

|  |  |  |  |
| --- | --- | --- | --- |
| Course Code | : | SWE306 | |
| Course Name | : | Programming Elective II (1) | |
| Lecturer | : | Xiao Han | |
| Academic Session | : | 2020/09 | |
| Assessment Title | : | Project Report | |
| Submission Due Date | : | 15 January 2021 | |
| Prepared by |  |  | |
|  | Student ID | Student Name |
|  | SWE1809651 | Khor Jia Cheng |
|  | SWE1809654 | Leong Wei Jun |
|  | SWE1809659 | Tan Ting Yi |
|  |  |  |
|  |  |  |
|  |  |  |
| Date Received | : |  | |
|  |  |  | |
| Feedback from Lecturer:  Mark: | | | |

**Own Work Declaration**

We hereby understand our work would be checked for plagiarism or other misconduct, and the softcopy would be saved for future comparison(s).

We hereby confirm that all the references or sources of citations have been correctly listed or presented and we clearly understand the serious consequence caused by any intentional or unintentional misconduct.

This work is not made on any work of other students (past or present), and it has not been submitted to any other courses or institutions before.

Signature:  *Leong, Khor, Tan*

Date: 14 January 2021

**Table of Contents**

[**1.0 Title 5**](#_Toc61491367)

[**2.0 Abstract 5**](#_Toc61491368)

[**3.0 Introduction 5**](#_Toc61491369)

[**4.0 Code Architecture 6**](#_Toc61491370)

[**5.0 Case Study 8**](#_Toc61491371)

[**5.1 Registration 8**](#_Toc61491372)

[**5.1.1 Picture of Registration Page of X-Trade 8**](#_Toc61491373)

[**5.1.2 Registration for X-Trade 9**](#_Toc61491374)

[**5.2 Log In 10**](#_Toc61491375)

[**5.2.1 Picture of Login Page of X-Trade 10**](#_Toc61491376)

[**5.2.2 Login for X-Trade 11**](#_Toc61491377)

[**5.3 Browse Product 12**](#_Toc61491378)

[**5.3.1 Picture of All Product Page of X-Trade 12**](#_Toc61491379)

[**5.3.2 Picture of Books and Stationery Page of X-Trade 12**](#_Toc61491380)

[**5.3.3 Picture of Food and Beverage Page of X-Trade 13**](#_Toc61491381)

[**5.3.4 Picture of Household Appliance Page of X-Trade 13**](#_Toc61491382)

[**5.3.5 Picture of Clothes and Accessories Page of X-Trade 14**](#_Toc61491383)

[**5.3.6 Picture of Others Page of X-Trade 14**](#_Toc61491384)

[**5.3.7 Browsing product in X-Trade 15**](#_Toc61491385)

[**5.4 Upload Product 16**](#_Toc61491386)

[**5.4.1 Picture of Upload Product Page of X-Trade 16**](#_Toc61491387)

[**5.4.2 Upload Product for X-Trade 17**](#_Toc61491388)

[**5.5 View Product Details 18**](#_Toc61491389)

[**5.5.1 Picture of View Product Details of X-Trade 18**](#_Toc61491390)

[**5.5.2 View Product Details for X-Trade 19**](#_Toc61491391)

[**5.6 View Currently Sell Product 20**](#_Toc61491392)

[**5.6.1 Picture of View Product Details of X-Trade 20**](#_Toc61491393)

[**5.6.2 View Currently Sell Product for X-Trade 21**](#_Toc61491394)

[**5.7 Delete Product 22**](#_Toc61491395)

[**5.7.1 Picture of Delete Product in Currently Sell Product Page 22**](#_Toc61491396)

[**5.7.2 Delete Product for X-Trade 23**](#_Toc61491397)

[**5.8 Change User Account Settings 24**](#_Toc61491398)

[**5.8.1 Picture of Change User Account Setting Page of X-Trade 24**](#_Toc61491399)

[**5.8.2 Change User Account Setting in X-Trade 25**](#_Toc61491400)

[**5.9 View Seller’s Profile 26**](#_Toc61491401)

[**5.9.1 Picture of View Seller Profile Page of X-Trade 26**](#_Toc61491402)

[**5.9.2 View Seller Profile Page in X-Trade 27**](#_Toc61491403)

[**5.10 View Product Sold by Particular Seller 28**](#_Toc61491404)

[**5.10.1 Picture of View Seller Profile Page of X-Trade 28**](#_Toc61491405)

[**5.10.2 View Seller Profile Page in X-Trade 29**](#_Toc61491406)

[**5.11 View Frequent Asked Question 30**](#_Toc61491407)

[**5.11.1 Picture of View Seller Profile Page of X-Trade 30**](#_Toc61491408)

[**5.11.2 View Frequent Asked Questions in X-Trade 31**](#_Toc61491409)

[**5.12 Log Out 32**](#_Toc61491410)

[**5.11.1 Picture of Log out of X-Trade 32**](#_Toc61491411)

[**5.12.2 Log out in X-Trade 33**](#_Toc61491412)

[**6.0 Difficulties Faced during the Implementation of Project 33**](#_Toc61491413)

[**6.1 Difficulties in using Flexbox as a Container 33**](#_Toc61491414)

[**6.2 Difficulties in Implementation for every Actor 33**](#_Toc61491415)

[**6.3 Difficulties in Creating Inbox 34**](#_Toc61491416)

[**6.4 Difficulties in Display Image which in Local Computer in JSP File 34**](#_Toc61491417)

[**6.5 Difficulties in Implementing Varieties of Payment Method 34**](#_Toc61491418)

[**6.6 Difficulties in Creating Responsive Website 34**](#_Toc61491419)

[**7.0 Future Improvement 34**](#_Toc61491420)

[**7.1 Create Inbox to Connect Buyer and Seller 34**](#_Toc61491421)

[**7.2 Create Responsive Website 34**](#_Toc61491422)

[**7.3 Increase Variety of Payment Method 35**](#_Toc61491423)

[**We hope to increase variety ways of payment method like Touch n-go, WeChat Pay, and e-Banking and also student card. 35**](#_Toc61491424)

[**8.0 Conclusion 35**](#_Toc61491425)

[**9.0 Work Breakdown 36**](#_Toc61491426)

[**10.0 Reference 37**](#_Toc61491427)

# Title

The title of our project will be named as **X-Trade**, which is an online platform which allows the staffs and students of XMUM to trade their products.

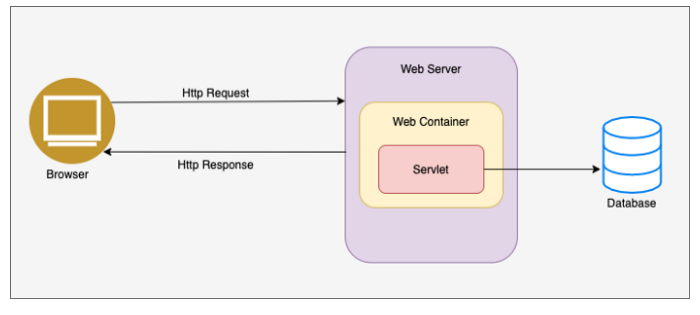
# 2.0 Abstract

The project is a Java EE website which is named as **X-Trade** which comprises HTML, CSS, Java, Java EE and SQL language. The website uses HTML and CSS to display the website appearance while Java EE is used to make the website interactive. SQL is used to query the MySQL database to retrieve the information and display them on the website. Besides that, our website is mainly designed for the university’s students to do trading of needs in academic and daily life in a virtual marketplace. Furthermore, client-server architecture is used in this project to display products dynamically according to the user's need. To implement client-server architecture, Apache Tomcat and Servlet are used and it will be explained further in the code architecture section.

# 3.0 Introduction

Since there are many social media’s second-hand trading market groups for Xiamen University Malaysia (XMUM) but no website developed for our university, our group decides to develop a website for XMUM’s students for our students and lecturer’s convenience. For this Java EE project, a second-hand trade market website is developed and built for students. The user can register through their unique student id as only XMUM students can use this website. In this website, any registered user can be a buyer and seller. However, they can only pay through face-to-face methods. They can upload the product’s picture, product’s description and initial price to the website. Therefore, the interested buyer can contact the seller to know more details about the website. Furthermore, the website creates a platform which collects second-hand things for sale and categorizes them according to five main categories which are clothes, household appliances, books & stationery, food & beverage and others. Therefore, the interested user can explore their needs according to category and contact the seller’s through the seller's profile and obtain their contact number. Besides that, the picture is compulsory for the seller to upload it in order to ensure the quality of the product and prevent users from fraud or cheat issues. In X-Trade website, an unregistered user can only browse the home page, product page but they cannot view product details and sell products. Only registered users can sell products and view product details in order to buy products as their identity is verified. Hence, the risk of trading can be minimized. Furthermore, the users can delete their listing product after the product to update that the product has been traded. In order to be a seller, there will be a button floating in the homepage, product pages of the website. By clicking on it, filling in the product details and the product can be listed for the visitor of the website to browse. There is also a Frequently Asked Question (FAQ) page in our website which mainly guides new registered website how to use our website.

# 4.0 Code Architecture



***Figure 4*** *Client Server Architecture*

For our project, X-Trade website uses client-server architecture which is shown in the Figure 4 above. From the Figure 4 above which shows the client-server architecture, the user sends HTTP requests to the web server. The web server has the web container containing servlet, which gathers data from the database and creates a response. Then, the response is created by the servlet and is sent through HTTP Response to the client browser so that the user can see the content they want. In our project, Java Servlets are used and Apache Tomcat is used as a web container and MySQL is used as a database in the X-Trade website. The HTML and CSS pages are converted to a JSP file in order to use Servlets while having a good appearance for the website.

Java Servlets are software components that run on a web server and are responsible to receive the request from the web server, process the request, and respond back to the server. The Servlets extend the capabilities of a server as they can respond dynamically and to many types of requests. They act like web containers for hosting web applications on web servers. They do support client-server protocols and are always used with HTTP and also known as HTTP servlet which is also known as HTTP servlet. For web containers, it is also known as a servlet container. It provides the runtime environment for Java EE (j2ee) applications. The web container is used in Java for the dynamic web pages. Therefore, if the user wants to read the web pages as per input the web container is needed. It implements the Servlet API and the services required to process HTTP requests. Web container initiates the Servlet that matches the requested URL by invoking the service () method of Servlet class. The service () method which is invoked for the given HTTP request is handled in a separate thread within the web container protocol.

