Web Design & Development Report 2

Edinburgh Napier University

Module Leader : **Amjad Ullah**

Matriculation Number: **40437531**

Word Count : **622**

Date: **21st April, 2024**

Table of Contents

Introduction

The purpose of the report is to demonstrate the various aspects that make the developed website, Chapter One, fully interactive and professional, focusing on the folder structure, cart handling feature, security, authentication, content management and error handling. Additionally, the report will also display the user journey, starting from landing page to checking out process within the website.

Folder Structure

The folder structure for the website is inspired by the MVC design pattern. MVC stands for model-view-controller and it is a design pattern that breaks down the large and complex development into manageable and maintainable components. This is the basic principle of how the MVC architecture works and my folder structure is inspired by this pattern. However, the MVC pattern that I used on the website is not an actual pattern; it is inspired by the idea of separating components and used to organize the folder structure as it provides more clarity, and a simple way to manage the code flow easily.

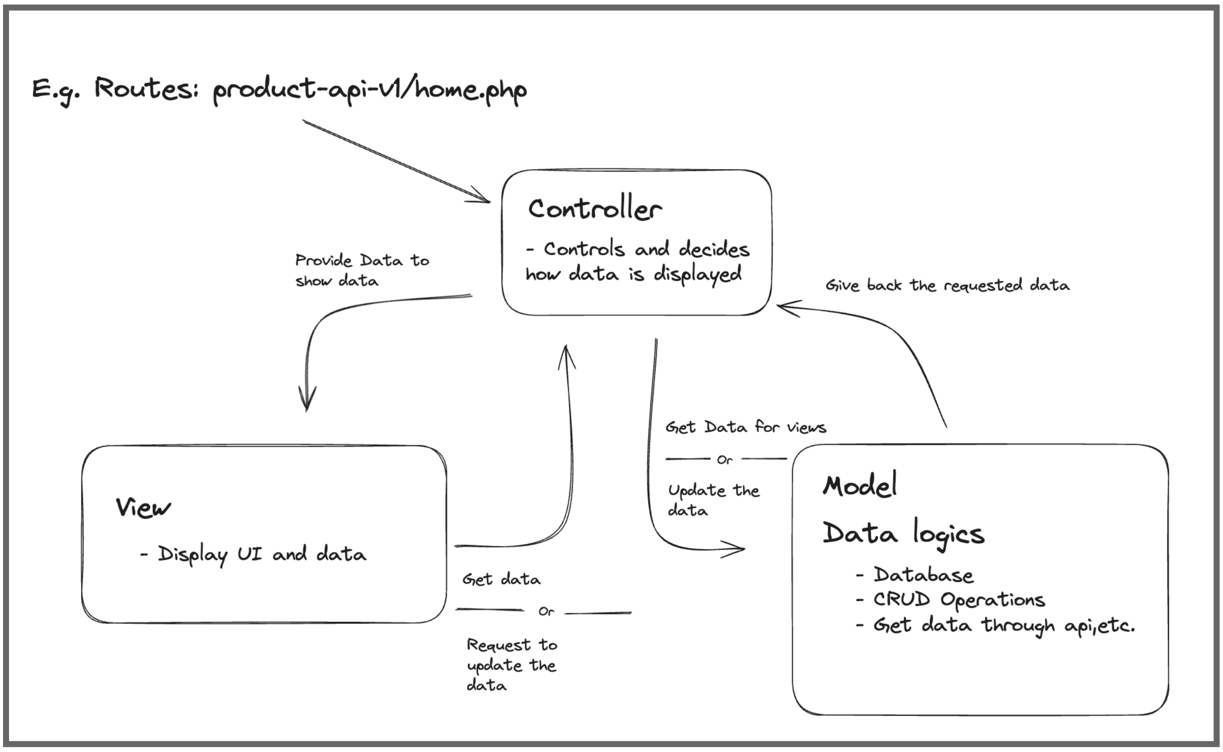

E.g. Routes: product-api-v1/home.php
Controller
- Controls and decides how data is displayed
Provide Data to show data
Give back the requested data
Get Data for views
Or
Update the data
View
- Display UI and data
Get data
Or
Request to update the data
Model
Data logics
- Database
- CRUD Operations
- Get data through api,etc.

