|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Web Technologies** | | 3 | 0 | 0 | 3 |
| **Total Units to be Covered:** | | **Total Contact Hours: 45** | | | | |
| **Prerequisite(s):** | **Basic knowledge of Computer and Designing** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To understand the basics of Web Designing using HTML, DHTML, and CSS.
2. To introduce the fundamentals of Internet, and the principles of web design.
3. To build dynamic web pages with validation using Java Script event handling mechanisms and Angular JS.
4. To understand the development of server-side applications using Node JS.
5. To introduce backend development using server side scripting language.

**Course Outcomes**

On completion of this course, the students will be able to

CO1. Understand web application architecture and can develop basic websites using HTML and Cascading Style Sheets.

CO2. Gain skills in different programming control structures and functions for development of dynamic client-side web applications.

CO3. To build dynamic web pages with validation using Java Script objects and by applying different event handling mechanisms

CO4. Implement a functional front-end web application using Angular JS.

CO5. Understand handling of data and will be able to perform basic database operations.

CO-PO Mapping

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course Outcomes | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
| CO1 | 1 | - | 2 | - | 3 | - | - | - | 2 | - | - | - | 3 | 2 | - |
| CO2 | 1 | - | 2 | - | 3 | - | - | - | 2 | - |  | - | 2 | 2 | - |
| CO3 | 1 | - | 1 | - | 3 | - | - | - | 2 | - | - | - | 3 | 2 | - |
| CO4 | 1 | - | 2 | - | 3 | - | - | - | 2 | - | - | - | 3 | 2 | - |
| Average | 1 |  | 2 | - | 3 | - | - | - | 2 | - | - | - | 3 | 2 | - |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium)

3 – Strongly Mapped (High) “\_” means there is no correlation

**Syllabus**

|  |
| --- |
| **Unit 1 Introduction to HTML No. of Lectures: 8** |
| Web Fundamentals: Internet, Intranet and Extranet, web browsers, web servers, working of search engine. Development Process: Client and Server scripting and languages, HTML tags, W3C Validation service, Headers, HTML form elements and designing web forms, meta tags, CSS and its types, Types of CSS Selectors, CSS properties, CSS Box Model, CSS Positioning (relative, absolute and fixed), Responsive design with CSS, HTML APIs – geolocation, drag and drop. |
| **Unit 2 Introduction to Client-side scripting No. of Lectures: 10** |
| JavaScript: Overview, Syntax, Data types, Enabling JavaScript, variables, functions, control statements, arrays, form validation, JS objects – Date, Math, String, event handling, JS Browser Object Model – window object, history object, navigator object, screen object, HTML Document Object Model. |
| **Unit 3 Angular JS No. of Lectures: 10** |
| Introduction to angular JS, features of angular JS, Architectural concepts, expressions, built-in filters, basic usage of filters, introduction to directives, controllers, and modules, working with angular forms and events, form validation. |
| **Unit 4 Development using Node JS No. of Lectures: 9** |
| Advantages of Node JS, Traditional Web Server Model, Node.js Process Model, Environment Setup, Node JS console, Node JS modules and its types, Functions, Buffer, Understanding Node event driven framework, Event Emitter class, Events and Event Loop, inheriting events, Node Package Manager. |
| **Unit 5 Working with Data No. of Lectures: 8** |
| Introduction to MongoDB: Setting up MongoDB, hosting and authenticating into database, Model Creation, Managing Database Connections, and Performing basic operations – Insert, update, delete, search etc. |

**Textbooks**

1. Steven Holzner,” HTML Black Book”, Dreamtech press.
2. M. Wandschneider, Learning Node.js : a hands-on guide to building Web applications in JavaScript. Upper Saddle River, Nj: Addison-Wesley, 2013.
3. Internet and World Wide Web How to program, P.J. Deitel & H.M. Deitel Pearson
4. AngularJS Essentials – Rodrigo Branas, Packt Publishing Ltd.

**Reference Books**

1. Robert W. Sebesta, “Programming with World Wide Web”, Fourth Edition, Pearson, 2008.

1. Fritz Schneider,Thomas Powell, JavaScript : The Complete Reference 2nd Edition, Tata McGraw - Hill Education
2. D. Herron, Node.js Web Development. Packt Publishing Ltd, 2018.

**Modes of Evaluation:**

Quiz/ Assignment/ presentation/ extempore/ Written Examination

**Examination Scheme:**

|  |  |  |  |
| --- | --- | --- | --- |
| Components | Internal Assessment | MSE | ESE |
| Weightage (%) | 30% | 20% | 50% |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Course Code** | **Course name** | | **L** | **T** | **P** | **C** |
|  | **Web Technologies Lab** | | 0 | 0 | 2 | 1 |
| **Total Units to be Covered:** | | **Total Contact Hours:** | | | | |
| **Prerequisite(s):** | **Basic knowledge of Computer and Designing** | | **Syllabus version: 1.0** | | | |

**Course Objectives**

1. To Understand the basics of web Designing.
2. To develop Dynamic Web Programs using Java Script and Angular JS and NodeJS.

**Course Outcomes**

On completion of this course, the students will be able to:

CO1. Learn markup languages HTML and scripting languages JavaScript.

CO2. Identify the basic Java Script programming structures.

CO3. Develop real world dynamic applications using client and server-side scripting.

