CSY2028: Web Programming

PHP & MySQL Assignment 1



A imagem [Esta Fotografia](http://craftclefojrwnj.wikidot.com/blog:_start/date/2015.8) de Autor Desconhecido está licenciada ao abrigo da [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

Ana Rosa Gaspar 18400473 BSc Computing

Índice

Introduction & Problem Specification 1

Technical documentation 2

Checklist 2

Database structure 2

Website structure 3

Testing 9

References 10

# Introduction & Problem Specification

The purpose of this assignment was to check our ability to create a database driven website using PHP and MySQL.

For that, it was provided a zip file containing the HTML layout with the relevant CSS and Images, that should be used for the look of an electronics shop website.

Using what was provided, the task was to implement functionalities into the website and allow users to find information about products the shop has in stock. The following list of functionalities that should be implemented into the website was given:

* Each product should have a page where a title, price, manufacturer and a longer description is showed.
* The company should be able to log in to a secure administration area and add new products, entering all the information about the product, making it visible to the public.
* Each product should fall into a category such as TVs, Cameras, Phones or Gaming.
* Categories should not be limited to these choices and new categories should be able to be added via the administration area.
* If a product is marked as Featured, it should be displayed in the sidebar.
* The public should be able to browse products and filter by category, for example, seeing all Phones that the shop has in stock.
* Members of the public should be able to add reviews to each product.
* The reviews for each product should be visible below the product description and reviews should only be visible on the page of the product they were posted to.
* When a user posts a review, their name, email address, review text and date the review was posted should be logged and displayed on the page.

# Technical documentation

## Checklist

|  |  |  |  |
| --- | --- | --- | --- |
| **Functionality** | **Completed?** | **File containing**  **code for this feature** | **Any comments** |
| Administrator log in | Yes | Login.php | Email: admin  Password: admin |
| User log in | Yes | Login.php |  |
| User registration | Yes | Login.php |  |
| Add products | Yes | Admin.php |  |
| Add categories | Yes | Admin.php |  |
| Assign product to category | Yes | Admin.php |  |
| Edit product | Yes | Admin.php |  |
| Delete product | Yes | Admin.php |  |
| Edit category | Yes | Admin.php |  |
| Delete category | Yes | Admin.php |  |
| Browse products (public) | Yes | Index.php |  |
| View categories (public) | Yes | Head.php |  |
| Filter products by category (public) | Yes | productslists.php | The category name must be specified (?category=) |
| Add review (public when login) | Yes | Product.php |  |
| View product reviews (admin) | Yes | Product.php |  |
| Searching for products | Yes | Head.php |  |
| Securely store passwords | Yes | Login.php |  |
| Social media review share | Partially | Product.php | There is only one button (facebook) |
| Upload image with product | Yes | admin.php |  |

## Database structure

The base of the website was the database because it stores all the information needed for the website use. The database consists of five tables: categories, featured\_product, products, reviews and users.

The database structure was the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tables** | **Attribute** | **Datatype** | **Defaults/ Constraints** |
| **users** | Email | Varchar(60) | Pk, NOT NULL |
| Password | Varchar(100) | NOT NULL |
| Type | Varchar(10) | ‘regular’ |
| Username | Varchar(45) | NOT NULL |
| **categories** | idcategories | Int(11) | Pk, NOT NULL |
| name | Varchar(45) | NOT NULL |
| **products** | Idproduct | Int(11) | Pk, NOT NULL |
| Name | Varchar(45) | NOT NULL |
| Description | Longtext | NOT NULL |
| Price | Decimal(5,2) | NOT NULL |
| Category | Varchar(45) | NOT NULL |
| Insertdate | Date | NOT NULL |
| Image | longtext |  |
| **reviews** | Idreviews | Int(11) | Pk, NOT NULL |
| Idproduct | Int(11) | Fk, NOT NULL |
| Reviewtext | Longtext | NOT NULL |
| Username | Varchar(45) | NOT NULL |
| Reviewdate | Date | NOT NULL |
| **featured\_product** | idfeatured\_product | Int(11) | Pk, NOT NULL |
| Name | Varchar(45) | NOT NULL |
| Description | Longtext | NOT NULL |
| Date | Date | NOT NULL |
| **Pk** – Primary key **Fk** – Foreign Key | | | |

All the data is inserted, altered, selected and deleted from the tables through the website. The categories, products and featured\_product tables are only accessed and edited by the admin. The other two, users and reviews, are edited by the costumers.

The only way to change the type of account in the users table is through the database, that way only who as access to it can authorize or remove access to the admin area.

## Website structure

The website consists of five main pages (index, admin, login, product, productlists) that can be seen and other three files (head, foot, logout) that are only used to complement the main pages. All pages and files interact with the database.

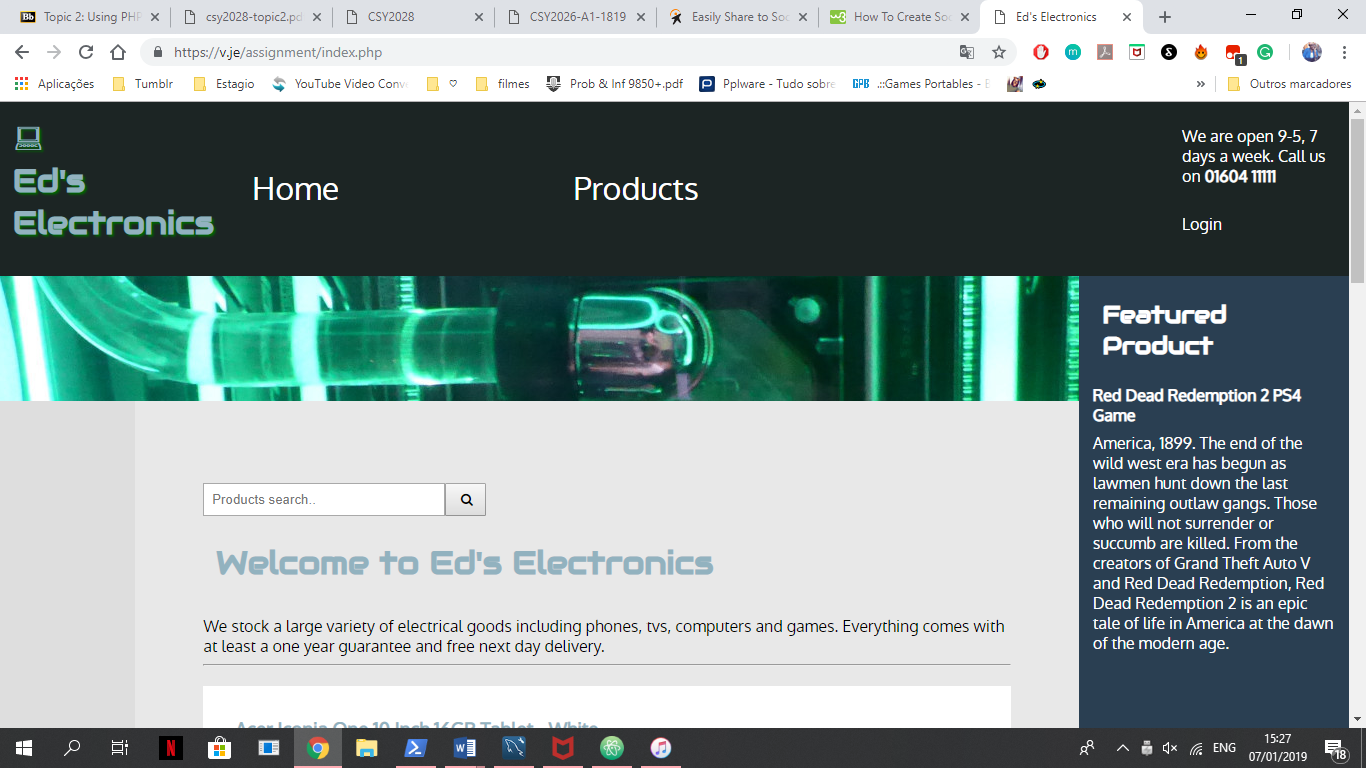
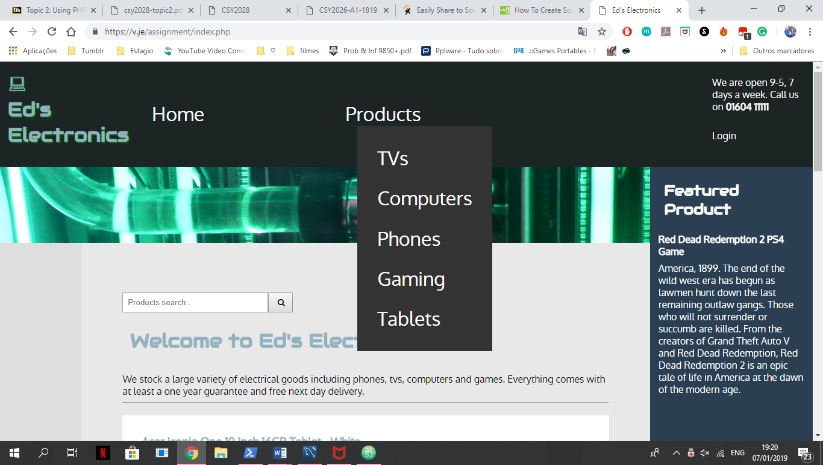
The head and foot files were created to reduce the code repetition as every page has a header and footer. To call for these pages the other pages have to “require ‘head.php’;” in the beginning of the page and “require ‘foot.php’;” at end of the page. In the head.php is the code for the navigation bar.

