APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 7 No. 1 (2026) 390-409
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

# Traveller-Friendly Vehicle Rental Web-Based Application (TWB): A Web-Based Application to Optimize Business Efficiency

**Nur Aina Natasha Hasan[1], Rosmamalmi Mat Nawi[1*]**

[1] *Faculty of Computer Science and Information Technology*
*Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

*Corresponding Author: rosmamalmi@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2026.07.01.021

**Article Info**

Received: 12 June 2025
Accepted: 18 June 2026
Available online: 30 June 2026

**Keywords**

Web-Based, vehicle rental system, rental, traveller

**Abstract**

Since the previous system for renting vehicles involved lots of paperwork and was inefficient, the Traveller-Friendly Vehicle Rental Web-Based Application was made to modernize it and remove these hassles. The purpose of this solution is to make things easier, more accurate, and boost productivity in the company. Knowledge from both customers and businesses pointed out the key issues and important needs that were important for the system. The main purpose of the platform was to provide better information, make the use of cars more effective, and offer top customer service which led to greater client satisfaction, more efficient operations, and a sustainable business for the rental vehicle industry by using PHP with Laravel and MySQL as the database. In the development phase, programmers kept testing and repairing to avoid any errors and to guarantee the application's functionality, safety, and dependability. The end product is a travel-friendly car rental web app with booking, payment, and car management modules. It provides an easy-to-use user interface, a well-structured database, and fully satisfies the project specifications. The project was successfully completed as all features have been implemented, and the platform is operating seamlessly without any disruptions.

## 1. Introduction

Because of the constantly changing needs of today's travelers, the Traveler-Friendly Vehicle Rental Web Application (TWB) was designed to meet this demand for quick, reliable, and adjustable transportation options. Although Sixt and Joe Saunders started car rentals in the early 1900s [1], a lot of today's owners and independent companies still use dated, traditional practices. Because of these technologies, companies may find it hard to manage bookings, create a proper inventory of vehicles, and look after customer details swiftly. Besides, handling paper by hand can create problems such as the possibility of theft or unauthorized access, leading to loss of vital data.

TWB aims to transform the rental experience by automating key functions, strengthening data security, providing real-time updates [2], and offering an intuitive, user-friendly interface. By doing so, the platform enhances accessibility, streamlines operations, and boosts efficiency for both car owners and users. Manual operations in the rental industry not only slow down business processes but also create opportunities for disputes, customer dissatisfaction, and operational bottlenecks. Managing large volumes of paperwork is time-consuming, labor-intensive, and prone to human error factors that hinder scalability and business growth.

Without a digital system in place [3], it becomes increasingly difficult to accurately track vehicle availability, maintain reliable records, or provide a seamless customer experience, thereby reducing a company's competitiveness in the market.

The primary goal of TWB is to deliver a user-friendly, web-based solution that simplifies the vehicle rental process, particularly for travelers [4]. With the system, users are able to see all the cars, fill in what they want to rent, and finish booking in a simple way. With the help of the app, renting a car, paying, and handling drivers' information has become automated, leading to better daily operations.

The platform by TWB is trusted and covers user registration, adding vehicles, scheduling trips, managing payments, and making contracts [5]. The user can enjoy an easy-to-use app, live updates on available vehicles, customers' feedback, effective communication tools, and suggestions for any favorite vehicles. Many users can be part of the platform, such as people who book transport, those who manage reservations, and people offering their cars. The presence of admin consoles, safe payment gateways, and recommend tools helps the site work better and safer [6].

Automation of everything from booking to managing data helps TWB offer a trustworthy and growing solution for anyone renting or owning a car. As a result, companies can handle fewer paper works, lessen the chance of errors, and spend less on daily tasks.

Consequently, TWB enhanced vehicle rental services by eliminating old manual tasks, boosting the protection of data, and improving how the operations are carried out. In addition, it helps boost the economy by letting private drivers share equipment and operate in a way that helps preserve the environment by giving people an alternative to buying cars. Since digital transactions are safe, rental agreements are understandable, and reviews are provided by users. Customers and business owners enjoy increased trust and reliability.

The remainder of this paper is organized as follows as in Section 2 examines previous works, domain background and comparable systems in the domain of rental car system. Section 3 explains the methodology adopted for developing the TWB. Section 4 illustrates the system's design and implementation process. Section 5 discusses user responses. Section 6 concludes the paper and outlines the potential areas for future development.

## 2. Related Work

The main purpose of the suggested solution is to establish a smooth and secure system that links users who want to rent a car with those who have vehicles to give out. Because it's easy to use, users can easily look at what's available, confirm the parking is vacant when they want it, and book a space without any trouble. To make the experience better, the system will also include things like secure logins, ways to communicate in real-time, and features that use location services.

### 2.1 Domain Background

Abbad Berkat Enterprise Sdn Bhd started its journey back in 2015. It began as a small car rental business, focusing on providing affordable transport options. Right now, though, the company still manages bookings and payments using a manual, paper-based system. As their fleet has grown, this old-fashioned method has become inefficient. The new system we're suggesting will replace this outdated approach. It will use MySQL to handle all the data and Laravel for the behind-the-scenes work. This digital switch will make managing their cars much easier, simplify how bookings and payments are handled, and give them up-to-the-minute tracking information.

### 2.2 Comparison on Existing Similar Systems

Shifting from their current manual system to the new digital platform is set to tackle some big problems they face now, like mistakes in bookings, keeping records efficiently, and not having real-time information. Their current system involves handwritten booking notes, physical paper records for transactions, and relying on phone calls or verbal agreements to check if a car is free. This often leads to human errors, mixed messages, and makes it hard to keep track of rental histories and how the cars are being used. With the new system, the car owners will get automated booking confirmations, payments will be processed more smoothly, and it will be easy to see the rental history and which vehicles are available. Plus, the simple design for customers will let them see the most current details about the cars and their availability, helping them make better choices. By using this new solution, Abbad Berkat Enterprise Sdn Bhd will operate much more efficiently and offer a service that's more reliable and convenient for both the people renting cars and the owners lending them out, which should help them compete better in the market. We'll outline the specific benefits of the new system compared to the old one to show clearly how much better the new approach is.

**Table 1** *Comparison between the existing system and the proposed system*

| Features | Socar.my [7] | Wahdah.my [8] | Rentalcars.com [9] | The Proposed System |
|---|---|---|---|---|
| User Registration & Authentication | ✓ | ✓ | ✓ | ✓ |
| Login with Email | ✓ | ✓ | ✓ | ✓ |
| Register for New User | ✓ | ✓ | ✓ | ✓ |
| Car Details Button | ✓ | ✗ | ✓ | ✓ |
| Booking and Reservation | ✓ | ✓ | ✓ | ✓ |
| Payment Options | ✓ | ✗ | ✓ | ✓ |
| Delivery Service Option | ✓ | ✗ | ✗ | ✓ |
| Personal Details Submission | ✗ | ✗ | ✗ | ✓ |
| Review and Rating System | ✓ | ✓ | ✓ | ✓ |
| Communication Between User & Car Owner | ✓ | ✗ | ✓ | ✓ |
| Car Suggestion System | ✗ | ✗ | ✗ | ✓ |
| Admin Dashboard for Management | ✓ | ✓ | ✓ | ✓ |
| Real-Time Car Availability Updates | ✓ | ✓ | ✓ | ✓ |

The new system designed for Abbad Berkat Enterprise Sdn Bhd is important for raising the efficiency and accuracy of car rental management processes in the company. Moving from paper records to an online platform will give the company fast and accurate booking, less error checking, and frequent updates on which vehicles are available. Because of digital transformation, vehicle owners will manage their listings more easily and clients will evaluate their options with greater help.

Showing how the modern system differs from the current one makes it evident why using automation and advanced technology is beneficial. The enhanced system allows Abbad Berkat Enterprise to offer better customer service and remain on par with competitors as the car rental industry keeps advancing.

## 2.2 Technology Used

The creators built TWB using Laravel, a framework that is reliable, performs well, and is known for its security and the ability to scale. It depends on MySQL as the relational database management system for efficient handling of info. All of these technologies work together to give a smooth and safe vehicle rental platform for everyone.

### 2.2.1 Laravel Framework

Using Model-View-Controller (MVC) architecture, Laravel gives web developers a clear framework for application creation. Because of Blade templating engine, built-in security tools, routing, and database migration, developers can make secure and rich applications. It is possible to setup the environment using Homestead or XAMPP applications. The main features of registration, login, vehicle administration, making and managing bookings, charges, and safe navigation are efficiently built in Laravel. Additionally, Laravel enables programmers to code RESTful APIs, which make it possible to integrate with future mobiles apps or other programs.

## 3. Methodology

The application is created through the use of the Agile [10] approach due to its quick adjustments, recurring progress, and commitment to users. Using Agile, the development team can gradually release new features, regularly listen to user input, and adjust fast to any changes in the market. This way of working helps the app develop faster and still satisfy users with its reliability, usefulness, and proper functionality.

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 390-409

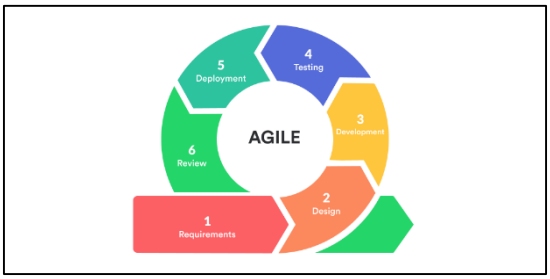


**Fig. 1** *Agile Methodology Phase*

Breaking up extensive jobs into clear steps and by joining forces with the team, Agile prevents risks, raises user satisfaction, and backs continual improvements. With this method, it is easy to make a flexible platform that meets the needs of today's travelers.

## 3.1 Requirements Phase

At the onset of the TWB project, the team focused on studying what the new system should have in order to solve the standard issues with vehicle rental. It was decided to use Laravel, MySQL, and Figma in the development process since this ensures the platform is easy to expand, safe, and user-friendly. They made detailed plans for the project and built a new Gantt chart to track the project milestones as well. Since the planning was strategic, progress followed the project goals all through the different stages of implementation.

## 3.2 Analysis Phase

During the analysis phase, interviews with stakeholders such as vehicle rental companies, potential users, and technical experts helped gather both functional and non-functional requirements for TWB application. Key functional needs identified included browsing car inventory, making reservations, handling payments, and generating reports, while non-functional requirements focused on system performance, scalability, data security, and usability. The interviews revealed challenges such as inefficiencies in manual processes, lack of transparency, and the need for a user-friendly interface with robust security. These findings led to the development of a digital platform designed to address these pain points, streamline the rental process, enhance customer satisfaction, and ensure a seamless, secure experience for users.

## 3.3 Design Phase

TWB team built a reliable database and developed a user-friendly UI to help simplify booking, managing accounts, and selecting cars. Figma was very important for speedy testing and for adjusting the design according to stakeholder feedback. The design of the interface focuses on what users require and raises the quality of their experience. At the same time, the database was set up to smoothly save customer data, car details, and information on all rentals. Such a well-designed system runs smoothly and securely deals with data.

## 3.4 Coding Phase

TWB's development process meant turning the designs and needs into actual code. HTML, CSS, and JavaScript made the user interface more enjoyable, while data security was handled by Laravel and PHP at the back end. MySQL made it easy to keep and look up data. By frequently testing and debugging, it was made sure that the system worked as expected. Functions such as user sign-up, searching for cars, organizing reservations, and handling payments were strictly put in place to ensure everything worked well and safely. The testing phase was very important because it allowed the project to be completed successfully with all the needed features in place.

## 3.5 Testing Phase

During the testing phase, the project team checks if the system meets all the required standards. People use a combination of testing methods for this reason. As an example, system testing makes sure that reservations, car browsing, and the payment process are error-free. By doing integration testing, all system components are tested to function, and unit testing ensures each part behaves properly on its own. Before the system is launched, people who will be using it perform User Acceptability Testing (UAT). Taking this step enables us to improve the project by relying on their feedback. This type of testing helps preserve users' confidential information and their payments. In turn, performance tests demonstrate that the system is able to support many activities and be active during busy times. Problems that are found are fixed by testing and validating the design several times. This strict method is followed so that the product provides the best quality.

### 3.6 Maintenance Phase

When the TWB Web-Based Application is fully set up, it begins the maintenance phase. During this stage, the team attempts to eliminate bugs, patch any holes in the security, and enhance the system's overall quality. They take users' feedback into account and develop new features as a result. To keep things in order, it is important to make backups regularly, maintain an efficient database, and watch out for errors or problems with using the system. Both security and up-to-date program features are maintained by using updates and fixes as soon as they are available. At the same time, what users say helps improve the design and experience offered by the platform. Keeping things updated regularly gives the application greater dependability, helps it match current user needs, and keeps people content.

## 4. Results and discussions

This section reports the key findings of the study and discusses them in detail in terms of what was set out in the research and in relation to relevant references.

### 4.1 System Analysis and Design

The analysis and design part were meant to thoroughly explore how TWB operates. At this point, different diagrams were made to represent the interactions between parts of the system and the flow of data. The concept of data exchange, interaction between parts of the system, and the role of users were shown by drawing flowcharts, a context diagram, a Level 0 Data Flow Diagram (DFD), and an Entity-Relationship Diagram (ERD).

### 4.2 Flowchart

In this phase, the important task is to keep the application up, safe, and easy to use. Among other things, this includes detecting and fixing bugs, securing the website from risks, making it faster, and bringing in new features as required by users or technologies. Regular maintenance tasks are system backups, keeping data organized, and checking the system for performance problems. User input is key in improving the interface, and updates help preserve the app's reliability, security, as well as users' satisfaction.

### 4.3 User Flowchart

The steps a user takes while using the TWB platform are clearly shown in the User Flowchart shown in Figure 2. This covers registration, browsing vehicles, and making bookings, which makes everything clearer and easier for users. The list of responsibilities in the Administrator Flowchart, which can be found in Appendix A, explaining all the vital activities for guaranteeing the security, legality, and function of the system. Moreover, the Renter Flowchart in Appendix A explains the process that individuals use to rent their cars by using the application. Car owners can use this guide to list their rides and handle them in a user-friendly way. To ensure the site works well, administrators oversee user accounts, cars for rent, and keep a watch on all booking and payment processes.

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 390-409

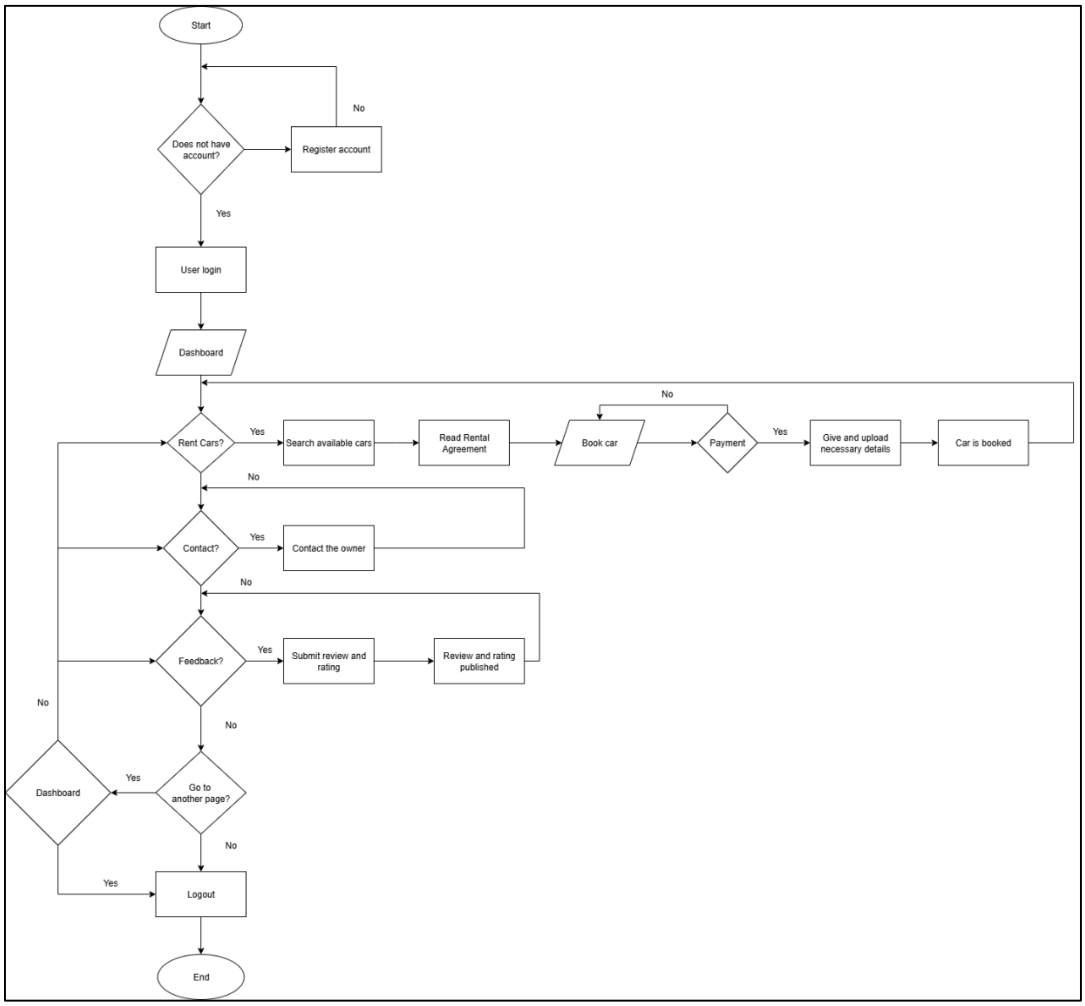


**Fig. 2** *User Flowchart of TWB system*

## 4.4 Context Diagram

Figure 3 presents the context diagram for the TWB system, which portrays the system's main connections with outside entities at a high level.

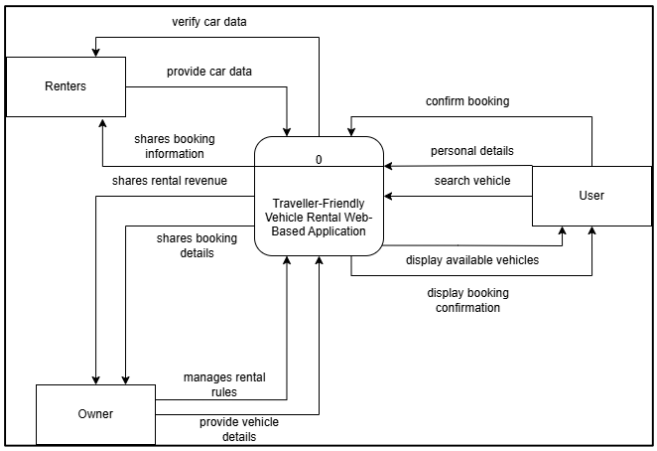


**Fig. 3** *Context Diagram of TWB system*

## 4.5 Data Flow Diagram

The Data Flow Diagram (DFD) can be located in Appendix B. All in all, these sections explain how data flows within the system and what its structure is.

## 4.6 Entity Relationship Diagram

The relationships between users, rental transactions, and vehicles of the TWB system are shown in Figure 4 which is an Entity Relationship Diagram (ERD). This guides how the database is structured and how data is organized and flows within the application.

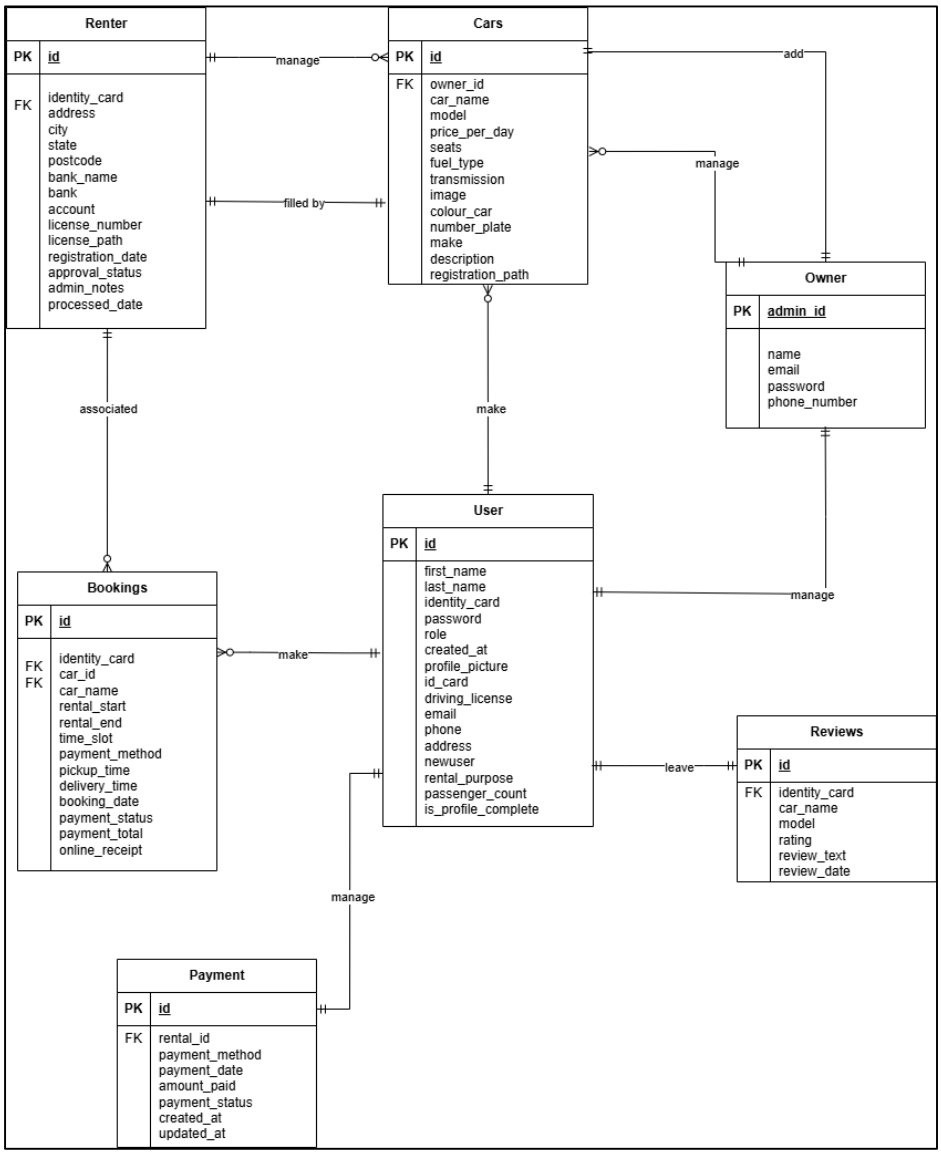


**Fig. 4** *Entity Relationship Diagram (ERD) of TWB system*

## 4.7 User Interface

TWB gives priority to making its user interface intuitive, fast, and responsive so that anyone using the application can navigate it with ease. The user interface is made to help users complete all steps, such as signing in and finishing a vehicle rental without any difficulty. Registration and login screens are easy to navigate and have been optimized for all users. The final part includes pictures of the UI components described in this appendix C.

# 5. Implementation

This chapter covers how TWB was assembled step by step. This section inspects the instructions to set up various aspects of the system, for example, user registration and login, fleet management, processing bookings and reservations, payments, agreements, reviews and ratings, dashboard, suggestions for vehicles, and car listing.

## 5.1 Implementation of User Registration and Authentication Module

Users will begin the process of User Registration and Authentication in the TWB system, by filling in their first name, last name, identity card number, password and confirm password in the form. The system will validate the form field to check each field is not left empty and that password matches the confirm password field. The identity card number will be checked with the existing records which in the database to avoid duplicates. If the validation fails the user will see an error message in order to correct the entry. If there are no errors and the validation passes, the system encrypts the password and stores user details into the database securely. After that, the user will also see a success message after registration and will redirect the user to the login page. Figure 5 shows the registration form interface which shows the input fields and the password criteria that must be adhere to for the users to successfully register.

