codeus

Učitelj Tasina 28/1
18000 Niš, Serbia
info@codeus.rs

PHP development task requirements

A web page is needed where a user will see a list of products available in the system and their prices with or without discount vouchers associated.

Consider the following entities:

- Products that have:
    - ID
    - Name
    - Price

- Vouchers that have:
    - ID
    - start date
    - end date

- Discount tiers (10%, 15%, 20% and 25% discounts applicable to product price)

A product can have one or more vouchers associated with it (or none). Each voucher, once introduced to the database, might be assigned to several products. Each voucher must be associated with one discount tier. The task is to produce REST API service that would allow:

- To create a voucher and associate it with a discount tier
- To add a product
- To add a voucher bind to a certain product
- To remove voucher bind from a certain product
- To buy a product

The goal is to produce a table of products with names and associated prices and a button (or a link or whatever control element is convenient) to buy a certain product. Prices in this table are calculated in the following way:

Price of the product * sum of discounts by all associated vouchers

Please note that the vouchers that are outdated (if today is earlier than start date or today is later than end date) should not be applied in the formula. Also, if the sum of all discounts is more than 60% then the summary discount is equal to 60% regardless of how many vouchers are associated with the product.

Clicking on “buying a product” should make the product and all associated vouchers unavailable via REST API and the web page.

You are completely free to design the database structure and project architecture at your will. Please note that aside from testing the project via web page you provide, the REST API will be also accessed multiple times separately from the web page by third-party tools in order to test the application with additional data.

Must have

1. We need to see your own code. No borrowed code is allowed.
2. Simple HTML on a web page is fine – there is no need to concentrate on UI at all
3. Back-end logic should be implemented with the use of a framework of your choice
5. Supported web browsers: Chrome (latest version) or Firefox (latest version)

Nice to have

1. Unit OR/AND integration tests.
2. Sort the product table after user clicks on a column name. Correct sorting should be provided for numbers and plain text.

Expected Deliverables

1. Source code.
2. Database migrations with pre-seeded data.
3. Sample request/response payload for REST API endpoints.
4. README file with clear instructions for project set up.