MAGENEST DEVELOPER ASSESSMENT TEST

Create a module which is able to manage vendors based on Magento 2’s existing customer system. Each vendor corresponds to a customer. A customer can request to be a vendor in the process of registering for a customer account on the system.

On the backend, the admin can do CRUD operations on the vendors, as well as other operations such as approve a vendor, assign a product to a vendor, …

Note: Question 1,2,3,4 and 9 are mandatory. Then you can choose either to do questions (5 and 6) or questions (7 and 8).

# 1) Module setup

Vendor name : Magenest

Module name : <name\_of\_developer>

Version : 1.0.0

Router name on backend : <name\_of\_developer> (lower case)

Router name on frontend : <name\_of\_developer> (lower case)

# 2) Setup database schema and install data

Create a database table by **InstallSchema.php** with name : **magenest\_test\_vendor\_(name of developer in lower case)**

|  |  |  |
| --- | --- | --- |
| Name | Type | Length |
| id | Int (primary key) | 11 |
| customer\_id | Int (required) | 11 |
| first\_name | text | 255 |
| last\_name | text | 255 |
| email | text (required) | 255 |
| company | text | null |
| phone\_number | text | 15 |
| fax | text | 20 |
| address | text | null |
| street | text | null |
| country | text | null |
| city | text | 50 |
| postcode | text | 20 |
| total\_sales | float | 11 |

Create a new customer attribute with **InstallSchema.php**

|  |  |  |
| --- | --- | --- |
| Name | Type | Length |
| <name of developer>\_is\_approved | int (not null) | 2 |

# 3) Create a model that can be used to access data from the above table

- Create model with path name on below:

+ <vendor\_name>/<module\_name>/Model/Vendor.php

+ <vendor\_name>/<module\_name>/Model/ResourceModel/Vendor.php

+ <vendor\_name>/<module\_name>/Model/ResourceModel/Vendor/Collection.php

# 4) Create a grid for admin to manage vendors.

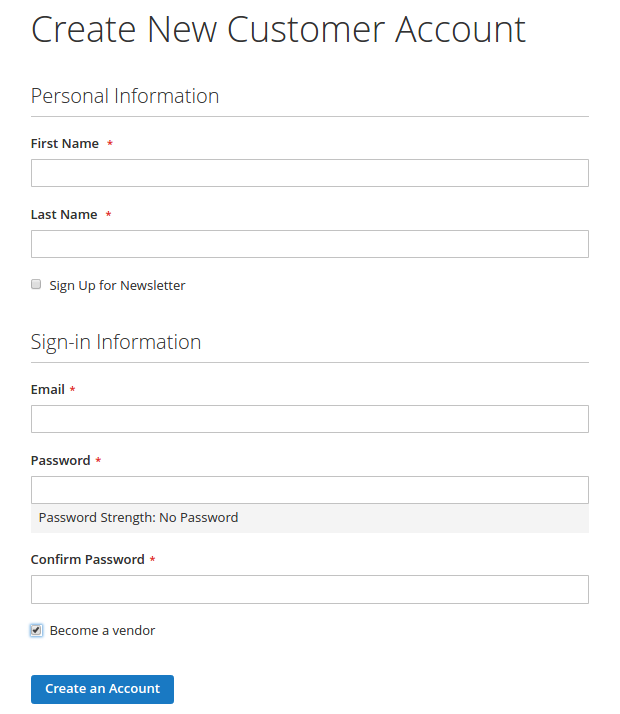
* The grid can be created using uicomponent or layout blocks.
* Admin can also edit a vendor’s information or add a new vendor with a form by clicking on a grid row’s edit button.

The following fields should be included in the form:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Field type** | **Data type** | **Note** |
| customer | Dropdown | Int (id of customers) (required) | List all available customers in the system in this dropdown |
| first\_name | Text input | text |  |
| last\_name | Text input | text |  |
| email | Text input | text (required) |  |
| company | Text input | text |  |
| phone\_number | Text input | text |  |
| fax | Text input | text |  |
| address | Text input | text |  |
| street | Text input | text |  |
| city | Text input | text |  |
| postcode | Text input | text |  |
| country | Dropdown | text | List all available countries in the system in this dropdown |

* Upon saving, you need to validate the vendor’s phone number (whether it contains alphabetical characters or not) and email address (whether its a valid email address).

