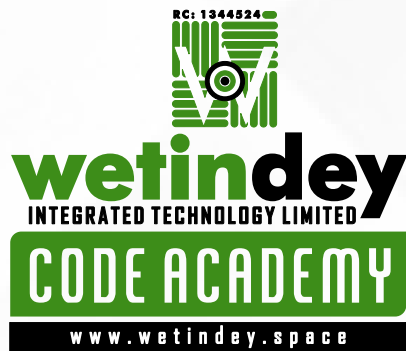




EXPERT LED COURSE CURRICULUM

WEB/SOFTWARE

DEVELOPMENT

...Coding with Passion!

Overview

The Comprehensive **Web/Software Development** Training program is designed to equip aspiring developers with the essential skills and knowledge to excel in the dynamic world of web and software development.

This intensive practical course covers a wide range of topics, from foundational HTML and CSS to advanced backend scripting and API integration.

With hands-on projects and real-world applications, students will gain practical experience in designing and building interactive websites, harnessing the power of databases, and integrating APIs for seamless functionality. In addition, the program offers addon courses in graphic design and digital marketing to provide a well-rounded skill set for creating, promoting, and maintaining successful web projects.

Estimated Time

This course will span over 12 weeks which includes the timeframe for your project, which will be given after the completion of all modules.

Prerequisites

- **Basic computer literacy:** Students should have a comfortable understanding of using computers, browsing the internet, and working with files and folders.
- **Familiarity with web browsing:** An ability to navigate websites and a basic understanding of how the internet works is beneficial.
- No prior knowledge of programming is assumed for this course.

Technical Requirements:

- **Personal computer:** Access to a computer (laptop or desktop) is required for practicing coding, running development environments, and completing assignments.
- **Operating system:** Windows, macOS, or Linux are suitable operating systems for web development.
- **Internet connection:** A stable and reliable internet connection is necessary for accessing online resources, tutorials, and collaboration tools.

Software and Tools:

- **Text editor** (any text editor of your choice) Recommended editors include Visual Studio Code, Sublime Text, or Atom.
- **Web browser:** A modern web browser like Google Chrome, Mozilla Firefox, or Microsoft Edge is necessary for testing and previewing web pages.
- **Optional:** For graphic design and image editing, access to Adobe Fireworks, CorelDRAW, and Photoshop

Week 1: Introduction to Web Development

Upon completing the "**Introduction to Web Development**" module, students will be able to:

LEARNING OUTCOMES

Understand Web Development Fundamentals:

- Define web development and its role in creating websites and web applications.
- Differentiate between front-end and back-end development and explain their respective functions.

Navigate Development Environments:

- Set up a basic development environment, including installing and configuring an Integrated Development Environment (IDE) and text editor.
- Familiarize themselves with the tools required for writing, testing, and debugging code.

Explore HTML and Structure:

- Explain the purpose of HTML (Hypertext Markup Language) in structuring web content.
- Identify key HTML elements, tags, and attributes used to create the basic structure of a web page.

Introduction to CSS Styling:

- Describe the role of CSS (Cascading Style Sheets) in styling and formatting web content.
- Understand the concept of selectors, properties, and values to style HTML elements.

Interact with Web Pages:

- Use a web browser to view and interact with web pages.
- Explain the relationship between HTML, CSS, and JavaScript in creating dynamic and interactive websites.

Basic Troubleshooting:

- Identify common issues that may arise while setting up a development environment or writing code.
- Apply basic troubleshooting techniques to identify and resolve simple errors.

Explore Further Learning Opportunities:

- Recognize the importance of continuous learning in the field of web development.
- Explore additional resources, online communities, and learning platforms to enhance their skills.

Hands-on Project: Basic Web Page Creation:

- Create a simple web page using HTML and apply basic CSS styling to elements.
- Embed images and links to enhance the content of the web page.
- Preview the web page in a browser to ensure proper rendering and functionality.

