|  |  |  |  |
| --- | --- | --- | --- |
| **Functionality** | **Completed? (Yes/No/Partially)** | **Relevant files containing code for this feature** | **Comment, \*Notes** |
| **DIR PREFIX, PLEASE GO TO THIS DIR TO SEE ALL FILE IN “ Relevant files containing code for this feature” SECTION:**  ***APP/modules/*** | | | |
| **Administrator log in** | Yes | Admin/controller/controller.php | -login function**.**  **-**postLogin function. |
| **Add categories** | Yes | Admin/controller/controller.php | -addCate function.  -postAddCate. |
| **Edit categories** | Yes | Admin/controller/controller.php | -editCate function.  -postEditCate function. |
| **Delete categories** | Yes | Admin/controller/controller.php | -removeCate function.  -postRemoveCate function. |
| **Added categories in website menu** | Yes | Admin/controller/controller.php | -addCate function  -postAddCate function |
| **User registration** | Yes | User/controller/user.php | -register function  -postRegister function |
| **User login** | Yes | User/controller/user.php | -login function  -postLogub function |
| **Post auction** | Yes | User/controller/user.php | -addAuction function  -postAddAuction function |
| **Edit auction** | Yes | User/controller/user.php | -editAuction function  -postEditAuction function |
| **Delete auction** | Yes | User/controller/user.php | -postRemoveAuction function |
| **Page user profile** | Yes | User/controller/user.php | -profile function |
| **Show all auctions of current user** | Yes | User/controller/user.php | -profile function |
| **Show all reviews of current user** | Yes | User/controller/user.php | -profile function |
| **Show all Bids of current user** | Yes | User/controller/user.php | -profile function |
| **Latest auctions on home page** | Yes | home/controller/home.php | -index function |
| **View products in chosen category** | Yes | home/controller/home.php | -search function |
| **View reviews of a user** | Yes | product/controller/product.php | -detail function |
| **Add review to user** | Yes | product/controller/product.php | -postReview function |
| **Calculate auction time remaining** | Yes | product/controller/product.php | -detail function |
| **Show all bids of the specific auction** | Yes | product/controller/product.php | -showBidAuction function |
| **Administrator approval auction before appearing on the website** | Yes | Admin/controller/controller.php | -approveAuction function |
| **Allow uploading an header image with the auctions** | Yes | User/controller/user.php | -postAddAuction function |
| **Allow uploading up to 5 images of each product (thumbnails image)** | Yes | User/controller/user.php | -postAddAuction function |
| **Social media buttons** | Yes | product/view/detail.php | -line 80 to line 89 |
| **Allow searching for products by string** | Yes | home/controller/home.php | -search function |
| **Allow searching for products by category** | Yes | home/controller/home.php | -search function |
| **Allow administrators to manage administrator accounts** | Yes | Admin/controller/controller.php | **-**showAccount function |
| **Securely store passwords** | Yes | User/controller/user.php | -postRegister function  -Using hash password from php support(password\_hash, password\_verify) |
| **The ability to click on a user and see any auctions they have created** | Yes | User/controller/user.php | -showUserProfile function |
| **Implement PayPal payments in the shopping basket.** | Yes | product/controller/product.php | -createPayment function |
| **Implement a buy it now price** | Yes | product/controller/product.php | -placeHotBid function |

**2.TESTING**

**1.Place bid on auction**

- First, I will implement the place bid auction, the user can place bid on many auctions, I will check if the amount(money) to place bid is empty or user who want place bid is empty and the id of the auction which will be placed is empty, if empty send error to user, otherwise place bid on the auction and update current bid of auction, the function will look like this:

Function placeBid{

If(amount == empty) or (user == empty) or (ID\_Auction == empty)

(

Error(‘Some require value is missing’)

StopExecution()

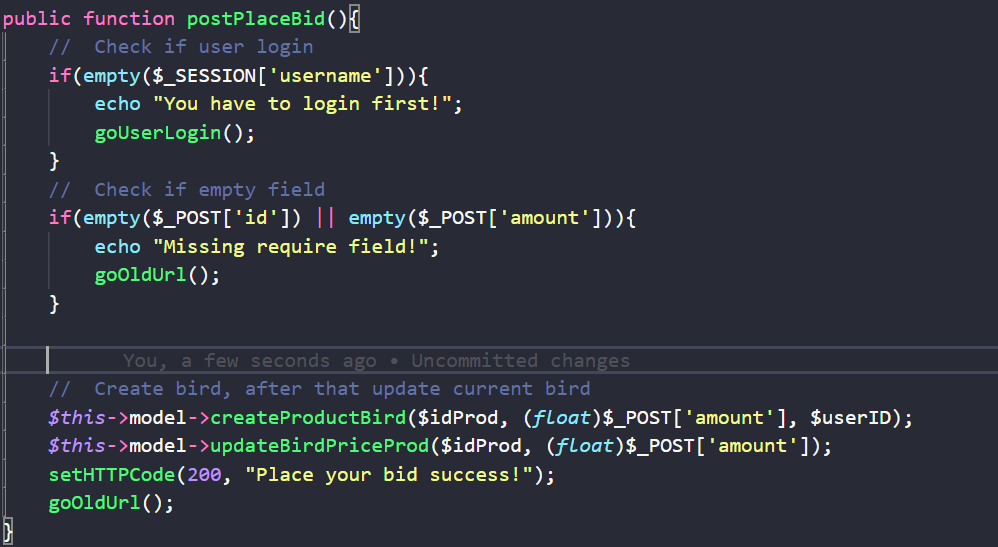
)

UserPlaceBid()

UpdateCurrentBidAuction()

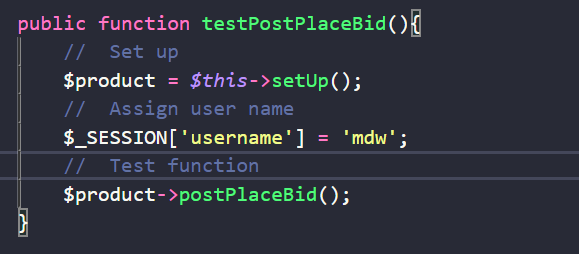
}

* Now the actually function is like this:

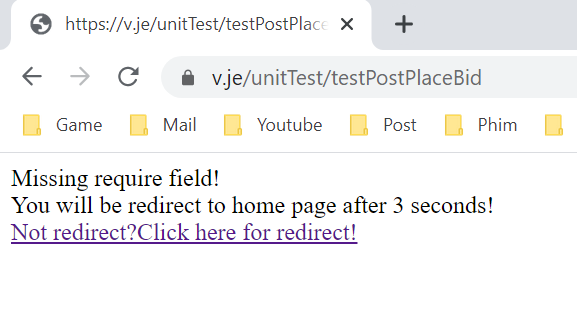


* I will go to test this by write a simple unit test, the unit test will check all condition and test every aspect to ensure there is no error.

