iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: V Month of publication: May 2025

DOI: https://doi.org/10.22214/ijraset.2025.69962

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Scalable Parking Management Using Cloud Infrastructure and Devops

Kedar Pattanshetti[1], Vedant Pawar[2], Pornima Vibhute[3], Tejaswini Dalavi[4], Prof. Sheetal Kapse[5]
[1, 2, 3, 4]*Department of Computer Engineering, Smt. Kashibai Navale College Of Engineering, Pune, India*
[5]*Assistant Professor, Department Of Computer Engineering, Smt. Kashibai Navale College Of Engineering, Pune, India*

Abstract: Parking management has become a critical issue in modern urban areas due to the increasing number of vehicles. Traditional parking methods rely heavily on manual intervention, leading to inefficiencies, congestion, and high operational costs. To address these challenges, this research proposes a cloud-based Parking Management System that integrates DevOps automation tools to ensure efficiency, scalability, and real-time monitoring. The system is developed using Flask, MySQL, HTML, CSS, and JavaScript and is hosted on AWS EC2 (Free Tier). It employs DevOps tools like Git, Jenkins, and Docker for automation and deployment. The proposed system allows users to check real-time slot availability, make bookings, and process payments online. It enhances parking efficiency by automating space allocation and reducing human intervention. By leveraging cloud infrastructure, this system ensures high availability and remote access. The CI/CD pipeline automates deployment and updates, improving reliability and reducing downtime.
Keywords: Parking Management, Cloud Infrastructure, AWS, Scalable, CI/CD pipeline

.

I. INTRODUCTION

As urban populations grow, the number of vehicles on the road continues to rise, leading to increased congestion and inefficient parking management. Parking facilities often struggle to handle high demand, resulting in frustration for drivers and operational difficulties for administrators. Traditional parking systems rely on manual ticketing and cash transactions, which can cause delays and lead to errors. Furthermore, the lack of real-time slot tracking leads to inefficiencies, where parking spaces are either overused or underutilized. With advancements in cloud computing and DevOps automation, these issues can be mitigated. This paper introduces a scalable, cloud-based Parking Management System that integrates automated deployment and monitoring tools to streamline operations. By implementing a web-based solution, users can check real-time slot availability, book spaces in advance, and make cashless transactions. The goal is to reduce congestion, optimize space usage, and improve the overall parking experience for users.

II. METHODOLOGY

The development methodology follows an Agile approach with iterative improvements:

1) Requirement Analysis: Identifying system needs and technical feasibility.
2) Design Phase: Architecting the cloud infrastructure and DevOps pipeline.
3) Development: Implementing frontend, backend, and database interactions.
4) Testing: Conducting unit testing, integration testing, and performance evaluation.
5) Deployment: Hosting on AWS EC2 with automated updates via Jenkins.
6) Maintenance: Continuous monitoring and improvements based on user feedback.

III. MODELLING AND ANALYSIS

In this section, we present the architecture and overall flow of our Cloud-Based Parking Management System. The system is lightweight, integrates DevOps tools for automation and continuous deployment, and is hosted using scalable cloud infrastructure.

A. System Architecture

The proposed system is developed using the Flask framework for the backend and MySQL for the database. The web interface is designed using HTML, CSS, and JavaScript to allow real-time interaction with users and administrators. The system is containerized using Docker and deployed on AWS EC2 instance with an Elastic IP for public access.

The Flask application handles core logic such as slot availability, booking, user management, and payment processing. Gunicorn acts as the WSGI server to serve the Flask app, and Nginx is used as a reverse proxy to manage client requests and enhance performance.

B. DevOps Integration

Version control is handled through Git, allowing developers to manage and track code efficiently. Jenkins is configured to automate the Continuous Integration and Continuous Deployment (CI/CD) pipeline. Every code push to the repository triggers Jenkins to build the application, run necessary tests, and deploy the Docker container onto the EC2 instance.

This DevOps pipeline ensures quick updates, automated deployments, and rollback capabilities, reducing downtime and manual interventions.

