

CHAPTER 1

INTRODUCTION

Technology has been the engine of transformation throughout life. Incorporating technology into daily life has been appreciated whether it is through television, the internet or moveable type. Within the confines of a civilized society, the advantages of technical breakthroughs vastly outweigh the negatives. The new ways for us to improve and streamline our lives are brought about by improving technologies. This resulted in the development of numerous branches, one among them is Data Science. Simply said, Data Science is the study of how and where data can be found and originated based on what it stands for and how frequently it becomes a significant resource in development of businesses. High amounts of organised data together with unstructured data can be mined for looking patterns that can be used to lower the cost, increase productivity, observe potential market developments and strengthen an organization's competitive advantage.[2]

Data science is a multidisciplinary blend of data inference, algorithm development and technology in order to solve analytically complex problems. Data science is used by almost all the industries like educational institutions, finance, healthcare and business to handle large volume of data. The practical applications range from predicting stock movement to predicting cancer; used in image processing to identity recognition, audio processing for speech to text prediction. Since most of the people in the world are facing problems in the field of authentication and security. We are able to provide a real time eye tracing for password authentication for people who authenticate themselves using Morse code.[1]

Advancement in the technology of authentication and authorization has been supported in the 21st century a lot as we know. Personal identification numbers (PIN) are widely used for the user authentication and security since the late 90's. Since PIN numbers can be easily forged these days, we prefer to follow different approach. PIN authentication with hands-off gaze-based PIN entry techniques on the other hand leaves no physical footprints behind and therefore offer a more secure password entry option.[4]

1.1 Problem Statement

We always have our cell phones within reach, but how many of us are aware of the dangers they pose? Threats to mobile security keep increasing: Over sixty percent of digital fraud now occurs on cell phones, including phishing scams and password theft. Security is even more crucial when using our phones for vital tasks like banking. According to Randy Pargman, senior director at cybersecurity firm Binary Defence, "the more you rely on your phone for daily tasks, the more it will impact you if your device is compromised." That's another reason you shouldn't keep certain items on your smartphone.

Fortunately, you can still use your phone safely by being aware of potential dangers and exercising caution. In order to help you protect yourself, your phone, and your data, we've compiled a list of this year's top dangers to smartphone security.[6]

Data breaches

Read the small print before downloading a new application to your cell phone. Almost all smartphone apps get data from your phone. Your name, date of birth, credit card and bank account information, location history, contact list, photographs, and other details may be included in this information. When you consider how much of your online behaviour is stored on servers run by the app creators, "it's quite scary." All of this information may be compromised if those servers are breached by hackers or whenever a technical problem leaves them exposed. They can be taken into account for scams by fenders.

Phishing attacks

Phishing is the practise of using texts, emails, or even phone conversations to trick a target into divulging a confidential information, visiting a link to download malware, or confirming a transaction. Phishing continues to be one of the most popular and effective methods used by cybercriminals to compromise users. Always check the identity of the person who is approaching you for your personal information to prevent falling into a phishing scam. For instance, informing the caller posing as your bank that you'll call them back at the lender's main line. Additionally, since these texts are probably scams, you need to eliminate them right away.

Spyware

Apps that claim to track the whereabouts of your family members and children should be avoided because they are actually spyware which is designed to allow extremely invasive digital surveillance through a smartphone. These applications allow abusers to do a variety of things, including read text messages and emails, locate the mobile device, listen in on adjacent discussions covertly, and take images. Even less nefarious apps can still gather data regarding your smartphone usage. While it can be challenging to make your phone impossible to track, it is still entirely feasible to do so to some extent in order to ensure safety. He advises staying away from apps that ask for a lot of permissions or any accessibility-related permission.[7]

Poor password security

According to the year 2019 Google/Harris research, more than half of Americans repeat identities across various accounts. Cybercriminals love those passwords because they may access dozens of accounts by buying huge databases of breached and leaked passwords on the dark web. Advises enabling authentication using multiple factors and utilising a password manager programme to create and save different passwords for each account in order to safeguard your online accounts against hackers. In this way, he argues, "you won't have to rely solely on the name of your pet to keep your money secure and stay out from the control of thieves." Avoid making the password errors that hackers want you to make when you secure your accounts with passwords.

Man-in-the-Middle (MitM) Attack

Attacks called "Man-in-the-Middle" (MitM) include a hacker stealing network conversations in order either to listen on or change the data being transmitted. While different systems may be vulnerable to this kind of attack, mobile devices are particularly prone to MitM attacks. SMS messages can be easily intercepted, and mobile applications may employ unencrypted HTTP for the transfer of possibly sensitive data, in contrast to web traffic, which frequently uses protected HTTPS during communication.

1.2 Solution for the Problem

Businesses need corporate mobile security measures due to the vast and varied mobile attack ecosystem. This is particularly so given that more and more people are turning to remote work, which makes these mobile devices an essential part of the information technology (IT) system of an organisation.

An efficient mobile threat defence solution must be capable to recognise and counter a wide range of assaults while delivering a satisfying user experience. To do this, it is necessary to put these guiding ideas into practise:

- A complete picture of security for devices, apps, and networks
- Complete adaptability and scalability
- Complete awareness of the risk posed by the mobility workforce
- By design, safeguarding confidentiality
- Optimum user encounter

To protect business information, Check Point's Synergy Mobile offers complete mobile protection. by protecting employees' cell phones against all threat vectors, including OS, system, and app attacks. Check Ask for a customised demo with a mobile security expert to see Harmony Mobile's features for yourself. You're more than welcome to use the trial period for free to give it an opportunity. Check out this mobile protection consumer guide for more details on the guiding principles and other crucial components of a mobile safety solution.

The majority of organisations can benefit from implementing strict password policies. Larger passwords, more intricate passwords, more frequent password changes, or a mix of these ideas may be used. Longer passwords that aren't frequently rotated are actually safer than shorter ones. Users may be prevented from choosing bad passwords through password authentication. Users should be obliged to use multifactor authentication whenever viewing sensitive information or websites, frequently with the help from solutions for MFA.[1]

1.3 Existing Technique

In the twenty-first century, it has been supported to advance the technology of authentication and authorization. Since the late 1990s, personal identification numbers (PINs) have been utilised extensively for user authentication and security. We prefer to use a different strategy these days because PIN codes are so simple to hack. On the other side, PIN authentication using hands-free gaze-based PIN entry techniques leaves no physical traces and thus provides a more secure password entry option. The current technology does not offer a safe way to authenticate partially sighted people.[3]

Recheck your security fundamentals

Using biometric security and/or a PIN, pattern, or password on any of your gadgets is a fast hardly any-brainer that should be mentioned. Now. Every safety expert you speak with will tell you the same thing: failing to secure your possessions is the most likely reason for a security breach. As the cool kids used to say 15 to 20 years ago, you are the weakest link. Putting aside laughably archaic pop culture allusions, consider the fact that if your phone is unprotected by a passcode, every bit of its information is available for theft if you accidentally or purposefully leave the gadget unattended. Your email, papers, social networking accounts, and other items fall under here. as well as the whole collection of pictures (yes, including those images; hey, I'm not here to judge).

Android makes maintaining the security of your devices as simple as possible. With the help of the applications Smart Lock feature, you can set up your phone to constantly leave itself unlocked in a number of pre-approved "safe" situations, such as when you're at home, when a particular reliable Bluetooth-enabled gadget is attached, or even when it's in your pocket. That implies you aren't confronted with the additional security the rest of the time; it only appears when it's actually required.

View the saved Smart Lock passwords you've saved

Speaking of Smart Lock, one of the less-discussed features of Google's security system is its capacity to store credentials for applications and websites that you browse on your smartphone or tablet. Examine the list of stored passwords that Googling has to feed your user account as part of this yearly check-up to ensure that you understand what's available and which, if any, of your credentials have been compromised (which Google will blatantly alert you about towards the very forefront of that exact same screen).

Examine your confidentiality management setup

Although Google's saved the username and password method is an improvement over nothing, using an encrypted password administration service will give you stronger security guarantees, more sophisticated and practical features, and more support for in-app password filling.

There are many great choices, with widely favoured favourites like LastPass, 1Password, and Bit warden. Any such trustworthy service will function just as well on the desktop as it will on iOS, and the majority of them have largely comparable specifications for security. The primary variations are in price, additional features, the user interface, and the whole user experience.

It's time for you to begin using one of those services if you don't already. And should you already utilise one of these services, Check the application's settings right away to make sure you're utilising all the within the device security features it offers. For instance, with LastPass, which is you should make sure the options to automatically lock the app and whenever it is idle for longer than a few minutes are turned on. Check to see if the app requires a PIN or biometric identification before using it. And you ought to make sure the app is configured for offline access, just in case. (The Security section of LastPass's settings contains each of these options)

Like Google, the majority of reliable password managers now give you the opportunity to review all of your passwords and determine which ones need to be changed. Analyse the two-factor authentication predicament you are in.[6]

These days, it takes more than one password to secure a significant account, let alone one as valuable and extensive as your Google account. With two-factor authorization, each time users try to log in, a unique timing-sensitive code must also be entered in addition to your password. This considerably raises your level of security and lowers the likelihood that anyone would ever be able to access your personal information because they would need to know both your password and the physical location of your password-generating device, which is probably your mobile device, in order to do so. Visit this website to get began if two-factor authorization isn't already enabled for your Google account. Also, don't limit yourself to using Google alone: If a service, such as the password boss, Facebook or Twitter accounts, or any other quasi-Google storage assistance, supports two-factor identification, you should look into activating it. Once everything is set up, you'll need to generate one-time use codes from your phone using either the company's authentication system or your telephone itself as an authentication key. Additionally, you can use a third-party substitute like Authy, which has more features than Google's authentication service and can be installed on numerous devices.[7]

Improve the security of your lock screen.

Your Android device's lock screen serves as the gatekeeper, and there are a few things you can do to give it more strength and make sure it's ready for the job.

Consider the alerts you receive and how much of that information you would like to be displayed on your phone's lock screen initial, as any individual who gets a hold of your phone may have access to all of that information. Go to your system preferences' Display tab and choose "Lock screen" or "Lock screen display" if you frequently get important messages or simply want to raise your security and privacy game. Once enabled, it provides a quick way to disable all biometric and Smart Lock security features on your phone, making it so that only a pattern, PIN, or password could be used to bypass your lock screen and access your device.

The intention is that you could turn on secure mode and know that your information could not be discovered without your explicit authorization had you ever felt like you might be forced to unlock your phone with your fingerprint or face, whether it was by a law enforcement official or just a regular. When that mode is turned on, even alerts won't appear on the locking screen, and that increased level of security will last till. Following that, if an emergency ever arises, just keep in mind that you can always find a button to turn on that "Lockdown" feature in your cell phone's strength menu in addition to the standard options for restarting and shutting down your device. Though ideally you won't require it, you are now prepared in the event that you do.

1.4 Proposed Technique

The model comprises of a user experience and a database that operates on the back end. Because of the design of the GUI, the user may communicate with the system. Frontend: The user must first register by providing a user ID, a password (PIN), and a search phrase of their selection. After registering, logging in requires the user's user id and password. A web digicam is used to record the PIN and convert it into Morse code for entry. A web camera is used to record the PIN and convert it into Morse code for input. - Backend The submitted PIN is contrasted with the user-input PIN that was recorded in the database upon registration. If the PIN entered is wrong, the screen disappears. In the event that a user disregards their password, they may authenticate using the keyword, change their password, and if necessary, generate a new one-time password (OTP) each time they login.

1.5 Objective

The objectives of our project “MORSE CODE SECURITY BREACH” are as follows:

- To create a secure system to authenticate users who aren't completely blind.
- To create a secure password authentication system which uses Morse code.
- To make sure that the specified parts of the face are recognized accurately by the system.
- This project gives catchphrase and less using of hardware sensors which is using in nowadays.

1.6 Scope of the Project

The primary goal of this project is to prevent fraud from occurring in government or Army credentials. Comparatively speaking, eye trackers offer greater security than any biometric authentication method. Eye trackers are the tools used to gauge visual activity. Users who are physically disabled can now use their eyes to interact with computers thanks to this. Our primary motivation is to offer an authentication process that includes physically challenged individuals for everyone.

1.7 Organization of Report

Chapter 1: Introduction• This chapter talks about the introduction to our Eye tracking password system, problem statement, Aim of the project and motivation factors.

Chapter 2: Literature Survey• This chapter majorly deals with existing system and proposed system and related research works.

Chapter 3: System Requirement Specification• This chapter speaks about the merchandise perspective, user characteristics, its assumptions and dependencies, specific requirements, non-functional requirements and functional requirements

Chapter 4: System Design and Architecture• This chapter deals with the advance software engineering where the whole flow of the project is represented by professional data flow chart. This chapter mainly deals with sequence diagram for the project representation.

Chapter 5: Implementation• This chapter deals with the steps involved within the creation of the project work. it's defined with the assistant of code explanation for the convenience of reader.

Chapter 6: Testing• This chapter mainly deals with the varied sorts of the test cases to prove

the validity of the project.

Chapter 7: Snapshots• This chapter mainly deals with the graphical interface of the project to point out the output of the appliance.

Chapter 8: Conclusion and Future Work• This chapter is especially the summary of the whole project development and it also suggest a number of the enhancement idea which couldn't be covered up thanks to constraint of your time and resources.

References • This section mainly highlights all the journal and IEEE papers being referred for the event of the project

CHAPTER 2

LITERATURE SURVEY

A Literature Survey is an overview of the previously published works on a specific topic. The term can refer to full scholarly paper or a section of a scholarly work such as a book or an article.

2.1 Real time Eye Tracking for Password Authentication [1]

proposes a authentication process where a real time application for gaze-based PIN entry, eye detection and tracking for PIN identification using a smart camera. This process leaves no traces of physical footprints behind, therefore offering one of the most secure ways to authenticate the password.

2.2 Quantitative Analysis of Tennis Experts Eye Movement Skill [2]

proposes measurement the eye movements of an actual expert tennis player and a beginner tennis player. The measured eye movements of the players are compared and analysed. The eye movements are recorded using an eye-tracker. Main observation made in this paper is that beginners have a tendency to follow the tennis ball unconsciously for a moment.

2.3 Smart-Eye Tracking System [3]

proposes a Smart Eye tracking system which is designed for people with disabilities and elder people. The concept of this research is to apply eye movement to control appliances, wheelchair and communicate with caretaker. This system comprises four components, imaging processing module, wheelchair-controlled module, appliances-controlled module and SMS manager module. The image processing module consists of webcam and C++ customized image processing, the eye movement image is captured and transmitted to Raspberry Pi microcontroller for processing with OpenCV to derive the coordinate of eye ball. The coordinate of eye ball is utilized for cursor control on the Raspberry Pi screen to control the system. Besides the eye movement, the eye blink is applied in this system for entering a command as when you press enter button on keyboard. The wheelchair-controlled module is a cradle with two servos that can be moved to two dimensions and also adaptable to other wheelchair joysticks. This system also remotely controls some appliances and communicates with caretaker via sending messages to smartphone.

2.4 Extension of Desktop Control to Robot Control by Eye Blinks using SVM [4]

proposes issues related to Accessibility which should eliminate or at least reduce the distance between disabled people and technology. For severely-impaired persons there are still many challenges that must be overcome. We present eye tracking as a valuable support for disability in the accomplishment of hands-free tasks. Moreover, we stress the potentials of eye-based interfaces to enhance the user-machine interaction process in “traditional” activities based on keyboard and mouse. Through the description of some of the projects they have recently developed a robot which can move according to the movements of the eye balls and can be triggered with some actions based on the eye blinks.

2.5 Eye Movement Related EEG Potential Pattern Recognition for Real-Time BMI [5]

proposes study which aims at rapid BMI (Brain Machine Interface) pattern recognition for the eye-ball movement which is considered to be removed factor from EEG (Electroencephalogram) as artefact. We investigated the repeatability of eyeball movement ERP (Event related Potential) and the characteristics which possess steady, high voltage and 50ms rapid reaction. As ERP pattern discriminator, this paper proposes 3 methods to extract and distinguish characteristic patterns induced by several directional ocular movements.

2.6 Eye Contact Game Using Mixed Reality for The Treatment of Children With Attention Deficit Hyperactivity Disorder [6]

proposes an observation where many children with ADHD perform poorly in their academics. They also face difficulty in their social lives due to lack of attention and also due to lack interpersonal skills and often continues to their adult life. Considering the problem, this paper offers a solution where they have introduced and demonstrated the benefits of a new type of treatment, an eye-contact game which successfully exploits mixed reality technology. To the best of our knowledge, this study is one of the first studies to use a mixed reality head-mounted display to treat children with attention deficit hyperactivity disorder and to prove its potential as a treatment for clinically diagnosed children.

