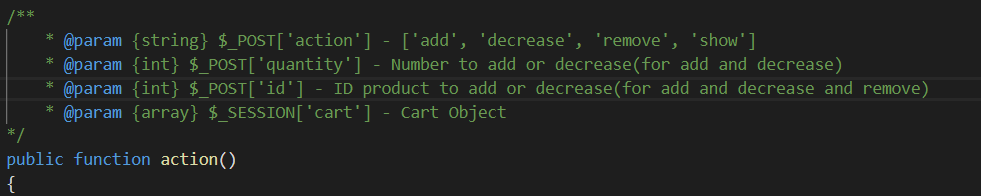
**TESTING**

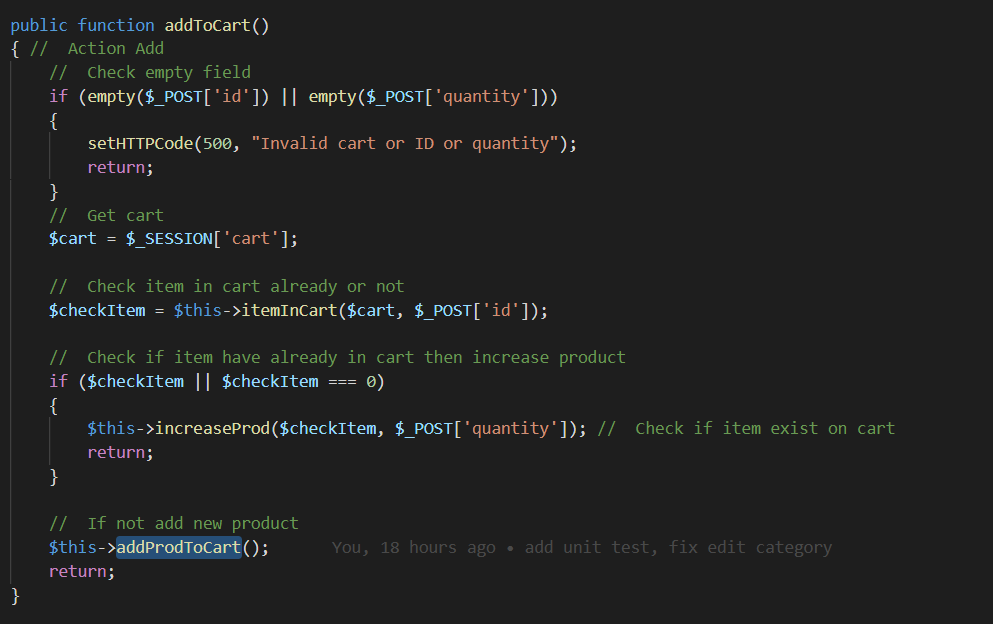
1.Add to cart function

- I will test the add to cart function, which includes adding an item to the cart, reducing the number of items in the cart by the quantity and removing the items from the cart.

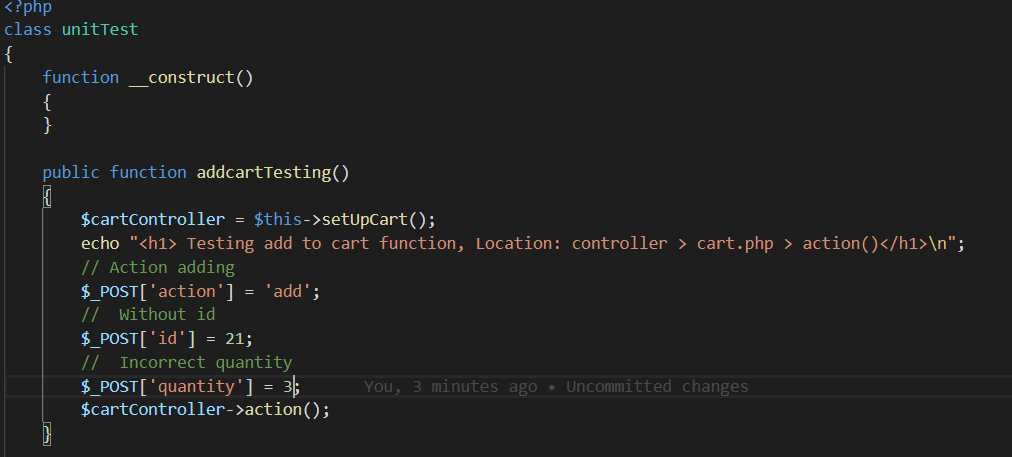
\* First I will testing add to cart function, this need the following parameters:

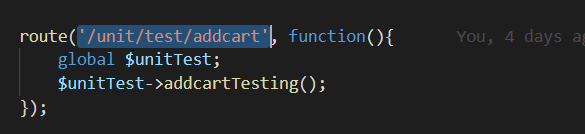


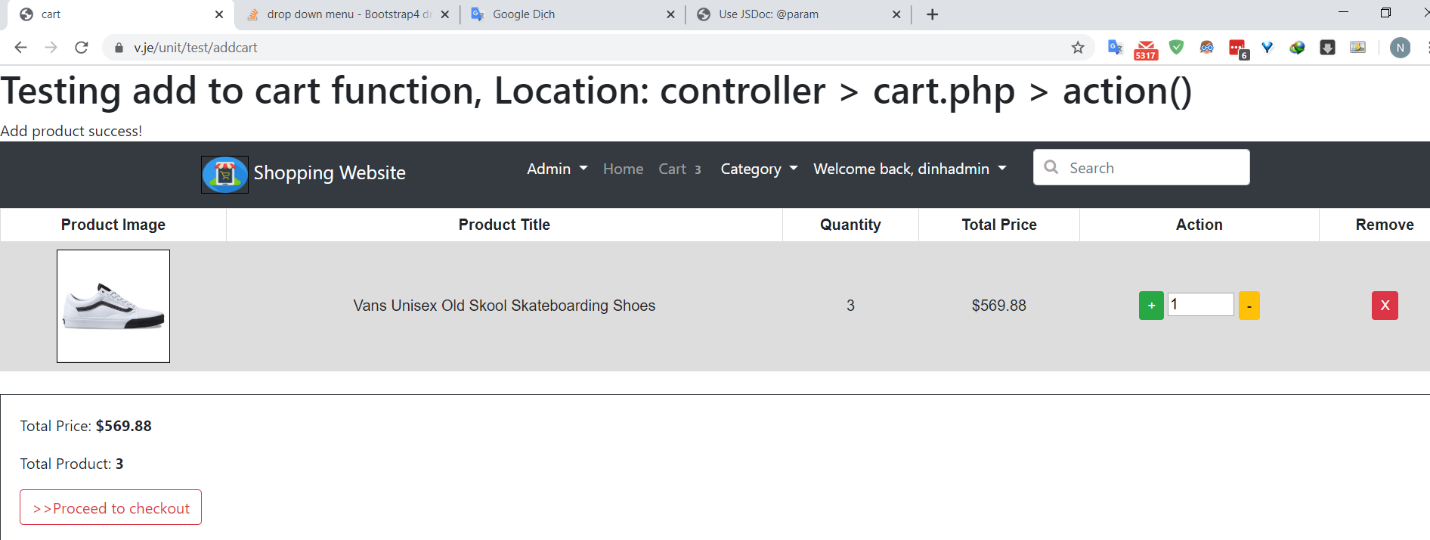
* Add to cart function will check if not have the ID of product to add or not have the quantity to add, it will show the error, otherwise it will add item to cart if the item doesn’t exist in the cart or will decrease the quantity of the item if it exist in the cart:



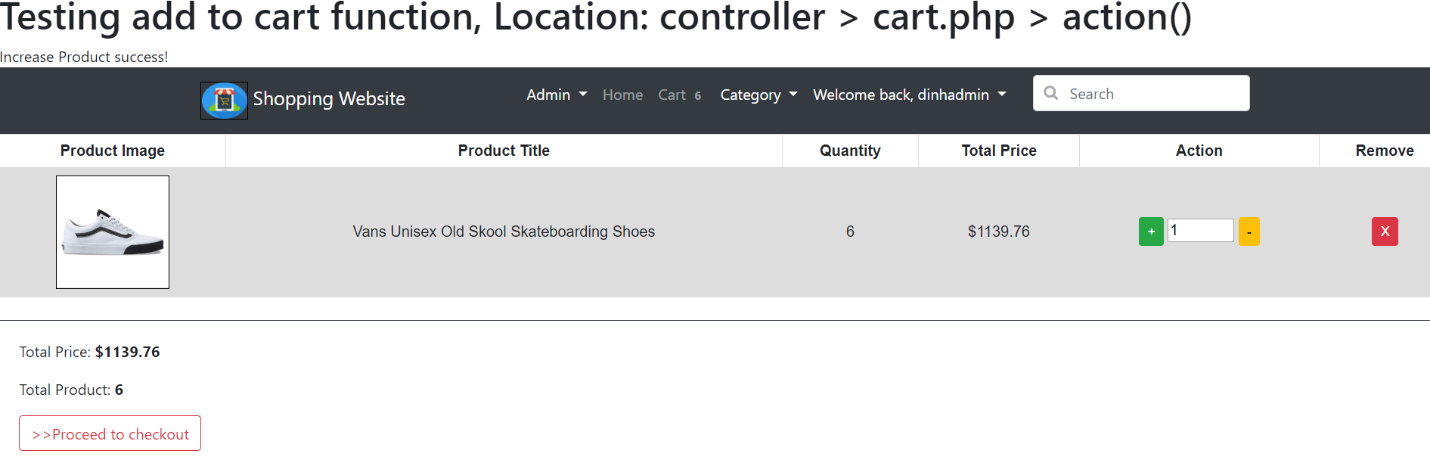
* Now I will write custom unit test to trigger this function and see what output is, here is the custom unit test:



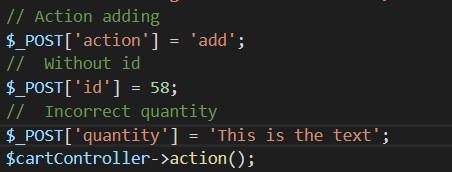
* The $\_SESSION[‘cart’] automacally create when I go to website, no need specific.
* To run this unit test, I go to url ‘/unit/test/addcart’ because I have specific the route to trigger it: 
* Now let see the output, the parameter I passed is:
* // Action adding
* $\_POST['action'] = 'add';
* //  Without id
* $\_POST['id'] = 58;
* //  Incorrect quantity
* $\_POST['quantity'] = 3;
* $cartController->action();
* Output:



