International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(2) 2026 pp. 1057-1067
https://doi.org/10.70917/ijcisim-2026-3535
Article

REAL-TIME EYE BLINKING FOR PASSWORD AUTHENTICATION

Nagashree S[1], Abhinav Aditya[2], Divya Kumari[3], Akshaya Raj[4]

[1,2,3,4]Department of Information Science and Engineering, JSS Academy of Technical Education, Bangalore
[*]Corresponding Author: Dr. Nagashree S[1], e-mail: nagashrees@jssateb.ac.in

Abstract: Personal Identification Number (PIN) is widely used for user authentication and security. Password verification using PINs requires users to enter the PIN, which could be a risk of hacking the PIN with many methods. Confirming the authenticity of the PIN by an eye blink input method does not leave physical marks/trats behind and thus offers the option to enter a more secure password. Blink-based verification receives instant blinks in all consecutive photo frames to produce a PIN. This paper introduces a real-time system that includes three levels of authentication face detection, instant eye blink PIN entry and OTP (One-Time Password) to avoid shoulder surfing (it is a type of attack in which hackers practice of spying on the user's finger movement when user typing their PIN or password) and thermal tracking attacks (when user typing their password or some personal secured information on a keypad/keyboard, they leaving heat traces behind that can be taken by hackers).
Keywords: PIN, Authentication, eye Blink-based Authentication, Thermal tracking attacks, Shoulder surfing

1. INTRODUCTION

The use of PINs, called personal identification numbers, is a simple system authentication approach for many applications, such as cash withdrawals (ATMs), electronic authorization, opening doors, and unlocking some personal devices like phones and laptops. But the challenge is seamless personal verification even when PIN verification is used. Based on European ATM Security, fraudulent attacks on ATMs from 2015 to 2016 increased by 26%. The reality that the legitimate person should enter the code in public makes the PIN vulnerable to password attacks, consisting of shoulder surfing and thermal tracking. Password verification requires a person to type the password physically. Verifying by Gaze-Based (Gaze-Based is a simple password entry method in which in place physically typing or touching the screen to enter a password, the user has to look at each character to trigger password digits like eye typing) PIN input authentication generally, the password uses a 4-digit PIN. It does not leave the actual steps and provides a higher security level for entering the password option. Gaze-based verification helps to locate the eye in all consecutive photo frames, as well as eye-tracking [3].

This paper introduces a real-time PIN entry system that is based on blinking eyes, PIN detection, and eye detection using a smart camera. This helps to avoid shoulder surfing and thermal tracking attacks. Due to poor interaction between users and systems, there are security problems. To overcome these types of attacks, this paper proposes a three-level security system: 1) the first level is face detection and face recognition, 2) the second level is authenticating PIN using eye blink PIN entry system, and 3) finally, at the last level, authenticate a user through OTP (One-Time Password) verification that is sent to the user's registered mobile number.

2. EXISTING SYSTEM

There is work related to the sample in the fields directly related to this work.

Alexander, Martin, and Heinrich (2009) introduced the "Eye-Pass Design Method"; EyePassShapes expands and develops two authentication methods for integration, PassShapes and EyePIN. The security, which is compared to a password or PIN in PassShape users to paint the shape (including stripes) in a certain order; this method enhances memory but does not improve. In place of entering numbers, eye movements are performed by the user representing the digits that are related [8].

Sonia, Robert, and Alain introduced "CGP enhancements" in 2010. CGP, abbreviated as Cued Gaze Points, can be considered an image transfer program that protects against such attacks using eye gaze password input, rather than clicking the mouse. Still, it requires certain techniques to improve viewing accuracy. This study introduced two improvements: a combination of viewing points close to the neighbours and a one-point measurement before entering the entry name [9].

Justin, Kenrich, and Bogdan (2011) introduced the "EyeDent System", which is a development of current eye-tracking systems that requires compression. Any time the user looks at anything, the trigger is triggered sign. This method has the advantage of allowing the user to be sure of their preferred speed, instead of a predetermined duration [10].

Albrecht, Florian, and Andreas introduced a "novel-based verification program" in 2012, a program that uses a professional pass-through memory for all images. This method uses visual acuity calculations to hide these possible parts of the image. They have created a virtual threat model related to potential attacks on public places, such as recording users' actions during the ATM withdrawal process [11].

