Lab 6

Validation Checklist Creation: plant shop system is an online platform for users to browse and purchase plants, with administrators managing inventory efficiently, ensuring a seamless experience for all users.

❖ Plant Shop

Plants Model Validation:

1. **ID:**
   - @NotNull: Cannot be null.
   - @Min(2) : Length must be more than 2 characters.

2. **Name:**
   - @NotNull: Cannot be null.
   - @Size(min = 3): Length must be more than 3 characters.
   - @Pattern(regexp = "^[a-zA-Z]+$"): Name should consist of characters.

3. **Description:**
   - @NotNull: Cannot be null.
   - @Size(min = 10): Length must be more than 10 characters.

4. **Price:**
   - @NotNull: Cannot be null.
   - @Positive: Must be a positive number.

5. **Availability:**
   - Must be initially set to true.

6. **Category:**
   - @NotNull: Cannot be null.

- @Pattern(regexp = "^(Indoor|Outdoor|Succulents)$"): Category must accept only Indoor, Outdoor or Succulents

7. **Stock Level:**
   - @NotNull: Cannot be null.
   - @Min(0): Minimum Stock will be 0
   - @Positive: Must be a positive number.

PlantShopController Endpoint:

1. **Get all plants:**
   - Retrieves a list of all plants.

2. **Add a new plant:**
   - Adds a new plant to the system.

3. **Update a plant:**
   - Updates an existing plant's information by using the ID.
   - Verify that the plant exists.

4. **Delete a plant:**
   - Deletes a plant from the system.
   - Verify that the plant exists.

5. **Search Plants by Category:**
   - Retrieves a list of plants based on their category (e.g., "Indoor", "Outdoor", "Succulents").
   - Ensure that the category parameter is valid.

6. **Search Plants by Availability:**
   - Retrieves a list of plants based on their availability (true or false).
   - Ensure that the availability parameter is valid.
   -
7. **Search Plants by Price Range:**
   - Retrieves a list of plants within a specified price range.
   - Ensure that minPrice and maxPrice are valid price values.

8. **Order Plant:**

- Allow customers to place an order for a plant.
- Verify that the plant exists and is available in stock.
- Update the stock level accordingly.

9. **Cancel Plant Order:**
   - Allow customers to cancel their plant orders.
   - Verify that the order exists and the plant is in stock.
   - Update the stock level accordingly.
10. **Get Plants Low in Stock:**
    - Retrieves a list of plants with stock levels below a specified threshold.

11. **Mark Plant as Available/Unavailable:**
    - Allow shop administrators to mark a plant as available or unavailable for sale.

12. **Promote Plant:**
    - Allows the shop administrator to promote a plant to a featured status if it meets certain criteria.
    - Verify that the plant with the specified ID exists.
    - Validate that the plant's price is above a certain threshold.
    - Confirm that the plant is not currently featured.
    - Change the plant's status to featured if it meets the criteria.

❖ User:

User Model:

1. **ID:**
   - @NotNull: Cannot be null.
   - @Min(2) : Length must be more than 2 characters.
2. **Username:**
   - @NotNull: Cannot be null.
   - @Size(min = 3): Length must be more than 3 characters.
   - @Pattern(regexp = "^[a-zA-Z0-9_-]+$"): Name should consist of characters and letters.
3. **Password:**
   - @NotNull

- @Size(min = 6)
- @Pattern(regexp = "^(?=.*[a-z])(?=.*[A-Z])(?=.*\\d)(?=.*[@$!%*?&])[A-Za-z\\d@$!%*?&]+$"): Password should consist of at least 1 capital letter, 1 small letter, 1 special character and numbers.

4. **Email:**
   - @NotNull: Cannot be null.
   - @Email

5. **Role:**
   - @NotNull
   - @Pattern(regexp = "^(admin|customer)$"): Role can only be admin or customer.

UserController Endpoint:

1. **Get all users:**
   - Retrieves a list of all users.

1. **Add New User:**
   - Allows users to register for a new account.
   - Validates user input data.
   - Assigns a customer or admin role to the user.

2. **Update User Information:**
   - Allows users to update their information by entering username.
   - Validates user input data.
   - Updates the user's information.

3. **Delete a user:**
   - Allows Admin to delete a customer by entering admin ID and customer username.
   - Validates admin input data.
   - If ID and username match then admin can delete the customer.