

AkarjayaMotor: The All-in-One Motorcycle Platform

Khairin Irfan Mohd Irwan¹, Mohd Zainuri Saringat^{1*}

¹ Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia (UTHM), Batu Pahat, 86400, MALAYSIA

*Corresponding Author: zainuri@uthm.edu.my

DOI: <https://doi.org/10.30880/aitcs.2026.07.01.044>

Article Info

Received: 15 June 2025

Accepted: 22 June 2026

Available online: 30 June 2026

Keywords

E-commerce platform, Secure payment integration, Service management, Role-based account

Abstract

This project aims to develop a comprehensive e-commerce platform for Akarjaya Motor Sdn Bhd, a leading motorcycle dealership in Parit Raja, Batu Pahat. The platform will support motorcycle sales, service bookings, and online payments, enhancing operational efficiency and customer convenience. Currently, Akarjaya Motor faces challenges due to its reliance on face-to-face transactions and lack of digital presence. To address this, the project adopts the Prototyping Model, enabling iterative development based on stakeholder feedback to ensure the system meets user needs. Key processes include requirement gathering, rapid prototyping, and continuous refinement. The final system will serve as the central hub for business operations, offering secure payment integration, a user-friendly interface, efficient service management, and role-based account features for administrators and customers. The platform is expected to significantly improve customer satisfaction and digital competitiveness, positioning Akarjaya Motor for future growth and strengthening its role as a key player in the motorcycle industry.

1. Introduction

Electronic commerce (e-commerce) is the use of online platforms to facilitate commercial transactions, connecting businesses and consumers in a seamless digital environment. E-commerce is the process of buying, selling or transferring products, services and information via digital platform [1]. The proposed system is an e-commerce platform that is intended to simplify the commercial operations of a motorcycle company. Digitization and innovative technologies are essential for growth and efficiency, facilitating their adaptation to market trends and enhancing operational effectiveness [2]. It enables users to purchase and sell motorcycles, schedule maintenance, and process payments online. The growth of e-commerce platforms has significantly impacted numerous industries, including the automotive sector which started in the 1960s with Electronic Data Interchange (EDI), has developed into a crucial element of the global economy and after that, platforms such as eBay and Amazon which developed in the 1990s, have transformed the way business works between businesses and consumers [3].

The objective of this system is to guarantee a seamless experience by offering users a dependable and intuitive interface. The objective of the solution is to enhance the organization's operational efficiency and satisfy the increasing needs of digital clients. Akarjaya Motor Sdn Bhd is a famous motorcycle dealer in the Parit Raja area, located at No. 1, Jalan Sri Indah, Taman Sri Indah, 86400 Parit Raja, Batu Pahat, Johor. Digital transformation, that involves the integration of digital technologies and processes to improve business operations and customer engagement, is essential for maintaining competitiveness in modern markets [4].

The company has been in operation since the 1980s and has a strong reputation for offering a diverse selection of motorcycles, including both high-end and budget-friendly models. Akarjaya Motor Sdn Bhd known as Akarjaya Motor is a motorcycle dealership that has established a significant presence in the city and

developed a devoted customer base by consistently delivering exceptional products and service. Akarjaya Motor continues applying traditional techniques for its operations, despite its strong reputation. These methods involve the use of paper-based systems like logbook for inventory processing, in-person interactions with clients, and manual inventory management. The company is unable to take advantage of the increasing popularity of online shopping and digital services due to the lack of a specialized web platform. This constraint results in inefficiencies in the administration of sales, services, and invoicing activities and impacts customer convenience. By reformulating critical business operations, the proposed e-commerce system addresses these challenges.

The development of this website will allow Akarjaya Motor to offer users the convenience of online browsing, purchasing, and service scheduling. Consequently, the system will improve the company's capacity to manage stocks, securely process payments, and provide real-time updates to customers, thereby resulting in increased customer satisfaction and operational efficiency.

2. Literature Review

2.1 E-Commerce System

E-commerce allows people to buy and sell goods and services online, offering features like product listings, secure payment options, and customer accounts. E-commerce is defined as business transactions being conducted online, with information technology as well as the global Internet network used for buying and selling many goods and services [5]. It allows companies to promote their products, engage with consumers, and handle transactions securely. E-commerce actively engages people in buying and selling products through multiple online platforms or by easily transmitting data over the Internet [6]. Akarjaya Motor needs an e-commerce system to streamline the purchase process for motorcycles, accessories, and parts. This system improves accessibility, customer satisfaction, and expands market access, lowering running expenses related to traditional retail.

2.2 Customer Relationship Management (CRM)

Customer Relationship Management (CRM) improves customer interactions, satisfaction, and loyalty by providing tools like contact management and sales tracking. CRM actively improves customer satisfaction, loyalty and profitability by effectively building, nurturing and sustaining strong relationships and interactions with customers and stakeholders [7]. Companies like Akarjaya Motor use CRM to enhance customer retention, satisfaction, and loyalty. CRM is a technology for business management that strengthens customer relationships as well as increases return on investment [8]. By implementing CRM, the platform attracts new customers, retains existing ones, and enhances overall engagement and profitability.

2.3 Inventory Management

Inventory management systems help companies maintain optimal stock levels, reduce storage costs, and prevent overstocking and out of stock issues. Organizational efficiency in inventory management comes from using an inventory management system (IMS), a software tool that helps oversee order placement, product reception, inventory maintenance, product tracking and sales [9]. It offers features like real-time stock tracking, automatic reordering, and reporting on turnover rates. Akarjaya Motor uses an inventory management system to track motorcycles, parts, and accessories, ensuring accurate stock information and reducing stockouts. IMS helps stores to accurately maintain their sales as well as purchase records, importantly reducing manual labor, human mistakes, along with delays, while rapidly speeding up the overall process [10]. This system improves sales efficiency and customer satisfaction by ensuring easy access to products and reducing stockouts.

2.4 Service Scheduling Management

The service scheduling system will allow customers to book maintenance appointments online, improving efficiency and customer satisfaction. Web-based booking platforms allow customers to easily schedule as well as pay for services or activities online, importantly easing staff workload and effectively preventing scheduling conflicts [11]. Many online booking platforms for vehicle services efficiently arrange booking data and provide multiple service-related information to many customers [12]. This will also reduce waiting times and minimize scheduling conflicts, ultimately improving functional efficiency.

2.5 Payment and Billing System

Payment and billing system ensures secure online transactions for products, services, and subscriptions. An online payment system allows users to transfer money electronically over the internet, computer networks and digital value storage systems [13]. It includes secure gateways, automatic invoicing, transaction records,

multiple payment methods, fraud detection, data encryption, and functionalities for recurring payments or invoicing. This eases smooth payments through web or other internet connections [14]. The system provides detailed billing records, allowing for close monitoring of customer transactions, financial management, and increased transparency.

2.6 Comparison with Existing System

The AkarjayaMotor: The All-in-One Motorcycle Platform actively matches some functions of existing systems, which offer important information. Table 1 provides a thorough comparison of required modules for the three existing system MotoMalaysia.com [15], WMoto [16] and iMotorbike [17], as well as the proposed system AkarjayaMotor.

Table 1 System Comparison

Features / System	MotoMalaysia [17]	WMoto [18]	iMotorbike [19]	AkarjayaMotor
Register and Login	No	Yes	Yes	Yes
Manage Item	Yes	Yes	Yes	Yes
Purchase Item	Yes	Yes	Yes	Yes
Manage Booking	No	Yes	No	Yes
Booking Service	No	Yes	No	Yes
Frequently Asked Question (FAQ)	Yes	Yes	Yes	Yes
Manage Payment and Billing	No	No	Yes	Yes
Generate Report	No	No	Yes	Yes

These systems enhance user experience and streamline transactions by focusing on key features like comprehensive motorcycle listings and service scheduling tools. AkarjayaMotor integrates functional modules that boost efficiency and financial transparency, including secure registration and login, payment integration, invoice management, and detailed report generation. With user-friendly filters and comparison tools, the platform simplifies navigation and decision-making for customers. Its service module allows easy scheduling of maintenance appointments, reinforcing customer convenience. By adopting best practices from modern platforms, AkarjayaMotor positions itself as a complete, customer-focused solution for motorcycle sales and services, promoting loyalty and competitiveness. Unlike platforms such as MotoMalaysia, WMoto, and iMotorbike, AkarjayaMotor delivers a broader feature set. WMoto focuses on promoting its own models, while MotoMalaysia and iMotorbike prioritize listings and information sharing. AkarjayaMotor stands out with secure multi-role access, advanced invoicing and payment management also robust reporting capabilities.

3. Methodology

The selection of the Prototyping Model for the AkarjayaMotor resulted from its adaptability along with its focus on user feedback. Prototyping is more appropriate for customised systems or software, distinguishing it from the waterfall model in the Software Development Life Cycle (SDLC) [18]. Complicated functionalities, including secure payment systems, as well as motorcycle service scheduling, are involved in this truly meaningful project. An intuitive user interface, along with smooth interaction, is necessary for customers, administrators, as well as staff to use these features. Stakeholders readily provide important input during Prototyping Model development. This iterative process truly guarantees the platform completely meets their needs besides improves user experience. This model's flexibility changes to changing requirements, making it perfect for this project's dynamic, user-concentrated environment. The Prototyping Model, with its iterative cycle along with its core phases, is depicted in Figure 1.

