**1** **Overview of Web**

The Internet is a worldwide collection of networks that links millions of businesses, government agencies, educational institutions, and individuals.

The Internet originated as ARPANET (Advanced Research Projects Agency Network) in September 1969.

**IP address:**

An IP address is a number that uniquely identifies each computer or device connected to the Internet.

**URL:**

We all use URLs to visit web pages and other resources on the web. The URL is an address that sends users to a specific resource online, such as a webpage, video or other document or resource. When you search Google, for example, the search results will display the URL of the resources that match your search query. The title in search results is simply a hyperlink to the URL of the resource.

A URL is one type of ***U***niform ***R***esource ***I***dentifier .The generic term for all types of names and addresses that refer to objects on the World Wide Web.URL stands for Uniform Resource Locator, and is used to specify addresses on the World Wide Web. A URL is the fundamental network identification for any resource connected to the web (e.g., hypertext pages, images, and sound files).

URLs have the following format:

Protocol: // hostname/ other\_ information

**Domain name:**

A domain name is the text version of an IP address .DNS server translates the domain name into its associated IP address.

**World Wide Web:**

World Wide Web or Web consists of a worldwide collection of electronic documents (Web pages) .A Web site is a collection of related Web pages and associated items.

Web server:

A Web server is a computer that delivers requested Web pages to your computer.

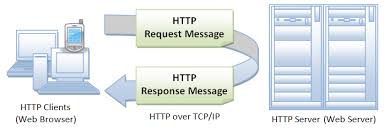
**Web Client:**

Web Clients request pages from the Server, and displays them to the user. In most cases, the client is a web browser.



**Protocols:**

Protocols define how messages are sent and received. A set of rules that defines how data is formatted and processed on a network. Types of Protocol are TCP/IP, SMTP etc. In Web technology HTTP protocol is used, which work on client/server based technique.



**2**  **Overview of HTML**

**What is HTML:**

* HTML stands for Hyper Text Markup Language.
* HTML describes the structure of Web pages using markup.
* HTML elements are the building blocks of HTML pages.
* HTML elements are represented by tags.
* HTML tags label pieces of content such as "heading", "paragraph", "table", and so on.
* Browsers do not display the HTML tags, but use them to render the content of the pages.

**Brief history of html:**

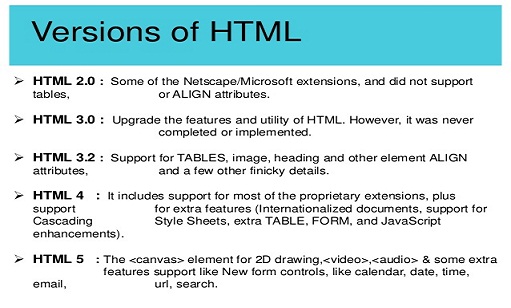
In 1989, Berners-Lee wrote a memo proposing an Internet- based hypertext system. Berners-Lee specified **HTML** and wrote the browser and server software in late 1990. ... **HTML** is a markup language that web browsers use to interpret and compose text, images, and other material into visual or audible web pages.

When Tim Berners-Lee was putting together his first elementary browsing and authoring system for the Web, he created a quick little hypertext language that would serve his purposes. He imagined dozens, or even hundreds, of hypertext formats in the future, and smart clients that could easily negotiate and translate documents from servers across the internet. The problem, however, turned out to be in the simplicity of Berners-Lee's language. Since it was text-based, you could use any editor or word processor to create or convert documents for the Web. And there was just a handful of tags – anyone could master HTML in an afternoon. The Web flourished. Everyone started publishing.

Early browsers were simply text-based, and there was an immediate desire to display figures and icons inline on a page. In 1993, a debate was exploding on the fledgling HTML mailing list, and finally a college student named Marc Andreessen  added < img> to his Mosaic browser. People objected, saying it was too limited. They wanted <include>, which would allow you to add any sort of media to a Web page with the much-touted content negotiation used on the client .

HTML continued to grow, with new, powerful, and exciting tags. We got <background>, <frame>, <font>, and of course, <blink>. Microsoft jumped into the game, and <marquee>, <iframe>, and <bgsound> started competing for room in the spec. And all this time, the W3C furiously debated something called HTML3, a sprawling document outlining all sorts of neat new features that nobody supported.It was now 1995, and things were an absolute mess.

HTML3 was dropped entirely, and work began on HTML 3.2, which, ironically, was far less technologically advanced than its predecessor. But, more importantly, it was realistic in its goal to give content providers and browser developers a common, if dated, reference from which to work.



Now currently HTML 5.1 is used which released in 2016.HTML 5.1 has many new features that makes users more helpful.

# HTML Tags:

This is a list of tags used in the HTML language. Each tag starts with a tag opener (a less than sign) and ends with a tag closer (a greater than sign). Many tags have corresponding closing tags which identical except for a slash after the tag opener. (For example, the TITLE tag). <TITLE> ... </TITLE>

