

FULL STACK WEB DEVELOPMENT CURRICULUM

**Two-Semester Program | Beginner Level | 1st & 2nd Year Undergraduate Students |
AI-Assisted Learning**

1. Program Overview

Program Duration	Two semesters (one full academic year), phased delivery
Theory Contact Hours	20 hours per semester (40 hours total): 17 hours of core/basics instruction + 3 hours of dedicated project studio time.
Projects	6 total (3 per semester), progressive in complexity.
Sequencing Logic	Semester 1: Builds foundational web literacy (markup, styling, scripting, relational data, server-side basics) using one language family (JS + PHP). Semester 2: Builds on the JavaScript investment first (Node/Express/React = MERN) before introducing Laravel as a contrasting MVC/ORM paradigm.

2. AI-Assisted Learning: Tools Used Across the Curriculum

Every basics module pairs its topic with a free AI coding assistant so students learn to use these tools correctly from day one, rather than picking up ad-hoc habits later.

Tool	Free Tier (Aug 2026)	Used For
GitHub Copilot (Student Plan)	Free for verified students via GitHub Education (GitHub Student Developer Pack):	Primary tool across all modules, inline completion + chat, works in VS Code

Tool	Free Tier (Aug 2026)	Used For
	unlimited completions + premium chat/agent credits. Non-verified users get a free tier of 2,000 completions + 50 chats/month.	
Codeium	Free forever for individuals, unlimited completions, no student verification needed	Backup/default tool for students without student-plan verification
Google Gemini Code Assist	Generous free tier in VS Code and JetBrains	SQL query generation/explanation, alternative chat assistant
Amazon Q Developer	Free for individual/personal use, unlimited completions	Backend/API modules, security-aware suggestions
Cursor (Hobby)	Free tier with a limited monthly pool of AI requests; full AI-native editor	Optional for React module, strong multi-file/component context handling

AI USAGE DISCLAIMER

At the basics stage, AI-assisted coding is a double-edged tool. Used well, it accelerates syntax fluency and shows idiomatic patterns; used carelessly, it lets a beginner submit working code they cannot explain, which defeats the purpose of a foundations course.

3. Semester 1 – Web Foundations (HTML, CSS, JavaScript, jQuery, SQL, PHP)

3.1 MODULE-WISE BREAKDOWN

Module	Topics Covered	AI Tool	Hrs
Orientation & Web Fundamentals	- Client-server model and request/response cycle - DNS, hosting, and how a URL resolves to a page - HTTP methods, status codes, headers - Browser rendering pipeline (HTML parse → CSSOM → render tree) - Dev environment setup: VS Code, browser DevTools, Git/GitHub basics - GitHub Education verification for student AI tool access	GitHub Copilot (setup walkthrough)	1.0
HTML5	- Document structure, doctype, head/meta tags, basic SEO tags - Semantic elements (header, nav, main, article, section, footer) - Text, list, and table elements - Forms: input types, labels, validation attributes - Media embedding: images, audio, video - Accessibility basics: alt text, ARIA roles, tab order	GitHub Copilot / Codeium	1.5
CSS3 & Responsive Design	- Box model, units (px, %, rem, vh/vw)• Selectors, specificity, and the cascade - Flexbox: axes, alignment, wrapping	GitHub Copilot / Codeium	2.0

Module	Topics Covered	AI Tool	Hrs
	- CSS Grid: tracks, areas, responsive grids - Media queries and mobile-first design - Pseudo-classes/elements, transitions, basic animations - CSS variables and reusable design tokens		
JavaScript Fundamentals	- Variables (let/const), data types, type coercion - Operators, control flow, loops• Functions, scope, closures (introductory level) - Arrays and objects: common methods (map, filter, forEach) - DOM selection and manipulation, event listeners - ES6 basics: arrow functions, template literals, destructuring, spread/rest - JSON basics and the Fetch API (introductory) - Basic error handling with try/catch	GitHub Copilot (chat + completion)	5.0
jQuery	- Selector syntax and method chaining - DOM manipulation shorthand (.html(), .css(), .addClass(), etc.) - Event handling with jQuery - Simple AJAX calls with \$.ajax() / \$.get() - Common jQuery plugins and where the library is still used in production	Codeium	1.0
SQL & Relational Databases (MySQL)	- Relational modeling and ER diagrams - Normalization basics (1NF-3NF) - DDL: CREATE, ALTER, DROP; constraints and keys	Gemini Code Assist (query generation/explanation)	3.5

Module	Topics Covered	AI Tool	Hrs
	- DML: INSERT, UPDATE, DELETE, SELECT - Filtering, sorting, aggregate functions, GROUP BY/HAVING - JOINS (INNER, LEFT) and basic subqueries - Indexes: what they are and why they matter		
PHP Fundamentals	- Syntax, variables, control structures, functions - Superglobals (\$_GET, \$_POST, \$_SESSION) - Form handling and server-side validation - Sessions and basic authentication flow - Database integration with MySQLi/PDO (prepared statements) - Basic password hashing (password_hash/password_verify) - Introductory OOP in PHP: classes, objects	GitHub Copilot	3.0
OVERALL HOURS			17.0

3.2 Semester 1 Projects

Project S1-1: Personal Portfolio Website

Tech Stack	HTML5, CSS3, vanilla JavaScript, GitHub Copilot / Codeium
Objective	Build a multi-section responsive personal portfolio (about, skills, project showcase, contact section) with a mobile-first layout and client-side form validation.
Deliverable	Deployed static site (3-5 sections), responsive across breakpoints, validated contact form.

