Whering Backend Engineer - Take Home Test

Overview

Build a RESTful API for managing clothing items in a digital wardrobe using NestJS and TypeScript. This test should take approximately **1.5-2 hours** to complete.

Feel free to use any AI coding tools (GitHub Copilot, ChatGPT, Claude, etc.) that you would normally use in your workflow.

Business Context

We're building a digital wardrobe platform where users can catalog their clothing items. This test focuses on building well-structured API endpoints with proper request/response handling.

Requirements

Data Model

A **Clothing Item** has the following fields:

- id (string/UUID)
- category (enum: tops, bottoms, dresses, outerwear, shoes, accessories)
- colour (string)
- user_id (string)
- brand (string)
- size (string)
- image_url (string)
- purchase_date (Date)
- purchase_price (number)

API Endpoints

Implement the following operations:

- `POST /items` - Create a new clothing item
- `GET /items` - List all items
- `GET /items/:id` - Get a single item's details
- `PATCH /items/:id` - Update an item
- `DELETE /items/:id` - Delete an item

Technical Guidelines

- Use NestJS with TypeScript
- Store data in-memory (no database required)
- Make architectural and technical decisions that you would make for production code

Time Expectation

1.5-2 hours - Focus on delivering quality over quantity. We're interested in seeing how you approach building a well-structured API within a reasonable timeframe.

Sample Request/Response

POST /items

json

```JSON
// Request
{
  "category": "tops",
  "colour": "blue",
  "user_id": "user-123",
  "brand": "Brooks Brothers",
  "size": "M",
  "image_url": "https://example.com/shirt.jpg",
  "purchase_date": "2024-01-15",
  "purchase_price": 89.99
}

// Response (201 Created)
{
  "id": "550e8400-e29b-41d4-a716-446655440000",
  "category": "tops",
  "colour": "blue",
  "user_id": "user-123",
  "brand": "Brooks Brothers",
  "size": "M",
  "image_url": "https://example.com/shirt.jpg",
  "purchase_date": "2024-01-15T00:00:00.000Z",
  "purchase_price": 89.99
}
```

GET /items

json

```JSON
// Response (200 OK)
{
  "data": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000",
      "category": "tops",
      "colour": "blue",
      "user_id": "user-123",
      "brand": "Brooks Brothers",
      "size": "M",
      "image_url": "https://example.com/shirt.jpg",
      "purchase_date": "2024-01-15T00:00:00.000Z",
      "purchase_price": 89.99
    }
  ],
  "count": 1
}
```

Deliverables

1. Please email us a link to a public **GitHub Repository** with your complete solution
2. The repository should also enclose a **README.md** that includes:
    - How to run your application
    - Any decisions or assumptions you made
    - Anything else you think we should know

Questions?

If anything is unclear, please reach out at **ben@whering.co.uk**.

What Happens Next?

After submitting, we'll review your code and discuss it further during your technical interview, we will focus on:

- Discussing your implementation and architecture choices
- Exploring how you'd extend the API
- Discussing how you'd add a database layer

- Talking about adding new features
- Reviewing authentication/authorization approaches
- Discussing scaling and production considerations