David Rozado (2013) introduced a "direct title measurement parameter" using this parameter

collected during the rating process to make it difficult for someone else to enter a password by staring or in a situation where the impostor knows the password [12].

Arsenovski, Trpkova, and Mihajlov introduced the "ImagePassEyeTrackingStudy" in 2013. Image Pass may be considered a visual-based verification program. The purpose of the study was to obtain user feedback and to respond to confirmation of image quality [13].

Emanuel, Regina, Mohamed, Florian, Mariam, and Andreas introduced the "GazeTouchPass Scheme" in 2016, a multimodal approach that combines touch and looks about ensuring shoulder-toshoulder mobility. GazeTouchPass accepts login by changing multiple input methods through the authentication process [14].

Zhenjiang et al. (2017) introduced the "iType System", an approach that uses eye-tracking to type a confidential key PIN on mobile commodity platforms [15].

3. DESIGN AND IMPLEMENTATION

The main goal of the proposed paper is to build an application with three levels of authentication. At present, in financial transaction humans have to physically interact with the related machine, like in ATM machines or in an open place; users have to enter their PIN physically. This leads to their PIN being hacked by shoulder surfing or thermal tracking. To avoid this hacking, this work proposed three levels of authentication.

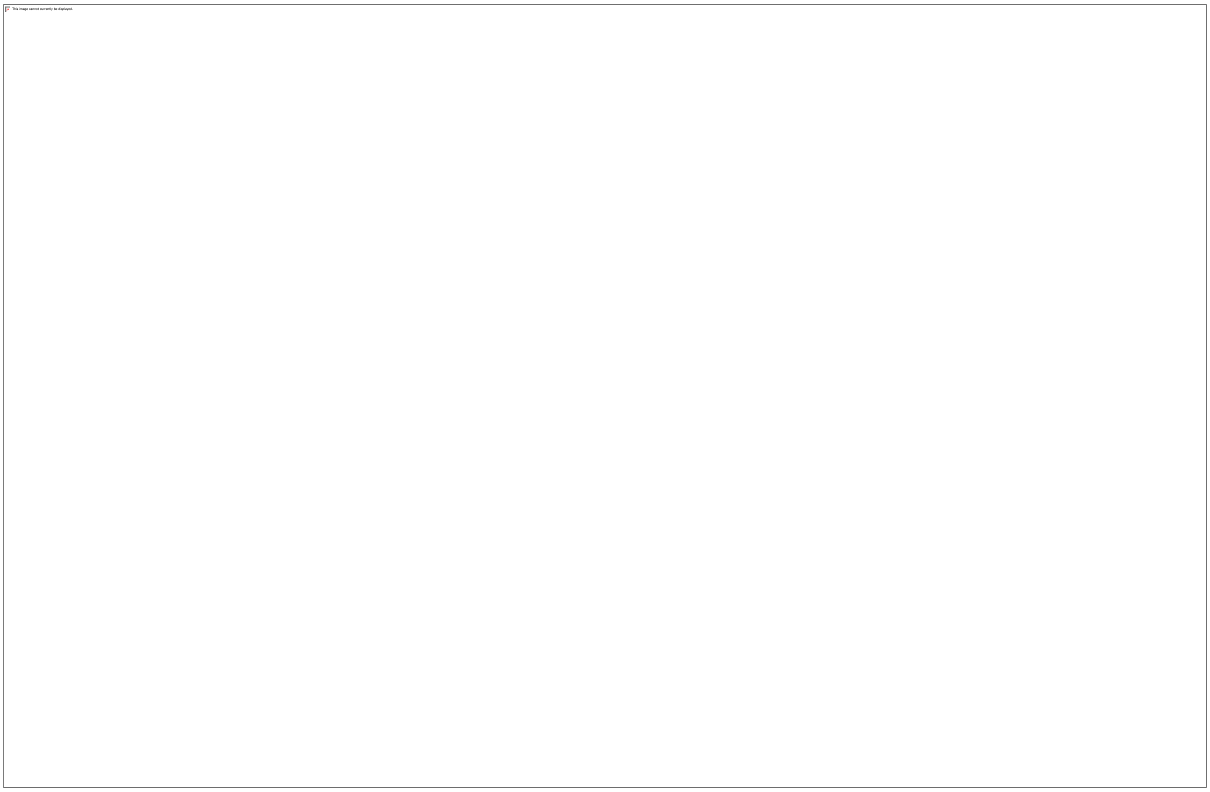


Fig: 3.1 Application Flow Diagram

Figure 3.1 shows the flow diagram of the application. When the application starts, if a new user, then the user has to register. In the registration, the application takes the user's details and captures the user's face and then trains the system. And then start login. In the login process, the three-level of authentication work.

