Pokhara University
Faculty of Management Studies

Course Code: CMP 173
Course title: **Internet Technology**
Nature of the course: Theory + Practical
Year: First, Semester I
Level: Bachelor

Full marks: 100
Pass marks: 45
Credit: 3Hrs
Total periods: 48 hours
Program: BCSIT

1. Course Description

This course introduces students to web technologies, encompassing HTML, CSS, and JavaScript. Students learn web basics, HTML structure, CSS styling, JavaScript programming, and advanced topics. The course emphasizes responsive design, DOM manipulation, asynchronous programming, and practical implementations. Practical sets reinforce skills in creating web content, styling layouts, and enhancing interactivity, culminating in a website project. The course includes understanding the fundamental concepts of web technology, including the Internet, Intranet, and World Wide Web (WWW), and demonstrating proficiency in designing and structuring web content using HTML, creating both static and dynamic web pages. The course explores advanced JavaScript topics, including scope, closures, error handling, DOM manipulation, and asynchronous programming, and develops hands-on experience by creating interactive web interfaces and responsive layouts through practical exercises.

2. General Objectives

- To understand the core concepts of web technology, including the Internet, Intranet, and World Wide Web (WWW).
- To acquire proficiency in designing and structuring web content using HTML, encompassing static and dynamic pages.
- To develop expertise in applying CSS for styling and layout, including typography, colors, backgrounds, and responsive design.
- To master the principles of client-side scripting with JavaScript, covering variables, control flow, functions, arrays, and objects.
- To explore advanced JavaScript topics, including scope, closures, error handling, DOM manipulation, and asynchronous programming.
- To gain hands-on experience with practical exercises to create interactive web interfaces and responsive layouts.
- To apply best practices in coding, optimization, and organization to enhance web development efficiency and performance.
- To cultivate a foundation in modern web technologies, enabling the creation of dynamic and engaging web applications.

3. Method of Instructions

General Instructional Technique: Lecture, Discussion, Readings, Question Answer

Specific Instructional Technique: Practical works, Project Based Learning, Self-Directed Learning, Industry Insights and Case Study

4. Content in Detail with Specific Objectives

Specific Objectives	Contents
● Explore internet essentials, differentiating between Internet, Intranet, and World Wide Web (WWW). ● Distinguish between static and dynamic web pages, and comprehend the roles of web clients and servers. ● Grasp the fundamentals of client-server architecture, including single-tier, two-tier, and multi-tier models. ● Gain insight into HTTP, understanding request and response processes, and interpreting URLs. ● Differentiate client-side scripting from server-side scripting, and recognize their significance in web development. ● Trace the evolution of the web from Web 1.0 to Web 3.0, and understand the evolving nature of internet technologies.	**Unit I: Introduction to web technology [3 Hrs.]** 1.1 Web Basics: 1.1.1 Internet, Intranet, WWW 1.1.2 Static and Dynamic Web Page 1.1.3 Web Clients 1.1.4 Web Servers 1.2 Client Server Architecture: 1.2.1 Single Tier 1.2.2 Two-Tier 1.2.3 Multi-Tier; 1.3 HTTP: HTTP Request and Response 1.4 URL 1.5 Client Side Scripting 1.6 Server Side Scripting 1.7 Web 1.0,Web 2.0 and web 3.0
Specific Objectives	**Contents**
● Understand HTML's role in web development and its key components. ● Comprehend the basic structure of HTML documents, including head and body sections. ● Identify and use tags, elements, and attributes within HTML. ● Apply proper HTML doctype declaration for consistent rendering. ● Utilize meta tags to specify character encoding, viewport, and other metadata. ● Format content using heading, paragraph, strong, em, underline, and strikethrough tags. ● Implement line breaks, and horizontal rules, and create hyperlinks with anchor tags. ● Establish navigation menus, and lists, and organize content effectively. ● Insert images and multimedia elements, setting attributes like src, alt, width, and height. ● Construct tables, including headers, data cells, merging cells, and adding captions. ● Create interactive forms with input elements, validate user input, and handle form submission.	**Unit 2: Hyper Text Markup Language [8 Hrs.]** 2.1 Introduction to HTML 2.2 Document Structure 2.3 Text Formatting 2.4 Links and Navigation 2.5 Hyperlink 2.6 Images and Multimedia 2.7 Lists, Tables, Forms and Input 2.8 Semantic HTML