# 5) Modify the layout of the customer registration form to add a new checkbox, as the following example:

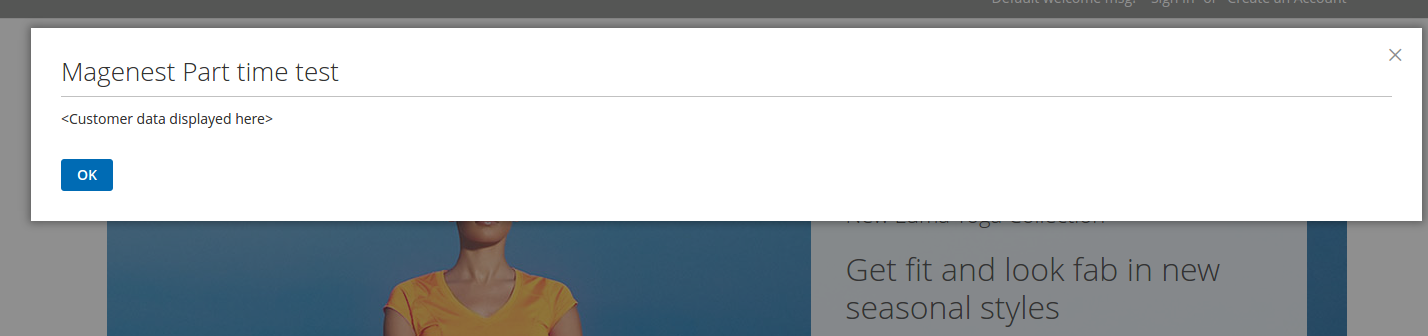


* Each time a customer registers for a new account:
  + Set the attribute of <name of developer>\_is\_approved of that customer to 0,
  + Add a new vendor with the information according to the customer’s information.

# 6) Create a new frontend page

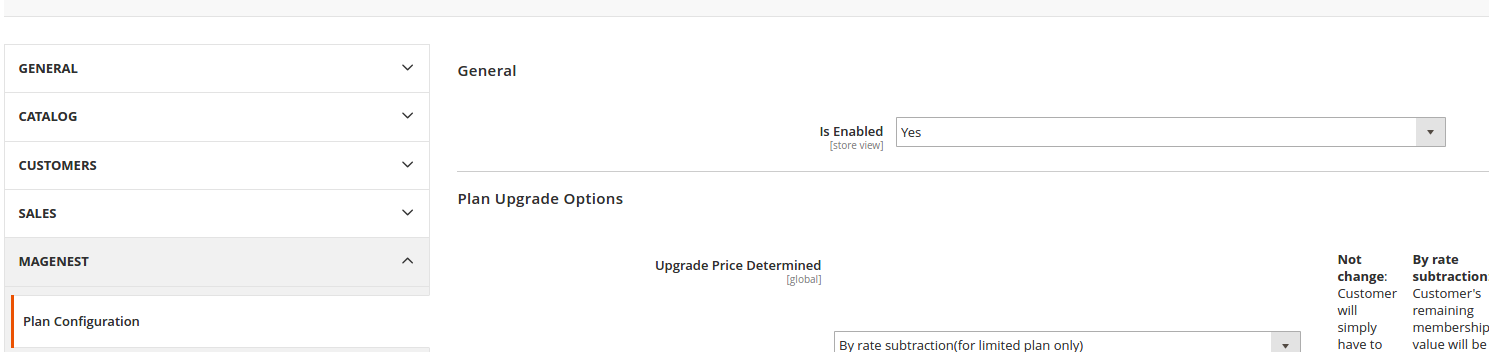
* Layout: 1column
* Content:
  + Create a text input field and a button
  + User input a customer’s id in text field and press the button
  + Send an AJAX request to a controller (you create) to retrieve a customer’s data (email, firstname, lastname) with the id inputted
  + Alert the customer’s ID, NAME, EMAIL using Magento ALERT MODAL (can be found in Magento\_Ui module)

- Result: (picture below is an example of Magento Alert Modal)