At the first level, the application captures the face. It detects the face from an image. After detecting the face, the application recognizes the face and compares the face from the database where the user's images are present and recognizes the user. In the second stage, after recognition, the user has to enter the PIN using the eye-blink method. The application compares the PIN with the saved PIN that is present in the database, and if the PIN is matched, it proceeds to the third level, in which the OTP is sent to the register mobile number of the user, and the user has to enter the OTP. If the OTP is correct, then the user has logged in. Then the user is able to do their transaction.

3.1 OBJECTIVE AND ALGORITHMS

In this work, the first objective is to detect faces using the Haar Cascade algorithm and then use the Local Binary Pattern Histogram(LBPH) algorithm for face recognition. For that, we are using a system camera as a resource. The second objective is to detect the Eye-blink with the help of OpenCV, which captures the image through the camera. Finally, the application has to generate an OTP with the help of the random number generation module, and this OTP is sent to register a mobile number using any Short Message Service (SMS) Application Programming Interface (API).

3.1.1 HAAR CASCADE (FACE DETECTION)

Haar Cascade is an algorithm of machine learning algorithm for the detection of objects in multimedia like snapshots or video. It is a technique where the cascade feature is skilled or taught from a lot of high-quality and terrible photographs. Then it is used for training to find targeted items in different images [17].

This algorithm has four stages: 1. Haar Feature Selection, 2. Creating Integral Images, 3. Adaboost Training, 4. Cascading Classifiers.

It is able to detect faces as well as body parts in an image, but it can be trained to identify almost any type of object. The Haar cascade has mainly three features: two rectangle features, three rectangle features, and four rectangle features. These features help to find out the edges, dots, or lines in an image. For a given image, we cannot apply all these features, as it will be hectic even for a high-performance computer; therefore, we create an integral image from the original image so that the complexity will be reduced, and it is also easy to find facial features from the integral image. The third step in haar cascade is using a technique called Adaboost, where integral image pixels are given as input, and it selects features; the classifiers are trained to get the best out of those features. Finally, images are fed to cascade classifiers consisting of a collection of stages; the stages reject negative samples and output all positive samples [16].

Fig: 3.2: Creating an Integral Image

In Figure 3.2, the left part of the image contains an 8x8 matrix that shows the original image in greyscale and then converts it into the integral image using haar rectangular feature, which is shown in the middle of the image, in which the pixels value is 0 or 1: 0 for light pixels on the left and dark pixels on the right. The Haar evaluation is done by taking the difference between the average of the pixel values of the dark region and the light region. If it is near 1, then it means there is an edge detected by the used Haar feature.

3.1.2 LBPH (FACE RECOGNITION)

This work performs face recognition or face classification every day and every moment of our life. But it may seem easy for humans, while it is quite difficult for the computer. As humans, in order to recognize someone, we first have to meet the person or see them for the first time so that the face of the person gets stored in our memory. In computer language, it is called a data set. Then, when we meet the person again next time, our brain compares the current image with the previously stored ones in our memory, and then it recognizes the person. Similarly, in a face recognition system, when the user comes in front of the camera, the computer first has to detect the face from the input image, which is achieved by haar cascade classifier (note: face recognition is different from face detection). After detecting the face, to recognize it, we use LBPH. LBPH is a simple yet effective texture operator that assigns values to the pixels of an image by thresholding the neighbors of each pixel and then finally considers the results as a binary number [17].

First, we need to train the algorithm. To achieve this, we need to use a dataset with the facial images of the people we want to recognize. We also need a unique ID for each person, so that the algorithm uses it to recognize the input image and give it as the output.

Fig: 3.3 Eye Blink Detection

Figure 3.3 shows the steps of LBPH. The first computational step of LBPH is to create an intermediate image from the original image. This is done in order to describe the original image in a better way, by highlighting the facial characteristics shown in the above figure.

In order to achieve this, the algorithm uses a sliding window on the basis of the parameters radius and neighbors.

First, we obtain the image of a face in grayscale values.

