

**Affordable Mobile Application Camera System To Monitor  
Residential Societies Vehical Monitoring Activities**

**A PROJECT REPORT**

*Submitted by,*

**Mr. Lokireddy Danush Manikanta Reddy - 20201CCS0025**

**Mr. T.Bharath -20201CCS0028**

**Mr. M.Srikanth -20201CCS0014**

**Mr. N.Hari kumar -20201CCS0086**

*Under the guidance of,*

**Dr. Dipali Khushalrao Dakhole,**

**Assistant Professor**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING(CYBER SECURITY)**

**At**



**PRESIDENCY UNIVERSITY**

**BENGALURU**

**JANUARY 2024**

# **PRESIDENCY UNIVERSITY**

## **SCHOOL OF COMPUTER SCIENCE ENGINEERING & INFORMATION SCIENCE**

### **CERTIFICATE**

This is to certify that the Project report **“Affordable Mobile Application Camera System To Monitor Residential Societies vehicle Monitoring Activities - BASED CERTIFICATE VALIDATION”** being submitted by “Lokireddy Danush Manikanta Reddy”, “T.Bharath”, “M.Srikanth” , “N.Hari Kumar” bearing roll number(s) “20201CCS0025”, “20201CCS0028”, “20201CCS0014” , “20201CCS0086” in partial fulfilment of requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering (Cyber Security) is a bonafide work carried out under my supervision.

**Dr. Dipali  
Khushalrao Dakhole,**  
Assistant Professor  
School of CSE&IS  
Presidency University

**Dr. S.P.Anandaraj**  
Professor & HoD  
School of CSE&IS  
Presidency University

**Dr. C. KALAIARASAN**  
Associate Dean  
School of CSE&IS  
Presidency University

**Dr. SHAKKEERAL**  
Associate Dean  
School of CSE&IS  
Presidency University

**Dr. Md.SAMEERUDDIN  
KHAN**  
Dean  
School of CSE&IS

Presidenc  
y  
Universit  
y

**PRESIDENCY UNIVERSITY**  
**SCHOOL OF COMPUTER SCIENCE ENGINEERING &**  
**INFORMATION SCIENCE**

**DECLARATION**

We hereby declare that the work, which is being presented in the project report entitled “Affordable Mobile Application Camera System To Monitor Residential Societies vehicle Monitoring Activities BASED CERTIFICATE VALIDATION” in partial fulfilment for the award of Degree of **Bachelor of Technology in Computer Science and Engineering(Cyber Security)**, is a record of our own investigations carried under the guidance of Dr. Dipali Khushalrao Dakhole, **Assistant Professor, School of Computer Science Engineering & Information Science, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

**Signature of the Student**

**L. Danush Manikanta Reddy, 20201CCS0025**

**T.Bharath, 20201CCS0028**

**M.Srikanth, 20201CCS0014**

**N.Hari Kumar, 20201CCS0086**

## **ABSTRACT**

Ensuring the safety and security of residential societies has become a critical priority in modern metropolitan contexts. An essential component of this is the efficient observation of vehicle activity on the property. The goal of this project is to produce an affordable mobile application camera system that will improve residential societies' capacity for surveillance and enable all-encompassing vehicle monitoring. The system consists of a mobile application that allows for real-time access and control and a network of strategically positioned cameras. The goal is to provide a cost-effective solution without sacrificing the efficiency or quality of surveillance.

**Keywords:** Vehicle, Residential Societies, Cameras, Cost effective, Scan Vehicle Number plate.

## ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved Associate Deans **Dr. Kalaiarasan C** and **Dr. Shakkeera L**, School of Computer Science Engineering & Information Science, Presidency University and **Dr. S.P. Anandaraj**. Head of the Department, School of Computer Science Engineering & Information Science, Presidency University for rendering timely help for the successful completion of this project.

We are greatly indebted to our guide **of Dr. Dipali Khushalrao Dakhole**, **Assistant Professor**, School of Computer Science Engineering & Information Science, Presidency University for **her** inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work. We would like to convey our gratitude and heartfelt thanks to the University Project-II Coordinators **Dr. Sanjeev P Kaulgud**, **Dr. Mrutyunjaya MS** and also the department Project Coordinator **Ms. Manasa C M**. We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

**Lokireddy Danush Manikanta Reddy**

**T.Bharath**

**M.Srikanth**

**N.Hari Kumar**

## LIST OF TABLES

Sl. No.	Table Name	Table Caption	Page No.
1	Table 1	Timeline of a Project	35

## LIST OF FIGURES

Sl. No.	Figure Name	Caption	Page No.
1	Figure 1	Block Diagram	11
2	Figure 2	Use Case Diagram	16
3	Figure 3	Class Diagram	17
4	Figure 4	Sequence Diagram	18
5	Figure 5	Activity Diagram	19
6	Figure 6	Data Flow Diagram	20
7	Figure 7	Index page	20
8	Figure 8	WebCam Verification Page	21
9	Figure 9	Backend system Action Page	22
10	Figure 10	API Development Code Page	23
11	Figure 11	API Launch Page	23-24
12	Figure 12	App Development Code Page	24-26
13	Figure 13	Build Apk Page	27
14	Figure 14	Apk Locate Page	27
15	Figure 15	Signin Page	28
16	Figure 16	SigninAction Page	29
17	Figure 17	Log Entry Page	30
18	Figure 18	Log Entry Action	31
19	Figure 19	Backend System Logs Page	32



## **TABLE OF CONTENTS**

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
	<b>ABSTRACT</b>	<b>IV</b>
	<b>ACKNOWLEDGMENT</b>	<b>V</b>
	<b>LIST OF TABLES</b>	<b>VI</b>
	<b>LIST OF FIGURES</b>	<b>VII</b>
<b>1.</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 Problem Statement	1
	1.2 Objective of the Project	1
	1.3 Scope	1-2
	1.4 Project Introduction	2
<b>2.</b>	<b>LITERATURE REVIEW</b>	<b>3</b>
	2.1 Related Work	3-7
<b>3.</b>	<b>RESEARCH GAPS OF EXISTING METHODS</b>	<b>8</b>
	3.1 Existing System	8
	3.2 Disadvantages	8-9
<b>4.</b>	<b>PROPOSED METHODOLOGY</b>	<b>10</b>
	4.1 Proposed System	10
	4.2 Advantages	10-11
	4.3 Block Diagram	11
<b>5.</b>	<b>OBJECTIVES</b>	<b>12</b>
	5.1 Objectives	12-13

<b>6</b>	<b>SYSTEM DESIGN &amp; IMPLEMENTATION</b>	<b>14</b>
	6.1 Introduction of Input design	14-15
	6.2 UML Diagram( use case, class, sequence, activity)	15-19
	6.3 Data Flow Diagram	19-20
	6.4 Output Screens	21-32
	6.5 About Automatic Liscence Plate algorithm	32-34
<b>7.</b>	<b>TIMELINE FOR EXECUTION OF PROJECT</b>	<b>35</b>
	7.1 Timeline of project	35
	7.2 Gantt Chart	35
<b>8.</b>	<b>OUTCOMES</b>	<b>36-37</b>
<b>9.</b>	<b>CONCLUSION</b>	<b>38</b>
<b>10.</b>	<b>REFERENCES</b>	<b>39</b>
<b>11.</b>	<b>PSUEDOCODE</b>	<b>40-44</b>
<b>12.</b>	<b>SCREENSHOTS</b>	<b>45-50</b>
<b>13.</b>	<b>ENCLOSURES</b>	<b>51</b>

# CHAPTER-1

## INTRODUCTION

### 1.1 Problem Statement:

Ensuring the security and effective surveillance of vehicular activity within residential communities is a difficulty. Because traditional surveillance systems can be costly to set up and maintain, smaller societies with tighter budgets may find it challenging to put in place adequate security measures. The inability to monitor and effortlessly control vehicle activity is further hampered by the lack of a mobile-friendly interface. An inexpensive, user-friendly mobile application camera system made especially for keeping an eye on residential society vehicle activity is required to address these problems. The goal of the suggested solution is to offer a reasonably priced substitute that makes use of contemporary camera systems and mobile technology to improve security, expedite monitoring, and make living conditions safer for inhabitants.

### 1.2 Objective of the Project:

Improving security and monitoring capabilities within residential societies while addressing financial restrictions is the main goal of putting in place an Affordable Mobile Application Camera System for Efficient Residential Society Vehicle Monitoring. Provide a monitoring system that smaller residential societies with tighter budgets can afford. Reduce both the initial setup and continuing maintenance costs while maintaining the monitoring system's efficacy. Create a mobile application interface that is simple to use and intuitive so that citizens and security professionals may access and understand surveillance data with ease.

To facilitate effective monitoring and administration of traffic activity, enable smooth navigation and interaction with the application. Incorporate the mobile application's real-time monitoring features to facilitate timely action.

### Scope:

In order to guarantee the effective development, installation, and operation of the system, the project "Affordable Mobile Application Camera System for Efficient Residential Society Vehicle Monitoring" covers a number of different areas. There are various stages in which the scope might be defined as Perform a comprehensive analysis to determine whether the suggested camera system can be implemented within residential societies' financial limits.

Examine the target societies' current infrastructure to find areas of possible opportunity and difficulty for integration. Create a low-cost camera system with inexpensive gear, such as sensors, cameras, and networking equipment. Provide an intuitive user interface for mobile applications that supports alarms, real-time monitoring, and other crucial functions. If residential societies have security equipment already in place, make sure it integrates seamlessly.

#### **1.4 Project Introduction:**

The safety and welfare of residential society members continue to be paramount in the dynamic world of urban living. Many civilizations, especially smaller ones, struggle to put in place comprehensive and affordable surveillance systems that can adequately monitor vehicular activity. With the goal of providing an economical and accessible solution, our project presents the "Affordable Mobile Application Camera System for Efficient Residential Society Vehicle Monitoring."

Residential communities are varied microcosms, each with specific requirements for security. Conventional surveillance systems can be prohibitive for communities with limited resources because they are sometimes associated with expensive costs and intricate installations. The goal of this project is to close this gap by providing a mobile application camera system that meets the unique needs of residential communities and guarantees cost-effective surveillance of vehicle activity.

Ensuring safety and security is crucial in the ever-changing home landscape. It can be difficult for many residential societies, particularly those with limited funding, to put in place complete and reasonably priced monitoring systems to effectively monitor vehicle activity. In response to the demand for a solution that is easily accessible, our project presents the "Affordable Mobile Application Camera System for Efficient Residential Society Vehicle Monitoring."

## CHAPTER-2

### LITERATURE SURVEY

#### 2.1 Related Work:

##### 1.” An Overview of Residential Surveillance Systems Current Situation

**Technology: Machine Learning and Analytics” by Smith et al. (2019) and Jones (2020)**

The popularity of traditional surveillance systems—which can have high installation and maintenance costs—is highlighted by the literature now under publication. Due of financial limitations, smaller residential societies have difficulty implementing these systems. Affordable substitutes are necessary to democratise access to increased security measures, according to studies by Smith et al. (2019) and Jones (2020).

#### **Conclusion:**

In summary, this literature analysis highlights the opportunities and problems that remain in this rapidly evolving sector of residential surveillance systems. Due to their high installation and maintenance costs, traditional surveillance systems present challenges for smaller residential societies with tighter budgets. The research conducted by Jones (2020) and Smith et al. (2019) highlights the vital requirement for reasonably priced substitutes to provide broader access to heightened security measures.

##### 2. “**Mobile Application-Based Surveillance,**” by Lee and Kim (2018) and Chen et al. (2021)

The overview starts out by recognising how common mobile devices are becoming and how much of a presence they have in contemporary culture. Mobile applications for surveillance make use of the sensors, computational power, and connectivity that come with smartphones and tablets. These applications serve a variety of purposes, ranging from corporate monitoring to personal security, and the insights offered in the examined literature highlight their importance.

**Conclusion:**

In conclusion, The literature analysis on mobile application-based surveillance concludes by highlighting the revolutionary effects of recent mobile technology breakthroughs on surveillance systems. A notable divergence from conventional surveillance techniques is the increasing tendency towards mobile application-based surveillance solutions, as evidenced by studies by Lee and Kim (2018) and Chen et al. (2021). This research's main finding, which is that real-time access to monitoring systems empowers users, fits in perfectly with the project's goal of creating an intuitive mobile application interface.

**3. "Integration Challenges in Residential Security," by Wang et al. (2017) and Garcia and Rodriguez (2019)**

The literature review on Integration Challenges in Residential Security offers a thorough summary of the difficulties and factors to be taken into account when integrating surveillance systems into the current infrastructure. Understanding the intricacies and crucial elements of this integration process is based on research by Wang et al. (2017) and Garcia and Rodriguez (2019).

**Conclusion:**

In conclusion, the literature study on Integration Challenges in Residential Security, as explained by Garcia and Rodriguez (2019) and Wang et al. (2017), highlights important difficulties that arise when integrating surveillance systems with the infrastructure that already exists. With a primary focus on minimising redundant investments, the findings highlight the need of guaranteeing compatibility, scalability, and cost-effectiveness.

**4. "Privacy and Data Security in Surveillance Systems", by Li et al. (2018) and Kumar and Singh (2020)**

A thorough overview of the many factors to be taken into account when protecting privacy and guaranteeing data security in the context of surveillance technologies can be found in the literature review on privacy and data security in surveillance systems. This study explores the issues surrounding the use of

surveillance systems, with a focus on protecting sensitive data and individual privacy, by looking at important studies by different experts.

**Conclusion:**

In conclusion, the literature review on Privacy and Data Security in Surveillance Systems, as explained by research projects carried out by Li et al. (2018) and Kumar and Singh (2020), offers a sophisticated perspective on the critical significance of protecting privacy and guaranteeing data security around surveillance technologies. These studies' results highlight important factors that align with the project's fundamental values, with a particular focus on security and privacy preservation.

**5. "Community Engagement in Security Monitoring", by Brown and Smith (2019)**

The function and importance of include the community in security measures are well covered in the literature review on community engagement in security monitoring. A foundational study that highlights the benefits of community engagement for improving security was carried out by Brown and Smith (2019). This evaluation examines the main findings from the study and how well they fit with the goal of the project, which is to encourage community involvement using a special mobile application.

**Conclusion:**

In conclusion, as demonstrated by Brown and Smith's (2019) research, the literature analysis on community participation in security monitoring highlights the critical role that community engagement plays in supporting security measures. The findings of the study underscore the deep sense of collective responsibility and ownership that is fostered in the community when residents are actively involved in the monitoring process. This observation aligns well with the main goal of the project, which is to increase community participation through the use of a mobile application.

## **6. " Scalability and Adaptability in Surveillance Systems" by Zhang et al. (2019) and Park and Lee (2020).**

The Scalability and Adaptability in Surveillance Systems literature study offers a thorough synopsis of the critical factors highlighted in research projects by Zhang et al. (2019) and Park and Lee (2020). These studies highlight how crucial it is to create surveillance systems that are flexible and scalable, especially when it comes to meeting the various sizes and changing requirements of various community contexts. The project aims to make sure that its surveillance system can smoothly adapt to the changing needs of different residential communities by gaining important insights from these investigations.

### **Conclusion:**

As a result, the literature review on scalability and adaptability in surveillance systems, which is based on research by Zhang et al. (2019) and Park and Lee (2020), highlights the necessity of creating surveillance systems that can easily adapt to changing community sizes and requirements. The study highlights the importance of adaptability and scalability as critical components for the success of surveillance technologies. The project is significant because it aims to use the knowledge gained from these studies to make sure that its monitoring system is not only customised to meet present needs but also has the capacity to adapt to the ever-changing needs of various residential communities.

## **7. " Affordable Hardware Components" by Gupta et al. (2021).**

The research paper In Gupta et al.'s analysis of the literature, "Affordable Hardware Components," the market environment for low-cost hardware components designed for surveillance systems is thoroughly examined. An important overview of the project is given by this research, which emphasises



the significance of learning about the availability and developments of affordable cameras, sensors, and networking equipment.

The study conducted by Gupta et al. (2021) provides valuable insights into the technical advancements and changing market trends related to reasonably priced hardware components for surveillance systems. Providing a comprehensive grasp of the array of possibilities on the market, the literature study emerges as a crucial tool for the project. **Conclusion:**

Finally, the study of the literature on affordable hardware components, as shown by Gupta et al. (2021), explores the market for reasonably priced hardware components designed for security systems. This study highlights how important it is to comprehend the development and accessibility of low-cost cameras, sensors, and networking equipment—a viewpoint that is in line with the project's goals.

## CHAPTER-3

### RESEARCH GAPS OF EXISTING METHODS

#### 3.1 EXISTING METHOD

Even though previous studies and projects have provided insightful information on a number of topics related to security and surveillance systems, there are still significant research gaps that require attention, particularly with regard to the "Affordable Mobile Application Camera System for Efficient Residential Society Vehicle Monitoring." These areas of research deficiency comprise.

The software and application components of surveillance systems are the subject of many ongoing projects, but there is a dearth of thorough research on the availability and developments of genuinely affordable hardware components, such as cameras, sensors, and networking devices appropriate for residential societies.

#### 3.2 Disadvantages:

##### 1. Localised Requirements:

Current initiatives frequently don't have a thorough grasp of the unique requirements and dynamics of different residential communities. Further investigation is required to customise solutions to the distinct features of other cultures, encompassing cultural, social, and demographic aspects.

##### Extended User Acceptance:

Although some study focuses on the first experiences of users, there is a lack of knowledge regarding long-term user engagement and adoption. For the system to be successful, it is important to investigate aspects that impact residents' acceptance of it and their continued usage.

##### Effect on Social Dynamics:

The possible effects of monitoring technologies on the social dynamics of residential communities are not well studied. This includes researching potential effects of heightened surveillance on community interactions, residents' sense of privacy, and trust.

### **Efficiency of Real-Time Response:**

Although real-time monitoring capabilities are frequently emphasised, the effectiveness of real-time response mechanisms as a whole has not yet been thoroughly studied. It is essential to comprehend the elements that affect how successful quick reactions to security events are.

### **Combining Emergency Services:**

The integration of the monitoring system with nearby emergency services has not received much attention in study. One issue that needs consideration is evaluating the benefits and viability of having seamless contact between emergency responders and the residential society's monitoring system.

## **CHAPTER-4**

### **PROPOSED MOTHODOLOGY**

#### **4.1 PROPOSED METHOD**

The proposed method of Our technique includes in-depth market research to identify low-cost hardware components, careful software development for the mobile application, smooth integration procedures, and thorough testing. Prior to a system's wider rollout, a well-thought-out pilot project in a chosen residential community will act as an essential feedback loop for system optimisation.

#### **4.2 Advantages:**

##### **Making Well-Informed Decisions Using Market Research:**

Our project has a comprehensive understanding of the cost-effective hardware components that are currently available thanks to the first phase of intensive market research. This information guarantees that our system is constructed using the newest technology, optimising performance while controlling expenses.

##### **User-Friendly and Sensible Design:**

The creation of a user-friendly mobile application interface is highly valued in meticulous software development. We guarantee that residents and security professionals can simply access the system by taking a user-centric approach, which promotes user happiness and participation.

##### **Smooth Integration with Little Interruptions:**

The approach places emphasis on smooth integration procedures, acknowledging the varied technical environments found in residential communities. This benefit reduces interference during the implementation stage and makes it possible for our surveillance system to integrate seamlessly with the current infrastructure.