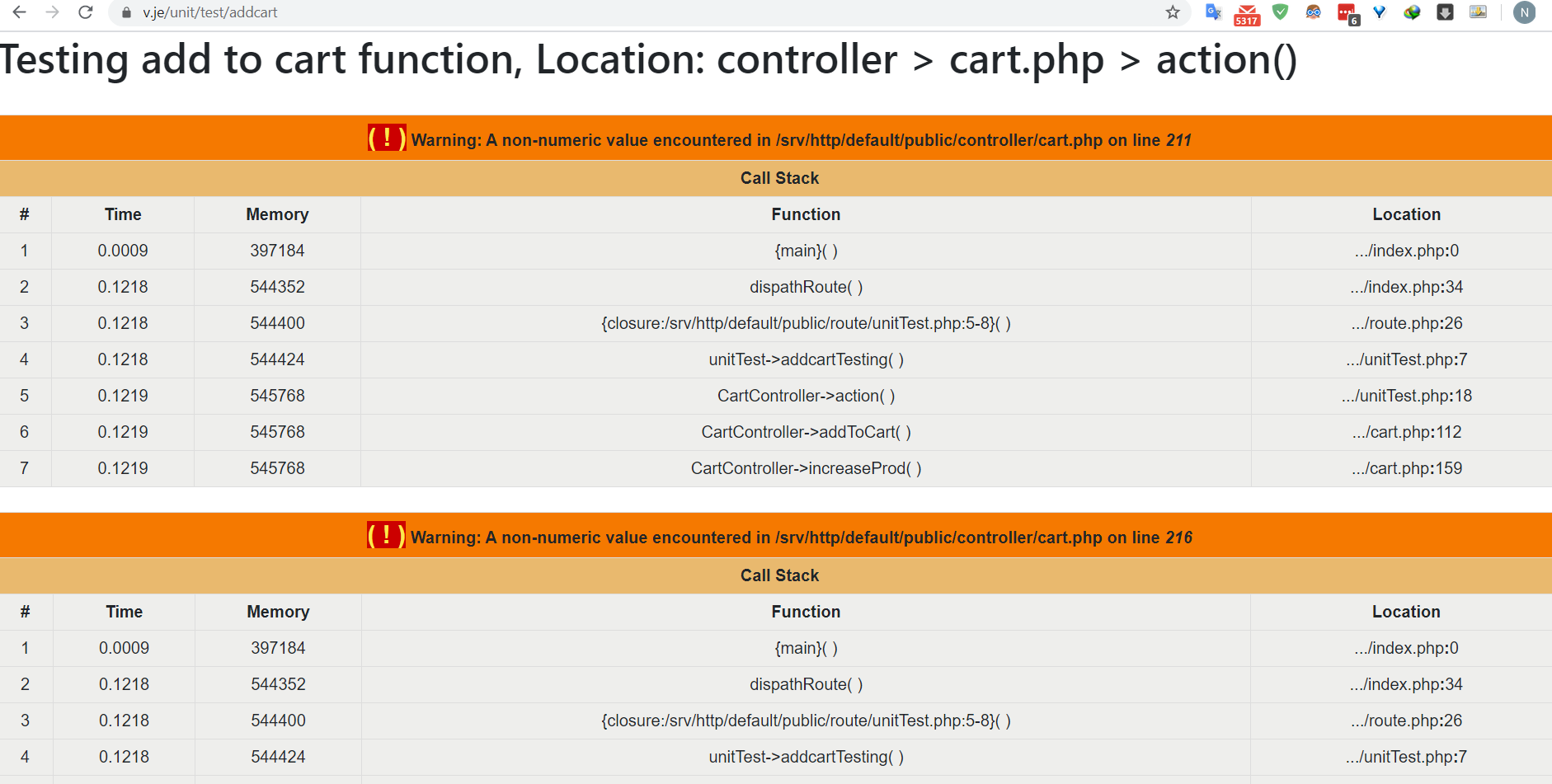
* The website said add product success, now I will re-testing again to check if whether the product is increase to 6(because I set the quantity is 3, current I have 3 in cart so add more 3 is 6) or something else, now testing again:



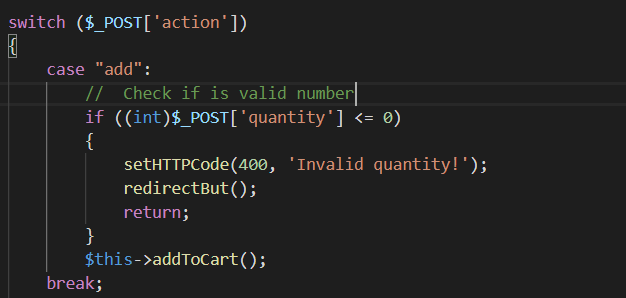
* The quantity is 6, the test successfull, now I will change the parameter, in this point if I submit the quantity which is not the number, but the text, I will change like this:



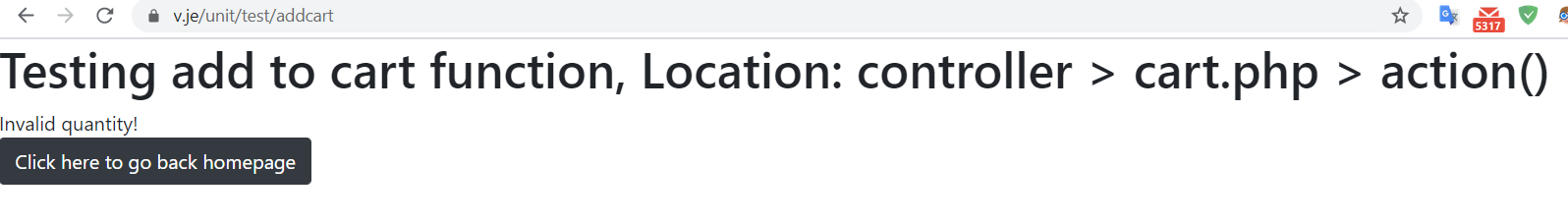
* Output:



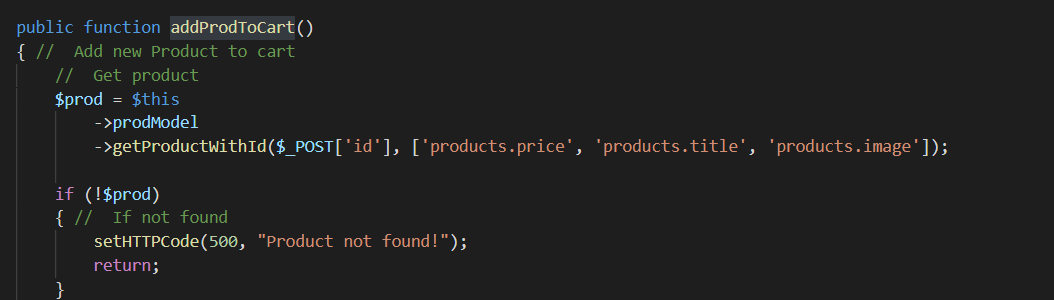
* Errors come as much as has never been seen, because I have plus the number with the string, so the error come, to fix this, I have to check if the $\_POST[‘quantity’] is valid, means it only an integer:



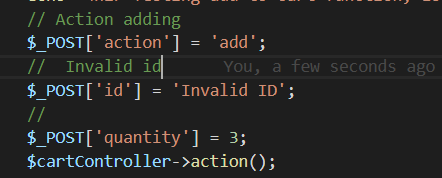
* Now try again, output:



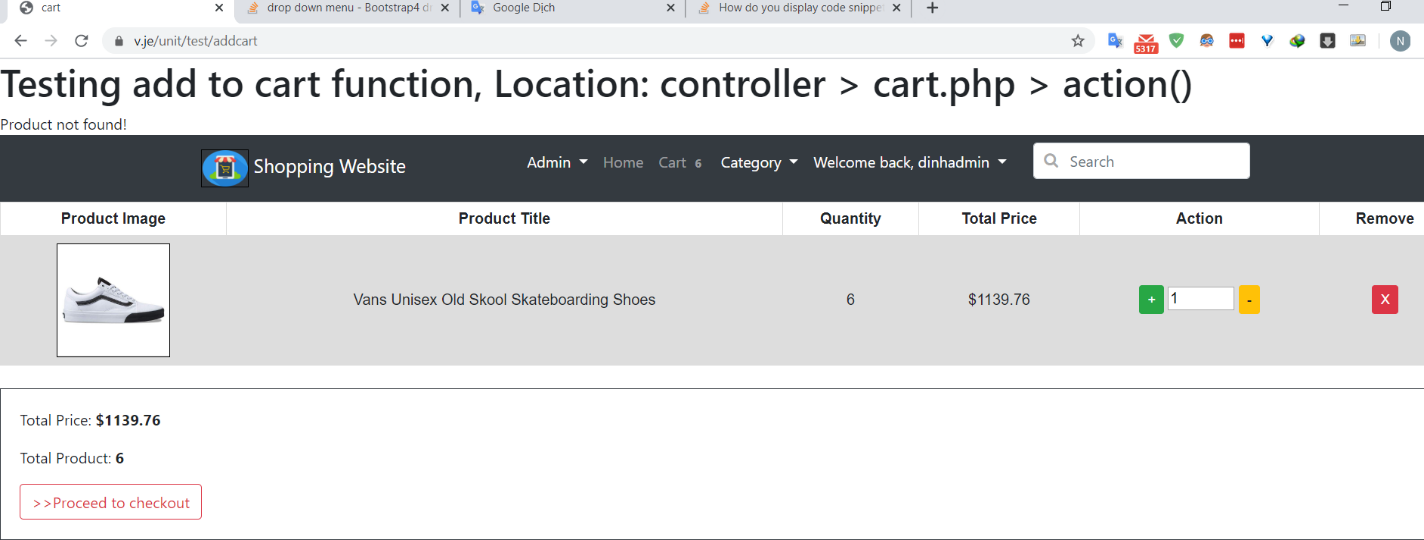
* The error has been fixed, the output like we expect, now I come to change the parameter again, but now is the ID of product, what happen if the ID is null or not correct, I have write solution for this, I will go check if the ID of product is not exist on our Database will return the error, null value also:



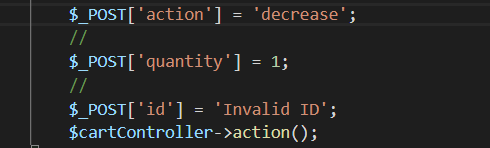
* Now let setup again unit test with invalid ID:



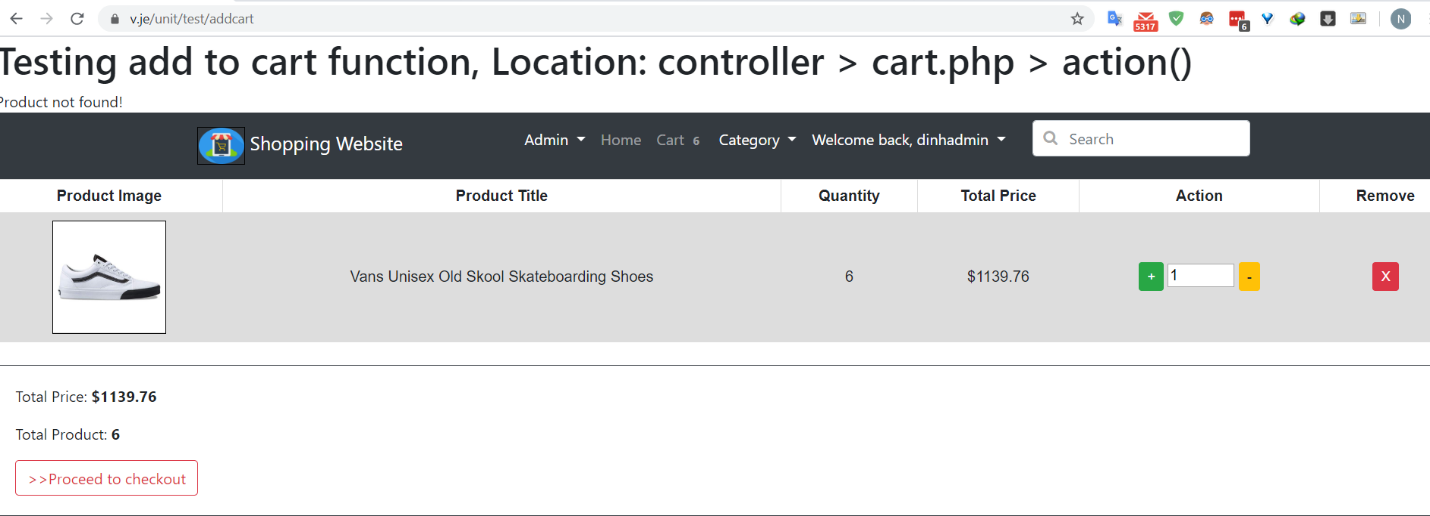
* The output:



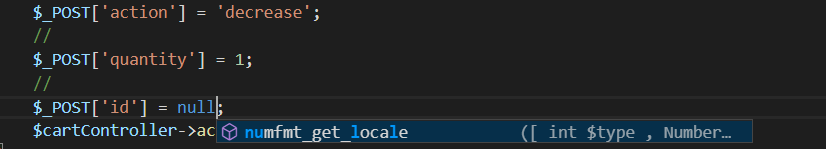
* The output said not found the product, the function work correctly as expect, the add function is finish, now come to decrease item, depend on the error I found in add to cart, I will check the valid input like $\_POST[‘id’] and $\_POST[‘quantity’] to make sure every thing is correct, now I will set up the unit test with invalid $\_POST[‘id’]:



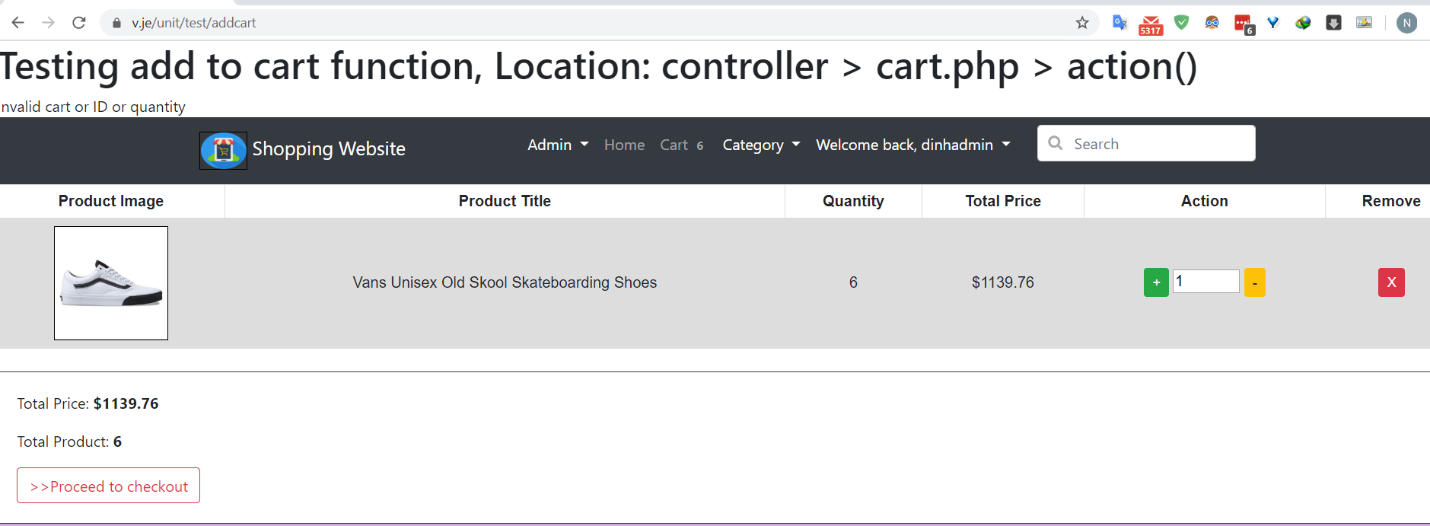
Output:



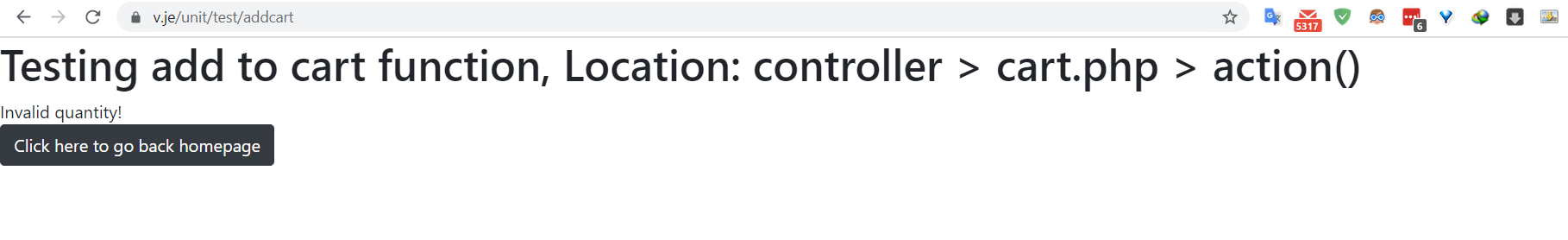
* Successfull prevent the error, now put NULL:



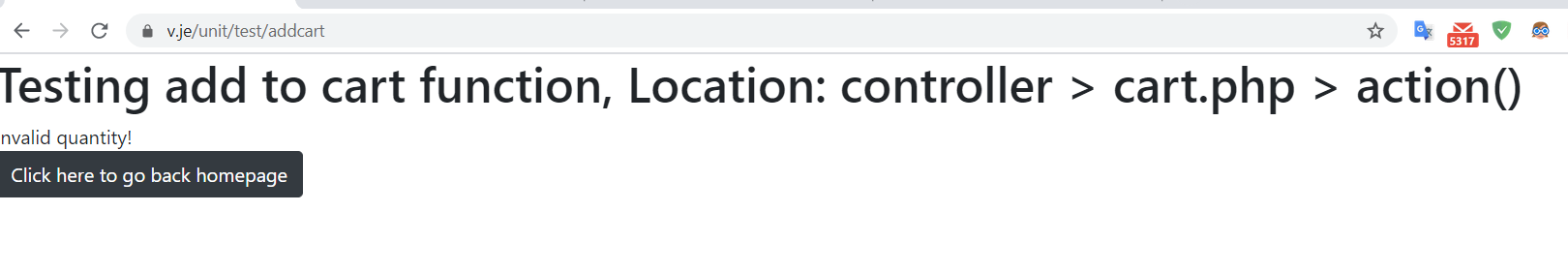
* Output



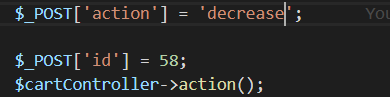
* Success with the output saying, next come with the $\_POST[‘quantity’] with invalid value and NULL value,
* With Invalid value:



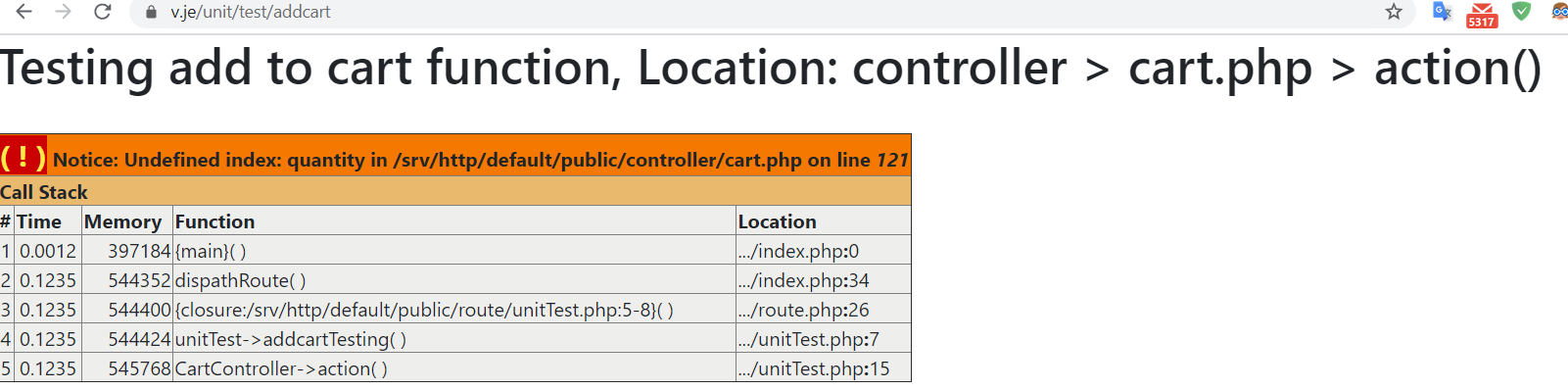
* With NULL value:



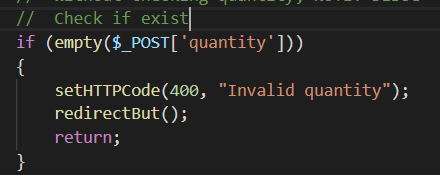
* Now let the empty $\_POST[‘quantity’], no create parameter to it:



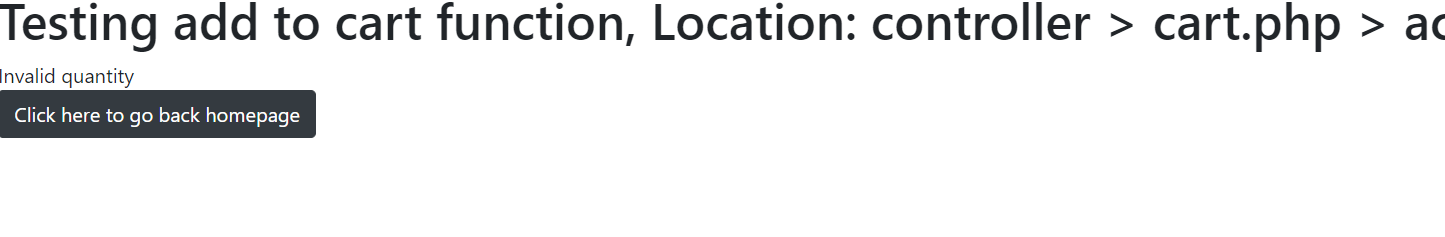
* Output:



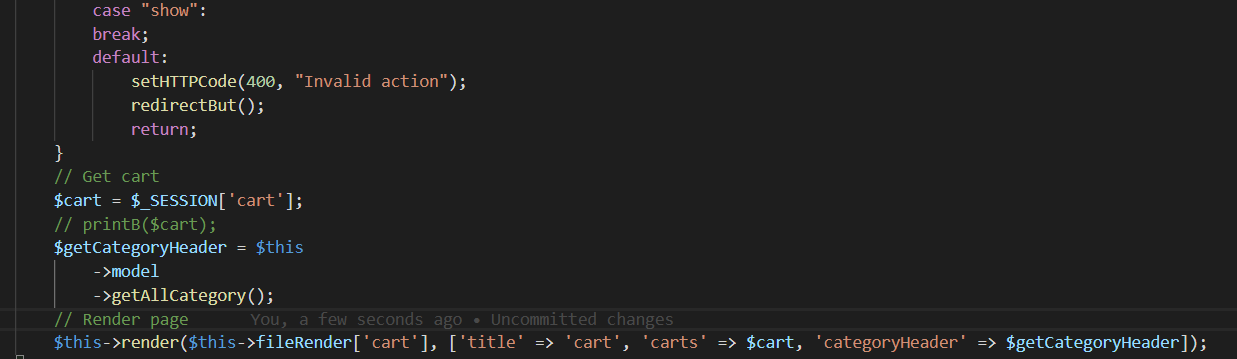
* The error said the $\_POST[‘quantity’] is not initialize but we still continue check $\_POST[‘quantity’] is valid or not so make the error , so I will check the $\_POST[‘quantity’] is exist or not before go check the valid of its:



* Now let see the output:



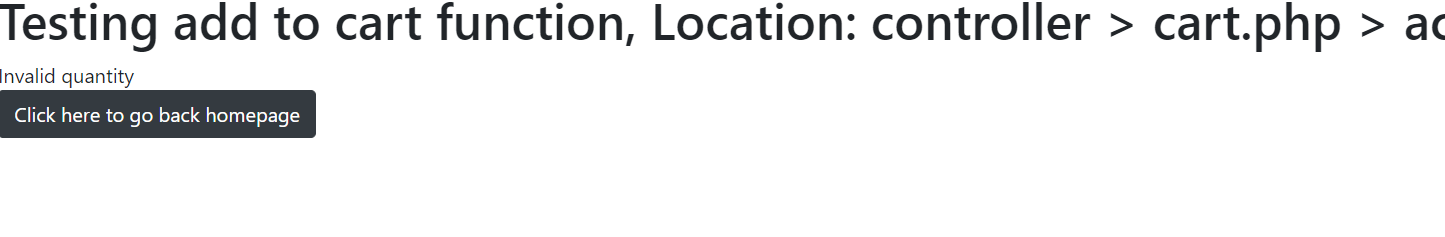
* The output show the error we make, was successfull fixed.
* Now come with action ‘show’, this will only show what we have in cart:



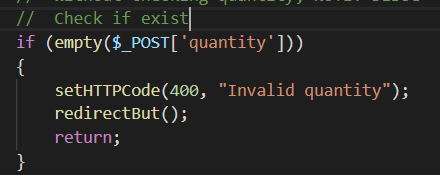
* Set up the unit test, because we only show the cart so no need to pass parameter like $\_POST[‘id’] or $\_POST[‘quantity’]:



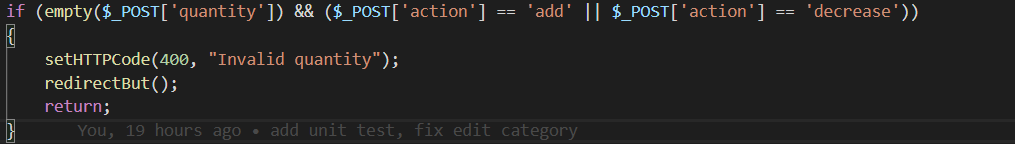
* Output:



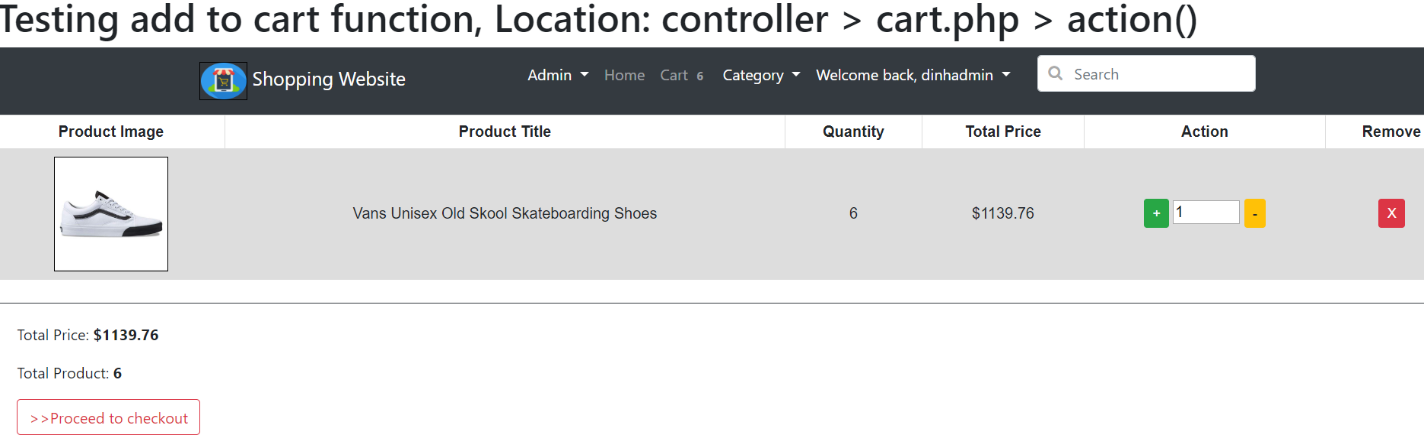
* Why so the error? I come to the code and firuge out, I have check the $\_POST[‘quantity’] if exist or not:



* Here action ‘show’ I didn’t create that, but we no need to create, so to make this work correctly, I need to edit some little about this If statement, only check if the action is ‘add’ or ‘decrease’:



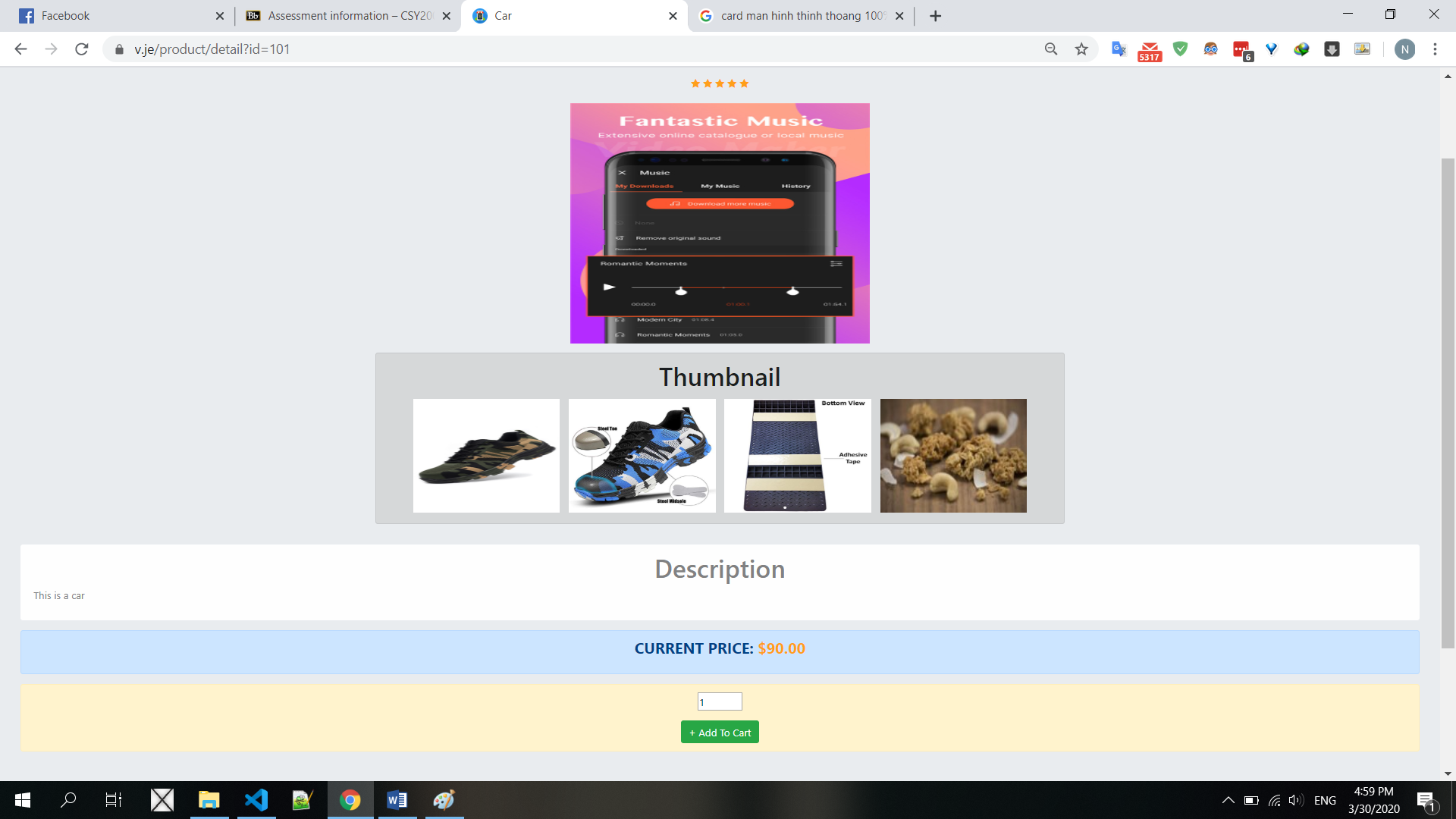
* Now try again:



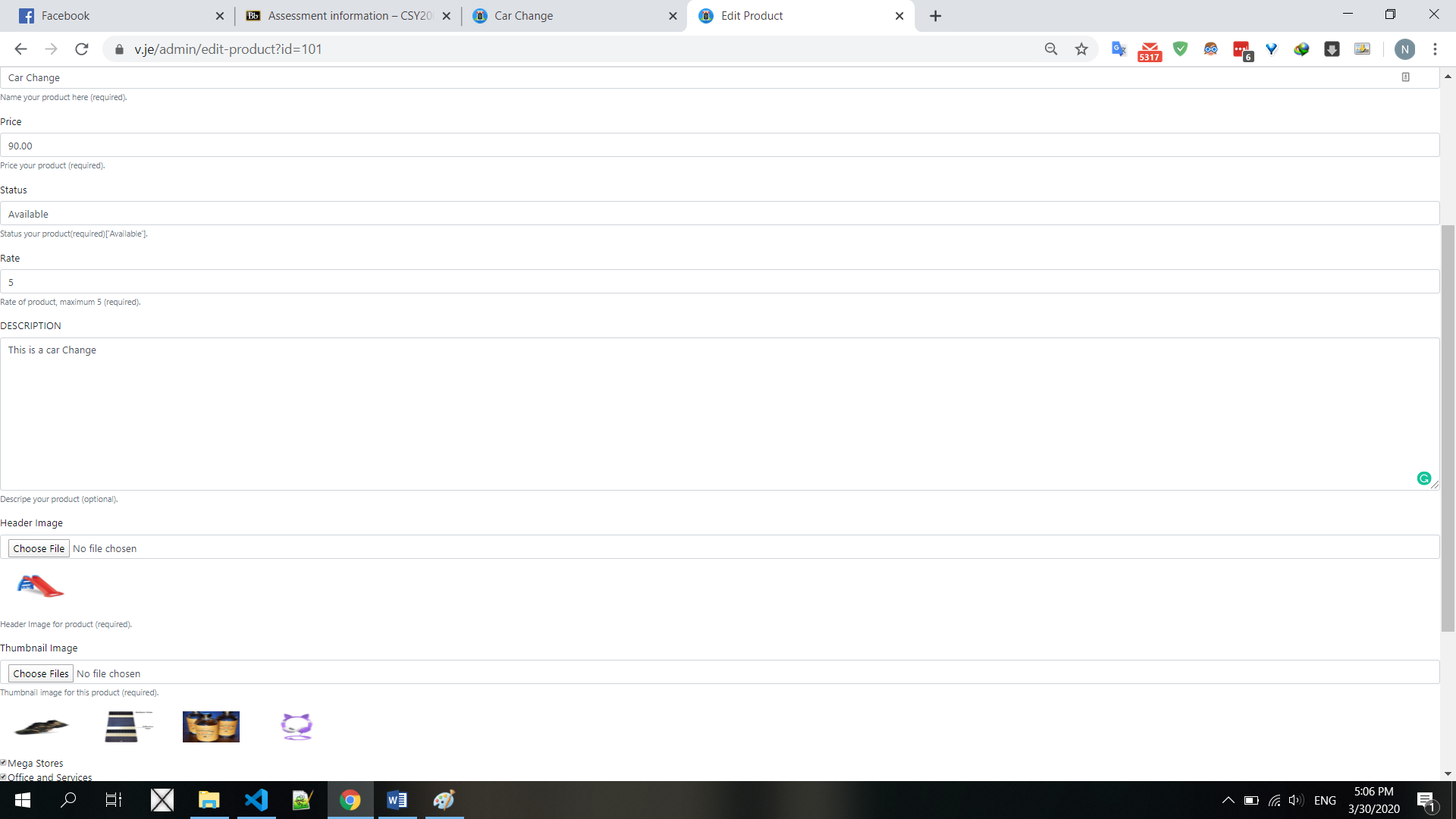
* The cart has been show, we have fixed the error, only check quantity if the action is ‘add’ or ‘decrease’, otherwise no check.

2.Edit product

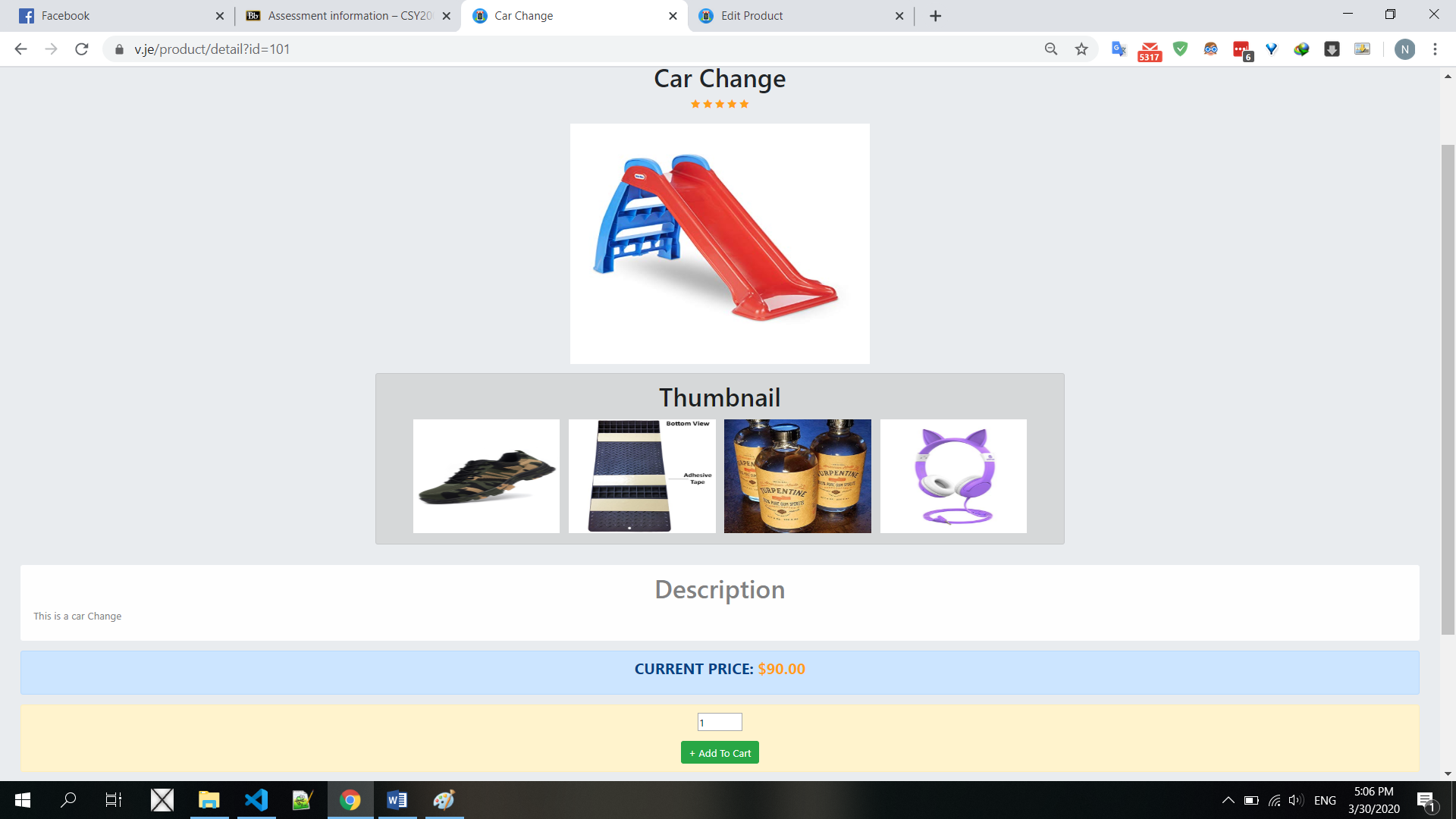
-Now I will come to test edit product functionality, this will update the information about product like title, price, description, image, thumbnail, … Now I will test change all the information of one product, the product like this:



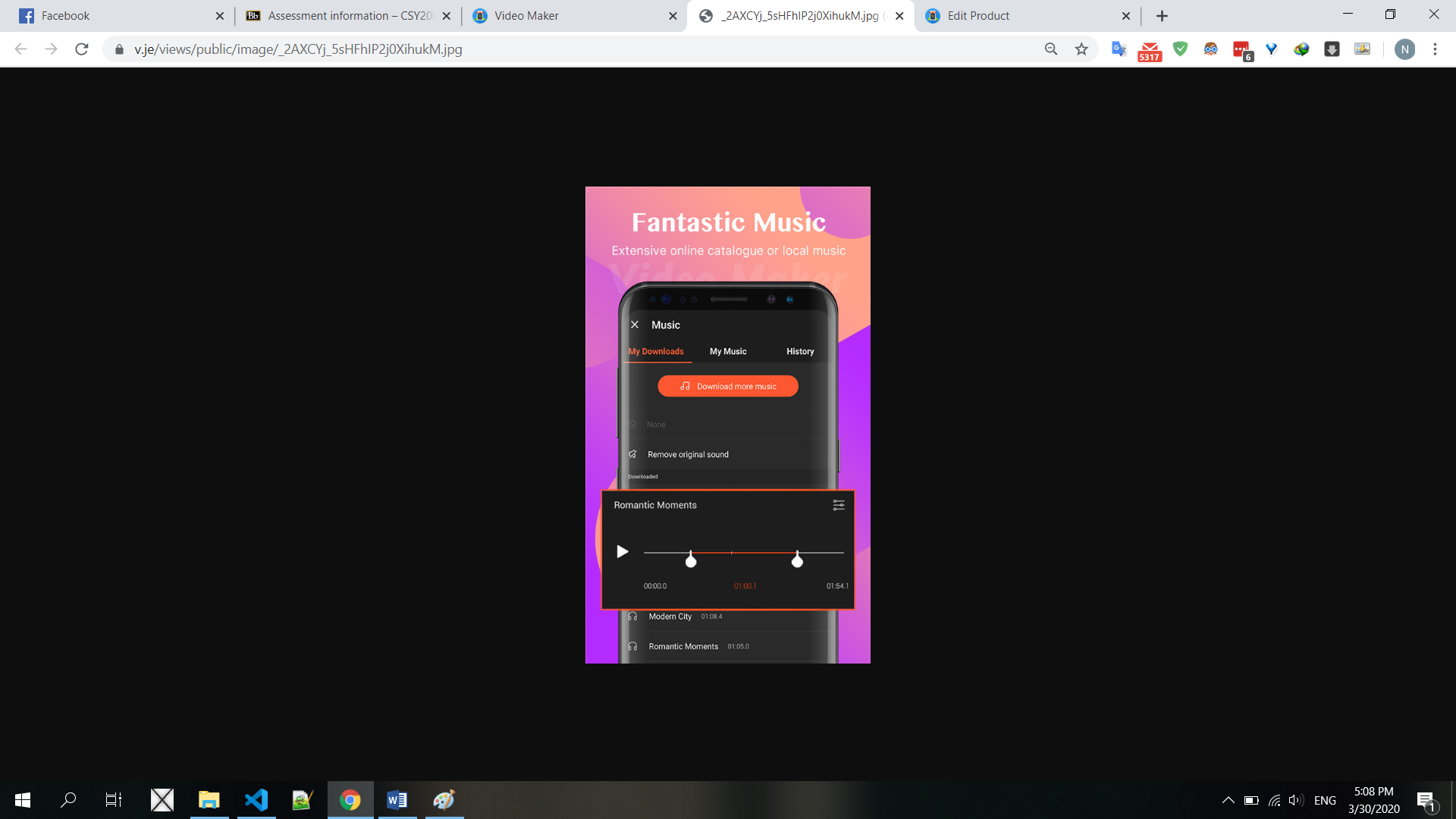
* I will change the detail of product such as image and thumbnail as follow:



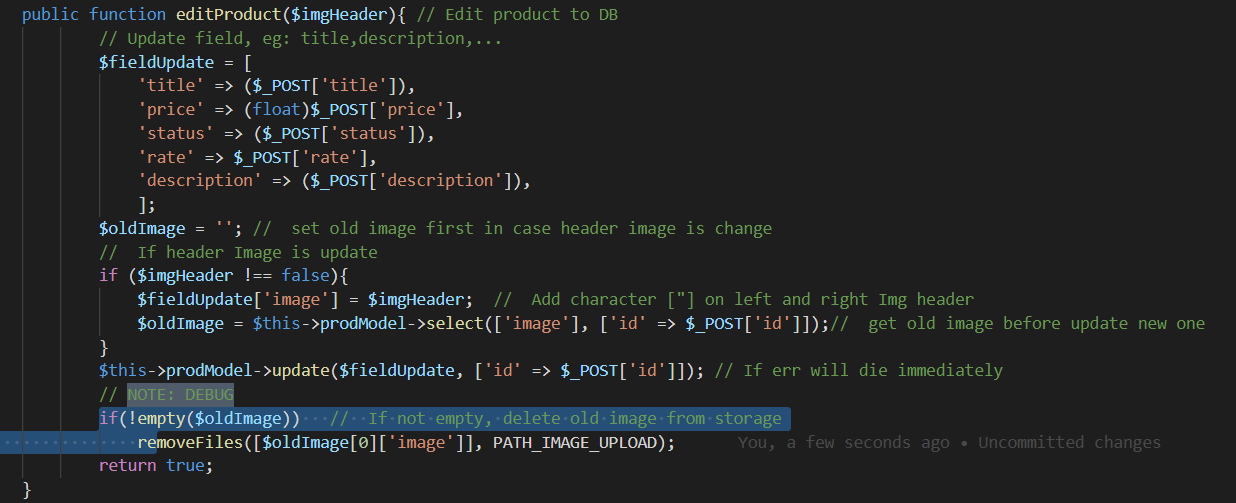
* And now the product will look like this:



* Edit product was successful, but I noticed that when I update an image, I just replace the name of Image to display, not delete it in actual storage, the od image still available on website:



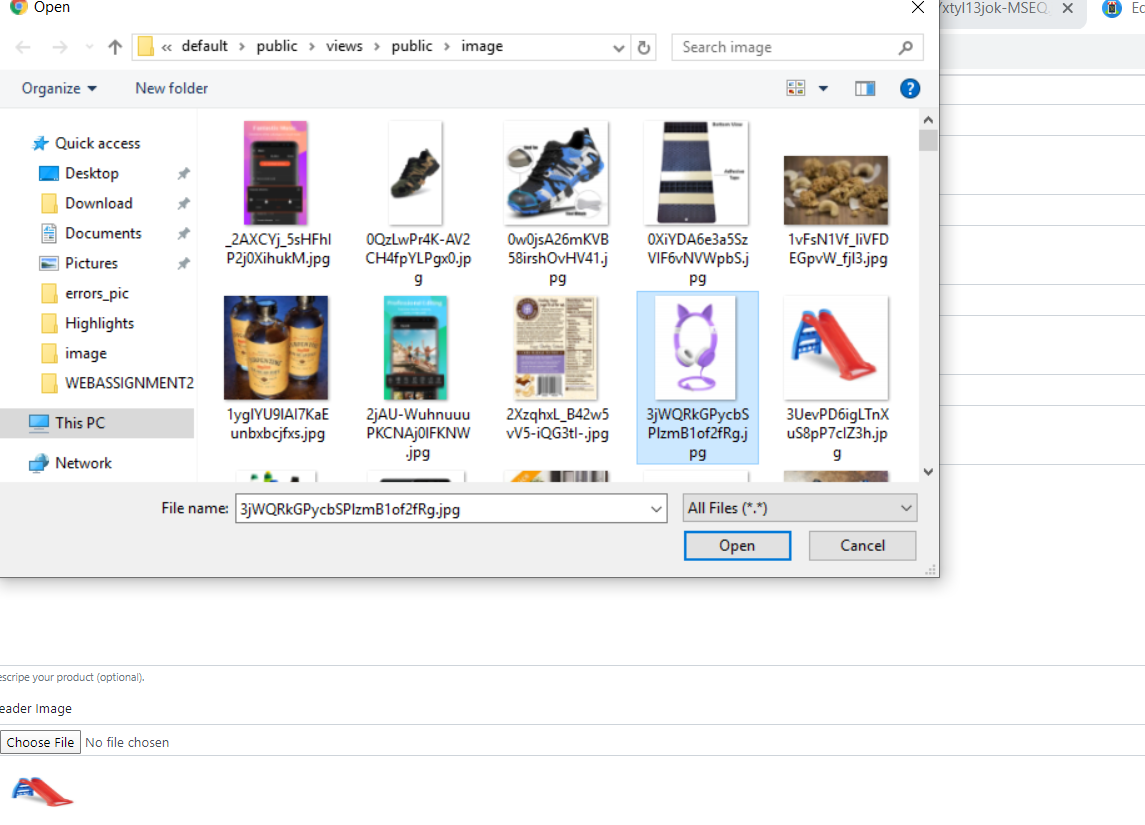
* To fix this, I have to delete the old image after update image for product, so I have to add the code get the old image first, then if have, remove it from storage after update success:



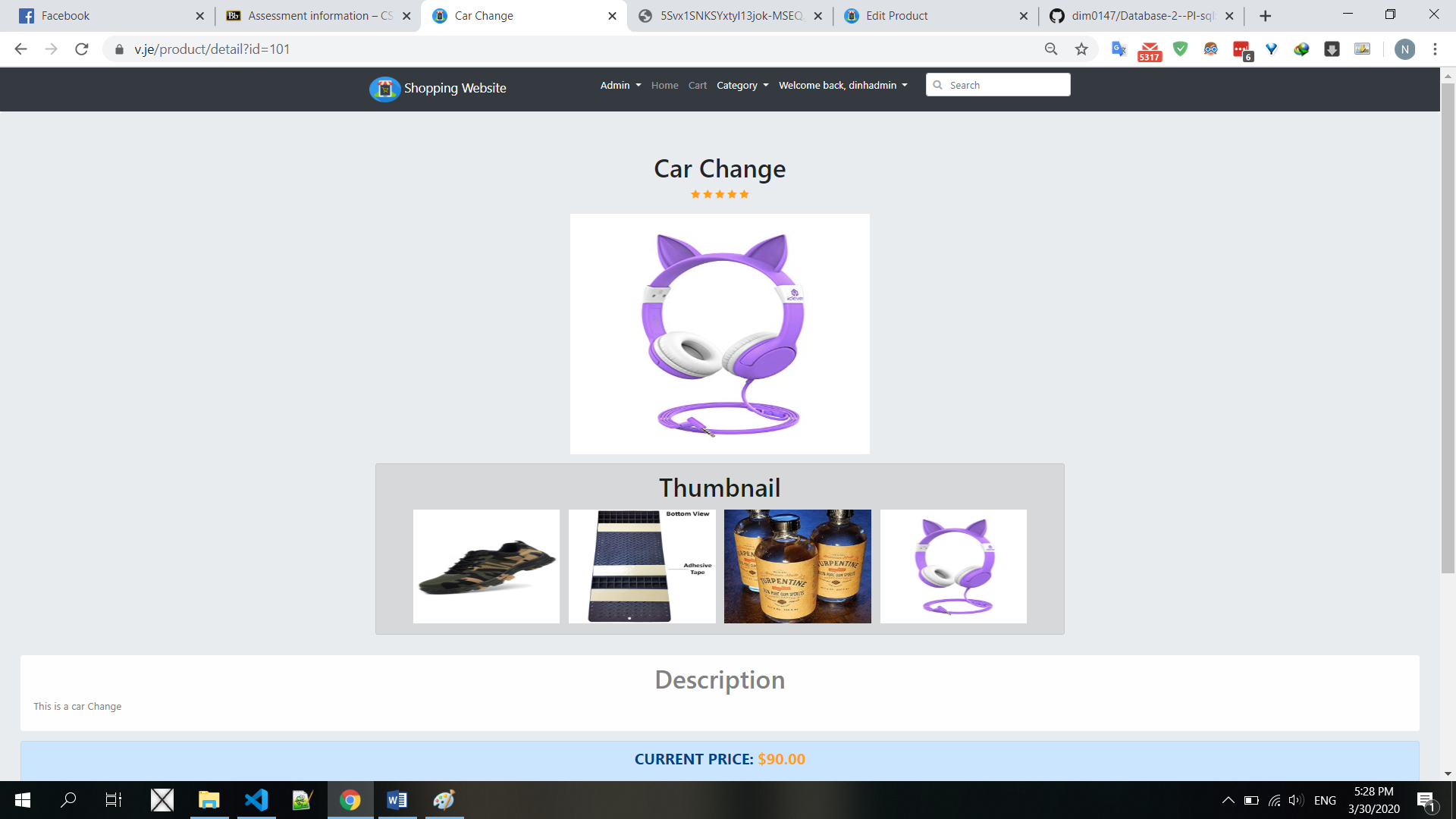
* First I will get detail update to variable $fieldUpdate, then check if have update header image, I will get current image of product and assign it into $oldImage, after that update the product, if update success, I will delete the old image. Now I will testing, first get the current image first:



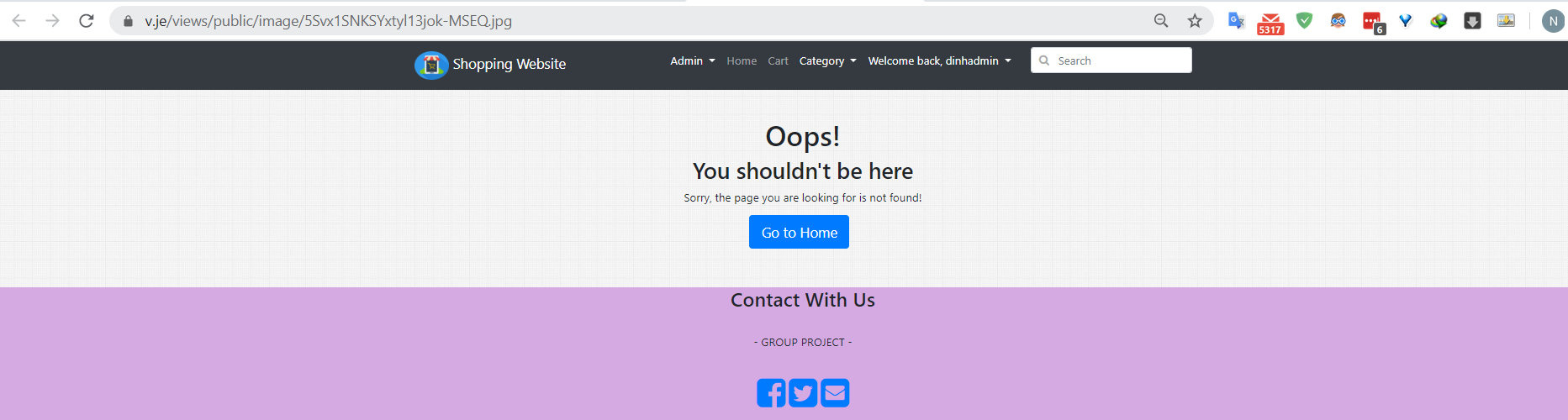
* Now I’m gonna go change the header image:



* After change finish, the product will look like this:



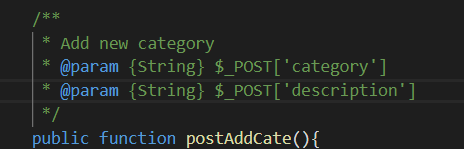
* Now I gonna go load the old image url to check if it still exist or not:



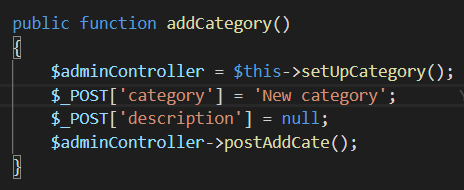
* The website cannot found old image in my storage, great, now edit product is actually edit the product which will remove the old image when change to new one.

3.Add Category

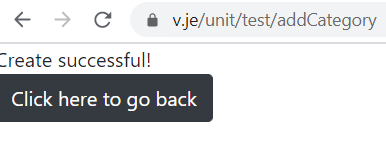
- Now I will come to add category functionality, this will add a new category, require two field:



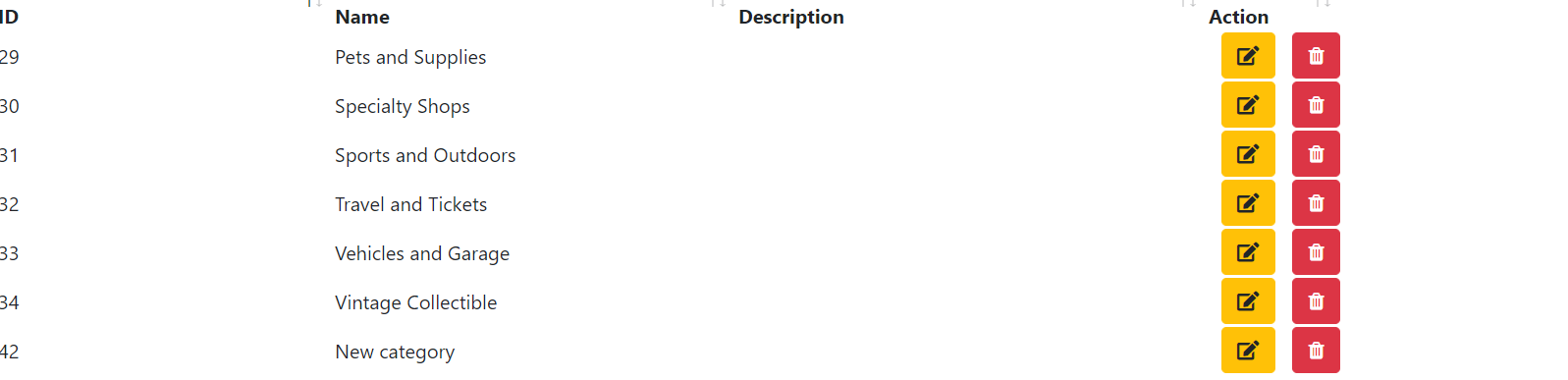
* I’m gonna write a little unit test for this:



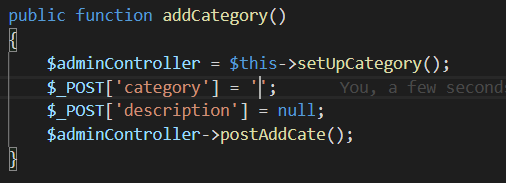
* This unit test will create two parameter as required above, the category name is ‘New category’ and the decription is null, now let run it and see if it can create new one:



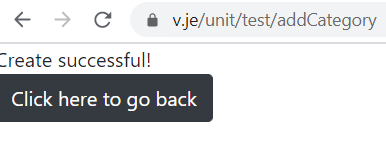
* The output said create successful, now I will go check if it exists or not:



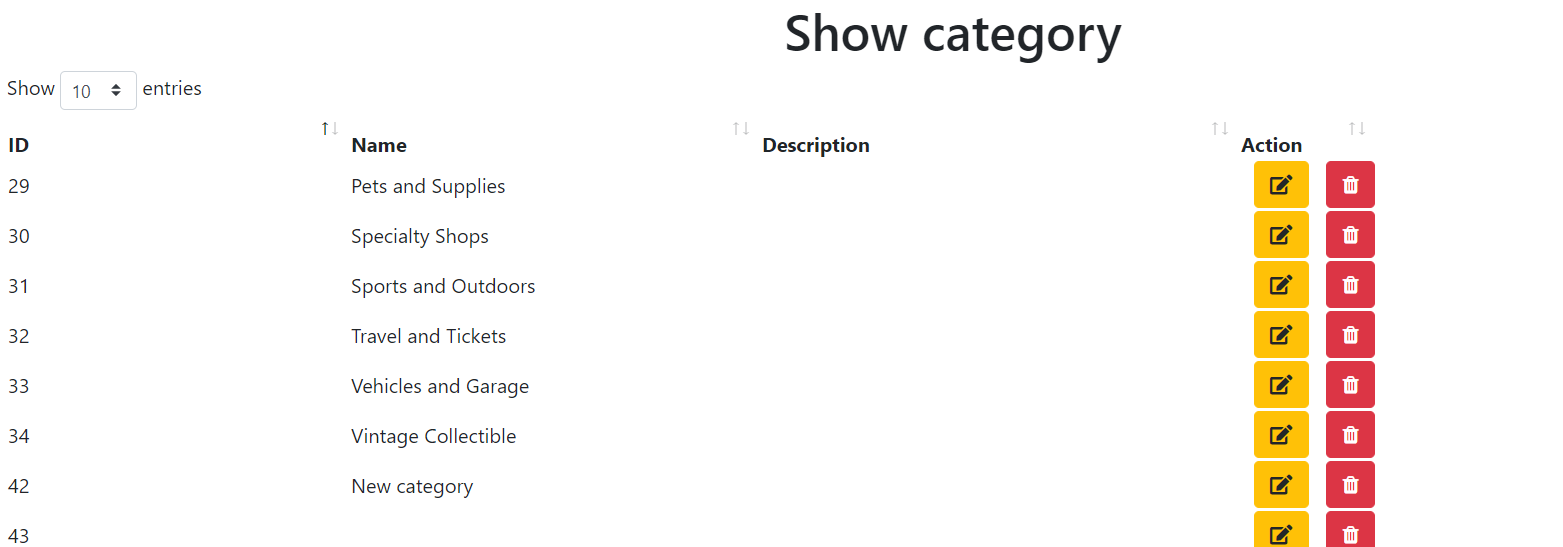
* The last one which ID 42 is a new category I just create, the function working, now if I pass an NULL value for category to create let see what happen:



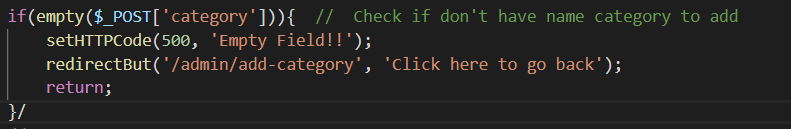
* Output:



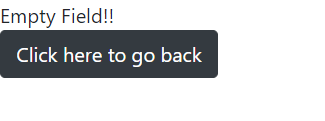
* Can create, something went wrong, I go to check the newest category just create and get this result:



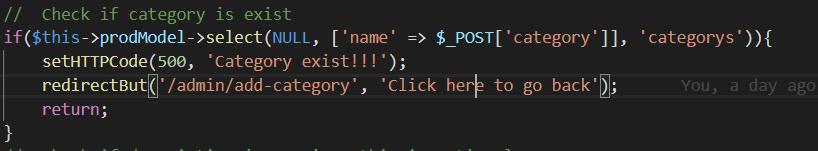
* As you can see, the ID 43 the name category is empty, to avoid this I have to check the category must not null, I add the following code to valid it:



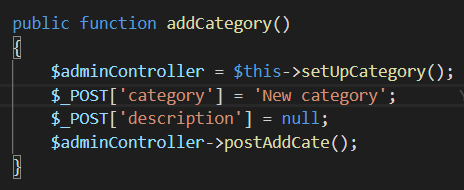
* Now run again, the output show:



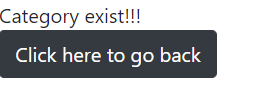
* Its show the error as we coding above, we have solve the problem, but if the user create a new category but if exist already, then will have duplicate category, to avoid this, I will add one more validation:



* This code will get the category by name, if found one, it will show the error, now I will re-create the category which I have create first one, let see what happen:



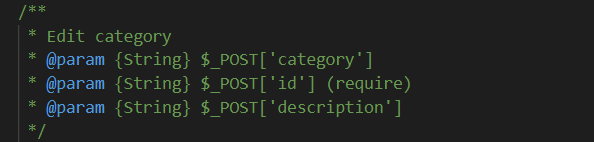
* Output:



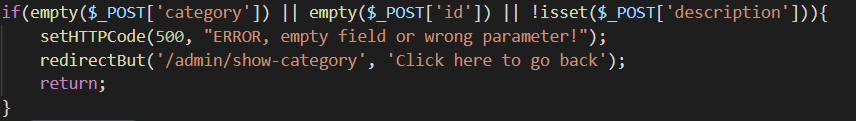
* The error show like we coding, now we make sure the category is not empty and cannot be duplicate.

4.Edit Category

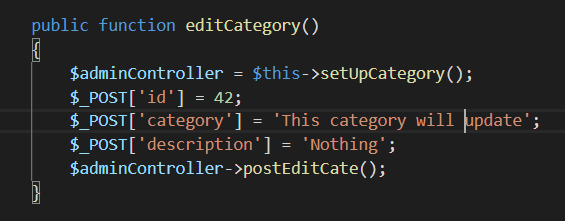
- Now I come with edit category, this will require 3 field:



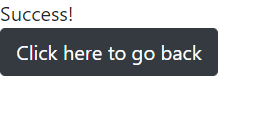
* The ‘category’ is the name category we want to edit, the ‘id’ is the id of category we identify with category should be update and the ‘description’ is the description we want to update also.
* As in add category section, I should check the field not empty, otherwise we will update the category with empty name, so to make sure everything works, I have to check these field before update:



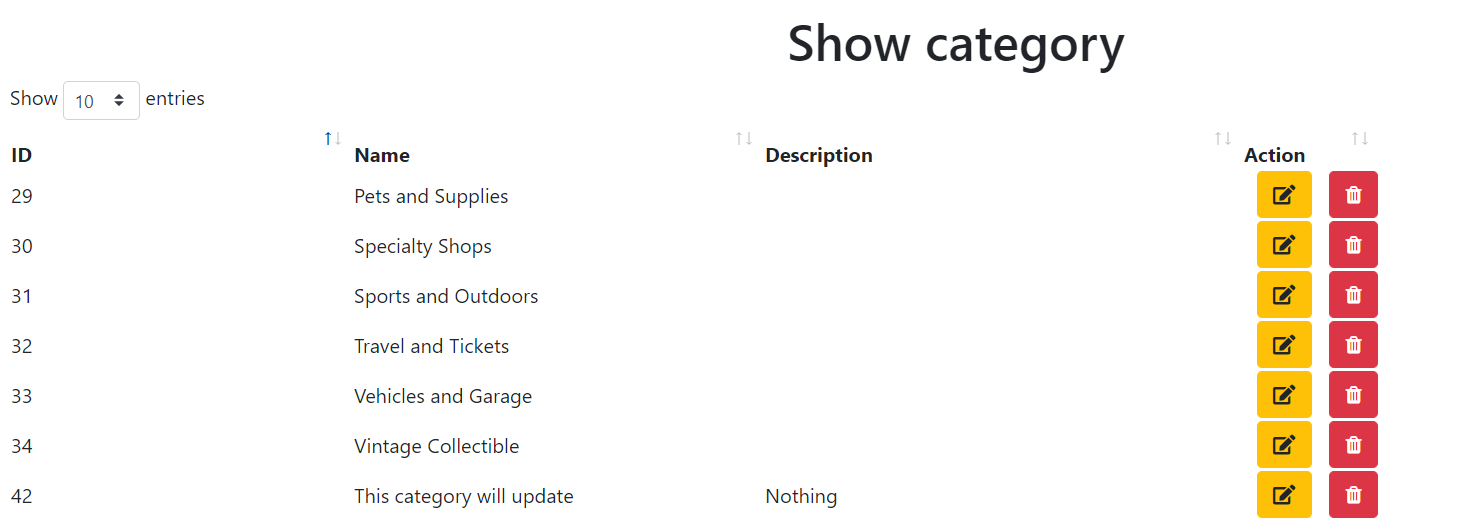
* Now I will create a unit test for this, I will update the category which I just create from add category section:



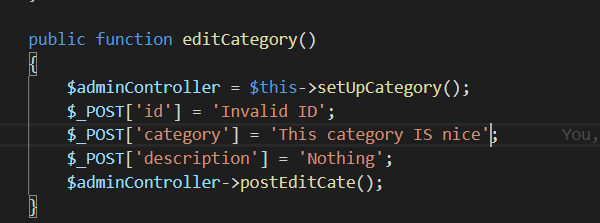
* Run the unit test, Output:



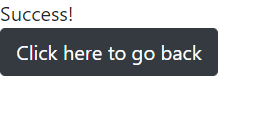
* The output so success, now we go to check:



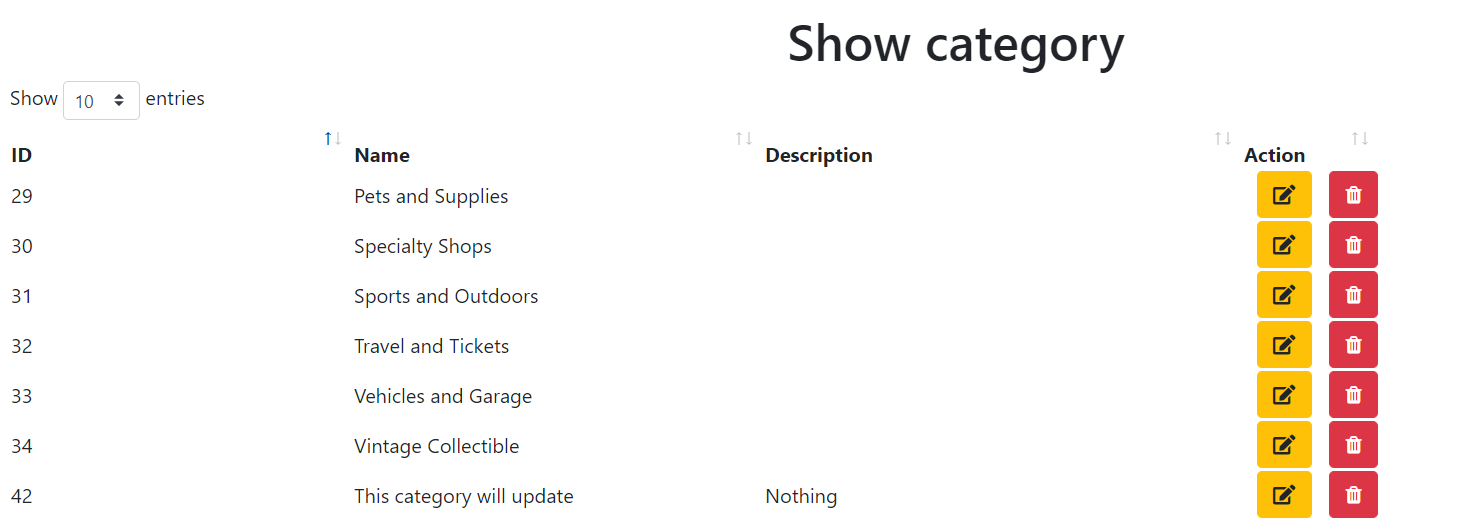
* As we can see, the ID 42 have the name exactly we expect, but if I put the wrong ID or invalid ID, what will happen? Let go check, edit the unit test:



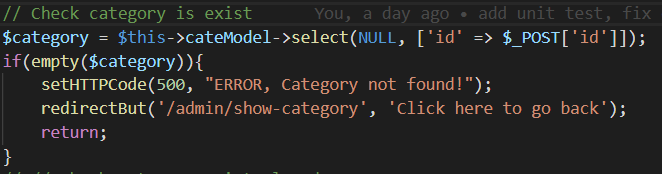
* Now run, output:



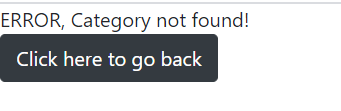
* Output said success, go check:



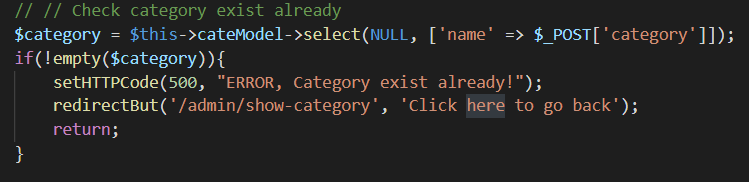
* Nothing is update, so to fix this, I have to check the id passed is found on Database before update, so the code will be:



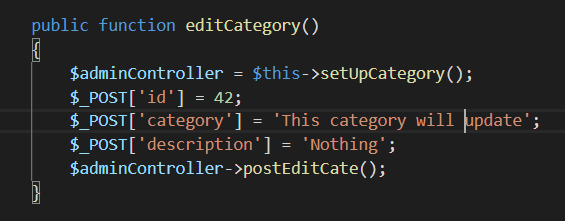
* Now run again:



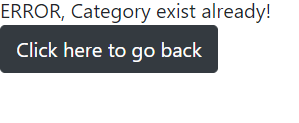
* The error show like we expect, but if the category name we want to edit is exist already? Well we need one more validation case for this:



* Change the unit test to update name of category as the name exist already in Database:



* Go testing, got the result:



* Hooray, now we can make sure the ID of category we want to update must exist and the check whether the category update name is exist already or not.