Thank you for applying for the Full Stack Software Developer position at Palitronica. As part of the application process, we require you to take the following coding challenge.

For this challenge, you will develop a simple application for calculating payments for customer orders. The application consists of three components:

1. A database containing customer and item information.
2. A backend service for payment calculation.
3. A single-page web application to take customer inputs and display payments.

Database

- The database stores customer information including a customer id and name.
- The database stores item information including item id and price.

Backend service

- For simplicity, the backend service seeds the database with hardcoded customer and item information when it first starts.
- The backend service exposes a REST API for payment calculation. The REST API takes the following inputs: customer id, list of item ids, and the quantity for each item. The API returns the following: customer name, total price for each item, the tax component, and the total price.
- The backend service uses the public API (https://developers.taxjar.com/api/reference) to retrieve the tax rate based on the customer's address (which is retrieved from the database). Use the "combined tax rate" to calculate the total taxes on all items (assuming that they are all taxable).
- For calling the public API:
    - Store any hardcoded customer address information in the database
    - The API token can be a configuration to your backend application

Single-page web application

- This is a simple web page that contains the following:
    - An input field for a customer id
    - Hardcoded item ids (matching ones in the database)
    - For each item, an input field for the quantity
    - A submit button that calls the REST API with the user inputs
- On the same page, display the output of the backend call (customer name, total prices for each item, total taxes, total price)

Code Submission

- Use a programming language of your choice
- Each application component is a separate docker container. Create the necessary docker files to containerize your services.
- A docker compose file should start all the services
- Add a ReadMe containing any instructions or notes for running your application
- Push your code to a public repo on your github account
- Send a link of your repo to recruiting@palitronica.com