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program Outcomes**  **Course Outcomes** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PSO1** | **PSO2** | **PSO3** |
| **CO1** | 1 |  | 2 |  | 1 |  |  |  | 2 |  |  | 2 | 3 |
| **CO2** | 1 |  | 2 |  | 1 |  |  |  | 2 |  |  | 2 | 3 |
| **CO3** | 1 |  | 1 |  | 1 |  |  |  | 2 |  |  | 2 | 3 |
| **Average** | 1 |  | 1.6 |  | 1 |  |  |  | 2 |  |  | 2 | 3 |

1 – Weakly Mapped (Low) 2 – Moderately Mapped (Medium) 3 – Strongly Mapped (High) “\_” means there is no correlation

**List of Experiments**

**Experiment 1: Basic HTML Tags**

Q1. Design a webpage to describe your university. The webpage should have properly aligned paragraphs to show textual information and images wherever required.

Q2. Design a webpage to show your timetable in proper format.

Q3. Create a webpage that should have a menu to show different tourist places in Dehradun. Use different fore ground, background colors, images and other properties.

**Experiment 2: Creating HTML forms using different form elements.**

1. Create a login page using different form elements.
2. Create a HTML form for a student for course registration which should have following fields:
3. Student Name (textbox)
4. Age (textbox with numbers only)
5. Date of Birth (Calendar)
6. Select Course (Drop Down)
7. Submit (Button)
8. Create HTML form for selecting sports in your university. The form should have fields like Name, Password, Address, Select Game, Gender, Age etc.

**Experiment 3: Use of Inline, Internal and External stylesheets and incorporating styles in HTML document.**

1. Use Inline style sheet and create a webpage.
2. Create a CSS based Sticky footer.
3. Using HTML and CSS, create a custom hover and focus effect for navigation items, using CSS transformations.
4. Using HTML, CSS creates a zoom in zoom out animation.

**Experiment 4: Responsive Design with Media Queries**

1. Add a CSS media query and appropriate styles so that the webpage looks similar even when resized to smaller widths. Specifically:
2. The sidebar should be hidden.
3. The body should have no padding.
4. The images shouldn't exceed the width of the window.
5. The navigation items should each be on their own line.
6. The header should be fixed, so that it stays at the top after scrolling.

**Experiment 5: Client-side Programming using Java Script – Use of control statements**

1. Write a JavaScript program that displays the largest integer among two integers.
2. Write a JavaScript function that accepts a string as a parameter and converts the first letter of each word into upper case.
3. Write a Java Script to create a simple calculator.
4. Write a JavaScript function that accepts a string as a parameter and finds the longest word within the string.
5. Write a JavaScript program to find odd and even numbers from 1 to 100.
6. Write a JavaScript program to generate a random string.
7. Write a JavaScript Program to Print All Prime Numbers in an Interval.
8. Write a JavaScript program to populate a drop-down box from 1 to 1000.

**Experiment 6: Java Script Event Handling and Functions**

1. Write a Java script program to Generate a Random Number and display it in a textbox.
2. Write a JavaScript function that changes the background color of an element on a particular event.
3. Write a Java script to validate course registration form.
4. Write a JavaScript program that adds a keydown event listener to a text input to detect when the "Enter key" is pressed.
5. Write a program to show the use of alert, prompt and confirm dialog boxes.
6. Write a JavaScript function to extract a specified number of characters from a string.
7. Write a function to accept date of birth from a user and calculate the difference till current date.

**Experiment 7: Using Angular JS Implement Input Validation**

1. Write a code to change the background of the textbox based on the color mentioned by the user in the same text box.
2. Write a code to create a cost calculator.
3. Write a code to build a simple search filter functionality to display a filtered list based on the search query entered by the user.
4. Use AngularJS Tables to perform the following.

Display a Table

Display contents of table with Order by Filter

Display Table with even and odd.

**Experiment 8: NodeJS basic exercises**

1. Create a simple “Hello, World!” server using Node.js and Express.

2. Write a node.js program to replace two or more a's with the letter b on the given string using Regular Expression.

3. Create a basic calculator that can perform arithmetic operations (addition, subtraction, multiplication, and division) through HTTP requests.

4. Write a node.js code to iterate over the given array.

**Experiment 9: Working with Data**

1. Create a NodeJS application to connect to a MongoDB database.
2. Create an application to store the details of students in a database.
3. Create a search application for finding the students based on given search criteria.
4. Write a program to create an application for a shopping center with all the facilities like add an item, delete an item, update an item detail, stock report, sale etc.

**Textbooks**

1. H. M. Deitel, P. J. Deitel, and A. Deitel, Internet and World Wide Web How to Program. Pearson Education UK, 2014.
2. ‌Web Technologies, Black Book, Dreamtech Press.
3. S. Stefanov, React: Up & Running. “O’Reilly Media, Inc.,” 2021.

‌**Reference Books**

1. J. C. HIATT, REACT JS FOUNDATIONS BUILDING USER INTERFACES WITH REACTJS: an approachable guide. S.l.: JOHN WILEY & SONS, 2020.
2. Jennifer Niederst Robbins, Learning web design: a beginner’s guide to HTML, CSS, Javascript, and web graphics. Sebastopol, Ca: O’reilly, 2018.

**Modes of Evaluation: Continuous Assessment**

**Examination Scheme**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Components** | **Quiz** | **Performance & Viva** | **Lab Report** | **Total** |
| Weightage (%) | 30 | 50 | 20 | 100 |