Later obtain the part of the image as a 3x3 matrix, which is shown in Figure 3.3. It can also be called a 3x3 matrix containing the pixels. Consider the center value of the matrix as i_c, which is used for thresholding the neighboring values of the matrix to assign a new value based on the i_c.

For each neighboring value of the matrix, we assign a new binary number as 1 if the value is higher than the i_c else 0 if the value is less than i_c.

Now the matrix only contains binary values except the i_c value.

Later, we have to concatenate each binary value starting from a position to the end and generate a new binary number. This binary number is converted to decimal and assigned to the center value, that is i_c. The same procedure is performed on each and every pixel in a facial image. In the end, we obtain a new image that contains better information than the original image. This image is converted into a histogram based on the pixel values to convert the complete image to the histogram.

In order to recognize the face, the histogram of the input image is compared with a histogram of the dataset based on different approaches such as Euclidean distance, chi-square, absolute value, etc.

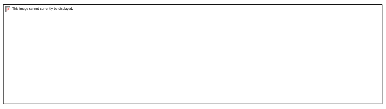
In this work Euclidean distance-based approach is used to identify closely related images. The algorithm returns the ID of the image with the least Euclidean distance compared to other images.

To detect eye blinks, we consider the metric Eye Aspect Ratio (EAR), which is a basic metric to detect and count the number of eye blinks from a live video stream. The input facial image consists of different features like eyes, nose, ears, jawline, etc. From these features, with the help of the Haar cascade classifier, we extract the eye feature, which is the only feature required for eye blink detection [18].

Each eye is divided into 6 (x, y) coordinates. Starting from the left corner of the eye and remaining in a clockwise direction.

Fig: 3.4: Six coordinates on eye (EAR coordinates)

Figure 3.4 shows all 6 coordinates on the eye that are helping to form the Eye Aspect Ratio(EAR) equation. Based on these coordinates, the EAR is given as,



Equation --- 1

According to equation-1, the numerator of the equation gives the distance between the vertical coordinates of the eye, while the denominator computes the distance between the horizontal coordinates of the eye. This EAR remains approximately constant throughout the video stream but immediately falls to 0 when an eye blink occurs. Hence, when EAR is approximately equal to 0, we can consider that an eye blink has occurred.

Fig: 3.5 EAR value trace

The above figure (Fig. 3 .5) shows the EAR value when the eye is opened and when the eye is blinked. In the figure graph at the beginning, the EAR value is approximately 0.25, and when the eye is blinked, the value decreases suddenly and reaches between 0.1 and 0.05.

4. CONCLUSION AND FUTURE WORK

The work was done by using a few machine learning algorithms and inbuilt modules of OpenCV. The web application is tested with a 6-digit PIN. This application has three levels of authentication that make it very secure, and it prevents shoulder surfing as well as thermal tracking, which is the main target of this work. The final level, which is OTP verification, makes the application more secure authentication as many methods are available to hack a PIN. In the case of a new user, first the new user has to register themselves and then start the login process.

The work is restricted to the actual user (account holder). That is the main limitation of this work. In future work, it can be removed with some techniques. Still, for now, if the user's guardian wants to access the system in place of the main user, the guardian has to enter all details manually and enter the OTP that is sent to the register mobile number.

5. RESULT

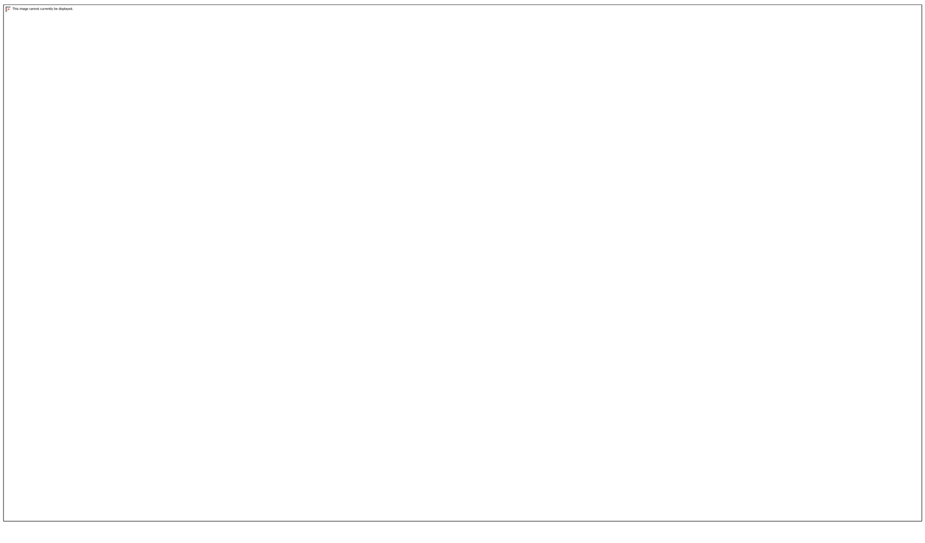


Figure 5.1: Home Page (Landing page)