Week 2: HTML 5 (Hypertext Markup Language)

Upon completing the "HTML 5 (Hypertext Markup Language)" module, students will be able to:

LEARNING OUTCOMES

Understand HTML's Role:

- Explain the fundamental role of HTML in structuring content on the web.
- Differentiate between HTML and other web technologies like CSS and JavaScript.

Create Structured Documents:

- Create well-structured web documents using appropriate HTML elements, tags, and attributes.
- Construct headings, paragraphs, lists, and other essential elements to organize content.

Link Resources:

- Utilize anchor (<a>) tags to create hyperlinks that connect web pages and external resources.
- Understand relative and absolute URLs for linking within the same website and to external websites.

Embed Media:

- Embed images () and multimedia content like audio and video using appropriate HTML tags.
- Set attributes to control image sizes, alt text, and media playback.

Construct Forms:

- Build interactive forms using form elements (<form>, <input>, <textarea>, <select>, etc.).
- Define form controls and attributes for collecting user input and data submission.

Use Semantic HTML:

- Understand the concept of semantic HTML and its importance for accessibility and SEO.
- Apply semantic elements like <header>, <nav>, <main>, and <footer> to enhance page structure.

Apply Basic Styling:

- Recognize the distinction between content and presentation by using inline styles and internal/external CSS for basic styling.
- Apply CSS classes and IDs to target specific HTML elements for styling purposes.

Semantic HTML5 Elements:

- Identify and utilize HTML5 semantic elements like <article>, <section>, <aside>, and <figure> for improved document structure.

Form Validation and Input Types:

- Implement form validation using HTML attributes and attributes like required and pattern.
- Explore different input types (<input type="text">, <input type="email">, <input type="date">, etc.) and their purposes.

Hands-on Project: Web Page with Structured Content:

- Create a comprehensive web page that demonstrates proper use of HTML elements for structured content.
- Include hyperlinks, images, and a form with various form controls and validation.

Week 3: CSS 3 (Cascading Style Sheets)

Upon completing the "CSS 3 (Cascading Style Sheets)" module, students will be able to:

LEARNING OUTCOMES

Understand the Role of CSS:

- Explain the importance of CSS in styling and formatting web content.
- Differentiate between inline, internal, and external CSS and their applications.

Apply Styles to HTML Elements:

- Use CSS selectors to target and apply styles to specific HTML elements.
- Modify properties like colors, fonts, margins, padding, and borders to enhance visual presentation.

Box Model and Layout Design:

- Understand the box model and its components (content, padding, border, margin).
- Utilize box model properties to control spacing and positioning of elements on the web page.

Responsive Design with Media Queries:

- Apply media queries to create responsive layouts that adapt to different screen sizes.
- Implement CSS rules to modify content display based on breakpoints.

Styling Text and Typography:

- Control typography by adjusting font properties (family, size, weight, style, spacing).
- Style text with emphasis, decoration, and alignment to improve readability.

Working with Colors and Backgrounds:

- Apply various color formats (hex, RGB, named colors) to elements.
- Set background colors, images, and gradients for elements.

CSS Box Properties:

- Utilize CSS properties like display, position, float, and clear for advanced layout control.
- Understand the difference between block, inline, and inline-block elements.

CSS Flexbox and Grid Layouts:

- Apply CSS Flexbox and Grid layout systems to create complex, responsive layouts.
- Arrange and align content using flex containers and grid containers.

CSS Transitions and Animations:

- Create smooth transitions and animations using CSS properties like transition and animation.
- Apply keyframes and timing functions to control animation behavior.

Hands-on Project: Stylish and Responsive Web Page:

- Implement a comprehensive CSS stylesheet for a given HTML webpage.
- Design a responsive layout that adapts to different screen sizes using media queries.
- Incorporate typography, colors, backgrounds, and box model properties to enhance visual appeal.

