Clothes Store

A simple clothes store were users can browse, search, and buy the items . A user can add or remove an item to his/her favorite list. The store Admin can create ,modify, and delete items

Task consists of building a Database model of the store and RESTFUL APIs with the following details:

- **Database**
  - There are four types of users guest user, registered user, worker and admin
  - Each registered user must have credit card credentials in order to buy something .
  - Each item must have unique name, gender, category, price, color, size and brand. (You can add any parameters you want)
  - Each item should have quantity by size and color
  - Create an enum for the category, color, and gender
  - Each item soled should be stored in DB
  - Each item have a rating decided by the users
  - **Extra feature** (do it when you have time): each item can have multiple images up to 5. Do not store images in database
  - Create a Digram of Database before building it. specify the parameters types and the relations between the tables

- **API**
  - The user can login, signup, and edit his/her data including password reset functionality.
    - The guest user can see the items for sale
    - The registered users can buy items or add/remove them from their favorite list
    - The worker can add, remove, modify items
    - The admin can remove/ add users
    - Each user type can use the functionality of the previous users types
  - The password must be at least 12 character long, contain small letters , capital letters, numbers , symbols and should not contain the email or the username
  - Use JWT for authentication, where user needs to send his username and password to get token to access API.
  - User will use generated tokens for accessing other endpoints in the api.

  - Item APIs:
    - List of items: can be filtered by category, name, gender, and price
    - use pagination when you want to get list of items
    - List of best seller items : can be filtered by gender, category and price
    - Update item API
    - Add item API
    - Delete item API
    - Add item to fav list
    - Remove item from fav list
    - Send credit credentials API
    - **How we can handle the quantity for item with color and size**

- Management command **Extra feature**:

you need to write a management command that reads a json file that contains items and add these items to DB and print the items that caused an error

Notes :

- Use PostgreSQL database.
- Create APIs urls before building them
- No need to build UI just build the APIS
- In the API, use Serializes: do not use it to create items
- use class-based views.
- Try to make the structure of the project files (views, serializers …etc) organized to make tracking and reading the code easy.
- Use postman to test the API, no need to write scripts to test the API.
- *Create a Github account, and create a repository, and push your code consistently to it to we you can track and your code and get back to any version of it if something goes wrong*

Good Luck