In-Class Exercise – Movie Ratings System with JPA & Custom Queries

Objective

Extend the Blockbuster API by introducing a new `Rating` entity, linking it to the existing `Movie` and `Customer` entities.

---

1. Create the Rating Entity

- Define a new JPA entity named **Rating**, backed by a `ratings` table.
- Include:
  - An auto-generated `Long id` as the primary key.
  - An integer `rating` field, validated to accept values from 1 to 10.
  - A many-to-one relationship to **Movie**.
  - A many-to-one relationship to **Customer**.

> *The `Rating` entity must have a bi-directional relationship with both the `Movie` and `Customer` entities.*

---

2. Repository Layer: Custom Queries

- In your `RatingRepository`, add two JPQL methods:
  1. Return the top 5 `Movie` entities along with their average ratings, ordered in descending order.
  2. Return each movie's title alongside its average rating across all ratings.
  3. Return each movie title and the number of ratings it has received, ordered from most to least.

---

3. Service Layer: Business Logic

- Create a `RatingService` class with methods to:
  - Call the repository's top-rated query with a page request for 5 items.
  - Call the repository's per-movie average query.
  - Call the repository's rating count per movie query.
  - Retrieve all `Rating` records.
  - Persist a new `Rating`.
  - Look up an existing `Rating` by ID, update its fields (rating value, movie, customer), and save.
  - Delete a `Rating` by ID.

---

4. Controller Layer: REST Endpoints

- Create a `RatingController` under `/api/ratings` with the following endpoints:
  1. **GET /top**
     – Returns the top 5 movies by average rating (raw `Movie` + average value).

2. **GET /averages**
   – Returns a list of movie titles with their average ratings.
3. **GET /**
   – Returns all `Rating` records.
4. **POST /**
   – Accepts a `Rating` JSON payload, saves it, and returns the created object (HTTP 201).
5. **PUT /{id}**
   – Accepts updated `Rating` data for the given ID and returns the updated object.
6. **DELETE /{id}**
   – Deletes the rating by ID.
7. **GET /most-rated**
   – Returns a list of movie titles along with the number of ratings each has received, ordered by count.