Assignment: E-commerce API Testing

You are tasked with testing an e-commerce API that provides information about products, user accounts, and shopping carts. The API documentation is as follows:

- Base URL: `https://localhost:8080`
- Authentication: The API uses JWT tokens for authentication. You need to obtain a token by sending a POST request to `/auth/login` with valid credentials (`username` and `password`) in the request body. The token will be returned in the response.

1. Authentication:

- Create a request to obtain an authentication token from the API.
- Use environment variables to store the token for later use in other requests.
- Write a test script to ensure that the token is obtained successfully, and it contains a valid JWT.

2. User Profile:

- Create a GET request to retrieve the user's profile information at `/user/profile`.
- Include the authentication token in the request headers.
- Write test scripts to verify that the response contains the user's information and has the correct status code.

3. Product Listing:

- Create a GET request to retrieve a list of available products at `/products`.
- Write a test script to ensure that the response contains a list of products and has the correct status code.
- Validate that each product in the list has essential information like `name`, `price`, and `description`.

4. Add to Cart:

- Create a POST request to add a product to the user's shopping cart at `/cart/add`.
- Include the authentication token in the request headers.
- Provide the necessary parameters in the request body, such as `product_id` and `quantity`.
- Write test scripts to verify that the product is added to the cart successfully and that the response contains the updated cart details.

5. Checkout:

- Create a POST request to checkout the items in the user's cart at `/cart/checkout`.
- Include the authentication token in the request headers.
- Write test scripts to ensure that the checkout process is successful and that the response contains an order confirmation.

6. Negative Testing:

- Create a request that deliberately sends incorrect authentication or invalid data to the API.
- Write test scripts to confirm that the API responds with the appropriate error status codes and error messages.

7. Collection and Environment:

- Organize all your requests into a Postman collection.
- Use environment variables to manage the token and any other dynamic data.
- Export your collection and environment as JSON files for sharing.