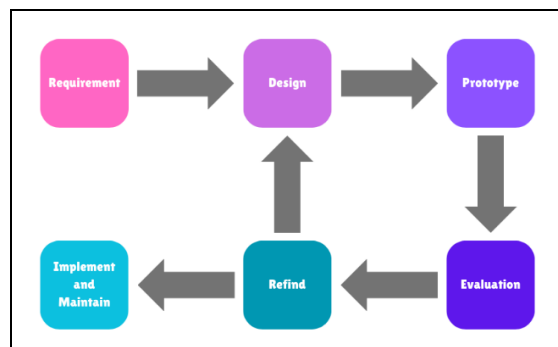


Fig. 1 Prototyping Model Phases

3.1 System Development Workflow

The system development workflow for the AkarjayaMotor adheres to the Prototyping Model. This structured workflow ensures that all development activities align with project objectives and are completed within the allocated timeline. Each phase is defined by specific tasks and deliverables, as shown in Table 2.

Table 2 *System development workflow*

Phase	Task	Output
Requirement Gathering	(i). Identify stakeholder and collect requirements through interviews, surveys and observation.	Requirement specification document.
	(ii). Prioritize system functionalities and document requirements.	
Rapid Design	(i). Create wireframes and mock-ups for the module's interfaces.	Wireframe designs and workflows.
	(ii). Define workflows for the system.	
	(iii). Obtain feedback from the stakeholders.	
Prototype Development	(i). Develop a functional prototype for all the modules such as home page, trade-in, payment systems and others.	Initial working prototype.
	(ii). Integrate basic UI and system functionality.	
User Evaluation	(i). Present the prototype to stakeholders for user acceptance testing and evaluation.	Stakeholder evaluation report.
	(ii). Gather feedback through observations, surveys and interviews.	
Prototype Refinement	(i). Analyse stakeholder feedback and do a change if needed.	Updated and refined prototype.
	(ii). Implement improvements and fix identified issues.	
	(iii). Conduct further testing and re-evaluation process until completely succeed.	
Implementation and Maintenance	(i). Deploy the finalize system into a real-world environment.	Fully operational system with ongoing updates.
	(ii). Monitor system performance and address ongoing issues.	

4. Analysis and Design

4.1 Functional Requirements Analysis

Functional requirements are functions of a system that must be implemented to enable the users to fulfil their tasks or business process. Table 3 shows the functional requirement analysis for each module of the AkarjayaMotor: The All-in-One Motorcycle Platform.

Table 3 *System Functional Requirements*

Modules	Functionalities
Register and Login	(i). Admin manages account registrations and logins.
	(ii). Customers securely log in and access the platform.
Manage Item	(i). Admin performs CRUD operations on item listings.
Purchase Item	(i). Customers browse and purchase items.

Table 3 Continued

Manage Booking	(i).	Admin manages service schedules by adding, updating or deleting service options.
Booking Services	(i).	Customer schedule and view service appointments through the platform.
Frequently Asked Question (FAQ)	(i).	Admin updates FAQs to address common queries.
	(ii).	Customers access FAQs for quick information.
Manage Payment and Billing	(i).	Admin tracks transactions and billing records.
	(ii).	Customers make payment securely and view history.
Implementation and Maintenance	(i).	Admin generates reports on sales, services and finances.

4.2 Non- Functional Requirements Analysis

Non-functional requirements, also known as quality requirements, describe how a system must behave and its quality attribute. Table 4 shows the non-functional requirement of the AkarjayaMotor: The All-in-One Motorcycle Platform.

Table 4 System Non-Functional Requirements

Modules	Functionalities
Security	(i). Passwords must include a combination of at least 8 characters (letters and numbers). (ii). Access is secured by username-password authentication.
Operational	(i). The system should function across all web browsers. (ii). Customers perform tasks such as login and transactions seamlessly.
Performance	(i). System interactions including page load times must not exceed 3 seconds.

4.3 Use Case Diagram

The use case diagram for AkarjayaMotor: The All-in-One Motorcycle Platform illustrates the interaction between two primary actors, Admin and Customer with the system's core functional modules. Customers can register and log in, purchase motorcycles, book service appointments, view FAQs, and manage their payments and billing. Admin users, on the other hand, have extended privileges including managing items, handling bookings, updating FAQs, overseeing payment and billing records, and generating system reports. Both Admin and Customer interact with the Register and Login modules, promoting accessibility and user engagement. Figure 2 effectively outlines the scope of the system's functionalities and clarifies the responsibilities and access levels of each user role, ensuring organized system development and user experience.

**Fig. 2 Use Case Diagram**

4.4 General System Architecture

The system architecture is built using the Model-View-Controller (MVC) framework to promote modular design and improve interaction between its parts. Figure 3 shows the Model layer, includes data entities such as motorcycle inventory, customer profiles, service appointments, transaction records, as well as reports representing the system's core data structure. The Controller layer works like a go-between. It manages the logic and operations for many modules. It makes sure the data is consistent, and it handles all user requests. The View layer gives a user-friendly interface. This includes dashboards for inventory, service bookings and billing.

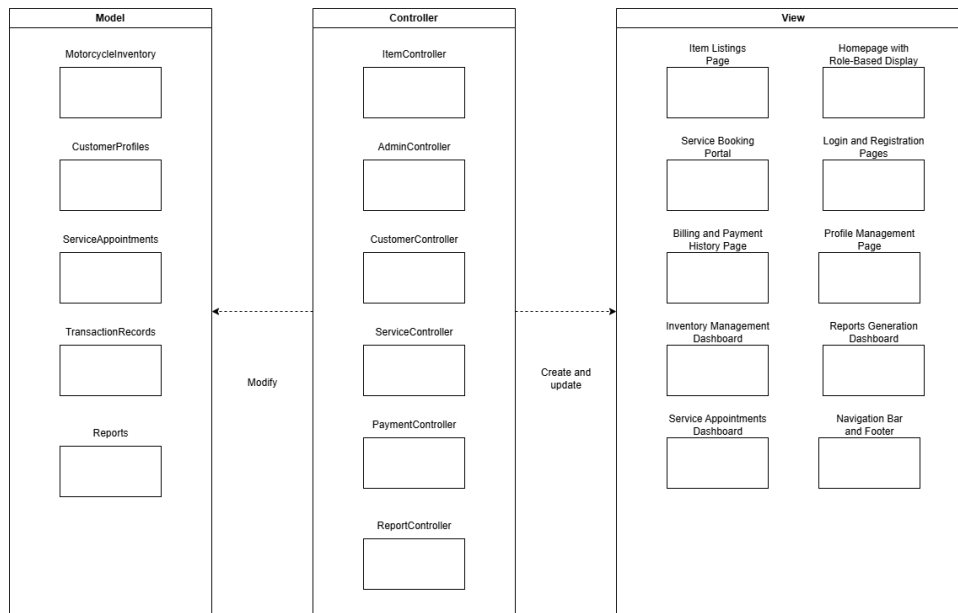


Fig. 3 General System Architecture Diagram

4.5 Class Diagram

Figure 4 shows a class diagram for AkarjayaMotor: The All-in-One Motorcycle Platform presents a well-structured data model that highlights the relationships between key system entities. At the core is the User class, which connects with various modules such as Purchase, Payment, Appointment, Report, and Faq, reflecting user interactions across the platform. The Purchase class links users with items, while Payment manages transactions tied to users and purchases. The Item class contains product details such as name, price, stock, and category. The Service and Appointment classes support the motorcycle service booking function, allowing users to schedule appointments for different services. The Report class captures system-generated insights, and the Faq class handles frequently asked questions, both managed by the admin. Additional classes like Session and PasswordResetToken manage system authentication and security. Overall, the class diagram illustrates clear entity relationships and supports CRUD operations vital for the platform's e-commerce and service functionalities.

