Pokhara University
Faculty of Science and Technology

Course Code: CMP 418 (3 Credits) Full Marks: 100
Course Title: ***Advanced Web Technology*** **(Elective)** Pass Marks: 45
Nature of the Course: Theory & Practical Total Lectures: 45
Level: Bachelor Program: BE Computer

1. Course Description

This course is designed to provide students with hands-on knowledge of client side and server-side development technology. After completing this course students will not only have theoretical knowledge but also practical, hands-on experience in building, securing, and deploying web applications, making them well-prepared for the challenges of professional web development. This course covers essential web technologies topics on client side such as web basics, HTML, CSS, and JavaScript, and server-side programming and framework such as Python Basics, OOP's, flask web frameworks, database integration using SQLAlchemy, developing and consuming REST services, and deployment.

2. General Objectives

The course is designed with the following objectives:

- To introduce the basic concepts of web development using python programming.
- To provide knowledge and skills on how to create web pages using HTML, including working with elements, attributes, and various components like text formatting, headings, lists, tables, links, images, and forms, etc.
- To enable students to style and layout web pages using CSS, covering basics, the CSS box model, responsive design, and advanced layout techniques like Flexbox and Grid.
- To build students' JavaScript skills, including how to write scripts, handle events, validate form, etc.
- To provide knowledge about the fundamentals of server-side development using python programming language.
- To enable students to develop web applications using flask web framework and manage databases in web applications.
- To develop skills in creating and consuming REST services and APIs.
- To prepare students for the deployment of web applications to production environments.

3. Methods of Instruction:

- Lecture, Class discussion, Practical's, Demonstration, Presentation and Project Work

4. Content in Details	
Specific Objectives	**Contents**
- Understand history and the concept of web technology. - Understand the basic of web protocols, DNS, web hosting, and client server scripting	**Unit 1: Overview of Web Technology (3hrs.)** 1.1 Introduction to WWW, URL, HTTP, HTTPS 1.2 The Evolution of Web Technologies 1.3 Website and Web Application 1.4 Web Client and Web Server 1.5 HTTP Request and Response 1.6 Client side and server-side scripting language 1.7 DNS and its hierarchy 1.8 Web Hosting
- Apply the HTML structure, tags and attributes in web design. - Implement lists, links, images, tables, and forms, and use semantic HTML for better web organization.	**Unit 2: HTML (7hrs.)** 2.1 Introduction 2.1.1 History and evolution of HTML 2.1.2 Concept and types of tag 2.1.3 HTML document structure 2.1.4 HTML elements and attributes 2.1.5 Basic text formatting tags 2.2 Lists, Links 2.2.1 Unordered, ordered and description lists 2.2.2 Creating hyperlinks, types of links 2.3 Adding images to a web page 2.4 Table and Forms 2.4.1 html table elements and attributes, merging row and columns 2.4.2 Common form elements like text fields, text areas, selects, buttons, radio button. 2.4.3 input date, time 2.4.4 form methods 2.5 Semantic HTML 2.5.1 Header, footer, nav, article, section 2.6 Audio and video elements 2.7 div, span
- Implement CSS including its syntax, box model, and layout techniques. - Understand the need for responsive web	**Unit 3: CSS (4hrs.)** 3.1 Introduction to CSS and CSS3 3.2 CSS Syntax, Selector and types (Internal, External, Inline) 3.3 CSS Properties 3.4. Box Model 3.5. Flexbox and Grid Layouts 3.6. Responsive Design Principles
- Explain the core concepts of JavaScript through variables, data types, and	**Unit 4: JavaScript Basics (6hrs.)** 4.1 Introduction to JavaScript 4.2 Embedding JavaScript to a webpage

