A PROJECT REPORT

On

E-Learning

Submitted in partial fulfillment of the requirement of
University of Mumbai for

Internet Programming Mini Project
In
Information Technology

Submitted By
Nishita Jagdale
Bharti Jagtap
Nishith Dubey
Nitish Palanivelu

Department Of Information Technology

PILLAI COLLEGE OF ENGINEERING

New Panvel – 410 206

UNIVERSITY OF MUMBAI

Academic Year 2020 – 21

DEPARTMENT OF INFORMATION TECHNOLOGY
Pillai College of Engineering
New Panvel – 410 206

CERTIFICATE

This is to certify that the requirements for the report entitled '**Project Title**' have been successfully completed by the following students:

Name	**Roll No.**
Nishita Jagdale	ITA564
Bharti Jagtap	ITA565
Nishith Dubey	ITA562
Nitish Palanivelu	ITA566

in partial fulfillment of Internet Programming mini Project in the Department of Information Technology, Pillai College of Engineering, New Panvel – 410 206 during the Academic Year 2019 – 2020.

Prof. Prerna Kulkarni
Subject Incharge

Department of information technology
Pillai College of Engineering
New Panvel – 410 206

PROJECT APPROVAL FOR

This project entitled “E-Learning” by Nishita Jagdale, Bharti Jagtap, Nishith dubey, and Nitish Palanivelu are approved for the degree of Bachelor of Engineering in Information Technology.

Examiners:

1. ________________

2. ________________

3. ________________

4. ________________

5. ________________

Date:

DEPARTMENT OF INFORMATION TECHNOLOGY
Pillai College of Engineering
New Panvel – 410 206

DECLARATION

We declare that this written submission for Internet Programming Mini Project entitled "Project Title" represent our ideas in our own words and where others' ideas or words have been included. We have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any ideas / data / fact / source in our submission. We understand that any violation of the above will cause for disciplinary action by institute and also evoke penal action from the sources which have not been properly cited or from whom prior permission have not been taken when needed.

Project Group Members:

Nishita Jagdale

____

Bharti Jagtap

____

Nishith Dubey

____

Nitish Palanivelu

____

Table of Contents

Abstract

E-learning is an integral part of smart education. There are many e-learning systems that are widely available to educational institutions. The challenge is to easily integrate the e-learning system into a smart educational environment based on the requirements of the users. The provided application rely on a software system that allows access to all the materials for the educational process and makes them electronically available to all the students on the Internet whenever they need and wherever they are. The design and development of this application, known as "E-Vidya" is a critical part of the educational process as it reflects on the usage of the system. In this work, the design and implementation of this systems is described where different techniques are explored and compared. The proposed e-learning system is designed using off-the-shelf and open-source software engineering models and programming tools and database models. The system is tested to prove the new design concepts and features. The method used in the back-end and front-end design and implementation allows flexible usage and integration of the e-learning systems by the educational institutions in smart cities.

Chapter 1
Introduction

E-learning is a medium for engaging learners in an online training course. E-learning courses can exist in many forms, using different types of technologies. Commonly used in organizations and corporations, e-learning can help learners complete training and education objectives with ease and flexibility as compared to traditional classroom-based learning. E-learning has worked towards bringing learners, tutors, experts, practitioners, and other interest groups to one place. Thus, there is a good practice of knowledge sharing followed through different online platforms. This is important in current times as competition is rising and the world is also growing. Hence, quick information helps in the better growth of an individual.

The E-learning system, known as "E-Vidya" has been developed to override the problems prevailing in the practicing manual system. This application is designed to eliminate and in some cases reduce the hardships faced by the existing system. Moreover, this system is supported for the particular need of the company to carry out operations in smooth and effective manner.

The proposed system consist of majorly two different domains. The admin side and the client side. Admin side is all about managing the backend by keeping a record of and updating the courses and the videos to be displayed to the client. The client side is the one who can log in or register the application to have a overview regarding the courses that are made being available for them. The two modules, i. e. the courses and the video page will direct them to the desired area.

The contact us and about us page will direct the user incase of any of the queries. This system is been designed to adapt the managerial requirements. It assist in strategic planning, and right level of information which allow to have better management for the resources.

Chapter 2

Requirement Analysis

E-learning utilizes a range of diverse technologies in addition to the fundamental needs of a computer and an Internet connection. Some of them are utilized to provide extra support and features for e-learning, while others are built and intended exclusively for e-learning courses.

1. Functional requirements
2. Non-functional requirements

1. Functional requirements

In this E-Learning system you will be able to add a student in a class, upload a file, a course, department, subject and video. This project will contain a lot of advanced modules which makes the backend system very powerful. The list of requirements for this project was also influenced by the work done in the background research section above.

USER	**FUNCTIONAL REQUIREMENTS**
Admin	● Login for Admin ● Forgot password for Admin ● Edit Profile for Admin ● Change password for Admin ● Logout functionality ● Dashboard for Admin User ➔ MANAGE DEPARTMENTS: Add a new department, edit the existing department, view details of the department and see a listing of all departments ➔ MANAGE EVENT: Add a new event, edit the existing event, view details of events and see a listing of all events ➔ MANAGE COURSES: Add a new course, edit the existing course, viewing the details of the course. Adding the necessary notes. ➔ MANAGE VIDEOS: Add a new video, updating the videos accordingly based on the requirements. ➔ MANAGE STUDENT: Add new student, edit the existing student, view details of the student and see a listing of all students

Student	➔ STUDENT REGISTRATION: Any Students can register on the website using the registration module ➔ STUDENT LOGIN: This is the login form from where students can login to the system ➔ STUDENT DASHBOARD: The student can view his/her details and the ongoing courses. ➔ CHANGE PASSWORD: A student can change his account password from this module.

2. Non-functional requirements

System qualities including security, reliability, performance, maintainability, scalability, and usability are defined by nonfunctional requirements (NFRs). They act as limitations or restrictions on how the system is designed for the various backlogs.

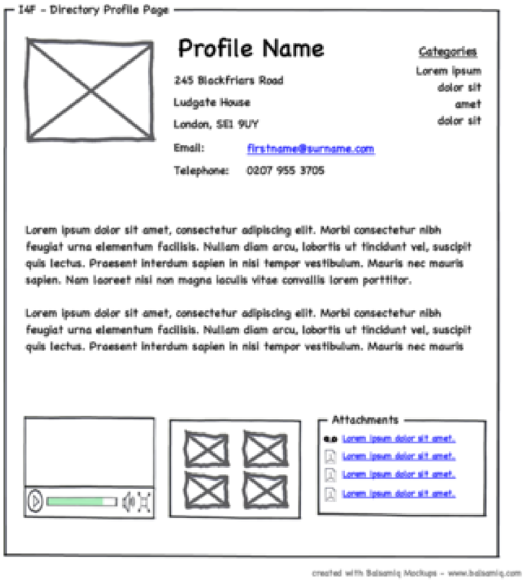
NON-FUNCTIONAL REQUIREMENT	**OBJECTIVE**
Application Security	The system should be protected in such a manner that one registered user should not be able to access another registered user's information ensuring privacy of information.
Database Security	Users of the system should not have direct access to the database to query it nor view data in it. The only access to the database should be via the application interface.
Browser Compatibility	The application should be accessible on Google Chrome, Mozilla Firefox and Internet Explorer browsers on any device.
Maintainability	The application should be developed so that one can easily add new products and easily facilitate changes to product information
Consistency	The appearance and delivery of the content should be consistent to reduce the learning curve. Layouts, buttons and the positioning of key elements should be consistent in each page
Usability	Application is expected to be user friendly so that it can

	be used in an efficient manner.
Scalability	Number of users supported will mainly depend on the server load, server processing capacity and its memory. It should scale maximum number of users
Availability	24 X 7 availability should be there so that student can use it at any time according to his/her convenience

Chapter 3

Wireframe

A website wireframe, also known as a page schematic or screen blueprint, is a visual guide that represents the skeletal framework of a website. Wireframes are created for the purpose of arranging elements to best accomplish a particular purpose. The purpose is usually being informed by a business objective and a creative idea. The wireframe depicts the page layout or arrangement of the website's content, including interface elements and navigational systems, and how they work together. The wireframe usually lacks typographic style, color, or graphics, since the main focus lies in functionality, behavior, and priority of content. In other words, it focuses on what a screen does, not what it looks like. Wireframes can be pencil drawings or sketches on a whiteboard, or they can be produced by means of a broad array of free or commercial software applications. Wireframes are generally created by business analysts, user experience designers, developers, visual designers, and by those with expertise in interaction design, information architecture and user research.



