**Kitchen365 - Full Stack Developer Technical Assessment**

**1. Introduction**

Developers at Kitchen365 work on challenging, demanding, and exciting projects for a range of large enterprise clients. Because we have high standards for our developers, we focus on writing quality code that is easy to understand and maintain.

This technical assessment is designed to evaluate your proficiency in **Next.js, Nest.js, and PostgreSQL**, as well as your problem-solving skills and coding best practices.

**2. Recruitment Process**

The interview process for developers at Kitchen365 consists of multiple stages. You are currently in the **technical assessment stage**. If your submission meets our expectations, you will proceed to a technical interview with our senior engineering team, followed by an HR interview.

**3. Technical Assessment**

**3.1 Overview**

You are required to develop a **simple Product Catalog** application using **Next.js (Frontend), Nest.js (Backend), and PostgreSQL (Database)**. This will allow us to evaluate your skills in full-stack development, API design, database management, and UI implementation.

**3.2 Requirements**

***Backend (Nest.js + PostgreSQL)***

* Create a **Nest.js** backend with a Product entity containing the following fields:
* id (UUID, primary key)
* name (string, required)
* price (float, required)
* description (string, optional)
* Implement a REST API with the following endpoints:
* GET /products → Fetch all products.
* POST /products → Create a new product (validate input data).
* DELETE /products/:id → Delete a product by ID.
* Use **TypeORM** for database interactions.
* Database should be **PostgreSQL**, running locally via **Docker**.

***Frontend (Next.js + React)***

* Create a simple UI for listing and adding products:
* Display all products retrieved from GET /products.
* A form to add a new product using POST /products.
* Use **React Hooks** for state management.
* Handle form validation (e.g., price should be a positive number, name should not be empty).
* Basic UI styling (Tailwind CSS preferred, but not mandatory).

***Bonus (Optional, but Encouraged)***

* Implement API authentication using JWT (login required to add/delete products).
* Add a search/filter feature for products.

**3.3 Expectations**

* Code should be well-structured, modular, and follow best practices.
* Include basic error handling (e.g., return proper HTTP status codes).
* Provide clear instructions for running the application.

**3.4 Submitting Your Assessment**

* Submit your project with Public Git repo with clear readme.
* Submit it within **3 days** of receiving the test.
* Email your submission to [**careers@kitchen365.com**](mailto:careers@kitchen365.com) with the subject **"Full Stack Developer Technical Assessment Submission"**.

npx @nestjs/cli new productCatalogApi - 11.0.4

Get-ExecutionPolicy

Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass

Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser

https://medium.com/simform-engineering/nestjs-and-postgresql-a-crud-tutorial-32aa78778752

npm install @nestjs/config @nestjs/typeorm pg typeorm class-validator class-transformer bcrypt jsonwebtoken passport passport-jwt

npm install -D @types/jsonwebtoken @types/passport-jwt

admin

5432