Figure - 1.1: The MVC design pattern

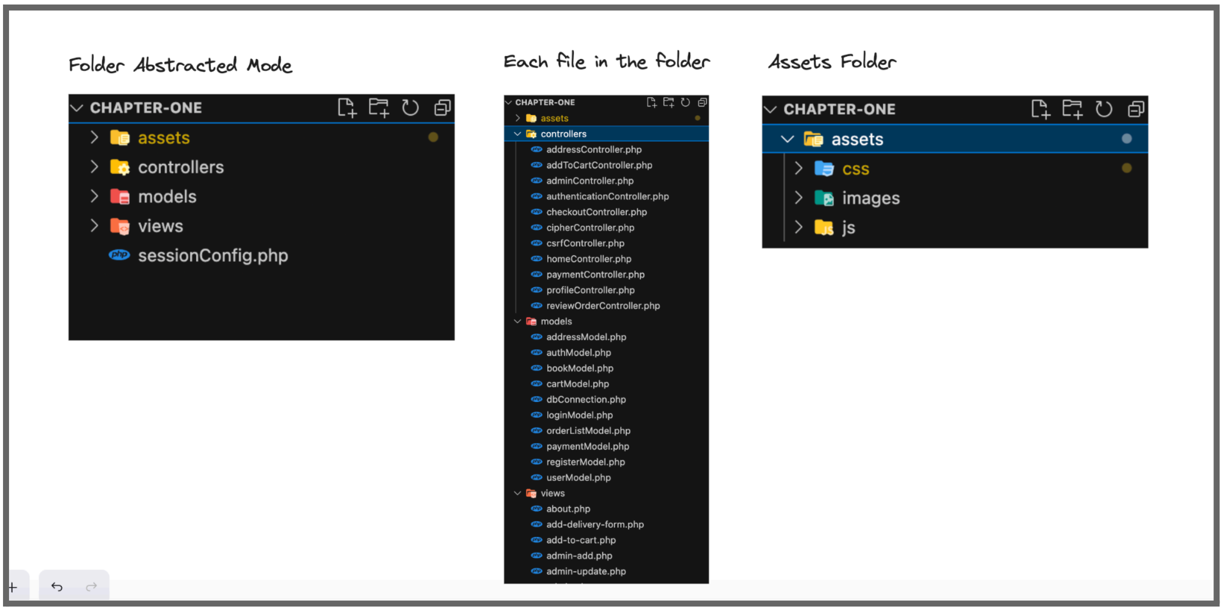


Figure - 1.2: Folder structure of the website

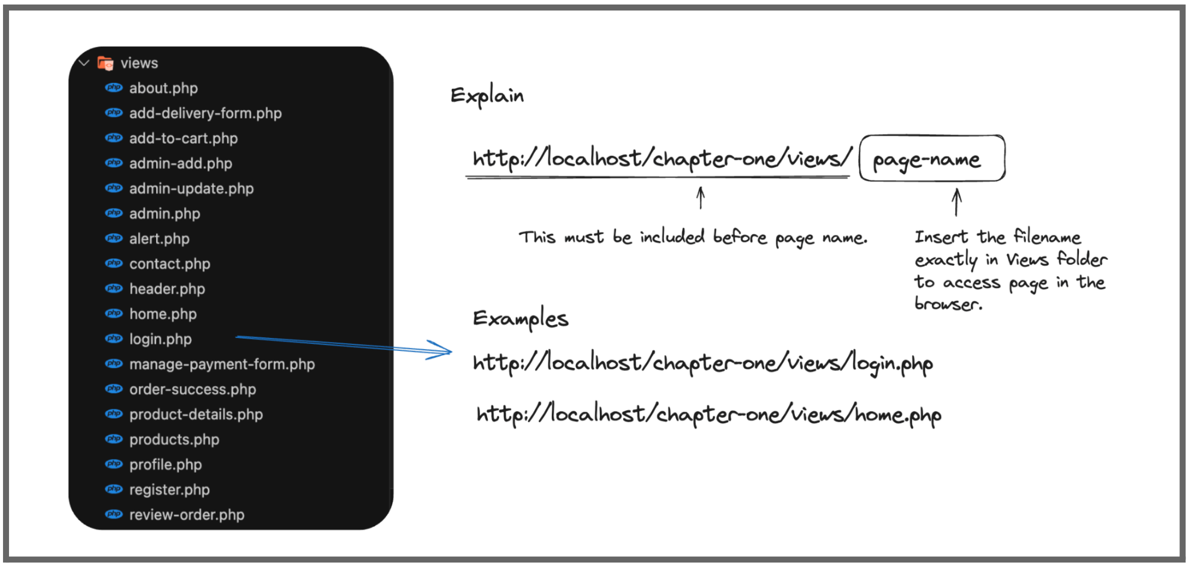
Figure 1.2 describes the overall files of the website. Aside from the main three folders, there is also another folder which is the assets folder. This is where various resources, such as css, images, and js, used in the website are stored.