Figura - Navigation bar

The following code make the menu and links each menu option to another page. So, if we need to add another page to the menu, we can easily do it by adding a new item (<li>) with a link (<a>) to the page between the <ul> tags in the head.php.

As the products option must show a drop-down menu with all the categories stored in the database a connection to the database is made (line 19 to 25).

Once the connection is established, a select is needed to get all categories names and link each name to a page that list all the products of that category (line 29 to 32).

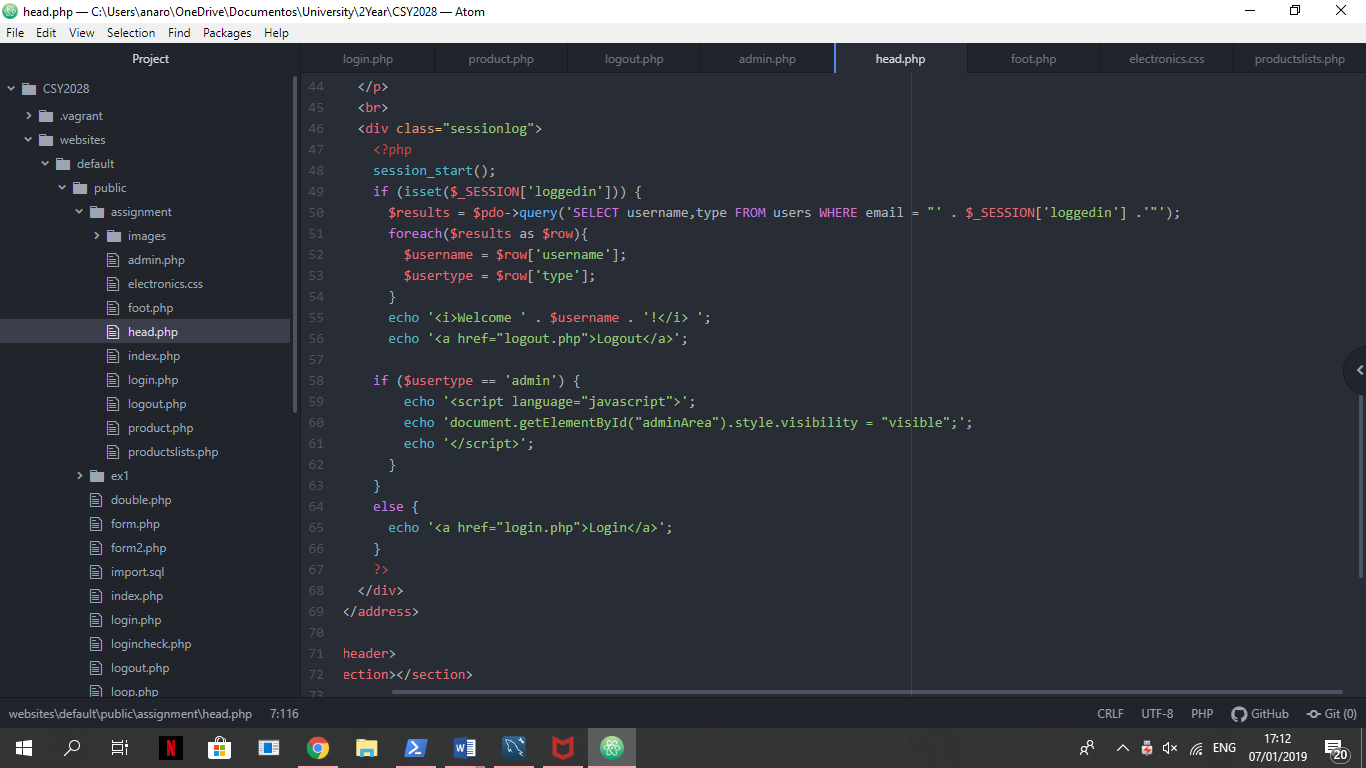
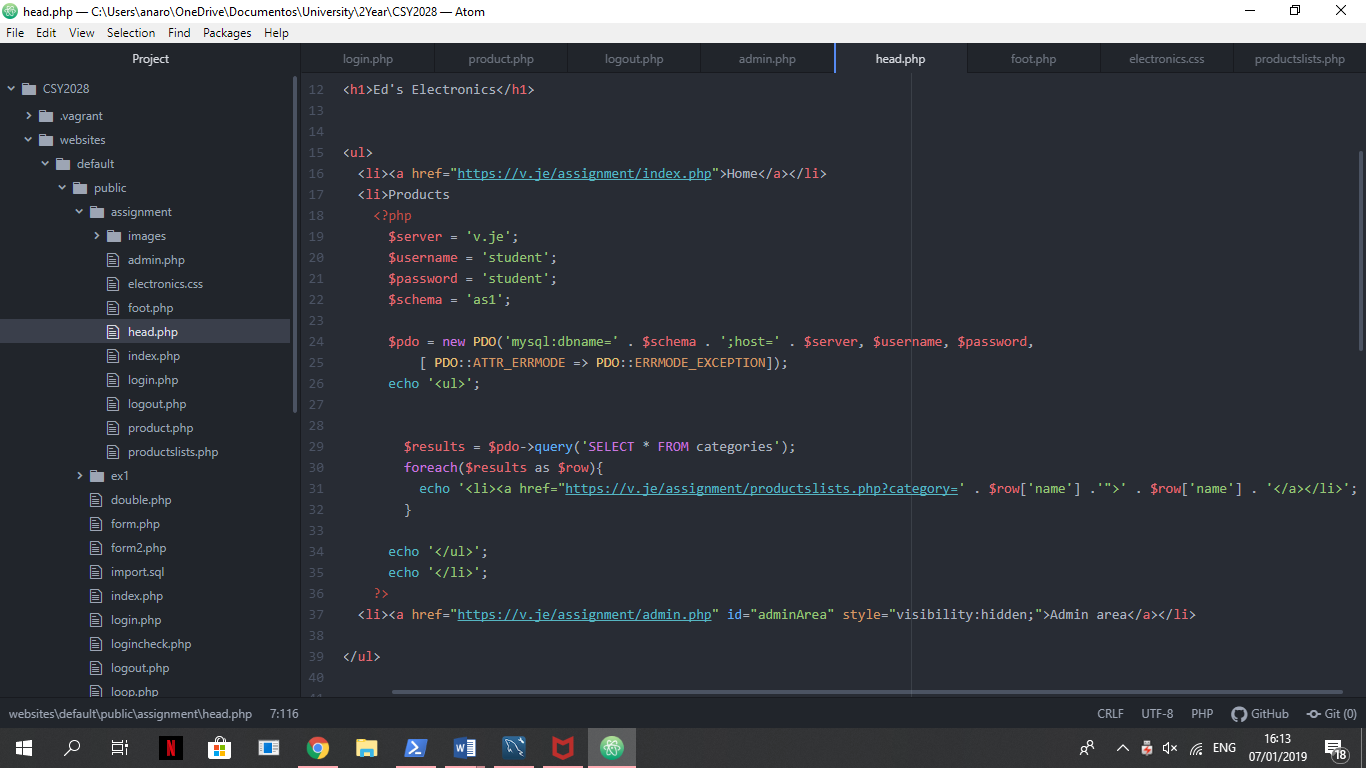
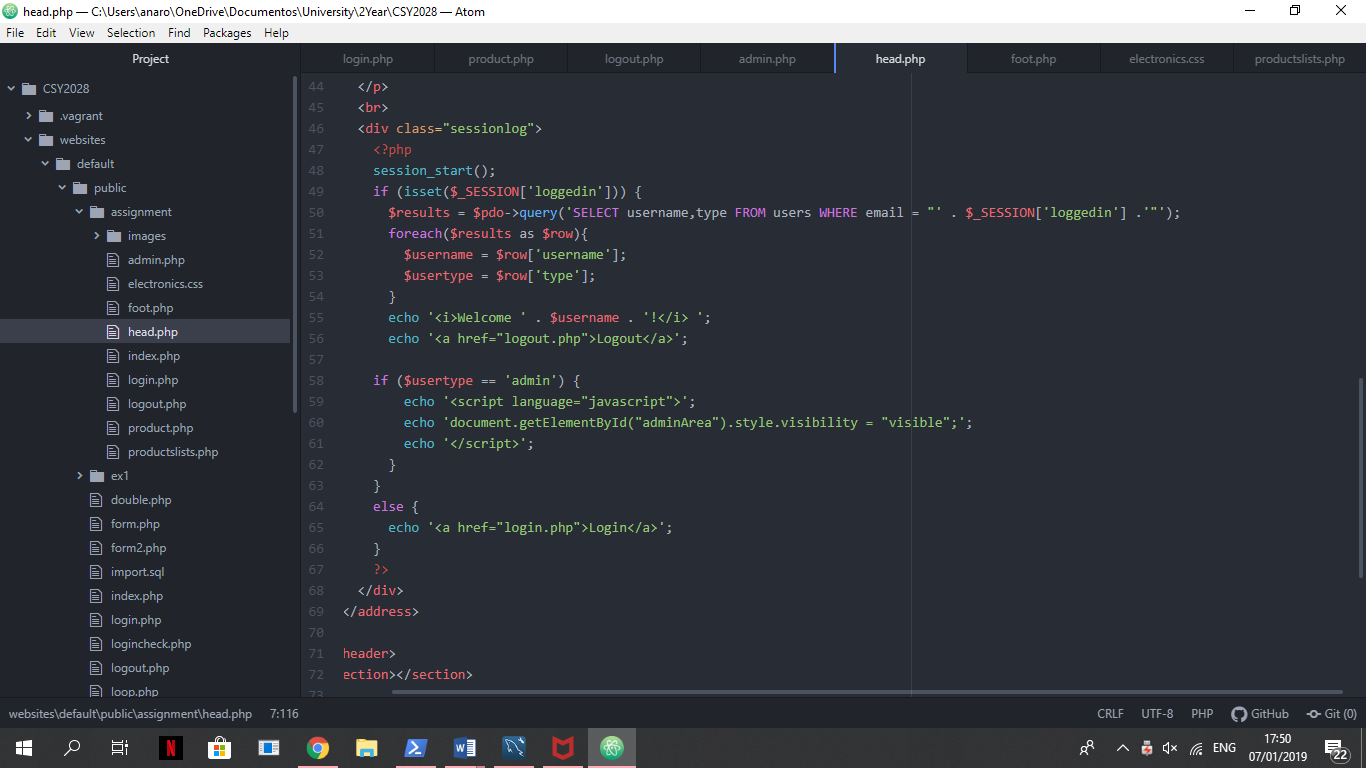
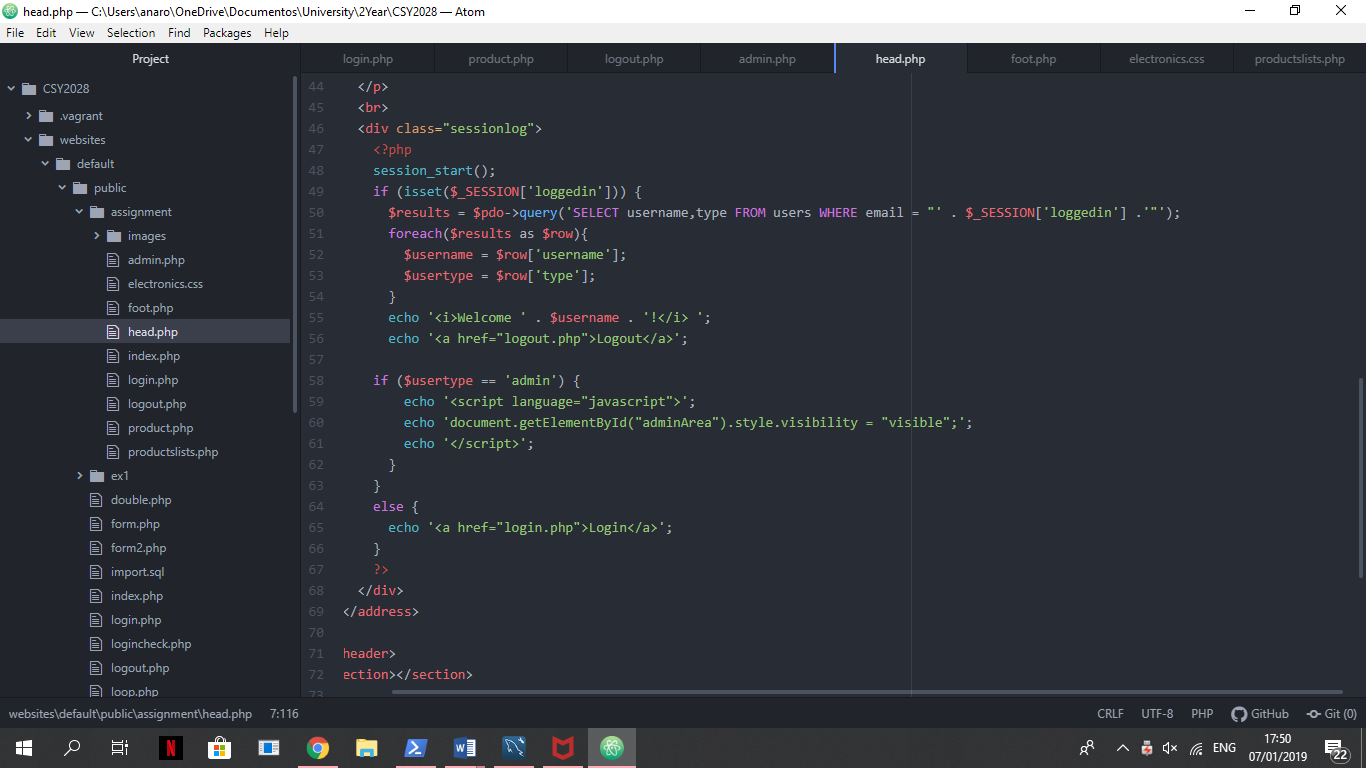
  
