

# **Internet and** **Web Technologies** **Lab Project**

Student Database System

## **Group 26:**

Vishala Reddy (14/IT/36)

Gollapudi Vamsi Krishna (14/IT/40)

Ashish Agarwal (14/IT/60)

## **Introduction:**

The total population of our college is about 4000 students. It is difficult to maintain records for all the students in hard copy, so it is vital to have a standardized database where all the information is stored and can be modified by the user.

For the above mentioned problem, we have created a database for teachers and students across the college so that they can view their profile. It also simplifies the task of teacher to authenticate the students by viewing the respective profiles directly on the teachers profile. We also have a root user to modify all the above profiles.

Some of the features of the database system are:

- Forms for signup and log in.
- Authentication of the filled in details on click
- Account usage only after authentication from admin.
- User profile for both teachers and student
- Usage of Javascript for showing/hiding objects.

## **Technologies used:**

For achieving the goal, we have taken use of the following technologies:

- Hyper Text Markup Language (HTML)
- Cascading Style Sheets (CSS)
- PHP
- Javascript
- MySQL

We have not used technologies like Bootstrap or any other framework.

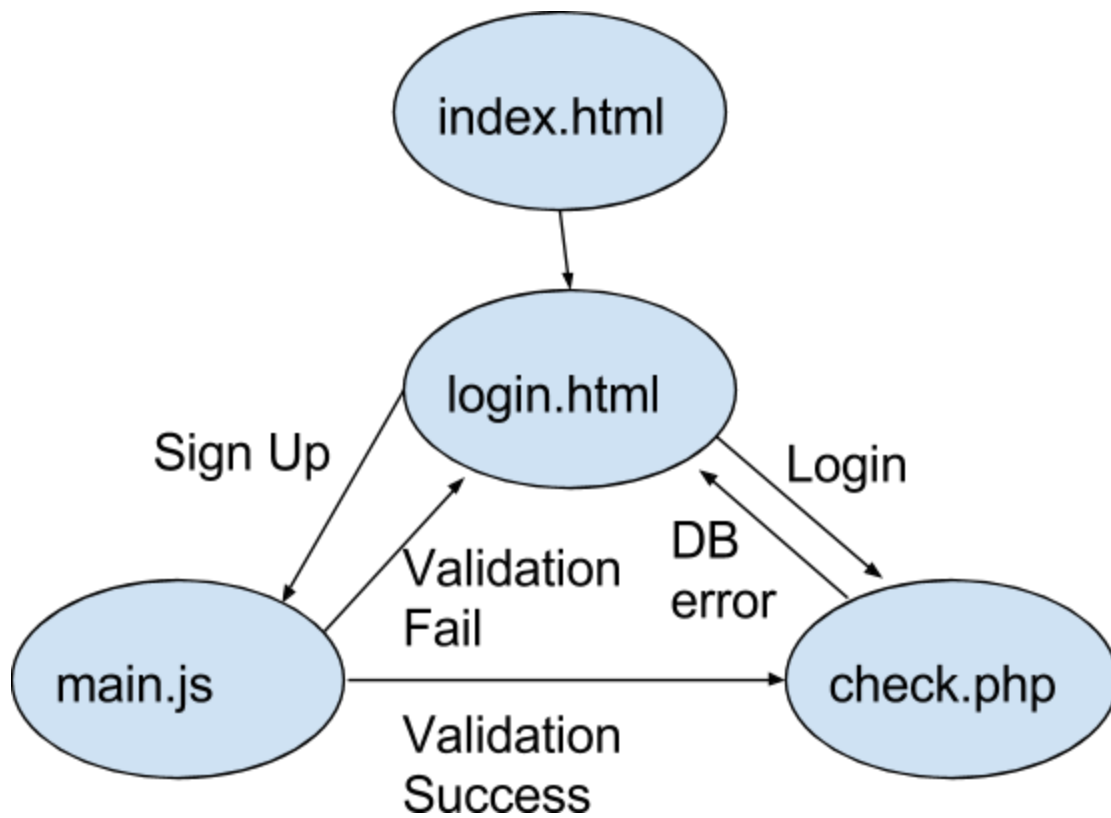
All of the code was written from basic understanding of the above mentioned technologies.

## **File Structure:**

The file structure of our project with description of each file is as given below.