control structures, and to show how to use functions, manipulate the DOM, and handle events to make web pages interactive. - Introduce JSON and its uses	4.3 Variables, Data Types, Control flow statement, Operators etc. 4.4 Functions and Scope 4.5 Dialog boxes 4.6 Document Object Model 4.7 Form Validation and Event handling 4.8 Introduction to JSON and its uses
- Explain about Python data types, functions, and control structures. - Explore lists, dictionaries, files, and understand OOP concepts. - Implement CSV and JSON files	**Unit 5: Introduction to Server-side Programming (9hrs.)** 5.1 Introduction to server-side programming languages (Python, PHP, Java) 5.2 Setting up the Python environment 5.3 Introduction to Python syntax and basic operations 5.4 Data Types and Variables 5.5 Control Structures: 5.5.1 Conditional statements: if, elif, else 5.5.2 Loops: for and while 5.6 Introduction to functions and its types 5.7 Scope and global variables 5.8 Working with Python Data Structures 5.8.1 Lists 5.8.2 Tuples 5.8.3 Sets 5.8.4 Differences between lists, tuples, and sets 5.8.5 Dictionaries 5.9 File Handling in Python 5.9.1 Introduction 5.9.2 Opening and Closing Files 5.9.3 Reading/Writing to/from Files 5.10 Working with CSV and JSON 5.11 Introduction to OOPs 5.11.1 Class 5.11.2 Object 5.11.3 Inheritance 5.11.4 Polymorphism etc.
- Describe the importance of framework in web development and MVC architecture - Implement HTTP requests, form submissions, cookies, and sessions	**Unit 6: Working with Web Framework (13 Hours)** 6.1 Overview of Web Framework & MVC (Model-View-Controller) Architecture 6.2. Introduction to Python Flask Framework 6.3. Setting Up a Flask Environment 6.4. Flask Basics: Routes, Views and Templates 6.5. HTTP Request and Response Handling 6.6. Building Web Functionalities 6.6.1. Implementing Form Submission and Input Validation

- Explore Flask features like file uploads and sending emails - Implement web application with database Implement basic CRUD operations and manage database transactions. - Explore user authentication and authorization. - Implement REST API	6.6.2. Managing Session and Cookies 6.7. Flask Redirect, Message Flashing, File Upload 6.8. Sending Email 6.9. Database Functionality (CRUD) using SQLAlchemy 6.10. User Authentication and Authorization 6.11 Webservice and RESTful Webservice 6.12. Creating and Consuming RESTful API using Flask
- Explore the development and deployment platforms - Apply security measures in deployment of web app or website	**Unit 7: Web Application Deployment (3hrs.)** 7.1. Introduction to Deployment Platform 7.2. Basic Difference Between Development and Production Environment 7.3. Security Factor During Deployment 7.4. Introduction to Gunicorn to Serve Flask Application

5. Practical Works:

Part I: Client-side Lab Works:

- Install VS Code, PyCharm or any other Web IDE

HTML lab tasks:

- HTML basic page to show greeting message
- In the head section of a web page, make sure to:
    - Add a title tag to set the page's title, which appears in the browser tab.
    - Link to external stylesheets using relative paths to apply the desired styles.
    - Include meta tags to specify important information such as the character encoding, description, keywords, author, and viewport settings for responsive design.
- In the body of a web page, include:
    - A table with headers, rows, and cells, adjusting attributes for spanning and sizing.
    - Text, images, sound clips, and videos with proper controls, sizing, and alternate text.
    - The `<div>` tag for styling and layout.
    - Text elements using tags like `<h1>`, `<p>`, and `<li>`.
    - Classes and styles for formatting, including lists.
- Create hyperlinks from text and images to:
    - bookmarks on the same page
    - other locally stored web pages
    - a website using the URL
    - send mail to a specified email address
    - to open in a specified location (the same window, a new window)
- Create form for:

    - Login and signup purpose
    - Design registration form including Textbox, TextArea, Select, Checkbox, Radio button, Button elements and group them using fieldset and legend.
- Use elements to break down in different blocks.

CSS Labs:

- Set background properties including colour and images, font properties, style table including size, background colour, horizontal and vertical alignment, spacing, padding, borders.
- Create external styles to be tagged in a web page including h1, h2, h3, p, li.
- Attach comments to an external stylesheet.
- Center a div with its elements.
- Design navigation menu for web page with proper styllying, when user put mouse over the menu the menu should be highlited.
- Apply css to image, links, list
- Create product grid including product name, price, image, add to cart button and add effects when user put cursor on the product. Also make it responsive.

JavaScript:

- Link an external JavaScript file and manipulate DOM elements.
- Add and remove textbox from web page with add and delete button.
- Write a script to input information like Name, Address, Contact, Age and display them in proper format when user clicks the submit button.
- Validate form inputs with JavaScript
    - Validate login and signup form. The password should contain atleast letters, symbol and a number and should be minimum 10 characters, username field cannot be blank, to sign up user must be audult.