As you can see in figure 2 in line 37 another menu option, “Admin area”, is there but doesn’t appear because it visibility is set as hidden. That happens because only admin users are supposed to have access to that page, and to make sure of that the following if statement was made:

Figura – Navegation bar menu

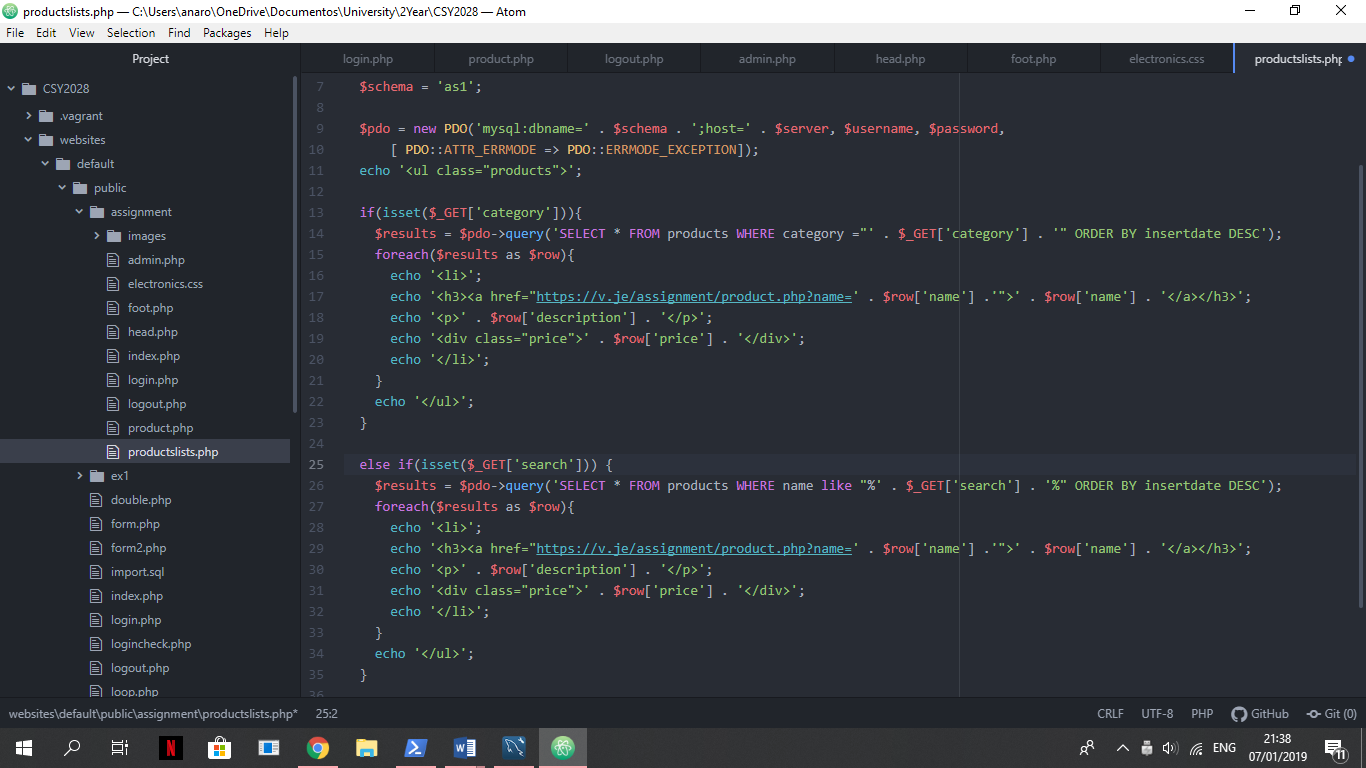
It is also in the head.php that the user status is checked, if is logged in or not, and the user identifier, in this case the email, is saved to be used in other operations.

