**IP**

**Experiment No: 01 - vs code**

**Aim – Create a static webpage using HTML and CSS.**

**Code –**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Habitat Harmony - About Us</title>

<style>

/\* Reset \*/

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: "Segoe UI", Arial, sans-serif;

background-color: #fdfdfd;

color: #333;

line-height: 1.6;

}

/\* Header \*/

header {

background: cyan;

padding: 15px 30px;

display: flex;

align-items: center;

justify-content: space-between;

box-shadow: 0 2px 5px rgba(0,0,0,0.1);

}

.logo {

display: flex;

align-items: center;

}

.logo img {

height: 45px;

margin-right: 12px;

}

.logo h2 {

font-size: 28px;

font-weight: bold;

color: #111;

}

nav a {

text-decoration: none;

margin: 0 12px;

font-weight: 600;

color: #111;

padding: 6px 10px;

border-radius: 6px;

transition: background 0.3s, color 0.3s;

}

nav a:hover {

background: #00bcd4;

color: white;

}

/\* Main content \*/

.container {

max-width: 900px;

margin: 40px auto;

padding: 0 20px;

}

.container h3 {

font-size: 26px;

margin-bottom: 15px;

color: #222;

}

.container p {

font-size: 18px;

margin-bottom: 25px;

color: #444;

}

.container h4 {

font-size: 22px;

margin-bottom: 12px;

color: #00796b;

}

ul {

list-style: none;

padding-left: 0;

}

ul li {

background: #e0f7fa;

margin: 10px 0;

padding: 12px 15px;

border-radius: 8px;

box-shadow: 0 2px 5px rgba(0,0,0,0.05);

transition: transform 0.2s;

}

ul li:hover {

transform: translateX(5px);

background: #b2ebf2;

}

/\* Footer \*/

footer {

background: black;

color: white;

text-align: center;

padding: 20px;

margin-top: 40px;

font-size: 14px;

}

</style>

</head>

<body>

<!-- Header -->

<header>

<div class="logo">

<img src="images/leaf-logo.png" alt="Habitat Harmony Logo">

<h2>Habitat Harmony</h2>

</div>

<nav>

<a href="index.html">About Us</a>

<a href="contact.html">Contact Us</a>

<a href="signin.html">Sign In</a>

</nav>

</header>

<!-- Main Content -->

<div class="container">

<h3>About Us</h3>

<p>

Welcome to <b>Habitat Harmony</b> – a platform dedicated to preserving and promoting

native Indian trees. Our mission is to spread awareness about trees that form

the backbone of India's natural ecosystem.

</p>

<h4>Some Native Trees:</h4>

<ul>

<li><b>Jamun (Syzygium cumini):</b> Known for its delicious purple fruit and medicinal value.</li>

<li><b>Banyan Tree (Ficus benghalensis):</b> National tree of India, symbol of longevity.</li>

<li><b>Neem (Azadirachta indica):</b> Famous for its antibacterial and healing properties.</li>

<li><b>Peepal (Ficus religiosa):</b> Considered sacred, improves air quality drastically.</li>

</ul>

</div>

<!-- Footer -->

<footer>

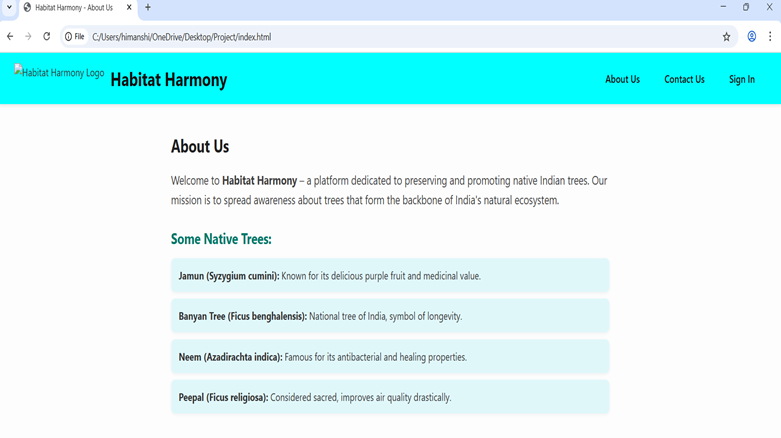
&copy; 2025 Habitat Harmony | All Rights Reserved