I4F - Directory Profile Page

Profile Name

245 Blackfriars Road
Ludgate House
London, SE1 9UY

Email: firstname@surname.com
Telephone: 0207 955 3705

Categories
Lorem ipsum
dolor sit
amet
dolor sit

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi consectetur nibh feugiat urna elementum facilisis. Nullam diam arcu, lobortis ut tincidunt vel, suscipit quis lectus. Praesent interdum sapien in nisi tempor vestibulum. Mauris nec mauris sapien. Nam laoreet nisi non magna iaculis vitae convallis lorem porttitor.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi consectetur nibh feugiat urna elementum facilisis. Nullam diam arcu, lobortis ut tincidunt vel, suscipit quis lectus. Praesent interdum sapien in nisi tempor vestibulum. Mauris nec mauris

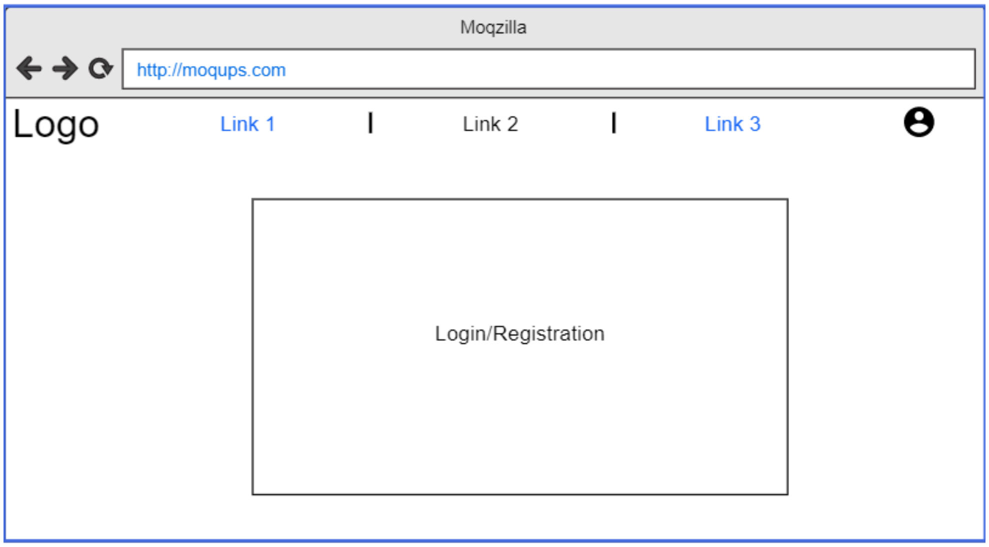
Attachments
- Lorem ipsum dolor sit amet.
- Lorem ipsum dolor sit amet.
- Lorem ipsum dolor sit amet.
- Lorem ipsum dolor sit amet.

created with Balsamiq Mockups - www.balsamiq.com

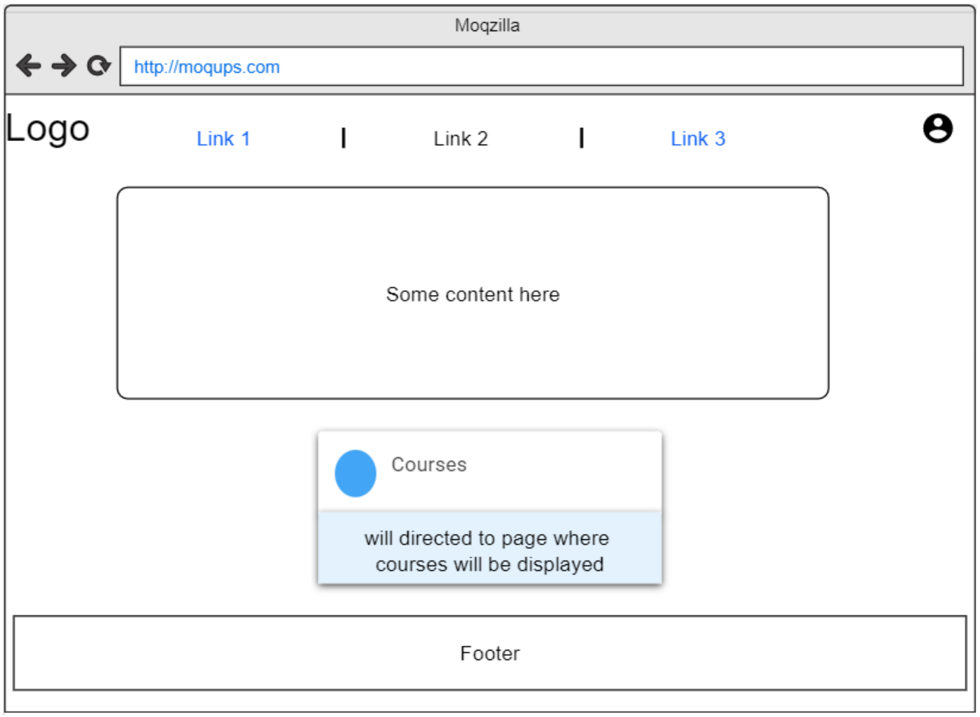
Wireframes focus on:

1. The range of functions available
2. The relative priorities of the information and functions
3. The rules for displaying certain kinds of information
4. The effect of different scenarios on the display

Wireframe 1


Moqzilla
http://moqups.com
Logo
Link 1
Link 2
Link 3
Login/Registration

Wireframe 2


Moqzilla
http://moqups.com
Logo
Link 1
Link 2
Link 3
Some content here
Courses
will directed to page where courses will be displayed
Footer

Chapter 4

Using different types of CSS

Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g., fonts, colors, spacing) to Web documents. CSS is a language that describes the style of an HTML document. CSS describes how HTML elements should be displayed.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content also makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable. The CSS specifications are maintained by the World Wide Web Consortium (W3C).

Syntax of CSS

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts :

Selector − A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.

Property - A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be color, border etc.

Value - Values are assigned to properties. For example, color property can have value either red or #F1F1F1 etc.

selector { property: value } selector { property: value, property: value } h1 {‘color’:’blue’}

Types of css selectors:

- The element Selector :h1 {color: red;}
- The id Selector / type selector :#para1 {text-align: center;color: red;}
- The Descendant Selectors : ul li {color: red;}
- The class Selector : .center {text-align: center;color: red;}
- The Attribute Selectors : input[type = "text"]{color: #000000; }
- The Child Selectors : body > p {color: #000000; }
- The Universal Selectors : * {color: red;}
- The Adjacent Sibling Selector : H2+P {color: red;}

There are three ways of inserting a style sheet:

1. External style sheet
2. Internal style sheet
3. Inline style

External style sheet

The <link> element can be used to include an external stylesheet file in your HTML document. An external style sheet is a separate text file with .css extension. You define all the Style rules within this text file and then you can include this file in any HTML document using <link> element.