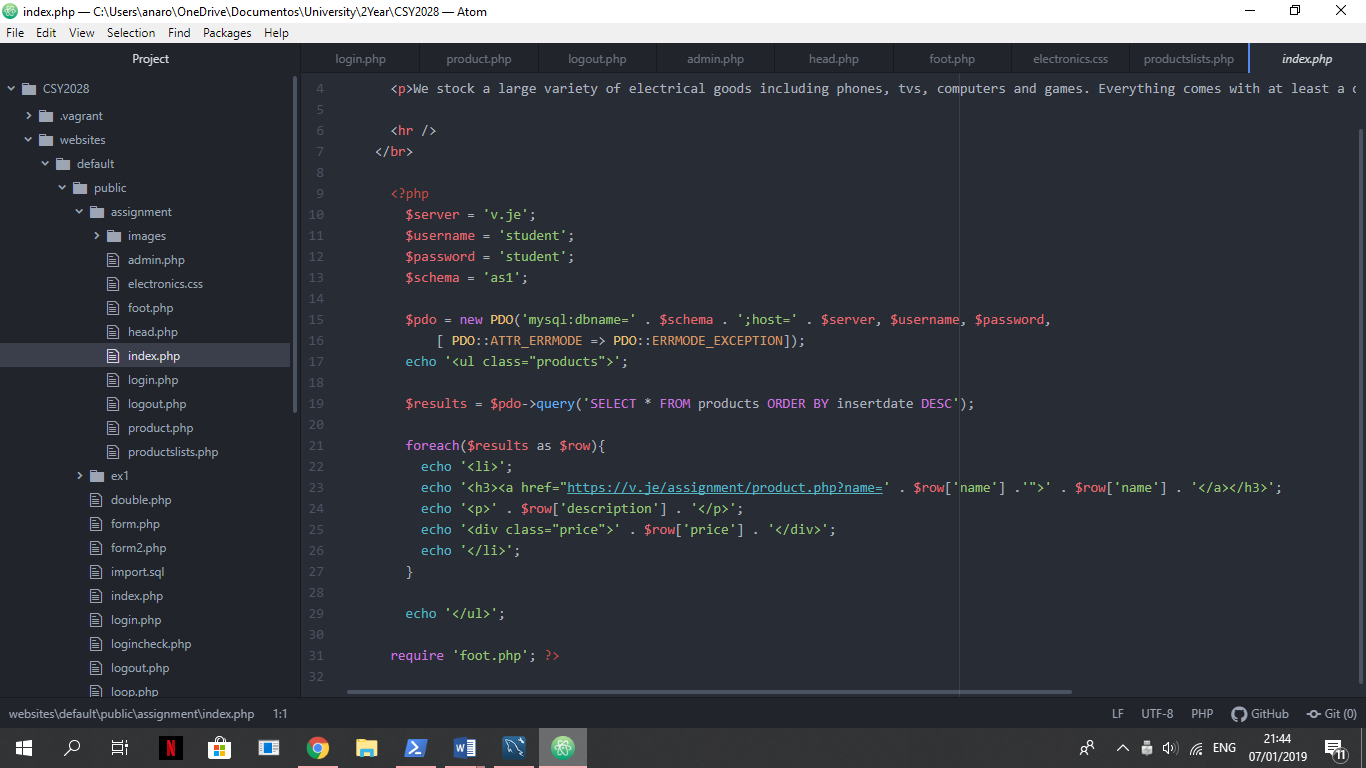
  
In the other end the foot.php contains the code to show the featured product, defined by an admin, in the side of the page and a footer with the site and programmer name. Almost all pages require for the foot.php with exception of the admin.php that as its own footer because contains a form to change the featured product.

If the user is logged in a welcome message and the logout option shows.

Otherwise the user is not logged in so the option to login shows.

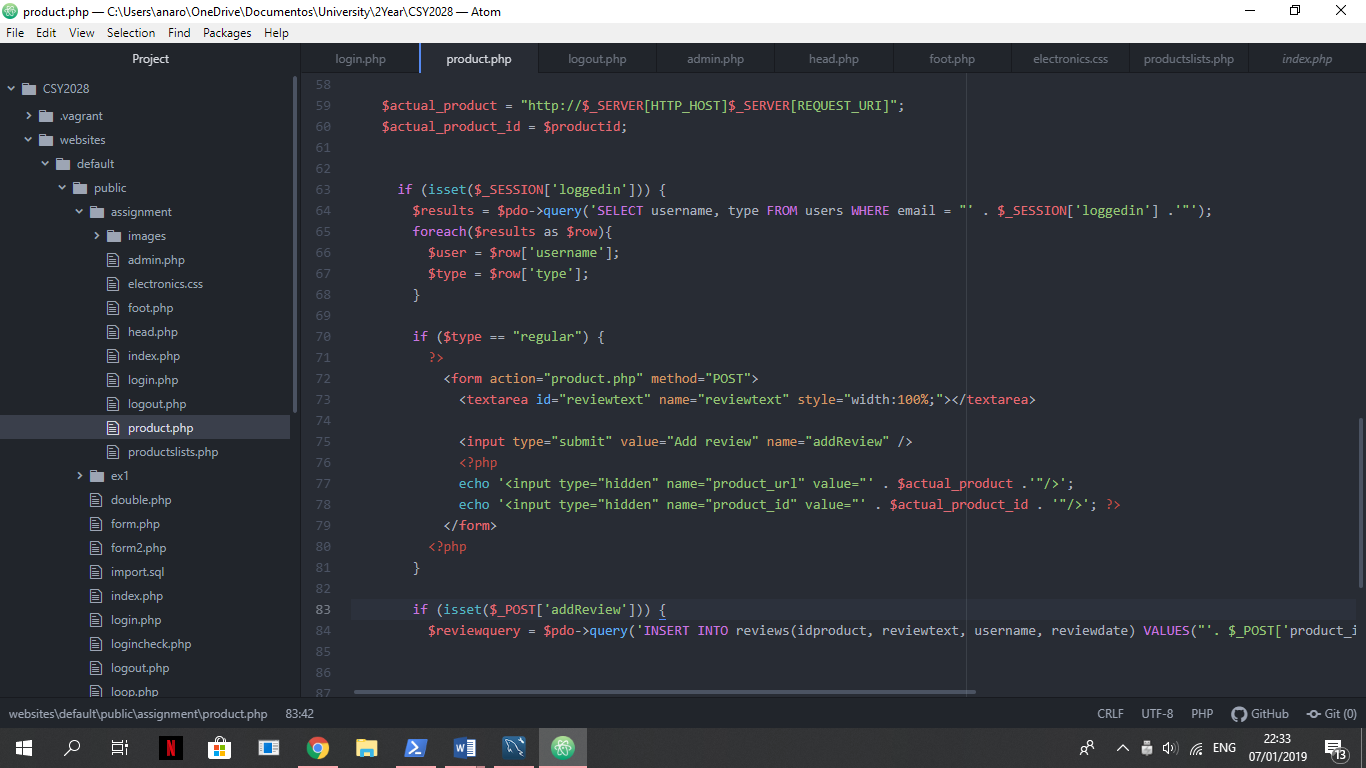
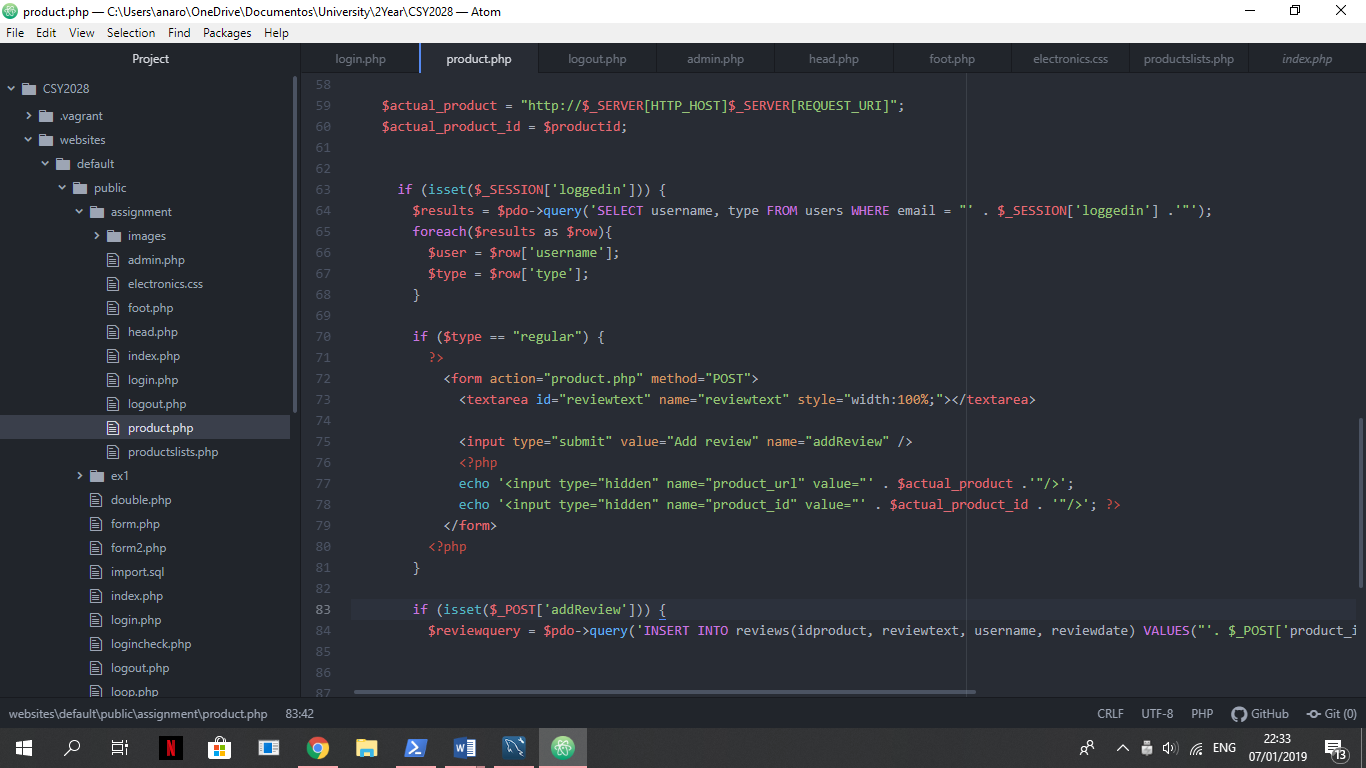
Other pages as index.php and productslists.php have similar code as bout list all products that the store as in stock. The difference between them is in the database query statement, as all the foreach statement is the same.