C. Feature Flow and Integration

1) User/Admin Access: The users interact through a browser-based interface.
2) Frontend: Developed using HTML/CSS/JS, the frontend captures slot booking requests.
3) Backend (Flask): Processes the booking logic and communicates with the MySQL database.
4) Database (MySQL): Stores user data, slot details, and transaction records.
5) Cloud Hosting (AWS EC2): Hosts the application using an Elastic IP for remote access.
6) Gunicorn & Nginx: Gunicorn handles Flask execution while Nginx manages request routing.
7) CI/CD Pipeline: Jenkins fetches code from Git, builds and deploys updates.

D. Evaluation and Reliability

The system was tested under various scenarios including high traffic and concurrent bookings. With DevOps integration, the CI/CD process ensures stability, minimizes downtime, and enables frequent and reliable updates. Logs and performance metrics are tracked to monitor system health and user interactions. Overall, the architecture achieves real-time efficiency, high availability, and seamless scalability using modern cloud and DevOps practices.

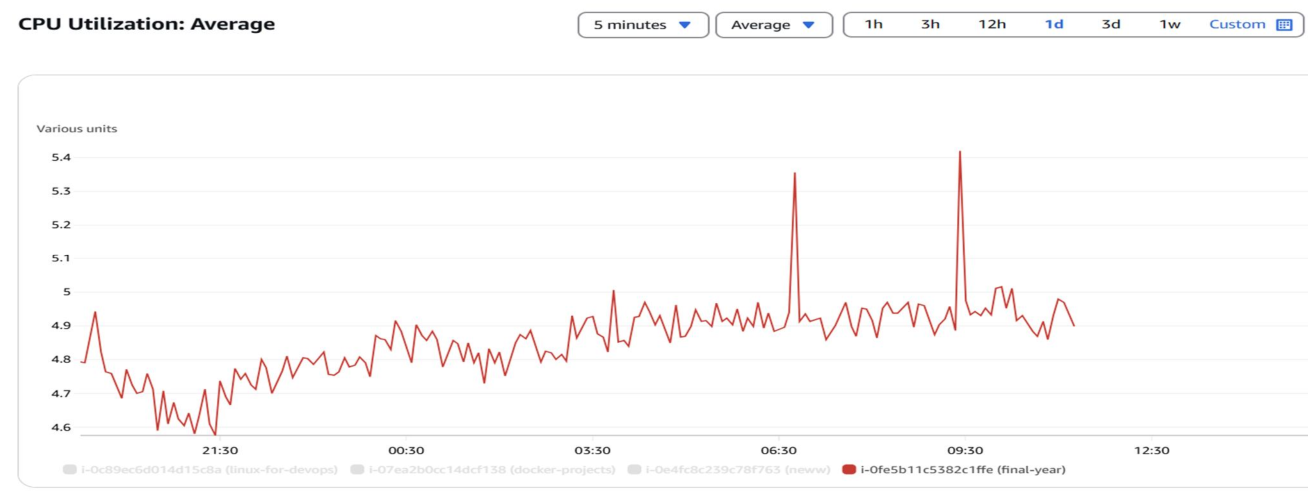


Fig 1. CPU Utilization

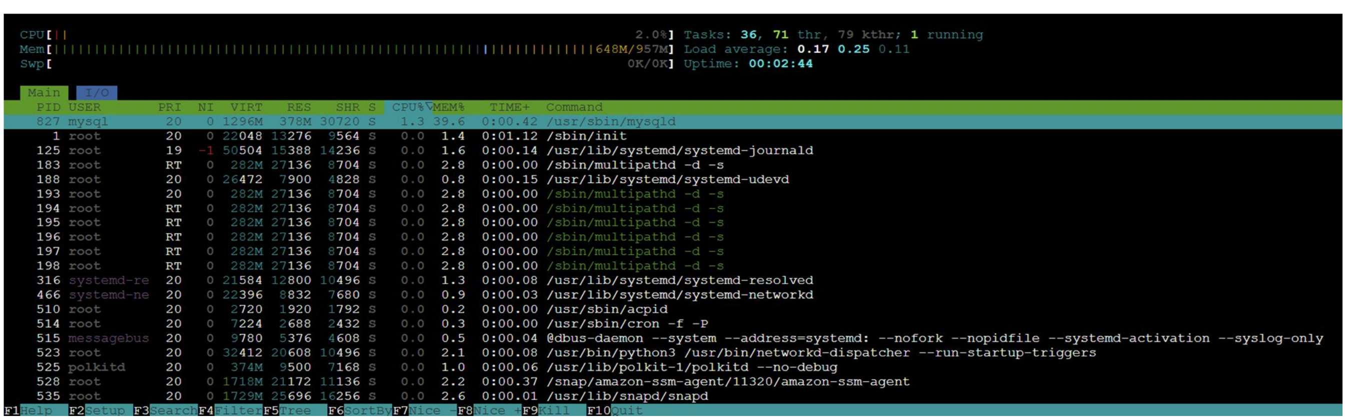


Fig 2. Server Performanc

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

E. Flow of System

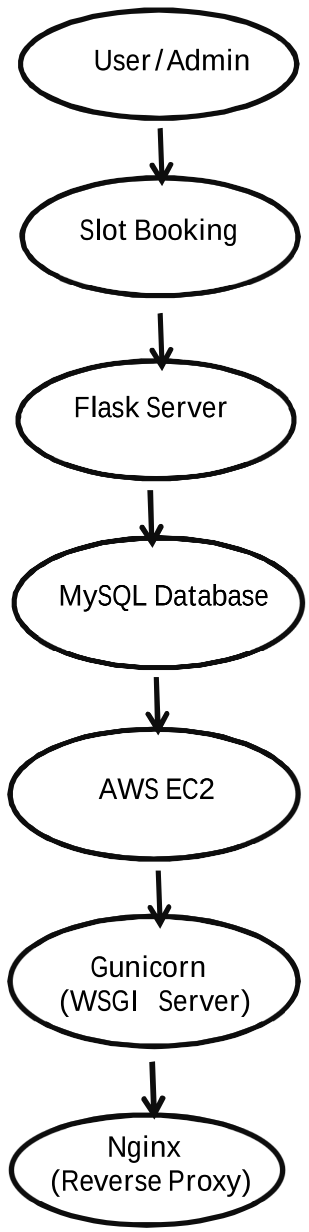
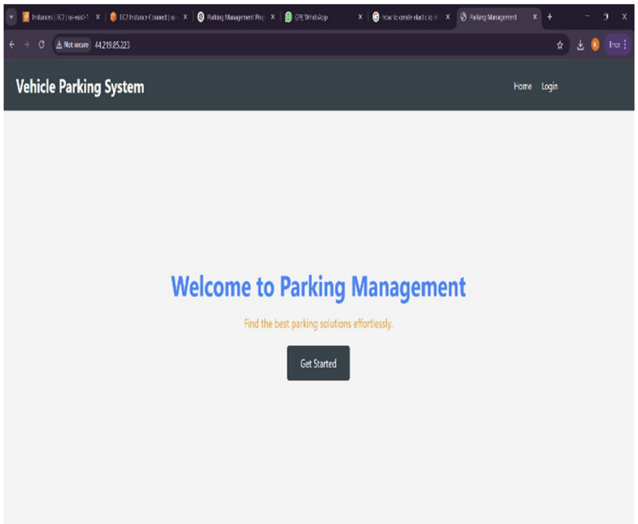


Fig 3. Flow Diagram

IV. RESULTS

1) Real-Time Slot Booking Works Reliably.
2) Cloud Hosting on AWS EC2 with Elastic IP Provides High Availability.
3) Reverse Proxy with Nginx Improves Performance and Load Handling
4) CI/CD Pipeline Ensures Seamless Deployment.
5) Database (MySQL) Integration Enables Persistent Data Storage.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com