Week 4: JavaScript

Upon completing the "JavaScript" module, students will be able to:

LEARNING OUTCOMES

Understand JavaScript's Role:

- Explain the purpose of JavaScript in adding interactivity and dynamic behavior to web pages.
- Differentiate between client-side and server-side scripting.

Variables and Data Types:

- Define variables and their use in storing data.
- Identify and work with different data types, including numbers, strings, booleans, and arrays.

Control Structures:

- Implement control structures like conditional statements (if, else if, switch) to make decisions in code execution.
- Utilize loops (for, while) for repetitive tasks and iterations.

Functions and Scope:

- Define functions to encapsulate reusable blocks of code.
- Understand variable scope (global vs. local) and apply best practices for variable declaration.

DOM Manipulation and Event Handling:

- Access and manipulate the Document Object Model (DOM) to interact with HTML elements.
- Attach event listeners to respond to user interactions (clicks, input, etc.) and trigger actions.

Arrays and Objects:

- Create and manipulate arrays to store and manage collections of data.
- Understand object-oriented concepts and work with objects, properties, and methods.

Asynchronous Programming and Callbacks:

- Explain the concept of asynchronous programming and its importance for non-blocking code execution.
- Implement callbacks and use them to handle asynchronous operations.

Debugging and Error Handling:

- Identify and fix common programming errors and bugs.
- Use browser developer tools and debugging techniques to troubleshoot issues.

Hands-on Project: Interactive Web Page Elements:

- Develop a web page with interactive features using JavaScript.
- Create form validation scripts to ensure proper user input.
- Implement event listeners to respond to user actions and update page content dynamically.
- Create a complete calculator using HTML, CSS and JAVASCRIPT

Week 5: Bootstrap Framework

Upon completing the "**Bootstrap Framework**" module, students will be able to:

LEARNING OUTCOMES

Understand Bootstrap's Role:

- Explain the purpose of the Bootstrap framework in creating responsive and visually appealing websites.
- Describe how Bootstrap simplifies the design and development process.
-

Grid System and Layouts:

- Understand the Bootstrap grid system and its use for creating responsive layouts.
- Divide the webpage into columns and rows, adjusting layout for different screen sizes.

Bootstrap Components:

- Identify and utilize common Bootstrap components such as navigation bars, buttons, modals, carousels, and forms.
- Customize and style these components to match the design of the web page.

Responsive Design:

- Apply responsive design principles using Bootstrap's classes to create a consistent user experience across devices.
- Use media queries and responsive utilities to adapt content for various screen sizes.

Styling and Theming:

- Apply Bootstrap's built-in styles and themes to enhance the visual aesthetics of the webpage.
- Customize Bootstrap styles to match the branding and design requirements.

Typography and Icons:

- Utilize Bootstrap typography classes for consistent and attractive text formatting.
- Incorporate Bootstrap's icon library to enhance user interface elements.

Navigation and Menus:

- Create responsive navigation menus using Bootstrap's navigation components.
- Design mobile-friendly navigation that collapses on smaller screens.

Bootstrap and JavaScript Interaction:

- Understand how Bootstrap components interact with JavaScript for enhanced functionality.
- Implement Bootstrap's JavaScript plugins for modal dialogs, tooltips, and more.

Building Complete Web Pages:

- Combine Bootstrap's grid system, components, and styles to create complete, responsive web pages.
- Develop interactive elements using Bootstrap's JavaScript features.

Hands-on Project: Responsive Website Design:

- Design and develop a complete responsive website using the Bootstrap framework.
- Incorporate a variety of Bootstrap components, styles, and layout techniques.
- Ensure seamless navigation and optimal user experience on different devices.

Week 5-6: Web Hosting

Upon completing the "**Web Hosting**" module, students will be able to:

LEARNING OUTCOMES

Understand Web Hosting Basics:

- Define web hosting and its role in making websites accessible on the internet.
- Differentiate between different types of web hosting services, including shared hosting, VPS, and cloud hosting.