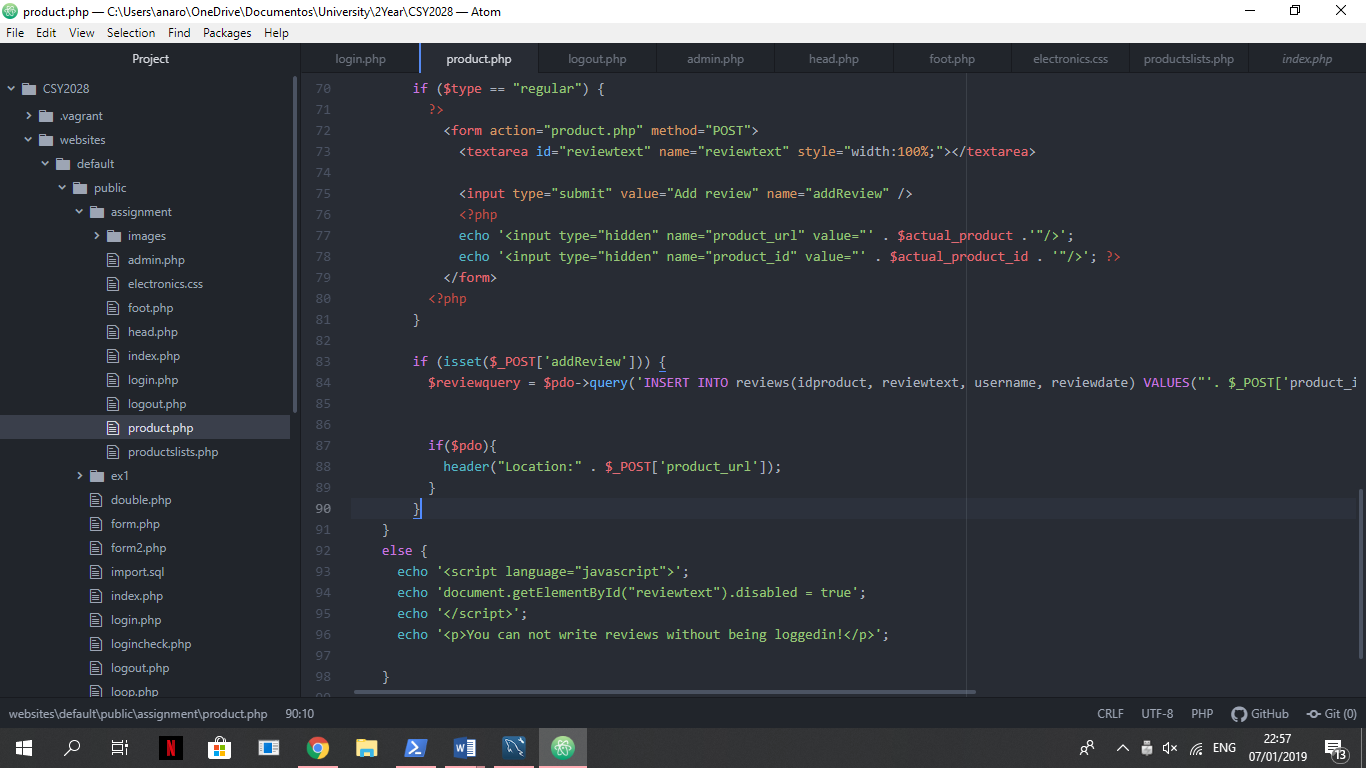
The productslists.php lists the products of a specific category (line 14) or the products that have a name like what was wrote in the searched box (line 26).

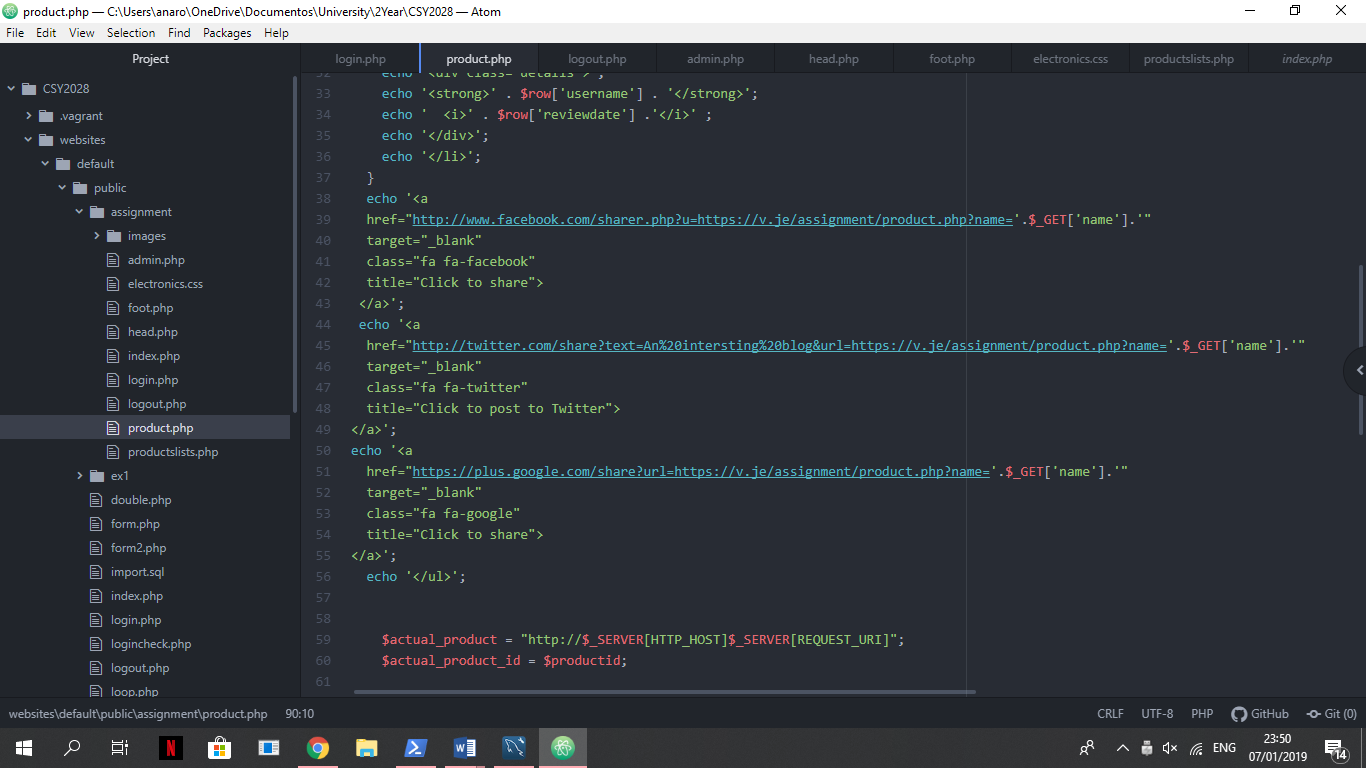
  
And the index.php lists all the products without any restrictions:

The product page (product.php) is meant to show a specific product information and its reviews. In this page the public is also able to add reviews to that product but only if they are logged in as the user id needs to be saved with the review.