Specific Objectives	Contents
● Create graphics using the HTML5 canvas element. ● Explore new form input types, including email, URL, date, time, and range. ● Master form validation using built-in attributes and placeholder text. ● Adhere to proper code indentation, formatting, and consistent naming conventions.	**Unit 3: HTML5 [7 Hrs.]** 3.1 HTML5 APIs 3.2 HTML5 Forms 3.3 Responsive Web Design 3.4 HTML5 3.5 Semantic Markup 3.6 Best Practices and Optimization
Specific Objectives	**Contents**
● Understand CSS's role in web development and its syntax. ● Apply CSS rules inline, internally, and externally. ● Master various CSS selectors, including elements, classes, IDs, and pseudo-classes. ● Grasp the concept of the box model, dimensions, and box-sizing property. ● Style typography with fonts, sizes, colors, alignment, and decorations. ● Apply diverse color formats and configure backgrounds with images and properties. ● Control element positioning using static, relative, absolute, and fixed positioning. ● Implement floating elements and handle clearing and clearfix techniques. ● Utilize display property for versatile element presentation. ● Acquire skills in creating layouts and enhancing webpage aesthetics with CSS. ● Understand viewport Meta tags for responsive design. ● Implement media queries for diverse screen sizes and responsiveness. ● Design responsive layouts using CSS for optimal viewing on various	**Unit 4: Cascading Style Sheets [8 Hrs.]** 4.1 Introduction to CSS 4.2 CSS syntax 4.3 Using CSS with HTML 4.4 CSS Selectors 4.5 CSS Comments 4.6 CSS Properties 4.6.1 Backgrounds 4.6.2 Border, Margin 4.6.3 Padding 4.6.4 Height 4.6.5 width 4.6.6 color(color wheel) 4.7 Text 4.8 Font 4.9 Alignment 4.10 Line Height 4.11 Box Model 4.12 working with images 4.13 Layout and Positioning 4.14 Media query 4.15 CSS website Layout
Specific Objectives	**Contents**
● Develop advanced proficiency in CSS, covering flexbox, grid layouts, transitions, animations, responsiveness, specificity, units, preprocessors, and optimization. ● Master CSS Flexbox and Grid for versatile layout control. ● Apply transitions and animations using various properties and keyframes. ● Create responsive designs using media queries,	**Unit 5: Advance Topics on CSS [7 Hrs.]** 5.1 CSS Flexbox 5.2 CSS Grid 5.3 CSS Transitions and Animations 5.4 Responsive Web Design 5.5 CSS Specificity and Inheritance 5.6 CSS Units and Values 5.7 CSS Preprocessors 5.8 CSS Best Practices and

breakpoints, and the mobile-first approach. ● Understand CSS specificity and inheritance principles for effective styling. ● Utilize different CSS units and values, including relative and absolute units. ● Explore CSS preprocessors, employing nesting, variables, mixins, and functions. ● Implement CSS best practices, including efficient organization, vendor prefixes, and optimization techniques. ● Enhance web development skills with advanced CSS techniques.	Optimization
Specific Objectives	**Contents**
● Understand JavaScript's role in web development and its syntax. ● Declare variables and utilize data types: numbers, strings, Booleans, null, undefined, objects, and arrays. ● Apply arithmetic, comparison, and logical operators to manipulate data. ● Implement control flow through if statements, switch statements, and ternary operators. ● Utilize loops such as while, for, and do-while for iterative processes. ● Define, call, and manage functions, including parameters, arguments, and return values. ● Create and manipulate arrays, using methods like push, pop, shift, unshift, splice, slice, and iteration techniques. ● Work with objects, access properties, employ methods, and understand object constructors and prototypes. ● Develop practical client-side scripting skills using JavaScript to enhance web interactivity and functionality.	**Unit 6: Client Side Scripting with JavaScript [8 Hrs.]** 6.1 Introduction to JavaScript 6.2 Using JS in HTML 6.3 JavaScript Output 6.4 JavaScript Comments 6.5 Variables and Data Types 6.6 Operators and Expressions 6.7 Control Flow and Conditionals 6.8 Loops 6.9 Functions 6.10 Arrays 6.11 Objects
Specific Objectives	**Contents**
● Grasp the intricacies of scope, variable visibility, and the concept of closures. ● Manage errors using try-catch blocks, exceptions, and employ debugging techniques. ● Manipulate the Document Object Model (DOM) to interact with HTML elements and handle events. ● Master asynchronous programming, including callback functions, promises, and async/await. ● Work with JSON data and make AJAX requests using fetch API or XMLHttpRequest. ● Explore ES6 features like arrow functions, template	**Unit 7: Advance Topics on JavaScripts [7 Hrs.]** 7.1 Scope and Closures 7.2 Error Handling and Debugging 7.3 DOM Manipulation 7.4 Asynchronous JavaScript 7.5 JSON and AJAX 7.6 ES6 and Modern JavaScript 7.7 JavaScript Libraries