In the process of the user browsing the X-Trade website, for example, the user wants to browse the all products available in X-Trade website, the user sends an HTTP Request to the web server that it requests “AllProduct.jsp”. Then, the web server forwards requests to the web container which is Apache Tomcat. Then, the web container forwards the request to the Servlet which is connected to the MySQL database which is named as “ProductDisplayServlet.java” in the form of the request object. Servlet retrieves the information from the database and builds the response object and sends it back to the web container. Web container transforms the response object to equivalent HTTP response and sends it to the web server. The web server sends the response via HTTP response back to the user. Lastly, the user is able to see all products listed on X-Trade website. This is the process of displaying all products to the user.

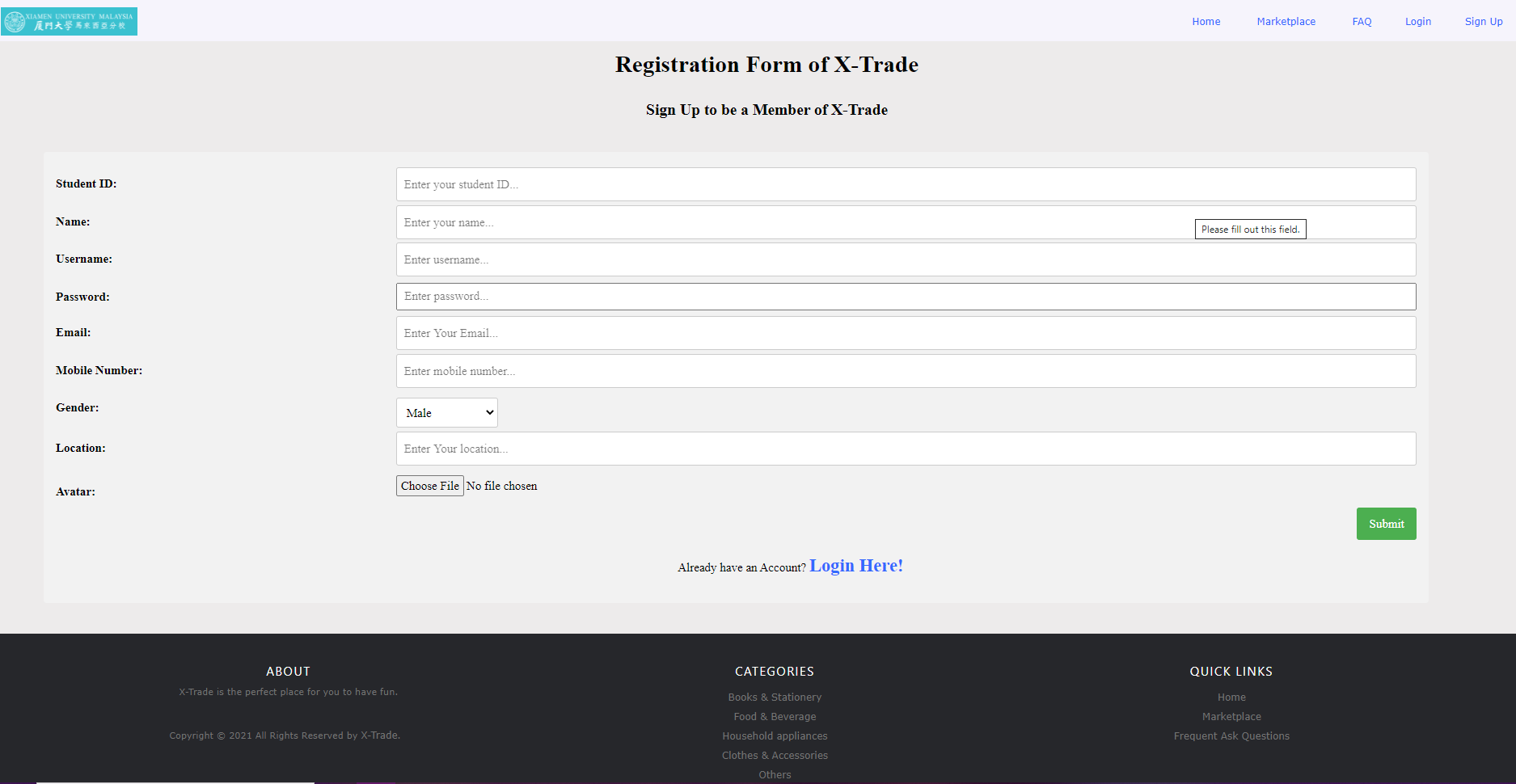
To make the web server, web container and servlet work together, the concurrent HTTP requests coming to the server are forwarded to the web container. Then the web container will initiate the servlet before invoking the service () method. The initiation process is invoked when the servlet is created when the user first invokes a URL corresponding to the servlet. Then, the web container handles multiple requests by spawning multiple threads, each thread executing the service () method of a single instance of the servlet to perform the actual task, requests coming from the user. After the servlet is used, it will invoke the destroy () method which gives the servlet to close MySQL database connections and halt background threads (Kothari, 2020).

This code architecture making X-Trade website can display content dynamically for the user.

# 5.0 Case Study

## 5.1 Registration

### 5.1.1 Picture of Registration Page on X-Trade



***Figure 5.1*** *Picture of Registration Page on X-Trade*

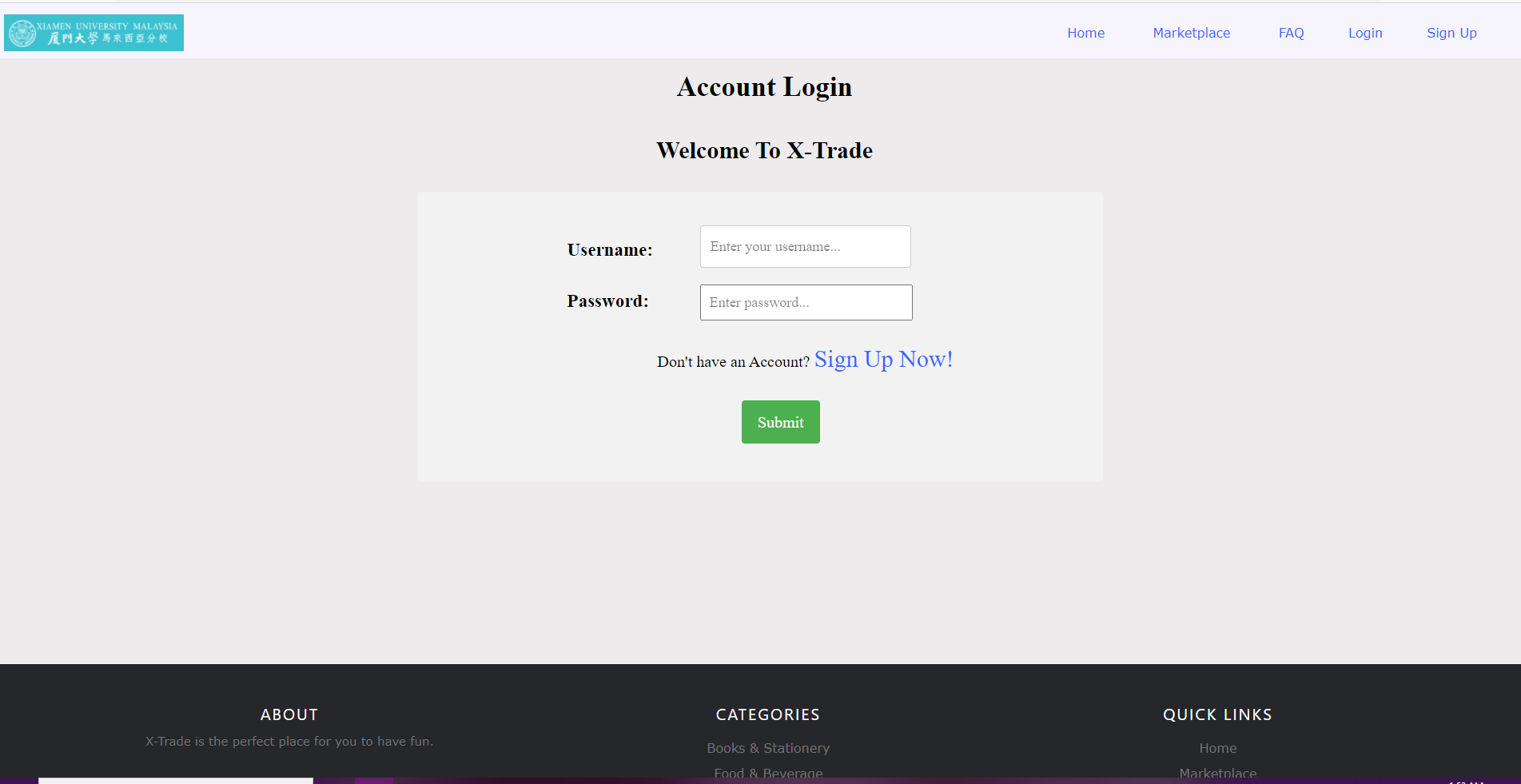
### 5.1.2 Registration for X-Trade

|  |  |
| --- | --- |
| **Functionality** | Registration |
| **Functionality Description** | The user wants to register. |
| **Pre-condition** | The user is unregistered. |
| **Post-condition** | The user is registered. |
| **Basic events** | 1) An unregistered user enters the X-Trade website.  2) The user clicks on the “register” on the menu bar.  3) The user fills in the registration form and fills in his student ID, name, username, password, email, mobile number, gender, location and profile picture.  4) The user clicks the “submit” button.  5) The user has registered successfully. |

***Table 5.1*** *Registration for X-Trade*

## 5.2 Log In

### 5.2.1 Picture of Login Page on X-Trade



***Figure 5.2*** *Picture of Login Page on X-Trade*

### 5.2.2 Login for X-Trade

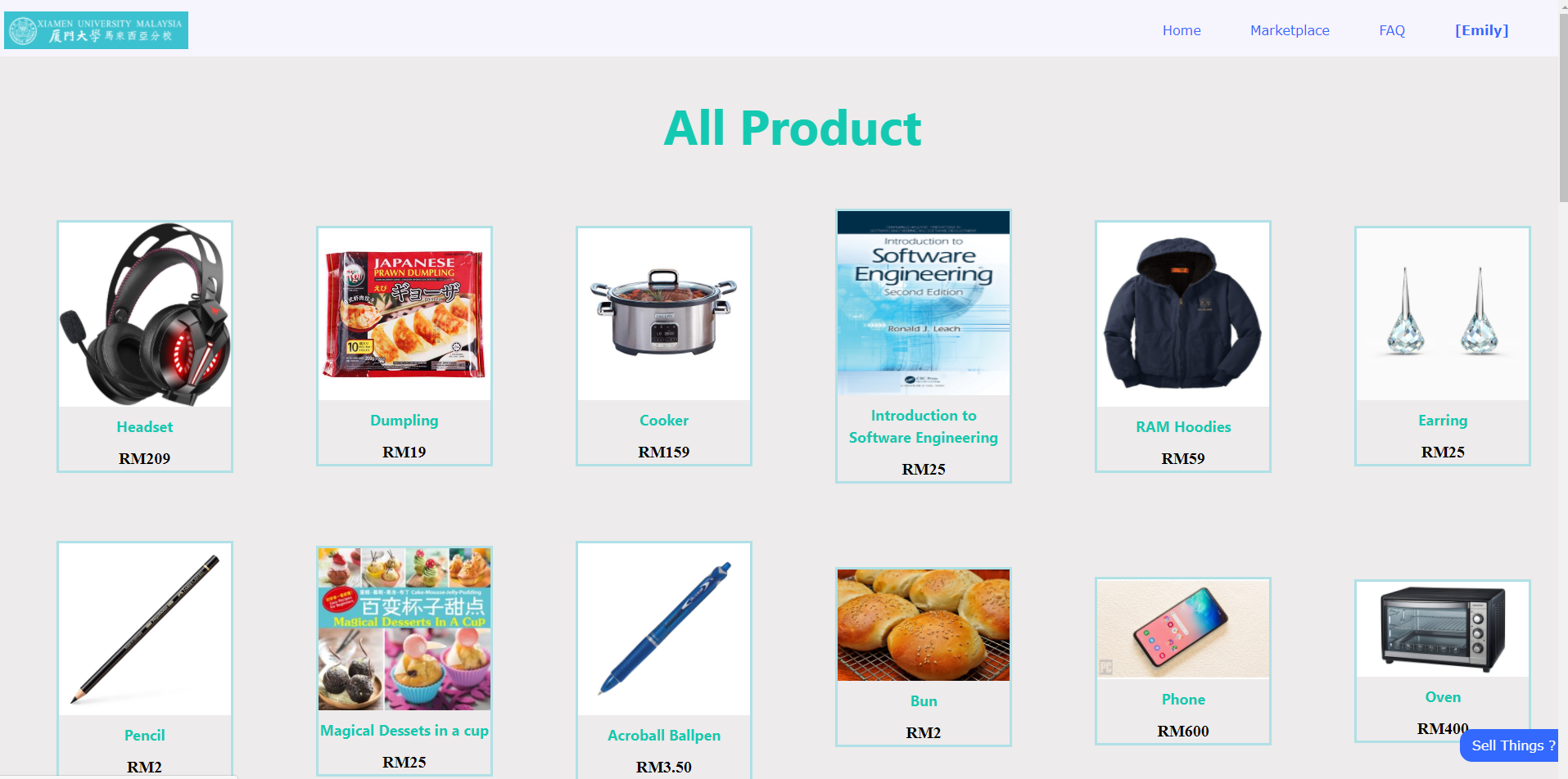
|  |  |
| --- | --- |
| **Functionality** | Log in |
| **Functionality Description** | The user wants to log in. |
| **Pre-condition** | The user is a registered user. |
| **Post-condition** | The user successfully logs in to his/her account. |
| **Basic events** | 1) An registered user enters the X-Trade website.  2) The user clicks on the “Login” on the navigation bar.  3) The user fills in his/her username and password.  4) The user clicks the “submit” button. |

***Table 5.2*** *Login for X-Trade*

## 5.3 Browse Product

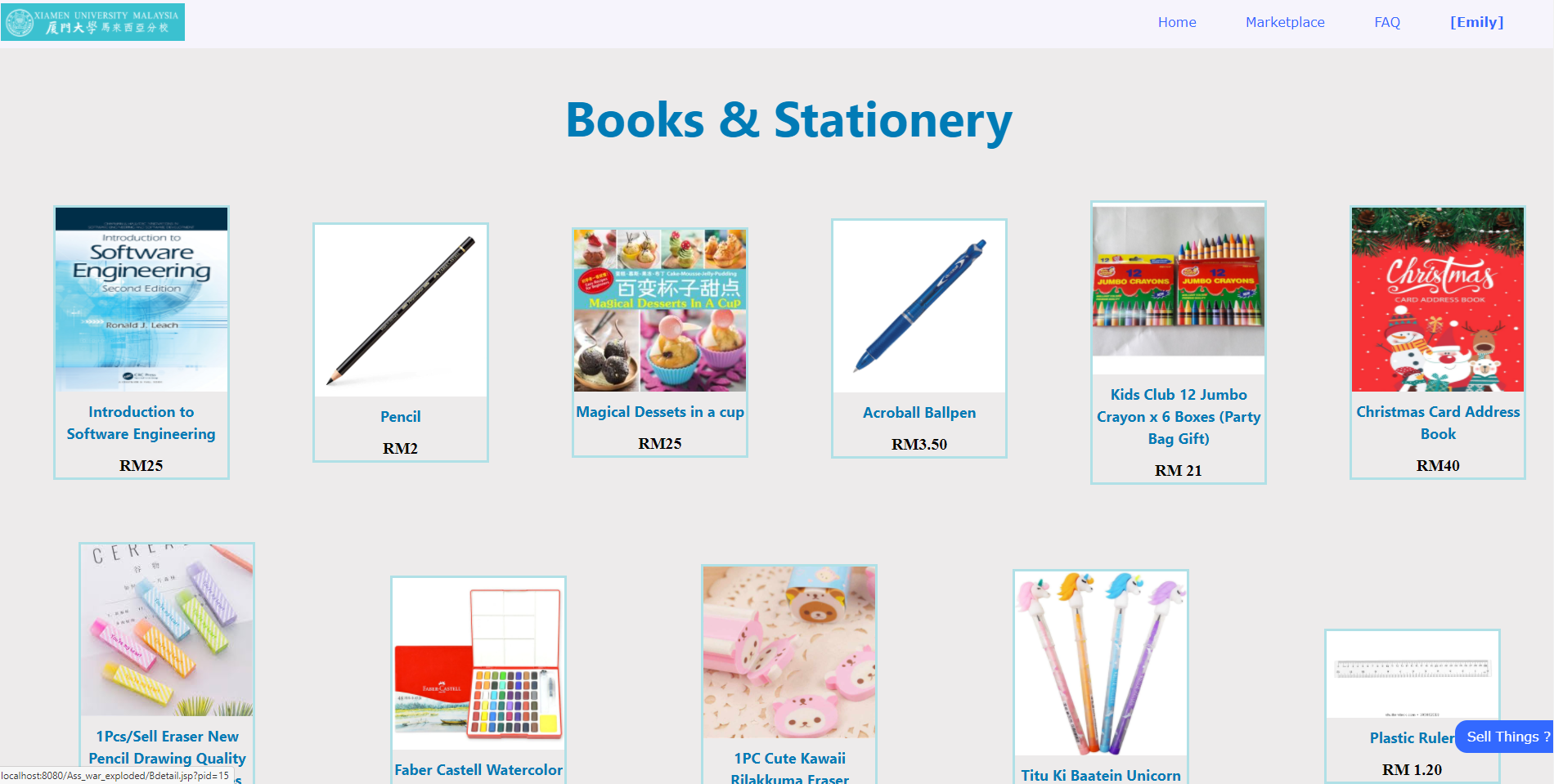
There were a total of **6** product pages which is the **Marketplace** pages which contains all product, **Clothes and Accessories** page, **Food and Beverage** page, **Household Appliance** page, **Books and Stationery** page, and **Others** page.

### 5.3.1 Picture of All Product Page on X-Trade

******

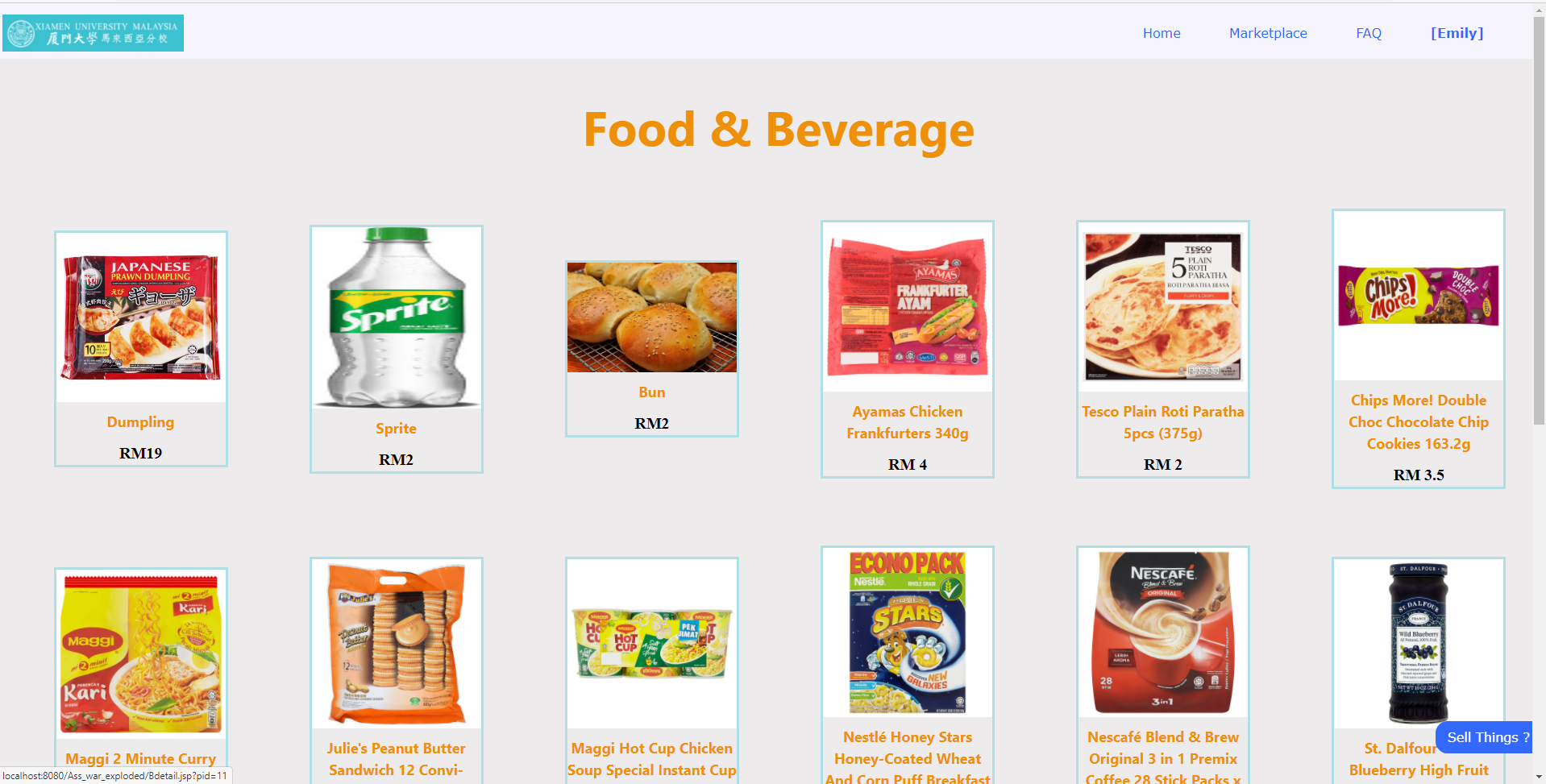
***Figure 5.3.1*** *Picture of Marketplace Page on X-Trade*