+ Assign username, if we correct, the output should be the string “Missing require field!” because we have username only, don’t have another field:

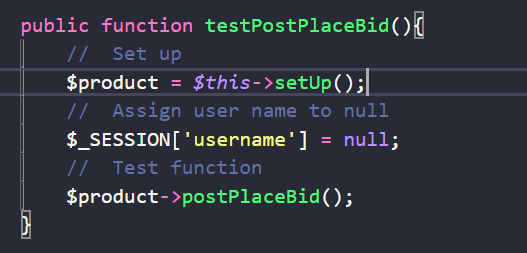


Result:

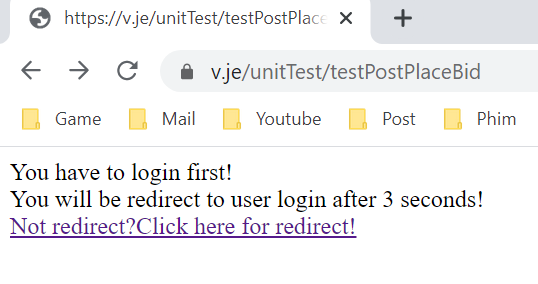


\* The test was success, the error show like we are expect, now we going to test without user name.

+ No assign username, the result should display the string “You have to login first” because we didn’t login yet, if login we will have username, in this case we don’t have:



Result:

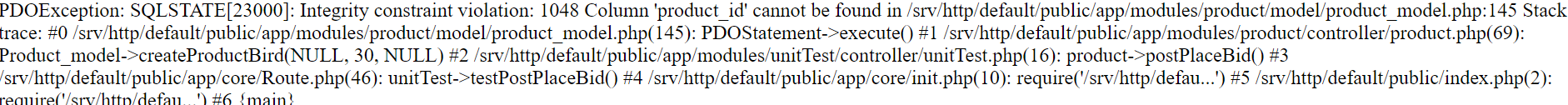


\* The test was success, the error show like we are expected, now we going to test with full require field.

+ Assign full require field, now go test:

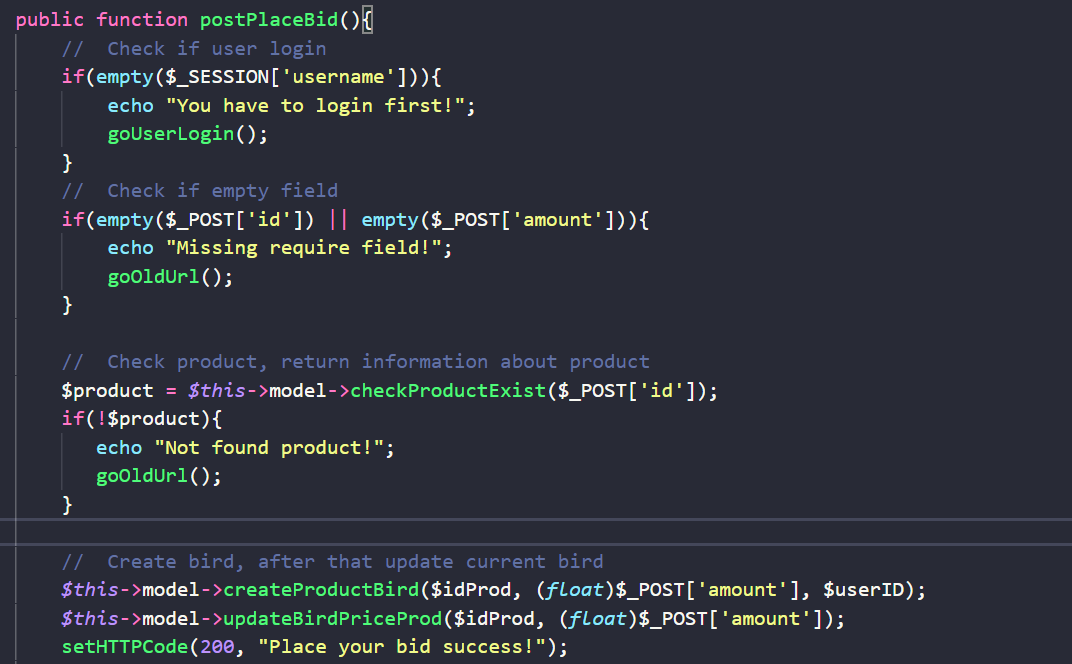


Result:

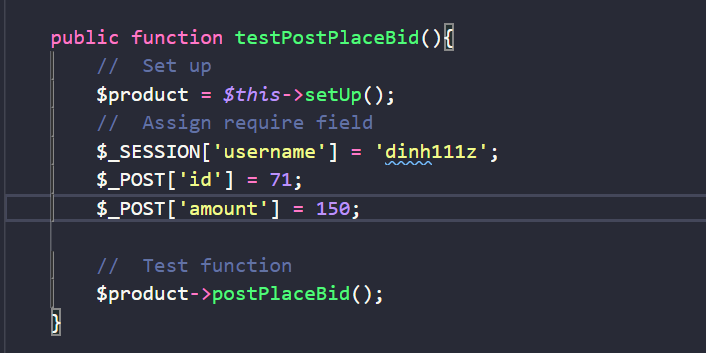


\*The test have an error, like you see, the product\_id(auction\_id) is not found, so that mean I have pass wrong ID auction and they didn’t found the product id to update, so I will need to implement testProductExist function to check if the product is exist or not.

