

INTRODUCTION OF COMAPNAY

I have completed our industrial training at StormSofts Technology. StormSofts Technology Pvt. Ltd. is a thriving company that excels in crafting vibrant and highly functional website designs tailored to the needs of both businesses and individuals. As a rapidly growing organization, StormSofts Technology has established itself as a prominent player in the field of Web Designing, setting new standards of excellence and innovation.

StormSofts Technology was founded in 1-1-2017 by Mr. Ankush S. Pol with a vision to become unique company offering unique IT solutions. We pride ourselves on having long-term relationships with our clients and focus on being a valued partner rather than merely a service provider.

Throughout our industrial training at StormSofts Technology, we were immersed in an environment that fostered learning and creativity. We had the privilege of working with a team of seasoned professionals who guided us through the intricacies of web design and development. Their passion for their craft and commitment to excellence left a lasting impression on us.

I am immensely grateful to the entire team at StormSofts Technology for imparting their invaluable knowledge and expertise to us. Their commitment to quality, innovation, and building enduring client relationships has inspired us to strive for excellence in all our endeavors. As we move forward, we carry with us the lessons learned during our training, and we are eager to apply these skills to contribute meaningfully to the world of web development and IT solutions.

INTRODUCTION ABOUT TRAINING

The Industrial Training Report presents an in-depth account of the invaluable practical learning experience attained during the Full Stack Web Development program. The primary objective of this training was to equip participants with essential skills in HTML, CSS, JavaScript, Bootstrap, PHP, and MySQL, spanning a well-defined duration. The report meticulously outlines the thoughtfully crafted curriculum, effective training methodologies, and the outstanding learning outcomes achieved throughout the program.

In today's digital landscape, Full Stack Web Development has become an integral aspect, meeting the demands for dynamic and interactive web applications. This report highlights the training's importance in addressing various technologies and their applications in real-world scenarios.

In this Training Several Technologies are covered like,

1. HTML
2. CSS
3. Bootstrap
4. JavaScript
5. PHP
6. MySQL

INTRODUCTION TO HTML

Markup Language: HTML is a markup language, not a programming language. It uses a system of tags (also known as elements) to define the structure and layout of content on a web page. These tags are surrounded by angle brackets, such as '`<tagname>content</tagname>`'.

Structure of an HTML Document: An HTML document consists of various elements, including the document type declaration, '`<html>`' element (root element), '`<head>`' element (contains metadata like the page title and links to external resources), and '`<body>`' element

• Commonly used HTML Tags

1. `<html>`: The root element that wraps the entire HTML document.
2. `<head>`: Contains meta-information about the document, such as the page title, character encoding, CSS styles, and other metadata.
3. `<title>`: Defines the title of the web page, which appears in the browser's title bar or tab.
4. `<body>`: Contains the visible content of the web page, including text, images, links, and other elements.
5. `<h1>` to `<h6>`: Header tags for creating headings, with `<h1>` being the highest level and `<h6>` the lowest.
6. `<p>`: Represents a paragraph of text.
7. `<a>`: Creates a hyperlink, allowing users to navigate to another web page or resource when clicked. It uses the href attribute to specify the target URL.
8. ``: Embeds an image on the web page. It requires the src attribute, which points to the image file's location.
9. `` and ``: Used to create an unordered list (bulleted list) with list items represented by `` tags.
10. `` and ``: Used to create an ordered list (numbered list) with list items represented by `` tags.
11. `<div>`: A generic container that is often used to group and style content within a web

HTML5: HTML5 is the latest version of HTML and comes with additional features and improvements, such as support for multimedia elements, form enhancements, canvas for graphics, and more.

HTML5 Tags :

Now, let's briefly introduce some HTML5-specific tags. HTML5 introduced new elements to provide more semantic meaning to web content, improve accessibility, and simplify website structuring.

1. **<header>** Represents the header section of a webpage or a section within the document.
2. **<nav>** Defines a container for navigation links.
3. **<section>** Represents a thematic grouping of content within a document, such as a chapter or a block of related content.
4. **<article>** Represents a self-contained piece of content that could be independently distributed and reused, such as blog posts, news articles, etc.
5. **<aside>** Defines content that is tangentially related to the content around it, like sidebars or call-out boxes.
6. **<footer>** Represents the footer section of a webpage or a section within the document.
7. **<video>** Used to embed videos on a web page. It supports various video formats and allows for easy playback.
8. **<audio>** Used to embed audio files on a web page, supporting various audio formats.

INTRODUCTION TO CSS

• Cascading Style Sheets :

CSS (Cascading Style Sheets) is a style sheet language used to describe the presentation and appearance of web pages written in HTML or XML. It provides web developers with the ability to control the layout, colors, fonts, and other visual aspects of a website, allowing them to separate the content from the design and layout.

• Key concept and features of CSS –

CSS works by associating style rules with HTML elements, which are defined using selectors. These rules determine how the elements should be displayed in the web browser. Here's an introduction to some of the key concepts and features of CSS:

1. **Selectors:** CSS selectors are patterns that are used to target HTML elements on a web page. They can select elements based on their tag name, class, ID, attributes, and more. For example:

Tag Selector: 'p' selects all paragraphs on the page.

Class Selector: '.classname' selects all elements with the specified class.

ID Selector: '#elementID' selects the element with the specified ID.

2. **Properties and Values:** CSS rules consist of one or more properties and their corresponding values. Properties define the aspect of an element that you want to style (e.g., color, font-size, padding, margin), and values specify the settings for those properties.
3. **Declaration Block:** CSS rules are contained within a declaration block, which is enclosed in curly braces '{ }'. Each declaration block consists of one or more property-value pairs separated by semicolons.
4. **Internal CSS:** CSS can be added directly to an HTML document using the '<style>' element in the head section. This is called internal or embedded CSS.
5. **External CSS:** To keep the styles separate from the HTML document and promote better organization and reusability, CSS can also be placed in an external file with a '.css' extension. The file is then linked to the HTML document using the '<link>' element in the head section.
6. **Inheritance:** CSS properties can be inherited from parent elements to their children. This means that if a property is not specified for a particular element, it will inherit the value from its parent.
7. **Cascading Order:** The "Cascading" in CSS refers to the order in which conflicting styles are applied. If multiple CSS rules target the same element, the rule with the highest

specificity takes precedence. In case of equal specificity, the rule that appears later in the stylesheet wins.

8. **Box Model:** The Box Model is a fundamental concept in CSS that defines how elements are displayed as boxes with content, padding, border, and margin. Understanding the Box Model is crucial for layout and spacing control.

CSS is a powerful and flexible tool for web developers, enabling them to create visually appealing and responsive websites. As web technologies have evolved, CSS has grown to support more advanced features, including animations, transitions, flexbox, grid layout, and more. Learning CSS is essential for anyone interested in web development and design.

INTRODUCTION TO BOOTSTRAP

• Bootstrap –

Bootstrap is a free and open-source front-end framework for building responsive and mobilefirst websites and web applications. It was originally developed by Twitter and is now maintained by a group of developers and contributors. Bootstrap provides a set of pre-designed HTML, CSS, and JavaScript components and styles that help streamline the web development process and ensure consistency and responsiveness across various devices and screen sizes

- Responsive web development using Bootstrap has become a popular approach in modern web design due to the increasing use of mobile devices and various screen sizes. Bootstrap is a front-end framework that provides a set of pre-designed HTML, CSS, and JavaScript components, making it easier to create responsive and mobile-friendly websites. Here are some key reasons why Bootstrap is widely used for responsive web development:
- Mobile-First Approach: Bootstrap follows a mobile-first approach, meaning that the default styles and components are designed for mobile devices. This ensures that the website looks great and functions well on smaller screens and progressively enhances as the screen size increases.
- Responsive Components: Bootstrap provides a collection of responsive CSS classes and components, such as navigation bars, buttons, forms, and carousels, which adapt their appearance and behavior based on the screen size. This saves development time and effort in customizing components for different devices.
- In conclusion, responsive web development using Bootstrap offers numerous advantages, including a mobile-first approach, a responsive grid system, pre-designed components, cross-browser compatibility, and extensive community support. These factors have made Bootstrap a popular choice for creating adaptive and user-friendly websites that work seamlessly across various devices and screen sizes.

• Bootstrap Classes

1. Grid System:

.container: Creates a fixed-width container with responsive padding.

.container-fluid: Creates a full-width container that spans the entire viewport.

2. Responsive Layout:

.row: Creates a row to contain columns within a grid layout.

.col-*: Defines columns within a row. The * can be replaced with numbers from 1 to 12, representing the column width.

3. Typography:

.lead: Makes text stand out by increasing its font size.

.text-*: Aligns text horizontally. Replace * with values like left, center, right, or justify.

4. Buttons:

.btn: Styles a button element.

.btn-*: Adds contextual styles to buttons, such as btn-primary, btn-danger, btn-warning, etc.

5. Alerts:

.alert: Styles an alert box.

.alert-*: Adds contextual styles to alerts, such as alert-success, alert-info, alert-warning, alert-danger, etc.

6. Navigation:

.navbar: Creates a navigation bar.

.navbar-brand: Styles the brand/logo area in the navigation bar.

.nav: Styles an unordered list to create a navigation menu.

.nav-item and .nav-link: Styles individual items and links in the navigation menu.

7. Forms:

.form-control: Styles form inputs, text areas, and select elements.

.form-group: Groups form elements together for proper spacing.

.form-check: Styles checkboxes and radio buttons.