### 5.3.2 Picture of Books and Stationery Page on X-Trade

**

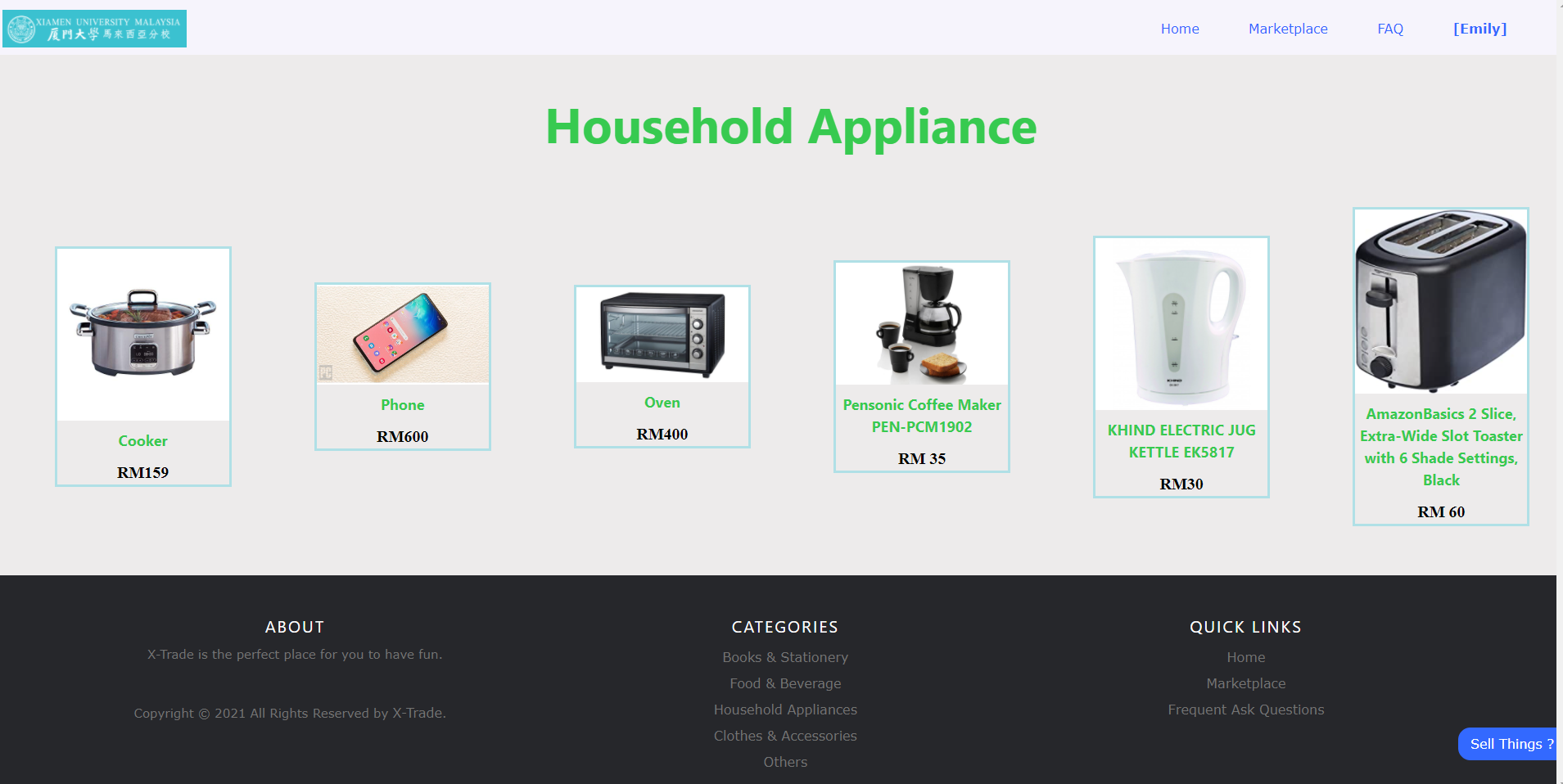
***Figure 5.3.2*** *Picture of Books and Stationery Page on X-Trade*

### 5.3.3 Picture of Food and Beverage Page on X-Trade

**

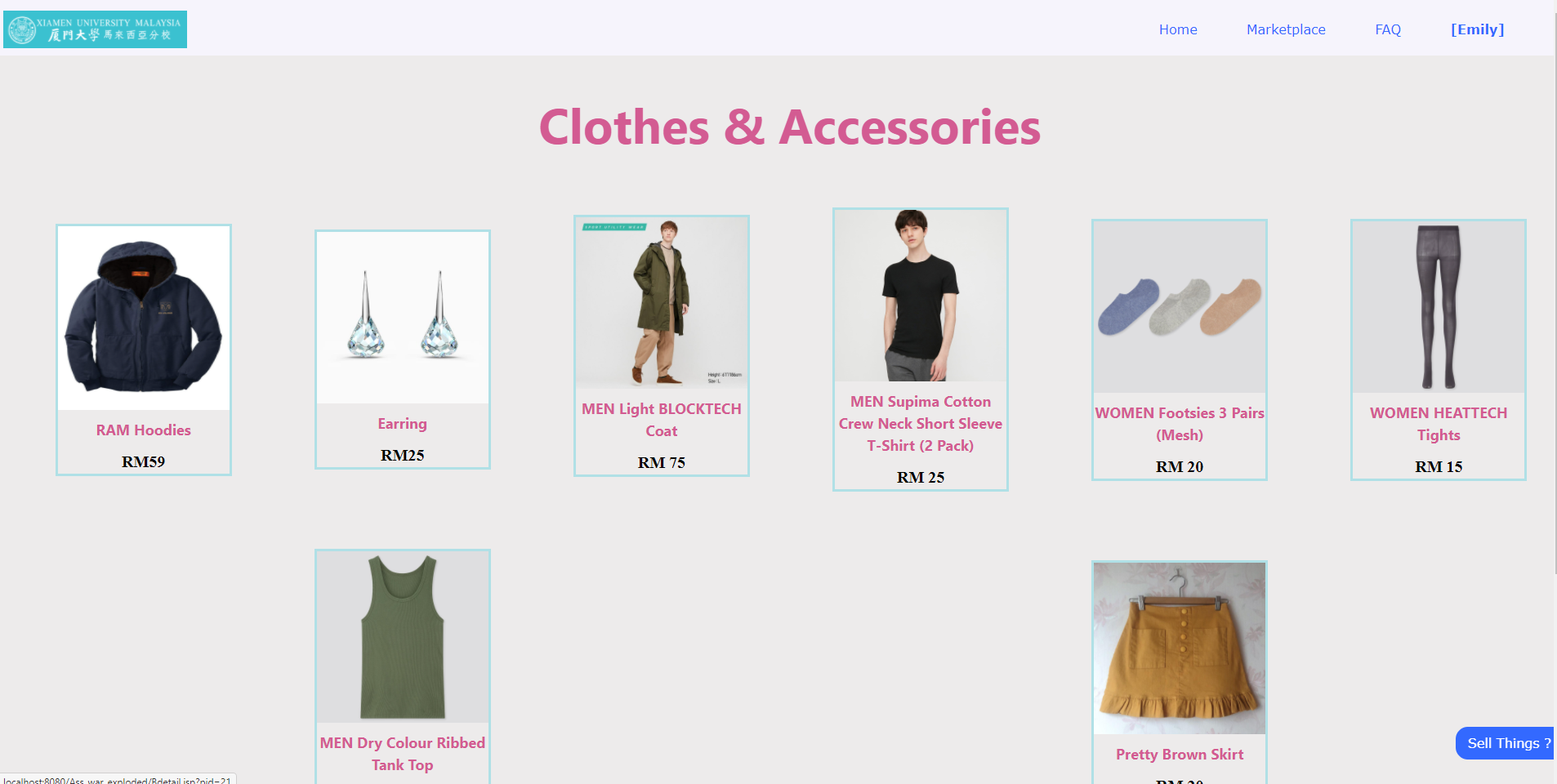
***Figure 5.3.3*** *Picture of Food and Beverage Page on X-Trade*

### 5.3.4 Picture of Household Appliance Page on X-Trade

**

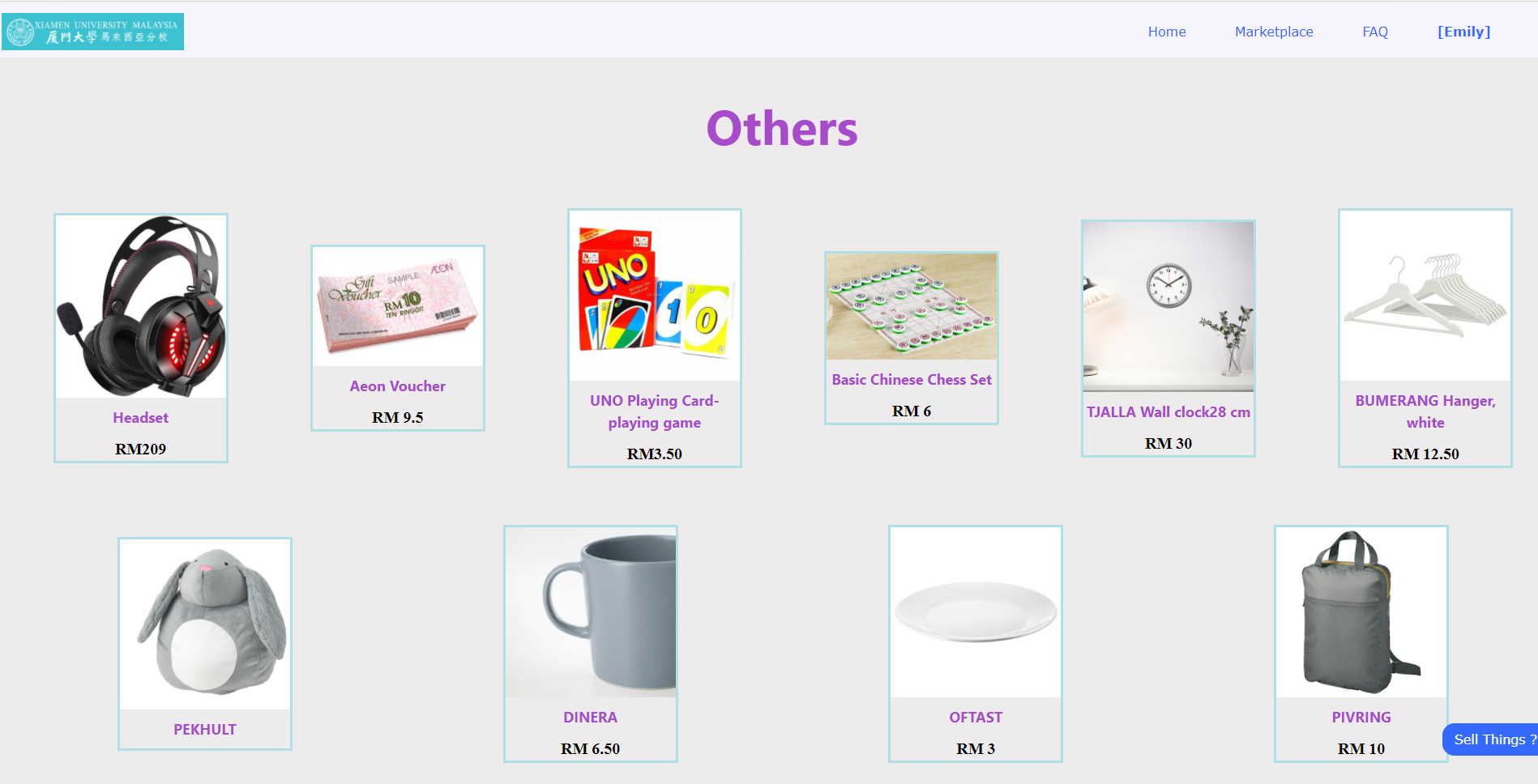
***Figure 5.3.4*** *Picture of Household Appliance Page on X-Trade*

### 5.3.5 Picture of Clothes and Accessories Page on X-Trade



***Figure 5.3.5*** *Picture of Clothes and Accessories* *Page on X-Trade*

### 5.3.6 Picture of Others Page of X-Trade

**

***Figure 5.3.6*** *Picture of Others Page on X-Trade*

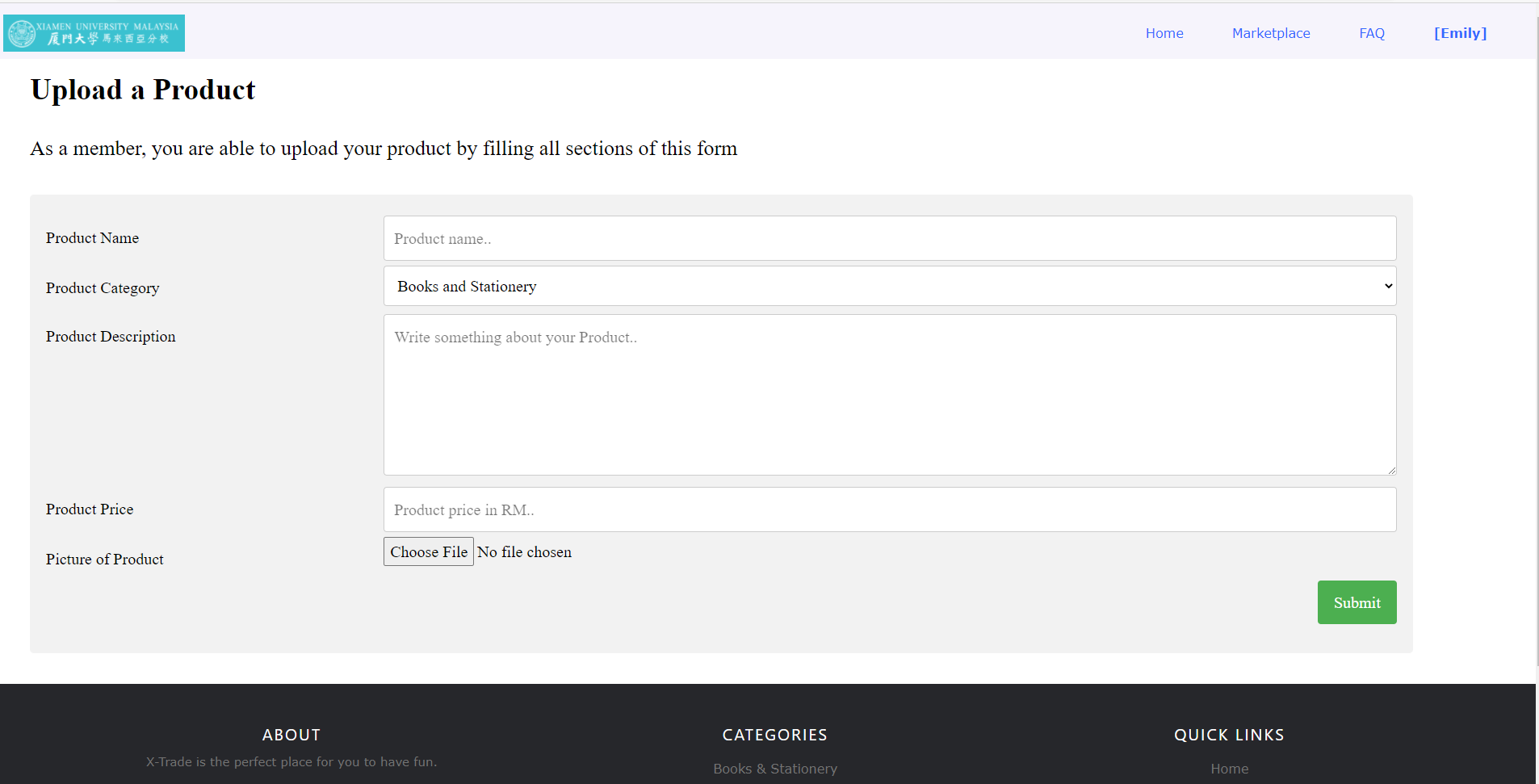
### 5.3.7 Browsing product on X-Trade

|  |  |
| --- | --- |
| **Functionality** | Browse Product |
| **Functionality Description** | The user wants to browse all products on X-Trade website |
| **Pre-condition** | The user browses the X-Trade website. |
| **Post-condition** | The brief information of all products on X-Trade website is displayed to the user. |
| **Basic events** | 1) The user clicks on one of the pages on “Marketplace” on the menu bar.  2)  The list of selected categories of products will be displayed to the user. |
| **Alternative events** | 1. The user clicks one of the drop-down menu on the marketplace section to navigation to particular section of product. |

***Table 5.3*** *Browsing Product on X-Trade*

## 5.4 Upload Product

### 5.4.1 Picture of Upload Product Page on X-Trade

**

***Figure 5.4*** *Picture of upload product page on X-Trade*

### 5.4.2 Upload Product on X-Trade

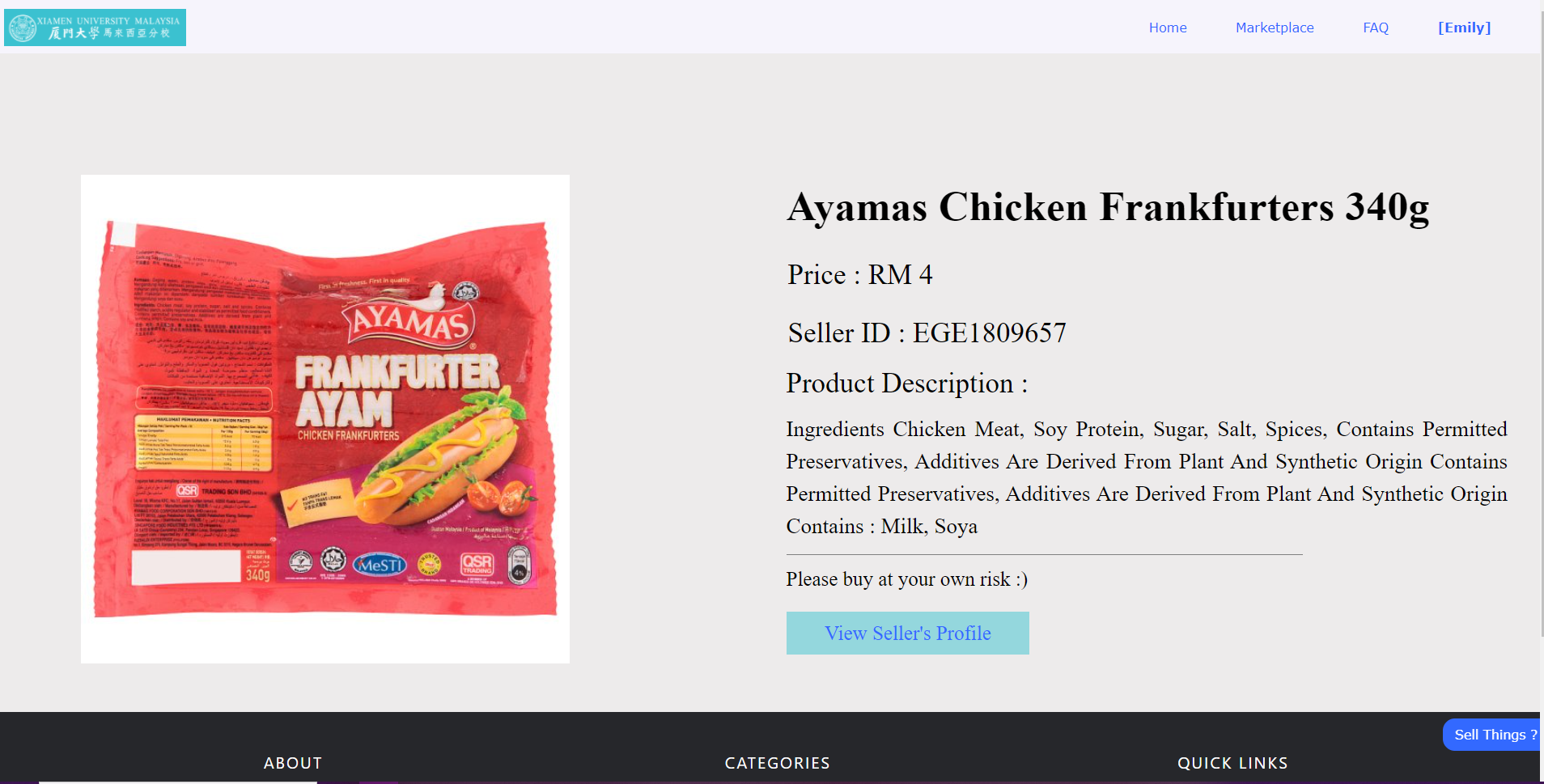
|  |  |
| --- | --- |
| **Functionality** | Upload Product |
| **Functionality Description** | The user wants to upload a product on X-Trade. |
| **Pre-condition** | The user is a member of X-Trade and had login to his/her account. |
| **Post-condition** | The product being added to the product pages and the user’s product table. |
| **Basic events** | 1) The user clicks the “Sell Things?” button at the bottom right on any pages  2) The user fills all the sections on the product upload pages  3) The selected product was successfully added to  the product page and the user’s product table. |

***Table 5.4*** *Upload Product on X-Trade*

## 

## 5.5 View Product Details

### 5.5.1 Picture of View Product Details on X-Trade



***Figure 5.5*** *Picture of view product details page on X-Trade*

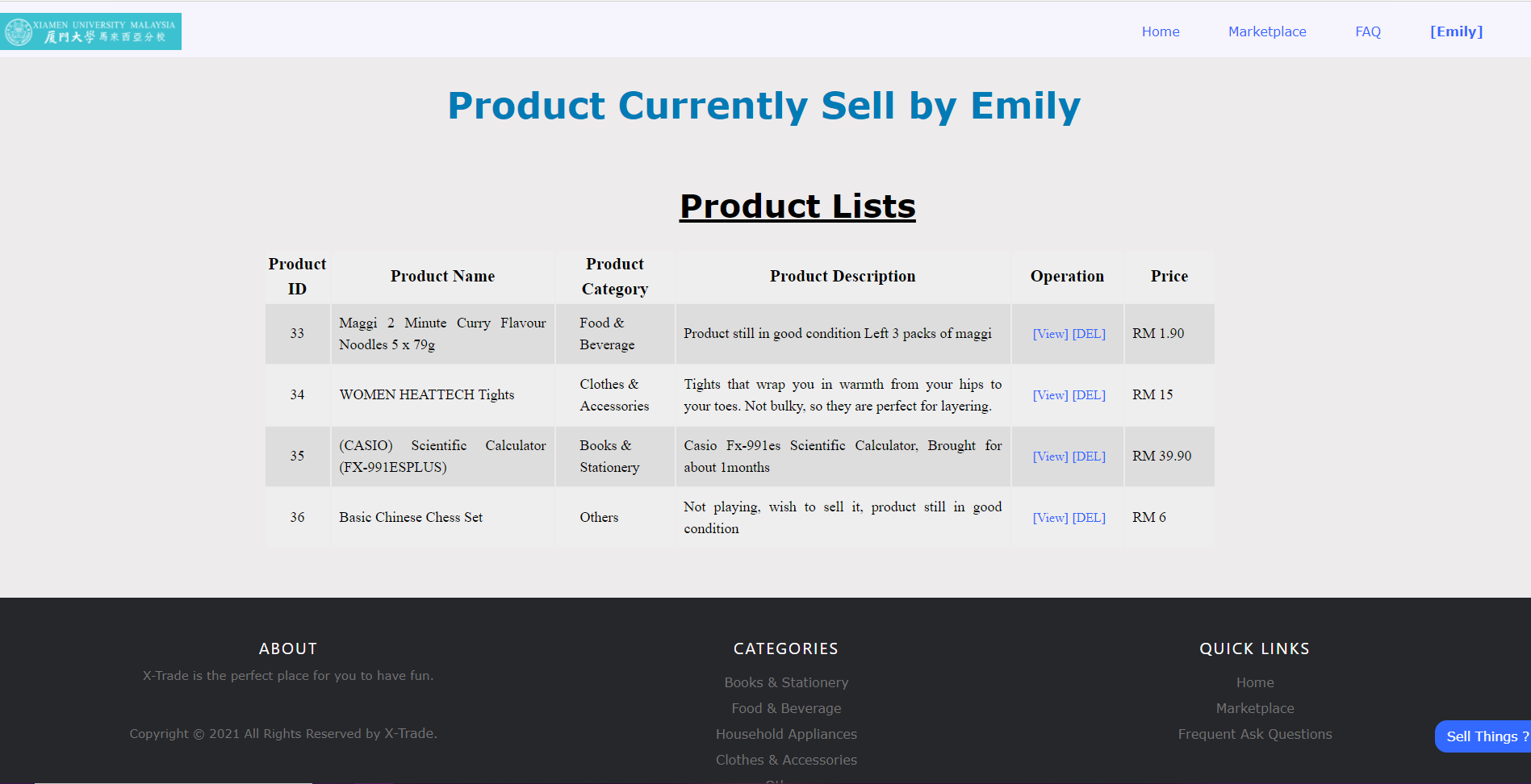
### 5.5.2 View Product Details on X-Trade

|  |  |
| --- | --- |
| **Functionality** | View Product Details |
| **Functionality Description** | The user wants to view a particular product details |
| **Pre-condition** | The user is a member of X-Trade and had login to his/her account. |
| **Post-condition** | The detailed information of the selected products on X-Trade website is displayed to the user. |
| **Basic events** | 1) The user clicks on one of the pages on the “Marketplace” section on the navigation bar.  2) The user clicks on the selected product that he/she wants to view.  3) The detailed information of the selected product is shown to the user. |

***Table 5.5*** *View Product on X-Trade*

## 5.6 View Currently Sell Product

### 5.6.1 Picture of View Product Details of X-Trade



***Figure 5.6*** *Picture of View Product Details on X-Trade*

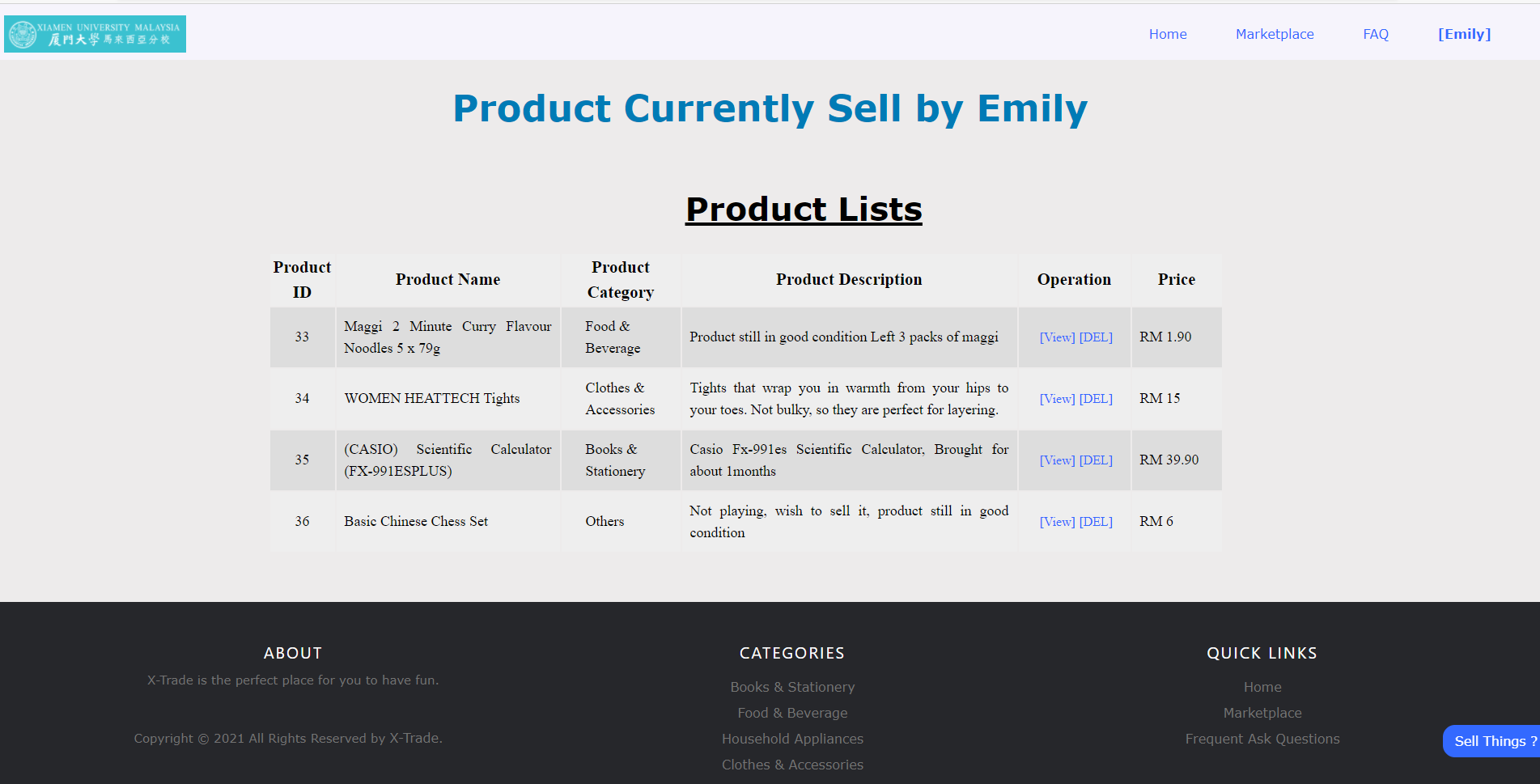
### 5.6.2 View Currently Sell Product on X-Trade

|  |  |
| --- | --- |
| **Functionality** | View Currently Sell Product |
| **Functionality Description** | The user wants to view currently sell product |
| **Pre-condition** | The user is a member of X-Trade and had product currently selling. |
| **Post-condition** | The information of product currently selling were display to the user. |
| **Basic events** | 1) The user clicks the “Currently Sell Product” selection which dropped down at the Profile Section  2) The currently sell product was display in a form of table to the user |

***Table 5.6*** *View currently sell product on X-Trade*

## 5.7 Delete Product

### 5.7.1 Picture of Delete Product in Currently Sell Product Page



***Figure 5.7*** *Picture of delete product in currently sell product page*

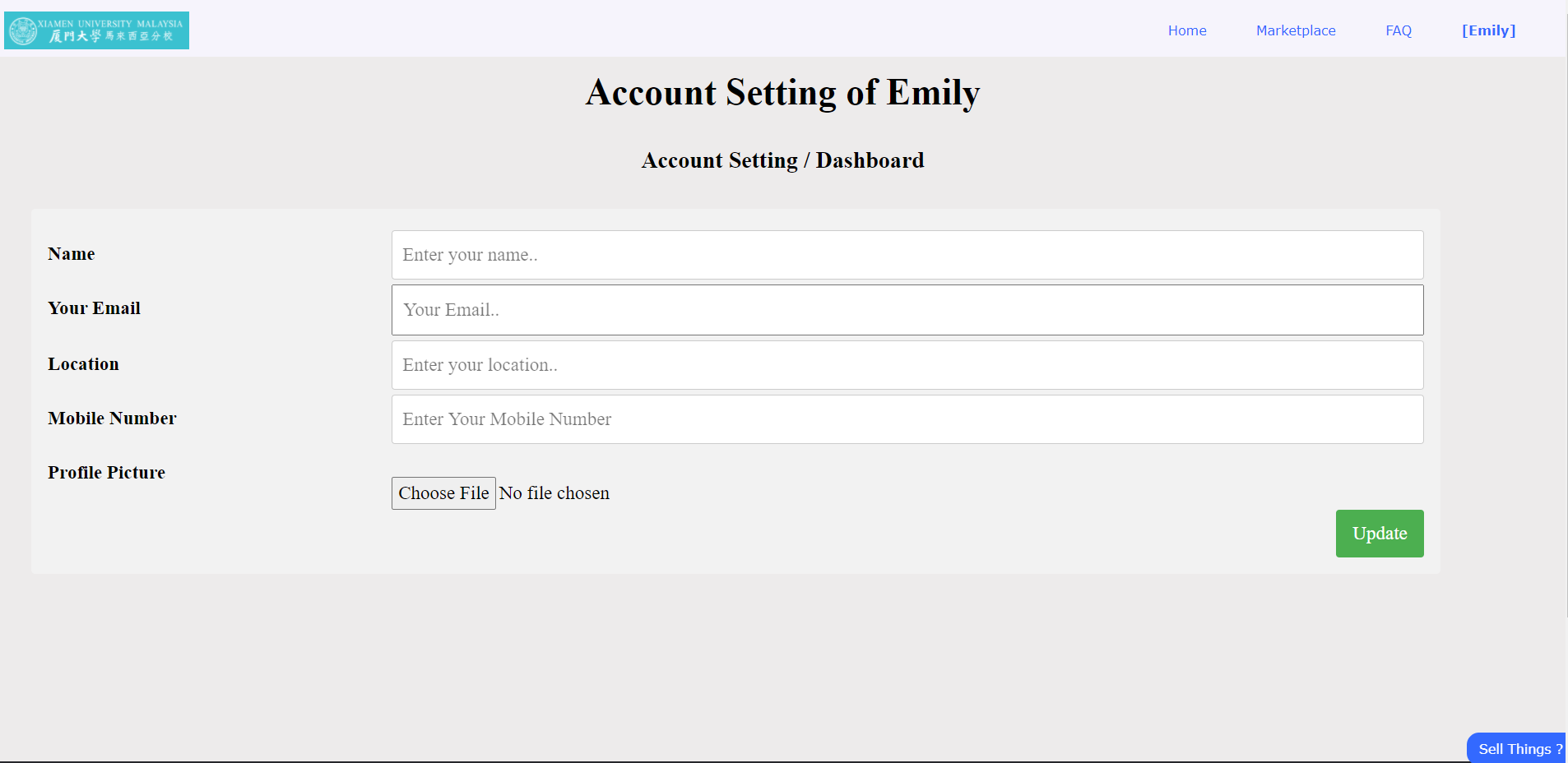
### 5.7.2 Delete Product for X-Trade

|  |  |
| --- | --- |
| **Functionality** | Delete Product |
| **Functionality Description** | The user wants to delete product from the product page via product Table |
| **Pre-condition** | The registered user already login to his/her account and have product current posted on the product pages. |
| **Post-condition** | The product being removed from the product pages and the user’s product table. |
| **Basic events** | 1) The user proceeds to the product table pages via navigation bar  2) The user clicks on the “DEL” on the operation column.  3) The selected product was successfully removed from the product page and the product table. |

***Table 5.7*** *Delete Product for X-Trade*

## 5.8 Change User Account Settings

### 5.8.1 Picture of Change User Account Setting Page on X-Trade

***Figure 5.8*** *Picture of change account setting page on X-Trade*

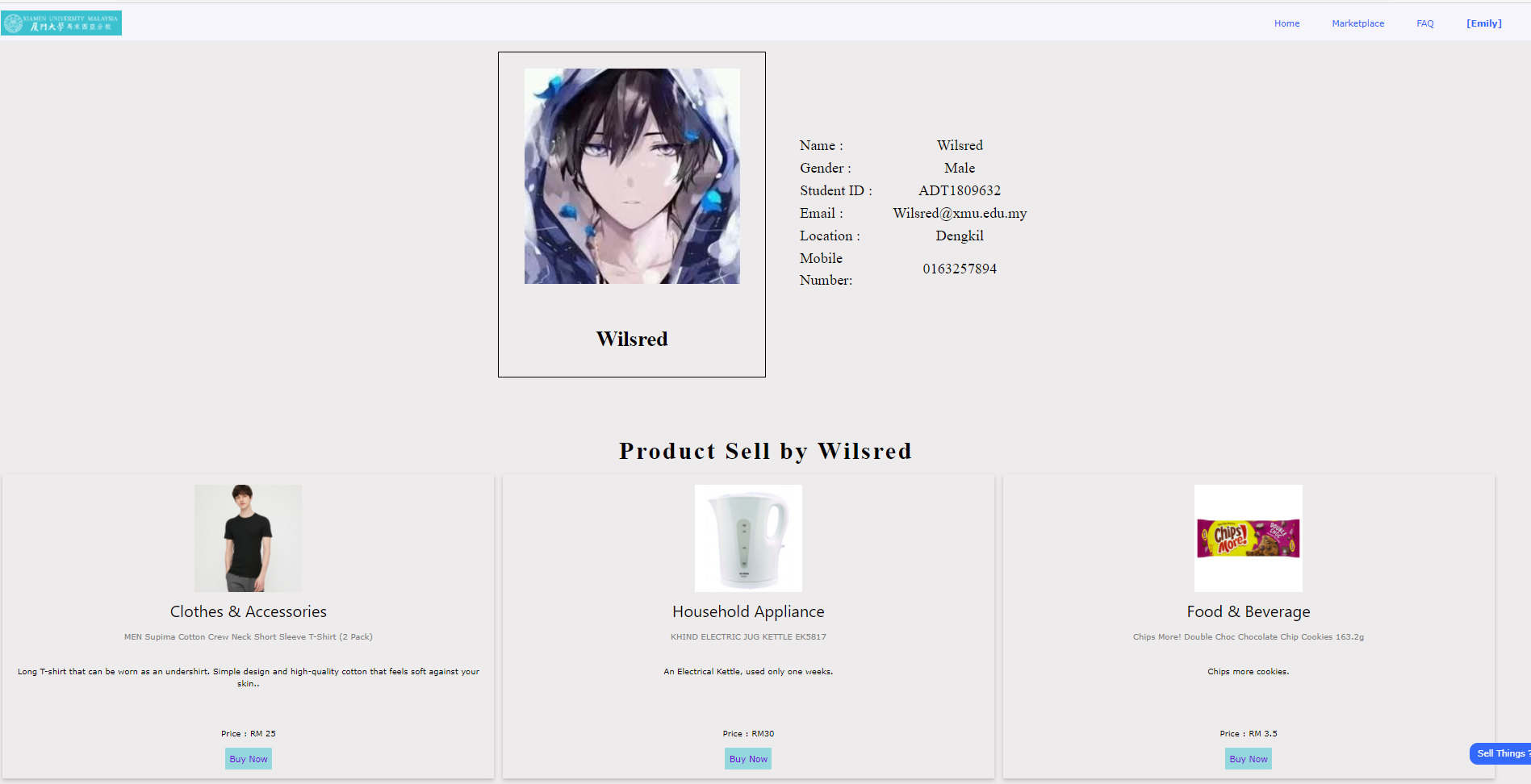
### 5.8.2 Change User Account Setting on X-Trade

|  |  |
| --- | --- |
| **Functionality** | Change User Account Settings |
| **Functionality Description** | The user wants to change his/her profile settings. |
| **Pre-condition** | The user is a member of X-Trade and had login to his/her account and enter the profile page |
| **Post-condition** | The information of the users were changed successful |
| **Basic events** | 1) The click the “Edit Profile” button.  2) The user fills all the sections on the change account setting page  3) The user click the “update” button. |

