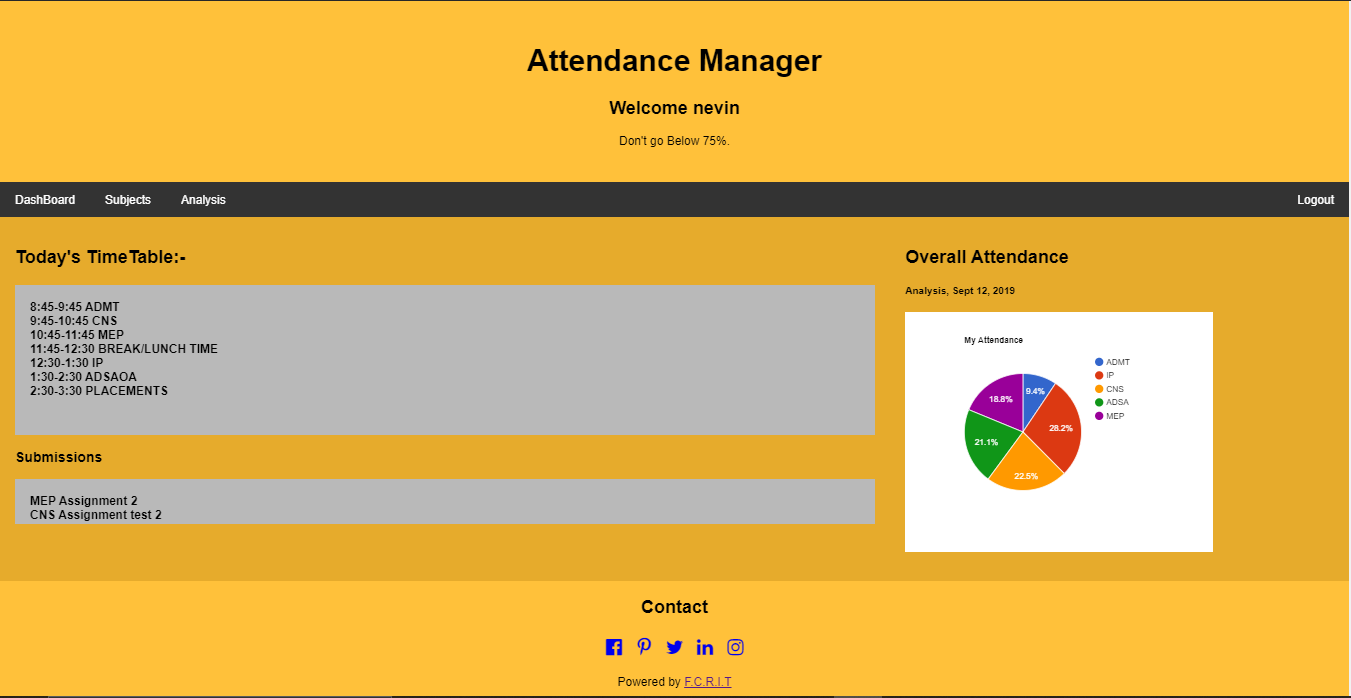


Fig 6.1.4 Dashboard

The page contains the overall percentage of the attendance of the student in the form of chart.

Also the page displays the day-to-day time table as per the students has entered



Fig 6.1.5 Subjects Updation Page

The page allows the student to add respective subjects with their respective subject teacher’s name. Here the student marks whether he/she attended or missed the lecture.



Fig 6.1.6 Analysis Page

The page displays the overall percentage of the attendance for each subject separately.