Here is the generic syntax of including external CSS file −

```
<head>
  <link type = "text/css" href = "..." media = "..." />
</head>
```

Consider a simple style sheet file with a name mystyle.css having the following rules −

```
h1, h2, h3 {
  color: #36C;
  font-weight: normal;
  letter-spacing: .4em;
  margin-bottom: 1em;
  text-transform: lowercase;
}
```

Now you can include this file mystyle.css in any HTML document as follows −

```
<head>
  <link type = "text/css" href = "mystyle.css" media = " all" />
</head>
```

Internal style sheet

You can put your CSS rules into an HTML document using the <style> element. This tag is placed inside <head>...</head> tags. Rules defined using this syntax will be applied to all the elements available in the document

<style> Attribute type =”text/css” <style type=”text/css” > </style>

Specifies the style sheet language as a content-type (MIME type). This is required attribute.

media attribute <style type = "text/css" media = "all">

```
<head>
<style type=”text/css” >
body {
   background-color: linen;
}
h1 {
   color: maroon;
   margin-left: 40px;
}
</style>
</head>
```

Inline style

An inline style may be used to apply a unique style for a single element. To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

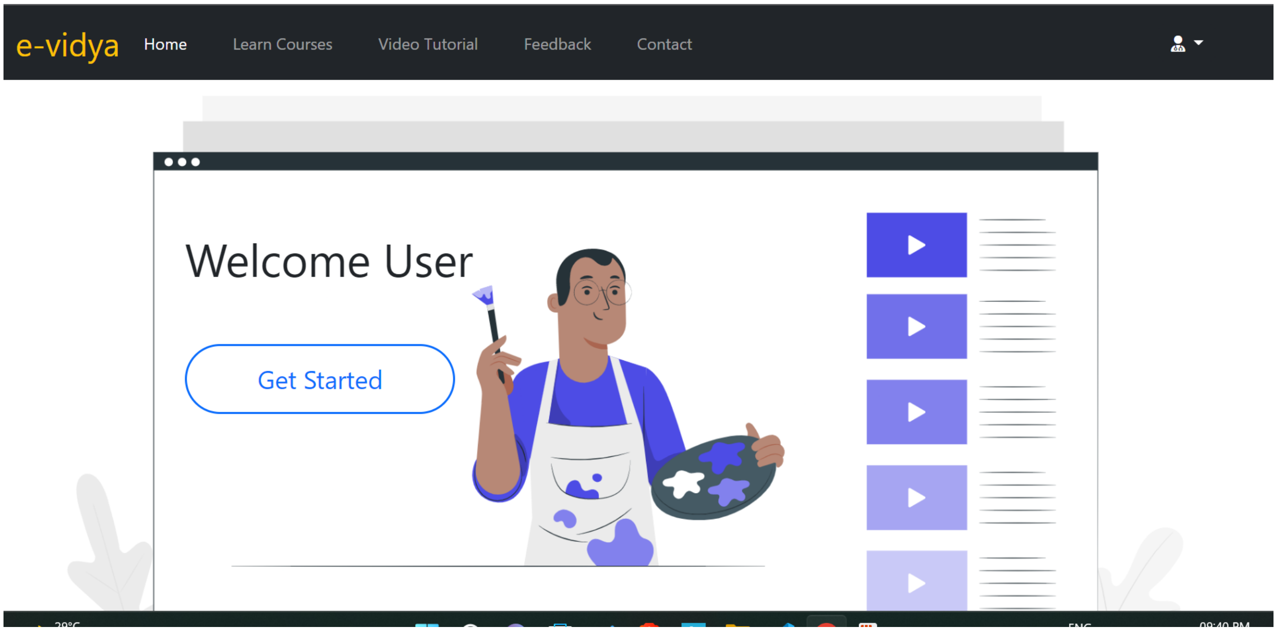
Syntax:

<element style = "...style rules....">

Attributes style “style rules”

The value of style attribute is a combination of style declarations separated by semicolon (;)

<h1 style = "color:#36C;">



Chapter 5

Responsive design using media queries

Media queries in CSS3 look at the capability of the device. Media queries can be used to check many things, such as:

- width and height of the viewport
- width and height of the device
- orientation (is the tablet/phone in landscape or portrait mode?)
- resolution

Using media queries are a popular technique for delivering a tailored style sheet to desktops, laptops, tablets, and mobile phones (such as iPhone and Android phones). Media queries are used for creating responsive web sites . You can also use media queries to specify that certain styles are only for printed documents or for screen readers (mediatype: print, screen, or speech).

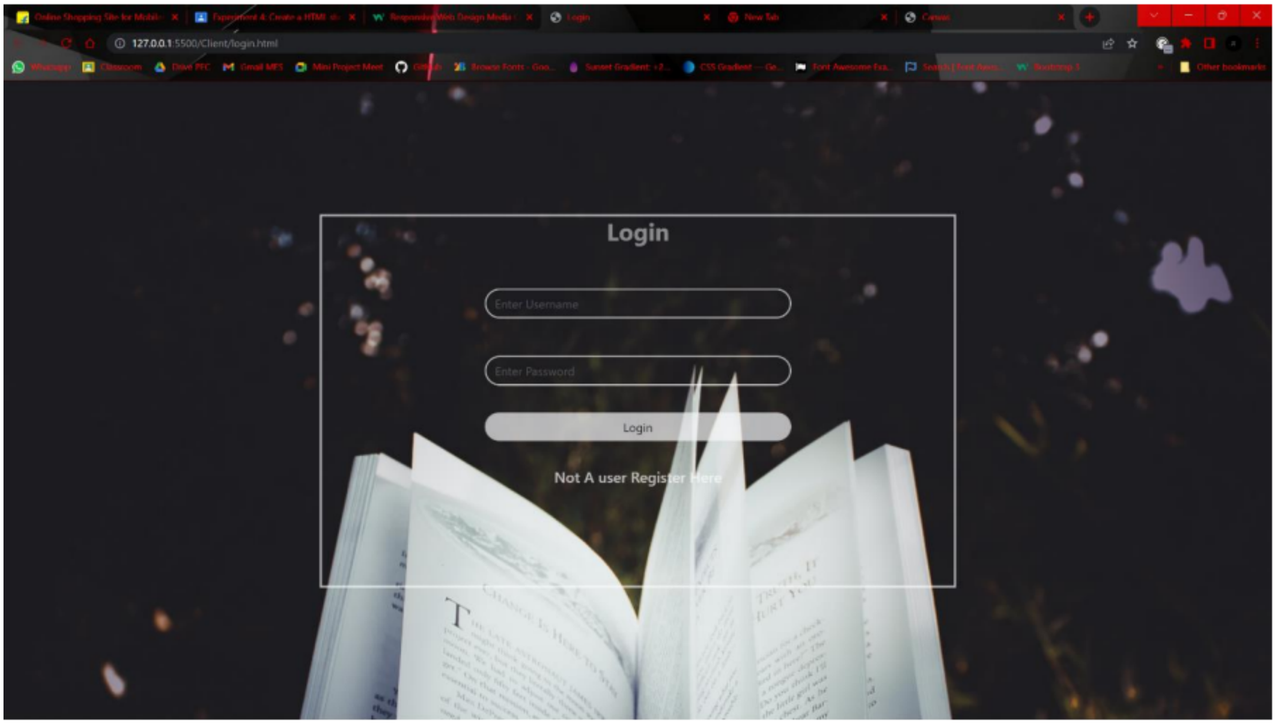
In addition to media types, there are also media features. Media features provide more specific details to media queries, by allowing to test for a specific feature of the user agent or display device. For example, you can apply styles to only those screens that are greater, or smaller, than a certain width.

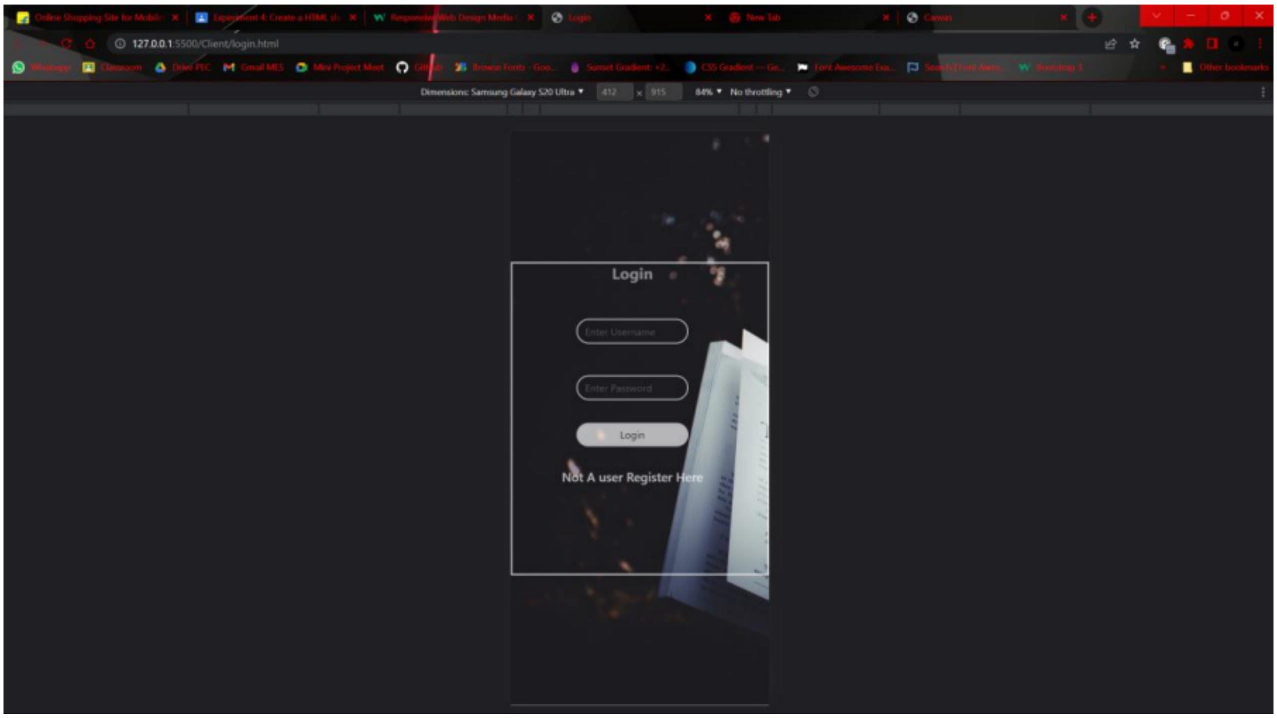
A media query consists of a media type and can contain one or more expressions, which resolve to either true or false.

```
@media not|only mediatype and (expressions) {

   CSS-Code;

}
```