***Table 5.8*** *Change User Account Setting on X-Trade*

## 5.9 View Seller’s Profile

### 5.9.1 Picture of View Seller Profile Page on X-Trade



***Figure 5.9*** *Picture of view seller profile page on X-Trade*

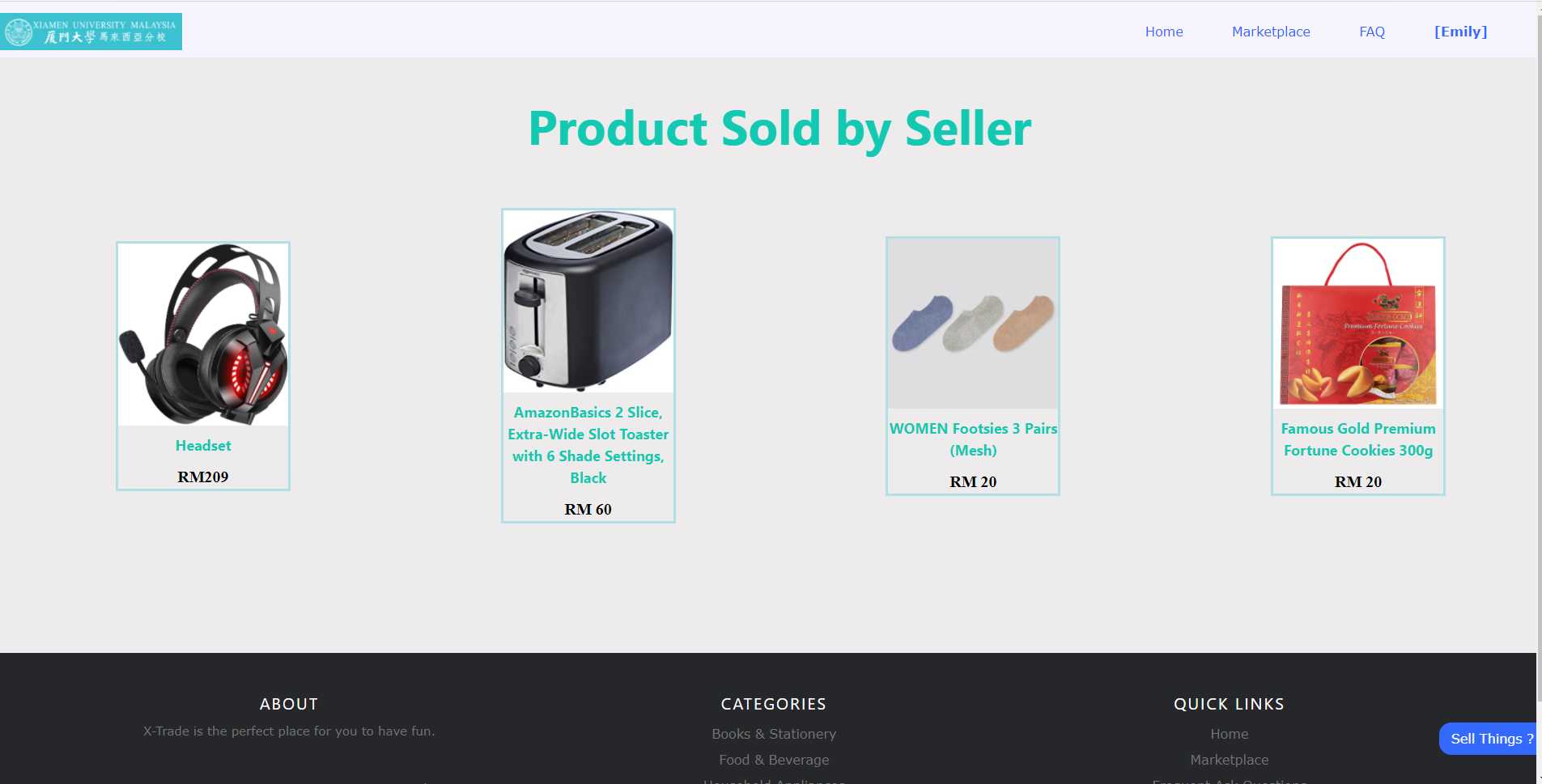
### 5.9.2 View Seller Profile Page on X-Trade

|  |  |
| --- | --- |
| **Functionality** | View Seller’s Profile |
| **Functionality Description** | The registered user wants to view the seller's profile to know the contact number of the seller. |
| **Pre-condition** | The user is in product-details page |
| **Post-condition** | The profile of the seller is displayed to the user |
| **Basic events** | 1) The user clicks the “View the seller’s profile” button on the profile details page. |

***Table 5.9*** *View Seller Profile page in X-Trade*

## 5.10 View Product Sold by Particular Seller

### 5.10.1 Picture of View Seller Profile Page of X-Trade



***Figure 5.9*** *Picture of view seller profile page of X-Trade*

### 5.10.2 View Seller Profile Page in X-Trade

|  |  |
| --- | --- |
| **Functionality** | View Product Sold by Particular Seller |
| **Functionality Description** | The registered user wants to view  product sold by particular seller |
| **Pre-condition** | The user already login to his/her account. |
| **Post-condition** | The product that the seller currently selling is shown to the user. |
| **Basic events** | 1) The user clicks the “View more” button on the Seller’s profile page |

***Table 5.10*** *View Product Sold by Particular Seller in X-Trade*

## 5.11 View Frequent Asked Question

### 5.11.1 Picture of View Seller Profile Page of X-Trade



***Figure 5.9*** *Picture of view seller profile page of X-Trade*

### 5.11.2 View Frequent Asked Questions in X-Trade

|  |  |
| --- | --- |
| **Functionality** | View Frequent Asked Questions |
| **Functionality Description** | The registered user wants to view frequent ask questions of X-Trade |
| **Pre-condition** | There is no any Pre-conditions |
| **Post-condition** | The user successfully view the frequent asked questions |
| **Basic events** | 1) The user clicks the “FAQ” button on the navigation bar |

***Table 5.11*** *Frequent Asked Questions in X-Trade*

## 5.12 Log Out

### 5.11.1 Picture of Log out of X-Trade



***Figure 5.9*** *Picture of log out page of X-Trade*

### 5.12.2 Log out in X-Trade

|  |  |
| --- | --- |
| **Functionality** | Log Out |
| **Functionality Description** | The user wants to log out from X-Trade. |
| **Pre-condition** | The user is a registered user and currently log in to his/her account |
| **Post-condition** | The user successfully logs in to his/her account. |
| **Basic events** | 1) The user clicks on the “Log Out” on the name of the user section on the navigation bar.  2) The user successfully log out from X-Trade. |

***Table 5.12*** *Log out in X-Trade*

# 6.0 Difficulties Faced during the Implementation of Project

## 6.1 Difficulties in using Flexbox as a Container

In X-Trade, all product categories are displayed using flex-box. Since the image has different size, we failed to compress the image, product name and product description in a same size container, making the container displayed with different size.

## 6.2 Difficulties in Implementation for every Actor

In X-Trade, there’s only two actors who are buyers and sellers. We failed to implement the view for admin due to the time constraint and lack of functionalities for admin.

## 

## 6.3 Difficulties in Creating Inbox

Inbox function was planned in our proposal initially to connect buyer with seller initially. However, lack of knowledge on JavaScript makes the implementation of inbox more difficult. We have tried to create a front-end part for the inbox page but difficulties switch inbox tabs and make it active causing us to replace the function with a contact seller via phone number.

## 6.4 Difficulties in Display Image which in Local Computer in JSP File

After research on Google, the image needs to be stored in the server in order to display in .jsp file. Therefore, we are forced to use internet images in order to display it successfully.

## 6.5 Difficulties in Implementing Varieties of Payment Method

We wanted to implement a variety of payment methods such as Payment Gateway, pay through e-wallet and other online payment platforms. We found out it is difficult to connect with that type of platform and decided to use a traditional payment method which is pay face-to-face.

## 6.6 Difficulties in Creating Responsive Website

We want to make the website responsive in any size of the window but due to lack of knowledge in HTML and CSS we are only able to display the website normally in a full-sized window.

# Future Improvement

## 7.1 Create Inbox to Connect Buyer and Seller

We hope to increase knowledge on JavaScript and can create inbox like inbox in Facebook to increase efficiency of connection between buyer and seller.

## 7.2 Create Responsive Website

We hope to create a website in normal size in any size of window and fit in any device.

## 

## 7.3 Increase Variety of Payment Method

We hope to increase variety ways of payment method like Touch n-go, WeChat Pay, and e-Banking and also student card.

# 

# 8.0 Conclusion

With the implementation of HTML, CSS, Java, Java EE and SQL language, the X-Trade website is built. The product details are displayed using client-server architecture by Apache Tomcat and Servlet. X-Trade also provides different functionality for the user to access X-Trade. Although we faced some difficulties during the project, we managed to solve them by using alternative ways and document them in our future plan. In our opinion, there is always still a large space to improve and evolve in website development. We hope that we can have more chances to participate in the future on developing websites.

# 9.0 Work Breakdown

|  |  |  |
| --- | --- | --- |
| **Member Details** | | **Work Portion** |
| **Name** | **Student ID** |
| **Khor Jia Cheng** | **SWE1809651** | * ***Leader*** * ***Website: All Product page and Product Category page, Converting Product page and Product Category page into. jsp format.*** * ***Database: User and Product Table*** * ***Report: Abstract, Introduction, Code Architecture, Case Study*** |
| **Leong Wei Jun** | **SWE1809654** | * ***Team Member (1)*** * ***Website: Other pages except products page, product details page and Homepage, and then convert it to. jsp format.*** * ***Database: User and Product Table*** * ***Report: Case Study, Compilation of the Report*** |
| **Tan Ting Yi** | **SWE1809659** | * ***Team Member (2)*** * ***Website: Homepage and Product Category page, Converting Home Page into. jsp format.*** * ***Designer for the whole website*** * ***Database: User and Product Table*** * ***Report: Conclusion*** |

# 10.0 Reference

Kothari, K. (2020). Understanding Java Servlet Architecture. Retrieved 14 January 2021, from https://codeburst.io/understanding-java-servlet-architecture-b74f5ea64bf4

# 11.0 Appendices

**For. jsp file**

* **AllP**: All Product category for user who haven’t register or login.
* **AllProduct**: All product category for user who has login.
* **Bdetail**: Product detail for the product.
* **BnS**: Books and Stationery’s category product for user who haven’t register or login.
* **Books&Stationery**: Books and Stationery’s category product for user who has login.
* **Clothes**: Clothes’ category product for user who has login.
* **CNA**: Clothes’ category product for user who haven’t register or login.
* **F&B**: Food&Beverage’ category product for user who has login.
* **FnB**: Food&Beverage’ category product for user who haven’t register or login.
* **FAQ**: FAQ page for user who has login.
* **FAQQ**: FAQ page for user who haven’t register or login.
* **HA**: Household Appliance’ category product for user who haven’t register or login.
* **HouseholdAppliance**: Household Appliance’ category product for user who has login.
* **Login**: Login page
* **Home**: Home page for the user who had login.
* **HomePage**: Home page for the user who haven’t register or login. It is also the default page for this project.
* **NewAccountSetting**: Account Setting page for the user.
* **O**: Others’ category product for user who haven’t register or login.
* **Others**: Others’ category product for user who has login.
* **ProductTable**: Display currently selling product table for the user.
* **ProductUpload**: Upload product details to become a seller.
* **Profile**: Display the user’s profile.
* **Registration**: Page for unregistered user to register.
* **SellerAllProduct**: Display all seller’s currently selling product.
* **ViewSellerProfile**: Display seller profile information.

**APPENDIX 1**

**Marking Rubrics**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Component Title** | **Project Report** | | | | | **Weight** | **100** | |
| **Criteria** | **Score and Descriptors** | | | | | | **Weight** | **Marks** |
| **Excellent (directly relevant)**  **25-30** | **Good**  **(Somewhat relevant)**  **20-25** | **Average**  **(Remotely related) 15-20** | **Need Improvement**  **(Not related)**  **10-15** | **Poor (not related / not present)**  **0-10** | |
| Report | Information is presented in effective order. Excellent structure of paragraphs and transitions enhances readability and comprehension. | Information is logically ordered with paragraphs and transitions. | Include vague information. | Details and examples are not organized, and hard to follow and understand. | Details are not related and badly written and hard to follow. | | 100 |  |
| **TOTAL** | | | | | | | **100** |  |

Note to students: Please print out and attach this appendix together with the submission of coursework