# 7) Backend Config

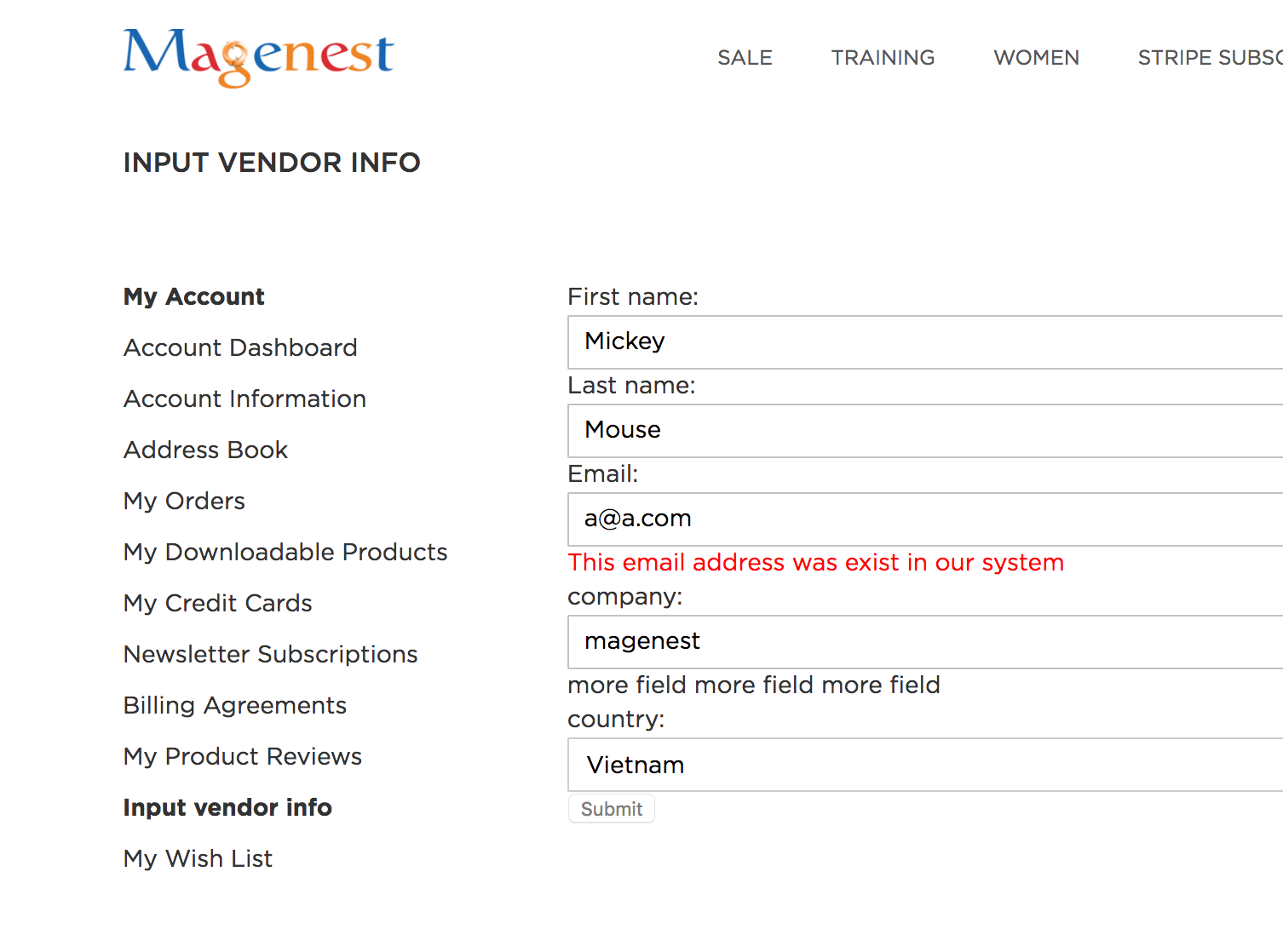
* Create a configuration in Configuration->Magenest->your\_name of type yes/no. You can refer to the image below.