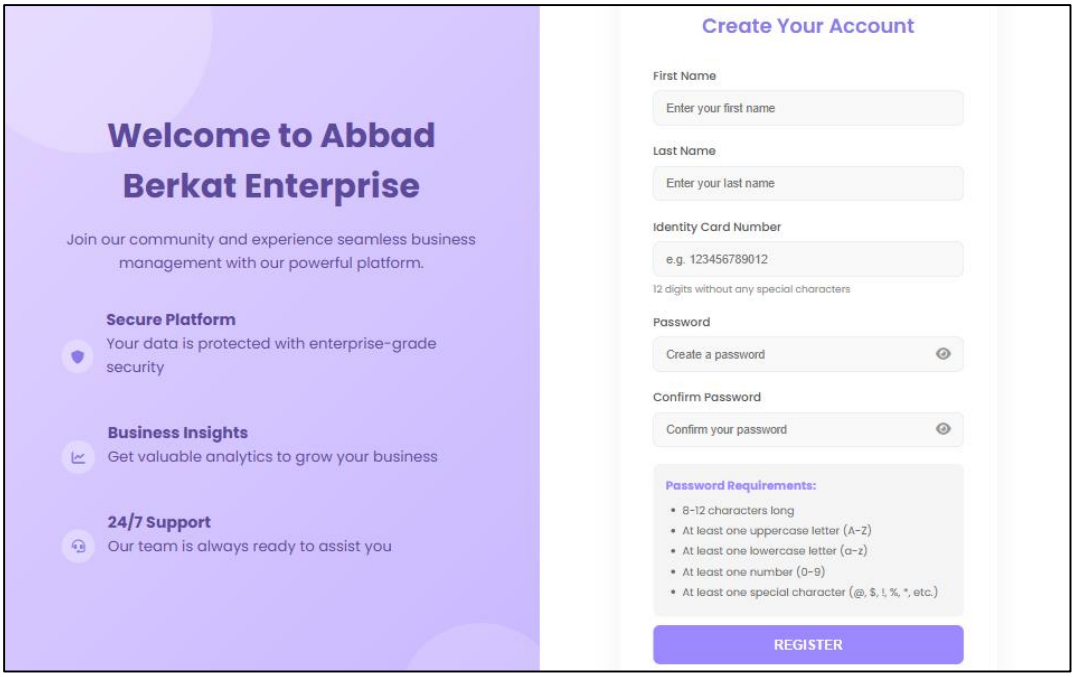


**Fig. 5** *Registration form of the user of TWB system*

## 5.2 Implementation of Vehicle Management Module

Let's look now at the user vehicle management module within the proposed system. Figure 6 shows the algorithm that is implemented in removing a vehicle from the list or system. The process works like this: when someone clicks the delete button, a confirmation window pops up. This window displays the car's name and asks the user to confirm if they really want to delete it. If they proceed by clicking 'delete', the system sends the specific car's ID to the server to ensure that exact vehicle is permanently removed. Meanwhile, Figure 7 gives a view of the vehicle management module as it appears on the admin page. This is where the administrator has the ability to add new cars, edit details of existing ones, and delete vehicles from the system.

```python
from sklearn.preprocessing import OneHotEncoder, MultiLabelBinarizer
import pandas as pd
import numpy as np

# Sample data
data = pd.DataFrame({
    'Owner': [['Aina'], ['Amir', 'Anis'], ['Aisyah']],
    'Brand': ['Toyota', 'Honda', 'Toyota'],
    'Type': ['SUV', 'Sedan', 'Truck'],
    'Year': [2005, 2012, 1999]
})

# Fill missing values
data.fillna({'Owner': ['Unknown'], 'Brand': 'Unknown', 'Type': 'Unknown'}, inplace=True)

# Encode Brand and Type
encoder_brand = OneHotEncoder(sparse=False)
brand_encoded = encoder_brand.fit_transform(data[['Brand']])

encoder_type = OneHotEncoder(sparse=False)
type_encoded = encoder_type.fit_transform(data[['Type']])

# Encode Owner (multi-label)
mlb_owner = MultiLabelBinarizer()
owner_encoded = mlb_owner.fit_transform(data['Owner'])

# Combine features
combined_features = np.hstack([brand_encoded, type_encoded, owner_encoded, data[['Year']]])

# Deletion logic (e.g., keep only Year >= 2010)
filtered_data = data[data['Year'] >= 2010]

print(filtered_data)
```

**Fig. 6** *Algorithm implemented in deleting vehicles in TWB system*

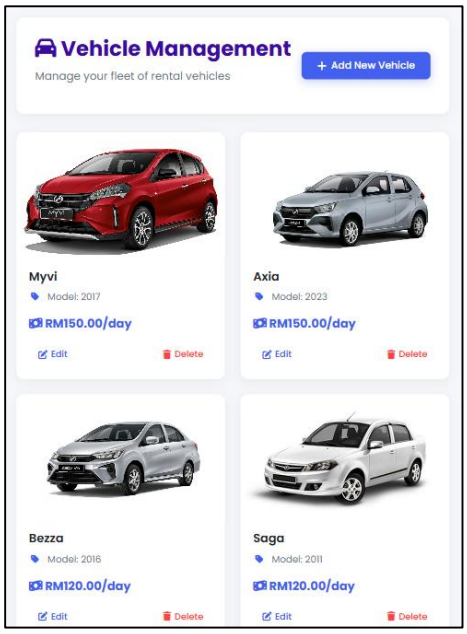


**Fig. 7** *Vehicle Management module of TWB system*

## 5.3 Implementation of Booking and Reservation Module

This section describes the implementation of the Booking and Reservation Module in the TWB system. This module becomes accessible to users only after they have successfully registered and logged in. In this module, there is a specific section of the booking form based on the selected dates. If the start date and end date are the same, the system reveals an additional row that allows users to select a specific time, supporting half-day bookings. The booking process begins by prompting users to enter the start date and end date of the intended rental period. As shown in Figure 8, the system checks the availability of vehicles based on the selected dates. If the rental spans an entire day or multiple days, the platform displays all available vehicles for full-day booking. However, if the start and end dates are the same, the system switches to the half-day booking section. Additionally, the system provides detailed information about each available vehicle, allowing users to make informed decisions about which car best suits their needs before confirming the reservation.

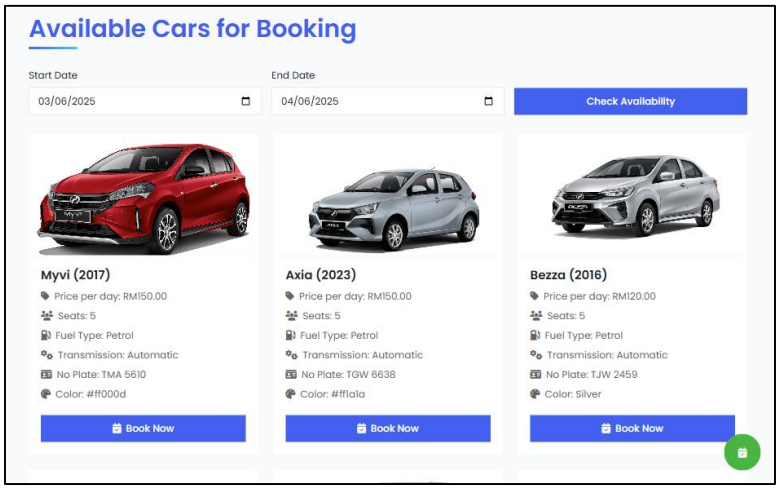


**Fig. 8** *Vehicle Management module of TWB system*

## 5.4 Implementation of Payment Processing Module

This section discusses the implementation of payment processing module. In this module, it will present a summary as shown in Figure 9 on how much a car rental will cost based on the rental dates and the selected time slot. The total cost is calculated by multiplying the number of days with the daily charge, but for a half day, pay just half of the normal price. With the system, anyone getting a rental car can pay by QR code or cash before receiving the vehicle. Collecting the vehicle is easy, since the payment is using cash or a QR code to pay on the platform**.** After getting the booking confirmation, everyone receives their receipt, and user can make payments just by scanning the QR code or through cash at the pickup point, making it safer and convenient for them.


**Fig. 9** *Summary of Payment Processing Module*

## 5.5 Implementation of Rental Agreement Management Module

In this system, user and car rental agency can view their rental agreements anytime they need. This aspect makes it easier for the two parties to be open and avoid misunderstandings. Any time a user needs to look at the agreement, they can call it up and view or confirm the stated details. Shown in Figure 10, users need to read the car rental terms first, which have been agreed upon, before they finish booking the car.

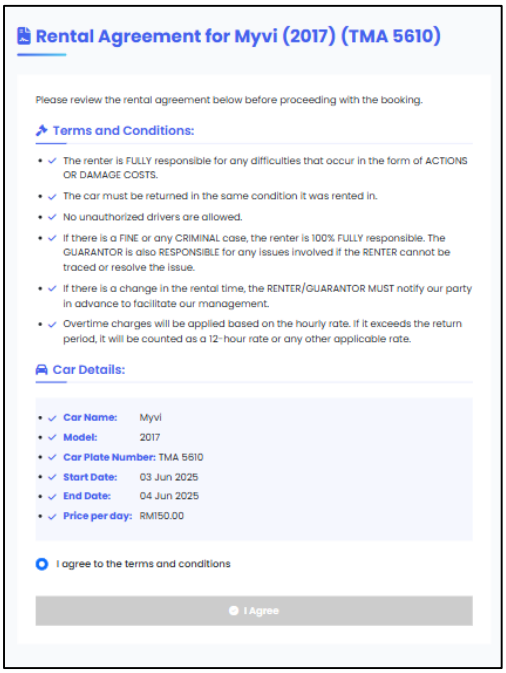


**Fig. 10** *Rental Agreement Management Module of TWB system*

## 5.6 Implementation of Rating and Review Module

The section describes how the system is built so that users can review and rate their stay after the rental. Having this feature helps the platform become more honest and trusted by everyone. The user's rental history will be recorded in the system including the car's name, model, and the day it was rented. Should a user not have any rental history, the table will show a message that there are no old bookings. Apart from writing their own thoughts, users are able to see the comments from others, allowing them to decide by reading different opinions. People who want to submit a review must select their car make, set the rating, and describe their experience through text as shown in Figure 11.

