DevOpsSchool

Course curriculum · November 2026

LEVEL 04 OF 04 · ADVANCED

Django Advanced Training

Take Django to production with caching, async and background tasks, PostgreSQL features, hardened security, containers and CI/CD.

LIVE ONLINE **Max 8-10 hours**	CORPORATE (DAY-WISE) **5 days · Days 1-5**	SELF-PACED VIDEO **1-year LMS access**
BATCH SIZE **Max 10 learners**	NEXT BATCHES **1st of every month · weekends: first Saturday**	CERTIFICATE **Verifiable certificate of completion**

Foundation	Essential	Intermediate	Advanced
1 day · Included	2 days · Included	3 days · Included	5 days · This curriculum

INSIDE THIS CURRICULUM

What you will achieve	Outcomes and level focus
Who should attend	Audience and prerequisites
Agenda	51 topics · Foundation to Advanced
Hands-on labs	4 labs
Formats and fees	Live, 1:1, self-paced, corporate
Trainer and reviews	Rajesh Kumar
Enrolment	Batches, payment, contacts

WHY DEVOPSSCHOOL

- Rated 4.6 on Google
- 1,500+ verifiable certificates issued in the public registry
- Trainer Rajesh Kumar: 20 years in industry
- Small batches: max 10 learners, every session recorded

Course page: www.devopsschool.com/courses/python/django-advance-training.html
Talk to us: WhatsApp +91 99057 40781 · +1 (469) 756-6329 · contact@DevOpsSchool.com · reply within 1 hour

OUTCOMES

What you will achieve

By the end of the Django course path, you will be able to:

1

Build Django web applications

Create Django projects with models, views, URL routing, forms and the admin.

2

Develop secure APIs and auth

Add authentication, permissions and REST APIs with Django REST Framework and tests.

3

Run Django in production

Optimise, secure, containerise and deploy Django applications with CI/CD.

WHAT THE ADVANCED LEVEL ADDS

Performance, async, deployment and production security

Hands-on: Deploy a production Django app. Containerise the bookings app with Gunicorn, Nginx and PostgreSQL, add Redis caching and Celery reminders, and deploy it through a CI pipeline.

WHO SHOULD ATTEND

- Experienced Django practitioners responsible for quality, security and scale.
- Senior engineers, architects and leads who design and troubleshoot Django solutions.
- Teams making Django an organisation-wide standard.

BEFORE YOU START

- The Intermediate level, or equivalent hands-on experience.
- You can already: Build maintainable Django apps with class-based views, efficient ORM queries, authentication, REST APIs, tests and safe settings.
- Basic computer skills and a laptop with a stable internet connection for the live sessions and labs.
- This curriculum includes every earlier level, so it is complete on its own.

AGENDA · FOUNDATION TO ADVANCED

Django Advanced Training agenda

51 topics. Each level builds on the one before it, so this curriculum lists every earlier level in full. Days are for corporate (day-wise) delivery; live online batches cover the same agenda in max 8-10 hours.

DAY 1 — Foundation: Django setup, first app, pages and models

Included from Foundation

1. Course overview and introducing Django: what Django is, batteries included, Django 5.2 LTS and 6.0
2. Prerequisites and project preview: Python 3.12 or later, virtual environments and the app you will build
3. Installing Django: pip install django in a venv, django-admin version check and editor setup
4. Starting a project: django-admin startproject, manage.py, settings.py, urls.py and runserver
5. Creating a Django app: python manage.py startapp, registering it in INSTALLED_APPS
6. Creating a simple web page: adding a page with a view function and a URL pattern
7. When it doesn't work: reading tracebacks, the debug page and common setup mistakes
8. Django and the flow of control: request, URL resolver, view and response; adding an About page
9. Introducing models and migrations: the ORM, running initial migrations and the default SQLite database
10. Creating a model class: fields, field options and __str__, then creating and running migrations
11. The admin interface: createsuperuser, registering models and managing data in Django admin
12. Bringing it all together: adding a second model class with a ForeignKey relationship

DAY 2 — Essential: MTV pattern, URLs, styling and ModelForms

Included from Essential

13. Combining model, view and page output: querying models in views and passing context
14. Introducing the MTV pattern: Django templating language and a page for the welcome view
15. Dynamic content: variables, filters and tags in Django pages
16. Completing the MTV pattern: taking a parameter from the URL and returning a 404 error with get_object_or_404
17. URLs and link building: project structure overview, path converters, named URLs and the url tag
18. Listing data: a page with a for loop and an exercise listing all rooms
19. Best practice for URLs and apps: app namespaces, include() and keeping apps focused
20. Styling and static content: adding styling with CSS, adding an image and the static files app
21. Page inheritance: base layouts with extends and block for consistent pages
22. Adding user interaction with ModelForms: a page with a ModelForm and submitting the form
23. Understanding the program flow for processing a form: GET vs POST, is_valid and redirect after POST
24. Processing user input: cleaned_data, saving model instances and user feedback
25. Validation and security: CSRF tokens, escaping output, customising form fields and validation

DAY 3 — Intermediate: Class-based views, ORM, auth, APIs and testing

Included from Intermediate

26. Class-based views: ListView, DetailView, CreateView, UpdateView and DeleteView with mixins
27. ORM queries: filter, exclude, Q and F objects, annotate, aggregate and ordering
28. Query efficiency: select_related, prefetch_related, only and defer, and spotting N+1 queries
29. Authentication: login and logout views, password reset, the User model and a custom user model
30. Authorization: permissions, groups, LoginRequiredMixin and PermissionRequiredMixin

