

# TRAINING DAY28 REPORT

25 JULY 2025

## Conclusion and Summary of Django & DRF Training

### Summary of Learning:

#### 1. Django Fundamentals

- Learned about **MVC (Model-View-Template) architecture** in Django.
  - Created **models** with fields like title, description, created\_at, and completed.
  - Understood **views** (Function-Based Views and Class-Based Views) to handle HTTP requests.
  - Used **templates** and template tags to render dynamic data on web pages.
  - Implemented **forms and model forms** for user input.
- 

#### 2. Database Handling

- Performed **CRUD operations** on models.
  - Learned about **Django ORM** for querying the database efficiently.
  - Explored **relationships between models** (ForeignKey, ManyToMany, OneToOne).
  - Used **migrations** to update database schema.
-

### 3. Django REST Framework (DRF) Basics

- Introduction to **APIs and RESTful principles**.
  - Learned **serializers** to convert models ↔ JSON data:
    - Serializer (manual)
    - ModelSerializer (automatic)
    - HyperlinkedModelSerializer (RESTful URLs)
  - Created **ViewSet**s and connected them with **routers** for API endpoints.
  - Practiced **GET, POST, PUT, PATCH, DELETE** operations using DRF.
  - Implemented **basic validations** and understood how `create()` and `update()` work.
- 

### 4. Advanced Features

- Implemented **pagination** for large data sets.
  - Explored **context processors** to make variables globally available in templates.
  - Learned **Django Channels basics** for real-time updates (WebSockets).
  - Understood **authentication and permissions** in DRF for secure APIs.
- 

### 5. Practical Projects

- Developed a **Task Manager API**:
  - CRUD operations on tasks
  - Serializer usage

- Optional integration with DRF for API testing
  - Worked on a **Daily Diary API**:
    - Model design with fields like title, content, mood, created\_at
    - Serializer implementation
    - Real-time updates using WebSockets (Channels) [basic concept]
  - Learned to **link frontend with backend** using Django templates and DRF APIs.
- 

## 6. Tools & Skills Acquired

- **Python** programming for backend logic
  - **Django** for building dynamic web applications
  - **Django REST Framework** for API development
  - **HTML, CSS, and Bootstrap** for templates
  - **JavaScript basics** for frontend interaction (optional DRF integration)
- 

## Challenges Faced & Overcome

- Understanding **asynchronous behavior** in Django Channels
- Correctly **serializing nested and related models**
- Implementing **pagination and filtering** in DRF
- Connecting **frontend templates with backend APIs** efficiently

## Conclusion

- The six-week training provided **hands-on experience** in Django and basic DRF.
- Gained confidence in building **dynamic web applications** with CRUD functionality.
- Learned to **design RESTful APIs**, serialize data, and implement secure and reusable code.
- Prepared to **extend Django knowledge** for real-world projects like e-commerce, task management, or chat apps.
- Developed an **end-to-end understanding** of backend development with Python and Django REST Framework.