The result of the query is true if the specified media type matches the type of device the document is being displayed on and all expressions in the media query are true. When a media query is true, the corresponding style sheet or style rules are applied, following the normal cascading rules. Unless you use the not or only operators, the media type is optional and the all type will be implied The following example changes the background-color to light green if the viewport is 480 pixels wide or wider (if the viewport is less than 480 pixels, the background-color will be pink):

```
@media screen and (min-width: 480px) {
    body {
    background-color: lightgreen;
    }
}
```





127.0.0.1:5500/Client/login.html

Dimensions: Samsung Galaxy S20 Ultra ▾ 412 x 915 84% ▾ No throttling ▾

Login

Enter Username

Enter Password

Login

Not A user Register Here

Chapter 6

Embedding Google Maps in web page

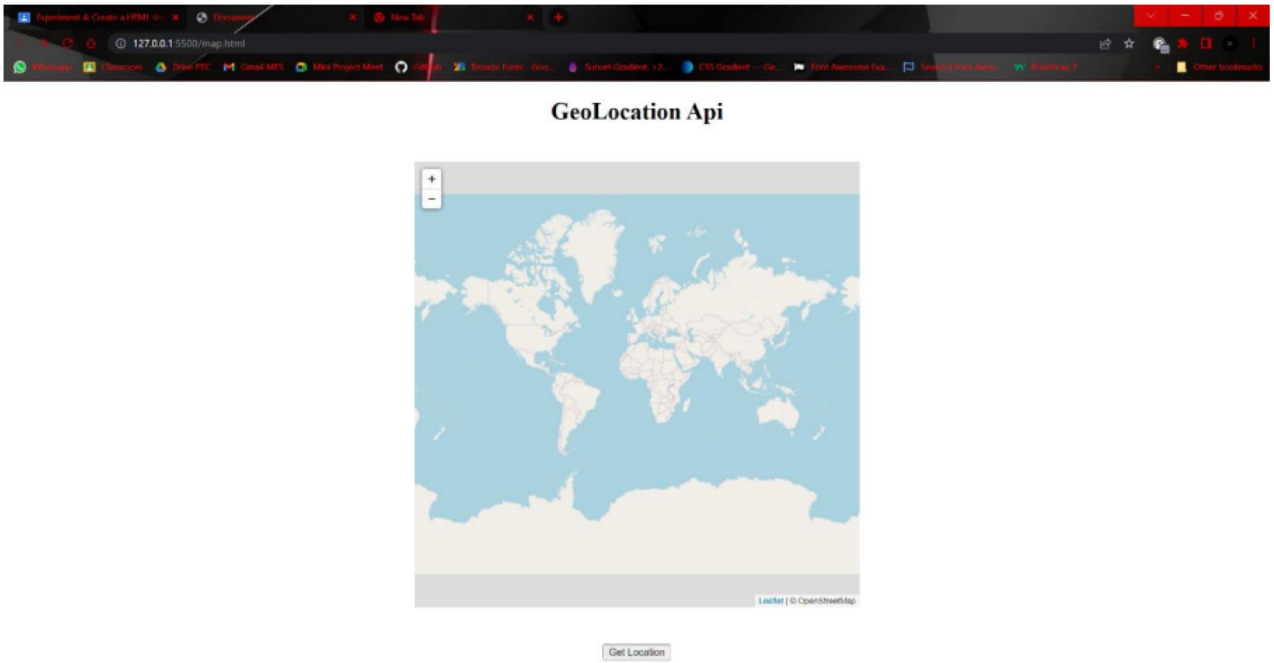
Google Maps is a web mapping service developed by Google. It offers satellite imagery, street maps, 360° panoramic views of streets (Street View), real-time traffic conditions (Google Traffic), and route planning for traveling by foot, car, bicycle (in beta), or public transportation. Google Maps API in June 2005[19] to allow developers to integrate Google Maps into their websites. It was a free service that didn't require an API key until June 2018 (changes went into effect on July 16), when it was announced that an API key linked to a Google Cloud account with billing enabled would be required to access the API. The API currently does not contain ads, but Google states in their terms of use that they reserve the right to display ads in the future.

By using the Google Maps API, it is possible to embed Google Maps into an external website, on to which site-specific data can be overlaid. Although initially only a JavaScript API, the Maps API was expanded to include an API for Adobe Flash applications (but this has been deprecated), a service for retrieving static map images, and web services for performing geocoding, generating driving directions, and obtaining elevation profiles. Over 1,000,000 web sites use the Google Maps API, making it the most heavily used web application development API.

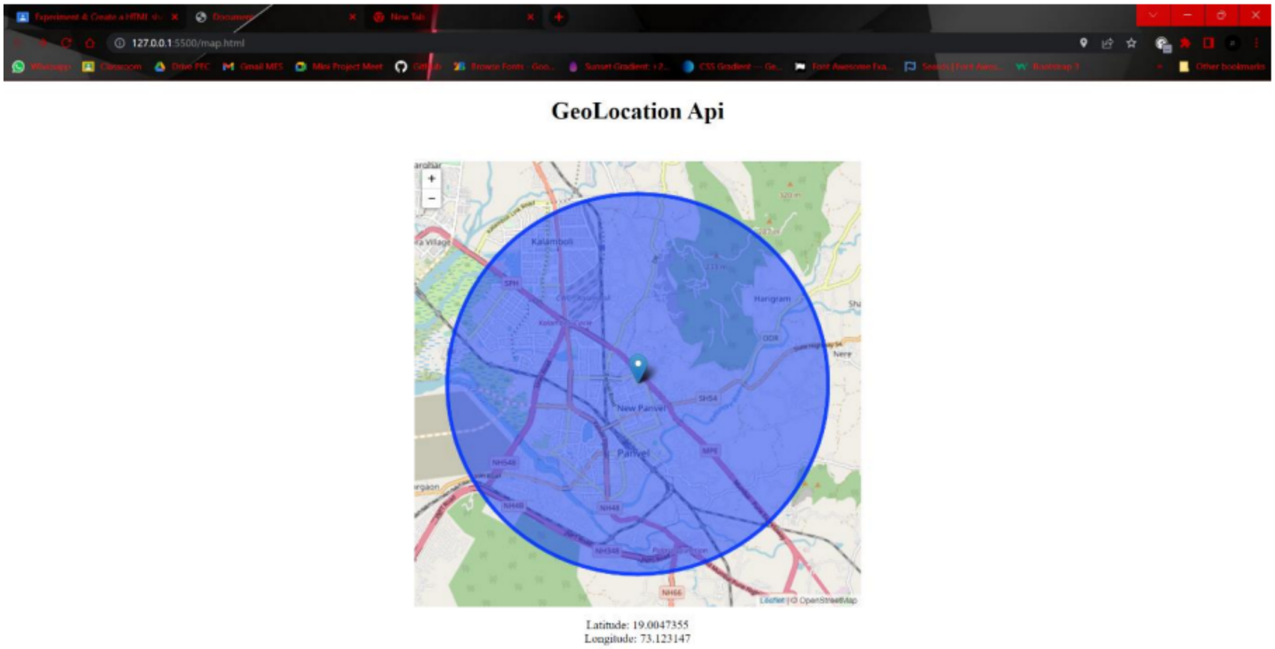
The Google Maps API is free for commercial use, provided that the site on which it is being used is publicly accessible and does not charge for access, and is not generating more than 25,000 map accesses a day. Sites that do not meet these requirements can purchase the Google Maps API for Business.

