Library Catalogue

1. Description of the Topic:

A **Library Catalogue API** to manage books, authors, and categories. This API allows users to view, add, update, and delete information about books and their metadata while maintaining RESTful design principles.

---

2. Entities and Relationships:

- **Books**: Represent the books in the library.
  - Attributes: `id`, `title`, `isbn`, `publication_year`, `description`, `category_id`, `author_ids`.
- **Authors**: Represent authors of books.
  - Attributes: `id`, `name`, `bio`.
- **Categories**: Represent genres or classifications.
  - Attributes: `id`, `name`, `description`.

Relationships:

- A book belongs to one category but can have multiple authors.
- Authors can write multiple books.
- Categories can have multiple books.

---

3. Operations Supported by the API:

For Books:

- **GET** `/books`: Retrieve all books (with pagination, filtering by title, category, and author).
- **GET** `/books/{id}`: Retrieve a specific book by ID.
- **POST** `/books`: Add a new book.
- **PUT** `/books/{id}`: Update an existing book.
- **DELETE** `/books/{id}`: Delete a book.

For Authors:

- **GET** `/authors`: Retrieve all authors (with pagination and filtering by name).
- **GET** `/authors/{id}`: Retrieve a specific author by ID.
- **POST** `/authors`: Add a new author.
- **PUT** `/authors/{id}`: Update an author.
- **DELETE** `/authors/{id}`: Delete an author.

For Categories:

- **GET** `/categories`: Retrieve all categories (with pagination).
- **GET** `/categories/{id}`: Retrieve a specific category by ID.
- **POST** `/categories`: Add a new category.
- **PUT** `/categories/{id}`: Update a category.
- **DELETE** `/categories/{id}`: Delete a category.

---

4. REST API Design (JSON-Based):

Base URL:
`https://api.library-catalogue.com/v1`

Collections and Filters:

- **Books**:
  - **Pagination**: `?page=1&limit=10`
  - **Filters**: `?title=keyword`, `?author_id=1`, `?category_id=3`
- **Authors**:
  - **Pagination**: `?page=1&limit=5`
  - **Filters**: `?name=John`
- **Categories**:
  - **Pagination**: `?page=1&limit=5`

Example API Endpoints:

1. `GET /books`

   Response:
   ```
   {
     "page": 1,
     "limit": 10,
     "total": 25,
     "data": [
       {
         "id": 1,
         "title": "RESTful Design",
         "isbn": "978-3-16-148410-0",
         "publication_year": 2022,
         "category": { "id": 3, "name": "Technology" },
         "authors": [
           { "id": 1, "name": "John Doe" },
           { "id": 2, "name": "Jane Doe" }
   ```

```
        ]
      }
    ],
    "_links": {
      "self": "/books?page=1&limit=10",
      "next": "/books?page=2&limit=10"
    }
  }
```

2. POST /books

   Request Body:

```
{
  "title": "New Book",
  "isbn": "123-4-56-789012-3",
  "publication_year": 2024,
  "category_id": 2,
  "author_ids": [1, 2]
}
```

   Response (201):

```
{
  "id": 4,
  "title": "New Book",
  "isbn": "123-4-56-789012-3",
  "publication_year": 2024,
  "category_id": 2,
  "author_ids": [1, 2]
}
```

Error Handling:

- **404 Not Found**: When a requested entity is not found.
- **400 Bad Request**: Invalid input data.
- **401 Unauthorized**: Authentication failure.

---

5. Authentication:

Use **JWT (JSON Web Tokens)** for secure access.

Header:

```
{
  "alg": "HS256",
  "typ": "JWT"
```

```
}
```
Payload (claims):
```
{
  "sub": "user_id",
  "name": "John Doe",
  "iat": 1516239022,
  "roles": ["admin", "user"]
}
```

---

6. Caching:

- **GET requests** are cacheable with expiration headers (`Cache-Control: max-age=3600`).
- **POST/PUT/DELETE requests** are **not** cacheable because they modify the state.

---

7. Richardson Maturity Model:

1. **Level 1**: Resources are identified using URLs (`/books`, `/authors`).
2. **Level 2**: Uses HTTP methods (`GET`, `POST`, `PUT`, `DELETE`) appropriately.
3. **Level 3**: Incorporates HATEOAS with `_links` for navigation.