CHAPTER 3

REQUIREMENT SPECIFICATION

A software requirement specification is description of a software system to be developed. It is modelled after business requirement specification (CONOPS). The software requirement specification lays out functional and non-functional requirements.

3.1 Software Requirements

The software requirements are as follows:

- Operating system : Windows XP / 7 /8/10
- Coding Language : Python
- IDE : Python

3.1.1. Python 3.7

Python is well renowned for being a quick and practical way to work with structured data. Python offers classes to organise data and associate common behaviours with representations of that information, but classes which include a number of initializers are frequently afflicted by the requirement for a lot of boilerplate code to instantiate them. The most recent release of this language, Python 3.7, which aims to simplify complicated tasks, is now available for production use. The most notable changes and upgrades in Python 3.7 involve:

- Reduced boilerplate when working with data in classes thanks to data classes.a modification to how generators handle exceptions that might not be backwards compatible.
- "Developing mode" for the language.
- Time objects with a nanosecond resolution.
- UTF-8 mode, which sets the surrounding environment's initial encoding to UTF-8.
- An updated built-in that activates the debugger.

In June 2018, the programming language Python saw the release of version 3.7. In comparison to earlier versions, it added a number of new features. Here is a quick rundown of some of Python 3.7's salient features:

Data Classes: A new decorator named data class was added to Python 3.7, making it simpler to

construct classes that serve primarily as data storage structures. Boilerplate code for initialising attributes, contrasting objects, and other tasks is generated automatically.

Type Hints: Type hints, which let you define the anticipated types of factors function arguments, and return values, were substantially improved in Python 3.7. Static code checkers or Integrated Development Environments may leverage this to give better code analysis and it helps to make the code clearer.

offering a method for generating and maintaining contextual parameters that are available inside a certain context, such as within a specific thread or job.

Asuncion has been improved in Python 3.7. The asyncio module is used for asynchronous programming. Notably, improvements were made to make working with coroutines and jobs simpler, including the addition of the `asyncio.create_task()` method.

speed Enhancements: Python 3.7 came with a number of optimisations and speed upgrades, including a faster built-in `unzip ()` function, quicker method calls, and quicker startup times for big codebases.

Additional Features: Python 3.7 added a number of minor enhancements, such as syntactic simplifications like the use of underscores in numerical literals to increase readability, more practical ways to customise class construction, and additional built-in modules.

It's important to remember that Python has developed further since the publication of Python 3.7, which is and later versions now have more capabilities and enhancements. Since Python 3.9 was the most recent stable version at the time of my expertise limit in September 2021, it is advised to consult the official Python website for the most recent details on Python functionality and version.

3.1.2. OpenCV

OpenCV is a sizable open-source framework for processing pictures, machine learning, and visual analysis. It now plays a significant part in real-time operation, which is crucial in modern systems. Using it, one can analyse pictures and videos to find faces, objects, and even human handwriting. Python is able to handle the OpenCV array structure for analysis when it is integrated with different libraries, such as NumPy. We use vector space and apply mathematical techniques to these characteristics to identify visual patterns and their various features. A library of programming functions, known as OpenCV (Open Source Computer Vision Library), is

mostly used for real-time computer vision.[1] It was initially created by Intel, then sponsored by Willow Garage and Itseez (which Intel[2] eventually acquired[3]). The framework is cross-platform and distributed under Apache Licence 2 as free and open-source software. OpenCV offers acceleration using the GPU for real-time activities starting in 2011.

OpenCV, which is an Open Source Library, is a machine learning and computer vision library that was first created by Intel. It offers a variety of tools and functionalities for processing images and videos, detecting and tracking objects, extracting features, and more. Here is a quick summary of OpenCV.

Processing of images and videos: OpenCV provides a wide range of functions for both simple and complex video and image processing jobs. It has the ability to handle colour spaces, read and write photos and videos, manipulate pixels, apply filters and transformations, change brightness and contrast, and execute geometric operations.

Object identification and tracking are made possible by the pre-trained models and methods included in OpenCV. It supports well-liked methods for identifying faces and other objects, such Haar cascades. Additionally, OpenCV allows more sophisticated methods like the tracking and object detection using deep learning-based models.

Extraction and Matching of Features in pictures: OpenCV offers a number of algorithms for collecting and matching characteristics in pictures. For tasks like picture identification, image sewing, and image registration, these properties can be exploited. For the extraction of features and matching, OpenCV provides methods like SIFT (Scale-Invariant Field Transformation) with SURF (Speeded Accelerated Robust Features).

3D Reconstruction and Camera Calibration: OpenCV has camera calibration routines that are necessary for determining camera settings and correcting distortions.

Computer vision tasks may be combined with deep learning methods because to OpenCV's integration with frameworks for machine learning like TensorFlow and PyTorch. This integration makes it possible to use neural networks for tasks like picture classification, object identification, and semantic segmentation.

Support for Multiple Operating Systems, Multiple Languages, and Mobile devices: OpenCV is a multi-platform library that works with macOS, Windows, Linux, and numerous mobile devices. It makes it available to a variety of developers by offering bindings to a number of languages, including C++, Java, Python, and MATLAB.

In fields including robotics, automobiles healthcare, security, and entertainment, OpenCV is extensively employed. It is a well-liked option for machine learning application because to its broad set of capabilities, thorough documentation, and huge community.

3.1.3. CSV FILE

A delimited text file that employs commas to separate values is known as a comma-separated values (CSV) file. The file's lines each contain a data record. Several fields, surrounded with commas, make up each record. The designation of the format of a file is derived from the fact that fields are separated by commas. Each line in a CSV file will normally have the same number of fields if the data being stored is tabular (numbers and text).

Incomplete standardisation exists for the CSV file format. Commas provide the foundation for separating fields, however comma in the information or imbedded line breaks need to be treated carefully. While some implementations forbid such material, others include the field in quotation marks, necessitating the requirement for escape if quotation marks are present in the data.

The word "CSV" also refers to a number of related field delimiter-separated formats that utilise semicolons instead of commas.[2] Among these are values with tabs and values with spaces between them. Parsing is made much simpler by a delimiter that is guaranteed not to be present in the data. Although a non-comma field separator is used, alternative delimiter-separated files are sometimes given a ".csv" extension. Inaccurate wording might hinder the sharing of data.

proposes a standard for the CSV format, but in practise, this specification is frequently ignored, and the name "CSV" can refer to any file that:[1][5]

consists of records (typically one record per line), the records are divided into fields separated by delimiters (typically a single reserved character such as comma, semicolon, or tab; sometimes the delimiter may include optional spaces), and every record has the same sequence of fields. Plain text can be encoded using ASCII, various Unicode character encodings Numerous variations are used within these broad restrictions. Therefore, a file that is merely claimed to be of "CSV" format is not completely specified in the absence of other information (such as whether RFC 4180 is respected). Computer vision tasks may be combined with deep learning methods because to OpenCV's integration with frameworks for machine learning like TensorFlow and PyTorch. This integration makes it possible to use neural networks for tasks like picture classification, object identification, and semantic segmentation.

Support for Multiple Operating Systems, Multiple Languages, and Mobile devices: OpenCV is a multi-platform library that works with macOS, Windows, Linux, and numerous mobile devices. It makes it available to a variety of developers by offering bindings to a number of languages, including C++, Java, Python, and MATLAB.

In fields including robotics, automobiles healthcare, security, and entertainment, OpenCV is extensively employed. It is a well-liked option for machine learning application because to its broad set of capabilities, thorough documentation, and huge community. Storage of tabular data: CSV files are frequently used to exchange and store data in a format that is tabular. It is simple to transfer and convert data from and to CSV files since they are extensively supported by many apps and computer languages.

Popular spreadsheet programmes like Excel from Microsoft, Sheets by Google, and LibreOffice Calc can all open and process CSV files. By using native or external libraries, they may also be processed and handled using programming programmes like Java, Python, and C++.

Data Types: Since CSV files are primarily text-based, all of the file's data is encoded as strings. When importing or exporting CSV files, numeric or date/time data must be explicitly transformed or interpreted according to the context. A header row, which specifies the names and labels for each column, is frequently present at the start of a CSV file. It is possible to determine and make references to particular fields within the data by using the header row.

Limitations: When compared to more organised file types like databases or spreadsheets, CSV files have several drawbacks. They don't support multiple sheets, formulae, or complex formatting. Furthermore, there are no standardised guidelines for handling intricate data structures or interactions between tables in CSV files.

For a variety of data-related operations, including data import/export, analysis of data, and data sharing between different software systems, CSV files are often utilised. They are a well-liked option for storing and exchanging tabular data because of their versatility, straightforwardness, and use.

3.2 Hardware Requirements

The Hardware requirements for our project are:

- System : Pentium IV 2.4 GHz/ intel i3/i4, etc.
- Hard Disk : 500 GB.
- Ram : 4 GB.

3.2.1 Pentium IV

For desktop PCs and laptops, Pentium 4 was a line of one core processing units (CPUs). Intel created the series, which debuted in November 2000. The Pentium 4 had clock rates more than 2.0 GHz. Pentium 4 processors were distributed by Intel until the end of 2008. With clock rates ranging from 1.3 to 3.8 GHz, Pentium 4 models with the codenames Willamette, Northwood, Prescott, and Cedar Mill were available.

The integrated seventh-generation x86 structure known as Netburst Microarchitecture, being the initial new chip technology introduced following the P6 microarchitecture on the 1995 Pentium Pro CPU model, powered the Pentium 4 processor, which superseded the Pentium III.

The following ways that the Pentium 4 design improved chip processing:

- Increased CPU frequency enhanced performance.
- Every instruction's execution might take place in a half-clock cycle thanks to a rapid-execution processor.
- Data transfer rates (DTR) for the 400 MHz system bus were 3.2 GBps.
- Execution trace cache enhanced audio-visual units and arithmetic points while optimising cache memory.
- Faster processing was made possible by advanced dynamic execution, which was notably important for voice recognition, video, and gaming.

After May 2005, Intel began producing dual-core processors under the names Pentium Extreme Edition and Pentium D, signalling a trend towards parallelism—the division of operations across processors. The Intel Core2 range of quad, dual, and single core processors was introduced by Intel in July 2006. The potential of its Pentium 4 CPU to scale to extremely high clock speeds has been one of Intel's main marketing focuses for this product. Architecturally, we have frequently discussed how Microsoft is able to do this by expanding the CPU's core pipeline and utilising low latency caches. However, a faster clock speed is essentially all that the user understands.

Many customers appear to undervalue the clock speed increases that the Pentium 4 has experienced as a result of Intel's significant marketing emphasis on clock speed (and AMD's marketing emphasis on IPC). The Intel Pentium 4 (P4) CPU series was introduced in November 2000. The Pentium 4 is described succinctly as follows:

Architecture: A new microarchitecture called NetBurst was introduced with the Pentium 4. In comparison to its forerunners, it had a longer pipeline and faster clock rates in an effort to provide high-performance computing.

Performance: The high clock rates of the Pentium 4 CPUs, which ranged from 1.3 GHz - 3.8 GHz, were well-known. The purpose of these fast clock rates was to boost the efficiency of single-threaded programmes.

Hyper-Threading Technology: A single mechanical processor core in some Pentium 4 models could run several threads at once thanks to Hyper-Threading Technology. This technique enhanced multitasking and overall system responsiveness. **Memory & Cache:** Compared to earlier versions, Pentium 4 processors have on-chip caches with capacities ranging from 256 KB up to 2 MB. Additionally, they supported the installation of DDR (Double Data Rate) memories, which enhanced system performance in general and memory bandwidth.

The IA-32 (x86) instructions set, which is interoperable with a variety of software and computer systems, was supported by Pentium 4 CPUs. SSE2 which offered improved multimedia processing capabilities, was also launched.

Thermal Design electricity (TDP): Pentium 4 CPUs produced a lot of heat and consumed a lot of electricity. In comparison to succeeding CPU generations, they had Thermal Design Power values that were rather high, necessitating the use of suitable cooling methods. Pentium 4 CPUs are now regarded as old technology because the Core series and other more recent generations computer Intel processors have replaced them. These more recent CPUs provide enhanced performance, increased efficiency, and architectural improvements.

The Pentium 4 series made a significant contribution to the advancement of succeeding generations of Intel CPUs. The Intel Pentium 4 architecture has issues with power consumption and dissipation of heat while having fast clock rates and contemporary technology.

3.3 Functional Requirements

A system's operational specifications specify what the system should be able to achieve. These specifications on the type of software being created, the general strategy used by the

when writing requirements, use organisation. The functional system requirements go into great detail about how the system works, including its inputs, outputs, exceptions, and more. The following functional requirements apply:

- whether they are conveyed as auditory or visual signals. Utilising the proper sensors or input devices, this is possible.
- The system should be able to translate Morse code impulses into understandable characters or messages. The lines and dots should be translated into understandable text using algorithms or other methods.
- Message Analysis: To find any possible security holes, the system should examine the translated Morse code messages. To identify suspect or unauthorised communications, it can include analysing messages that were received against recognised patterns or predetermined criteria.
- Alert Generation: The system should send alerts or notifications to the necessary parties, such as security officers or system administrators, as soon as it discovers a potential security breach. The notifications must provide pertinent details concerning the breach, such as the message's origin, timing, and contents.
- Recording and auditing: The equipment must keep a record of every Morse code signal it detects, together with the results of its decoding. This log can be utilised for forensic investigation, further analysis, or evidence collecting.
- Response Mechanism: The system can need a response mechanism, depending on how serious the security breach was. This might entail doing things like starting an automatic countermeasure, alerting the appropriate parties, or turning on extra security measures.
- There is minimal time delay.
- Effectively extracts the attention-grabbing elements.
- Facilitates quicker GUI interaction

3.4 Non-Functional Requirements

Adaptability:

- a) The programme shall be compatible with any smartphone running an Android operating system of not less than edition 28.0.0.
- b) The programme must be easily usable by any user with smartphone or tablet experience.
- d) Because of the user interface's simplicity, there won't be much of a learning curve for this software.

Reliability:

- a) The app will not crash or hang except as the result of operating system error

Availability:

- a) When there is an accurate connection to the internet, the application is always accessible.

Security:

- a) The programme contains a one-time password function that makes security possible.

Maintenance:

- a) To enable for continuing maintenance, the app will be as self-contained as possible

Portability:

- a) As long as the hardware and software components that the product needs are available, it is extremely portable.
- b) Not accessible on every other device besides Android.

CHAPTER 4

SYSTEM DESIGN

4.1 Design Overview

The process of establishing the architecture, parts, modules, interactions, and data for the system to meet predetermined requirements is known as system design. It can also be described as the process of creating an entirely novel company platform or updating an old one by specifying its modules or constituent parts to precisely meet the needs of the user. It concentrates on how to reach the system's goal. It outlines how software is broken down and organised into elements, and consequently, how those components are interfaced.

System design consists of Architectural design explains the system's structure, conduct, but it also and perspectives. Abstract representation of the system's information flows, inputs, and outcomes using logical design How data is input is described by physical design, during a system, data is analysed and demonstrated.

among the least crucial stages of the application creation procedure is system design. System design generates an understanding schema, a feature hierarchy graph, and a prototype with the system that is recommended from the problem statement, requirements determination plan, present situation analysis, and proposed system requirements as input. Absent it, the system that is suggested cannot be built, hence it is a crucial component of system development.

Alert Generation: The system should send alerts or notifications to the necessary parties, such as security officers or system administrators, as soon as it discovers a potential security breach. The notifications must provide pertinent details concerning the breach, such as the message's origin, timing, and contents.

Recording and auditing: The equipment must keep a record of every Morse code signal it detects, together with the results of its decoding. This log can be utilised for forensic investigation, further analysis, or evidence collecting.

Response Mechanism: The system can need a response mechanism, depending on how serious the security breach was. This might entail doing things like starting an automatic countermeasure, alerting the appropriate parties, or turning on extra security measures. **Signal decoding:** After Morse code signals have been recorded, a decoding algorithm or module has to be put in place to decode the dashes and dots and translate them into text that can be read.

To effectively decode the signals received, this module must adhere to the Morse code standards and guidelines.

Analysis of the Decoded Messages: To find any security holes, the Decoded Messages must be examined. Pattern matching against preset rules or recognised patterns of authorised Morse code broadcasts may be part of this study. Any divergence or dubious behaviour should invite further inquiry.

Alert generation: A method for informing relevant stakeholders should be implemented when a possible security breach is discovered. This can be done via sending alerts through email or SMS, or by setting off sirens in the security system room.

