# PROJECT REPORT

**On**

# Vehicle Movement Analysis and Insight Generation in a College Campus

**Submitted By:**

**SOUMYAJIT ROY**

roysoumyajit36@gmail.com

**MANAV MALHOTRA**

manavmalhotra173@gmail.com

**ISHTAJ KAUR DEOL**

ishtajdeol@gmail.com

**SWARNAV KUMAR**

swarnav786@gmail.com

**Submitted to**

**Intel**
**Unnati Industrial Training Program**

# Contents

# Chapter 1

## Introduction

Parking on college campuses is frequently a significant difficulty due to the high volume of automobiles and the scarcity of parking places. Students, teachers, and guests are often frustrated by this circumstance because they must spend a lot of time looking for open slots. Furthermore, the absence of a reliable monitoring system makes it challenging to properly manage parking spaces, which leads to underutilization or overcrowding of certain areas.

Our initiative aims to improve college campus parking to tackle these problems. The goal is to create a comprehensive system that enables precise identification of the individual parking space that each vehicle occupies, real-time monitoring of parked cars, and the production of useful insights from the parking data gathered. This system seeks to offer several advantages:

- Enhanced Parking Administration
  Campus authorities can better manage parking resources and ensure efficient use of available spaces by properly measuring the occupancy of parking slots in real-time. By planning and assigning parking spots more effectively, the system can lessen the likelihood of shortages or overflows of parking spaces.

- Improved Convenience for Users
  Users may keep an eye on their cars from a distance, always tracking their location and status. This feature provides ease and peace of mind, especially for customers who might forget where they parked or are worried about the security of their vehicle.

- Data-Driven Insights
  Usage trends and patterns can be found through the long-term collecting and analysis of parking data. Decisions like increasing parking facilities or changing pricing tactics, for example, might be made with these facts in mind while improving infrastructure. Additionally, data-driven insights can aid in more efficient peak parking hour management and demand forecasting, which will lessen traffic and improve the parking experience overall.

# Chapter 2

## Challenges

College campus parking poses a number of difficulties that have a big impact on user convenience and operational effectiveness. In a dynamic setting where parking places are scarce and the number of vehicles is growing, effective management is essential. These difficulties, specifically, are:

- Time Wastage
  The amount of time that visitors, teachers, and students must spend looking for open parking spaces is one of the most significant problems they encounter. This issue is made worse by the absence of an effective system to direct drivers to open slots. People frequently circle the parking lot more than once, which causes delays, annoyance, and sometimes missing appointments or classes.

- Inefficient Space Utilization
  Parking spots are frequently utilized inefficiently in the absence of sophisticated allocation software and real-time monitoring. While some spaces grow busy, others might continue to be unused. Because of this mismatch, there may be a large number of unoccupied spots in less popular locations while popular zones are completely booked, which could lead to an underutilization of the available resources.

- Security Concerns
  Car owners are very concerned about security, especially in big, unattended parking lots. These worries are increased by the inability to remotely monitor the condition of parked cars. Particularly at night or in dimly lit regions, there is a chance of theft, vandalism, and other criminal activity.

- Inconvenience
  It's not uncommon to forget where you parked your car, especially in big parking lots with lots of spaces. People may become overly stressed and waste a lot of time looking for their cars as a result. Due to schedule constraints on campus, this inconvenient situation may cause you to be late for meetings, classes, or other crucial events.

- Absence of Data-Driven Inference
  It is challenging for campus authorities to make well-informed decisions about parking management, infrastructure development, and policy implementation because they

frequently lack comprehensive data on parking trends and space utilization. It is difficult to pinpoint peak usage hours, comprehend the unique requirements of various user groups, and make plans for upcoming additions or changes without precise and thorough data. This lack of information may lead to less-than-ideal investments and policies, which in turn may not adequately address the underlying problems.

- Problems with Accessibility
  Another major issue is making parking accessible for those with impairments. It is challenging to oversee and enforce reserved accessible parking spaces without real-time surveillance. This may lead to abuse of these spaces or their non-availability for those who actually require them, which would cause more annoyance and possibly even prejudice toward people with disabilities.

- Inefficiencies in Operations
  There are several operational inefficiencies that can result from a disjointed parking management system. For instance, college security officers could have to physically patrol parking lots, enforce regulations, and help people locate spaces. This manual error is prone to human error and requires a lot of resources. By automating these procedures, you can increase productivity and free up staff for other important work.

- Financial Consequences
  For the campus's finances, ineffective parking management may also have consequences. While busy parking lots might result in higher maintenance expenses owing to increased wear and tear, underutilized parking spaces represent lost revenue opportunities. Furthermore, user annoyance has the potential to harm the campus's reputation and sway prospective students.

# Chapter 3

## Objective

A sophisticated system with real-time parking spot monitoring and management capabilities is desperately needed, especially considering the difficulties with parking on college campuses. In order to facilitate effective use of available spaces by directing drivers to empty slots and guaranteeing a balanced distribution of vehicles, this system should precisely track and show the occupancy status of each parking slot. It should also make remote vehicle monitoring easier, enabling customers to always monitor the position and condition of their parked cars, improving security, and bringing comfort. To provide insightful information about usage trends, peak periods, and overall utilization, the system should also gather and analyse parking data.

The primary objectives of this project are as follows:

- Develop a Real-Time Monitoring System:
  Install a cutting-edge artificial intelligence system that can instantly identify and detect license plates. Give consumers the ability to remotely view the position and status of their automobiles by integrating this technology with an intuitive user interface.

- Accurate Slot Recognition:
  Make sure the system can precisely determine which parking space a car is in, giving users and campus officials precise information.

- Produce Useful insights:
  Examine gathered parking data to find trends, peak usage periods, and overall parking spot consumption. Provide reports and visuals to assist in making defensible choices regarding infrastructure development and parking management.

The project's goal is to create an Edge AI-based system that improves the experience of parking cars on college campuses in order to solve these issues. The particular goals are to increase customer convenience, optimize parking management, deliver actionable insights from data analysis, and, in the end, develop a more intelligent and effective parking system for college campuses.

# Chapter 4

# Dataset Description

## Image Dataset:

In this project, 433 photos make up the image collection. Each image has bounding boxes that clearly identify the license plate of an automobile. The aforementioned annotations are organized within PASCAL VOC XML files and provide crucial metadata, like file and folder names, image dimensions, and exact coordinates that define the bounding box of every license plate. This dataset provides essential visual input for training and validating the ability of the edge AI system to detect and recognize license plates with accuracy.

