Coincent 3-Year Program in Web Development (Full Stack)
Partnered by Movidu

Year 1: Live Industrial Training – Build Your Foundation

Gain hands-on industry exposure from day one with 2.5 months of live training in a professional environment. Learn the latest tools and technologies through skill-focused sessions, guided by expert mentors from the industry

Web Development Curriculum
Front-End Development Curriculum

Chapter 1: HTML Fundamentals

1. Introduction to HTML

- **Purpose**: HTML (HyperText Markup Language) is the backbone of all web pages. It structures content for browsers.
- **Key Concepts**: Markup language, HTML versions (HTML5 is the latest), static vs dynamic pages.

2. Structure of HTML

- **Tags**:
  - <!DOCTYPE html>: Declares the HTML version.
  - <html>: Root element of the document.
  - <head>: Metadata, scripts, styles.
  - <body>: Visible content of the web page.

3. Basic HTML Elements

- Usage of tags such as <h1>–<h6> for headings, <p> for paragraphs, <br> for line breaks, <strong> for bold, and <em> for italics.

Coincent
Learning Drives Excellence

MOVIDU

4. HTML Link Tags

- <a href="url">: Used for hyperlinks.
- target="_blank" opens links in a new tab.

5. Ordered and Unordered Lists

- <ul>: Unordered (bulleted)
- <ol>: Ordered (numbered)
- <li>: List item within lists

6. Head and Meta Tags

- Metadata tags like <meta charset="UTF-8">, <meta name="viewport" content="width=device-width, initial-scale=1.0">
- <title> sets the browser tab name.

7. Styles and Semantic Tags

- **Inline Styles**: style="color:blue" (used sparingly)
- **Semantic Tags**: <header>, <footer>, <article>, <section>, <aside>, <nav> improve readability and SEO.

8. HTML Element Attributes

- id, class, style, title, alt (for images), href, src
- Provide additional info to elements for styling and scripting.

9. HTML Form Elements

- Forms collect user input using <input>, <textarea>, <select>, <button>.
- Validation with required, type="email" etc.

10. HTML Table Elements

- Table structure using <table>, <tr>, <td>, <th>.
- Attributes: colspan, rowspan, border, cellpadding.

Chapter 2: CSS Fundamentals

1. Fundamentals of CSS

- Cascading Style Sheets control layout and design.
- Types: Inline, Internal (<style>), External (.css file).

2. CSS Selectors

- Select elements by tag (p), class (.class), id (#id), attributes ([type="text"]), pseudo-classes (:hover), pseudo-elements (::after).

3. Color and Backgrounds

- color, background-color, gradients using linear-gradient(), image backgrounds using background-image.

4. Fonts

- font-family, font-size, import Google Fonts, fallback fonts.

5. Text Properties

- text-align, text-decoration, letter-spacing, line-height, text-transform.

6. CSS Float

- Used to float elements left/right; requires clearing with clear: both;.

7. CSS Box Model

- Each element has: content + padding + border + margin.
- Visual spacing is managed here.

8. CSS Display

- block, inline, inline-block, none (for hiding), flex, grid.

9. Flexbox

- One-dimensional layout system for aligning items horizontally or vertically.
- justify-content, align-items, flex-direction.

10. Responsive Web Design

- Adapting layout for devices using @media queries.
- Units: %, vh, vw, em, rem for scalability.

Chapter 3: JavaScript Fundamentals

1. Introduction

- JS enables interactivity.
- Runs on client-side. Supported by all browsers.

2. Variables and Outputs

- var, let, const: ES6 introduced let and const.
- Outputs: alert(), console.log(), document.write().

3. Data Types

- Strings, numbers, booleans, null, undefined, and symbols.

4. Conditional Statements

- if, else if, else, switch to make decisions.

5. Loops

- for, while, do...while loops for repeating code.

6. Functions

- Reusable blocks of code.
- Parameters, return values, function myFunc(){}

7. Scope

- Local vs Global variables.
- Hoisting: Function and variable declarations are moved to the top.