31. Sessions and messages: the session framework, the messages framework and cookie settings
32. Django REST Framework: serializers, APIView, viewsets and routers for a JSON API
33. File uploads and media: FileField, ImageField, MEDIA_ROOT and safe handling of uploads
34. Extending Django: custom middleware, signals and custom management commands
35. Testing: TestCase, the test client, factories with factory_boy and pytest-django
36. Settings management: environment variables, separate settings per environment and secret keys
37. Security basics: CSRF, XSS, SQL injection and clickjacking protections, manage.py check --deploy
38. Debugging: Django Debug Toolbar, logging configuration and troubleshooting migrations

DAYS 4-5 **Advanced: Performance, async, deployment and production security** **New at this level**

39. Caching: per-view and low-level caching with Redis, cache invalidation and database indexes
40. Async Django: async views, ASGI servers and when async helps
41. Real-time features: Django Channels, WebSockets and channel layers
42. Background work: Celery with Redis or RabbitMQ, periodic tasks and the tasks framework in Django 6.0
43. PostgreSQL features: JSONField, full-text search, constraints and database functions
44. Advanced DRF: token and JWT authentication, throttling, pagination and OpenAPI docs with drf-spectacular
45. Security hardening: HTTPS settings, HSTS, secure cookies, Content Security Policy and dependency audits
46. Deployment: Gunicorn or Uvicorn behind Nginx, WhiteNoise for static files and Docker images
47. CI/CD: running tests, linting and migrations in GitHub Actions or GitLab CI with zero-downtime deploys
48. Scaling: database routers, read replicas, connection pooling and horizontal scaling
49. Larger products: internationalisation with gettext and locale middleware, multi-tenancy patterns
50. Upgrading Django: deprecation warnings, moving between LTS releases and third-party package compatibility
51. Capstone: build, secure and deploy a production-ready Django application with an API and background jobs

PRACTICE

Hands-on labs

You build these yourself during the course, guided by the trainer.

FOUNDATION

Start a meeting rooms website

Create a Django project and app with home and About pages, define Room and Meeting models, migrate them and manage records in the admin.

ESSENTIAL

Add booking forms to the site

Add room detail pages with URL parameters, a styled base layout with static files and a ModelForm to book meetings with custom validation.

INTERMEDIATE

Expose bookings through a tested API

Convert views to class-based views, add login and permissions, build a DRF API for bookings and cover it with automated tests.

ADVANCED

Deploy a production Django app

Containerise the bookings app with Gunicorn, Nginx and PostgreSQL, add Redis caching and Celery reminders, and deploy it through a CI pipeline.

HOW YOU CAN LEARN

Formats and fees

The same agenda in four formats. Fees are in Indian rupees as of November 2026; the course page shows them in your currency.

FORMAT	DURATION	WHAT YOU GET	FEE
Live & interactive (online)	Max 8-10 hours	Instructor-led public batch of max 10 learners with demos, labs, Q&A and assignment review. Weekday batches start on the 1st of every month, weekend batches on the first Saturday of every month.	**INR 34,999**
1:1 live training (online)	Max 8-10 hours, one-to-one	The same agenda, taught one-to-one by the trainer. Starts immediately, without waiting for the next batch; sessions planned around your working hours, at your pace.	**INR 99,999**
Self-paced video	1-year LMS access	All four Django levels, plus 20+ more courses and 50+ tools, at your own pace.	**INR 833/month** billed yearly at INR 9,996
Corporate (day-wise)	5 days for this level	A private batch for your team, on demand. Foundation 1 day, Essential 2, Intermediate 3, Advanced 5 (cumulative).	**Quote on request**

ASSESSMENT AND CERTIFICATE

Your progress is checked through hands-on labs and assignments at every level. On completion you receive a DevOpsSchool certificate of completion with a unique credential ID that anyone can verify at devopsschool.com/certificates.

INCLUDED WITH EVERY LIVE BATCH

- Every live session is recorded and stays in your LMS for a year.
- Can't attend your batch? Move to next month's batch.
- Free demo class before you pay.
- Free 10-15 minute call with an advisor or the trainer to check the course fits you.

YOUR TRAINER

Learn from Rajesh Kumar

Principal Architect, trainer and mentor in DevOps, SRE, DevSecOps and platform engineering, with 20 years across 8+ software MNCs, including Staff DevOps/SRE Engineer for a leading Japanese mobile-payment platform and Principal DevOps Architect and Manager in Europe.

Profile: www.rajeshkumar.xyz · Videos: youtube.com/@TheDevOpsSchool

20 years in industry	**200+** corporate trainings
10,000+ engineers mentored	**70+** organisations coached

REVIEWS

What learners say

Rated 4.6 on Google · more reviews at devopsschool.com/testimonials and in our video testimonials.

> “Rajesh is very good trainer. Rajesh was able to resolve our queries and question effectively. We really liked the hands-on examples covered during this training program.”
>
> Indrayani, India

> “The training was very useful and interactive. Rajesh helped develop the confidence of all.”
>
> Abhinav Gupta, Pune

START YOUR DJANGO TRAINING

Join the next batch

Weekday batches start on the 1st of every month and weekend batches on the first Saturday of every month, with max 10 learners each. 1:1 training can start immediately; corporate batches run on demand. We reply within 1 hour.

Pay by UPI, credit and debit cards, net banking or PayPal. Before you pay, read our refund policy. All levels of this course: www.devopsschool.com/courses/python/

ENROL OR RESERVE A SEAT
Django Advanced Training

WHATSAPP · INDIA
+91 99057 40781

WHATSAPP · USA
+1 (469) 756-6329

EMAIL
contact@DevOpsSchool.com