**🎓 Unit 01: Introduction to Django and Setting Up the Environment (Week 1)**

**Learning Outcomes**: understand Django's background, its advantages, and complete project setup using best practices.

**Topics & Add-ons:**

* **History of Django & Its Ecosystem**
  + Real-world usage: Instagram, Pinterest, Disqus
* **What is Django & Why Django?**
  + Comparisons with Flask, Laravel, Node.js
* **MVT Architecture**
  + Real-world analogy to clarify differences from MVC
* **Setup Environment**
  + Use VS Code with extensions (Python, Django)
  + Setup using venv, pip, .env with python-decouple
  + Introduce django-admin and manage.py commands
  + Explain folder structure and responsibilities

**Activities:**

* Quick install and setup
* Create and run first Django project
* Live demo with basic homepage view

**🎓 Unit 02: Django Models and Database (Week 2)**

**Learning Outcomes**: Learn Django ORM, how models interact with DB, and admin management.

**Topics & Add-ons:**

* Create basic models (Student, Course, etc.)
* Migrations and their purpose
* SQLite for development, intro to MySQL/PostgreSQL
* Custom queries: .filter(), .get(), .exclude()
* Setup Admin panel, add custom display methods

**Activities:**

* Create 2–3 related models with ForeignKey
* Customize admin interface
* Explain Django shell usage for ORM operations

**🎓 Unit 03: Django Views and Templates (Week 3)**

**Learning Outcomes**: Master views and frontend rendering with templates.

**Topics & Add-ons:**

* Function vs Class-based views (when and why)
* Template rendering with context
* DTL (Django Template Language) basics
* Use Bootstrap via CDN for UI
* Static files configuration (settings.py, collectstatic)
* Template inheritance with base layout

**Activities:**

* Create Blog layout: base.html + index.html + detail.html
* Navigation bar using template inheritance
* Create dynamic templates using passed data

**🎓 Unit 04: Forms, Authentication and CRUD (Weeks 4–5)**

**Learning Outcomes**: Implement secure, dynamic form handling and user auth.

**Topics & Add-ons:**

* ModelForm vs Form
* Validating form data (clean\_ methods)
* Create login, register, logout views
* Session usage & flash messages
* Password hashing, authentication backends
* Custom user model (extend AbstractUser)
* CRUD operations (Add/Edit/Delete students/courses)
* SweetAlert2 for UI messages (cdn + JS + message tags)

**Activities:**

* Build Register/Login system with auth
* Form error handling and success message
* Create/update/delete records with feedback

**🎓 Unit 05: Advanced Concepts (Weeks 6–7)**

**Learning Outcomes**: Handle real-life project requirements using advanced features.

**Topics & Add-ons:**

* Profile model linked via OneToOneField
* Role-based access using decorators and middleware
* Send OTP using Email backend (SMTP)
* Upload profile images, timestamp save
* Use pandas and openpyxl for Excel/CSV handling
* Generate downloadable PDFs (with xhtml2pdf or WeasyPrint)

**Activities:**

* Profile page with image upload
* Admin vs Student dashboard using roles
* Password reset via email simulation
* Import/export Excel and generate PDF reports

**🌟 Bonus Suggestions to Keep Students Engaged:**

* Assign a **mini-project at the end of each unit**
* Create a **GitHub classroom** for pushing code
* Conduct a **capstone project**: "Student Portal" or "Mini CRM"
* Show them how to deploy project on **Render / Railway**
* Introduce Django REST Framework at end (Optional bonus unit)
* Schedule 1 hands-on lab every 2 weeks
* Show 1 open-source Django repo walkthrough