iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** X **Month of publication:** October 2026

DOI: https://doi.org/10.22214/ijraset.2026.85543

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue X Oct 2026- Available at www.ijraset.com

Personal Vehicle Maintenance and Service Management System - A Web-Based Platform for Vehicle Maintenance, Repair Booking, and Service Management

Mr. S. Vinoth Kumar[1], Srisha R[2], Sudharsan M[3], Yeshwanth Kumar V[4]
[1]*Assistant Professor,* [2, 3, 4]*B.E. Students, Department of Computer Science and Engineering, Adhiyamaan College of Engineering (An Autonomous Institution), Dr. M.G.R. Nagar, Hosur – 635130, Tamil Nadu, India*

Abstract: ***The Personal Vehicle Maintenance and Service Management System is a full-stack web application developed to simplify and digitalize the vehicle servicing process. The system allows vehicle owners to register, add their vehicles, and book maintenance or repair services online without the need to search for nearby mechanics manually. It supports various types of vehicles, including two-wheelers, cars, electric vehicles and other heavy vehicles. After a service request is submitted, the assigned mechanic can accept the request, arrange vehicle pickup, update the service progress, record repair details, generate an electronic bill (E-Bill) with a QR code for bill verification, and mark the service as completed. Emergency Roadside Assistance saves emergency details offline and automatically sends them when the internet is restored. Customers can track the real-time status of their vehicles, make secure online payments, download invoices, and provide ratings and reviews based on their service experience. The system also maintains vehicle service history, maintenance schedules, and reminder notifications for future servicing. The application includes three user roles: Customer, Mechanic, and Administrator. The Administrator manages users, mechanics, service requests, vehicle categories, and system reports, ensuring efficient operation of the platform. The system is developed using HTML5, CSS3, JavaScript, Bootstrap 5, Python (Flask), and MySQL. The proposed solution reduces the time and effort required to find reliable vehicle service providers, improves communication between customers and mechanics, promotes paperless billing through electronic invoices, and enhances service transparency by providing real-time status updates. The project offers a practical, user-friendly, and scalable solution for modern vehicle maintenance management, including support for electric vehicles***.

Index Terms: Vehicle Maintenance, Flask, MySQL, Online Service Booking, E-Billing,Roadside Assistance,Electric Vehicle

I. . INTRODUCTION

The Personal Vehicle Maintenance and Service Management System is a web-based application designed to simplify and digitalize vehicle servicing. Traditional vehicle maintenance methods often involve difficulties in finding reliable mechanics, booking appointments, tracking repairs, and maintaining service records. These activities can consume time and create communication gaps between customers and service providers.

The proposed system provides a centralized platform connecting customers, mechanics, and administrators. Customers can register their vehicles, book maintenance and repair services online, track service progress, make online payments, and download electronic invoices. The system supports two-wheelers, cars, electric vehicles, and heavy vehicles.

Mechanics can manage service requests, arrange vehicle pickup, update repair progress, and generate electronic bills with QR codes for verification. Additional features include emergency roadside assistance with offline request storage, maintenance reminders, service history management, and customer ratings and reviews.

The application is developed using HTML5, CSS3, JavaScript, Bootstrap 5, Python Flask, and MySQL. The main objective is to reduce manual work, improve communication, promote paperless billing, and provide a convenient, transparent, and efficient vehicle maintenance experience.

The system integrates an AI-powered chatbot that helps customers understand vehicle problems, provides basic maintenance suggestions, and estimates repair costs. Customers can also find nearby mechanics, compare available services, and select a suitable mechanic based on their location, improving convenience and supporting informed service decisions.