Figure 5.1 shows the home page after the application started. It is a landing page, in which the buttons and two input field is available for client ID and PIN.

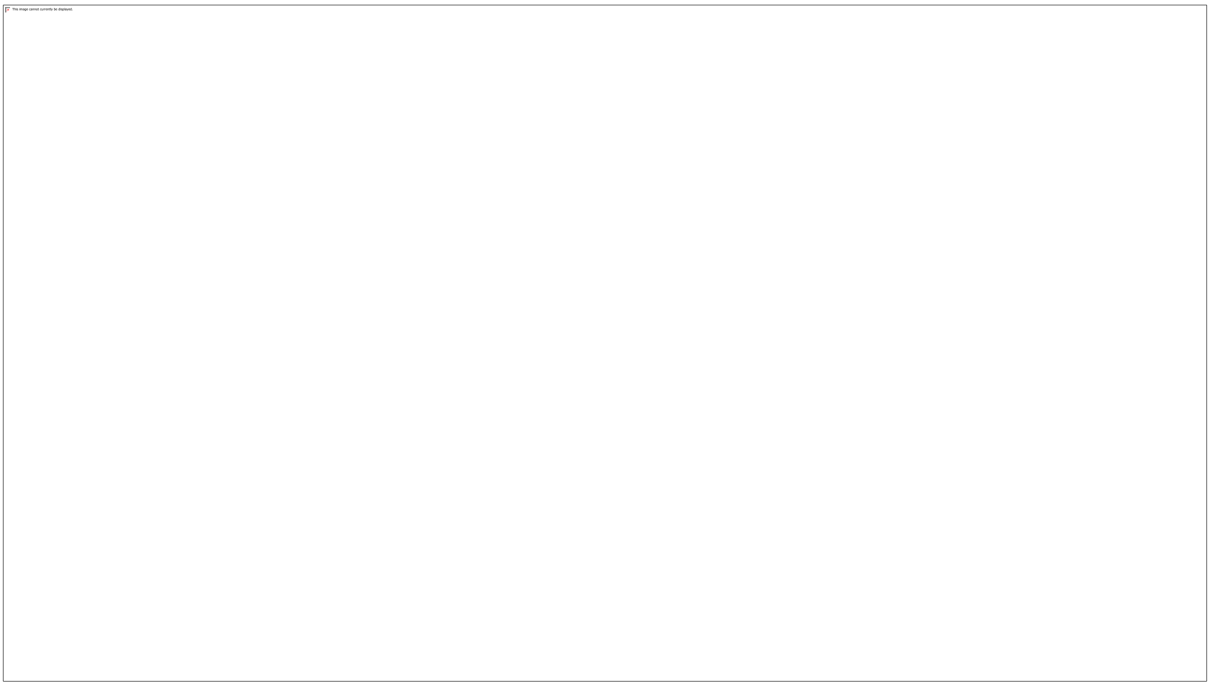


Figure 5.2: Face Recognised

Figure 5.2 depicts Recognition and Face detection. The face is detected using the Haar Cascade algorithm and recognized using the LBPH algorithm. The application is found to detect faces in an input image. If the user is recognized, then highlight the same with a rectangular blue color border together with their name and ID. If the user is not recognized, a message "unknown" will be displayed.