The success of the Google Maps API has spawned a number of competing alternatives, including the HERE Maps API, Bing Maps Platform, Leaflet and OpenLayers via self-hosting. The Yahoo! Maps API is in the process of being shut down.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
       /* Set the size of the div element that contains the map */
      #map {
        height: 400px;  /* The height is 400 pixels */
      width: 100%;  /* The width is the width of the web page */
      }
    </style>
  </head>
  <body>
    <h3>My Google Maps Demo</h3>
    <!--The div element for the map -->
    <div id="map"></div>
    <script>
// Initialize and add the map
function initMap() {
  // The location of Uluru
  var uluru = {lat: -25.344, lng: 131.036};
  // The map, centered at Uluru
  var map = new google.maps.Map(
      document.getElementById('map'), {zoom: 4, center: uluru});
  // The marker, positioned at Uluru
  var marker = new google.maps.Marker({position: uluru, map: map});
}
    </script>
    <!--Load the API from the specified URL
    * The async attribute allows the browser to render the page while the API loads
    * The key parameter will contain your own API key (which is not needed for this tutorial)
    * The callback parameter executes the initMap() function
    -->
    <script async defer
    src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap">
    </script>
  </body>
</html>
```



GeoLocation Api



GeoLocation Api

Latitude: 19.0047355
Longitude: 73.123147

Chapter 7

HTML5 based form validation

Forms are used in webpages for the user to enter their required details that are further send it to the server for processing. A form is also known as web form or HTML form. Form validation helps us to ensure that users fill out forms in the correct format, making sure that submitted data will work successfully with our applications.

Go to any popular site with a registration form, and you will notice that they give you feedback when you don't enter your data in the format they are expecting. You'll get messages such as:

"This field is required" (you can't leave this field blank)

"Please enter your phone number in the format xxx-xxxx" (it enforces three numbers followed by a dash, followed by four numbers)

"Please enter a valid e-mail address" (if your entry is not in the format of "somebody@example.com")

"Your password needs to be between 8 and 30 characters long, and contain one uppercase letter, one symbol, and a number"

This is called form validation — when you enter data, the web application checks it to see that the data is correct. If correct, the application allows the data to be submitted to the server and (usually) saved in a database; if not, it gives you an error message explaining what corrections need to be made. Form validation can be implemented in a number of different ways.

We want to make filling out web forms as easy as possible. So why do we insist on validating our forms? There are three main reasons:

We want to get the right data, in the right format — our applications won't work properly if our user's data is stored in the incorrect format, or if they don't enter the correct information, or omit information altogether.

We want to protect our users' accounts — by forcing our users to enter secure passwords, it makes it easier to protect their account information.

We want to protect ourselves — there are many ways that malicious users can misuse unprotected forms to damage the application they are part of (see Website security).

Different types of form validation

There are two different types of form validation which you'll encounter on the web:

Client-side validation is validation that occurs in the browser before the data has been submitted to the server. This is more user-friendly than server-side validation as it gives an instant response. This can be further subdivided:

JavaScript validation is coded using JavaScript. It is completely customizable.

Built-in form validation using HTML5 form validation features. This generally does not require JavaScript. Built-in form validation has better performance, but it is not as customizable as JavaScript.

Server-side validation is validation which occurs on the server after the data has been submitted. Server-side code is used to validate the data before it is saved into the database. If the data fails authentication, a response is sent back to the client to tell the user what corrections to make. Server-side validation is not as user-friendly as client-side validation, as it does not provide errors until the entire form has been submitted. However, server-side validation is your application's last line of defence against incorrect or even malicious data. All popular server-side frameworks have features for validating and sanitizing data (making it safe).

1. Specialized Input Types

HTML5 introduced several new input types. They can be used to create input boxes, which will accept only a specified kind of data.

The new input types are as follows:

Color, date, datetime, email , month .number, range . search, tel , time , url , week

To use one of the new types, include them as the value of the type attribute:

<input type="email"/>

2. Required Fields

By simply adding the "required" attribute to a <input>, <select> or <textarea>, you tell the browser that a value must be provided in this field. Think of this as the red asterisk* we see in most registration forms.

```
<input type="checkbox" name="terms" required >
```

3. Limits

We can set some basic limitations like max length and minimum and maximum values for number fields. To limit the length of input fields and textareas, use the "maxlength" attribute. What this does is to forbid any string longer than the field's "maxlength" value to be entered at all. If you try and paste a string witch exceeds this limit, the form will simply clip it.

```
<input type="text" name="name" required  maxlength="15">
```

The <input type="number"> fields use "max" and "min" attributes to create a range of possible values - in our example we've made the minimum allowed age to be 18 (too bad you can be whatever age you want on the internet).

```
<input type="number" name="age" min="18" required>
```

Add your snapshot:

Chapter 8

Javascript based form validation

Validating form input with JavaScript is easy to do and can save a lot of unnecessary calls to the server as all processing is handled by the web browser. It can prevent people from leaving fields blank, from entering too little or too much or from using invalid characters.

Forms validation on the client-side is essential — it saves time and bandwidth, and gives you more options to point out to the user where they've gone wrong in filling out the form. Having said that, I don't mean that you don't need server-side validation. People who visit your site may use an old browser or have JavaScript disabled, which will break client-only validation. Client and server-side validation complement each other, and as such, they really shouldn't be used independently.

Why is Client Side Validation Good?

There are two good reasons to use client-side validation:

1. It's a fast form of validation: if something's wrong, the alarm is triggered upon submission of the form.
2. You can safely display only one error at a time and focus on the wrong field, to help ensure that the user correctly fills in all the details you need.

Two Major Validation Approaches

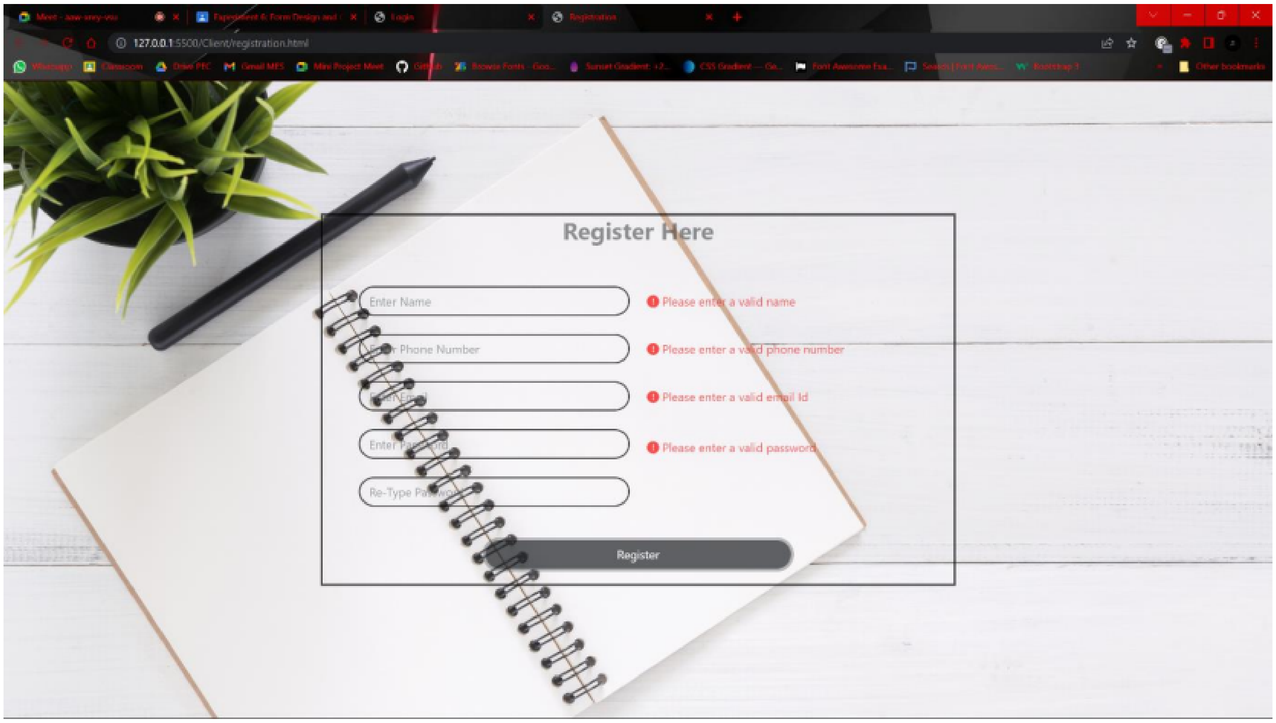
1. Display the errors one by one, focusing on the offending field
2. Display all errors simultaneously, server-side validation style

While displaying all errors simultaneously is required for server-side validation, the better method for validation on the client-side is to show one error at a time. This makes it possible to highlight only the field that has been incorrectly completed, which in turn makes revising and successfully submitting the form much easier for the visitor. If you present users with all errors at the same time, most people will try to remember and correct them at once, instead of attempting to re-submit after each correction.

```
 function validateForm() {
    var x = document.forms["myForm"]["fname"].value;

    if (x == "") {

        alert("Name must be filled out");

        return false;

    }

}

<form name="myForm" action="/action_page.php" onsubmit="return validateForm()" method="post">

Name: <input type="text" name="fname">

<input type="submit" value="Submit">

</form>

function checkpassword(pform1){

var str=pform1.password.value;
//check required fields
//password should be minimum 4 chars but not greater than 8
if ((str.length < 4) || (str.length > 8)) {
function checkpassword(pform1){
var str=pform1.password.value;

//check required fields

//password should be minimum 4 chars but not greater than 8

if ((str.length < 4) || (str.length > 8)) {
alert("Invalid password length.")
pform1.password.focus()
return false
}


}
function checkemailphone(pform1){
var email = pform1.email.value;
var phone = pform1.phone.value;
var cleanstr = phone.replace(/[().- ]/g, ");
var validemail =/^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+.[a-zA-Z]{2,4}$/;
if(!(validemail.test(email))){
alert("Invalid email address")
pform1.email.focus()
return false
}
//check phone number
if (isNaN(parseInt(cleanstr))) {
alert("The phone number contains unwanted characters.")
}
}
```

Registration Page

```
function regVal() {
   const fname = document.getElementById("name").value;
   const phno = document.getElementById("phno").value;
   const email = document.getElementById("email").value;
   const pass = document.getElementById("password").value;
   const nameErr = document.getElementById("nameErr");
   const phErr = document.getElementById("phErr");
   const emailErr = document.getElementById("emailErr");
   const passErr = document.getElementById("passErr");
   if (/^[a-zA-Z ]+$/.test(fname)) {
      nameErr.innerHTML = "<i class='bi bi-check-circle-fill' style='color:green;'></i>";
   }
   else {
      nameErr.innerHTML = "<i class='bi bi-exclamation-circle-fill'></i> Please enter a valid
name";
   }
   if (/^\+?[1-9][0-9]{9}$/.test(phno)) {
      phErr.innerHTML = "<i class='bi bi-check-circle-fill' style='color:green;'></i>"
   }
   else {
      phErr.innerHTML = "<i class='bi bi-exclamation-circle-fill'></i> Please enter a valid
phone number";

   }

   if (/^\+?[1-9][0-9]{9}$/.test(phno)) {
      phErr.innerHTML = "<i class='bi bi-check-circle-fill' style='color:green;'></i>"
   }
   else {
      phErr.innerHTML = "<i class='bi bi-exclamation-circle-fill'></i> Please enter a valid
phone number";

   }
   if (/^\S+@\S+\.\S+$/.test(email)) {
      emailErr.innerHTML = "<i class='bi bi-check-circle-fill' style='color:green;'></i>";
   }
   else {
        emailErr.innerHTML = "<i class='bi bi-exclamation-circle-fill'></i> Please enter a valid
email Id";
    }

    if (pass == "" || pass.length <= 10) {
        passErr.innerIITML = "<i class='bi bi-exclamation-circle-fill'></i> Please enter a valid
password";

    }
    else {
        passErr.innerHTML = "<i class='bi bi-check-circle-fill' style='color:green;'></i>";
        rpassCheck(pass);
    }

}
function rpassCheck(pass) {
    const rpassword = document.getElementById("rpassword").value;
    const rpassErr = document.getElementById("rpassErr");
    if (rpassword == pass) {
        rpassErr.innerHTML = "<i class='bi bi-check-circle-fill' style='color:green;'></i>"
    }
    else {
        rpassErr.innerHTML = "<i class='bi bi-exclamation-circle-fill'></i> The passwords do
not match"
    }
}
```



Please enter a valid name
Please enter a valid phone number
Please enter a valid email Id
Please enter a valid password

Chapter 9

Server side programming using PHP

PHP stands for Hypertext Preprocessor. PHP is a very popular and widely-used open source server-side scripting language to write dynamically generated web pages. PHP was originally created by Rasmus Lerdorf in 1994. It was initially known as Personal Home Page.

PHP scripts are executed on the server and the result is sent to the web browser as plain HTML. PHP can be integrated with the number of popular databases, including MySQL, PostgreSQL, Oracle, Microsoft SQL Server, Sybase, and so on.

What You Can Do with PHP

- There are lot more things you can do with PHP.
- You can generate pages and files dynamically.
- You can create, open, read, write and close files on the server.
- You can collect data from a web form such as user information, email, phone no, etc.
- You can send emails to the users of your website.
- You can send and receive cookies to track the visitor of your website.
- You can store, delete, and modify information in your database.
- You can restrict unauthorized access to your website.
- You can encrypt data for safe transmission over internet.
- The list does not end here, there are many other interesting things that you can do with PHP. You will learn about all of them in detail in upcoming chapters.

Advantages of PHP over Other Languages

- If you're familiar with other server-side languages like ASP.NET or Java, you might be wondering what makes PHP so special. There are several advantages why one should choose PHP.
- Easy to learn: PHP is easy to learn and use. For beginner programmers who just started out in web development, PHP is often considered as the preferable choice of language to learn.
- Open source: PHP is an open-source project. It is developed and maintained by a worldwide community of developers who make its source code freely available to download and use.

- Portability: PHP runs on various platforms such as Microsoft Windows, Linux, Mac OS, etc. and it is compatible with almost all servers used today such Apache, IIS, etc.
- Fast Performance: Scripts written in PHP usually execute or runs faster than those written in other scripting languages like ASP, Ruby, Python, Java, etc.
- Vast Community: Since PHP is supported by the worldwide community, finding help or documentation related to PHP online is extremely easy.

Setting Up a Local Web Server

PHP script execute on a web server running PHP. So before you start writing any PHP program you need the following program installed on your computer.

1. The Apache Web server
2. The PHP engine
3. The MySQL database server

You can either install them individually or choose a pre-configured package for your operating system like Linux and Windows. Popular pre-configured package are XAMPP and WampServer.

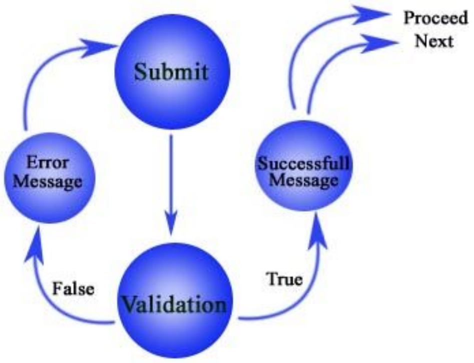
Creating Your First PHP Script

```
<!DOCTYPE HTML>
<html>
<head>
   <title>PHP Application</title>
</head>
<body>
<?php
// Display greeting message
echo 'Hello World!';
?>
</body>
</html>
```

Add your PHP code snapshot with output

Chapter 11
PHP form validation

It is very essential to have the input to your form validated before taking the form submission data for further processing. When there are many fields in the form, the PHP validation script becomes too complex.



Validation rules:

- Name : Should required letters and white-spaces
- Email : Should required @ and .
- Website : Should required a valid URL
- Radio : Must be selectable at least once
- Check Box : Must be checkable at least once
- Drop Down menu : Must be selectable at least once

Validation for non-empty, alphabets and whitespace only

The following code is added within the form

```
<label>Full name<span class="note">*</span>:</label>
  <input type="text" name="full_name" placeholder="FirstName LastName"
autofocus="autofocus" value="<?php echo $_POST['full_name']; ?>">
  <?php echo "<p class='note'>".$msg_name."</p>";?>
  <?php echo "<p class='note'>".$msg2_name."</p>";?>
```

Code for validation

```
if (isset($_POST['submit'])) {
//checking name
if(empty($_POST['full_name']))
$msg_name = "You must supply your name";
$name_subject = $_POST['full_name'];
$name_pattern = '/^[a-zA-Z ]*$/';
preg_match($name_pattern, $name_subject, $name_matches);
if(!$name_matches[0])
$msg2_name = "Only alphabets and white space allowed";
}
```

email Validation

Code added within the form

```
<label>Email address<span class="note">*</span>:</label>
  <input type="text" name="email_addr" value="<?php echo $_POST['email_addr']; ?>">
   <?php echo "<p class='note'>".$msg_email."</p>";?>
  <?php echo "<p class='note'>".$msg2_email."</p>";?>
```

Code for validation

```
if (isset($_POST['submit'])) {
//check email
if(empty($_POST['email_addr']))
$msg_email = "You must supply your email";
$email_subject = $_POST['email_addr'];
$email_pattern = '/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/';
preg_match($email_pattern, $email_subject, $email_matches);
if(!$email_matches[0])
$msg2_email = "Must be of valid email format";
}
```

Selection list Validation

Code added within the form

```
<label>Select Tour Package<span class="note">*</span>:</label>
  <select name="package">
        <option value="Goa" <?= ($_POST['package'] == "1")? "selected":"";?>>Goa</options>
        <option value="Kashmir" <?= ($_POST['package'] == "2")?
"selected":"";?>>Kashmir</options>
        <option value="Rajasthan" <?= ($_POST['package'] == "3")?
"selected":"";?>>Rajasthan</options>
  </select>
```

Code for validation

```
if (isset($_POST['submit'])) {
if(empty($_POST['package']))
$msg_package = "You must select a package";
}
```

Add your PHP validation snapshot

Chapter 12

PHP MySQL database operations

With PHP, you can connect to and manipulate databases. MySQL is the most popular database system used with PHP.

What is MySQL?

- MySQL is a database system used on the web
- MySQL is a database system that runs on a server
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, and easy to use
- MySQL uses standard SQL
- MySQL compiles on a number of platforms
- MySQL is free to download and use

The data in a MySQL database are stored in tables. A table is a collection of related data, and it consists of columns and rows. Databases are useful for storing information categorically. A company may have a database with the following tables:

- Employees
- Products
- Customers
- Orders

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
// Create connection
$conn = mysqli_connect($servername, $username, $password);

// Check connection
if (!$conn) {
   die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";
?>
```

If you want to use PHP to query your MySQL database you can do that by either entering the MySQL query command in the PHP script or define the command as a variable and use the variable when needed

```
mysqli_query($query);
```

The command can be repeated again in the source code. All you need to do is to change the $query variable.

For example, here is the complete code that could be used to create a MySQL table in PHP:

```
<?php
$username = "your_username";
$password = "your_password";
$database = "your_database";
$mysqli = new mysqli("localhost", $username, $password, $database);
$query="CREATE TABLE tablename(id int(6) NOT NULL auto_increment,first varchar(15) NOT NULL,last varchar(15) NOT NULL,field1-name varchar(20) NOT NULL,field2-name varchar(20)NOT NULL,field3-name varchar(20) NOT NULL,field4-name varchar(30) NOT NULL, field5-name varchar(30)NOT NULL,PRIMARY KEY (id),UNIQUE id (id),KEY id_2 (id))";
$mysqli->query("$query");
$mysqli->close();
?>
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
```

```
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
    // output data of each row
    while($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}
mysqli_close($conn);
?>
```

Add your page and code snapshot where database interaction is done

Chapter 13

RIA using AJAX

JAX stands for Asynchronous JavaScript and XML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script. Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display. Conventional web applications transmit information to and from the sever using synchronous requests. It means you fill out a form, hit submit, and get directed to a new page with new information from the server. With AJAX, when you hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.

XML is commonly used as the format for receiving server data, although any format, including plain text, can be used. AJAX is a web browser technology independent of web server software. A user can continue to use the application while the client program requests information from the server in the background. Intuitive and natural user interaction. Clicking is not required, mouse movement is a sufficient event trigger. Data-driven as opposed to page-driven.

Rich Internet Application Technology

AJAX is the most viable Rich Internet Application (RIA) technology so far. It is getting tremendous industry momentum and several tool kit and frameworks are emerging. But at the same time, AJAX has browser incompatibility and it is supported by JavaScript, which is hard to maintain and debug. AJAX is Based on Open Standards

AJAX is based on the following open standards −

- Browser-based presentation using HTML and Cascading Style Sheets (CSS).
- Data is stored in XML format and fetched from the server.
- Behind-the-scenes data fetches using XMLHttpRequest objects in the browser.
- JavaScript to make everything happen.

The XMLHttpRequest Object

All modern browsers support the XMLHttpRequest object. The XMLHttpRequest object can be used to exchange data with a server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

Send a Request To a Server

To send a request to a server, we use the open() and send() methods of the XMLHttpRequest object:

```
xhttp.open("GET", "ajax_info.txt", true);
xhttp.send();
```

The onreadystatechange Property The readyState property holds the status of the XMLHttpRequest. The onreadystatechange property defines a function to be executed when the readyState changes. The status property and the statusText property holds the status of the XMLHttpRequest object.

```
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML =
      this.responseText;
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
}
```

Add your ajax based page part snapshot and code snapshot here

Chapter 14

SQL injection and its prevention

SQL injection is a code injection technique that might destroy your database. SQL injection is one of the most common web hacking techniques. SQL injection is the placement of malicious code in SQL statements, via web page input.

SQL in Web Pages

SQL injection usually occurs when you ask a user for input, like their username/userid, and instead of a name/id, the user gives you an SQL statement that you will unknowingly run on your database.

SQL Injection Based on 1=1 is Always True

Look at the example above again. The original purpose of the code was to create an SQL statement to select a user, with a given user id.

If there is nothing to prevent a user from entering "wrong" input, the user can enter some "smart" input like this:

UserId:

```
105 OR 1=1
```

Then, the SQL statement will look like this:

```
SELECT * FROM Users WHERE UserId = 105 OR 1=1;
```

The SQL above is valid and will return ALL rows from the "Users" table, since OR 1=1 is always TRUE

Does the example above look dangerous? What if the "Users" table contains names and passwords?

The SQL statement above is much the same as this:

```
SELECT UserId, Name, Password FROM Users WHERE UserId = 105 or 1=1;
```

A hacker might get access to all the usernames and passwords in a database, by simply inserting 105 OR 1=1 into the input field.

SQL Injection Based on Batched SQL Statements Most databases support batched SQL statement. A batch of SQL statements is a group of two or more SQL statements, separated by semicolons. The SQL statement below will return all rows from the "Users" table, then delete the "Suppliers" table.

Example

```
SELECT * FROM Users; DROP TABLE Suppliers
```

Primary Defenses:

Option 1: Use of Prepared Statements (with Parameterized Queries)

Option 2: Use of Stored Procedures

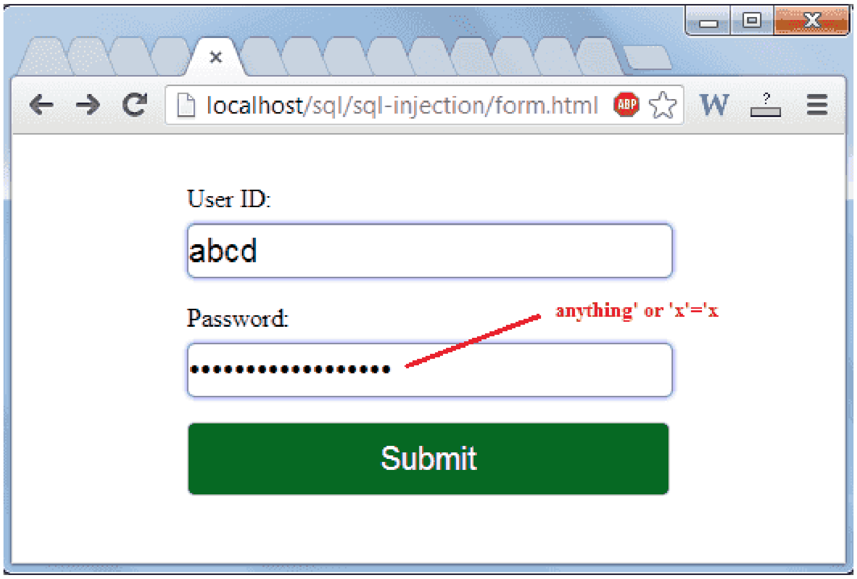
Option 3: White List Input Validation

Option 4: Escaping All User Supplied Input

Additional Defenses:

Also: Enforcing Least Privilege

Also: Performing White List Input Validation as a Secondary Defense

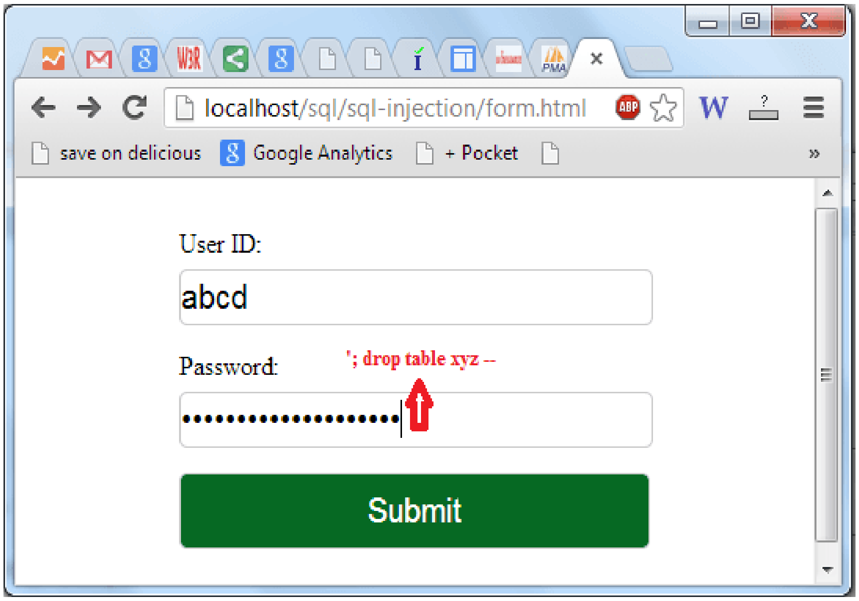


localhost/sql/sql-injection/form.html

User ID: abcd

Password: ••••••••••••••••••

anything' or 'x'='x



User ID: abcd

Password: ••••••••••••••••••••

'; drop table xyz --

Replace above image with your project code snapshot and image

Chapter 15

Stored procedure using MySQL

A stored procedure is a set of Structured Query Language (SQL) statements with an assigned name, which are stored in a relational database management system as a group, so it can be reused and shared by multiple programs.

Stored procedures can access or modify data in a database, but it is not tied to a specific database or object, which offers a number of advantages.

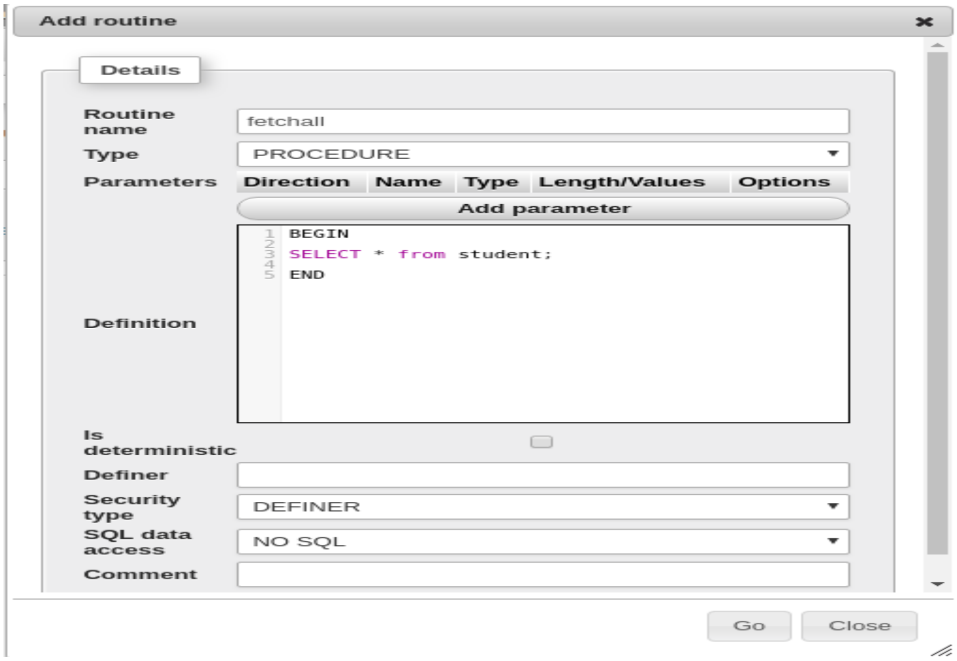
Benefits of using stored procedures

A stored procedure provides an important layer of security between the user interface and the database. It supports security through data access controls because end users may enter or change data, but do not write procedures. A stored procedure preserves data integrity because information is entered in a consistent manner. It improves productivity because statements in a stored procedure only must be written once.

Creating SQL stored procedures.

Stored procedures offer advantages over embedding queries in a graphical user interface (GUI). Since stored procedures are modular, it is easier to troubleshoot when a problem arises in an application. Stored procedures are also tunable, which eliminates the need to modify the GUI source code to improve its performance. It's easier to code stored procedures than to build a query through a GUI.

Use of stored procedures can reduce network traffic between clients and servers, because the commands are executed as a single batch of code. This means only the call to execute the procedure is sent over a network, instead of every single line of code being sent individually.



Add routine

Details

Routine name: fetchall

Type: PROCEDURE

Parameters: Direction | Name | Type | Length/Values | Options

Definition:

```
BEGIN
SELECT * from student;
END
```

Is deterministic: □

Definer:

Security type: DEFINER

SQL data access: NO SQL

Comment:

Replace above image with your project procedure snapshot and page image

Chapter 16

Web hosting

Web hosting is a service that allows organizations and individuals to post a website or web page onto the Internet. A web host, or web hosting service provider, is a business that provides the technologies and services needed for the website or webpage to be viewed in the Internet. Websites are hosted, or stored, on special computers called servers. When Internet users want to view your website, all they need to do is type your website address or domain into their browser. Their computer will then connect to your server and your web pages will be delivered to them through the browser.

Most hosting companies require that you own your domain in order to host with them. If you do not have a domain, the hosting companies will help you purchase one.

Here are some features you should be expecting from your hosting provider:

Email Accounts : As mentioned earlier, most hosting providers require users to have their own domain name. With a domain name (e.g. www.yourwebsite.com) and email account features provided by your hosting company, you can create domain email accounts (e.g. yourname@yourwebsite.com).

FTP Access : The use of FTP lets you upload files from your local computer to your web server. If you build your website using your own HTML files, you can transfer the files from your computer to the web server through FTP, allowing your website to be accessed through the internet.

Add your hosted website URL

Snapshot of Landing page

Snapshot of Login Page

Snapshot of Registration Page

Snapshot of About us page

Snapshot of Contact page

Snapshot of User Home page

Snapshot of Admin Home page

Acknowledgement

(The following text shall be 12pt Font, Times New Roman, Justified, 1.5 line spacing)

Use separate paragraph for each Guide, HOD, and Principal.

………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
…………………………………………………………………

………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
……………………………………………………………………

………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
………………………………………………………………………………………………
……………………………………………………………………

Project member 1 name
Project member 2 name
Project member 3 name
Project member 4 name