</footer>

</body>

</html>

**Output –**

****

**EXPERIMENT NO:02 - vs code**

**AIM:** Implement JavaScript validations on registration form.

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<title>

Form validation using HTML and JavaScript

</title>

<link rel="stylesheet" href="style.css">

<script src="validations.js"></script>

</head>

<body>

<h1 style="text-align: center;">

REGISTRATION FORM

</h1>

<form name="RegForm" onsubmit="return Validations()" method="post">

<p>Name:

<input type="text" size="65" name="Name" />

</p>

<br />

<p>Address:

<input type="text" size="65" name="Address" />

</p>

<br />

<p>E-mail Address:

<input type="text" size="65" name="EMail" />

</p>

<br />

<p>

Password:

<input type="text" size="65" name="Password" />

</p>

<br />

<p>

Telephone:

<input type="text" size="65" name="Telephone" />

</p>

<br />

<p>

SELECT YOUR COURSE

<select type="text" value="" name="Subject">

<option>BTECH</option>

<option>BBA</option>

<option>BCA</option>

<option>B.COM</option>

<option>GEEKFORGEEKS</option>

</select>

</p>

<br />

<br />

<p>

Comments:

<textarea cols="55" name="Comment"> </textarea>

</p>

<p>

<input type="submit" value="send" name="Submit" />

<input type="reset" value="Reset" name="Reset" />

</p>

</form>

</body>

</html>

**Style.css** div { box-sizing: border-box; width: 100%; border: 100px solid black;

float: left; align-content: center; align-items: center;

} form { margin: 0 auto; width: 600px;

}

**Validation.txt**

function Validations(){

var name = document.forms.RegForm.Name.value; var email = document.forms.RegForm.EMail.value; var phone = document.forms.RegForm.Telephone.value;

var what = document.forms.RegForm.Subject.value; var password = document.forms.RegForm.Password.value;

var address = document.forms.RegForm.Address.value;

//Javascript reGex for Email Validation.

var regEmail=/^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$/g;

// Javascript reGex for Phone Number validation.

var regPhone=/^\d{10}$/;

// Javascript reGex for Name validation

var regName = /\d+$/g;

if (name == "" || regName.test(name)) { window.alert("Please enter your name properly.");

name.focus(); return false;

}

if (address == "") { window.alert("Please enter your address."); address.focus();

return false;

}

if (email == "" || !regEmail.test(email)) { window.alert("Please enter a valid e-mail address."); email.focus();

return false;

}

if (password == "") { alert("Please enter your password");

password.focus();

return false;

}

if(password.length <6){ alert("Password should be atleast 6 character long");

password.focus();

return false;

}

if (phone == "" || !regPhone.test(phone)) { alert("Please enter valid phone number."); phone.focus();

return false;

}

if (what.selectedIndex == -1) { alert("Please enter your course."); what.focus();

return false;

}

return true;

}

**Output:**

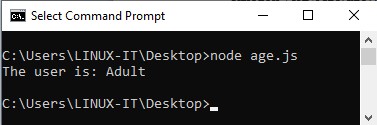
**Experiment No:3 - online js compiler / notepad & cmd**

**1. Age function():**

function verifyAge(age){ return age >= 18 ? "Adult" : "Minor";

} let userAge= 21; let verificationResult=verifyAge(userAge);

console.log("The user is: "+ verificationResult);



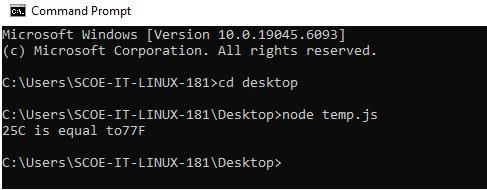
**2.Temperature function():** function temperature(celsius)

{

return(celsius \* 9/5)+32;

} let currentcelsius=25;

let fahrenheit = temperature(currentcelsius); console.log(currentcelsius + "C is equal to" + fahrenheit + "F");