©IJRASET: All Rights are Reserved

II. LITERATURE SURVEY

1) T. MATHEEBAN (2017)The paper titled *Web-Based Vehicle Service Management System for Trinco Automobile*, authored by T. Matheeban in 2017 as an Undergraduate Dissertation at the University of Colombo School of Computing, focuses on improving automobile service center operations through web-based technologies. It provides a foundation for understanding online service booking and the management of customer and vehicle information. The proposed system extends this concept by integrating service tracking, mechanic management, electronic billing, and emergency roadside assistance.
2) TIAGO ZONTA ET AL. (2020)The research paper titled *Predictive Maintenance in the Industry 4.0: A Systematic Literature Review*, authored by Tiago Zonta, Cristiano André da Costa, and Rodrigo da Rosa in 2020 and published in Computers & Industrial Engineering, discusses digital technologies and data-driven approaches for maintenance planning. It highlights the importance of organized maintenance information, monitoring systems, and predictive maintenance techniques. These concepts are relevant to the proposed system's service history management, scheduled maintenance reminders, and digital vehicle maintenance records.
3) FABIO ARENA ET AL. (2022) The research paper titled *Predictive Maintenance in the Automotive Sector: A Literature Review*, authored by Fabio Arena, Mario Collotta, Liliana Luca, and Marianna Ruggieri in 2022 and published in Machines, examines various methods and technologies used to improve vehicle maintenance and identify potential failures. It emphasizes the importance of systematic vehicle monitoring, maintenance planning, and early fault identification. The proposed system applies related concepts through digital service records, maintenance reminders, and AI-powered vehicle maintenance suggestions.

Table I. Summary of surveyed work and identified gap

Ref.	Focus	Limitation addressed by the proposed system
[1]	Web-based vehicle service management	Limited service tracking and mechanic management.
[2]	Predictive maintenance using Industry 4.0	No integrated online service booking.
[3]	Automotive predictive maintenance	Limited digital maintenance reminders
[4]	ML-based vehicle fault diagnosis	No integrated customer service platform.
[5]	Garage management web application	Limited AI-based assistance.
[6]	Automobile service management	Limited online payment and digital billing.
[7]	AI-based vehicle fault diagnosis	No complete service management facilities.
[8]	Vehicle damage detection and cost estimation	Limited mechanic booking and service tracking.
[9]	Automated vehicle fault diagnosis	No integrated emergency roadside assistance.
[10]	AI-based vehicle maintenance assistance	Limited location-based mechanic searching.

Existing systems focus on vehicle servicing, predictive maintenance, and fault diagnosis but lack complete service integration. The proposed system combines online booking, AI assistance, digital billing, service tracking, and emergency support in one platform.

III. EXISTING SYSTEM

In the traditional vehicle servicing system, customers generally contact mechanics through telephone calls or visit service centers directly to book maintenance and repair services. Service details and customer information are maintained using registers, paper documents, or separate applications. Customers often need to contact mechanics repeatedly to obtain updates about their vehicles, while manual billing and record maintenance create difficulties in retrieving previous service information and managing vehicle maintenance efficiently.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

IV. PROPOSED SYSTEM

A. *Overview*

The proposed system provides a centralized online platform for managing vehicle maintenance and repair services, allowing customers to register their vehicles, book services, track repair progress, make online payments, and download electronic invoices. Mechanics can accept assigned service requests, arrange vehicle pickup, update repair details, and generate bills, while administrators can manage users, mechanics, service requests, vehicle categories, and system reports efficiently.

B. *Proposed Solution*

The proposed solution is developed as a full-stack web application using HTML5, CSS3, JavaScript, Bootstrap 5, Python Flask, and MySQL. The frontend provides an interactive interface for customers, mechanics, and administrators, while Flask handles authentication, service requests, business logic, and database operations. MySQL stores user details, vehicle information, bookings, repair records, invoices, payment details, and reviews. When a customer submits a service request, the application validates and stores the information in the database, allowing the assigned mechanic to process the request and update its status. Customers can easily view the latest service progress and access their vehicle maintenance information.

C. *Ideation and Brainstorming*

The project idea was developed by identifying the common problems faced by vehicle owners and serviceproviders. The major concerns included difficulty in booking services, limited repair updates, manual billing, and poor maintenance record management.

D. *Problem–Solution*

Problem: Traditional vehicle servicing involves time-consuming booking, lack of repair updates, poor maintenance record management, paper-based billing, difficulty verifying bills, network interruptions during emergencies, missed maintenance schedules, limited customer feedback, and difficult service administration.

Proposed Solution: The proposed system provides online service booking, service status tracking, a centralized database, electronic invoices, QR code verification, offline storage and synchronization, maintenance reminder notifications, customer ratings and reviews, and an administrator dashboard to overcome these problems efficiently.