- Source: For this project, an image dataset was sourced and manually annotated for the purpose of training license plate detection.
  Dataset Link: https://www.kaggle.com/datasets/andrewmvd/car-plate-detection
- Size: 433 photos with PASCAL VOC XML annotations are included, offering comprehensive spatial data that is essential for training AI models.
- Features: To enable accurate license plate detection, each image annotation contains information about the folder and file, the image's dimensions, and the bounding box coordinates.

## Simulated Parking Dataset:

Python is used to build the simulated parking dataset programmatically over a period of 197 days, from 1st January 2024 to 15th July 2024, simulating real-world parking dynamics. It mimics 100 permitted automobiles pulling into and out of 15 parking spaces. This dataset contains detailed information about every entry, including the vehicle number, the time of entry (in-time), the time of departure (out-of-time), and the assigned parking space number. This dataset offers a practical framework to maximize parking space use and identify patterns in parking behaviour, which is helpful for testing and improving the functionality of the parking management system.

The creation of the simulated parking dataset is made easier by our Python script. It initializes variables such start_date, which sets the start date of the simulation, entries_per_day, which defines daily parking entries, and num_days, which specifies the duration of the simulation. To imitate permitted cars parking, authorized vehicle numbers are listed in authorized_vehicle_numbers.

The helper methods is_time_slot_available(slot, in_time, out_time, slot_usage) guarantees that each parking slot is used effectively without overlap, while random_datetime(date) generates random entry times. Every day, the script generates parking entries by selecting vehicle numbers at random and timing entry and exit. It tracks usage in the slot_usage dictionary and verifies slot availability.

The generated data is output by the script into a CSV file called "indian_vehicle_parking_data.csv" after it has finished running. This file has columns for the car number, entry time, exit time, and slot number. Analysis of parking patterns and occupancy rates is made possible by this structured dataset, which is essential for using AI-driven insights to optimize parking management tactics.

- Source: This project's simulation was created using programming.
- Size: Contains maximum 100 parking entries every day for 197 days, or roughly 5537 records.
- Features: Car numbers, entry and exit timestamps, and parking slot allocations are all included in the records, which makes it easier to analyze parking patterns and use AI algorithms to optimize space use tactics.

# Chapter 5

# Methodology

## License Plate Detection Model Training

- Importing Required Libraries and Dependencies: Installing and importing OpenCV, Pandas, NumPy, and PyTorch are among the necessary libraries that must be imported before the project can proceed. These libraries are essential for a variety of activities, including building deep learning models and handling data and images. Verifying GPU availability is a good way to make sure the project can take advantage of accelerated processing capabilities, which can help accelerate the training and inference of models.

- Importing Data: 433 photos with bounding boxes around vehicle license plates make up the image collection utilized for this research. The metadata that these annotations contain, including image dimensions and exact coordinates that define the bounding box of each license plate, is recorded in PASCAL VOC XML files. Python's XML parsing powers are employed to retrieve this metadata and arrange it for additional handling.

- Preprocessing: Processing Images and Annotations Preprocessing entails reading each image into OpenCV and extracting the pixel data and dimensions. Bounding box coordinates are processed and normalized simultaneously from the XML annotations. In order to prepare the data for consistent input into the model during training, this normalization phase makes sure that all bounding box coordinates are scaled appropriately in relation to the image dimensions.

- Train-Test Split: Using sklearn's train_test_split function, the dataset is divided into training, validation, and test sets. By keeping the split at 80/10/10, we can make sure that the model is trained on a significant amount of the data and is validated and tested on different, unseen subsets. In order to assess the model's performance on fresh data and avoid overfitting, this separation is essential.

- Converting data into the popular YOLO (You Only Look Once) format is necessary for object detection models. The bounding box coordinates are transformed into normalized values appropriate for YOLO model input using a custom function that makes directories for the photos and labels that go with them. The model will be able to precisely identify and locate license plates within photos thanks to this format.

- YOLOv8n Model Selection: Due to its effectiveness in real-time object detection tasks, Ultralytics' YOLOv8n model was selected. The model is tuned to perform well on the unique features of the license plate identification dataset using parameters like epochs, batch size, and image size.

- Training Procedure: The dataset is formatted in YOLO format and the selected parameters are used to start the training process. By minimizing a predetermined loss function, the model learns to reliably detect license plates by adjusting its internal parameters through backpropagation during training. Until the model converges to an accuracy level that is deemed satisfactory, this iterative procedure is continued.

- Prediction and Visualization: Bounding box predictions and visualizations on a different set of test images are used to assess trained models. The model predicts where license plates will be located in each test image using the saved model weights. Bounding boxes are superimposed over the photos to display these predictions, together with confidence scores that represent the model's level of assurance for each detection.

- Post-Processing and Analysis: In order to extract and analyse text from the identified license plate sections, post-processing procedures incorporates optical character recognition (OCR). By offering actionable insights from the identified license plate data, such as recognizing specific license plate numbers or examining vehicle movement patterns, this extra step improves the practical utility of the model.

## Vehicle Movement Insights

- Importing Requirements and Necessary Libraries: The project starts with the import of required libraries that help with different phases of data processing and analysis. These libraries include Seaborn, which makes visually appealing and educational statistical visuals, Matplotlib, which plots and visualizes data, and Pandas, which manipulates and analyses data. For the purpose of completing the data exploration and visualization tasks needed for this project, these libraries are essential.

- Importing Dataset: Pandas' read_csv function is used to import the dataset, which goes by the name "indian_vehicle_parking_data.csv." The comprehensive records of car movements within a parking facility are included in this collection. We can obtain a preliminary understanding of the structure and contents of the data by using the head function to examine the first few rows of the dataset.

- Data exploration and pre-processing: Using the info function, which offers insights into the data kinds and non-null counts of each column, the exploration starts by presenting a basic overview of the dataset. The describe technique is used to display summary statistics, which provide a numerical overview of the dataset. Using the isnull method, missing values are examined. To enable time-based analysis, datetime columns—more precisely, "In Time" and "Out Time"—are processed and transformed into Pandas datetime objects. Extra time-related information is collected, like the day and hour of car entries and exits. The hours-long duration of every parking event is likewise obtained by deducting the "In Time" from the "Out Time."

## Generating Insights

- Distribution of Vehicle Entries by Hour: The distribution of vehicle entries throughout the day is visualized using a histogram. A plot is created with Seaborn's histplot function, which displays the frequency of vehicle entries for each hour of the day. This visualization helps identify peak entry times, providing insights into when the parking facility experiences the highest influx of vehicles.