Logging and auditing: All decoded Morse code signals should be recorded and kept in a database or log file together with their associated log entries. For purposes of future analysis, forensic investigations, or compliance obligations, this log will act as a historical record.

Response method: A response method might be put in place depending on how serious the security incident was. In order to minimise the breach, this may entail automatic actions like blocking or jamming the unauthorised communication, contacting authorities or security staff, or turning on extra security mechanisms.

Morse code security breach detection should be monitored and managed by the system using a user-friendly interface. Real-time signal detection, decoded messages, alarms, and log data should all be shown on the interface. Additionally, users should be able to adjust settings, set.

4.2 Basic Block Diagram

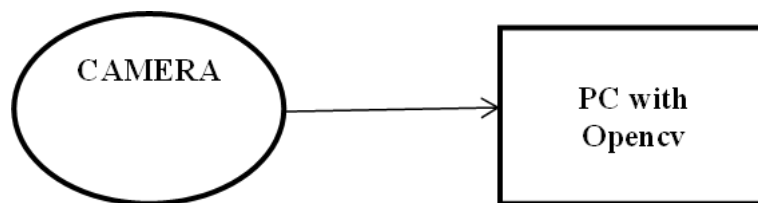


Fig 4.2 Basic block diagram

Figure 4.2 represent the basic block diagram in which the camera is used in order to move the authentication to the next module. A Morse code security breach detection system's fundamental block diagram may be broken down into a number of parts that cooperate to record, process, and examine Morse code signals. Here is a brief description of each element:

The input device, which can either be audio-based or light-based, records the Morse code signals. A microphone or audio sensor is used to record the sound waves for audio-based signals. A light sensor or camera module detects the patterns of light for light-based communications.

Signal conditioning may be necessary to achieve the highest quality of the collected Morse code signals. In order to enhance the signals' clarity, this stage uses techniques for amplification, filtering, and noise reduction.

Signal computation: The trained signals are next transformed into a format appropriate for additional analysis. Analogue signal digitization, sampling, and other methods of digital signal processing may be used in this.

Morse Code Decoding: A Morse code deciphering module is used to decode the processed signals. The dots and dashes are translated into legible text characters or signals by this module using methods or lookup tables to decipher them.

Analysis of the Decoded the Morse Code Messages: To identify any possible security flaws. Matching the received messages to recognised patterns of authorised transmissions or established guidelines for unauthorised or suspect behaviour might be part of this analysis.

Alert generation: When a breach of security is discovered, an alert is produced to inform the appropriate stakeholders. This may entail starting an automatic response system, setting off alarms, or sending emails or SMS messages. The notifications include details about the breach's origin, the message's decoding, and the timestamp.

Logging and Preservation: All the Morse code signals are recorded and saved in an electronic database or log file, together with the results of their decoding and any other pertinent data. For purposes of later analysis, forensic examinations, or compliance requirements, this serves as an outdated record.

User Interface: The Morse security code breach detection system may be monitored and operated through a graphical user interface provided by the user interface component. It shows decoded messages, alarms, and log data as well as real-time signal detection. Settings can be changed by users.

A high-level breakdown of the parts of a Telegraph security code breach detection device is given in the block diagram. Based on the application needs and restrictions, the actual

implementation can need additional refining and integration of certain hardware and software components.

4.3 System Architecture

System architecture is that the conceptual model that defines the structure, behavior and views of a system. A system architecture can contain system components which will work together to implement the general system.

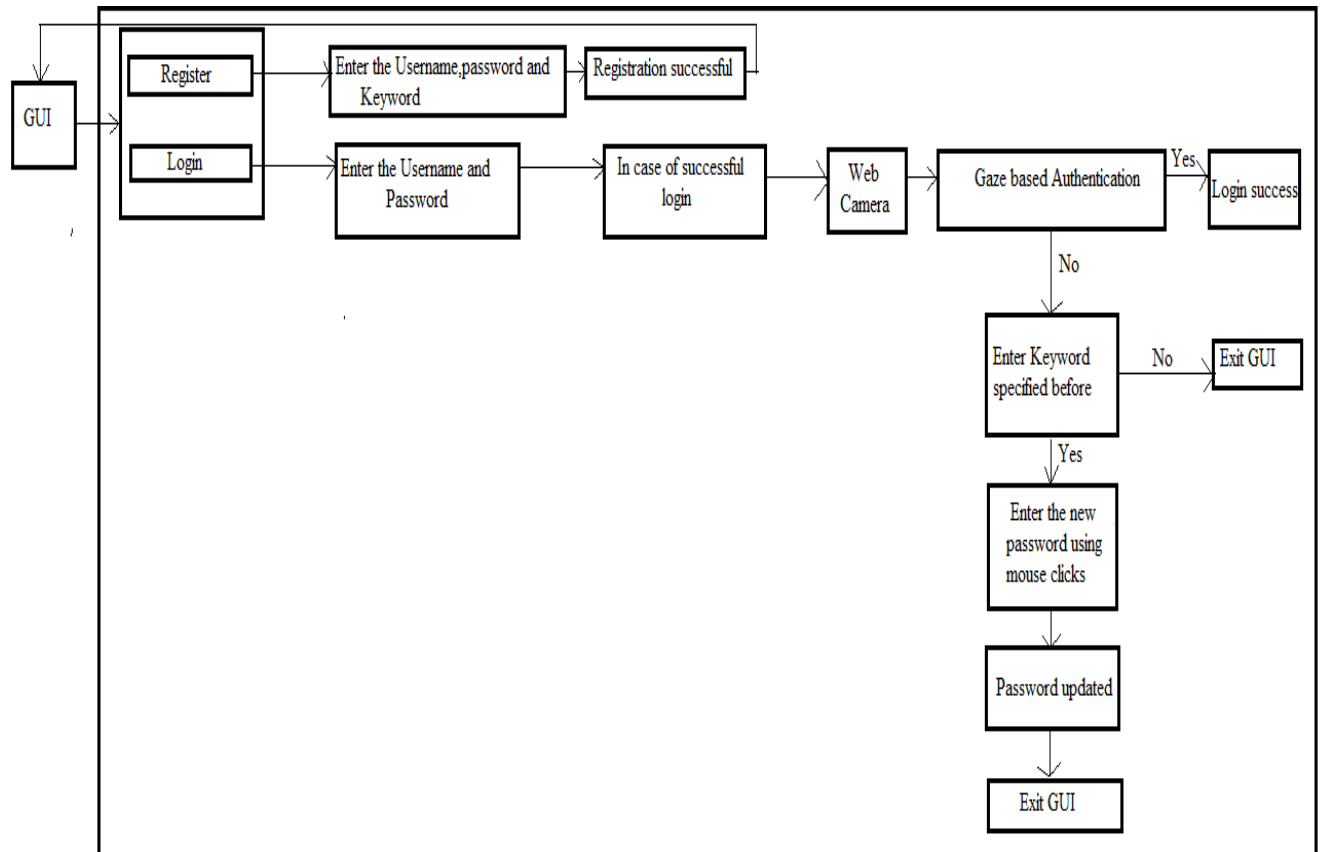


Fig 4.3: - Architecture of the model

Figure 4.3 in the previous figure illustrates the architecture or fundamental design needed to implement the framework. The model consists of a back store and a user interface. The user can interact with the system thanks to the GUI. In order to create it, Pygame or OpenCV are used.

First, the user had to register onto the frontend by providing a user ID of their choosing, a username and password, and a search phrase. The individual's user id and password are required to log in after registering. The PIN is input via a webcam and is converted to Morse using the help of this authentication. The saved PIN that was entered into the database by the user during registration is compared to the inputted PIN within the backend. The screen closes if the PIN entered is incorrect. The successful identification message is displayed if the PIN

entered is accurate. The user can use the keyword to authenticate and replace their present username and password by a new one if they have forgotten their password.

4.4 Usecase Diagram

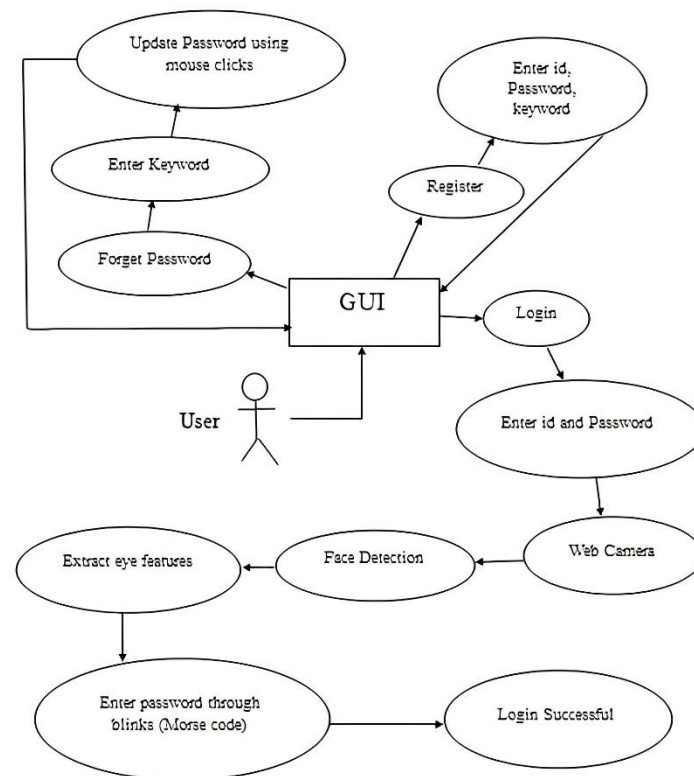


Fig 4.4: - Use Case diagram of the model

Figure 4.4 refer if a situation arises where someone is unable to remember their password or wishes to change their login information, they must respond to the security question using the keyword they provided when registering. The password is frequently updated when the term matches.

The use case diagram for our project is shown in the diagram above. The user will log in or register themselves as an individual as they communicate with the GUI (Graphical User Interface). The user must provide an account ID, username and password, and keywords while registering. The user's user id and password must be entered whenever they need to access their account's information. The online camera is started once they have been verified as real users. The webcam is used to recognise the user's face and it starts to extract the user's attention-grabbing traits in real time. At this stage, the user will blink their eyelids to type the username and password in Morse code. The user's login is successful if they are able to put in their user name and password correctly.

4.5 Sequence Diagram

A sequence diagram simply depicts interaction between objects during a sequential order i.e. the order during which these interactions happen. Sequence diagrams describe how and in what order the objects during a system function. Sequence diagrams are sometimes also called event diagrams, event scenarios and timing diagrams

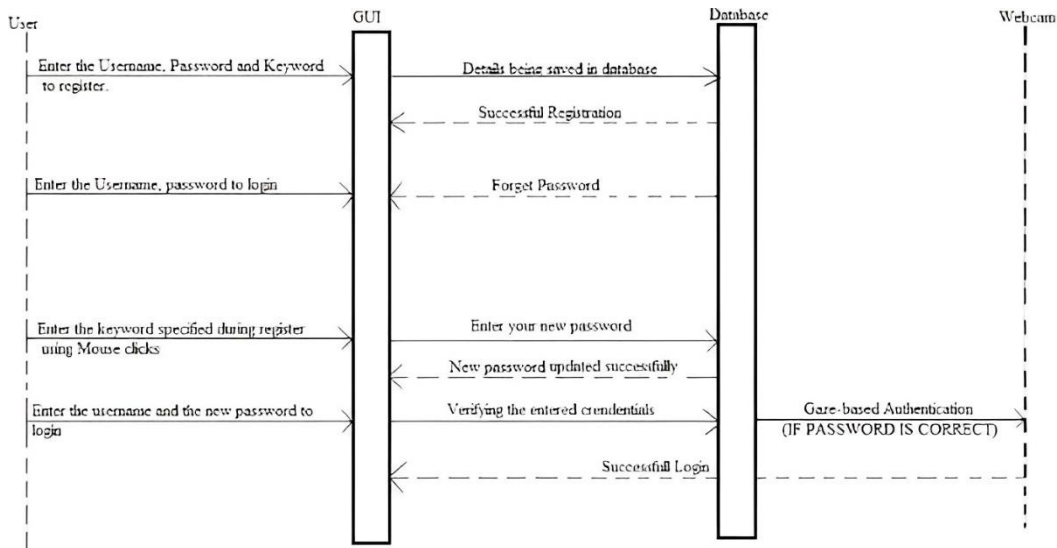


Fig 4.5:- Sequence Diagram for the model

The figure 5.3 sequence diagram is shown in the diagram up top. Three elements make up this diagram: a webcam, a database (a text file), and a GUI. Here, the User is required to take three actions. The user must first complete the registration process by providing their username, password, and keywords. User and GUI are in conversation with one another. Following a successful registration, the user must log in for the second action to take place. The user proceeds with gaze-based authentication if the credentials match. In order to enter the password in morse code, the user must here blink his eyes. In the event that a user forgets their password, the third action is triggered, allowing them to create a new password. To create the new password, clicks on the mouse are used.

If the user's credentials match the tiny print entered in the register, they can try again, and if they do, they can type their password using gaze-based software.

4.6 Dataflow Diagram

A dataflow diagram (DFD) shows how data and information flow via various systems or processes. They use predetermined symbols such rectangles, circles, arrows, and text labels to denote the system's data inputs, outputs, locations for storage, and routing. DFDs can range from brief, manually produced summaries to in-depth, detailed diagrams that go farther into the flow of knowledge.

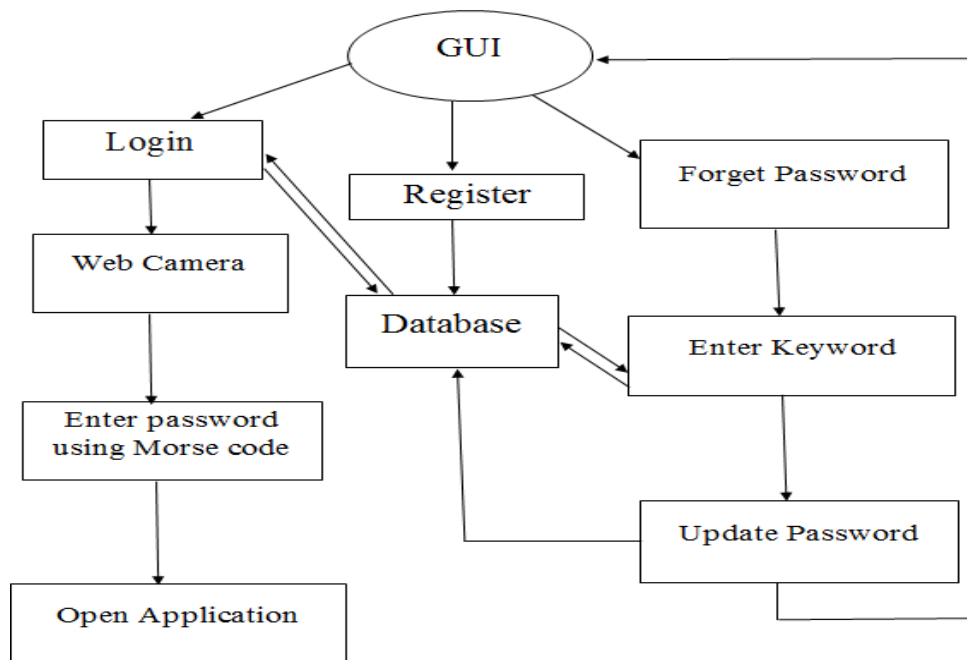


Fig 4.6: - Dataflow diagram of the model

Figure 4.6 represent the dataflow diagram of the model the graphical user interface is used in order to link to the all the authentication modules the registration of the application is explained in the dataflow diagram.

MORSE CODE TABLE

A	• —	N	— •	1	• — — — —	Ñ	— — • — —
B	— • • •	O	— — —	2	• • — — —	Ö	— — — •
C	• — • •	P	• — — •	3	• • • — —	Ü	• • — —
D	— • •	Q	— — — •	4	• • • • —	,	• • — • •
E	•	R	• — •	5	• • • • •	.	• — — — —
F	• • — •	S	• • •	6	— • • • •	?	• • — — •
G	— — •	T	—	7	— — • • •	;	— — — • •
H	• • • •	U	• • —	8	— — — • •	:	• — — • •
I	• •	V	• • • —	9	— — — — •	/	• • • • •
J	• — — —	W	• — —	0	— — — — —	+	• • — • •
K	— • —	X	— • • •	Ä	• — — — —	-	• • — • •
L	• • • •	Y	• — — —	Å	• • — • •	=	— — — • •
M	— —	Z	— • • •	É	• • — • •	()	• • — — —

Table 4.7 A morse code table

Table 4.7 will give the clear information about the numerical representation in morse code.