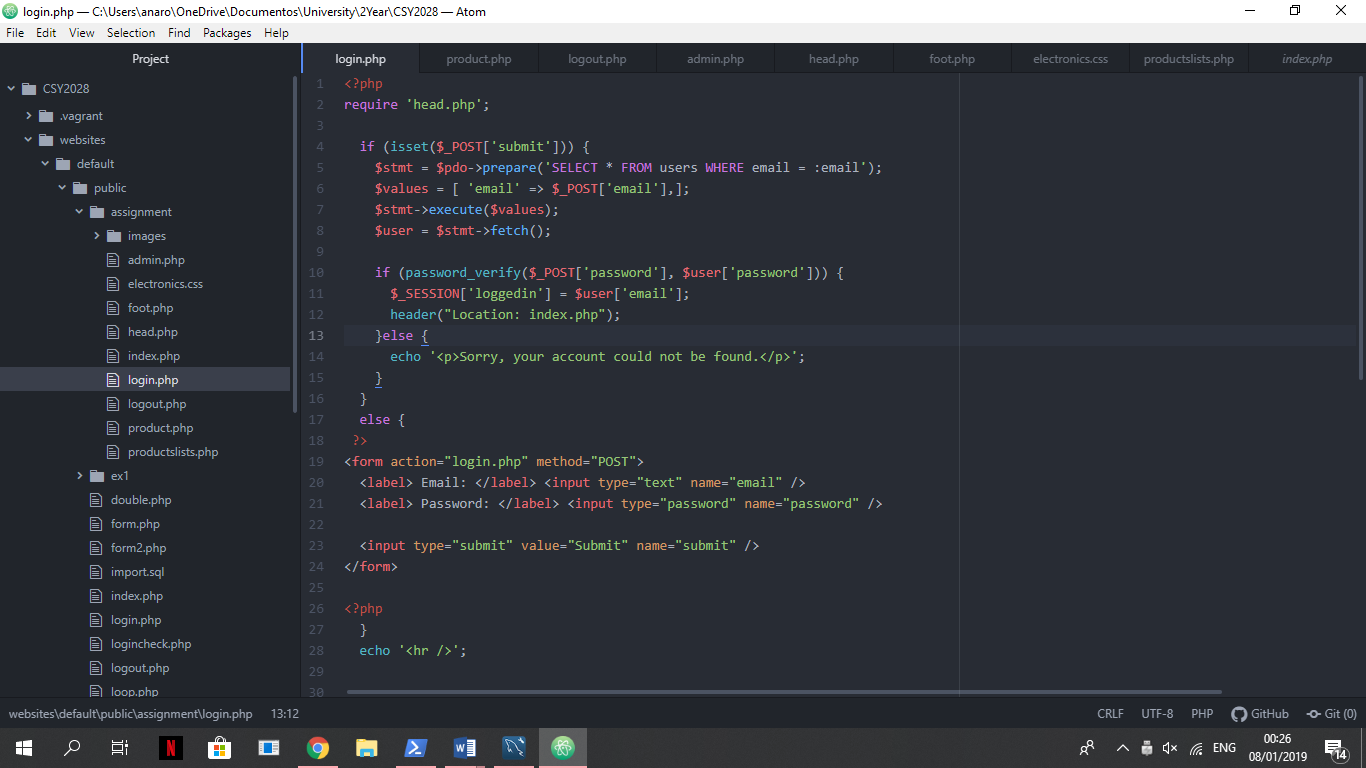
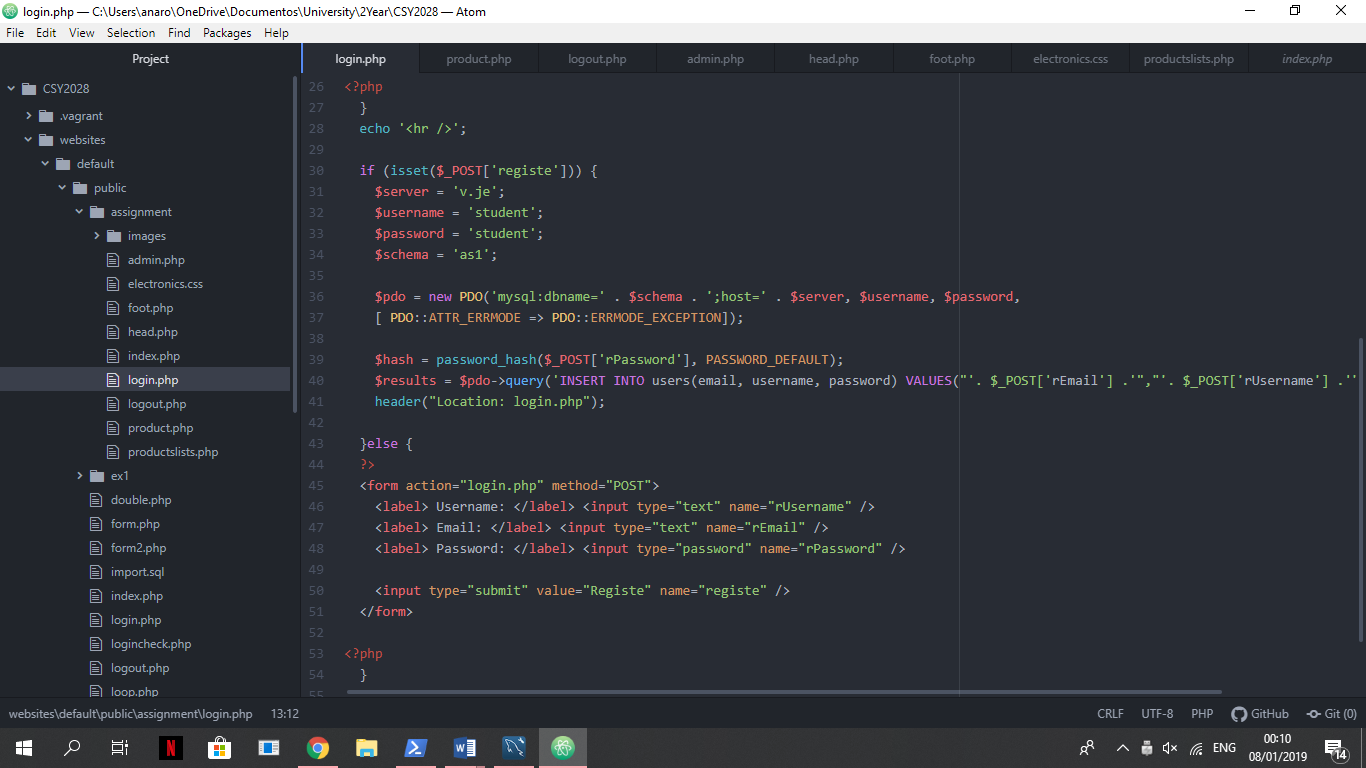
The form for add reviews was a bit different as it had to save the current product id so that the review can be associated to that product and the product url so that we can refresh the page after adding a review.