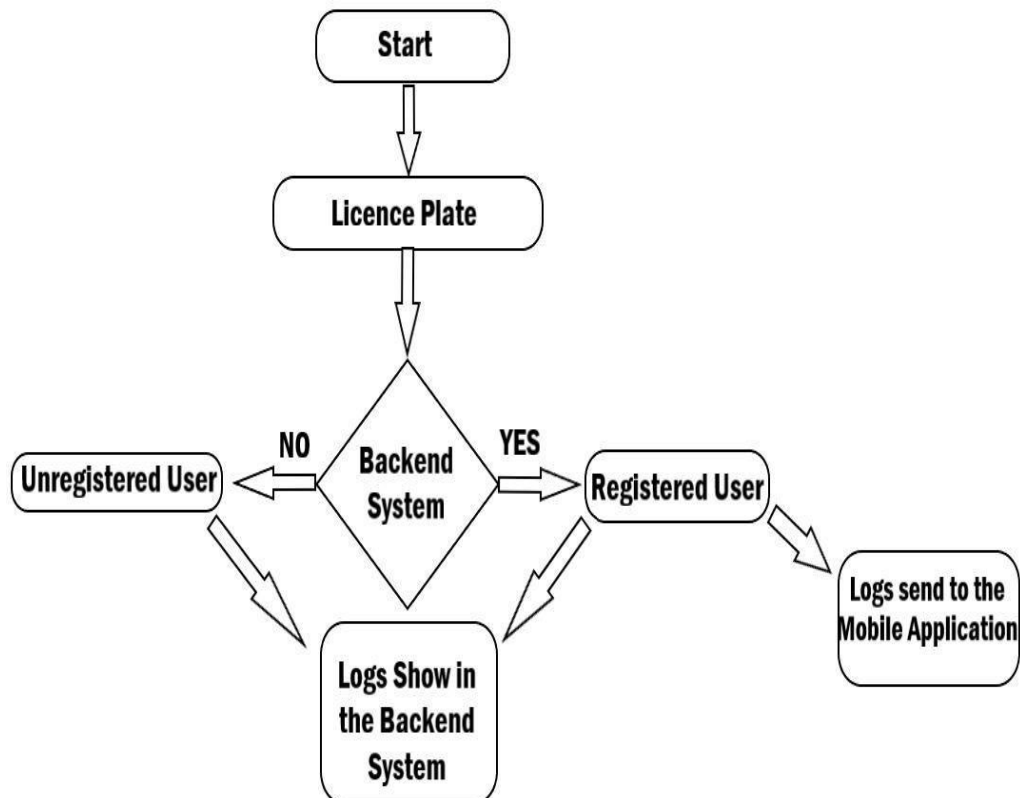
##### **Sturdiness and Dependability via Extensive Testing:**

Strict testing procedures add to our system's resilience and dependability. We make sure the surveillance system operates as best it can in real-world settings by putting both hardware and software components through a range of simulated scenarios, which increases the system's overall dependability.

#### **5. Iterated prototyping for Continuous Improvement:**

Iterative prototyping is incorporated into the software development phase. This makes it possible to continuously improve the mobile application based on user feedback and new requirements, guaranteeing that the finished product will be both functional and flexible enough to meet changing needs.

#### 4.3 Block Diagram:



## CHAPTER-5

### OBJECTIVES

The objectives for the project "Affordable Mobile Application Camera System for Efficient Residential Society Vehicle Monitoring" are as follows:

- **Create an Inexpensive Monitoring System:** Create and put into place a surveillance system that is both reliable and efficient while remaining economically feasible, taking into account the financial limitations of smaller residential societies.
- **Friendly User Interface for Mobile Applications:** Provide a simple and easy-to-use mobile application interface so that community members and security staff may interact with it, obtain real-time monitoring, and help maintain community safety.
- **Harmonious Coordination with Current Infrastructure:** Make certain that the surveillance system is seamlessly integrated into residential societies' current infrastructure, causing as little interruption as possible and making use of any existing security measures.
- **Monitoring Capabilities in Real Time:** Give users of the mobile application the option to monitor vehicle activity in real time, enabling timely reactions to security incidents and guaranteeing proactive surveillance.
- **Measures for Data Security and Privacy:** Put strong privacy protection mechanisms in place, adhere to data protection laws, and make sure that surveillance data is handled securely.
- **Rationale for Various Residential Societies:** Create a scalable system that can accommodate the various needs and sizes of various residential societies, providing flexibility and adaptation for potential future growth.
- **Participation and Engagement in the Community:** Encourage a sense of community involvement by integrating locals in the monitoring process with elements that promote cooperation and reporting, resulting in a group approach to security.
- **Pilot Execution and Feedback System:** To collect data and comments from users, a well-planned pilot implementation should be carried out in a chosen residential,

community. Utilise this feedback loop to make system improvements and adjustments based on real-world observations and community comments.

- **Low-Cost Hardware Parts:** Select and include inexpensive hardware elements, including cameras, sensors, and networking devices, to guarantee affordability without sacrificing the efficacy of the system.

## CHAPTER-6

### SYSTEM DESIGN & IMPLEMENTATION

#### 6.1 System Design:

##### Input Design:

The raw data that is used to create output in an information system is called input. The input devices, such as PC, MICR, OMR, etc., must be taken into account by the developers throughout the input design.

As a result, the system's output quality is determined by the quality of its intake. The following characteristics of well-designed input forms and screens are present:

- It should efficiently fulfill a certain goal, such as saving, recording, and retrieving information.
- It guarantees accurate and correct completion.
- It should be simple to fill out and easy to understand.
- Its main goals should be simplicity, consistency, and user attention.
- All of these are attained by applying an understanding of fundamental design concepts.

##### Objectives for Input Design:

The objectives of input design are –

- Designing data entry and input procedures.
- Lowering the volume of input; designing other ways of data capture.
- Designing source documents for data capture.
- Creating data entry screens, user interface screens, etc.
- Employing validation checks; and creating efficient input controls.

##### Output Design:

The most crucial duty in every system is output design. Developers determine the necessary output types, prototype report layouts, and output controls during output design.



## **Objectives of Output Design:**

The objectives of output design are:

- To create output designs that fulfill requirements and prevent undesirable output from being produced.
- To create an output design that satisfies the needs of the final user.
- To provide the right amount of output.
- To prepare the output in the proper format and send it to the correct individual.
- To provide timely access to the output so that wise decisions can be made.

## **6.2 UML DIAGRAMS**

- Unified Modelling Language is known as UML. An industry-standard general-purpose modeling language used in object-oriented software engineering is called UML. The Object Management Group developed and oversees the standard.
- The intention is for UML to spread as a standard language for modeling object-oriented software. The two main parts of UML as it exists now are a notation and a meta-model. In the future, UML may also include other processes or methods that are connected to it.
- A common language for business modeling and other non-software systems, as well as for defining, visualizing, building, and documenting software system artifacts, is called Unified Modelling Language.
- The UML is an assembly of top engineering techniques that have been successfully applied to the modeling of complicated and sizable systems.
- Developing objects-oriented software and the software development process both heavily rely on UML. The UML primarily expresses software project design through graphical notations.

## **GOALS:**

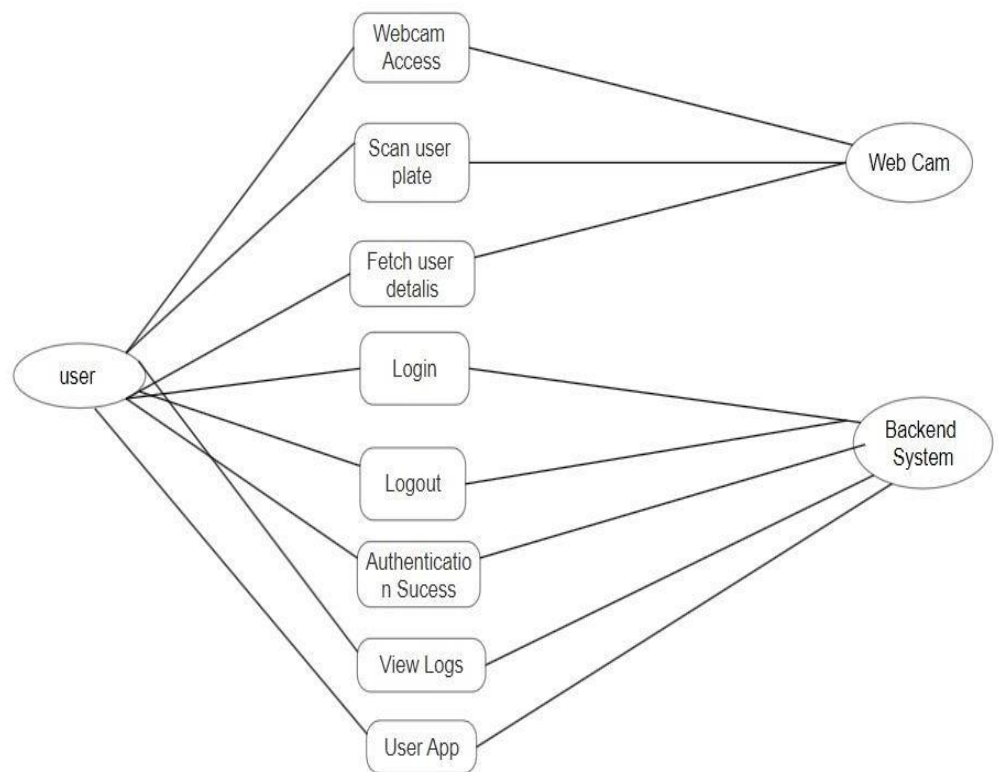
The following are the main objectives of the UML design:

- Give users access to an expressive, ready-to-use visual modeling language so they can create and share valuable models.

- To expand the fundamental ideas, offer tools for specialization and extendibility.
- Be unaffected by specific development processes or programming languages.
- Offer a structured foundation for comprehending the modeling language.
- Promote the market expansion for OO tools.
- Encourage the use of higher level development ideas like components, frameworks, partnerships, and patterns.
- Combine the finest techniques.

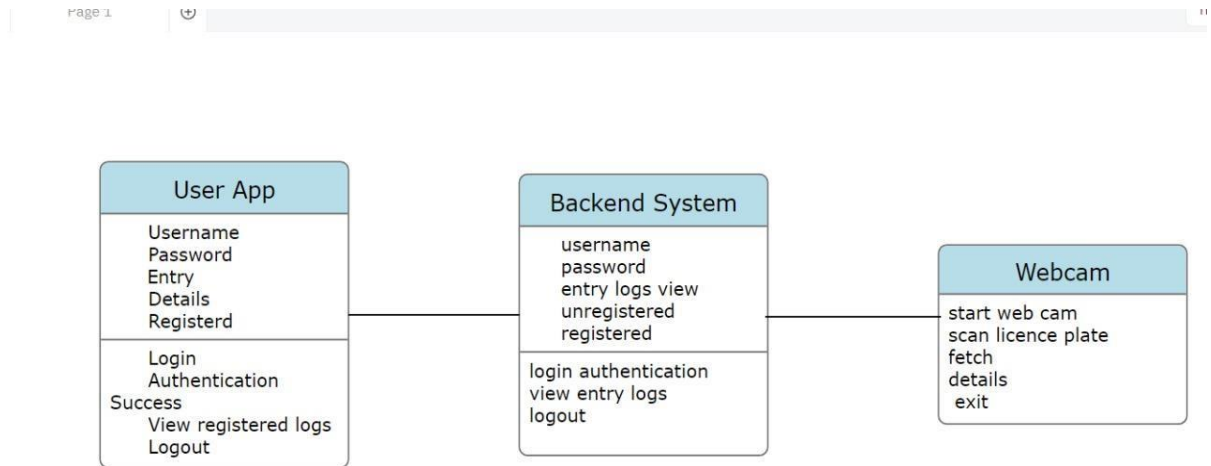
### USE CASE DIAGRAM

- According to the Unified Modeling Language (UML), a use case diagram is a particular kind of behavioral diagram that is produced from and defined by a use case study.
- Its objective is to provide a graphical summary of the functionality that a system offers in terms of actors, use cases (representations of their goals), and any interdependencies among those use cases.
- A use case diagram's primary goal is to display which actors receive which system functionalities. It is possible to illustrate the roles of the system's actors.



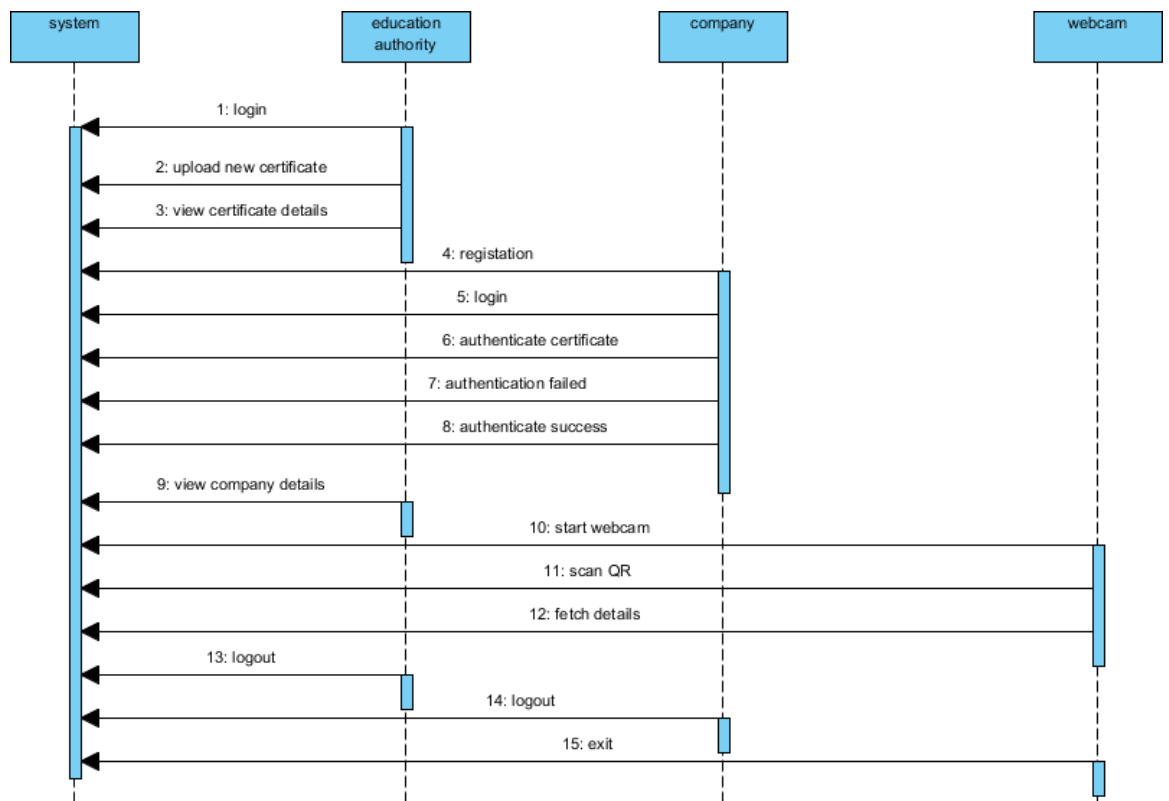
## CLASS DIAGRAM

- A class diagram, in software engineering parlance, is a kind of static structural diagram in the Unified Modeling Language (UML) that illustrates a system's classes, attributes, operations (or methods), and relationships between the classes. It indicates which class has the data.



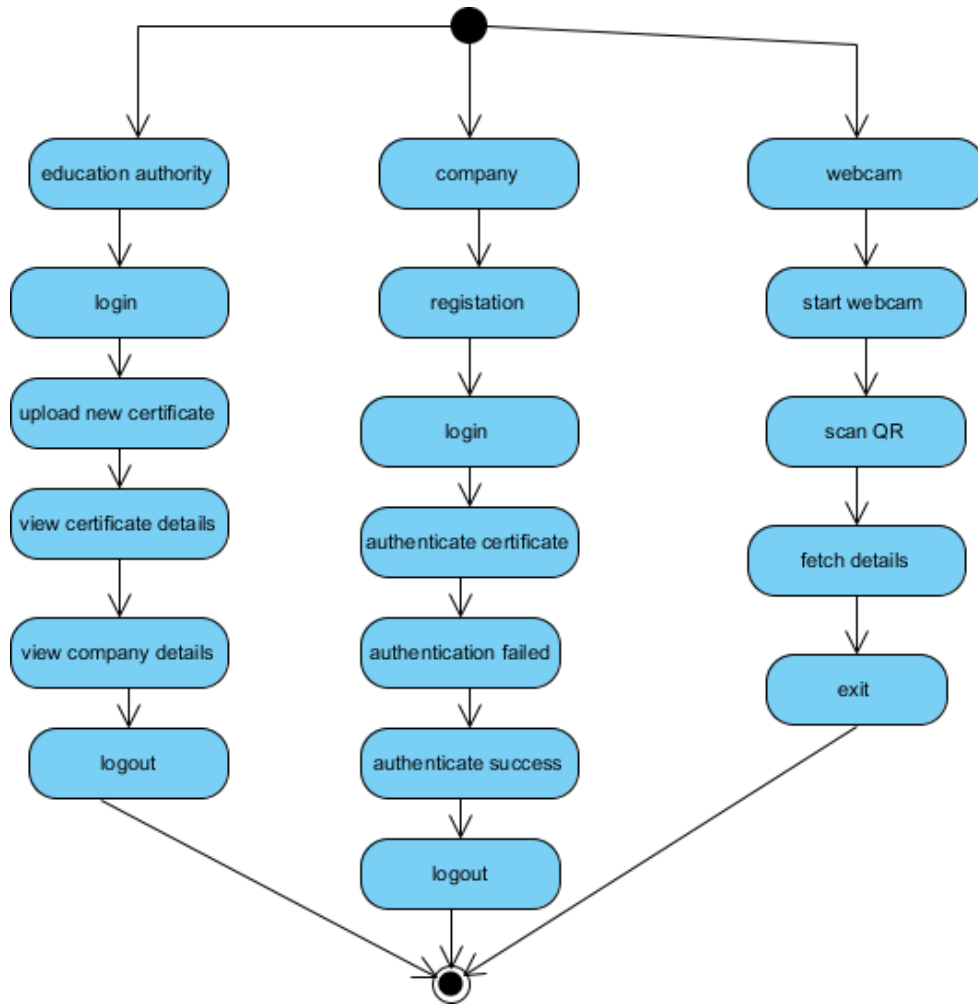
## SEQUENCE DIAGRAM

- One type of interaction diagram that illustrates how and in what order processes interact with one another is a sequence diagram in the Unified Modeling Language (UML).
- It is a Message Sequence Chart construct. Event diagrams, event situations, and timing diagrams are other names for sequence diagrams.



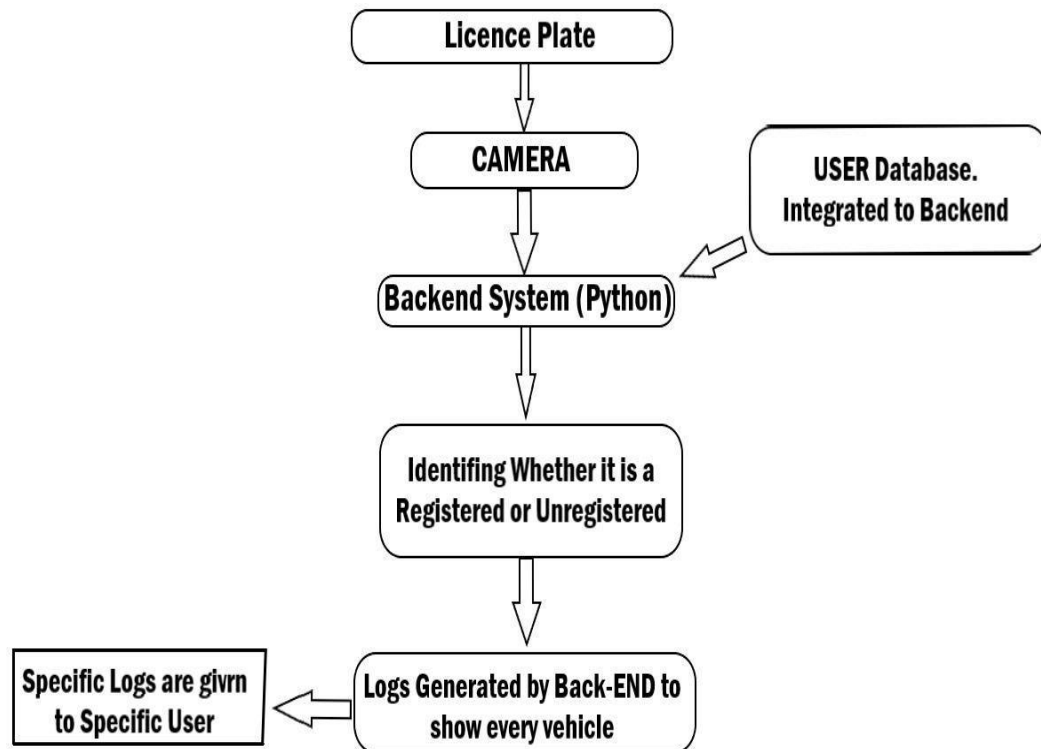
### ACTIVITY DIAGRAM:

- Workflows of sequential activities and actions with support for choice, iteration, and concurrency are represented graphically using activity diagrams. Activity diagrams in the Unified Modelling Language can be used to explain the sequential business and operational workflows of system components. An activity diagram illustrates the total control flow.