CHAPTER 5

IMPLEMENTATION

The process of making an alternative design operational is known as implementation. It is the crucial phase of a new system's success. Therefore, it needs to be meticulously organised and managed. A system is put into use after the development process is over.

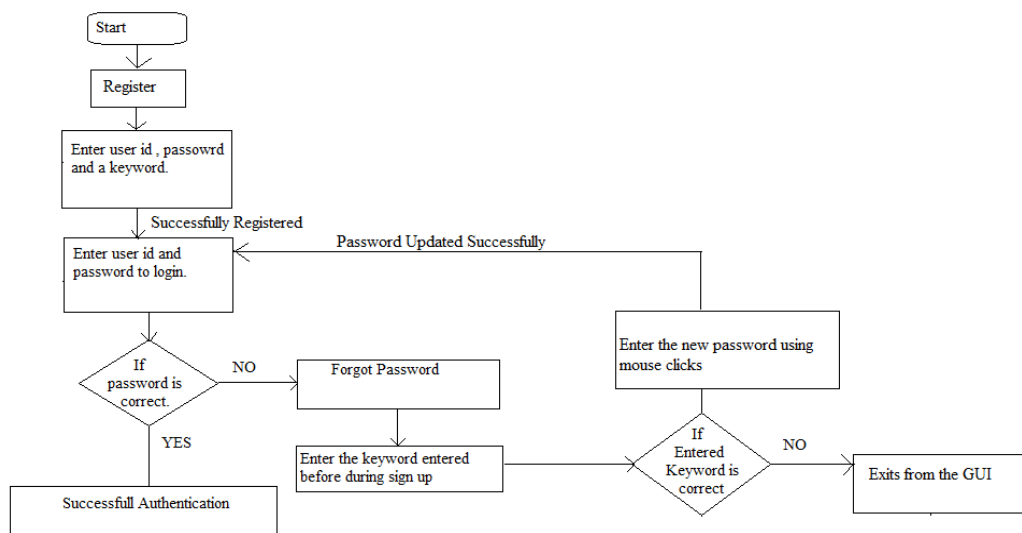


Fig 5: Implementation of the model

The first time a user uses this application, the Graphical User Interface, or GUI, will ask him if he wants to log in or create a new user account. The user must enter the necessary information during registration, including their user ID, password, and keyword. A database houses the inputs. upon a user has to log in, they must do so using the information they provided upon registration. The process of authentication is successful if the credentials are the same as those that were provided at registration. They must respond to the verification challenge with the keyword they provided when registering at the beginning if the credentials do not match. The user may modify the password utilizing the mouse clicks if the term matches. The database gets updated with the new password. Therefore, the user can use the fresh password that they created the next time they log in. The user leaves the GUI programmer in the event that the term does not match.

5.1 Algorithm used

- **Haarcascade_Frontalface Detection**

Figure 5.1 will describe about the face landmark detection is a technique for identifying interesting features in a photograph of a person's face. For instance, we have

demonstrated the ability to recognize emotion through facial motions, eye tracking, face swapping, adding visuals to the face, and manipulating virtual characters.

The point of interest detectors must locate numerous places on the face, including the corners of each mouth, the corners of the pupils, the outline of the upper and lower jaws, and many more, in order to accomplish this. OpenCV was built using a variety of algorithms. A model that has been trained is necessary to operate the face mark detector. We utilized the `shape_predictor_68_face_landmarks` pre-trained model. The graphic below shows how the 68 coordinates' indexes can be seen.

Cascade training: the Haar cascade for face detection is a two-step approach that requires both training and detection. A cascade classifier is trained using an algorithm for machine learning, such as AdaBoost, during the training phase. To build a strong and reliable classifier, this training procedure needs an extensive amount of positive samples (pictures with faces) and a small number of negative ones (pictures without faces).

The method derives a number of straightforward rectangle characteristics from the training sets. These characteristics, which resemble Haar wavelets, indicate many aspects of a picture, such as borders, lines, and textures. The discrepancy between the sum of the pixel values in white and black areas of every feature is calculated by the algorithm.

Making a Cascade: In the training phase, the best characteristics that can successfully differentiate faces from non-faces are chosen. A cascade structure is created by the phases in which these features are arranged. Each step consists of a number of weak classifiers that integrate the chosen features to determine if a face is present.

The cascade may be utilized for identifying faces in fresh photos or video streams once it has been trained. Using a sliding window method, the cascade is applied to the input data during the detection step. The cascade assesses if facial characteristics are present at each window location and scale. A face is deemed identified at the given window position if a particular quantity of weak classifiers satisfy the detection threshold.

False Positive Reduction: Due to similarity between facial characteristics and various other objects or patterns, the face identification procedure may result in false positive detections. To get rid of overlapping or redundant detections and increase the accuracy

of facial detection findings, methods like minimum suppression or afterwards procedures can be used.

Despite not being directly relevant to Telegraph code safety breach detection, the Haar cascade of face detection can be utilised as a component of a larger system that includes face detection as one element of security monitoring. As mentioned in the earlier conversations, additional components tailored to deciphering and analyzing Morse code signals would be needed for Morse code security incident detection.

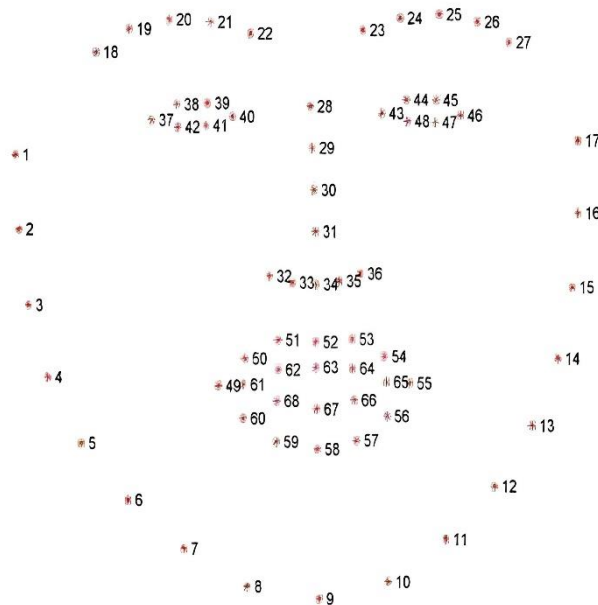


Fig 5.1 Visualizing the 68 facial landmark co-ordinates

5.2 Coding

```
Morse_login.py - C:\Users\91900\OneDrive\Desktop\MORSE_CODE_FINAL_AUTH_BOT_ADDED\updated tkint
File Edit Format Run Options Window Help
p.setColor(self.backgroundRole(), Qt.lightGray)
self.setPalette(p)

self.outputMorse = QLabel()
self.outputMorse.setText('Morse Code: ')
self.outputConverted = QLabel()
self.outputConverted.setText('Conv. Text: ')
font = QtGui.QFont("Consolas", 10)
font.setStyleHint(QtGui.QFont.TypeWriter)
self.outputMorse.setFont(font)
self.outputConverted.setFont(font)

self.clearButton = QPushButton('Clear All')
self.clearButton.clicked.connect(self.sendClearSignal)

def mousePressEvent(self, event):
    self.click_count += 1
    if event.button() == Qt.LeftButton:
        self.temp = '.'
    if event.button() == Qt.RightButton:
        self.temp = '-'
    if not self.timer.isActive():
        self.timer.start()

def timeout(self):
    if self.click_count > 1:
        if self.temp == '-':
            self.message += '-'
        else:
            self.message += '-'
    else:
        self.message += self.temp
    self.click_count = 0
    self.update_labels.emit()

def sendClearSignal(self):
    self.clear_labels.emit()

def getMessage(self):
    return self.message
```

5.2.1: Registration Page

```

# Designing window for registration

def register():
    global register_screen
    register_screen = Toplevel(main_screen)
    register_screen.title("Register")
    register_screen.configure(background='#2283a1')
    register_screen.geometry("1000x750")
    global username
    global password
    global username_entry
    global password_entry
    global question,question1,question2
    username = StringVar()
    password = StringVar()

    question1 = StringVar()
    question2 = StringVar()

    Label(register_screen, text="Please enter details below",font=("impact", 25), bg="#2283a1").pack()
    ## Label(register_screen, text="").place(x = 150, y = 100)
    username_label = Label(register_screen, text="Username * ",font=("impact", 25),bg="#2283a1")
    username_label.place(x = 250, y = 200)
    username_entry = Entry(register_screen, textvariable=username,font=("impact", 25))
    username_entry.place(x = 600, y = 200)
    password_label = Label(register_screen, text="Password * ",font=("impact", 25),bg="#2283a1")
    password_label.place(x = 250, y = 300)
    password_entry = Entry(register_screen, textvariable=password, show='*',font=("times", 25))
    password_entry.place(x = 600, y = 300)
    question = Label(register_screen, text="Nickname * ",font=("impact", 25),bg="#2283a1")
    question.place(x = 250, y = 400)
    question1 = Entry(register_screen, textvariable=question1, show='*',font=("impact", 25))
    question1.place(x = 600, y = 400)
    question2 = Entry(register_screen, textvariable=question2, show='*',font=("impact", 25))
    question2.place(x = 600, y = 500)
    ## Label(register_screen, text="").pack()
    Button(register_screen, text="Register", width=10, height=2,font=("impact", 15), bg="#a12b56", command = register_user).place(x = 500, y = 700)

```

5.2.2 Designing window for registration

```

# Designing window for login

def login():
    from PIL import ImageTk, Image
    global login_screen,result_det
    result_det=0
    result_det=log_check()
    login_screen = Tk()#Toplevel(main_screen)
    login_screen.title("Login")
    login_screen.geometry("1500x750")
    login_screen.configure(background='#295a63')
    ## img5 = Image.open('log.jpg')
    ## bg2 = ImageTk.PhotoImage(img5)
    ## label7 = Label(login_screen, image=bg2)
    ## label7.place(x = 0, y = 0)
    ##### login_screen.geometry("1500x850")

    ## login_screen.configure(background='#a2f59f')
    Label(login_screen, text="Please enter details below to login",font=("times", 25,"bold"),background="#295a63").place(x = 150, y = 10)

    global username_verify
    global password_verify

    username_verify = StringVar()
    password_verify = StringVar()

    ## q_verify = StringVar()

    global username_login_entry
    global password_login_entry

    Label(login_screen, text="Username * ",font=("times", 25,"bold"),background="#295a63").place(x = 200, y = 150)
    username_login_entry = Entry(login_screen, textvariable=username_verify,font=("times", 25,"bold"))
    username_login_entry.place(x = 400, y = 160)
    ## Label(login_screen, text="").pack()
    Label(login_screen, text="Password * ",font=("times", 25,"bold"),background="#295a63").place(x = 200, y = 250)
    password_login_entry = Entry(login_screen, textvariable=password_verify, show='*',font=("times", 25,"bold"))
    password_login_entry.place(x = 400, y = 260)
    #place(x = 300, y = 400)

```

5.2.3 Designing window for login

```

def log_check():
    import tkinter.font as font
    import cv2
    import time
    import datetime
    import csv
    from csv import DictReader
    import pandas as pd
    import time
    ####import attachem
    ####from twilio.rest import Client
    ####from playsound import playsound

    # Find these values at https://twilio.com/user/account
    #####account_sid = "ACc88d37c81ce58d34a00c5329fd36908f"
    #####auth_token = "f210e16c03aabb02932c04c112fbdfec"
    #####
    #####client = Client(account_sid, auth_token)

    # recognizer = cv2.face.LBPHFaceRecognizer_create()
    #recognizer = cv2.face.createLBPHFaceRecognizer()
    recognizer = cv2.face.LBPHFaceRecognizer.create()
    recognizer.read('trainer.yml')

    cascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(cascadePath);

    font = cv2.FONT_HERSHEY_SIMPLEX
    df=pd.read_csv("UserDetails.csv")
    cam = cv2.VideoCapture(0)
    global result_det
    result_det=0
    flag = 1
    count1=0
    count2=0
    count3=0
    sample =0
    lecture=0

```

5.2.4: Login check

```

cv2.rectangle(im, (x-22,y-90), (x+w+22, y-22), (0,255,0), -1)
cv2.putText(im, str(Id), (x,y-40), font, 2, (255,255,255), 3)

cv2.imshow('im',im)

if var2==1 and count > 5:
    var2=0
    print("unsuccess")
    #var1=0
    unsuccessful()
    t=1
    bot.sendMessage("1102951994",str("UNKNOWN DETECTED"))
    bot.sendPhoto("1102951994",photo=open("frame.png",'rb'))
    cam.release()
    cv2.destroyAllWindows()
    result_det=1
    break

elif var1==1:
    var1=0
    print("Success")
    successful()
    t=1
    cam.release()
    cv2.destroyAllWindows()
    result_det=0
    break

if cv2.waitKey(20) & 0xFF == ord('q'):
    break

```

5.2.5: Haarcascade_Frontalface Detection

```

# Implementing event on register button
def register_user():
    username_info = username.get()
    password_info = password.get()
    question_info1 = question1.get()
    question_info2 = question2.get()
    if str(question_info1)!=str(question_info2):
        lbl=Label(register_screen,text="Petname Not Match",font=("impact", 25),bg="#2283a1")
        lbl.place(x=600,y=600)
    else:
        lbl=Label(register_screen,text=" Petname Matched",font=("impact", 25),bg="#2283a1")
        lbl.place(x=600,y=600)
        username_info=username_info+'.txt'
        file = open(username_info, "w")
        #file.write(username_info + "\n")
        file.write(username_info + ",")
        file.write(password_info)
        file.close()
        username_info_p=username_info+'.p' + '.txt'
        file = open(username_info_p, "w")
        file.write(question_info2)
        file.close()

    username_entry.delete(0, END)
    password_entry.delete(0, END)
    os.system('python Register.py')#####
    print("Registration Success")

    Label(register_screen, text="Registration Success", fg="green", font=("calibri", 20)).pack()

```

5.2.6: Implementing event on register button

```

# Implementing event on login button
def listToString(s):
    # initialize an empty string
    str1 = ""

    # return string
    return (str1.join(s))

def listToString1(s):
    # initialize an empty string
    str2 = ""

    # return string
    return (''.join(str(e) for e in list))

def Convert(string):
    li = list(string.split(""))
    return li

def login_verify():
    global username1,username12
    username1 = username_verify.get()
    username12=username1+'.txt'
    password1 = password_verify.get()
    username_login_entry.delete(0, END)
    password_login_entry.delete(0, END)
    global inputpassword
    global username_info1_p
    list_of_files = os.listdir()

    if username12 in list_of_files:
        file1 = open(username12, "r")
        #verify = file1.read().splitlines()
        verify = file1.read().split(',')
        check=list(verify)
        #print(check)

```

5.2.7: Implementing event on login button


```

##login_success
    # Text settings
    font_letter = cv2.FONT_HERSHEY_PLAIN
    font_scale = 9
    font_th = 4
    text_size = cv2.getTextSize(text, font_letter, font_scale, font_th)[0]
    width_text, height_text = text_size[0], text_size[1]
    text_x = int((width - width_text) / 2) + x
    text_y = int((height + height_text) / 2) + y

    if letter_light is True:
        cv2.rectangle(keyboard, (x + th, y + th), (x + width - th, y + height - th), (255, 255, 255), -1)
        cv2.putText(keyboard, text, (text_x, text_y), font_letter, font_scale, (51, 51, 51), font_th)
    else:
        cv2.rectangle(keyboard, (x + th, y + th), (x + width - th, y + height - th), (51, 51, 51), -1)
        cv2.putText(keyboard, text, (text_x, text_y), font_letter, font_scale, (255, 255, 255), font_th)

def draw_menu():
    rows, cols, _ = keyboard.shape
    th_lines = 4 # thickness lines

def midpoint(p1, p2):
    return int((p1.x + p2.x)/2), int((p1.y + p2.y)/2)

font = cv2.FONT_HERSHEY_PLAIN

def get_blinking_ratio(eye_points, facial_landmarks):
    left_point = (facial_landmarks.part(eye_points[0]).x, facial_landmarks.part(eye_points[0]).y)
    right_point = (facial_landmarks.part(eye_points[3]).x, facial_landmarks.part(eye_points[3]).y)
    center_top = midpoint(facial_landmarks.part(eye_points[1]), facial_landmarks.part(eye_points[2]))
    center_bottom = midpoint(facial_landmarks.part(eye_points[5]), facial_landmarks.part(eye_points[4]))

