

Django Developer Task

Project & Multi-Tenancy Setup

Your first task is to establish the foundation of the application. You will create a new Django project and configure it for a multi-tenant architecture.

Tasks:

- **Project Initialization:** Set up a new Django project and a single app. Use a virtual environment and install necessary dependencies, including a multi-tenancy package like `django-tenant-schemas`.
- **Multi-Tenancy Integration:** Implement the chosen multi-tenancy solution. This includes configuring the `settings.py` file, setting up the `Tenant` and `Domain` models, and ensuring tenant-specific schemas can be created and managed.
- **Tenant-Specific Model:** Create a basic tenant-specific model (e.g., a `Product` model) and confirm it can be migrated and accessed correctly within individual tenant schemas.

Docker & CI Pipeline Foundation

Tasks:

- **Dockerization:** Write a `Dockerfile` to containerize the Django application. Use a multi-stage build to optimize the image size. Create a `docker-compose.yml` file to run the Django app alongside a PostgreSQL database locally.
- **Local Test:** Confirm that the application runs successfully using `docker-compose up`.
- **GitHub Actions Workflow:** Create a GitHub Actions workflow file (`ci.yml`) to perform the following actions on every `push` and `pull_request`:
 - Run the Django test suite (`python manage.py test`).
 - Build the Docker image for the application.
- **Versioning:** Establish a versioning strategy (e.g., `MAJOR.MINOR.PATCH`) for the application and its Docker images. Plan to use Git tags to trigger new releases.

Advanced CI/CD & Documentation

Tasks:

- **Refine CI Pipeline:** Enhance the `ci.yml` workflow. Add a linting step (e.g., `flake8` or `black`) to enforce code standards. Configure the pipeline to push the Docker image to a container registry (e.g., GitHub Container Registry) when a new Git tag is created. Ensure the image is tagged with both the version (e.g., `v1.0.0`) and `latest`.
- **Deployment Preparation:** Create a separate production settings file. Update the Docker configuration to use these settings.
- **Documentation:** Write a comprehensive `README.md` file. This should explain how to set up the project locally with Docker Compose and detail the CI/CD process and versioning strategy you implemented.