# *Week 1* : From Frontend Fundamentals to Django Deployment

*Web Development Fundamentals*

**M1:HTML and CSS Essentials (2 Days)**

Introduction to HTML structure, tags, tables, forms, and semantic elements

Styling with CSS: selectors, flexbox, layouts, and responsive design best practices

**Mini Project**: Create a profile card and a responsive contact form

**M2:JavaScript and jQuery Fundamentals (2 Days)**

Core JavaScript concepts: variables, functions, loops, and DOM manipulation

Introduction to jQuery: event handling, selectors, and basic AJAX operations

**Mini Project**: Build a dynamic To-Do application with jQuery (add, edit, delete)

**M3:Working with APIs and AJAX (1 Day)**

Introduction to `fetch()`, handling JSON data, and integrating public APIs

**Mini Project**: Create a simple app using a public Weather or Joke API

## *M1 Materials*

Topics Covered:

Introduction to HTML structure and document flow

Key HTML tags: headings, paragraphs, links, images, tables, forms, semantic elements (header, nav, article, section, footer)

CSS Basics: selectors (class, ID, attribute), box model, flexbox, layout techniques

Responsive design using media queries

Introduction to modern best practices (accessibility, semantic HTML)

Materials Needed:

**Text Editor**: VSCode (recommended)

**Browser**: Chrome or Firefox (latest version)

**Extensions**: Live Server extension for VSCode (for live preview)

**Reference Docs**:

https://www.w3schools.com/html/

https://www.w3schools.com/css/

Mini Project:

**Create a Profile Card and Responsive Contact Form**

Design a simple personal profile card (photo, name, title, short bio).

Build a contact form with Name, Email, Message fields, and ensure mobile responsiveness.

.

## *M2 Materials*

Topics Covered:

JavaScript Fundamentals:

Variables (let, const)

Functions and Arrow Functions

Loops (for, while) and Conditionals

DOM Selection and Manipulation (querySelector, getElementById, .value, .innerText)

Event Handling (onclick, addEventListener)

Introduction to jQuery:

jQuery selectors: $("#id"), $(".class")

Event Handling: .click(), .submit()

Basic AJAX methods: $.get(), $.post()

Materials Needed:

**Text Editor**: VSCode

**Browser Console**: Chrome Developer Tools

**Reference Docs**:

[https://www.w3schools.com/js/default.asp](https://www.w3schools.com/js/default.asp)

[https://www.w3schools.com/jquery/default.asp](https://www.w3schools.com/jquery/default.asp)

Mini Project:

**Build a Dynamic To-Do Application with jQuery**

Allow users to add, edit, and delete tasks.

Use DOM manipulation for list items.

Validate empty inputs before adding a task.

## *M3 Materials*

Introduction to HTTP Requests: GET, POST

Using `fetch()` API to call external services

Handling JSON data (parse and display)

AJAX Requests using jQuery (`$.ajax()`)

### Materials Needed:

**Libraries**:

jQuery (already included from Module 2)

**Reference Docs**:

Using Fetch

Free Public APIs Collection

### Example Public APIs:

https://official-joke-api.appspot.com/random_joke

https://api.adviceslip.com/advice

## 📦 Deliverables for Week 1:

1. ✅ Responsive Profile Card
2. ✅ Responsive Contact Form
3. ✅ Functional To-Do List App with jQuery
4. ✅ Public API App (weather, jokes, advice, etc.)

# Week 2

### Day 1: Bootstrap Setup + HTML Recap

Topics:

- Add Bootstrap CDN to HTML template
- Layout: container, row, col, mt-4, text-center
- Components: Navbar, Card, Button, Image
- Reusable HTML structure using base.html

Task:

- Create a simple personal profile layout using Bootstrap
- Add Bootstrap navbar with "Home" and "Posts" links

### Day 2: Python Fundamentals Refresher

Topics:

- Variables, conditionals, loops, functions
- Lists, dictionaries, and iterating over data
- OOP: Define and use classes
- File I/O: open, readlines, write
- Exception handling: try-except

Task:

- Build a **Student CLI Manager**: Add/view/delete students via terminal
- Save and load data using a text file

### Day 3: Django Installation + Project Initialization

Topics:

- Install Django via pip install django
- Create a new project & app
- Explore Django project structure:
- settings.py, urls.py, views.py
- Run the development server and open in browser

Task:

- Create a Django project called mysite
- Set up a home view that returns "Hello Django"

## Day 4: Views, URLs & Templates

Topics:

- Function-based views (FBV)
- Django URL dispatcher (project & app-level)
- Create templates folder and render HTML files using render()
- Pass dynamic data from views to templates
- Use base.html with {% block content %}

Task:

- Create views: Home and About pages
- Use Bootstrap to style templates
- Navigation using Django URLs and Bootstrap navbar

## Day 5: Models + Django Admin

Topics:

- Define a Post model with title, content, created_at
- Apply migrations: makemigrations, migrate
- Register the model in admin.py
- Create and manage data via Django Admin

Task:

- Add 3 blog posts from the admin panel
- Display post list using Bootstrap card layout in /posts/

## Day 6: Dynamic Templates & Looping Data

Topics:

- Template tags: for, if, url, static
- Render dynamic content from model
- Include partials: navbar.html, footer.html
- Use filters: {{ created_at|date:"M d, Y" }}

Task:

- List blog posts with title, short content, and date
- Use Bootstrap Cards with post data inside a loop

## Day 7: Navigation & Wrap-up

Topics:

- Create consistent layout using base.html
- Polish design: spacing, grid, responsive image/text
- Link all pages in navbar
- Review all static files, templates, and views

Task:

- Final check: Homepage, Posts page, About page
- Mobile-friendly layout test using Chrome DevTools

---

## 📦 Week 2 Deliverables

✅ Bootstrap base.html with navbar and layout
✅ Python CLI: Student Manager
✅ Django project with working URLs, views, templates
✅ Blog post model with admin integration
✅ Responsive blog post list page using Bootstrap