ABOUT THE COURSE:

TOTAL DURATION:	45 HRS
MODE OF DELIVERY	Virtual Instructor-Led Training + Self-Paced Learning
TRAINER TO STUDENT RATIO:	Delivered Virtually
TOTAL MARKS:	75

TABLE 1	
OVERALL COURSE OBJECTIVE:	This is a comprehensive program designed to equip learners with the necessary skills and knowledge to become Front End Development. The program covers key concepts, tools, and technologies in web development. This will be the first Step for the Students who want to Become Front-End Developers. Students will be Going through the Basic Concepts of HTML, CSS and JavaScript. AS well as they also understand the basic Concepts of Cloud Computing. Students will also learn to deploy their Webpages in Git and GitHub.
LEARNING OUTCOME:	1. Understand web development basics and its relevance to personal profile webpages. 2. Learn HTML to create structured webpages with headings, paragraphs, and lists. 3. Apply CSS styles for webpage enhancement 4. Develop responsive web design skills to ensure the webpage looks good on various devices. 5. Learn JavaScript to add interactivity and dynamic content to the webpage. 6. Manipulate the DOM with JavaScript for dynamic updates to webpage content. 7. Explore web hosting options and deploy the webpage online for public access. 8. Customize the webpage with advanced CSS techniques, animations, and transitions. 9. Integrate third-party libraries or frameworks for added functionality and design options. 10. Create a functional personal profile webpage as a final project, demonstrating acquired skills.

TABLE 2: MODULE-WISE COURSE CONTENT AND OUTCOME				
SL.NO	**MODULE NAME**	**MODULE CONTENT**	**MODULE LEARNING OUTCOME**	**DURATION (HRS)**
Week 1	Program Introduction	1. Introduction to Skills Build Program (Orientation and Registration in Portal) 2. Introduction to Coding and Cloud Computing (Explaining the Syllabus and Future Program Instruction)	• Understand the structure and objectives of the Skills Build Program, including successful registration on the portal. • Gain an overview of coding concepts and cloud computing, along with clarity on the syllabus and future program expectations.	2 HRS
Week 2	Introduction to HTML	1. Understanding of WEB. 2. Introduction to HTML- Fundamentals of Required tools and technologies like Visual Studio Code, notepad++, Eclipse for frontend web application development. 3. HTML- Elements, Tags, Components and	• Gain foundational knowledge of web development concepts, including understanding the web and its components. • Develop the ability to utilize essential tools and	2 HRS

		Structures, Formatting	technologies like Visual Studio Code, Notepad++, and Eclipse for creating and managing HTML-based frontend web applications.	
Week 3	HTML Web Applications	1. Web Application: List, Table, Form, Media, graphics, Semantic tags 2.Link HTML 5 APIs: Geolocation, Web Storage	• Develop the ability to create and structure web applications using lists, tables, forms, media, graphics, and semantic tags to ensure accessible and organized content. • Gain practical knowledge of integrating HTML5 APIs, such as Geolocation and Web Storage, to enhance web application functionality and interactivity.	2 HRS
MileStone-1	**Students should share the Screenshot of their HTML page (Self- Paced)**			4 HRS
Week 4	Introduction to CSS	1. Introduction to CSS 2. CSS syntax and embedding,	• Understand the fundamental concepts of	2 HRS

		CSS selector	CSS, including its purpose and how it enhances the presentation of web pages. • Gain the ability to write and apply CSS rules using proper syntax, embedding methods, and various selectors to style HTML elements effectively.	
Week 5	CSS Properties	1. CSS properties: Colour, Background, Text, Font, Position, List style, table 2. CSS Properties: pseudo-element, Transformations, Animation, and Media Queries, grid, flex	• Demonstrate the ability to effectively use core CSS properties such as color, background, text, font, position, list style, and table to design visually appealing and well-structured web pages. • Apply advanced CSS concepts, including pseudo-elements, transformatio ns,	2 HRS

			animations, media queries, grid, and flexbox, to create responsive and dynamic web designs suitable for various devices and user experiences.	
MileStone-2	**Students should share the Screenshot of their HTML & CSS page (Self- Paced)**			4 HRS
Week 6	Introduction of JS	1.Java Script: Types of JS, JS console, Dialog box, Operators and Functions 2. Java Script: Control Structures, Document Object Model (DOM)	• Gain an understanding of JavaScript fundamentals, including its types, console operations, dialog boxes, operators, and functions, to effectively build dynamic web applications. • Develop skills to implement JavaScript control structures and manipulate the Document Object Model (DOM) for interactive and responsive user interfaces.	2 HRS