VCS:

- Start a Git repository with **git init**, stage files with **git add**, and commit with **git commit** make some changes and push with **git push**

UI/UX:

- Prepare Low-Fidelity wireframe of products and product details page.

Part II: Server-side Lab Works

Fundamentals Part:

- Install & Setup Python Environment

- Create a simple script that performs basic arithmetic operations.

- Practice using data types like lists, dictionaries, and sets. Write a Python program to create a list of integers, append values, sort the list, and remove duplicates.

- Write a Python program that stores student names as keys and their marks as values in a dictionary. Print the average marks and the name of the top scorer.
- Perform operations like slicing a string and list manipulation.
- Practice control structures using if-else conditions.
- Write a Python function that takes two numbers as input and returns their sum, difference, product, and quotient.
- write program using recursion technique.
- Perform basic read and write operations.
- Write a Python program to read data from a CSV file and display the content. Then, modify the program to write the output to a JSON file.
- Write a Python class called Student with attributes like name, roll number, and marks. Create methods to display student details and calculate the average marks.
- Create a base class Vehicle with attributes like make and model. Derive a class Car from Vehicle with additional attributes like mileage and capacity. Create objects of child classes and display their details.

Framework Part:

=> Install and configure a local web server (Apache or Nginx). Create a simple HTML. page and serve it via the web server.

=> HTTP Request and Response Simulation

1. Install flask and understand directory structure.
2. Create a basic Flask application with a route that returns “Hello, World!” when accessed via the browser.
3. Develop a web form using HTML and Flask to capture user input (e.g., name, email) and display the input on another page.
4. Installation and seting MySql database server.
5. Create a simple MySql database and connect it to a Flask application. Develop a form that inserts user data into the database.
6. Write a Flask application that connects to a MySQL database using mysql-connector or SQLAlchemy. Create a table called students with columns id, name, age, and grade. Insert records and fetch data from the database to display on a webpage.

7. Implement HTML Dropdown to display dynamic options, Dynamic HTML tables, product list and product details using the flask.
8. Create a Flask app that allows users to add, update, and delete records from a database using a web interface.
9. Sending an email using flask.
10. Template integration
11. Use Bootstrap with Flask to style the CRUD application developed in the above task. Create a form with proper layout and design. Use Bootstrap classes for buttons, tables, and forms.
12. Implement input validation for the form created in the previous exercise.
13. Develop a login page using Flask that includes session management and secure password storage using hashing
14. Develop a REST API using Flask that performs CRUD operations on a student's table in the MySQL database. Implement API endpoints for creating, retrieving, updating, and deleting student records.
15. Use Python's requests module to fetch data from an open API (e.g., weather or currency exchange API) and display the results.

6. Evaluation System and Students' Responsibilities:

6.1 Evaluation System:

In addition to the formal exam(s) conducted by the Office of the Controller of Examination of Pokhara University, the internal evaluation of a student may consist of class attendance, class participation, quizzes, assignments, presentations, written exams, etc. The tabular presentation of the evaluation system is as follows.

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		**30**	**Semester End Examination**	**50**
Attendance / Class Participation	10%			
Assignments	20%			
Project Work/Presentations	10%			
Term Exam	60%			
Practical		**20**		
Attendance and Lab Participation	10%			
Lab Report	20%			
Lab Examination	40%			

Viva Examination	30%			
Total Internal Marks		**50**		
Full Marks=50+50				

6.2 Students' Responsibilities:

To be eligible for the Semester End Examinations, each student must secure at least 45% marks in the internal evaluation with 80% attendance in the class to appear in the Semester End Examination. Failing to obtain such a score will be given NOT QUALIFIED (NQ) and the student will not be eligible to appear in the End-Term examinations. Students are advised to attend all the classes and complete all the assignments within the specified time period. If a student does not attend the class(es), it is his/her sole responsibility to cover the topic(s) taught during the period. If a student fails to attend a formal exam, quiz, test, etc. there won't be any provision for a re-exam.

7. Prescribed Books and References:

1. Felke-Morris, T. (2020). *Web development and design foundations with HTML5*. Pearson.
2. Simpson, K. (2021). *You don't know JS yet: ES6 & beyond*.
3. Independently published Chacon, S., & Straub, B. (2014). *Pro Git*. Apress.
4. Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python (2nd ed.). O'Reilly Media.
5. Beazley, D. (2020). *Python distilled*. Addison-Wesley.