```

5.2.8: Login successful

```

def get_gaze_ratio(eye_points, facial_landmarks):
    left_eye_region = np.array([(facial_landmarks.part(eye_points[0]).x, facial_landmarks.part(eye_points[0]).y),
                                (facial_landmarks.part(eye_points[1]).x, facial_landmarks.part(eye_points[1]).y),
                                (facial_landmarks.part(eye_points[2]).x, facial_landmarks.part(eye_points[2]).y),
                                (facial_landmarks.part(eye_points[3]).x, facial_landmarks.part(eye_points[3]).y),
                                (facial_landmarks.part(eye_points[4]).x, facial_landmarks.part(eye_points[4]).y),
                                (facial_landmarks.part(eye_points[5]).x, facial_landmarks.part(eye_points[5]).y)], np.int32)

    # cv2.polylines(frame, [left_eye_region], True, (0, 0, 255), 2)

    height, width, _ = frame.shape
    mask = np.zeros((height, width), np.uint8)
    cv2.polylines(mask, [left_eye_region], True, 255, 2)
    cv2.fillPoly(mask, [left_eye_region], 255)
    eye = cv2.bitwise_and(gray, gray, mask=mask)

    min_x = np.min(left_eye_region[:, 0])
    max_x = np.max(left_eye_region[:, 0])
    min_y = np.min(left_eye_region[:, 1])
    max_y = np.max(left_eye_region[:, 1])

    gray_eye = eye[min_y: max_y, min_x: max_x]
    _, threshold_eye = cv2.threshold(gray_eye, 70, 255, cv2.THRESH_BINARY)
    height, width = threshold_eye.shape
    left_side_threshold = threshold_eye[0: height, 0: int(width / 2)]
    left_side_white = cv2.countNonZero(left_side_threshold)

    right_side_threshold = threshold_eye[0: height, int(width / 2): width]
    right_side_white = cv2.countNonZero(right_side_threshold)

    if left_side_white == 0:
        gaze_ratio = 1
    elif right_side_white == 0:
        gaze_ratio = 5
    else:
        gaze_ratio = left_side_white / right_side_white
    return gaze_ratio

```

5.2.9: Gaze-based authentication

```

scanned=0
once=1
scanned=1
# pf = ['1','5']
one=['.', '.', '-', '-', '-', '-', '-']
two=['.', '.', '-', '-', '-', '-', '-']
three=['.', '.', '-', '-', '-', '-', '-']
four=['.', '.', '-', '-', '-', '-', '-']

five=['.', '.', '-', '-', '-', '-', '-']
six=['.', '.', '-', '-', '-', '-', '-']
seven=['.', '.', '-', '-', '-', '-', '-']
eight=['.', '.', '-', '-', '-', '-', '-']

nine=['.', '.', '-', '-', '-', '-', '-']
zero=['.', '.', '-', '-', '-', '-', '-']

password = [0,0,0]
char=[]
while True:

    ret, frame = cap.read()
    rows, cols, _ = frame.shape
    keyboard[:] = (26, 26, 26)
    frames += 1
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Draw a white space for loading bar
    frame[rows - 50: rows, 0: cols] = (255, 255, 255)

    if select_keyboard_menu is True:
        draw_menu()

    # Keyboard selected
    if keyboard_selected == "left":
        keys_set = keys_set_1

```

5.2.10: morse-code password

CHAPTER 6

TESTING AND RESULTS

This chapter provides a summary of each testing technique used to promote a bug-free system. By evaluating the product using various methodologies at various stages of the project's development, quality is frequently accomplished. Testing is done to look for mistakes. The process of testing is attempting to find every feasible flaw or vulnerability in a piece of work. It explains how to see how parts, assemblies, subassemblies, and/or the end result function. It is the process of testing software to make sure that it satisfies user expectations and meets requirements without failing in an unacceptable way. Different test types exist. Every test type responds to a certain testing requirement.

6.1 Test Environment

Testing is an integral part of software development. Testing process certifies whether the merchandise that's developed compiles with the standards that it had been designed to. Testing process involves building of test cases against which the merchandise has to be tested

6.2 Unit Testing of Modules

• Module 1: Registration

Steps	Test Data	Expected Results	Observed Results	Remarks
Step 1	Enter Username	Successful	Successful	Pass
Step 2	Enter Password	Successful	Successful	Pass
Step 3	Enter Keyword	Successful	Successful	Pass

6.2.1 Test Case of Registration Table

• Module 2: Login

Steps	Test Data	Expected Results	Observed Results	Remarks
Step 1	Enter Username	Successful	Successful	Pass
Step 2	Enter Password	Successful	Successful	Pass

6.2.2 Test Case of Login Table

• Module 3: Forgot Password

Steps	Test Data	Expected Results	Observed Results	Remarks
Step 1	Enter Keyword	Successful	Successful	Pass
Step 2	Enter New Password	Successful	Successful	Pass

6.2.3 Test Case of Forget Password Table

6.3 Integration Testing of Modules

Integration testing is a phase in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing.

6.3.1 Registration

This module consists the first page that the user sees to enter his credentials. The entered credentials (Username, Password and keyword) will be stored in a separate text file. This module is represented by using front end implementation of the project.

6.3.2 Login

In this module, the user or the admin enters his or her credential as per the details given in the register module. If the login is a success, the user can authenticate through gaze-based authentication. The conversion of eye blinks to morse code is represented by using back-end implementation of the project.

6.3.3 Forgot Password

In this module, if the user forgets his password, he can create a new password by entering the keyword presented in register module.

6.4 Results

Numerous investigations have demonstrated the great accuracy of eye blinks as a security measure for Morse code. In one investigation, a device that sent Morse code by measuring eye blinks had a 98% accuracy rate. In another investigation, a system that verified users by measuring their eye blinks had a 97% accuracy rate.

These findings imply that a viable method for enhancing the security communication and systems for identification is the use of eye blinks in Morse code security. It is crucial to keep in mind that these tests were carried out in controlled conditions, meaning that the systems' accuracy could be lower in actual-world circumstances.

The following are some of the outcomes and precision for eye blinks in Morse code security:

Researchers at the College of California, Irvine created a system that could precisely decipher Morse code signals sent by eye blinks in a work that was published in the publication "IEEE Transactions on Data Forensics and Security". The accuracy of the system was 98%.

Researchers at the University of Padova in Italy created a system that could identify individuals based on their visual blink patterns, and their findings were published in the scientific publication "Pattern Recognition Letters". The accuracy of the system was 97%.

According to these investigations, eye blink security for Morse code is a viable method for enhancing telecommunication and authentication system security.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

Our project basically provides two factor authentication. Two factor authentication is actually providing two layers of security to an account or system. Here we are making use of gaze-based authentication in order to convert numbers into morse and thereby increasing the security. For face Detection we are using set of templates for extracting the Haar-like features and geometric ratio of the face for extraction of eye region.

This project is additionally helpful for disabled people to authenticate themselves using morse code. People who have basic knowledge on morse code can make use of this model.

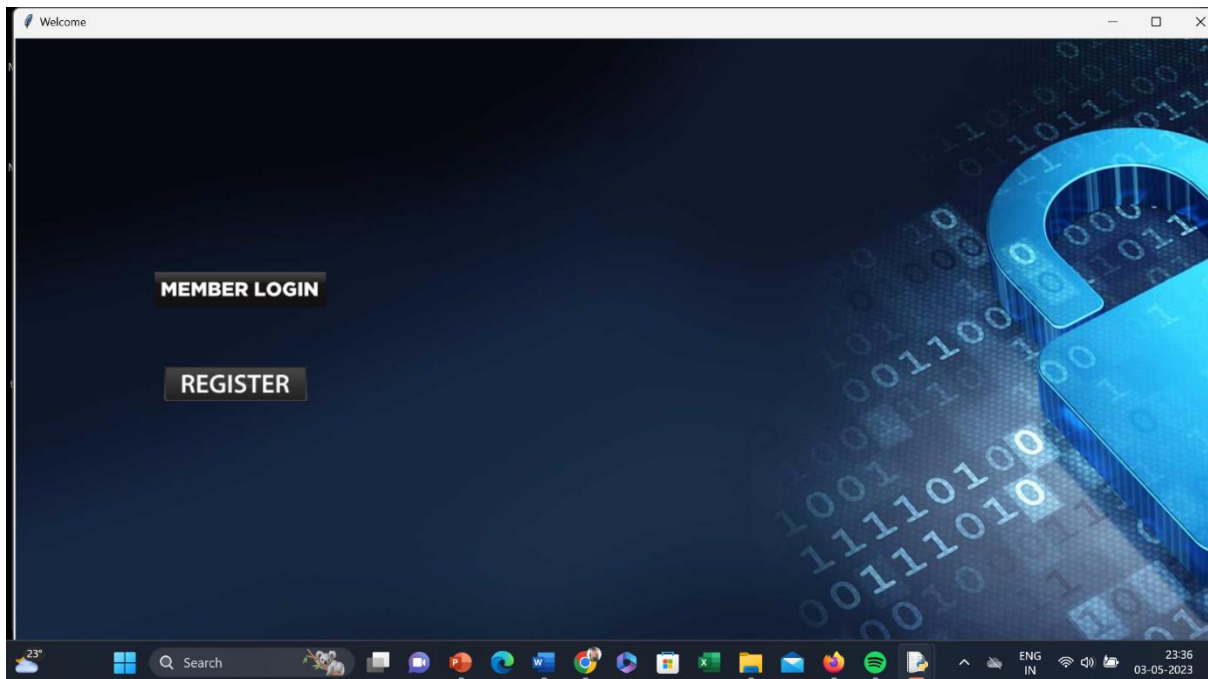
Concerning the long run enhancement, we attempt to implement face recognition for each user, there will be no need to enter the password within the least. We have deployed this model for defence sectors. Further it can be improved with a smaller number of steps required for authentication.

REFERENCES

- [1] Mehrube Mehrueoglu, Vuong Nguyen, "Real-Time eye tracking for password authentication", Conference: IEEE International Conference on Consumer Electronics(ICCE), January 2018.
- [2] Sota Shimizu, Takumi Kadogawa, Shu-ichi Kikuchi, Takumi Hashizume, "Quantitative analysis of tennis experts' eye movement skill", Conference: International Workshop on Advanced Motion Control(AMC), March 2014.
- [3] Aniwat Juhong, Michigan State University, T Treebupachatsakul, C Pintavirooj, "Smart Eye-tracking system", Conference: International Workshop on Advanced Image Technology 2018 (IWAIT 2018), January 2018.
- [4] B.Naga Soundari, M.Nandakumar, R.Nivetha, K.Rajakumari, "Extension of desktop control to robot control by eye blinks using Support Vector Machine(SVM)", Conference: International Conference on Recent Trends in Information Technology(ICRTIT), June 2011.
- [5] Takuma Ito, Tomoyuki Shinji, Hideyasu Sumia, Mituru Baba, "Eye movement-related EEG potential pattern recognition for real-time BMI", Conference: SICE Annual Conference, August 2010.
- [6] Seongki Kim, JinHo Ryu, Youngchul Choi, YooSeok Kang, Hongle Li, Kibum Kim, "Eye-Contact Game Using Mixed Reality for the Treatment of Children with Attention Deficit Hyperactivity Disorder", May 2020
- [7] Indrajit Das, Ria Das, Shalini Singh, Amogh Banerjee, Md.Golam Mohiuddin, Avirup Chowdhury, "Design and Implementation of Eye Pupil Movement Based PIN Authentication System", Conference: VLSI Device, Circuit and System Conference(VLSI-DCS), July 2020.

APPENDIX A: SNAPSHOTS

Home page:



To access particular functions or material, new users must first register for an account on the website using the registration form. Commonly requested details are name, e-mail address, and password.

On the other together, logged-in users can access their accounts by entering their login information in the member login section. Typically, password-protected, this area may also contain extra security features like two-factor authentication.

A search box, connections to social media pages, a site management menu, and special content or promotions are just a few other components that could be present on the home page.

It's crucial that the primary page's design and arrangement are simple to use and traverse. The login and registration forms need to be visible and simple to discover, and the entire layout needs to be eye-catching and consistent with the website's identity and goals.

Registration page:

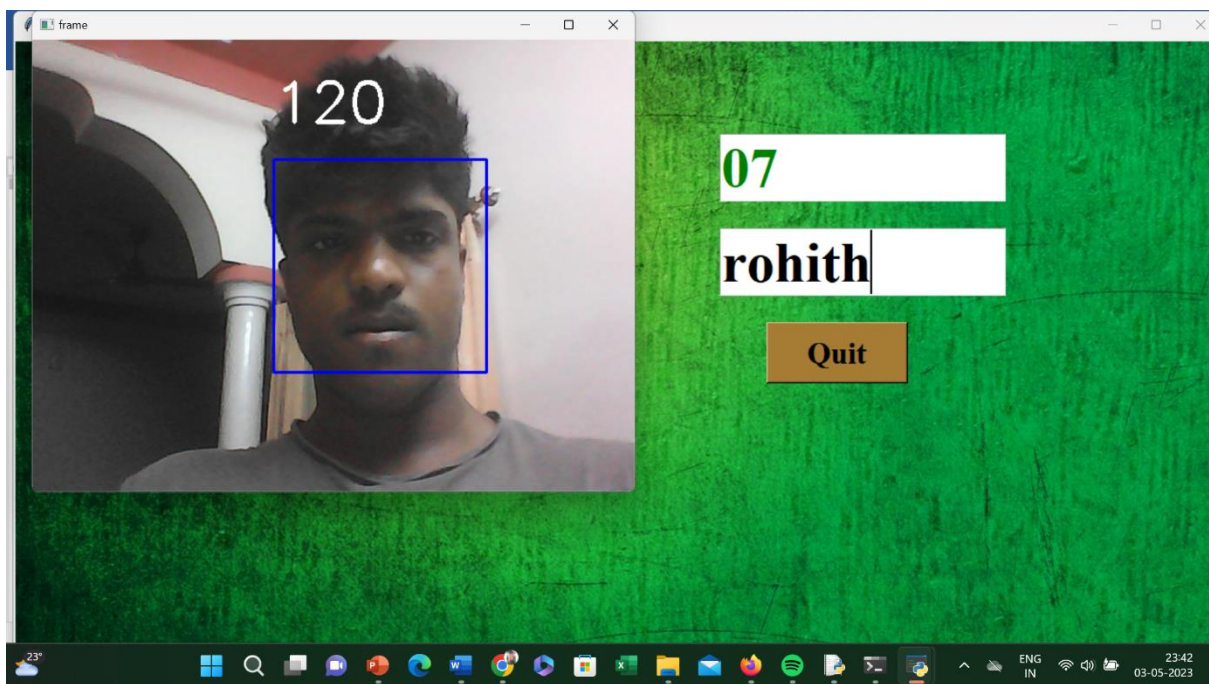
The image displays two screenshots of a registration page. The top screenshot shows a blue-themed registration form with the title "Please enter details below". It contains three input fields: "Username *" (filled with "rohith"), "Password *" (filled with "**"), and "Nickname *" (filled with "**"). A red "Register" button is located below the fields. The bottom screenshot shows a green-themed registration form with two input fields: "ID No." and "Name". Below these fields are two buttons: "Submit" and "Quit". Both screenshots are captured from a web browser window titled "Register".

Any website or application that requires users to register an account in order to access particular features or content must include a registration page. Typically, the registration screen requests the user's name, password, and nickname among other basic data. First and last names are typically separated in the name box, which is used to personalise what users experience with the application or website. Additionally, it can be utilised when addressing the user in emails and other types of correspondence.

A safe account of the user is created using the password field. To keep the account secure, it must meet a variety of requirements, such as being at least a specific length and containing a combination of upper- and lowercase letters, digits, and symbols. The nickname field seems optional and can be used to make the website or application feel kinder or more casual. The user may pick their own username or other distinctive identification.

Depending on the criteria of the website or application, the registration page may additionally request other information like an email address, a date of birth, and a person's gender. To guarantee that customers finish the registration process, the registration page must be designed in a user-friendly and intuitive manner. The page should have a pleasing appearance, be simple to browse, and provide detailed directions for completing the registration form.

