

## Novel's Marketplace Website

Syed Muizzuddin Syed Anuar<sup>1</sup>, Rozlini Mohamed<sup>1\*</sup>

<sup>1</sup> *Fakulti Sains Komputer dan Teknologi Maklumat,*

*Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

\*Corresponding Author: [rozlini@uthm.edu.my](mailto:rozlini@uthm.edu.my)

DOI: <https://doi.org/10.30880/aitcs.2025.06.01.133>

### Article Info

Received: 29 July 2024

Accepted: 19 June 2025

Available online: 30 June 2025

### Keywords

Novel, Marketplace, E-commerce,  
Web-based, Object-Oriented  
Approach, Payment, Review

### Abstract

The Novel's Marketplace Website project addresses the emerging need for a centralized platform in the digital literary landscape that focusing on the fragmented nature of novel e-commerce, review and interactions within the reader community. The main reason of this development is because of the challenges in the current system, including no dedicated platform exclusively for buying and selling that focus on novels and limited accessibility and inclusivity. The project aims to develop a comprehensive, user-centric website for sharing literary insights, experiences and facilitating the buying and selling of novels that involves buyer and seller of novel's enthusiasts. By applying an object-oriented approach and modern web technologies, this projects develops a web-based platform tailored to the preferences of novel enthusiasts. The methodology involves the design and implementation of key features such as user registration, log in, seller management, novel cataloging, purchase and payment, review submission, report management. The outcome is a novel's marketplace that effectively involves novel reviews, enhances user interaction, and provides an inclusive platform for both readers and sellers. This website has significantly improved accessibility, user engagement and the quality of literary discourse. The project paves the way for future enhancements in online literary communities, suggesting areas for further research in user experience optimization and community-driven features. The Novel's Marketplace Website stands as a testament to the potential of digital platforms in transforming literary engagement and fostering a vibrant, inclusive community of novel lovers.

## 1. Introduction

This study delves into the creation of an innovative Novel's Marketplace Website, a pivotal solution addressing the fragmented landscape of novel reviews and sales. Online user reviews are crucial for consumers, serving as a supplement to word-of-mouth communication and fostering brand loyalty through community content on websites [1]. Developing a dedicated platform for literary enthusiasts has become paramount in a world shaped by the written word, where literature serves as a gateway to new worlds, ideas, and emotions.

Recognizing the challenges in the current system, including the prevalence of e-commerce websites, none are solely focused on the buying and selling of novels. Most of the current platforms cater to a wide range of products, lacking a specialized focus on novels. Even Goodreads, which is a well-known platform for book

enthusiasts, does not facilitate the direct sale of novels between users. Instead, Goodreads redirects users to Amazon platform for purchasing and does not offer a feature for users to sell their used novels directly on the site. This gap in the market highlights the need for a dedicated platform where users can exclusively buy and sell novels. Then, most of e-commerce platform's limited accessibility and exclusivity have hindered its potential for inclusivity and engagement among users. The exclusivity of the platform, where only the company can sell novel's product.

Novel's Marketplace website is developed to overcome these challenges. The proposed system is tailored for a diverse audience including administrators, sellers, and buyers and will feature user-centric design and functionality, enabling seamless interaction, transaction, and engagement within the literary community. The objectives encompass the design, development, and testing of this web-based platform, focusing on user interaction, novel transactions, and content sharing. This study anticipates that the resulting platform will bridge significant gaps in literary engagement, offering an inclusive, and comprehensive environment for novel enthusiasts. The expected outcomes include a robust, scalable database and an intuitive interface facilitating novel discovery, sales, and insightful discourse, ultimately transforming this platform into a cornerstone for novel enthusiasts.

## 2. Related Work

Novels are personal, narrating individual characters' experiences, exploring feelings, thoughts, and complex ideas, often conflicting with previous literature, catering to individual readers [2]. Marketplaces connect vendors with customers, driving sales through a multi-vendor platform, and enhancing visibility for sellers [3]. Navigating the current landscape of novel marketplaces presents challenges for users, who search multiple platforms for reliable reviews. A centralized platform is needed to provide a streamlined experience for buyers and sellers. The platform aims to facilitate the sale of preloved novels, offering opportunities for both sellers and buyers to share their literary insights and experiences. This system aims to encourage user interaction, open opportunities for both sellers and buyers, and foster a thriving community for novel lovers.

The Novel's Marketplace Website will employ e-commerce system method features in its development. The main concern in the development of this system is to manage buying and selling process, review submission process and fostering engagement of novel's enthusiast's community. E-commerce involves using the internet and electronic media to trade products and services directly with customers. It's rapidly expanding in the global corporate sector due to ICTs, allowing consumers to compare costs and observe variations [4]. The e-commerce system comprises seven key components: dynamic pricing, supply chain management, customer engagement, rich product information, shipping and returns management, product quality, and loyalty and marketing [5].

This system is developed using web-based technology. This system involves multiple users and buyers, sellers and administrators who can access the system at the same time. The proposed system is also more manageable as buyers, sellers and administrators can access the system using an internet connection, a common browser and do not need to install the system to a computer. Web-based e-commerce enhances communication between customers, sellers, and business relationship managers through electronic resources like email and web pages, making it more accessible and efficient. [6]. A web-based e-commerce system should have a user-friendly navigation system, organized products into logical categories, and a site search feature. Product reviews and ratings are essential for providing social evidence [7].

There are several e-commerce systems that allow sellers to list their products for sale and enable buyers to buy these products online. Therefore, conducting a review of existing systems becomes crucial as it helps in understanding the system architecture and provides guidelines for developing the project. Three existing systems have been reviewed and studied which are Carousell, Goodreads, and eBay, as shown in Table 1.

**Table 1: Comparison of existing systems with proposed system**

Feature/System	Carousell	Goodreads	eBay	Novel's Marketplace Website
Register and log in	Yes	Yes	Yes	Yes
Catalog	Yes	Yes	Yes	Yes
Review submission	Yes	Yes	Yes	Yes

Seller management	Yes	Link to another platform (Amazon.com)	Yes	Yes
Purchase and Payment	Yes	Yes	Yes	Yes
Content (novels/books)	No	Yes	No	Yes
User	Seller Buyer	Reader Community	Seller Buyer	Seller Buyer

From this comparison, it shows that the similarities and differences in terms of the features and target users. The registration features of Carousell, eBay and Goodreads are related to the Novel’s Marketplace Website system. The log in feature that contains email and password. Next, the comparison of users is the Carousell, eBay and Novel’s Marketplace Website has a related users that focuses on an e-commerce aspect, facilitating the buying and selling of novels in addition to community engagement such as the system’s users are among sellers and buyers which are any users can be a seller and buyers. While Goodreads primarily focuses on building a reading community and book recommendations. The Goodreads and proposed system have related content which focuses on novel-related context content that a platform for novel enthusiasts, but Goodreads is linked to another platform for a buying and selling process different with the proposed system. While the Carousell and eBay are covered a wide range of product.

### 3. Methodology/Framework

This section provides an overview of the methodology and analysis employed in this project. The system development for the web-based novel’s marketplace system follows the prototyping model.

#### 3.1 System Development

The Prototyping Model is chosen for this project. The prototyping model is a systematic process in systems development that involves building, testing, and reworking a prototype until an acceptable outcome is achieved [8]. Prototyping aims to align product requirements, designer ideas, and user models through usability testing, evaluation methods, and tools [9]. The prototyping model comprises several phases including initial requirements, system design, prototype construction, user evaluation, review and update, implementation, and testing of the final product.

The initial requirement phase involves planning and analysis to determine the problem and estimated time required for the development of a novel's marketplace website. This phase includes analyzing the information process and collecting critical data through interviews with novel sellers, reviewers and buyers through social media and observing existing e-commerce systems. The unified modelling language (UML) diagram, Entity Relationship Diagram (ERD), and flowchart diagram are created. The design phase involved designing mock-up user interfaces using draw.io and a database schema using Laragon, ensuring a robust and scalable structure for user information, reviews, and reports. The first prototype phase begins with designing an early version of the system using Laravel framework and Visual Studio Code software. This involves coding user interfaces for registration, login, profile management, catalog, and seller management. The second prototype enhances the initial design by incorporating feedback, focusing on product details, shopping cart, review submission, payment module, and admin dashboard user interfaces. The implementation phase involves developing the final system based on the prototype, ensuring the code is suitable for development and testing. The testing phase involves testing the novel's marketplace website by administrators, sellers, and buyers to ensure smooth performance of each module and identify any errors. Table 2 shows system development activities and their task.

**Table 2: System Development activities and their tasks**

Phase	Task	Output
Initial Requirements	Determining the problem and project timeline. Analyze the requirements, interviews and observations. Visualize the design of the system.	<ul style="list-style-type: none"> <li>• Proposal</li> <li>• Project requirement</li> <li>• UML</li> <li>• ERD</li> <li>• Flowchart</li> </ul>

Design	The system architecture was designed using draw.io mockups user interface each module and Laragon for database schema.	<ul style="list-style-type: none"> <li>• Database design</li> <li>• User interface design</li> </ul>
Prototype 1	Designing and coding early versions of the registration, login, profile management, catalog, and seller management user interfaces using Laravel framework and Visual Studio Code software.	<ul style="list-style-type: none"> <li>• Prototype I</li> </ul>
User's Evaluation and Modification	Determine whether the prototype satisfies the user's requirement or not. Any modification is needed based on buyer's and seller's feedback.	<ul style="list-style-type: none"> <li>• Buyer's and Seller's reviews</li> </ul>
Prototype 2	Focusing on product details, shopping cart, review submission, payment module, and admin dashboard user interfaces.	<ul style="list-style-type: none"> <li>• Prototype II</li> </ul>
Implementation	Develop the final product based on all user's requirements, feedback and prototype.	<ul style="list-style-type: none"> <li>• Full report of project</li> <li>• Completed website</li> </ul>
Testing	Test the validity of the website.	<ul style="list-style-type: none"> <li>• Test cases</li> <li>• Workable website</li> </ul>

### 3.2 Requirement Analysis

System Requirement Analysis is the method used to identify the requirements and needs for a new product [8]. System requirements analysis is a crucial process for determining the system's behavior and limitations, guiding the development of Novel's Marketplace. It involves a comprehensive breakdown of modules, functional, non-functional, and user requirements.

**Table 3: System's module**

Module	Function	User
Registration, login and manage profile	Create accounts, login securely and manage their profile.	<ul style="list-style-type: none"> <li>• Seller</li> <li>• Buyer</li> </ul>
Seller management	Seller can add their novel's product for sale. The notification will pop up when their novels product has been purchased.	<ul style="list-style-type: none"> <li>• Seller</li> </ul>
Catalog	Search for novels, view novels details, access comprehensive listings of novels and update the latest novels that sellers add for sale.	<ul style="list-style-type: none"> <li>• Seller</li> <li>• Buyer</li> <li>• Administrator</li> </ul>
Purchase and payment	Buyers can proceed with a payment for purchasing the novels.	<ul style="list-style-type: none"> <li>• Buyer</li> </ul>
Review submission	Submit novel reviews, ratings, and comments, post comments, facilitate community discussions, and rate books for valuable feedback of novels that have been purchased.	<ul style="list-style-type: none"> <li>• Buyer</li> </ul>
Report Management	Manage report of the novel's sale, reviews and others.	<ul style="list-style-type: none"> <li>• Administrator</li> </ul>

Table 3 outlines the modules of the proposed system. The first module allows both buyer and seller to create accounts and login to the website. Seller management allows sellers to manage their product such as add novel's product for sale. The catalog module facilitates novel discovery, enabling Sellers, Buyers, and Administrators to search, view details, and access comprehensive listings, ensuring users stay updated on the latest additions. The

reviews submission allows buyers to facilitate the submission of reviews, ratings and comments to provide feedback for the novels. The purchase and payment allow buyers to securely proceed with novel purchases. Report management is handled by administrators to encompass the management of sales, reviews, and other essential reports for effective system oversight.

Table 4 summarizes the functional requirements of the proposed system. It outlines the registration, login, and manage profile modules for new users, allowing them to securely log in and manage profiles. The seller management module allows sellers to add novel products, while the purchase and payment module allows buyers to proceed with purchases. The review submission module allows buyers to rate and review novels.

**Table 4: Functional requirements for the proposed system**

Requirement	Description
Registration, login and manage profile	<ul style="list-style-type: none"> <li>New users can register as sellers or buyers, existing users can login, manage profiles, and be directed to the catalog upon successful login.</li> </ul>
Seller management	<ul style="list-style-type: none"> <li>The system allows registered sellers to add new products and edit existing ones, including setting prices, descriptions.</li> </ul>
Catalog	<ul style="list-style-type: none"> <li>Displays a list of sellers' novels with details like image, price, and title, allows the administrator to update, allows users to browse, and adds products to the cart.</li> </ul>
Purchase and payment	<ul style="list-style-type: none"> <li>The user can proceed with purchasing products in their cart, fill in shipping information, transfer payment method, and receive a notification from the seller.</li> </ul>
Review submission	<ul style="list-style-type: none"> <li>The system allows users to rate a purchased novel product using a numerical rating system and post comments and reviews.</li> </ul>
Report Management	<ul style="list-style-type: none"> <li>Allow administrator to generate report of review data, purchase data, novel's data and user's data.</li> </ul>

Table 5 outlines the non-functional requirements of the proposed system, including performance, operational characteristics, and security. It aims to ensure reliable system performance, dependency on internet connectivity, and robust security measures to protect user data and access.

**Table 5: Non-functional requirement for the proposed system**

Module	Description
Performance	<ul style="list-style-type: none"> <li>The system should consistently smooth and responsive user experience</li> </ul>
Operational	<ul style="list-style-type: none"> <li>The loading time required for a website is no more than 1 minute.</li> </ul>
Security	<ul style="list-style-type: none"> <li>The system should be user friendly, robust encryption and authentication processes to protect user information.</li> </ul>
Cultural and political	<ul style="list-style-type: none"> <li>The system should be able to work on any web browser.</li> </ul>

Table 6 outlines the user requirements for the proposed system. User requirement analysis systematically explores the functionality and features desired by different user roles including sellers, buyers and administrators.

**Table 6: User requirements of the proposed system**

User	Description
Administrator	Login, manage and moderate platform content, update product listings, monitor buyer reviews, and generate reports on sales, user activity, and product reviews to ensure compliance with community guidelines and standards.

Seller	Register, login, manage profile and list their novel products, including title, author, genre, and pricing, and receive notifications when a product is sold.
Buyer	Register, login, manage profile and browse the catalog, select novels, add them to their cart, purchase by shipping information, select payment method, receive a receipt, and submit reviews, rate, and comment on product pages for community feedback.

For development, Windows 7 or above is recommended, with a 2.0 GHz CPU, 4 GB RAM, and 256 GB SSD storage. Wi-Fi or hotspot connections are recommended. Test websites with Google Chrome, Internet Explorer, or Microsoft Edge, and use MySQL for database management. Laragon, Laravel framework, and Visual Studio Code are recommended development software.