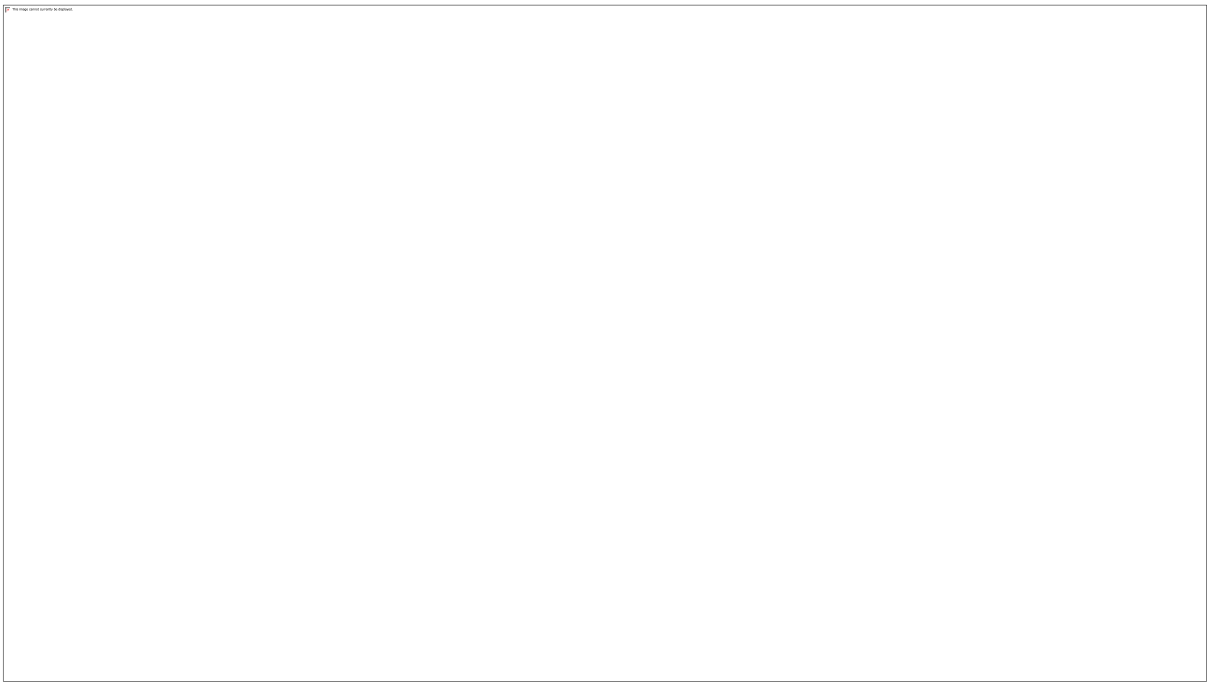


Figure 5.3: Auto-Fill Client ID

Figure 5.3 shows that the client ID is filled. It is automatically filled after the recognition of the image. The client is not supposed to enter it physically.

Figure 5.4: Detect eyes (open) and generate virtual keypad

Figure 5.4 shows the virtual keypad and also detects the open eyes. In the application, eye detection is done by the OpenCV module. The eye is detected as open and shown in a red border.

Figure 5.5: Detect eyes (Closed)

Figure 5.5 shows the blinked eyes. When the eyes are blinked, the red color changes to blue, and at that time the cursor is pointed to the particular number taken as the PIN of that digit. It is generated 4 times because in this application the PIN is 4 digit number.

