Evaluation Task – Build a Mini Blog API (Python + Django REST Framework)

Objective:
Evaluate your ability to:

- Work with Python fundamentals (lists, dicts, validation).
- Implement APIs using Django REST Framework (DRF).
- Understand basic API design, request/response handling, and simple authentication.

Requirements (Tier 1 – Mandatory)

Models (in-memory storage only)

You do not need a database. Just store data in **Python lists/dictionaries**.

- **User**
  - Use a static list of users (e.g., `users = [{"id":1,"name":"Shubham","token":"abc123"}]`)
  - Authentication → via a **hardcoded token** or simple login simulation.
- **Post**
  - Fields → `id`, `title`, `content`, `author`, `created_at`
- **Comment**
  - Fields → `id`, `post`, `text`, `author`, `created_at`

API Endpoints

- **POST** /api/posts/
  → Create a new blog post (**authentication required**).
- **GET** /api/posts/
  → List all posts (**latest first**).
  - Support pagination → ?page=1&page_size=5.
- **GET** /api/posts/{id}/
  → Get details of a single post with its comments.
- **POST** /api/posts/{id}/comments/
  → Add a comment to a post (**authentication required**).

Authentication (Simplified)

- Use a **hardcoded token** (e.g., "abc123") or static login user.
- Pass token in headers (e.g., Authorization: Token abc123).

Acceptance Criteria

✅ Input validation (title/content cannot be empty, must be within reasonable length).
✅ Posts should be returned **latest first**.
✅ Pagination should work correctly.
✅ Clean code with comments.
✅ A **README.md** file with:

- Setup instructions
- Example requests/responses for each endpoint

Optional (Bonus)

- Replace in-memory storage with **SQLite/MySQL** using Django ORM.

- Implement **JWT authentication** using DRF’s built-in mechanisms.

- Add **Update/Delete APIs** for posts and comments.

Submission Guidelines

1. Create a **GitHub repo** (public or private with access).
2. Include all code + README.md with setup & usage instructions.
3. Ensure API is testable via **Postman or curl commands**.

Timeline

Total Time: **4-5 Days**