- Distribution of Vehicle Exits by Hour: Similarly, the distribution of vehicle exits by hour is plotted to understand when vehicles typically leave the parking facility. Using a histogram with Seaborn, this plot reveals the times of the day when vehicle exits peak, complementing the entry distribution analysis.

- Distribution of Parking Duration: The duration for which vehicles remain parked is analyzed using a histogram. Seaborn's histplot function is employed to illustrate the distribution of parking durations. This analysis helps in understanding the common parking durations and identifying any outliers with unusually long or short parking times.

- Most Frequently Occupied Parking Slots: To identify the most utilized parking slots, the frequency of occupancy for each slot is calculated. A bar plot is generated using Seaborn's barplot function, highlighting which parking slots are most frequently occupied. This information can be useful for facility management to optimize parking space allocation.

- Real-Time Occupancy Patterns Over Time: The real-time occupancy patterns are examined day-wise, month-wise, and week-wise to uncover trends and peak times. By grouping the data by date and hour, a line plot is created with Seaborn's lineplot function to visualize the daily occupancy patterns. Similar plots are generated for month-wise and week-wise occupancy, providing a comprehensive view of how parking patterns vary over different time scales.

- Top 10 Cars by Total Parking Duration: The total parking duration for each vehicle is summed up to identify the top 10 vehicles that spend the most time parked in the facility. A bar plot is created to display these vehicles, providing insights into which vehicles or owners use the parking facility the most.

- Top 10 Vehicles by Entry Frequency: The frequency of entries for each vehicle is counted to determine the top 10 vehicles with the highest entry frequency. This analysis is visualized using a bar plot, highlighting vehicles that frequently enter the parking facility.

- Heatmap of Hourly Occupancy by Day of the Week: To analyze occupancy patterns in a more granular manner, a heatmap is created to show the occupancy count for each hour of the day across different days of the week. This visualization helps identify specific hours and days with the highest or lowest occupancy, facilitating better understanding of weekly parking trends.

- Parking Duration Distribution by Day of the Week: The parking duration is analyzed for different days of the week using a box plot. This plot, created with Seaborn's boxplot function, shows the distribution of parking durations, highlighting any variations in parking behavior across different days.

- Parking Slot Utilization: The total parking duration for each slot is calculated and visualized using a bar plot to show which slots are used the most. This analysis can help in optimizing slot usage and managing parking space more efficiently.

- Mean Parking Duration by Day of the Week: The average parking duration for each day of the week is calculated and visualized using a bar plot. This plot provides insights into how long vehicles typically stay parked on different days, which can be useful for planning and managing parking resources.

- Violin Plot of Parking Duration by Day of the Week: A violin plot is created to show the distribution of parking durations for each day of the week. This plot combines aspects of box plots and kernel density plots, offering a detailed view of the parking duration distributions and any variations across different days.

- Strip Plot of Parking Duration by Day of the Week: Finally, a strip plot is generated to show the individual parking duration values for each day of the week. This plot provides a granular view of parking durations, with each dot representing a single parking event, allowing for the identification of patterns and outliers.

# Chapter 6

# Web App

1. ## home.py

   The home.py script serves as the main entry point for the Vehicle Registration and Monitoring App, built using Streamlit. The script presents a welcoming interface with a title and descriptive text about the application. It offers four primary functionalities: Register Vehicle, Vehicle Log, Vehicle Entry, and Parking Insights, each displayed as a card with an icon and a brief description. These cards allow users to easily navigate to different sections of the app. Additionally, the script includes a section detailing the contributors to the project and a link to the GitHub repository, providing recognition to the developers and easy access to the source code.

2. ## parking_insights.py

   The parking_insights.py script is designed to provide detailed insights into parking data on a college campus. It loads data from CSV files containing information about authorized vehicles and parking logs. The script offers various analytical tools, including the ability to find the hour with the maximum number of cars parked on a selected date, check parking slot occupancy at a specific date and time, and search for vehicle records by plate number and date. The results are displayed in a structured table format for easy interpretation.

3. ## register_vehicle.py

   The register_vehicle.py script is designed to facilitate the registration of vehicles on a college campus through image processing. It employs various Python libraries, including Streamlit for the web interface, NumPy for numerical operations, OpenCV for image processing, and pandas for data handling. The script allows users to upload images of vehicle number plates, which are then processed to detect and extract text. The extracted text is cleaned and analyzed to identify the state code and correct the format of the number plate. The corrected number plate is then verified against a CSV file of authorized vehicles. Users can confirm or correct the detected number plate manually, and newly authorized plates are added to the CSV file for future reference.

This ensures a streamlined process for vehicle registration and enhances the user experience with an intuitive web interface.

## 4. vehicle_entry.py

The vehicle_entry.py script focuses on managing vehicle entry in a college campus parking system by utilizing image recognition to detect vehicle number plates. Similar to register_vehicle.py, this script uses Streamlit for creating the web interface, OpenCV for image processing, and pandas for data management. Users can upload images of vehicle number plates, which are then processed to detect and extract text. The extracted text undergoes cleaning and analysis to verify and correct the number plate format. The script then checks if the detected number plate is authorized by comparing it against a list of pre-authorized vehicles stored in a CSV file. If the number plate is confirmed as correct and authorized, access is granted; otherwise, access is denied with instructions to register the vehicle first. This script helps maintain security and proper authorization for vehicle entry on campus.

## 5. vehicle_log.py

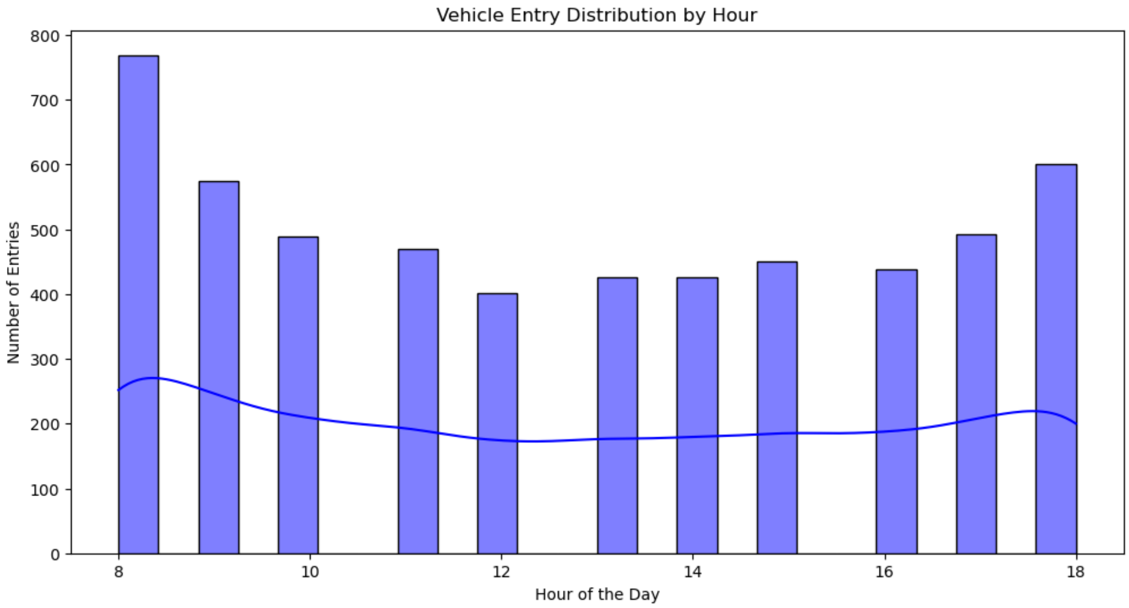
The vehicle_log.py script is designed to log vehicle entry and exit times, providing a detailed history of parking activities on a college campus. Utilizing Streamlit for the web interface and pandas for data handling, the script reads data from CSV files containing authorized vehicles and parking logs. Users can input a vehicle number plate to check its log history. The script verifies if the vehicle is authorized and retrieves corresponding logs from the parking data. It calculates the total parking time for each log entry and formats the times for display. The vehicle logs are presented in a tabular format, showing slot number, in time, out time, and total parking time. This feature enables efficient tracking of vehicle movements and ensures accurate record-keeping for campus parking management.

# Chapter 7

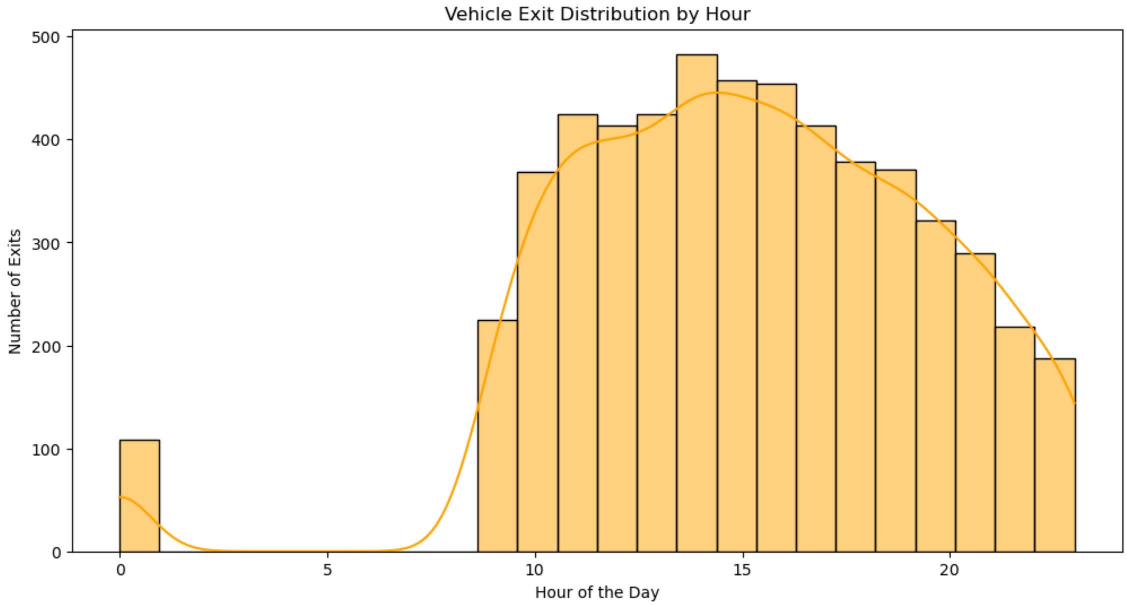
## Results

The results of this study encompass various analyses on vehicle movements, focusing on license plate detection, vehicle entries and exits, parking durations, and overall parking behaviour patterns.