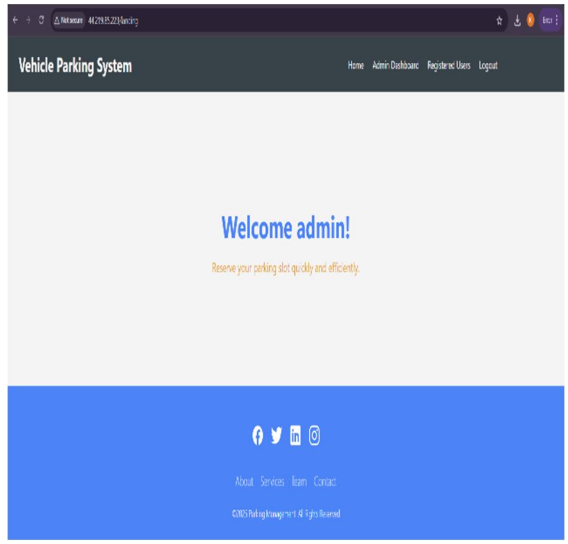
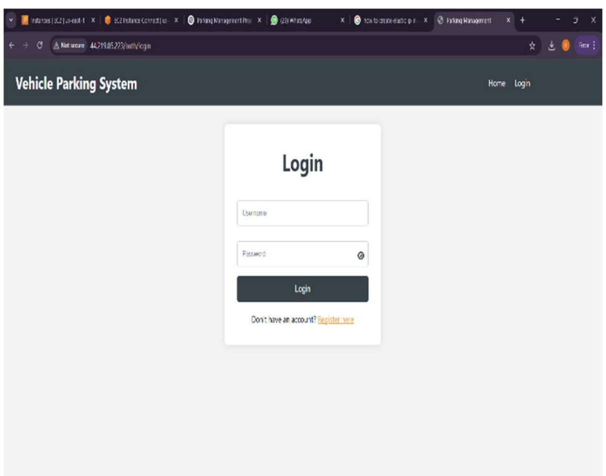


Fig 4. Home Page(Without Login)

Fig 5. Admin Lading Page



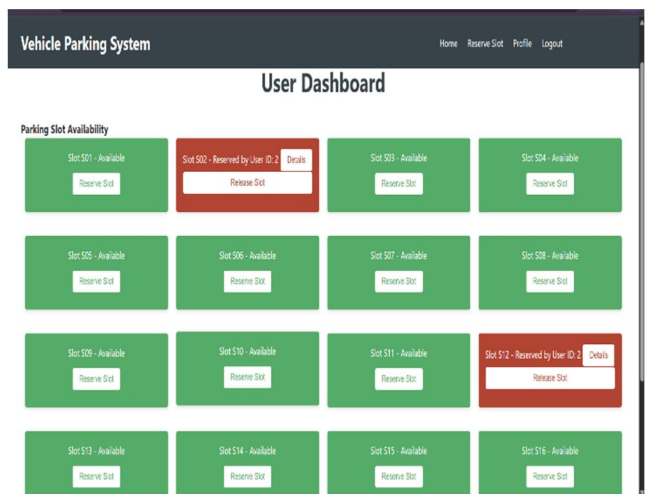


Fig 6. Login Page

Fig 7. Slot Reservation Page

V. CONCLUSION

This project demonstrates by using CNN based approach for waste classification using image analysis. This desktop application provides simple tool for realtime waste classification.Although results are good, the model requires larger dataset and model advancement through vision transformers. These improvements will enable model to generalize better and can be deployed in real world environment.

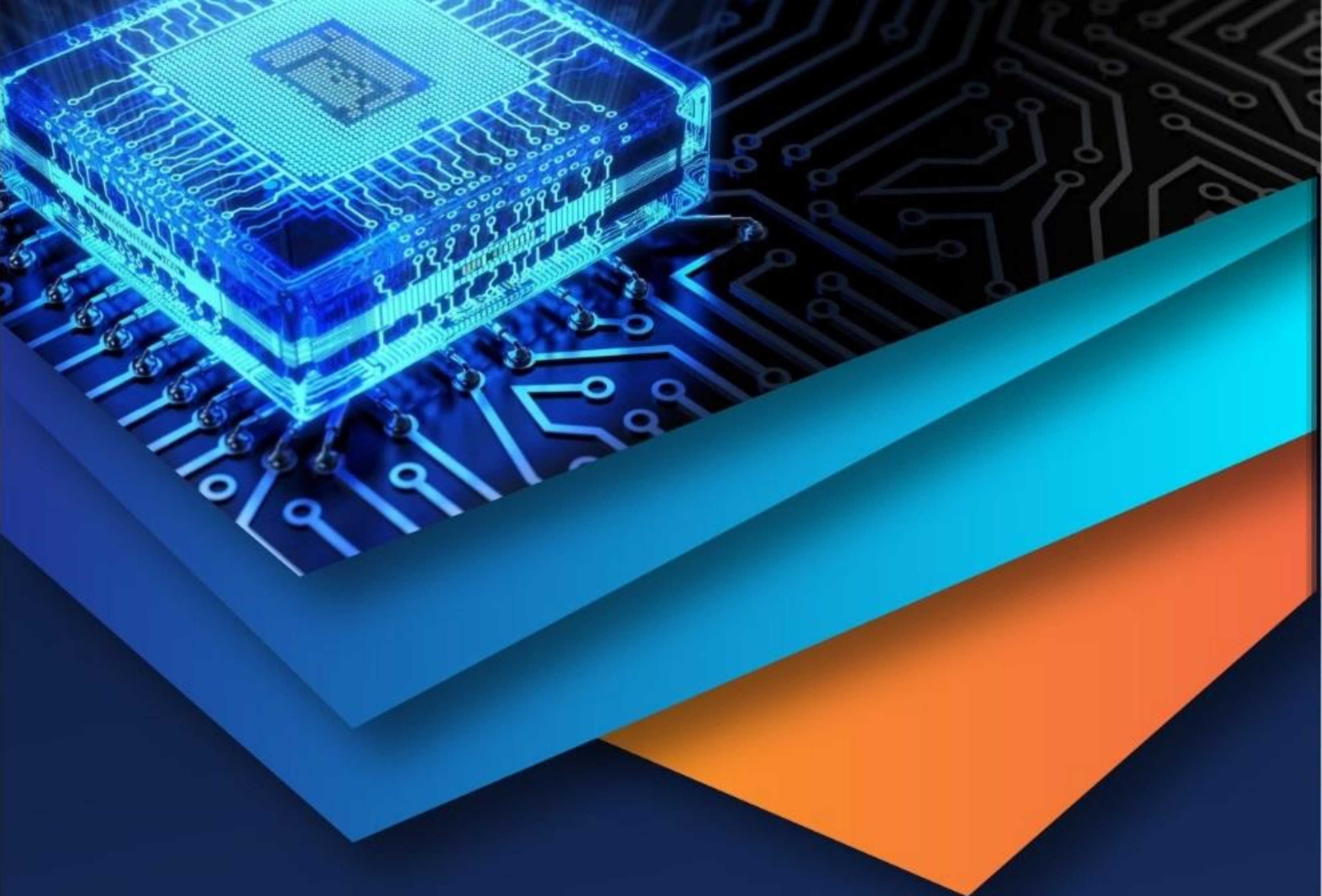
REFERENCES

[1] R. Sharma, M. Patel, and S. Singh, "Cloud-based smart parking system using IoT and DevOps tools," International Journal of Computer Applications, vol. 182, no. 31, pp. 15–20, 2024. doi: 10.5120/ijca2024912893

[2] A. Gupta, R. Verma, and N. Roy, "DevOps-based CI/CD pipeline for scalable web applications on AWS," IEEE Access, vol. 12, pp. 67540–67548, 2024. doi: 10.1109/ACCESS.2024.3355612

[3] P. Kulkarni, T. Deshmukh, and A. Kale, "Development and deployment of real-time parking management using cloud computing," International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE) ,vol. 11, no.2, pp.390–396,2024. doi:10.15680/IJIRCCE.2024.1102065.

[4] S. Mehta and K. Joshi, "Integrating Jenkins and Docker for efficient CI/CD deployment of Python Flask applications," International Research Journal of Engineering and Technology (IRJET), vol. 11, no. 1, pp. 1023–1028, 2024. doi: 10.5281/zenodo.11020701.
[5] L. Thomas and A. Nair, "An efficient smart parking system using PHP and MySQL hosted on AWS," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 10, no. 3, pp. 321–327, 2024. doi:10.6084/m9.figshare.23922123.
[6] P. G. Das, A. N. Ranjan, and T. B. Kumar, "Deep learning techniques for efficient solid waste classification," Waste Management, vol. 123, pp. 65-73, 2023. doi: 10.1016/j.wasman.2022.11.0



10.22214/IJRASET

INDEX COPERNICUS

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY