ASSIGNMENT 2 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Business**		
Unit number and title	**Unit 30: Application Development**		
Submission date		**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name	Doan Tran Vinh Kha	**Student ID**	GCS190930
Class	GCD0901	**Assessor name**	Hoang Nhu Vinh
Student declaration I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.			
		Student's signature	

Grading grid

P4	P5	P6	M3	M4	M5	D2	D3
✓	✓	✓	✓	✓	✓	✓	✓

❒ **Summative Feedback:**		❒ **Resubmission Feedback:**
3.1		
Grade:	**Assessor Signature:**	**Date:**
Internal Verifier's Comments:		
Signature & Date:		

Chapter 1 – Peer Review & Feedback Analysis (P4 – M3 – D2) 9
I. Questionnaires To Review The Application 9
II. Collecting Review Feedbacks (P4) 11
III. Interpreting Peer-Review Feedbacks (M3) 13
IV. Evaluate New Insights, Ideas Or Potential Improvements Of The Application (D2) 14
Chapter 2 – Application Development (P5 – M4) 18
I. Folder Structure (P5) 18
II. Source Code Examples (M5) 19
1. Creating A New Book 19
2. Edit Book's Information 20
3. Add Item To Cart 21
4. Search and Show Home Page 22
5. View Order Book 23
III. Screenshot Of The Website (M5) 24
1. Unregistered Users 24
2. Customer Role 27
3. Store Owner Role 31
4. Administrator Role 40
IV. Version Control Of The Project (M5) 44
V. IIS Deployment (M5) 46
Chapter 3 - Application Evaluation (P6 – M5 – D3) 55
I. Performance Of The Application (P6) 55
1. The Functionality Of The Website 55
2. Test Case 56

3. The Performance Of The Website 69
II. Personal Evaluation (P6) 74
III. Factors That Influence The Performance Of The Application (M5) 75
1. Components Impact The Application Performance 75
2. Conclude The Application Review 76
IV. Evaluate The Strengths And Weaknesses Of The Application (D3) 77
References 79

Figure 1: Questionnaires about the functions of the website 10
Figure 2: Review Feedbacks 12
Figure 3: Folder Structure 18
Figure 4: Create Book Function 19
Figure 5: Edit book function 20
Figure 6: Add item to cart 21
Figure 7: Search and show home page 22
Figure 8: Order Function 23
Figure 9: Home Page screenshot 24
Figure 10: Book Detail page 25
Figure 11: Register Page 26
Figure 12: Login Page 26
Figure 13: Customer Home Page 27
Figure 14: FAQs Page 28
Figure 15: Shopping Cart Page 29
Figure 16: Order List Page 29
Figure 17: Order Detail Page 30
Figure 18: Store Owner Home Page 31
Figure 19: Notification 31
Figure 20: Book Management Page 32
Figure 21: Create Book Page 33
Figure 22: View Book Detail Page 34
Figure 23: Edit Book Information Page 35
Figure 24: Category Management Page 36
Figure 25: Create Category Page 37
Figure 26: Edit Category Information Page 37
Figure 27: List Of Customer's Orders Page 38
Figure 28: List Of Customer Page 39
Figure 29: Home Page Of Administrator 40
Figure 30: Customer Management Page 41

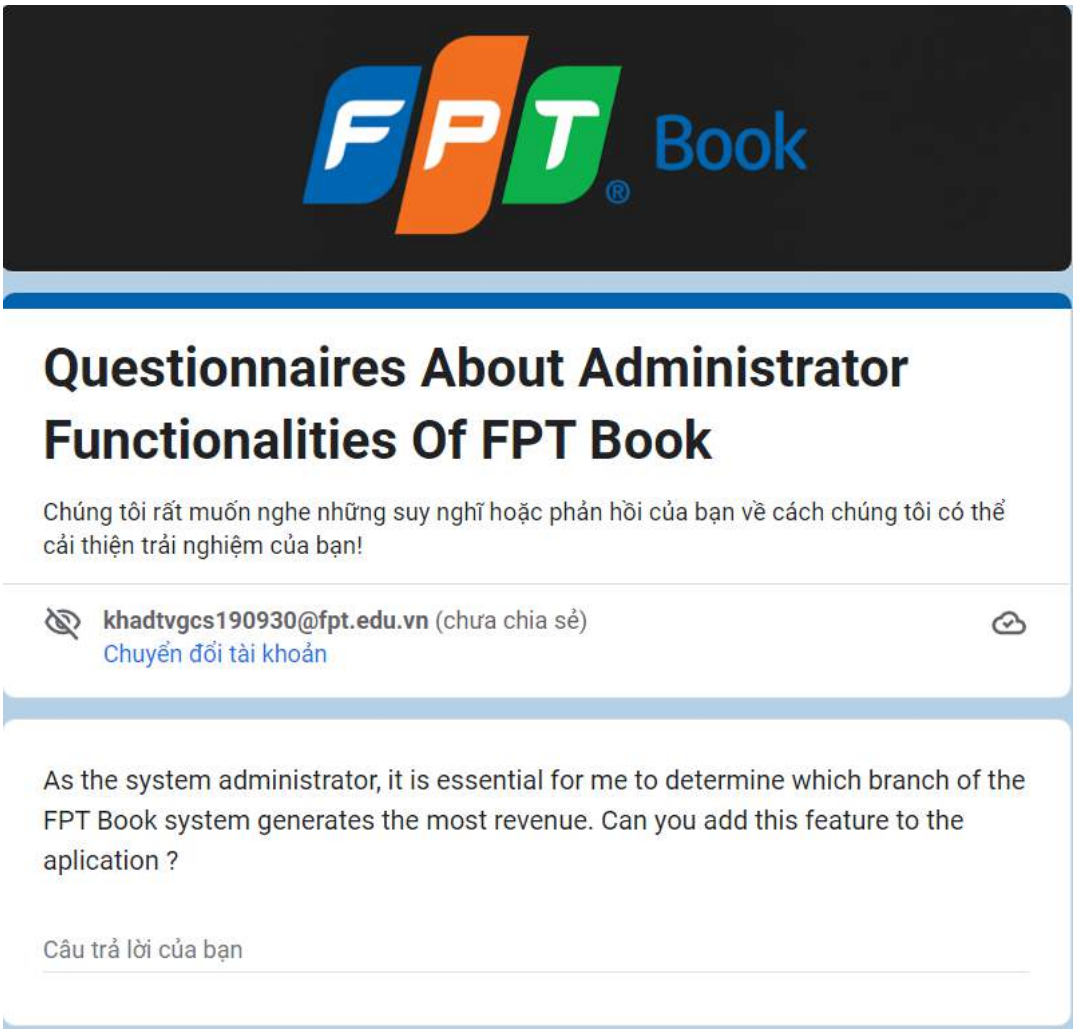
Figure 31: Change Password For Customer Page 41
Figure 32: Store Owner Management Page 42
Figure 33: Change Password For Store Owner Page 42
Figure 34: Category Requets Page 43
Figure 35: Approved Categories Page 43
Figure 36: Project On GitHub 44
Figure 37: Level of source code contribution of each members 44
Figure 38: Our teams's commits 45
Figure 39: Window IIS 46
Figure 40: Create a new login 47
Figure 41: Configure the account 47
Figure 42: Configure the account (2) 48
Figure 43: Change sign in option 48
Figure 44: Restart the SQL Configuration 49
Figure 45: Create new connection string 49
Figure 46: Edit the project configuration method 50
Figure 47: Public window 50
Figure 48: Choosing Web Deploy 51
Figure 49: Configure the deploy connection 51
Figure 50: Finishing configuring the publish profile 52
Figure 51: Ready to publish 52
Figure 52: Access to the website after being deployed 53
Figure 53: Access to the website from mobile device 54
Figure 54: Test case result (1) 59
Figure 55: Test case result (2) 60
Figure 56: Test case result (2) 60
Figure 57: Test case result (3) 61
Figure 58: Test case result (4) 61
Figure 59:Test case result (4) 61
Figure 60: Test case result (5) 62

Figure 61: Test case result (5) 62
Figure 62: Test case result (6) 62
Figure 63: Test case result (6) 63
Figure 64: Test case result (6) 64
Figure 65: Test case result (6) 65
Figure 66: Test case result (6) 65
Figure 67: Test case result (7) 65
Figure 68: Test case result (7) 65
Figure 69: Test case result (8) 66
Figure 70: Test case result (8) 66
Figure 71: Test case result (9) 67
Figure 72: Test case result (9) 67
Figure 73: Test case result (9) 67
Figure 74: Test case result (10) 68
Figure 75: Test case result (10) 68
Figure 76: Test case result (10) 68
Figure 77: Starting the break point 69
Figure 78: Ending the break point 69
Figure 79: Diagnostic tools 70
Figure 80: Statistical figure about website performance 71
Figure 81: Statistical figure about website performance (2) 72
Figure 82: Statistical figure about website performance (3) 73
Figure 83: Statistical figure about website performance (4) 73

Chapter 1 – Peer Review & Feedback Analysis (P4 – M3 – D2)

I. Questionnaires To Review The Application

Following the demonstration of the app's functions with the client who plays as the system administrator, they have some opinions and comments that review our business application and are related to the problem definition statement, the proposed solution, and the development strategy. From the administrator's point of view, this will be done from the perspective of reviewing our business application.The software development team has received the following five questions from the user's perspective as an administrator on the functional development of the website.



It is obvious that our business can not be progressed if I do not cooperate with the parties. I would like it if the FPT Book website could be used to communicate with third parties to boost the company's earnings.

Câu trả lời của bạn

One of the essential criteria of our business is that our shipping service must try our best to bring customers the most convenience. Therefore, I want to link with a third party so the system can provide many forms of payment for home delivery services.

Câu trả lời của bạn

As an administrator of the FPT Book store, I have to know the system's profits as well as human resources for planning suitable award plans. Thus, I want to know which branch has the highest yield and who is the store owner of this branch.

Câu trả lời của bạn

As an administrator of the system, I want an account-creating function for store owners instead of seeding the account information them into the system, which required the technical knowledge about Information Technology

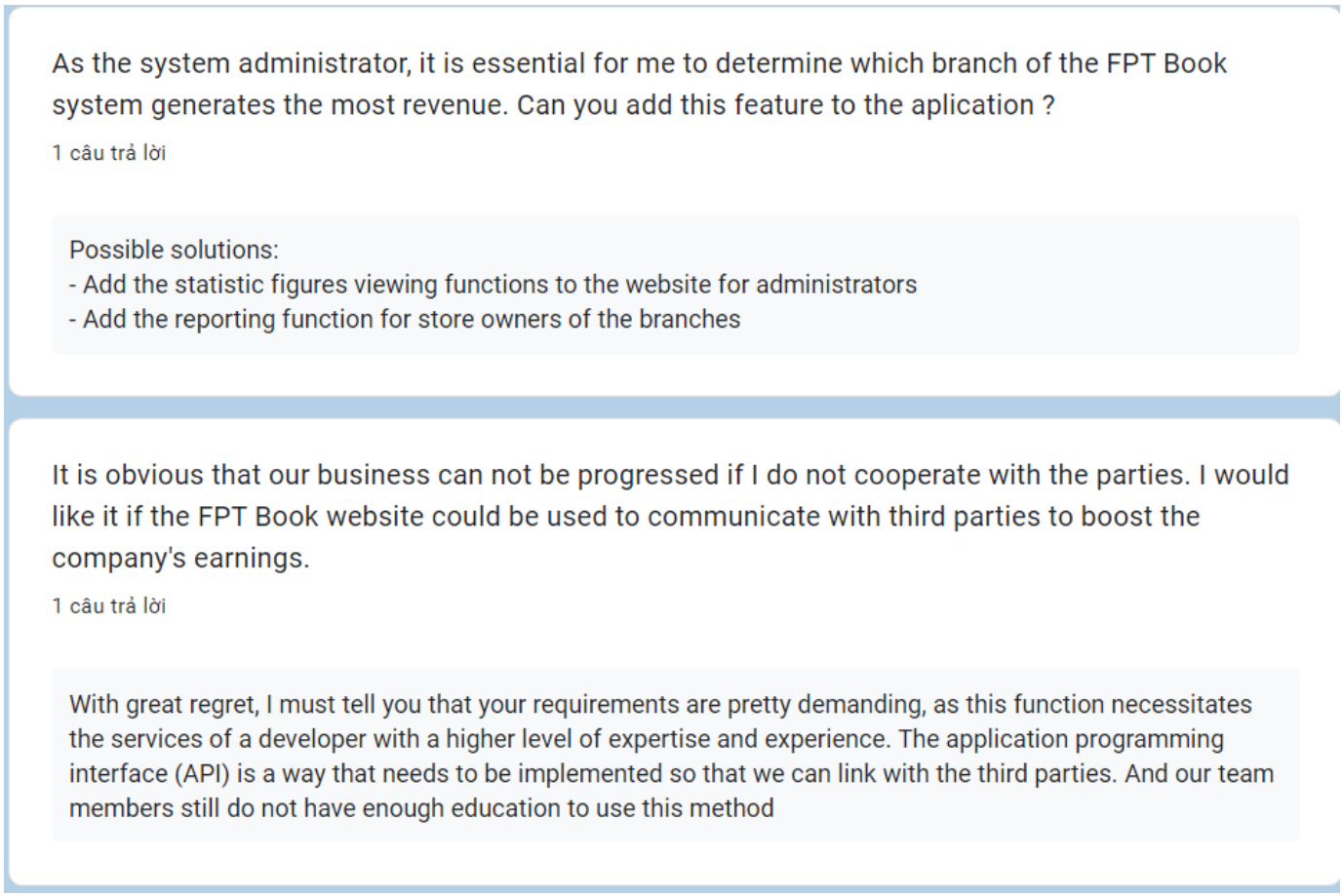
Câu trả lời của bạn

Figure 1: Questionnaires about the functions of the website

II. Collecting Review Feedbacks (P4)

Our team members, in addition to providing our customers, who are also administrators, with the functions pertinent to their duties, also review each other's responsibilities. As a functional developer for the store owner user position, I will present the functionality I have created in conjunction with my coworkers, who are accountable for developing functions for the customer and admin user roles. Our product must be reviewed and evaluated.

As a result, based on the above client questionnaire form that was developed (which will also be utilized as a part of a peer-review), our team has reviewed and briefly provided alternative ways to answer the issues presented above. The following is a list of the feedback that has been documented following the peer review:



One of the essential criteria of our business is that our shipping service must try our best to bring customers the most convenience. Therefore, I want to link with a third party so the system can provide many forms of payment for home delivery services.

1 câu trả lời

Regarding your requirements for a home delivery service, we have the appropriate solutions as follows:
- Hire a third party to help contact the home delivery service when an order is placed
- Integrate the home delivery service into the store's service, and create a functional page about the delivery service for customers to use the service

As an administrator of the FPT Book store, I have to know the system's profits as well as human resources for planning suitable award plans. Thus, I want to know which branch has the highest yield and who is the store owner of this branch.

1 câu trả lời

Regarding your requirements for this function, I have a few suggestions as follows:
- Add the function of revenue statistics by store branch, and display information about the store owner as well as quarterly, yearly, or monthly revenue

As an administrator of the system, I want an account-creating function for store owners instead of seeding the account information them into the system, which required the technical knowledge about Information Technology

1 câu trả lời

Regarding your requirements for this function, I have the following solutions:
- Add the ability to create an account for the store owner for the admin role.
- Add the ability to create an account for the store owner on the registration page, and the admin must approve the account before it can be signed in

Figure 2: Review Feedbacks

III. Interpreting Peer-Review Feedbacks (M3)

When I have finished collecting the input from the peer review, the next step is to interpret it so that the client may comprehend it and find opportunities that were not previously addressed. In particular, the following will serve as an explanation of the responses provided to customers about the fulfillment of their issues and requirements:

1. The problem is that the administrator wants to see which branch of the FPT book store chain has the highest revenue. To do this, we need:
   - Add a data reporting function to store owners' pages.
   - Add function to view and sort data by month, year, or quarter for administrators with other information such as time and address.
   - Create a sort function to sort revenue from low to high or vice versa for admins.
2. The problem is that the administrator of the system wants to cooperate with a third party to increase the store's revenue. In this regard, we have clearly stated that we are unable to meet the customer's request due to limited manpower and potential.
3. My customers want to link up with a third-party home delivery service to provide the best experience for their customers. In this regard, I think:
   - We can hire a third party to manage the shipping service and provide information about the customer's address when an order is placed.
   - We can also integrate the functionality of choosing a shipping service and time.
   - Shipping time and shipping rates are right on the order page.
   - From there, customers will be able to select the service that is right for them.
4. My client wants to understand the sales and staffing situation of the chain stores to plan the appropriate reward, especially for the store owners who earn the highest revenue in a period of time. Therefore, I think we should:
   - Add the function of revenue statistics by store branch
   - Add the function of displaying information about the store owner as well as quarterly, yearly, or monthly revenue

- Allows administrators to manipulate the above list, such as sorting and searching to filter out the necessary information for their needs.

5. My client wants to have the ability to create an account for the store owner instead of seeding the system. For this request, I think:
    - Add store owner account creation function for admin or store owner only
    - If an administrator creates the account, no authentication is required to work
    - Conversely, if a store owner creates the account, the admin will have to approve the account before it can be used

IV. Evaluate New Insights, Ideas Or Potential Improvements Of The Application (D2)

I find my website only meets the basic requirements of customers but has not met the needs of the current market. So I've listed my potential ideas and insights based on my assessments for future site development and explained why I didn't choose to include them in the development phase but be included in the list of potential development

No.	Improvements	Description	Explain the reason
1	**Responsive With All Devices**	In today's technology era, users are equipped with many devices such as smartphones, tablets, laptops, desktop computers, etc., to access the data source of the Internet. Therefore, the website's interface also requires improvement, suitable for many different screen sizes. Most users prioritize convenience when using websites, so they often tend to access websites on phones, tablets, or Ipads instead of computers. Therefore, a website needs to be responsive on multiple devices. It enhances the user experience and helps bring users optimal convenience during use.	I do not make a responsive website due to the short period and the workforce restriction. We only have about one month to gather users' requirements, plan a development phase, pick up technology, and make the application. Creating a responsive website requires a long time to test on multiple devices. With some of the vast scale projects, they may make a differentiate mobile apps for their business to bring customers the best experience.
2	**Attractive Website Interface**	Every user is attracted by a website with an elegant, professional, and attractive interface. Investing in interface images is very important because they help create an excellent first impression with customers and make them feel that the website they visit belongs to a business that cares and invests carefully. A website with a catastrophic interface design and layout will make customers frustrated and not want to use the service. An outdated web design will give the impression that you	The current look and feel of our website are at an acceptable level. We didn't spend much time on the interface because we only had a very short deadline to complete a basic, fully functional e-commerce website system for three user roles. Use. Moreover, to have a website with a gorgeous and attractive interface, we have to delve deeper into front-end programming languages like React JS or Vue JS, which takes more time to complete. Therefore, we could only create a

		don't care about your business either. Therefore, creating an attractive website is very important, as it will help attract customers at first sight.	website with an average interface. In the future, we will study and improve the look and feel of the website to make it look professional and attractive.
3	**Customer Discussions**	Another vital aspect of e-commerce websites is allowing customers to discuss and rate their services. In popular e-commerce sites, customer comments and reviews on each item are paramount. Because it helps business owners to understand clearly the problems or problems that their customers are facing or the things that make their customers unsatisfied while using the service; moreover, displaying good reviews of customers who have used the service will make new customers of the store better understand the service quality of the store. The best advertising strategy is for customers to advertise the service they experience.	As I mentioned in the previous issues, time and workforce constraints were also why we could not include this functionality in the project. Creating customer reviews and viewing other reviews requires us to scale up our database and use real-time reflection technology, which involves much time for research and development. Therefore, in the future, this is also a potential development that FPT book business owners should consider in developing their websites.
4	**User-friendly Layouts**	One of the factors that makes a website successful is easy access for all segments of customers. Some customers are willing to spend much time thoroughly understanding the functions of a website; on the contrary, some customers who are elderly or lack information technology knowledge will feel discouraged and do not	Currently, our website has too little information to display. Therefore, accessing the functions on the website seems to be relatively easy for users. However, in the future, when business owners want to grow their store and the size of the website increases, it means that the number of functions and

		want to use a complicated website with lots of distracting information. Therefore, it is essential to arrange the layout and functionality of a website so that it is accessible to new users. Because the top criterion of a website is the ease of use for all ages and all customer segments	the layout of the website will become complicated to approach with new users. Thus, at that time, I think it is necessary to change and arrange the design of the website components.
5	**Linked To Other Platforms**	The development of different social networking accounts such as Youtube, Facebook, Instagram, and Twitter is essential. Today, with the widespread popularity of social networks, allowing customers to link their accounts to social networking sites will make it more convenient for customers to use the website. Linking to social networking sites will allow customers to share products from the website with their friends and acquaintances. In addition, these channels also contribute to increasing website traffic and positively impacting SEO.	Developing functionality to connect to social networks requires us to use APIs - a new technology we need to spend a lot of time researching. However, we could not complete this function in time for the deadline that our customers required. However, in the future, I think this is also a useful function that FPT Book business owners should consider adding to their website.

Chapter 2 – Application Development (P5 – M4)

I. Folder Structure (P5)

Here is the folder structure of our project

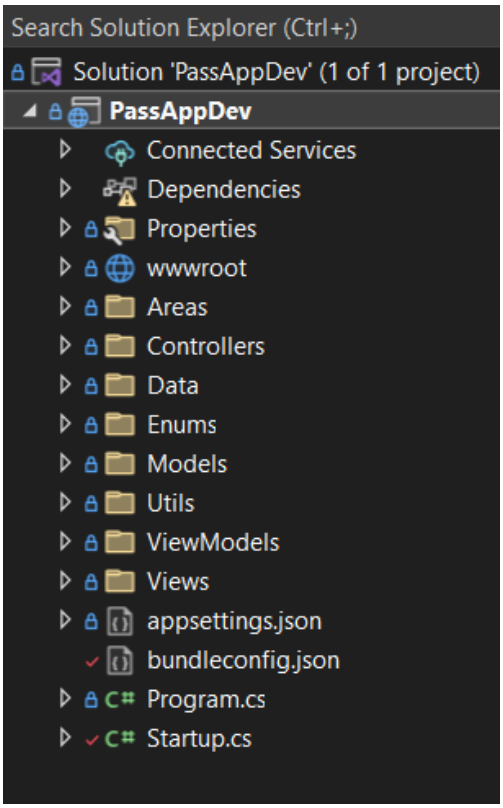


Figure 3: Folder Structure

- The **Areas** folder contains the sub-folders and files about the login/ register function of the project.
- The **Controllers** folders contain all the controller's files of the project, which are responsible for organizing the programming flow of the project.
- The **Data** folders are used to save all files created when users change the database; these steps are also called migrations.
- The **Enums** folders are used to save the file that enumerates the status of components in the project.

- The **Models** folders contain the project's model files, which are responsible for storing all the application's data and are a bridge between the two components, View and Controller.
- The **Utils** folder contains the constant values of the project-like roles.
- The **ViewModels** folder contains the view models responsible for displaying the information to the application views.
- The **Views** folder is used to contain the view files, which are responsible for displaying the information to the application views.

II. Source Code Examples (M5)

In this section, I will represent some of feature functions in my project.

1. Creating A New Book

```
[HttpPost]
public async Task<IActionResult> CreateAsync(BookCategoriesViewModel viewModel)
{
  if (!ModelState.IsValid)
  {
    viewModel = new BookCategoriesViewModel
    {
      Categories = _context.Categories
        .Where(t => t.Status == Enums.CategoryStatus.Approved)
        .ToList()
    };
    return View(viewModel);
  }

  using (var memoryStream = new MemoryStream())
  {
    await viewModel.FormFile.CopyToAsync(memoryStream);

    var newBook = new Book
    {
      Title = viewModel.Book.Title,
      Author = viewModel.Book.Author,
      Price = viewModel.Book.Price,
      Description = viewModel.Book.Description,
      CategoryId = viewModel.Book.CategoryId,
      ImageData = memoryStream.ToArray()
    };

    _context.Add(newBook);
    await _context.SaveChangesAsync();
  }

  return RedirectToAction("Index");
}
```

Figure 4: Create Book Function

This is the code snippet of the book creating function. First, the program will check whether the data entered is valid. If not, the program will display a screen informing the user that the data they entered is not valid. Conversely, if the input data is valid, the program will receive all the data such as book title, author, price, description, genre, and image and save it in the database to create a new book. Then, it return to the homepage

2. Edit Book's Information

```
[HttpPost]
public async Task<IActionResult> Edit(BookCategoriesViewModel viewModel)
{
  var bookInDb = _context.Books.SingleOrDefault(t => t.Id == viewModel.Book.Id);
  if (bookInDb is null)
  {
    return BadRequest();
  }

  if (!ModelState.IsValid)
  {
    viewModel = new BookCategoriesViewModel
    {
      Book = viewModel.Book,
      Categories = _context.Categories
        .Where(t => t.Status == Enums.CategoryStatus.Approved).ToList(),
    };

    ViewBag.ImageData = ConvertByteArrayToStringBase64(bookInDb.ImageData);
    return View(viewModel);
  }
  bookInDb.Title = viewModel.Book.Title;
  bookInDb.Author = viewModel.Book.Author;
  bookInDb.Price = viewModel.Book.Price;
  bookInDb.Description = viewModel.Book.Description;
  bookInDb.CategoryId = viewModel.Book.CategoryId;


  if (viewModel.FormFile != null)
  {
    using (var memoryStream = new MemoryStream())

    {
      await viewModel.FormFile.CopyToAsync(memoryStream);

      if (memoryStream != null)
        bookInDb.ImageData = memoryStream.ToArray();
    }

  }
  await _context.SaveChangesAsync();
  return RedirectToAction("Index");
}
```

Figure 5: Edit book function

This is the code snippet of the book creating function. First, the program will check whether the book is existed in the database by book ID. If not, the program will notify the user that the book does not exist. Conversely, if the book exists, the program will display all of the existed book's information to the users. Then it will check the validation of the input information that the user wants to update. If the input is invalid, the program will return the ex-book information with the warning alert. If the input is valid, it will receive all the data such as book title, author, price, description, genre, and image and update it in the database to replace the ex information. Then, it returns to the homepage.

3. Add Item To Cart

```
public async Task<IActionResult> AddToCart(int id)
{
  var bookInStore = _context.Books.SingleOrDefault(t => t.Id == id);
  var currentUserId = _userManager.GetUserId(HttpContext.User);

  var bookInCart = _context.CartBooks.SingleOrDefault(
      t => t.ApplicationUserId == currentUserId &&
      t.BookId == id);
  if (bookInCart == null)
  {
    var cartBook = new CartBook()
    {
      ApplicationUserId = currentUserId,
      BookId = id,
      Quatity = 1,
      Price = bookInStore.Price
    };
    _context.Add(cartBook);
  }
  else
  {
    bookInCart.Quatity++;
  }
  await _context.SaveChangesAsync();
  return RedirectToAction("Index");
}
```

Figure 6: Add item to cart

This is the code snippet of the adding book to cart function. First, the program checks whether the book is already in the user's cart by examining the book and user id. If the book chosen by the user is in the cart, the quantity of this book will increase. In contrast, if the book is not in the cart yet, it will be added to the cart with quantity 1. Then the program will return to Index.

4.Search and Show Home Page

```
public IActionResult Index(string keyWord)
{
  var currentUserId = _userManager.GetUserId(HttpContext.User);
  BookNotificationViewModel bookNotification = new BookNotificationViewModel();
  IEnumerable<Notification> notifications = _context.Notifications.Where(t => t.ApplicationUserId == currentUserId)
    .OrderByDescending(t => t.NotifiedAt).ToList();
  var bookVMList = new List<BookViewModel>();
  if (!string.IsNullOrWhiteSpace(keyWord))
  {
    var result = _context.Books
      .Include(t => t.Category)
      .Where(t => t.Category.Name.Contains(keyWord)
          || t.Title.Contains(keyWord)
      )
      .ToList();

    foreach (var book in result)
    {
      string imageBase64 = Convert.ToBase64String(book.ImageData);

      string image = string.Format("{0}", imageBase64);

      var newbookVM = new BookViewModel()
      {
        Title = book.Title,
        Author = book.Author,
        Price = book.Price,
        Id = book.Id,
        ImageUrl = image
      };
      bookVMList.Add(newbookVM);
    }
    bookNotification.BookViewModels = bookVMList.AsEnumerable();
    bookNotification.Notifications = notifications;
    return View(bookNotification);
  }
```

```
IEnumerable<Book> books = _context.Books
  .Include(t => t.Category)
  .ToList();

foreach (var book in books)
{
  string imageBase64 = Convert.ToBase64String(book.ImageData);

  string image = string.Format("{0}", imageBase64);

  var newbookVM = new BookViewModel()
  {
    Title = book.Title,
    Author = book.Author,
    Price = book.Price,
    Id = book.Id,
    ImageUrl = image
  };
  bookVMList.Add(newbookVM);
}
bookNotification.BookViewModels = bookVMList.AsEnumerable();
bookNotification.Notifications = notifications;
return View("~/Views/Home/Index.cshtml", bookNotification);
```

Figure 7: Search and show home page

This is the code snippet showing the home page and searches function. First, the program checks whether any keywords are typed in the search box. If it has, the program will access the database to get all the book's information with the category or name matched with the keyword in the search box. If there is nothing in the search box, it will show all of the books in the store's database. Moreover, in case the user signs in to the website as the store owner, they will see a notification icon, which can be clicked on to show the notification about the status of the category.

5. View Order Book

```
public async Task<IActionResult> Order()
{
    var currentUserId = _userManager.GetUserId(HttpContext.User);
    var booksInCart = _context.CartBooks.Where(t => t.ApplicationUserId == currentUserId).ToList();
    if (!booksInCart.Any())
    {
        return RedirectToAction("Index");
    }
    var newOrder = new Order
    {
        ApplicationUserId = currentUserId
    };
    _context.Add(newOrder);
    await _context.SaveChangesAsync();
    var orderInDb = _context.Orders.OrderByDescending(t => t.Id).First();
    var booksInOrder = new List<OrderedBook>();
    foreach (var item in booksInCart)
    {
        var newOrderedBook = new OrderedBook
        {
            OrderId = orderInDb.Id,
            BookId = item.BookId,
            Quantity = item.Quatity,
            Price = item.Price
        };
        booksInOrder.Add(newOrderedBook);
    }

    _context.OrderedBooks.AddRange(booksInOrder);
    _context.RemoveRange(booksInCart);
    await _context.SaveChangesAsync();
    return RedirectToAction("Index");
}
```

Figure 8: Order Function

This is the code snippet showing the list of order items. First, the program checks whether there is any book in the cart; if not, it will return to the Index page. If a book exists in the cart, it will create a new list of ordered items by customer ID, add the items from the shopping cart to the latest list of orders, and then remove this list from the customers' shopping cart.

III. Screenshot Of The Website (M5)

1. Unregistered Users

➢ Home Page

This is the home page of the app which will be displayed to users when they access to the web page

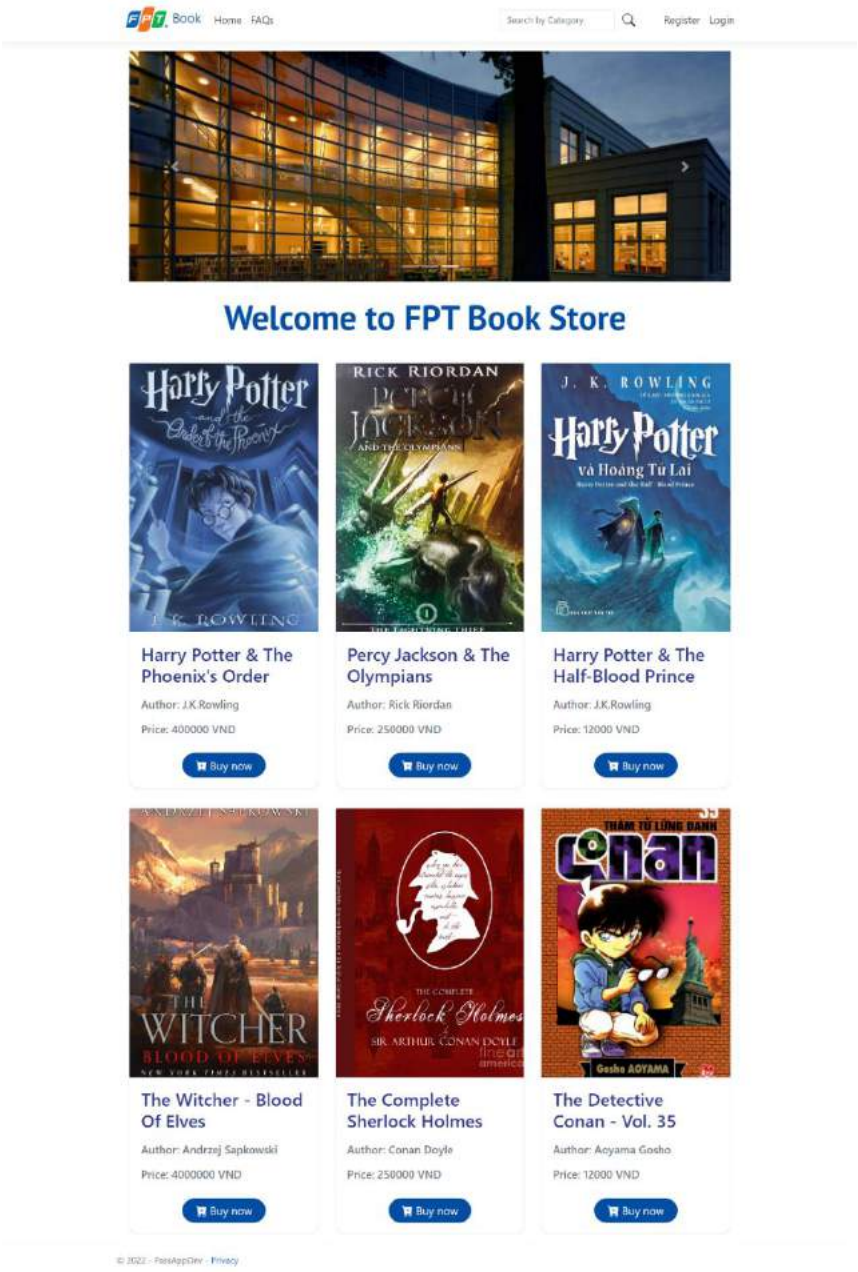


Figure 9: Home Page screenshot

➢ Book Detail Page

This is the detail page which take responsibility for displaying full information of a book in the store

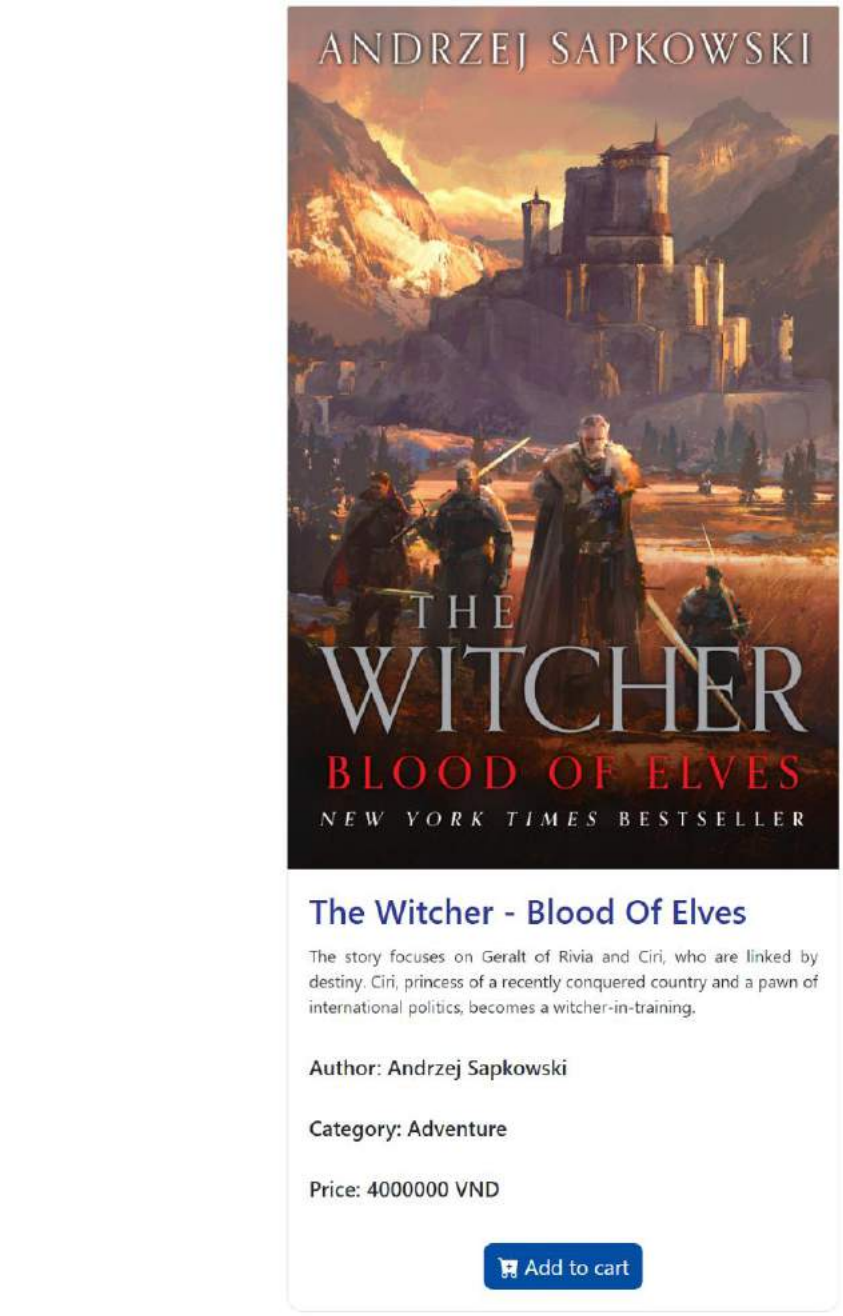


Figure 10: Book Detail page

➢ Register Page

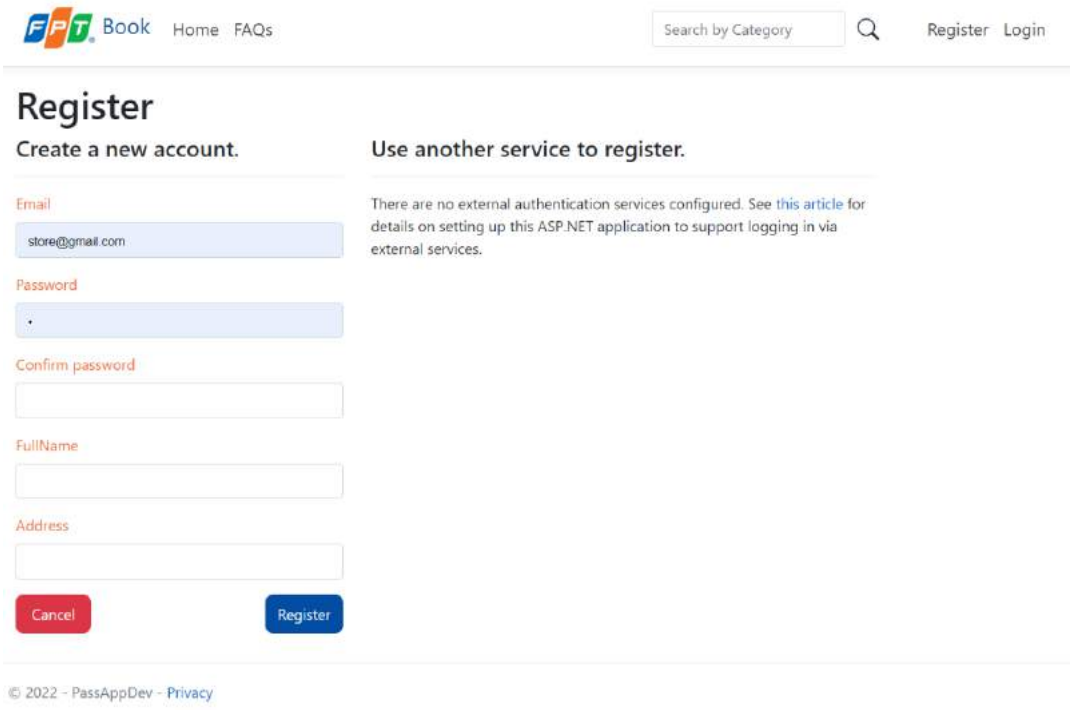


Figure 11: Register Page

➢ Login Page

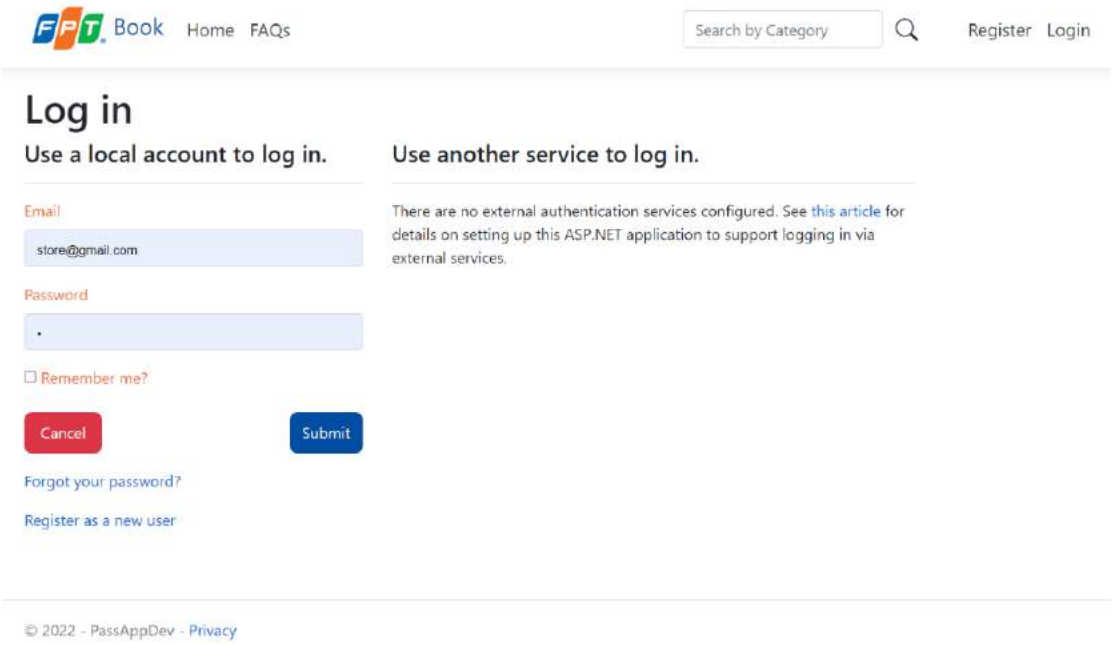


Figure 12: Login Page

2. Customer Role

➢ Home Page

This is the home page of the customer role, when a user sign in to the website as a customer, they will see this home page

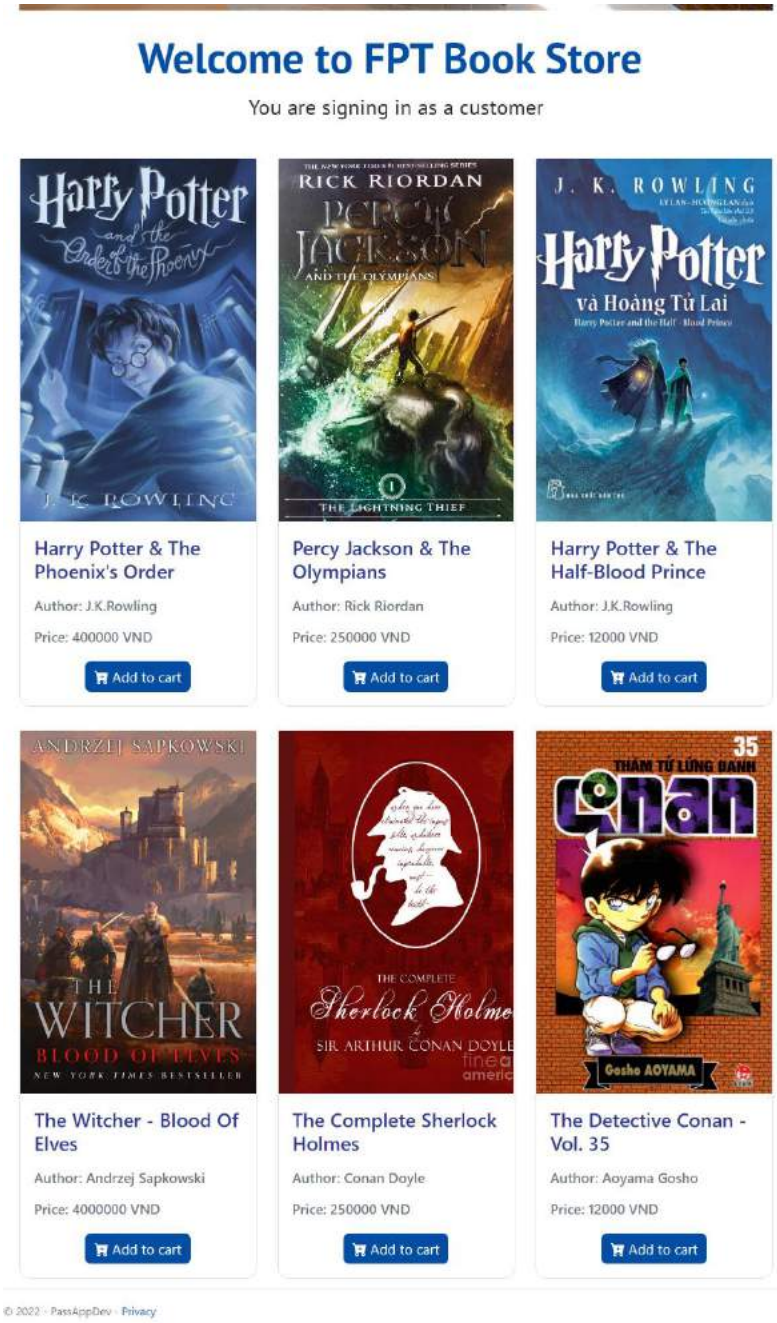


Figure 13: Customer Home Page

➢ FAQs Page

This page displays all of the frequent questions and answers of customers

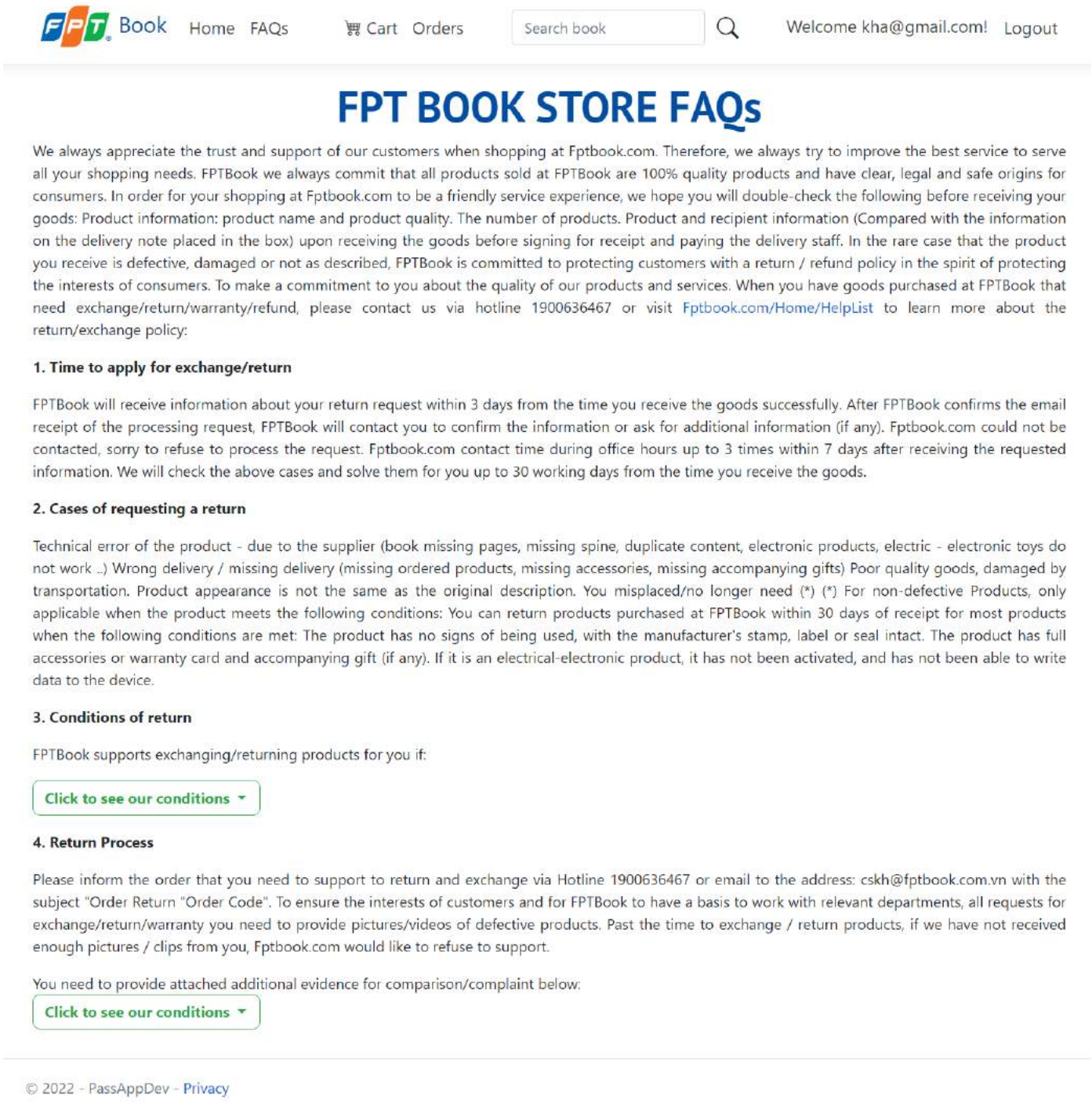


Figure 14: FAQs Page

➢ Shopping Cart Page

This is the shopping cart page which displayed list of item that customer want to purchase

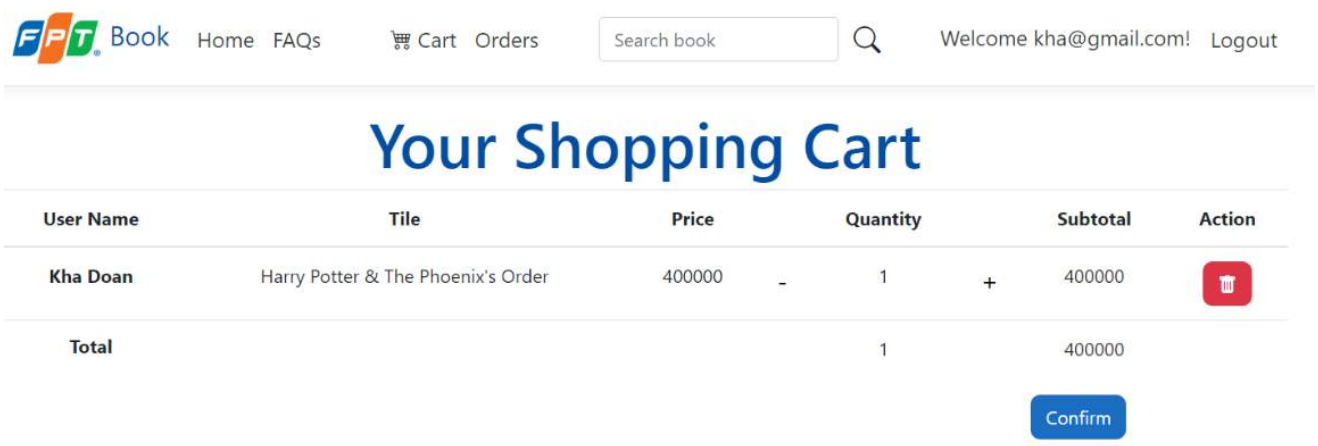


Figure 15: Shopping Cart Page

➢ Order Page

This page is responsible for displaying list of orders of a customer

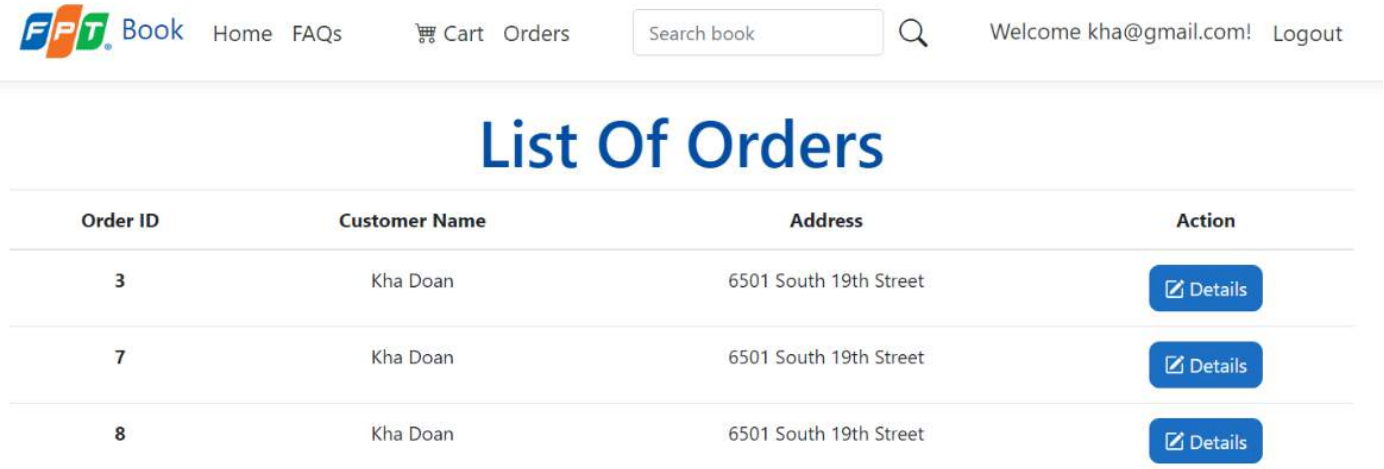


Figure 16: Order List Page

➢ Order Detail Page

This page displays the detail of a certain order by order ID

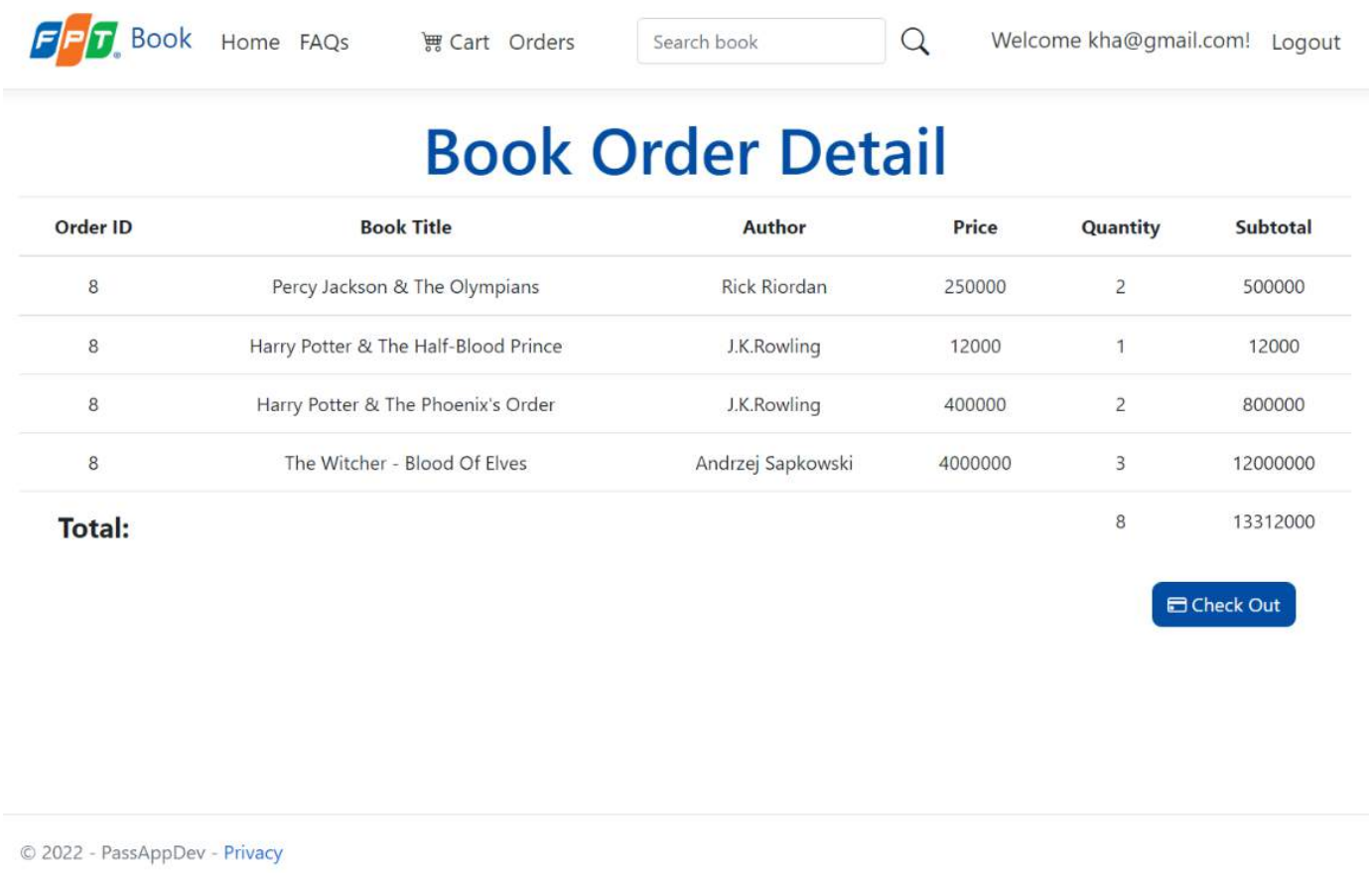


Figure 17: Order Detail Page

3.Store Owner Role

➢ Home Page

This page displays the home page for users who sign in as the store owners

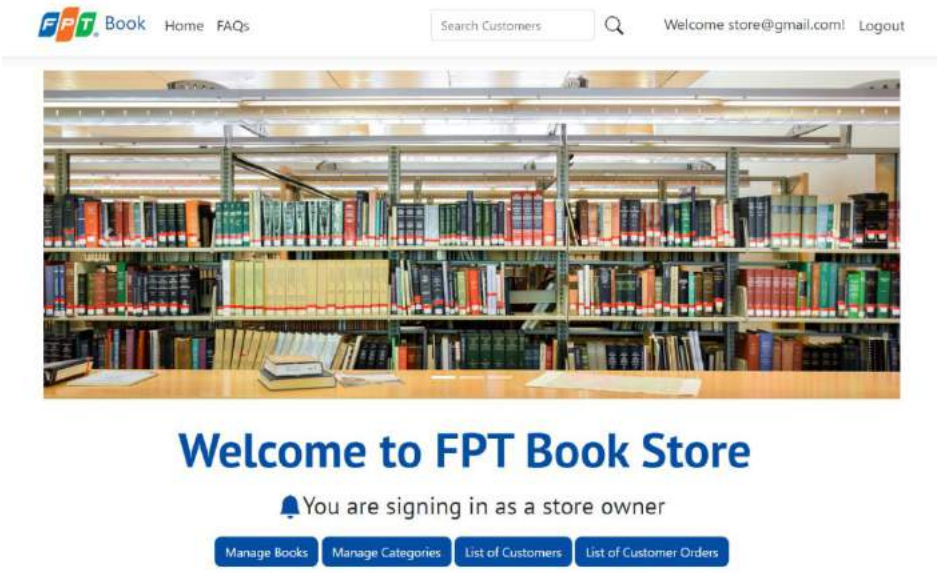


Figure 18: Store Owner Home Page

➢ Notification

This is the section used to display the status of category requests of store owners

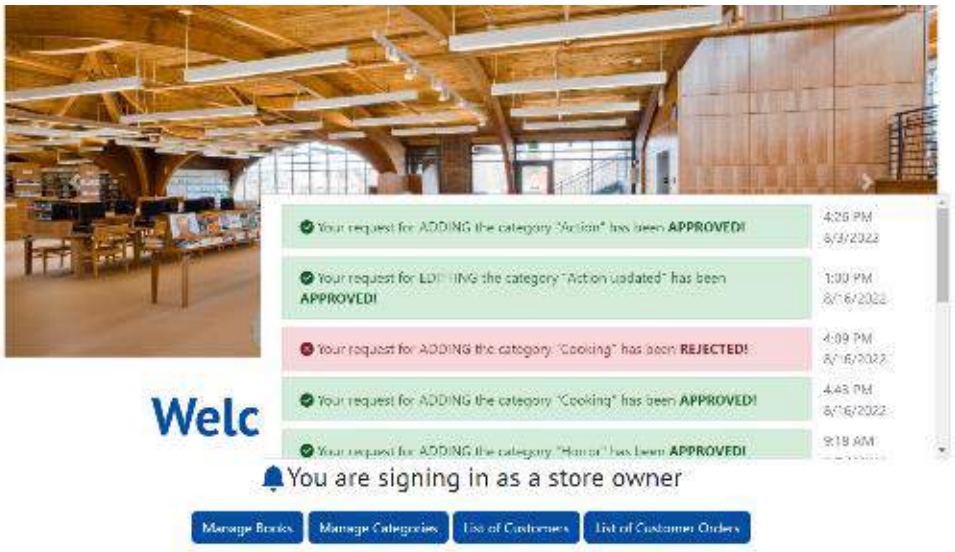


Figure 19: Notification

➢ Book Management Page

This page is used to display the list of book in the store

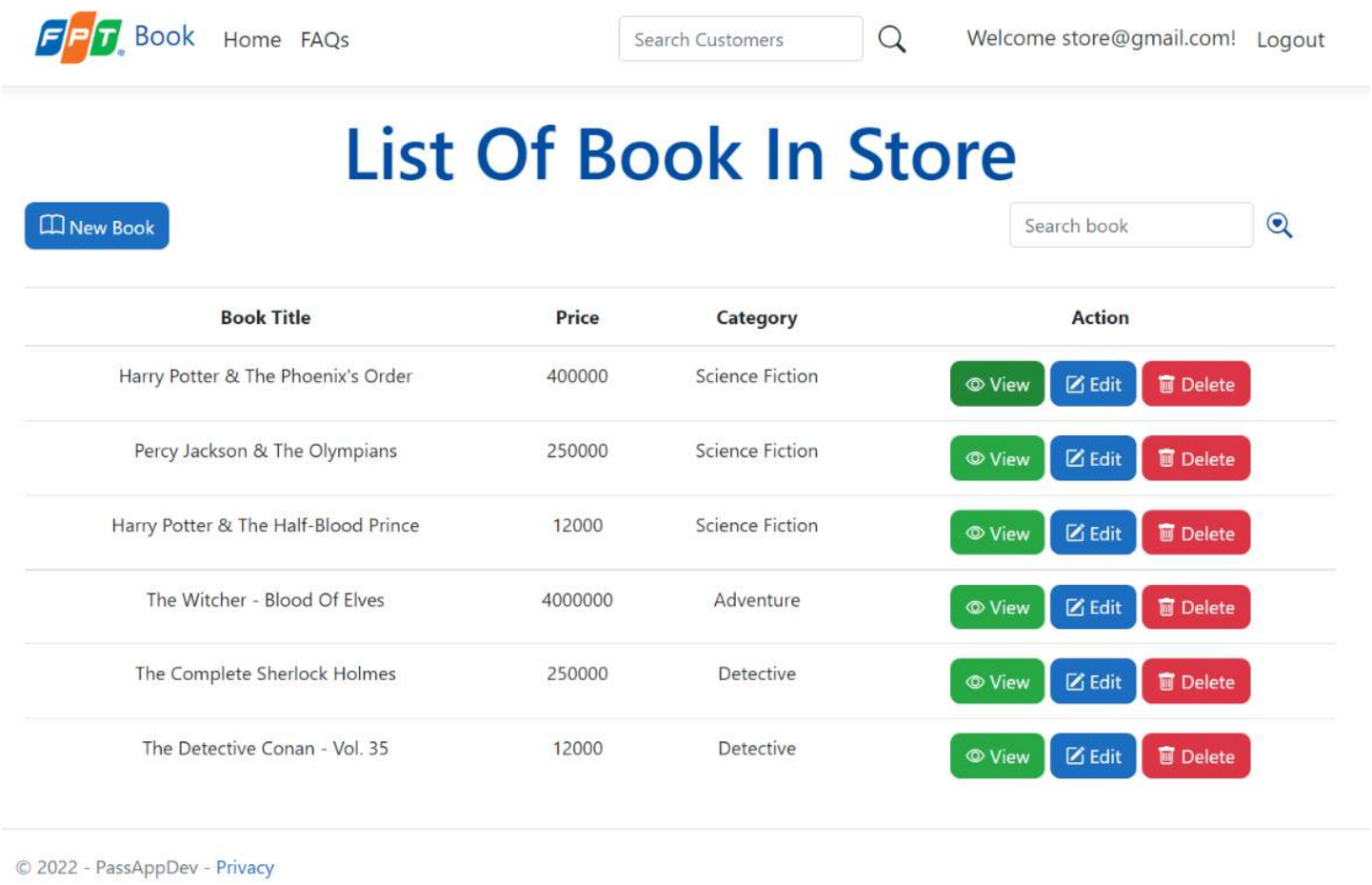


Figure 20: Book Management Page

➢ Create Book Page

This page is used to display a form to add new book to the store

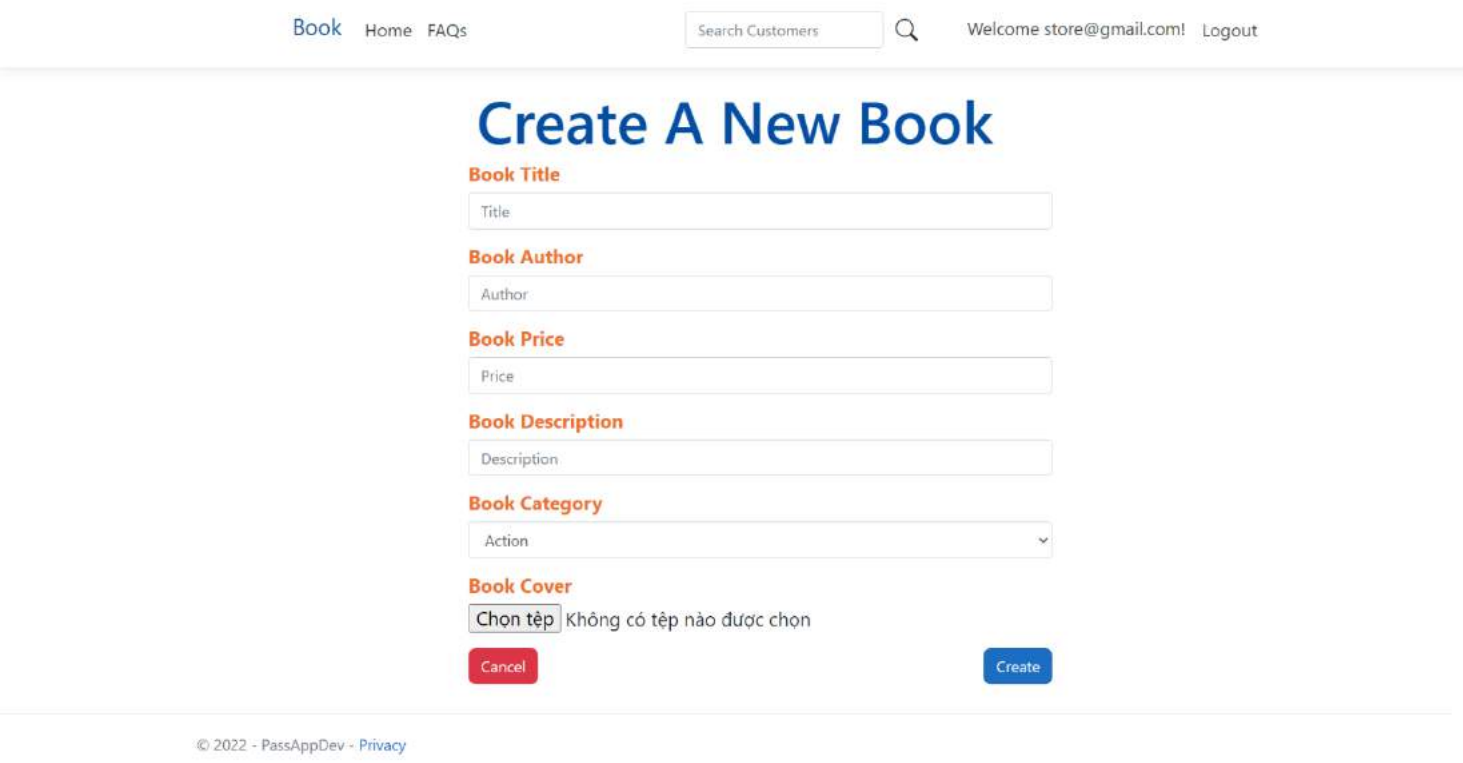


Figure 21: Create Book Page

➢ View Book Detail Page

This page is used to display the book detail for store owner

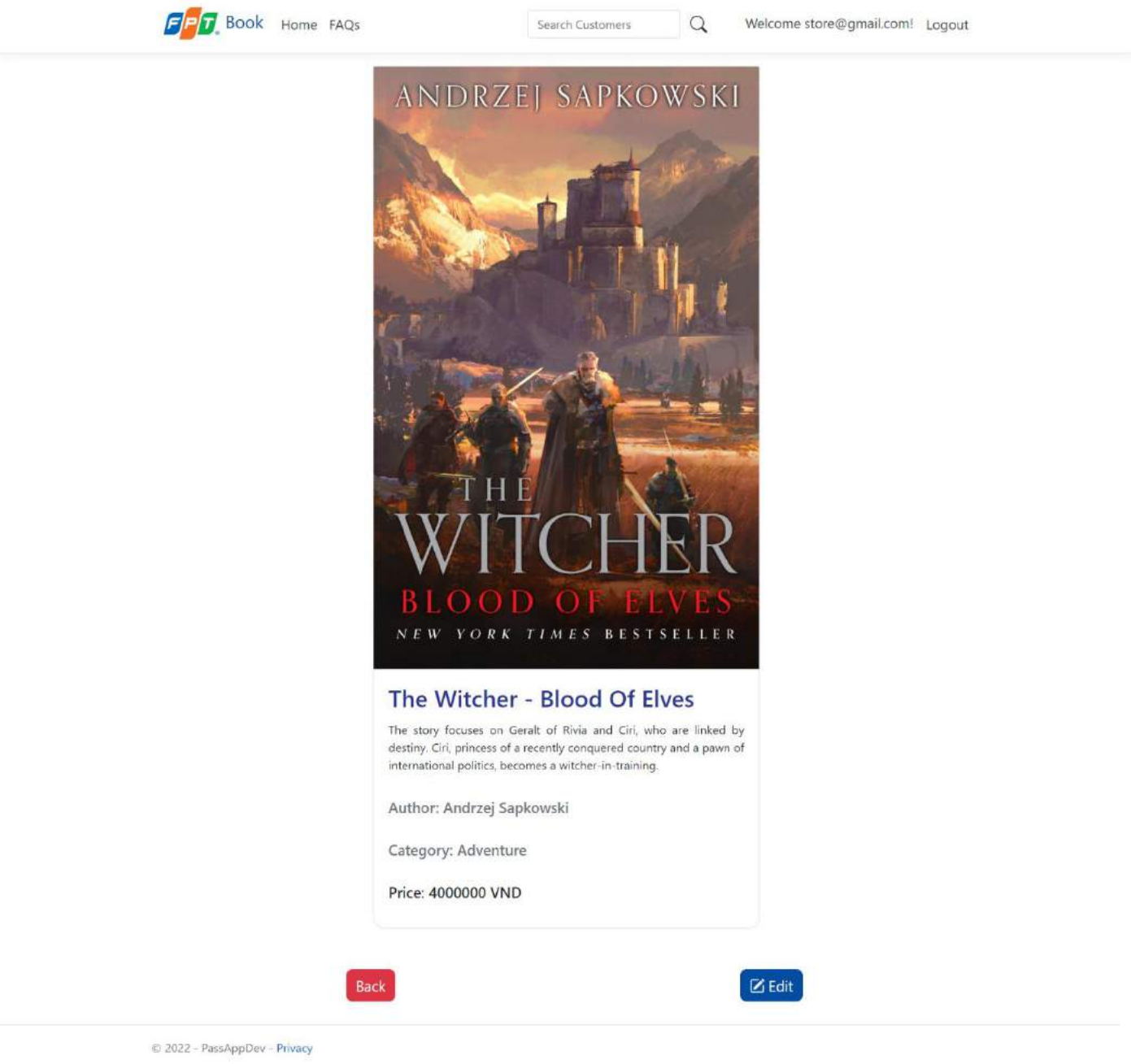


Figure 22: View Book Detail Page

➢ Edit Book Information Page

This page is used to allow store owner to change the book's information

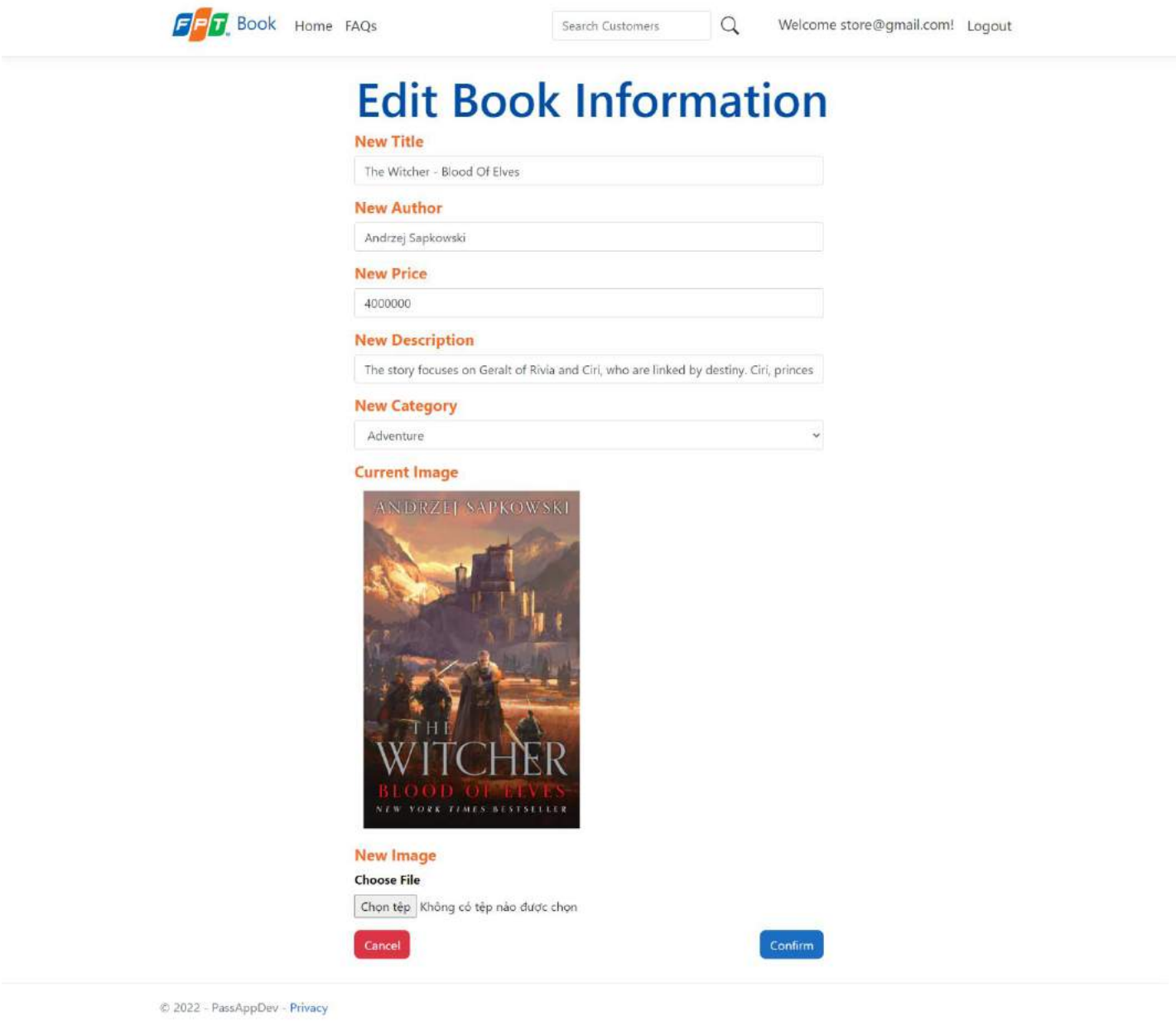


Figure 23: Edit Book Information Page

➢ Category Management Page

This page is used to show list of categories to the store owner

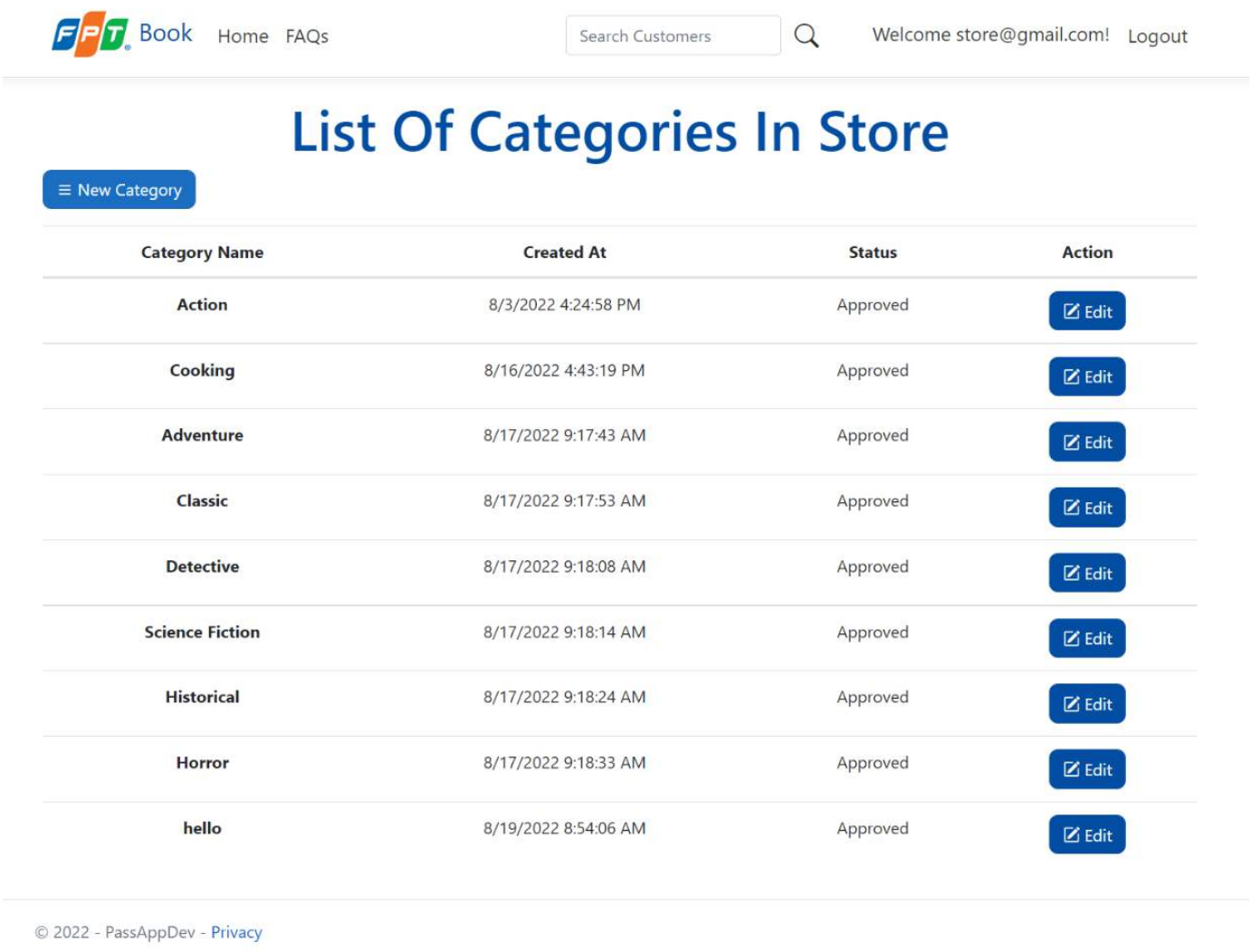


Figure 24: Category Management Page

➢ Create New Category Page

This page is used for the store owner to create a new cateogry

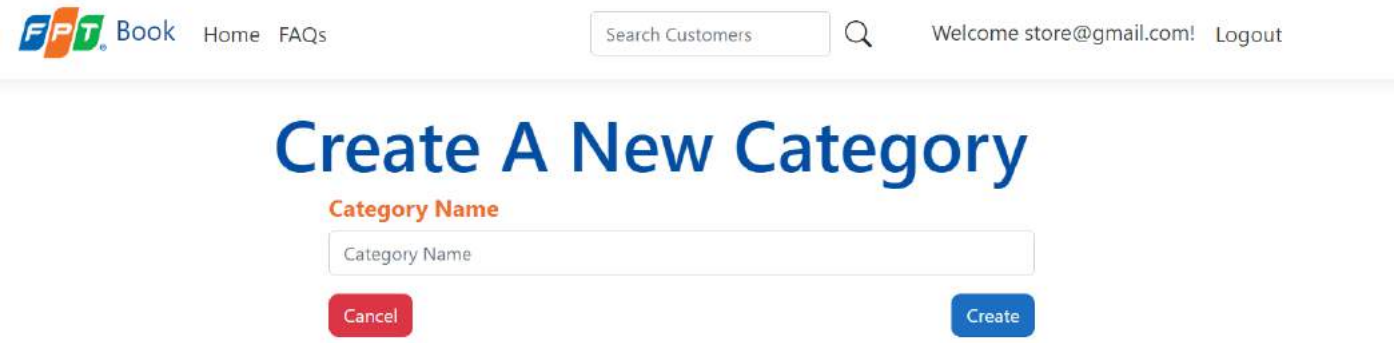


Figure 25: Create Category Page

➢ Edit Category Information Page

This page is used for the store owner to change a cateogry's name

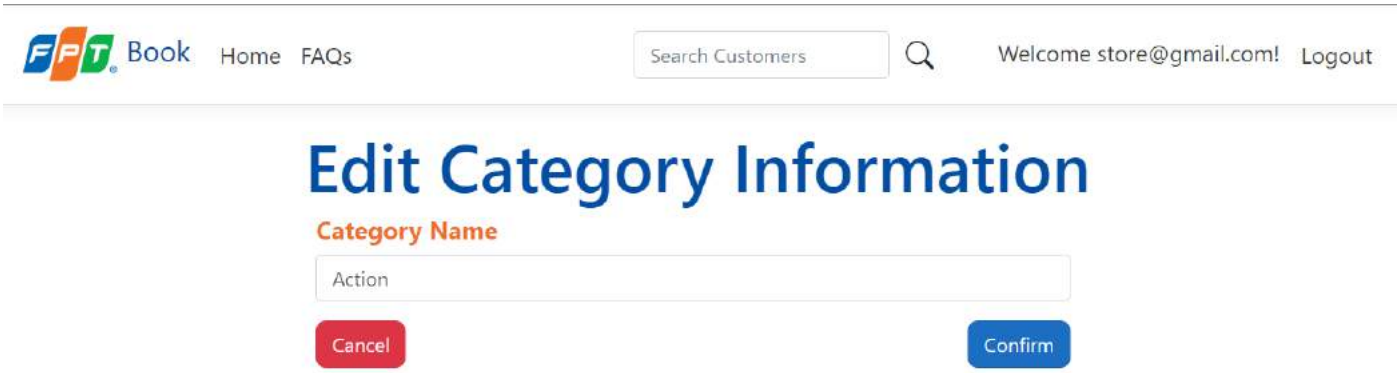


Figure 26: Edit Category Information Page

➢ List Of Customer's Orders Page

This page is used to display list of customer's orders for store owners

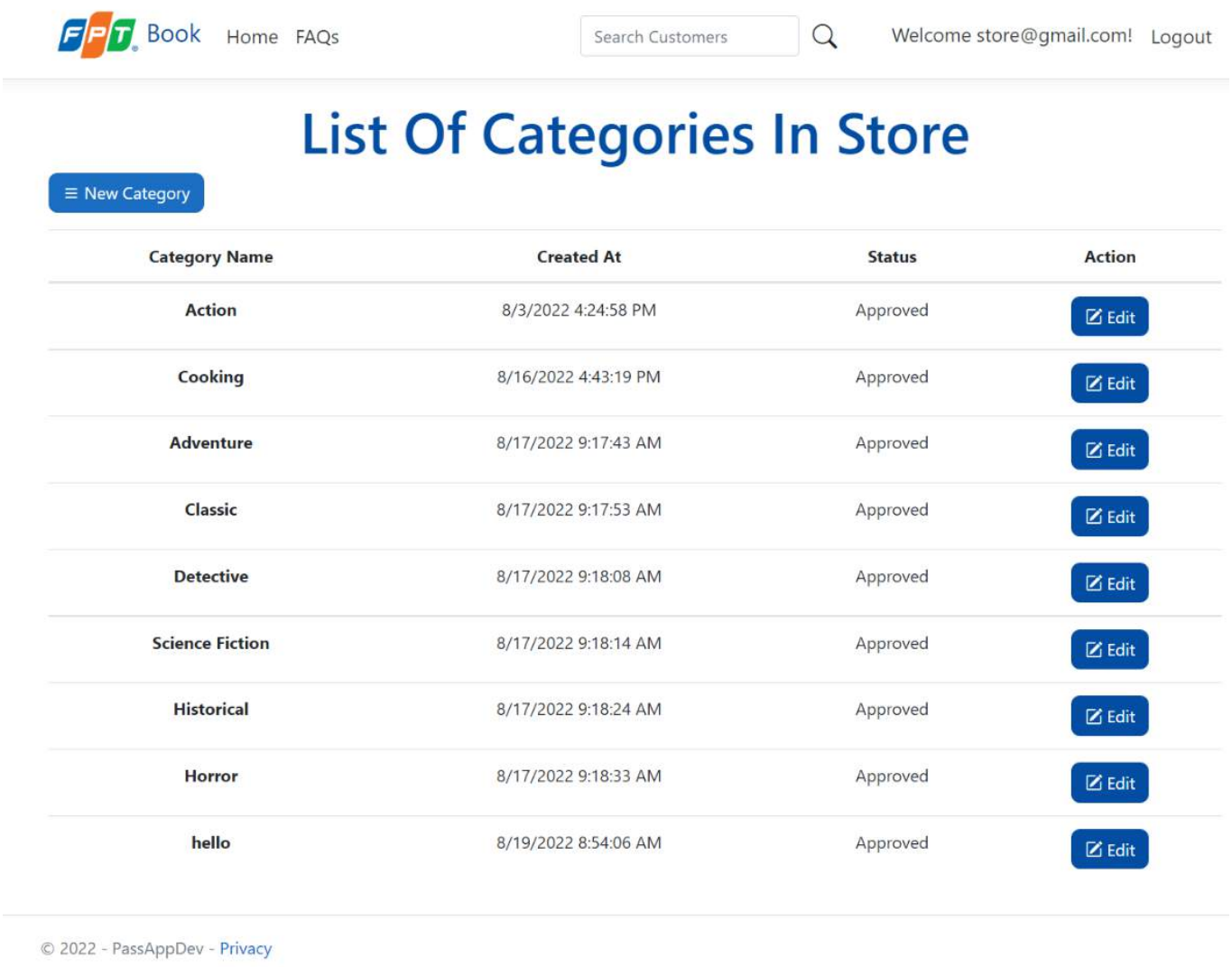


Figure 27: List Of Customer's Orders Page

➢ List Of Customer Page

This page is used to display list of customers in the store for store owner

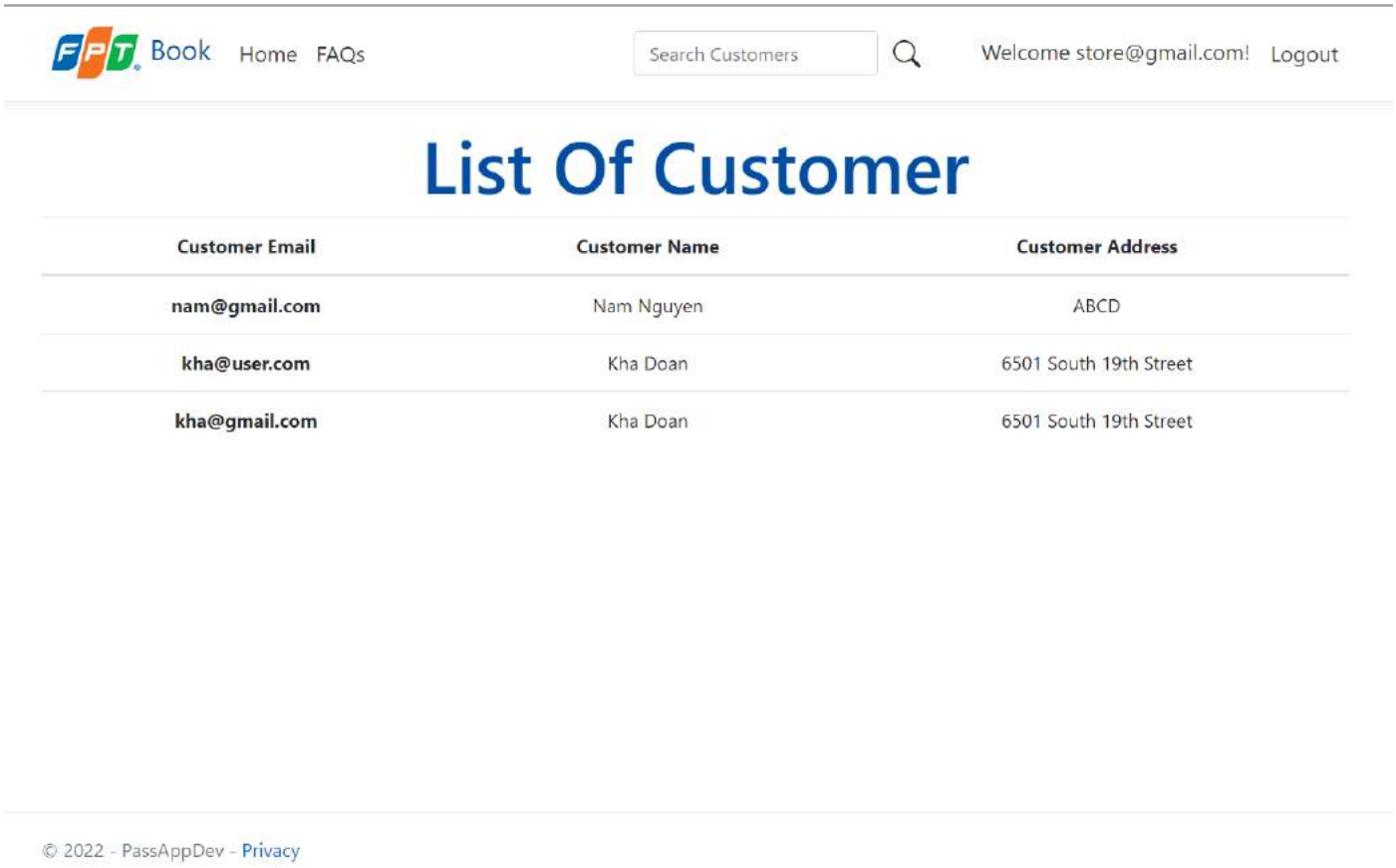


Figure 28: List Of Customer Page

4.Administrator Role

➢ Home Page

This is the home page which displayes to user when they sign in as the administrators

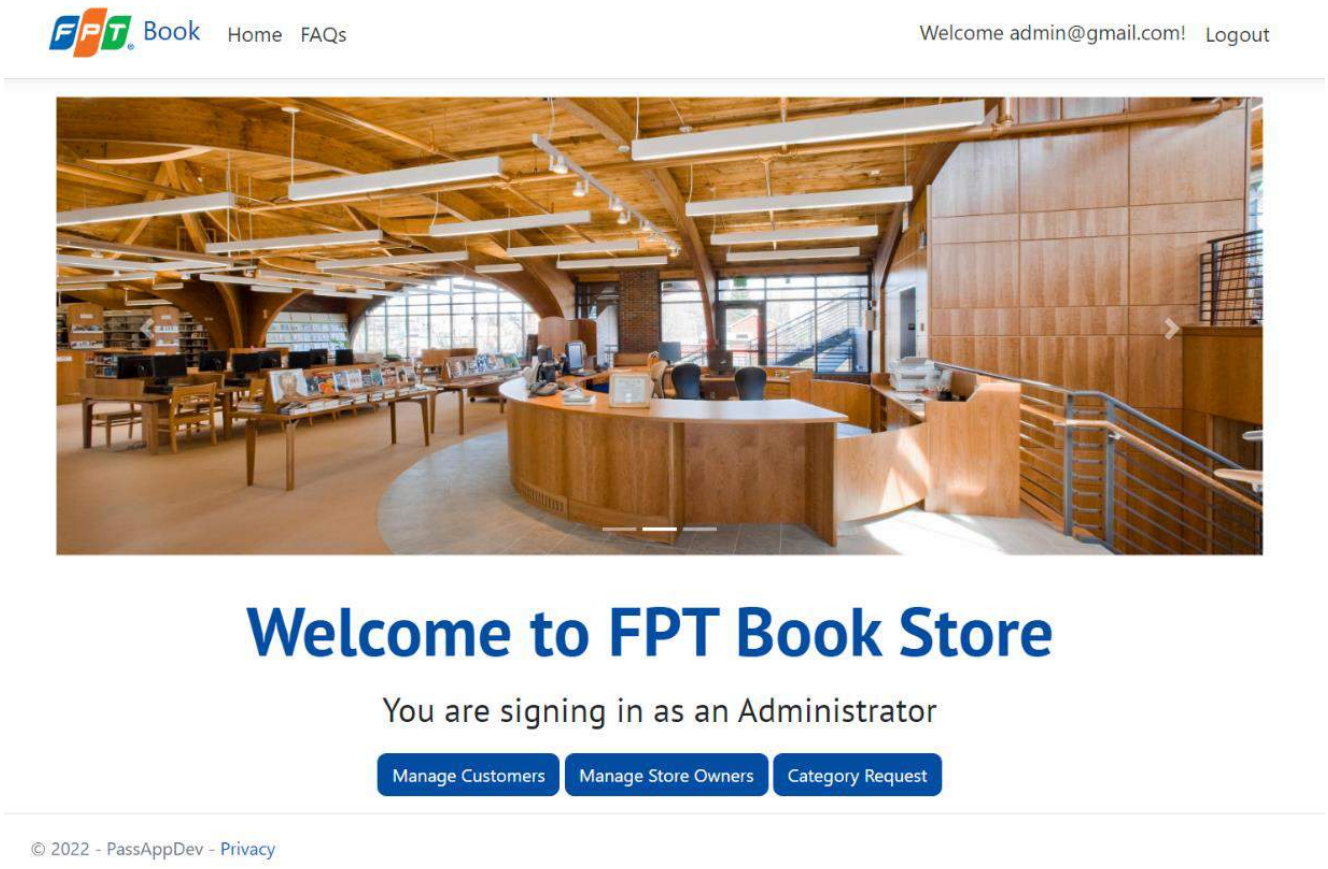


Figure 29: Home Page Of Administrator

➢ Customer Management Page

This page is used to display list of customer in the store for administrators

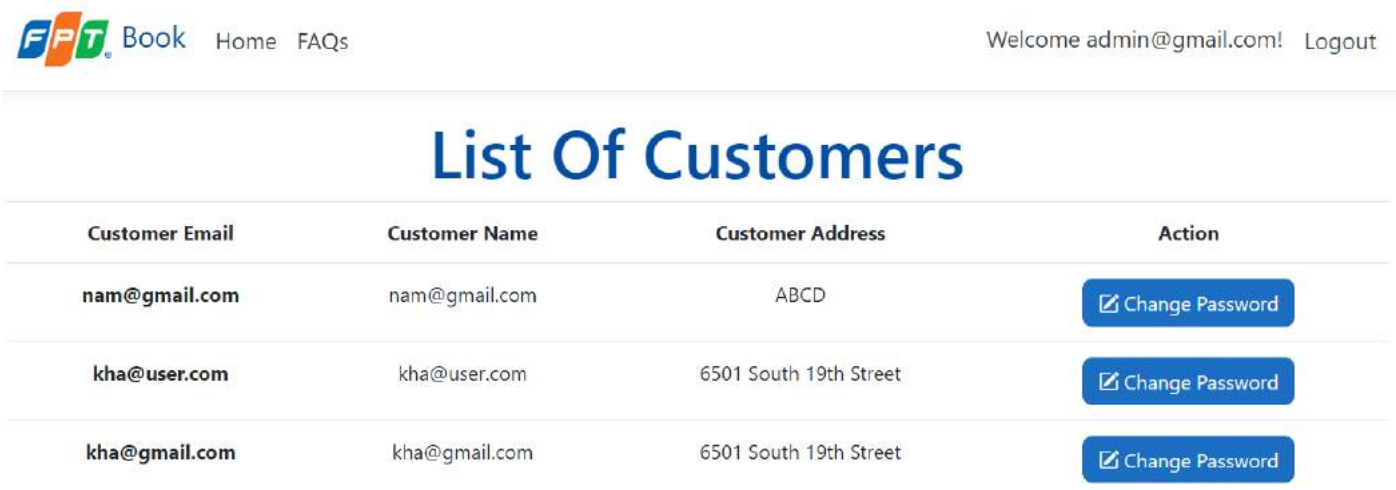


Figure 30: Customer Management Page

➢ Change Password For Customer Page

This page is used to change password for customer accounts in the store for administrators by customer ID

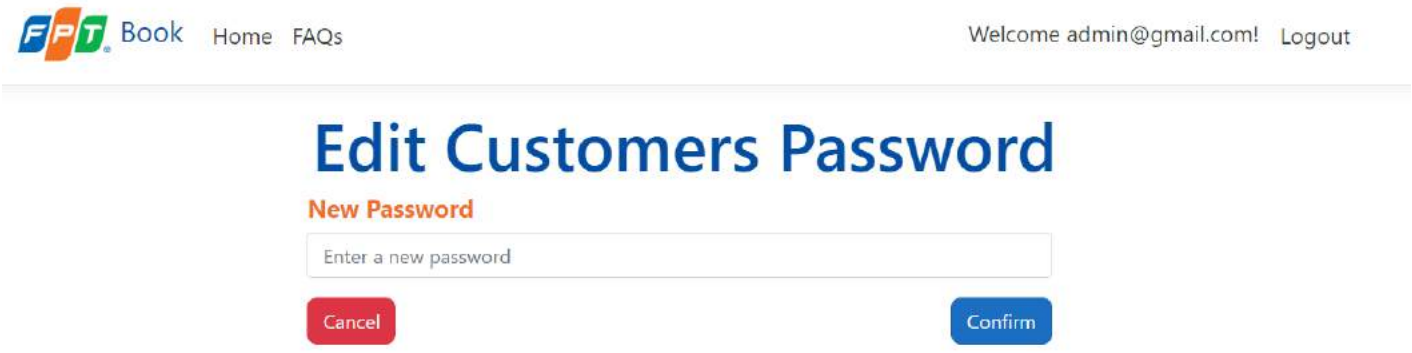


Figure 31: Change Password For Customer Page

➢ Store Owner Management Page

This page is used to display list of store owner accounts in the store for administrators

Figure 32: Store Owner Management Page

➢ Change Password For Customer Page

This page is used to change password for store owner accounts in the store for administrators by user ID

Figure 33: Change Password For Store Owner Page

➢ Category Requets Page

This page is used to display list of waiting categories from store owner accounts in the store for administrators to approve

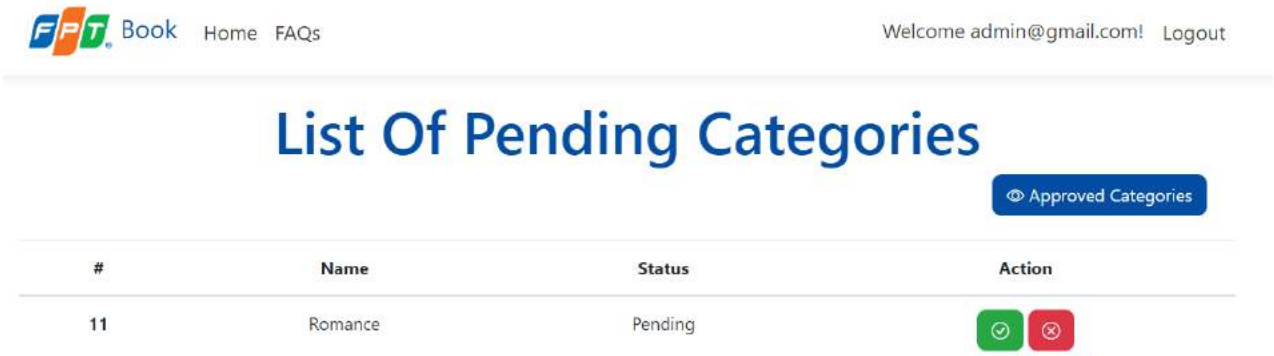


Figure 34: Category Requets Page

➢ Approved Categories Page

This page is used to display list of approved categories from store owner accounts in the store

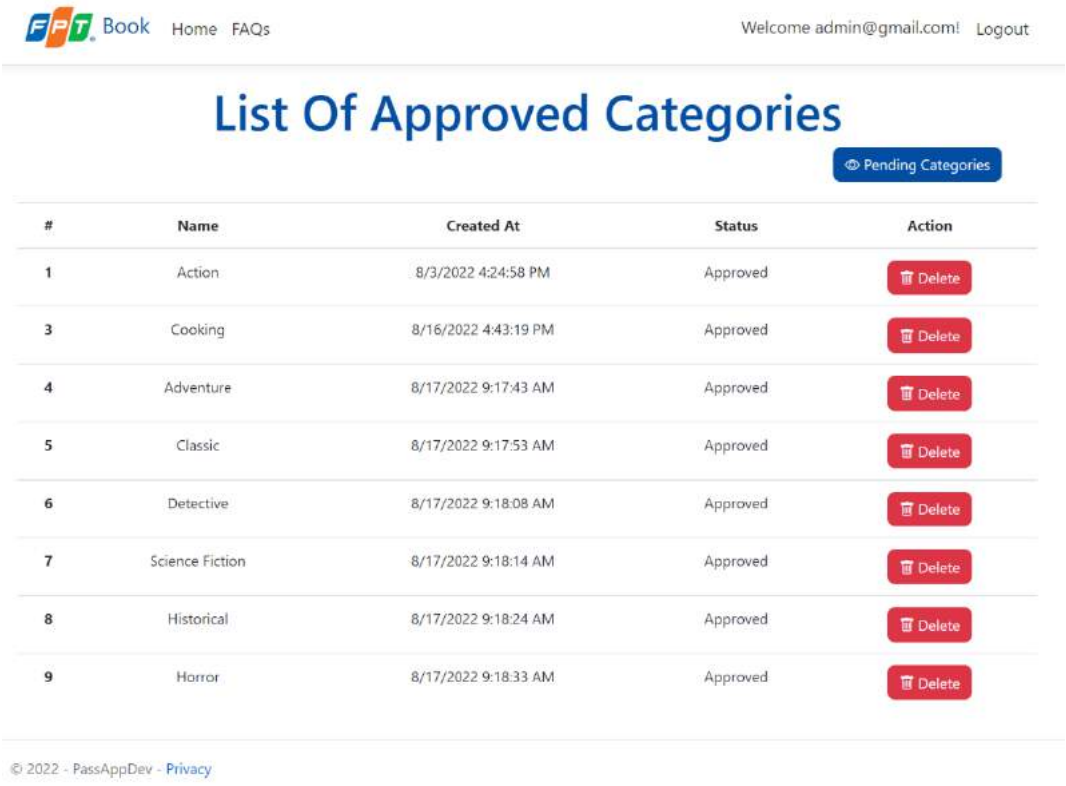


Figure 35: Approved Categories Page

IV. Version Control Of The Project (M5)

During the project's development period, my team and I used GitHub to manage the source code versions of the application in stages. We create a repository called PassAppDev that includes a master branch and other sub-branches. After we have completed a function of the website, we will commit to GitHub to save the source code version for backup purposes.

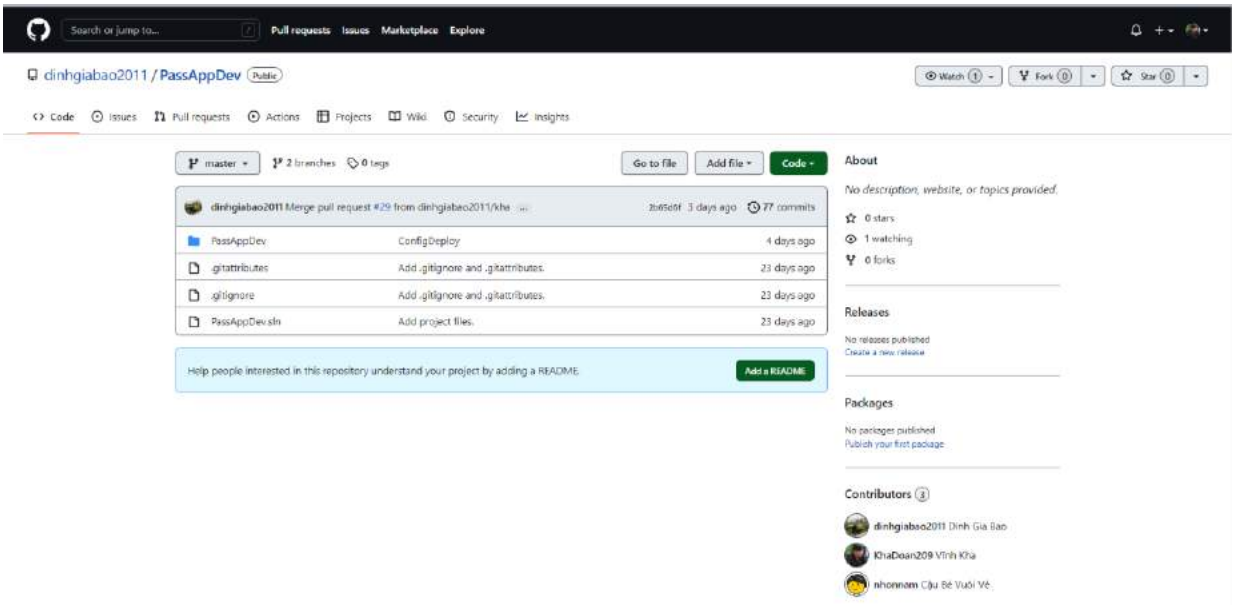


Figure 36: Project On GitHub

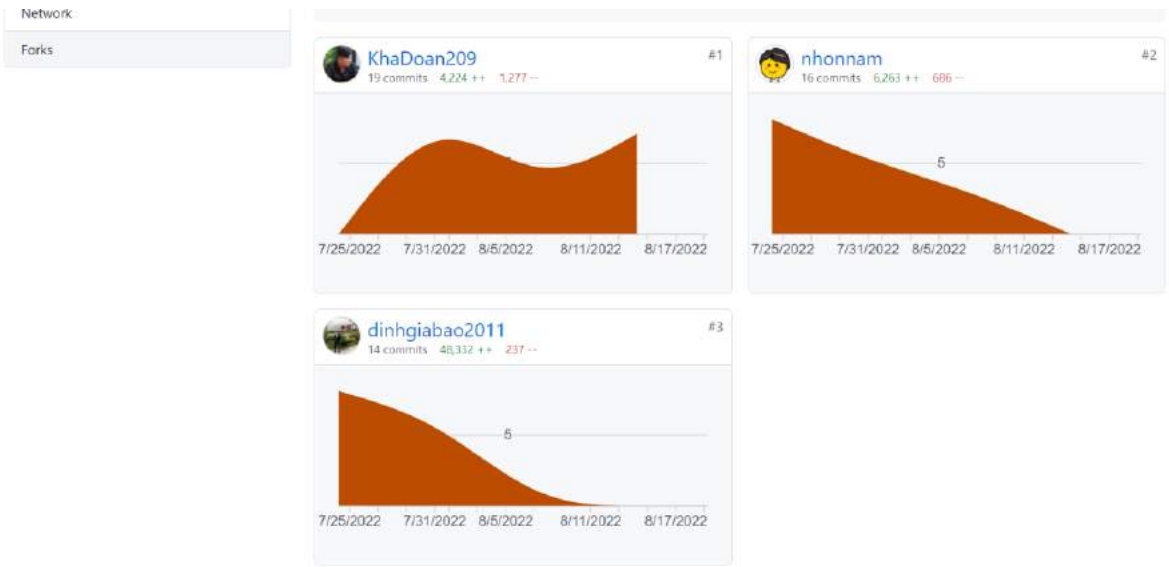


Figure 37: Level of source code contribution of each members

Here is the commits that we have done during the project making

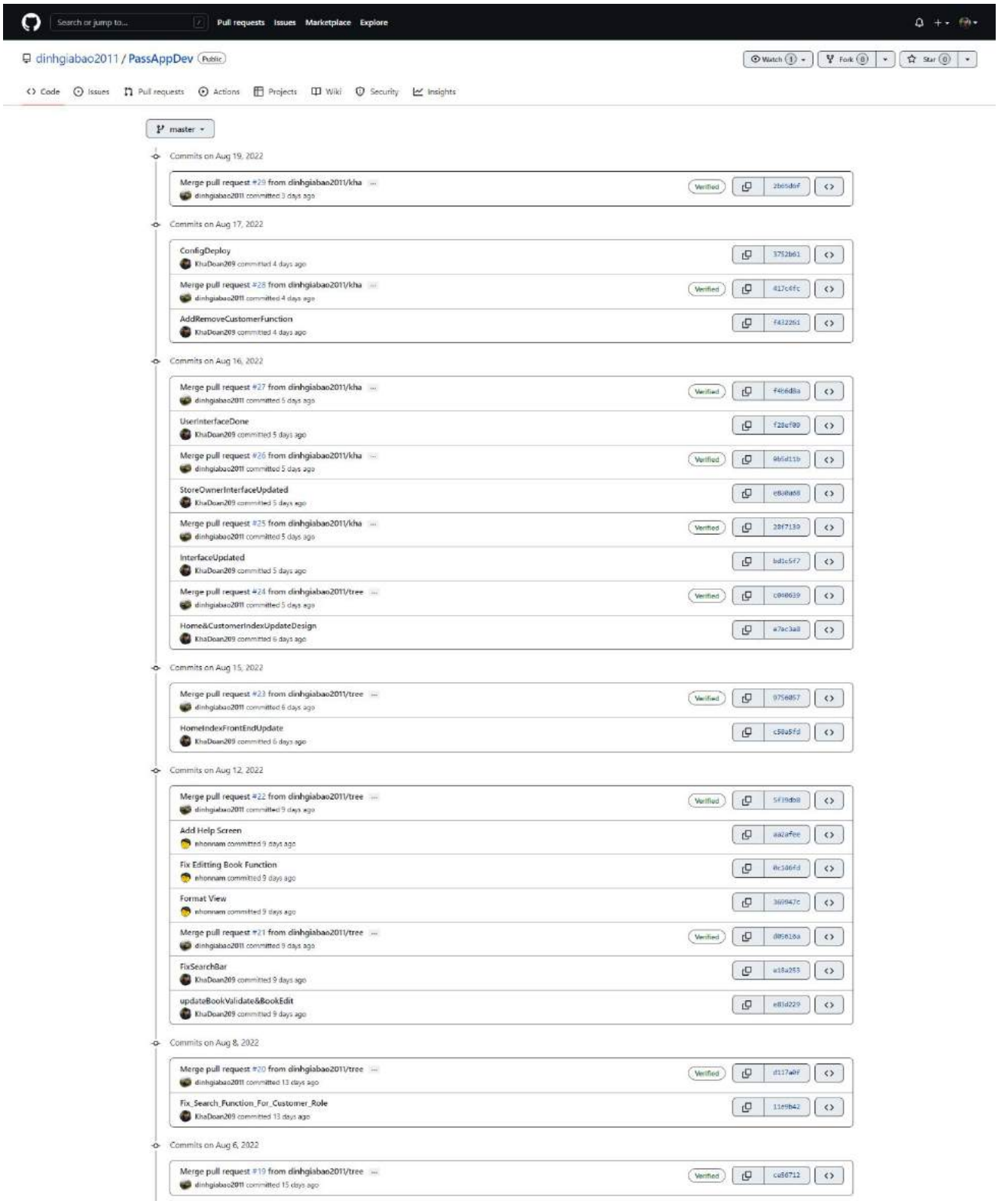


Figure 38: Our teams's commits

V. IIS Deployment (M5)

After completing the project, the last stage we have to take is to deploy the project to a public server so everyone can access and use it. Following are the steps to deploy our project from the development environment to the production environment

First, I have to open the **Internet Information Server** integrated with Windows OS, and start the **Default Web Site**

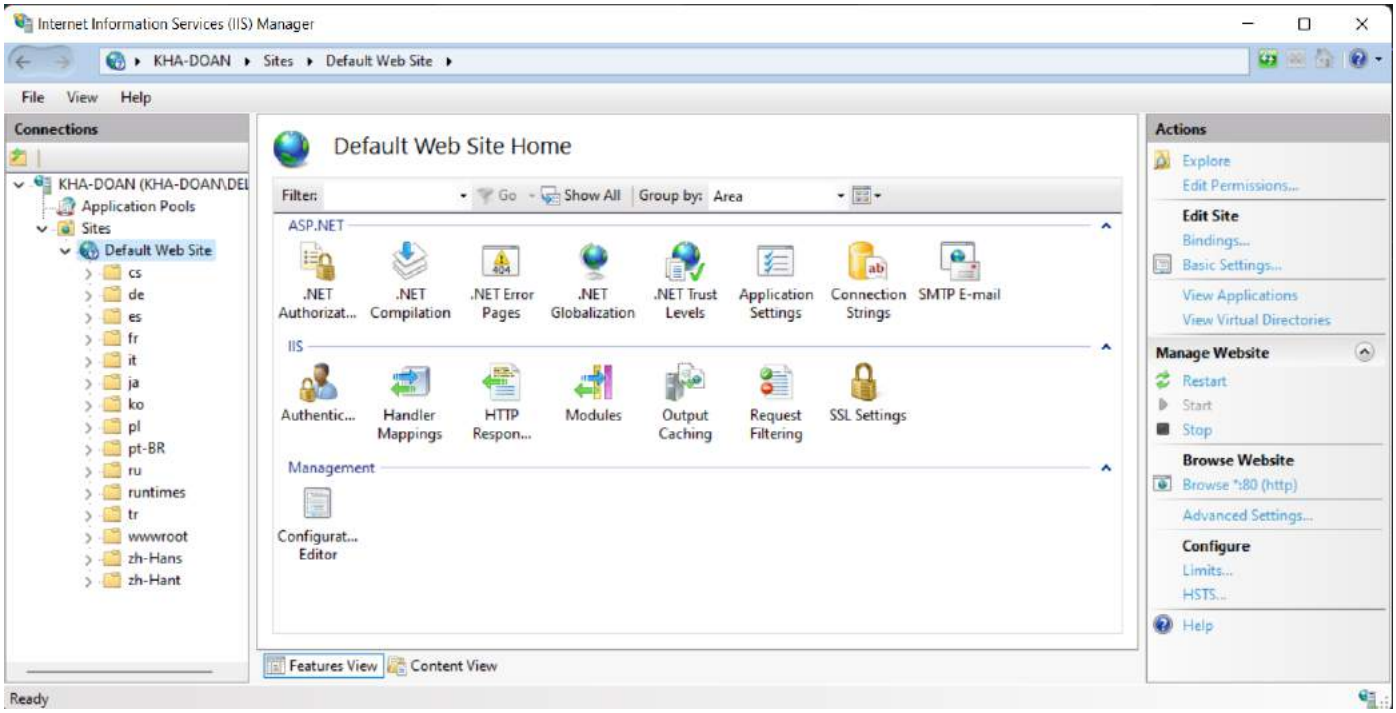


Figure 39: Window IIS

Then, I access to **SSMS (SQL Server Management Studio)** to create a new login account

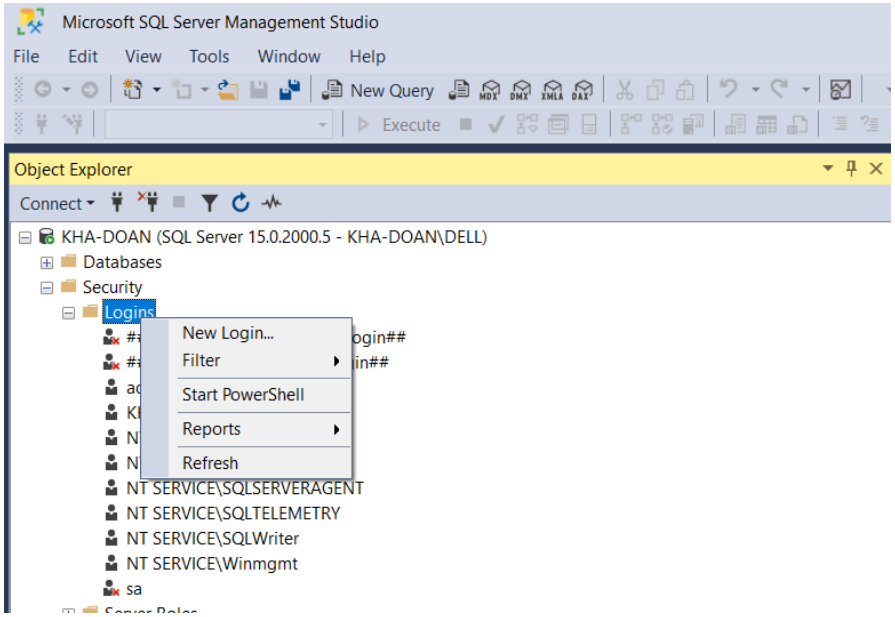


Figure 40: Create a new login

Then choose the login with **SQL Server Authentication**, and start to configure

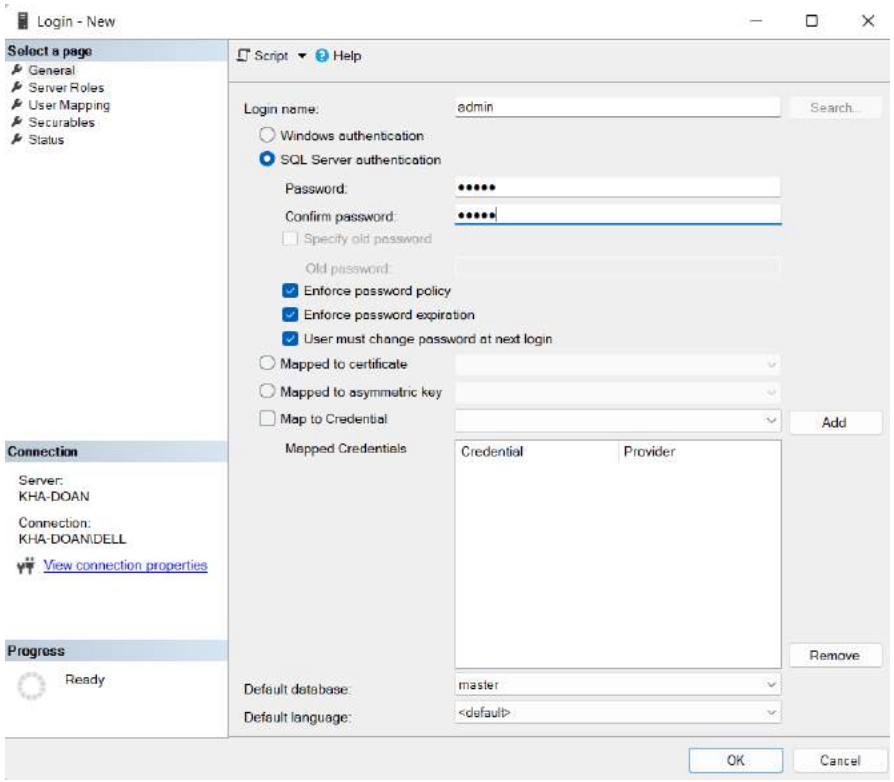


Figure 41: Configure the account

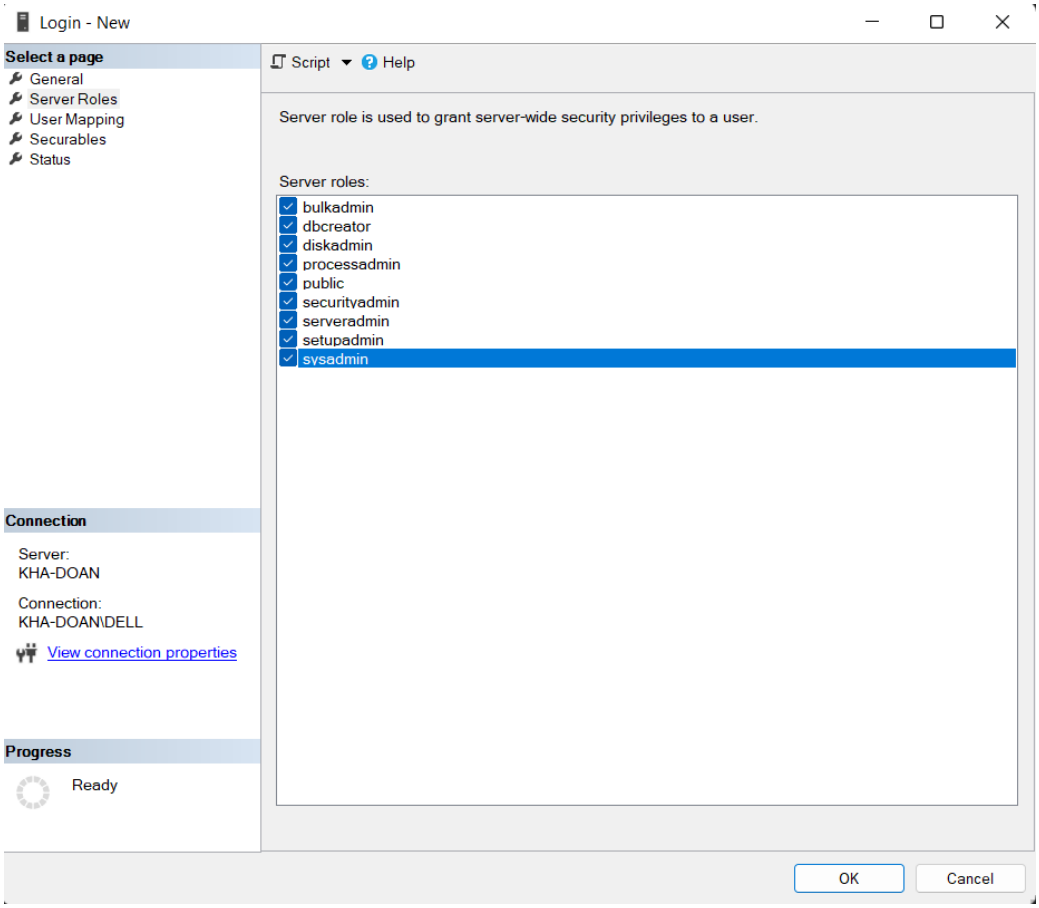


Figure 42: Configure the account (2)

Then, I disconnect the datbase in **SQL Server** and configure the sign in option

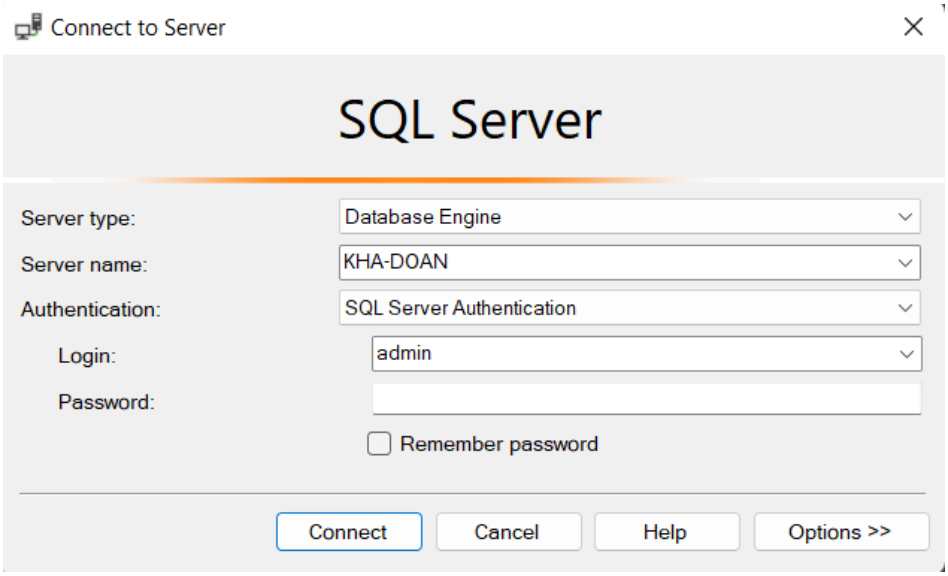


Figure 43: Change sign in option

After that, I open the **SQL Server Configuration Manager** to reset the database so that the new login configuration can be applied

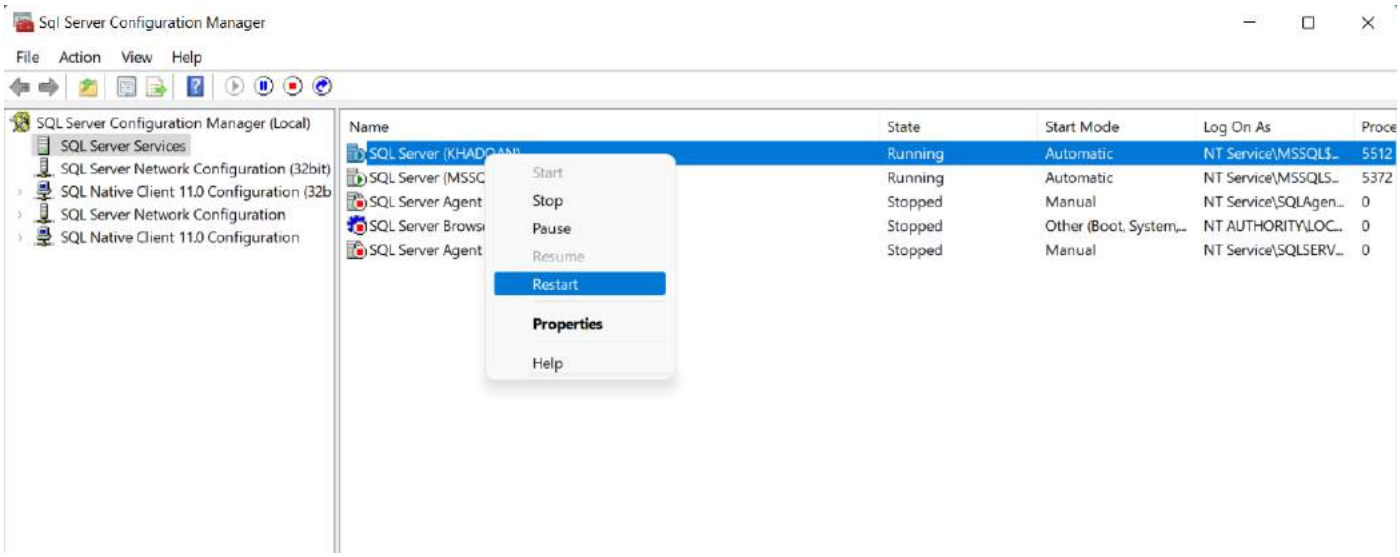


Figure 44: Restart the SQL Configuration

When I done with the SQL configuration, I must configure the web deployment in the Visual Studio. First, I go to the setting file of my project and create a new connection string named "**DeployConnection**". This connection will help my project connect with the database automatically when I deploy to the IIS server

```
{
  "ConnectionStrings": {
    "DefaultConnection": "Server=KHA-DOAN;Database=PassAppDev;Trusted_Connection=True;MultipleActiveResultSets=true",
    "DeployConnection": "Server=KHA-DOAN;Database=PassAppDev;User Id=admin;Password=12345"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "AllowedHosts": "*"
}
```

Figure 45: Create new connection string

Then, I go to the Startup.cs file in my project to change the connection from default to new connection "**DeployConnection**"

```
public IConfiguration Configuration { get; }

// This method gets called by the runtime. Use this method to add services to the container.
public void ConfigureServices(IServiceCollection services)
{
  services.AddDbContext<ApplicationDbContext>(options =>
      options.UseSqlServer(
          Configuration.GetConnectionString("DefaultConnection")));
  services.AddDefaultIdentity<ApplicationUser>(options => options.SignIn.RequireConfirmedAccount = true)
    .AddRoles<IdentityRole>()
    .AddEntityFrameworkStores<ApplicationDbContext>();
  services.AddControllersWithViews();
  services.AddRazorPages();
}
```

Figure 46: Edit the project configuration method

After completed the above steps, I start to deploy my project. Fisrt, I choose the option Public and a deployment window will be displayed. To configure, I choose the **Web Server** (**IIS**)

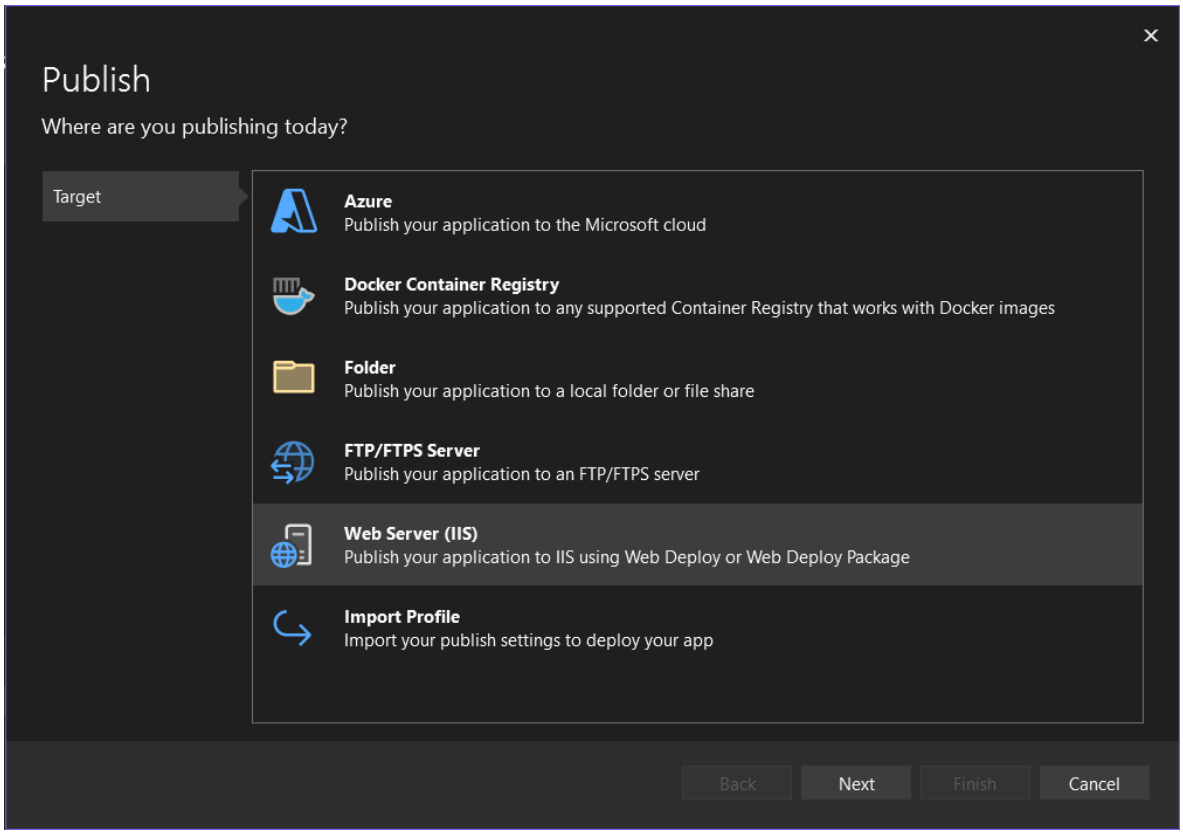


Figure 47: Public window

Then, I choose to publish our project to IIS using Web Deploy.

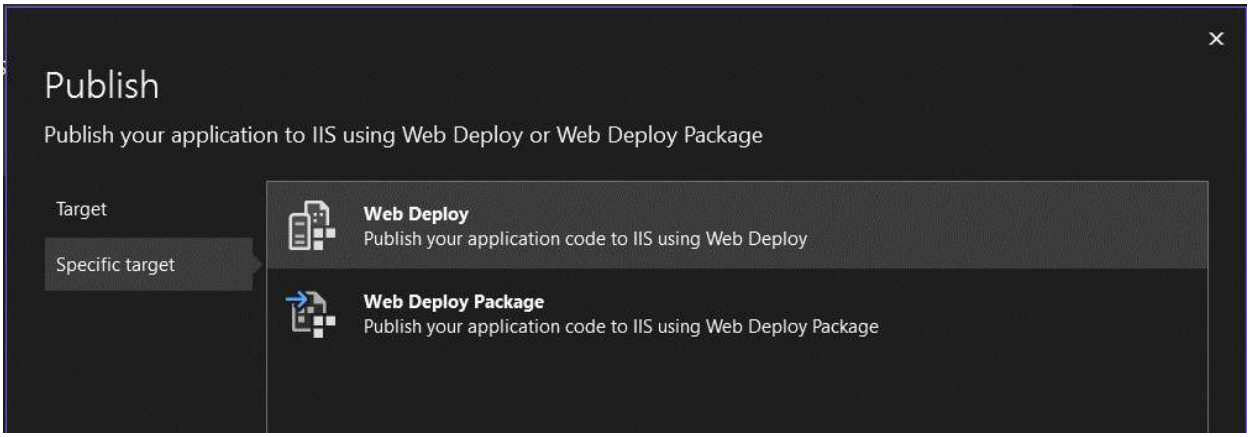


Figure 48: Choosing Web Deploy

Next, I have to configure the deployment using the **Default Web Site** that already turned on

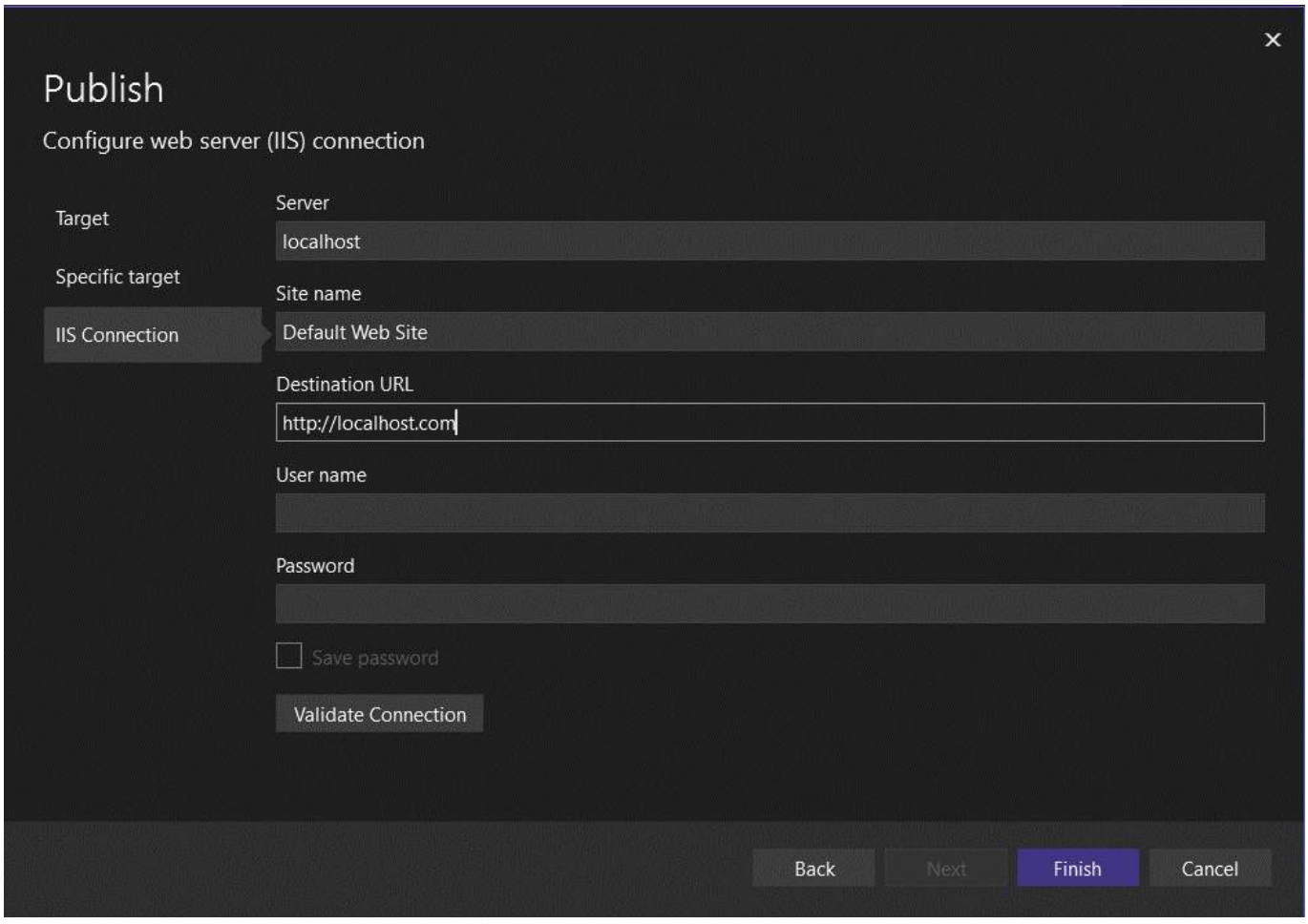


Figure 49: Configure the deploy connection

The public configuration validated

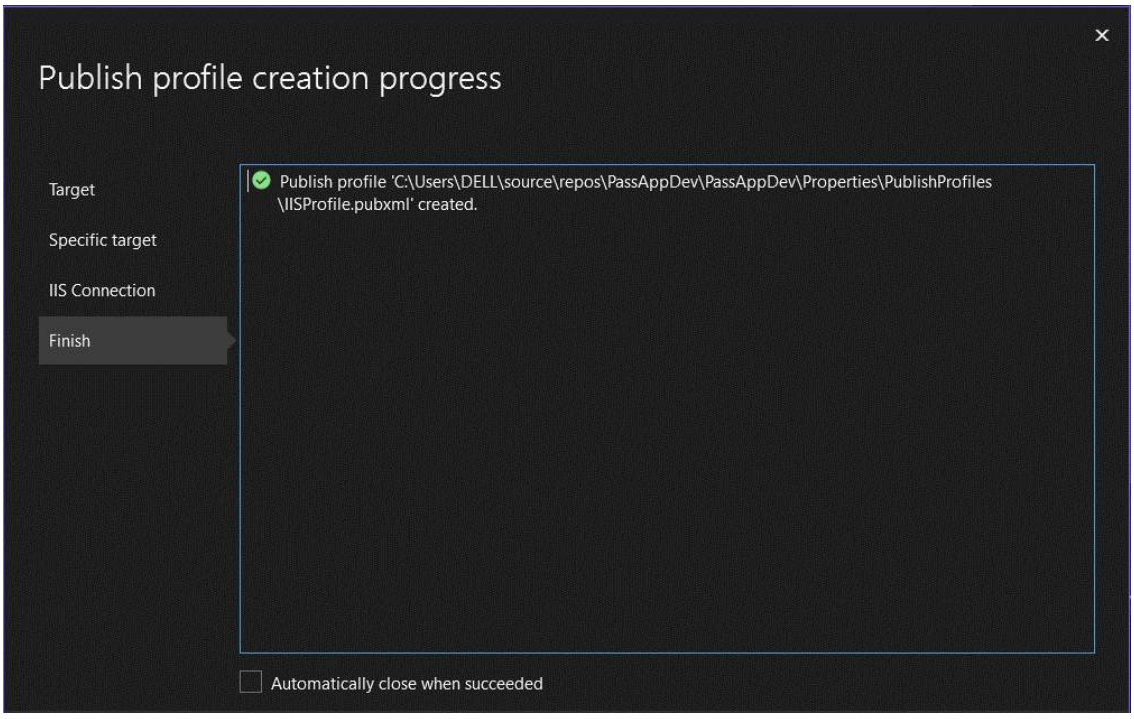


Figure 50: Finishing configuring the publish profile

After the configuration process done, the below interface will be displayed.

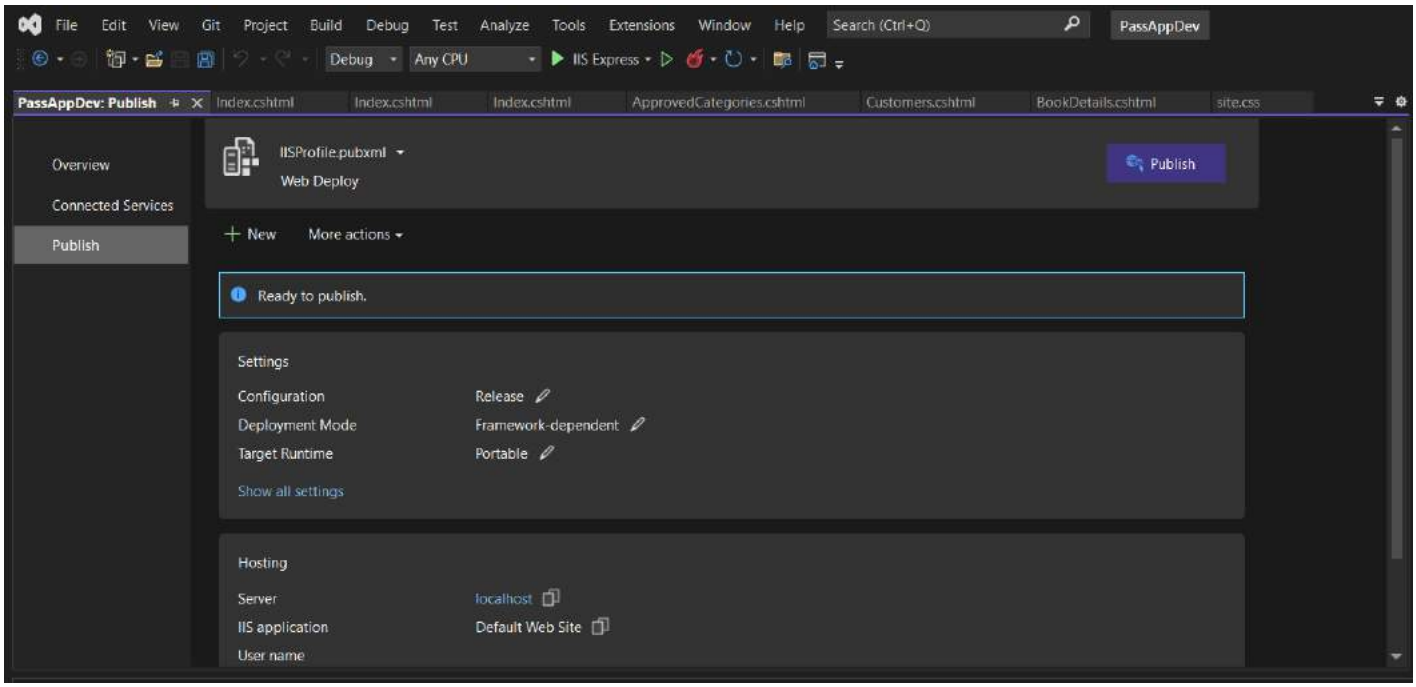


Figure 51: Ready to publish

After the successful deployment, instead of using the IIS Server in Visual Studio to open the project, I can access the application by entering localhost in the browser.

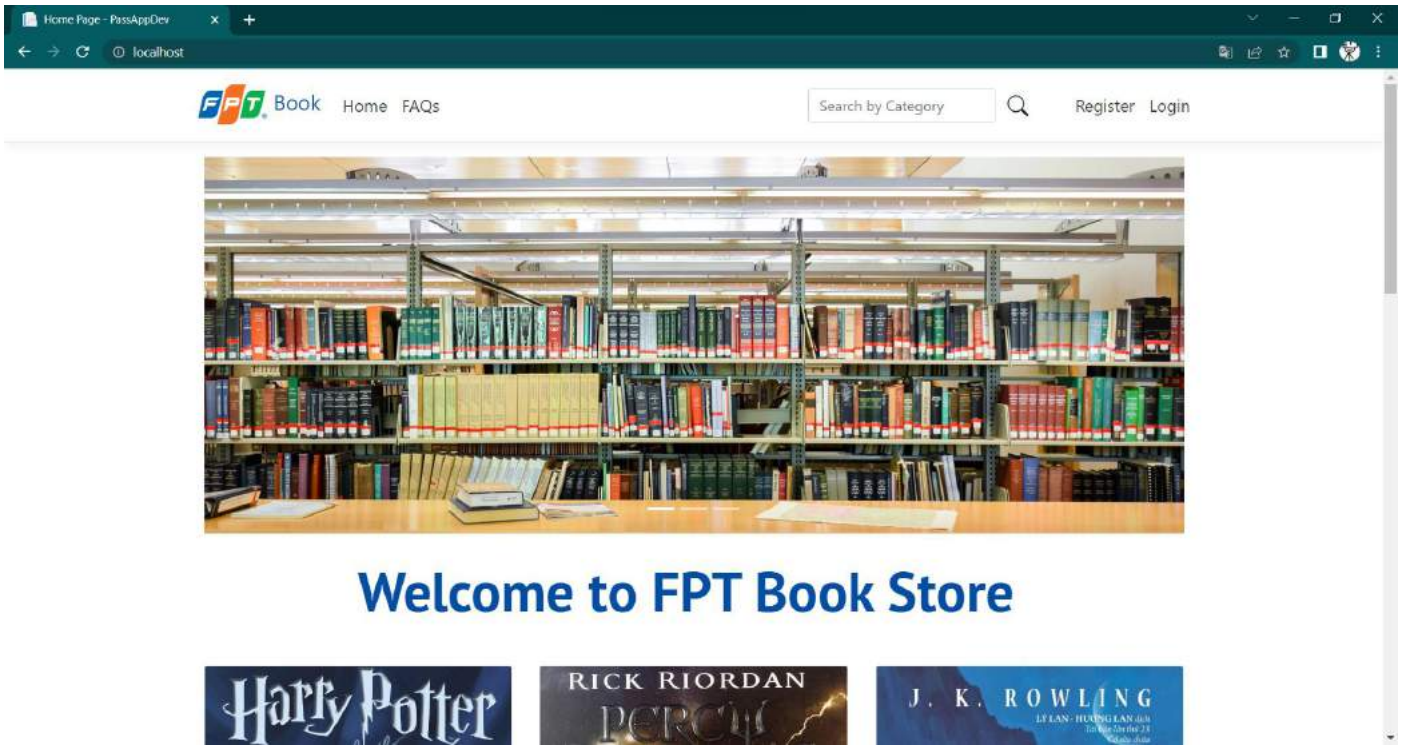


Figure 52: Access to the website after being deployed

Moreover, by using my laptop' IP address, I can access to the website from my phone. For example, my laptop IP is 192.168.120.212, after entering this ip into my phone's browser, this is the result

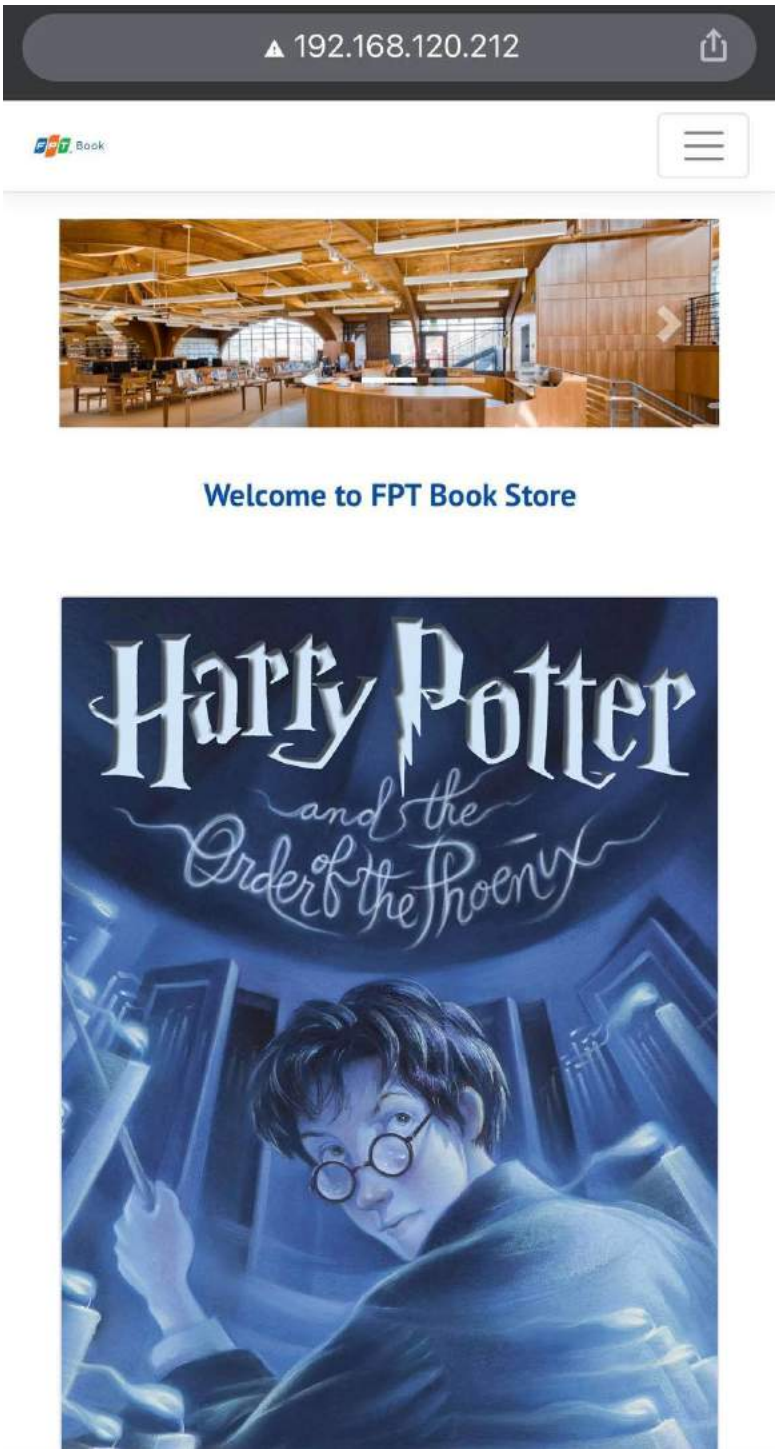


Figure 53: Access to the website from mobile device

Chapter 3 - Application Evaluation (P6 – M5 – D3)

I. Performance Of The Application (P6)

1. The Functionality Of The Website

Basically, here is the list of our website's functionalities

With customers role:

When the user is not logged in or has logged in as the customer, all of the book products that are available for purchase will be displayed to the user.

- The **Book Detail** page will pop up when you click on any book's title. This screen will help provide detailed information on a particular book to help users to ensure that this is the book they want.
- The customer's account information can be found and updated on the **Profile Page**, which also displays the customer's information.
- The **Register Page** allows users to purchase books before becoming a customer. On this page, users are prompted to provide various information, including their email address, full name, password, re-password, and home address.
- The **Cart Page** displays the items that the consumer has currently added to the cart, as well as the price, the quantity, and the subtotal and total amounts for the items in the shopping cart. Customers can place their orders and make payments via the "Confirm Order" button below.
- Once the order has been processed, the customer will be taken to the **Order Page**, where they can examine details about any past orders they have placed. Customers can view each item they have added to the order on the List of ordered items page
- The **FAQs Page** answers most of customers' concerns about orders, shipping, or return policies.
- A search bar on the top of every screen helps customers to search for a specific book by book category or title.

With store owners role:

When the user has logged in as the store owner, they can access to the following page:

- A **Book Management Page** allows the store owner to view the information and specifics of the books while also allowing them to add, update, and delete the information of the books in the store
- A **Category Management Page** that allows the store owner to send to the administrators the requests about creating a new category or editing an existed category name
- An **Ordered Recorded Page** allows the store owner to view the orders that all customers have placed in addition to the item-specifics of those orders, such as the item name, quantity, price, and total amount of the order.
- A **notification** icon on the Home Screen for store owners to click will provide the status of their request to add or amend categories.

With administrators role:

- A **Customer Page** that allows the administrators to view the account details of all customers and change their passwords if necessary
- A **Store Owner Page** gives admin access to view the account details of all store owners and change their passwords if necessary
- A **Category Request Page** allows the administrators to view the categories waiting to be approved or rejected by store owners.
- A **Category Page** allows the administrator to view the list of approved categories in the store and enable them to remove these categories if necessary.

2. Test Case

The list above is the list of functional requirements, now I will write a test case to see whether all functions of my application work well.

No.	Test Purposes	Description	Expected Result	Actual Result	Status
1	Test the operation of the search bar	I will access to the website via browser, and type name of any book and category in the store	The website will display the book that I want to find	**Figure 54**	Pass
2	Test the operation of register function	I will register a new account named kdoan@gmail.com	The account has been updated in the database	**Figure 55** **Figure 56**	Pass
3	Test the operation of login function	I will sign in with the a new account kdoan@gmail.com	The home page will change to the home page for customer	**Figure 57**	Pass
4	Test the operation of adding book to cart and editing book quantity	I will add the new book to my cart list and then change the book number to see whether the subtotal amount change	The book added to cart with the quantity is 1, when the quantity change, the subtotal amount is also change	**Figure 58** **Figure 59**	Pass
5	Test the operation of view order list and order detail	I will confirm the order in the shopping cart, after that, I can view my order list, and the detail of each orders	My ord has been processed to list of order page, and can see the order detail	**Figure 60** **Figure 61**	Pass
6	Test the operation of book management function	I will sign in with the store owner account, then I access the book management page to create a new book, update book's price, view detail, and remove the book from the store	The new book is created, the book price is updated, the book detail is correct, and the book is successfully removed from the list	**Figure 62** **Figure 63** **Figure 64** **Figure 65** **Figure 66**	Pass
7	Test the operation of sending category request	I will create a new category named Finance and send the reques to the administrators. Then I will sign in as the administrator to view whether I can receive the request.	The request from store owner is displayed to the administrator	**Figure 67** **Figure 68**	Pass

8	Test the operation of approving category request	After receiving the store owner's request, I will approve the request and view the list of approved categories to check. Then, I sign in with the store owner's account to consider whether the notification about tapprovalent is sent to me.	The approved request appeared in the list, and the store owner receive the notification	**Figure 69** **Figure 70**	Pass
9	Test the operation of changing customer's password	I will sign in with the administrator account and change the password of the account kdoan@gmail.com to 1	The password is changed successfully and customer can sign in with the new password	**Figure 71** **Figure 72** **Figure 73**	Pass
10	Test the operation of approving category request	I will sign in with the administrator account and change the password of the account store@gmail.com to 1	The password is changed successfully and storeowner can sign in with the new password	**Figure 74** **Figure 75** **Figure 76**	Pass

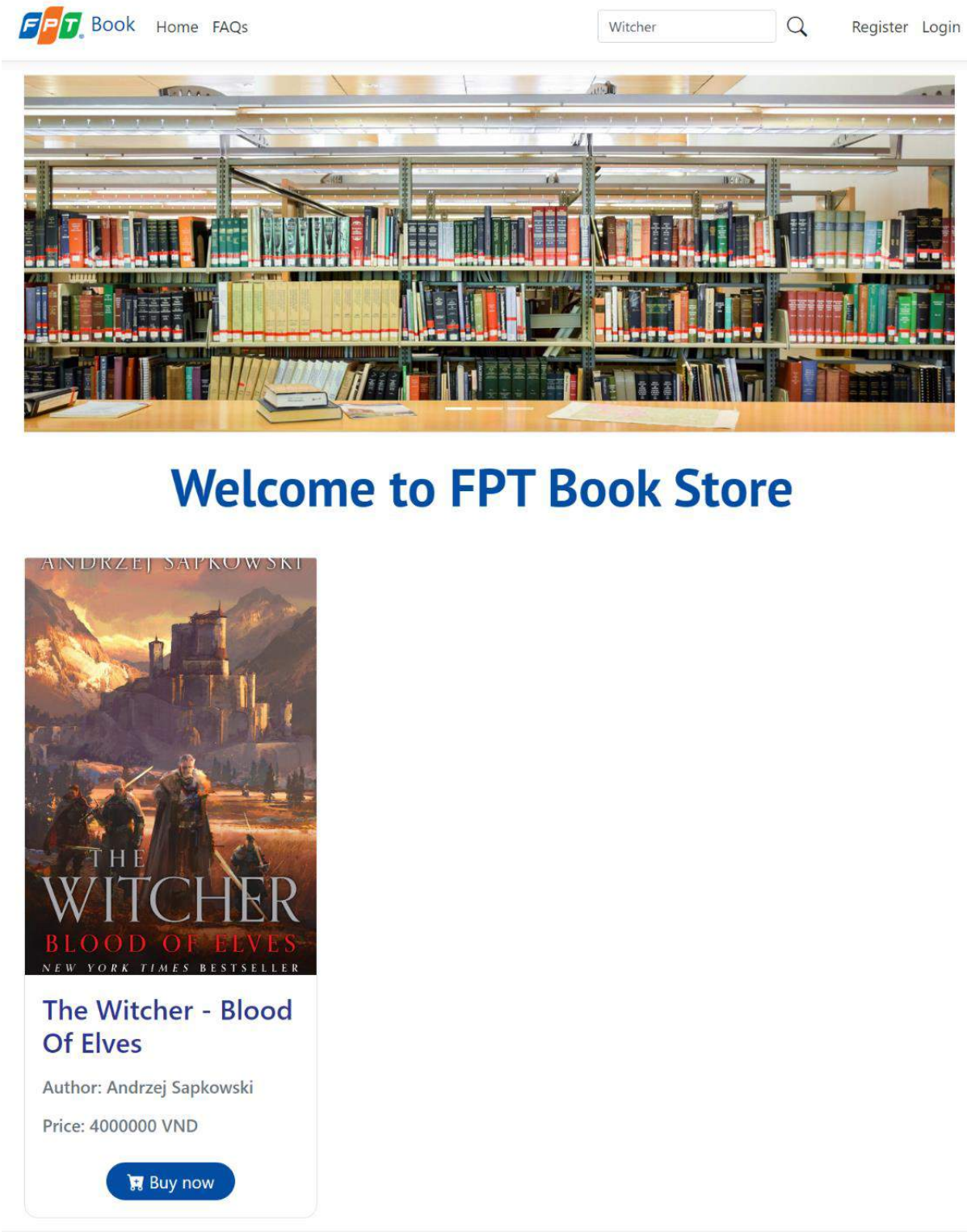


Figure 54: Test case result (1)

Register

Create a new account.

Email

kdoan@gmail.com

Password

••••••••

Confirm password

••••••••

The password and confirmation password do not match.

FullName

Kha Doan

Address

6501 South 19th Street

Cancel

Register

Use another service to register.

There are no external authentication services configured. See this article for details on setting up this ASP.NET application to support logging in via external services.

Figure 55: Test case result (2)

	Id	UserName	NormalizedUserName	Email	NormalizedEmail	EmailConfirmed
1	5e290c6f-0501-4f2d-aa6f-9fc2319b784b	nam@gmail.com	NAM@GMAIL.COM	nam@gmail.com	NAM@GMAIL.COM	1
2	a0554bfd-1d4d-4a61-97d4-d827530e6883	store@gmail.com	STORE@GMAIL.COM	store@gmail.com	STORE@GMAIL.COM	1
3	b74ddd14-6340-4840-95c2-db12554843e5	admin@gmail.com	ADMIN@GMAIL.COM	admin@gmail.com	ADMIN@GMAIL.COM	1
4	bdbe8e6e-ccb2-4987-88ed-52ee946b01ce	kha@user.com	KHA@USER.COM	kha@user.com	KHA@USER.COM	1
5	cdb162cf-5ea7-40e3-8a8d-ce8813a09c15	kha@gmail.com	KHA@GMAIL.COM	kha@gmail.com	KHA@GMAIL.COM	1
6	d58310cd-2595-4723-9cc4-0cabd3483396	kdoan@gmail.com	KDOAN@GMAIL.COM	kdoan@gmail.com	KDOAN@GMAIL.COM	1

Figure 56: Test case result (2)

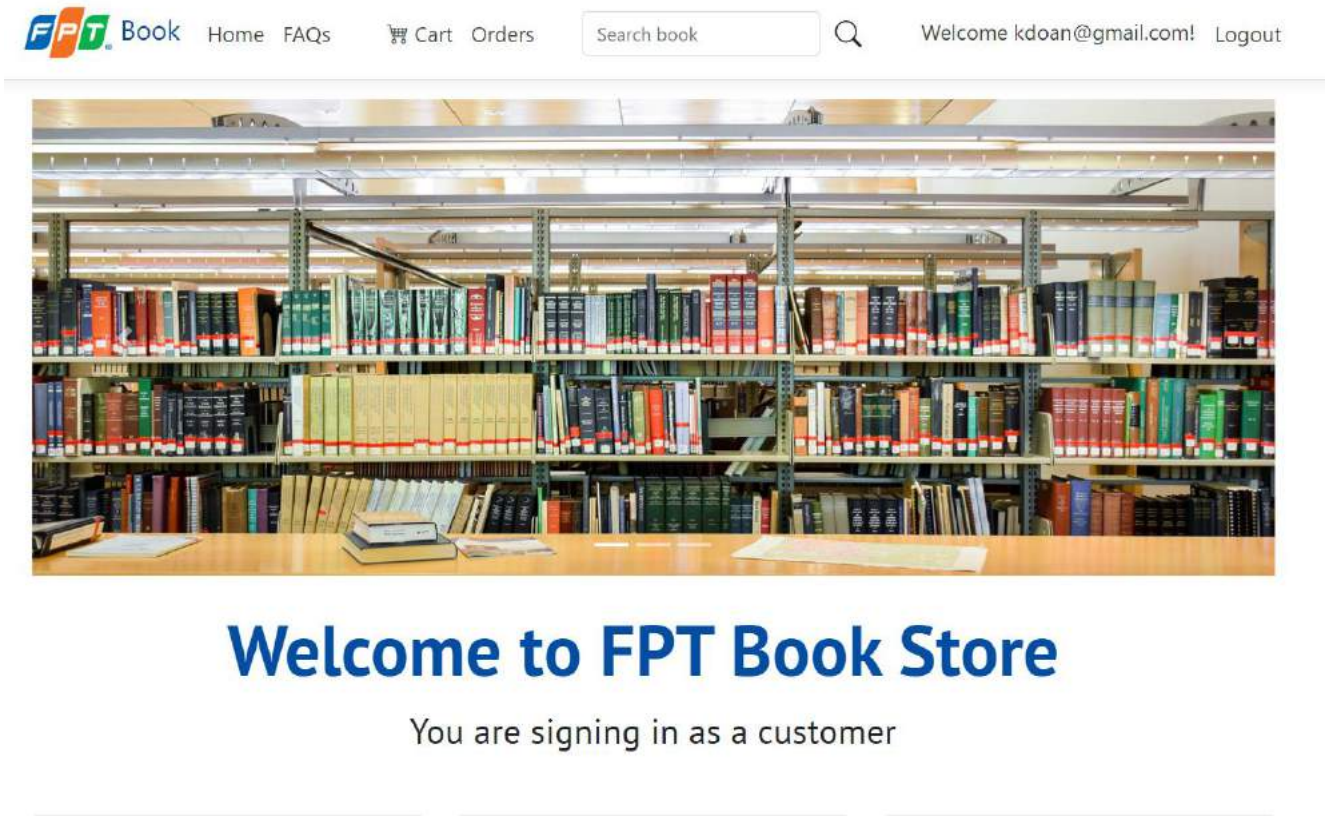


Figure 57: Test case result (3)

Figure 58: Test case result (4)

Figure 59:Test case result (4)

Figure 60: Test case result (5)

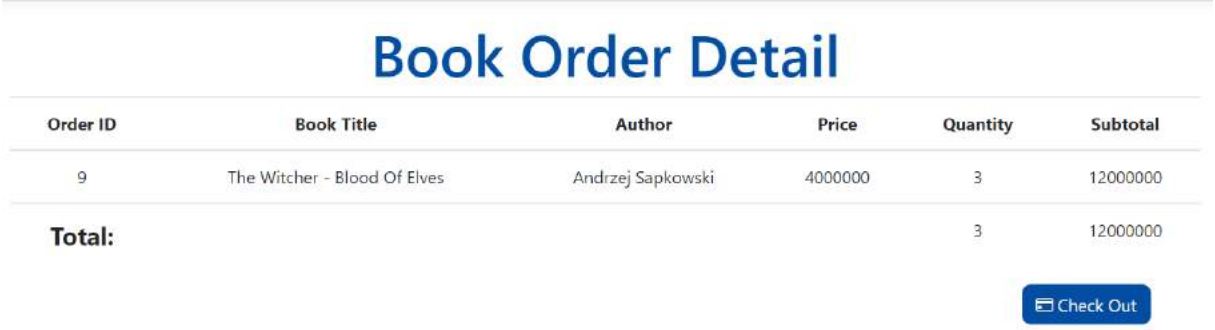


Figure 61: Test case result (5)

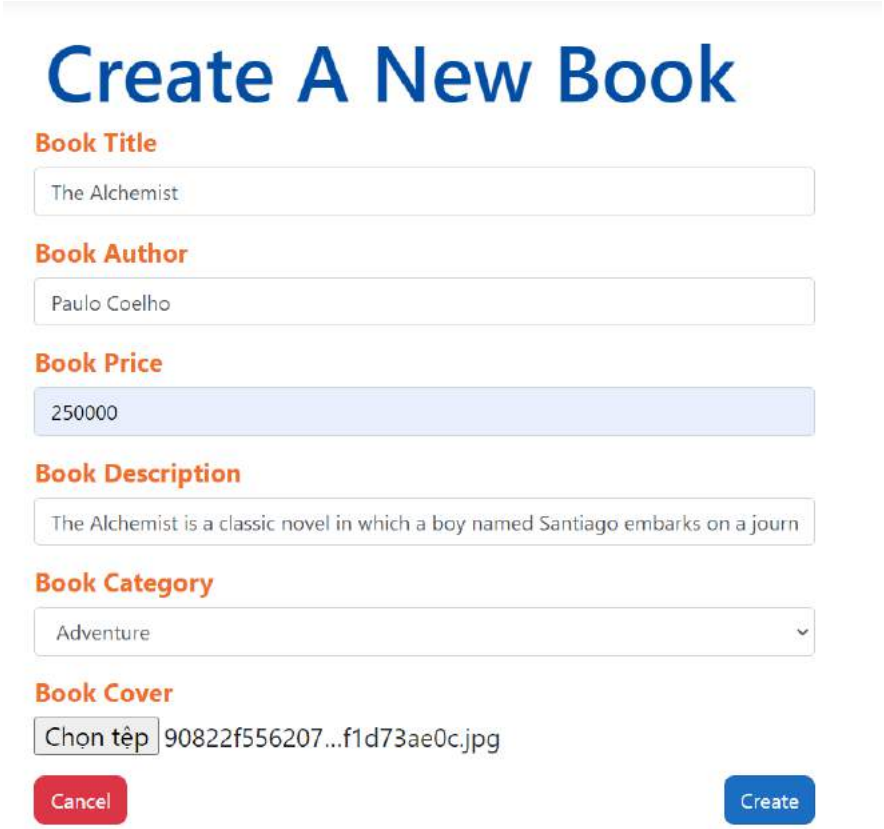


Figure 62: Test case result (6)

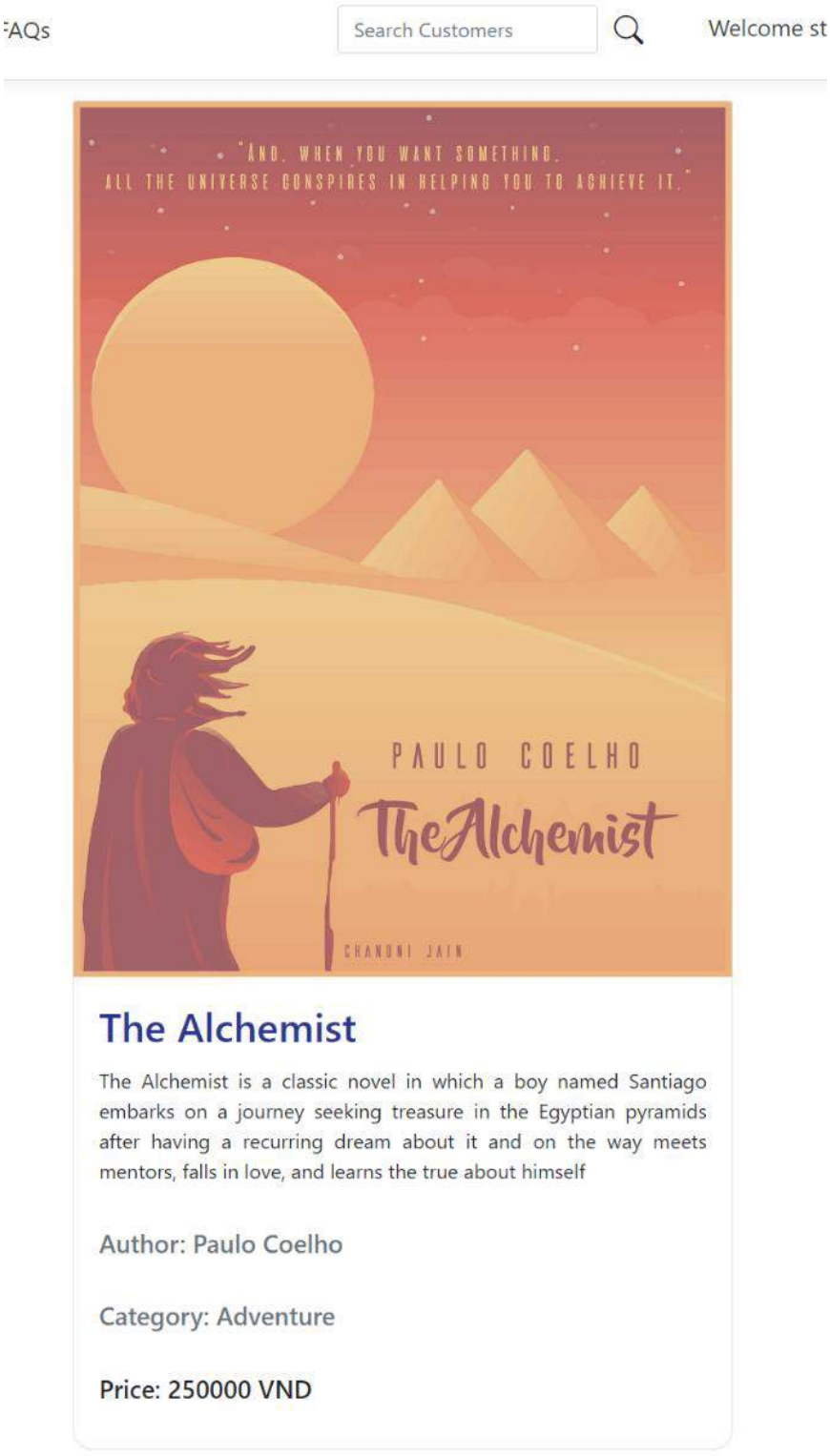
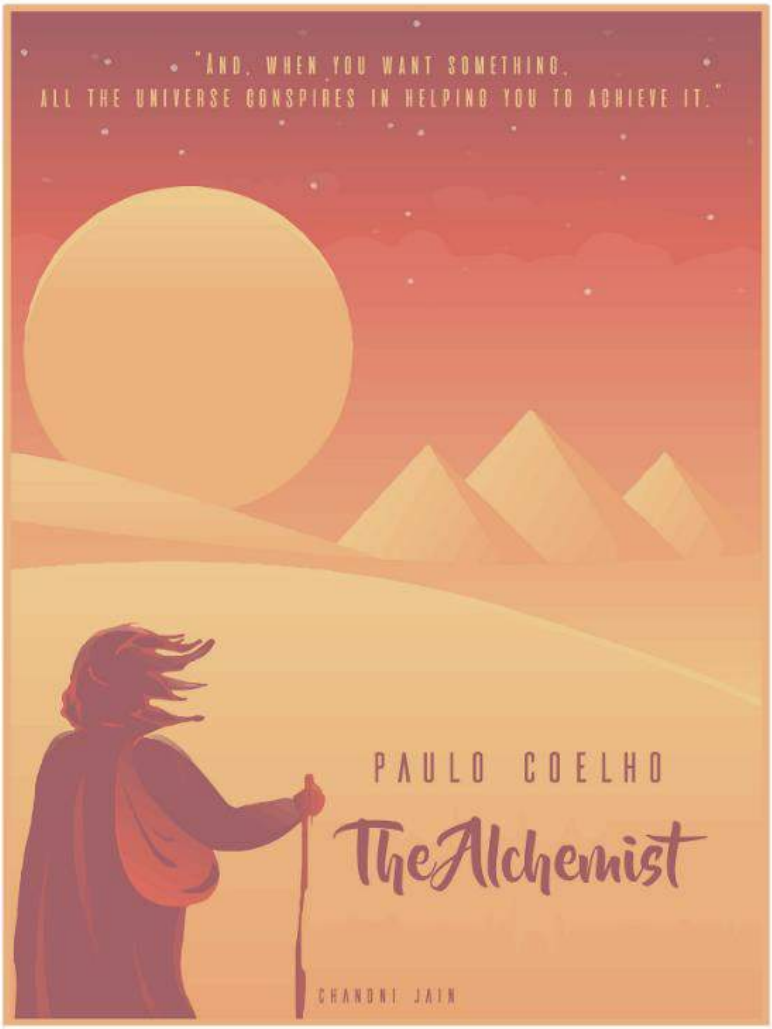


Figure 63: Test case result (6)



Nhà giả kim

The Alchemist is a classic novel in which a boy named Santiago embarks on a journey seeking treasure in the Egyptian pyramids after having a recurring dream about it and on the way meets mentors, falls in love, and learns the true about himself

Author: Paulo Coelho

Category: Adventure

Price: 400000 VND

Figure 64: Test case result (6)

The Witcher - Blood Of Elves	4000000	Adventure	View Edit Delete
The Complete Sherlock Holmes	250000	Detective	View Edit Delete
The Detective Conan - Vol. 35	12000	Detective	View Edit Delete
Nhà giả kim	400000	Adventure	View Edit Delete

Figure 65: Test case result (6)

The Witcher - Blood Of Elves	4000000	Adventure	View Edit Delete
The Complete Sherlock Holmes	250000	Detective	View Edit Delete
The Detective Conan - Vol. 35	12000	Detective	View Edit Delete

Figure 66: Test case result (6)

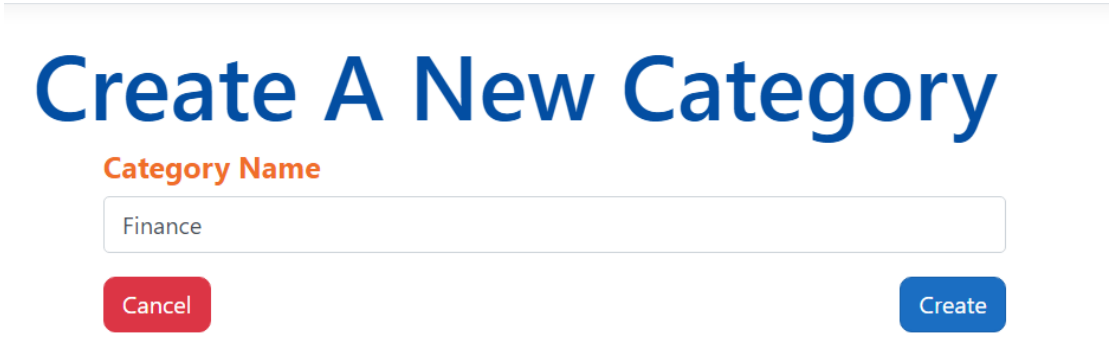


Figure 67: Test case result (7)

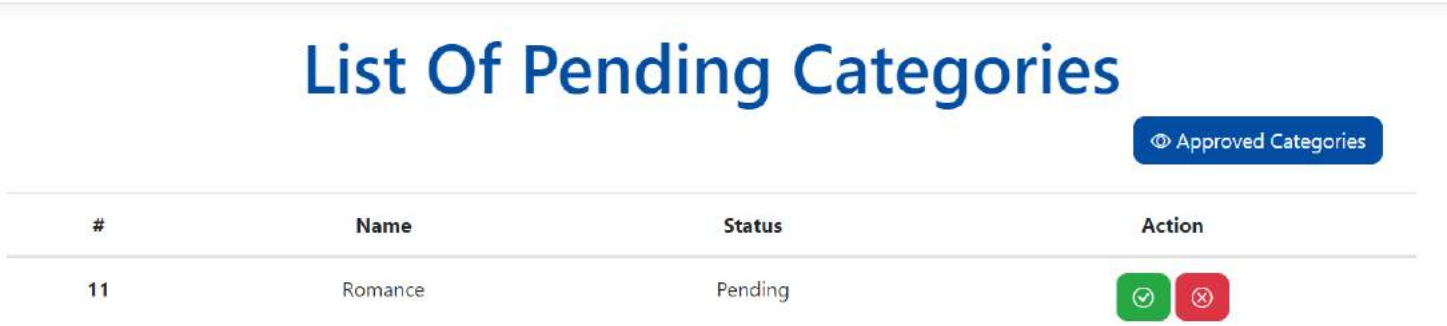


Figure 68: Test case result (7)

List Of Approved Categories

Pending Categories

#	Name	Created At	Status	Action
1	Action	8/3/2022 4:24:58 PM	Approved	Delete
3	Cooking	8/16/2022 4:43:19 PM	Approved	Delete
4	Adventure	8/17/2022 9:17:43 AM	Approved	Delete
5	Classic	8/17/2022 9:17:53 AM	Approved	Delete
6	Detective	8/17/2022 9:18:08 AM	Approved	Delete
7	Science Fiction	8/17/2022 9:18:14 AM	Approved	Delete
8	Historical	8/17/2022 9:18:24 AM	Approved	Delete
9	Horror	8/17/2022 9:18:33 AM	Approved	Delete
11	Romance	8/21/2022 1:40:57 PM	Approved	Delete

Figure 69: Test case result (8)

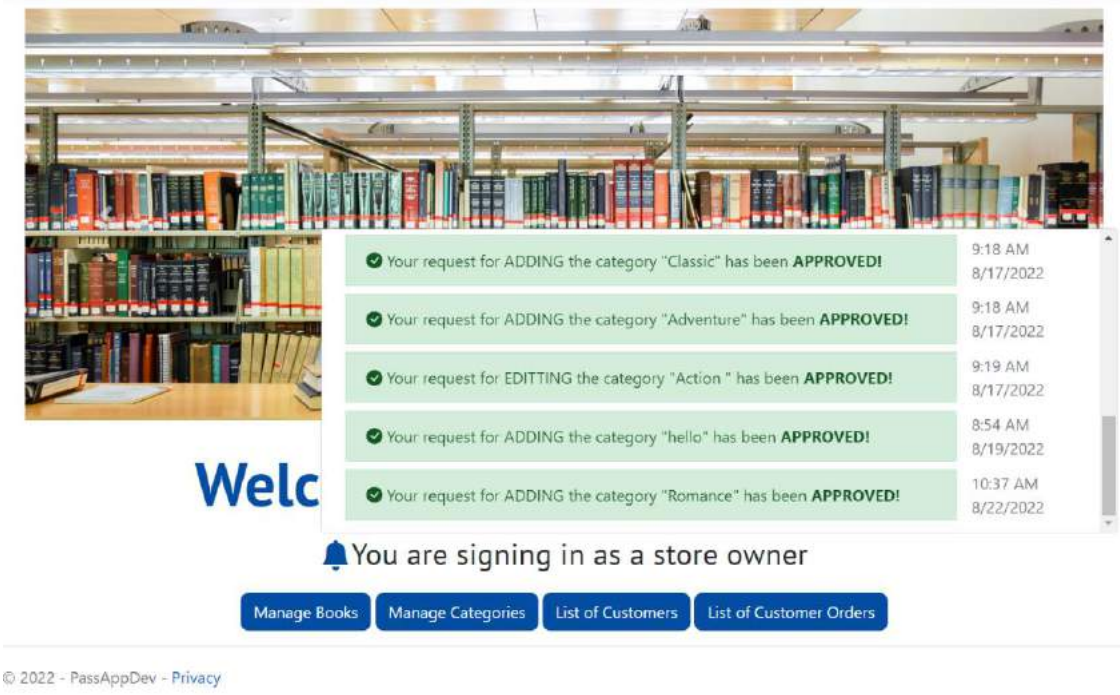


Figure 70: Test case result (8)

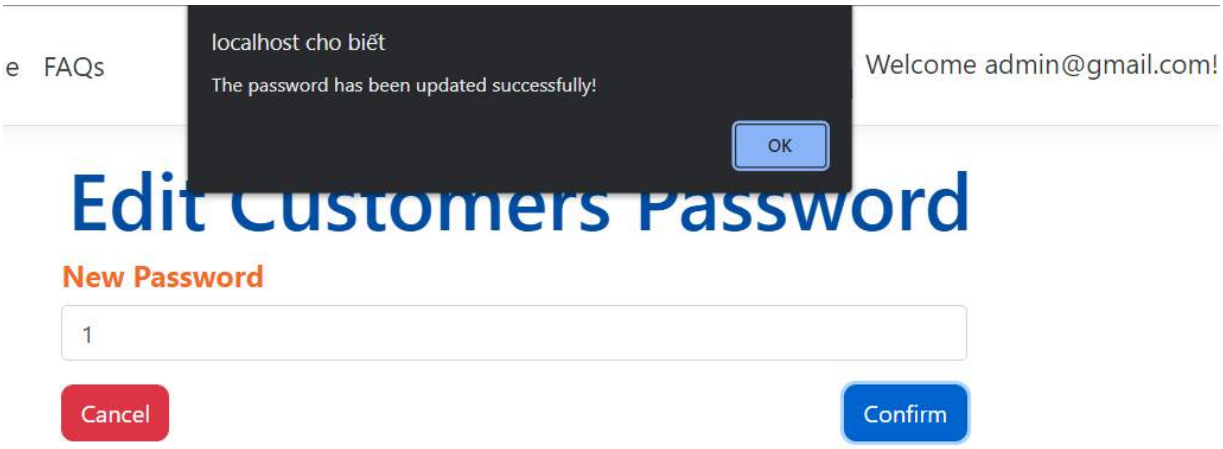


Figure 71: Test case result (9)

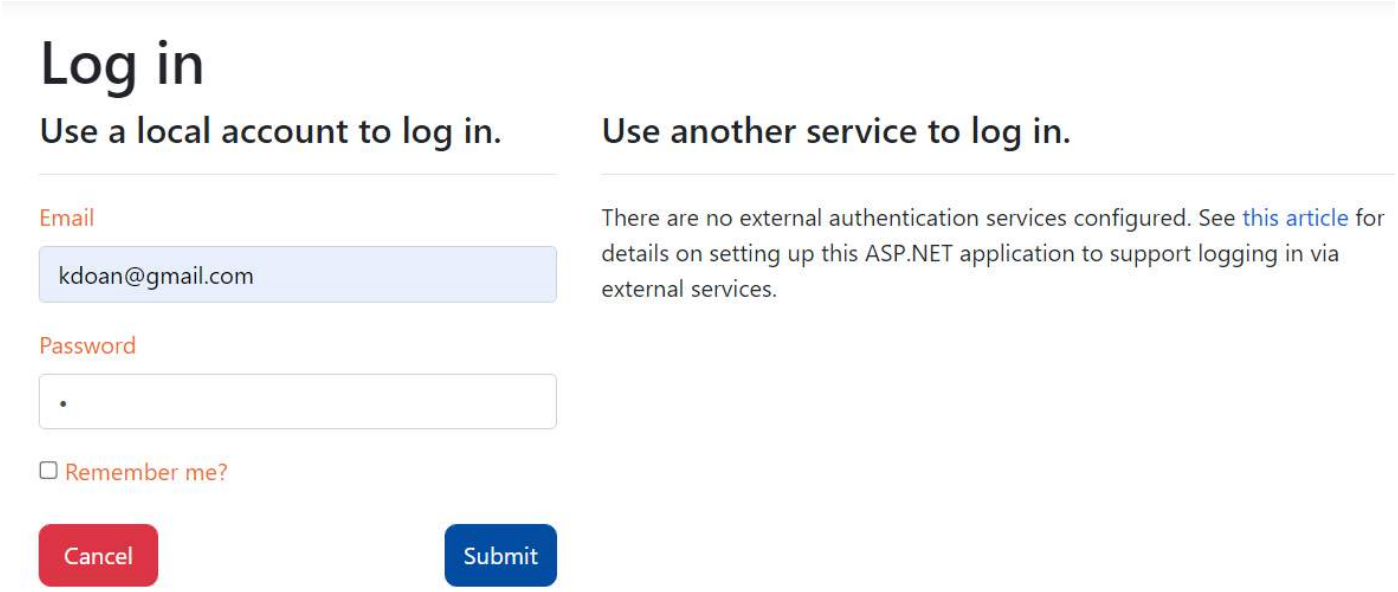


Figure 72: Test case result (9)

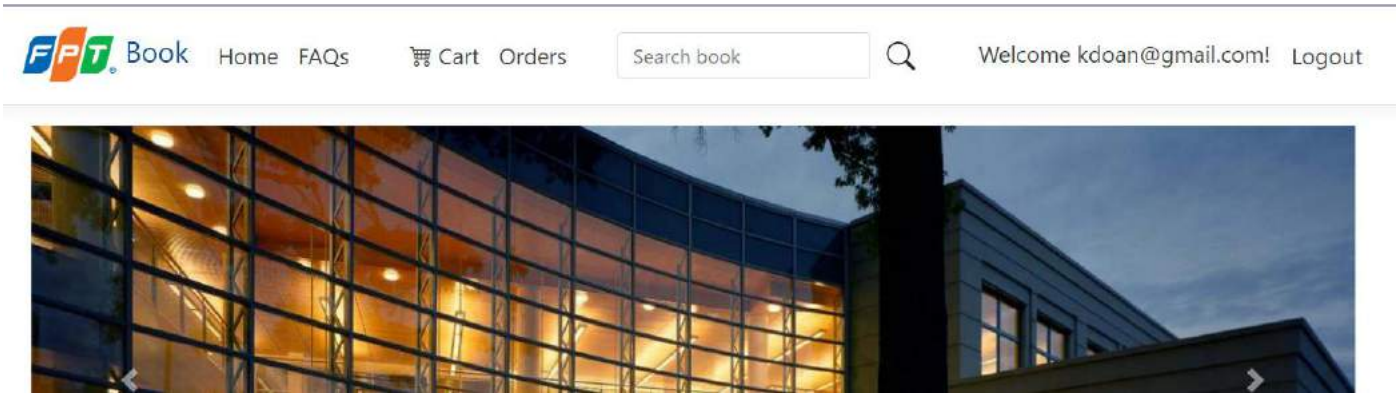


Figure 73: Test case result (9)

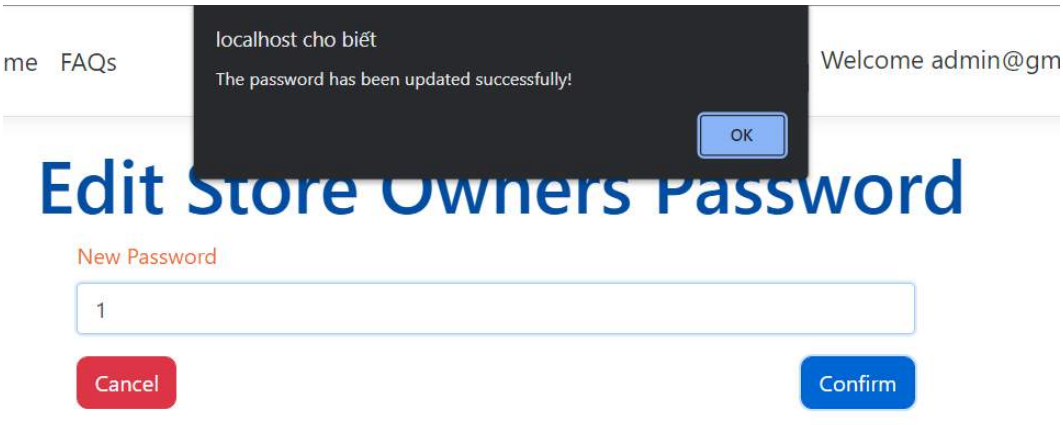


Figure 74: Test case result (10)

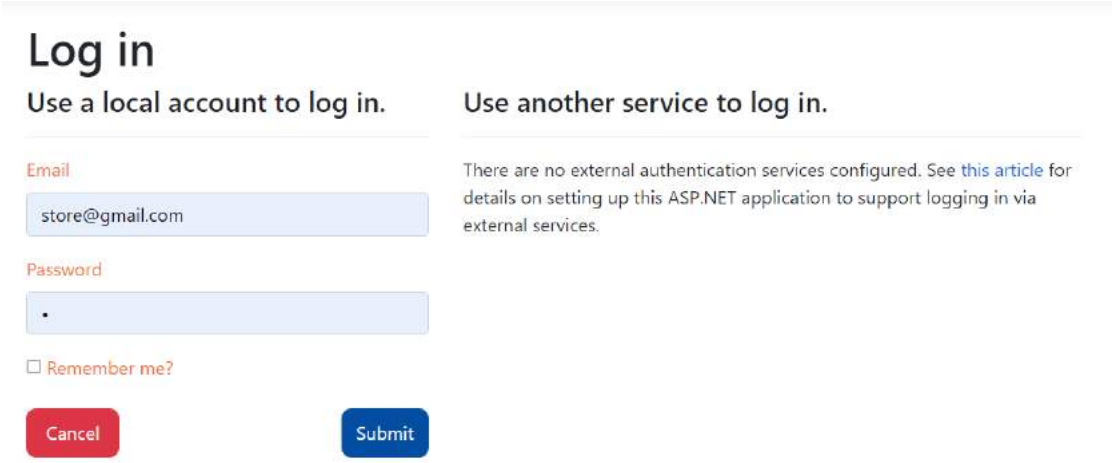


Figure 75: Test case result (10)

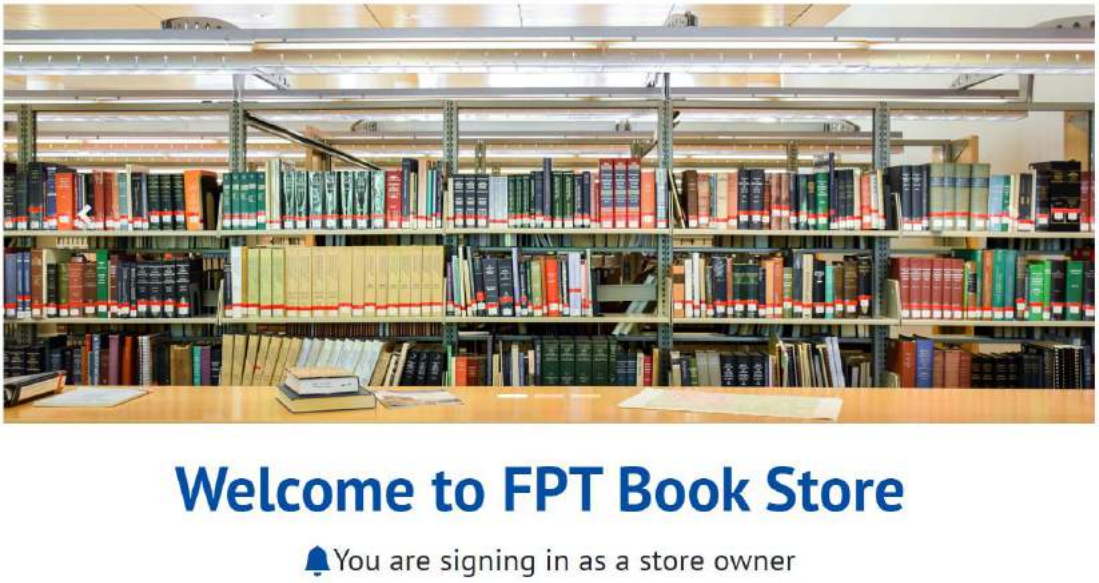


Figure 76: Test case result (10)

3. The Performance Of The Website

For performance evaluation, I run my software in Debug mode to see how well it performs using the analysis tool integrated with Visual Studio. The Diagnostic Tools section can be found in the very rightmost bar of the interface. Here, we can view information such as the amount of process memory consumption and the percentage of CPU currently being utilized. To evaluate the performance of the website, I chose the function of showing book to the index page when user start the appication as an example.

First, I put two break point at the starting point and ending point of the function Index

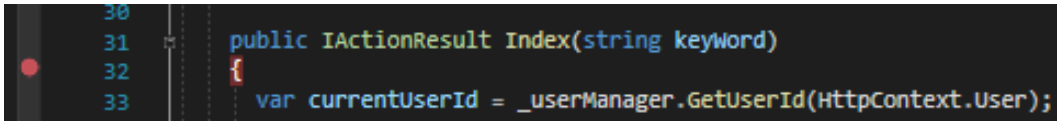


Figure 77: Starting the break point

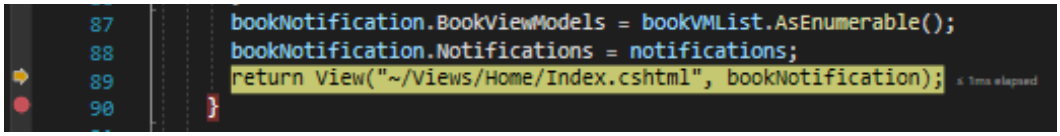


Figure 78: Ending the break point

Then, I opened the dianostic tool in the Visual Studio IDE, this tool will help to display the statistical figure of my application's performance when I run it

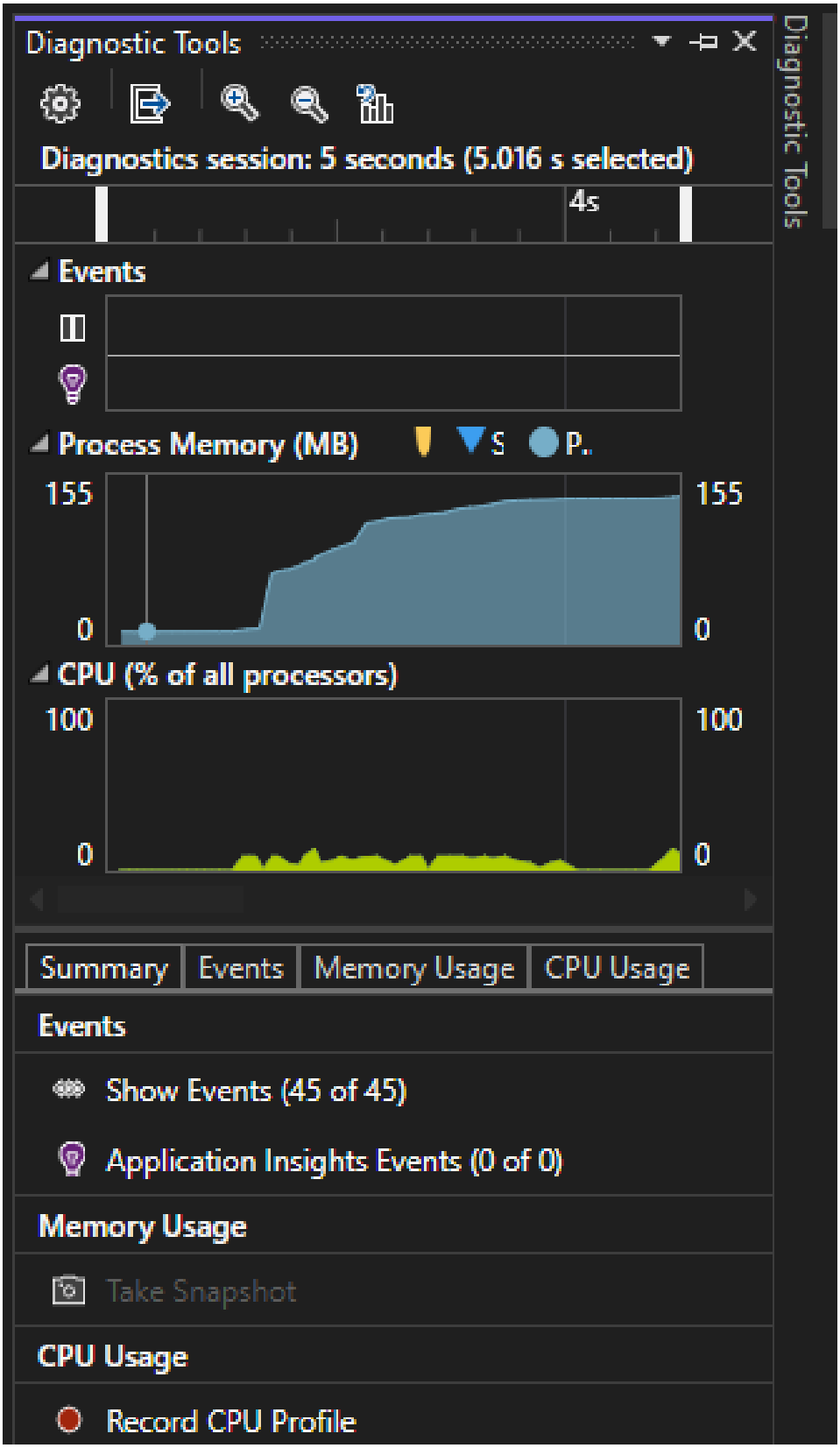


Figure 79: Diagnostic tools

After I ran the application with the Debug mode, here is some of statistical figure about the performance of my website

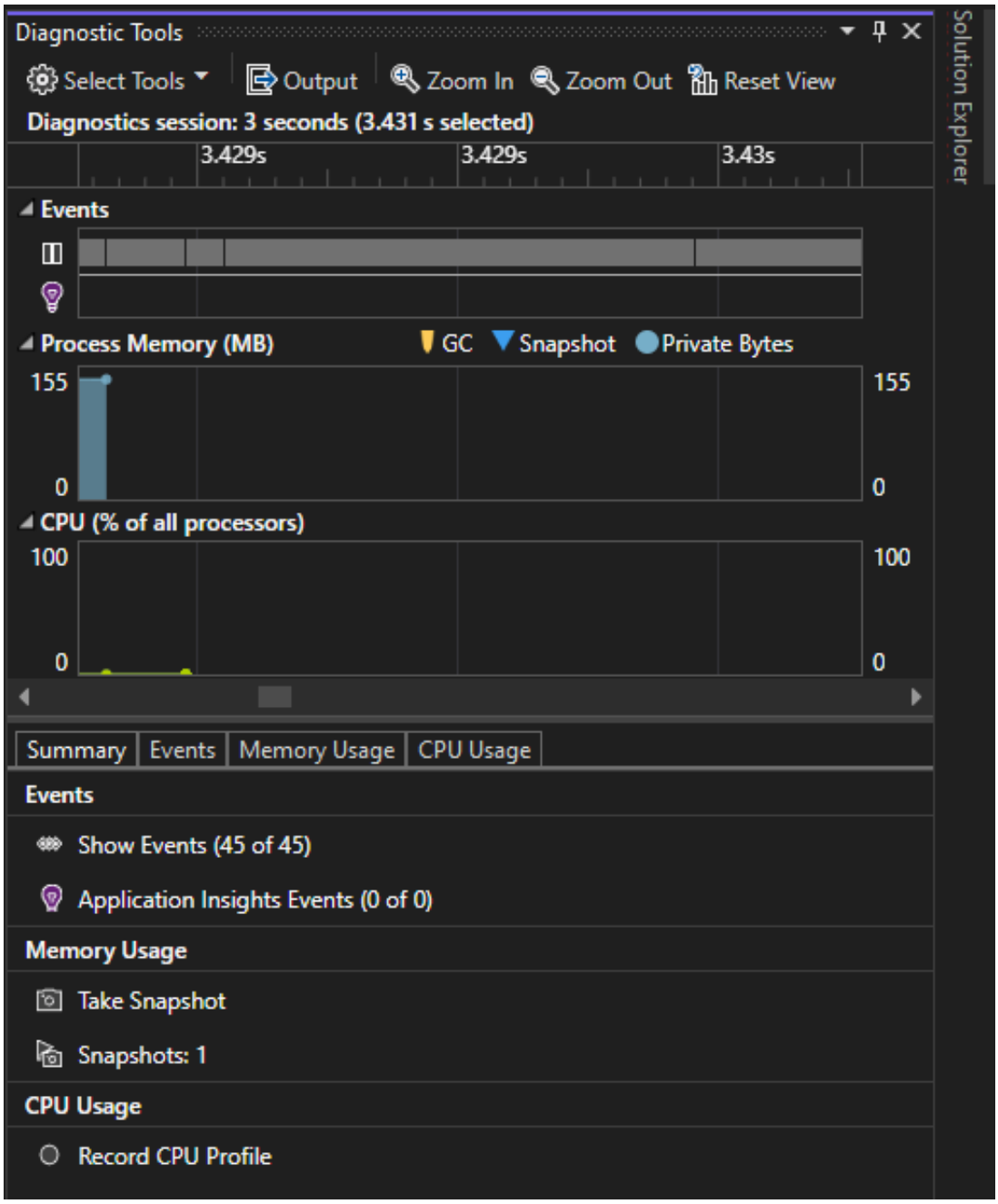


Figure 80: Statistical figure about website performance

Based on the image above, we can see that rendering and displaying the book data on the screen took 3,431 seconds. During the execution, the program performed 45 events and consumed 155 MB of process memory

Here is the list of events happened when the program implement the function. We can see that the program implements the function step by step.

Event	Time	Duration	Thread
Breakpoint: HomeController.cs line 32	2.26s	2,263ms	[16956]
Step: HomeController.cs line 33	2.26s	1ms	[16956]
Step: HomeController.cs line 34	2.26s	2ms	[16956]
Step: HomeController.cs line 35	2.26s	1ms	[16956]
Step: HomeController.cs line 36	3.28s	1,020ms	[16956]
Step: HomeController.cs line 37	3.28s	1ms	[16956]
Step: HomeController.cs line 67	3.28s	1ms	[16956]
Step: HomeController.cs line 71	3.40s	118ms	[16956]
Step: HomeController.cs line 71	3.40s	1ms	[16956]
Step: HomeController.cs line 71	3.40s	1ms	[16956]
Step: HomeController.cs line 71	3.40s	1ms	[16956]
Step: HomeController.cs line 72	3.40s	1ms	[16956]
Step: HomeController.cs line 73	3.40s	1ms	[16956]
Step: HomeController.cs line 75	3.40s	1ms	[16956]
Step: HomeController.cs line 77	3.40s	1ms	[16956]
Step: HomeController.cs line 85	3.41s	7ms	[16956]
Step: HomeController.cs line 86	3.41s	1ms	[16956]
Step: HomeController.cs line 71	3.41s	1ms	[16956]
Step: HomeController.cs line 71	3.41s	1ms	[16956]
Step: HomeController.cs line 72	3.41s	1ms	[16956]
Step: HomeController.cs line 73	3.41s	1ms	[16956]
Step: HomeController.cs line 75	3.41s	1ms	[16956]
Step: HomeController.cs line 77	3.41s	4ms	[16956]
Step: HomeController.cs line 85	3.41s	2ms	[16956]
Step: HomeController.cs line 86	3.41s	1ms	[16956]
Step: HomeController.cs line 71	3.41s	1ms	[16956]
Step: HomeController.cs line 71	3.41s	1ms	[16956]
Step: HomeController.cs line 72	3.41s	1ms	[16956]
Step: HomeController.cs line 73	3.41s	1ms	[16956]
Step: HomeController.cs line 75	3.42s	1ms	[16956]
Step: HomeController.cs line 77	3.42s	2ms	[16956]

Figure 81: Statistical figure about website performance (2)

After that, using the Diagnostic Tools, we can determine how much CPU usage the app consumes while it is executed. When we look at the CPU Usage, we can see a list of functions ranked according to the time they have been running, with the process running the longest, displaying first under Top Functions. Here is the pie chart of CPU usage

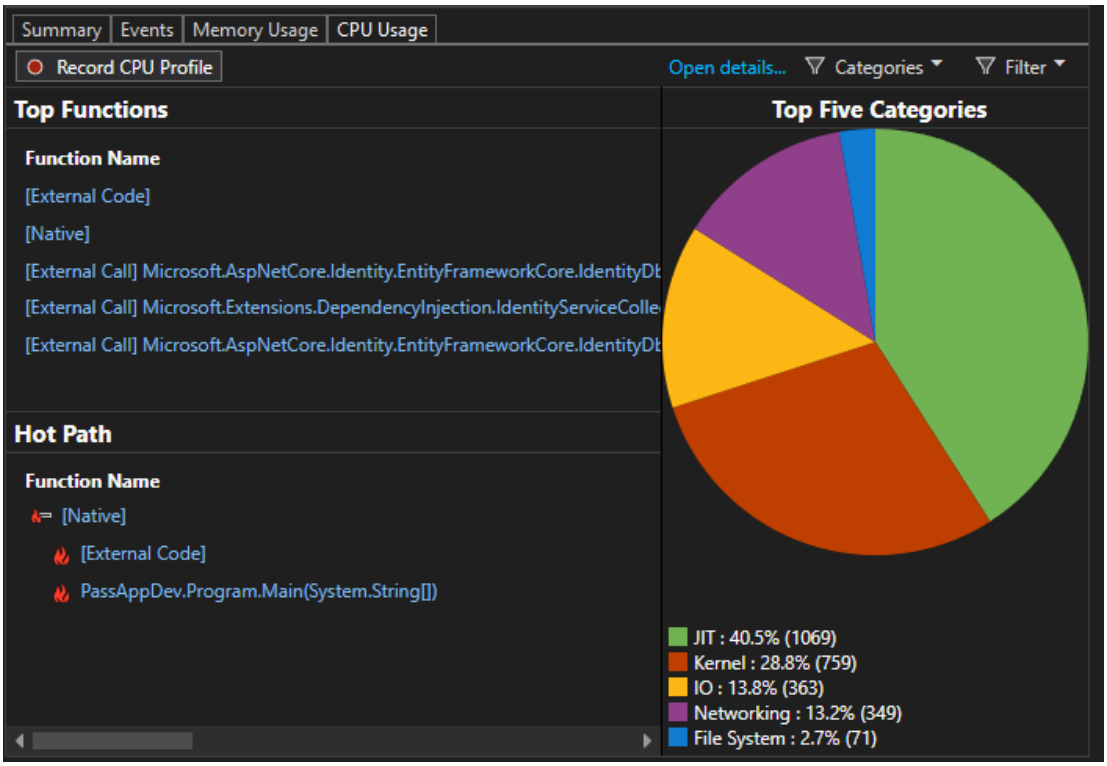
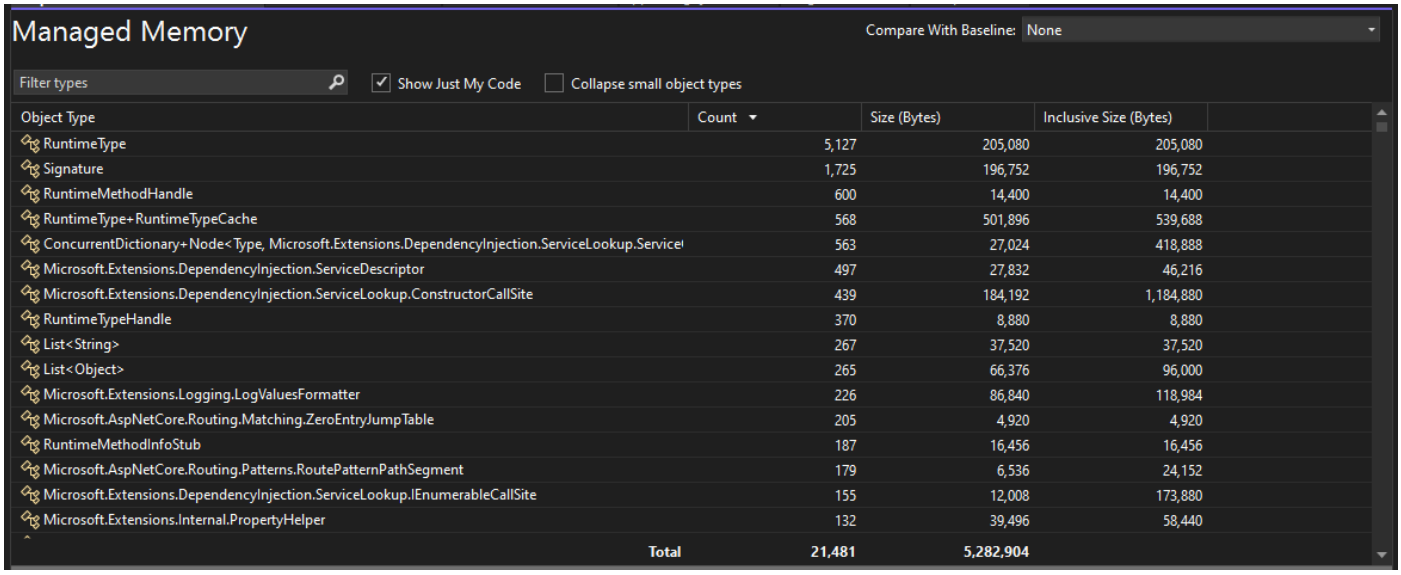


Figure 82: Statistical figure about website performance (3)

I can gain insight into the precise condition of the system at any given point in time if at least one memory snapshot is taken. When we check out the program's characteristics, we can see the memory usage-related parameters by looking at the figure provided below.



Object Type	Count	Size (Bytes)	Inclusive Size (Bytes)
RuntimeType	5,127	205,080	205,080
Signature	1,725	196,752	196,752
RuntimeMethodHandle	600	14,400	14,400
RuntimeType+RuntimeTypeCache	568	501,896	539,688
ConcurrentDictionary+Node<Type, Microsoft.Extensions.DependencyInjection.ServiceLookup.Service	563	27,024	418,888
Microsoft.Extensions.DependencyInjection.ServiceDescriptor	497	27,832	46,216
Microsoft.Extensions.DependencyInjection.ServiceLookup.ConstructorCallSite	439	184,192	1,184,880
RuntimeTypeHandle	370	8,880	8,880
List<String>	267	37,520	37,520
List<Object>	265	66,376	96,000
Microsoft.Extensions.Logging.LogValuesFormatter	226	86,840	118,984
Microsoft.AspNetCore.Routing.Matching.ZeroEntryJumpTable	205	4,920	4,920
RuntimeMethodInfoStub	187	16,456	16,456
Microsoft.AspNetCore.Routing.Patterns.RoutePatternPathSegment	179	6,536	24,152
Microsoft.Extensions.DependencyInjection.ServiceLookup.IEnumerableCallSite	155	12,008	173,880
Microsoft.Extensions.Internal.PropertyHelper	132	39,496	58,440
Total	21,481	5,282,904	

Figure 83: Statistical figure about website performance (4)

II. Personal Evaluation (P6)

Good Points:

- ✓ Regarding user requirements about functionalities, our application has fully met most of the requirements for all three roles: customer, store owner, and admin.
- ✓ The application has an excellent interface; the layout is straightforward and visual, and the color scheme is tasteful, which is simple for users to comprehend and enables them to interact with the application when they use it without difficulty.
- ✓ The application's performance is still satisfactory for the quantity of data currently being used

Future Improvements:

- Regarding the performance aspect, the database of the FPTBook system only contains a relatively small quantity of data, this is not a remarkable difference. However, as the system data becomes bigger with more data added, it will cause an overall delay in the website's loading speed. It will affect both the functionality and the experience of the endpoint users. Pagination and lazy loading will help to improve the performance problems. The data can be segmented so that the subsequent data is only shown when the users go deeper into the website
- Regarding the technical aspect, the source code is not clean because it has many things that need to be rearranged, edited, and refactored to make it standard and easier to maintain in the future.
- Regarding the functional aspect, our website has met most of the essential functions of a website for buying and selling books. However, .for today's users' needs, our website lacks a lot in terms of functionality. In the future, we will try to improve our website by adding more functions like a book review, adding books to the favorite list, commenting on books, etc.

III. Factors That Influence The Performance Of The Application (M5)

1. Components Impact The Application Performance

Algorithm Complexity

One of the factors that have the most significant impact on an application's performance is the complexity level of the program's algorithm. Today's applications and services, particularly those provided via the Internet, are a mosaic of components sourced from multiple locations such as the data center, the cloud, third-party suppliers, and so on. Algorithms are described as how a program works; the more steps and complexity an algorithm has, the longer the program takes to execute. Therefore, algorithm optimization is essential in optimizing the performance of the application. It will help the other components like the cloud or browser can render the application faster.

Application Design

The way an app is designed significantly affects how well it performs. There is a requirement for performance design. Once applications are created, performance goals and the specifics of the environment in which they will operate need to be planned. Low application performance can only be avoided if the application developers are subject to strict quality control and processes early in the application lifecycle. If an application is not planned in detail and meticulously, its performance will not be as expected.

Networking

With online applications accessed through the internet, the most critical factors are latency and network capacity (remote workforce, customer-facing websites, web applications, etc.). The network the program is connecting to significantly affects its performance, and this is especially true for cloud computing and mobile computing. Variable bandwidth, high jitter, prolonged latency, and packet loss negatively impact application performance. So, having a solid and stable internet connection between the client and server sides will help improve website performance.

Server Hardware

A server is a computer connected to a computer network or the Internet, has a static IP, and has a high processing capacity. On it, people install software to serve other computers to request services and resources. In short, the server is a computer like the client computers but must meet higher requirements in terms of features and performance. Therefore, it is essential to use a computer with a powerful configuration as a server for the application because it helps to process requests from customers quickly and efficiently. Using a low-end computer as a server will significantly reduce the website's performance.

2. Conclude The Application Review

Regarding **application design,** our website is created with the Entity framework - a robust framework supporting the creation of websites according to the MVC model. That means the project folders and files are all organized intuitively and easily, for example, the Model folder, the Controllers folder, the Views folder, the Enums folder, etc. Structurally, we need to improve the structure of files and folders and refactor the source code to make the code cleaner and easier to maintain and upgrade to new functions in the future.

Regarding **performance,** our website has wholly met our customers' requirements in terms of functionality. The site has three roles, and the roles interact with each other logically with no errors. However, our source code is still not called perfect in terms of performance optimization. The site's algorithms have very complex syntax and are not optimized for the fastest site performance. So in the future, we will have to refactor the algorithm to work and handle large amounts of data as the site's scale become bigger.

Regarding **testing,** due to human resource constraints, we could only perform manual testing - a simple technique to determine if there was an error during the operation. However, one of the limitations of this method is that it cannot find hidden errors in the application. Therefore, in the future, we will try to use the automation testing method - a testing method using source code written in Python language. This method will help us find hidden errors in the operation and fix them quickly to optimize the user experience.

IV. Evaluate The Strengths And Weaknesses Of The Application (D3)

To assess the strengths and weaknesses of FPT Book's corporate website, I apply the SWOT method - Strengths, Weaknesses, Compewastors, Threats, which is invented by Albert Humphrey invented in the 1960s-1970s in the US. The main objective of a SWOT analysis is to raise awareness of the factors that lead to business decision-making or strategy formulation. To do this, SWOT analyzes the internal and external elements that can affect the viability of a business. In the case of the FPT Book enterprise, it can be divided into four main parts: internal strengths, external strengths, internal weaknesses, external weaknesses.

- **Internal Strengths (Strengths)** are the advantages of the website compared to user requirements and proposed functions compared to other websites in the market.
- **Internal Weaknesses (Weaknesses)** are the limitations of a website compared to user requirements and suggested functions, as well as compared to other websites on the market.
- **External Strength (Opportunity)** is the advantage of a website over the general requirements of the market and the opportunity where a business can improve its website to enhance competitive performance.
- **External Weaknesses (Threats)** are technological, technical limitations, or changes in user factors and competitors' advantages in the market.

Internal Strengths	External Strengths
• Meet the user's requirements • Designed according to the MVC model, convenient for later development • Easy access for new users • Functions interact with each other clearly, smoothly, and without errors • Intuitive, easy-to-see interface	• Many technologies support interface design as well as back-end processing today. • An easy-to-use website design will help attract many customer segments of all ages • More effective marketing tactics • Positive changes in social factors

• Accessible navigation and layout for an essential website • Show necessary information to customers • Works well with current data levels	
Internal Weaknesses • Website interface design is at an acceptable level but has not attracted users • Many functions are not outstanding; for example, the notification function • The site only works well for low amounts of data • The source code is not optimized and needs to be refactored • Lack of responsive cross-device support • The application's algorithm is still lengthy, so the performance is not optimized. • There is a lack of necessary functions such as a rating function, an avatar update function for users, or a link to social networking platforms.	**External Weaknesses** • The scale of users grows, so the size of the website must also increase • The level of security is not high; users are expected to be easily attacked by hackers • The service's stability depends on the server hardware's strength and the internet connection's stability. If these two factors are not guaranteed, the customer's use of the website will be restricted • Unwanted spam and advertisements • Upgraded versions of the browser may lead to missing information in the rendering of website contents.

References

Kenton, W., 2022. *investopedia.com.* [Online]

Available at: https://www.investopedia.com/terms/s/swot.asp

[Accessed 23 August 2022].

microsoft.com, 2022. *microsoft.com.* [Online]

Available at: https://docs.microsoft.com/en-us/visualstudio/profiling/profiling-feature-tour?view=vs-2022

[Accessed 20 August 2022].

Index of comments

3.1 P4
The questionnaire is clear and interesting. Information extracted from questionnaire is useful and could be used for future development
P5
The application is fully developed with images evidence.
The GitHub repository is included.
Sample sources are included
There is proof of the application deployment
Student can answer technical questions properly
P6
I can see you compare between the design and the final product
M3
You extract quite useful information to develop your web app further
M4
You show good evidence about what you have completed the app with source code
M5
Very good information about your app performance based on measurements
Very good insights from the team analysis and ideas for further development
The team uses various tools to measure then analyzes it very well in order to have better insights for further improvements