

PruebaFullStack

Descripción

Este proyecto es una prueba para el cargo de desarrollador Full Stack en la empresa.

Solución

La solución se divide en backend, frontend, base de datos y despliegue.

Base de datos

Se utilizó PostgreSQL para la base de datos. Se crearon tablas para los usuarios, los productos y los datos de empleos por estados. Se realizó limpieza de los archivos CSV para la carga de los datos de empleos por estados. Se realizó la carga de datos por medio de archivos CSV con DBeaver. Las tablas fueron creadas con TypeORM por medio de las entidades de NestJS. Se utilizó el archivo "Employment by Location.csv" de Retail Trade USA para la carga de datos de empleos por estados. Se utilizó el archivo "Product.csv" de Adventure Works para la carga de datos de productos.

Backend

API

El backend es una API REST desarrollada en NestJS, la cual se conecta a una base de datos PostgreSQL. Se utilizó TypeORM para la conexión a la base de datos y para el manejo de las entidades. Se crearon entidades para los usuarios, los productos y los datos de empleos por estados. Se crearon controladores para los usuarios y los productos, y se crearon servicios para los usuarios, los productos y los datos de empleos por estados. Se creó servicio de autenticación para el manejo de los tokens de los usuarios y sus roles. Se utiliza JWT para la autenticación de los usuarios.

Frontend

Frontend

El frontend es una aplicación web desarrollada en NextJS, la cual se conecta a la API REST. Se utilizó TailwindCSS y daisyUi para el diseño de la aplicación. Se crearon páginas para el login, productos, detalles de productos, graficas y usuarios. Se almacenaron los tokens de los usuarios en las cookies del navegador. Se utiliza manejador de estados global para el manejo de los datos de los usuarios.

Despliegue

El despliegue de la aplicación se realizó en Render y Vercel. Se utilizó Render para el despliegue del backend y Vercel para el despliegue del frontend. Se configuró el despliegue continuo en ambos servicios para que se actualicen automáticamente cuando se realicen cambios en el repositorio de GitHub. Se creó un servicio de base de datos en Render para la base de datos PostgreSQL.

Tecnologías

- NextJS
- NestJS
- PostgreSQL
- Render
- Vercel

Instalación

Leer los archivos README.md de cada carpeta para la instalación de cada parte del proyecto.

Repositorio

<https://github.com/jblancoh/PruebaFullStack/>

Pagina desplegada

<https://prueba-full-stack.vercel.app/> <https://prueba-fullstack-nestjs.onrender.com>

```
usuario: jonathan@example.com  
password: 123123123
```