URLs

About the pages, every file in the Views folder can be reached through the browser url. However, trying to navigate to the other files in the Models or Controllers via the URL will cause an error. Therefore, only pages in the Views folder can be accessed via the browser URL. The below figure shows how the pages in the Views folder can be accessed in the browser.

Although I said every page in the Views can be accessed, some of them are used as a UI component rather than a page. Following files serve as a UI component and cannot not be accessed.

- alert.php
- add-to-cart.php
- header.php
- footer.php



Explain

http://localhost/chapter-one/views/ page-name

This must be included before page name.

Insert the filename exactly in Views folder to access page in the browser.

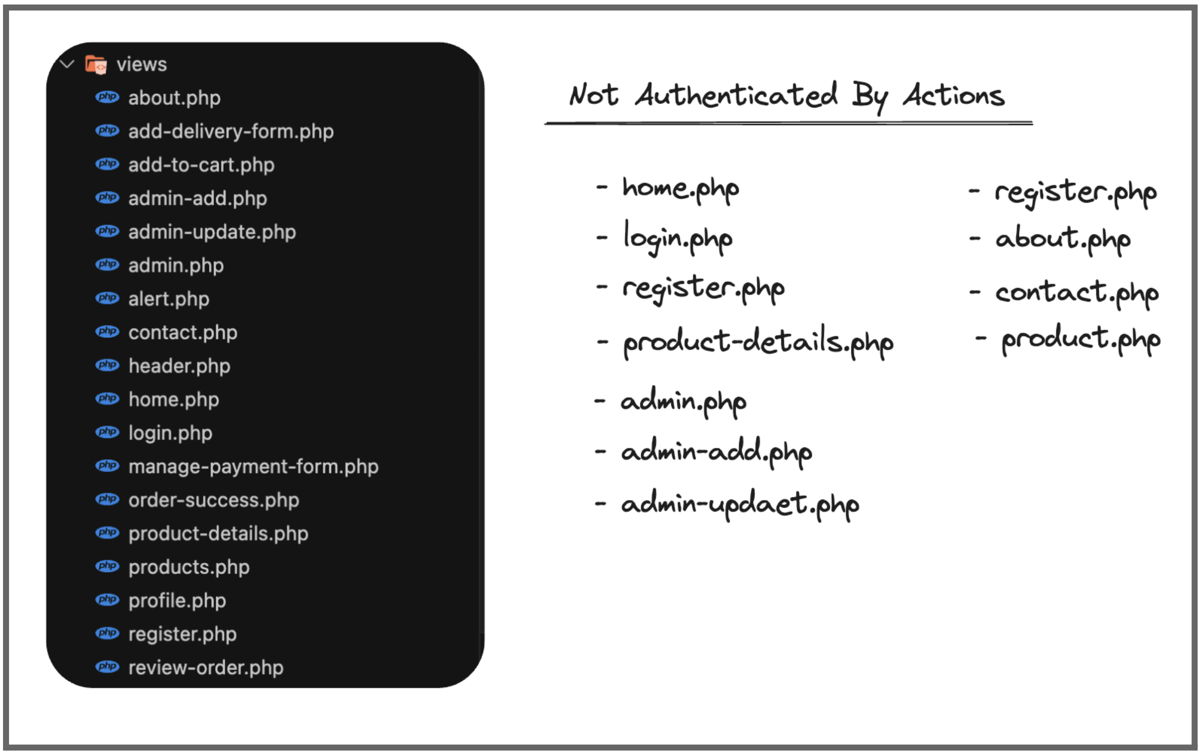
Examples