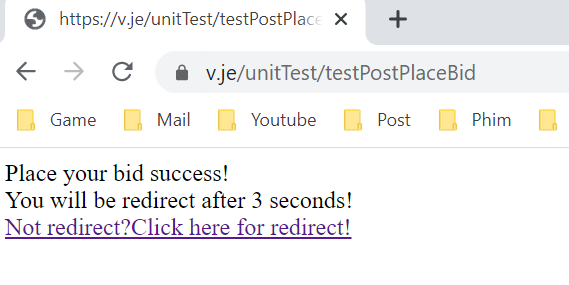
+ Implement check product if exist, now if product not found we will show the error and redirect user:



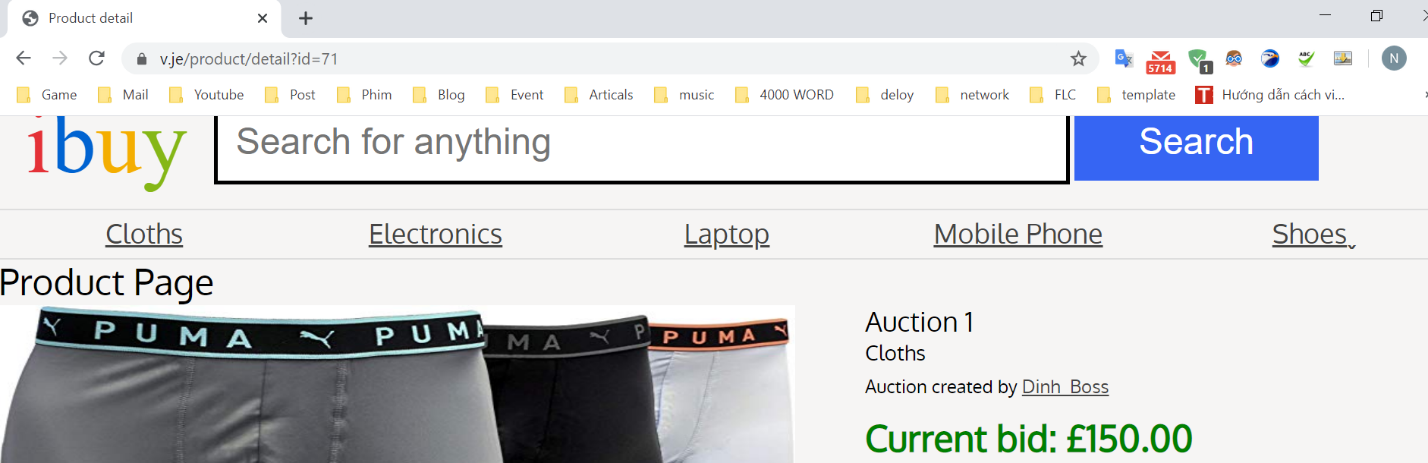
+Setup again unit test:



+ Now go test again, the output need to show ‘Place your bid success’:

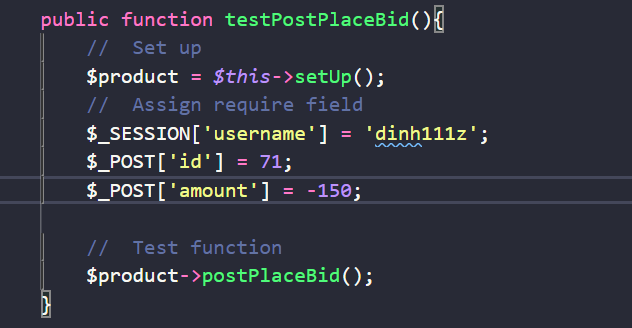


\*Horayy, we success place bid on this auction, now go to that auction to see our bid:

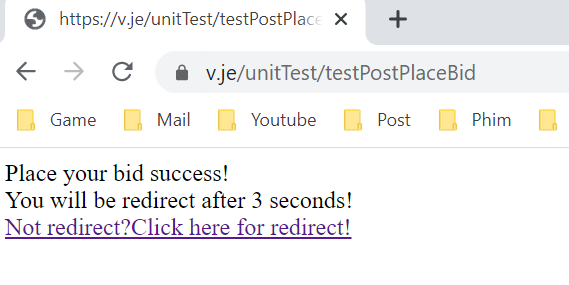


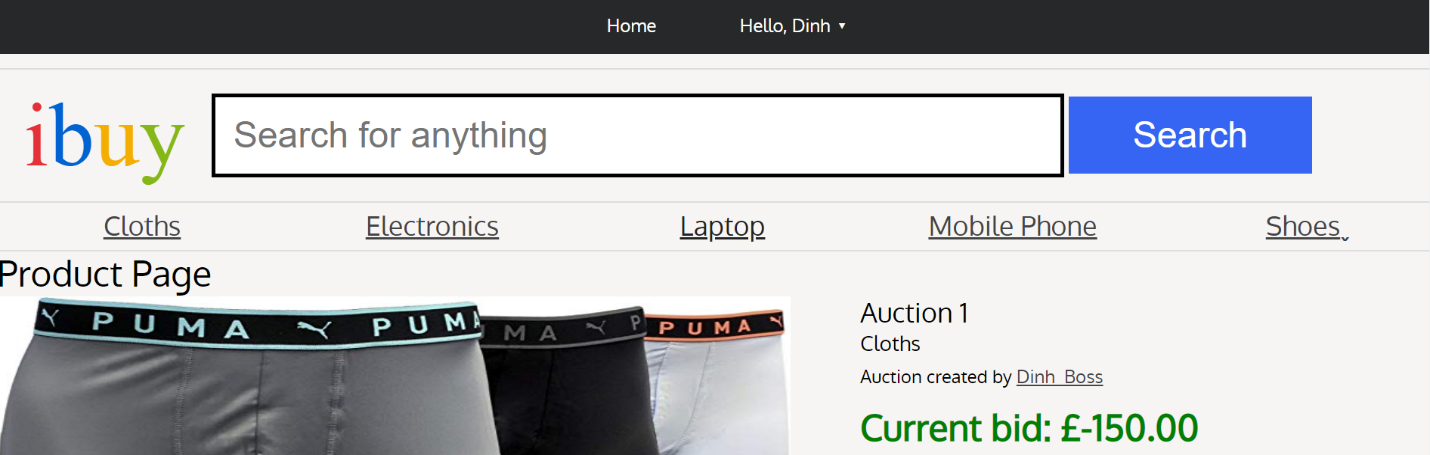
\*The result show current bid is 150, which is exactly the amount number that we pass from unit test.