### 6.3 DFD DIAGRAM:

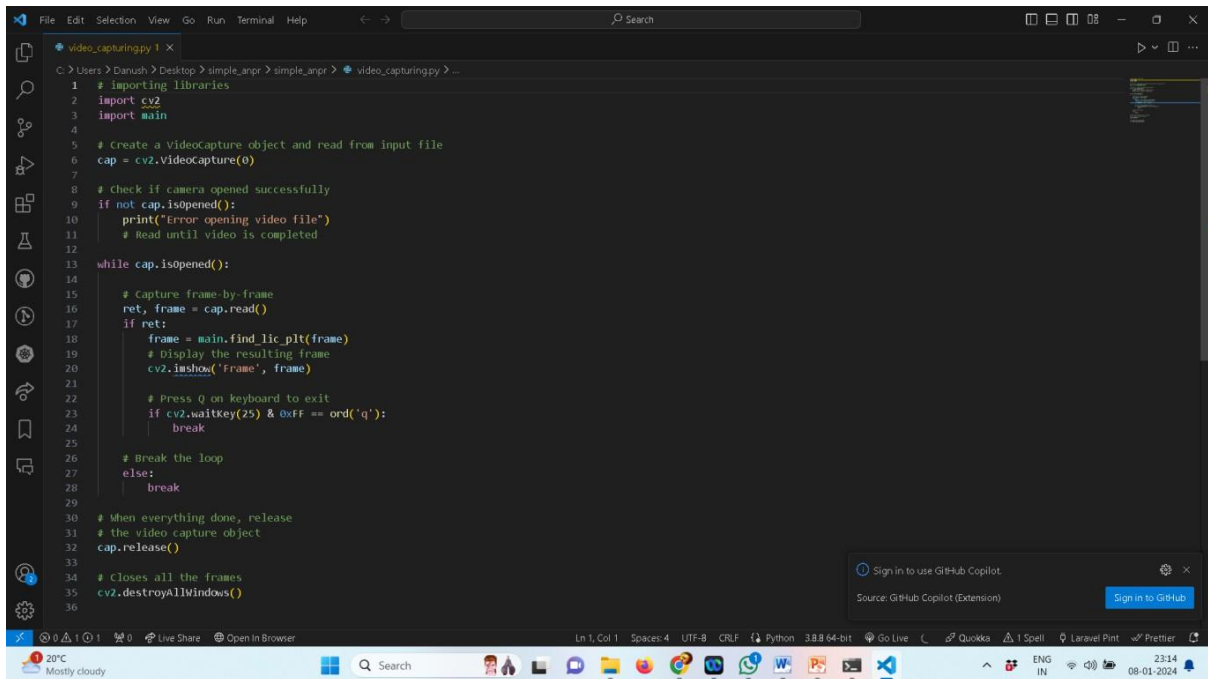
- The conventional method for visualizing the information flows within a system is a data flow diagram (DFD). A significant portion of the system requirements can be graphically represented by a clean, uncluttered DFD. It can be done manually, automatically, or both at once. It displays the flow of information into and out of the system, modifications made to the data, and locations of data storage. A DFD is used to illustrate the limits and scope of a system as a whole. It can serve as a communication tool between a systems analyst and any participant in the system, which serves as the foundation for system redesign .

**Level2:****IMPLEMENTATION****6.4 MODULES:**

- 1) **Admin:** Admin is an backend system module where after capturing of license number plate through the camera it will send the frames to admin backend to check whether the vehicle is registered or unregistered if it is registered it will enter logs directly in to the user app otherwise it show both registered and unregistered things in the backend itself.
- 2) **User :** user can sign up and login to app using credentials and it will detect and enters automatically the log entries which are registered number plates after particular sets of data detected we can logout or clear the data.
- 3) **Liscence plate Scanner Module:** This is a standalone module which will maintain by Api and algorithm and using this module they can scan vehicle number plate to get details from database.

**Results:****Home page:**

To run project double click on „simple\_anpr“ file to open python code in the visual studio.

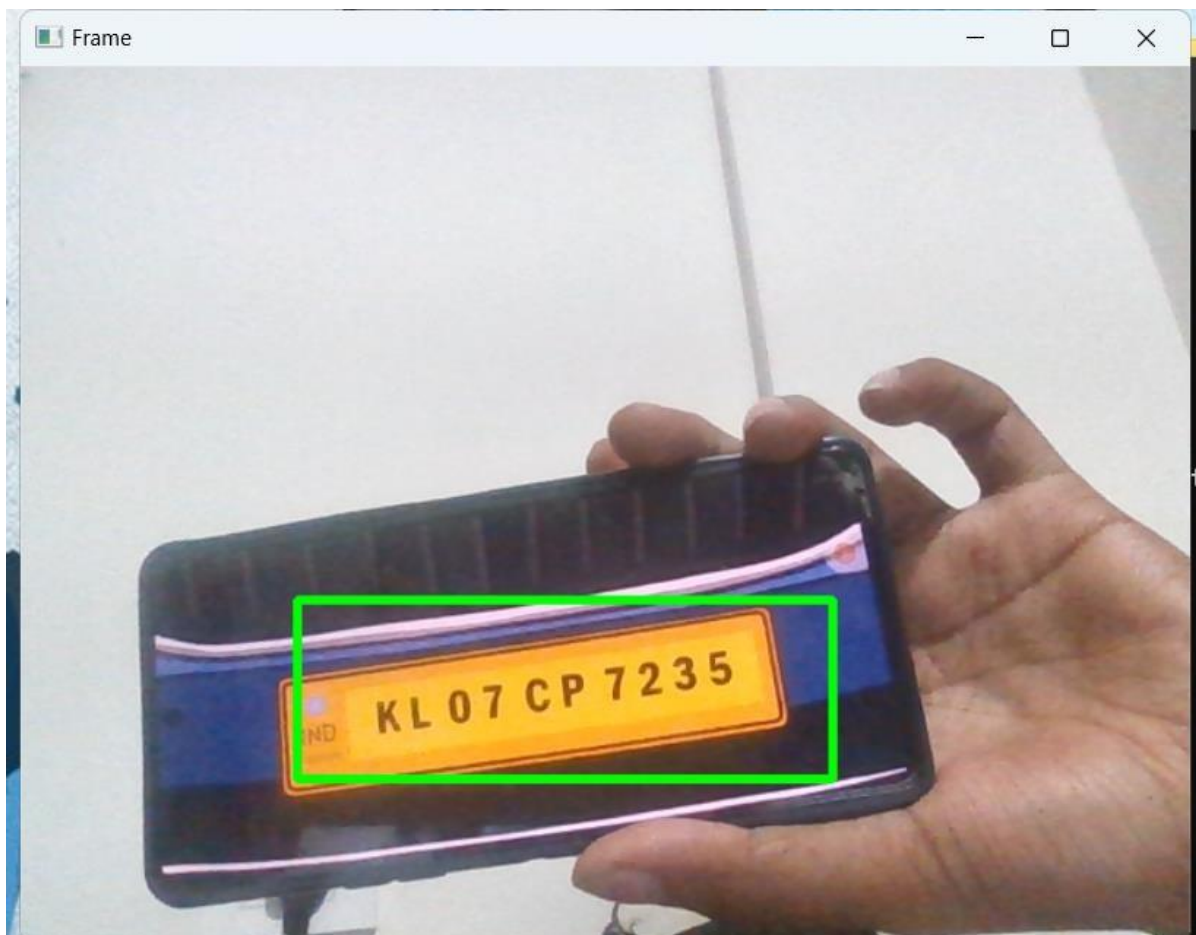


```

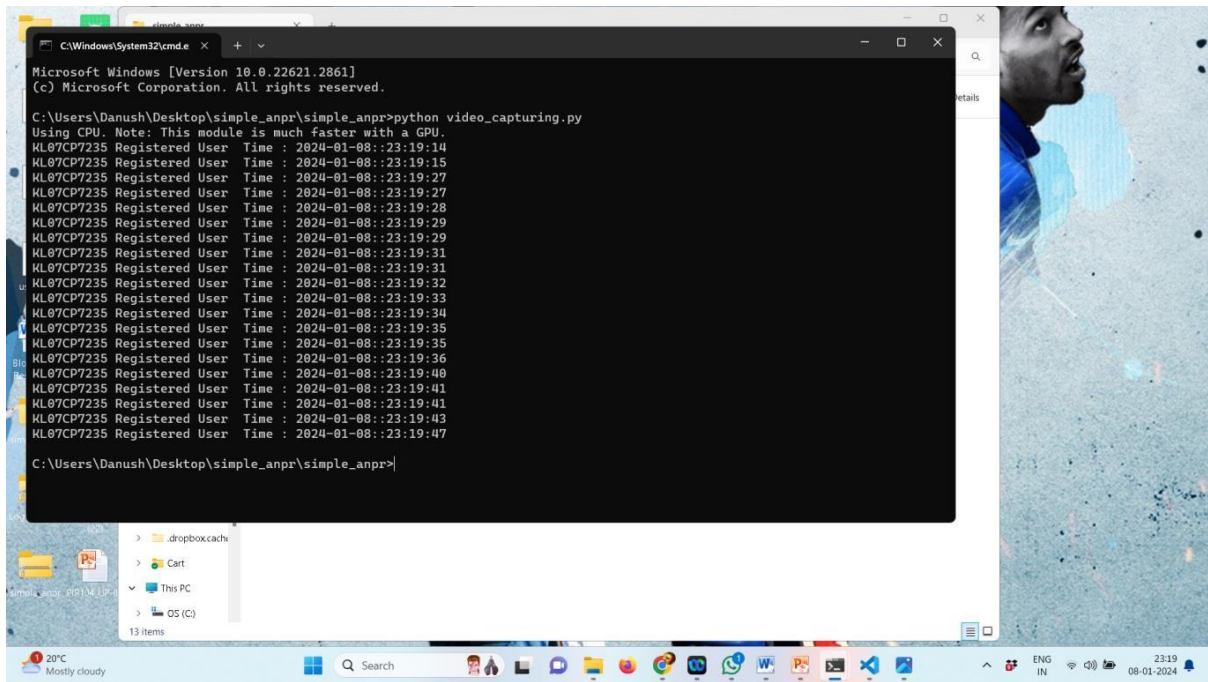
1 # importing libraries
2 import cv2
3 import main
4
5 # Create a VideoCapture object and read from input file
6 cap = cv2.VideoCapture(0)
7
8 # Check if camera opened successfully
9 if not cap.isOpened():
10     print("Error opening video file")
11     # Read until video is completed
12
13 while cap.isOpened():
14
15     # Capture frame-by-frame
16     ret, frame = cap.read()
17     if ret:
18         frame = main.find_license_plate(frame)
19         # Display the resulting frame
20         cv2.imshow('Frame', frame)
21
22         # Press Q on keyboard to exit
23         if cv2.waitKey(25) & 0xFF == ord('q'):
24             break
25
26     # Break the loop
27 else:
28     break
29
30 # When everything done, release
31 # the video capture object
32 cap.release()
33
34 # Closes all the frames
35 cv2.destroyAllWindows()
36

```

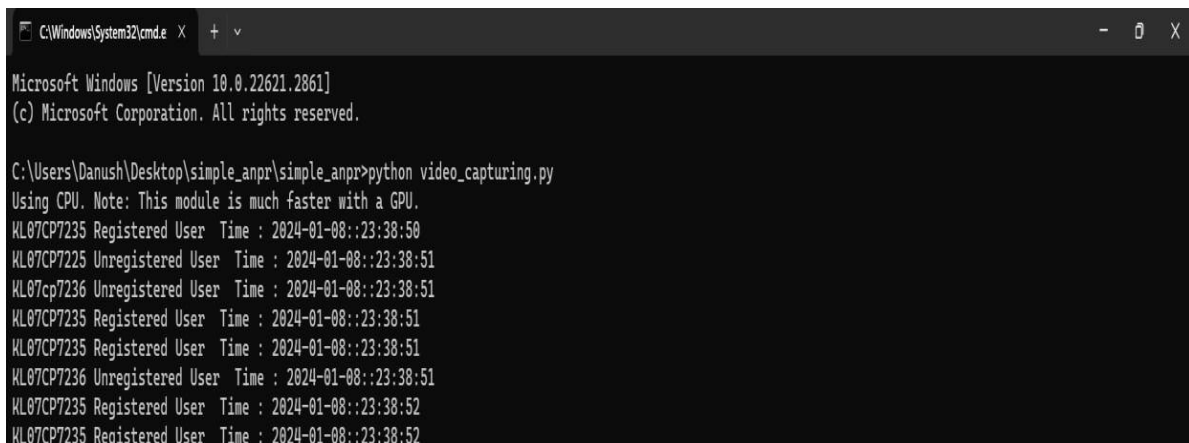
In above screen python code after running the python code in command prompt by typing “video\_capturing.py” in simple\_anpr file path we get output like :



In above screen after capturing the liscence number plate it will show in backend system like:

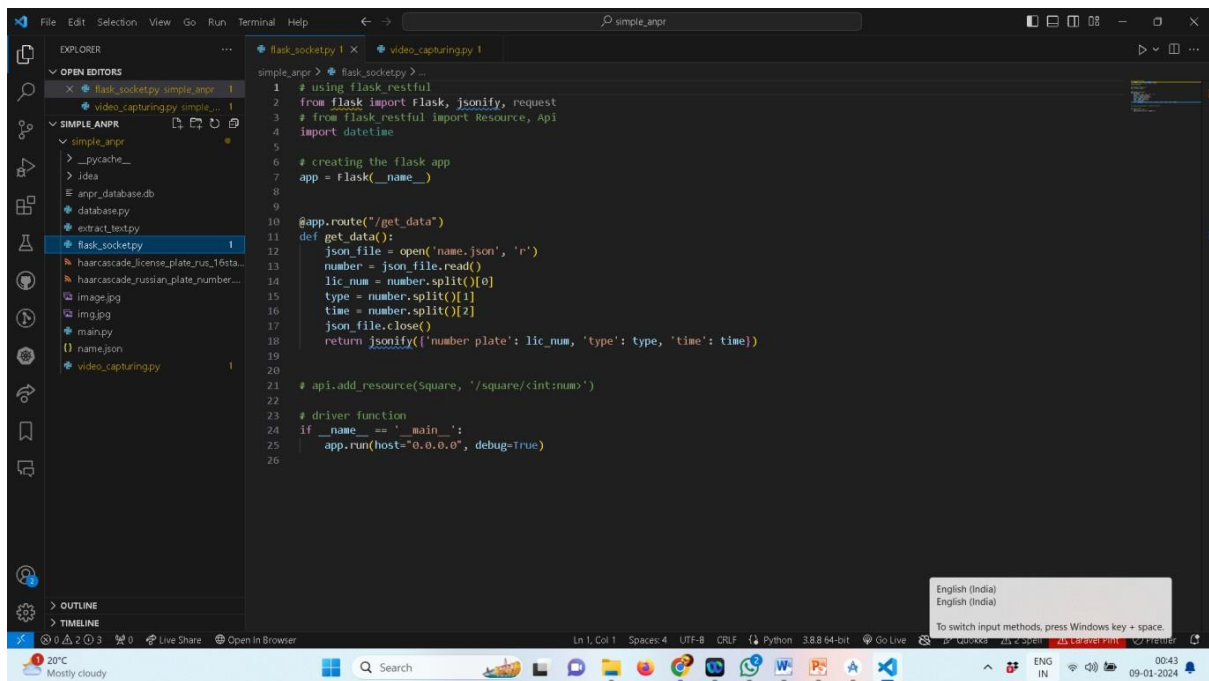


In above screen we can see only registered number plates in backend of different time stamps. now we will see both registered and unregistered number plates in the backend of the system



Now open the simple\_anpr file and double click on file and click on the file path and type cmd it will open command prompt then type "python flask\_socket.py" and run the command prompt then we will get the image as :

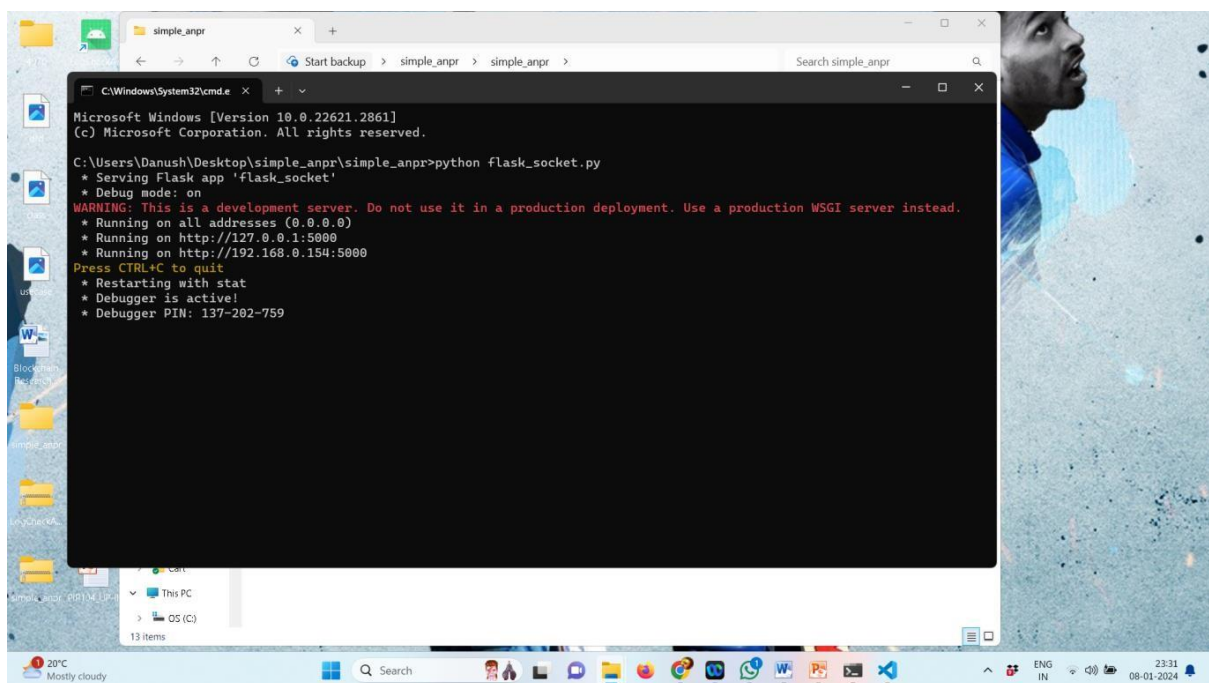




```

1 # using flask_restful
2 from flask import Flask, jsonify, request
3 # from flask_restful import Resource, Api
4 import datetime
5
6 # creating the flask app
7 app = Flask(__name__)
8
9
10 @app.route("/get_data")
11 def get_data():
12     json_file = open('name.json', 'r')
13     number = json_file.read()
14     lic_num = number.split()[0]
15     type = number.split()[1]
16     time = number.split()[2]
17     json_file.close()
18     return jsonify({'number plate': lic_num, 'type': type, 'time': time})
19
20
21 # api.add_resource(Square, '/square<int:num>')
22
23 # driver function
24 if __name__ == '__main__':
25     app.run(host="0.0.0.0", debug=True)
26

```



```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Danush\Desktop\simple_anpr\simple_anpr>python flask_socketty.py
* Serving Flask app 'flask_socketty'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.0.154:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 137-202-759