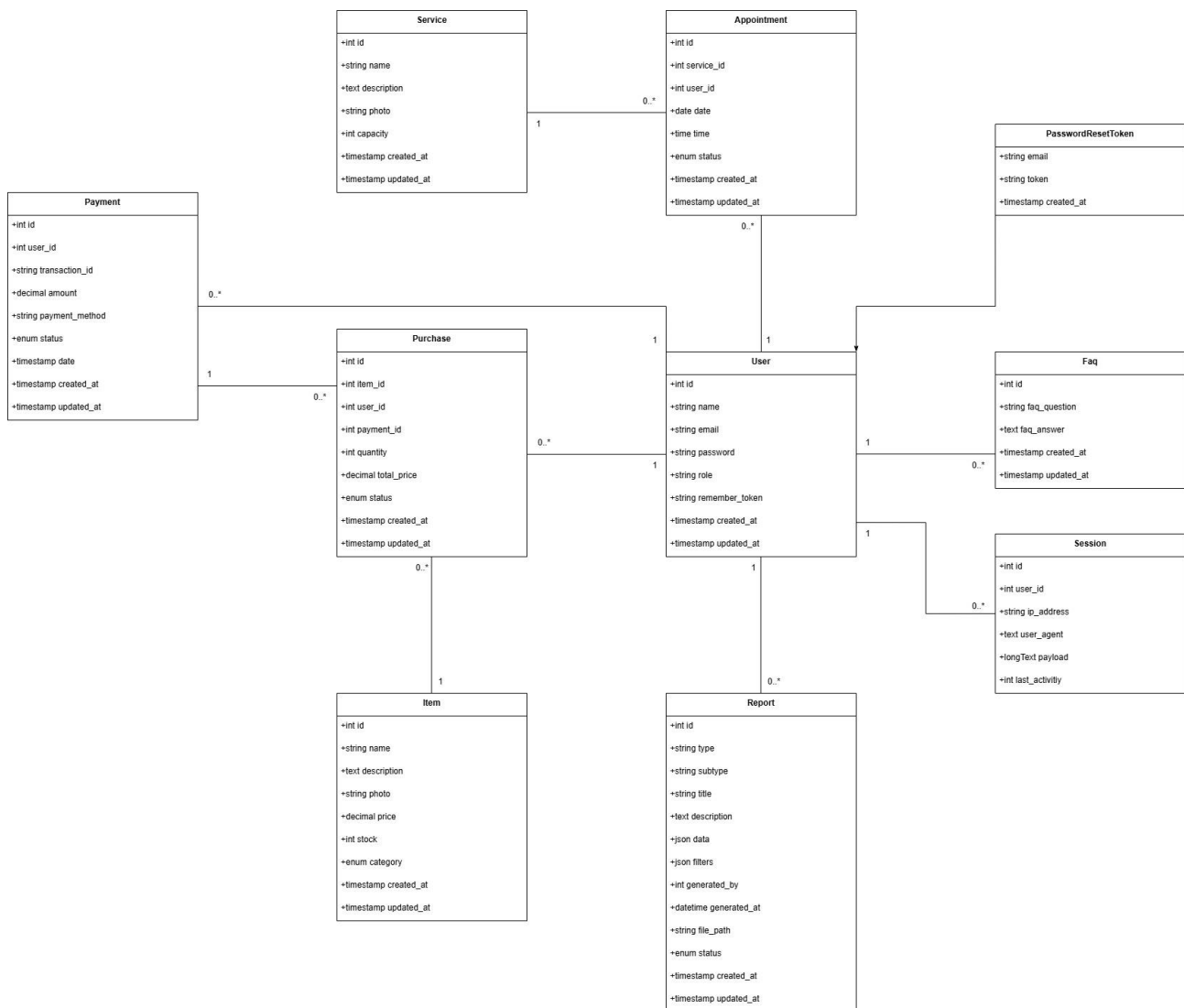


Fig. 4 Class Diagram

4.6 Interface Design

The interface designs for AkarjayaMotor include two main parts, a login page and a sign-up page for both admin and customer. Figure 5 (a) the log in page has spaces for the username and password. It also has choices for password recovery and account creation. Figure 5 (b) the registration page offers input fields for a username, email address, a password, as well as password confirmation, thus providing an outstandingly straightforward process for new user registration. Clear and prominent headings and readily apparent action buttons are present, guaranteeing effortless use and accessibility for every user.

(a)

(b)

Fig. 5 (a) Register Interface; (b) Login Interface

The design of AkarjayaMotor's e-commerce platform is easy to use and has clear features for customers. Figure 6 (a) the homepage shows available services, promotions, as well as company information. Figure 6 (b) the "Purchase Item" section offers a thorough catalogue of motorcycle options, including outstandingly detailed pricing information, while Figure 6 (c) the "Booking Service" module allows for convenient appointment bookings and progress tracking.

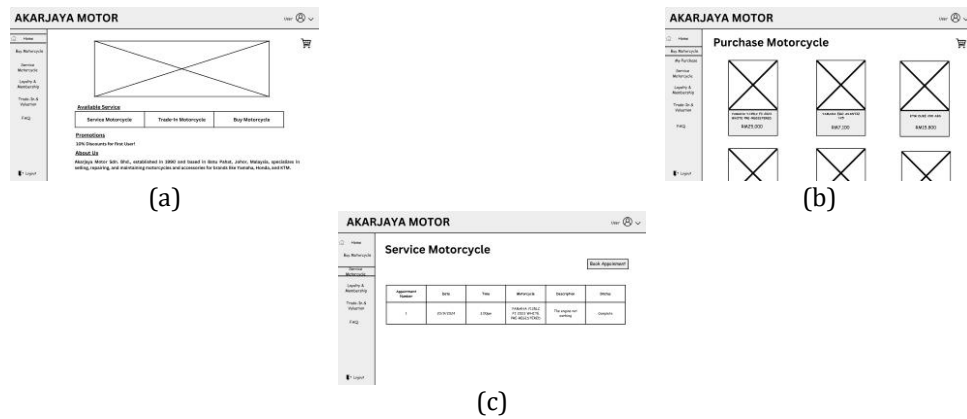


Fig. 6 (a) Home Page Interface; (b) Purchase Item Interface; (c) Booking Service Interface

The admin interfaces for AkarjayaMotor are designed to manage important parts of the platform effectively. Figure 7 (a) the "Manage Homepage" module lets administrators customize banners, services, along with promotions. Figure 7 (b) the "Manage Item" interface allows users to add, edit, or delete motorcycle listings. The interface provides instantaneous updates on sales and inventory levels. Figure 7 (c) the "Manage Booking" section lets administrators handle appointment requests and mark them as finished. Figure 7 (d) the "Generate Report" modules offer the ability to create several detailed reports about sales and performance, making sure that administrative tasks are easy to complete.

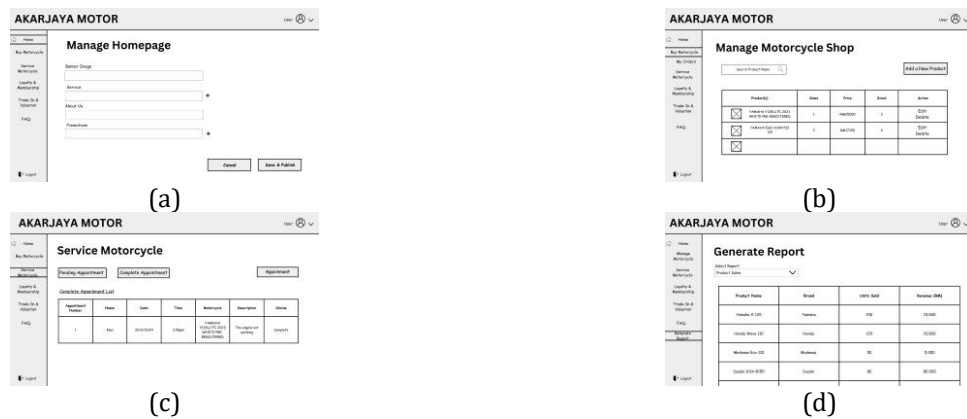


Fig. 7 (a) Manage Homepage Interface; (b) Manage Motorcycle Shop Interface; (c) Service Motorcycle Interface; (d) Generate Report Interface

5. Implementation and Testing

5.1 Functional Module Development

Each module was designed to handle core operations like authentication, inventory, bookings, payments, and user communication. Backend logic was managed by dedicated controllers, while the views focused on usability. Validations and secure data handling ensured smooth, reliable performance.

5.1.1 Register and Login Module

The Register and Login module is responsible for handling user account creation, login, and logout processes. Figure 8 (a) shows the RegisteredUserController provides functionality for new users to register by validating input such as name, email, and password, ensuring password strength, and then creating a new user record with encrypted credentials. After successful registration, users are automatically logged in and redirected to the dashboard. Meanwhile, Figure 8 (b) shows the AuthenticatedSessionController handles the login and logout processes. It validates login credentials using a custom LoginRequest, ensures session regeneration for security, and supports proper logout by invalidating and regenerating the session token, redirecting users back to the homepage. Together, these controllers enforce secure access control for all platform users. The user interfaces for this module include the registration form, login form, and logout function embedded within the navigation bar or dashboard. The registration view `auth.register` provides a clean and simple form for new users to sign up, collecting essential details like name, email, and password. The login view `auth.login` offers an intuitive interface with fields for email and password input, along with validation feedback for incorrect entries. Upon successful

login, users are directed to the dashboard, and a logout option is readily available to ensure secure session termination.

