

# Backend Development Course Syllabus

**Total Duration: 90hrs**

## Module 1: Core Backend Foundations

### 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

## **Module 2: Real-World Backend Features & Deployment**

### **Chapter 5: File Uploads, Logging, and Deployment**

- Handling File Uploads for Images and Documents
- Validating Uploaded Files for Type, Size, and Path
- Serving Static Files from the Backend
- Implementing Logging and Error Tracking Systems
- Managing Environment Configurations in Production
- Deploying Django Projects with Gunicorn, Nginx, and PostgreSQL
- Deploying Laravel Projects using Laravel Forge or Shared Hosting
- Deploying Node.js Projects with Railway, PM2, or Render

### **Chapter 6: Final Capstone Project**

#### **Build One of the Following:**

- Mini Content Management System (CMS)
- Inventory Management Application
- Admin Dashboard with Role-Based Access

#### **Project Requirements:**

- Authentication System with Login, Register, and JWT



- SQL Database Integration with Proper Relationships
- RESTful API with CRUD, Filtering, and Pagination
- File Upload Handling for Images or Documents
- Admin and User Role-Based Access Management
- Hosted Live Project with GitHub Repository
- API Testing Collection using Postman or Thunder Client

**Final Deliverables:**

- Deployed Project Live on Hosting Platform
- GitHub Repository with Complete Source Code
- API Documentation Using Swagger or Postman
- Final Project Presentation and Walkthrough