Choose a Hosting Provider:

- Evaluate and compare various web hosting providers based on features, pricing, and customer support.
- Select an appropriate hosting plan that meets the needs of a specific project or website.

Domain Registration and DNS Management:

- Register a domain name and understand the domain registration process.
- Configure Domain Name System (DNS) settings to point a domain to a web hosting server.

Control Panel and Server Management:

- Navigate and utilize a web hosting control panel (e.g., cPanel, Plesk) for managing domains, databases, files, and email accounts.
- Upload and manage website files using File Transfer Protocol (FTP).

Database Creation and Management:

- Create and manage databases using tools provided by the hosting control panel.
- Understand the importance of secure database management and user permissions.

Backup and Security Practices:

- Implement regular backup strategies to safeguard website data and files.
- Implement basic security measures, such as using strong passwords and configuring security settings.

Email Setup and Management:

- Set up and configure email accounts associated with a hosted domain.
- Understand email protocols (SMTP, IMAP, POP3) and how to manage email accounts.

Web Hosting Troubleshooting:

- Identify and troubleshoot common web hosting issues, such as domain propagation, server errors, and email problems.
- Utilize hosting provider resources and support channels for problem resolution.

Website Deployment and Maintenance:

- Deploy a website from a development environment to a live web hosting server.
- Implement version control and continuous integration practices for seamless updates and maintenance.

Hands-on Project: Hosting a Live Website:

- Register a domain name and select a suitable web hosting provider and plan.
- Configure DNS settings to point the domain to the hosting server.
- Upload website files, set up databases, and deploy a live website.



Week 5-6: Adobe Fireworks (Web Graphic)

Upon completing the "Adobe Fireworks (Web Graphic)" module, students will be able to:

LEARNING OUTCOMES

Introduction to Adobe Fireworks:

- Understand the purpose and capabilities of Adobe Fireworks as a tool for creating web graphics and images.
- Differentiate between bitmap and vector graphics and their applications.

Creating Web Graphics:

- Use Adobe Fireworks to create web graphics, banners, buttons, icons, and other visual elements.
- Utilize drawing tools, shapes, and brushes to design custom graphics.

Image Optimization for Web:

- Optimize images for web display by reducing file sizes while maintaining visual quality.
- Export images in appropriate formats (JPEG, PNG, GIF) and adjust compression settings.

Working with Layers and Effects:

- Organize graphics using layers to manage elements separately.
- Apply effects, gradients, and filters to enhance the visual appeal of web graphics.

Slicing and Exporting:

- Learn the concept of slicing graphics for optimized web layouts.
- Export sliced graphics and generate HTML/CSS code for web designs.

Prototyping Web Elements:

- Create interactive prototypes of web elements such as navigation menus, buttons, and forms.
- Simulate user interactions and demonstrate functionality.

Integration with Other Tools:

- Understand how Adobe Fireworks can integrate with other design and development tools in a web workflow.
- Export graphics and assets for use in HTML, CSS, and JavaScript.

Responsive Design Graphics:

- Design graphics and assets that adapt to various screen sizes in responsive web design.
- Create scalable vector graphics (SVG) for modern web interfaces.

Web Graphic Best Practices:

- Apply design principles, color theory, and typography to create visually appealing and effective web graphics.
- Follow industry best practices for creating graphics that enhance user experience.

Hands-on Project: Designing Web Graphics:

- Create a set of web graphics (banners, buttons, icons) using Adobe Fireworks.
- Optimize graphics for web display and export in appropriate formats.
- Develop interactive prototypes to showcase user interactions and functionality.

Week 7-8: PHP (Hypertext Preprocessor)

Upon completing the "**PHP (Hypertext Preprocessor)**" module, students will be able to:

LEARNING OUTCOMES

Introduction to PHP:

- Understand the role of PHP as a server-side scripting language for creating dynamic web content.
- Differentiate between client-side and server-side scripting.