### 3.3 System Analysis

The use case diagram in Figure 1 outlines the registration, login, and profile management process for buyers and sellers. Users create accounts, log in, and manage their profiles. They access a catalog with detailed information, add products, and update the catalog. Buyers can browse, search, and add novels to their carts. The purchase and payment use case allows buyers to purchase products, select payment methods, receive notifications, generate receipts, rate products, post comments, and generate reports on review, purchase, novel, and user data.

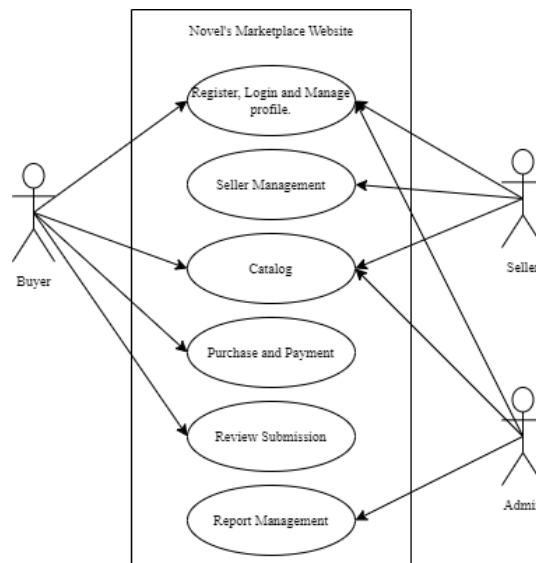


Figure 1: Use Case Diagram

The connections between classes are identified and will continue to be described using Entity Relationship Diagram (ERD). In Figure 2, eight classes are identified in the proposed system which are Admin, Buyer, Seller, Order, Novel, Review, Category and Notification. Each class has its own operations, with the admin class managing admin details, the Buyer class handling user-related operations, the Seller class managing novel listings, the Order class managing order processes, the Novel class managing novel-specific information, the Review class handling review processes, the Category class organizing genres, and the Notification class for timely communication.

A To-Be Model or Flowchart as shown in Figure 3, aids in the analysis and design of a Novel's Marketplace Website, illustrating future state and identifying system efficiencies. It streamlines buying and selling processes, managing user interactions, and allowing buyers, sellers, and admins.

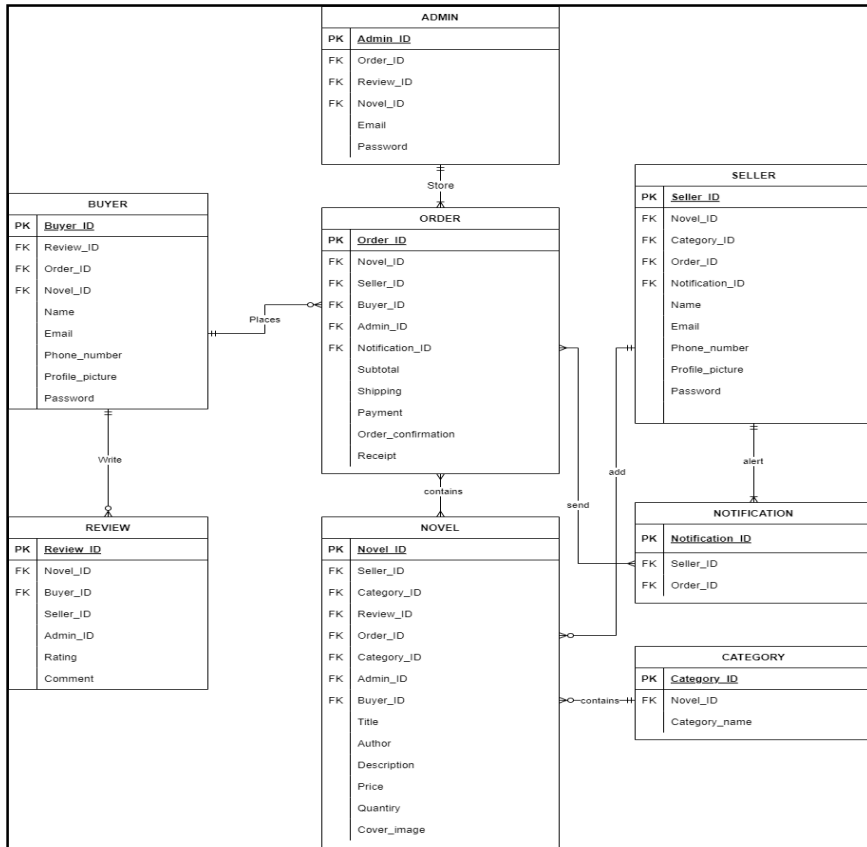


Figure 2: Entity Relationship Diagram (ERD)

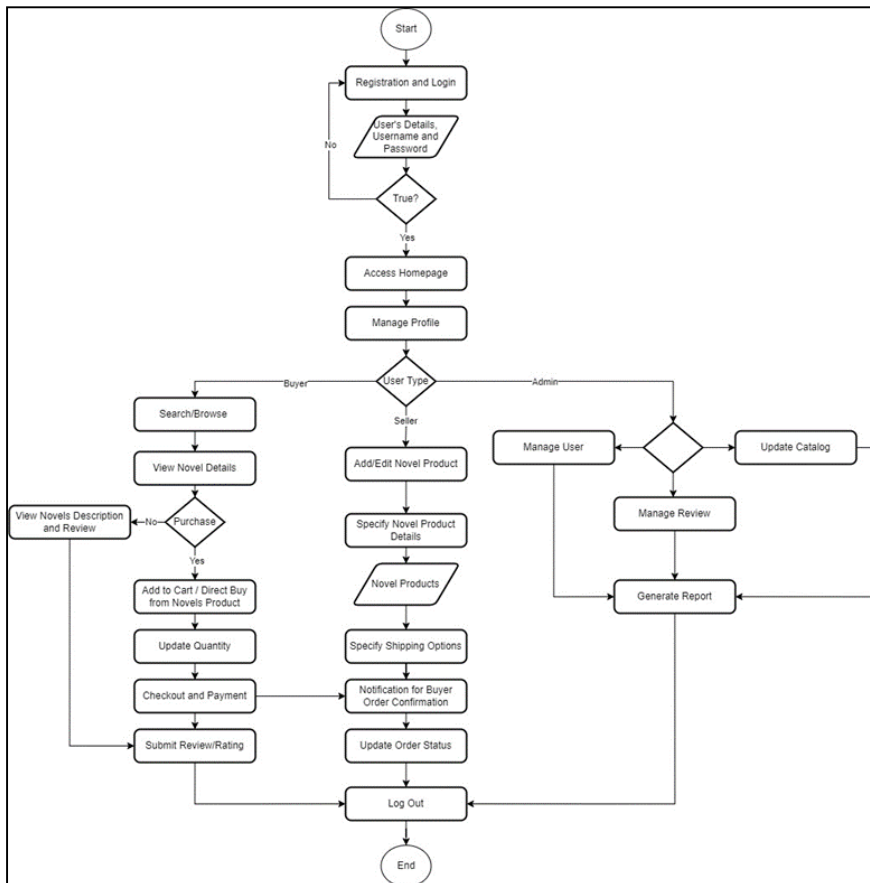


Figure 3: Flowchart

### 3.4 Design

Novel's Marketplace Website features a system architecture diagram as shown in Figure 4, with three user roles: buyers, sellers, and admins, ensuring a seamless online marketplace experience.

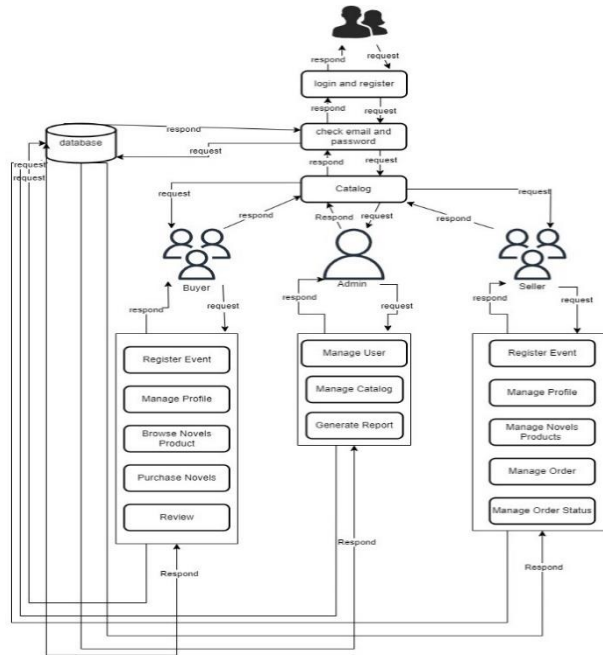


Figure 4: System Architecture

The following are the interfaces that have been designed based on each process. User interface design is a crucial aspect of software or system development, shaping the user's first impressions of website interfaces [10].

#### 3.4.1 Registration, Login and Manage Profile Interface

The user interface of registration, login and manage profile as shown in Figure 5. Users fill out a form with their name, email address, password, and registration role, choosing buyer or seller. They can register by clicking the register button. Then, users need to login to the system in login interface in Figure 6 using their correct email and password, which are then validated. If correct, they are directed to the main page, and if incorrect, an error message is displayed, requiring re-entry.

The registration form includes a logo at the top left. It contains the following fields: 'Name' (text input), 'Email' (text input), 'Password' (text input), 'Confirm Password' (text input), and 'What are you registering as?' (dropdown menu with 'Buyer' selected). At the bottom, there is a link 'Already registered?' and a 'REGISTER' button.

Figure 5: Registration

The login form includes a logo at the top left. It contains the following fields: 'Email' (text input), 'Password' (text input), and a 'Remember me' checkbox. At the bottom, there is a link 'Forgot your password?' and a 'LOG IN' button.

Figure 6: Login

Users can manage their profile, update account information, and enhance security by updating their password in Figure 7.

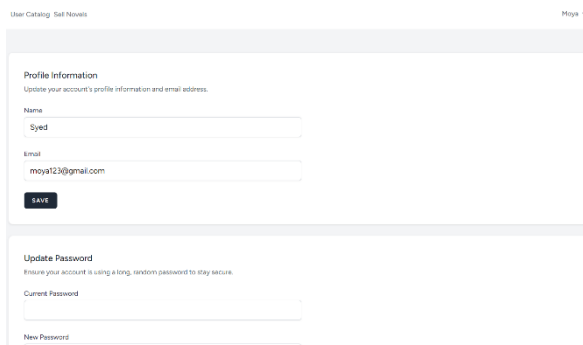


Figure 7: Manage Profile

### 3.4.2 Catalog Interface

Then the system displays a curated selection of novels in Figure 8, with details like title, author, genre, and cover image. Users can search for specific novels or filters based on preferences. Additional details can be found in each entry. The catalog also allows adding novels to the shopping cart.

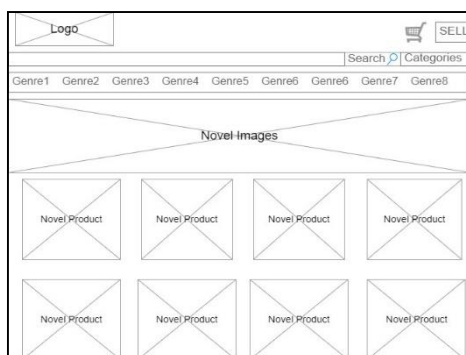


Figure 8: Catalog

### 3.4.3 Seller Interface

The seller's interface in Figure 9 allows them to manage their offerings by adding novels, entering titles, prices, author information, and classifying them. They can upload cover images and submit their listing to the catalog.

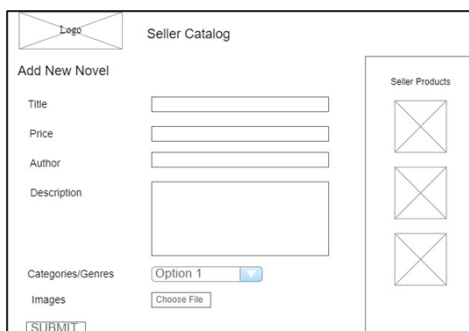


Figure 9: Seller Catalog

### 3.4.4 Payment Interface

After selecting a novel product, buyers can proceed to the checkout interface in Figure 10, where they can enter their delivery address, review product details, choose shipping, and choose a payment method, including secure options like online banking, and complete the transaction.

Figure 10: Payment

### 3.4.5 Review Interface

Buyers can submit reviews for a novel's product using the review interface in Figure 11, which allows rating and commenting on the product with a star.

Figure 11: Review

## 4. Result and Discussion

This section focuses on system implementation and testing for the web-based Novel's Marketplace system. It covers the implementation of registration, login, profile, seller management, catalog, review submission, payment and reporting module functionalities. Additionally, it discusses the comprehensive functional testing conducted to the system's performance.

### 4.1 Implementation

This section describes the development of functional modules in a system. Program code is provided to aid clarification.

#### 4.1.1 Registration, Login and Manage Profile Module

Figure 12 shows a code form about the register method validates and processes it that ensures the unique email and matching passwords. It creates a new user with a hashed password if everything is valid. Then, the code in Figure 13 shows the login method authenticates the user. If successful, it redirects to the catalog page. Otherwise, it returns to the login form with an error message. Next, Figure 14 show the updateProfile function validates input data, update user details, handles profile photo changes, saves it to the database, and redirects to the profile page with a success message.

```
9 class UserController extends Controller
10 {
11     public function showRegistrationForm()
12     {
13         return view('registration'); // Replace 'registration' with the actual view name for your registration form
14     }
15
16     public function register(Request $request)
17     {
18         // Validate the form data
19         $validatedData = $request->validate([
20             'name' => 'required|string|max:255',
21             'phonenumber' => 'nullable|string|max:255',
22             'email' => 'required|string|email|max:255|unique:users',
23             'password' => 'required|string|confirmed',
24             'userrole' => 'required|in:buyer,seller',
25         ]);
26
27         // Create a new user using the User model and store it in the database
28         $user = User::create([
29             'name' => $validatedData['name'],
30             'phonenumber' => $validatedData['phonenumber'],
31             'email' => $validatedData['email'],
32             'password' => bcrypt($validatedData['password']),
33             'userrole' => $validatedData['userrole'],
34         ]);
35     }
36 }
```

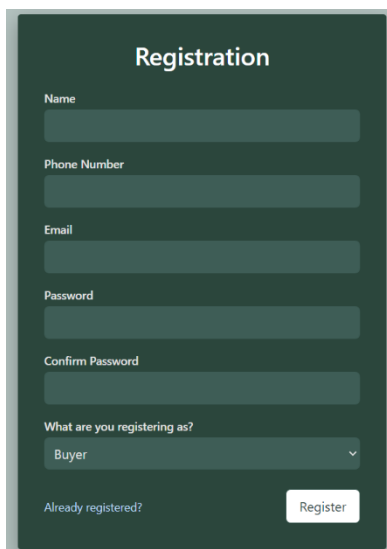
Figure 12: Registration Source Code

```
45 public function login(Request $request)
46 {
47     $credentials = $request->only('email', 'password');
48
49     if (Auth::attempt($credentials)) {
50         // Authentication passed, user is now logged in
51         return redirect()->intended('/'); // Replace '/' with your desired redirect route
52     }
53
54     // Authentication failed, redirect back to the login form with an error message
55     return redirect()->route('login')->with('error', 'Invalid credentials. Please try again.');
```