- css/
  - style.css : This is the main CSS file of the whole project which has the styling of various elements of the project
- images/
  - This folder has all the assets of the project, includes images used on various pages
- js/
  - main.js : This file handles the javascript of the entire project with various functions in it. Some of the functions included in it are:
    - Display/Hide items when required.
    - Validate the forms on submission and display the error as required.
    - Load the options of combo boxes.
- index.html : This is the starting placeholder page of the website which displays our group info
- login.html : This is the most important page of the website which includes the login/signup option for teacher and student.
- student.php : This is the student user page, where he can view his info.
- teacher.php : This is the teacher user page, where they can validate the student profiles and they can also view their info here.
- check.php : This is a file used to update the user database with new signups
- activate.php : This is a background page which is used to activate profiles. This will be called whenever the teacher wishes to activate a profile.
- update.php : This is called when any update function is called by student/teacher.

### **FLOW OF CONTROL:**



Login/Signup Data Flow

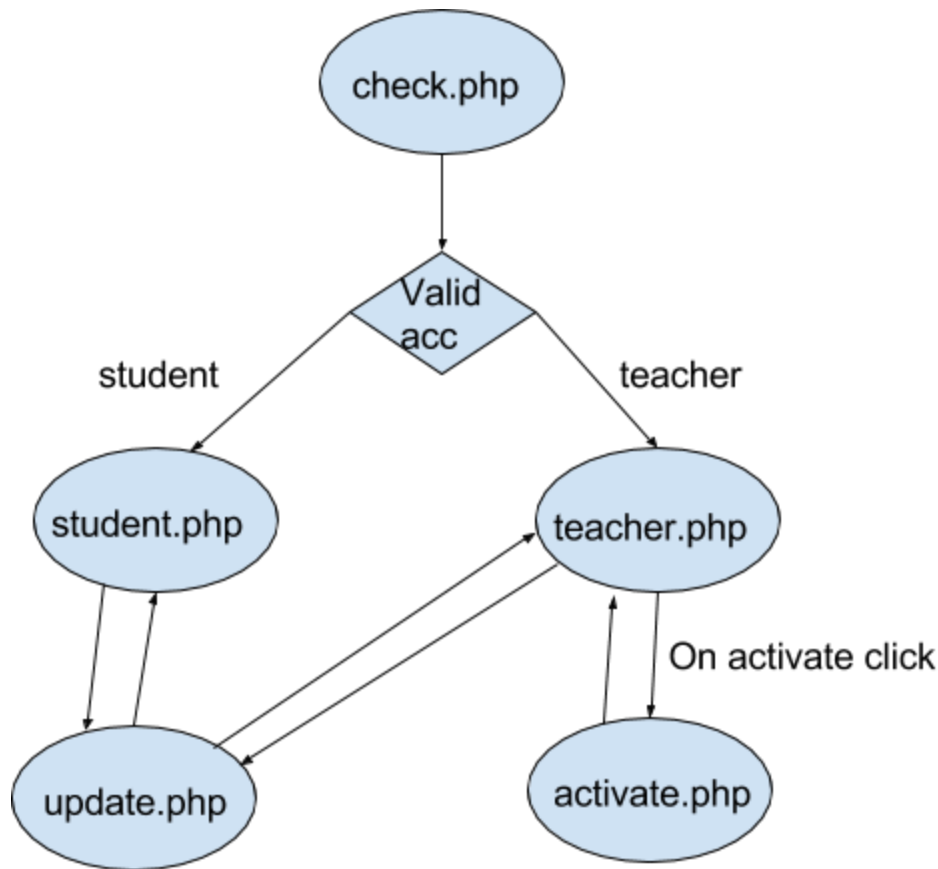
When a user submits a form of signup in `login.html`, it first goes through the **validate()** present in `main.js`.

This function checks if the required fields are filled, it also checks if the filled fields have followed the syntax and returns customized errors where required. Here is one of the example of customized error display which `main.js` handles:

<div>Username: <input type="text" value="Username"/></div> <div>This should not be blank</div>	<div>Username: <input type="text" value="user12"/></div> <div>This should have only alphabets</div>
--	---

Later, the submitted form goes to `check.php` which displays if there is any database error or if the signup was successful.

For login option, if the user exists and if the password is right, they will be directed to `student.php` or `teacher.php` respectively.