- But if the user entered the amount is negative number or smaller than current bid, we will have trouble here, let test this, I will assign amount with negative number:



+ Now run and see the result:



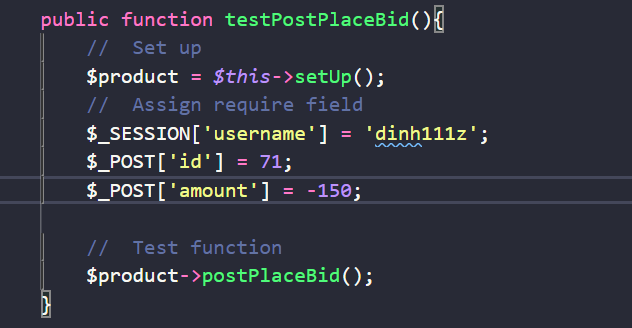


\*So funny, this will make our website like a joke, so to prevent that, I will implement checkAmount function to check if amount is negative number or below then current bid, will show the error and redirect user:

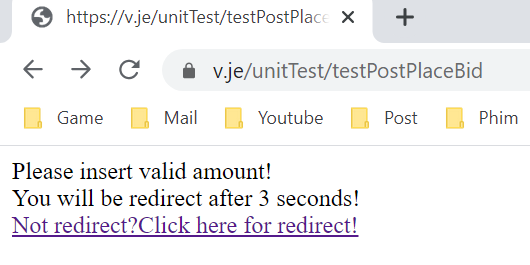


-Now I will test, first test is amount is negative number, second test amount is smaller than current bid, and the final is amount is a string

+Assign amount is negative number, the output expected “Please insert valid amount”:

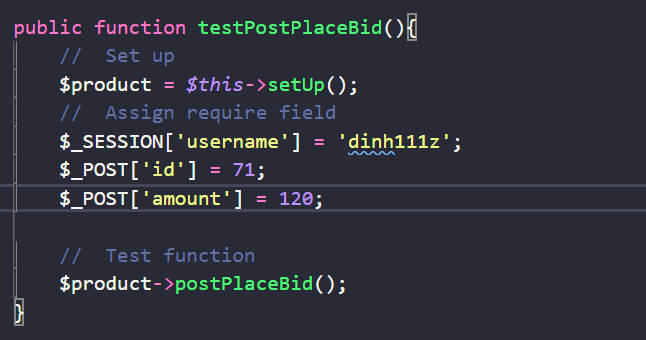


Result

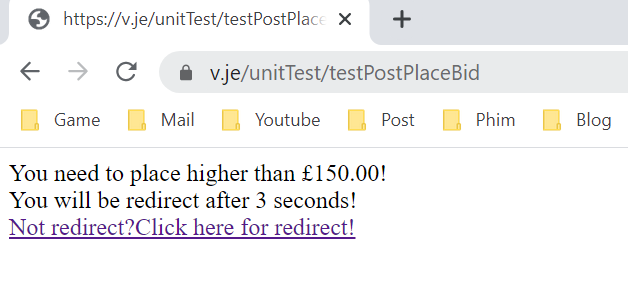


\*The result has success like we expect, but this is not done yet, now go to the next one.

+Assign amount is lower than current bid, here I will assign 120, the output expect “You need to place higher than 150” because I have reset current bid to 150 dollar, so the bid must be more than 150 to place bid:

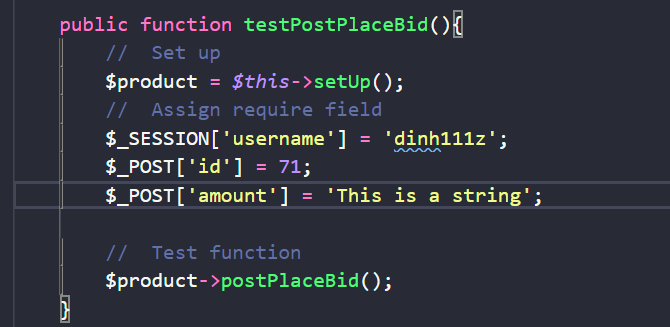


Result:

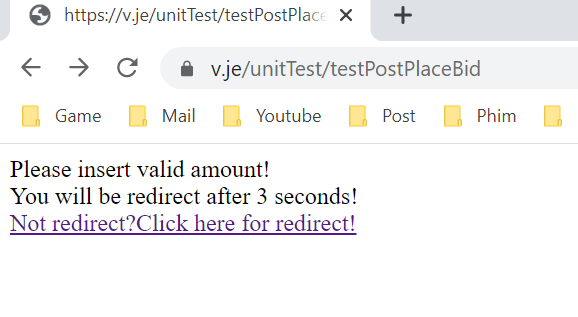


\*The output is like we expect, so we need to place bid more than 150, now come to pass a string into amount to see what happen.

+ Assign a string into the amount, the output should be “Please insert valid amount”:

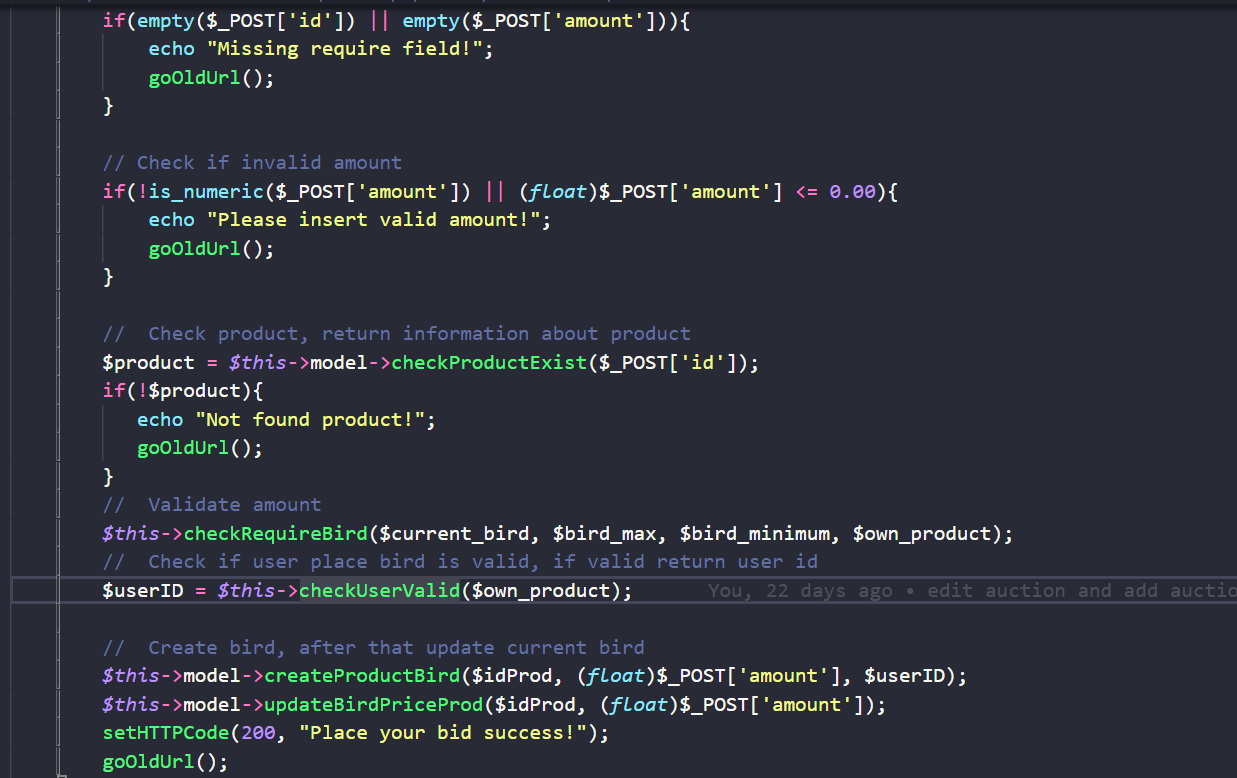


Result:



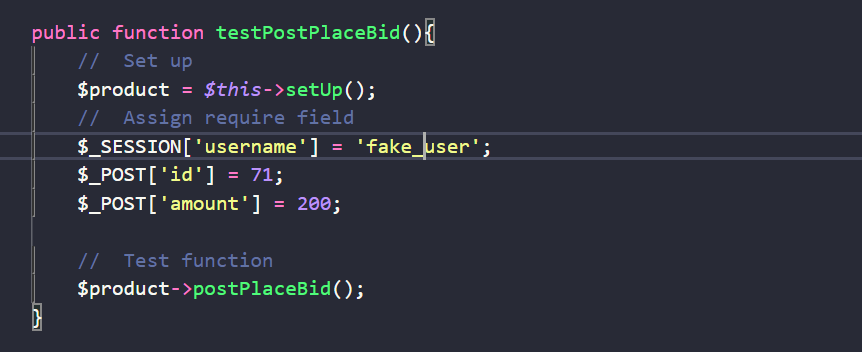
\*The result like 100% that we are expected, the unit test was done his job. But if the owner of product is place bid, should we accept that? Completely no, So now I will need to add a checkUser function to test if any invalid user (like the owner of product or don’t found the user) place bid, I will show the error and redirect them.

- Implement checkUser function:

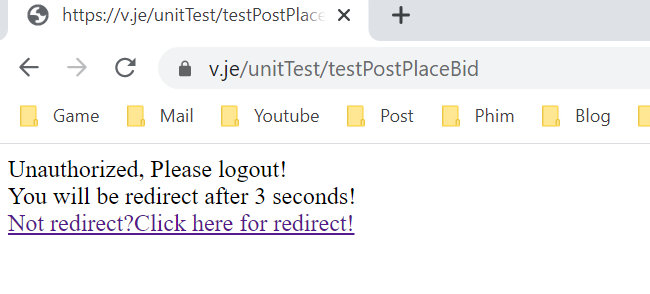


-Now I will go test 2 case, first test one is a fake user which user is not true, not register yet, and second test is the owner of the product.

+Assign fake user, the expect should be “Unauthorized, Please logout”:

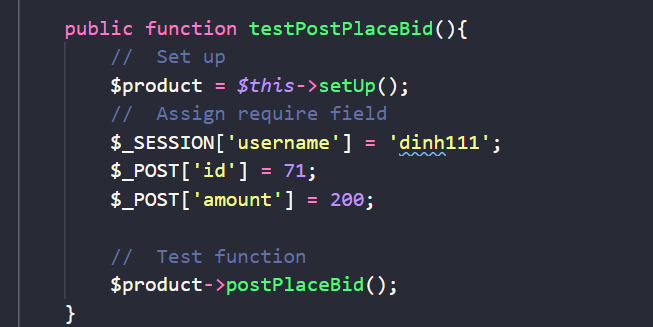


Result:

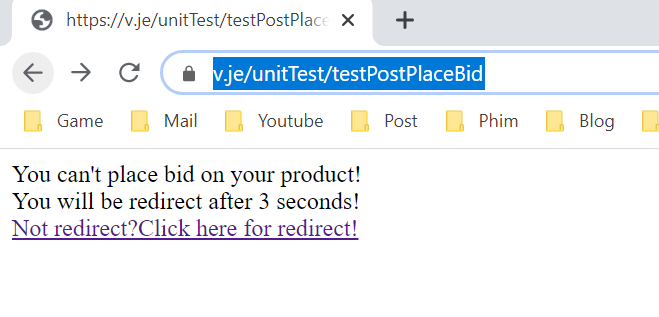


\*The output exactly like we expect, now come to second test.

+ Assign owner of auction, the output should be “You can’t place bid on your product!”



Result:



\*The result was wonderful, the output showing exactly like we expect, the test now complete, we ensure there is only valid case, other case will return the error depend on error that we check.

