Linux Command Line Exercise: Set Up an E-commerce Project

Objective:

Learn how to use basic Linux commands to create a directory structure and essential files for an e-commerce project.

Step-by-Step Instructions:

1. Create the Main Project Directory

Objective: Create a main directory for your e-commerce project called 'ecommerce_project'.

2. Navigate into the Main Directory

Objective: Move into the 'ecommerce_project' directory.

3. Create Subdirectories for the Project

Objective: Create subdirectories that represent different components of the e-commerce project.

4. Navigate into the Frontend Directory and Create Files

Objective: Inside the 'frontend' directory, create HTML, CSS, and JavaScript files for the e-commerce site's homepage.

5. Navigate into the Backend Directory and Create Files

Objective: Inside the 'backend' directory, create files for API and business logic.

6. Navigate into the Database Directory and Create Files

Objective: Create SQL scripts or other database-related files.

7. Navigate into the Assets Directory and Create Subdirectories

Objective: Create subdirectories for storing images, CSS, and JS files.

8. Move a File into the Correct Directory

Objective: Move 'style.css' from the 'frontend' directory to the 'assets/css' directory.

9. List the Directory Structure

Objective: Verify the directory structure and files are created.

Expected Output:

You should see a directory tree similar to this:

```
ecommerce_project/
├── assets
│   ├── css
│   │   └── style.css
│   ├── images
│   └── js
├── backend
│   ├── api_routes.js
│   └── app.js
├── database
│   ├── create_tables.sql
│   └── seed_data.sql
└── frontend
    ├── app.js
    └── index.html
```

Bonus Challenge:

- Remove a file: Use the 'rm' command to remove any unnecessary file from the project.

- Create a compressed archive: Create a '.zip' of your 'ecommerce_project' directory

- Check file and directory sizes.