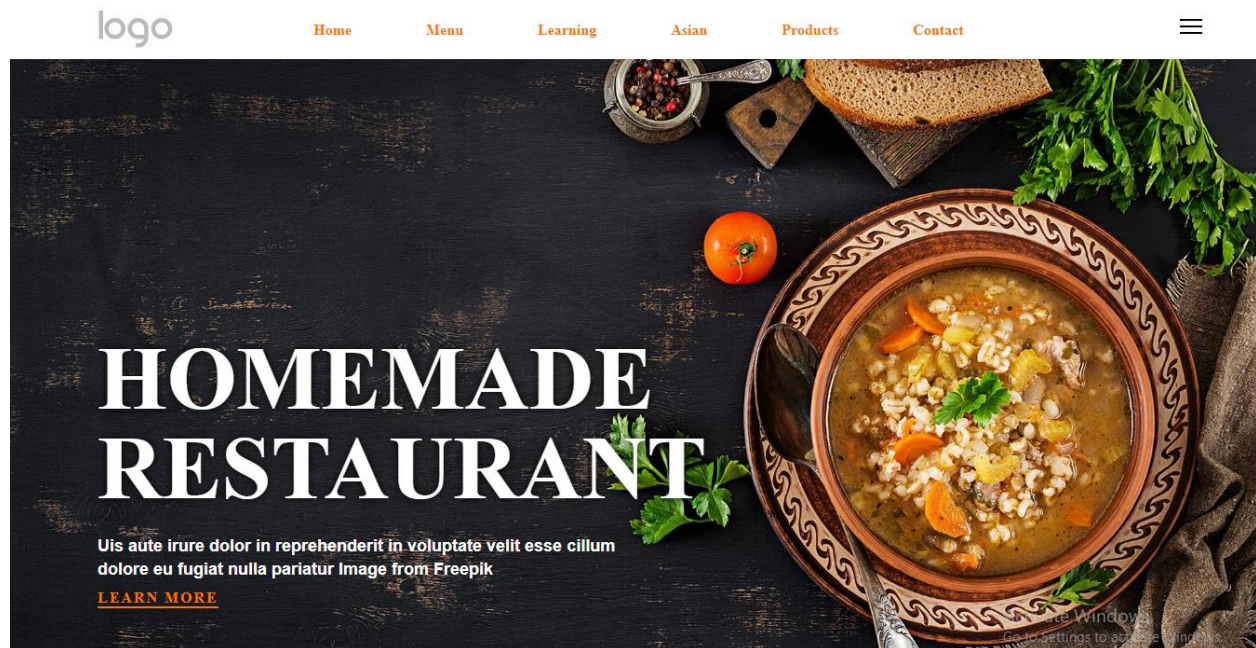
.btn-group: Styles a group of buttons together.



8. Spacing:

.m-*: Applies margin to an element. Replace * with values like 1, 2, 3, auto, etc.

.p-*: Applies padding to an element. Replace * with values like 1, 2, 3, auto, etc.

HTML CSS Implementation In simple Project –







Menu

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt


[learn more](#)






FRESH FOOD EVERYDAY


Our Restaurant is a family-owned restaurant that has been proudly serving around the 1940s located on the corner of Broadway and West 112th Street in New York. [learn more](#)




OUR MENU



SPECIALS



DRINKS



Activate Windows
Go to Settings to activate Windows.

INTRODUCTION TO JAVA SCRIPT

Basic Introduction: JavaScript is a client-side scripting language, meaning it runs directly in the web browser on the user's device. This enables web developers to create interactive and responsive websites without relying on server-side processing. JS can be seamlessly integrated into HTML and CSS, making it the backbone of many web applications and enhancing user experience.

❖ Features of JavaScript:

a) Lightweight: JavaScript is a lightweight scripting language with a small file size, allowing web pages to load quickly.

b) Versatility: JS is a versatile language that supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

c) Event-Driven: JavaScript uses an event-driven programming model, allowing developers to respond to user actions such as clicks, form submissions, or mouse movements.

d) Cross-Browser Compatibility: JavaScript is supported by all major web browsers, ensuring consistent behavior across different platforms.

e) Interactivity: By manipulating the Document Object Model (DOM), JavaScript enables developers to modify the content and structure of web pages in real-time, creating interactive experiences for users.

f) Asynchronous Execution: JavaScript supports asynchronous programming, allowing certain tasks to be performed in the background without blocking the main execution flow. This is commonly used for handling network requests and optimizing performance.

❖ Important Points in JavaScript

Variables and Data Types:

JavaScript uses variables to store data. It is a dynamically typed language, meaning you don't need to specify the data type explicitly.

Example:

```
let age = 25; // Declaring a variable 'age' and assigning the value 25
```

```
let name = 'John'; // Declaring a variable 'name' and assigning the value 'John'
```

Functions:

Functions in JavaScript are blocks of code that can be defined and invoked to perform specific tasks.

Example:

```
function greet(name) {  
    return 'Hello, ' + name + '!';  
}
```

```
let message = greet('Alice'); // message will be 'Hello, Alice!'
```

Conditional Statements:

Conditional statements are used to make decisions in code based on certain conditions.

Example:

```
let temperature = 30;  
if (temperature > 25) {  
    console.log('It is a hot day.');} else {  
    console.log('It is a moderate day.');}
```

Loops:

Loops are used to execute a block of code repeatedly until a condition is met.

Example:

```
for (let i = 1; i <= 5; i++) {  
    console.log('Number: ' + i);  
}
```

Arrays:

Arrays are used to store multiple values in a single variable.

Example:

```
let fruits = ['apple', 'banana', 'orange'];  
  
console.log(fruits[0]); // Output: 'apple'  
console.log(fruits.length); // Output: 3
```

Objects:

Objects in JavaScript are used to store key-value pairs and represent complex data structures.