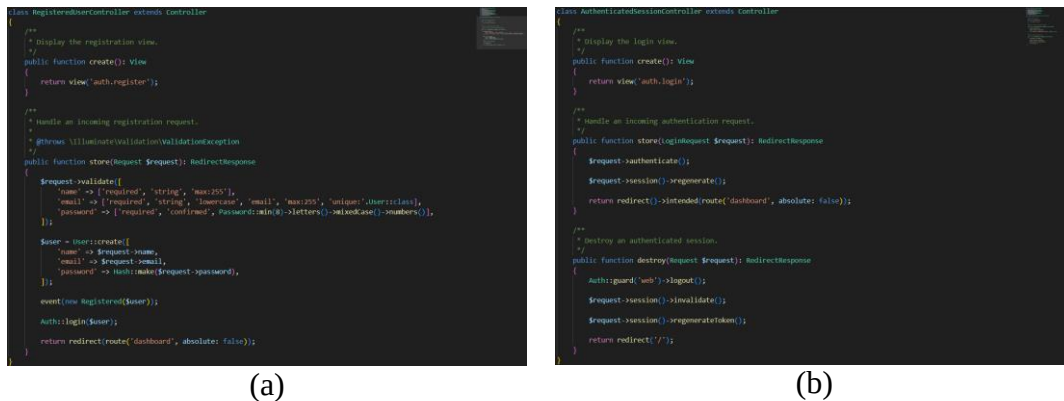


Fig. 8 (a) RegisteredUserController; (b) AuthenticatedSessionController

The user interface for the Register and Login module consists of two clean and minimalistic forms designed for user onboarding and access. Figure 9 (a) shows the registration interface features the Akarjaya Motor logo prominently at the top, followed by input fields for name, email, password, and password confirmation, enabling users to easily create an account. A link is also provided for existing users to navigate to the login page. Figure 9 (b) shows the login interface follows a similar visual structure, focusing on email and password fields, with an optional “Remember me” checkbox and a navigation link to the registration page. Both interfaces maintain a professional and consistent brand identity while ensuring usability, responsiveness, and clarity.

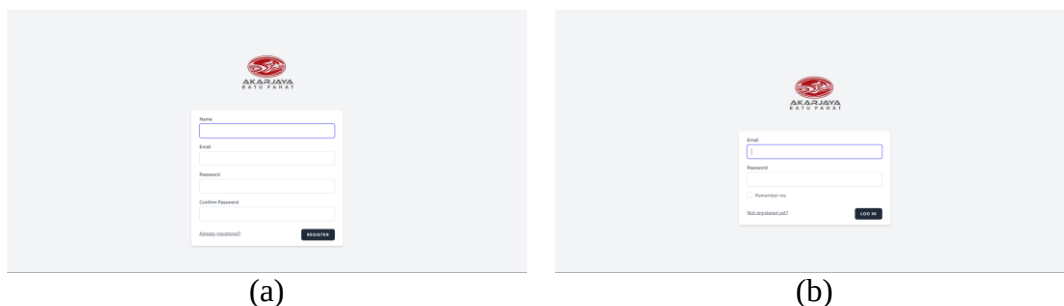


Fig. 9 (a) Register User; (b) Login User Interface

5.1.2 Manage Item Module

The Manage Item module enables administrators to perform full CRUD operations on inventory items such as motorcycles, accessories, and brands. Figure 10 shows the ItemController handles functionalities including listing, creating, viewing, editing, updating, and deleting items. It supports searching and filtering by category and includes customizable sorting by attributes like name, price, and stock level. The store and update methods also manage image uploads, saving item photos securely within the public/items directory. The module enforces validation to ensure data consistency, such as requiring unique item names, valid image formats, and non-negative prices and stock levels. Overall, this controller plays a central role in maintaining accurate and organized item listings within the platform.

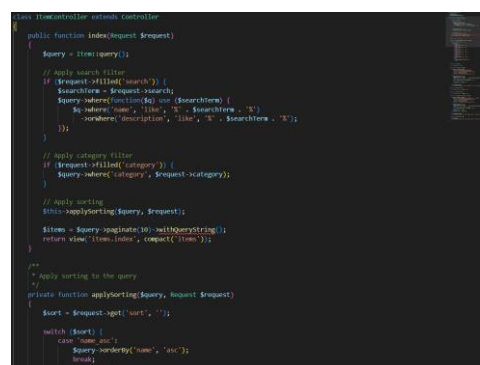


Fig. 10 ItemController

Figure 11 (a) shows the Inventory Management (index) page displays a clear dashboard of total stock statistics, along with a searchable and filterable table of items including details such as name, category, price, stock, and actions for edit or delete. Figure 11 (b) shows the Add New Item interface provides a structured form for admins to input essential item details such as name, category, price, stock quantity, and description, all within an intuitive card layout. Figure 11 (c) shows the Edit Item interface mirrors the create form but is pre-filled with the selected item's data, allowing administrators to easily update existing records.

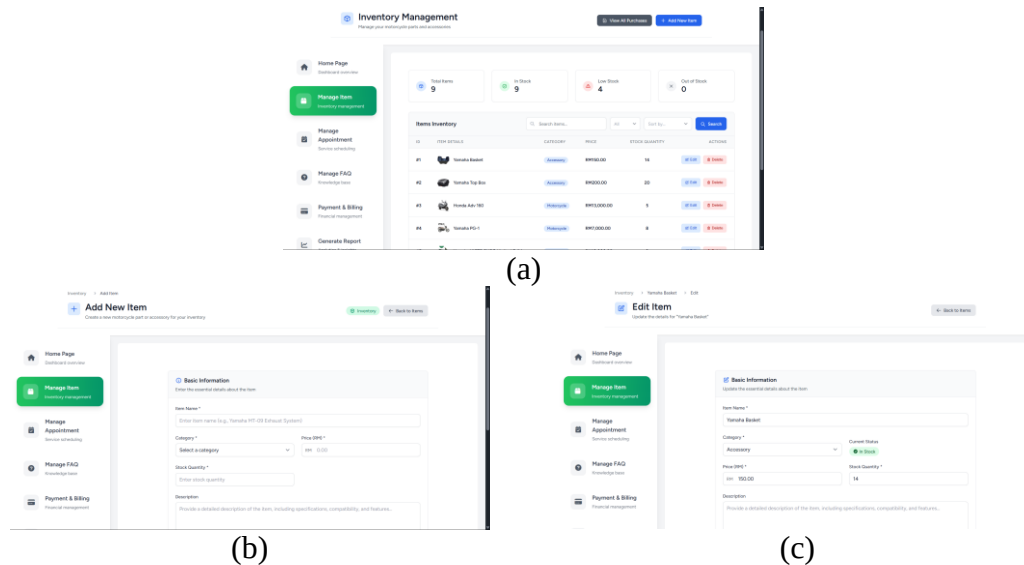


Fig. 11 (a) Inventory Management; (b) Add New Item; (c) Edit Item Interface for Admin

5.1.3 Purchase Item Module

The Purchase Item module allows users to browse, filter, and purchase available items, while also enabling administrators to monitor and manage purchase activities. Figure 12 shows the PurchaseController supports several core features: users can add items to their cart (pending purchases), apply various filters (category, price, stock), and sort item listings before purchasing. Upon purchase, item stock is deducted, and the transaction is stored with a status of "pending". Users can update or delete pending purchases, which will adjust stock accordingly. Additionally, a separate method provides access to completed purchases for user purchase history tracking. For administrators, the module offers a detailed view with advanced filtering by date, status, user, item name, and allows updating purchase statuses from pending to completed or canceled. The controller also includes AJAX-based item fetching and enhanced search, ensuring scalability and responsiveness. This comprehensive setup supports both user shopping experiences and administrative oversight effectively.

```

class PurchaseController extends Controller
{
    /**
     * Display a listing of the resource.
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $purchases = Purchase::with('item')
            ->where('user_id', auth()->id())
            ->where('status', 'pending') // only pending purchases
            ->paginate(10);

        return view('purchases.index', compact('purchases'));
    }

    /**
     * Show the form for creating a new resource with filtering and sorting.
     */
    public function create(Request $request)
    {
        $query = Item::query();

        // Apply filters
        $this->applyFilters($query, $request);

        // Apply sorting
        $this->applySorting($query, $request);

        // Get paginated results
        $items = $query->paginate(12)->withQueryString();

        return view('purchases.create', compact('items'));
    }

    /**
     * Apply filters to the query
     */
    private function applyFilters($query, Request $request)
    {

```

Fig. 12 PurchaseController

The user interfaces for the Purchase Item module encompass multiple key screens tailored for both customers and administrators. Figure 13 (a) shows the customer purchase page displays pending cart items in card format with product images, pricing details, and action buttons for canceling or proceeding to checkout. Figure 13 (b) shows the browse items page allows users to filter and sort products by category, price, or availability, enhancing the shopping experience. Completed transactions are shown in Figure 13 (c) shows the Purchase History interface, listing previously bought items with quantity, total price, and completion dates. For administrative users, Figure 13 (d) shows the Purchase Management Dashboard offers a high-level overview of

order statistics, including total revenue, order status distribution, and a visual sales graph. These interfaces collectively offer a seamless and interactive experience for item selection, transaction tracking, and order oversight within the platform.

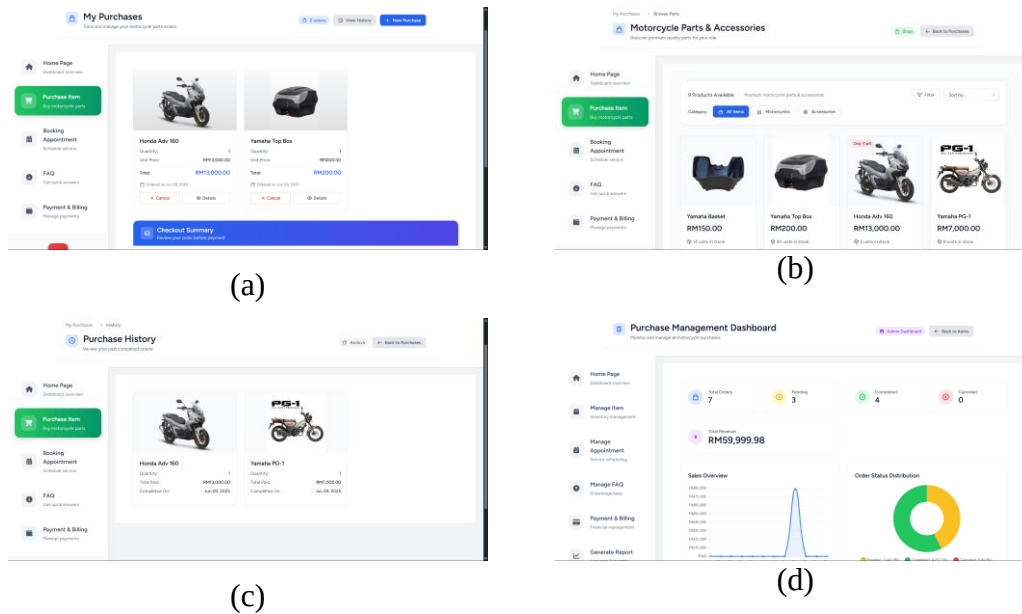


Fig. 13 (a) My Purchases; (b) Motorcycle Part and Accessories; (c) Purchase History Interface; (d) Purchase Management Dashboard Interface for Admin

5.1.4 Manage Booking Module

The Manage Booking module, specifically handled through Figure 14 shows the ServiceController, allows administrators to manage motorcycle service offerings by performing full CRUD operations. Services can be listed, created, updated, and deleted with support for pagination. When creating or editing a service, administrators must input a name, optional description, and a capacity value indicating how many bookings a service slot can handle. Optional image uploads are handled securely, storing photos in the public/services directory for easy referencing. The controller ensures input validation, including checks for unique names, minimum capacity values, and valid image formats. This setup forms the foundation for booking functionalities by defining the types of services customers can schedule with the platform.

```

use Illuminate\Http\Request;
use App\Models\Service;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Redirect;
use Illuminate\Support\Facades\Response;
use Illuminate\Support\Facades\Storage;

class ServiceController extends Controller
{
    /**
     * Display a listing of the services.
     */
    public function index()
    {
        $services = Service::paginate(10);
        return view('services.index', compact('services'));
    }

    /**
     * Show the form for creating a new service.
     */
    public function create()
    {
        return view('services.create');
    }

    /**
     * Store a newly created service in storage.
     */
    public function store(Request $request)
    {
        $validated = $request->validate([
            'name' => 'required|string|max:255|unique:services',
            'description' => 'nullable|string|max:255',
            'photo' => 'nullable|image|mimes:jpg,jpeg,png|max:2048',
            'capacity' => 'required|integer|min:1',
        ]);

        if ($request->hasFile('photo')) {
            $path = $request->file('photo')->store('services', 'public');
            $validated['photo'] = $path;
        }

        $service = Service::create($validated);
        return redirect('services.index')->with('message', 'Service created successfully.');
```

Fig. 14 ServiceController

The user interfaces for the Manage Booking module include three essential views designed for administrative control over service scheduling. Figure 15 (a) shows the Service Management page provides an overview of all registered services in a tabular layout, displaying service names, photos, capacity, and quick-action buttons for editing or deleting entries. Figure 15 (b) shows the Create New Service interface offers a form layout where administrators can enter the service name, description, photo, and booking capacity per day, making it easy to onboard new maintenance offerings. Figure 15 (c) shows the Edit Service view mirrors the creation form but is pre-populated with existing service data, allowing for seamless updates.

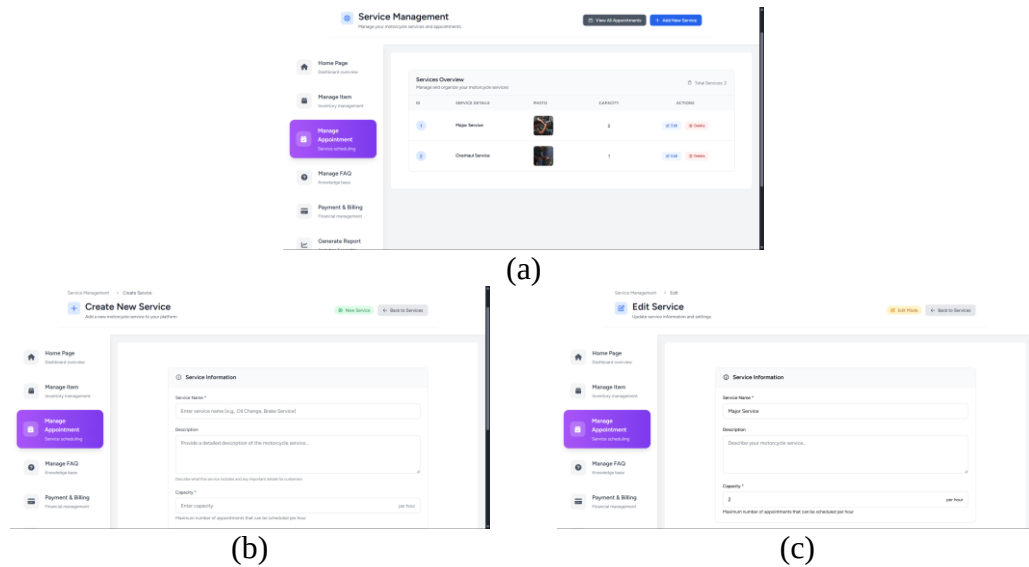


Fig. 15 (a) Service Management; (b) Create New Service; (c) Edit Service Interface for Admin

5.1.5 Booking Service Module

The Booking Service module, managed via Figure 16 shows the AppointmentController, facilitates appointment scheduling for motorcycle services by customers, while also offering robust tools for administrative oversight. Users can create, view, edit, or cancel their appointments, with validation ensuring that selected time slots do not exceed booking limits. Appointments are linked to specific services and users, and each booking records the service type, date, time, and booking status. For administrators, the module provides advanced filtering, real-time dashboard statistics (total bookings, status breakdown), and historical trends through weekly and monthly data visualizations. Admins can update appointment statuses to approved, rejected or pending. This module ensures structured, conflict-free service scheduling while equipping the admin team with analytical insights and control.

```

class AppointmentController extends Controller
{
    /**
     * Display a listing of appointments for the user.
     */
    public function index(Request $request)
    {
        $appointments = Appointment::with('service')
            ->where('user_id', auth()->id())
            ->paginate(10);

        return view('appointments.index', compact('appointments'));
    }

    /**
     * Display the specified appointment (not used, but added to avoid errors).
     */
    public function show(Appointment $appointment)
    {
        abort(404, 'This page is not implemented.');
```

Fig. 16 AppointmentController

The user interfaces for the Booking Service module consist of four main views that cater to both customer and admin users. Figure 17 (a) shows the Service Appointments page provides users with a clear history of their bookings, showing service names, scheduled dates, statuses, and quick actions to edit or cancel appointments. Figure 17 (b) shows the Book Service Appointment form offers a user-friendly interface for selecting a service, preferred date, and time, with helpful prompts and visual highlights explaining the benefits of booking. For administrators, Figure 17 (d) shows the Booking Management Dashboard presents a comprehensive overview of current bookings, categorized by status (approved, pending, rejected), alongside a pie chart and a weekly trends graph to visualize performance over time.

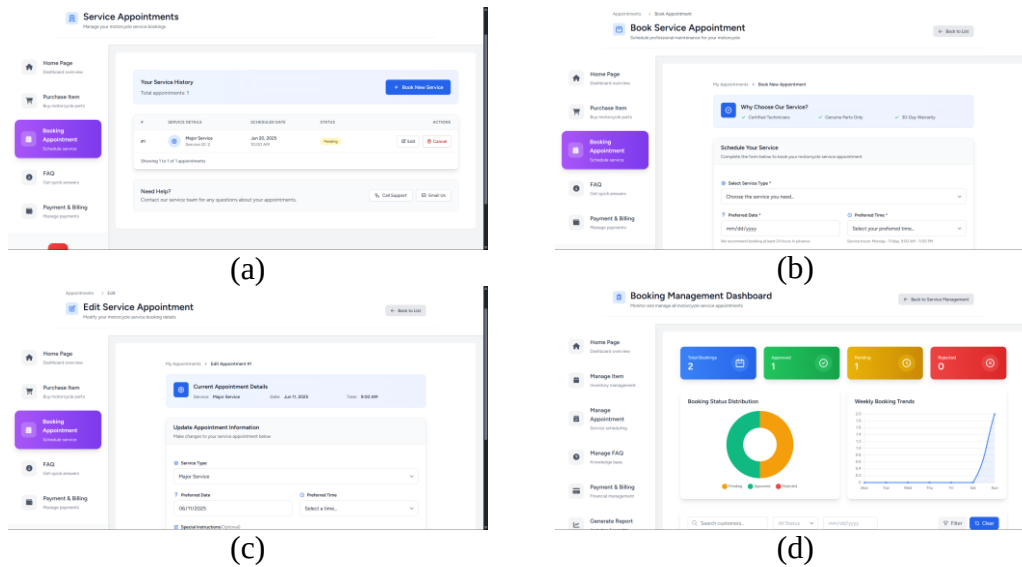


Fig. 17 (a) Service Appointment; (b) Book Service Appointment; (c) Edit Service Appointment; (d) Booking Management Dashboard Interface for Admin