**Chapter 7**

**TESTING**

**7.1 HTML form validation**

<form action="" method="POST" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]);?>">

<p>Username</p>

<input type="text"name="user" placeholder="username" required>

<script type="text/javascript">

</script>

<p> Password</p>

<input type =password name="pass1" id="pass1" placeholder="enter your password" pattern="(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z]).{8,}" title="Must contain at least one number and one uppercase and lowercase letter, and at least 8 or more characters" required>

<p> confirm Password</p>

<input type =password name="pass2" id="pass2" placeholder="enter your password" required>

<div id="message">

<h6>Password must contain the following:Lower case,Capital letter,Number,Minimum 8 characters</h6>

<p id="letter" class="invalid"> </p>

<p id="capital" class="invalid"> </p>

<p id="number" class="invalid"> </p>

<p id="length" class="invalid"> </p>

</div>

<script>

var myInput = document.getElementById("pass1");

var letter = document.getElementById("letter");

var capital = document.getElementById("capital");

var number = document.getElementById("number");

var length = document.getElementById("length");

// When the user clicks on the password field, show the message box

myInput.onfocus = function() {

document.getElementById("message").style.display = "block";

}

// When the user clicks outside of the password field, hide the message box

myInput.onblur = function() {

document.getElementById("message").style.display = "none";

}

// When the user starts to type something inside the password field

myInput.onkeyup = function() {

// Validate lowercase letters

var lowerCaseLetters = /[a-z]/g;

if(myInput.value.match(lowerCaseLetters)) {

letter.classList.remove("invalid");

letter.classList.add("valid");

} else {

letter.classList.remove("valid");

letter.classList.add("invalid");

}

// Validate capital letters

var upperCaseLetters = /[A-Z]/g;

if(myInput.value.match(upperCaseLetters)) {

capital.classList.remove("invalid");

capital.classList.add("valid");

} else {

capital.classList.remove("valid");

capital.classList.add("invalid");

}

// Validate numbers

var numbers = /[0-9]/g;

if(myInput.value.match(numbers)) {

number.classList.remove("invalid");

number.classList.add("valid");

} else {

number.classList.remove("valid");

number.classList.add("invalid");

}

// Validate length

if(myInput.value.length >= 8) {

length.classList.remove("invalid");

length.classList.add("valid");

} else {

length.classList.remove("valid");

length.classList.add("invalid");

}

}

</script>

<p>E-mail id</p>

<input type="email" name="email" placeholder="abvc@gmail.com">

<input type="submit" name="submit" name="" value="register" onsubmit="validateemail()" onclick="return Validate12()" >

<script>

function validateemail()

{

var x=document.myform.email.value;

var atposition=x.indexOf("@");

var dotposition=x.lastIndexOf(".");

if (atposition<1 || dotposition<atposition+2 || dotposition+2>=x.length){

alert("Please enter a valid e-mail address \n atpostion:"+atposition+"\n dotposition:"+dotposition);

return false;

}

}

</script>

<script type="text/javascript">

function Validate12() {

var password = document.getElementById("pass1").value;

var confirmPassword = document.getElementById("pass2").value;

if (password != confirmPassword) {

alert("Passwords do not match.");

return false;

}

return true;

}

</script>

</form>

**7.2 Website Security testing for SQL injection**

if(!empty($\_POST['user']) && !empty($\_POST['pass1']) && !empty($\_POST['pass2']) && !empty($\_POST['email'])) {

$user= mysqli\_real\_escape\_string($con,$\_POST['user']);

$pass1=mysqli\_real\_escape\_string($con,$\_POST['pass1']);

$pass2=mysqli\_real\_escape\_string($con,$\_POST['pass2']);

$email=mysqli\_real\_escape\_string($con,$\_POST['email']);

}

$query=mysqli\_query($con,"SELECT \* FROM login WHERE username='".$user."'");

$numrows=mysqli\_num\_rows($query);

if($numrows==0)

{

$sql="INSERT INTO login(username,password,confirmpassword,email) VALUES('$user','$pass1','$pass2','$email')";

**Chapter 8**

**WEBSITE LAUNCHING**

**Steps for website launching:**

1. Select and register a domain name.
2. Find, choose and purchase web hosting.
3. Make a backup copy of your website files.
4. Strive to make your website easy to navigate.
5. Validate your code.
6. Implement a site map.
7. Test your website in a variety of web browsers.
8. Ensure that you're using SEO-friendly code.
9. Install website analytics to keep track your website's success and current status.

Transfer your website's files to your web host.

**Chapter 9**

**CONCLUSION AND FUTURE SCOPE**

**Conclusion:**

We have successfully accomplished in implementing our proposed idea. Our website provides the user to keep approximate record of the attendance for their academics. It allows user to enter their respective subjects and their respective attendance. Users can add their day-to-day time table and submissions.

**Future Scope:**

In future we aim to take this website further more advance by adding more features, making it more user friendly by adding such styles, cookies so that we can track the students interest and provide them a better range chat box by which the students can directly interact with the other students and be updated.

We would also like to develop an app of our website for better personalization, ease of sending notifications.

**Chapter 10**

**APPENDIX I: CODE SAMPLE**

/\*ABOUT THE WEBSITE ANS IT’S POLICY\*/

<section>

<div id="about">

<h1> About Us</h1>

<p>Furnspace is a Mumbai based company that helps people living in Mumbai to have a variety of options in furniture to rent upon.We provide furniture on rent so that people can live better at their homes.Our furniture is inspired by the urban lifestyle.We believe in a sustainable developement rather than just achieving more and more things in life.And whole idea behing this is why buy stuffs that lose their value by 50% as you buy them.

<h4>Our policy</h4>

<p>-Verify Profile</p> <p>

Join the community

Share a few essential details with us to be a part of our community and enjoy faster delivery of your orders.

Your details are safe with us

The security of your information is of utmost importance to us, it helps us build a secure network of trusted individuals. Your details are safe with us.</p>

<p>-Pay refundable Deposits</p><p>

Bespoke products to suit your needs!

With tenures starting at just 3 months.Pay the deposit at start and collect while return.</p>

<p>-Pay at once</p><p>

The furnitures are specified with the tenure at which they can be rented.But you can always rent it again once your tenure is over.

You just have to pay the whole amount at the start of the tenure.The best thing is our staff woould come install furniture and while returning disassemble and recollect the furniture from your place itself.</p><p>\*All terms and conditions applied.</p>

</p>

</div>

</section>

<section>

<div id="about">

<h1> Why We?</h1>

<p>Furnspace provided an end to end furnishing products.Customers need not worry about the quality of the product at all.

Not just that we provide services and installation of furniture for free.We provide you with furnitures for every room,every nook and corner of your house.</p>

</p>

</div>

</section>

/\*RESPONSIVE\*/

@media screen and (max-width: 600px) {

.topnav a:not(:first-child) {display: none;}

.topnav a.icon {

float: right;

display: block;

}

}

@media screen and (max-width: 600px) {

.topnav.responsive {position: relative;}

.topnav.responsive .icon {

position: absolute;

right: 0;

top: 0;

}

.topnav.responsive a {

float: none;

display: block;

text-align: left;

font-size: 12px;

}

.topnav a{

font-size: 12px;

}

#intro{

font-size: 12px;

}

#about{

font-size: 12px;

}

footer h1{

font-size: 12px;

}

.textfooter{

font-size: 12px;

}

footer p{

font-size: 12px;

}

}

/\*RESPONSIVE CODE ENDS HERE!\*/

**ACKNOWLEDGEMENT**

We would like to express special thanks of gratitude to our guide **Asst. Prof. Kalpana Wani** who gave us the golden opportunity to do this wonderful project on the topic of **Attendance Management System,** which helped us in doing a lot of research and we came to know about so many new things. We would also like to thank our HOD **Dr. H.K. Chavan** for providing us the opportunity to implement our project. We are really thankful to them. Finally, we would also like to thank our friends who helped us a lot in finalizing this project within the limited time frame.

1. Rony Benny (501750)

2. Nevin Joseph (501751)

3.Rachel Angelin (501745)

**REFERENCES**

* <https://www.w3schools.com/>
* “Learning php 5”,David Sklar,O’Reilly Publications
* “Web Technologies:Black Book”,Dreamtech publications
* [www.tutorialspoint.com](http://www.tutorialspoint.com/)
* [www.stackoverflow.com](http://www.stackoverflow.com/)