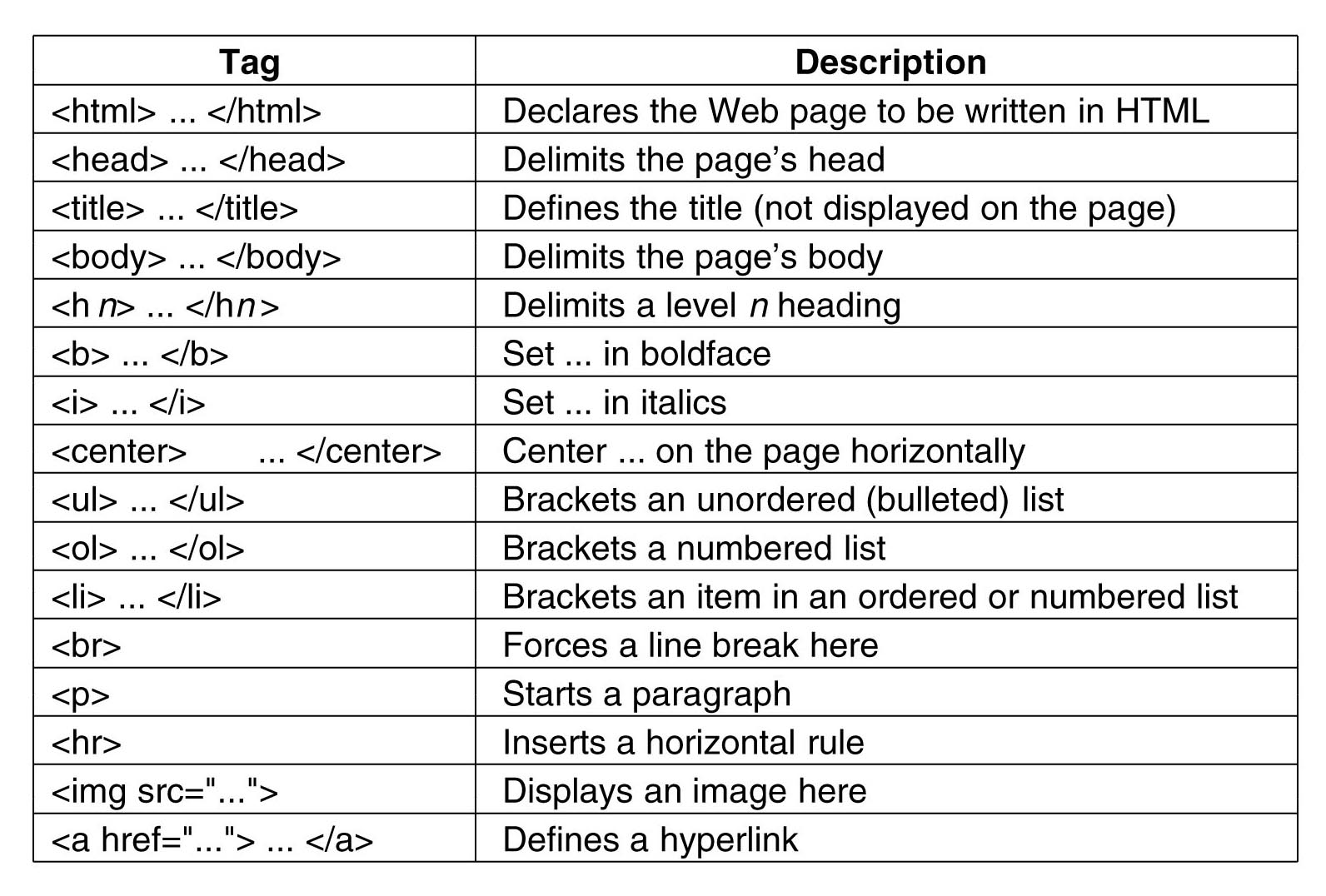
## Anchors:

The format of an anchor is as follows:

<A NAME=xxx HREF=XXX> ... </A>

The text between the opening tag and the closing tag is either the start or destination (or both) of a link. HREF gives the address of link.

Some other tags are in the given image,



**HTML FORM:**

A web form, web form or HTML form on a web page allows a user to enter data that is sent to a server for processing. Forms can resemble paper or database forms because web users fill out the forms using checkboxes, radio buttons, or text fields.

<!DOCTYPE html>

<html>

<head>

</head>

<body>

<form action='profile.php'method='post' onsubmit="return login()">

<table>

<tr><td>Email id</td><td><input type='text' /> ></td>

<td></td>

<tr><td>Password</td><td><input type='password' td>

<tr><td><input type='submit' value='Login'/><span></td>

<td></td>

</tr>

</table>

</form>

</body>

</html>



HTML forms are basically way to communication between user and host of the website. As above we have to fill the login as email and password to submit the request. Here user provide the details using form to the host.

# 3 Overview of CSS

Cascading style sheets (CSS) contain style rules that are applied to elements in a webpage. These styles define how elements are displayed and where they are positioned on the page. Visual Studio provides tools that you can use to work with CSS.

### Writing CSS Styles:

You can write CSS style rules in several places, including the following:

* Inline with the HTML markup.
* In a style element in a webpage.
* In an external style sheet (.css file) that is linked or imported into the webpage.

In general, you write rules that apply to the whole Web site in an external style sheet. You write style rules that apply only to a page in the page's style element. You write style rules that apply to a single page element as an inline style. Many designers and developers find that writing all style rules in one or more external style sheets makes maintaining styles easier.

#### Creating Inline Styles:

An inline style rule is defined in an element's opening tag by using the style attribute. Use an inline style when you want to define properties for a single element in a webpage and you do not want to re-use that style. The following example shows an inline style.

Eg-

<p style="font-weight: bold; font-style: italic; color: #FF0000">

#### Creating Internal (Page-Specific) CSS Styles:

CSS style rules can be defined in a style element inside the head element of a webpage. In that case, the style rules apply only to elements in that page.

<!DOCTYPE HTML>

<html>

<head>

<title>CSS Element Style Example</title>

<style>

h1{text-align:center; color:blue;}

</style>

</head>

<body>

<h1> This text is centered and blue</h1>

</body>

</html>

#### Creating External Cascading Style Sheets:

An external style sheet is a text file that has a .css file name extension and that contains only style rules. You can link a style sheet to a webpage by using a link element.

<link rel="stylesheet" href="myStyles.css" />

Style rules that are listed in an external style sheet are written the same way that they are in a style element.

External stylesheet are as .css extension.

h1 { text-align: center; color :blue; }

.head2 { font-size:14pt; text-align: center; color :blue;}

**4** **Overview of JavaScript**

JavaScript, often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

Javascript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as LiveScript, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name LiveScript. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

* JavaScript is a lightweight, interpreted programming language.
* Designed for creating network-centric applications.
* Complementary to and integrated with Java.
* Complementary to and integrated with HTML.
* Open and cross-platform

## Client-side JavaScript:

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field.