http://localhost/chapter-one/views/login.php

http://localhost/chapter-one/views/home.php

Figure - 1.3: Accessing pages from the browser URL

When visiting the pages in the Views Folder, some pages are validated meaning a user cannot go to those pages until a certain action is done. For instance, the user cannot access the checkout page without nothing in the cart or logging as a user. Below, the report provides the list of the pages, describing what pages are authenticated, and what are not.



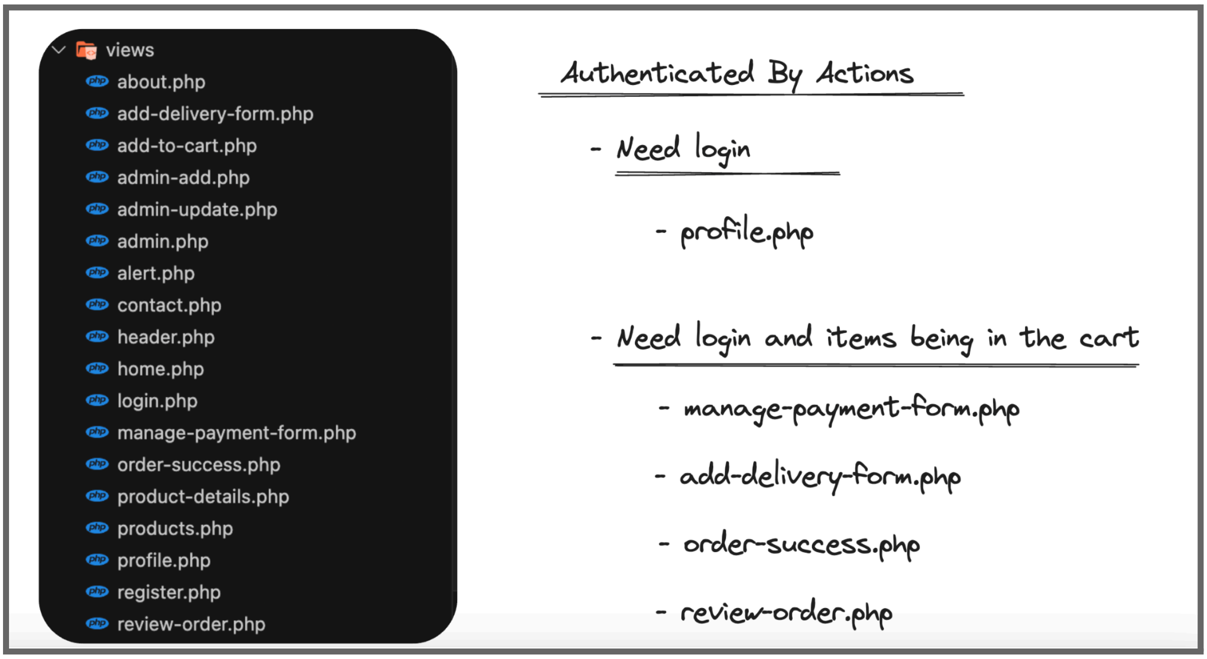
```
views
  about.php
  add-delivery-form.php
  add-to-cart.php
  admin-add.php
  admin-update.php
  admin.php
  alert.php
  contact.php
  header.php
  home.php
  login.php
  manage-payment-form.php
  order-success.php
  product-details.php
  products.php
  profile.php
  register.php
  review-order.php
```

Not Authenticated By Actions

- home.php
- login.php
- register.php
- product-details.php
- admin.php
- admin-add.php
- admin-updaet.php

- register.php
- about.php
- contact.php
- product.php

Figure - 1.4: Pages with no authentication



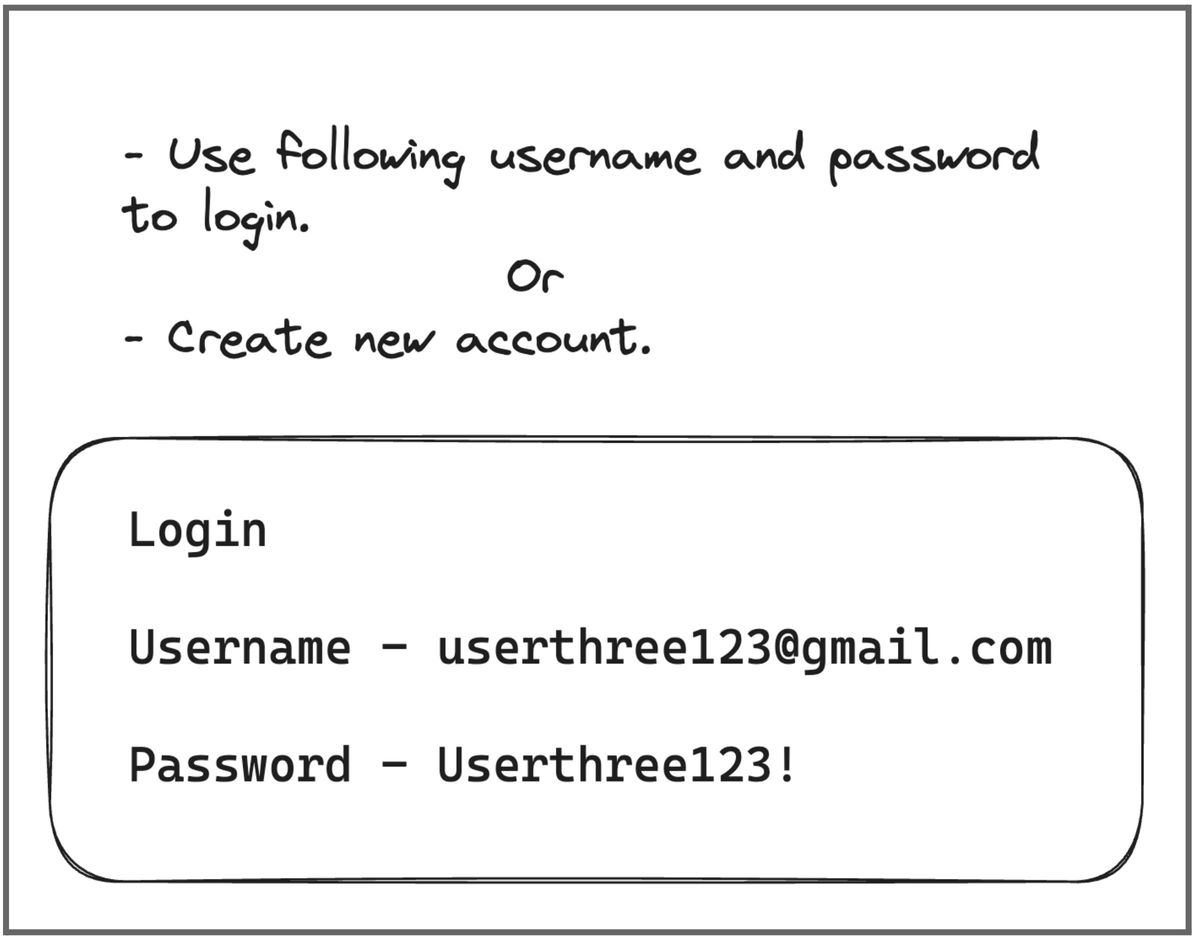
```
views
  about.php
  add-delivery-form.php
  add-to-cart.php
  admin-add.php
  admin-update.php
  admin.php
  alert.php
  contact.php
  header.php
  home.php
  login.php
  manage-payment-form.php
  order-success.php
  product-details.php
  products.php
  profile.php
  register.php
  review-order.php
```

Authenticated By Actions

- Need login
  - profile.php
- Need login and items being in the cart
  - manage-payment-form.php
  - add-delivery-form.php
  - order-success.php
  - review-order.php

Figure - 1.5: Pages with authentication

Login Information



- Use following username and password to login.