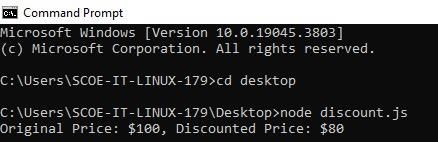
**3.Discount function():** function calculateDiscount(price,discountPercentage) { return price - (price \* discountPercentage / 100);

}

let originalPrice = 100; let discount =20;

let discountedPrice = calculateDiscount(originalPrice,discount);

console.log("Original Price: $" + originalPrice + ", Discounted Price: $"+ discountedPrice);

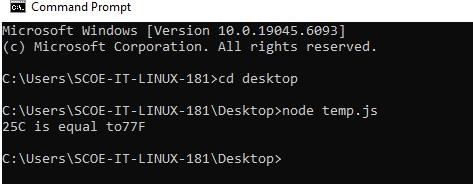


**4.Anonymous function on temperature:** let convertTemperature = function(celsius) { return (celsius \* 9/5) + 32;

};

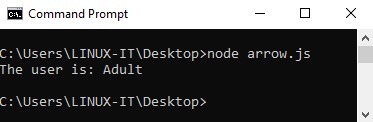
let currentCelsius = 25;

let fahrenheit = convertTemperature(currentCelsius); console.log(currentCelsius + "°C is equal to " + fahrenheit + "°F");



**5.Arrow function on Age:** const verifyAge = age => age >= 18 ? "Adult" : "Minor"; let userAge = 21;

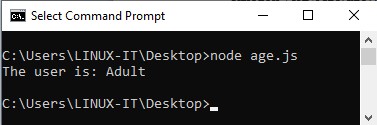
let verificationResult = verifyAge(userAge); console.log("The user is: " + verificationResult);



**6.Generator function on temperature:** function\* verifyAge(age){ yield age >= 18 ? "Adult" : "Minor";

} let userAge = 21;

let verification =verifyAge(userAge); let verificationResult = verification.next().value console.log("The user is: " +verificationResult);



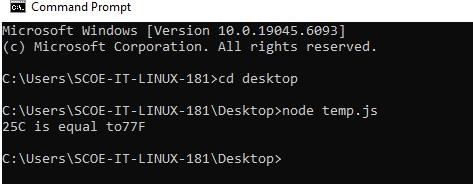
**7.Rest function on Temperature:**

function convertTemperature(...celsiusValues) { return celsiusValues.map(celsius => (celsius \* 9/5) + 32);

}

let currentCelsius = 25;

let [fahrenheit] = convertTemperature(currentCelsius); console.log(currentCelsius + "°C is equal to " + fahrenheit + "°F");



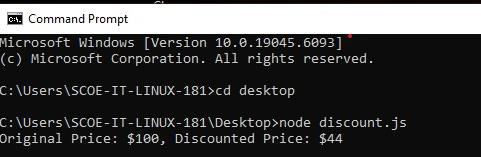
**8.Default parameter on discount:** function calculateDiscount(price,discountPercentage=56) { return price - (price \* discountPercentage / 100);

}

let originalPrice = 100; let discount =20;

let discountedPrice = calculateDiscount(originalPrice);

console.log("Original Price: $" + originalPrice + ", Discounted Price: $"+ discountedPrice);



**Experiment No:4 - note pad & cmd prompt**

**1)**

**Class-Object :** Class Person{ constructor(name,age){ this.name=name; this.age=age;

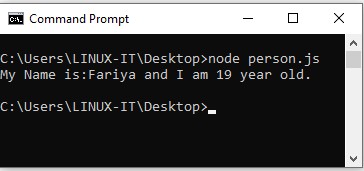
} greet(){

console.log("My Name is:"+this.name+ " and I am " + this.age+ " year old.");

}}

let person1 = new Person ("Fariya","19"); person1.greet();

**OUTPUT:**

****

**Exp 5 - notepad & cmd**

**Inheritance:**

class Animal{

constructor (name){

this.name = name;

}

animalName(){

return `${this.name}`;

}}

class Pets extends Animal{

constructor(name){

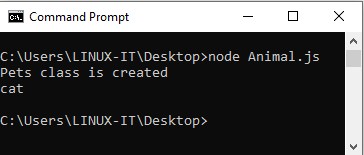
super(name)

console.log("Pets class is created");

}}

let p1=new Pets("cat"); console.log(p1.animalName());

**OUTPUT:**

****

**EXPERIMENT : 06 IP - vs code & react**

**index.js**

**import React from 'react';**

**import ReactDOM from 'react-dom/client';**

**import './index.css';**

**import App from './App';**

**import reportWebVitals from './reportWebVitals';**

**const root = ReactDOM.createRoot(document.getElementById('root'));**

**root.render(**

**<React.StrictMode>**

**<App name="himanshi" age={19} dept="IT" collegename="Saraswati" location="Kharghar" />**

**</React.StrictMode>**

**);**

**App.js**

**import './App.css';**

**import React from 'react'**

**const App=(props)=>{**

**return(**

**<div className="person">**

**<h2>Name:{props.name}</h2>**

**<h2>Age:{props.age}</h2>**

**<h2>Dept:{props.dept}</h2>**

**<h2>College Name:{props.collegename}</h2>**

**<h2>Location:{props.location}</h2>**

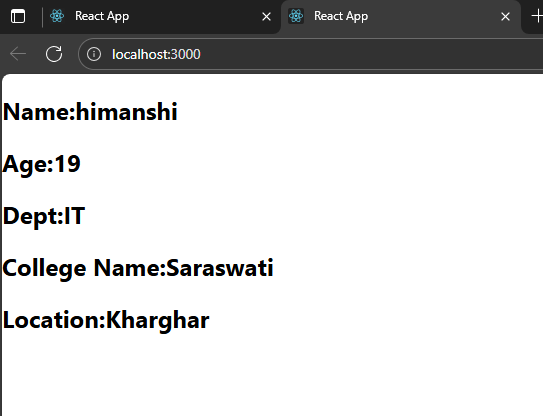
**</div>**

**)**

**};**

**export default App;**

**Output:**

****

**Experiment No - 8**

**Code:**

**StyledComponent.js**

import React, { useState, useEffect } from 'react';

function StyledComponent() {

const [color, setColor] = useState('blue');

useEffect(() => {

const style = document.createElement('style');

style.textContent = `

.dynamic-color {

color: ${color};

} `;

document.head.appendChild(style);

return () => { document.head.removeChild(style);

};

}, [color]);

return (

<div>

<h1 className="dynamic-color">useEffect Example</h1>

<button onClick={() => setColor('red')}>Change to red</button>

<button onClick={() => setColor('green')}>Change color to green</button> </div>

);

}

export default StyledComponent;

Output:

**Experiment No:09**

import react,{useState, useInsertionEffect} from "react";

function StyledComponent(){

const[color, setColor] = useState('blue');

useInsertionEffect(() => {

const style = document.createElement('style');

style.textContent =`

.dynamic-color{

color: ${color};

} `;

document.head.appendChild(style);

return () => {

document.head.removeChild(style);

};

},[color]);

return (

<div>

<h1 className="dynamic-color">useInsertionEffect Example</h1>

<button onClick={() => setColor('red')}>Change to Red</button>

<button onClick={() => setColor('green')}>Change to Green</button>

</div> );}

export default StyledComponent;



**Experiment No: 07**

[**Home.js**](http://home.js)

import react from 'react'; function Home(){ return <h1>Welcome to the world of React Router!!</h1>

}

export default Home;

**Aboutus.js** import react from 'react'; function Aboutus(){ return<div>

<h2> It's a About Us Page!!</h2> read more about us:

<a href='https://www.google.com/'>

</a>

</div>

}

export default Aboutus;

**Contactus.js** import React, {Component} from "react"; class Contactus extends Component{ render(){ return(

<div>

<h2> GOT QUESTIONS??</h2>

<p> The easiest thing to do is post on our

<a href="http://forum.kirupa.com">formus</a>

</p>

</div>

); }} export default Contactus;