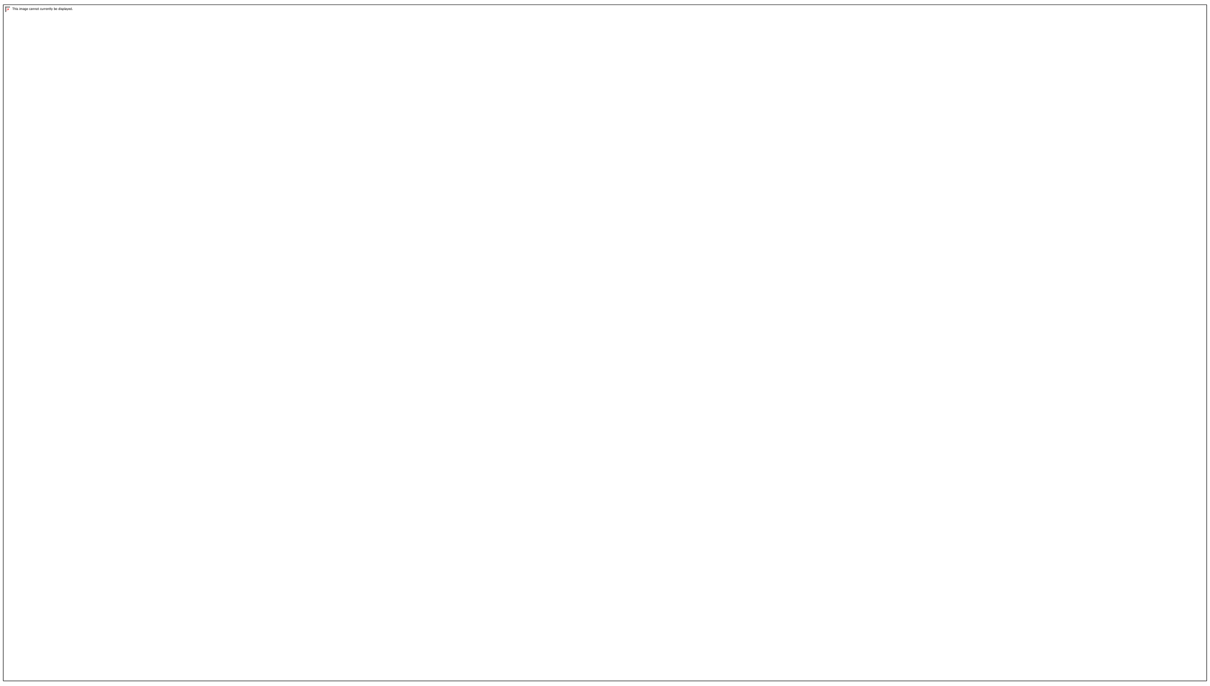


Figure 5.6: Auto fill PIN using Eye Blink

In Figure 5.6, the PIN is filled. After blinking four times, the PIN is taken as input and filled automatically, and then the client has to hit the login button; then the OTP is generated and sent to the registered mobile number.

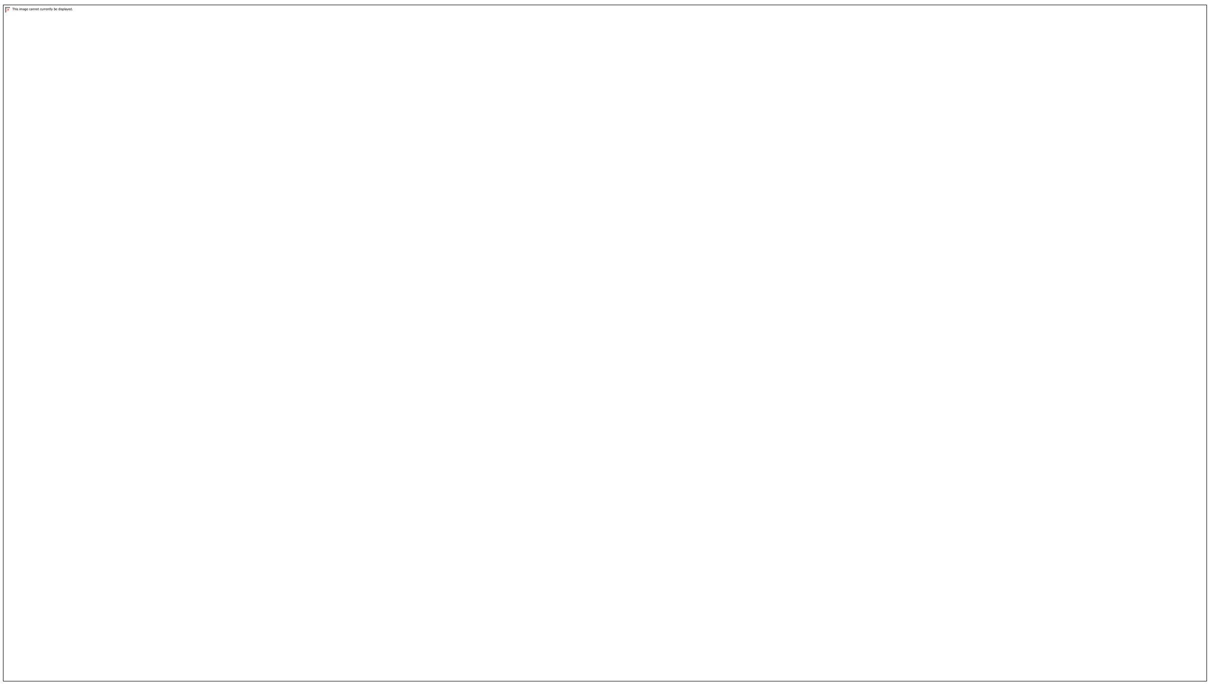


Figure 5.7: Generate OTP.