Figure 13: Login Source Code

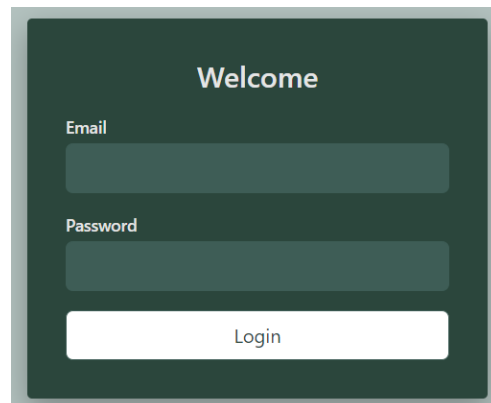
```
71 public function showProfile()
72 {
73     $user = Auth::user();
74     return view('profile', compact('user'));
75 }
76
77 // Method to update the user profile
78 public function updateProfile(Request $request)
79 {
80     $user = Auth::user();
81
82     // Validate the form data
83     $validatedData = $request->validate([
84         'name' => 'required|string|max:255',
85         'phonenumber' => 'nullable|string|max:255',
86         'email' => 'required|string|email|max:255|unique:users,email,'.$user->id,
87         'current_password' => 'nullable|string',
88         'new_password' => 'nullable|string|confirmed',
89         'profile_photo' => 'nullable|image|mimes:jpeg,png,jpg,gif|max:98980',
90     ]);
91
92     // Check if the current password matches
93     if ($request->filled('current_password') && Hash::check($request->current_password, $user->password)) {
94         return back()->withErrors(['current_password' => 'Current password is incorrect']);
95     }
96
97     // Update user details
98     $user->name = $validatedData['name'];
99     $user->phonenumber = $validatedData['phonenumber'];
100     $user->email = $validatedData['email'];
101
102     // Update password if a new password is provided
103     if ($request->filled('new_password')) {
104         $user->password = bcrypt($validatedData['new_password']);
105     }
106 }
```

Figure 14: Manage Profile User Controller Source Code



The registration form is titled "Registration" and is set against a dark green background. It contains several input fields: "Name", "Phone Number", "Email", "Password", and "Confirm Password". Below these is a dropdown menu labeled "What are you registering as?" with "Buyer" selected. At the bottom left, there is a link "Already registered?" and at the bottom right, a white "Register" button.

Figure 15: Registration User Interface



The login form is titled "Welcome" and is set against a dark green background. It contains two input fields: "Email" and "Password". At the bottom, there is a white "Login" button.

Figure 16: Login User Interface

Figure 17: Profile Management User Interface

Next, based on the figure 15 displays the registration page with fields for users to input their details. Then, they can click “Already registered?” to be redirected to the login page. The “Register” button redirects users to login after submission. Then, figure 16 shows the login page features a centered login form with fields for email and password, accompanied by a welcoming message. Upon submitting credentials, users are redirected to the catalog page. Next, figure 17 shows the profile management user interface includes fields for buyer and seller that can update profile info like name, email, password, and upload a new photo. The validation ensures data submission, followed by a success message.

#### 4.1.2 Seller Management Module

```

22 public function create()
23 {
24     return view('sellermanagement'); // Assuming you have a view for the form
25 }
26
27 public function store(Request $request)
28 {
29     // Validate the form data
30     $validatedData = $request->validate([
31         'title' => 'required|string|max:255',
32         'price' => 'required|numeric',
33         'description' => 'required|string',
34         'author' => 'required|string|max:255',
35         'categories' => 'required|string|max:255',
36         'image' => 'nullable|image|mimes:jpeg,png,jpg,gif|max:2048', // Image validation rules
37         'quantity' => 'required|integer|min:1',
38     ]);
39

```

Figure 18: Seller Management Source Code

Based on the Figure 18, the code snippet contains two functions for managing sellers. The 'create' function displays a form for sellers to input item details. The 'store' function validates the submitted data (title, price, description, author, categories, image, and quantity) to ensure format and length constraints are met. If validation passes, the data can be processed further which saving it to a database.

Figure 19: Seller Management User Interface

Next, Figure 19, displays the seller management user interface with a form for adding new products. Sellers input novel details such as title, price, description, author, and category via dropdown. They can upload an image and specify quantity. Upon submission, the product appears on the catalog.

### 4.1.3 Catalog Module

```

40 // Handle file upload if an image is provided
41 if ($request->hasFile('image')) {
42     $titleSlug = Str::slug($validatedData['title']); // Use Str::slug() to format the title
43     $directoryPath = 'assets/' . $titleSlug; // Define the directory path using the title slug
44
45     // Create the directory if it does not exist
46     if (!file_exists(public_path($directoryPath))) {
47         mkdir(public_path($directoryPath), 0777, true);
48     }
49
50     // Store the image in the specific directory
51     $imagePath = $request->file('image')->store($directoryPath, 'public');
52     $validatedData['image_path'] = $imagePath;
53 }
54
55 // Create a new catalog entry in the database
56 Catalog::create($validatedData);
57
58 return redirect()->route('catalog.index')->with('success', 'Catalog entry created successfully!');
59 }
60

```

Figure 20: Catalog Controller Source Code

```

61 public function show($id)
62 {
63     $product = Catalog::findOrFail($id);
64     return view('product', compact('product'));
65 }
66

```

Figure 21: Product Details Catalog Controller Source Code

```

68 public function showCategory($category)
69 {
70     // Retrieve catalogs for the specified category
71     $catalogs = Catalog::where('categories', $category)->get();
72
73     // Pass the catalog data to the view
74     return view('category', ['catalogs' => $catalogs, 'category' => $category]);
75 }
76 }
77

```

Figure 22: Category Catalog Controller Source Code

Based on Figure 20, the code handles the upload image and creates a new catalog entry. It organizes images by creating a directory named after the item's title, checks if it exists, and stores the image there. Then it adds the image path to the data, creates a catalog entry in the database, and redirects the seller with a success message. Next, Figure 21 shows a code for a product details catalog controller that retrieves a specific product by its ID from the database. If found, it passes the product data to the view using the compact method for display. Then, Figure 22 shows a code for Category function that displays catalog item for a selected category by fetching matching items from the catalog model and passing them to the 'category' view along with the category name for dynamic display.