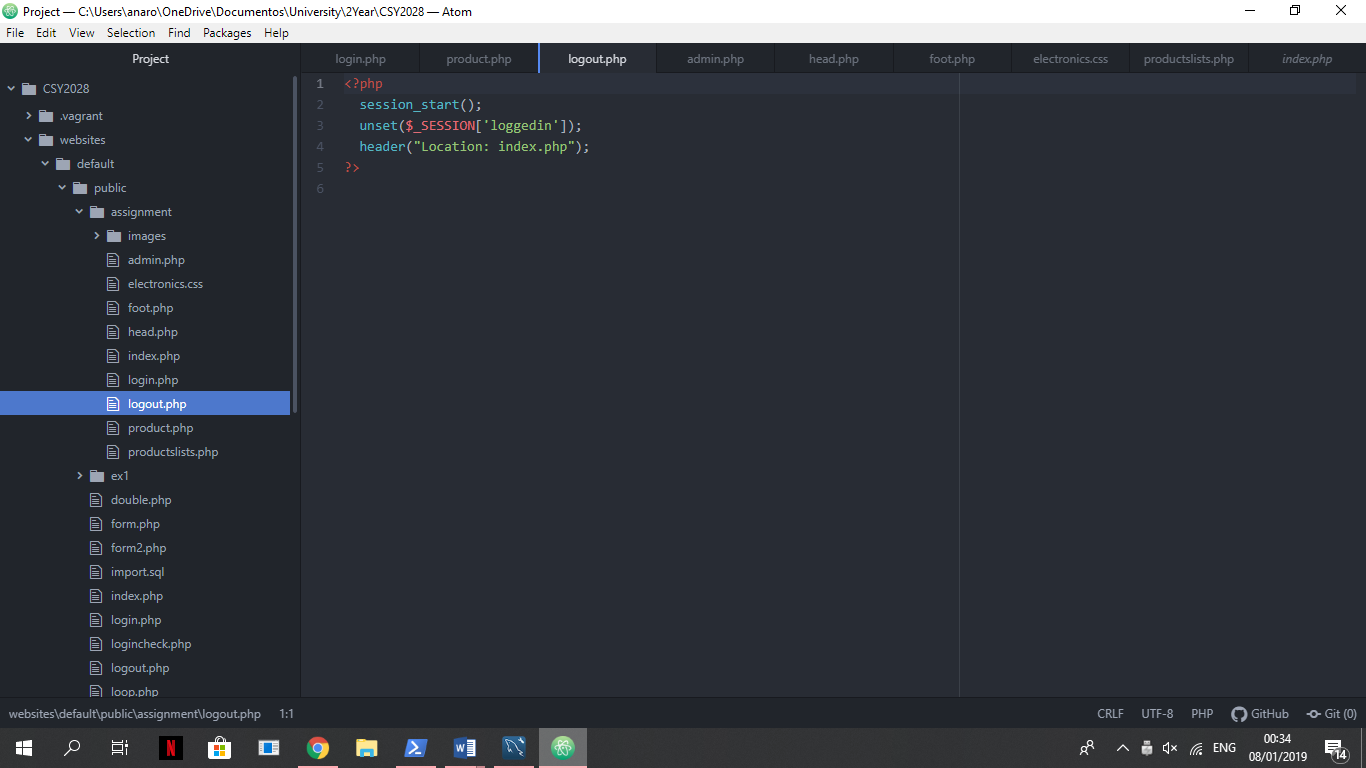
  
To get the current page url the following code was used:

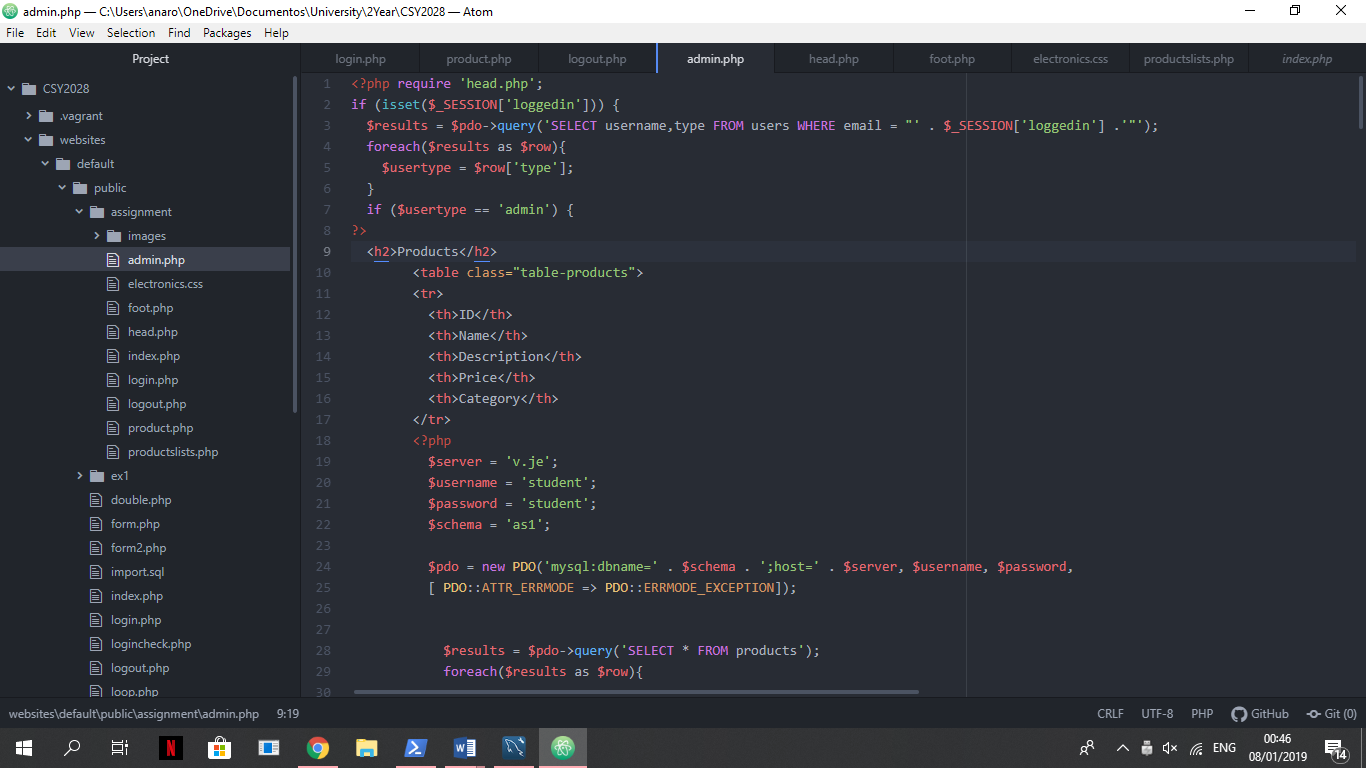
And later, after inserting into the database the review, the following code was added to refresh the page:

In the review area there is also a share button that allow people to share the page on facebook. The code was taken from a website, that is referenced in the references session of this document.

The login page (login.php) allows users to register into the website or login in they account, if they are already register. The register form is a simple form with three inputs (username, email, password) and a submission button. When the submission button is pressed, because one of the inputs is a password and we don’t want that password to be visible to everybody that has access to the database, the following method was used:

  
That way the database stores the password in a codified way. Is also in this page that the login is executed by checking if the inputted email and password match the information in the users table of the database.

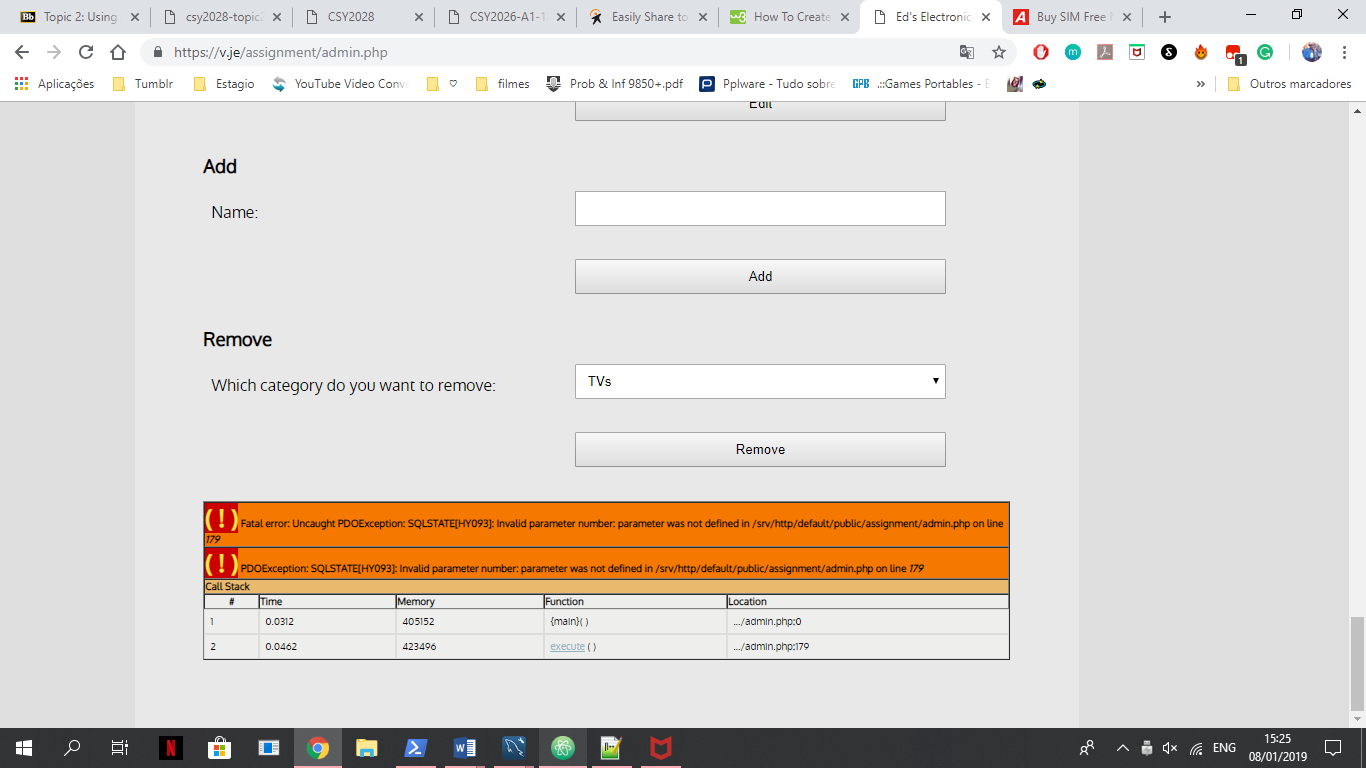
Unlike the login page, the logout page is the website simplest page as only contains the statements to ends the session started before by the login page and redirects to the homepage.

