

Full-Stack Web Development Foundations with Django and Vue

HTML, CSS, Python, APIs, front-end integration and guided deployment

FA-0766 · 10 days

Software Development · DevOps, Cloud & Infrastructure

Introduction

Follow a beginner progression through web markup, styling, Python and a small Django/Vue application. Connect front-end interaction to backend data and API concepts, then review a guided deployment.

The ten-day course uses focused exercises and a prepared project to introduce a broad stack. It does not promise independent professional mastery, network-administrator qualification or complete production readiness.

Learning outcomes

- Create basic HTML pages and style responsive layouts with CSS.
- Use introductory JavaScript and prepare simple web images.
- Write small Python programs and organise functions, files and objects.
- Build basic Django views, models, forms and API endpoints.
- Connect selected Vue screens to a prepared API and manage shared state.
- Explain authentication, permissions, uploads and data exports in the sample project.
- Follow a prepared deployment workflow and recognise domain, TLS and operational requirements.

Prerequisites

Basic computer, internet and file-management skills. No previous web-development experience is required.

A computer with a current browser, editor and compatible supported Python/Django and Vue tooling. Prepared server access and approved installation permissions are used; requirements depend on the chosen lab, not an arbitrary universal 4GB/1Mbps minimum.

Training outline

1. Module 1 – HTML foundations

- Introduction
- Editors
- Basic
- Elements
- Attributes
- Headings
- Paragraphs
- Styles
- Formatting
- Quotations
- Comments
- Colors
- CSS
- Links
- Images
- Favicon
- Tables
- Lists
- Block & Inline
- Classes
- Id
- Iframes
- JavaScript
- File Paths

- Head
- Layout
- Responsive
- Style Guide
- Entities
- Symbols
- Emojis
- Charset
- URL Encode
- Forms
- Form Attributes
- Form Elements
- Input Types
- Input Attributes
- Input Form Attributes
- Graphics
- Canvas
- SVG

2. Module 2 – CSS and layout

- Introduction
- Syntax
- Selectors
- How To
- Comments
- Colors
- Backgrounds
- Borders
- Margins
- Padding
- Height/Width
- Box Model
- Outline
- Text
- Fonts

- Icons
- Links
- Lists
- Tables
- Display
- Max-width
- Position
- Z-index
- Overflow
- Float
- Inline-block
- Align
- Combinators
- Pseudo-class
- Pseudo-element
- Opacity
- Navigation Bar
- Dropdowns
- Image Gallery
- Image Sprites
- Attr Selectors
- Forms
- Counters
- Website Layout
- Units
- Specificity
- !important
- Math Functions

3. Module 3 — Browser programming and assets

- Introductory JavaScript for page interaction and form behaviour.
- Basic image preparation for web use, using Photoshop where available or a suitable image editor.
- Relate HTML, CSS and JavaScript in a small page; this is not a complete graphics or JavaScript specialisation.

4. Module 4 – Python language foundations

- [Python Intro](#)
- [Python Get Started](#)
- [Python Syntax](#)
- [Python Comments](#)
- [Python Variables](#)
- [Python Data Types](#)
- [Python Numbers](#)
- [Python Casting](#)
- [Python Strings](#)
- [Python Booleans](#)
- [Python Operators](#)
- [Python Lists](#)
- [Python Tuples](#)
- [Python Sets](#)
- [Python Dictionaries](#)
- [Python If...Else](#)
- [Python While Loops](#)
- [Python For Loops](#)
- [Python Functions](#)
- [Python Lambda](#)
- [Python Arrays](#)
- [Python Classes/Objects](#)
- [Python Inheritance](#)
- [Python Iterators](#)
- [Python Scope](#)

5. Module 5 – Python modules and files

- [Python Modules](#)
- [Python Dates](#)
- [Python Math](#)
- [Python JSON](#)
- [Python RegEx](#)
- [Python PIP](#)

- Python Try...Except
- Python User Input
- Python String Formatting
- File Handling
- Python File Handling
- Python Read Files
- Python Write/Create Files
- Python Delete Files

6. Module 6 – Django application foundations

- Basics
- Overview
- Environment
- Creating a Project
- Apps Life Cycle
- Admin Interface
- Creating Views
- URL Mapping
- Template System
- Models
- Page Redirection
- Sending E-mails
- Generic Views
- Form Processing
- File Uploading
- Apache Setup
- Cookies Handling
- Sessions
- Caching
- Comments

Comments are an application feature example, not an assertion that Django includes a current built-in comments framework. Apache deployment is a selected alternative demonstration.

7. Module 7 — REST API and backend data

- Models
- Serializers
- Write Only Fields
- Password storage using Django password-management APIs
- Login
- JWT token flow using a compatible authentication library
- Getting the Authenticated User
- Logout and token-expiry/revocation behaviour
- Foreign Keys
- Fixtures
- Getting Permissions
- ViewSets
- Roles
- Related Fields
- Completing Orders
- Generic Views
- Pagination
- Update Request Data
- Profile
- User Permissions
- Routes
- Uploading Images
- Serializer Method Fields
- Export CSV
- Parameterised raw SQL overview
- View Permissions

Use framework password APIs, authentication and permission checks. JWT is supplied through a compatible library; logout/revocation depends on the implemented token policy, and deleting a client token alone does not invalidate every issued token. Raw SQL is a bounded parameterised example, not a replacement for the ORM.

8. Module 8 — Vue interface integration

- Installation
- Template

- Router
- Form Submit
- Register
- Login
- Child Routes
- Logout
- Component Props
- Users
- Router Link Active
- Pagination
- Deleting Users
- Classes and Interfaces
- Creating Users
- Updating Users
- Roles
- Creating Roles
- Updating Roles
- Paginator Component
- Uploading Images
- Image Component
- Export CSV files
- Chart
- Profile

Classes/interfaces are introductory JavaScript/TypeScript language concepts used where the prepared Vue example requires them.

9. Module 9 — State and permissions

- Shared state, actions and getters using current Pinia patterns in the prepared Vue application.
- Compare legacy Vuex dispatch/modules and TypeScript getter examples so the source terminology remains understandable.
- UI permission controls complement, rather than replace, server-side checks.

10. Module 10 — Guided deployment and review

- Domain configuration and A/AAAA DNS records for IPv4/IPv6.
- A prepared Unicorn/Nginx deployment, TLS certificate configuration and authorised remote access.

- Development versus deployment settings, secrets, host checks, static/media handling and review of the integrated project.
- Deployment practice occurs in a prepared lab; store, hosting and infrastructure operations require the relevant access and maintenance.