8. Arrow Functions

- const add = (a, b) => a + b;
- Shorter syntax, lexical this.

9. Arrays

- Store ordered data.
- Methods: push(), pop(), shift(), map(), filter().

10. Objects

- Key-value pairs. { name: "John", age: 30 }
- Access via dot or bracket notation.

11. Event Handling

- onclick, onmouseover, onchange, addEventListener.

12. DOM & BOM

- DOM: Access and manipulate HTML structure.
- BOM: Browser-specific APIs (window, navigator).

13. JSON

- Data exchange format.
- JSON.stringify(), JSON.parse() for working with APIs.

Chapter 4: jQuery Fundamentals

1. Role of jQuery

- Simplifies DOM manipulation and event handling.

2. Getting Started

- Use via CDN or download. Requires jQuery library.

3. Basic Syntax

- $(selector).action() e.g., $("#btn").click().

4. Selectors

- Same as CSS: #id, .class, element.

5. Events

- Attach behaviors: click(), hover(), submit().

6. Show/Hide/Toggle

- Easy UI visibility changes: show(), hide(), toggle().

7. Fading Effects

- Animate opacity using fadeIn(), fadeOut().

8. Sliding Effects

- Vertical transitions with slideUp(), slideDown().

9. Animations

- Animate any CSS property. Use stop() to interrupt.

10. Chaining

- Combine multiple actions: $(selector).css().slideUp().fadeOut();

Chapter 5: Bootstrap Fundamentals

1. Bootstrap Intro

- CSS framework for responsive UI development.
- Includes grid system, prebuilt components.

2. Getting Started

- Include via CDN or install locally.

3. Containers & Typography

- .container, .container-fluid for layout.
- Text utilities for alignment, size, and colors.

4. Grid System

- 12-column layout with breakpoints (col-sm, col-md, etc.).

5. Tables & Forms

- Style tables with .table, form controls with .form-control.

6. Buttons & Images

- Predefined styles using .btn, responsive images via .img-fluid.

7. Cards

- Flexible content containers for text, images, links.

8. Components

- Navs, navbars, accordions, breadcrumbs, paginations, modals, spinners, tooltips, popovers, alerts, carousels.

Back-End Development Curriculum
Chapter 1: PHP Fundamentals

1. PHP Basics

- Open-source scripting language executed on server.

2. Syntax

<?php ?> tags, semicolons, comments (//, /* */).

3. Constants and Variables

- define("SITE", "MySite");
- Dynamic typing: $x = "hello";

4. Echo/Print

- Output data to browser.

5. Data Types

- Strings, numbers, arrays, objects, null.

6. String Functions

- strlen(), strpos(), str_replace() etc.

7. Operators and Conditions

- Arithmetic, comparison, logical.
- if, switch, while, for.

8. Arrays & Objects

- Loop through arrays, create class instances.

9. Functions

- User-defined functions with parameters and return values.

10. Forms & Validation

- Validate user input from $_POST or $_GET.

11. Sessions & Cookies

- Manage user state across pages.

12. File Handling & Emails

- Read/write files. Send email via mail() or SMTP.

13. JSON, Filters

- Data formatting and input sanitation using filter_var().

14. MySQL Integration

- CRUD operations using mysqli or PDO.

Chapter 2: SQL Fundamentals

- SQL operations to manage relational databases.
- From creating tables to writing complex queries:
    - SELECT, WHERE, JOIN, GROUP BY, ORDER BY
    - DDL: CREATE, DROP, ALTER
    - DML: INSERT, UPDATE, DELETE

Chapter 3: Laravel Fundamentals

1. Laravel Basics

- MVC framework for PHP. Uses Eloquent ORM.
- Install via Composer.

2. Artisan CLI

- Commands for generating code: php artisan make:controller, migrate, serve.

3. Routing, Middleware

- Define URL routes and control access via middleware.

4. Controllers, Views (Blade)

- Business logic in controllers, UI in Blade templates.

5. Models, Migration

- Define schema with migrations and interact using models.

