1. App Initialization & Configuration

- app = Flask(__name__): Initializes the Flask app.
- CORS(app): Enables Cross-Origin Resource Sharing, allowing requests from different origins.
- app.config[...]: Sets up email configurations (server, port, username, password) for sending alerts.
- mail = Mail(app): Initializes the Flask-Mail extension with the app for email functionality.

2. Scheduled Price Drop Alert

- **Function**: schedule_price_drop_alert()
- **Purpose**: Queries users from the database and sends price drop alerts every 180 seconds. If a price drop occurs, an alert is sent to the user's email.
- **Scheduler**: BackgroundScheduler triggers schedule_price_drop_alert at defined intervals, which is stopped when the app exits using atexit.register(lambda: scheduler.shutdown()).

3. Root Route

- **Route**: /
- **Purpose**: Serves as a health check, returning "Scheduler is running!" when the app is up.

4. User Postings

- **Route**: /postings/<username> (GET)
- **Purpose**: Retrieves all product postings by a user and checks if the user exists. If the user does, it fetches products they posted, returning details such as name, date, description, and price.

5. Price Extraction

- **Function**: extract_price(price_str)
- **Purpose**: Extracts the numeric price from a string using regex.

6. Item Search API

- **Route**: /api/search (GET)
- **Purpose**: Searches items across multiple e-commerce sites (e.g., Walmart, eBay, Target, Best Buy).
- **Implementation**: Uses site-specific scrapers like scrape_walmart, scrape_ebay, and filters results within a specified price range.

7. Web Scrapers

- **Functions**:
    - scrape_walmart(): Fetches item details from Walmart.

    - scrape_target(): Fetches item details from Target.
    - scrape_ebay(): Fetches item details from eBay.
    - scrape_bestbuy(): Fetches item details from Best Buy.
- **Purpose**: Each function sends a GET request to the respective website's search page and uses BeautifulSoup to parse product details such as title, price, and link.

8. Wishlist Management

- **Route**: /api/wishlist (POST)
    - **Purpose**: Adds an item to the user's wishlist, saving it if it doesn't already exist.
- **Route**: /api/wishlist/<product_id> (DELETE)
    - **Purpose**: Removes an item from the user's wishlist based on product_id.
- **Route**: /api/wishlist/<username> (GET)
    - **Purpose**: Retrieves a user's wishlist, displaying details like title, price, link, and image URL.

9. Posting Management

- **Route**: /add-posting (POST)
- **Purpose**: Adds a new product posting by a user, requiring product details and username. If the user exists, it stores the posting in the database.

10. Email Alert

- **Function**: send_email(email, content)
- **Purpose**: Sends a price drop alert email to a user using Flask-Mail. The function attempts to send an email and logs any errors if they occur.

11. Price Drop Alert API

- **Route**: /api/price-drop-alert (POST)
- **Purpose**: Sends a price drop alert to a user if an item in their wishlist has decreased in price on eBay.

12. User Authentication

- **Route**: /login (POST)
    - **Purpose**: Logs a user in, verifying username and password with database records.
- **Route**: /register (POST)
    - **Purpose**: Registers a new user by saving their details and hashed password in the database.

13. Database Setup

- **Function**: models.Base.metadata.create_all(bind=engine)
- **Purpose**: Creates all defined tables in the database, ensuring the required schema is set up before the app runs.