

ES Tech Group – The test typically takes an average of 3 hours.

Usecase: You have a user logged into an ecommerce store. Only prices for their account should be shown to the user. The storefront makes a request to the pricing API (simulated by a JSON feed) passing the account and product id. If there is no match, you must find the match in the Database. If there is no match in the Database, then that product is not available for that user and no price should be returned.

Given the attached database and files

- import.csv - This file has references to products, accounts and users
- live\_prices.json - this file simulates an external service where you can read dynamic live prices in real time. A price is always related to a product, and it could optionally be related to an account.

Please complete what you can of the following tasks

- Write a script/function to import the prices provided in import.csv - this file has columns with references to each entity - into the prices table
- Read the JSON price file in real-time and do not load it into the database. This represents a “live pricing” feed.
- Develop an output for the function below. This can be done using JSON api, command line or UI based on your preference.
  - get a product price
    - required parameter: product code (can be one or many)
    - optional parameter: account id (one only, if null it is a public product price being requested)
    - Pricing logic
      - Pricing taken from the database should allow the lowest price to win for a product. A product price in the database with no account id is considered a public price and a match for all queries.
      - A product price with an account ID should match exactly and not be used for pricing without an account ID.
      - Any price in the JSON (live price) should win over any price in the database so long as it is a match for product and account (use null if no account is provided). If there is no match in the JSON pricing file use the database lowest matching price rule.
    - Output: returns a product sku and product price value in your preferred format.

You can use any technology you want to complete the test, but of course we'd like to see:

PHP

Laravel

MySQL Stored procedures

VueJS

We would love to see any type of UI, if you have time.