[**index.js**](http://index.js)

import React from 'react'; import ReactDOM from 'react-dom/client'; import './index.css'; import App from './App'; import reportWebVitals from './reportWebVitals'; import FinalRoute from './FinalRoute'; const root = ReactDOM.createRoot(document.getElementById('root')); root.render(

<React.StrictMode>

<FinalRoute></FinalRoute>

</React.StrictMode>

);

// If you want to start measuring performance in your app, pass a function

// to log results (for example: reportWebVitals(console.log)) // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals reportWebVitals();

[**FinalRoute.js**](http://finalroute.js)

import React, { Component } from "react"; import { BrowserRouter as Router,Routes,Route,Link} from 'react-router-dom'; import Home from "./home"; import Aboutus from "./Aboutus"; import Contactus from "./Contactus"; import './App.css'; class FinalRoute extends Component{ render(){ return(

<Router>

<div className="App">

<ul className="App-header">

<li>

<Link to ="/">Home</Link>

</li>

<li>

<Link to ="/Aboutus">About us</Link>

</li>

<li>

<Link to ="/Contactus">Contact us</Link>

</li>

</ul>

<Routes>

<Route exact path="/" element={<Home/>}></Route> <Route exact path="/Aboutus" element={<Aboutus/>}></Route>

<Route exact path="/Contactus" element={<Contactus/>}></Route>

</Routes>

</div>

</Router>

); }} export default FinalRoute;

**IP Exp : 10**

**App.js**

import React, { useState, useRef, useEffect } from 'react';

function CountPrev() {

const [count, setCount] = useState(0);

const prevCountRef = useRef();

useEffect(() => {

prevCountRef.current = count;

}, [count] ) ;

return (

<div>

<p> Current count : {count}</p>

<p>Previous count: {prevCountRef.current}</p>

<button onClick={() => setCount(count + 1)}>Increment</button>

</div>

)

}

export default CountPrev ;

**Index.js**

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import App from './App';

import reportWebVitals from './reportWebVitals';

import FinalRoute from './FinalRoute';

import CountPrev from './CountPrev';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

<React.StrictMode>

< App/>

</React.StrictMode>

);

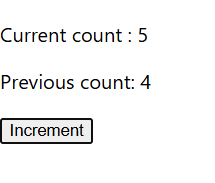
// If you want to start measuring performance in your app, pass a function

// to log results (for example: reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more:<https://bit.ly/CRA-vitals>

reportWebVitals();

**Output :**

****

**IP EXP : 11**

import React, { useState } from 'react';

function login() {

const [inputName, setInputName] = useState('');

const [inputPassword, setInputPassword] = useState('');

const hardcodedName = 'Himanshi';

const hardcodedPassword = '9876';

const handleSignIn = () => {

if (inputName === hardcodedName && inputPassword === hardcodedPassword) {

alert('Signed in');

} else {

alert('Wrong password or name');

}

};

return (

<div>

<h2>Sign In</h2>

<div>

<label>Name: </label>

<input

type="text"

value={inputName}

onChange={(e) => setInputName(e.target.value)}

/>

</div>

<div>

<label>Password: </label>

<input

type="password"

value={inputPassword}

onChange={(e) => setInputPassword(e.target.value)}

/>

</div>

<button onClick={handleSignIn}>Sign in</button>

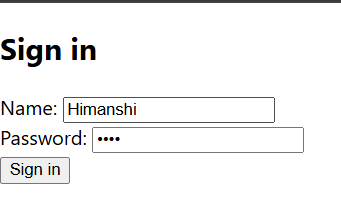
</div>

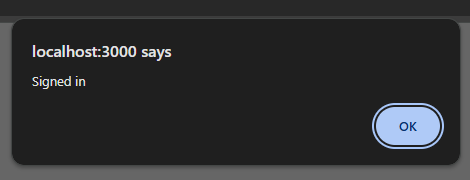
);

}

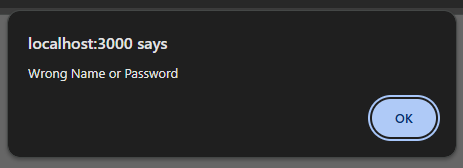
export default login;

**Output :**

****



If the user enters wrong name or password then



Note - how to create folder on react commands