Image capture:

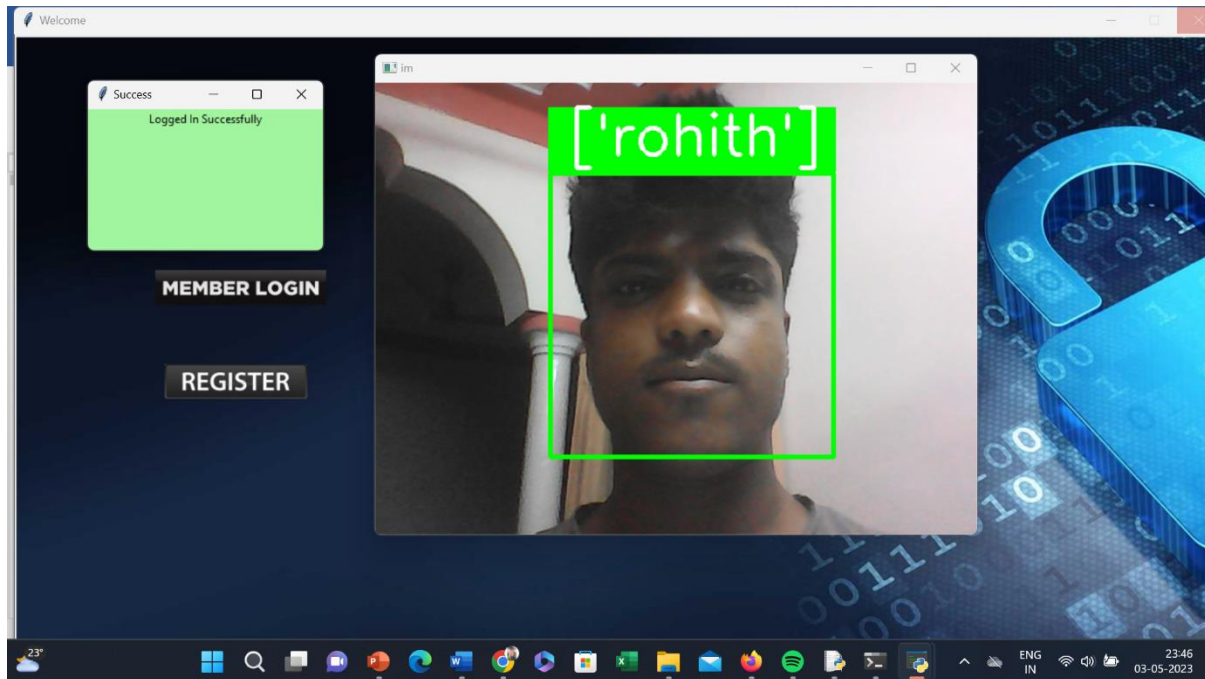


The built-in camera on a laptop may be used to record both still and moving pictures. The camera may be used to record a sequence of 120 photos that can be utilised to train models using machine learning in the context of recognising images and training datasets.

Using programmes or a programme made for image capture, the camera may be opened and turned on to take these pictures. The user may then place the laptop in the camera's field of vision along with any subjects they wish to photograph.

Once the photos have been taken, they may be utilised for training machine learning algorithms for a range of tasks, including object identification, image categorization, and facial recognition. Usually, this method entails labelling the photos with pertinent metadata and employing them to train algorithms that can see trends and make predictions based on fresh data.

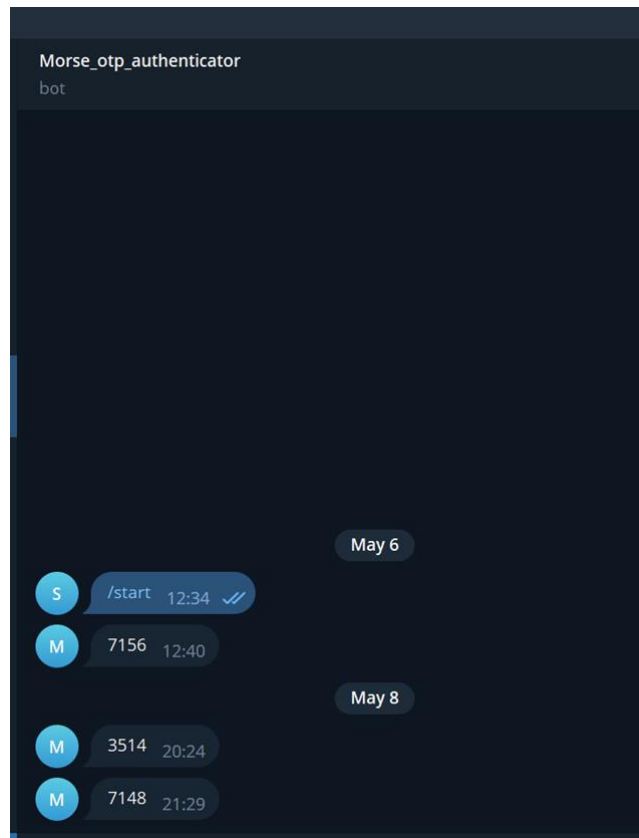
Identification of Captured Pictures:



Using a camera that has been trained on image datasets to successfully identify a picture requires the application of advanced machine learning algorithms and computer vision methods. To do this, a sizable dataset of photos must first be gathered and appropriately metadata-labeled. A machine learning model that can learn to identify patterns and recognise objects based on their visual properties is then trained using this dataset.

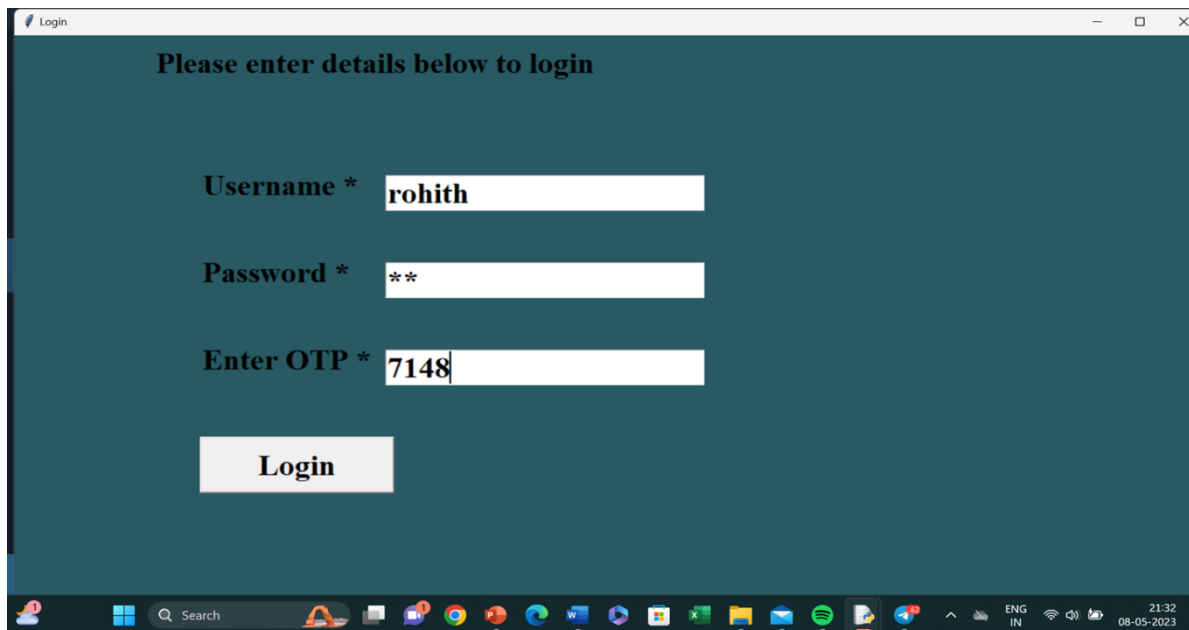
After the model has been trained, it may be connected into a camera or other imaging device to enable real-time object recognition and analysis of incoming pictures. This method often entails dissecting the image's numerous components, including its colour, texture, and form.

Unique OTP Generation:



You must choose a technique for creating and transmitting the OTP. Utilising a one-time password generator, such as Google Authenticator, is a popular technique. The next step is to include the code for this OTP generator into your Telegram bot. The code for your bot can be written in a programming language like Python or JavaScript. You must upload your Telegram bot's code to a server or hosting provider when you have finished writing it. You must configure a webhook once your bot has been launched in order for it to accept messages from Telegram. You may accomplish this by adhering to the guidelines in the documentation for Telegram's Bot API. You may build a command that creates the OTP and transmits it to the user after your bot is configured and connected to Telegram. As an instance, you may develop the `/otp` command, which will generate the OTP and communicate it to the user. You must call the OTP generator function and save the created OTP in a variable in the code for your `/otp` command. The OTP may then be sent to the user using the Telegram API.

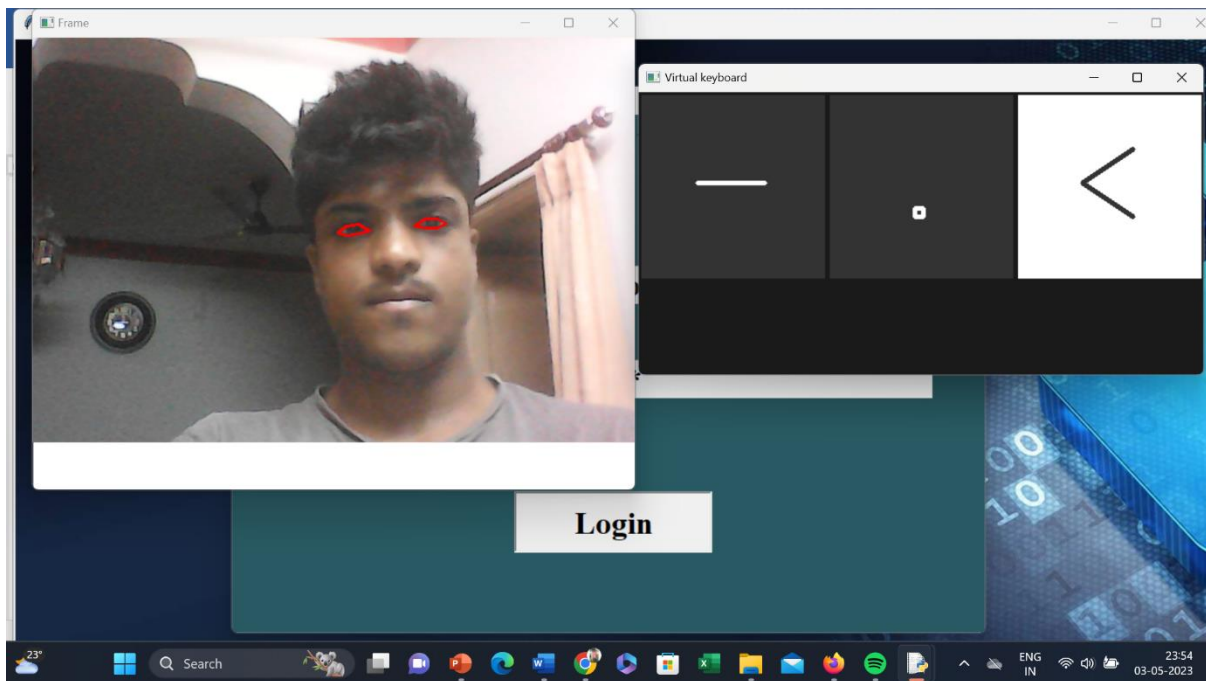
Login Page:



A fast and safe user experience should be guaranteed by the layout of the login page. It needs to be aesthetically pleasing, simple to use, and responsive to various devices and screen sizes. The website should also clearly explain how to input the username, password, and OTP code and provide the user feedback on any mistakes or problems they may have had when attempting to log in. The username field, which might be an email address, a username, or another special identifier, is used to identify the user's account. To confirm the user's identity and provide access to their account, the password field is employed.

Some websites or programmes may utilise an additional layer of security to secure user accounts in addition to the login and password. Use of a one-time password (OTP), which is often supplied to the user via an external communication channel like Telegram, is one such security mechanism. To confirm their identity and obtain access to their account, the user must input the 4-digit OTP number within a predetermined window of time.

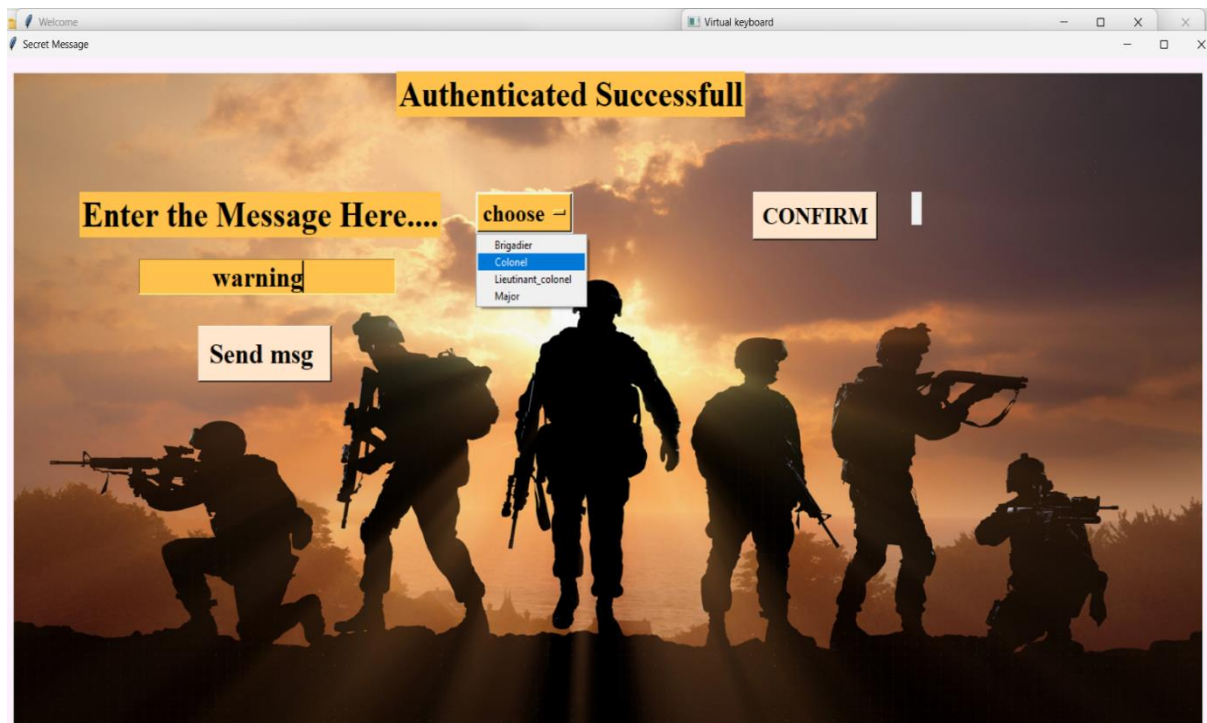
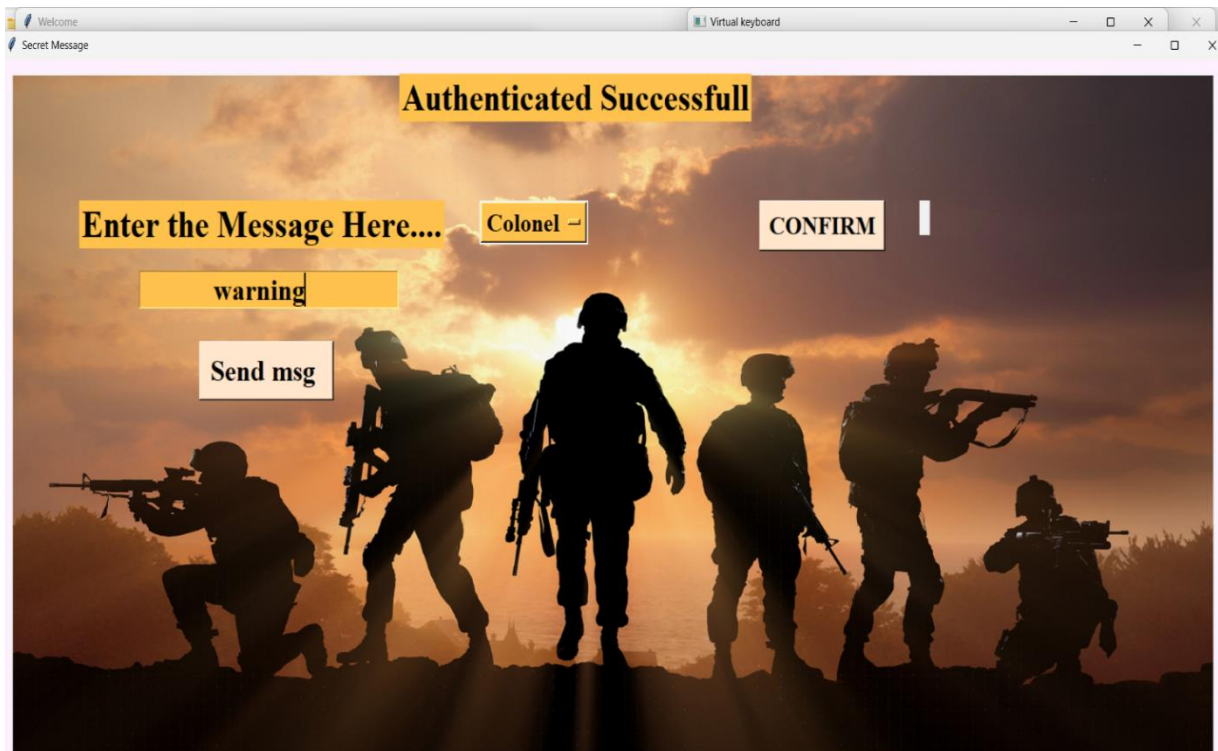
Morse-Code Password:

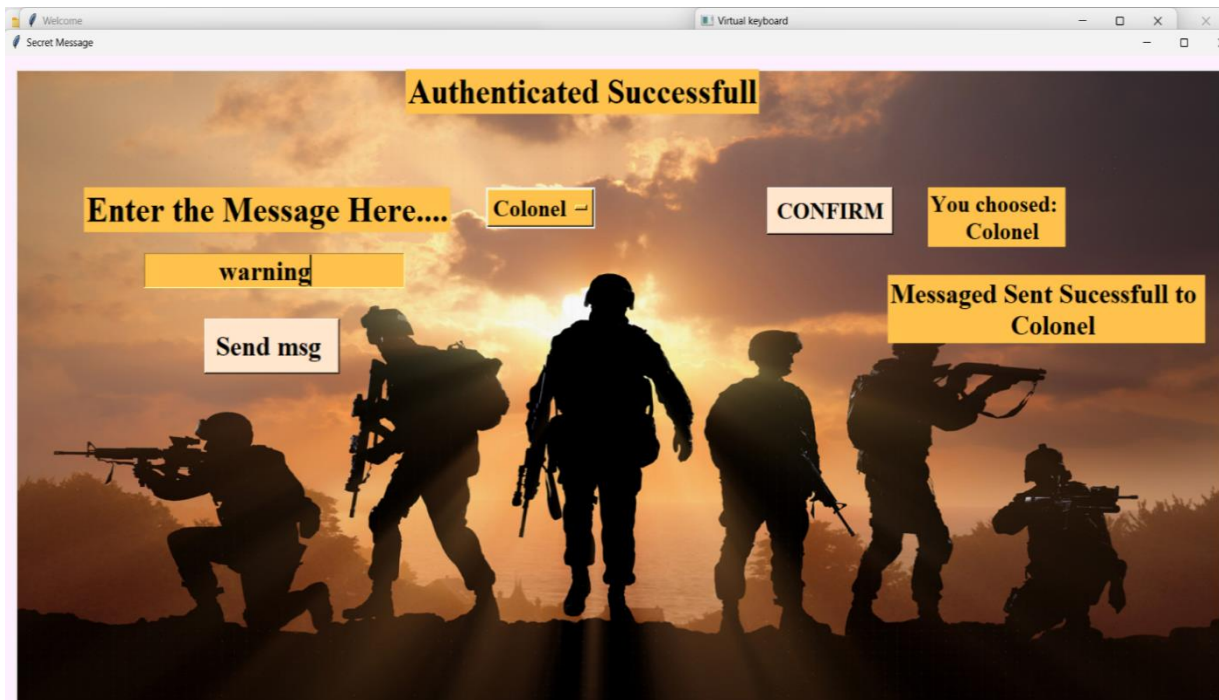


The page records a live video feed of the user's face, focusing on their eyes, in order to track their blinking patterns. A virtual keyboard is then presented on the screen using the blinking pattern as input.

Without a real keyboard or mouse, the virtual keyboard can be used to input passwords or other confidential data. The technology is usable for those with impairments or in situations where a physical keyboard is not available since the user may select keys on the virtual keyboard by blinking their eyes. In order to safeguard user data and prevent unauthorised access to the password or other sensitive information entered through the page, it should be built to assure user privacy and security.

Army Message box:





A page for the Army that contains This programme or tool is probably used by military personnel to send and receive secure communications. Enter the message, to whom it is addressed, and confirm.

The user can enter the message's content in the "Enter the message" area to send it. To protect the security of the sent information, the communications are probably encrypted.

The user can choose the receiver or recipients of the message by using the "To whom" field. Other military members, commanders, or other authorised persons with access to the secure communications system may be included in this.

The user may click "Confirm" to send the message after entering the message and recipient. The system could provide a warning or confirmation message before the message is delivered to make sure the user has entered the right information and has permission to send the message.

APPENDIX B: PAPER PUBLICATION

MORSE-CODE SECURITY BREACH

<i>Prof. Radha Rao</i> <i>radha.rao@saividy</i> <i>ya.ac.in</i> <i>Sai Vidya Institute of</i> <i>Technology</i>	<i>Rohith.M</i> <i>rohithm.19ec@saivid</i> <i>ya.ac.in</i> <i>Sai Vidya Institute of</i> <i>Technology</i>	<i>Surya Venkatesh</i> <i>swryavenkatesh</i> <i>.19ec@saiv idya.ac.in</i> <i>Sai Vidya Institute of</i> <i>Technology</i>	<i>Tanusha.v</i> <i>tanusha.v.18is@</i> <i>saividy.ac.in</i> <i>Sai Vidya Institute of</i> <i>Technology</i>	<i>Sheetal P.S</i> <i>pssheetal.19is@s</i> <i>aividy.ac.in</i> <i>Sai Vidya Institute of</i> <i>Technology</i>
---	--	---	--	--

ABSTRACT- In order to address analytically challenging issues, Technology, algorithm design, and data inference are all combined in data science. Such as handle massive volumes of data, practically all industries, including those in education, finance, healthcare, and business, use data science. The operational uses range from detecting cancer to estimating stock movement; when used for person identification, speech recognition, and text prediction in audio processing. Since the majority of individuals worldwide are having issues with security and authentication. For those who choose to use Morse code for their own authentication, we can offer real-time eye tracing for password authentication. As is well documented, technological developments in authentication and authorization have gotten a lot of support in the twenty-first century. Personal identifying numbers (PINs) have been widely used for user authentication and security since the late 1990s. These days, we prefer to employ a different tactic because PIN codes are so easy to crack. On the other hand, hands-free gaze-based PIN entry approaches for PIN authentication leave no physical traces and offer a more secure password entering option. The phrase "gaze-based authentication" describes the process of identifying the eyes in a series of frames and tracking their centers over time. For password verification, Morse code will be utilized, and digits will be depicted by slashes and dotted lines. This model presents a real-time application for gaze-based PIN entering together with an eye detection and tracking system for PIN identification utilizing a smart camera. **Keywords:** Morse-code, gaze-based authentication, PIN code, eye tracking.

I.INTRODUCTION

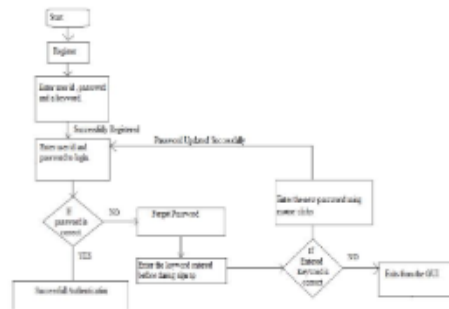
Data inference, algorithm design, and technology are all combined in data science, a multidisciplinary field. To manage vast volumes of data, practically all industries, including those in education, finance, healthcare, and business, use data science. Applications in real-world settings include text prediction, audio processing for speech, image processing for cancer prognosis, and identity recognition. Since the majority of individuals worldwide are having issues with security and authentication. For those who choose to use Morse code for their own authentication, we can offer real-time eye tracing for password authentication.

II.METHODOLOGY

Registration: The user enters his credentials on the first page that appears in this module. The username, password, and keyword that you input will be saved in a separate text file. The front-end implementation of the project is used to represent this module.

Login: The user or administrator inputs their credentials in this module in accordance with the information provided across the registered module. Once logged in, the client can use gaze-based authentication to confirm the password. The back-end implementation of the project is used to depict the conversion of eye blinks to morse code.

Forgot password: a user forgets their password in this module, they can generate a new one by typing the keyword from the register module.



III. LITERATURE SURVEY

[1] Mr. Kaustubh S. Sawant¹, Mr. Pange P.D; The goal of this effort is to use a smart camera and real-time eye recognition and tracking to enter and recognize gaze-based PINs. Eye tracking and real-time on-camera recording of the eye center location are done using NI Vision Builder and LabVIEW. The smart camera enables data collecting and analysis on-board.

[2] Sota Shimizu, Takumi Kadogawa, Shu-ichi Kikuchi, Takumi Hashizume, The primary objective of this work is a quantitative analysis of eye movement between the two groups. Using an eye-tracking device, a time series of gaze points are recorded from each subject. They are changed using a straightforward parametric model of the tennis forehand shot, and then they are examined using a significant test.

[3]Aniwat Juhong,T. Treebupachatsakul and C.Pintavirooj, This theory offers a Smart Eye surveillance system for those in their senior years and people with disabilities. The goal of this research is to develop eye movements that can be used to interact with carers and operate wheelchairs and other devices. The four components of this system are the imaging processing module, wheelchair control module, appliance control module, and SMS administration module. To find the coordinate of the eyeball, the eye movement image is captured and delivered to the Raspberry Pi microcontroller for OpenCV processing. A camera and bespoke C++ image segmentation make up the image processing module. The coordinate of the

eyeball is used to move the cursor on the Raspberry Pi's display.

[4] Leo Pauly, Deepa Sankar, For eye tracking, it employs a Haar-based cascade classifier, and for detecting eye blinks, it combines HOG characteristics with an SVM classifier. The proposed method offers a comfortable user engagement because it is not obtrusive.

[5] Hideyasu Sumiya, Takuma Itoh, The eye-ball movement is thought to be a factor that is deleted from the electroencephalogram (EEG) as an artefact, and this model suggests research that seeks to quickly identify BMI (Brain Machine Interface) patterns of that movement. We investigated the visual stimuli ERP's repeatability and its characteristics, which include constancy, high voltage, and a 50ms rapid response. As an ERP pattern discriminator, this study recommends three methods for extracting and detecting different patterns caused by varied directional ocular motions.

[6] Morse-code was applied in a way that will enable disabled individuals to converse with others normally and make their intended points clear. The use of led light to enter the patient's eye will have an adverse effect on the patient's sight, and this document also uses Morse-code for the entire alphabet (from A to Z) and for the numerals. Thus, it will be difficult for people to recollect all of the Morse code. If the patient doesn't know the code, they might not know how to utilize this application.

[7] we learn how specially abled persons communicate using Morse code by blinking their eyes and how their code is translated into audio. Utilizing a pyttx3 library function in this model This tool mostly turns any text into audio. we primarily use facial sensors and eye blink sensors to make it easier for those with paralysis to use the Morse code system. Zoran DURIZ, proposed a system that uses a web crawler to gather data and an EAP detector and extractor.

[8] Establish secure communication between the two users utilizing facial and eye blink detection to prevent

information leaks. It is an open-source CV program uses the dlib module of Python 3.6 to recognize facial shapes. The important takeaway from this concept is that any type of data sent between users will be protected utilizing face detectors and the eye blink mechanism.

[10] a password with a specific level of security is set using gaze-based verification. After the password was entered, there was a set amount of time for password authentication. The average recognition rate was then stated to be around 8%. All of the passwords that were entered had a time limit of roughly 1.05 seconds. These lead to the recognition of gaze-based verification, which reduces the risk during the system's password identification procedure.

[11] In order to improve predictive power, this work attempts to handle complex problems. Speech loss and motor impairments from accidents are not specifically answered. They employed IR-LED sensors connected to pin of the new loop, which blinks. Microcontrollers made by Arduino are user-friendly, adaptable, and very affordable. They had employed IR sensors, which were more dangerous and difficult to use. therefore, there must be compulsory VAPT testing to prevent cyber-attack cases and strengthen system security.

[12] the video was used in a way that allowed the message to be concealed. By using the AES method, they may store the most memory possible inside the video while yet maintaining the highest levels of security and retrieval speed. However, the project's biggest drawback is that viewers could not realize if there is a message concealed within the video, causing them to miss it. People would overlook the spot where the message had been corrected.

[13] Morse-code was used in this project in a way that prevented other hackers from deciphering the users' use of it, allowing messages to be transferred from one location to another. In essence, this project operates as two users of the system, each at a different end. In order to prevent other hackers from intercepting the signal and understanding the code used by the users, the user

can input any private messages he wants to transmit and have them compressed using Morse code before sending them to the recipient. however, one drawback is that hackers unable to comprehend the personal message.

[14] For the crippled people involved in this study, Morse-code was utilized so they could speak to others without stammering. The paralyzed patient should blink for each letter of the alphabet in front of the camera (A-Z). The biggest drawback, however, is that the patient will find it difficult to memories the codes for every letter of the alphabet (A-Z).and if regular eye blinking is also considered to be part of the code, the message the patient is trying to convey will differ from the one shown on the screen.

[15] Morse code provides more dependable communication in noisy communication channels than other coding. A fuzzy logic had been developed by them. These are neural networks, which mimic human decision-making but much more quickly. This system's accuracy is primarily dependent on faulty inputs and data. They are not generally favored. The system that has been presented can be utilized to communicate within a 50-meter radius. The system has the ability to respond to WIFI or touchpad-entered morse code by acting accordingly. Which lowers errors.

[16] The 3D simulators employed in this study, which focuses on Robot Operating System, are used for multi-robot systems. Robotics simulators are useful tools in research because they can be used to evaluate new ideas, concepts, efficiency, and resilience. A multi-UAV simulator built on ROS and Unity 3D was introduced in 2016. These also receive assistance from the graphical user interface, simulator mailing list, and tutorials, where MORSE offers a configurable interface at variable level that is simple to code. With Ubuntu 14.04.4 and ROS Jade, they used MORSE 1.4 and Gazebo 5.0. In this instance, MORSE's real-time ratio is 1.0 while Gazebo's is 0.9.

[17] Article discusses wireless online. The goal of this study is achieved by communication amongst

authorized users who have developed a simple and affordable method to help patients with speech disorders. Since secure communication between user is essential in today's environment, this technology is employed for that purpose. Using image processing techniques, we are computing eye blinks. detection of eye measurement, which produces dots and dashes by blinking the eye at various intervals and giving the appearance of a particular ratio.

[18] the gestures were detected using the gaze technique and Morse code so that the words could be framed and shown in the system. To portray the alphabets, they had employed eye movements. They must move the eye in the directions of North, South, East, and West in order to create single letter at a time. The key issue with this project is that participants must recognize each movement for the alphabet they created. This would have caused further problems for the project's system.

[19] Characters and images are classified using this neural network machine learning system. By enhancing resilience and general capabilities, as well as network performance, the implications of data size have been investigated. Arabic numbers and symbols were used to create the algorithm. Morse code is made up of dots and dashes. The dashes are 3.9 values in length. The introduction of Dilation made these simpler. which is its capacity for several learning algorithms. These archives have more precision thanks to neural networks and algorithms.

[20] students talked about the security of the ATM Pin and how simple it is to change it. They have determined that the main problem with the current ATM pin method is that anyone may readily see the pin number and abuse it. They are employing Eye-gaze interaction system techniques to address this problem. They are improving three different eye-based approaches. For further security, they use gaze basis synchronization with hardware button click, Typical gaze-based pin input and Recalling pins using shapes

IV. EXISTING SYSTEM

In the twenty-first century, it has been supported to advance the technology of authentication and authorization. Since the late 1990s, personal identification numbers (PINs) have been utilised extensively for user authentication and security. We prefer to use a different strategy these days because PIN codes are so simple to hack. On the other side, PIN authentication using hands-free gaze-based PIN entry techniques leaves no physical traces and thus provides a more secure password entry option. The current technology does not offer a safe way to authenticate partially sighted people.

V. PROPOSED SYSTEM

The model consists of a back-end database with user interface. The user can interact with the system thanks to the creation of the GUI. Frontend: The user must first register by giving a user ID of their choosing, a password (PIN), and a keyword. The user's user id and password are required to log in after registration. The PIN is captured using a web camera as input in the form of Morse code. The PIN is captured using a web camera as input in the form of Morse code. Backend: - The user-input PIN that was stored in the database during registration is compared to the entered PIN. The screen closes if the PIN entered is incorrect. The successful authentication message is displayed if the PIN entered. If a user forgets his password, he can use the keyword to authenticate, replace the current password with a new one, and generate a different OTP for each login if necessary.

VI. RESULTS

For conveying essential information inside the army, this effort offers a three-way authentication mechanism of communication. It also offers a safe environment for communication. Every time the user attempts to log in, he is given a special pair of numbers, and this continues until the user blinks the special number in morse code.



