

Title: WriteFlow – Blog Content Management System

1. Project Goals

- Develop a backend-powered **Blog Content Management System** to manage blog posts (create, read, update, delete).
- Containerize the backend using **Docker** for portability and consistent deployment.
- Implement a **CI/CD pipeline** for automated building, testing, and deployment.
- Deploy the application on a **cloud platform** (AWS EC2 / DigitalOcean / Render).
- Practice **DevOps best practices** like version control, environment variable management, and automated workflows.

2. Project Requirements

2.1 Functional Requirements

1. **Post Management**
 - Create new blog posts.
 - View all blog posts.
 - View individual blog post details.
 - Update existing blog posts.
 - Delete blog posts.
2. **User Authentication (Optional for Beginner)**
 - Secure login for authors.
 - Token-based authentication (JWT).
3. **API Documentation**
 - Swagger or Postman collection for API testing.

2.2 Non-functional Requirements

1. **Performance**
 - API should respond within **500ms** for typical requests.
2. **Scalability**
 - Should be deployable in cloud environments with minimal configuration changes.
3. **Portability**
 - Entire application should run inside Docker containers.
4. **Reliability**
 - Persistent data storage using PostgreSQL with volume mapping.
5. **Maintainability**
 - Code should follow standard conventions and be version-controlled on GitHub.
6. **Security**
 - Use environment variables for credentials and database configuration.
7. **CI/CD Automation**
 - Automatic build and deploy upon push to the main branch.

3. Project Roadmap

Phase	Tasks	Tools/Tech
Phase 1 – Planning	Define features, architecture, tech stack	Notion, Draw.io
Phase 2 – Backend Development	Create Django/Node.js API with CRUD endpoints	Django REST Framework / Express.js, PostgreSQL
Phase 3 – Containerization	Create Dockerfile & docker-compose for backend + DB	Docker, Docker Compose
Phase 4 – CI/CD Setup	Configure GitHub Actions to build & push Docker image	GitHub Actions, Docker Hub
Phase 5 – Deployment	Deploy containerized app to cloud (AWS EC2 / Render)	AWS CLI, Docker
Phase 6 – Testing & Documentation	Test APIs, write README, add API docs	Postman, Swagger, Markdown