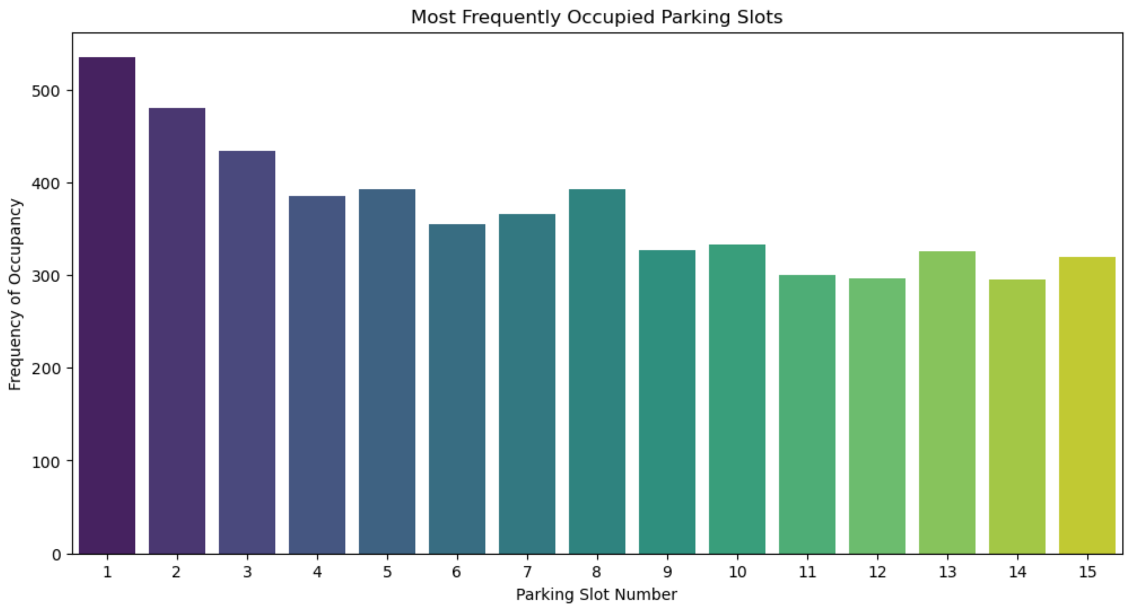
### Parking Insights and Statistics



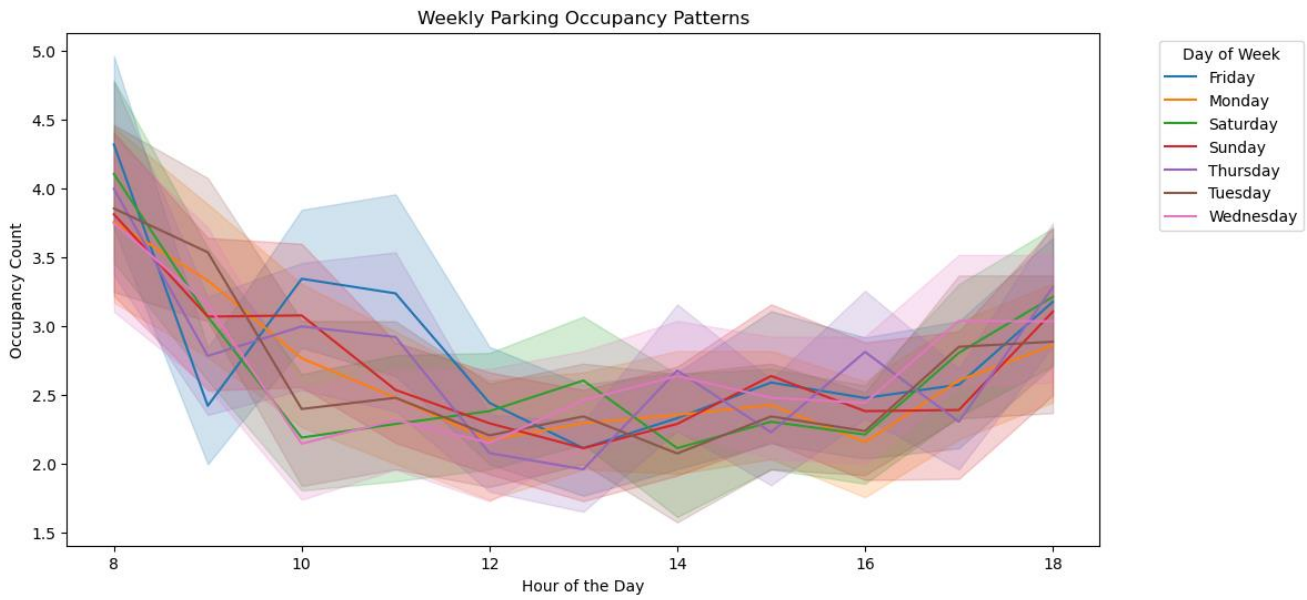

The above graph of vehicle entry distribution by hour reveals distinct patterns in daily vehicle influx. The highest number of entries, around 800, occurs at 8 AM, indicating a significant morning peak likely associated with the start of work or school. Following this peak, entries gradually decline, with a noticeable decrease between 9 AM and 10 AM. Throughout midday and afternoon, the entry rate remains relatively stable but lower, with a slight dip around noon and early afternoon. There is a secondary increase in vehicle entries around 6 PM, likely corresponding to the end of the workday or school day, indicating another peak during the early evening. Overall, the data shows two main peaks: a prominent one in the early morning and a smaller one in the early evening, with more consistent and lower entry rates during midday. These insights can help optimize resource allocation and traffic management strategies to handle peak periods effectively and improve vehicle movement efficiency within the facility.



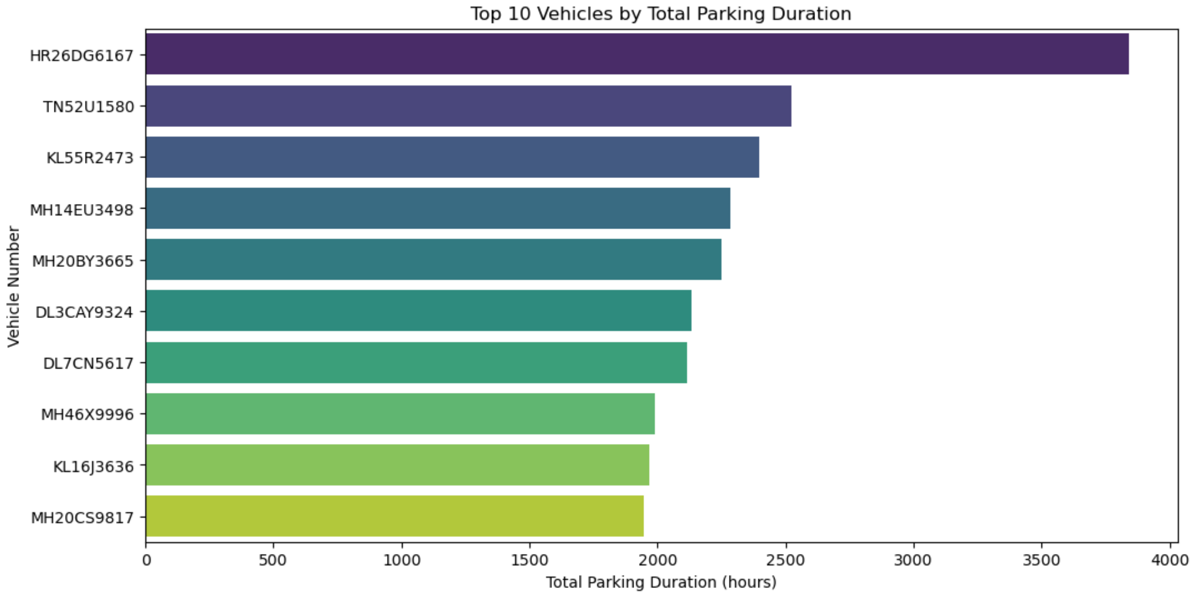

The bar plot displaying vehicle exit distribution by hour reveals distinct patterns in exit times. There is minimal activity during the early morning hours, with a slight increase around midnight. The number of exits starts to rise sharply after 6 AM, peaking between 2 PM and 3 PM, where the highest frequency of vehicle exits occurs, reaching nearly 500. After this peak, the number of exits gradually declines throughout the evening, with a noticeable drop-off after 8 PM. This pattern suggests that most vehicle exits occur during typical daytime working hours, particularly in the early to mid-afternoon, indicating high activity likely related to work schedules, errands, and other daytime commitments. The decline in exits during the late evening and early morning hours reflects reduced activity during these times.



