**Translation of the Assignment Description to English:**

**Course Name: Frontend Development - Server**

**Class**: Frontend Developer Web Security  
**Term**: Term 1, Fall 2024

**REST API (ONLINE MARKET APPLICATION)**

**INTRODUCTION**

**Background Description, Problem Statement, Scope, and Objectives**

**Background**:  
Our client has just launched an online shop for their new brand "Drezztore" and has entered into a supplier agreement with a Chinese warehouse. The client wants us to help them set up an API to simplify product management. In this task, you will create a REST server to apply what we have learned about APIs, server programming, and JSON.

**Objective**:  
A REST server that can handle **GET**, **POST**, **PUT**, and **DELETE** requests to manage products in an e-commerce application. The application should also have a route that renders an HTML page displaying all products.

**Scope**:  
You can implement this using any method (or package) you prefer.

**Why Perform This Task?**

**Purpose**:

* The student will gain an extended understanding of creating a REST application using Node.js and NPM.
* The student will work on problem-solving and designing JavaScript code following good practices.

**Deliverables**

1. A folder containing an NPM project (package.json, etc.).
   * Exclude the node\_modules folder from the project.
2. A written report on how to use the API (e.g., "How do I start the application? How do I test it?").
3. Oral presentation of the application, including comprehensive answers to questions from the course instructor.

**PROJECT ASSIGNMENT**

**What Should You Do?**

Create a **REST API** using Node.js and Express that interacts with a MongoDB database for an e-commerce application. The application does not need a frontend, but the root route (/) must render HTML. Write a report describing how to use the API.

**Steps to Solve the Task:**

1. Create a folder for your project (e.g., assignment) and place it on your desktop.
2. Open the terminal and navigate to the folder:
3. cd ~/Desktop/assignment
4. Run the following command to initialize an NPM project:
5. npm init
6. Install the following NPM packages:
   * cors, express, mongoose, jest, supertest, and any other packages you wish to use.
7. Add a .gitignore file in the folder with the following content:
8. node\_modules
9. Initialize a Git repository and link it to GitHub. Ensure the repository is **private** to prevent exposing sensitive data (e.g., MongoDB credentials). Follow [this tutorial](https://www.digitalocean.com/community/tutorials/how-to-push-an-existing-project-to-github) to push the project to GitHub.
10. Set up a basic REST API and ensure that the root route (/) responds with the message:
11. Welcome to Drezztore API

**API CRUD Operations to Implement:**

| **Method** | **URL** | **Action** |
| --- | --- | --- |
| GET | /api/products | Fetch all products |
| GET | /api/products/:id | Fetch product by ID |
| POST | /api/products | Create a new product |
| PUT | /api/products/:id | Update product by ID |
| DELETE | /api/products/:id | Delete product by ID |
| DELETE | /api/products | Delete all products |
| GET | /api/products?name=kw | Fetch products whose names include "kw" |
| GET | / | Returns an HTML page with a table of all products |

Each product must have the following properties:

* name (String)
* description (String)
* price (Number)
* quantity (Number)
* category (String)

**Testing the API**

Use **Postman** or **Insomnia** to test the API during development.

1. Write tests for all the routes in your REST API using **supertest** and **jest**.

**SUBMISSION AND PRESENTATION**

**Submission:**

Submit your project by **8th December** at 11:59 PM via the LearnPoint platform.  
Deliver the following:

1. A zipped folder containing all project files.
2. A written report explaining how to use the API.

**Assessment Criteria:**

**Pass**:

* Demonstrates knowledge and skills in server programming with Node.js.
* Demonstrates understanding of structuring a REST API.
* Shows knowledge of backend development, including database integration.
* Demonstrates understanding of API testing.

**Distinction**:

* Uses MongoDB (in the **Cloud Atlas** environment) to store all products.
* Provides a report explaining REST API routes with examples of usage and nuanced reasoning for various outcomes (e.g., handling invalid and valid data in requests).

**Feedback:**

Oral feedback will be provided during the presentation, and written feedback will be available on LearnPoint by **22nd December**.

Let me know if you'd like assistance with any specific part of this task! 😊