

Backend Development Course Syllabus

Total Duration: 90hrs

Module 1: Fundamentals of Backend

Chapter 1: Backend Fundamentals & Environment Setup

- What is Backend? Server, Client & Database
- MVC / MVT Architecture Overview
- Folder Structure in Different Stacks
- Routing, Controllers, and Middleware
- Templating Engines (Blade, Django Templates, EJS)
- Setting Up Local Development Environment
- Git and GitHub Basics
- Environment Variables using .env Files

Chapter 2: Programming in Your Selected Language

Option A: Python (Django / Flask)

- Django Project Structure and Flask Blueprints
- Django ORM and SQLAlchemy
- Model Relationships: One-to-Many and Many-to-Many
- Basic CRUD using ORM
- Django Admin Panel and Flask Extensions
- REST API using Django REST Framework or Flask-Restful
- JWT Authentication with SimpleJWT or Flask-JWT

Option B: PHP (Laravel)

- Laravel Folder Structure and MVC
- Eloquent ORM and Database Relationships
- Laravel Breeze Authentication Setup
- API Routes, Controllers, and Resource Management
- API Authentication using Laravel Sanctum or Passport
- Middleware Creation, Request Validation, and File Upload Handling

Option C: JavaScript (Node.js + Express)

- Setting up Express.js in MVC Format
- Using Sequelize ORM with SQL Database
- Building RESTful API Endpoints
- JWT Authentication with Role-Based Access
- File Uploads with Multer
- Custom Middleware and Error Handling

Chapter 3: SQL and ORM Integration

- Understanding Relational Databases
- Writing SQL Queries: SELECT, JOIN, WHERE, GROUP BY
- Creating Database Schemas, Migrations, and Relationships
- Performing CRUD Operations using ORM
- Seeding Data and Ensuring Data Integrity
- Debugging SQL with Logs and Query Builders

Chapter 4: REST API Development and Authentication

- Designing RESTful APIs for Applications



- Creating CRUD Routes with Server-Side Validation
- Implementing JWT Token-Based Authentication
- Protecting Routes with Authentication Middleware
- Validating and Sanitizing Input Data
- Testing APIs with Postman or Thunder Client
- Sending Proper HTTP Status Codes and Error Responses