Project S1-2: Interactive Task / Expense Tracker

Tech Stack	JavaScript, jQuery, Browser LocalStorage, Codeium
Objective	Build a client-side CRUD tool (add/edit/delete/mark-complete) with data persisted in LocalStorage and dynamic DOM updates driven by event handling.
Deliverable	Working interactive tracker with persistent state across page reloads.

Project S1-3 (Capstone): Student Record Management System

Tech Stack	PHP, MySQL, HTML5, CSS3, GitHub Copilot, Gemini Code Assist (SQL)
Objective	Build a full server-rendered CRUD application: session-based login, relational schema (students, courses, enrollments), server-side validation, and MySQL persistence.
Deliverable	Functional multi-page PHP application with authentication and a normalized 3+ table schema.

4. Semester 2 – Modern Frameworks (MongoDB, MERN, Laravel)

4.1 MODULE-WISE BREAKDOWN

Module	Topics Covered	AI Tool	Hrs
NoSQL Concepts & MongoDB	- Document model vs. relational tables - Collections, documents, embedded vs. referenced data - Schema design trade-offs and when to choose NoSQL over SQL - CRUD operations with the MongoDB shell/driver - Basic querying and filtering	Amazon Q Developer	2.0
Node.js Fundamentals	- Node runtime model and the V8 engine (conceptual) - CommonJS/ES modules, npm and package.json - Callbacks, Promises, async/await - The event loop and non-blocking I/O (why Node handles concurrency this way) - File system module and environment variables (.env)	GitHub Copilot	2.5
Express.js & REST API Design	- Routing and route parameters - Middleware and middleware chaining - Request/response lifecycle - Building RESTful CRUD endpoints - Centralized error-handling middleware - Basic CORS configuration	GitHub Copilot / Amazon Q Developer	2.5

Module	Topics Covered	AI Tool	Hrs
React.js Fundamentals	- Components (function components), JSX syntax - Props and state, one-way data flow - Hooks: useState, useEffect - Conditional rendering and rendering lists (keys) - Controlled forms - Consuming REST APIs with fetch/axios - Basic client-side routing (React Router)	Cursor (Hobby) / GitHub Copilot	4.5
Laravel Fundamentals	- MVC architecture and request lifecycle - Routing and route parameters - Controllers and resource controllers - Blade templating: directives, layouts, includes - Artisan CLI: generators, migrations, common commands - Request validation basics	GitHub Copilot	2.5
Eloquent ORM & Laravel Queues/Jobs	- Migrations and schema definition - Eloquent models, relationships (hasMany, belongsTo, many-to-many) - Eager loading and avoiding N+1 queries (conceptual) - Creating and dispatching queued jobs - Queue drivers, workers, and basic failed-job handling	GitHub Copilot	2.0
Stack Comparison &	- Node's event-driven model vs. PHP-FPM's process-per-request model - SQL vs. NoSQL decision criteria in practice	Discussion-led	1.0

Module	Topics Covered	AI Tool	Hrs
Architecture Discussion	- When to reach for MERN vs. Laravel on a real project		
OVERALL HOURS			17.0

4.2 Semester 2 Projects

Project S2-1: Task Manager REST API

Tech Stack	Node.js, Express.js, MongoDB, GitHub Copilot, Amazon Q Developer
Objective	Build a RESTful API with full CRUD endpoints for a task-management domain, using MongoDB for persistence and middleware for validation/error handling.
Deliverable	Running API (Postman/curl-testable) with documented endpoints and a MongoDB schema.

Project S2-2: Task Manager Frontend – MERN Integration

Tech Stack	React.js consuming the Project S2-1 API, Cursor (Hobby) / GitHub Copilot
Objective	Build a single-page frontend that consumes the previously built API: component-based UI, hooks-based state management, and end-to-end data flow.
Deliverable	Working MERN application (Mongo + Express + React on Node) demonstrating full-stack integration.

Project S2-3 (Capstone): Library / Inventory Management System

Tech Stack	Laravel, Eloquent ORM, MySQL, Laravel Queues/Jobs, GitHub Copilot
Objective	Build an MVC application with relational models (books, members, transactions), and implement a queued background job (e.g., overdue-item reminder) to contrast asynchronous handling against the Node.js event loop covered earlier.
Deliverable	Functional Laravel application with Eloquent relationships and at least one working queued job.