The above provided bar plot illustrates the frequency of occupancy for different parking slots, revealing key insights into usage patterns. Slot 1 stands out as the most frequently occupied, with over 500 instances of use, significantly higher than other slots. Slots 2 and 3 also show high occupancy, each exceeding 400 uses. Slots 4 through 10 exhibit moderate usage, with frequencies between 300 and 400. In contrast, Slots 11 to 15 are the least occupied, all below 350 instances. This descending trend from Slot 1 to Slot 15 suggests that certain slots are more popular, possibly due to their location or ease of access. The findings imply a need to investigate the factors contributing to Slot 1's popularity, and there may be opportunities to redistribute demand by promoting or enhancing the less frequently used slots. Understanding these patterns can aid in better managing parking slot utilization.



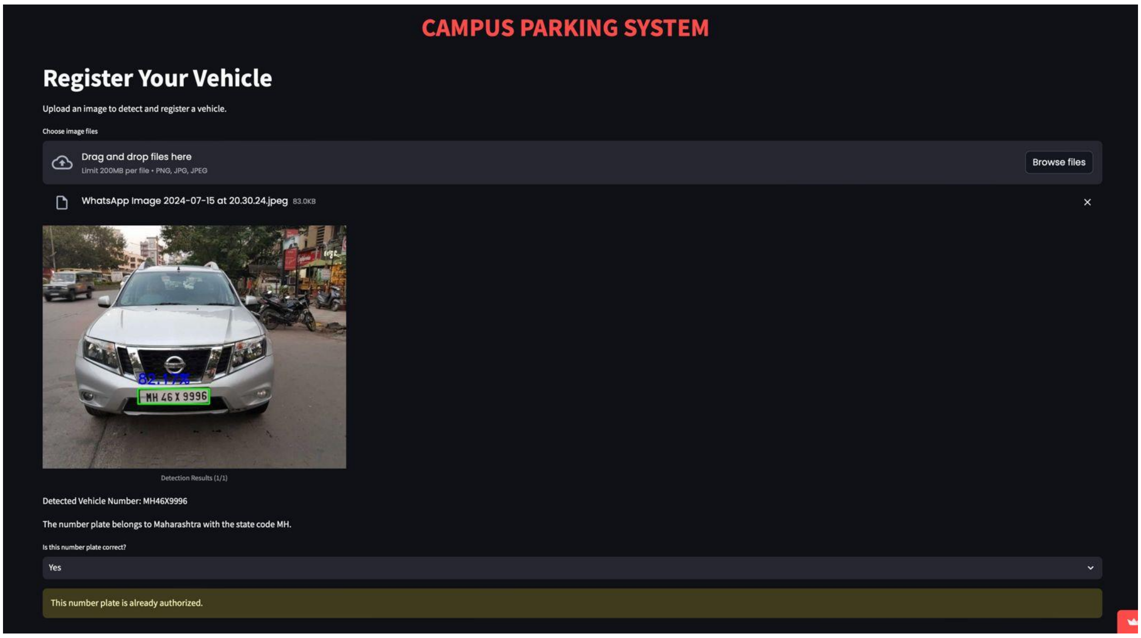

The line plot depicting weekly parking occupancy patterns by hour and day of the week reveals several trends in parking behavior. Across all days, there is a noticeable peak in occupancy around 8 AM, followed by a decline that continues until around noon. This pattern suggests a high initial demand for parking spaces in the morning, likely due to work or school-related activities. After the noon dip, occupancy remains relatively low and stable through the afternoon, with minor fluctuations. Another increase in occupancy is observed in the late afternoon and early evening, around 5 PM to 6 PM, indicating a second wave of parking demand as people return from work or other activities. The occupancy patterns are fairly consistent across different days of the week, with slight variations in peak times and levels, reflecting routine daily schedules. This insight can help in managing parking resources more efficiently by anticipating high-demand periods and planning accordingly.



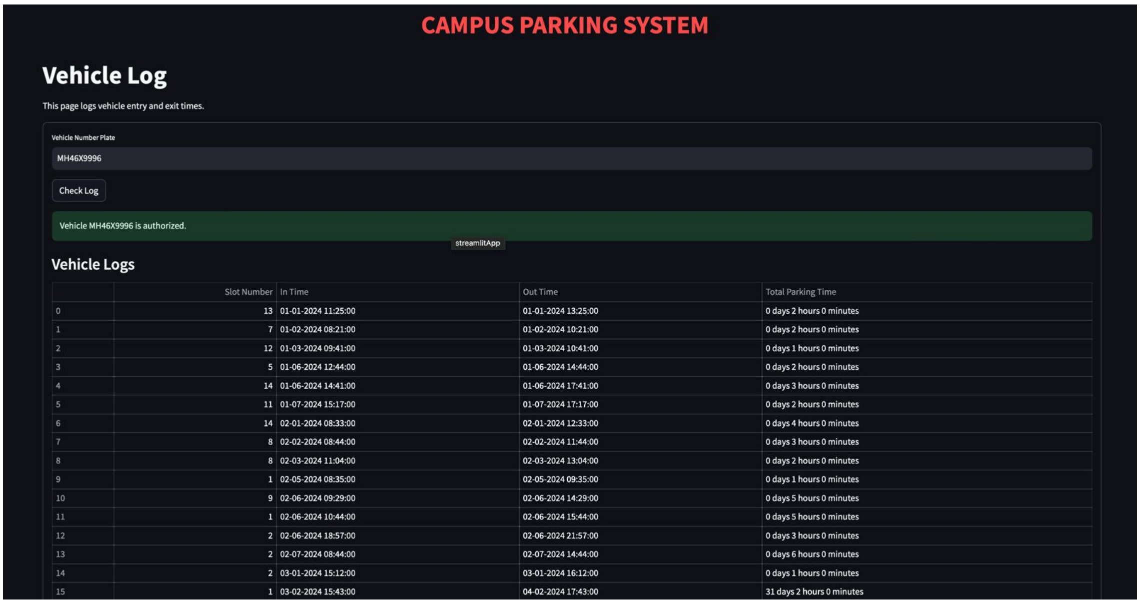

The bar graph depicts the top 10 vehicles based on their total parking duration in hours. Vehicle HR26DG6167 has the highest parking duration, significantly exceeding 3500 hours, indicating it is parked for extended periods compared to others. The second highest, TN52U1580, shows a substantial drop, parking just over 2500 hours. The remaining vehicles, KL55R2473, MH14EU3498, MH20BY3665, DL3CAY9324, DL7CN5617, MH46X9996, KL16J636, and MH20CS9817, range between 1500 to just under 2500 hours. This suggests a trend where a few vehicles dominate parking space usage, potentially indicating long-term parking habits or vehicle inactivity. The clear disparity between the top vehicle and others might reflect specific usage patterns or ownership types, such as commercial versus personal vehicles.

