Role-Based Content Management System (CMS)

Objective

Develop an API for a Content Management System that allows different types of users (e.g., Admins, Editors, Authors, and Readers) to manage and access content based on their roles. The API will include role-based access to specific actions, such as creating, editing, reviewing, and publishing articles.

Features

1. **User Management**
   - Register new users with roles like Admin, Editor, Author, and Reader.
   - Log in users and assign JWT tokens for session management.
   - Admins can update user roles and permissions.
2. **Role-Based Permissions**
   - Define access permissions for each role:
     - **Admin**: Full access to all resources, including managing user roles and system settings.
     - **Editor**: Can review, edit, and publish articles created by Authors.
     - **Author**: Can create, edit, and submit articles for review.
     - **Reader**: Can view published articles only.
   - Implement role-based restrictions at the API level to ensure users can only perform actions allowed by their roles.
3. **Content Management**
   - CRUD operations for managing articles, categories, and tags.
     - **Admin** and **Editor**: Full CRUD on all articles, categories, and tags.
     - **Author**: Can create and edit their own articles, submit them for review.
     - **Reader**: Can only view published articles.
   - Track article status (Draft, In Review, Published, Rejected).
4. **Comments and Feedback**
   - Allow users to add comments on published articles.
   - Role-based comment moderation: Admins and Editors can manage comments.
5. **Search and Filtering**
   - Enable readers to search for articles by categories, tags, or keywords.
   - Filter articles by publication status, author, or date.

Technology Stack

- **FastAPI**: For building the REST API.
- **SQLAlchemy**: To manage the database schema and relationships.
- **Pydantic**: For defining request and response models, and validating input data.
- **JWT**: For authentication and role-based authorization.
- **PostgreSQL**: For storing user data, roles, and content.