6. API and Web Routes

- Create REST APIs with routes/api.php.

7. Validation, Sessions

- Validate requests using FormRequest or controller methods.
- Handle user data across sessions.

8. Mail, Queue, Helpers

- Use Mailtrap for testing emails.
- Queue background jobs.
- Helpers like url(), asset().

Year 2: Real-Time Projects – Apply What You've Learned

Transform your knowledge into real-world experience by working on 8 industry-level projects that build your technical and professional skills. Each project enhances your portfolio, strengthening your resume and showcasing your practical abilities. You'll also collaborate in teams, gaining valuable experience in communication, teamwork, and project management—just like in a real work environment.

PROJECTS

1. Advance E-Commerce Website

An advanced e-commerce website offers a seamless online shopping experience with features like user authentication, dynamic product listings, cart management, and secure payment gateway integration. It supports multiple product categories, real-time search and filtering, product reviews, and order tracking. Admins can manage inventory, process orders, and offer promotions via an intuitive backend. The frontend is designed to be responsive and visually engaging using HTML, CSS, Bootstrap, and JavaScript frameworks. Backend logic is handled using PHP or Laravel with SQL database integration. It also includes features like wishlist, coupon codes, and email notifications. SEO optimization and performance tuning are part of the deployment process. This project gives full-stack exposure to modern e-commerce systems.

2. Job Portal Dashboard

This dashboard allows users to search and apply for jobs while companies can post and manage job listings. It includes role-based access for job seekers, recruiters, and admins. Features include user registration/login, profile management, resume upload, and job alerts. Recruiters can filter applications, schedule interviews, and manage shortlists. Admins have full control over users, job posts, and analytics. The frontend is built with responsive UI libraries like Bootstrap and React or jQuery, while the backend uses PHP/Laravel or Node.js with a MySQL or MongoDB database. The system also supports application tracking, status updates, and real-time notifications. It's an excellent project for learning CRUD operations, authentication, and dashboard design.

3. Doctor Appointment Dashboard

This healthcare-oriented dashboard enables patients to book appointments, consult doctors, and maintain medical history online. It provides separate portals for doctors and patients with features like appointment scheduling, real-time calendar integration, and prescription uploads. Doctors can manage their availability, view patient records, and send consultation notes. Patients can search for doctors by specialty or location and receive automated SMS/email confirmations. Admins can manage doctor profiles, bookings, and system reports. Built using frontend tools like Bootstrap and JavaScript, with a backend powered by PHP or Node.js and a relational database. The system ensures data security and smooth UX, ideal for digitizing clinical workflows.

4. Chat Application

This real-time chat application supports one-to-one and group messaging with features like typing indicators, message timestamps, and online status tracking. It allows user sign-up/login with JWT authentication and secure password hashing. Messages are sent and received using WebSockets (e.g., Socket.io) for real-time performance. The UI is built with HTML, CSS, and React or plain JavaScript, while the backend uses Node.js and Express, with MongoDB or Firebase for storage. Optional features include emojis, file sharing, chat history, and notifications. It's a strong full-stack project demonstrating WebSocket communication, state management, and responsive design. The system can be deployed on platforms like Heroku or Vercel.

Year 3 – Placement & Internship Phase:

In the 3rd year of Coincent's program, students are guaranteed an internship with partner companies, complete with a formal Internship Offer Letter and a Completion Certificate upon successful completion. This internship is a complimentary part of the 3-Year "Industrial Training + Internship" model, which also includes live classes, expert mentorship, and hands-on project work. This phase bridges academic learning with real-world application, providing students with valuable professional exposure before graduation.

Coincent also offers structured placement preparation to ensure students are job-ready. This includes portfolio building through 8 real-time projects, certifications aligned with Microsoft standards, and dedicated training for interviews. From mock interviews to resume reviews and HR/technical round prep, every element is designed to transition students from classroom learning to career success. By the 4th year, students are equipped not just with knowledge, but with experience, credentials, and confidence to enter the workforce.