Assignment

Build a REST API for an e-commerce marketplace.

You will be building a set of REST API endpoints that enable the following functionality

- **Buyers** and **sellers** can **register** and **login** to the system
- Sellers can build a catalog of **items**, with each item having a **name** and **price**
- Buyers can **GET** a list of sellers
- Buyers can **GET** a specific seller's **catalog** (list of items)
- Buyers can create an **Order** that contains a list of items from the seller's catalog
- Sellers can **GET** a list of all **orders** they've received

Entities

Following are the different entities in the system:

1. Users
    - Two types: **buyers** and **sellers**
    - A user can sign up as a buyer or as a seller
2. Catalogs
    - A catalog belongs to a **seller**
    - **One seller** can have **one catalog**
    - A catalog consists of **Products**

3. Products
    - A product has a **name** and a **price**
4. Orders
    - An order can be created by a **buyer** to purchase items from a **seller's** catalog
    - An order consists of a list of **products**

APIs

Following are a few examples of the API endpoints you should expose.

Auth APIs

POST /api/auth/register

- Register a user (accept username, password, type of user - buyer/seller)

POST /api/auth/login

- Let a previously registered user log in (e.g. retrieve authentication token)

APIs for buyers

GET /api/buyer/list-of-sellers

- Get a list of all sellers

GET /api/buyer/seller-catalog/:seller_id

- Get the catalog of a seller by seller_id

POST /api/buyer/create-order/:seller_id

- Send a list of items to create an order for seller with id = seller_id

APIs for sellers

POST /api/seller/create-catalog

- Send a list of items to create a catalog for a seller

GET /api/seller/orders

- Retrieve the list of orders received by a seller

Instructions

- Use **Node.js** along with any necessary libraries to complete the assignment
- You can store data in either a **SQL** or **NoSQL** database - your choice.
- Create a public git repository (e.g. on **GitHub**) and try to make frequent small commits as you work through the problem.
- Take as much or as little time as you need. There is no deadline, however most candidates respond within 1-3 days.
- Once you are done, **share the link** to the git repository. Ensure that you've also added any **instructions** needed for building and running the project.
- **No need to host** the live project on a server, simply provide instructions for running it on **localhost**
- No need to create any UI, only API endpoints are required
- If you find any instructions missing, feel free to use your best judgement and **fill in the blanks yourself** as to what the functionality of the APIs should be like.
- If there's anything you'd like to clarify, you can **reply to the email** directly

Our philosophy

- This assignment is designed to not only test your skills, but also to give you a taste of **what working at UnityLabs.ai is like**
- We do not believe in testing candidates for data structures or algorithms, but **only for real world applications**

Next steps

1. Once you've completed the assignment, **reply to the email** with a link to the git repository hosting the project
2. After that, we will schedule a **screen share** call with you to have a technical discussion around the project
3. This is the **final round**. Upon successfully completing the above, you will receive an offer from UnityLabs.ai within 24 hours

Happy coding!