🗂️ Models

RoleModel
Field Name	Datatype	Primary Key	Comments
id	Integer	Yes	Autoincrement
rolename	String	No	Values: "AUTHOR", "EDITOR"

UserModel
Field Name	Datatype	Primary Key	Foreign Key	Comments
id	Integer	Yes	No	Autoincrement
username	String	No	No	-
email	String	No	No	Unique
password	String	No	No	-
role	Integer	No	Yes	FK to RoleModel

ArticleModel
Field Name	Datatype	Primary Key	Foreign Key	Comments
id	Integer	Yes	No	Autoincrement
title	String	No	No	-
content	String	No	No	-
category	String	No	No	Example: "Tech", "Health"
wordCount	Integer	No	No	Auto-calculated from content
createdDate	String	No	No	Default: Current date (dd/MM/yyyy)
status	String	No	No	Default: "draft"
authorId	Integer	No	Yes	FK to UserModel

🔐 JWT Authentication
- JWT token required in Authorization header:
  `
  Authorization: Bearer <JWT_TOKEN>
  `
- Roles (AUTHOR / EDITOR) are identified from the token.
- All endpoints except /login require authentication.

🌐 Endpoints

1. POST /login
- Purpose: Authenticate user and return JWT.
- Request Body:
`json
{
  "email": "author1@gmail.com",
  "password": "auth123$"
}
`
- Success Response (200 OK):
`json
{
  "email": "author1@gmail.com",
  "access": "<access_token>",
  "refresh": "<refresh_token>",
  "status": 200
}
`

---

2. POST /article/add (AUTHOR only)
- Adds a new article.
- Request Body:
`json
{
  "title": "AI in 2025",
  "content": "Artificial Intelligence is evolving rapidly...",
  "category": "Tech"
}
`
- Manipulation: wordCount auto-calculated from content.
- Success Response (201 CREATED):
`json
{
  "id": 301,
  "title": "AI in 2025",
  "category": "Tech",
  "wordCount": 6,
  "createdDate": "29/11/2025",
  "status": "draft",
  "authorId": 1
}
`

---

3. GET /article/list?category=Tech&minWords=500 (Both roles)
- Returns all articles filtered by:
  - category
  - minWords (articles with wordCount ≥ given value)
- Success: 200 OK with list.
- Error: 400 Bad Request → "no articles found"

---

4. PATCH /article/update/{id} (EDITOR only)
- Updates article status.
- Request Body:
`json
{ "status": "published" }
`
- Success Response (200 OK): Updated article object.
- Error: 400 Bad Request → "article not found"

---

5. DELETE /article/delete/{id} (Conditional access)
- Accessible by:
  - Editor
  - Author who created the article
- Success: 204 No Content → "deleted successfully"
- Error:
  - 400 Bad Request → "article not found"
  - 403 Forbidden → "you don't have permission"

📑 Response Codes
- 200 OK – Success
- 201 CREATED – Resource created
- 204 No Content – Deleted successfully
- 400 Bad Request – Invalid input / Not found
- 403 Forbidden – Permission denied

✅ This new problem statement adds:
- RoleModel with FK (AUTHOR / EDITOR).
- Manipulation: wordCount auto-calculated from content.
- Query params: filter by category and minWords.