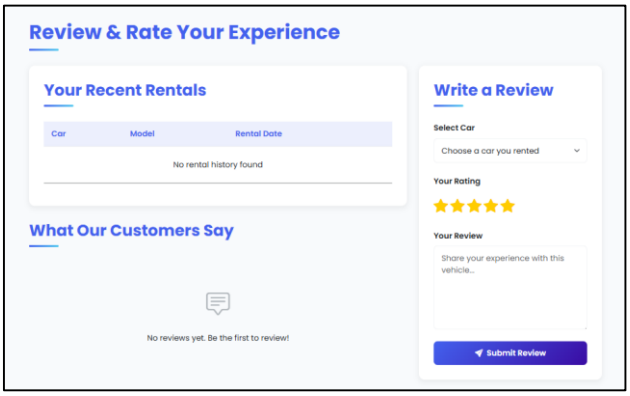


**Fig. 11** *Rating and Review Module of TWB system*

## 5.7 Implementation of Admin Dashboard Module

This part focuses on the admin dashboard module. Think of it as the central command center for administrators, giving them one place to handle everything from vehicle rentals and user accounts to managing the fleet itself. It keeps admins in the loop with live updates on ongoing rentals, which vehicles are available, and what users are doing, making it easier to run the system efficiently. The dashboard is packed with useful tools too, like adding or removing vehicles, deciding whether to approve or deny rental requests, and keeping an eye on how the whole system is performing. For instance, Figure 12 displays the schedule of booked cars, serving as a handy reminder for the admin.

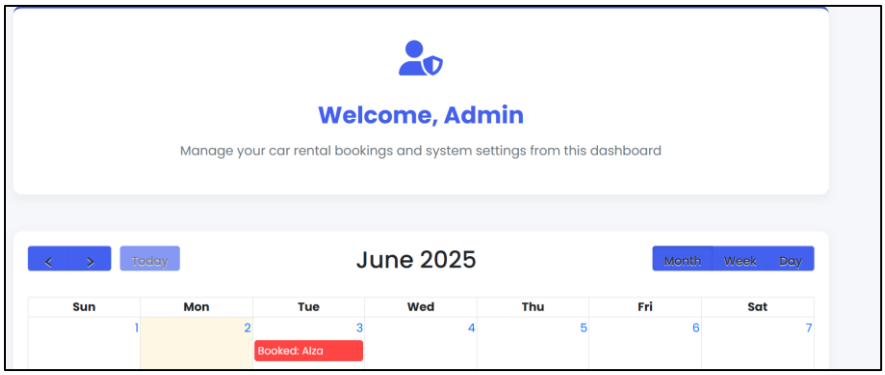


**Fig. 12** *Admin dashboard of TWB system*

## 5.8 Implementation of Communication Module

Let's delve into how the communication module was implemented. To make it easier for users to connect within the TWB system, we incorporated a floating action button. This button provides quick access to the communication features. It's designed to stay put in the bottom right corner of the screen, ensuring users can always easily open a chat window or find the contact form.

## 5.9 Implementation of Car Suggestion System

This section focuses on the car suggestion system. This particular module works to offer personalized vehicle recommendations based on the user's stated travel purpose, a detail gathered during their initial registration. By understanding the reason for their trip, the system can suggest cars that are most likely to fit what the user needs, thereby making the rental process more tailored and user-friendly. Figure 13 illustrates the algorithm used in the car suggestion system and Figure 14 demonstrates what this car suggestion system looks like when presented on the user's dashboard.

```
from surprise import Dataset, Reader, SVD, KNNBasic
from surprise.model_selection import train_test_split
import numpy as np

# Step 1: Load dataset
# Sample data format: user_id, car_id, rating
data = Dataset.load_from_df(car_ratings_df[['user_id', 'car_id', 'rating']], Reader(rating_scale=(1, 5)))
trainset, testset = train_test_split(data, test_size=0.2)

# Step 2: Train SVD
svd = SVD()
svd.fit(trainset)

# Step 3: Train KNN with cosine similarity
sim_options = {'name': 'cosine', 'user_based': True}
knn = KNNBasic(sim_options=sim_options)
knn.fit(trainset)

# Helper: Map raw ID to inner ID and back
def get_top_k_neighbors(user_id, k):
    inner_id = knn.trainset.to_inner_uid(user_id)
    neighbors = knn.get_neighbors(inner_id, k)
    raw_neighbors = [knn.trainset.to_raw_uid(inner_id) for inner_id in neighbors]
    return raw_neighbors

def get_user_latent_factors(svd, user_id):
    inner_id = svd.trainset.to_inner_uid(user_id)
    return svd.pu[inner_id]

def get_item_latent_factors(svd, car_id):
    inner_id = svd.trainset.to_inner_iid(car_id)
    return svd.qi[inner_id]

def predict_group_rating(group_user_ids, car_id):
    latent_vectors = [get_user_latent_factors(svd, uid) for uid in group_user_ids]
    group_mean = np.mean(latent_vectors, axis=0)
    item_factors = get_item_latent_factors(svd, car_id)
    predicted_rating = np.dot(group_mean, item_factors)
    return predicted_rating

# Example target user and car to recommend
target_user_id = 'U001'
car_id = 'C456'
group_user_ids = get_top_k_neighbors(target_user_id, k=3)
group_user_ids.append(target_user_id)

# Predict group rating
predicted_rating = predict_group_rating(group_user_ids, car_id)

# Evaluate SVD
from surprise import accuracy
predictions = svd.test(testset)
rmse = accuracy.rmse(predictions)

print(f"Predicted Group Rating for Car {car_id}: {predicted_rating:.2f}")
print(f"RMSE on test set: {rmse:.2f}")
```

**Fig. 13** *Algorithm for Car Suggestion System*

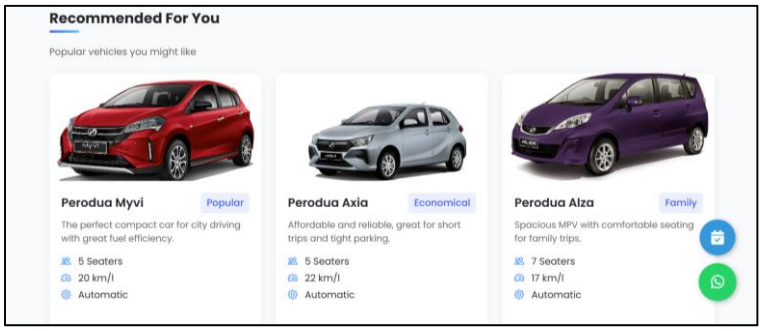


**Fig. 14** *Car suggestion system of TWB system*

## 5.10 Car Listing and Management Module

This section details the module that lets renters add their cars to the platform. Renters use it to input all the necessary information about their vehicles, like the make and model, when it's available for rent, the price, and photos. They also have full control to manage their listings, allowing them to easily edit, update, or even remove a car's details whenever necessary. This ensures customers always see accurate and current rental options. As shown in Figure 15, the system neatly displays all the cars a renter has listed, complete with their details, right on their personal dashboard.

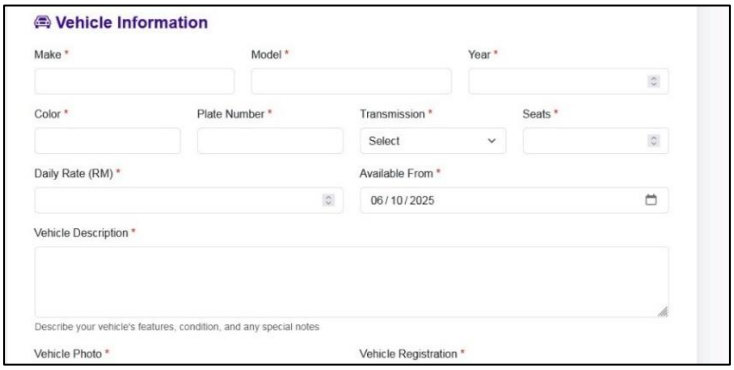


**Fig. 15** *Car Listing and Management of TWB system*

## 6. Results and Discussion

This chapter presents the outcomes of testing the proposed system. The two main types of results are shown: those from the formal test plan and feedback from user acceptance testing.

## 6.1 Test Plan Result

Table 2 details the findings from the functionality test plan, comparing what was expected to happen with what actually occurred.