Example:

```
let person = {  
  name: 'John',  
  age: 30,  
  profession: 'Engineer'  
};  
  
console.log(person.name); // Output: 'John'
```

DOM Manipulation:

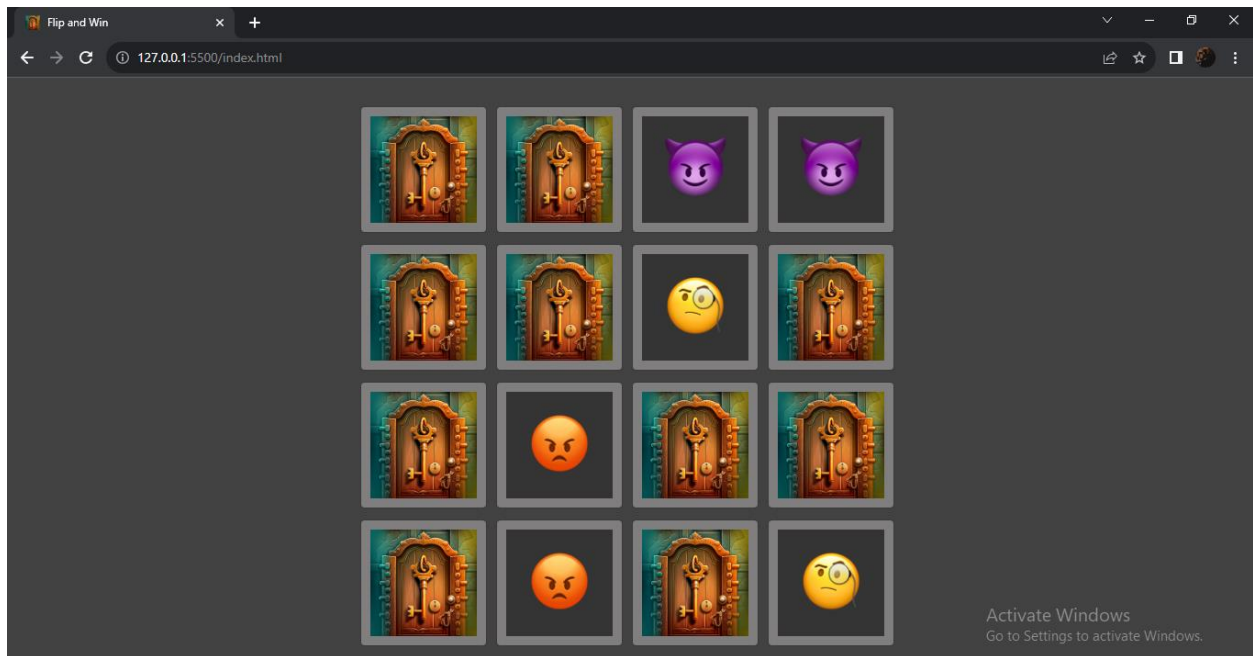
JavaScript can interact with HTML elements using the Document Object Model (DOM) to manipulate and change their content or appearance dynamically.

Example:

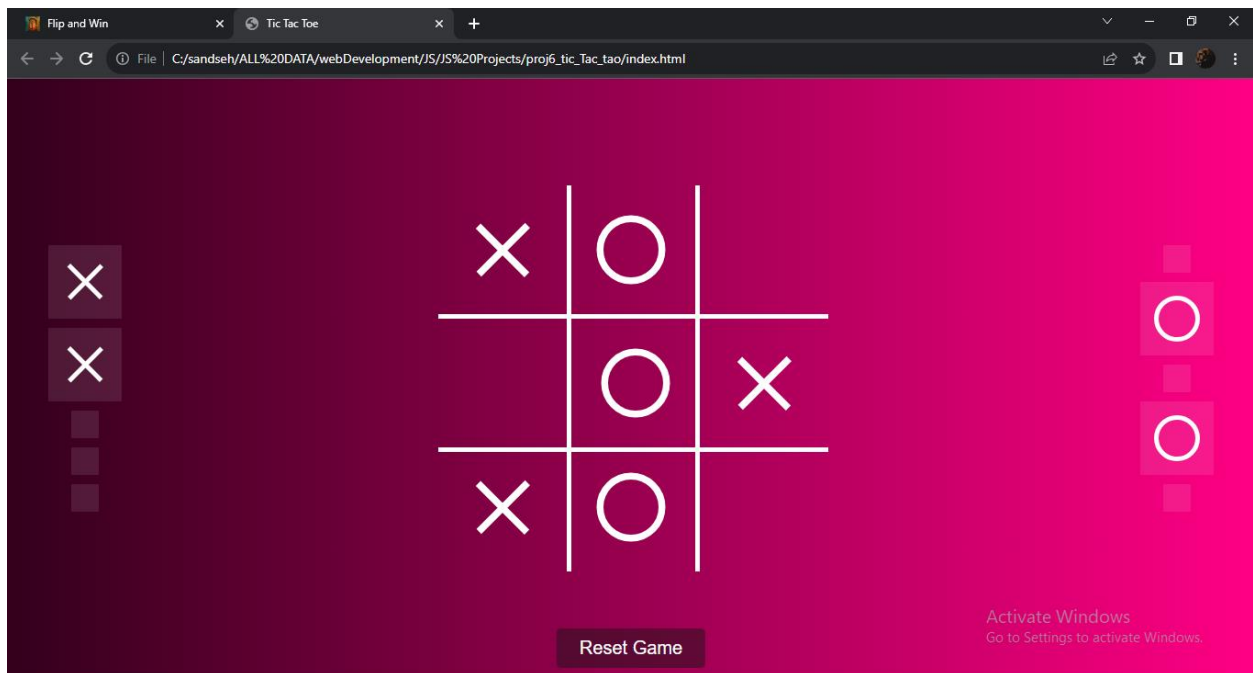
```
<!DOCTYPE html>  
  
<html>  
  
  <head>  
  
    <title>DOM Manipulation</title>  
  
  </head>  
  
  <body>  
  
    <p id="message">Hello, World!</p>  
  
    <script>  
  
      let messageElement = document.getElementById('message');  
  
      messageElement.innerText = 'Welcome to JavaScript!';  
  
    </script>  
  
  </body>  
  
</html>
```

HTML CSS Java Script Implementation In simple Projects:

1. Flip and Win Game using JS:



2. Tic Toe Tac Game:



INTRODUCTION TO PHP

PHP (Hypertext Preprocessor) is a popular server-side scripting language designed primarily for web development. It is embedded within HTML and executed on the web server, generating dynamic content that is then sent to the client's web browser. PHP is open-source, easy to learn, and widely used to create dynamic websites, web applications, and server-side functionalities.

❖ Features of PHP :

1. Versatility: PHP can be used on various operating systems (Windows, macOS, Linux) and supports multiple web servers (Apache, Nginx).
2. Simple Syntax: PHP has a straightforward and easy-to-understand syntax, making it accessible to both beginners and experienced developers.
3. Integration: PHP can be easily embedded into HTML, allowing developers to mix PHP code with HTML, making web development more seamless.
4. Extensive Library Support: PHP has a vast collection of libraries and frameworks that simplify common tasks and accelerate development.
5. Database Support: PHP supports numerous databases, such as MySQL, PostgreSQL, and SQLite, allowing developers to interact with databases efficiently.
6. Open-Source Community: The large PHP open-source community contributes to its continuous improvement, updates, and bug fixes.
7. Security: PHP has built-in security features, such as data filtering and encryption functions, making it suitable for secure web application development.

❖ Important Points in JavaScript

Basic PHP Syntax:

PHP code is enclosed within `<?php ... ?>` tags. It can be embedded in HTML or used as a standalone script.

Example:

```
<?php  
echo "Hello, World!";  
?>
```

Variables:

PHP variables start with the \$ symbol. They are dynamically typed and do not require explicit type declarations.

Example:

```
<?php
$name = "John";
$age = 25;
echo "My name is $name and I am $age years old.";
?>
```

Data Types:

PHP supports various data types, including strings, integers, floats, arrays, booleans, etc.

Example:

```
<?php
$str = "Hello";
$num = 42;
$pi = 3.14;
$isTrue = true;
$fruits = array("Apple", "Banana", "Orange");
?>
```

Conditional Statements:

Example:

```
<?php
$age = 20;

if ($age >= 18) {
    echo "You are an adult.";
} else {
```



```
        echo "You are a minor.";
    } ?>
```

Loops:

PHP offers different types of loops like for, while, do-while, and foreach.

Example:

php

Copy code

```
<?php
for ($i = 1; $i <= 5; $i++) {
    echo "$i ";
}

$numbers = array(1, 2, 3, 4, 5);
foreach ($numbers as $num) {
    echo "$num ";
}

?>
```

Functions:

PHP allows defining custom functions that can be reused throughout the code.

Example:

```
<?php
function addNumbers($a, $b) {
    return $a + $b;
}

echo addNumbers(5, 3); // Output: 8

?>
```

Superglobals:

PHP provides predefined variables called superglobals that can be accessed from anywhere in the PHP script.

Example:

```
<?php
echo $_SERVER['REMOTE_ADDR']; // Displays the client's IP address

echo $_GET['param'];          // Accessing URL parameters

echo $_POST['field'];         // Accessing form data

?>
```

Include and Require:

PHP supports including other PHP files to reuse code or modularize the application.

Example:

```
<!-- file1.php -->

<?php
$message = "Hello from file1.php";

?>


<!-- file2.php -->

<?php
require 'file1.php';

echo $message; // Output: Hello from file1.php

?>
```

Object-Oriented Programming (OOP):

PHP supports object-oriented programming with classes and objects.

Example:

```
<?php

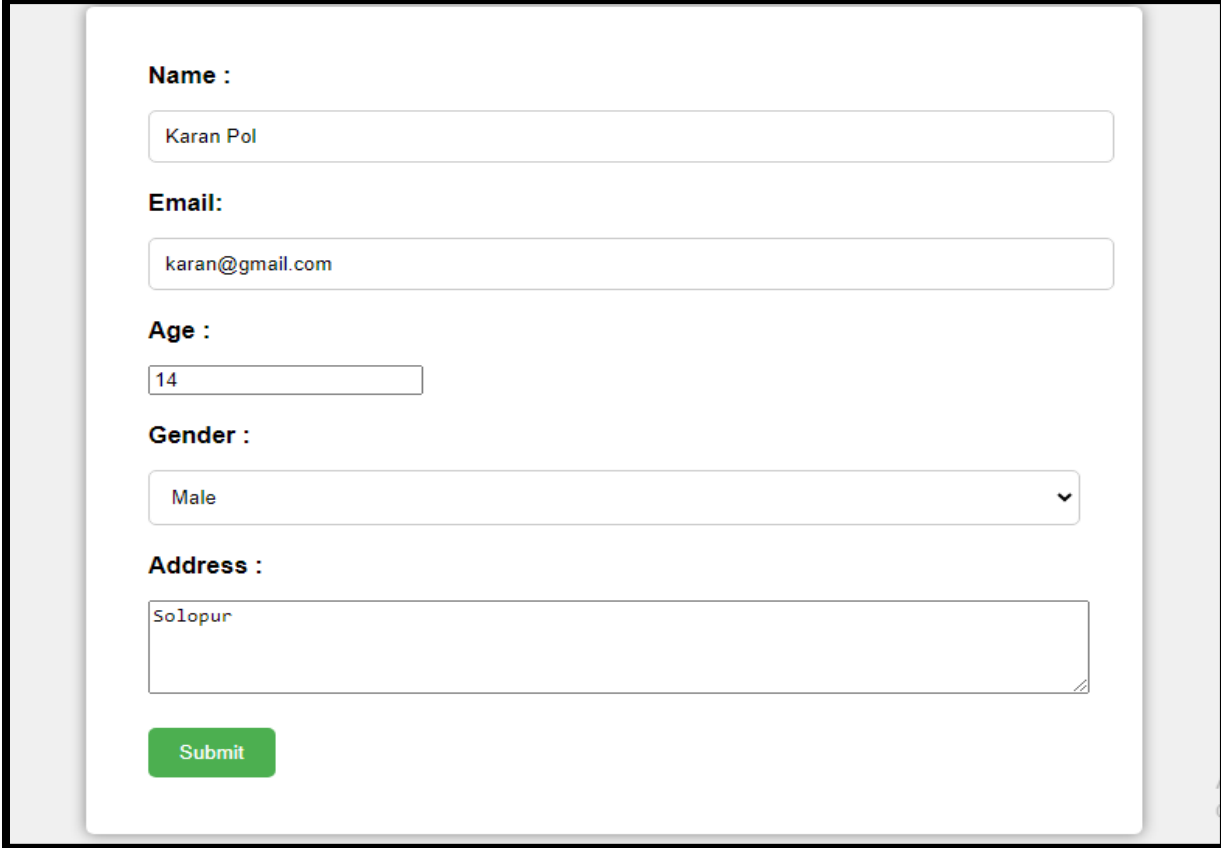
class Person {

    public $name;
```

```
public function greet() {  
    echo "Hello, my name is $this->name."; }  
  
$person = new Person();  
$person->name = "Alice";  
$person->greet(); // Output: Hello, my name is Alice.  
?>e.
```

Working with Databases (MySQL):

```
MySQL $conn = mysqli_connect("localhost", "username", "password", "database_name");  
$sql = "SELECT * FROM users"; $result = mysqli_query($conn, $sql);  
while ($row = mysqli_fetch_assoc($result))  
{  
    echo $row['name'] . "<br>";  
}  
mysqli_close($conn);
```

Performing Database Operation on Form:

A screenshot of a web form for user registration. The form is titled "Name :" and has a text input field containing "Karan Pol". Below this is an "Email:" field with "karan@gmail.com". The "Age :" field contains "14". The "Gender :" field is a dropdown menu with "Male" selected. The "Address :" field contains "Solapur". A green "Submit" button is at the bottom.

Name :

Karan Pol

Email:

karan@gmail.com

Age :

14

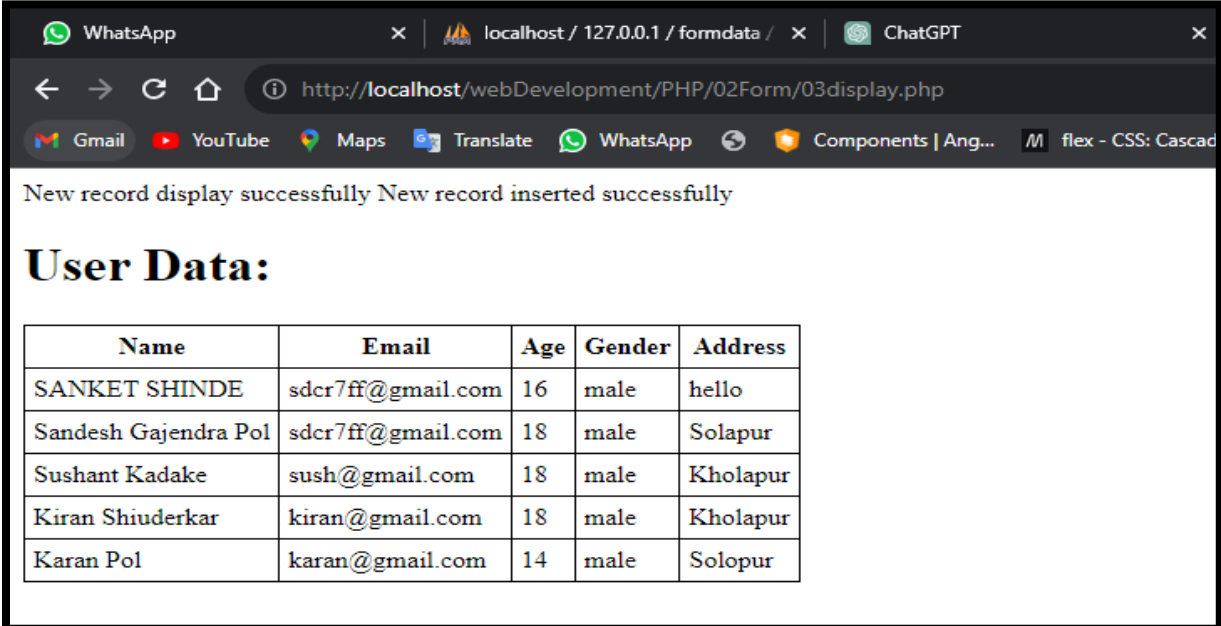
Gender :

Male

Address :

Solapur

Submit



A screenshot of a web browser window. The address bar shows "http://localhost/webDevelopment/PHP/02Form/03display.php". The page content includes a message "New record display successfully New record inserted successfully" and a section titled "User Data:" followed by a table of user data.

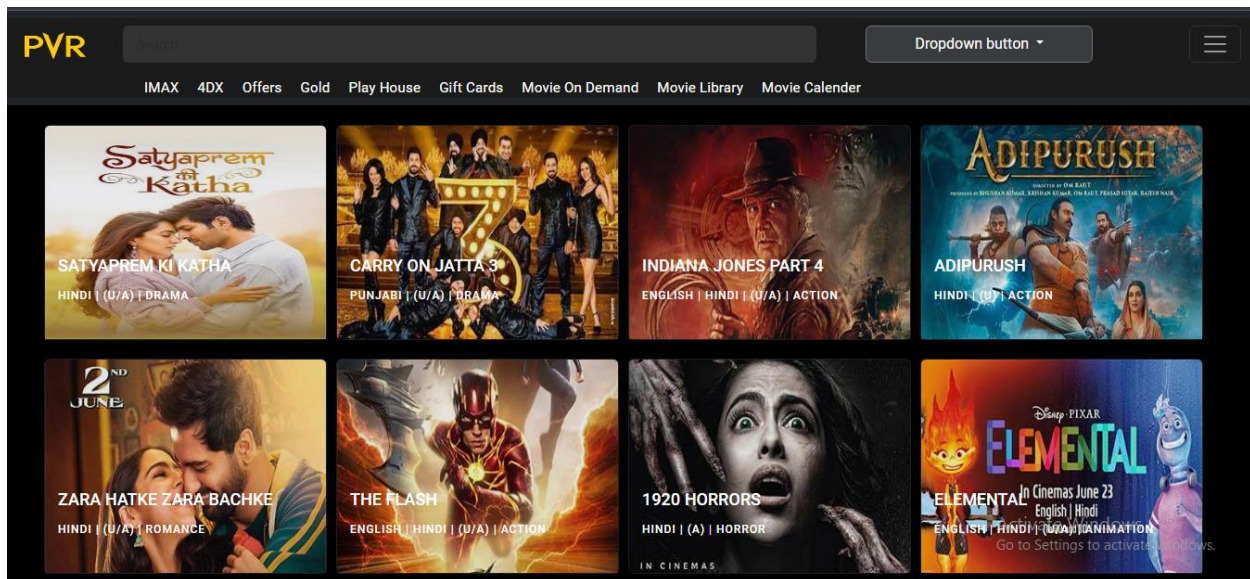
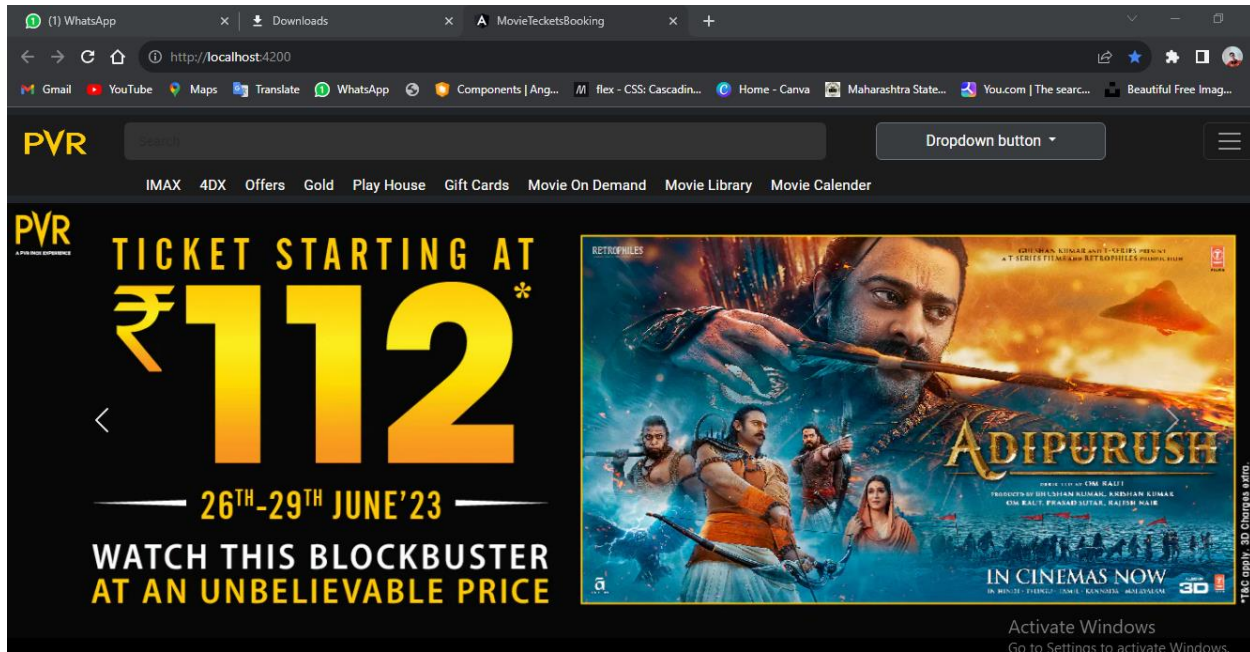
New record display successfully New record inserted successfully

