

# User Define Pre-Owned Vehicle Exchange Portal

## Author

### 1. Dr. K.ANANDHRAJ M.SC.,M.PHIL.,PH.D.P

Associate Professor

PG & Research Department of Computer Science

Sri Ramakrishna College of Arts & Science

Coimbatore – 641006

anandharaj@srcas.ac.in

### 2. Yuvaraj .M

PG & Research Department of Computer Science

Sri Ramakrishna College of Arts & Science

## Abstract

The User Friendly Pre Owned Vehicle Exchange Portal is a premium web application written in PHP 8, utilizing a handcrafted MVC architecture. This application is created from scratch to prove architectural proficiency, avoiding frameworks and opting for a custom-built router, controller, and security middleware. This application aims to resolve existing marketplace problems, such as an uncluttered user interface, verified user listings, and security risks such as SQL injection, cross-site scripting, and cross-site request forgery attacks. These security risks have been mitigated by implementing PDO Prepared Statements, Output Escaping, and IP Rate Limiting. This application has been designed with features such as an administrator verification system, which will reduce fraudulent activity, an internal messaging system, and a favorites system for buyers. This application has been technically designed with MySQL 8 for data management, Bootstrap 5.3, and Vanilla CSS 3 for the frontend. This application has been designed with a glass morphism theme, allowing for native dark and light modes through CSS variables, providing a premium, responsive, and easy-to-navigate user interface.

## Keywords

Used Vehicle Exchange Portal, PHP 8, MVC Architecture, MySQL Database, RBAC, Web Security, Bootstrap 5.3, SQL Injection Prevention.

## I. INTRODUCTION

User Friendly Pre-Owned Vehicle Exchange Portal

The User Friendly Pre-Owned Vehicle Exchange Portal is a premium web application that is safe and user-friendly, addressing the aforesaid challenges. This application has been built using a custom Model View Controller architecture, thus providing a high-end 'glass morphism' user experience. This application is a marketplace where trust is established through administrative intervention and safe communication.

Due to the increased use of the internet for business activities, online marketplaces have become an integral part of modern business. With the help of online exchange websites, buyers and sellers can easily interact with each other, which is not possible through traditional business practices. Although exchange websites are helpful, the lack of transparency, security, and verification is a major drawback of these websites, resulting in fraudulent activities.

This system aims to overcome these limitations by introducing a new web application architecture. This application has been built using the latest web development tools, including PHP 8, MySQL 8, and Bootstrap 5.3. This application has a custom MVC architecture, which is helpful in improving the

maintainability of the application by separating it into modules.II.

## II. LITERATURE REVIEW

The rapid growth of internet technology has significantly changed the way people buy and sell vehicles. Online vehicle exchange platforms have provided users with an opportunity to list, search, and buy used vehicles without the constraints of physical marketplaces. Various studies and existing systems have shown the significance of developing secure, efficient, and user-friendly web applications for online marketplaces.

Initially, web marketplace systems have been developed using basic server-side scripting languages like PHP. These systems have provided users with the opportunity to list their products and communicate with other users. However, the systems have failed to incorporate proper security features. Various web applications have been developed using basic database management systems. Due to the lack of proper security features, the systems have become vulnerable to various cyber attacks. Some of the major attacks include SQL injection, Cross-Site Scripting, and Cross-Site Request Forgery.

Today, web applications have been developed using structured architectural patterns. Various web applications have been developed using the MVC framework. MVC stands for Model-View-Controller. MVC separates the system into logical parts, making the system easier to maintain, scale, and debug. Various frameworks like Laravel and CodeIgniter have been developed for this purpose.

## III. PROPOSED SYSTEM

The methodology used in this project is based on the design and development of a secure and efficient web application using a custom PHP 8-based MVC framework for a used vehicle exchange portal. The overall development process is based on a structured approach, which includes system planning, requirement analysis, system architecture design, implementation, security integration, and testing.

### 1. System Planning and Requirement Analysis

The first phase of the methodology is based on the requirement analysis of the used vehicle exchange system. The system is designed to provide the following features to the users:

- Users can create their own accounts.
- Users can post their used vehicles.

### 2. System Architecture Design

The web application is developed based on the Model-View-Controller (MVC) architecture pattern.

- Model: Model is used to handle database operations and business logic.
- View: View is used to handle the user interface and presentation of the data to the user

#### .A. System Overview

The application, "Used Vehicle Exchange Portal," is a web-based application that aims to simplify the process of buying and selling used vehicles through a user-friendly interface. The application enables users to create accounts, list used vehicles, search for available used vehicles, and interact with other users who are selling used vehicles. The primary goal of this application is to create a trusted platform where users can buy and sell used vehicles, removing the limitations associated with traditional used vehicle trading.