Figure 5.7 shows that the OTP is sent successfully to the registered mobile number and also the OTP is printed on the terminal.

Figure 5.8: Get OTP on the registered mobile number

Figure 5.8 is a screenshot of the mobile message in which the OTP is received using the FAST2SMS API.

Figure 5.9: Login done

Figure 5.9 shows the login page after the OTP is entered; the system verifies the OTP and sends the client to the login page, where the client details are available.

References

1. R. Revathy and R. Bama, "Advanced Safe PIN Entry Against Human Shoulder-Surfing," IOSR Journal of Computer Engineering, vol 17, issue 4, ver. II, pp. 9-15, July-Aug. 2015. (Available:http://www.iosrjournals.org/iosrjce/papers/Vol17issue4/Version2/B017420915.pdf).
2. Sonia, Robert and Alain, "Input precision for gaze-based graphical passwords". In proceedings of the 28th International Conference on Human Factors in Computer Systems, CHI 2010. (Available:https://www.researchgate.net/publication/221518954_Input_precision_for_gazebased_graphical_passwords.
3. D. Rozado, 2013, "Using Gaze-Based Passwords as an Authentication Mechanism for Password Input", In proceedings of the 17th European Conference on Eye Movements (ECEM) 2013.
4. Drewes, Heiko (2010): Eye Gaze Tracking for Human Computer Interaction. Dissertation, LMU München: Faculty of Mathematics, Computer Science and Statistics[Book].
5. Mun-Kyu Lee (February 2014): "Security Notions and Advanced Method for Human Shoulder-Surfing Resistant PIN-Entry", Published in: IEEE Transactions on Information Forensics and Security (Volume: 9, Issue: 4, April 2014). (Available: https://ieeexplore.ieee.org/abstract/document/6746671).
6. Sota Shimizu, Takumi Kadogawa, Shu-ichi Kikuchi, Takumi Hashizume (March 2014): "Quantitative analysis of tennis experts' eye movement skill", Published in: 2014 IEEE 13th International Workshop on Advanced Motion Control (AMC) 2014.
7. Hari Krishna S M, Gautam Pradyumna, Aishwarya B, Chinamaya Gayathri (April 2021): "Development of Personal Identification Number Authorization Algorithm Using Real- Time Eye Tracking & Dynamic Keypad Generation" Published in: 2021 6th International Conference for Convergence in Technology (I2CT).
8. De Luca, Alexander, Martin Denzel, and Heinrich Hussmann. "Look into my eyes! Can you guess my password?" Proceedings of the 5th Symposium on Usable Privacy and Security. ACM, 2009.
9. Forget, Alain, Sonia Chiasson, and Robert Biddle. "Input precision for gaze-based graphical passwords." CHI'10 Extended Abstracts on Human Factors in Computing Systems. ACM, 2010.
10. Weaver, Justin, Kenrick Mock, and Bogdan Hoanca. "Gaze-based password authentication through automatic clustering of gaze points." Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on. IEEE, 2011.
11. Bulling, Andreas, Florian Alt, and Albrecht Schmidt. "Increasing the security of gaze-based cued-recall graphical passwords using saliency masks." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2012.
12. Rozado, David. "Using gaze-based passwords as an authentication mechanism for password input." 3rd International Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction. 2013.

13. Martin, Mihajlov, Trpkova Marija, and Arsenovski Sime. "Eye tracking recognition-based graphical authentication." Application of Information and Communication Technologies (AICT), 2013 7th International Conference on. IEEE, 2013.
14. Khamis, Mohamed, et al. "Gazetouchpass: Multimodal authentication using gaze and touch on mobile devices." Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems. ACM, 2016.
15. Li, Zhenjiang, et al. "iType: Using eye gaze to enhance typing privacy." INFOCOM 2017-IEEE Conference on Computer Communications, IEEE. IEEE, 2017.
16. https://towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd08 .
17. Preetham G, Sandarsha H.M, Vineeth S, Vipruth L.M, Mouna K.M (June 2021): "Password Authentication Provided by Three-Layer Security Using Machine Learning" Published in International Advanced Research Journal in Science, Engineering and Technology (Vol. 8, Issue 6, June 2021).
18. Akihiro K, Kazu N, Rin H, Hideaki K, Yoshihisa N (January 2022): "Eye fatigue estimation using blink detection based on Eye Aspect Ratio Mapping" [Book].