INTELLIGENT POTHOLE DETECTION SYSTEM

PROJECT REPORT

Submitted by

MANIKANDAN M (DLAVBCA046)

SUCHETH CS (DLAVBCA050)

MANEESH EM (DLAVBCA018)

DIYOOF (DLAVBCA016)

Under the Guidance of

MR. RAMSHEEL M

(HOD, Department of Computer Science)

Submitted in partial fulfilment of the requirements for

the award of the Degree of

BACHELOR OF COMPUTER APPLICATION
UNIVERSITY OF CALICUT

SADANAM KUMARAN COLLEGE, MANKARA, PALAKKAD

MARCH (2024)

SADANAM KUMARAN COLLEGE

MANKARA, PALAKKAD

(Affiliated to The University of Calicut)

Department of Computer Science

<u>BONAFIDE CERTIFICATE</u>

This is to certify that the project work entitled

Intelligent Pothole Detection System

is the bonafide record of work done by

MANIKANDAN M (DLAVBCA046)

SUCHETH CS (DLAVBCA050)

MANEESH EM (DLAVBCA018)

DIYOOF (DLAVBCA016)

for the award for the Degree of BACHELOR OF COMPUTER APPLICATION, University of Calicut submitted for the viva-voce examination held on…………………………………

Project Guide Head of The Department Principal

Internal Examiner External Examiner

ACKNOWLEDGEMENT

We express our sincere gratitude to the almighty for the client and the support blessing showered upon to complete this project successfully. It is an opportunityto thank the institution members for giving me an opportunity for doing the project in the concern.

We highly obliged to Prof. MURUKAN BABU CR, Principal of SADANAM KUMARAN COLLEGE, Mankara, who encourage us throughout our project work.

We express our sincere gratitude to Mr. RAMSHEEL M, Head of Computer Science Department for giving valuable suggestions for the completion of the project work and for his constant guidance valuable suggestions and encouragement throughout the course of our project

We would like to express our thanks to our Project guide Mr. RAMSHEEL M, Head of Computer Science Department, for their cheerful and valuable guidance for the project work that led us to complete the project in a determined way within a stipulated time

MANIKANDAN M (DLAVBCA046)
SUCHETH CS (DLAVBCA050)
MANEESH EM (DLAVBCA018)
DIYOOF (DLAVBCA016)

DECLARATION

We hereby declare that this project work entitled "INTELLIGENT POTHOLE DETECTION SYSTEM" submitted to the University of Calicut in partial fulfilment of the requirement for the award of the degree in Bachelor of Computer Application is a record of original work done by us under the guidance of MR. RAMSHEEL M, Assistant Professor Department of Computer Science, during the period of study in SADANAM KUMARAN COLLEGE, Mankara, Palakkad.

Place: Pathiripala
Date:

MANIKANDAN M
SUCHETH CS
MANEESH EM
DIYOOF

CONTENTS

ABSTRACT

1. ABSTRACT

The Intelligent Pothole Detection System presents a novel approach to address the perennial issue of road maintenance by leveraging computer vision and machine learning techniques. Conventional methods of pothole detection often suffer from limitations such as high costs, inefficiency, and lack of real-time capabilities. In contrast, our system harnesses the ubiquity and affordability of webcams to continuously monitor road surfaces. Through sophisticated computer vision algorithms and machine learning techniques, the system autonomously identifies and analyses anomalies indicative of potholes within the live video feed and report potholes on road surfaces, enabling timely repairs and improvements to enhance overall road

An automated system is designed to detect potholes through a webcam. The user can login or sign up using a setup page, which redirects them to the actual program. There, they can set the route they're about to cover. A webcam will be set up to capture the road. The footage is then reviewed by the program, and if any potholes are detected, the images are sent to the cloud. Government agencies can access these images to prioritize road repairs.

Extensive testing and validation have demonstrated the system's robustness, accuracy, and real-time performance. This innovative solution offers a cost-effective, scalable, and proactive approach to pothole detection, enabling timely maintenance interventions and ultimately contributing to significant improvements in road safety and infrastructure management

In conclusion, Pothole Detection offers a comprehensive solution to tackle the persistent problem of potholes, contributing to safer and more efficiently maintained road networks.

INTRODUCTION

2. INTRODUCTION

This project aims to investigate the prevalence and impact of potholes on Kerala's roads, focusing on recent news and factual data. Potholes are a persistent issue in Kerala, causing road accidents, vehicle damage, and inconvenience to commuters.

Through this study, we will analyze the frequency of pothole-related incidents, government initiatives to address the problem, and public responses. By leveraging data analysis and information gathering techniques, this project will provide valuable insights into the pothole situation in Kerala, contributing to potential solutions and policy recommendations

Traditional methods of pothole detection are often inefficient and reactive. In contrast, our system utilizes advanced computer vision and machine learning techniques to autonomously detect and locate potholes in real-time, offering a proactive approach to road maintenance and enhancing overall road safety and infrastructure management

MODULE DESCRIPTION

3. MODULE DESCRIPTION

The Intelligent Pothole Detection System project can be divided into different modules or components, each responsible for a specific task or functionality. Here is an overview of the main modules of the system:

1.Camera Module: The camera module is responsible for capturing images and videos of the monitored area. It is equipped with a high-quality camera with internet connectivity that can capture high-resolution images and videos, even in low light conditions.

2.Image Processing Module: The image processing module processes the images and videos captured by the camera and prepares them for analysis by the AI model. This module can also be used to enhance the quality of the images and videos, which can improve the accuracy of the AI model.

3.AI Model Module: The AI model module is responsible for identifying Potholes in the monitored area. The AI model is trained on a dataset of labelled images of pothole in the area. When a pothole is detected, the AI model triggers an alert to the notification system.

4.Notification Module: The notification module is responsible for sending alerts to the designated recipient when a pothole is detected. The notification system can be configured to send alerts immediately or with a certain time delay. The alert contains the image or video of the potholes and the time and location of the detection.

5. Cloud Storage Module: The cloud storage module stores the images and videos captured by the camera securely in the cloud. This module enables easy access to the footage and provides a backup in case the camera is damaged or stolen.

6.Mobile App Module: A mobile app could be developed to provide remote access to the camera and the notification system. The user can login or sign up using a setup page, which redirects them to the actual program. There, they can set the route they're about to cover.

SYSTEM STUDY

4. SYSTEM STUDY

4.1 EXISTING SYSTEM

- Current methods rely on manual inspections and citizen reports, leading to delayed detection and repair of potholes. There are several existing pothole detection systems that use various techniques, such as:
- Computer Vision and Image Processing- These systems use cameras mounted on vehicles or infrastructure to capture images of the road surface. Then, image processing algorithms are applied to detect potholes based on their shape, size, and texture.
- Machine Learning and Deep Learning- These systems use machine learning or deep learning algorithms to learn the patterns and features of potholes from large datasets of road images. Once trained, these algorithms can detect potholes in real-time or from existing video footage.
- Sensor-Based Systems- These systems use sensors mounted on vehicles or infrastructure to detect potholes based on changes in the road surface or vehicle behaviour.
- Video-Based Pothole Detection System - Develop a computer application that uses machine learning techniques to detect potholes from existing video footage. The system can use a dataset of road images to train a deep learning algorithm to detect potholes based on their features.

LIMITATIONS:

- While Intelligent Pothole Detection System have many benefits, there are also some limitations that should be considered. Here are some potential limitations of the Intelligent Pothole Detection System project:
- Dependence on environmental conditions: Intelligent pothole detection systems that rely on computer vision and image processing can be affected by lighting and weather conditions. For instance, low light conditions, shadows, and rain can make it difficult to accurately detect potholes.
- Computational cost: Deep learning models used in intelligent pothole detection systems can be computationally expensive, requiring high computational power and memory. This can be a challenge for edge devices with limited resources.
- Limited accuracy: While intelligent pothole detection systems can achieve high accuracy, they may still miss some potholes or false positives. This can be due to various factors, such as the size and shape of potholes, lighting conditions, and sensor performance.
- Real-time processing: Real-time processing of video data can be challenging, especially when using resource-constrained edge devices. It is essential to optimize the models and systems for real-time processing to ensure accurate and timely pothole detection.