Fig 2 Welcome Page

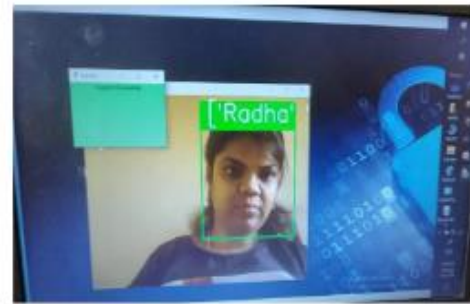


Fig 6 Registration Successful



Fig 3 Registration Page



Fig 7 Authentication by Morse Code



Fig 4 Login Page



Fig 8 Successful Page



Fig 5 Face Capturing



Fig 9 Message Successfully Sent

VI. CONCLUSION

Our project essentially provides two-factor authentication. To safeguard an account or system, two layers of protection are essentially provided by two factor authentication. Here, we're using mouse clicks to turn letters or numbers into source code and gaze-based authentication to further increase security. This model demonstrates a real-time application for gaze-based PIN entry as well as eye detection and tracking for PIN identification using a smart camera.

REFERENCES

- [1] S. Muller, M. Deicke, and R. W. De Doncker, "Doubly fed induction generator systems for wind turbines," *IEEE Ind. Appl. Mag.*, vol. 8, no. 3, pp. 26–33, May/Jun. 2002.
- [2] D. Zhi and L. Xu, "Direct power control of DFIG with constant switching frequency and improved transient performance," *IEEE Trans. Energy Convers.*, vol. 22, no. 2, pp. 110–118, Mar. 2007.
- [3] National Grid Transco, Appendix 1. (Feb. 2004). Extracts from the grid code—Connection conditions [Online].
- [4] IEEE Recommended Practices and Requirements for Harmonic Control in Electrical Power Systems, IEEE Standard 519-1992, 1993.
- [5] J. Hu, H. Xu, and Y. He, "Coordinated control of DFIG's RSC and GSC under generalized unbalanced and distorted grid voltage conditions," *IEEE Trans. Ind. Electron.*, vol. 60, no. 7, pp. 2808–2819, Jul. 2013.
- [6] H. Xu, J. Hu, and Y. He, "Integrated modeling and enhanced control of DFIG under unbalanced and distorted grid voltage conditions," *IEEE Trans. Energy Convers.*, vol. 27, no. 3, pp. 725–736, Sep. 2012.
- [7] C. Liu, D. Xu, N. Zhu, F. Blaabjerg, and M. Chen, "DC-voltage fluctuation elimination through a DC-capacitor current control for DFIG converters under unbalanced grid voltage conditions," *IEEE Trans. Power Electron.*, vol. 28, no. 7, pp. 3206–3218, Jul. 2013.
- [8] Mehrube Mehruoglu, Vuong Nguyen, "Real-Time eye tracking for password authentication", Conference: IEEE International Conference on Consumer Electronics (ICCE), January 2018.
- [9] Sota Shimizu, Takumi Kadogawa, Shu-ichi Kikuchi, Takumi Hashizume, "Quantitative analysis of tennis experts' eye movement skill", Conference: International Workshop on Advanced Motion Control (AMC), March 2014.
- [10] Aniwat Juhong, Michigan State University, T Treebupachatsakul, C Pintavirooj, "Smart Eye tracking system", Conference: International Workshop on Advanced Image Technology 2018 (IWAIT 2018), January 2018.
- [11] B. NagaSoundari, M. Nandakumar, R. Nivetha, K. Rajakumari, "Extension of desktop control to robot control by eye blinks using Support Vector Machine (SVM)", Conference: International Conference on Recent Trends in Information Technology (ICRTIT), June 2011.
- [12] Takuma Ito, Tomoyuki Shinji, Hideyasu Sumia, Mituru Baba, "Eye movement-related EEG potential pattern recognition for real-time BMI", Conference: SICE Annual Conference, August 2010.
- [13] Mircea Constantin Scheau, Larisa Gabudeanu, "Risk-based approach in preventing mobile banking cyber-attacks" Conference: 25th RSEP International Conference on Economics, Finance & Business, At: Paris, France; Volume: ISBN: 978-605-70583-8-6
- [14] Seongki Kim, JinHo Ryu, Youngchul Choi, YooSeok Kang, Hongle Li, Kibum Kim, "Eye Contact Game Using Mixed Reality for the Treatment of Children with Attention Deficit Hyperactivity Disorder", May 2020.

- [15] Elyas Baray, Nitish Kumar Ojha, "WLAN security protocols and WAP3 security approach measurement through aircrack-ng technique", 2021 5th International Conference on computing Methodologies and communication (ICCMC), 23-30, 2021
- [16] Indrajit Das, Ria Das, Shalini Singh, Amogh Banerjee, Md. Golam Mohiuddin, Avirup Chowdhury, "Design and Implementation of Eye Pupil Movement Based PIN Authentication System", Conference: VLSI Device, Circuit and System Conference (VLSI-DCS), July 2020.
- [17] S. Dey, K. M. Chugg and P. A. Beerel, "Morse Code Datasets for Machine Learning," 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2018.
- [18] Kingshuk Mukherjee, Debdatta Chatterjee "augmentative and alternative communication device based on eye-blink detection and conversion to morse-code to aid paralyzed individuals" iee 2015.
- [19] K. Niu et al., "WiMorse: A Contactless Morse Code Text Input System Using Ambient WiFi Signals," in IEEE Internet of Things Journal, vol. 6, no. 6, pp. 9993-10008, Dec. 2019 .
- [20] S. N. Daskalakis, G. Goussetis, S. D. Assimonis, M. M. Tentzeris and A. Georgiadis, "A uW Backscatter-Morse-Leaf Sensor for Low-Power Agricultural Wireless Sensor Networks," in IEEE Sensors Journal, vol. 18, no. 19, pp. 7889-7898, 1 Oct. 1, 2018.

Paper Acceptance Status:

Acceptance Notification and Review Result of Your Paper - TIJER104218 | TIJER (ISSN:2349-9249) | www.tijer.org | editor@tijer.org Your Email id: suryavenkatesh.19ec@saividya.ac.in

1 message

Editor Tijer <editor@tijer.org>
To: suryavenkatesh.19ec@saividya.ac.in

Fri, May 12, 2023 at 17:28

TIJER-International Research Journal
An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor 8.57 Calculate by Google Scholar and Semantic Scholar | AI-Powered Research Tool, Multidisciplinary, Monthly, Multilanguage Journal Indexing in All Major Database & Metadata, Citation Generator, Peer-Reviewed, Refereed, Indexed, automatic Citation Open Access Journal

Acceptance Notification and Review Result of Your Paper - TIJER104218 | TIJER (ISSN:2349-9249) | www.tijer.org | editor@tijer.org Your Email id: suryavenkatesh.19ec@saividya.ac.in
Track Your Paper Link [Track Your Paper](https://www.tijer.org/track.php?r_id=104218)
https://www.tijer.org/track.php?r_id=104218

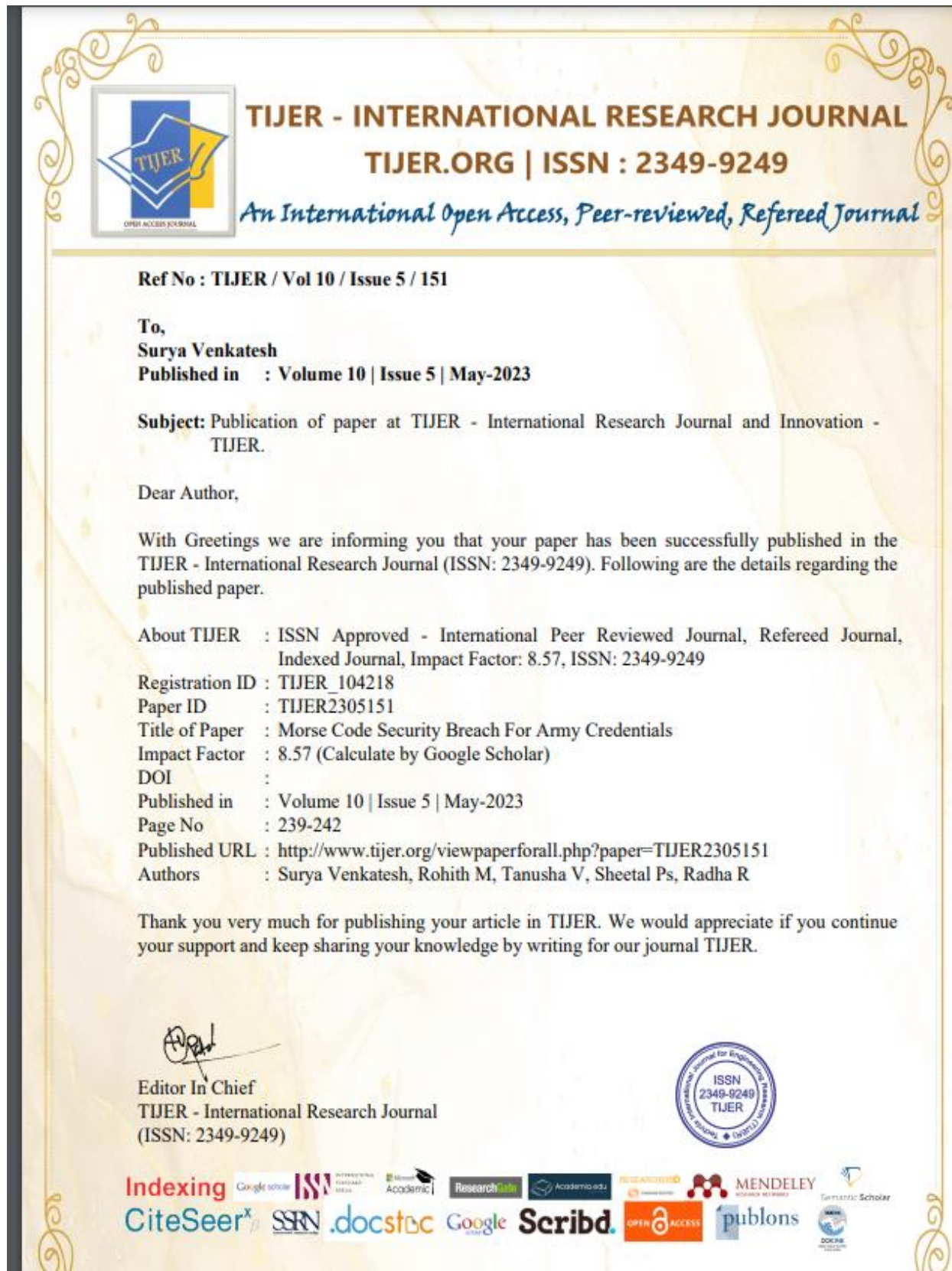
Dear Surya Venkatesh,
Your manuscript with Registration ID: TIJER104218 has been **Accepted** for publication in the TIJER-International Research Journal (www.tijer.org). Track Your Paper Link [Track Your Paper](https://www.tijer.org/track.php?r_id=104218) https://www.tijer.org/track.php?r_id=104218 Your Review Report is as follows:

An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor 8.57 Review Results:

Registration ID	TIJER104218
Email ID	suryavenkatesh.19ec@saividya.ac.in
Paper Title	Morse Code Security Breach For Army Credentials
Review Status	Accepted
Impact Factor & Licence:	Open Access, Peer-Reviewed, Refereed, Indexing,ISSN Approved,DOI and Creative Common Approved & 8.57 Calculated by Google Scholar
Unique Contents	94 %
Comments	Paper Accepted Complete Payment and documents Process. Complete Phase 2 Payment and Phase 3 Documents Process so within 1 to 2 day it will publish

Phase 2: Pay Publication Fees and Open Access Processing Charges
Option 1 : For Indian Author Payment Details

Paper Publication Certificate:



APPENDIX C: Plagiarism Report

DrillBit			
DrillBit Similarity Report			
11	70	B	A-Satisfactory (0-10%) B-Upgrade (11-40%) C-Poor (41-60%) D-Unacceptable (61-100%)
SIMILARITY %	MATCHED SOURCES	GRADE	
LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	blog.entegraty.com	1	Internet Data
2	IEEE 2018 IEEE International Conference on Consumer Electronics (IC, by Mehrubeoglu, Mehrub- 2018	1	Publication
3	datarecoverydigest.com	1	Internet Data
4	pdfcoffee.com	<1	Internet Data
5	www.aalimec.ac.in	<1	Publication
6	steamcommunity.com	<1	Internet Data
7	hotel.wmisports.com	<1	Internet Data
8	dspace.daffodilvarsity.edu.bd 8080	<1	Publication
9	Economic reforms in the industrial sector of the Peoples Republic of by Wolfgan-1983	<1	Publication
10	Ideal Timex2013Frequency Masking Algorithms Lead to Different Speech Intellig by Koning-2015	<1	Publication
11	PATIENT MONITORING AND ALERT SYSTEM USING MOBILE APPLICATION (DIGITAL PNEWS INDIA) BY MANTENA VISHWA PRAPULLA VARMA YR-2021,JNTUHCE	<1	Student Paper
12	docplayer.net	<1	Internet Data

13	docplayer.net	<1	Internet Data
14	IEEE 2020 IEEE 5th International Conference on Image, Vision and Computing (IC	<1	Publication
15	Intelligent Computing Everywhere	<1	Publication
16	www.readbag.com	<1	Internet Data
17	www.pyimagesearch.com	<1	Internet Data
18	Pattern Classification Approaches for Breast Cancer Identification via MRI Stat by Yin-2020	<1	Publication
19	www.10tv.com	<1	Internet Data
20	biomedcentral.com	<1	Internet Data
21	biomedcentral.com	<1	Internet Data
22	byjus.com	<1	Internet Data
23	moam.info	<1	Internet Data
24	Multidisciplinary perspectives on attention and the development of self-regulati by Andre-2007	<1	Publication
25	biomedcentral.com	<1	Internet Data
26	docplayer.net	<1	Internet Data
27	adoc.pub	<1	Internet Data
28	IEEE 2020 IEEE VLSI Device Circuit and System (VLSI DCS) - Kolkata, India (202	<1	Publication
29	oneplussupport.s3.amazonaws.com	<1	Publication
30	qdoc.tips	<1	Internet Data

31	Projection Methods for the Analysis of Complex Motions in Macromolecules by Hinsien-2000	<1	Publication
32	www.frontiersin.org	<1	Publication
33	www.ijarcs.info	<1	Publication
34	en.wikipedia.org	<1	Internet Data
35	moam.info	<1	Internet Data
36	Quantum Neural Network with Improved Quantum Learning Algorithm by Chen-2020	<1	Publication
37	alethonews.com	<1	Internet Data
38	ijircce.com	<1	Publication
39	Robot Control Using Anticipatory Brain Potentials by Boinovski-2011	<1	Publication
40	www.intechopen.com	<1	Publication
41	www.monell.org	<1	Internet Data
42	www.readbag.com	<1	Internet Data
43	Article Published by International Journal Of Engineering & Computer Science - www.ijecs.in	<1	Publication
44	Thesis Submitted to Shodhganga Repository	<1	Publication
45	www.lozano-hemmer.com	<1	Internet Data
46	www.readbag.com	<1	Internet Data
47	citeseerx.ist.psu.edu	<1	Internet Data
48	journalofbigdata.springeropen.com	<1	Internet Data

49	mdpi.com	<1	Internet Data
50	publications.iowa.gov	<1	Publication
51	Seeing the Unseen Revealing Mobile Malware Hidden Communications via by Caviglione-2015	<1	Publication
52	Soft dry electroophthalmogram electrodes for human machine interaction by Cheng-2019	<1	Publication
53	www.dx.doi.org	<1	Publication
54	www.dx.doi.org	<1	Publication
55	www.jatit.org	<1	Publication
56	blog.kore.ai	<1	Internet Data
57	buslibguides.smu.edu	<1	Internet Data
58	coek.info	<1	Internet Data
59	Comparative study of reflector antennas for small earth stations by Clarricoats-1987	<1	Publication
60	docplayer.net	<1	Internet Data
61	documents.mx	<1	Internet Data
62	moam.info	<1	Internet Data
63	schedule.yale.edu	<1	Internet Data
64	smile-ohm.co.uk	<1	Internet Data
65	The Internet as Social Support for Older Carers of Adults With Intellectual Disa by Elizabet-2012	<1	Publication

66	The scientific dimensions of social knowledge and their distant echoes in 20th-c by MIROWSKI-2004	<1	Publication
67	www.abbeyboilers.com	<1	Internet Data
68	www.kharkov.davr.gov.ua	<1	Internet Data
69	www.readbag.com	<1	Internet Data
70	www.thefreelibrary.com	<1	Internet Data