TESTING TABLE:

|  |  |  |
| --- | --- | --- |
| **Case** | ***First time test result*** | ***Second time test result*** |
| **Empty user name** | Success, output "You have to login first" |  |
| **Empty Id Product or empty amount** | Success, output "Missing require field" |  |
| **Fill full required field (not empty username, Id Product, amount)** | Error, output "product not found" | Add checkProduct function then test, again, Success, output "Place your bid success" |
| **Amount is negative number** | Error, output "Place your bid success", the expect output should be “Invalid amount” | Add checkAmount function then test again, Success, output " Please insert valid amount" |
| **Amount is lower than current bid** | Add checkAmount function then test, Success, output " You need place higher than ... dollar" |  |
| **Amount is a string** | Add checkAmount function then test,  Success, output " Invalid amount" |  |
| **If user is not existing?** | Add checkUser  function then test, Success, output " Unauthorized, please logout" |  |
| **If user is the owner of product?** | Add checkUser  function then test, Success, output " You can’t place bid on your product" |  |

**2.Admin Register**

- Now I’m going to test the admin account register, the admin account has 3 field: username, password and name of user, to register the admin account, first I will check the username, password and name of user not empty, if yes, I will show the error, otherwise create a new account, the function look like this:

Function CreateAdmin(

If(username or password or nameofuser == empty)

(

ShowError

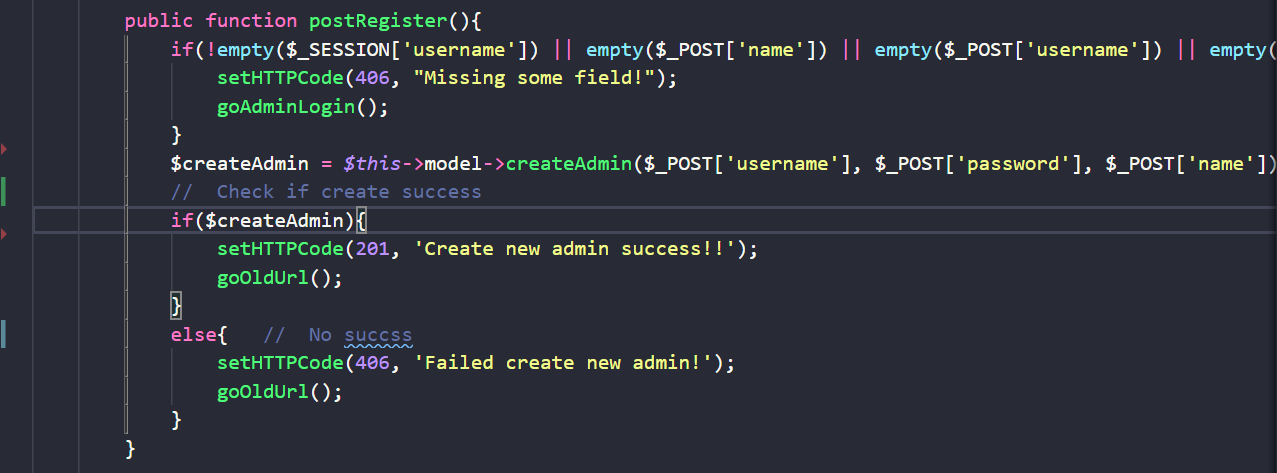
Exit

)

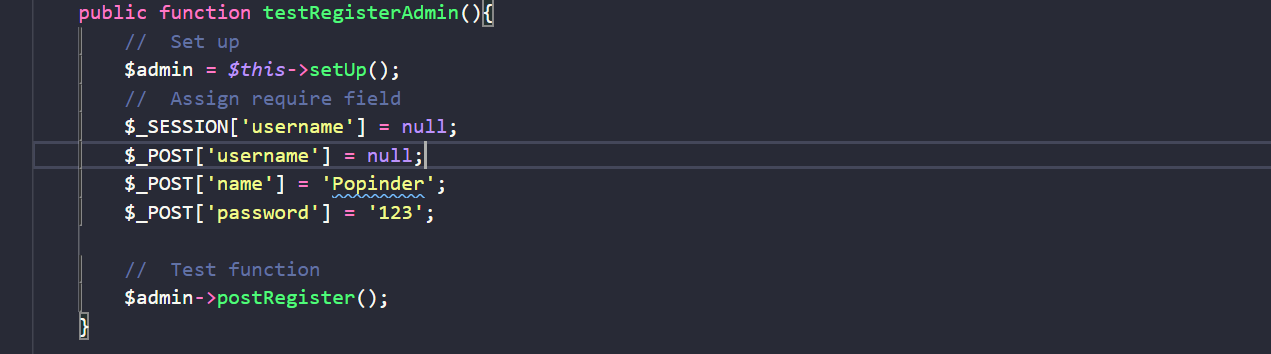
CreateAdminAccount

)

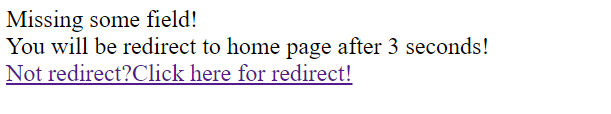
-The actual function:



+First I will test without username, but other field have value, the output must be “Missing some field”:

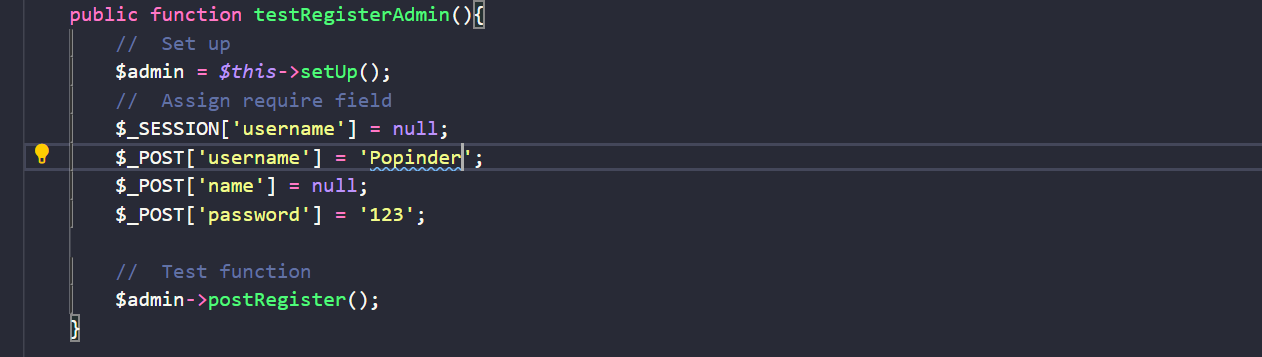


Result:

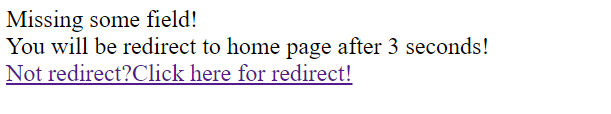


\*The output like we expected, the test was successful, now we continue to empty each field.