The application has been created using the latest version of the PHP programming language, i.e., PHP 8, along with a custom Model-View-Controller architecture. Model-View-Controller is a design pattern that organizes the application into three parts, namely Model, View, and Controller. This design pattern improves the application by separating the application logic into three parts, which makes it easier to maintain, modify, and extend the application

A custom router has been used to route the URL requests to the appropriate controllers

#### B. Frontend Design and User Interface

The frontend of the User-Friendly Pre-Owned Vehicle Exchange Portal is designed to provide an intuitive, responsive, and visually appealing interface for users.

The primary objective of the frontend design is to enhance usability while ensuring that users can easily browse vehicle listings, communicate with sellers, and manage their accounts. The interface is developed using modern web technologies including **HTML**, **CSS3**, **JavaScript**, and the responsive framework **Bootstrap 5.3**.

The platform adopts a responsive design approach to ensure compatibility across multiple devices such as desktops, tablets, and smartphones. The grid system provided by Bootstrap enables flexible layout management, allowing the application to dynamically adjust its interface according to screen size and device resolution. This responsiveness improves accessibility and ensures a consistent user experience across different platforms.

A distinctive visual feature of the system is the implementation of a **glass-morphism user interface design**. This design style incorporates translucent layers, blurred backgrounds, and subtle shadow effects to create a modern and aesthetically pleasing interface. Additionally, the system supports both light and dark themes through the use of CSS variables, enabling users to switch themes based on their preferences and environmental lighting conditions.

The homepage and navigation structure are designed to minimize complexity and improve user interaction. A structured navigation bar provides quick access to essential features such as vehicle browsing, listing management, favorites, and user messaging. This design ensures that users can easily locate and access the required functionalities without unnecessary navigation steps.

Vehicle listings are displayed using a card-based layout that presents key information including vehicle images, price, specifications, and seller details. This structured presentation allows users to quickly compare different vehicles and identify relevant listings. Furthermore, advanced filtering and search mechanisms are integrated into the interface, enabling users to refine search results based on parameters such as brand, price range, location, and vehicle condition.

The frontend also incorporates interactive features such as a favorites system that allows buyers to bookmark preferred vehicles for future reference. Additionally, a secure messaging interface is implemented to facilitate communication between

buyers and sellers within the platform, ensuring privacy and protecting personal contact information.

An administrative user interface is also provided for system administrators. This dashboard allows administrators to review submitted vehicle listings, approve or reject advertisements, and monitor user activities. The dashboard interface is designed with structured data tables and management controls to support efficient administrative operations.

Overall, the frontend design emphasizes usability, responsiveness, and modern visual aesthetics. By integrating contemporary user interface principles with responsive web technologies, the platform delivers a seamless and engaging experience for all users interacting with the vehicle exchange portal.

## C. Backend System

The backend plays a crucial role in handling all The backend system of the User-Friendly Pre-Owned Vehicle Exchange Portal is responsible for managing the application logic, database management, security, and interaction with the user interface and the database. The backend is created using the PHP 8 programming language with a custom implementation of the Model-View-Controller pattern. This pattern is useful in maintaining the separation of the code into logical layers, thus improving the maintainability of the codebase.

The application uses a custom implementation of the MVC pattern, unlike other applications that use heavy frameworks. This custom MVC framework is useful in managing the HTTP requests that are received by the application. The custom router processes the HTTP request and directs it to the appropriate controller. The controller is responsible for managing the interaction with the user, including authentication, managing the vehicle listing, messaging, and other operations.

The backend system of the User-Friendly Pre-Owned Vehicle Exchange Portal is responsible for managing the application logic, database management, security, and interaction with the user interface and the database. The backend is created using the PHP 8 programming language with a custom implementation of the Model-View-Controller pattern. This pattern is useful in maintaining the separation of the code into

logical layers, thus improving the maintainability of the codebase.

The application uses a custom implementation of the MVC pattern, unlike other applications that use heavy frameworks. This custom MVC framework is useful in managing the HTTP requests that are received by the application. The custom router processes the HTTP request and directs it to the appropriate controller. The controller is responsible for managing the interaction with the user, including authentication, managing the vehicle listing, messaging, and other operations.

### E. Benefits of the Proposed System

The proposed User-Friendly Pre-Owned Vehicle Exchange Portal has various advantages over the conventional method of trading vehicles and the conventional online trading platforms. This is because the proposed platform is designed using the latest technologies in web development and secure coding practices, which provide a secure environment for trading vehicles.

#### 1. Structured and Maintainable Architecture

The system is designed in such a way that it follows a custom Model View Controller (MVC) architecture using **\*\*PHP 8\*\***. This architecture divides the system into independent components that handle the data, the user interface, and the logic of the application separately. This improves the readability of the code and the maintainability of the system, making it easier for the developer to extend the system with more features.

## IV. IMPLEMENTATION AND RESULTS

The implementation of the User Friendly Pre-Owned Vehicle Exchange Portal is based on the integration of modern web technologies along with a structured system architecture. The application is developed using the latest version of the PHP programming language, namely, PHP 8, which is based on the custom Model View Controller architecture. In the custom MVC architecture, the entire system is divided into three parts: the model, the view, and the controller. Such an architecture helps to improve the maintainability of the system.