The last and most important page for admin proposes is the admin.php. That page should only be accessed by users identified as administrator and to make sure of that this page starts with the following code:

In the admin.php the admin can add, remove and alter products and categories. So, it was easier to read I opted to put all the forms for the products first and then all the categories forms and the submissions buttons actions all together before the footer of the page. And to make it easier to check the alterations a table of products and categories is show before the forms.

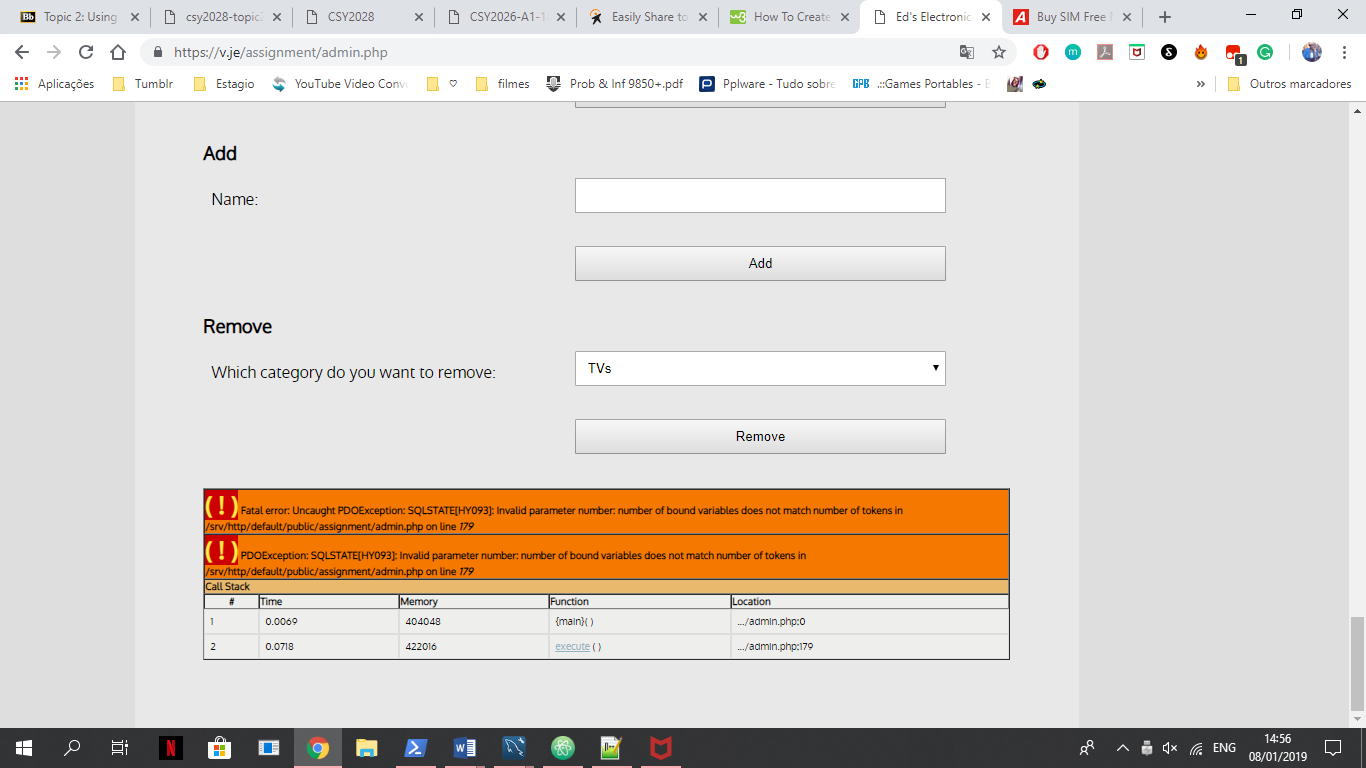
# Testing

Throughout the development of this project a lot of errors occurred, mainly whit the database queries. One of the queries that was more difficult was the inserts because wrong columns names, values names or datatype or missing quotes/comas make the page show this type of errors:

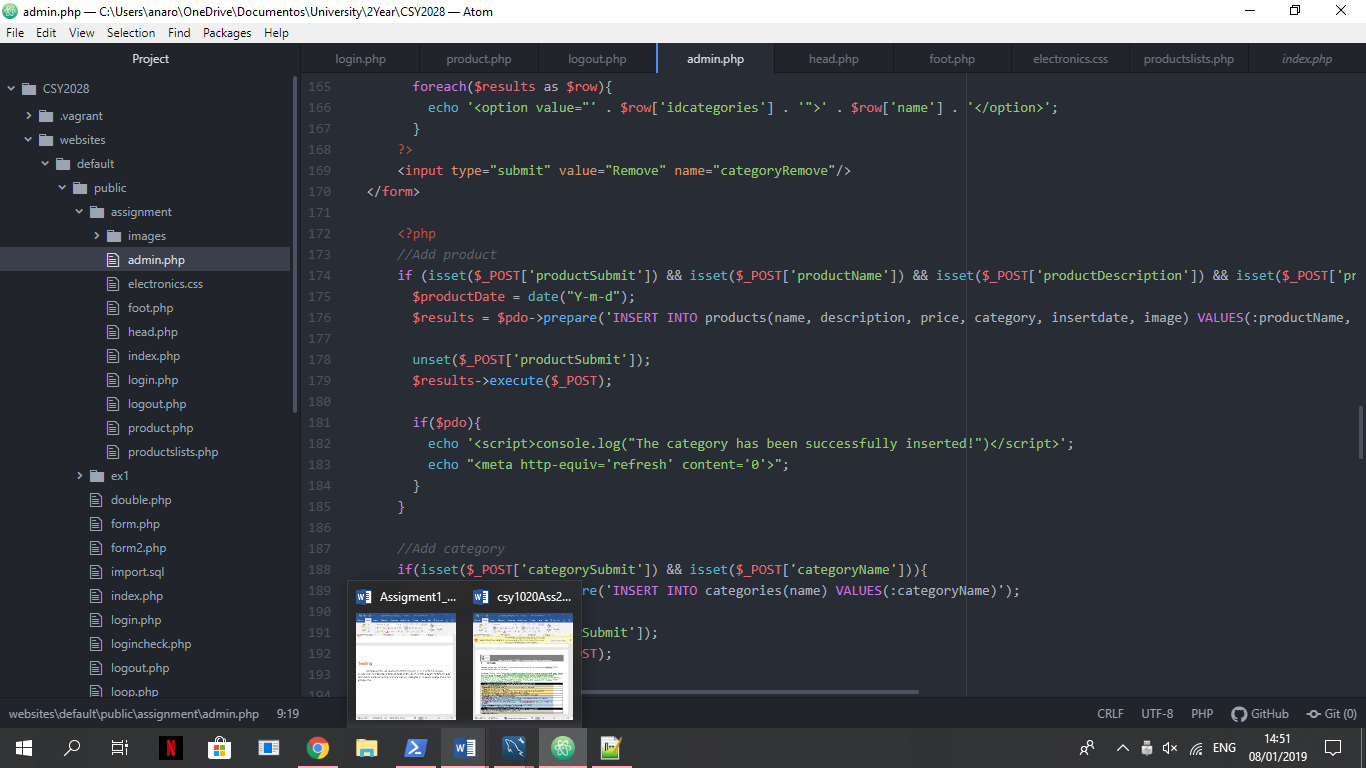


To solve that type of errors we have to check all the form inputs names, database columns and table names and then the submit button action code to make sure all the names match.

In the following image we can see what happens when we don’t unset the submission button of a form.



To resolve this an unset statement is needed before executing the SQL query.



Another error I currently have is with the share buttons in the review session of the product.php. I used three buttons, one for Facebook, other for Twitter and another one for Google+, the problem is that the only one that is working is the Facebook one as the other two say that the URL is not valid. Because of that I end up living just the facebook one there.

# References

How TO - Social Media Buttons. Available from: <https://www.w3schools.com/howto/howto_css_social_media_buttons.asp>

Easily Share to Social Media using Links. Available from: <https://www.pontikis.net/blog/easily-share-to-social-media-using-links>

Thomas Butler. Modules activities topics 2,6,8 and 9. Available from: <https://nile.northampton.ac.uk/>