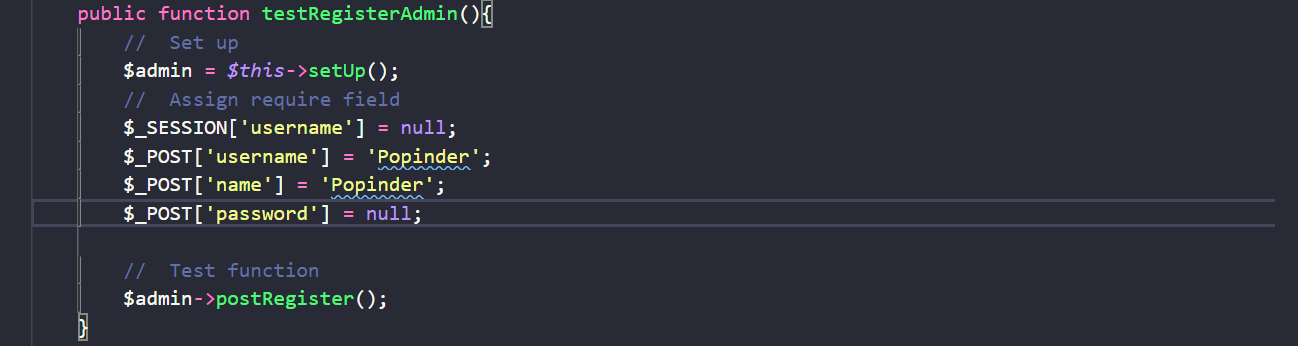
+Empty name, but other field not empty:



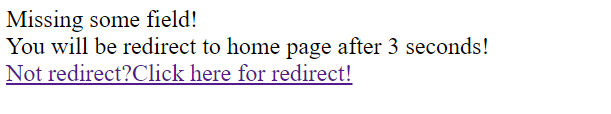
Result:



+Empty password, but other field not empty:

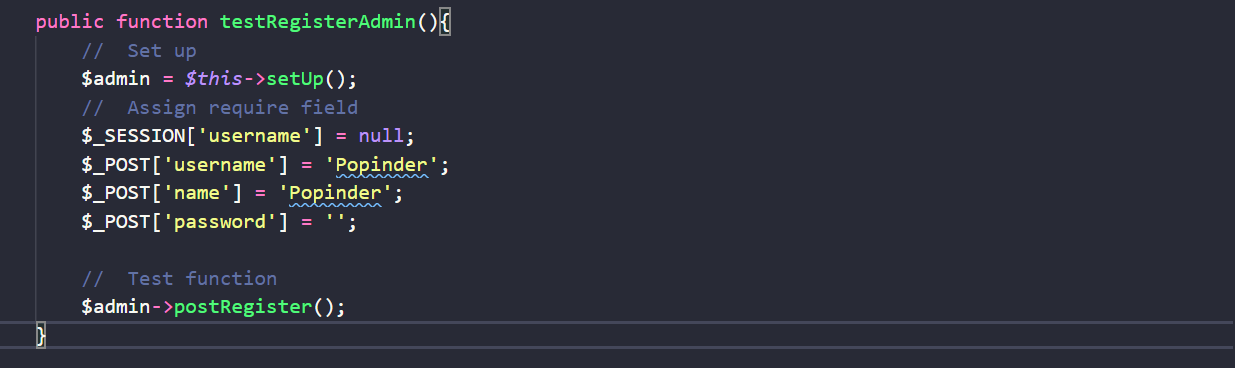


Result:

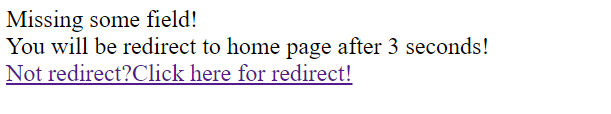


\* The above result return same output “Missing some field” as we expected, but null is not enough, if the user pass the empty string “” is it consider as null? Let test this to figure out.

+Assign password with empty string “”, output expected “Missing some require field”:



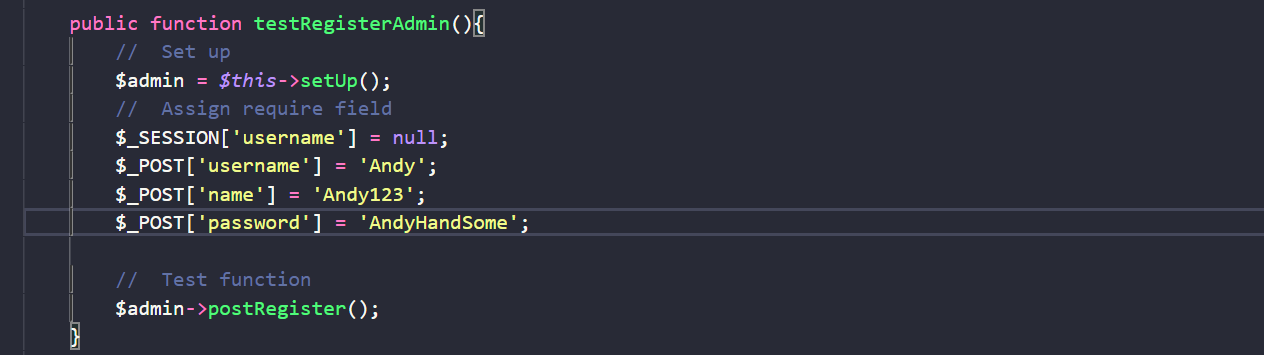
Result:



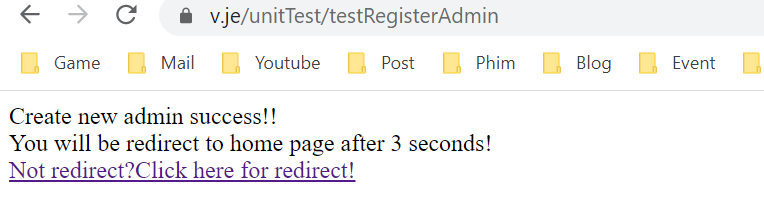
\*The output was the same, still output “Missing some field”, that we are talking about, actually in the php docs it say empty function will consider some case as null, include this case, so we can come there to knows how they consider a variable as null, no need to test this.

- But if the username has already exist? So there will be two admin in our server, sound not good, so I should implement another function call CheckAdminExist to check if admin exist already, if not create new admin.

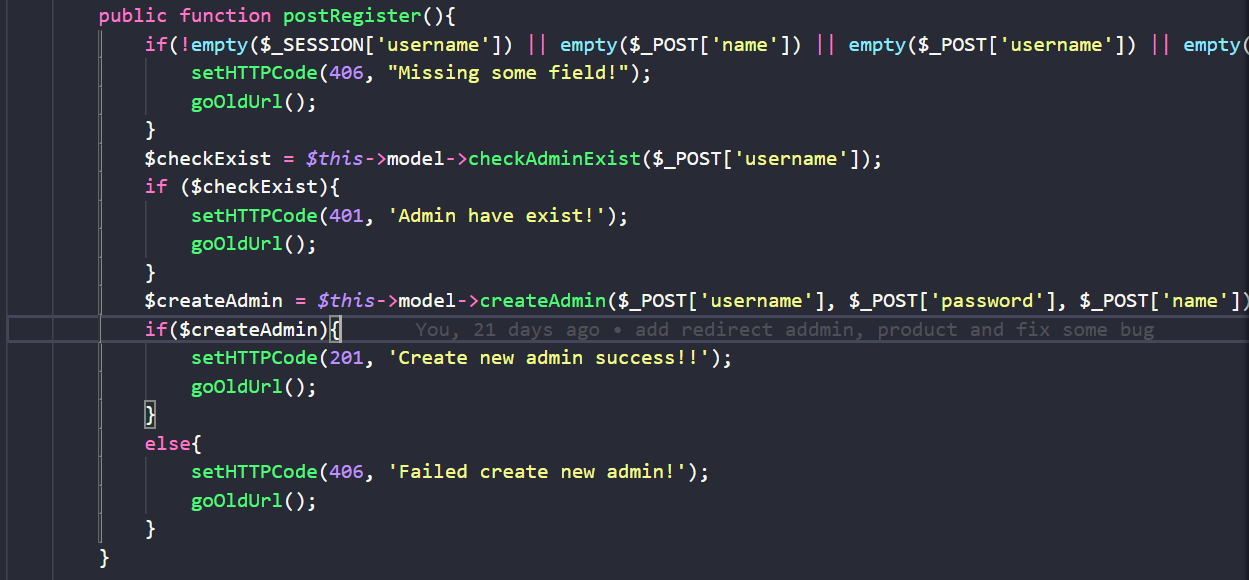
+First I will create admin with username is Andy:



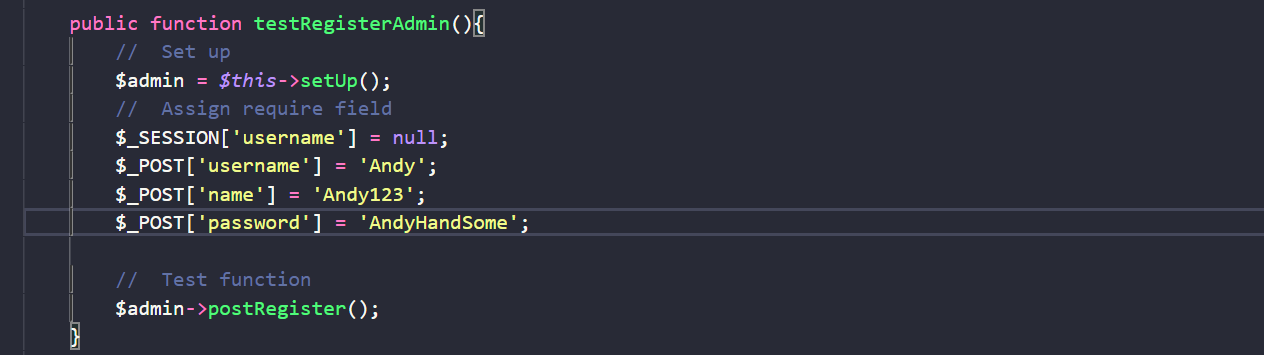
Result:



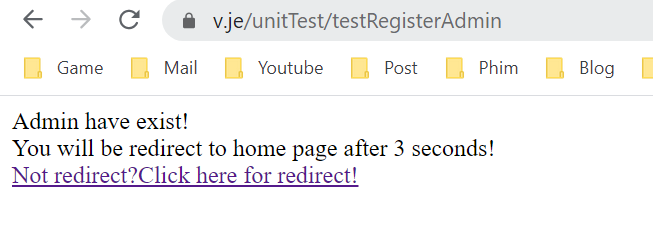
-The admin Andy has created, now we will implement checkAdminExist function:



+Now we going to test again, create new admin with name is Andy, the output must be “Admin have exist!”



Result:



\*The test was success, the output was predicted before, now the admin cannot register with same username, avoid unnecessary error.

-This test part seem to be easy than the first one, this just the basic so the test was easy to pass by simple unit test, in future if we want to extend more functionality, we will implement more features and then the register function will become harder so the unit test will be our hero to detect error and fix it.

|  |  |  |
| --- | --- | --- |
| **Case** | ***First time test result*** | ***Second time test result*** |
| **Empty username** | Success, output "Missing require field" |  |
| **Empty name** | Success, output "Missing require field" |  |
| **Empty password** | Success, output "Missing require field" |  |
| **Assign password with empty string “ “** | Success, output "Missing require field" |  |
| **Same username** | Add checkAdminExist function, Success, output "Missing require field" |  |

* **Conclusion:**
* From this test section I realized that in addition to writing code, we have to test the code, test the cases to make sure the code runs correctly, avoid potential errors, and know how to write basic unit tests, although it is lacking, omitted and not partially detailed but from this I have a bit of experience in testing code.