**Table 2** *The functionality test plan*

| Check List | Expected Result | Actual Result |
|---|---|---|
| Register (All required input fields are in correct format) | Register is successful and would be directing lo login | Pass |
| Login (Input correct identity card and correct password) | Login successfully and would be directed to admin and user dashboard | Pass |
| Login (Input correct identity card and wrong password) | Login fail and message displayed “Invalid Identity Card or Password!” | Pass |
| Login (Input wrong identity card and wrong password) | Login fail and message displayed “Invalid Identity Card or Password!” | Pass |
| Delete car (Admin delete the car) | Confirmation deletion is displayed to confirm the action | Pass |
| Cancel booking (User cancel the booking) | Confirmation to cancel the booking is displayed | Pass |

**Table 3** *Security Test Plan*

| Check List | Actual Result |
|---|---|
| User and admin can register and login based on their role | Pass |
| User and admin need to be logged in with strong unique passwords | Pass |
| The process before logging out is saved | Pass |
| Password match with the requirements needed | Pass |
| Maximum length for identity card is specified | Pass |

## 6.2 User Acceptance Form Results for User Module

The results of the user acceptance test for the User Module indicate overwhelmingly positive feedback, particularly from the target users of Abbad Berkat Enterprise Sdn Bhd. Users successfully completed core actions such as registering and logging in, as well as uploading critical documents including driving licenses and identity cards. In addition, the vehicle booking process was reported to be smooth and user-friendly. The system's built-in security features designed to protect sensitive data were also well-received. Overall, users found the platform intuitive, functional, and aligned with their expectations and needs. All 10 respondents expressed satisfaction with key modules, including the registration, login, and data security features. The results confirm that the User Module effectively meets its intended goals and supports a positive user experience.

## 6.3 User Acceptance Form Results for Admin Module

The results from the user acceptance testing for the admin module were positive, aligning well with what the system administrator expected. It was proven that the module assists well in managing the system. The administrator was able to securely access the system and take care of important tasks, for example, creating and managing accounts, checking uploaded ID documents, deciding on reservations, and monitoring the system. They further observed that the interface was responsive, so it was simple to operate and navigate through the game. Besides, the advantages of restricted access and data validation were regarded as useful. On the whole, the admin module was praised and recognized for being effective for meeting the company's operations. As shown in Table 4, these are the administrator's responses from Abbad Berkat Enterprise Sdn Bhd.

**Table 4** *User Acceptance form result for Admin Module*

| | **Module** | **Result (1) Dissatisfied 1 – (5) Very Satisfied** |
|---|---|---|
| **Login Module** | | |
| 1 | Admin can login successfully | 5 |
| **Admin Dashboard Module** | | |
| 1 | Admin can view the schedule successfully | 4 |
| 2 | Admin can view the booking info successfully | 5 |
| 3 | Navigation buttons are functioning well | 5 |
| **Vehicle Management Module** | | |
| 5 | Admin can add car successfully | 4 |
| 6 | Admin can edit car successfully | 5 |
| 7 | Admin can delete car without any problem | 5 |
| **Overall System** | | |
| 8 | Admin can use the system without a problem | 4 |
| 9 | The interface is user-friendly | 5 |

## 7. Conclusion

All in all, the Traveler-Friendly Vehicle Rental Web-Based Application fulfilled its main task by creating a strong and handy platform for renting vehicles. The system makes renting easier for everyone by adding key features such as registration, putting documents online, secure access, control over bookings, and handling administrative work. Both the interaction and the security of the data are efficient on this network. The fact that the user acceptance testing had positive outcomes confirms both the system's efficiency and that it is liked by the people meant to use it, including Abbad Berkat's users.

Nevertheless, even with its positive points, there are still areas where the web application can improve. Some users could find it helpful to have a livelier user interface, as well as live customer support features. Using more advanced tools for tracking and reporting helps administrators oversee how the system is doing and affects their decision making.

Applying some changes could boost the efficiency of the system. This requires including an online payment system, sending instant notifications via text message or cell phone, upgrading the layout and ease of use, and making administration abilities include more statistics and detailed reports. Upcoming updates might include support for multiple languages and make the application work smoothly on any mobile device, making it accessible to many people. Focusing on these aspects will allow the web-based application to become more complete, expand, and concentrate on the user experience.

## Acknowledgement

I am very thankful to Dr. Rosmamalmi binti Mat Nawi for her important guidance, constructive feedback, and encouragement while this project was developed. Gratitude extends to Muhammad Khairul Adib bin Mohd Zulkeffli, the owner of Abbad Berkat Enterprise Sdn Bhd, for giving his assistance and providing real-world inputs to develop the system. We must also express our gratitude to everyone involved in system testing, since what they suggested and shared contributed to making the application more useful and reliable.

## Conflict of Interest

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

## Author Contribution

*The authors confirm contribution to the paper as follows:* ***study conception and design:*** *Nur Aina Natasha binti Hasan, Dr. Rosmamalmi binti Mat Nawi;* ***data collection:*** *Nur Aina Natasha binti Hasan****; analysis and interpretation of results:*** *Nur Aina Natasha binti Hasan, Dr. Rosmamalmi binti Mat Nawi;* ***draft manuscript preparation:*** *Nur Aina Natasha binti Hasan. All authors reviewed the results and approved the final version of the manuscript.*

## References

[1] A. C. C. & V. Rental and J. Wood, "History of car rental: what has changed?," *www.arnoldclarkrental.com*. https://www.arnoldclarkrental.com/latest-news/275-history-of-car-rental-what-has-changed

[2] M. Kamargianni, W. Li, M. Matyas, and A. Schäfer, "A Critical Review of New Mobility Services for Urban Transport," *Transportation Research Procedia*, vol. 14, pp. 3294–3303, 2016, doi: https://doi.org/10.1016/j.trpro.2016.05.277.

[3] Y. Duan, J. S. Edwards, and M. X. Xu, "Web-based expert systems: benefits and challenges," *Information & Management*, vol. 42, no. 6, pp. 799–811, Sep. 2005, doi: https://doi.org/10.1016/j.im.2004.08.005.

[4] A. Sahu, Avani Katlana, A. Sharma, M. Y. Baniya, Sumitra Sureliya, and Shubhrata Kanungo, "Vehicle Rental Management System: OnDemand Vehicle Ride," pp. 1–11, Sep. 2024, doi: https://doi.org/10.1109/acroset62108.2024.10743592.

[5] F. Y. H. Ahmed, M. al Thiruchelvam, and S. L. Fong, "Improvement of Vehicle Management System (IVMS)," *IEEE Xplore*, Jun. 01, 2019. https://ieeexplore.ieee.org/abstract/document/8825030/ (accessed Sep. 20, 2022).

[6] S. Durai, C. Shyamalakumari, and S. Florence, "Online High Priced Motor Vehicle Rental System," *Journal of Computational and Theoretical Nanoscience*, vol. 15, no. 9, pp. 2848–2852, Sep. 2018, doi: https://doi.org/10.1166/jctn.2018.7550.

[7] "Car Rental Malaysia | Drive From Only RM6/Hour - SOCAR," *socar.my*. https://www.socar.my/consumer

[8] "Car Rental in Malaysia | Book A Car Online - WAHDAH," *www.wahdah.my*. https://www.wahdah.my/

[9] "Cheap Car Hire, Compare Rental Prices - Rentalcars.com," *Rentalcars.com*, 2018. https://www.rentalcars.com/

[10] *Slite.com*, 2021. https://slite.com/en/learn/agile

[11] I. Ahmed, "Model for Monitoring Sustainable Green SLA Parameters in Modern Computing Industries," *International Journal of Computer Applications*, vol. 177, no. 38, pp. 27–32, Feb. 2020, https://doi.org/10.5120/ijca2020919876.

[12] Zhang, M., Xie, Y., Huang, L., & He, Z. (2013). Service quality evaluation of car rental industry in China. *International Journal of Quality & Reliability Management*, *31*(1), 82–102. https://doi.org/10.1108/ijqrm-11-2012-0146

[13] T *Ground Access to Major Airports by Public Transportation*. (2008). Transportation Research Board. https://doi.org/10.17226/13918
[14] Beatriz Brito Oliveira, Maria Antónia Carravilla, José Fernando Oliveira, Raicar, P., Acácio, D., Ferreira, J., & Paulo Araújo. (2015). Pricing for Internet Sales Channels in Car Rentals. *Studies in Big Data*, 139–147. https://doi.org/10.1007/978-3-319-24154-8_17
[15] NIMITHA ABOOBAKER. (2015). Innovative Strategies for Talent Management: a Case Study of Enterprise Rent-a-Car. *Mendeley*, *6*(10), 84–86. https://www.mendeley.com/catalogue/9a193f99-2132-35b1-a2da-eeac6cfa5c25
[16] The Economist. (2015, January 22). *Uber is now more popular than taxis or car rental with business people*. The Economist. https://www.mendeley.com/catalogue/4a878fc6-4daa-3a67-8312-95db1946dd95
[17] Williams, Z. (2018). Rideshare Companies Uber and Lyft Strongly Affect Taxi and Rental Car Industries. *Mendeley*, *4*(1), undefined-undefined. https://www.mendeley.com/catalogue/3e66c57a-0915-31e0-bd8f-0ffb876c75b2
[18] Lei, S., Wang, H., Yang, C., Du, B., Zhong, R., & Huang, R. (2017, August 1). *Forecasting car rental demand based temporal and spatial travel patterns*. IEEE Xplore. https://doi.org/10.1109/UIC-ATC.2017.8397484
[19] I. Lazov, "Profit management of car rental companies," *European Journal of Operational Research*, vol. 258, no. 1, pp. 307–314, Apr. 2017, doi: https://doi.org/10.1016/j.ejor.2016.08.064.