# Web Interface

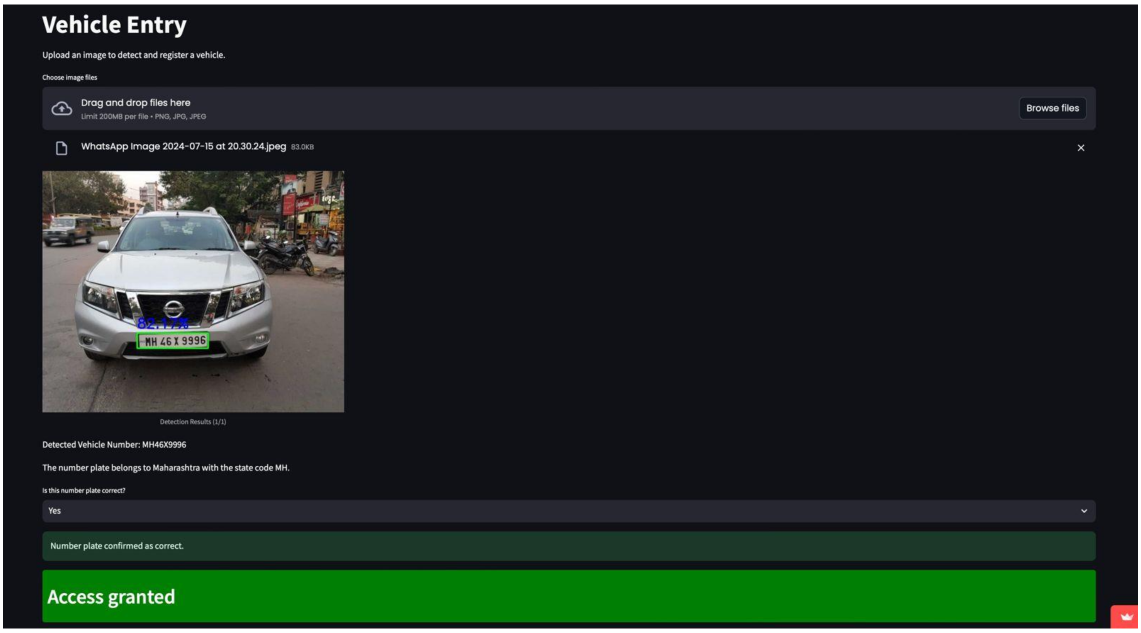

This is the home page of the "Campus Parking System" web application, designed to facilitate various parking-related tasks. It offers a user-friendly interface with clear navigation options for registering vehicles, viewing parking logs, managing vehicle entry, and gaining insights into parking trends. Each section is designed to provide quick access to specific functionalities related to campus parking management.



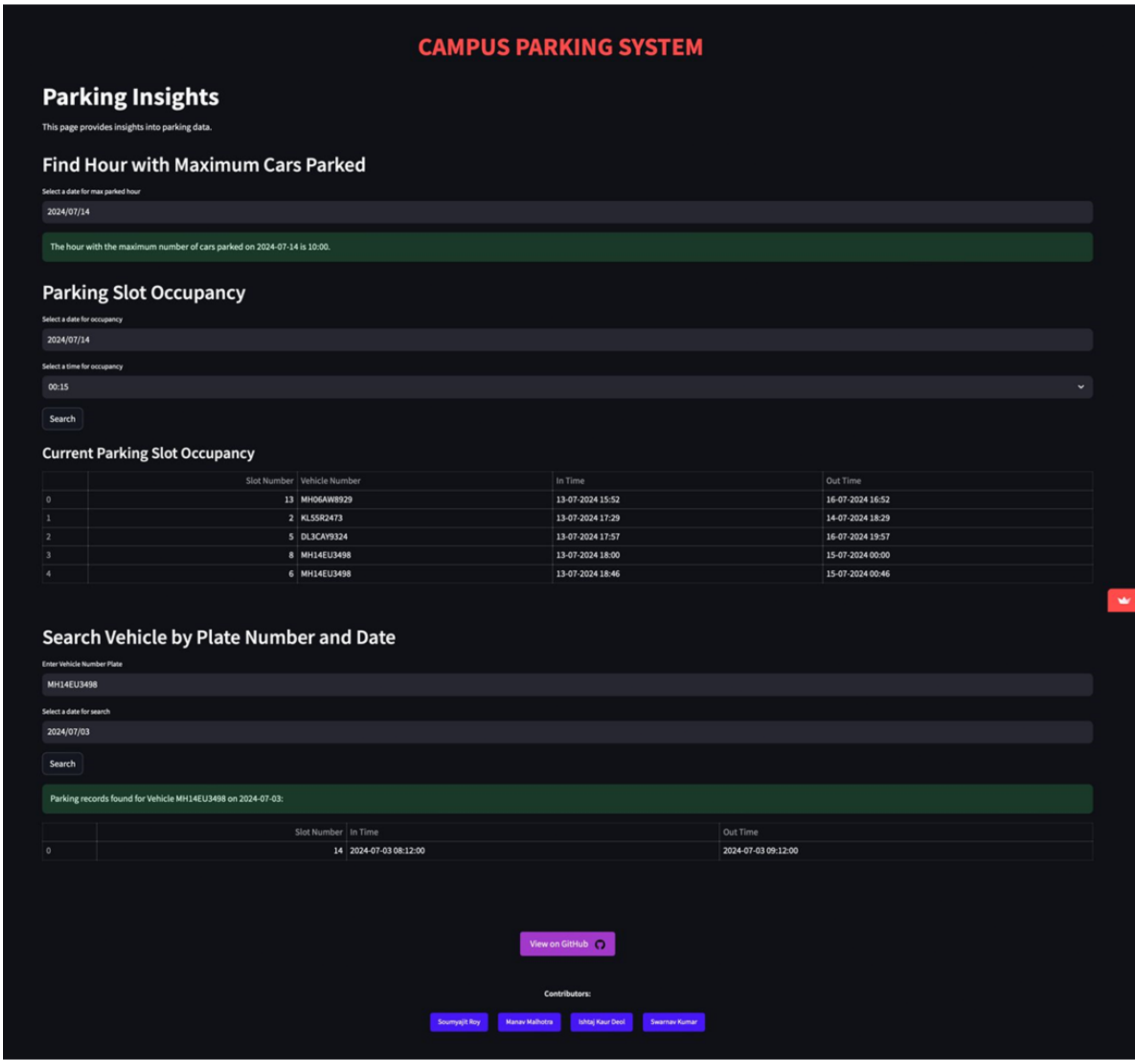

This webpage allows users to upload an image of their vehicle, automatically detects the vehicle's license plate number, identifies the state of registration, and provides a mechanism for users to confirm the detected information. If the vehicle is already authorised for parking, the system informs the user accordingly otherwise adds the vehicle number to the authorised vehicles database




The "Vehicle Log" page allows users to enter a vehicle's license plate number, check if the vehicle is authorized for campus parking, and view detailed logs of the vehicle's entry and exit times, parking slot numbers, and total parking durations. This functionality helps users keep track of their vehicle's parking activities on campus.