### Flow after successful login

When you are a student, after login, you will be redirected to the **student.php** page where the info of the logged in student is displayed. A few fields are editable, which on update option gets redirected to update.php and back to student.php.

When you are a teacher, you will be redirected to **teacher.php** page. You will also have the user information displayed. But, in addition to that, you will be given a list of students whom you can activate so that they can login.

There is also a special user named “**root**” who can activate both teacher and student accounts.

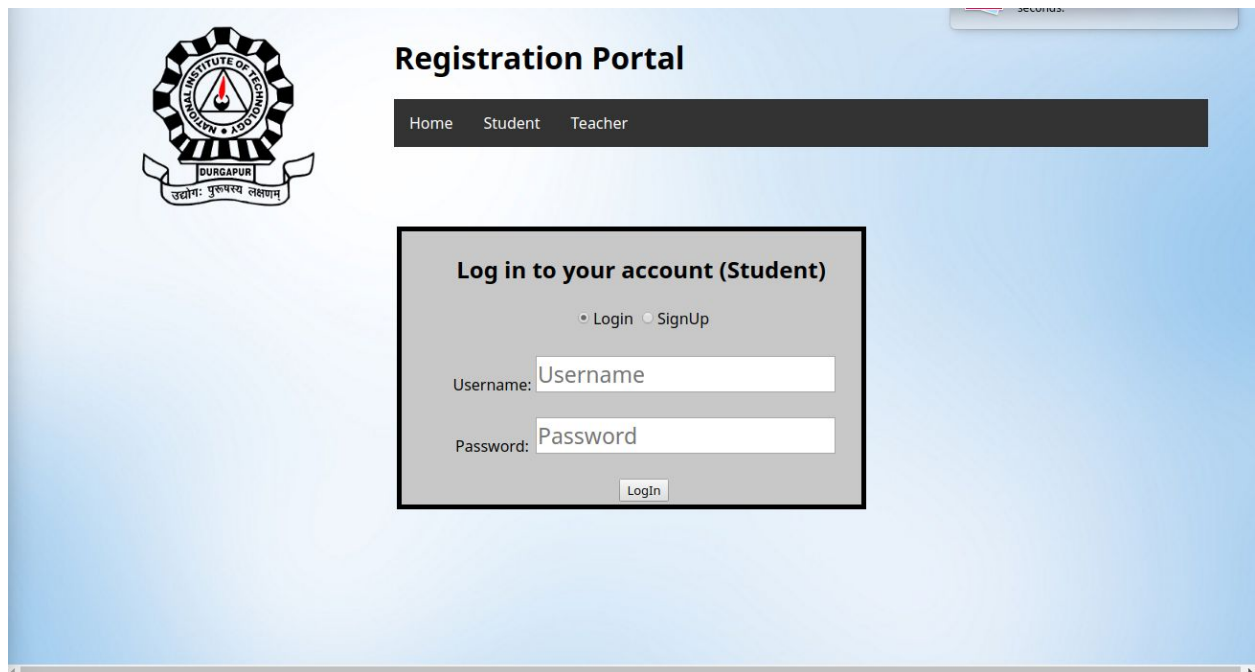
### Various Screenshots of Pages:

index.html:



This is the main landing page of the website which opens first.

login.html:



This is the main page of the entire interface, the default login option is set to student.

On clicking 'teacher' the center changes to the following:

**Welcome Teacher! Log in to your account (Teacher)**

☐ Login 
 ☐ SignUp

Username:

Password:

This above functionality is achieved by the following piece of code in javascript:

```
function studentT() {
  if(document.getElementById('teacher').style.display=='none'){
    document.getElementById('student').style.display = 'none';
    document.getElementById('teacher').style.display = 'block';
    var btn=document.getElementById('loginCheck');
    btn.checked=true;
    loginSignUp();
  }else{
    document.getElementById('teacher').style.display = 'none';
    document.getElementById('student').style.display = 'block';
    var btn=document.getElementById('loginCheck2');
    btn.checked=true;
    loginSignUp2();
  }
}
```

Similar function is also applied to switch between the login and signup option:

```
function loginSignUp(){
  if(document.getElementById('loginCheck').checked){
    document.getElementById('login').style.display = 'block';
    document.getElementById('signup').style.display = 'none';
  }else{
    document.getElementById('login').style.display = 'none';
    document.getElementById('signup').style.display = 'block';
  }
}
```

On clicking SignUp, we get the following form:

☐ Login ☒ SignUp

Username:

First Name:

Last Name:

Gender:

☐ Male ☐ Female

Date Of Birth:

Department

IT

Roll No:

Email ID:

Phone number:

Password:

Re Enter Password:

Address:

### Education qualifications

Highest level of Education: 

M Tech

10th Class

Institute:

Percentage:

Year of Passing:

12th Class

Institute:

Percentage:

Year of Passing:

B Tech

Institute:

Percentage:

Year of Passing:

M Tech

Institute:

Percentage:

Year of Passing:

SignUp

Reset



The overall structure of the page is as follow:

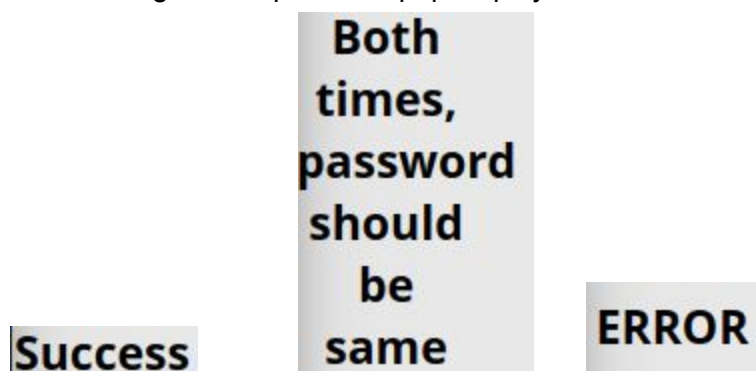
```
<div id="student" class="boxx">
  <table id="form">
    <tr><td><h2 align="center">Log in to your account (Student)</h2></td></tr>
    <tr><td>
      <center>...
    </center>
    <br></td></tr></table>
    <div id="login" align="center" >...
    </div>
    <div id="signup" align="center">...
    </div>
  </div>
  <div id="teacher" class="boxx" style="display: none">
    <h2 align="center">Welcome Teacher! Log in to your account (Teacher)</h2>
    <center>...
    </center>
    <br>
    <div id="login2" align="center" >...
    </div>
    <div id="signup2" align="center" style="display: none">...
    </div>
  </div>
```

This helps, as at any particular moment only one of the following has display is set to block:

- student->login
- student->signup
- teacher->login
- student->signup

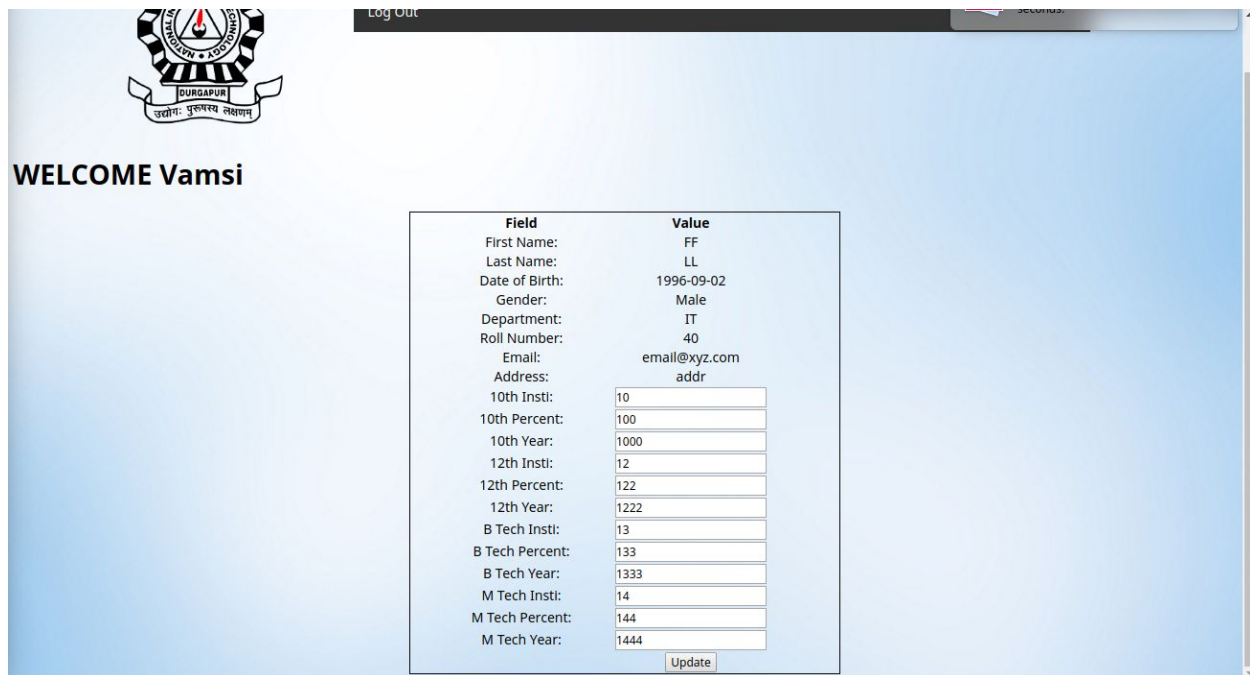
#### check.php:

Later, on login attempt, check.php displays one of the following depending on the db change:



student.php:

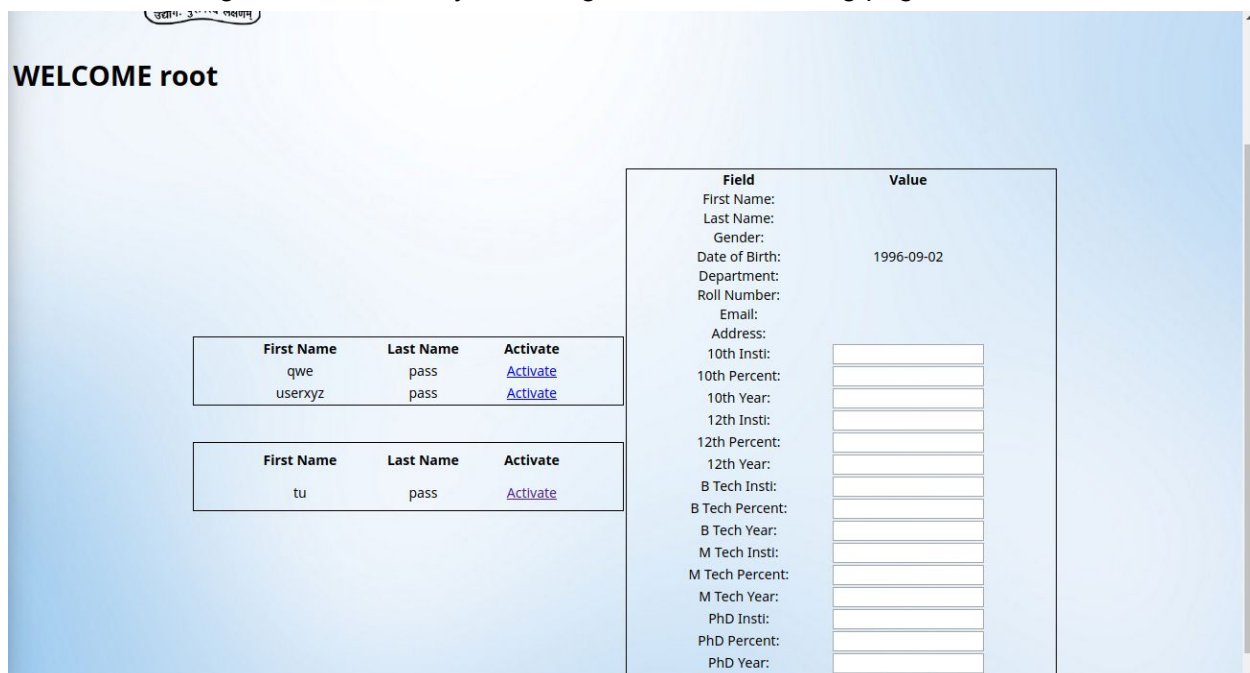
After this, if the student logs, he is navigated to the following page:



Field	Value
First Name:	FF
Last Name:	LL
Date of Birth:	1996-09-02
Gender:	Male
Department:	IT
Roll Number:	40
Email:	email@xyz.com
Address:	addr
10th Insti:	<input type="text" value="10"/>
10th Percent:	<input type="text" value="100"/>
10th Year:	<input type="text" value="1000"/>
12th Insti:	<input type="text" value="12"/>
12th Percent:	<input type="text" value="122"/>
12th Year:	<input type="text" value="1222"/>
B Tech Insti:	<input type="text" value="13"/>
B Tech Percent:	<input type="text" value="133"/>
B Tech Year:	<input type="text" value="1333"/>
M Tech Insti:	<input type="text" value="14"/>
M Tech Percent:	<input type="text" value="144"/>
M Tech Year:	<input type="text" value="1444"/>