## Appendix A: Flowchart

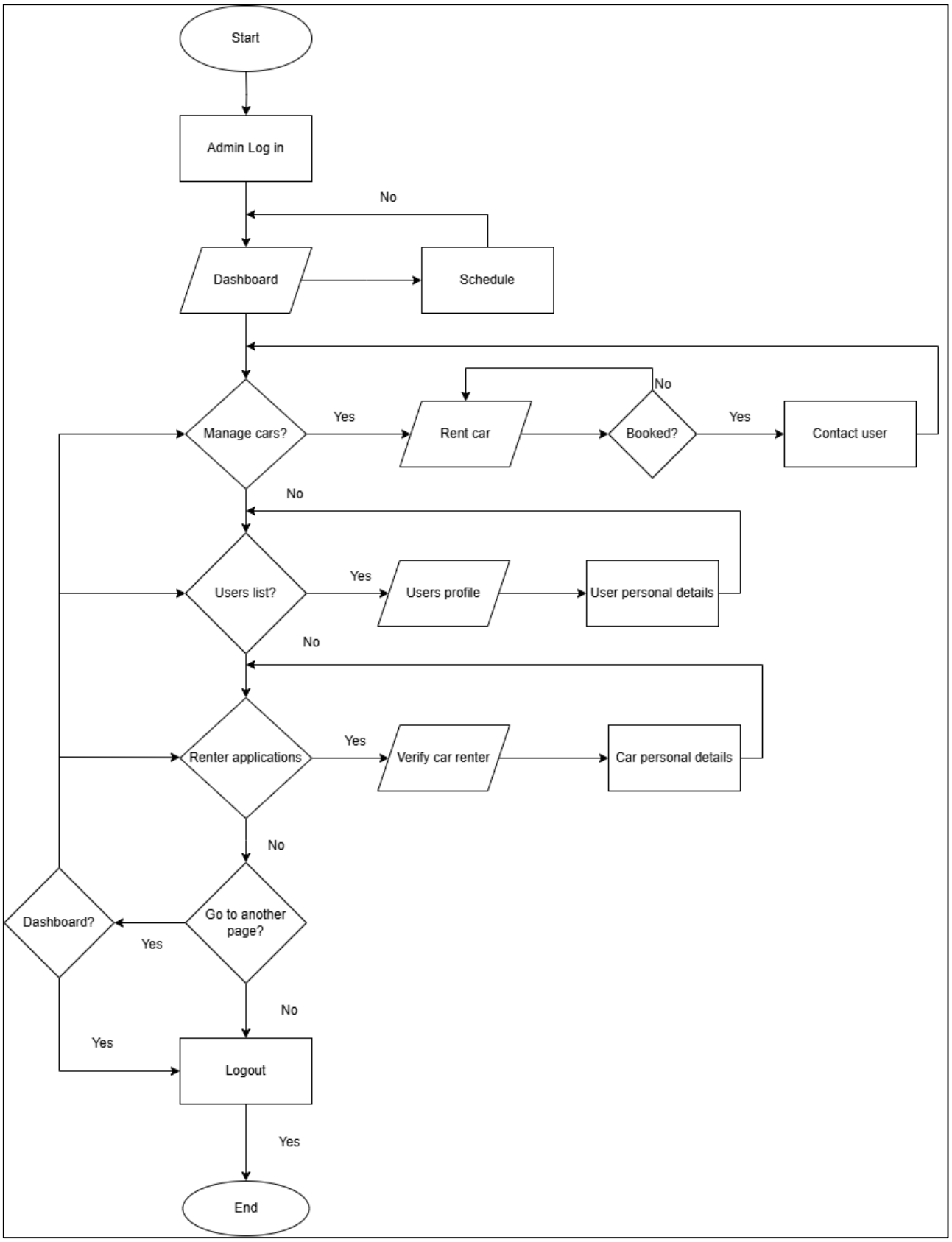


**Fig. 16** *Admin Flowchart of TWB system*

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 390-409

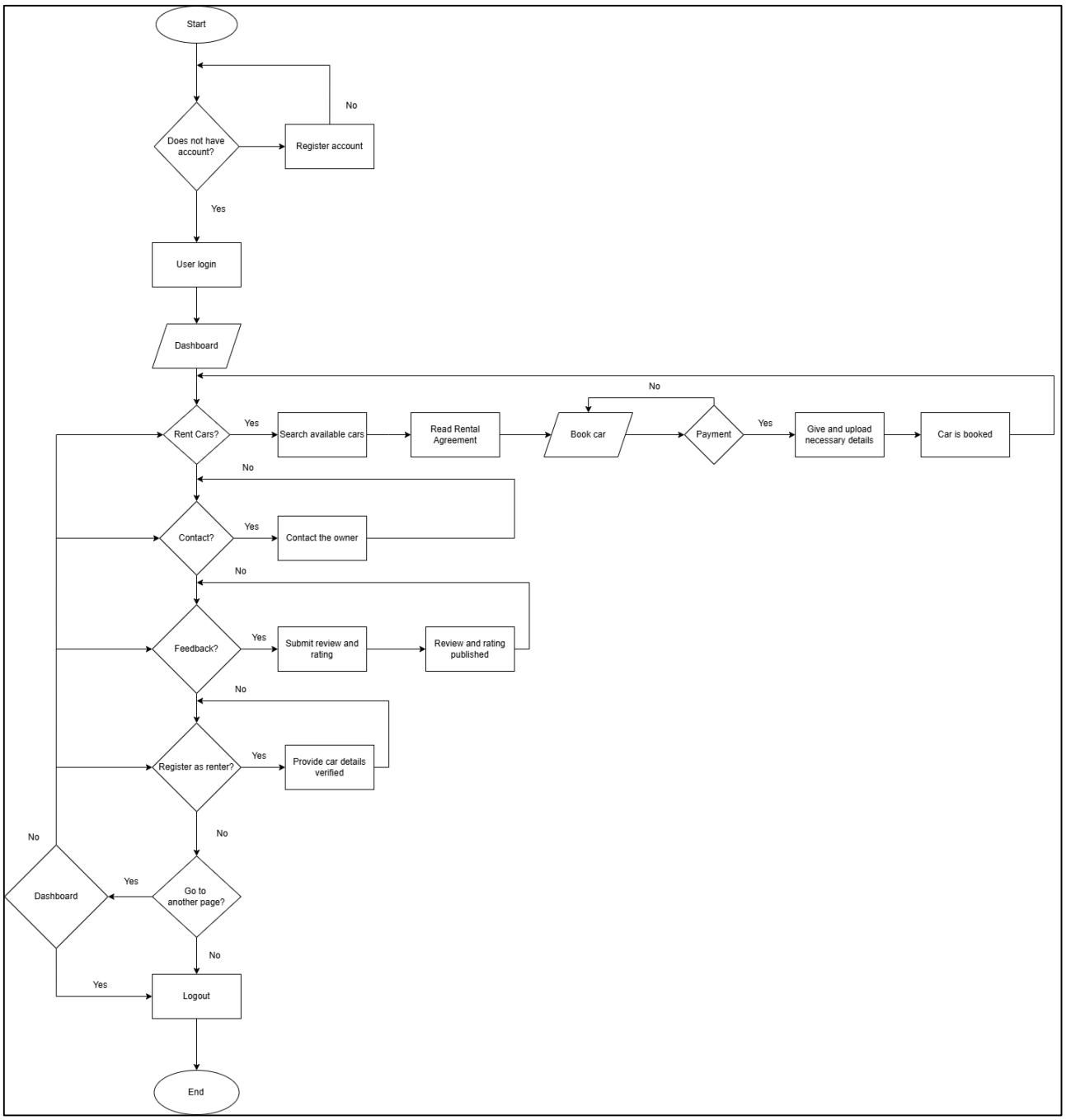


**Fig. 17** *Renter Flowchart of TWB system*

## Appendix B: Data Flow Diagram (DFD) Level 0

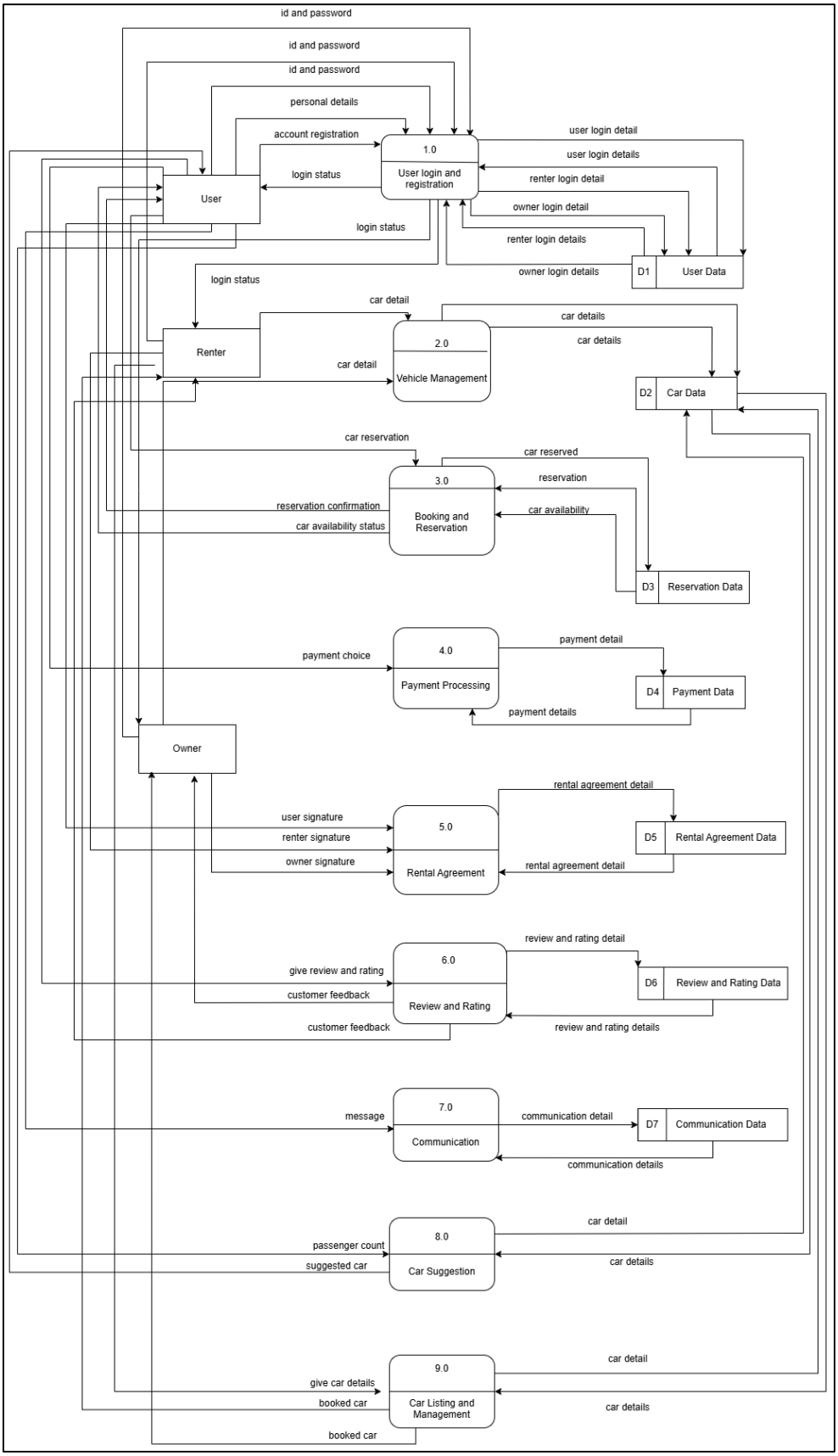