Or

- Create new account.

Login

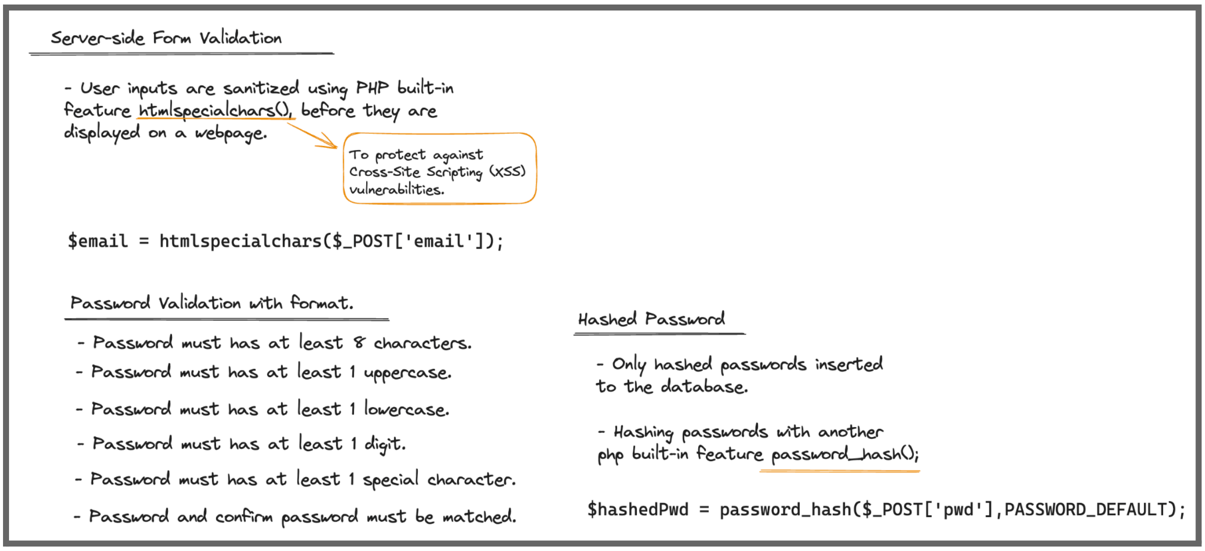
Username - userthree123@gmail.com

Password - Userthree123!

Figure - 1.6: Username and password

Login and Register Form Validation

All inputs from login and register validated properly, especially email and password formats. All the validations are from the server-side and there is no front-end validation using JavaScript. Implemented form validation is shown below.



Server-side Form Validation

- User inputs are sanitized using PHP built-in feature htmlspecialchars(), before they are displayed on a webpage.

To protect against Cross-Site Scripting (XSS) vulnerabilities.

```
$email = htmlspecialchars($_POST['email']);
```

Password Validation with format.

- Password must has at least 8 characters.
- Password must has at least 1 uppercase.
- Password must has at least 1 lowercase.
- Password must has at least 1 digit.
- Password must has at least 1 special character.
- Password and confirm password must be matched.

Hashed Password

- Only hashed passwords inserted to the database.