Week 7	Applications of JS	1. Java Script: Objects and Nodes, Handling DOM using JavaScript 2.Java Script: JavaScript Events, Animation, Cookies & session	• Develop the ability to manipulate the Document Object Model (DOM) using JavaScript, enabling dynamic interactions with web page elements through objects and nodes. • Gain proficiency in managing JavaScript events, creating animations, and handling cookies and sessions to enhance user experience and web application functionality.	2 HRS
Week 8	Introduction of Cloud	1. Introduction of Cloud Computing-(Git & GitHub) 2. Deploying the Web page in GIT & GitHub	• Understand the fundamentals of cloud computing, including version control using Git and GitHub. • Gain hands-on experience in deploying a web page	2 HRS

			using Git and GitHub platforms.	
MileStone-3	**Students should share the Screenshot of their Deployment Model (Self- Paced)**			4 HRS

TABLE 3: OVERALL COURSE LEARNING OUTCOME ASSESSMENT CRITERIA AND USECASES			
LEARNING OUTCOME	**ASSESSMENT CRITERIA**	**PERFORMANCE CRITERIA**	**USE CASE**
Understand web development basics and its relevance to personal profile webpages.	Explain the structure of the web and its role in personal webpage creation.	Demonstrate understanding of web components, including structure, tools, and technologies.	Use Case: **Building a Personal Blog** – A user creates a basic blog explaining their background and interests.
Learn HTML to create structured webpages with headings, paragraphs, and lists.	Create HTML documents with headings, paragraphs, lists, and other structural elements.	Develop error-free and semantically correct HTML pages, ensuring structured and accessible content.	Use Case: **Creating a Resume Page** – Build a webpage to display a well-structured resume using headings and lists.
Apply CSS styles for webpage enhancement.	Implement CSS to style webpages, including text, layout, and design properties.	Demonstrate the ability to enhance webpage presentation with effective CSS rules and advanced selectors.	Use Case: **Styling a Portfolio Page** – Customize fonts, colors, and layout to make a portfolio visually appealing.
Develop responsive web design skills to ensure the webpage looks good on various devices.	Use media queries and responsive layouts (grid and flexbox) for different screen sizes.	Build dynamic and visually responsive web designs compatible across devices and resolutions.	Use Case: **Building a Mobile-Friendly Page** – Ensure a profile page works seamlessly on phones, tablets, and PCs.
Learn JavaScript	Write JavaScript	Apply JavaScript	Use

to add interactivity and dynamic content to the webpage.	code for basic interactivity, including functions, control structures, and events.	to add interactive features, ensuring dynamic content and user engagement.	Case: **Interactive Contact Form** – Add validation and interactivity to a contact form for user engagement.
Manipulate the DOM with JavaScript for dynamic updates to webpage content.	Use JavaScript to manipulate the DOM elements dynamically.	Demonstrate DOM manipulation to update webpage content based on user input or interaction.	Use Case: **Dynamic Content Update**– Update a user's project gallery dynamically based on clicks or filters.
Explore web hosting options and deploy the webpage online for public access.	Successfully deploy a personal webpage on platforms like GitHub using Git version control.	Host and share a fully functional webpage, verifying accessibility and correctness in deployment.	Use Case: **Deploying a Personal Site** – Upload a personal website to GitHub for professional visibility.
Customize the webpage with advanced CSS techniques, animations, and transitions.	Create CSS animations, transitions, and advanced styling techniques for visual appeal.	Demonstrate proficiency in advanced CSS, creating engaging animations and ensuring smooth transitions.	Use Case: **Animating Portfolio Sections** – Use transitions to make portfolio sections smoothly appear on scroll.
Integrate third-party libraries or frameworks for added functionality and design options.	Use external libraries (e.g., Bootstrap) to enhance the webpage design and add pre-built features.	Incorporate third-party libraries seamlessly into the project to improve functionality and design.	Use Case: **Using Bootstrap for Quick Design** – Integrate Bootstrap to create a professional layout with less effort.
Create a functional personal profile webpage as a final project, demonstrating acquired skills.	Develop and showcase a complete personal profile webpage incorporating HTML, CSS, JavaScript, and hosting.	Demonstrate integration of learned skills into a cohesive final project, meeting functional, responsive, and design standards.	Use Case: **Showcasing a Personal Profile** – Develop a comprehensive webpage summarizing skills, projects, and bio.

TABLE 4: LIST OF FINAL PROJECTS (PROJECTS THAT

COMPREHENSIVELY COVER ALL THE LEARNING OUTCOME)	
SL.NO	**FINAL PROJECT**
1	Completion of the Learning Plan
2	Submission of Student Digital Portfolio using GitHub

TABLE 5: COURSE ASSESSMENT RUBRICS (TOTAL MARKS: 75)				
ASSESSMENT CRITERIA	**DESCRIBE THE CRITERIA OF THE BELOW CATEGORY PERFORMANCE**			**TOTAL MARKS**
	FAIR	**GOOD**	**EXCELLENT**	
1	33	50	75	75