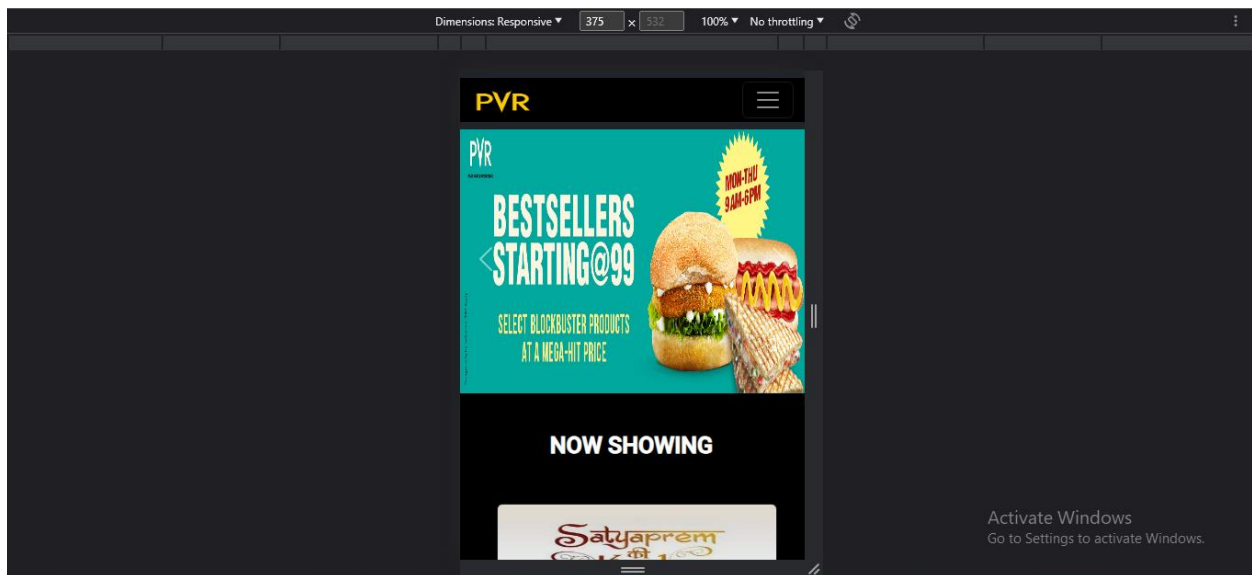
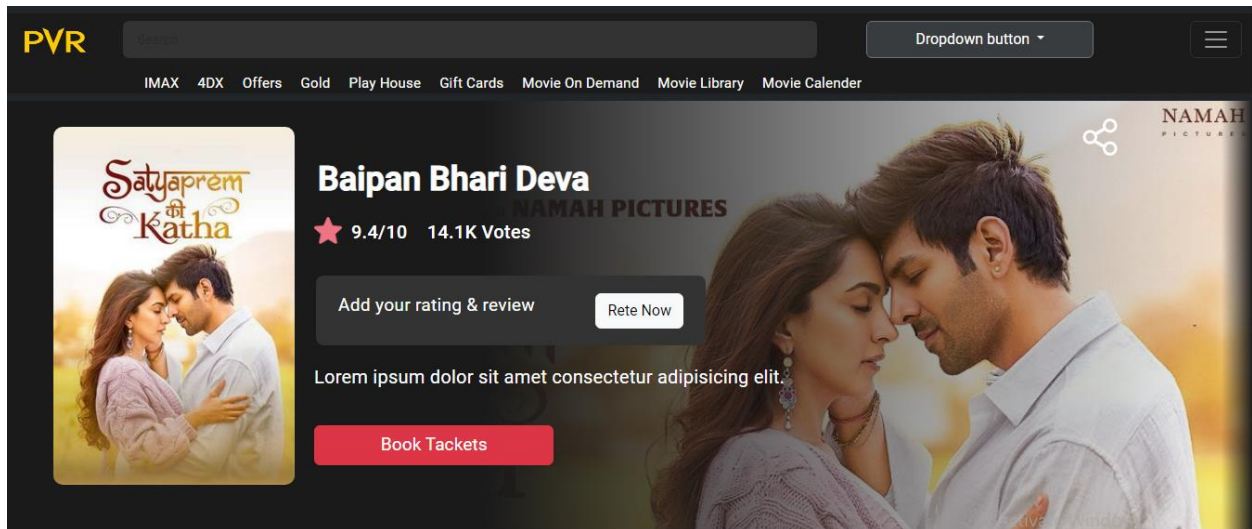
User Data:

Name	Email	Age	Gender	Address
SANKET SHINDE	sdcr7ff@gmail.com	16	male	hello
Sandesh Gajendra Pol	sdcr7ff@gmail.com	18	male	Solapur
Sushant Kadake	sush@gmail.com	18	male	Kholapur
Kiran Shiuderkar	kiran@gmail.com	18	male	Kholapur
Karan Pol	karan@gmail.com	14	male	Solapur

Web Development Project

Dynamic Website for Movie Tickets Booking:





CONCLUSION

In this 6 weeks of industrial training I have learned lot of things which I used in today's IT industry. I have learned some concepts of Web Development. This industrial training Covered all bits and pieces of different interesting topics. Most important of all I learned that How to do perfect use of html, css, javascript and basic php. We learned how to create any website or any template using html and css. We also get knowledge about how to make our web page responsive and advanced or more effective using bootstrap and javascript it under the guidance of training supervisor and their functionary.