PHP Basics:

- Define variables, data types, and operators in PHP.
- Write PHP code to perform basic arithmetic operations and manipulate data.

Working with Forms and User Input:

- Receive and process user input from HTML forms using PHP.
- Validate and sanitize user input to enhance data security and integrity.

Control Structures and Loops:

- Implement control structures (if statements, switch statements) to make decisions in PHP code execution.
- Use loops (for, while) for repetitive tasks and iterations.

Functions and Modularization:

- Define and call functions to encapsulate reusable blocks of PHP code.
- Understand the importance of modularization and code organization.

Session Management and Cookies:

- Implement sessions and cookies to manage user data between requests.
- Understand how to maintain user state and track user activity.

Database Interaction with MySQL:

- Establish connections to a MySQL database using PHP.
- Execute SQL queries for data retrieval, insertion, updating, and deletion.

Building Dynamic Web Pages:

- Generate dynamic content by embedding PHP code within HTML templates.
- Display database-driven content on web pages, such as user profiles and product listings.

Error Handling and Debugging:

- Implement error handling techniques to catch and manage PHP errors and exceptions.
- Use debugging tools and techniques to identify and fix issues in PHP code.

Security Considerations in PHP:

- Apply security best practices to prevent common vulnerabilities, such as SQL injection and cross-site scripting (XSS).
- Understand the importance of input validation, data sanitization, and secure coding practices.

Hands-on Project: Dynamic Web Application:

- Build a dynamic web application using PHP that interacts with a MySQL database.
- Develop user registration, login, and profile management features.
- Display dynamic content based on user input and database queries.

Week 8-9: MySQL & Database Management

STRUCTURED QUERY LANGUAGE

</SQL>

Upon completing the "MySQL & Database Management" module, students will be able to:

LEARNING OUTCOMES

Introduction to Databases:

- Understand the purpose and importance of databases in web development.
- Differentiate between relational databases and other types of databases.

Database Design and Structure:

- Design a relational database schema, including tables, fields, and relationships.
- Understand primary keys, foreign keys, and indexes for efficient data storage and retrieval.

Creating and Managing Databases:

- Create a MySQL database using SQL commands and a graphical interface (phpMyAdmin).
- Manage databases, tables, and records through a database management system.

SQL Fundamentals:

- Write SQL queries to perform CRUD operations (Create, Read, Update, Delete) on database records.
- Retrieve specific data using SELECT statements with conditions and sorting.

Advanced SQL Queries:

- Perform complex SQL queries involving joins to retrieve data from multiple tables.
- Aggregate data using functions like COUNT, SUM, AVG, and GROUP BY.

Data Validation and Integrity:

- Implement data validation and constraints to ensure data integrity.
- Use triggers and stored procedures to automate database actions.

Normalization and Database Optimization:

- Understand database normalization and its role in reducing data redundancy and improving data integrity.
- Optimize database performance through indexing and query optimization techniques.

Backup and Recovery:

- Create regular database backups and understand the importance of data recovery.
- Restore databases from backups in case of data loss.

Security and Access Control:

- Implement access control and user privileges to restrict data access and manipulation.
- Understand database security practices to prevent unauthorized access.

Database Integration with Web Applications:

- Integrate MySQL databases with web applications using server-side scripting languages (e.g., PHP).
- Perform database CRUD operations (Create, Read, Update, Delete) from web interfaces.

Hands-on Project: Database-Driven Web Application:

- Develop a web application that interacts with a MySQL database.
- Create a user registration and login system with secure password storage.
- Implement data retrieval and manipulation features using SQL queries.

Week 9: API Integration (Online Payment, SMS, Email)

Upon completing the "API Integration (Online Payment, SMS, Email)" module, students will be able to:

LEARNING OUTCOMES

Introduction to APIs:

- Understand the concept of Application Programming Interfaces (APIs) and their role in enabling communication between different software systems.