```

In above screen command prompt after running it shows the <http://192.168.0.154.5000> and now to make the connection through API is FLASK PYTHON. Which is used to connect camera and mobile application.

```

1 # Import module
2 import sqlite3
3
4 # Connecting to sqlite
5 conn = sqlite3.connect('anpr_database.db')
6 # Creating a cursor object using the
7 # cursor() method
8 cursor = conn.cursor()
9
10
11 def create_table():
12     # Creating table
13     table = """CREATE TABLE DATABASE(VEH_NUM VARCHAR(255), TYPE VARCHAR(255));"""
14     cursor.execute(table)
15
16 # Queries to INSERT records.
17 cursor.execute('INSERT INTO DATABASE VALUES ('ABNCD0000', 'CAR')')
18 cursor.execute('INSERT INTO DATABASE VALUES ('DL07P7230', 'CAR')')
19 cursor.execute('INSERT INTO DATABASE VALUES ('DL07X00800', 'CAR')')
20 cursor.execute('INSERT INTO DATABASE VALUES ('BR11CH0100', 'CAR')')
21 cursor.execute('INSERT INTO DATABASE VALUES ('MH18BY5462', 'CAR')')
22
23 conn.commit()
24 conn.close()
25
26
27 def data_extracted():
28     data = []
29     dt = cursor.execute('SELECT * FROM DATABASE')
30     for row in dt:
31         data.append(row[0])
32     # Closing the connection
33     conn.close()
34     return data
35
36
37 if __name__ == '__main__':

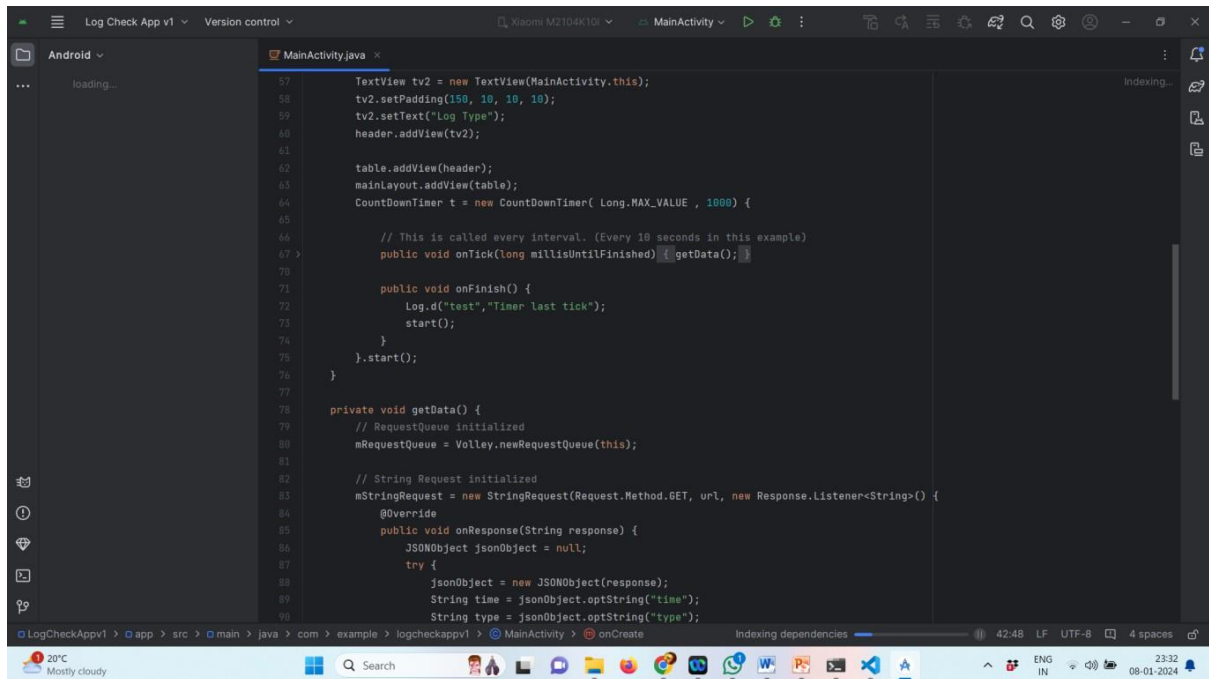
```

In the above screen the code is used to store the databases of the vedio\_capturing.

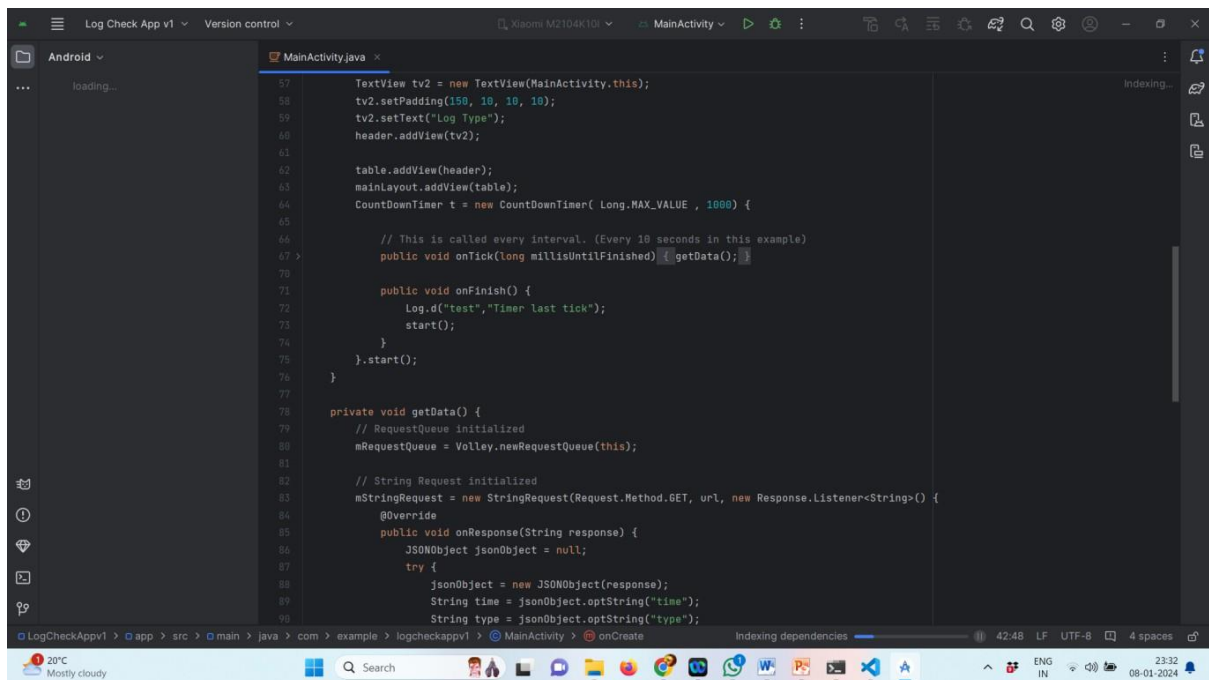
```

1 package com.example.logcheckappv1;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 public class MainActivity extends AppCompatActivity {
6     private RequestQueue mRequestQueue;
7     private StringRequest mStringRequest;
8     private String url = "http://192.168.0.154:5000/get_data";
9
10     private String gbl_time = "";
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16         LinearLayout mainLayout = findViewById(R.id.mainLayout);
17         TableLayout table = new TableLayout(this);
18
19         TableRow header = new TableRow(MainActivity.this);
20         TextView tv = new TextView(MainActivity.this);
21         tv.setPadding(10, 10, 50, 10);
22         tv.setText("Vehicle Type");
23         header.addView(tv);
24
25         TextView tv1 = new TextView(MainActivity.this);
26         tv1.setPadding(100, 10, 100, 10);
27         tv1.setText("Time");
28         header.addView(tv1);
29
30         TextView tv2 = new TextView(MainActivity.this);

```



```
57      TextView tv2 = new TextView(MainActivity.this);
58      tv2.setPadding(150, 10, 10, 10);
59      tv2.setText("Log Type");
60      header.addView(tv2);
61
62      table.addView(header);
63      mainLayout.addView(table);
64      CountdownTimer t = new CountdownTimer( Long.MAX_VALUE , 1000) {
65
66          // This is called every interval. (Every 10 seconds in this example)
67          public void onTick(long millisUntilFinished) { getData(); }
68
69          public void onFinish() {
70              Log.d("test", "Timer last tick");
71              start();
72          }
73      }.start();
74
75      private void getData() {
76          // RequestQueue initialized
77          mRequestQueue = Volley.newRequestQueue(this);
78
79          // String Request initialized
80          mStringRequest = new StringRequest(Request.Method.GET, url, new Response.Listener<String>() {
81              @Override
82              public void onResponse(String response) {
83                  JSONObject jsonObject = null;
84                  try {
85                      jsonObject = new JSONObject(response);
86                      String time = jsonObject.optString("time");
87                      String type = jsonObject.optString("type");
88                  } catch (JSONException e) {
89                      e.printStackTrace();
90                  }
91              }
92          });
93          mRequestQueue.add(mStringRequest);
94      }
```



```
57      TextView tv2 = new TextView(MainActivity.this);
58      tv2.setPadding(150, 10, 10, 10);
59      tv2.setText("Log Type");
60      header.addView(tv2);
61
62      table.addView(header);
63      mainLayout.addView(table);
64      CountdownTimer t = new CountdownTimer( Long.MAX_VALUE , 1000) {
65
66          // This is called every interval. (Every 10 seconds in this example)
67          public void onTick(long millisUntilFinished) { getData(); }
68
69          public void onFinish() {
70              Log.d("test", "Timer last tick");
71              start();
72          }
73      }.start();
74
75      private void getData() {
76          // RequestQueue initialized
77          mRequestQueue = Volley.newRequestQueue(this);
78
79          // String Request initialized
80          mStringRequest = new StringRequest(Request.Method.GET, url, new Response.Listener<String>() {
81              @Override
82              public void onResponse(String response) {
83                  JSONObject jsonObject = null;
84                  try {
85                      jsonObject = new JSONObject(response);
86                      String time = jsonObject.optString("time");
87                      String type = jsonObject.optString("type");
88                  } catch (JSONException e) {
89                      e.printStackTrace();
90                  }
91              }
92          });
93          mRequestQueue.add(mStringRequest);
94      }
```