**Fig. 18** *Data Flow Diagram (DFD) Level 0*

## Appendix C: User Interface

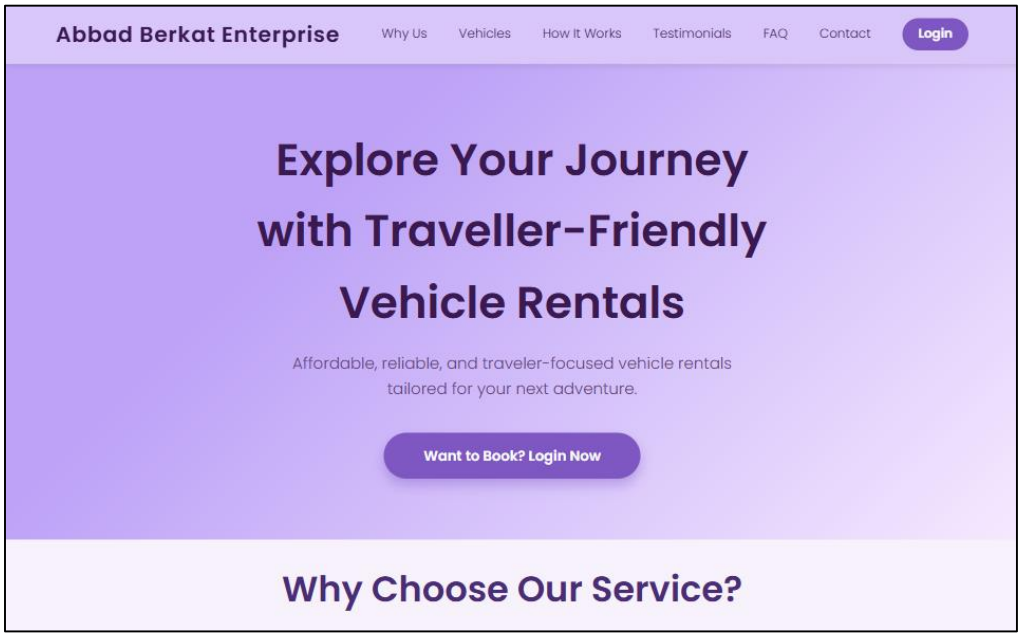


**Fig. 19** *Main home page interface of TWB system*

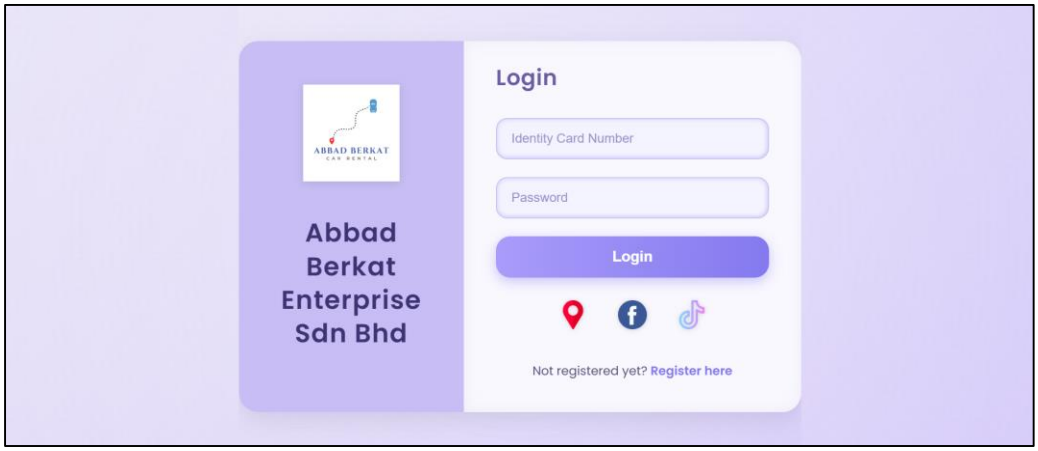


**Fig. 20** *The administrator and user login page of TWB system*

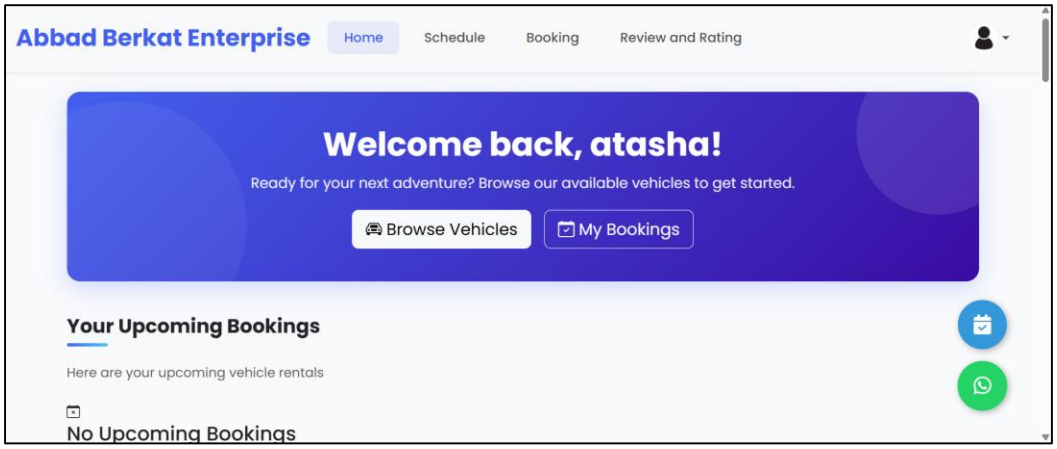


**Fig. 21** *The user dashboard of TWB system*