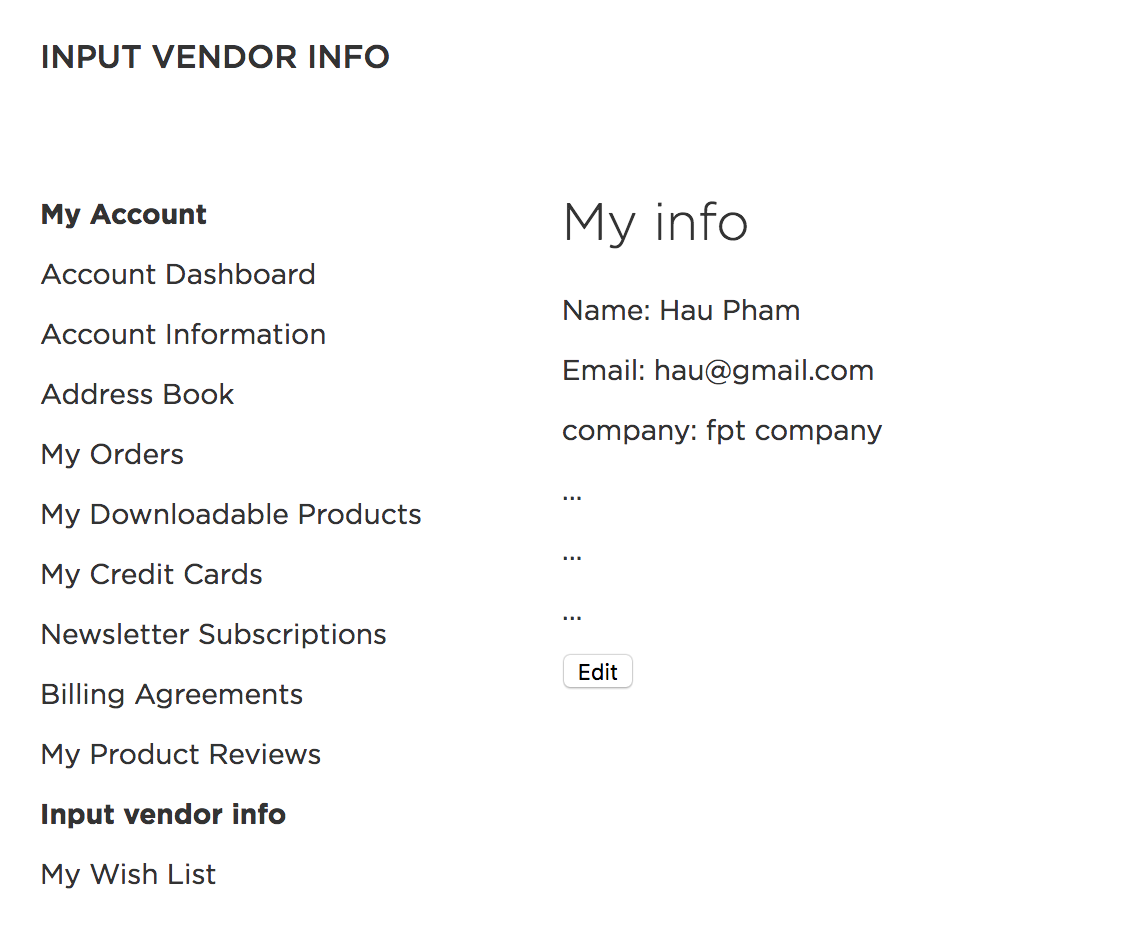


* When the admin save an admin user (You can look for admin users in Settings->Permissions->All users) check all available vendors in the database and delete all vendors that have the exact email address as the saved admin user

# 8) Vendor info

* Create a new vendor form in customer’s detail page (Customer->My Account) as image below and save to the database.
* If there exist a record in the database for that customer already then display that detail, along with an edit button to edit the information
* Require : Ajax Validation
  + If email exist in database, show error for customer, customer cannot submit form (submit button disabled). If email doesnt exist, show valid message and customer can submit form.
  + All require field. Email need to be email format, Phone number and postcode(if has) contain number only.





# 9. Event Observer

* Create a new product attribute <name\_of\_developer>\_product\_vendor.
* In product edit page, use a dropdown to allow admin to assign one of the available vendors to the product
* Each time a customer makes an order, if that order’s status is Processing then increases the total\_sales of the vendor assigned to the products in that order