API Authentication and Authorization:

- Explain different methods of API authentication (e.g., API keys, OAuth) and understand the importance of secure authorization.
- Implement API authentication to ensure secure data exchange.

Online Payment Integration:

- Integrate online payment gateways (e.g., Paystack, Vella, Vpay) into web applications for processing transactions.
- Implement payment forms, handle payment callbacks, and manage payment statuses.

SMS Integration:

- Integrate SMS APIs (e.g., BulkSMSlive) to send and receive SMS messages from web applications.
- Implement SMS notifications, verification codes, and two-factor authentication.

Email Integration:

- Integrate email APIs (e.g., SendGrid, Mailgun) to send transactional and marketing emails.
- Create email templates, send personalized messages, and handle email delivery and tracking.

Handling API Responses:

- Understand API response formats (e.g., JSON, XML) and how to parse and process API responses.
- Handle errors and exceptions in API responses and provide user-friendly feedback.

API Documentation and Testing:

- Read and understand API documentation to identify endpoints, request formats, and response structures.
- Use tools like Postman to test and debug API requests and responses.
-

Data Privacy and Security:

- Implement data privacy measures when transmitting sensitive information via APIs.
- Handle and protect user data in compliance with data protection regulations.

API Integration Best Practices:

- Follow best practices for API integration, including efficient use of API requests, error handling, and rate limiting.
- Optimize API calls for performance and responsiveness.

Hands-on Project: Integrated Web Functionality:

- Integrate online payment, SMS, or email functionality into a web application.
- Implement API requests and handle responses to achieve desired functionality.
- Create user-friendly interfaces for users to interact with integrated features.

Week 10: Graphic Design (CorelDRAW and Photoshop)

Upon completing the "Graphic Design (CorelDRAW and Photoshop)" module, students will be able to:

LEARNING OUTCOMES

Introduction to Graphic Design:

- Understand the principles of graphic design and its importance in creating visually appealing and effective visual communication.

CorelDRAW Basics:

- Navigate the CorelDRAW interface and tools for creating vector-based graphics.
- Create and manipulate shapes, lines, and curves to design logos, illustrations, and layouts.

Adobe Photoshop Basics:

- Familiarize with the Adobe Photoshop interface and tools for editing and enhancing raster images.
- Apply filters, adjustments, and retouching techniques to improve image quality.

Typography and Layout Design:

- Apply typography principles to create well-designed layouts with proper font selection, alignment, and spacing.
- Design posters, flyers, and banners with attention to visual hierarchy and composition.

Image Editing and Manipulation:

- Edit and manipulate images using selection tools, layers, and masks in Photoshop.
- Remove backgrounds, combine images, and apply creative effects to enhance visual impact.

Vector Illustration and Logo Design:

- Create vector-based illustrations and logos using CorelDRAW's drawing tools and techniques.
- Design scalable logos and artwork suitable for various applications.

Color Theory and Application:

- Understand color theory and its application in graphic design.
- Create color palettes and use color effectively to evoke emotions and convey messages.

Print and Digital Design:

- Design materials for both print (brochures, business cards) and digital (web graphics, social media posts) formats.
- Understand the differences between print and digital design requirements.

Mockups and Visual Prototyping:

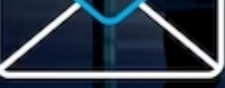
- Create mockups and visual prototypes for web and mobile interfaces.
- Showcase design concepts and interactions using graphical representations.

Graphic Design Best Practices:

- Apply design principles, alignment, contrast, and balance to create visually harmonious designs.
- Understand file formats (JPEG, PNG, PDF) and resolution considerations for different output requirements.

Hands-on Project: Graphic Design Portfolio:

- Create a portfolio of graphic design projects, showcasing various skills learned in CorelDRAW and Photoshop.
- Design logos, promotional materials like flyers, complimentary cards to demonstrate design versatility.