The frontend of the system is developed using HTML5, CSS3, JavaScript, and the responsive framework Bootstrap 5.3. Such an implementation

helps the system to improve the user experience, making the system responsive across various devices such as desktop, tablet, and mobile phones. Moreover, the user interface of the system is based on the glass morphism theme, which helps to improve the aesthetics of the system.

## V. SECURITY ANALYSIS

The aspect of security is vital in the development of the User Friendly Pre-Owned Vehicle Exchange Portal, especially since it deals with user information and vehicle details. To ensure the reliability and safety of the application, various security features are integrated into the system to ensure it is safe from various web application vulnerabilities. One of the vulnerabilities that a web application might face is SQL Injection, where an attacker tries to inject malicious SQL statements into the database using user input. To ensure the application is safe from this vulnerability, the system uses PHP Data Objects, which separates SQL commands from user input. Another vulnerability that might affect the application is Cross Site Scripting, which enables an attacker to inject malicious scripts into web pages. This vulnerability has been addressed by incorporating input validation, which ensures that all user input is sanitized before being echoed to the user. Additionally, the system has been designed to protect it from Cross Site Request Forgery, which involves the use of CSRF tokens in all forms submitted to the application.

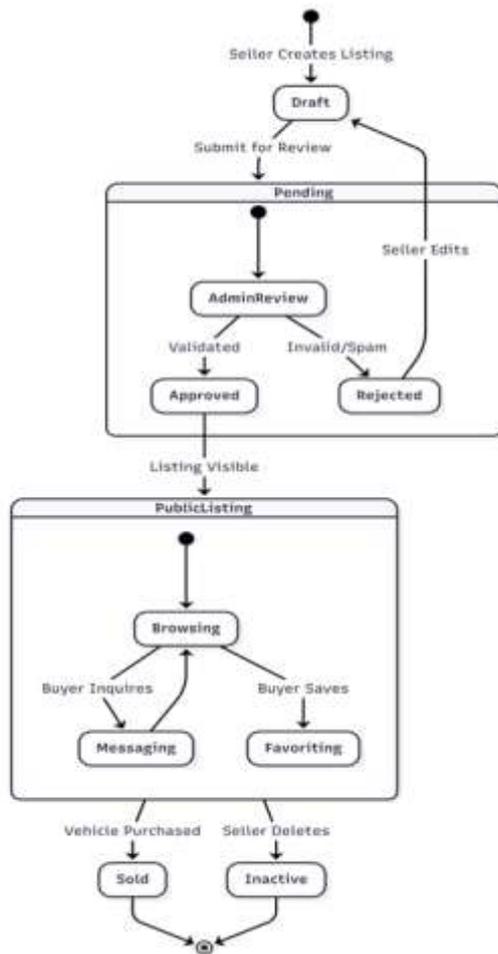
## V. CONCLUSION

The User Friendly Pre-Owned Vehicle Exchange Portal is an efficient and secure way of purchasing and selling pre-owned vehicles in a virtual environment. The system is developed using PHP custom Model-View-Controller architecture, which ensures a well-designed and maintainable system. By dividing the system into models, views, and controllers, the system is well organized and enhances the maintainability of the system.

The system incorporates the latest frontend tools, including Bootstrap 5.3, HTML, CSS3, and JavaScript, to develop a user-friendly and interactive user interface. By implementing the glass morphism design style, the system is user-friendly and accessible on different devices.

Security is one of the main features of the system, and several security measures are taken to protect the user data and system functionality. PHP Data Objects Prepared Statements are used to protect the database from SQL Injection attacks, and input validation and output escaping are used to protect the system from Cross-Site Scripting (XSS) attacks. Furthermore, token-based systems are used to protect the system from Cross-Site Request Forgery (CSRF) attacks.

## VI. FLOW DIAGRAM



## VII. REFERENCES

[1] Martin Fowler, Patterns of Enterprise Application Architecture. Boston, MA: Addison-Wesley, 2002.

[2] Rasmus Lerdorf, "Introduction to PHP and Modern Web Development," PHP Documentation, 2023.

[3] Oracle Corporation, MySQL 8 Reference Manual, Oracle, 2023.

[4] OWASP Foundation, "OWASP Top 10: The Ten Most Critical Web Application Security Risks," 2021.

[5] Bootstrap Team, Bootstrap 5.3 Documentation, 2023.

[6] PHP Group, "Database Access with PHP Data Objects (PDO)," PHP Official Documentation, 2023.

[7] Ian Sommerville, Software Engineering, 10th ed. Boston, MA: Pearson Education, 2016.

[8] Robert C. Martin, Clean Architecture: A Craftsman's Guide to Software Structure and Design. Boston, MA: Prentice Hall, 2017.