4.2 PROPOSED SYSTEM

- The proposed system for the Intelligent Pothole Detection System project would consist of the following components:
- Camera with internet connectivity - A high-quality camera with internet connectivity would be used to capture images and videos of the monitored area. The camera should be capable of capturing high-resolution images and videos even in low light conditions.
- AI model - An AI model would be trained on a dataset of labelled images to identify known and unknown potholes. The AI model should be able to distinguish between different potholes and objects in the monitored area and trigger an alert when it detects a pothole.
- Notification system - A notification system would be used to send alerts to the designated recipient when a pothole is detected. The notification system can be configured to send alerts immediately or with a certain time delay. The alert should contain the image or video of the pothole and the time and location of the detection.
- Cloud storage - The images and videos captured by the camera would be stored securely in the cloud. The Government agencies can access these images to prioritize road repairs. This would enable easy access to the footage and provide a backup in case the camera is damaged or stolen.
- Mobile app - A mobile app could be developed to provide remote access to the camera and the notification system. The user can login or sign up using a setup page, which redirects them to the actual program. There, they can set the route they're about to cover.

4.3 FEASIBILITY STUDY

The feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and time that is spent on it. Feasibility study lets the developer foresee the future of the project and its usefulness. Feasibility study is a test of the system proposed regarding its workability, impact on the organization, ability to meet the needs and effective use of resources. Thus, when a new project is proposed, it normally goes through a feasibility study before it is approved for development.

Steps in feasibility analysis eight steps involved in the feasibility analysis are: Form a project team and appoint a project leader. Prepare system flowcharts. Enumerate potential proposed system. Define and identify characteristics of proposed system. Determine and evaluate performance and cost effectiveness of each proposed system. Weight system performance andcost data. Select the best-proposed system. Prepare and report final project directive to management.

This document provides the feasibility of the project that is being designed and lists variousareas that were considered very carefully during the feasibility study of this project such as:

- Technical feasibility
- Economic feasibility
- Operational feasibility
- Legal feasibility

A feasibility study for the AI security camera project would involve assessing the practicality and viability of the project. The following factors should be considered:

4.3.1 Technical Feasibility

- The technical feasibility of the project would involve assessing the availability of the necessary hardware and software components, as well as the technical skills required to implement and maintain the system. The camera and AI model must be compatible with each other, and the notification system must be reliable and secure.

4.3.2 Economic Feasibility

The economic feasibility of the project would involve assessing the costs and benefits of implementing the system. This would include the cost of hardware, software, installation, maintenance, and training. The benefits of the system would include improved security and reduced reliance on physical security teams.

4.3.3 Operational Feasibility

The operational feasibility of the project would involve assessing the ease of use and effectiveness of the system. This would include testing the accuracy and reliability of the AI model, as well as the responsiveness and effectiveness of the notification system.

4.3.4 Legal Feasibility

The legal and regulatory feasibility of the project would involve assessing the applicable laws and regulations related to the use of AI-based security cameras. This would include privacy laws, data protection laws, and regulations related to the use of surveillance equipment in public places.

SYSTEM SPECIFICATION

5. SYSTEM SPECIFICATION

SCRIPTING LANGUAGE

Python

Python is a high-level, interpreted programming language. It is designed to be easy to read and write, with a clean syntax that emphasizes code readability. Python's popularity has grown significantly over the years, thanks in part to its ease of use and versatility.

Python is used for a wide variety of tasks, including web development, data analysis, scientific computing, and artificial intelligence. Its large and active community has developed numerous libraries and tools that make Python even more powerful and useful for a variety of applications.

One of the key features of Python is its dynamic typing system. Unlike languages like Java or C++, Python is not statically typed, which means that variable types are determined at runtime rather than being explicitly declared. This allows for more flexibility in writing code, and can lead to more efficient development.

Python is also interpreted, which means that code is executed by an interpreter rather than being compiled into machine code like some other programming languages. This makes it easy to write and run Python code on a wide variety of platforms, including Windows, macOS, and Linux.

Python supports object-oriented programming, allowing developers to create reusable code in the form of objects and classes. Python also supports functional programming paradigms, making it a highly versatile language that can be used in a wide variety of contexts.

Finally, Python comes with a large standard library that provides a wide range of modules and functions for common tasks, such as working with files, networking, and data processing. This library, combined with the numerous third-party libraries and tools available, makes Python a powerful and flexible language for developing a wide variety of applications.

Visual Studio Code

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux, and Mac OS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, NodeJS, Python and C++. It is based on the Electron framework, which is used to develop Node.js Web Applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (codename “Monaco”) used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services.

ABOUT THE OPERATING SYSTEM

Windows

Windows is a family of operating systems developed and maintained by Microsoft Corporation. It was first introduced in 1985 and has since become the most widely used desktop operating system in the world. Windows is designed to be user-friendly and versatile, and is used by millions of people for a wide variety of tasks, including web browsing, document editing, gaming, and multimedia playback.

Windows is known for its graphical user interface, which provides a visual representation of the operating system and allows users to interact with it using a mouse and keyboard. Windows also supports touch input, allowing users to interact with the operating system using their fingers on a touch-enabled screen.

The latest version of Windows is Windows 11, which was released in 2021. Windows 11 includes a number of new features and enhancements, such as a redesigned Start menu, improved touch input, and better support for multiple monitors.

Windows supports a wide range of hardware and software, including desktop computers, laptops, tablets, and smartphones. It also supports a wide range of software applications, including web browsers, productivity software, games, and multimedia players.

One of the key features of Windows is its ability to run multiple applications simultaneously, allowing users to switch between applications and perform multiple tasks at once. Windows also includes a number of built-in tools and utilities, such as a file manager, task manager, and control panel, which allow users to manage and customize their operating system.

Overall, Windows is a versatile and user-friendly operating system that is used by millions of people around the world. It is an essential tool for many businesses, organizations, and individuals, and continues to evolve and improve with each new release.

HARDWARE SPECIFICATION

Processor

HARDWARE REQUIREMENTS:

Processor : i3 Above

RAM : 4GB Above

Solid State Drive :256GB(min)

SOFTWARE REQUIREMENTS:

Development Platform : Windows

IDE : Visual Studio Code for software development

Front End : Python(Tkinder)

Back-End : Dataset

SYSTEM DESIGN

6. SYSTEM DESIGN

The Intelligent Pothole Detection System is composed of several components working together to detect potholes and trigger alerts. Here's an overview of the system design:

Camera: The system uses a camera with internet connectivity to capture images and videos of the monitored area.

AI model: An AI model would be trained on a dataset of labelled images to identify potholes. The AI model should be able to distinguish between different potholes and objects in the monitored area and trigger an alert when it detects a pothole.

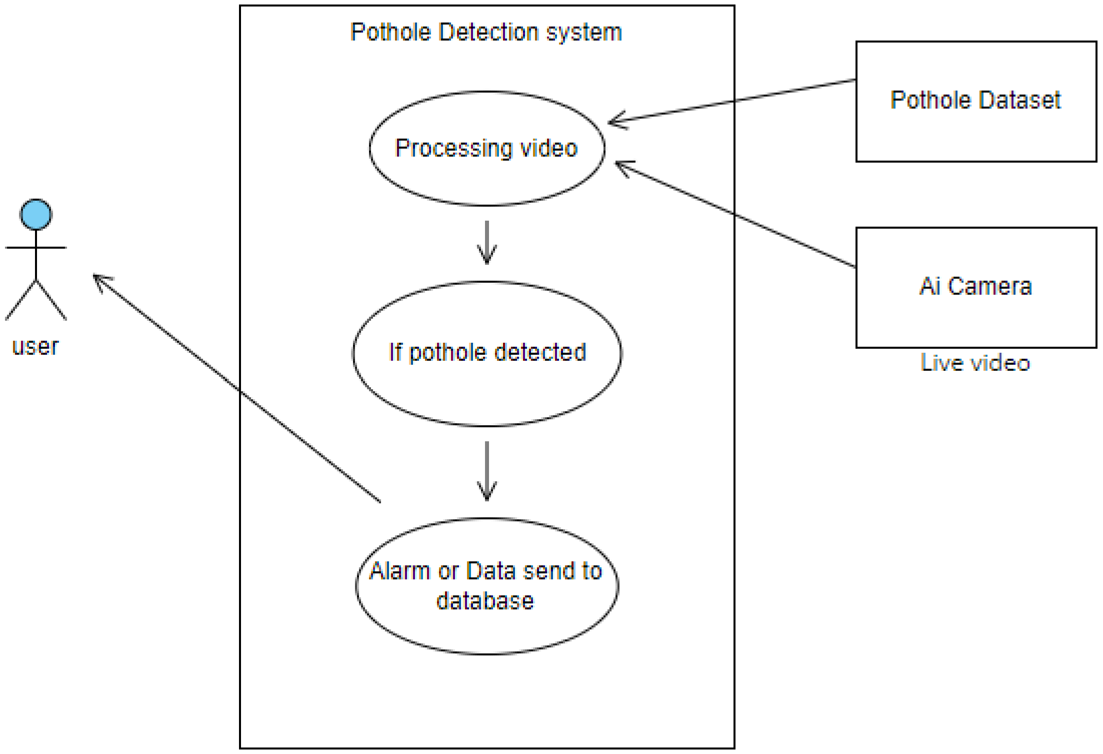
Notification system: The system includes a notification system that sends email alerts to designated recipients when a pothole is detected. The system can be configured to send alerts immediately or with a certain time delay.

User interface: The system includes a user interface that allows users to view camera feeds and configure system settings. The user interface can be accessed through a mobile application.

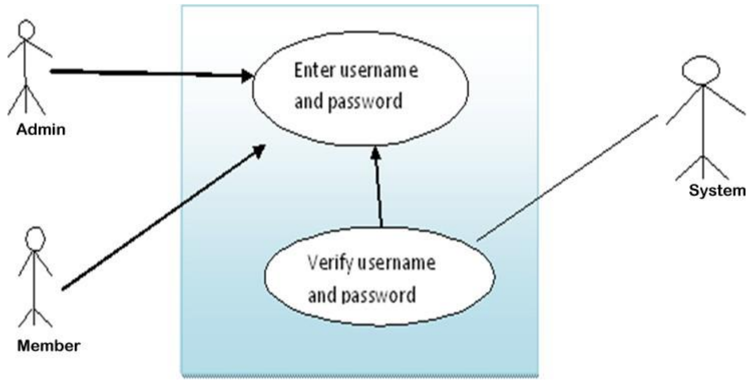
Cloud platform: The system can be hosted on a cloud platform, which provides scalability, security, and accessibility. This enables easy access to the footage and provides a backup in case the camera is damaged or stolen

Additional features: The system can be expanded to include additional features, such as automatic sound alarms or integration with other security systems.

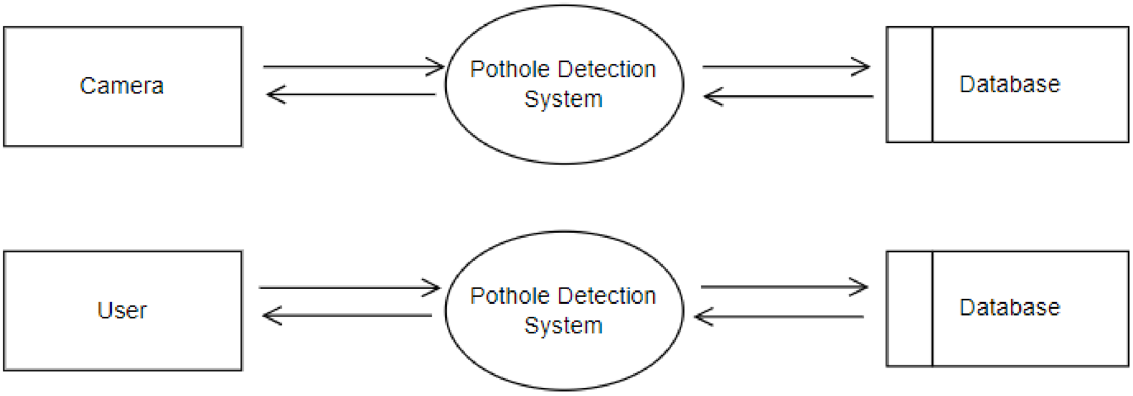
6.1 USECASE DIAGRAM



6.2 Login User Case:



6.3 DATA FLOW DIAGRAM



SYSTEM TESTING AND IMPLEMENTATION

7. SYSTEM TESTING AND IMPLEMENTATION

7.1 TESTING

Testing is an important part of the AI security camera project to ensure that the system functions correctly and meets its intended objectives. The following are the types of testing that can be performed on the system:

Unit testing: Unit testing involves testing each module of the system individually to ensure that it performs the intended functionality correctly. This can be done through manual or automated testing methods.

Integration testing: Integration testing involves testing the interaction between different modules of the system to ensure that they work together seamlessly. This can be done through manual or automated testing methods.

System testing: System testing involves testing the system as a whole to ensure that it meets its intended objectives. This can be done through real-world testing scenarios, such as simulating unknown persons entering the monitored area and evaluating the accuracy and responsiveness of the system.

Performance testing: Performance testing involves testing the system under different workloads to evaluate its performance and scalability. This can be done through stress testing, load testing, and endurance testing to ensure that the system can handle a large number of users and data.

Security testing: Security testing involves testing the system's security features to ensure that it is secure from cyber-attacks, data breaches, and unauthorized access. This can be done through vulnerability testing, penetration testing, and ethical hacking to identify and address any security vulnerabilities.

7.2 IMPLEMENTATION

Implementation is the stage in the project where the theatrical design is turned in to the working system and giving the confidence on the new system for the users that it will work eminently and effectively it involve careful planning. Investigation and design of method to achieve the changeover method A part from planning major task for preparing the implementation is education and training of users. The more complex system being implemented, the more involved will be the system analysis and the design effort required implementation

Implementation is the final and important phase. The most critical stage in achieving a successful new system and in giving the user confidence that the new system will work and being effective. The system can be implemented only after and if it is found to be working according to the specification.

At the beginning of the developing phase and initial implementation plan. Its creator to schedule and manage the many deferent activity that must be integrated throughout the development phase culminating in a change order plan for the operation phase.

TEST METHOD

8. TEST METHODS

The wide diffusion of Internet has produced a significant growth of the demand of Web-based applications with more and more strict requirements of reliability, usability, inter-operability and security. Due to market pressure and very short time-to-market, the testing of Web-based applications is often neglected by developers, as it is considered too time-consuming and lacking a significant payoff. This depreciable habit affects negatively the quality of the applications and, therefore triggers the need for adequate, efficient and cost-effective testing approaches for verifying and validating them. Though the testing of Web-based applications (Web applications, in the remaining of the paper) shares the same objectives of 'traditional' application testing, in most cases, traditional testing theories and methods cannot be used just as they are, because of the peculiarities and complexities of Web applications. Indeed, they have to be adapted to the specific operational environment, as well as new approaches for testing them are needed.

In common a web-based application is tested using:

- Performance testing
- Functionality testing
- Compatibility testing
- Accessibility testing
- Security testing
- Usability testing

8.1 PERFORMANCE TESTING

Performance testing objective is to verify specified system performances (e.g. response time, service availability). It is executed by simulating hundreds, or more, simultaneous users access over a defined time interval. Information about accesses are recorded and then analysed to estimate the load levels exhausting system resources. For Web applications, system performances is a critical issue because Web users don't like to wait too long for a response to their requests, also they expect that services are always available. Performance testing of Web applications should be considered as an everlasting activity to be carried out by analysing data from access log files, in order to tune the system adequately. Failures uncovered by performance testing are mainly due to running environment faults (such as scarce resources, or not well deployed resources, etc.), even if any software component of the application level may contribute to inefficiency.

This testing Include:

- Connection speed
- Load
- Stress

8.2 FUNCTIONAL TESTING

The functionality of the application like, calculation, business logic, validation links and navigation should be proper. In web-based application the following functional tests are carried out.

- Links

o Internal Links
o External Links
o Mail Links
o Broken Links

- Forms

o Field validation
o Error message for wrong input
o Optional and mandatory fields

- Database

o Testing will be done on the database integrity.

- Cookies

o Testing will be done on the client system side, on the temporary Internet files.

8.3 COMPATIBILITY TESTING

Compatibility testing will have to uncover failures due to the usage of different Web server platforms or client browsers, or different releases or configurations of them. The large variety of possible combinations of all the components involved in the execution of a Web application does not make it feasible to test all of them, so that usually only most common combinations are considered. As a consequence, just a subset of possible compatibility failures might be uncovered. Both the application and the running environment are responsible for compatibility failures.

8.4 ACCESSIBILITY TESTING

It can be considered as a particular type of usability testing whose aim is to verify that access to the content of the application is allowed even in presence of reduced hardware/ software configurations on the client side of the application (such as browser configurations disabling graphical visualization, or scripting execution), or of users with physical disabilities (such as blind people). In the case of Web applications, accessibility rules such as the one provided by the Web Content Accessibility Guidelines have been established, so that accessibility testing will have to verify the compliance to such rules. The application is the main responsible for accessibility, even if some accessibility failures may be due to the configuration of the running environment (e.g., browsers where the execution of scripts is disabled).

SYSTEM SECURITY

9. SYSTEM SECURITY

System security is a branch of technology known as information security as applied to computers and networks. The objective of system security includes protection of information andproperty from theft, corruption, or natural disaster, while allowing the information and property to remain accessible and productive to its intended users. The terms system security, means the collective processes and mechanisms by which sensitive and valuable information and services are protected from publication, tampering or collapse by unauthorized activities or untrustworthy individuals and unplanned events respectively. The technologies of system security are based on logic. As security is not necessarily the primary goal of most computer applications, designing a program with security in mind often imposes restrictions on that program's behaviour.

Internet is a part of everyday life; web applications are an essential component of every business activity. Customers and trading partners expect fast, accurate and secure applications with robust functionality. Companies want sites that are easy to maintain and update, yet cost effective. Auditors and security officers want to ensure that the web applications are controlled and that there is strong data integrity. All of these requirements need to be blended to ensure that each web application meets the company‘s goals, satisfies the customers and trading partners, and is secure and reliable.

The major security issues of web applications are: Types of unauthorized access

- ➢ Network security
- ➢ Firewalls
- ➢ Routers
- ➢ Intrusion detection and monitoring
- ➢ Virus detection and monitoring
- ➢ Encryption

- Operating system security
- Business continuance and disaster preparedness
- Hacker sites
- Control and security checklists

This is the process to determine that an Information System protects data and maintainsfunctionality as intended. The six basic security concepts are:

a) CONFIDENTIALITY

A security measure which protects against the disclosure of information to parties otherthan the intended users that is by no means the only way of ensuring.

b) INTEGRITY

A measure intended to allow the receiver to determine that the information which itreceives has not been altered in transit or by other than the originator of the information.

Integrity schemes often use some of the same underlying technologies as confidentiality schemes, but they usually involve adding additional information to a communication to form thebasis of an algorithmic check rather than the encoding all of the communication.

c) AUTHENTICATION

A measure designed to establish the validity of a transmission, message, or originator. Allows a receiver to have confidence that information it receives originated from a specificknown source.

d) AUTHORIZATION

This is the process of determining that a requester is allowed to receive a service orperform an operation. Access control is an example of authorization.

e) AVAILABILITY

Assuring information and communications services will be ready for use when expected. Information must be kept available to authorized persons when they need it.

f) NON-REPUDIATION

A measure intended to prevent the later denial that an action happened, or a communication that took place etc. In communication terms this often involves the interchange of authentication information combined with some form of provable time stamp.

DATA SECURITY

The focus behind data security is to ensure privacy while protecting personal or corporatedata. Data is the raw form of information stored as columns and rows in our databases, network servers and personal computers. This may be a wide range of information from personal files andintellectual property to market analytics and details intended to top secret. Encryption has become a critical security feature for thriving networks and active home users alike. This securitymechanism uses mathematical schemes and algorithms to scramble data into unreadable text. It can only by decode or decrypted by the party that possesses the associated key.

Data security wouldn't be complete without a solution to back up your critical information. Though it may appear secure while confined away in a machine, there is always a chance that your data can be compromised. You could suddenly be hit with a malware infectionwhere a virus destroys all of your files.

Someone could enter your computer and thieve data by sliding through a security hole inthe operating system. Perhaps it was an inside job that caused your business to lose those sensitive reports. If all else fails, a reliable backup solution will allow you to restore your data instead of starting completely from scratch.

USER SECURITY

User security lets your application use security rules to determine what it displays. It has twoelements:

Ensures that a valid user is logged-in, based on an ID and password provided by the user. ColdFusion (or, in some cases if you use web server authentication, the web server) maintains the user ID information while the user is logged-in.

AUTHORIZATION

Ensures that the logged-in user is allowed to use a page or perform an operation.

Authorization is typically based on one or more *roles* (sometimes called groups) to which the user belongs. For example, in an employee database, all users could be members of either the employee role or the contractor role. They could also be members of roles that identify their department, position in the corporate hierarchy, or job description. For example, someone couldbe a member of some or all of the following roles such as Employees, Human Resources, Benefits, and Managers. You can also use the user ID for authorization.

AUTHENTICATING USERS

You can use either, or both, of the following forms of authentication to secure your ColdFusionapplication:

- Web server authentication, where the web server authenticates the user and does notallow access to the website by users without valid login IDs.
- Application authentication, where the ColdFusion application authenticates the userand does not allow access to the application by users without valid login IDs.

POST IMPLEMENTATION

A Post-Implementation Review (PIR) is an assessment and review of the completed working solution. It will be performed after a period of live running; some time after the projectis completed. There are three purposes for a Post-Implementation Review:

- To ascertain the degree of success from the project, in particular, the extent to which it met its objectives, delivered planned levels of benefit, and addressed the specific requirements as originally defined.
- To examine the efficiency of all elements of the working business solution to see if further improvements can be made to optimize the benefit delivered.
- To learn lessons from this project, lessons which can be used by the team members andby the organization to improve future project work and solutions.

SYSTEM EVALUATION

The system evaluation involves the hardware and software as a unit. The hardware selectionis based on performance categories. The evaluation phase ranks vendor proposal and determines the one suited to the user's needs. It looks in to items such as price, availability and technical support.

In the operation phase, the system performance must be monitored not only to determinewhether or not they perform as planned, but also to determine if they should be modified to meetchanges in the information needs of the business. In the evaluation phase, the first step adopted was to look at the criteria listed earlier andrank them in the order of importance. Three sources of information are used in evaluating hardware and software. They are benchmark program, experience of other users and product reference manuals.

MAINTENANCE

10. MAINTENANCE

Software maintenance is the modification of a software product after delivery to correctfaults, to improve performance or other attributes, or to adapt the product to a modified environment. Maintenance covers a wide range of activities, including correcting, coding and design errors, updating documentation and test data and upgrading user support. Maintenance means restoring something to its original condition. After the installation phase is completed and the user staff is adjusted to the changes created by the candidate system, evaluation and maintenance begin. The maintenance phase ofthe software cycle is the time in which a software product performs the useful work. If the newinformation is inconsistence with the design specification, then changes have to be made. The importance of maintenance is to continue to bring the new system to standards. The system should be maintained and upgraded according to the technological advancements. It ensures the data integrity, data control and security. The system must be protected from fire and other natural calamities. The backup copies of data must be maintaineddaily so that we can prevent the loss of data due to various reasons.

Types of changes that can be encountered during the maintenance phase:

CORRECTIVE MAINTENANCE

Even with the best quality assurance activities, it is likely that the customer will uncoverdefects in the software. Corrective maintenance changes the software to correct the defects.

ADAPTIVE MAINTENANCE

Over time, the original environment (CPU, Operating System, Business Rules, External Product Characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

ENHANCEMENT MAINTENANCE

As software is used, the user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

PREVENTIVE MAINTENANCE

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs sothat they can be more easily corrected, adapted and enhanced.

ACTIVITIES OF A MAINTENANCE PROCEDURE

Maintenance activities begin where conversion leaves off. Maintenance is handled by the same planning and control used in a formal system project. The maintenance staff receives arequest for service from an authorized user, followed by a definition of the required modifications.

The source program and written procedures for the system acquired from the programming library. Program changes are then tested and submitted to the user for approval. Once approved, the modified documentation is filled with the library and a project completion notice is sent to the user, signalling the termination of the project. Although software doesn’t wear out like a piece of hardware, it ages and evenly fails to perform because of cumulative maintenance. A major problem with the software maintenance is its labour-intensive nature andtherefore the likelihood of errors.

CONCLUSION

11. CONCLUSION

In conclusion, the implementation of a pothole detection system represents a significant step forward in enhancing road safety and infrastructure maintenance. By harnessing advanced technologies such as sensors, machine learning, and data analytics, this system offers real-time monitoring and early detection of road defects, mitigating potential hazards for drivers and pedestrians alike. Furthermore, its ability to provide valuable data insights enables proactive maintenance strategies, ultimately contributing to cost savings and the prolonged lifespan of roads.

The pothole detection systems that utilize deep learning models and TensorFlow have shown significant potential in improving road safety and reducing infrastructure costs. TensorFlow, an open-source machine learning framework developed by Google, has been used to train and deploy high-performing convolutional neural networks (CNNs) for pothole detection. These systems that utilize TensorFlow and deep learning models represent a promising solution for addressing the issue of damaged road surfaces and improving road safety. These systems offer accurate and real-time pothole detection, enabling prompt repairs and reducing the risk of accidents and damage to vehicles. Additionally, the use of edge devices enables cost-effective deployment, making these systems accessible to a wider range of municipalities and organizations.

FUTURE SCOPE

12. FUTURE SCOPE

The INTELLIGENT POTHOLE DETECTION SYSTEM project has the potential for several future enhancements. Here are some of the possible areas for future development:

- Integrate with other systems: To make the pothole detection system more useful, you could consider integrating it with other systems such as traffic management systems or autonomous vehicles. This would enable the system to provide real-time alerts to drivers or traffic management systems, improving road safety and reducing maintenance costs.
- Crowdsourcing and Citizen Engagement: Implement crowdsourcing and citizen engagement features within the system, enabling drivers and pedestrians to report road defects they encounter in real-time via mobile applications or connected vehicles, thereby augmenting the system's data collection capabilities and enhancing community involvement in road maintenance efforts.
- Multi-camera support: The system can be expanded to support multiple cameras, providing wider coverage and more comprehensive security.
- Implement real-time inference: To make the pothole detection system more practical and useful, you could consider implementing real-time inference on edge devices such as smartphones or embedded systems. This would enable the system to detect potholes in real-time, without the need for cloud computing resources.
- Mobile app enhancements: The mobile app can be enhanced to include additional features such as live video streaming, real-time alerts, and remote access to the system.

BIBLIOGRAPHY

13. BIBLIOGRAPHY

- "OpenCV 4 Computer Vision with Python Recipes" by Joseph Howse and Prateek Joshi

- "Computer Vision: Algorithms and Applications" by Richard Szeliski

- "Python Machine Learning" by Sebastian Raschka and Vahid Mirjalili

- Fundamentals of Software Engineering.-Ragib mall.

APPENDIX

PROGRAM CODE

```
from email.mime.image import MIMEImage
import customtkinter as ctk
import tkinter.messagebox as tkmb
import shelve
import os
import argparse
import cv2
import numpy as np
import time
from threading import Thread
import importlib.util
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.image import MIMEImage

# Selecting GUI theme - dark, light , system (for system default)
ctk.set_appearance_mode("dark")

# Selecting color theme - blue, green, dark-blue
ctk.set_default_color_theme("blue")

user_entry = None
user_pass = None
logged_in_user = None
from_location_entry = None
to_location_entry = None
ready_button = None
capture_count = 0

def login():
    global logged_in_user
    username = user_entry.get()
    password = user_pass.get()

    with shelve.open("user_database.db") as db:
        if username in db:
            if db[username] == password:
                logged_in_user = username
                tkmb.showinfo(title="Login Successful", message=f"Welcome, {username}! You
have logged in Successfully")
```

```
                clear_existing_widgets()
                create_welcome_page()
                return
            else:
                tkmb.showwarning(title='Login Failed', message='Incorrect Password. Please
try again.')
        else:
            tkmb.showwarning(title='Login Failed', message='Username not found. Please
sign up first.')

def signup():
    global user_entry, user_pass  # Declare global variables
    username = user_entry.get()
    password = user_pass.get()

    with shelve.open("user_database.db") as db:
        if username in db:
            tkmb.showwarning(title='Sign Up Failed', message='Username already taken.
Please choose another username.')
        else:
            db[username] = password
            tkmb.showinfo(title="Sign Up Successful", message="You have signed up
successfully. Please login.")

def create_signup_page():
    global user_entry, user_pass  # Declare global variables

    clear_existing_widgets()

    label = ctk.CTkLabel(app, text='SIGNUP PAGE')
    label.pack(pady=12, padx=10)

    # Username
    user_entry = ctk.CTkEntry(app, placeholder_text="Username")
    user_entry.pack(pady=12, padx=10)

    # Password
    user_pass = ctk.CTkEntry(app, placeholder_text="Password", show="*")
    user_pass.pack(pady=12, padx=10)

    # Sign Up Button
    button_signup = ctk.CTkButton(app, text='Sign Up', command=signup)
    button_signup.pack(pady=12, padx=10)

    # Back Button
    button_back = ctk.CTkButton(app, text='Back', command=create_login_page)
    button_back.pack(pady=12, padx=10)


def create_login_page():
    global user_entry, user_pass  # Declare global variables

    clear_existing_widgets()

    label = ctk.CTkLabel(app, text="")
    label.pack(pady=20)

    frame = ctk.CTkFrame(app)
    frame.pack(pady=20, padx=40, fill='both', expand=True)

    label = ctk.CTkLabel(frame, text='LOGIN PAGE')
    label.pack(pady=12, padx=10)

    user_entry = ctk.CTkEntry(frame, placeholder_text="Username")
    user_entry.pack(pady=12, padx=10)

    user_pass = ctk.CTkEntry(frame, placeholder_text="Password", show="*")
    user_pass.pack(pady=12, padx=10)

    button_login = ctk.CTkButton(frame, text='Login', command=login)
    button_login.pack(pady=12, padx=5, anchor="center")

    button_signup = ctk.CTkButton(frame, text='Sign Up', command=create_signup_page)
    button_signup.pack(pady=12, padx=5, anchor="center")

    checkbox = ctk.CTkCheckBox(frame, text='Remember Me')
    checkbox.pack(pady=12, padx=10)

def create_welcome_page():
    global logged_in_user, from_location_entry, to_location_entry, ready_button

    clear_existing_widgets()

    label_user = ctk.CTkLabel(app, text=f"Welcome, {logged_in_user}!", font=("Algerian",
38, "bold"), anchor="center")
    label_user.pack(pady=20)

    label_from = ctk.CTkLabel(app, text="From:")
    label_from.pack(pady=5)
    from_location_entry = ctk.CTkEntry(app, placeholder_text="Enter from location")
    from_location_entry.pack(pady=5)

    label_to = ctk.CTkLabel(app, text="To:")
    label_to.pack(pady=5)
    to_location_entry = ctk.CTkEntry(app, placeholder_text="Enter to location")
    to_location_entry.pack(pady=5)

    ready_button = ctk.CTkButton(app, text='READY FOR POTHOLE DETECTING',
command=check_location_and_start_detection)
    ready_button.pack(pady=20)
    button_quit = ctk.CTkButton(app, text='Quit', command=app.quit)
    button_quit.pack(pady=10)

def check_location_and_start_detection():
    global from_location_entry, to_location_entry
    from_location = from_location_entry.get()
    to_location = to_location_entry.get()

    if from_location.strip() == "" or to_location.strip() == "":
        tkmb.showwarning(title='Incomplete Information', message='Please fill in both from
and to locations.')
    else:
        clear_existing_widgets()
        start_detection(from_location, to_location)  # Pass the from and to locations


def start_detection(from_location, to_location):
    global frame_rate_calc, capture_count

    # Initialize frame rate calculation
    frame_rate_calc = 1
    freq = cv2.getTickFrequency()

    # Initialize video stream
    videostream = VideoStream(resolution=(imW,imH),framerate=30).start()
    time.sleep(1)

    while True:
        # Start timer (for calculating frame rate)
        t1 = cv2.getTickCount()

        # Grab frame from video stream
        frame1 = videostream.read()

        # Acquire frame and resize to expected shape [1xHxWx3]
        frame = frame1.copy()
        frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        frame_resized = cv2.resize(frame_rgb, (width, height))
        input_data = np.expand_dims(frame_resized, axis=0)

        # Normalize pixel values if using a floating model (i.e. if model is non-quantized)
        if floating_model:
            input_data = (np.float32(input_data) - input_mean) / input_std

        # Perform the actual detection by running the model with the image as input
        interpreter.set_tensor(input_details[0]['index'],input_data)
        interpreter.invoke()

        # Retrieve detection results
        boxes = interpreter.get_tensor(output_details[boxes_idx]['index'])[0] # Bounding box coordinates of detected objects
        classes = interpreter.get_tensor(output_details[classes_idx]['index'])[0] # Classindex of detected objects
        scores = interpreter.get_tensor(output_details[scores_idx]['index'])[0] # Confidence of detected objects

        # Loop over all detections and draw detection box if confidence is above minimum threshold
        for i in range(len(scores)):
            if ((scores[i] > min_conf_threshold) and (scores[i] <= 1.0)):
                # Get bounding box coordinates and draw box
                ymin = int(max(1,(boxes[i][0] * imH)))
                xmin = int(max(1,(boxes[i][1] * imW)))
                ymax = int(min(imH,(boxes[i][2] * imH)))
                xmax = int(min(imW,(boxes[i][3] * imW)))

                cv2.rectangle(frame, (xmin,ymin), (xmax,ymax), (10, 255, 0), 2)

                # Draw label
                object_name = labels[int(classes[i])]
                label = '%s: %d%%' % (object_name, int(scores[i]*100))
                labelSize, baseLine = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.7, 2)
                label_ymin = max(ymin, labelSize[1] + 10)
                cv2.rectangle(frame, (xmin, label_ymin-labelSize[1]-10), (xmin+labelSize[0], label_ymin+baseLine-10), (255, 255, 255), cv2.FILLED)
                cv2.putText(frame, label, (xmin, label_ymin-7), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 0), 2)

                # Check if pothole is detected
                if object_name == "Pothole":
                    # Capture the image when pothole is detected
                    image_path = f"Pothole_{capture_count}.jpg"
                    cv2.imwrite(image_path, frame)  # Saving the frame as an image
                    capture_count += 1
                    # Send the captured image as an email attachment
                    send_email(image_path, from_location, to_location)  # Pass from and to locations

        # Draw framerate in corner of frame
        cv2.putText(frame,'FPS: {0:.2f}'.format(frame_rate_calc),(30,50),cv2.FONT_HERSHEY_SIMPLEX,1,(255,255,0),2,cv2.LINE_AA)

        # Display frame
        cv2.imshow('Object detector', frame)

        # Calculate framerate
        t2 = cv2.getTickCount()
```

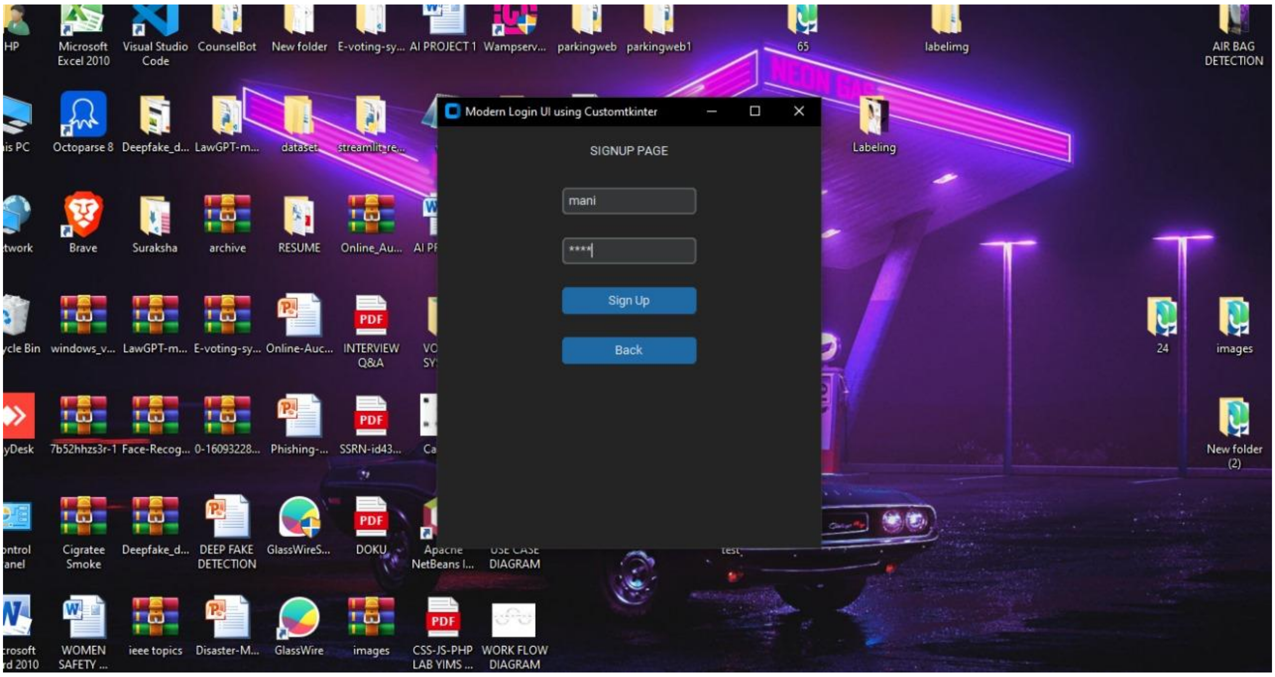
```
        time1 = (t2-t1)/freq
        frame_rate_calc= 1/time1

        # Press 'q' to quit
        if cv2.waitKey(1) == ord('q'):
            break

    # Clean up
    cv2.destroyAllWindows()
    videostream.stop()



def send_email(image_path, from_location, to_location):
    global capture_count  # Assuming capture_count is a global variable

    sender_email = "vishnudevp2003@gmail.com"  # Your email address
    receiver_emails = ["vishnudevp@yahoo.com", "arunmgcomsci@gmail.com"]  # List of
receiver email addresses
    password = "hqqi obps wuop azjg"  # Your email password

    # Create message container - the correct MIME type is multipart/related
    msg = MIMEMultipart('related')
    msg['Subject'] = "Emergency: Pothole Detected! Alert"
    msg['From'] = sender_email
    msg['To'] = ", ".join(receiver_emails)

    # Create the body of the message (a plain-text and an HTML version)
    text = f"Pothole has been detected From: {from_location} To: {to_location}. Please act
accordingly."
    html = f"""\
    <html>
      <body>
        <p>Pothole has been detected From: {from_location} To: {to_location}. Please act
accordingly.</p>
        <img src="cid:image1">
      </body>
    </html>
    """

    # Turn these into plain/html MIMEText objects
    part1 = MIMEText(text, 'plain')
    part2 = MIMEText(html, 'html')

    # Add HTML/plain-text parts to MIMEMultipart message
    # The email client will try to render the last part first
    msg.attach(part1)
    msg.attach(part2)

    # Add attachment image
    with open(image_path, 'rb') as attachment:
        image_part = MIMEImage(attachment.read(), name=f"Pothole_{capture_count}.jpg")
        image_part.add_header('Content-ID', '<image1>')
        msg.attach(image_part)

    try:
        # Create secure connection with server and send email
        with smtplib.SMTP_SSL("smtp.gmail.com", 465) as server:
            server.login(sender_email, password)
            server.sendmail(sender_email, receiver_emails, msg.as_string())
        print("Email sent successfully!")
    except Exception as e:
        print(f"An error occurred while sending the email: {e}")




def clear_existing_widgets():
    # Destroy all widgets currently present in the app
    for widget in app.winfo_children():
        widget.destroy()

# Main application
app = ctk.CTk()
app.geometry("400x400")
app.title("Modern Login UI using Customtkinter")

# Initial creation of login page
create_login_page()

# Define VideoStream class to handle streaming of video from webcam in separate
processing thread
class VideoStream:
    """Camera object that controls video streaming from the Picamera"""
    def _init_(self,resolution=(640,480),framerate=30):
        # Initialize the PiCamera and the camera image stream
        self.stream = cv2.VideoCapture(0)
        ret = self.stream.set(cv2.CAP_PROP_FOURCC, cv2.VideoWriter_fourcc(*'MJPG'))
        ret = self.stream.set(3,resolution[0])
        ret = self.stream.set(4,resolution[1])

        # Read first frame from the stream
        (self.grabbed, self.frame) = self.stream.read()

        # Variable to control when the camera is stopped
        self.stopped = False

    def start(self):
        # Start the thread that reads frames from the video stream
        Thread(target=self.update,args=()).start()
        return self

    def update(self):
        # Keep looping indefinitely until the thread is stopped
        while True:
            # If the camera is stopped, stop the thread
            if self.stopped:
                # Close camera resources
                self.stream.release()
                return

            # Otherwise, grab the next frame from the stream
            (self.grabbed, self.frame) = self.stream.read()

    def read(self):
        # Return the most recent frame
        return self.frame

    def stop(self):
        # Indicate that the camera and thread should be stopped
        self.stopped = True

# Define and parse input arguments
parser = argparse.ArgumentParser()
parser.add_argument('--modeldir', help='Folder the .tflite file is located in',
                    required=True)
parser.add_argument('--graph', help='Name of the .tflite file, if different than detect.tflite',
                    default='detect.tflite')
parser.add_argument('--labels', help='Name of the labelmap file, if different than
labelmap.txt',
                    default='labelmap.txt')
parser.add_argument('--threshold', help='Minimum confidence threshold for displaying
detected objects',
                    default=0.5)
parser.add_argument('--resolution', help='Desired webcam resolution in WxH. If the
webcam does not support the resolution entered, errors may occur.',
                    default='1280x720')
parser.add_argument('--edgetpu', help='Use Coral Edge TPU Accelerator to speed up
detection',
                    action='store_true')

args = parser.parse_args()

MODEL_NAME = args.modeldir
GRAPH_NAME = args.graph
LABELMAP_NAME = args.labels
min_conf_threshold = float(args.threshold)
resW, resH = args.resolution.split('x')
imW, imH = int(resW), int(resH)
```

```
use_TPU = args.edgetpu

# Import TensorFlow libraries
# If tflite_runtime is installed, import interpreter from tflite_runtime, else import from
regular tensorflow
# If using Coral Edge TPU, import the load_delegate library
pkg = importlib.util.find_spec('tflite_runtime')
if pkg:
    from tflite_runtime.interpreter import Interpreter
    if use_TPU:
        from tflite_runtime.interpreter import load_delegate
else:
    from tensorflow.lite.python.interpreter import Interpreter
    if use_TPU:
        from tensorflow.lite.python.interpreter import load_delegate

# If using Edge TPU, assign filename for Edge TPU model
if use_TPU:
    # If user has specified the name of the .tflite file, use that name, otherwise use default
'edgetpu.tflite'
    if (GRAPH_NAME == 'detect.tflite'):
        GRAPH_NAME = 'edgetpu.tflite'

# Get path to current working directory
CWD_PATH = os.getcwd()

# Path to .tflite file, which contains the model that is used for object detection
PATH_TO_CKPT = os.path.join(CWD_PATH,MODEL_NAME,GRAPH_NAME)

# Path to label map file
PATH_TO_LABELS = os.path.join(CWD_PATH,MODEL_NAME,LABELMAP_NAME)

# Load the label map
with open(PATH_TO_LABELS, 'r') as f:
    labels = [line.strip() for line in f.readlines()]

# Have to do a weird fix for label map if using the COCO "starter model" from
# https://www.tensorflow.org/lite/models/object_detection/overview
# First label is '???', which has to be removed.
if labels[0] == '???':
    del(labels[0])

# Load the Tensorflow Lite model.
# If using Edge TPU, use special load_delegate argument
if use_TPU:
    interpreter = Interpreter(model_path=PATH_TO_CKPT,
                              experimental_delegates=[load_delegate('libedgetpu.so.1.0')])
    print(PATH_TO_CKPT)
else:
    interpreter = Interpreter(model_path=PATH_TO_CKPT)
```

```
interpreter.allocate_tensors()

# Get model details
input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()
height = input_details[0]['shape'][1]
width = input_details[0]['shape'][2]

floating_model = (input_details[0]['dtype'] == np.float32)

input_mean = 127.5
input_std = 127.5

# Check output layer name to determine if this model was created with TF2 or TF1,
# because outputs are ordered differently for TF2 and TF1 models
outname = output_details[0]['name']

if ('StatefulPartitionedCall' in outname): # This is a TF2 model
    boxes_idx, classes_idx, scores_idx = 1, 3, 0
else: # This is a TF1 model
    boxes_idx, classes_idx, scores_idx = 0, 1, 2

# Main application loop
app.mainloop()
```

SCREENSHOTS

SCREENSHOTS

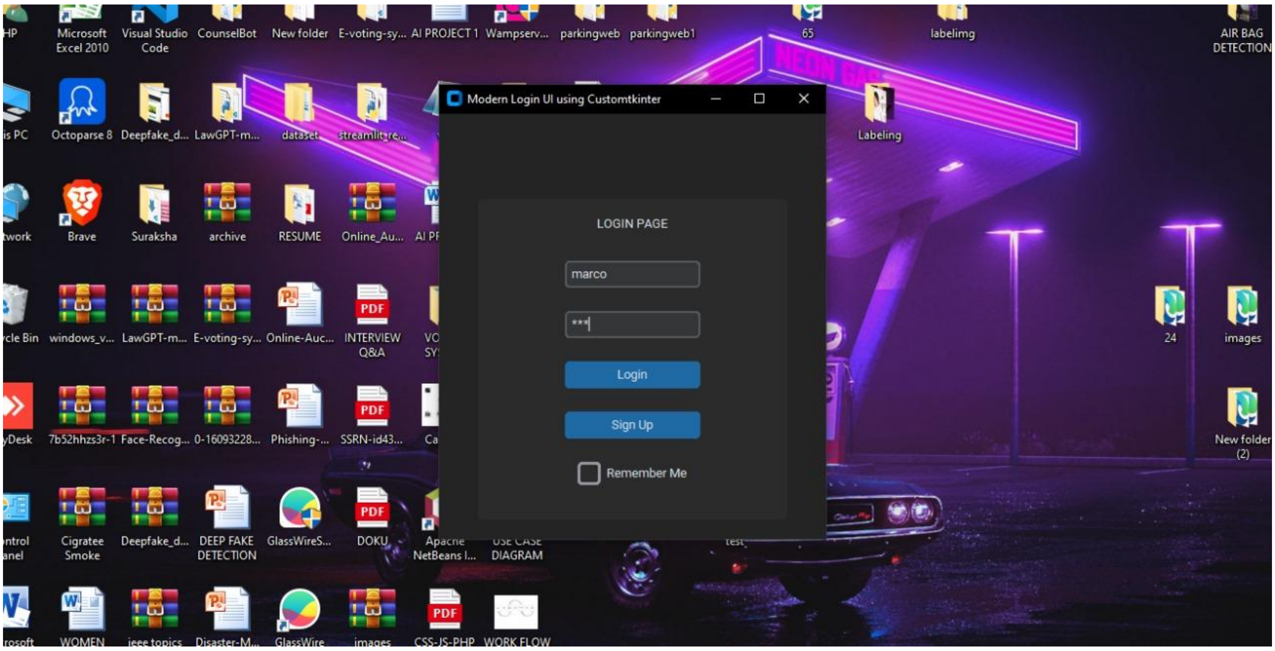


Modern Login UI using Customtkinter

SIGNUP PAGE

mani

****

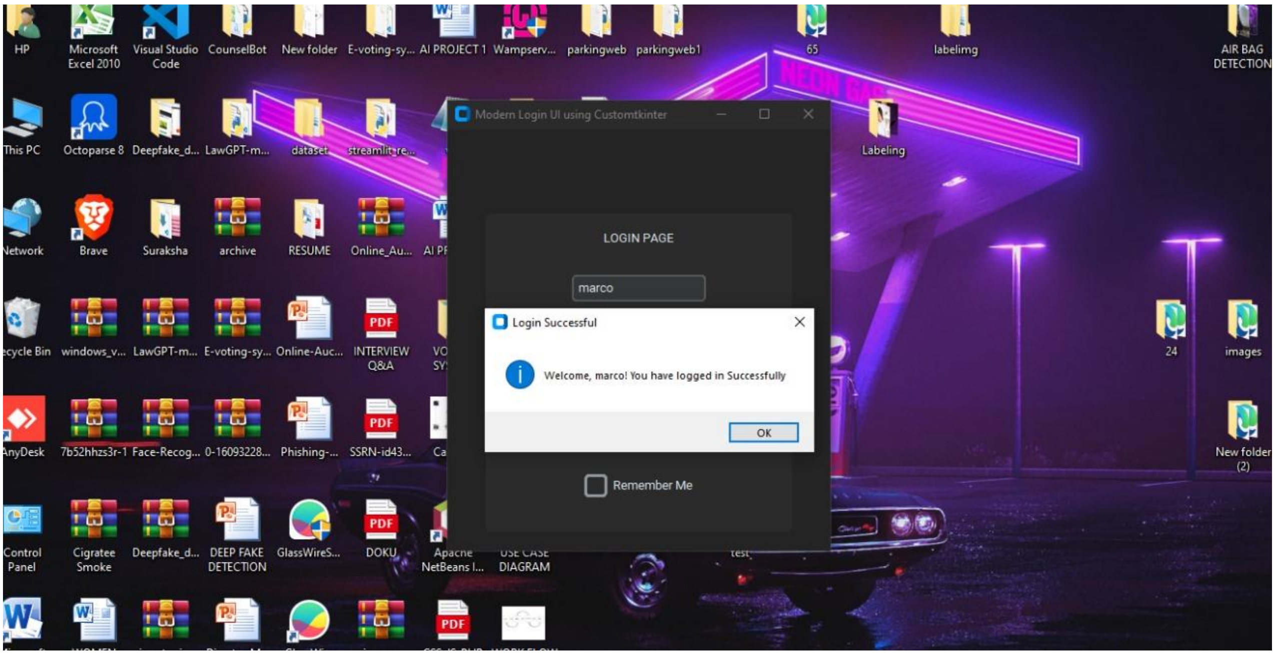


Modern Login UI using Customtkinter

LOGIN PAGE

marco

***



Modern Login UI using Customtkinter

LOGIN PAGE

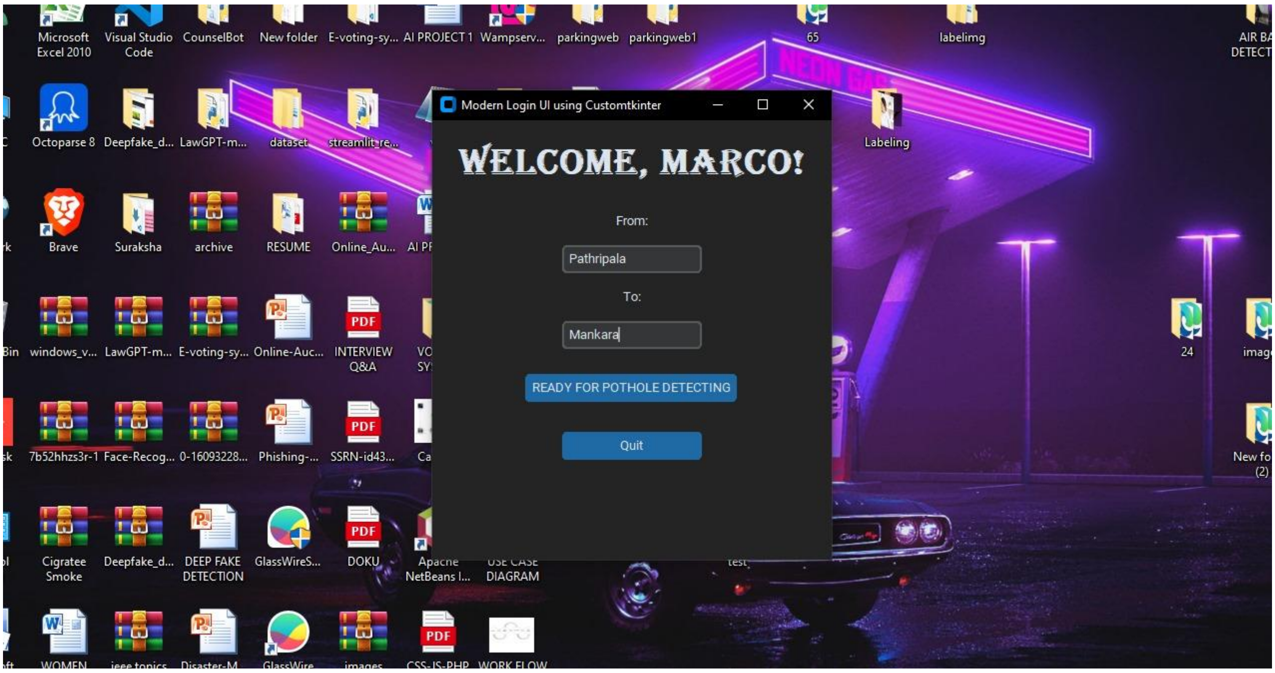
marco

Login Successful

Welcome, marco! You have logged in Successfully

OK

□ Remember Me



Modern Login UI using Customtkinter

WELCOME, MARCO!

From:

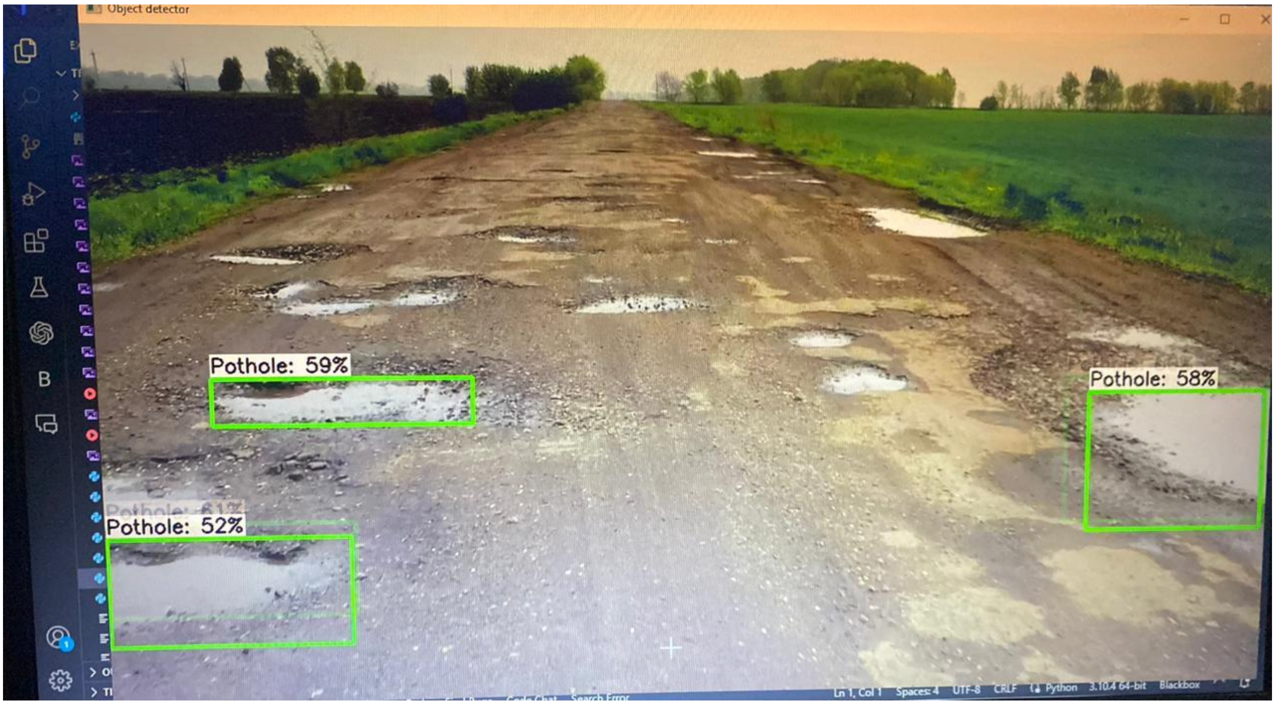
Pathripala

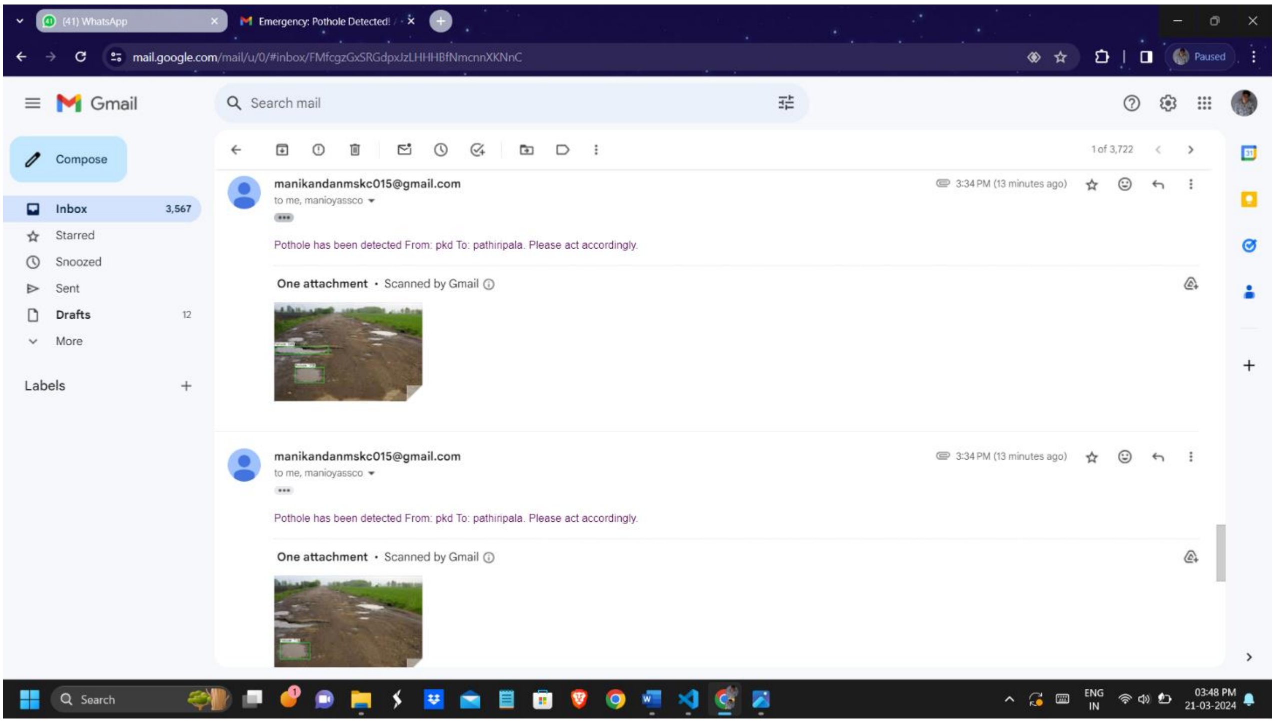
To:

Mankara

READY FOR POTHOLE DETECTING

Quit





manikandanmskc015@gmail.com 3:34 PM (13 minutes ago)

to me, manioyassco

Pothole has been detected From: pkd To: pathiripala. Please act accordingly.

One attachment • Scanned by Gmail

manikandanmskc015@gmail.com 3:34 PM (13 minutes ago)

to me, manioyassco

Pothole has been detected From: pkd To: pathiripala. Please act accordingly.

One attachment • Scanned by Gmail