5.1.6 Frequently Asked Question Module

The FAQ module, managed through Figure 18 shows the `FaqController`, allows administrators to efficiently create, view, edit, and delete frequently asked questions related to AkarjayaMotor's services. The controller facilitates full CRUD operations, with validation to ensure that each FAQ entry includes a clear question and a well-defined answer. FAQs are stored in the database and presented to users in a concise and accessible format.

```

1 use Illuminate\Http\Request;
2 use Illuminate\Support\Facades\Redirect;
3 use Illuminate\Support\Facades\View;
4 use App\Models\Faq;
5 use App\Http\Controllers\Controller;
6
7 class FaqController extends Controller
8 {
9     /**
10      * Display a listing of the resource.
11      */
12     public function index()
13     {
14         $faqs = Faq::all(); // use plural for clarity
15         return view('faq.index', compact('faqs'));
16     }
17
18     /**
19      * Show the form for creating a new resource.
20      */
21     public function create()
22     {
23         return view('faq.create');
24     }
25
26     /**
27      * Store a newly created resource in storage.
28      */
29     public function store(Request $request)
30     {
31         $validatedData = $request->validate([
32             'faq_question' => 'required|string|max:255',
33             'faq_answer' => 'required|string|max:500',
34         ]);
35
36         $faq = Faq::create($validatedData);
37
38         return redirect()->route('faq.index')->with('message', 'FAQ created successfully!');
39     }
40
41     /**
42      * Show the form for editing the specified resource.
43      */
44     public function edit($faq)
45     {
46         //
47     }
48 }

```

Fig. 18 `FaqController`

The user interfaces for the Frequently Asked Questions (FAQ) module include a streamlined design that helps administrators manage and present key information efficiently. Figure 19 (a) shows the FAQ index view displays a collapsible list of common questions with expandable answers, along with options to edit or delete each entry. Figure 19 (b) shows the Create New FAQ interface offers a form with labeled fields for entering a new question and its answer, supported by sidebar tips for writing clear and helpful responses. Figure 19 (c) shows the Edit FAQ view allows for easy modification of existing entries with pre-filled content and simple action buttons. These interfaces collectively promote a user-friendly knowledge base that enhances customer self-service while maintaining a professional and organized layout.

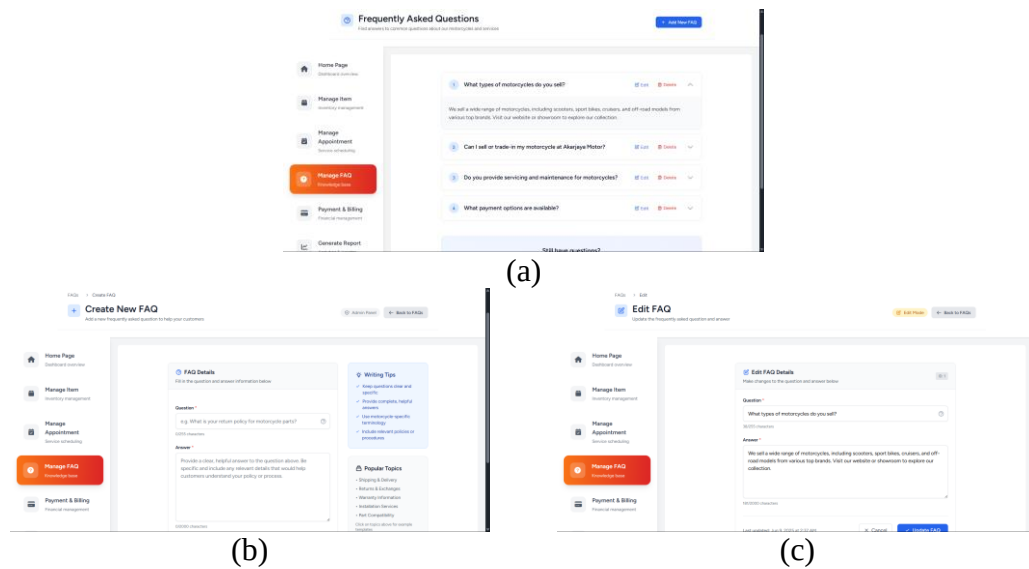


Fig. 19 (a) Frequently Asked Question; (b) Create New FAQ; (c) Edit FAQ Interface for Admin

5.1.7 Manage Payment and Billing Module

The Manage Payment and Billing module handles the entire transaction flow for purchases, from initiating payments to generating invoices. Integrated with Stripe, Figure 20 shows the PaymentController initiates secure checkout sessions for users with pending purchases and records successful transactions. Upon confirmation, purchases are updated to “completed” and linked to the corresponding payment record. Admin and users can view payment histories, and detailed invoices can be generated and downloaded as PDFs. The module ensures that only authorized users can view or download invoices, reinforcing data security. This module streamlines financial processing and enhances transparency in transaction handling for both customers and administrators.

```
class PaymentController extends Controller {
  //**
  // Display a listing of the payments.
  //
  public function index() {
    $user = Auth::user();

    $payments = $user->role == 'admin'
      ? Payment::where('user_id', $user->id)->paginate(10)
      : Payment::where('user_id', $user->id)->paginate(10);

    return view('payment.index', compact('payments'));
  }

  public function create() {
    return view('payment.create');
  }

  public function store(Request $request) {
    // not used
  }

  public function show(Payment $payment) {
    return view('payment.show', compact('payment'));
  }

  public function edit(Payment $payment) {
    return view('payment.edit', compact('payment'));
  }

  public function update(Request $request, Payment $payment) {
    // not used
  }
}
```

Fig. 20 *PaymentController*

The user interfaces for the Manage Payment and Billing module provide a seamless and professional transaction experience for users and administrators. Figure 21 (a) shows the Payment History page presents a clear summary of completed transactions, including total paid, payment status, and a detailed list of past payments with access to invoices. Figure 21 (d) shows the Stripe Checkout Page offers a secure and user-friendly form for entering card details and completing payment, fully integrated with Stripe's trusted platform. After a successful payment, users are directed to Figure 21 (b) shows the Payment Invoice view, which displays an itemized breakdown of purchases, payment details, and customer information. Figure 21 (c) shows a downloadable PDF invoice version is also available, styled professionally for record-keeping or printing.

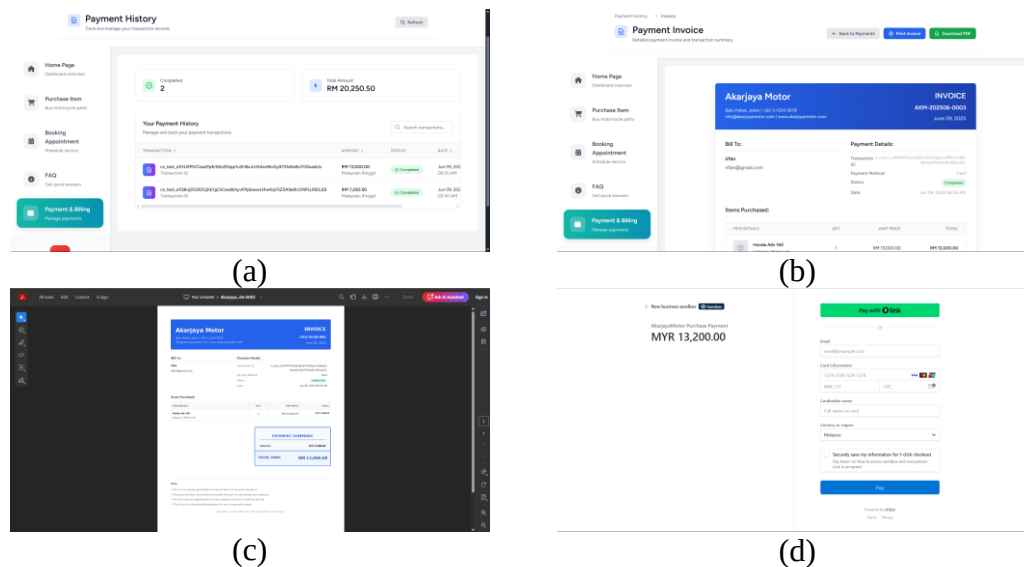


Fig. 21 (a) Payment History; (b) Payment Invoice; (c) PDF for Invoice Interface for Admin and Customer; (d) Stripe Payment Gateway Interface for Customer

5.1.8 Generate Report Module

The Generate Report module provides administrators with a powerful tool for creating analytical reports related to purchases and service appointments. Through Figure 22 shows the ReportController, users can filter data by date, status, customer, and subtype for example, sales summary and booking trends, and then generate professional PDF reports summarizing the selected metrics. This module supports dynamic data aggregation, such as top customers, most booked services, and revenue statistics. It logs generation activities, handles exceptions, and stores the final PDF securely for later access or download. The reports are presented with descriptive titles and summaries, making them suitable for internal performance analysis or managerial decision-making. Overall, this module enhances administrative capabilities with automated insights and streamlined reporting.

```

class ReportController extends Controller
{
    /**
     * Display the report generation interface
     */
    public function index()
    {
        $recentReports = Report::with('generatordata')
            ->orderBy('created_at', 'desc')
            ->take(10)
            ->get();

        $reportTypes = Report::getReportTypes();

        return view('report.index', compact('recentReports', 'reportTypes'));
    }

    /**
     * Generate a new report
     */
    public function store(Request $request)
    {
        $validated = $request->validate([
            'type' => 'required|string|in:purchases,appointments',
            'subtype' => 'required|string',
            'date_from' => 'nullable|date',
            'date_to' => 'nullable|date|after_or_equal:date_from',
            'status' => 'nullable|string',
            'customer_id' => 'nullable|exists:customers,id',
        ]);

        // Generate report data first
        $reportData = $this->generateReportData($validated);

        // Create report record
        $report = Report::create([
            'type' => $validated['type'],
            'subtype' => $validated['subtype'],
        ]);
    }
}

```

Fig. 22 ReportController

Figure 23 (a) shows the Report Generation Center offers a user-friendly form to select report types between purchase or appointment, specify filter criteria such as date range, status, or customer, and generate reports instantly. On the right, recent reports are listed for quick access, along with helpful statistics on total reports generated. Once a report is created, Figure 23 (b) shows the Report Details View presents a detailed breakdown of metrics including order status, revenue, top customers, and top items, with a download option for the PDF file. Figure 23 (c) shows the PDF report view mirrors this data in a clean, printable format ideal for documentation and analysis.

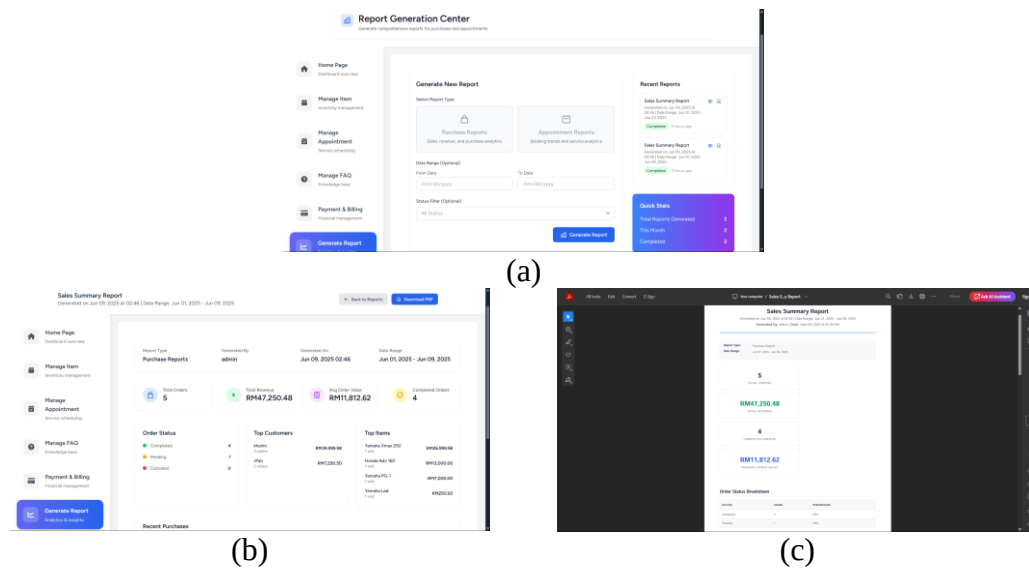


Fig. 23 (a) Report Generation Center; (b) Report Summary; (c) PDF for Report for Admin

5.2 Requirement Testing

The testing phase focused on verifying that each system module aligns with its defined functional requirements. This process involved executing structured test cases in Table 5 across various components such as user authentication, item and service management, purchasing, bookings, payments, FAQs, and report generation. Each test was designed to assess whether the system behaved as expected under specific conditions, ensuring the reliability, accuracy, and completeness of implemented features without evaluating the success or failure outcomes at this stage.

Table 5 Test Cases for All Module

Test Case ID	Requirement ID	Description	Status
TC_100	REQ_100	Register and Login	
TC_100_01	REQ_101	The system must allow users to register.	Pass
TC_100_02	REQ_102	The system must allow users to input their name, email and password.	Pass
TC_100_03	REQ_103	The system must validate registration inputs.	Pass
TC_100_04	REQ_104	The system must prevent registration with invalid or duplicate inputs.	Pass
TC_100_05	REQ_105	The system must allow users to log in using valid credentials.	Pass
TC_100_06	REQ_106	The system must validate login inputs and provide appropriate feedback.	Pass
TC_100_07	REQ_107	The system must restrict login if credentials are invalid.	Pass
TC_100_08	REQ_108	The system must provide role-based redirection upon successful login.	Pass
TC_200	REQ_200	Manage Item	
TC_200_01	REQ_201	The system must allow admin to create new item listings.	Pass
TC_200_02	REQ_202	The system must allow admin to update item details such as name, price and other details.	Pass
TC_200_03	REQ_203	The system must allow admin to delete item listings.	Pass
TC_200_04	REQ_204	The system must display all items in a structured list or table.	Pass
TC_200_05	REQ_205	The system must store uploaded images for each item.	Pass

Table 5 Continued

TC_200_06	REQ_206	The system must validate item input fields before submission.	Pass
TC_300	REQ_300	Purchase Item	
TC_300_01	REQ_301	The system must allow customers to view available items for purchase.	Pass
TC_300_02	REQ_302	The system must display detailed information for each item.	Pass
TC_300_03	REQ_303	The system must allow customers to add items to a cart or purchase directly.	Pass
TC_300_04	REQ_304	The system must process customer purchase orders.	Pass
TC_300_05	REQ_305	The system must generate a purchase record linked to the customer profile.	Pass
TC_300_06	REQ_306	The system must update the status of purchased items after successful payment.	Pass
TC_400	REQ_400	Manage Booking	
TC_400_01	REQ_401	The system must allow admin to create new service booking slots.	Pass
TC_400_02	REQ_402	The system must allow admin to edit existing booking details.	Pass
TC_400_03	REQ_403	The system must allow admin to delete outdated or cancelled bookings.	Pass
TC_400_04	REQ_404	The system must display all booking records in an organized view.	Pass
TC_400_05	REQ_405	The system must notify the customer if a booking is edited or cancelled.	Fail
TC_500	REQ_500	Booking Service	
TC_500_01	REQ_501	The system must allow customers to view available service booking slots.	Pass
TC_500_02	REQ_502	The system must allow customers to submit a service appointment request.	Pass
TC_500_03	REQ_503	The system must validate user inputs during the booking process.	Pass
TC_500_04	REQ_504	The system must confirm and store booking details in the customer profile.	Pass
TC_500_05	REQ_505	The system must notify the customer of a successful booking via the interface.	Pass
TC_600	REQ_600	Frequently Asked Question	
TC_600_01	REQ_601	The system must allow admin to add new FAQ entries.	Pass
TC_600_02	REQ_602	The system must allow admin to update existing FAQ entries.	Pass
TC_600_03	REQ_603	The system must allow admin to delete outdated or incorrect FAQ entries.	Pass
TC_600_04	REQ_604	The system must display all FAQ entries clearly for customers.	Pass
TC_600_05	REQ_605	The system must allow customers to access FAQ without logging in.	Fail
TC_700	REQ_700	Manage Payment and Billing	
TC_700_01	REQ_701	The system must allow customers to make secure online payments.	Pass
TC_700_02	REQ_702	The system must record each payment transaction with reference details.	Pass
TC_700_03	REQ_703	The system must allow an admin to view and manage all payment records.	Pass
TC_700_04	REQ_704	The system must display payment history to customers.	Pass

Table 5 Continued

TC_700_05	REQ_705	The system must update the status of purchases upon successful payment.	Pass
TC_700_06	REQ_706	The system must generate a digital invoice after payment.	Pass
TC_800	REQ_800	Generate Report	
TC_800_01	REQ_801	The system must allow admin to generate sales reports.	Pass
TC_800_02	REQ_802	The system must allow admin to generate service appointment reports.	Pass
TC_800_03	REQ_803	The system must allow admin to generate payment and billing reports.	Pass
TC_800_04	REQ_804	The system must allow admin to filter reports by date range or category.	Pass
TC_800_05	REQ_805	The system must support export reports in a readable format (e.g., PDF).	Pass

5.3 Overall Test Result

The testing process for the AkarjayaMotor platform encompassed 8 main modules and a total of 46 test cases, each mapped to specific functional requirements in Table 6. These modules include Register and Login, Manage Item, Purchase Item, Manage Booking, Booking Service, Frequently Asked Questions (FAQ), Manage Payment and Billing, and Generate Report.