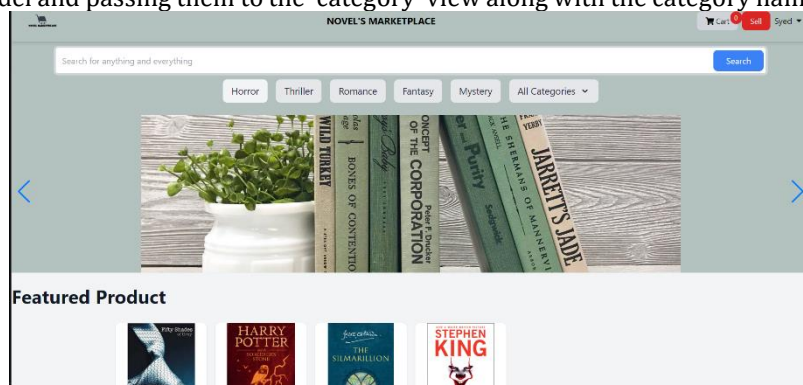


Figure 23: Catalog User Interface

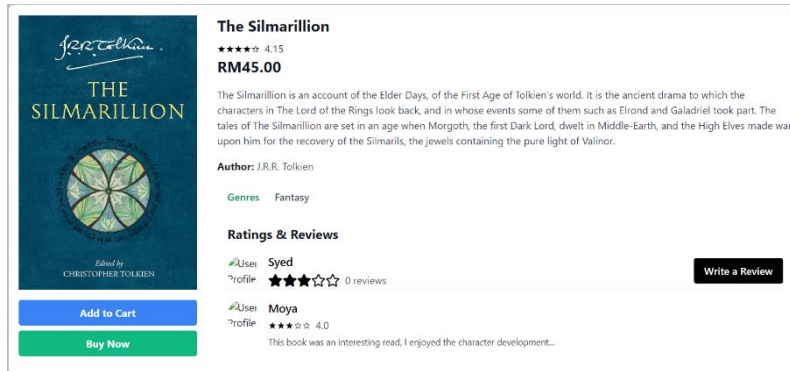


Figure 24: Product Page User Interface

Besides, Figure 23 shows a catalog page that features a search bar for keyword searches and quick access to categories. The navigation bar includes icons for the shopping cart, profile, and a "Sell" button for sellers. A swiper below showcases featured book images. Next, Figure 24 shows a product details page for viewing novel details. It features the book cover, title, rating, and price prominently. Below, a detailed description, author's name, and genres aid in decision-making. Buttons allow for easy purchase, and a review section adds credibility for potential buyers.

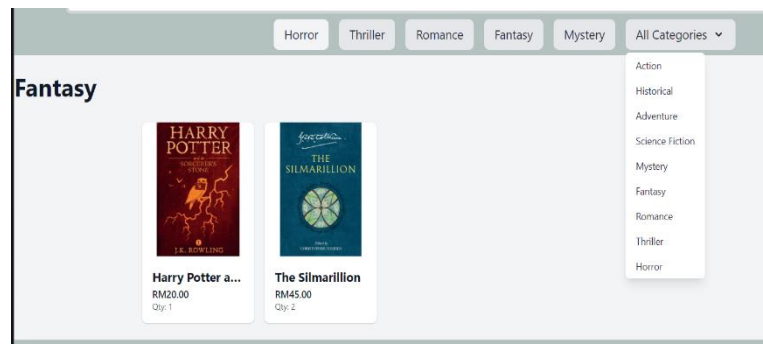


Figure 25: Category User Interface

Other than that, Figure 25 shows a category user interface with a heading showing the selected category and a grid layout of product cards, each featuring details like title, price, quantity, and an image. Clicking a product redirect both buyer and seller to the product page.

#### 4.1.4 Review Submission Module

```

12 public function store(request $request, $catalog_id)
13 {
14     $validatedData = $request->validate([
15         'rating' => 'required|integer|min:1|max:5',
16         'review' => 'nullable|string',
17     ]);
18
19     Review::create([
20         'catalog_id' => $catalog_id,
21         'user_id' => Auth::id(),
22         'rating' => $validatedData['rating'],
23         'review' => $validatedData['review'],
24     ]);
25
26     return redirect()->route('catalog.show', ['id' => $catalog_id])->with('success', 'Review submitted successfully!');
27 }
28
29

```

Figure 26: Review Controller Source Code

Based on Figure 26, the "store" function in the code handles new review submissions for a specific catalog item. It validates the rating input, creates a new review linked to the item and the user, saves it, and redirects the user back to the product page with a success message. This ensures accurate recording and association of reviews with products and buyers.

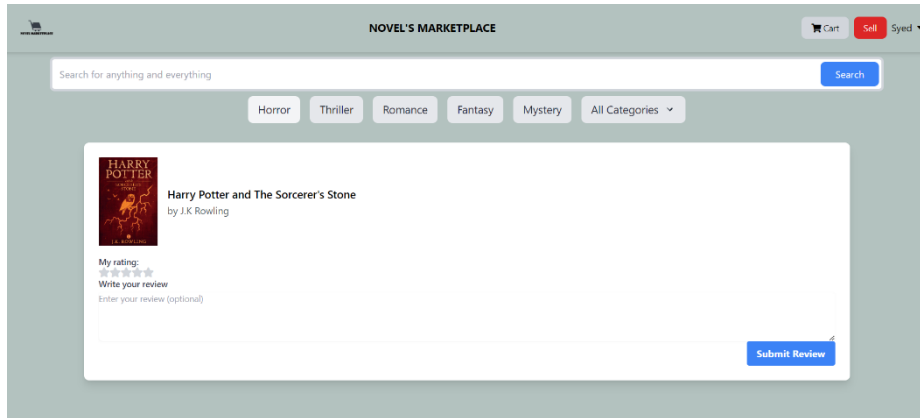


Figure 27: Review Submission User Interface

Next, Figure 27 displays a user-friendly interface for submitting reviews of novels. It includes the novel's image, title, and author's name, with options to rate from one to five stars and submit a review. This minimalist design encourages user engagement and feedback.

#### 4.1.5 Purchase and Payment Module

```

13 class CheckoutController extends Controller
14 {
15     // Show checkout page
16     public function showCheckoutPage(Request $request)
17     {
18         $cart = json_decode($request->input('cart'), true);
19         $user = auth()->user();
20
21         return view('checkout', ['cart' => $cart, 'user' => $user]);
22     }
23
24     // Process checkout
25     public function processCheckout(Request $request)
26     {
27         // Validate the request data
28         $validatedData = $request->validate([
29             'full_name' => 'required|string|max:255',
30             'phone_number' => 'required|string|max:255',
31             'email' => 'required|string|email|max:255',
32             'full_address' => 'required|string|max:255',
33             'cart' => 'required|json',
34         ]);

```

Figure 28: Checkout Source Code

The code manages a shopping cart in JavaScript by fetching product stock, rendering cart items dynamically, updating quantities within stock limits, removing items, and clearing the cart. Total amount is updated in real-time. Besides, Figure 28 shows the code of CheckoutController that manages the checkout process, decoding cart data, retrieving authenticated users, validating input data, calculating subtotal, delivery, platform fee, total amount, creating orders, clearing the cart, redirecting users to payment page, and displaying the payment page.

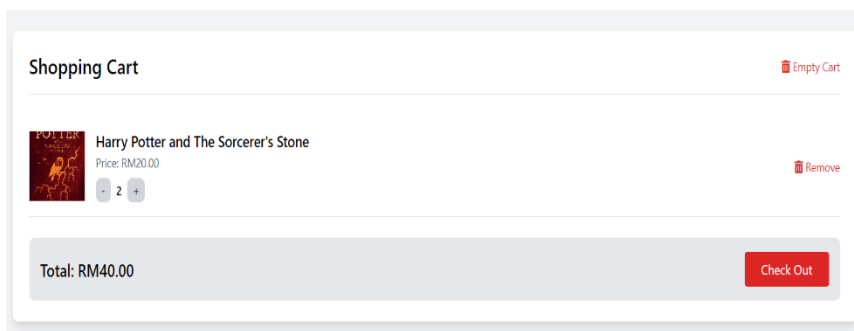


Figure 29: Shopping Cart User Interface

**Address**

Full Name: Syed Phone Number: 0162428044

Email: Muizzuddin621@gmail.com

Full Address: No.16 Jalan 1/10 Taman Ixora, Bandar Baru Salak Tinggi, 43900 Sepang, Selangor

**Order summary**

	Happy Place Quantity: 1	RM16.00
	Harry Potter and The Sorcerer's Stone Quantity: 1	RM15.00
Subtotal		RM31.00
Delivery fee		RM4.90
Platform fee		RM1.00
<b>Total</b>		<b>RM36.90</b>

(Incl. GST where applicable)

[Proceed to Payment](#)

Figure 30: Checkout Page user interface

Then, Figure 29 displays shopping cart user interface displays product details, including image, title, price, and quantity. Users can adjust quantities using real-time updates and stock validation. Error messages are displayed if items exceed stock limits. The interface also includes a remove button, "Empty Cart" button, total cost calculation, and checkout page redirection. Next, Figure 30 show the checkout user interface allows users to input shipping details, review orders, and view items in the cart. The summary includes titles, quantities, prices, subtotal, delivery fee, and platform fee.

```

77 // Show payment page
78 public function showPaymentPage(Order $order)
79 {
80     return view('payment', ['order' => $order]);
81 }
82
83 // Generate receipt
84 public function generateReceipt(Request $request)
85 {
86     $order = Order::with('orderItems.catalog')->find($request->order_id);
87
88     if (!$order) {
89         return redirect()->back()->with('error', 'Order not found.');
```

Figure 31: Payment Source Code

**Payment Details**

**Name:** Syed  
**Phone Number:** 0162428044  
**Email:** Muizzuddin621@gmail.com  
**Address:** No.16 Jalan 1/10 Taman Ixora, Bandar Baru Salak Tinggi, 43900 Sepang, Selangor

Total Price: RM36.90

**Recipient Name:** Nor Alia Husna Binti Mohd Nasir  
**Bank Name:** Maybank  
**Account Number:** 122425252  
**Receipt Reference:** Syed\_Order6  
**File Name:** Syed\_Order6.pdf

[Generate Receipt](#)

Figure 32: Payment User Interface

Next, Figure 31 shows the payment function that manages order payments and receipt generation. It displays payment pages, generates receipts, and sends receipts to users. The sendReceipt method formatted the receipt details and redirects to the catalog page with a success message. Then, Figure 32 the payment user interface displays a summary of the buyer's order, payment information, and necessary bank transfer details. It also features a "Generate Receipt" button, allowing the buyer to generate and download a receipt for their order.

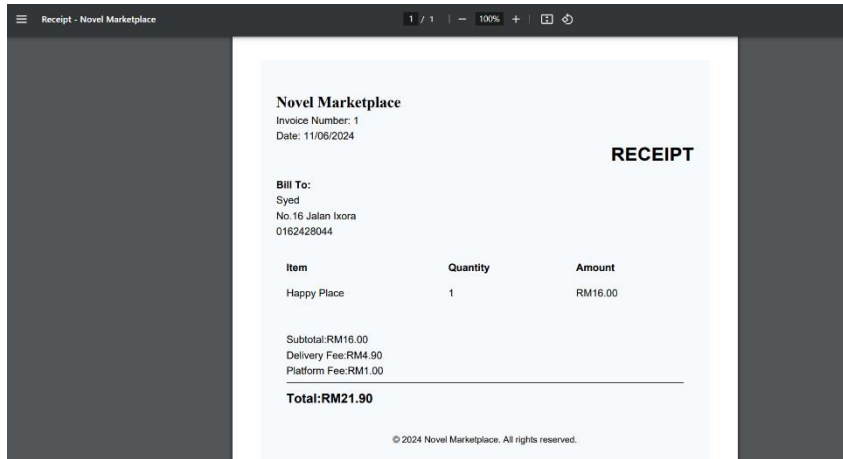


Figure 33: Receipt User Interface

Additionally, Figure 33 shows the receipt PDF generated for an order from the Novel Marketplace displays the essential transaction details in a structured format. It includes the invoice number, date, and billing information with the customer's name, address, and phone number.

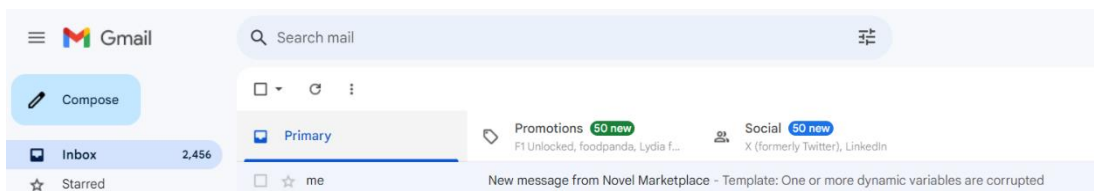


Figure 34: Email Service

Then, Figure 34 shows the the email interface indicates that an email from Novel Marketplace has been received.

### 4.1.6 Report Module

```
14 <body>
15 <div class="flex h-screen bg-gray-200">
16 <!-- Sidebar -->
17 <div class="custom-green-bg text-white w-64 space-y-6 py-7 px-2 absolute inset-y-0 left-0 transform -translate-x-full md:re
18 <h3 class="text-3xl font-semibold text-center text-white">Novel's Marketplace</h3>
19 <div class="mt-10">
20 <a href="dashboard.php" class="block py-2.5 px-4 rounded transition duration-200 hover:bg-gray-400">Dashboard</a>
21 <a href="#" class="block py-2.5 px-4 rounded transition duration-200 hover:bg-gray-400">Users</a>
22 <a href="#" class="block py-2.5 px-4 rounded transition duration-200 hover:bg-gray-400">Catalog</a>
23 <a href="#" class="block py-2.5 px-4 rounded transition duration-200 hover:bg-gray-400">Order</a>
24 <a href="#" class="block py-2.5 px-4 rounded transition duration-200 hover:bg-gray-400">Review</a>
25 </div>
26 </div>
27
```

Figure 35: Admin Dashboard Source Code

Based on the Figure 35, shows the code for admin dashboard that featuring a header, and sidebar for selected report of users, novels, sales, and reviews and summary of the data. The report will be generated and retrieved from the database.

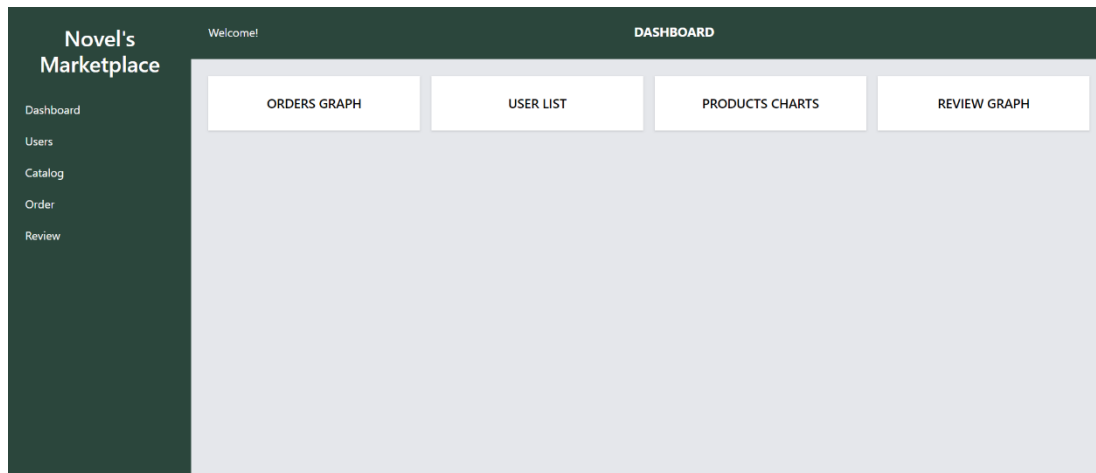


Figure 36: Admin Dashboard User Interface

Next, Figure 36 shows the admin dashboard interface that is designed with a sidebar for easy navigation and a clean, organized content area that displays key metrics like orders, sales, review and product numbers. The centre of the dashboard displays the summary of the data with the data visualize such as graph.

## 4.2 Testing Result

**Table 7: Testing Results**

Module	Description	Expected Result	Actual Result	Pass/Fail
Registration, login and manage profile	To check whether buyer, seller and administrator can register an account	Buyer, seller and administrator able create account	Buyer, Seller and administrator successfully create account	Pass
	To check whether buyer, seller and administrator can log in to system	Buyer, seller and administrator able log in to system	Buyer, seller and administrator successfully log in	Pass
	To check whether buyer and seller can edit their profile	Buyer and seller able to edit their profile	Buyer and seller successfully edit their profile	Pass
Seller management	To check if seller can add novel product	Seller should be able add novel product	Seller successfully added novel product	Pass
Catalog	To check if buyer and seller can search for novels	Buyer and seller should be able search for novels	Buyer and seller successfully searched novels	Pass
	To check if buyer and seller can view novel's details	Buyer and seller should be able to view novels details	Buyer and seller successfully viewed novels details	Pass
	To check if buyer and seller can view novels through categories	Buyer and seller should be able to view novels through categories	Buyer and seller successfully viewed novels through categories	Pass
Purchase and payment	To check if buyer can add novels into cart	Buyer should be able to add novels into cart	Buyer successfully adds novels into cart	Pass

	To check if system can calculate the total price in cart and payment page	System should be able calculate total price in cart and payment page	System successfully calculates total price in cart and payment page	Pass
	To check if buyer can proceed payment and received receipt	Buyer should be able to proceed payment and received receipt	Buyer successfully proceeds payment and received receipt	Pass
Review submission	To check if buyer can submit novels review and rate	Buyer should be able to submit novels review and rate the novels	Buyer successfully submits the novels review and rate novels	Pass
	To check if the buyer's review appears in product details page	Buyer's review should be able to appears in product details page	Buyer's review successfully appears in product page	Pass
Report Management	To check if administrator can manage report in admin dashboard	Administrator should be able to manage report in admin dashboard	Administrator successfully manage report in admin dashboard	Pass

Based on the Table 7, the system was thoroughly tested with tested users and performed exceptionally well across all areas. Buyers, sellers, and administrators could easily create accounts, log in, and manage their profiles. Sellers successfully added new novels to the platform. Both buyers and sellers found it simple to search for, view, and categorize novels. Buyers had no trouble adding novels to their cart, seeing accurate price calculations, making payments, and receiving receipts. Additionally, buyers could effortlessly submit and see their reviews on the product pages. Administrators efficiently managed reports through the admin dashboard. Overall, the system met all expectations and passed every test

### 5. Conclusion

The project successfully accomplished its objectives by developing a web-based Novel's Marketplace system for novel enthusiasts. The system underwent testing and demonstrated its functional capabilities. The key advantages of this system are providing a single platform for buying, selling, and reviewing novels, which addresses the fragmented nature of current marketplace systems. Besides, the system enables administrators to generate detailed reports on sales, reviews, and other essential data for effective system oversight. Then, the system allows buyer and seller to create accounts and manage profiles and easy access to the website features. The system also allows for efficient management and listing of products by sellers.

However, there are limitations to the project. The payment gateway is currently in test mode and lacks necessary document verification, limiting its ability to handle actual financial transactions. Next, there is no notification system to inform seller about novel's purchased and buyer's order. Then, the system lacks automated order tracking features, providing only basic display information about orders without real-time updates or notifications. The registration and login processes also are not highly secure, as they do not include two-factor authentication or other advanced security features.

Future work for the project is improvements that can be made to address these limitations. It includes implementing advanced security features, such as two-factor authentication, to enhance the security of the registration and login processes. Additionally, developing automated order tracking features and a robust notification system will provide real-time updates and notifications to both buyers and sellers about order statuses and purchases. Upgrading the payment gateway from test mode to a fully functional system with necessary document verification.

In conclusion, the development of the Novel's Marketplace Website is an ambitious and innovative venture that seeks to revolutionize the way novels are bought, sold, and discussed online. Its success could significantly

influence the literary world, making novel access and discussion more cohesive, inclusive, and engaging for readers and sellers alike. As the project progresses, its impact on the digital literary landscape is eagerly anticipated, with the potential to foster a more connected and vibrant community of novel enthusiasts.

## Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

## Author Contribution

This journal requires that all authors take public responsibility for the content of the work submitted for review. The contributions of all authors must be described in the following manner:

*The authors confirm contribution to the paper as follows: **study conception and design:** Syed Muizzuddin Bin Syed Anuar, Rozlini Binti Mohamed; **data collection:** Syed Muizzuddin Bin Syed Anuar, Rozlini Binti Mohamed; **analysis and interpretation of results:** Syed Muizzuddin Bin Syed Anuar, Rozlini Binti Mohamed; **draft manuscript preparation:** Syed Muizzuddin Bin Syed Anuar, Rozlini Binti Mohamed. All authors reviewed the results and approved the final version of the manuscript.*

## References

- [1] Chevalier, Judith A. and Mayzlin, Dina (December 2003)., The Effect of Word of Mouth on Sales: Online Book Reviews. NBER Working Paper No. w10148, Available at SSRN: <https://ssrn.com/abstract=476105>
- [2] Prah, A. (2019, May 2). What is a novel? definition and characteristics. ThoughtCo. <https://www.thoughtco.com/what-is-a-novel-4685632>
- [3] Franc, V. (2019, March 20). What is a marketplace? our understanding of multi-seller businesses. Shopery. <https://www.shopery.com/insights/what-is-a-marketplace>
- [4] Jain, Vipin & Malviya, Bindoo & Arya, Satyendra. (2021). An Overview of Electronic Commerce (e-Commerce). Journal of Contemporary Issues in Business and Government. 27. 665-670. 10.47750/cibg.2021.27.03.090.
- [5] Wilkinson, J. (2022, August 15). 7 key components of Ecommerce. LinkedIn. <https://www.linkedin.com/pulse/7-key-components-ecommerce-josh-wilkinson>
- [6] Thiru. (2016.). *Thiru*. Webbased ecommerce architecture. www.myreadingroom.co.in. <https://www.myreadingroom.co.in/notes-and-studymaterial/66-e-commerce/521-webbased-ecommerce-architecture.html>
- [7] Kristi Hines (2022, June 9). 21 must-have features for Ecommerce sites. Search Engine Journal. <https://www.searchenginejournal.com/ecommerce-guide/must-have-website-features/>
- [8] Lewis, S. (2023, June 9). prototyping model. CIO. <https://www.techtarget.com/searchcio/definition/Prototyping-Model>
- [9] Nacheva, Radka. (2017). *prototyping approach in user interface development*. [https://www.researchgate.net/publication/317414969\\_PROTOTYPING\\_APPROACH\\_IN\\_USER\\_INTERFACE\\_DEVELOPMENT](https://www.researchgate.net/publication/317414969_PROTOTYPING_APPROACH_IN_USER_INTERFACE_DEVELOPMENT)
- [10] Grace, M. (2022, January 10). What is user interface design? And why is it important? <https://www.linkedin.com/pulse/what-user-interface-design-why-important-mathew-grace>