Week 11: Digital Marketing

Upon completing the "**Digital Marketing**" module, students will be able to:

LEARNING OUTCOMES

Introduction to Digital Marketing:

- Understand the concept and importance of digital marketing in promoting products and services online.
- Differentiate between traditional marketing and digital marketing channels.

Content Creation and Strategy:

- Develop compelling and relevant content for various digital platforms, including websites, social media, and blogs.
- Formulate a content strategy that aligns with target audiences and business goals.

Social Media Marketing:

- Create and manage social media profiles and pages on platforms like Facebook, Instagram, Twitter, and LinkedIn.
- Develop content calendars, schedule posts, and engage with the audience to build brand awareness.

Search Engine Optimization (SEO):

- Understand SEO principles and techniques to optimize website content for better search engine visibility.
- Implement on-page and off-page SEO strategies to improve search rankings.

Search Engine Marketing (SEM) and Pay-Per-Click (PPC) Advertising:

- Set up and manage PPC advertising campaigns on platforms like Google Ads.
- Create targeted ads, choose keywords, and monitor campaign performance.

Email Marketing:

- Design and execute email marketing campaigns to reach and engage with subscribers.
- Craft compelling email content, segment audiences, and track email metrics.

Online Advertising and Display Ads:

- Understand different types of online advertising (banner ads, video ads, native ads) and their formats.
- Create eye-catching display ads for websites and social media platforms.

Analytics and Data Interpretation:

- Utilize digital marketing analytics tools (e.g., Google Analytics) to measure and interpret website and campaign performance.
- Make data-driven decisions to optimize marketing strategies.

Conversion Optimization and User Experience (UX):

- Implement strategies to improve website and landing page conversion rates.
- Understand UX principles and apply them to enhance user engagement and satisfaction.

Social Media Advertising:

- Develop and manage social media advertising campaigns on platforms like Facebook, Instagram, and LinkedIn.
- Target specific demographics, set budgets, and analyze ad performance.

Hands-on Project: Digital Marketing Campaign:

- Plan and execute a comprehensive digital marketing campaign for a specific product or service.
- Incorporate multiple digital marketing channels (content creation, social media, SEO, SEM, etc.).
- Monitor and analyze campaign results, making adjustments based on performance metrics.

After completing this comprehensive web and software development training program, students will have gained a diverse set of skills that empower them to work on a variety of projects. Here are some project ideas and examples that students could pursue:

PROJECT EXAMPLES

Personal Portfolio and Business Websites:

Create a portfolio/Business website showcasing their web development skills, projects, and achievements. Implement responsive design, interactive elements, and display their graphic design work.

E-commerce Website:

Develop an online store with product listings, shopping cart functionality, and secure payment gateways. Integrate APIs for online payment processing and order management.

Blog Platform:

Build a blogging platform with user registration, authentication, and the ability to create, edit, and publish blog posts.

Implement features such as categories, tags, and commenting.

Content Management System (CMS):

Create a CMS for managing website content, allowing users to easily add, edit, and delete pages, articles, and multimedia content.

Social Media Platform:

Design and develop a niche social media platform with user profiles, posts, comments, likes, and friend connections.

Implement real-time updates using WebSockets.

Online Learning Platform:

Build a platform for online courses with user registration, course enrollment, video lessons, quizzes, and progress tracking.

Job Board or Freelance Platform:

Develop a job board where employers can post job listings and job seekers can search and apply for positions.

Create a platform for freelancers to showcase their skills and connect with clients.

Real-Time Chat Application:

Design a chat application with real-time messaging, user presence indicators, and support for private and group chats.

Online Booking System:

Build an appointment booking system for services like spa appointments, restaurant reservations, or doctor appointments.



Week 13

Project Defense

Project defense is a crucial component of assessing and examining the skills and knowledge students have acquired during the training program.





Contact Info



support@wetindey.space



+234 807 440 0351



54A, Oba-Adesida Road
Opp. Wema Bank, Akure



wetindey.space