

- 1) When user registers he will receive email with activation code, and on specific endpoint if he provides this code, only then he will be able to be authorised;
- 2) He can upload his painting for sale, with some details he will also be able to upload the picture of the painting;
- 3) User can search the paintings on this web app;
- 4) User can order and pay for painting;
- 5) User can leave the review

(Endpoint naming might change)

#### 1. User Registration and Activation

- Endpoint: /register
  - Users register by providing necessary details (username, email, password).
  - An activation code is sent to the user's email.
- Email Service: Sends activation emails with a unique code and an activation link.
- Endpoint: /activate-account
  - Users activate their accounts by visiting this endpoint with the activation code.
  - The code is validated, and upon success, the user's account status is updated to "enabled".

#### 2. Authentication and Authorization

- Endpoints: /login, /logout
  - Uses Spring Security for handling authentication.
  - Only activated users can log in.
- Role Management: Roles like USER, ADMIN are defined to manage permissions.

#### 3. Painting Management

- Endpoint: /paintings/upload
  - Authenticated users can upload their paintings.
  - Details include title, description, price, and the image of the painting.
- Image Storage:
  - Painting details are stored in a database table (e.g., paintings).

#### 4. Searching Paintings

- Endpoint: /paintings/search
  - Users can search for paintings by keywords
- Service Layer:
  - Handles the search logic and querying the database.

#### 5. Ordering and Payment

- Endpoint: /order
  - Users can place orders for paintings which means filling out order form.
  - Order details include user info, painting info, payment details.
  - User will fill out payment form.
  - Order status will be updated

#### 6. Review System

- Endpoint: /reviews
  - Users can leave reviews for paintings;
  - Reviews include comments.

#### 7. Technical Components

- Spring Boot: Main framework for building the backend application.
- Thymeleaf: Template engine for rendering frontend views.
- Spring Data JPA: For interacting with the database.
- Spring Security: For managing authentication and authorization.
- Mail Dev: For sending activation and notification emails.
- Rest Controllers: For handling API requests and responses.
- Services: Business logic and service layer.
- Repositories: Data access layer.

#### Example Flow

1. User Registration
  - User registers via /register endpoint.

- Application sends an email with an activation code.
  - User activates their account via /activate endpoint.
2. Uploading Painting
    - Authenticated user accesses /paintings/upload.
    - Provides painting details and image.
    - Application saves details in the database and stores the image.
  3. Searching Paintings
    - User accesses /paintings/search and enters search criteria.
    - Application retrieves and returns matching paintings from the database.
  4. Placing an Order
    - User selects a painting and places an order via /order.
    - User fills out payment form
    - Order status is updated and stored in the database.
  5. Leaving a Review
    - user leaves a review via /reviews.
    - Application stores the review and optionally, admin can moderate it.