literals, let and const keywords, and destructuring assignments. ● Understand modern JavaScript concepts, such as modules and import/export functionality. ● Gain optional exposure to popular JavaScript libraries/frameworks like React, Angular, or Vue.js for building web applications. ● Elevate your JavaScript skills to enable dynamic and interactive web development.	

5. Laboratory Work

It builds the foundation for how to write a program using any high-level language. Hence, this course requires a lot of programming practice so that students will be able to develop good logic building and program developing capability which is essential throughout the course.
Some important contents that should be included in lab exercises are as follows:

1. Creating a simple static web site with 4 pages, using HTML5 (include all tags included in HTML and HTML 5
2. Create a simple image gallery using CSS
3. Create a responsive web page using box model
4. Create a form with all the elements and validate it using client-side scripting
5. Using JavaScript create a simple calculator
6. Create a user registration form using HTML 5 and validate it using JavaScript
7. Creating jQuery Slider and Image Gallery
8. Use jQuery date picker and sort
9. Work with JSON data and make AJAX requests using fetch API or XMLHttpRequest
10. General concept of React, Angular, or Vue.js for building web applications

Note:

1. Motivate students to create small project work integrating all of the above concepts.
2. Each of the above lab session should cover more than 4 hours of practical work.

6. Evaluation system and Student's Responsibility

In addition to the formal exam(s), the internal evaluation of a student may consist of quizzes, assignments, lab reports, projects, class participation, etc. The tabular presentation of the internal evaluation is as follows.

External Evaluation	**Marks**	**Internal Evaluation**	**Weight**	**Marks**
Semester-End examination	50	**Theory**		30
		Attendance & Class Participation	10%	
		Assignments	20%	
		Presentations/Quizzes	10%	
		Internal Assessment	60%	
		Practical		
		Attendance & Class Participation	10%	20

		Lab Report/Project Report	20%	
		Practical Exam/Project Work	40%	
		Viva	30%	
Total External	50	Total Internal		50
Full Marks: 50 + 50 = 100				

7. Student's Requirements

Each student must secure at least 45% marks separately in both internal assessment and practical evaluation with 80% attendance in the class in order to appear in the Semester End Examination. Failing to get such a score will be given NOT QUALIFIED (NQ) to appear in the Semester-End Examinations. Students are advised to attend all the classes, formal exams, tests, etc., and complete all the assignments within the specified time period. ***Students are required to complete all the requirements defined for the completion of the course.***

8. Prescribed Books and References

Prescribed Text Books:

1. Robbins, J. N. (2018). *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics*. O'Reilly Media.

Reference Books:

1. Holzner, S. (2000). *HTML Black Book*. John Wiley & Sons.
2. Inc, K. L. S. (2009). *Web Technologies: Html, Javascript, Php, Java, Jsp, Asp.Net, Xml And Ajax, Black Book (With Cd)*. Wiley India
3. Knuckles. (2006). *Web Applications: Concepts & Real World Design*. John Wiley & Sons.
4. Deitel, P. J., Deitel, H. M., & Deitel, A. (2012). *Internet and world wide web: How to Program*. Prentice Hall.