### **Project: Django Sneakers**

#### **Store App Endpoints**

1. **Product Management**
   * **List Products**
     + **Endpoint:** /products/
     + **Method:** GET
     + **Description:** Returns a list of all available products with pagination, filtering, and sorting options.
     + **Tests status:**
   * **Product Details**
     + **Endpoint:** /products/<product\_id>/
     + **Method:** GET
     + **Description:** Returns details of a specific product identified by product\_id.
     + **Test status:**
   * **Search Products**
     + **Endpoint:** /products/search/
     + **Method:** GET
     + **Description:** Allows users to search for products by name, description, category, or other attributes.
   * **Product Reviews & Ratings**
     + **Endpoint:** /products/<product\_id>/reviews/
     + **Method:** GET / POST
     + **Description:** Retrieves user reviews for a product or allows users to submit a review and rating.
   * **Product Categories**
     + **Endpoint:** /categories/
     + **Method:** GET
     + **Description:** Returns a list of product categories, with possible subcategories.
2. **Cart Management**
   * **Add to Cart**
     + **Endpoint:** /cart/add/
     + **Method:** POST
     + **Description:** Adds a product to the user's cart. Requires product ID and quantity in the request body.
   * **View Cart**
     + **Endpoint:** /cart/
     + **Method:** GET
     + **Description:** Returns the current state of the user's cart, including products, quantities, and total cost.
   * **Edit Cart**
     + **Endpoint:** /cart/edit/
     + **Method:** PUT
     + **Description:** Updates quantities or removes items in the cart. It requires the product ID and the new quantity in the request body.
     + **Code:** Done
     + **Test status:**
   * **Remove from Cart**
     + **Endpoint:** /cart/remove/
     + **Method:** DELETE
     + **Description:** Removes a product from the cart. Requires product ID in the request body.
     + **Test status:**
   * **Clear Cart**
     + **Endpoint:** /cart/clear/
     + **Method:** DELETE
     + **Description:** Clears all items from the user's cart.
3. **Checkout & Payment**
   * **Checkout**
     + **Endpoint:** /checkout/
     + **Method:** POST
     + **Description:** Processes the checkout for the current cart, including shipping and payment options.
   * **Payment**
     + **Endpoint:** /payment/
     + **Method:** POST
     + **Description:** Processes the payment for the current order. Handles payment gateway integration and returns the payment status.
     + **Gateway 🡪 ZarinPal.ir or Pay.ir**
   * **Apply Coupon**
     + **Endpoint:** /checkout/apply-coupon/
     + **Method:** POST
     + **Description:** Applies a discount coupon to the current order.
4. **Order Management**
   * **Create Order**
     + **Endpoint:** /orders/
     + **Method:** POST
     + **Description:** Create a new order after checkout is completed.
   * **List Orders**
     + **Endpoint:** /orders/
     + **Method:** GET
     + **Description:** Returns a list of the user's past orders with order status.
   * **Order Details**
     + **Endpoint:** /orders/<order\_id>/
     + **Method:** GET
     + **Description:** Returns details of a specific order identified by order\_id.
   * **Order Tracking**
     + **Endpoint:** /orders/<order\_id>/tracking/
     + **Method:** GET
     + **Description:** Provides tracking information for a specific order, including shipping status and delivery ETA.
   * **Cancel Order**
     + **Endpoint:** /orders/<order\_id>/cancel/
     + **Method:** POST
     + **Description:** Cancels an order if it's not yet shipped.
5. **User Account Management**
   * **User Profile**
     + **Endpoint:** /auth/profile/
     + **Method:** GET / PUT
     + **Description:** Retrieves and updates the user's profile information.
   * **User Address Management**
     + **Endpoint:** /auth/addresses/
     + **Method:** GET / POST / PUT / DELETE
     + **Description:** Manages user's saved addresses for shipping and billing.
6. **Wishlist Management**
   * **View Wishlist**
     + **Endpoint:** /wishlist/
     + **Method:** GET
     + **Description:** Returns the user's wishlist.
   * **Add to Wishlist**
     + **Endpoint:** /wishlist/add/
     + **Method:** POST
     + **Description:** Adds a product to the user's wishlist.
   * **Remove from Wishlist**
     + **Endpoint:** /wishlist/remove/
     + **Method:** DELETE
     + **Description:** Removes a product from the user's wishlist.
7. **Notifications**
   * **Email Notifications**
     + **Endpoint:** /notifications/email/
     + **Method:** POST
     + **Description:** Sends email notifications for order confirmations, shipping updates, etc.
   * **Push Notifications**
     + **Endpoint:** /notifications/push/
     + **Method:** POST
     + **Description:** Sends push notifications to users about important updates or promotions.

#### **Authentication App Endpoints**

1. **User Registration**
   * **Register**
     + **Endpoint:** /auth/register/
     + **Method:** POST
     + **Description:** Registers a new user.
   * **Activation**
     + **Endpoint:** /auth/activate/<uid>/<token>/
     + **Method:** GET
     + **Description:** Activates a user's account.
   * **Resend Activation**
     + **Endpoint:** /auth/activate/resend/
     + **Method:** POST
     + **Description:** Resend the account activation email.
2. **User Authentication**
   * **Login (JWT)**
     + **Endpoint:** /auth/login/
     + **Method:** POST
     + **Description:** Authenticates a user and returns a JWT token.
   * **Logout**
     + **Endpoint:** /auth/logout/
     + **Method:** POST
     + **Description:** Logs out the user and invalidates the token.
   * **Password Reset Request**
     + **Endpoint:** /auth/password-reset/
     + **Method:** POST
     + **Description:** Initiates a password reset by sending an email with a reset link.
   * **Password Reset**
     + **Endpoint:** /auth/password-reset/<uid>/<token>/
     + **Method:** POST
     + **Description:** Resets the user's password using the reset link.
3. **Two-factor authentication (2FA)**
   * **Enable 2FA**
     + **Endpoint:** /auth/2fa/enable/
     + **Method:** POST
     + **Description:** Enables two-factor authentication for the user.
   * **Disable 2FA**
     + **Endpoint:** /auth/2fa/disable/
     + **Method:** POST
     + **Description:** Disables two-factor authentication.
   * **2FA Verification**
     + **Endpoint:** /auth/2fa/verify/
     + **Method:** POST
     + **Description:** Verifies the 2FA code during login.
4. **Social Media Login**
   * **Social Login**
     + **Endpoint:** /auth/social-login/
     + **Method:** POST
     + **Description:** Allows users to log in using social media accounts (e.g., Google, Facebook).