**teacher.php:**

On successful login of teacher, they are navigated to the following page:



First Name	Last Name	Activate
qwe	pass	<a href="#">Activate</a>
userxyz	pass	<a href="#">Activate</a>

Field	Value
First Name:	
Last Name:	
Gender:	
Date of Birth:	1996-09-02
Department:	
Roll Number:	
Email:	
Address:	
10th Insti:	<input type="text"/>
10th Percent:	<input type="text"/>
10th Year:	<input type="text"/>
12th Insti:	<input type="text"/>
12th Percent:	<input type="text"/>
12th Year:	<input type="text"/>
B Tech Insti:	<input type="text"/>
B Tech Percent:	<input type="text"/>
B Tech Year:	<input type="text"/>
M Tech Insti:	<input type="text"/>
M Tech Percent:	<input type="text"/>
M Tech Year:	<input type="text"/>
PhD Insti:	<input type="text"/>
PhD Percent:	<input type="text"/>
PhD Year:	<input type="text"/>

This is the page of special user **root**.

## Validations:

One of the major part of the project consists of validation of the data entered by the user during sign up.

The different fields were first divided into different input groups, as the following:

- Text
  - These include fields like “Username”, “First Name”, “Last Name”, “Roll”, “Email”, “Phone”;
- Password
  - These include “Password” and “Re-type Password”

Initially, we check if the input box is empty or not and display the corresponding error.

```
if(val==""){  
  try{  
    document.getElementById(ele[i].id+"Error").textContent="This should not be blank";  
    chk=false;  
    chk2=false;  
  }catch(err){}  
  //continue;  
}
```

For authenticating the “text” fields, we have set the following regex option:

```
emailRegex=/^[A-Za-z](\.)?[0-9A-Za-z+]*@[A-Za-z](\.)?[0-9A-Za-z+]*(\.)[A-Za-z]{2,3}$/;  
phoneRegex=/^[0-9]{9}$/;  
nameRegex=/^[a-zA-Z]+$/;  
rollRegex=/^[0-9]{2}$/;
```

The entered input are checked with the above regex to see if they are valid or not.

As for the other input fields like “Username”, “First Name” and “Last Name”, we just check if it is all words and not any other characters, as defined by the nameRegex.

After we have checked, we display the corresponding error with respect to the regex.

```
if(ele[i].type === 'text'){
    //uname, fname, lname, roll, email, phone
    emailRegex=/^[A-Za-z]((\.)?[0-9A-Za-z]+)*@[A-Za-z]((\.)?[0-9A-Za-z]+)*(\.)[A-Za-z]{2,3}$/;
    phoneRegex=/^[0-9]{9}$/;
    nameRegex=/^[a-zA-Z]+$/;
    rollRegex=/^[0-9]{2}$/;
    if(ele[i].id=="roll"){
        if(!rollRegex.test(val)){
            document.getElementById(ele[i].id+"Error").textContent="This should be a two Ddigit Number.";
            chk=false;
        }
    }else if(ele[i].id=="phone" || ele[i].id=="phone2"){
        if(!phoneRegex.test(val)){
            document.getElementById(ele[i].id+"Error").textContent="This should be 9 digits.";
            chk=false;
        }
    }else if(ele[i].id=="email" || ele[i].id=="email2"){
        if(!emailRegex.test(val)){
            document.getElementById(ele[i].id+"Error").textContent="This should be of the form xyz@email.com";
            chk=false;
        }
    }else{
        if(!nameRegex.test(val)){
            try{
            }catch(err){
                document.getElementById(ele[i].id+"Error").textContent="This should only have small and captital alphabets";
                console.log(ele[i].id+"Error");
            }
            chk=false;
        }
    }
}
}else{
```

## Conclusion:

We have learnt the usage of html, css, javascript in making the front end of the website, including user interactions. We now know to manipulate the above to improve the user experience by knowing the usage of UI/UX.

Additionally, we have also learnt the powerful PHP which helps to maintain databases and transit between various pages with information.