

Full Stack Development Syllabus

With PHP

Introduction to Full Stack Development

- ✓ Understanding Full Stack Development
- ✓ Roles and responsibilities of a Full Stack Developer
- ✓ Overview of Web Development: Front-end, Back-end, and Database

HTML, CSS, and JavaScript (Front-end Basics)

- ✓ **HTML:** Structure of web pages, forms, tables, semantic HTML, and HTML5 elements.
- ✓ **CSS:** Styling web pages, CSS selectors, box model, flexbox, grid layout, CSS animations, and responsive design.
- ✓ **JavaScript:** Introduction to JavaScript, variables, functions, loops, conditions, DOM manipulation, events, and ES6 features (arrow functions, promises, async/await).

Version Control with Git

- ✓ Introduction to Git and GitHub
- ✓ Basic Git commands: init, clone, commit, push, pull
- ✓ Working with branches, merge conflicts, and collaborative workflows.

Advanced Front-end Development

- ✓ **Responsive Web Design:** Media queries, mobile-first design
- ✓ **JavaScript Frameworks:** Introduction to front-end frameworks like React.js or Vue.js (optional but beneficial for modern development)
- ✓ **AJAX:** Introduction to Asynchronous JavaScript and XML for making dynamic web pages.
- ✓ **Web APIs:** Fetching data from external APIs (JSON, XML).

Introduction to PHP

- ✓ **PHP Basics:** Syntax, data types, variables, operators, control structures (if, switch, loops).
- ✓ **Functions:** Creating and using functions, variable scope, and passing arguments.
- ✓ **Forms and User Input:** GET and POST methods, sanitizing and validating user input.
- ✓ **Sessions and Cookies:** Managing sessions and cookies in PHP.

PHP with MySQL (Back-end Development)

- ✓ **Database Basics:** Introduction to SQL, MySQL database creation, tables, and relationships.
- ✓ **CRUD Operations:** Performing Create, Read, Update, and Delete operations using PHP and MySQL.
- ✓ **Prepared Statements and Security:** Protecting against SQL injection using prepared statements and parameterized queries.
- ✓ **Database Connection:** Connecting PHP to MySQL using PDO (PHP Data Objects) or MySQLi.
- ✓ **Error Handling:** Handling exceptions and debugging PHP code.

Building Dynamic Websites with PHP

- ✓ **Form Handling:** Building and processing forms in PHP.
- ✓ **File Uploading:** Handling file uploads with PHP.
- ✓ **Email Handling:** Sending emails from PHP using mail() function or PHPMailer.
- ✓ **Authentication and Authorization:** Building a login and registration system with PHP, user authentication using sessions and hashing passwords with password_hash().

Object-Oriented Programming (OOP) in PHP

- ✓ **Classes and Objects:** Introduction to classes, objects, constructors, and destructors.
- ✓ **Inheritance, Polymorphism, and Encapsulation:** Concepts of OOP and how they apply to PHP.
- ✓ **Namespaces and Autoloading:** Organizing code using namespaces and autoloading classes.
- ✓ **Design Patterns:** Introduction to commonly used design patterns like MVC (Model-View-Controller).

PHP Frameworks (Optional but Recommended)

- ✓ **Laravel:** Introduction to Laravel (most popular PHP framework), routing, views, controllers, and middleware.
- ✓ **CodeIgniter:** Basics of another popular PHP framework, Model-View-Controller (MVC) architecture.

Web Security

- ✓ **Cross-Site Scripting (XSS):** Preventing XSS attacks.
- ✓ **Cross-Site Request Forgery (CSRF):** Securing forms and preventing CSRF attacks.
- ✓ **SQL Injection:** Understanding and mitigating SQL injection attacks.
- ✓ **Session Management:** Securing user sessions and cookies.

API Development with PHP

- ✓ **RESTful APIs:** Designing and developing RESTful APIs with PHP.
- ✓ **API Authentication:** Implementing authentication for APIs (JWT, OAuth).
- ✓ **Consuming External APIs:** Making HTTP requests using PHP (cURL or Guzzle).

Front-end Integration with PHP

- ✓ **Template Engines:** Using template engines like Twig or Blade (if using Laravel) to separate logic from presentation.
- ✓ **AJAX with PHP:** Creating dynamic web applications that interact with the server without refreshing the page.
- ✓ **Real-time Applications:** Introduction to real-time communication with WebSockets (using tools like Ratchet).

Testing and Debugging

- ✓ **PHP Debugging:** Using Xdebug and error logs.
- ✓ **Unit Testing:** Introduction to PHPUnit and writing test cases for PHP code.
- ✓ **Functional Testing:** Testing user flows and forms.

Deployment and Hosting

- ✓ **Server Setup:** Introduction to server environments like Apache/Nginx.
- ✓ **Web Hosting:** Deploying PHP applications to shared hosting, VPS, or cloud platforms (AWS, Heroku, DigitalOcean).
- ✓ **CI/CD:** Setting up continuous integration and deployment pipelines.
- ✓ **Domain and SSL Setup:** Configuring a domain and SSL for secure communication.