Table 6 Test Cases Summary for All Module

Test Cases	Total Test Cases	Total Success	Total Failed
TC_100	8	8	-
TC_200	6	6	-
TC_300	6	6	-
TC_400	5	4	1
TC_500	5	5	-
TC_600	5	4	1
TC_700	6	6	-
TC_800	5	5	-

5.4 User Acceptance Testing

User Acceptance Testing (UAT) for the AkarjayaMotor: The All-in-One Motorcycle Platform was conducted to ensure the system meets user expectations in real-world usage, focusing on functionality, usability, and alignment with business needs. UAT covered eight core modules including user registration, motorcycle purchases, service booking, payments, FAQs, and report generation, evaluated through 46 test cases and feedback from 20 respondents using a five-point Likert scale. Results showed high user satisfaction, with 90% finding the interface easy to navigate and 95% reporting a smooth purchasing process. All test cases passed successfully, with no critical issues found. User suggestions, such as mobile support and additional interactive features, were also gathered for future improvements. The results confirm the system is stable, user-friendly, and ready for deployment, effectively fulfilling its intended objectives and ensuring confidence among stakeholders.

6. Conclusion

This report concludes the development of AkarjayaMotor: The All-in-One Motorcycle Platform by highlighting its implementation process, achievements, and future improvement areas. The system has successfully met all the outlined objectives. It was designed using an object-oriented approach, developed as a web-based application, and thoroughly tested through both Functional Testing and User Acceptance Testing. The platform effectively resolves key challenges faced by Akarjaya Motor Sdn Bhd in managing motorcycle sales, service bookings, user interactions, and payment processing. With well-integrated modules for user registration, item management, purchases, service bookings, payments, FAQs, and report generation, the system enhances operational efficiency and customer experience. Although the current version reveals some limitations such as lack of mobile optimization and limited report customization, it establishes a strong digital foundation. The platform is well-positioned for future enhancements, supporting the company's growth through improved functionality, accessibility, and sustainability in the competitive motorcycle industry.

6.1 Advantages

The AkarjayaMotor: The All-in-One Motorcycle Platform provides numerous advantages to both the company and its users, driven by its fully integrated and tested core modules. One of the major benefits is the streamlined management of motorcycle listings, allowing administrators to easily add, update, or remove items, ensuring accurate and efficient inventory control. At the same time, customers benefit from a user-friendly browsing and purchasing experience, where they can view available motorcycles, add items to their cart, and complete secure payments through the integrated Stripe system.

Beyond sales, the platform supports post-purchase engagement by enabling customers to schedule and manage service appointments directly within the system. Multi-user authentication ensures that admins and customers access only the modules relevant to their roles, which enhances both security and overall usability. Customers can also track their payment history and download invoices, while administrators are equipped to monitor and manage transaction data effectively, improving operational oversight.

In addition, the platform includes a centralized FAQ module that provides instant answers to common questions, reducing the burden on customer service teams. Administrators can also generate detailed reports on sales, bookings, and payments, supporting data-driven decision-making and strategic planning. Overall, the platform not only modernizes the company's operations but also enhances the customer experience, positioning AkarjayaMotor for sustainable digital growth.

6.2 Limitation

Despite the successful development and deployment of the AkarjayaMotor: The All-in-One Motorcycle Platform, several limitations were identified during the testing and implementation phases. These limitations, while not critical to system functionality, highlight areas where future enhancements could significantly improve user experience and operational efficiency. Recognizing these issues is essential for planning ongoing development and ensuring the platform remains competitive and user-friendly.

One notable limitation is the lack of a fully responsive or mobile-optimized interface. The current system is primarily designed for desktop users, which may lead to reduced usability and accessibility for customers who access the platform through smartphones or tablets. In today's mobile-first environment, this could limit user engagement and convenience, particularly for customers who prefer managing purchases and bookings on the go.

Another limitation lies in the reporting module. While the platform supports basic report generation for administrators, it lacks advanced customization options such as filtering by specific date ranges, motorcycle categories, or user types. This restricts the depth of analysis that admins can perform and may hinder effective decision-making. Addressing these limitations in future system updates would enhance the overall flexibility and functionality of the platform.

6.3 Future Works

Based on the limitations identified in the current version of the AkarjayaMotor: The All-in-One Motorcycle Platform, several key areas have been proposed for future development. These enhancements aim to address current gaps and further optimize the platform's functionality, accessibility, and overall user experience. By focusing on continuous improvement, the system can better support both customer needs and administrative operations.

One of the primary recommendations is to improve mobile accessibility. As the current version is optimized mainly for desktop use, future updates should include a fully responsive web interface or a dedicated mobile application. This will ensure a seamless experience for users accessing the platform via smartphones or tablets, allowing them to browse motorcycles, book services, and make payments with greater ease and convenience.

Another important area for improvement is the reporting module. While basic reporting functions are in place, future iterations could benefit from more advanced features such as customizable filters, exportable data formats and graphical analytics. These improvements would enable administrators to perform more in-depth analysis, track performance trends, and make informed decisions to drive the business forward.

Acknowledgement

The author wants to sincerely thank everyone who supported him during his adventure to finish this thesis. Above all, the author extends his deepest gratitude to his lecturers and supervisor. Their extraordinary guidance, deep expertise and constant encouragement have been extraordinarily important in the development of this work and in importantly expanding their understanding. The author is also deeply grateful to his friends for their firm support, fellowship, as well as the many moments of motivation shared during difficult times. His family deserves special recognition for their unconditional love, patience, as well as belief in his abilities, which have been his pillars of strength throughout this adventure.

Conflict of Interest

Authors declare that there is no conflict of interest regarding the publication of the paper.

Author Contribution

The study's conceptualization and design, data collection, analysis and interpretation of the findings, and article writing are all solely the author's responsibility.

References

- [1] Z. Kedah, "Use of E-Commerce in The World of Business," *Startupreneur Bisnis Digital (SABDA)*, vol. 2, no. 1, 2023.
- [2] M. Fusko, L. Dulina, P. Bubeník, M. Bučková, M. Kasajová and R. Svitek, "THE IMPORTANCE OF DIGITIZATION AND INNOVATIONS FOR SMALL AND MEDIUM-SIZED ENTERPRISES," *Proceedings of CBU in Economics and Business*, vol. 2, pp. 32-40, 2021.
- [3] G. Taher, "E-Commerce: Advantages and Limitations," *International Journal of Academic Research in Accounting, Finance and Management Sciences*, vol. 11, no. 1, 2021.
- [4] S. Kraus, P. Jones, N. Kailer, A. Weinmann, N. Chaparro-Banegas and N. Roig-Tierno, "Digital Transformation: An Overview of the Current State of the Art of Research," *SAGE Open*, vol. 11, no. 3, 2021.
- [5] H. Grigoryan, "'Electronic business", "electronic commerce": Concepts and main characteristics," *Alternative*, 2023.
- [6] T. Bhumika, Jyoti, G. Neha and K. Santosh, "Overview of Electronic commerce (E-commerce)," *I-manager S Journal on Information Technology*, vol. 11, no. 2, p. 29, 2022.
- [7] Y. Baashar, H. Alhussian, A. Patel, G. Alkaws, A. I. Alzahrani, O. Alfarraj and G. Hayder, "Customer relationship management systems (CRMS) in the healthcare environment: A systematic literature review," *Elsevier B.V.*, 2020.
- [8] V. Guerola-Navarro, H. Gil-Gomez, R. Oltra-Badenes and J. Sendra-García, "Customer relationship management and its impact on innovation: A literature review," *Journal of Business Research*, vol. 129, pp. 83-87, 2021.
- [9] P. A. R, N. R and K. K. M. N, "Inventory Management System," *International Journal of Advanced Research in Science, Communication and Technology*, pp. 296-300, 2024.
- [10] D. P. G. Mehar, A. Pardhi, A. Borkar, D. R. Vaidya, H. R. Umredkar and R. Somkuwar, "Development of Inventory Management System Used in Production Industry," *International Journal for Research in Applied Science and Engineering Technology*, vol. 11, no. 4, pp. 4540-4543, 2023.
- [11] M. A. Ali, N. Hussin and I. A. Abed, "Electronic payment systems: Architecture, elements, challenges and security concepts: An overview," *Journal of Computational and Theoretical Nanoscience*, vol. 16, no. 11, pp. 4826-4838, 2019.
- [12] A. P. Thangamuthu, "A Survey on Various Online Payment and Billing Techniques," *Shanlax International Journal of Arts, Science and Humanities*, vol. 7, no. 3, pp. 86-91, 2020.
- [13] Motomalaysia, "Motomalaysia," [Online]. Available: <https://www.motomalaysia.com/>. [Accessed 18 12 2024].
- [14] Wmoto, "Wmoto," [Online]. Available: <https://www.wmoto.my/>. [Accessed 18 December 2024].
- [15] Imotorbike, "Imotorbike," [Online]. Available: <https://imotorbike.com/my/en/>. [Accessed 18 December 2024].
- [16] J. Smith, "2023 Business Valuation Considerations," Smith Schafer, 2023. [Online]. Available: <https://www.smithschafer.com/blog/2023-business-valuation-considerations/>. [Accessed 18 December 2024].
- [17] B. Black and G. Blue, "How a Dealership Evaluates a Trade," Reddit, 2023. [Online]. Available: https://www.reddit.com/r/askcarsales/comments/t9to4h/how_a_dealership_evaluates_a_trade_and_why_kbb_is/. [Accessed 18 December 2024].
- [18] N. Rachma and I. Muhlas, "Comparison of waterfall and prototyping models in research and development (R&D) methods for android-based learning application design," *Jurnal Inovatif: Inovasi Teknologi Informasi Dan Informatika*, vol. 5, no. 1, pp. 36-39, 2022.