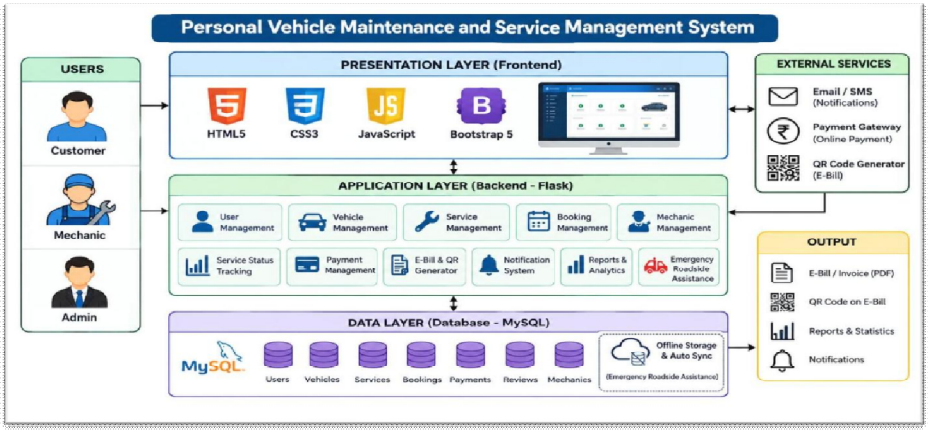


Fig. 3: System Architecture of Personal Vehicle Maintenance and Service Management System

Table II. Existing approach versus proposed system

Aspect	Existing approach	Proposed system
Service Booking	Manual booking through phone calls or direct visits	Online service booking
Service Tracking	Manual enquiry with mechanics	Online service status tracking
Billing	Paper-based billing	Electronic billing with QR code
Notifications	Manual communication	Email and SMS notifications
Emergency Assistance	Direct phone communication	Online roadside assistance

V. IMPLEMENTATION

Implementation is the process of converting the system design into a working application. The Personal Vehicle Maintenance and Service Management System is developed using HTML5, CSS3, JavaScript, Bootstrap 5, Python Flask, and MySQL. The frontend provides forms and dashboards for different users, while Flask processes submitted information and communicates with the database. MySQL stores all necessary records related to service booking, vehicle details, repair tracking, billing, and administration. The following sections explain the main implementation modules of the system.

A. User Registration

The user registration module allows new users to create accounts by entering their name, email address, phone number, password, and user role. The application validates the submitted information by checking required fields, email format, and duplicate accounts. Passwords are securely hashed before being stored in the MySQL database. After successful registration, a confirmation message is displayed, and the user can log in using their credentials to access the appropriate dashboard.

B. Vehicle Registration Module

The vehicle registration module enables customers to store their vehicle details in the application for future service bookings. The registration form includes vehicle registration number, manufacturer, model, vehicle type, year of manufacture, fuel or power type, and additional notes. After logging in, the customer enters the vehicle details, which are validated by Flask and stored in the MySQL database. The registered vehicle is displayed in the customer's vehicle list and can be selected while creating a service request. The application ensures that customers can access and modify only their authorized vehicle records.

C. Online Service Booking Module

The online service booking module allows customers to request vehicle maintenance or repair services through the application, reducing the need for manual appointment scheduling. After logging in, the customer selects a registered vehicle, chooses the required service category, describes the vehicle problem, and enters the preferred appointment details. Once the booking is submitted, the Flask application validates the entered information and stores the booking details in the MySQL database with a unique booking ID. Finally, the system displays a booking confirmation along with the current service status, allowing customers to track their requests easily.

D. Mechanic Service Management Module

The mechanic service management module allows mechanics to manage vehicle repair activities through the application. After logging in, the mechanic can view the assigned service requests, review the customer and vehicle details, and accept the requests. The mechanic then inspects the vehicle, identifies the problem, records the inspection results, and updates the service status. Repair details, required spare parts, labour charges, and other expenses are also entered into the system. These details are stored in the MySQL database and used for generating the final electronic invoice.

E. E-Billing and QR Code Verification Module

The electronic billing module generates invoices using the service and charge details stored in the database. After the mechanic completes the repair work and records the spare parts cost, labour charges, and other applicable expenses, the application retrieves the booking and repair details, calculates the total amount, and generates an invoice with a unique invoice number. The invoice includes the booking ID, vehicle details, repair description, applicable charges, and total amount. It is saved in the MySQL database and displayed to the customer for viewing, downloading, or printing, if these features are available. The system also generates a QR code containing a unique invoice identifier or a verification URL. When scanned, the QR code opens the invoice verification page, which checks the identifier against the stored invoice record and confirms its validity. This makes invoice retrieval easier and improves billing transparency without exposing sensitive customer or payment information.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

F. Emergency Roadside Assistance Module

The emergency roadside assistance module allows users to submit requests during unexpected vehicle breakdowns or urgent service situations. The user opens the roadside assistance page and enters the vehicle registration number, contact details, breakdown location, and a description of the issue. The application validates the request details and checks the network connection. If the device is online, the request is submitted directly to the server and stored in the MySQL database. If offline functionality is implemented, the request is temporarily stored in the browser's local storage and submitted automatically when the internet connection is restored. The system identifies pending requests and prevents duplicate submissions during synchronization. After successful submission, the request status is displayed to the user, helping the service team respond to emergencies efficiently.

G. Software Description

The software components work together to provide the complete functionality of the application. HTML5 is used to structure web pages and forms, while CSS3 provides styling and layout. JavaScript supports interactive features, and Bootstrap 5 helps create responsive pages that work on different screen sizes. Python Flask is used as the backend technology to handle application routing, request processing, data validation, user authentication, service booking, repair updates, and invoice-related operations. MySQL is used as the database technology to store and manage important system information, including user details, vehicle information, mechanic details, service bookings, service records, invoices, payment details, and roadside assistance requests. These technologies are integrated to provide an efficient and user-friendly vehicle maintenance and service management system.

VI. SYSTEM REQUIREMENTS

The system requires a computer with an Intel Core i3 processor, a minimum of 4 GB RAM, and 250 GB of storage. Windows 10/11 or Linux can be used as the operating system. HTML5, CSS3, JavaScript, and Bootstrap 5 are used for frontend development. Python Flask is used for backend development, and MySQL is used for database management. Visual Studio Code is used as the code editor, and Google Chrome or Microsoft Edge is used to access the application. A stable internet connection is required for online booking, service tracking, and other application features.

VII. RESULTS AND DISCUSSION

Fig.5.9.1–CustomerDashboard**:**The Customer Dashboard displays registered vehicles, service bookings, completed services, and cancelled requests. Customers can book services, track repair progress, view service history, check reminders, and download invoices.**5**.9.5–MechanicDashboard**:**The Mechanic Dashboard displays assigned service requests, available jobs, repair status, and pickup schedules. Mechanics can manage service requests, update repair progress, and handle roadside assistance efficiently.

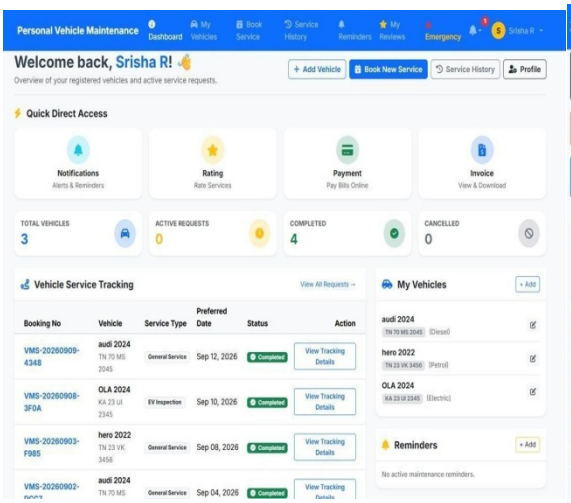


Fig.5.9.1:Customer Dashboard:

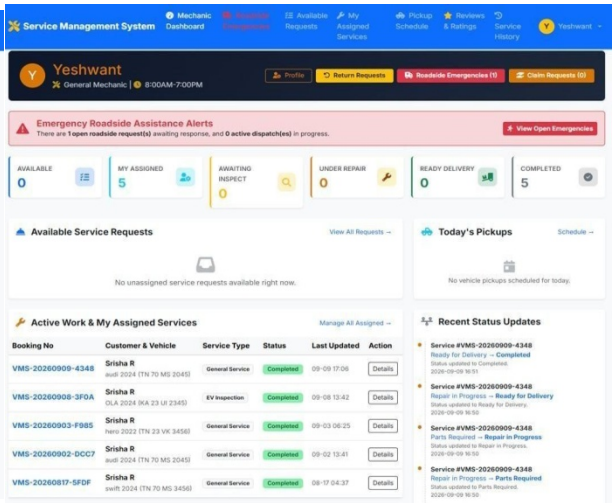


Fig. 5.9.5: Mechanic Dashboard:

Fig.5.9.9–CustomerReviewsandFeedback**:**This module displays customer ratings and feedback about the services provided. Mechanics can view customer reviews, check rating details, filter feedback, and respond to pending reviews.Fig. 5.9.12 – EODAmountReceivedFinancialReport**:**This admin-only module displays the total amount received at the end of the day.

It provides payment details based on different payment methods, including UPI, credit card, debit card, and cash, along with transaction details and payment status.

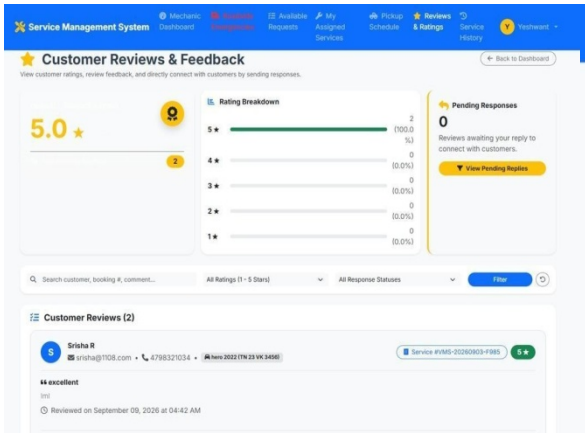


Fig. 5.9.9: Customer Reviews and Feedback:

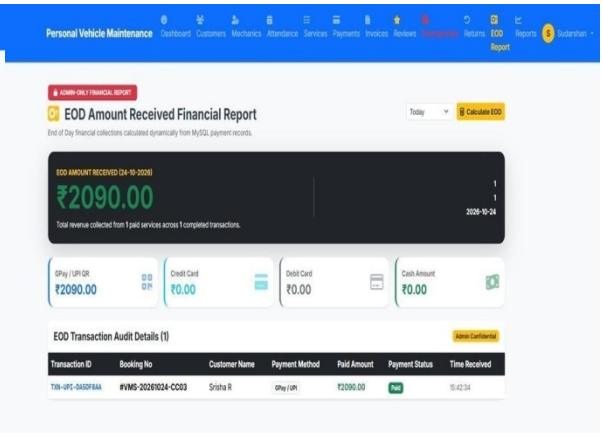


Fig. 5.9.12: EOD Amount Received Financial Report:

VIII. CONCLUSION AND FUTURE SCOPE

The Personal Vehicle Maintenance and Service Management System provides a web-based solution for organizing vehicle maintenance and repair activities. It simplifies the interaction between customers, mechanics, and administrators by integrating essential service-related functions into a single application. The system supports customer registration, vehicle management, online service booking, mechanic service management, service status tracking, maintenance records, electronic billing, QR code-based invoice verification, and emergency roadside assistance. It is developed using HTML5, CSS3, JavaScript, Bootstrap 5, Python Flask, and MySQL to provide a responsive and database-driven application. Centralized data storage helps manage customer, vehicle, booking, repair, and billing information efficiently. Overall, the project reduces dependence on manual records, improves communication, and makes vehicle servicing more convenient and transparent. The effectiveness of the system depends on the successful implementation and testing of its features.

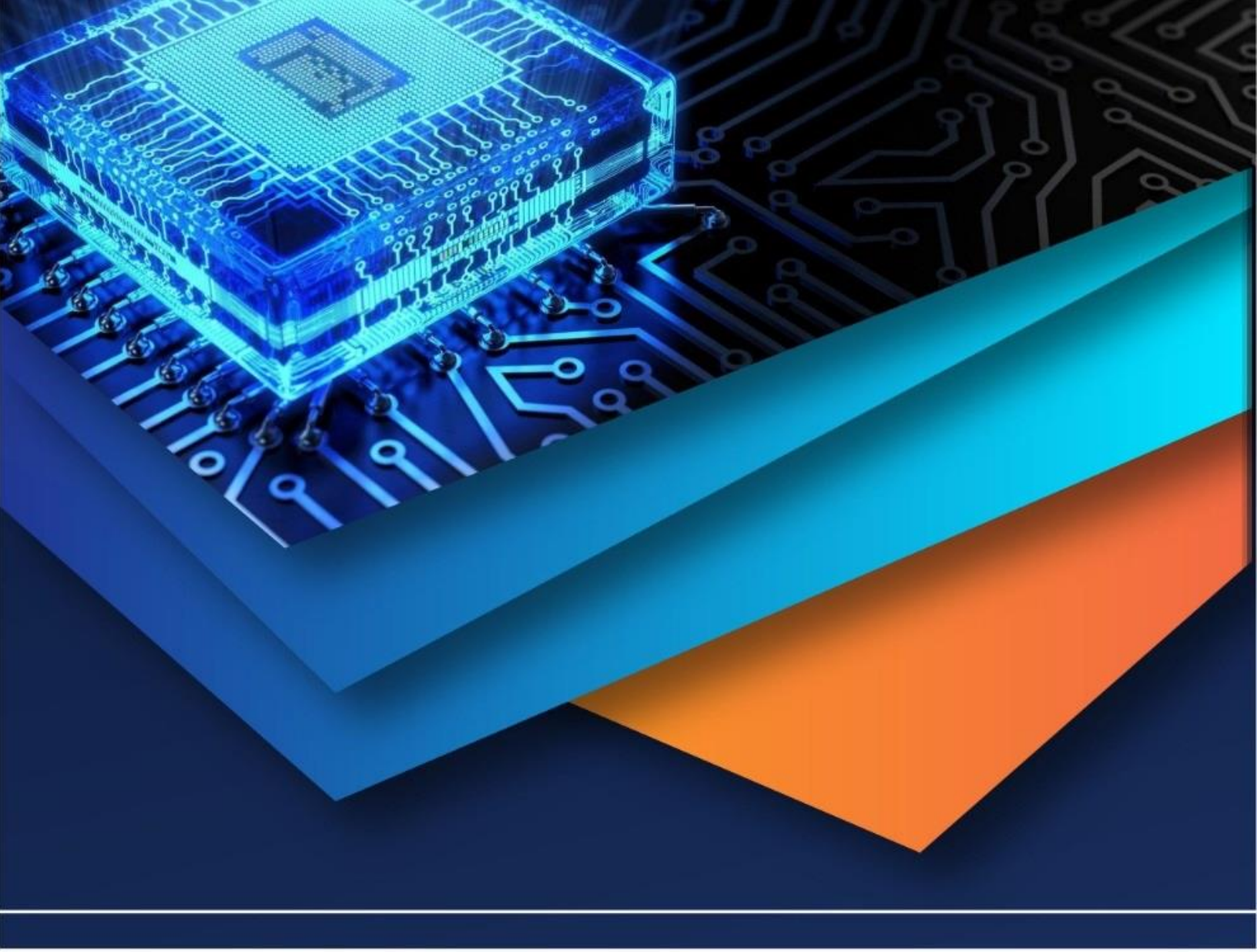
IX. ACKNOWLEDGMENT

The authors thank Dr. R. Radhakrishnan, Principal, and Dr. G. Fathima, Professor and Head, Department of Computer Science and Engineering, Adhiyamaan College of Engineering, Hosur, for the opportunity and encouragement, and the project coordinator Mr.S.Vinoth Kumar AP\CSE and faculty of the department for their support.

REFERENCES

[1] Python Software Foundation, Python Documentation. Python Documentation
[2] Pallets Projects, Flask Documentation. Flask Documentation
[3] Oracle, MySQL Reference Manual. MySQL Documentation
[4] Bootstrap Team, Bootstrap Documentation. Bootstrap Documentation
[5] Mozilla Developer Network, HTML: HyperText Markup Language. HTML Documentation
[6] Mozilla Developer Network, CSS: Cascading Style Sheets. CSS Documentation
[7] Mozilla Developer Network, JavaScript Guide. JavaScript Documentation
[8] Python Package Index (PyPI), qrcode Package. QR Code Python Package
[9] Pallets Projects, Flask-SQLAlchemy Documentation. Flask-SQLAlchemy Documentation
[10] Pallets Projects, Flask-Login Documentation. Flask-Login Documentation
[11] Flask-Login Developers, Flask-Login Documentation. Flask-Login Documentation
[12] Python Software Foundation, Python Package Index (PyPI). Python Package Index
[13] Oracle, MySQL Workbench Manual. MySQL Workbench Documentation
[14] Werkzeug Team, Werkzeug Documentation. Werkzeug Documentation
[15] Jinja Team, Jinja Template Designer Documentation. Jinja Documentation
[16] MDN Web Docs, Fetch API. Fetch API Documentation
[17] MDN Web Docs, Geolocation API. Geolocation API Documentation
[18] Google Developers, Google Maps Platform Documentation. Google Maps Platform Documentation
[19] Python Package Index (PyPI), Requests: HTTP for Humans. Requests Documentation
[20] Python Package Index (PyPI), python-dotenv Documentation. python-dotenv Documentation

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)