- Hashing passwords with another php built-in feature password_hash();

```
$hashedPwd = password_hash($_POST['pwd'],PASSWORD_DEFAULT);
```

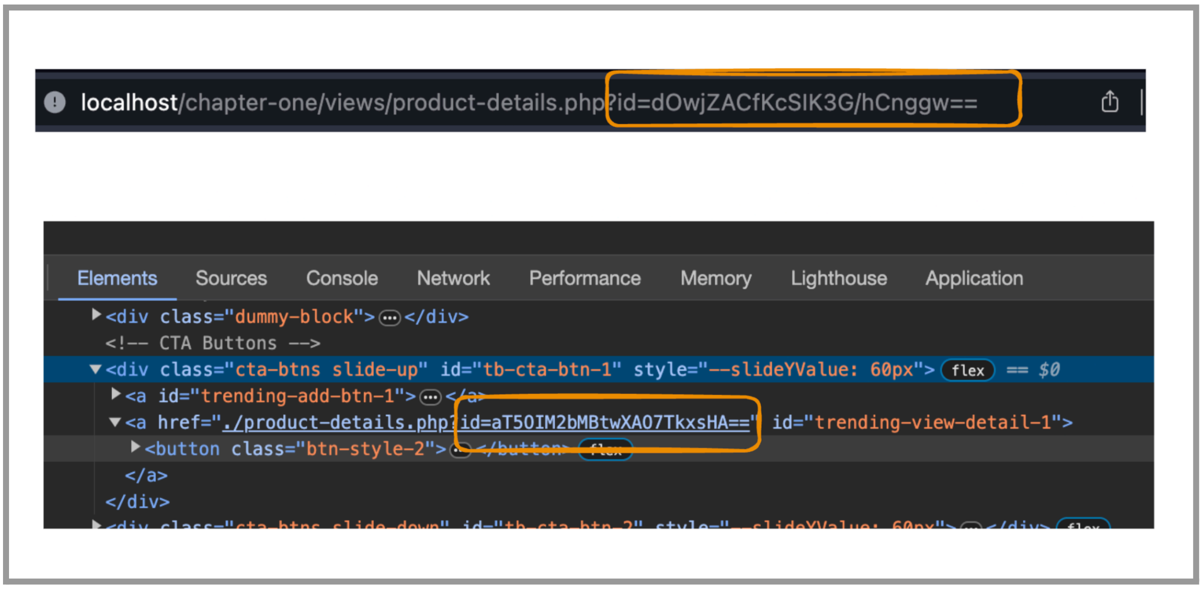
Figure - 1.7: Form Validation

CSRF Attack Protection

Login and register forms are protected with CSRF tokens. CSRF protection is implemented in the website therefore unauthorized or malicious requests cannot access the web application because the server checks the CSRF token and makes sure it matches the one it issued. Only accept the forms that are coming from the Chapter One website and any forgery attack will be blocked.

ID Encryption and Decryption

The book ids in the website are encrypted. Instead of providing actual id, encrypting id is a bit more safe because using the inspector of the browser, ids can be exposed and one can use those ids to do malicious things. The following figure shows encrypted ids in the inspector and the browser URL. The encryption and decryption has been achieved with openssl_encrypt and openssl_decrypt provided by PHP.



```
localhost/chapter-one/views/product-details.php?id=dOwjZACfKcSIK3G/hCnggw==
```

```
  ▶<div class="dummy-block">⋯</div>
   <!-- CTA Buttons -->
  ▼<div class="cta-btns slide-up" id="tb-cta-btn-1" style="--slideYValue: 60px"> flex == $0
    ▶<a id="trending-add-btn-1">⋯</a>
    ▼<a href="./product-details.php?id=aT5OIM2bMBtwXA07TkxsHA==" id="trending-view-detail-1">
      ▶<button class="btn-style-2">⋯</button> flex
     </a>
   </div>
  ▶<div class="cta-btns slide-down" id="tb-cta-btn-2" style="--slideYValue: 60px">⋯</div> flex
```

Figure - 1.8: Encrypted Ids

Session Timeout

Session timeout feature is also implemented in the website. Every 30 minutes, the session will be destroyed and created again.

Error Messages

The website displays errors to users when something unexpected happens and it lets them know that the actions they perform are not successful.

Content Management

There are pages related to admin and they allow you to create, update, and delete and view the data.

Front-end Cart Handling Processes

Every process related to the cart handling is done in the front-end by communicating and exchanging data with the server using API. This includes adding books to the cart, increasing/decreasing the amount in the cart, and removing items from it.