

Welcome!

Here is a skill test to see if you fit our Junior developer requirements.

Make sure to test your work and DO NOT plainly copy code from internet. Trust me, we'll know.

Prepare

Before doing the test, please make sure to go through these resources. If you don't understand some of the topics covered in those resources, please make sure to invest time in learning them. It will help you a lot when doing the Junior test

[HTML](#), [CSS](#) - Free, interactive and good for basics

[HTML](#), [JavaScript](#) - Free, interactive and even includes jQuery

PHP the right way <http://www.phptherightway.com/>

Here are links to videos that might also help you with the task:

- <https://www.linkedin.com/learning/css-essential-training-1>
- <https://www.linkedin.com/learning/css-essential-training-2>
- <https://www.linkedin.com/learning/css-essential-training-3>
- <https://www.linkedin.com/learning/responsive-css-workflow-with-sass-bourbon-and-susy>
- <https://www.linkedin.com/learning/jquery-essential-training-2>
- <https://www.linkedin.com/learning/javascript-essential-training-3>
- <https://www.linkedin.com/learning/php-design-patterns>
- <https://www.linkedin.com/learning/learning-composer-the-php-dependency-manager>
- <https://www.linkedin.com/learning/advanced-php>
- <https://www.linkedin.com/learning/php-tips-tricks-and-techniques>
- <https://www.linkedin.com/learning/code-clinic-php>
- <https://www.linkedin.com/learning/php-object-oriented-programming-2>

Assignments

Hello!

We are happy our company took your attention, however it is not the only thing we are expecting from you. We have created a trial for you, to prove your bravery.

General requirements:

PHP: ^7.0, plain classes, no frameworks, OOP approach **jQuery:** optional

jQuery-UI: prohibited

Bootstrap - optional

SASS - advantage

MySQL: ^5.6 - obligatory

PSR-1/2 - optional

Code should be shared as [bitbucket](#) repository that is shared with a user whose email is as follows: hr@scandiweb.com

Task description:

The expected outcome is 2 separate pages for:

- 1) product list
- 2) product add

1. Product list should list all existing product and details, like:

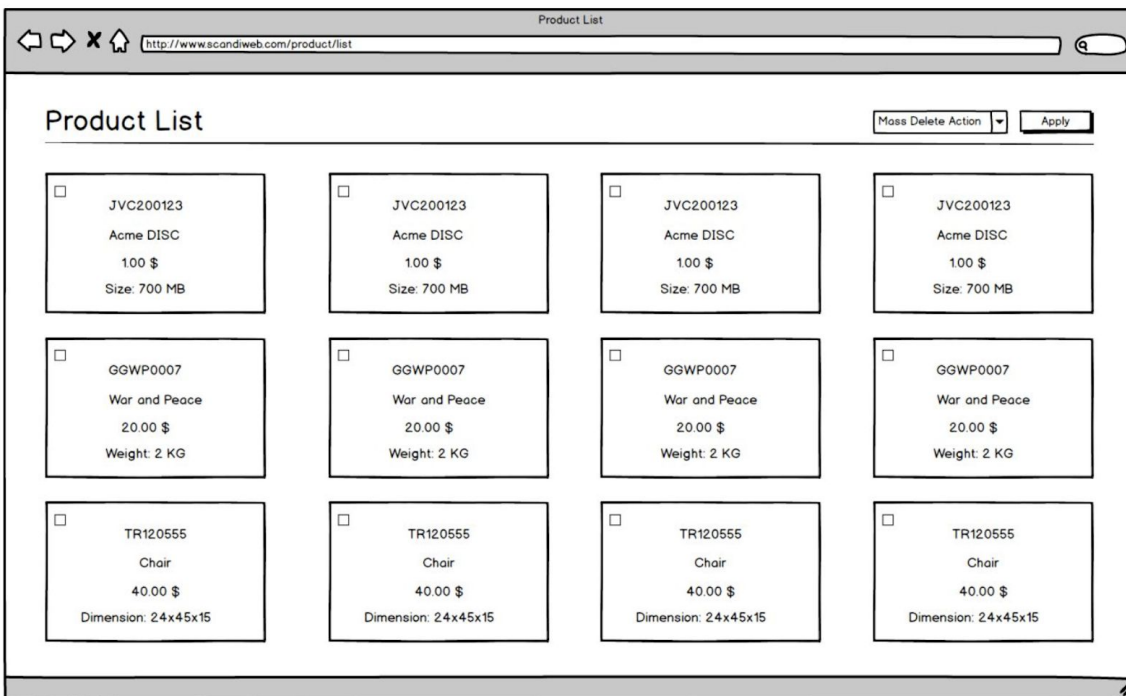
- **SKU (unique for each product)**
- **Name**
- **Price**

Also each product type has special attribute, which we expect you would be able to display as well (one of based on type):

- Size (in MB) for *DVD-disc*
- Weight (in Kg) for *Book*
- Dimensions (HxWxL) for *Furniture*

An advantage would be implementation of the optional feature: mass *delete* action, implemented as checkboxes next to each product.

Here is an example of product list page:



2. Product add page should display a form, with following fields

- SKU
- Name
- Price
- Type switcher (buttons for each type)
- Special attribute [please note: the form should be dynamically changed when type is switched]

Additional notes:

- Special attribute should have a description with helpful information, related to its type ex.: "Please provide dimensions in HxWxL format", when type: Furniture is selected.
- Utilize OOP principles to handle differences in type logic/behavior.
- Meeting PSR standards is an advantage (<https://www.php-fig.org>)

Additional requirements:

- All fields are mandatory for submission.
 - Avoid using conditional statements for handling differences in product types.
- An advantage* would be fields value (or format whether suitable) validation.

Here is an example of product add page:

Main point of this task is to test code standards, code commenting and understanding of OOP.