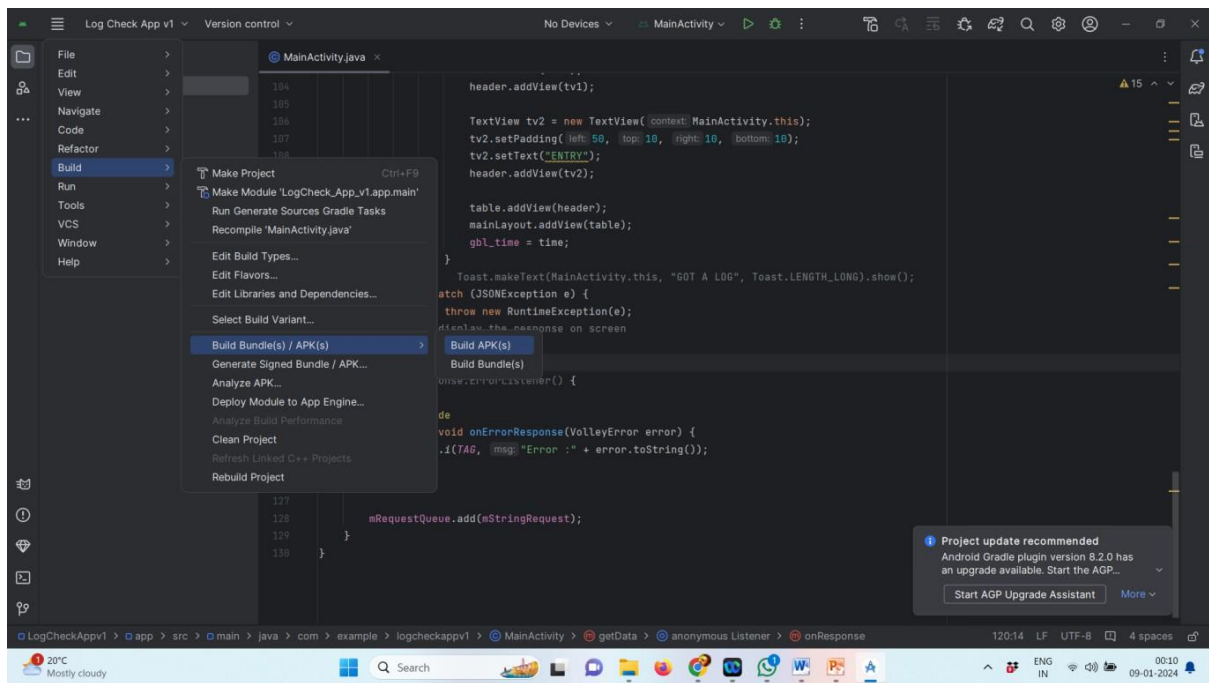
```

90  JSONObject jsonObject = null;
91  try {
92      jsonObject = new JSONObject(response);
93      String time = jsonObject.optString("time");
94      String type = jsonObject.optString("type");
95      if (!Objects.equals(gbl_time, time)) {
96          LinearLayout mainLayout = findViewById(R.id.mainLayout);
97          TableLayout table = new TableLayout(MainActivity.this);
98
99          TableRow header = new TableRow(MainActivity.this);
100          TextView tv = new TextView(MainActivity.this);
101          tv.setPadding(70, 10, 70, 10);
102          tv.setText(type);
103          header.addView(tv);
104
105          TextView tv1 = new TextView(MainActivity.this);
106          tv1.setPadding(50, 10, 50, 10);
107          tv1.setText(time);
108          header.addView(tv1);
109
110          TextView tv2 = new TextView(MainActivity.this);
111          tv2.setPadding(50, 10, 10, 10);
112          tv2.setText("ENTRY");
113          header.addView(tv2);
114
115          table.addView(header);
116          mainLayout.addView(table);
117          gbl_time = time;
118      }
119      // Toast.makeText(MainActivity.this, "GOT A LOG", Toast.LENGTH_LONG).show();
120      } catch (JSONException e) {
121          throw new RuntimeException(e);
122      }
123  }
124  }
125  }
126  }
127  }
128  }
129  }
130  }
131  }
132  }
133  }
134  }
135  }
136  }
137  }
138  }
139  }
140  }
141  }
142  }
143  }
144  }
145  }
146  }
147  }
148  }
149  }
150  }
151  }
152  }
153  }
154  }
155  }
156  }
157  }
158  }
159  }
160  }
161  }
162  }
163  }
164  }
165  }
166  }
167  }
168  }
169  }
170  }
171  }
172  }
173  }
174  }
175  }
176  }
177  }
178  }
179  }
180  }
181  }
182  }
183  }
184  }
185  }
186  }
187  }
188  }
189  }
190  }
191  }
192  }
193  }
194  }
195  }
196  }
197  }
198  }
199  }
200  }
201  }
202  }
203  }
204  }
205  }
206  }
207  }
208  }
209  }
210  }
211  }
212  }
213  }
214  }
215  }
216  }
217  }
218  }
219  }
220  }
221  }
222  }
223  }
224  }
225  }
226  }
227  }
228  }
229  }
230  }
231  }
232  }
233  }
234  }
235  }
236  }
237  }
238  }
239  }
240  }
241  }
242  }
243  }
244  }
245  }
246  }
247  }
248  }
249  }
250  }
251  }
252  }
253  }
254  }
255  }
256  }
257  }
258  }
259  }
260  }
261  }
262  }
263  }
264  }
265  }
266  }
267  }
268  }
269  }
270  }
271  }
272  }
273  }
274  }
275  }
276  }
277  }
278  }
279  }
280  }
281  }
282  }
283  }
284  }
285  }
286  }
287  }
288  }
289  }
290  }
291  }
292  }
293  }
294  }
295  }
296  }
297  }
298  }
299  }
300  }
301  }
302  }
303  }
304  }
305  }
306  }
307  }
308  }
309  }
310  }
311  }
312  }
313  }
314  }
315  }
316  }
317  }
318  }
319  }
320  }
321  }
322  }
323  }
324  }
325  }
326  }
327  }
328  }
329  }
330  }
331  }
332  }
333  }
334  }
335  }
336  }
337  }
338  }
339  }
340  }
341  }
342  }
343  }
344  }
345  }
346  }
347  }
348  }
349  }
350  }
351  }
352  }
353  }
354  }
355  }
356  }
357  }
358  }
359  }
360  }
361  }
362  }
363  }
364  }
365  }
366  }
367  }
368  }
369  }
370  }
371  }
372  }
373  }
374  }
375  }
376  }
377  }
378  }
379  }
380  }
381  }
382  }
383  }
384  }
385  }
386  }
387  }
388  }
389  }
390  }
391  }
392  }
393  }
394  }
395  }
396  }
397  }
398  }
399  }
400  }
401  }
402  }
403  }
404  }
405  }
406  }
407  }
408  }
409  }
410  }
411  }
412  }
413  }
414  }
415  }
416  }
417  }
418  }
419  }
420  }
421  }
422  }
423  }
424  }
425  }
426  }
427  }
428  }
429  }
430  }
431  }
432  }
433  }
434  }
435  }
436  }
437  }
438  }
439  }
440  }
441  }
442  }
443  }
444  }
445  }
446  }
447  }
448  }
449  }
450  }
451  }
452  }
453  }
454  }
455  }
456  }
457  }
458  }
459  }
460  }
461  }
462  }
463  }
464  }
465  }
466  }
467  }
468  }
469  }
470  }
471  }
472  }
473  }
474  }
475  }
476  }
477  }
478  }
479  }
480  }
481  }
482  }
483  }
484  }
485  }
486  }
487  }
488  }
489  }
490  }
491  }
492  }
493  }
494  }
495  }
496  }
497  }
498  }
499  }
500  }
501  }
502  }
503  }
504  }
505  }
506  }
507  }
508  }
509  }
510  }
511  }
512  }
513  }
514  }
515  }
516  }
517  }
518  }
519  }
520  }
521  }
522  }
523  }
524  }
525  }
526  }
527  }
528  }
529  }
530  }
531  }
532  }
533  }
534  }
535  }
536  }
537  }
538  }
539  }
540  }
541  }
542  }
543  }
544  }
545  }
546  }
547  }
548  }
549  }
550  }
551  }
552  }
553  }
554  }
555  }
556  }
557  }
558  }
559  }
560  }
561  }
562  }
563  }
564  }
565  }
566  }
567  }
568  }
569  }
570  }
571  }
572  }
573  }
574  }
575  }
576  }
577  }
578  }
579  }
580  }
581  }
582  }
583  }
584  }
585  }
586  }
587  }
588  }
589  }
590  }
591  }
592  }
593  }
594  }
595  }
596  }
597  }
598  }
599  }
600  }
601  }
602  }
603  }
604  }
605  }
606  }
607  }
608  }
609  }
610  }
611  }
612  }
613  }
614  }
615  }
616  }
617  }
618  }
619  }
620  }
621  }
622  }
623  }
624  }
625  }
626  }
627  }
628  }
629  }
630  }
631  }
632  }
633  }
634  }
635  }
636  }
637  }
638  }
639  }
640  }
641  }
642  }
643  }
644  }
645  }
646  }
647  }
648  }
649  }
650  }
651  }
652  }
653  }
654  }
655  }
656  }
657  }
658  }
659  }
660  }
661  }
662  }
663  }
664  }
665  }
666  }
667  }
668  }
669  }
670  }
671  }
672  }
673  }
674  }
675  }
676  }
677  }
678  }
679  }
680  }
681  }
682  }
683  }
684  }
685  }
686  }
687  }
688  }
689  }
690  }
691  }
692  }
693  }
694  }
695  }
696  }
697  }
698  }
699  }
700  }
701  }
702  }
703  }
704  }
705  }
706  }
707  }
708  }
709  }
710  }
711  }
712  }
713  }
714  }
715  }
716  }
717  }
718  }
719  }
720  }
721  }
722  }
723  }
724  }
725  }
726  }
727  }
728  }
729  }
730  }
731  }
732  }
733  }
734  }
735  }
736  }
737  }
738  }
739  }
740  }
741  }
742  }
743  }
744  }
745  }
746  }
747  }
748  }
749  }
750  }
751  }
752  }
753  }
754  }
755  }
756  }
757  }
758  }
759  }
760  }
761  }
762  }
763  }
764  }
765  }
766  }
767  }
768  }
769  }
770  }
771  }
772  }
773  }
774  }
775  }
776  }
777  }
778  }
779  }
780  }
781  }
782  }
783  }
784  }
785  }
786  }
787  }
788  }
789  }
790  }
791  }
792  }
793  }
794  }
795  }
796  }
797  }
798  }
799  }
800  }
801  }
802  }
803  }
804  }
805  }
806  }
807  }
808  }
809  }
810  }
811  }
812  }
813  }
814  }
815  }
816  }
817  }
818  }
819  }
820  }
821  }
822  }
823  }
824  }
825  }
826  }
827  }
828  }
829  }
830  }
831  }
832  }
833  }
834  }
835  }
836  }
837  }
838  }
839  }
840  }
841  }
842  }
843  }
844  }
845  }
846  }
847  }
848  }
849  }
850  }
851  }
852  }
853  }
854  }
855  }
856  }
857  }
858  }
859  }
860  }
861  }
862  }
863  }
864  }
865  }
866  }
867  }
868  }
869  }
870  }
871  }
872  }
873  }
874  }
875  }
876  }
877  }
878  }
879  }
880  }
881  }
882  }
883  }
884  }
885  }
886  }
887  }
888  }
889  }
890  }
891  }
892  }
893  }
894  }
895  }
896  }
897  }
898  }
899  }
900  }
901  }
902  }
903  }
904  }
905  }
906  }
907  }
908  }
909  }
910  }
911  }
912  }
913  }
914  }
915  }
916  }
917  }
918  }
919  }
920  }
921  }
922  }
923  }
924  }
925  }
926  }
927  }
928  }
929  }
930  }
931  }
932  }
933  }
934  }
935  }
936  }
937  }
938  }
939  }
940  }
941  }
942  }
943  }
944  }
945  }
946  }
947  }
948  }
949  }
950  }
951  }
952  }
953  }
954  }
955  }
956  }
957  }
958  }
959  }
960  }
961  }
962  }
963  }
964  }
965  }
966  }
967  }
968  }
969  }
970  }
971  }
972  }
973  }
974  }
975  }
976  }
977  }
978  }
979  }
980  }
981  }
982  }
983  }
984  }
985  }
986  }
987  }
988  }
989  }
990  }
991  }
992  }
993  }
994  }
995  }
996  }
997  }
998  }
999  }
1000  }

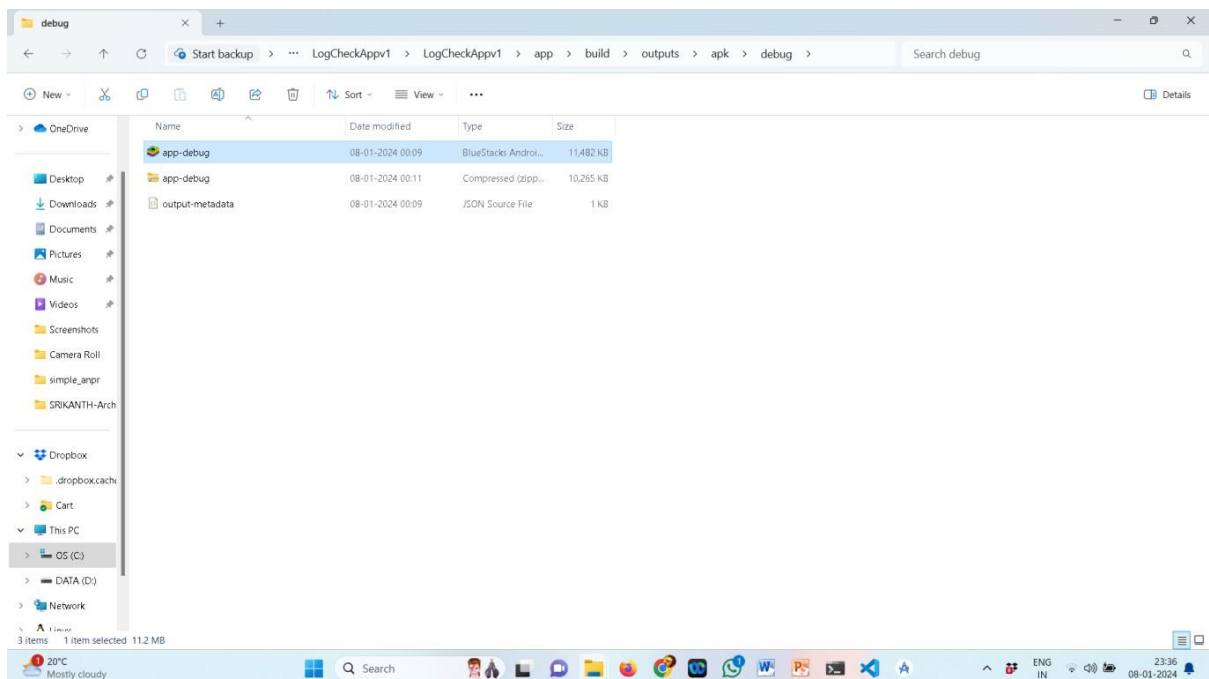
```

```

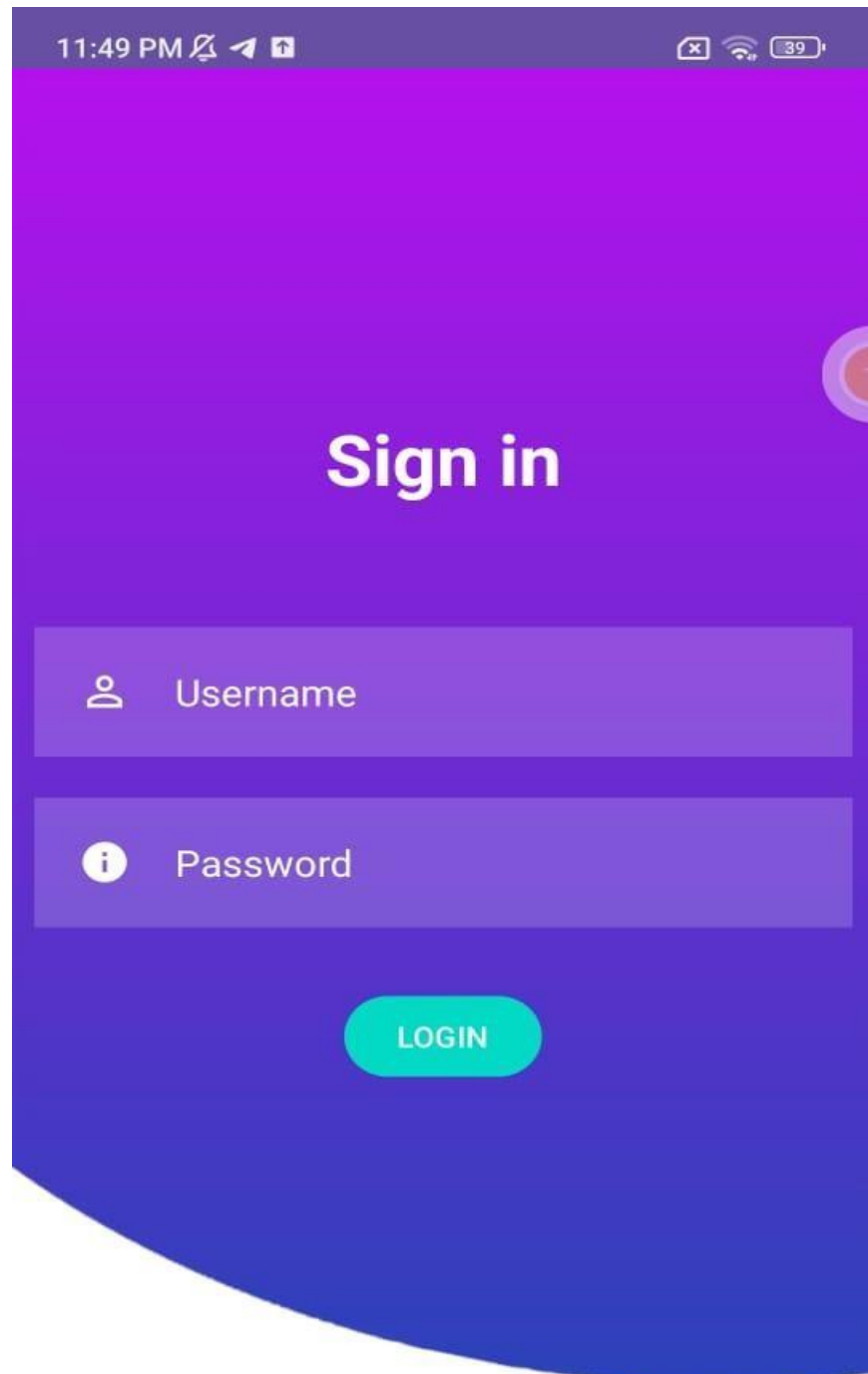
1001  }
1002  }
1003  }
1004  }
1005  }
1006  }
1007  }
1008  }
1009  }
1010  }
1011  }
1012  }
1013  }
1014  }
1015  }
1016  }
1017  }
1018  }
1019  }
1020  }
1021  }
1022  }
1023  }
1024  }
1025  }
1026  }
1027  }
1028  }
1029  }
1030  }
1031  }
1032  }
1033  }
1034  }
1035  }
1036  }
1037  }
1038  }
1039  }
1040  }
1041  }
1042  }
1043  }
1044  }
1045  }
1046  }
1047  }
1048  }
1049  }
1050  }
1051  }
1052  }
1053  }
1054  }
1055  }
1056  }
1057  }
1058  }
1059  }
1060  }
1061  }
1062  }
1063  }
1064  }
1065  }
1066  }
1067  }
1068  }
1069  }
1070  }
1071  }
1072  }
1073  }
1074  }
1075  }
1076  }
1077  }
1078  }
1079  }
1080  }
1081  }
1082  }
1083  }
1084  }
1085  }
1086  }
1087  }
1088  }
1089  }
1090  }
1091  }
1092  }
1093  }
1094  }
1095  }
1096  }
1097  }
1098  }
1099  }
1100  }
1101  }
1102  }
1103  }
1104  }
1105  }
1106  }
1107  }
1108  }
1109  }
1110  }
1111  }
1112  }
1113  }
1114  }
1115  }
1116  }
1117  }
1118  }
1119  }
1120  }
1121  }
1122  }
1123  }
1124  }
1125  }
1126  }
1127  }
1128  }
1129  }
1130  }
1131  }
1132  }
1133  }
1134  }
1135  }
1136  }
1137  }
1138  }
1139  }
1140  }
1141  }
1142  }
1143  }
1144  }
1145  }
1146  }
1147  }
1148  }
1149  }
1150  }
1151  }
1152  }
1153  }
1154  }
1155  }
1156  }
1157  }
1158  }
1159  }
1160  }
1161  }
1162  }
1163  }
1164  }
1165  }
1166  }
1167  }
1168  }
1169  }
1170  }
1171  }
1172  }
1173  }
1174  }
1175  }
1176  }
1177  }
1178  }
1179  }
1180  }
1181  }
1182  }
1183  }
1184  }
1185  }
1186  }
1187  }
1188  }
1189  }
1190  }
1191  }
1192  }
1193  }
1194  }
1195  }
1196  }
1197  }
1198  }
1199  }
1200  }
1201  }
1202  }
1203  }
1204  }
1205  }
1206  }
1207  }
1208  }
1209  }
1210  }
1211  }
1212  }
1213  }
1214  }
1215  }
1216  }
1217  }
1218  }
1219  }
1220  }
1221  }
1222  }
1223  }
1224  }
1225  }
1226  }
1227  }
1228  }
1229  }
1230  }
1231  }
1232  }
1233  }
1234  }
1235  }
1236  }
1237  }
1238  }
1239  }
1240  }
1241  }
1242  }
1243  }
1244  }
1245  }
1246  }
1247  }
1248  }
1249  }
1250  }
1251  }
1252  }
1253  }
1254  }
1255  }
1256  }
1257  }
1258  }
1259  }
1260  }
1261  }
1262  }
1263  }
1264  }
1265  }
1266  }
1267  }
1268  }
1269  }
1270  }
1271  }
1272  }
1273  }
1274  }
1275  }
1276  }
1277  }
1278  }
1279  }
1280  }
1281  }
1282  }
1283  }
1284  }
1285  }
1286  }
1287  }
1288  }
1289  }
1290  }
1291  }
1292  }
1293  }
1294  }
1295  }
1296  }
1297  }
1298  }
1299  }
1300  }
1301  }
1302  }
1303  }
1304  }
1305  }
1306  }
1307  }
1308  }
1309  }
1310  }
1311  }
1312  }
1313  }
1314  }
1315  }
1316  }
1317  }
1318  }
1319  }
1320  }
1321  }
1322  }
1323  }
1324  }
1325  }
1326  }
1327  }
1328  }
1329  }
1330  }
1331  }
1332  }
1333  }
1334  }
1335  }
1336  }
1337  }
1338  }
1339  }
1340  }
1341  }
1342  }
1343  }
1344  }
1345  }
1346  }
1347  }
1348  }
1349  }
1350  }
1351  }
1352  }
1353  }
1354  }
1355  }
1356  }
1357  }
1358  }
1359  }
1360  }
1361  }
1362  }
1363  }
1364  }
1365  }
1366  }
1367  }
1368  }
1369  }
1370  }
1371  }
1372  }
1373  }
1374  }
1375  }
1376  }
1377  }
1378  }
1379  }
1380  }
1381  }
1382  }
1383  }
1384  }
1385  }
1386  }
1387  }
1388  }
1389  }
1390  }
1391  }
1392  }
1393  }
1394  }
1395  }
1396  }
1397  }
1398  }
1399  }
1400  }
1401  }
1402  }
1403  }
1404  }
1405  }
1406  }
1407  }
1408  }
1409  }
1410  }
1411  }
1412  }
1413  }
1414  }
1415  }
1416  }
1417  }
1418  }
1419  }
1420  }
1421  }
1422  }
1423  }
1424  }
1425  }
1426  }
1427  }
1428  }
1429  }
1430  }
1431  }
1432  }
1433  }
1434  }
1435  }
1436  }
1437  }
1438  }
1439  }
1440  }
1441  }
1442  }
1443  }
1444  }
1445  }
1446  }
1447  }
1448  }
1449  }
1450  }
1451  }
1452  }
1453  }
1454  }
1455  }
1456  }
1457  }
1458  }
1459  }
1460  }
1461  }
1462  }
1463  }
1464  }
1465  }
1466  }
1467  }
1468  }
1469  }
1470  }
1471  }
1472  }
1473  }
1474  }
1475  }
1476  }
1477  }
1478  }
1479  }
1480  }
1481  }
1482  }
1483  }
1484  }
1485  }
1486  }
1487  }
1488  }
1489  }
1490  }
1491  }
1492  }
1493  }
1494  }
1495  }
1496  }
1497  }
1498  }
1499  }
1500  }
1501  }
1502  }
1503  }
1504  }
1505  }
1506  }
1507  }
1508  }
1509  }
1510  }
1511  }
1512  }
1513  }
1514  }
1515  }
1516  }
1517  }
1518  }
1519  }
1520  }
1521  }
1522  }
1523  }
1524  }
1525  }
1526  }
1527  }
1528  }
1529  }
1530  }
1531  }
1532  }
1533  }
1534  }
1535  }
1536  }
1537  }
1538  }
1539  }
1540  }
1541  }
1542  }
1543  }
1544  }
1545  }
1546  }
1547  }
1548  }
1549  }
1550  }
1551  }
1552  }
1553  }
1554  }
1555  }
1556  }
1557  }
1558  }
1559  }
1560  }
1561  }
1562  }
1563  }
1564  }
1565  }
1566  }
1567  }
1568  }
1569  }
1570  }
1571  }
1572  }
1573  }
1574  }
1575  }
1576  }
1577  }
1578  }
1579  }
1580  }
1581  }
1582  }
1583  }
1584  }
1585  }
1586  }
1587  }
1588  }
1589  }
1590  }
1591  }
1592  }
1593  }
1594  }
1595  }
1596  }
1597  }
1598  }
1599  }
1600  }
1601  }
1602  }
1603  }
1604  }
1605  }
1606  }
1607  }
1608  }
1609  }
1610  }
1611  }
1612  }
1613  }
1614  }
1615  }
1616  }
1617  }
1618  }
1619  }
1620  }
1621  }
1622  }
1623  }
1624  }
1625  }
1626  }
1627  }
1628  }
1629  }
1630  }
1631  }
1632  }
1633  }
1634  }
1635  }
1636  }
1637  }
1638  }
1639  }
1640  }
1641  }
1642  }
1643  }
1644  }
1645  }
1646  }
1647  }
1648  }
1649  }
1650  }
1651  }
1652  }
1653  }
1654  }
1655  }
1656  }
1657  }
1658  }
1659  }
1660  }
1661  }
1662  }
1663  }
1664  }
1665  }
1666  }
1667  }
1668  }
1669  }
1670  }
1671  }
1672  }
1673  }
1674  }
1675  }
1676  }
1677  }
1678  }
1679  }
1680  }
1681  }
1682  }
1683  }
1684  }
1685  }
1686  }
1687  }
1688  }
1689  }
1690  }
1691  }
1692  }
1693  }
1694  }
1695  }
1696  }
1697  }
1698  }
1699  }
1700  }
1701  }
1702  }
1703  }
1704  }
1705  }
1706  }
1707  }
1708  }
1709  }
1710  }
1711  }
1712  }
1713  }
1714  }
1715  }
1716  }
1717  }
1718  }
1719  }
1720  }
1721  }
1722  }
1723  }
1724  }
1725  }
1726  }
1727  }
1728  }
1729  }
1730  }
1731  }
1732  }
1733  }
1734  }
1735  }
1736  }
1737  }
1738  }
1739  }
1740  }
1741  }
1742  }
1743  }
1744  }
1745  }
1746  }
1747  }
1748  }
1749  }
1750  }
1751  }
1752  }
1753  }
1754  }
1755  }
1756  }
1757  }
1758  }
1759  }
1760  }
1761  }
1762  }
1763  }
1764  }
1765  }
1766  }
1767  }
1768  }
1769  }
1770  }
1771  }
1772  }
1773  }
1774  }
1775  }
1776  }
1777  }
1778  }
1779  }
1780  }
1781  }
1782  }
1783  }
1784  }
1785  }
1786  }
1787  }
1788  }
1789  }
1790  }
1791  }
1792  }
1793  }
1794  }
1795  }
1796  }
1797  }
1798  }
1799  }
1800  }
1801  }
1802  }
1803  }
1804  }
1805  }
1806  }
1807  }
1808  }
1809  }
1810  }
1811  }
1812  }
1813  }
1814  }
1815  }
1816  }
1817  }
1818  }
1819  }
1820  }
1821  }
1822  }
1823  }
1824  }
1825  }
1826  }
1827  }
1828  }
1829  }
1830  }
1831  }
1832  }
1833  }
1834  }
1835  }
1836  }
1837  }
1838  }
1839  }
1840  }
1841  }
1842  }
1843  }
1844  }
1845  }
1846  }
1847  }
1848  }
1849  }
1850  }
1851  }
1852  }
1853  }
1854  }
1855  }
1856  }
1857  }
1858  }
1859  }
1860  }
1861  }
1862  }
1863  }
1864  }
1865  }
1866  }
1867  }
1868  }
1869  }
1870  }
1871  }
1872  }
1873  }
1874  }
1875  }
1876  }
1877  }
1878  }
1879  }
1880  }
1881  }
1882  }
1883  }
1884  }
1885  }
1886  }
1887  }
1888  }
1889  }
1890  }
1891  }
1892  }
1893  }
1894  }
1895  }
1896  }
1897  }
1898  }
1899  }
1900  }
1901  }
1902  }
1903  }
1904  }
1905  }
1906  }
1907  }
1908  }
1909  }
1910  }
1911  }
1912  }
1913  }
1914  }
1915  }
1916  }
1917  }
1918  }
1919  }
192
```



In above screen after all the coding part we should open the code file in “ANDROID STUDIO” which is a software for mobile application development after opening the file in this software then click on top-left corner and select the build option and select the build apk option and click on the “Build APK” ,it automatically converts the code into apk file and locates it.

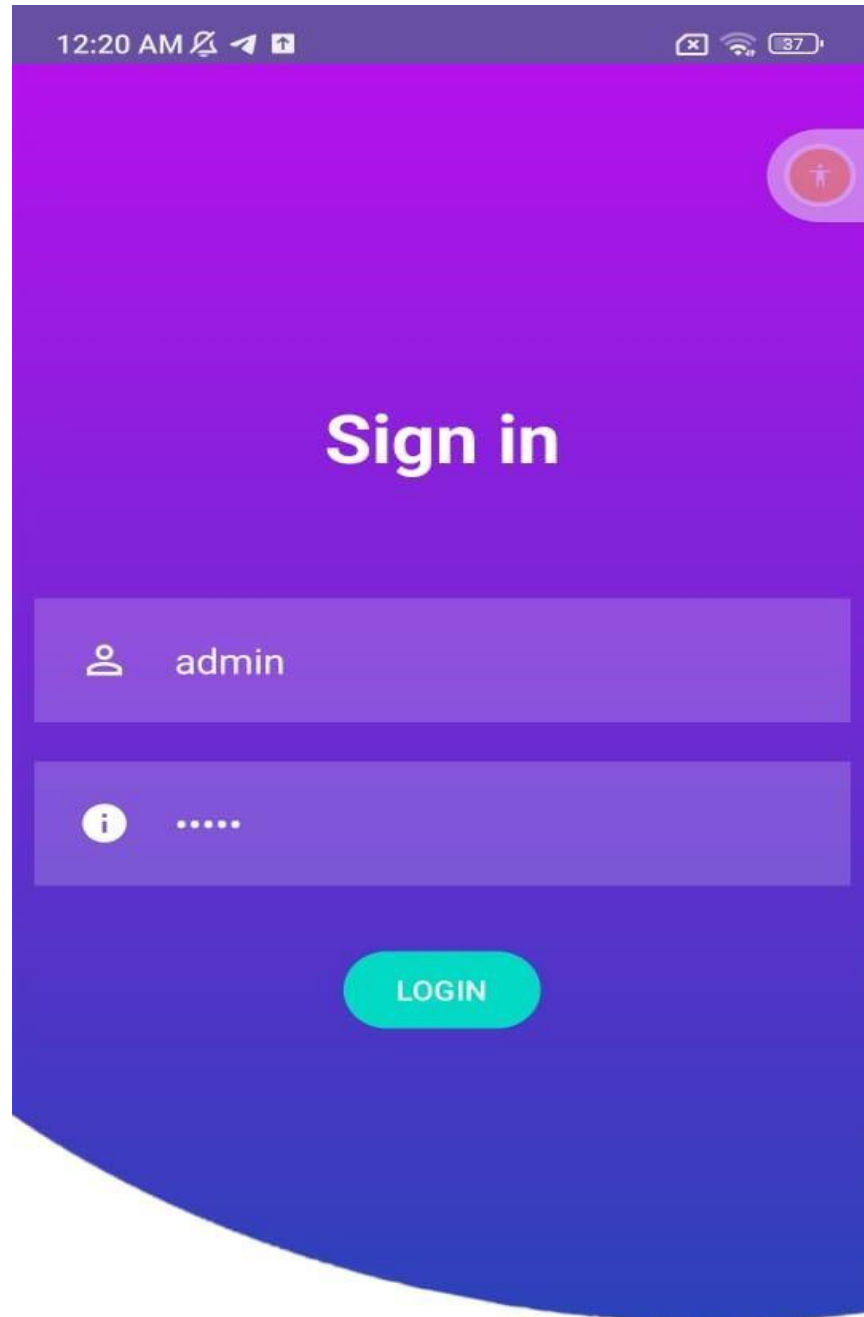


In above screen we can see the app\_debug after locating the apk file share it on the android phone and install it on the phone.

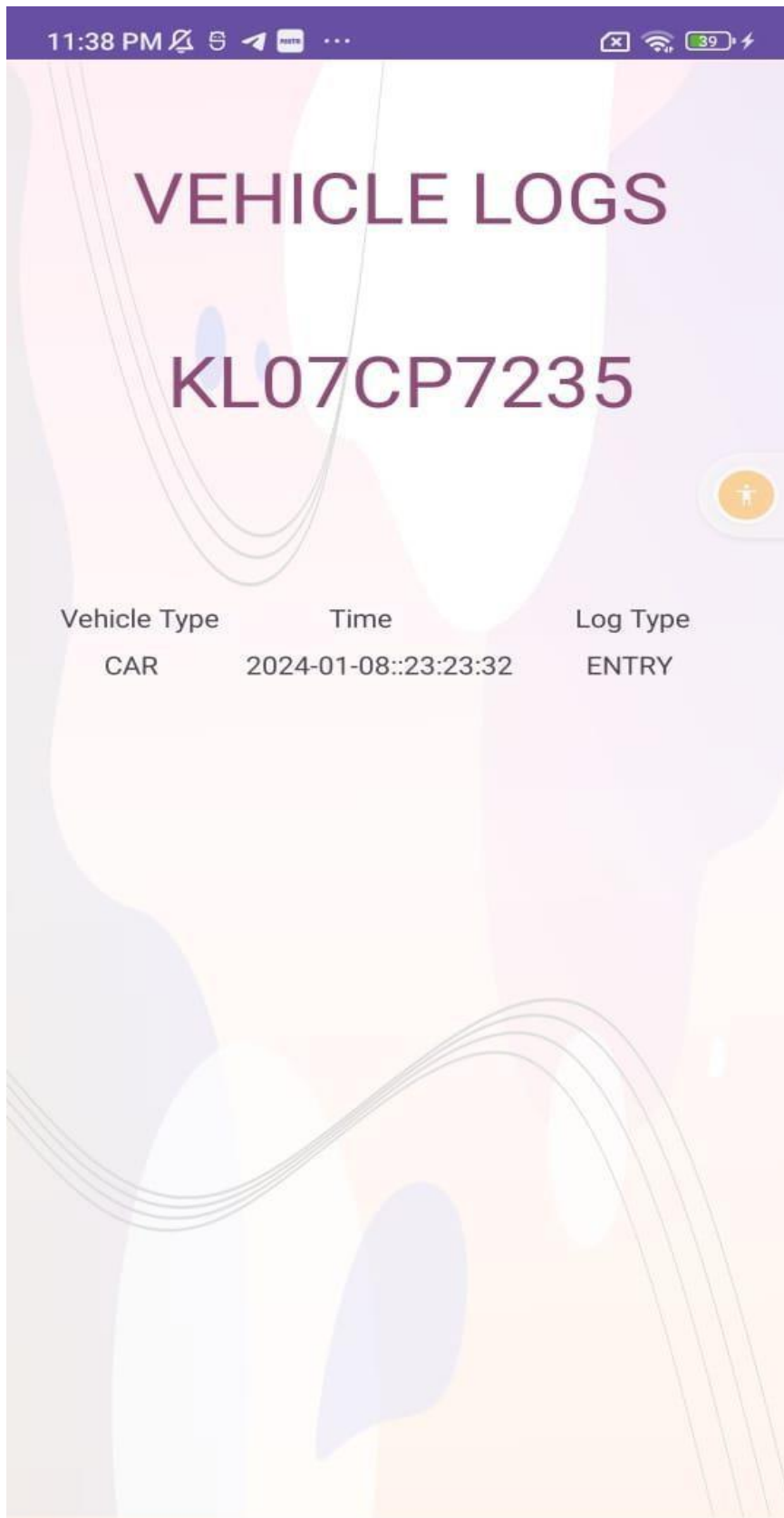




In above screen after login into the apk application it will ask the login credentials.

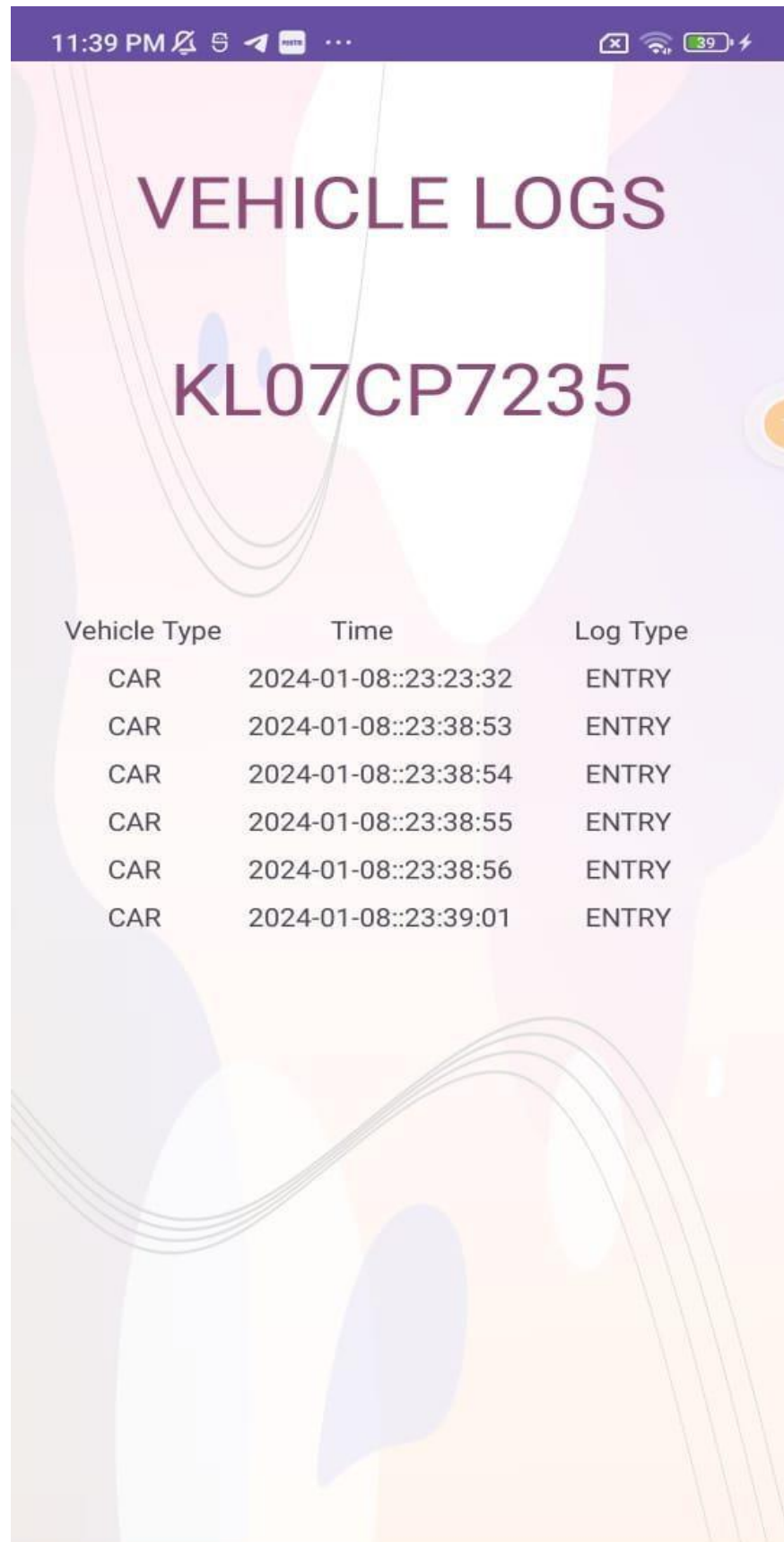


In above screen after entering credentials it will check and it will login only if we enter correct credentials other wise it will show login failed. Here i keep login\_id :admin and password: admin. And click on the submit button, it will enter inside the application.

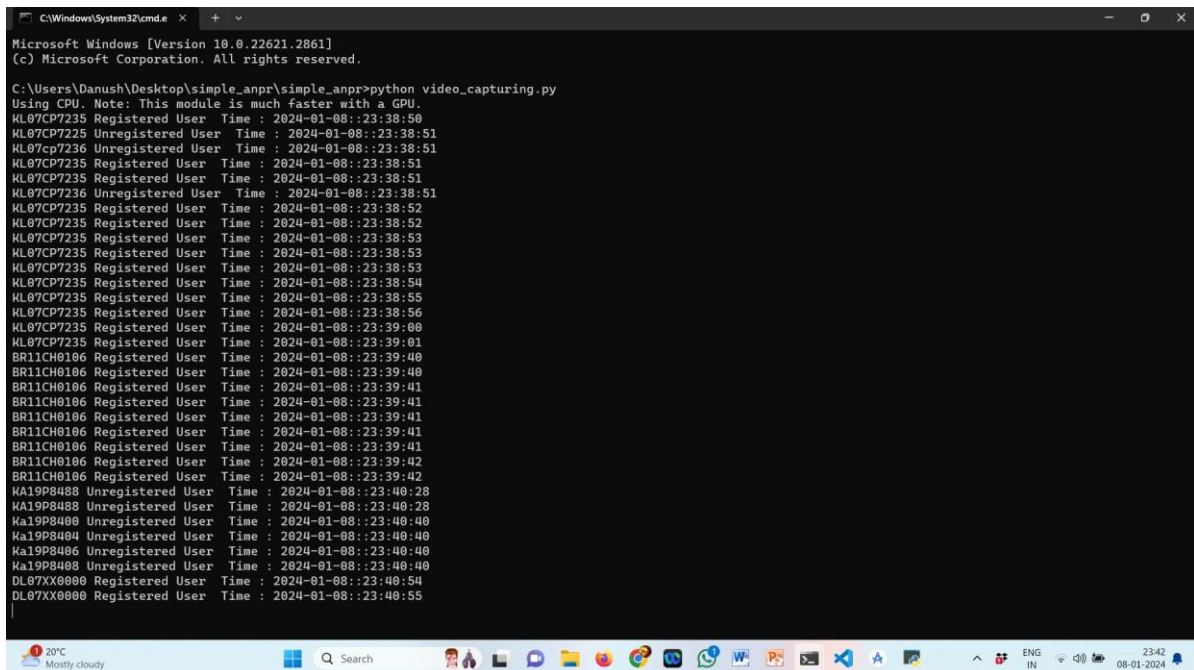




In above screen we can see the 3 things of category they are vehicle type ,time ,log\_type.  
Here in the above first log we consider dummy log of random values for the vehicle number.



In above screen will get log\_capturing with including time stamp and log\_type it will take if the camera captures 4 to 5 time stamps in one second the whole 4,5 time stamps considered as a single log and it will reflect only registered number plates of the vehicle.



```

C:\Users\Danush\Desktop\simple_anpr>python video_capturing.py
Using CPU. Note: This module is much faster with a GPU.
KL07CP7235 Registered User Time : 2024-01-08:23:38:50
KL07CP7225 Unregistered User Time : 2024-01-08:23:38:51
KL07CP7236 Unregistered User Time : 2024-01-08:23:38:51
KL07CP7235 Registered User Time : 2024-01-08:23:38:51
KL07CP7235 Registered User Time : 2024-01-08:23:38:51
KL07CP7236 Unregistered User Time : 2024-01-08:23:38:51
KL07CP7235 Registered User Time : 2024-01-08:23:38:52
KL07CP7235 Registered User Time : 2024-01-08:23:38:52
KL07CP7235 Registered User Time : 2024-01-08:23:38:53
KL07CP7235 Registered User Time : 2024-01-08:23:38:53
KL07CP7235 Registered User Time : 2024-01-08:23:38:53
KL07CP7235 Registered User Time : 2024-01-08:23:38:54
KL07CP7235 Registered User Time : 2024-01-08:23:38:55
KL07CP7235 Registered User Time : 2024-01-08:23:38:56
KL07CP7235 Registered User Time : 2024-01-08:23:39:00
KL07CP7235 Registered User Time : 2024-01-08:23:39:01
BR11CH0106 Registered User Time : 2024-01-08:23:39:40
BR11CH0106 Registered User Time : 2024-01-08:23:39:40
BR11CH0106 Registered User Time : 2024-01-08:23:39:41
BR11CH0106 Registered User Time : 2024-01-08:23:39:41
BR11CH0106 Registered User Time : 2024-01-08:23:39:41
BR11CH0106 Registered User Time : 2024-01-08:23:39:41
BR11CH0106 Registered User Time : 2024-01-08:23:39:41
BR11CH0106 Registered User Time : 2024-01-08:23:39:42
BR11CH0106 Registered User Time : 2024-01-08:23:39:42
KA19P8488 Unregistered User Time : 2024-01-08:23:40:28
KA19P8488 Unregistered User Time : 2024-01-08:23:40:28
KA19P8488 Unregistered User Time : 2024-01-08:23:40:40
KA19P8488 Unregistered User Time : 2024-01-08:23:40:40
KA19P8488 Unregistered User Time : 2024-01-08:23:40:40
KA19P8488 Unregistered User Time : 2024-01-08:23:40:40
DL07XX0800 Registered User Time : 2024-01-08:23:40:54
DL07XX0800 Registered User Time : 2024-01-08:23:40:55

```

In above screen we can see both unregistered and registered number plates with the time stamps of each second this backend we can consider as another side of the vehicle detection whether the vehicle is registered or unregistered .

Technologies Used to build this applications are:

- 1] Application programming interface:- Flask python.
- 2] Opencv and pillow :-ANPR(Automatic Number Plate Recognizer System)
- 3]Backend system:- Python to check whether the number is registered or not.
- 4]Hardwares:- Camera,Computers(As backend system).

## 6.5 ABOUT AUTOMATIC LICENSE PLATE ALGORITHM:

Automatic Number Plate Recognition (ANPR) is a system that reads number plates from photos by using optical character recognition. Law enforcement, traffic control, toll collecting, and parking management are just a few of the many uses for ANPR systems. For the purpose of precisely extracting and interpreting licence plate information from pictures or video streams, the ANPR algorithm consists of multiple crucial processes.

### 1. Purpose:

The automatic extraction, interpretation, and recognition of number plate information from pictures or video streams is the main goal of the Automatic Number Plate Recognition

(ANPR) algorithm.

**Security and Surveillance:** By giving authorities a tool to track and identify vehicles coming into or going out of particular zones, ANPR improves security and surveillance. It is useful for protecting vital sites including border crossings, airports, and government buildings.

## **2. Key Features:**

A number of characteristics of image processing, character recognition, and system functionality are included in the major features of an Automatic Number Plate Recognition (ANPR) algorithm. Together, these characteristics help ANPR systems be more accurate, effective, and versatile. The salient characteristics are as follows:

Accurately identifying and positioning the licence plate within a picture or video frame is known as "licence plate localization." This entails identifying the precise area inside the picture where the licence plate is situated.

## **3. Development Workflow:**

A number of procedures are involved in the creation of an Automatic Number Plate Recognition (ANPR) workflow, all of which help to accurately identify and interpret number plates. An extensive methodology for creating an ANPR algorithm is shown below:

1. **Image Acquisition:** The first step in the procedure is acquiring pictures or video frames that show cars and their licence plates. Surveillance, traffic, or specialised ANPR cameras can all be used to take these pictures.
2. **Preprocessing:** To improve the photos' quality and the algorithm's capacity to recognise licence plates accurately, apply preprocessing techniques. Resizing, normalisation, noise reduction, and contrast modification are a few examples of this.
3. **Licence Plate Localization:** Use algorithms to precisely locate the region of interest (ROI) in the picture that contains the licence plate. For this, methods like colour analysis, edge detection, and morphological processes can be used.
4. **Character Segmentation:** To get ready for optical character recognition (OCR), separate the various characters on the licence plate. For efficient character segmentation, use methods like contour analysis and linked component analysis.
5. **Optical Character Recognition (OCR):** Create and implement OCR algorithms to correctly identify and decipher the licence plate's alphanumeric letters. To improve flexibility to various fonts and styles, use machine learning models for character recognition.

**Key Takeaway:**

The importance of an Automatic Number Plate Recognition (ANPR) algorithm in several applications, the intricacy of its parts, and the possible advantages it presents are important lessons to learn. The main conclusions are as follows:

Versatility of Applications: ANPRS technology is used in many different domains, such as parking systems, toll collecting, traffic control, security, and law enforcement. Because of its adaptability, it is a useful tool for improving security and efficiency in a variety of contexts.

Advanced Neural Plate Recognition (ANPR) algorithms utilise intricate image processing methods, including optical character recognition (OCR), character segmentation, and licence plate localization. To precisely recognise and decipher licence plate information, these procedures need complex algorithms.

The project's commitment to usability and efficacy is demonstrated by important features including real-time monitoring capabilities, intuitive mobile application interfaces, and seamless connection with current infrastructure. The system complies with the strictest data protection regulations, and the emphasis on privacy and security measures guarantees that people's trust is preserved.

## CHAPTER-7

### TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)

#### 7.1 Timeline of Project:

Phase	REVIEW	Timeline
1.Title Finalization with Supervisor.	Review-0	16-10-23
2. Problem Definition/Novelty	Review-1	08-11-23
3. 50% implementation details	Review-2	28-11-23
4. 100% implementation details	Review-3	27-12-23
5. Demonstration	Final viva	10-01-24

#### 7.2 GANTT CHART



## CHAPTER-8

### OUTCOMES

#### 8.1 Expected Outcomes:

- The project "Affordable Mobile Application Camera System for Efficient Residential Society Vehicle Monitoring" has a number of goals, the main one being the impact and effective deployment of the surveillance system in residential communities. The following are the expected results:
- **Improved Security Infrastructure:** It is anticipated that residential communities would have a better security infrastructure as a consequence of the mobile application camera system's successful adoption. Security threats will be discouraged and lessened with the use of real-time monitoring and surveillance capabilities.
- **Effective Vehicle Monitoring:** The project's goal is to set up an effective vehicle monitoring system that will enable security guards and residents to follow and keep an eye on vehicle activity inside the residential community. This will enhance communal well-being and safety in general.
- **Cost-Effective Solution:** Providing a cost-effective solution that complies with the financial limitations of smaller residential societies is one of the main objectives. The project's capacity to deliver a cost-effective yet effective surveillance system will determine its level of success.
- **User-Friendly Mobile Application:** It is anticipated that a mobile application with an interface that is easy to use will have a high rate of user adoption. The application should be easy to use, accessible, and encourage both residents and security staff to actively participate in the monitoring process.
- **Seamless Integration with Existing Infrastructure:** One important result is the seamless integration of residential societies with their current infrastructure. The monitoring system ought to be in harmony with current security structures in order to reduce interference and promote a seamless transition of the new system.

- **Collaboration and Community participation:** By inviting locals to actively participate in the monitoring process, the project hopes to promote community participation. The application should have features that encourage cooperation, communication, and shared security responsibility.
- **Insights from the Carefully Planned Pilot Implementation:** The system's acceptability and practical functionality in a real-world residential environment will be revealed through this well-executed pilot project. The input obtained in this stage will be crucial for system optimisation and improvement.
- **Scalability Across Societies:** One important result of the technology is its scalability across various residential societies. The project's goal is to provide a solution that can be expanded into a variety of community settings and can adjust to differing sizes and requirements.
- **Data Security and Privacy Measures:** Robust data security and privacy measures will be ensured through successful deployment. Residents' expectations regarding privacy will be fulfilled, and the system will abide by applicable laws, building user confidence.

Improved security infrastructure, effective vehicle tracking, cost-effectiveness, community involvement, and scalability across many societies are all anticipated project objectives. The project's main objective is in line with the beneficial effects on lowering security incidents and creating a safer living environment.

## **CHAPTER-9**

### **CONCLUSION**

In Conclusion, the project "Affordable Mobile Application Camera System for Efficient Residential Society Vehicle Monitoring" addresses the unique requirements and financial limitations of smaller communities while making a substantial contribution to strengthening the security framework of residential societies. The goal of the monitoring system's thorough development and execution is to bring in a new era of efficiency, safety, and community involvement in residential settings.

With thorough market research, careful software development, smooth integration procedures, and stringent testing, the project has established the groundwork for a novel and economical solution. The project's methodology demonstrates a dedication to providing a system that not only satisfies technology requirements but is also customised to the distinct features of various residential communities.

An important stage is the meticulously thought-out pilot implementation in a chosen residential community, which offers priceless insights and user feedback. By using an iterative process, the system is improved and optimised based on real-world observations, resulting in a solution that is effective and meets end users' needs and preferences.

The surveillance system has the ability to significantly alter residential societies' security environments and further the development of safe and intelligent communities as it is ready for wider use. By utilising technology, this project shows how innovation, affordability, and the special dynamics of residential living can work together to create a future where safety and community well-being are prioritised.



## REFERENCES

1. Deitel, P., & Deitel, H. (2017). "Android for Programmers: An App-Driven Approach." Pearson.
2. Gonzalez, J. (2016). "Programming Mobile Applications for Android Handheld Systems: Part 1." University of Maryland.
3. Clarke, R. (2018). "Surveillance, Privacy and Security: Citizens' Perspectives." Routledge.
4. Norris, C., McCahill, M., & Wood, D. (2004). "The Growth of CCTV: A Global Perspective on the International Diffusion of Video Surveillance in Publicly Accessible Space." *Surveillance & Society*, 2(2/3), 110-135.
5. Caragliu, A., Del Bo, C., & Nijkamp, P. (2011). "Smart Cities in Europe." *Journal of Urban Technology*, 18(2), 65-82.
6. Horton, I. (2015). "Android Programming for Beginners." Que Publishing.
7. McFarland, D. (2018). "Swift Programming: The Big Nerd Ranch Guide." Big Nerd Ranch.
8. Giffinger, R., Fertner, C., Kramar, H., Kalasek, R., Pichler-Milanović, N., & Meijers, E. (2007). "Smart Cities: Ranking of European Medium-Sized Cities." Centre of Regional Science, Vienna UT.
9. Caragliu, A., Del Bo, C., & Nijkamp, P. (2011). "Smart Cities in Europe." *Journal of Urban Technology*, 18(2), 65-82.
10. Zanella, A., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M. (2014). "Internet of Things for Smart Cities." *IEEE Internet of Things Journal*, 1(1), 22-32.

## APPENDIX-A

### PSUEDOCODE

```

package com.example.logcheckappv1;

import static android.content.ContentValues.TAG;

import androidx.appcompat.app.AppCompatActivity;
import androidx.constraintlayout.widget.ConstraintLayout;

import android.os.Bundle;
import android.os.CountDownTimer;
import android.util.Log;

import android.widget.LinearLayout;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.Objects;

public class MainActivity extends AppCompatActivity {
    private RequestQueue mRequestQueue;
    private StringRequest mStringRequest;
    private String url = "http://192.168.0.154:5000/get_data";

    private String gbl_time = "";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        LinearLayout mainLayout = findViewById(R.id.mainLayout);
        TableLayout table = new TableLayout(this);

        TableRow header = new TableRow(MainActivity.this);
        TextView tv = new TextView(MainActivity.this);
        tv.setPadding(10, 10, 50, 10);
        tv.setText("Vehicle Type");
        header.addView(tv);

        TextView tv1 = new TextView(MainActivity.this);
        tv1.setPadding(100, 10, 100, 10);
        tv1.setText("Time");
        header.addView(tv1);

        TextView tv2 = new TextView(MainActivity.this);

```

```

        tv2.setPadding(150, 10, 10, 10);
        tv2.setText("Log Type");
        header.addView(tv2);

        table.addView(header);
        mainLayout.addView(table);
        CountDownTimer t = new CountDownTimer( Long.MAX_VALUE , 1000) {

            // This is called every interval. (Every 10 seconds in this
example)
            public void onTick(long millisUntilFinished) {
                getData();
            }

            public void onFinish() {
                Log.d("test", "Timer last tick");
                start();
            }
        }.start();
    }

    private void getData() {
        // RequestQueue initialized
        mRequestQueue = Volley.newRequestQueue(this);

        // String Request initialized
        mStringRequest = new StringRequest(Request.Method.GET, url, new
Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                JSONObject jsonObject = null;
                try {
                    jsonObject = new JSONObject(response);
                    String time = jsonObject.optString("time");
                    String type = jsonObject.optString("type");
                    if (!Objects.equals(gbl_time, time)) {
                        LinearLayout mainLayout =
findViewById(R.id.mainLayout);
                        TableLayout table = new
TableLayout(MainActivity.this);

                        TableRow header = new TableRow(MainActivity.this);
                        TextView tv = new TextView(MainActivity.this);
                        tv.setPadding(70, 10, 70, 10);
                        tv.setText(type);
                        header.addView(tv);

                        TextView tv1 = new TextView(MainActivity.this);
                        tv1.setPadding(50, 10, 50, 10);
                        tv1.setText(time);
                        header.addView(tv1);

                        TextView tv2 = new TextView(MainActivity.this);
                        tv2.setPadding(50, 10, 10, 10);
                        tv2.setText("ENTRY");
                        header.addView(tv2);

                        table.addView(header);
                        mainLayout.addView(table);
                        gbl_time = time;
                    }
                }
            }
        });
        Toast.makeText(MainActivity.this, "GOT A LOG",

```

```

Toast.LENGTH_LONG).show();
    } catch (JSONException e) {
        throw new RuntimeException(e);
    } //display the response on screen

    }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Log.i(TAG, "Error :" + error.toString());
        }
    });

    mRequestQueue.add(mStringRequest);
}

```

## PSUEDOCODE FOR VIDEO\_CAPTURE.PY:

```

# importing libraries
import cv2
import main

# Create a VideoCapture object and read from input file
cap = cv2.VideoCapture(0)

# Check if camera opened successfully
if not cap.isOpened():
    print("Error opening video file")
    # Read until video is completed

while cap.isOpened():

    # Capture frame-by-frame
    ret, frame = cap.read()
    if ret:
        frame = main.find_lic_plt(frame)
        # Display the resulting frame
        cv2.imshow('Frame', frame)

        # Press Q on keyboard to exit
        if cv2.waitKey(25) & 0xFF == ord('q'):
            break

    # Break the loop
    else:
        break

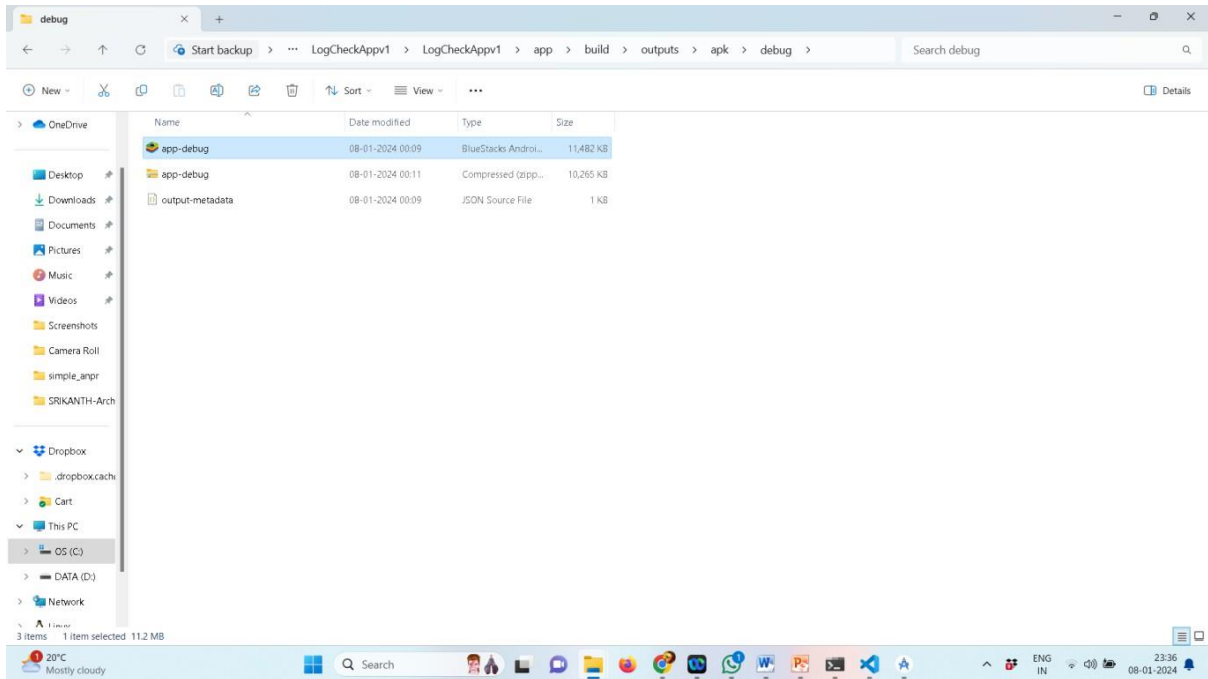
# When everything done, release
# the video capture object

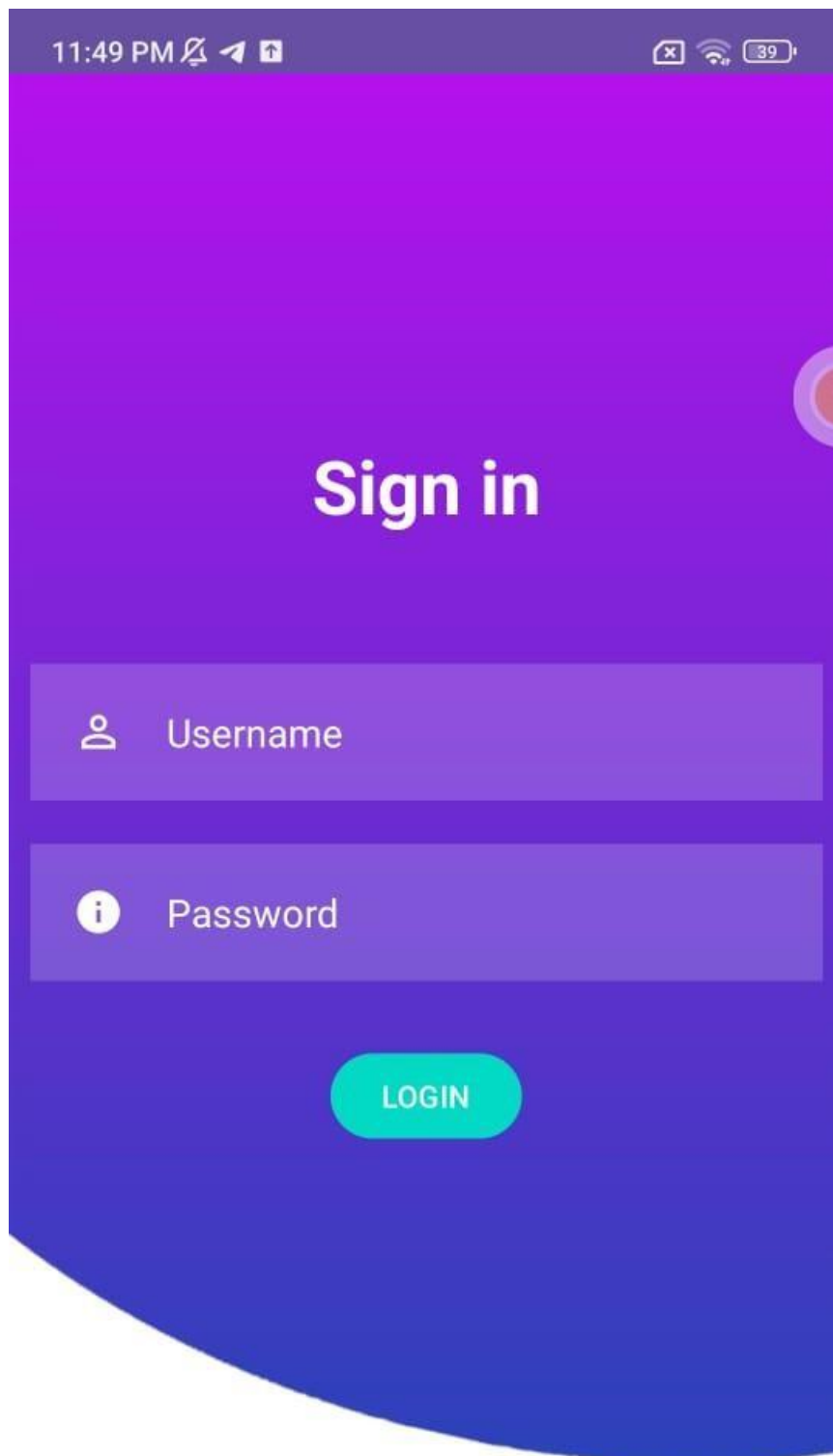
```

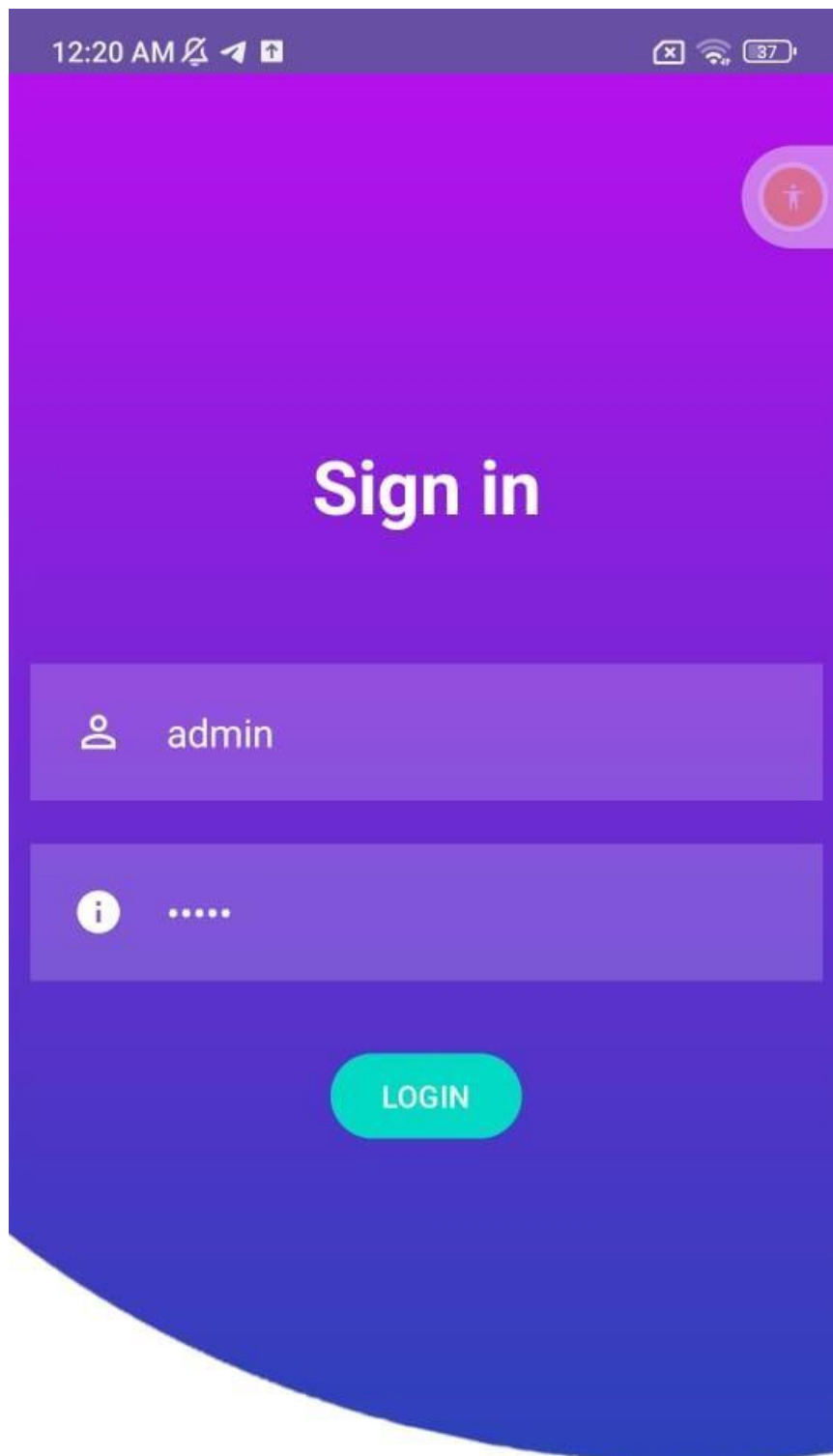
```
cap.release()  
  
# Closes all the frames  
cv2.destroyAllWindows()
```

## APPENDIX-B

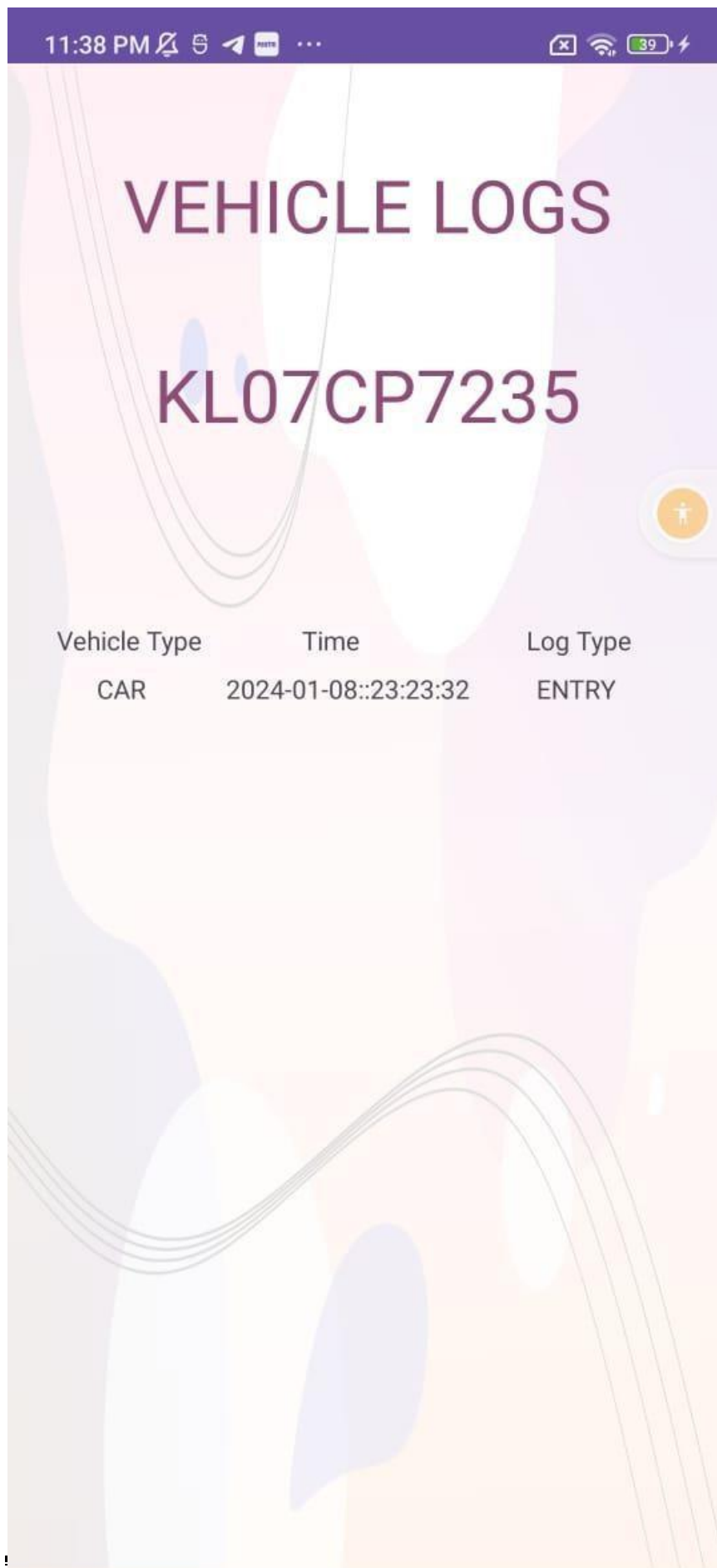
### SCREENSHOTS

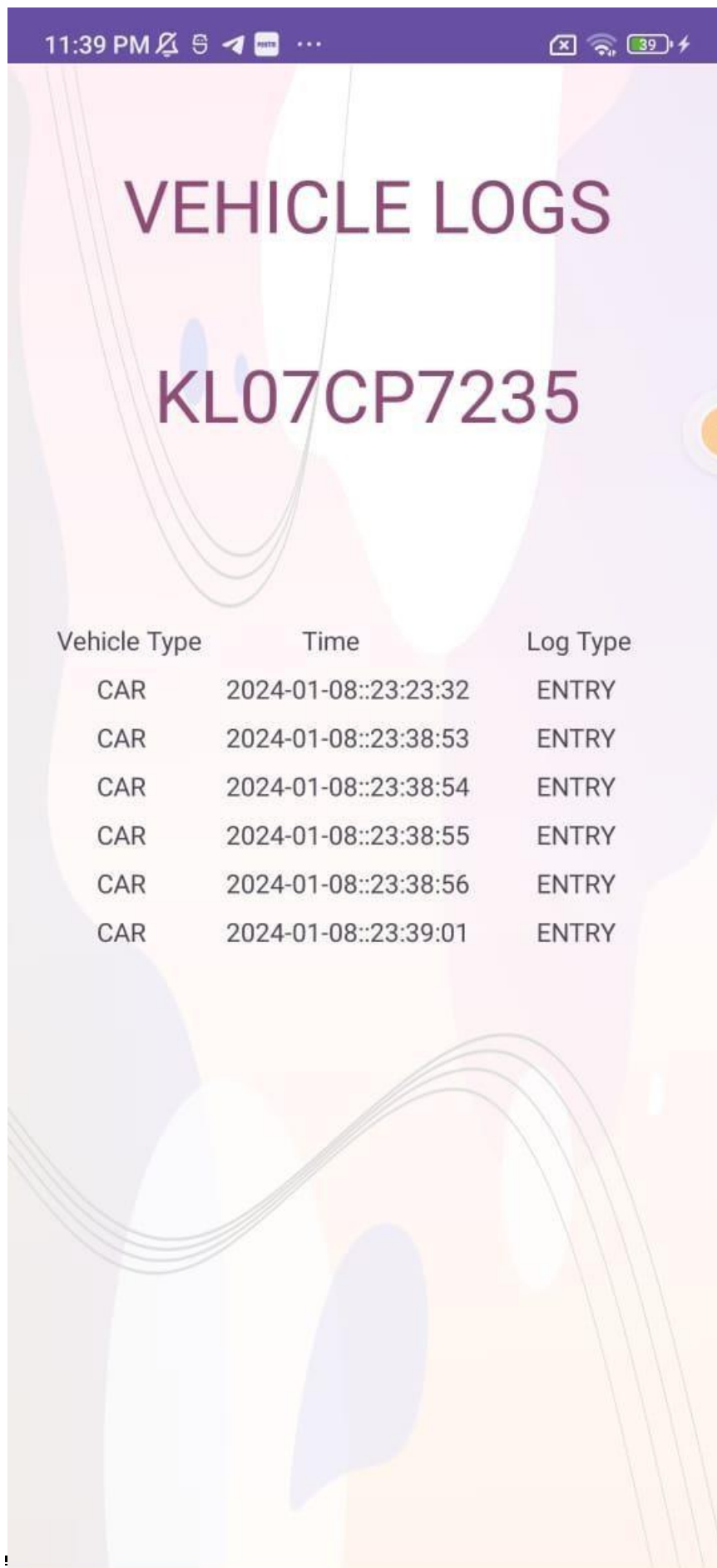


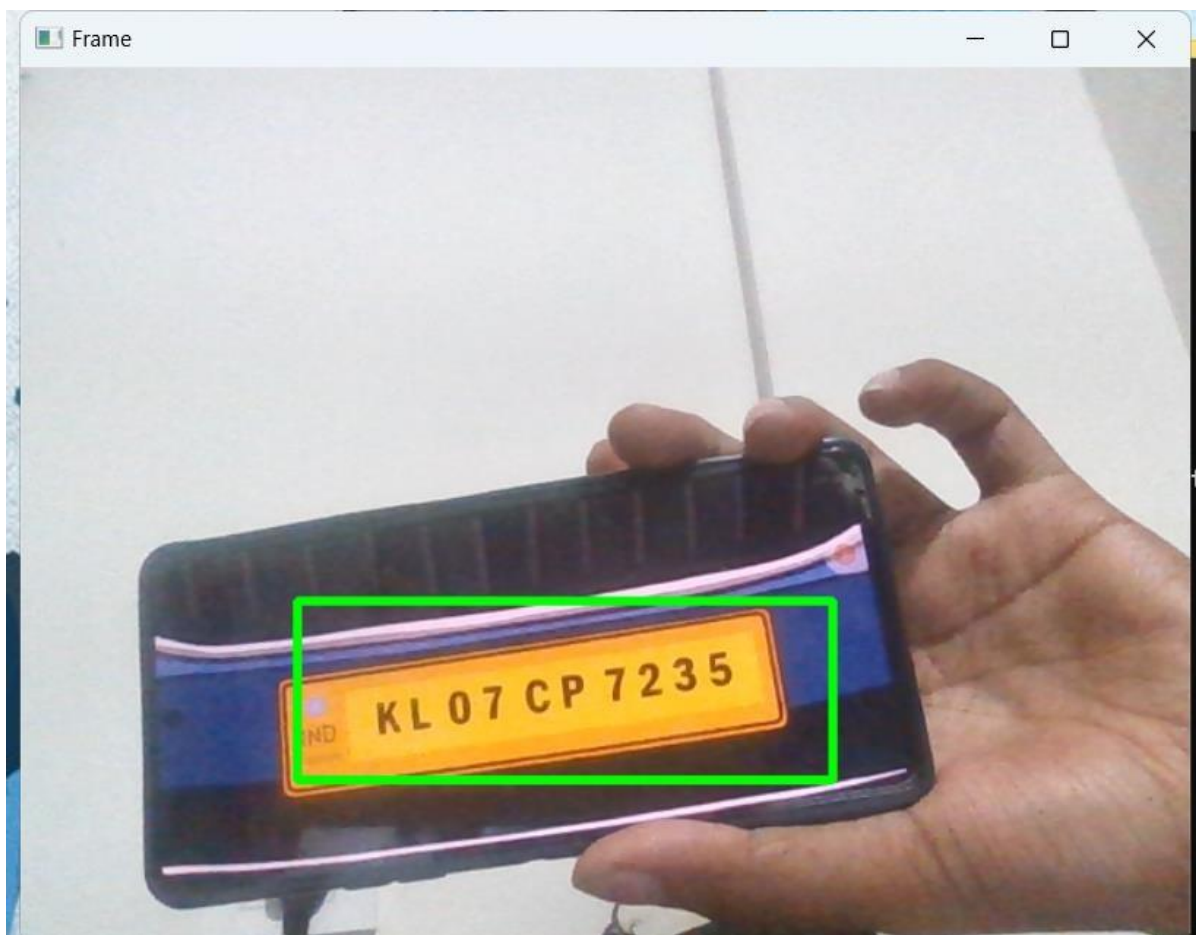
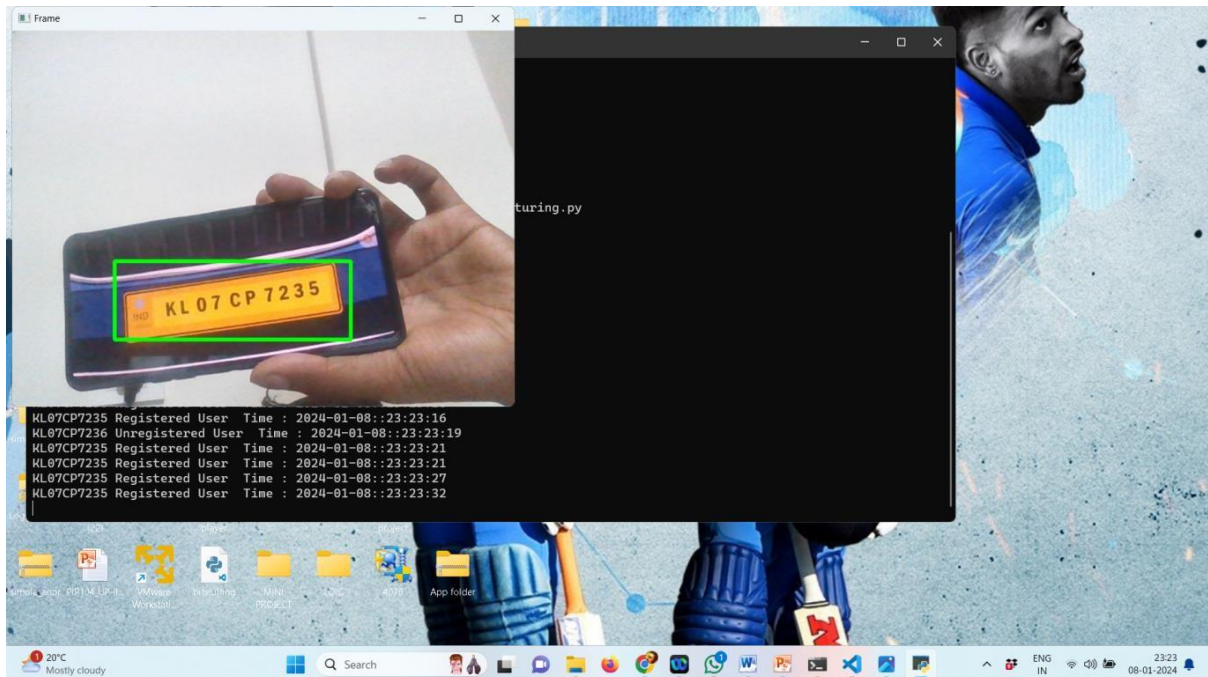












```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

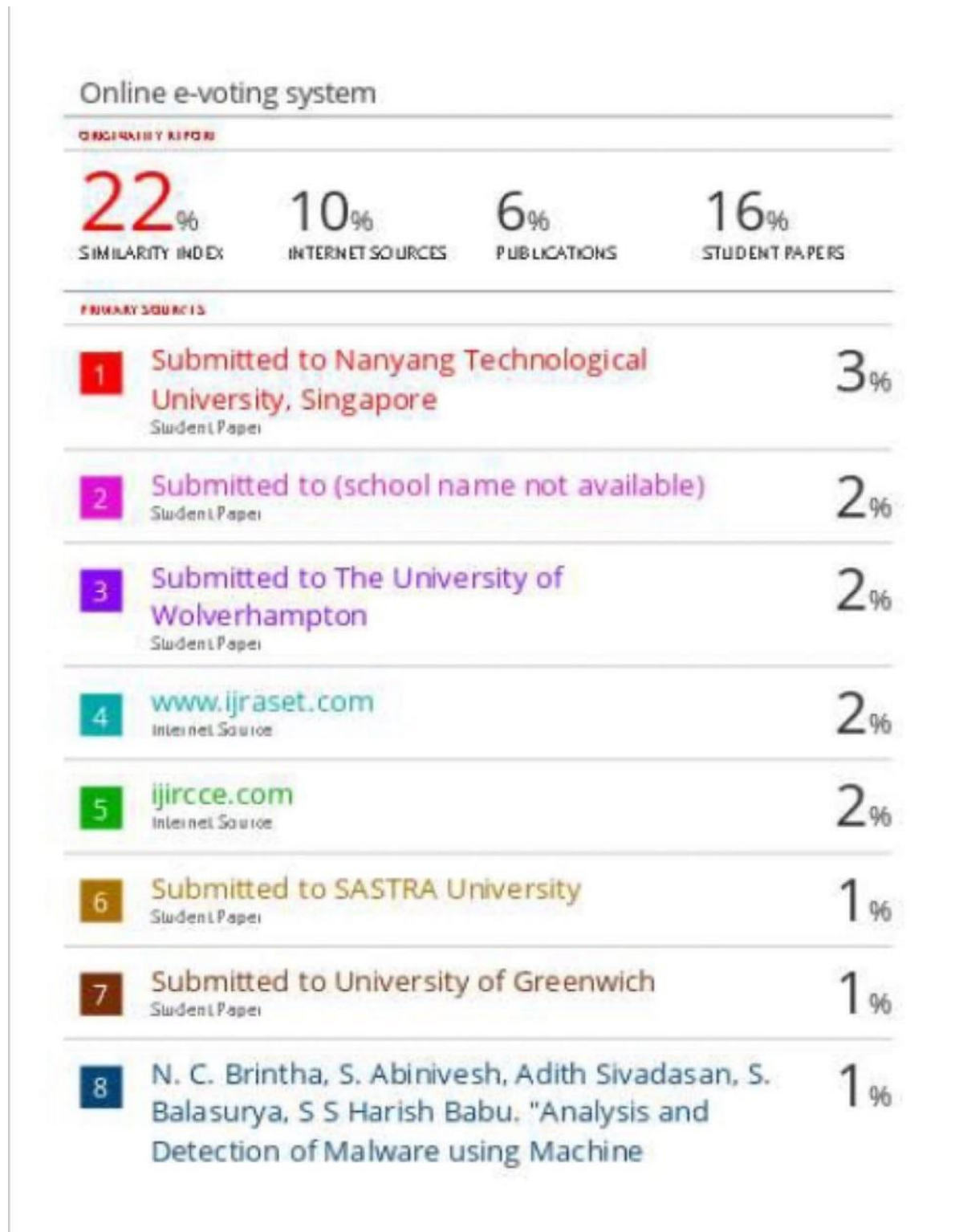
C:\Users\Danush\Desktop\simple_anpr\simple_anpr>python video_capturing.py
Using CPU. Note: This module is much faster with a GPU.
KL07CP7235 Registered User Time : 2024-01-08:23:38:50
KL07CP7235 Unregistered User Time : 2024-01-08:23:38:51
KL07CP7236 Unregistered User Time : 2024-01-08:23:38:51
KL07CP7235 Registered User Time : 2024-01-08:23:38:51
KL07CP7235 Registered User Time : 2024-01-08:23:38:51
KL07CP7236 Unregistered User Time : 2024-01-08:23:38:51
KL07CP7235 Registered User Time : 2024-01-08:23:38:52
KL07CP7235 Registered User Time : 2024-01-08:23:38:52
KL07CP7235 Registered User Time : 2024-01-08:23:38:53
KL07CP7235 Registered User Time : 2024-01-08:23:38:53
KL07CP7235 Registered User Time : 2024-01-08:23:38:53
KL07CP7235 Registered User Time : 2024-01-08:23:38:54
KL07CP7235 Registered User Time : 2024-01-08:23:38:55
KL07CP7235 Registered User Time : 2024-01-08:23:38:56
KL07CP7235 Registered User Time : 2024-01-08:23:39:00
KL07CP7235 Registered User Time : 2024-01-08:23:39:01
BR11CH0106 Registered User Time : 2024-01-08:23:39:40
BR11CH0106 Registered User Time : 2024-01-08:23:39:41
BR11CH0106 Registered User Time : 2024-01-08:23:39:41
BR11CH0106 Registered User Time : 2024-01-08:23:39:41
BR11CH0106 Registered User Time : 2024-01-08:23:39:41
BR11CH0106 Registered User Time : 2024-01-08:23:39:42
BR11CH0106 Registered User Time : 2024-01-08:23:39:42
KA19P8488 Unregistered User Time : 2024-01-08:23:40:28
KA19P8488 Unregistered User Time : 2024-01-08:23:40:28
KA19P8490 Unregistered User Time : 2024-01-08:23:40:40
KA19P8494 Unregistered User Time : 2024-01-08:23:40:40
KA19P8496 Unregistered User Time : 2024-01-08:23:40:40
KA19P8490 Unregistered User Time : 2024-01-08:23:40:40
DL07XX0800 Registered User Time : 2024-01-08:23:40:54
DL07XX0800 Registered User Time : 2024-01-08:23:40:55
```

## **APPENDIX-C**

### **ENCLOSURES**

- 1. Conference Paper Presented Certificates of all students.**
- 2. Include certificate(s) of any Achievement/Award won in any project related event.**
- 3. Similarity Index / Plagiarism Check report clearly showing the Percentage (%). No need of page-wise explanation.**

# Plagiarism Check





# Sustainable Development Goals (SDGs)



**The Project work carried out here is mapped to SDG-11 Sustainable Cities And Communities**

The project work carried here Contributes to creating safer and more sustainable residential communities through effective vehicle monitoring.

Promote inclusive, resilient, and sustainable urban development. Enhance access to basic services, green spaces. Foster community engagement for a harmonious and equitable urban environment.