This webpage allows users to upload an image of a vehicle. Once an image is uploaded, the application displays the image and detects the vehicle number plate. Users are then prompted to confirm the accuracy of the detected number plate. Upon confirmation, if the vehicle is authorised, the webpage displays a message confirming the number plate as correct and grants access to the vehicle, indicated by a green "Access granted" message; else it denies access to the vehicle, indicated by a red “Access denied” message, and asks the user to register vehicle.




This webpage is designed to provide insights into parking data. It features sections for finding the hour with the maximum cars parked on a selected date, viewing current parking slot occupancy by selecting a date and time, and searching for vehicle parking records by entering a plate number and date. The occupancy data is displayed in a table format, showing slot numbers, vehicle numbers, in times, and out times.

# Chapter 8

# Conclusion and Future Work

## Conclusion

This study demonstrates the effectiveness of using the custom trained YOLOv8n model for real-time license plate detection and vehicle movement analysis. The model, trained on a manually annotated dataset of 433 images, showed high accuracy in detecting and locating license plates in various images. Integrating Optical Character Recognition (OCR) further enhanced the model's practical utility by extracting text from the detected license plates, allowing for detailed analysis of vehicle movements. Through extensive data exploration and visualization of the "indian_vehicle_parking_data.csv" dataset, significant patterns in vehicle entries, exits, and parking durations were identified. Insights into occupancy and utilization patterns revealed peak times, frequently occupied parking slots, and the behaviour of frequent users, providing valuable information for optimizing parking management. This project highlights the potential of edge AI in improving real-time parking solutions, making parking management more efficient and user-friendly.

## Future Work

We would focus on several areas to enhance the capabilities and applications of the current model. First, expanding the dataset to include more diverse images with varying lighting conditions, weather scenarios, and different types of vehicles can improve the model's robustness and generalizability. Additionally, incorporating advanced techniques such as transfer learning and fine-tuning on specific subsets of data can further enhance model performance. Integrating additional features, such as vehicle type classification and anomaly detection for identifying unusual parking behaviours, can provide deeper insights into vehicle movements. Furthermore, developing a real-time deployment pipeline with edge devices can enable immediate application in parking facilities, enhancing the scalability and practical utility of the system. Lastly, exploring the integration of this model with other smart city solutions, such as traffic management systems and automated toll collection, can create a comprehensive intelligent transportation ecosystem, improving overall urban mobility and infrastructure management.

# Team Member Contribution Report

## Soumyajit Roy

- Specific tasks and responsibilities handled
  - Led the overall project planning and coordination.
  - Designed and implemented data visualization and insights generation using Seaborn and Matplotlib.
  - Designed the user interface and user experience for the web application.
  - Contributed to the data analysis and visualization efforts.

- Code sections written or major contributions to the project
  - Developed the frontend design for the web application, focusing on user experience and accessibility.
  - Code for server deployment and maintenance, ensuring reliable hosting for the web application.
  - Code for generating insights from parking data, including visualizations of vehicle entry and exit distributions, parking durations, and real-time occupancy patterns.

- Any challenges faced and how they were overcome
  - Designing an intuitive and user-friendly interface for the web application. Solution: Conducted user testing and iterated on the design based on feedback to improve usability.

- Collaboration with other team members
  - Worked closely with Manav Malhotra on integrating the license plate detection model with the web application.
  - Coordinated with Ishtaj Kaur Deol on data collection and annotation processes.

# Team Member Contribution Report

## Manav Malhotra

- Specific tasks and responsibilities handled
  - Focused on the development and implementation of the web application for real-time monitoring.
  - Integrated the AI model with the web app for data processing.
  - Managed the deployment of the application on the server and ensured its accessibility.

- Code sections written or major contributions to the project
  - Implemented frontend functionalities for displaying real-time vehicle movement data.
  - Contributed to data visualization scripts, particularly in generating parking insights and statistics.

- Any challenges faced and how they were overcome
  - Ensuring seamless integration between the AI model and the web application. Solution: Collaborated closely with Soumyajit Roy to synchronize the model output with the web application requirements.

- Collaboration with other team members
  - Worked with Swarnav Kumar on integrating the AI model with the web application.
  - Collaborated with Ishtaj Kaur Deol on designing the user interface and user experience aspects of the web application.

# Team Member Contribution Report

## Ishtaj Kaur Deol

- Specific tasks and responsibilities handled
  - Led the data collection and annotation process for training the AI model.
  - Assisted in data preprocessing and analysis.

- Code sections written or major contributions to the project
  - Scripts for data collection and annotation management.
  - Developed the frontend design for the web application, focusing on user experience and accessibility.
  - Assisted in the development of data visualization scripts for insights generation.

- Any challenges faced and how they were overcome
  - Ensuring high-quality data annotation for training the AI model.
    Solution: Implemented a rigorous data annotation process and performed multiple rounds of validation.

- Collaboration with other team members
  - Collaborated with Soumyajit Roy on data preprocessing and analysis tasks.
  - Coordinated with Manav Malhotra on ensuring the web application met user requirements and was easy to use.

# Team Member Contribution Report

## Swarnav Kumar

- Specific tasks and responsibilities handled
  - Developed and implemented the license plate detection model.
  - Managed data preprocessing and training the YOLOv8n model for license plate recognition.

- Code sections written or major contributions to the project
  - Code for importing libraries and dependencies, including OpenCV, Pandas, NumPy, and PyTorch.
  - Data preprocessing scripts for image and annotation handling.
  - Model training script for YOLOv8n.

- Any challenges faced and how they were overcome
  - Ensuring accurate detection of license plates in various lighting conditions. Solution: Enhanced the training dataset with diverse images and performed extensive model tuning.

- Collaboration with other team members
  - Worked closely with Manav Malhotra on integrating the license plate detection model with the web application.
  - Coordinated with Ishtaj Kaur Deol on data collection and annotation processes.