The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.

JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

## Advantages of JavaScript

The merits of using JavaScript are −

* **Less server interaction** − You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
* **Immediate feedback to the visitors** − They don't have to wait for a page reload to see if they have forgotten to enter something.
* **Increased interactivity** − You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
* **Richer interfaces** − You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

## Limitations of JavaScript

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features −

* Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
* JavaScript cannot be used for networking applications because there is no such support available.
* JavaScript doesn't have any multithreading or multiprocessor capabilities.

Once again, JavaScript is a lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages.

## The HTML<script>Tag:

The **<script>** tag is used to define a client-side script (JavaScript).

The <script> element either contains scripting statements, or it points to an external script file through the **src** attribute.

Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content.

To select an HTML element, JavaScript very often use the document.getElementById(id) method.

This example calls a function which performs a calculation and alert and returns the result:

<!DOCTYPE html>

<html>

<head>

<title>Calculation</title>

</head>

<body>

<p id="demo"></p>

<script>

var x = myFunction(4, 3);

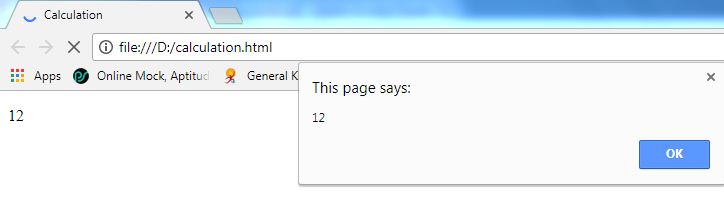
document.getElementById("demo").innerHTML = x;

function myFunction(a, b) {

alert(a\*b);

return a \* b;}

</script></body></html>

****

**5 Overview of PHP**

## What is PHP?

* PHP is an acronym for "PHP: Hypertext Preprocessor"
* PHP is a widely-used, open source scripting language
* PHP scripts are executed on the server
* PHP is free to download and use

## What is a PHP File?

* PHP files can contain text, HTML, CSS, JavaScript, and PHP code
* PHP code are executed on the server, and the result is returned to the browser as plain HTML
* PHP files have extension ".php"

## What Can PHP Do?

* PHP can generate dynamic page content
* PHP can create, open, read, write, delete, and close files on the server
* PHP can collect form data
* PHP can send and receive cookies
* PHP can add, delete, modify data in your database
* PHP can be used to control user-access
* PHP can encrypt data

With PHP you are not limited to output HTML. You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.

## Why PHP?

* PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
* PHP is compatible with almost all servers used today (Apache, IIS, etc.)
* PHP supports a wide range of databases
* PHP is free. Download it from the official PHP resource: www.php.net
* PHP is easy to learn and runs efficiently on the server side

## Basic PHP Syntax:

A PHP script can be placed anywhere in the document. A PHP script starts with

**<?php** and ends with **?>**.

The default file extension for PHP files is ".php". A PHP file normally contains HTML tags, and some PHP scripting code. Below, we have an example of a simple PHP file, with a PHP script that uses a built-in PHP function "echo" to output the text "Hello Wor<!DOCTYPE html>

<html>

<body>

<h1>My first PHP page</h1>

<?php

echo "Hello World!";

?>

</body>

</html>

PHP uses many tag of HTML as well as other also like echo.

eg:PHP function call in example

<!DOCTYPE html>

<html>

<body>

<?php

function familyName($fname, $year) {

echo "$fname Refsnes. Born in $year <br>";

}

familyName("Hege","1975");

familyName("Stale","1978");

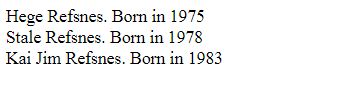
familyName("Kai Jim","1983");

?>

</body>

</html>

**Output:**



**What is a PHP Session?**

When you work with an application, you open it, do some changes, and then you close it. This is much like a Session. The computer knows who you are. It knows when you start the application and when you end. But on the internet there is one problem: the web server does not know who you are or what you do, because the HTTP address doesn't maintain state.

## Start a PHP Session:

## A session is started with the session\_start() function.Session variables are set with the PHP global variable: $\_SESSION.

## Destroy a PHP Session:

To remove all global session variables and destroy the session, use session\_unset() and session\_destroy():

## PHP include and require:

It is possible to insert the content of one PHP file into another PHP file (before the server executes it), with the include or require statement.

## PHP Global Variables – Superglobals:

Several predefined variables in PHP are "super globals", which means that they are always accessible, regardless of scope - and you can access them from any function, class or file without having to do anything special.

The PHP super global variables are:

* $GLOBALS
* $\_SERVER
* $\_REQUEST
* $\_POST
* $\_GET
* $\_FILES
* $\_ENV
* $\_COOKIE
* $\_SESSION

## PHP $GLOBALS:

$GLOBALS is a PHP super global variable which is used to access global variables from anywhere in the PHP script (also from within functions or methods).

PHP stores all global variables in an array called $GLOBALS[*index*]. The index holds the name of the variable.

## PHP $\_SERVER:

## $\_SERVER is a PHP super global variable which holds information about headers, paths, and script locations.

## PHP $\_REQUEST:

PHP $\_REQUEST is used to collect data after submitting an HTML form .The example below shows a form with an input field and a submit button. When a user submits the data by clicking on "Submit", the form data is sent to the file specified in the action attribute of the <form> tag. In this example, we point to this file itself for processing form data. If you wish to use another PHP file to process form data, replace that with the filename of your choice. Then, we can use the super global variable $\_REQUEST to collect the value of the input field.

## PHP $\_POST:

PHP $\_POST is widely used to collect form data after submitting an HTML form with method="post". $\_POST is also widely used to pass variables.The example below shows a form with an input field and a submit button. When a user submits the data by clicking on "Submit", the form data is sent to the file specified in the action attribute of the <form> tag. In this example, we point to the file itself for processing form data. If you wish to use another PHP file to process form data, replace that with the filename of your choice.

## PHP $\_GET:

PHP $\_GET can also be used to collect form data after submitting an HTML form with method="get".$\_GET can also collect data sent in the URL.Assume we have an HTML page that contains a hyperlink with parameters.

Mostly these above superglobals are used with every PHP page or FORM.

**6 Overview Of MySql**

## What is a Database?

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.

Other kinds of data stores can also be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those type of systems. Nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as Foreign Keys.

A **Relational DataBase Management System (RDBMS)** is a software that −

* Enables you to implement a database with tables, columns and indexes.
* Guarantees the Referential Integrity between rows of various tables.
* Updates the indexes automatically.
* Interprets an SQL query and combines information from various tables.

## MySQL Database:

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

* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
* MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development.
* MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
* MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

## Installing MySQL on Windows

The default installation on any version of Windows is now much easier than it used to be, as MySQL now comes neatly packaged with an installer. Simply download the installer package, unzip it anywhere and run the setup.exe file.

The default installer setup.exe will walk you through the trivial process and by default will install everything under C:\mysql.

Test the server by firing it up from the command prompt the first time. Go to the location of the **mysqld server** which is probably C:\mysql\bin.

Some MySql Commands are as

Create new database: create database [database];

List all indexes on a table: show index from [table];

Create new table with columns: CREATE TABLE [table] ([column] VARCHAR(120), [another-column] DATETIME);

Adding a column: ALTER TABLE [table] ADD COLUMN [column] VARCHAR(120);

Adding a column with an unique, auto-incrementing ID: ALTER TABLE [table] ADD COLUMN [column] int NOT NULL AUTO\_INCREMENT PRIMARY KEY;

Inserting a record: INSERT INTO [table] ([column], [column]) VALUES ('[value]', [value]');

Selecting records: SELECT \* FROM [table];

Explain records: EXPLAIN SELECT \* FROM [table];

Selecting parts of records: SELECT [column], [another-column] FROM [table];

Counting records: SELECT COUNT([column]) FROM [table];

Counting and selecting grouped records: SELECT \*, (SELECT COUNT([column]) FROM [table]) AS count FROM [table] GROUP BY [column];

Selecting specific records: SELECT \* FROM [table] WHERE [column] = [value]; (Selectors: <, >, !=; combine multiple selectors with AND, OR)

Select records containing [value]: SELECT \* FROM [table] WHERE [column] LIKE '%[value]%';

Select records starting with [value]: SELECT \* FROM [table] WHERE [column] LIKE '[value]%';

Select a range: SELECT \* FROM [table] WHERE [column] BETWEEN [value1] and [value2];

Select with custom order and only limit: SELECT \* FROM [table] WHERE [column] ORDER BY [column] ASC LIMIT [value];(Order: DESC, ASC).

Updating records: UPDATE [table] SET [column] = '[updated-value]' WHERE [column] = [value];

Deleting records: DELETE FROM [table] WHERE [column] = [value];

Delete all records from a table (without dropping the table itself): DELETE FROM [table];

 (This also resets the incrementing counter for auto generated columns like an id column.)

Delete all records in a table: truncate table [table];

Removing table columns: ALTER TABLE [table] DROP COLUMN [column];

Deleting tables: DROP TABLE [table];

Deleting databases: DROP DATABASE [database];

Custom column output names: SELECT [column] AS [custom-column] FROM [table];

Logout: exit;

## Aggregate functions

Calculate total number of records: SELECT SUM([column]) FROM [table];

Count total number of [column] and group by [category-column]: SELECT [category-column], SUM([column]) FROM [table] GROUP BY [category-column];

Get largest value in [column]: SELECT MAX([column]) FROM [table];

Get smallest value: SELECT MIN([column]) FROM [table];

Get average value: SELECT AVG([column]) FROM [table];

## Users functions:

## List all users: SELECT User ,Host FROM mysql.user;

## Create new user: CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';

Grant ALL access to user for \* tables: GRANT ALL ON database.\* TO 'user'@'localhost';

**7 Overview of XAMPP**

XAMPP stands for Cross-Platform (X), Apache (A), MySQL (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing purposes. Everything you need to set up a web server – server application (Apache), database (MySQL), and scripting language (PHP) – is included in a simple extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server is extremely easy as well. Web development using XAMPP is especially beginner friendly

**What is included in XAMPP:**

XAMPP has four primary components. These are:

**1. Apache:** Apache is the actual web server application that processes and delivers web content to a computer. Apache is the most popular web server online, powering nearly 54% of all websites.

**2. MySQL:** Every web application, howsoever simple or complicated, requires a database for storing collected data. MySQL, which is open source, is the world’s most popular database management system. It powers everything from hobbyist websites to professional platforms like WordPress.

**3. PHP:** PHP stands for Hypertext Preprocessor. It is a server-side scripting language that powers some of the most popular websites in the world, including WordPress and Facebook. It is open source, relatively easy to learn, and works perfectly with MySQL, making it a popular choice for web developers.

**4. Perl**: Perl is a high-level, dynamic programming language used extensively in network programming, system admin, etc. Although less popular for web development purposes, Perl has a lot of niche applications.

Different versions of XAMPP may have additional components such as phpMyAdmin, OpenSSL, etc. to create full-fledged web servers.

**Download XAMPP:**

XAMPP is available in three file formats:

* .EXE – Self-executable file; easiest to install.
* .7z – 7zip file. Favored by purists, although it requires working with more complicated .bat files to install.
* .ZIP – Compressed zip file. Like .7z, installing through .ZIP files is considerably more difficult than using .EXE

Since .EXE is the easiest to install, we will use this file format for this tutorial.

You can download the XAMPP installer from Sourceforge here (102MB).

**Installing XAMPP:**

Follow these steps for installing XAMPP:

**Step 1:** Disable your anti-virus as it can cause some XAMPP components to behave erratically.

**Step 2**: Disable User Account Control (UAC). UAC limits write permissions to XAMPP’s default installation directory (c:/Program Files/xampp), forcing you to install in a separate directory. You can learn how to disable UAC here. (Optional)

**Step 3:** Start the installation process by double-clicking on the XAMPP installer. Click ‘Next’ after the splash screen.

**Step 4**: Here, you can select the components you want to install. Choose the default selection and click ‘Next’.

**Step 5**: Choose the folder you want to install XAMPP in. This folder will hold all your web application files, so make sure to select a drive that has plenty of space.

**Step 6:** The next screen is a promo for BitNami, an app store for server software. Deselect the ‘Learn more about BitNami for XAMPP’ checkbox, unless you actually enjoy receiving promo mails!

**Step 7:** Setup is now ready to install XAMPP. Click Next and wait for the installer to unpack and install selected components. This may take a few minutes. You may be asked to approve Firewall access to certain components (such as Apache) during the installation process.

**Step 8:** Installation is now complete! Select the ‘Do you want to start the Control Panel now?’ checkbox to open the XAMPP control panel.

**XAMPP Control Panel:**

The XAMPP control panel gives you complete control over all installed XAMPP components. You can use the CP to start/stop different modules, launch the Unix shell, open Windows explorer and see all operations running in the background.

Here is a quick overview of the Control Panel. For now, you only need to know how to start and stop an Apache server.

**Test Your PHP and MySql:**

Follow these steps to test your XAMPP installation by launching the Apache web server and creating a simple PHP file.

**Step 1:** In the XAMPP control panel, click on ‘Start’ under ‘Actions’ for the Apache module. This instructs XAMPP to start the Apache webserver.

**Step 2:** Open your web browser and type in: http://localhost or 127.0.0.1

**Step 3:** Select your language from the splash screen.

**Step 4:** You should see the following screen. This means you’ve successfully installed XAMPP on your computer.

**Step 5:** We will now test whether XAMPP has installed PHP successfully. To do this, fire up Notepad and type the following into a new document:

<?php

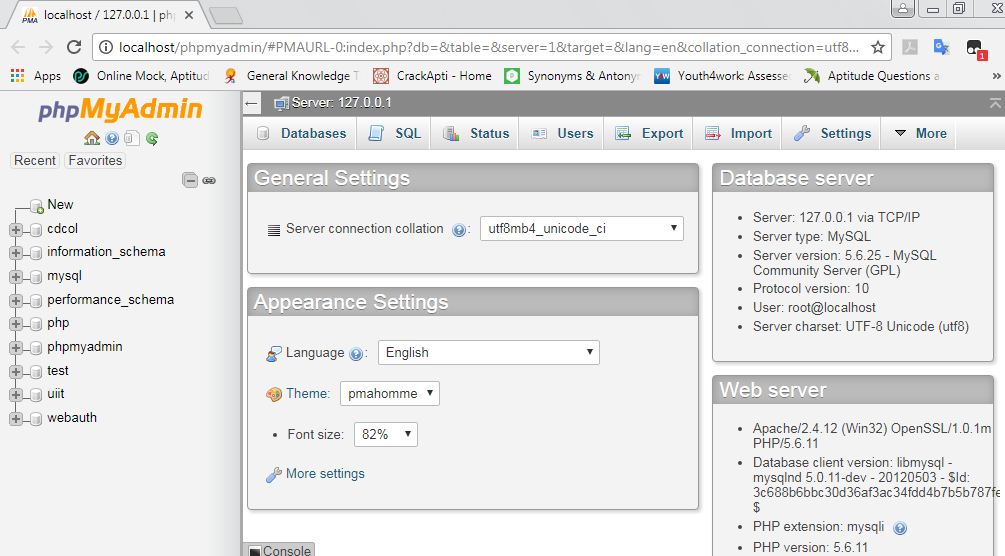
   echo ‘Hello world’;

?>

Save this file as ‘test.php’ in c:\xampp\htdocs\ (or whichever directory you installed XAMPP in).

**Step 6**: Navigate to localhost/test.php. You should see the “Hello World” message.

Now your XAMPP is completely working.



**8 Project Report**

**Student Online registration System**

The Online Student Registration System is an web based portal developed in PHP. It can be used by educational institutes or colleges to maintain the records of students easily. It also provides a less time consuming process for viewing, adding, editing and deleting the marks of the students. Online Registration system will allow online submission of student application. Achieving this objective is difficult using a manual system as the information is scattered, can be redundant and collecting relevant information may be very time consuming. Online Student Registration System Project consists of two functional elements: an enhanced Student module for Registration, Edit Profile, View Test history, Change Password. The project provides facilities like online registration and profile creation of students thus reducing paperwork and automating the record generation process in an educational institution.

**8.1 Proposed Online Student Registration System:**

Student Management System is software which is helpful for students as well as the school authorities. Proposed online student registration system will eliminate all the manual intervention and increase the speed of whole process. In our proposed system we have the provision for adding the details of the students by themselves. So the overhead of the school authorities and the teachers is become less. The objective of Student information System is to allow the administrator of any organization to edit and find out the personal details of a student and allows the student to keep up to date his profile. This student information management system project mainly explains the various actions related to student details. System will allow student to fill the form online, system has inbuilt validation system to validate the entered data. It’ll also facilitate keeping all the records of students, such as their id, name, mailing address, phone number, DOB etc. So all the information about an student will be available in a few seconds. After successful submission, system will give unique registration no for each student.

**8.2 Present student registration system:**

In the current system all the activities are done manually. Currently student submits hardcopy of filled application form to the college/university, office staff enters all data into excel file and write same in manual register. Issue hall ticket / admit card to student and conduct test for student. In the current system we need to keep a number of records related to the student and want to enter the details of the student and the marks manually. It is very time consuming and costly. Office staff check all the paper calculate results and type all the details for taking printout and display it in college premises. College inform student to check the result, student comes to college and view the results. In this student information management system only the teacher or the school authority views the mark of the student and they want to enter the details of the student. In present Student information System, managing and maintaining the details of the student is a tedious job for any organization. This student information management system project is time consuming and

has much cost.

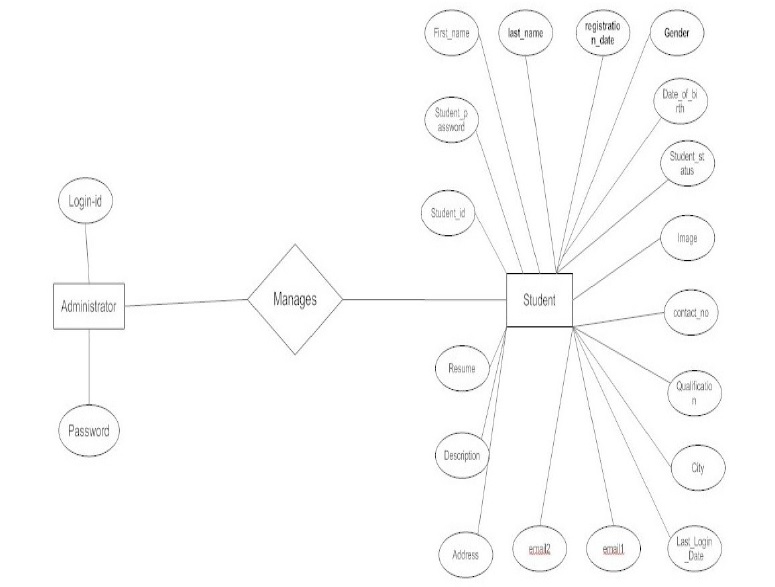
There are two different users who will be using this product:  
1.Administrator   
2. Student

Administrator can view and edit the details of any students. Admin can add new users and he can edit and delete a user.  
Students can view their profile details and they can edit their details. A student can register as user and can add edit and delete his profile. The features that are available to the Administrator are: Admin user have to login into the system to use the system. Admin can enable or disable student account. He can also edit student information in database. He can also search any student in the system by entering his name or username.

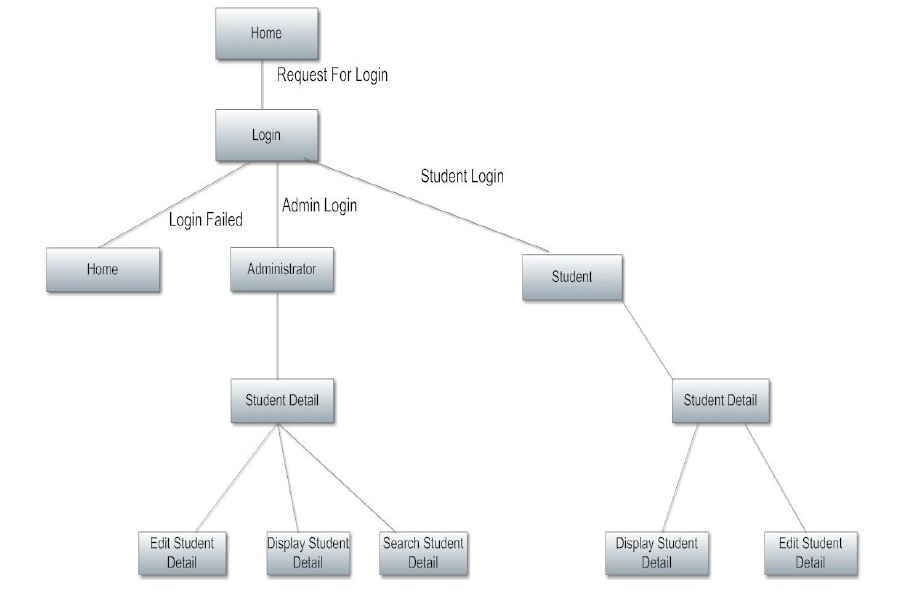
**8.3 Modules of the online student registration system**

Following are the main module of online student registration system.

**Login module:**  
Login module will help in authentication of user accounts. Admin or student should have valid login id and password to login into the system.  
  
**Search module:**  
  
This module is useful to search any student information. User can enter student name and system will give all the details of this system. In manual system this task is very time consuming and tedious but in proposed student registration system, you can do it within few seconds. So using this module administrator can easily search the student by specifying the name of the student in the search criteria.   
  
**Registration Module**  
  
This module help student to get registration. Using this system user can get registered by filling up the registration form online. After registration student can also edit the personal information.  
  
**User Management:**  
  
Only admin can access this module. Admin can enable or disable a student account from this module. He can also edit student information from this module.

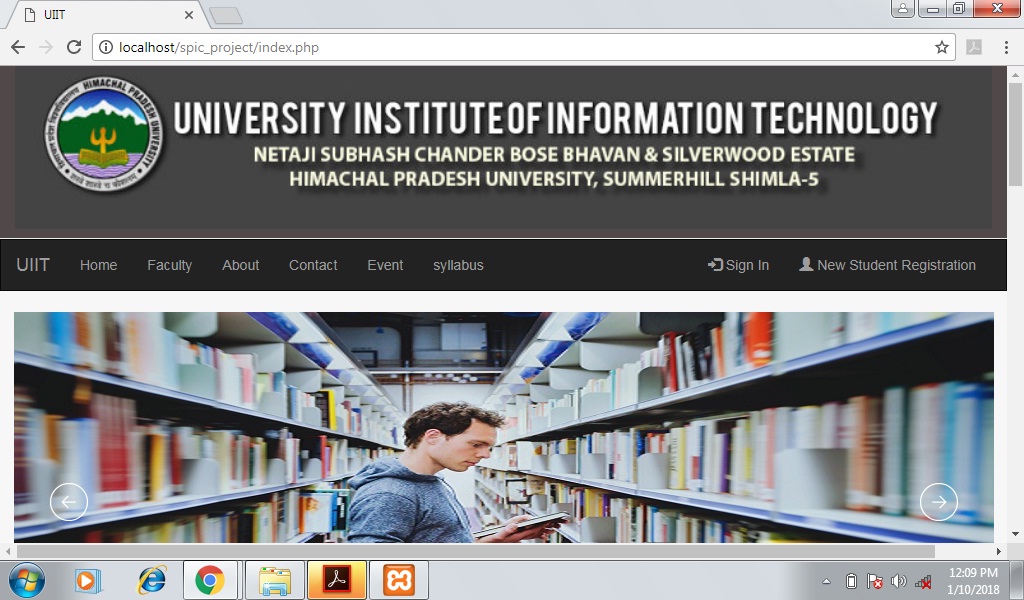
**8.4 ER DIAGRAM OF ONLINE STUDENT REGISTRATION  
**

**8.5 DECISION TREE DIAGRAM OF ONLINE STUDENT REGISTRATION SYSTEM:**

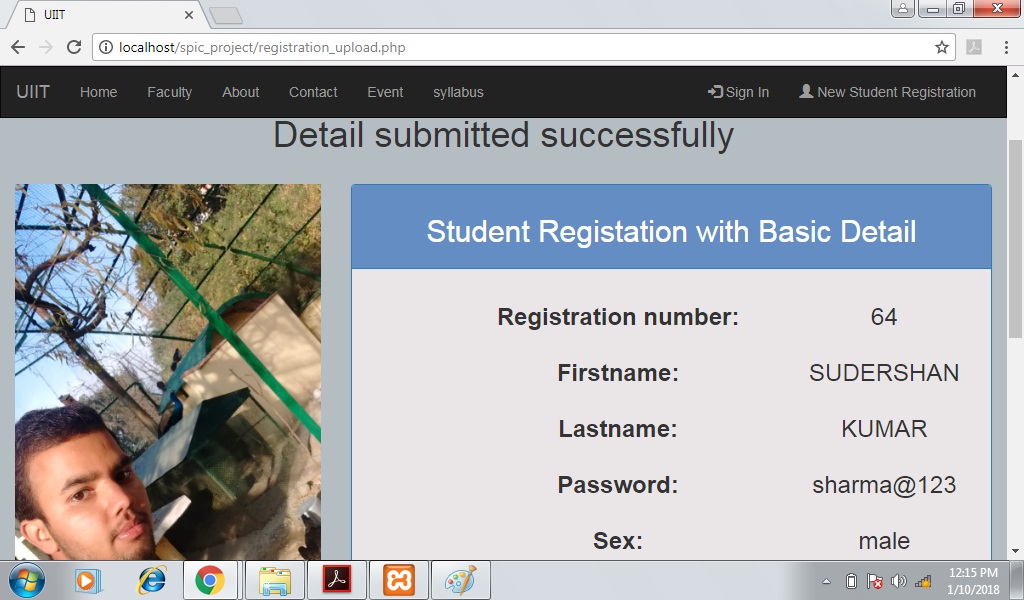
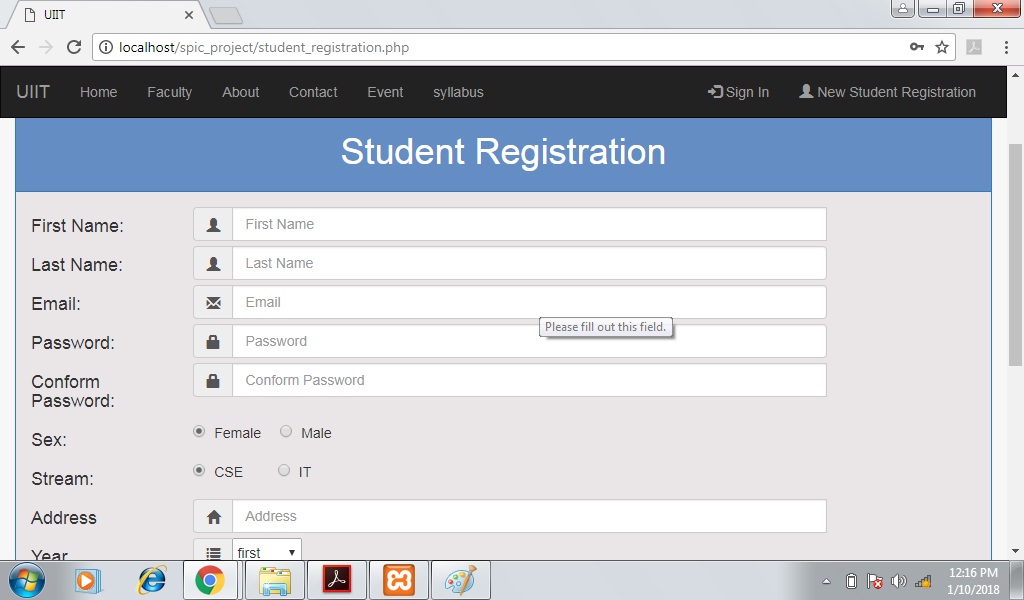
****

**SNAPSHOT OF PROJECT**

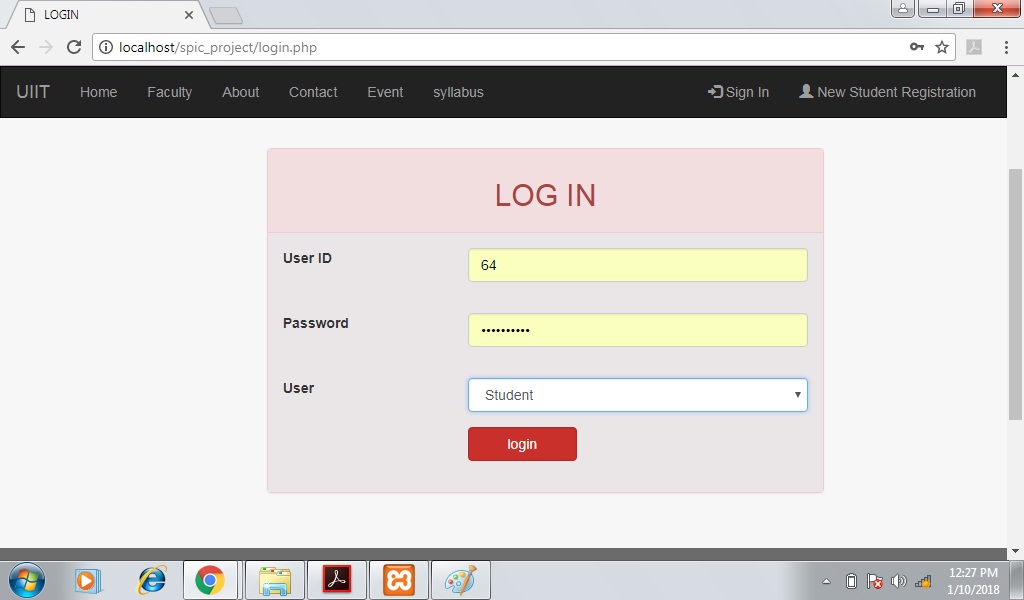
INDEX PAGE OF WEBSITE:

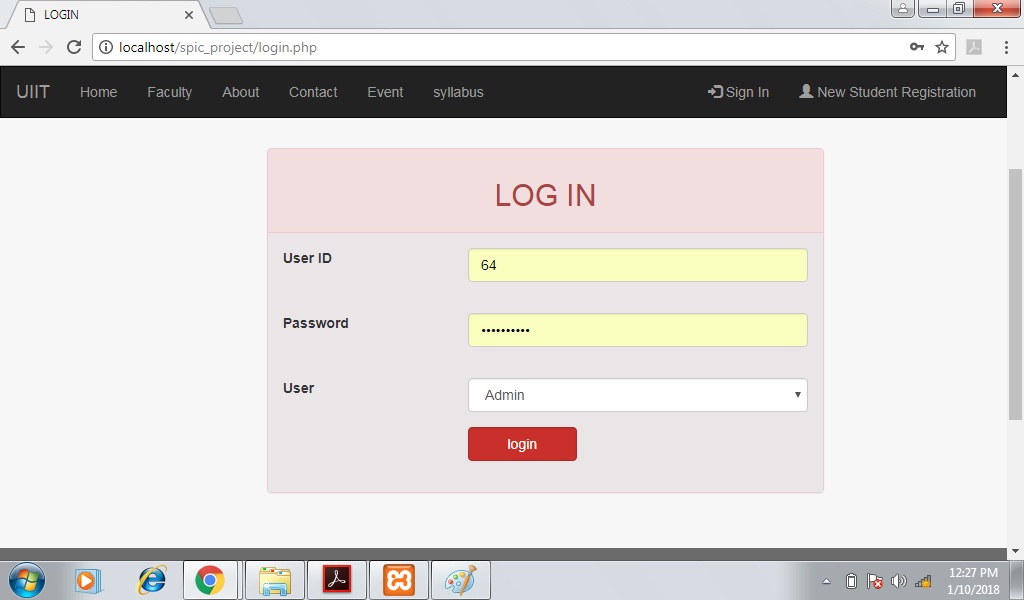
****

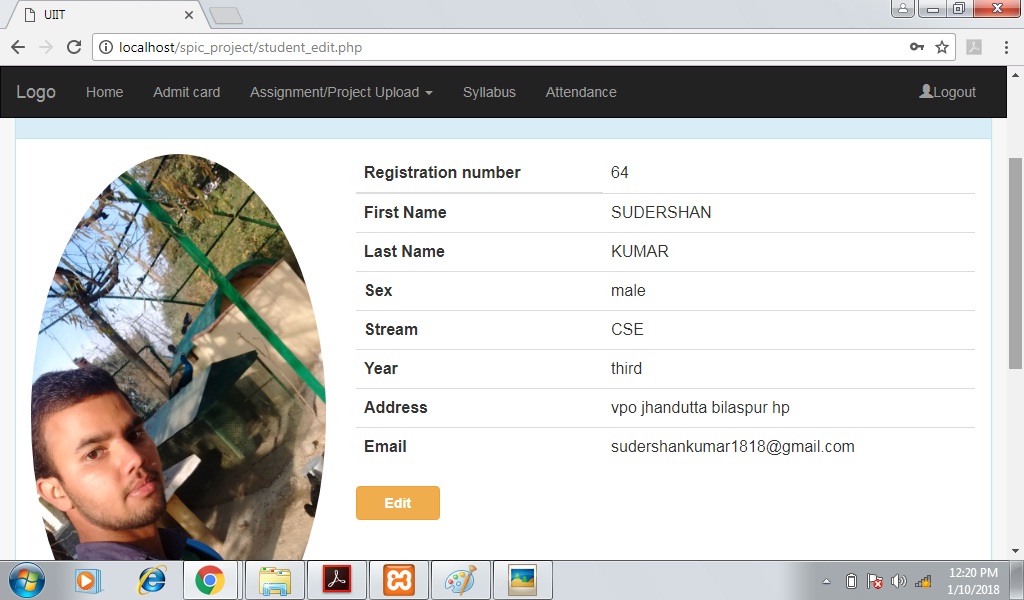
STUDENT REGISTRATION PAGE:



LOGIN PAGE: STUDENT AND ADMIN



Student Index:



Admin index:

