University of Ibadan
Journal of Science and Logics in ICT Research

Design of a Variable-length Accented Character-based CAPTCHA System

[1]✉Adetunji, A. O., [2]Osunade, O., [3]Olanrewaju, O. T. and [4]Asoro, O. B.

[1,2,4]Department of Computer Science, University of Ibadan, Nigeria
[3]Department of Computer Science, Federal College of Animal Health and Production Technology, Moor Plantation, Apata, Ibadan, Nigeria
[1]adetunjiadekunle8@gmail.com, [2]o.osunade@ui.edu.ng, [3]ayotundetaiwo@gmail.com, [4]raymondblessing5@gmail.com

Abstract

Completely Automated Public Turing test to Tell Humans and Computers Apart (CAPTCHA) is aimed at distinguishing humans from computers by using text, images, audio, or video. Text-based CAPTCHAs, are the most common but face vulnerabilities due to their limited use of latin characters. This study enhanced the design of the NaijaCAPTCHA by generating variable-length accented character-based CAPTCHA codes, utilising latin and accented characters to bolster online transaction security. The design was implemented using Javascript, PHP, HTML, and CSS. The enhanced NaijaCAPTCHA comprised of four modules: CAPTCHA generator, obfuscator, display unit, and database. The generator employed the Gimpy algorithm to create codes of varying lengths (4 to 7 characters) with at least two accented characters. The obfuscator manipulated the code's appearance through color, text distortion, background noise, and skewing. User presentation and authentication occurred in the display unit, with code correctness verified against the obfuscated value. Thirty CAPTCHA categories were generated from the combination of background and text modification. The enhanced NaijaCAPTCHA system presents a formidable challenge to bots, suggesting its potential adoption for government websites and transactional purposes.

Keywords: *CAPTCHA, web security, NaijaCAPTCHA, variable-length characters, cyber-threats*

1. Introduction

Completely Automated Public Turing Test to Tell Humans and Computers Apart, or "CAPTCHA," is a well-known and essential authentication method that is extensively used on the Internet. Its main goal is to carefully identify and validate users' real identities by differentiating between automated computer programs, or "bots," and people. This important distinction is critical to protecting the integrity and security of online platforms because CAPTCHA effectively blocks the illegal activities of bad actors, like spammers and scammers, who use automated bots to perform tasks like creating fake accounts and flooding websites with disruptive or fraudulent submissions. This guarantees that only real human interactions take place, thereby improving the overall dependability and trustworthiness of online platforms.

Securing non-physical assets is one of the most crucial aspects of living in the information age. Several security solutions have approached the issue in different ways, while user authentication has historically involved passwords and biometrics, security flaws persist [1]. The implementation of CAPTCHA, which stands as a dynamic and dependable authentication method, functions as a robust deterrent against individuals who lack authenticity and attempt unauthorised access to web services. The mechanism distinguishes between legitimate human users and automated, robotic entities. An individualised authentication code is generated within the CAPTCHA procedure and presented to the human user, who is required to input it as a verification step to gain access to the requisite resources. In a gesture focused on safeguarding user privacy, CAPTCHAs not only ensure anonymity but also uphold the confidentiality of personal data. To achieve this, CAPTCHAs

Adetunji, A. O., Osunade, O., Olanrewaju, O. T. and Asoro, O. B. (2024). Design of a Variable-length Accented Character-based CAPTCHA System. *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 12 No. 1, pp. 33 – 44.

utilise a versatile array of combinations involving text, voice, and imagery to transmit information that serves as confirmation of the user's genuine human identity.

CAPTCHA systems, like NaijaCAPTCHA, have become essential in the contemporary digital age, where protecting online services has become crucial due to the rising prevalence of automated bots and cyber dangers. These technologies have a pivotal function in upholding the credibility and dependability of digital engagements by distinguishing genuine individuals from automated bots. However, designing a CAPTCHA that is easily understood by humans (usability) but not by a machine (security) at the same time possesses a sweet spot property [2]. In particular, NaijaCAPTCHA has been painstakingly created to provide one-of-a-kind challenges that typically consist of five characters, including two accented characters. This makes it a handy and reliable tool in the ongoing fight against the unlawful and potentially dangerous activities of automated bots in Nigeria's digital environment.

The NaijaCAPTCHA system represents a pioneering CAPTCHA solution engineered with the explicit purpose of bolstering the security framework of online transactions. It achieves this goal through the strategic incorporation of Latin and accented characters drawn from two indigenous Nigerian languages, namely Yoruba and Igbo, collectively known as NaijaCAPTCHA. The architectural essence of this innovative approach lies in the utilisation of Latin and accented characters to construct CAPTCHA codes, a process underpinned by a refined and customised Gimpy algorithm. Nonetheless, the current configuration of the system, characterised by the generation of CAPTCHA challenges featuring fixed text lengths, introduces thought-provoking considerations that warrant exploration and refinement.

The existing NaijaCAPTCHA system is intricately designed to generate CAPTCHA codes with a fixed length of precisely five characters and incorporating two accented characters into its composition. While this configuration has thus far effectively fulfilled its intended purpose, it precipitates a pivotal question concerning the system's resilience and operational integrity under potential adjustments to the CAPTCHA length, be it an expansion beyond five characters or a reduction. The question of how much these changes might strengthen or weaken the security and overall efficacy of the system is yet unanswered and needs more investigation and study. This research work aims to design an enhanced CAPTCHA system called NaijaCAPTCHA to allow the generation of CAPTCHA codes with 4 to 7 characters including accented characters.

Web security is the cornerstone of confidence in digital interactions, and the problem of automated bots will never go away. This research is significant because it aims to enhance the current NaijaCAPTCHA system, which is well-known for its role in web security. When a word in the CAPTCHA has a predetermined character length, character locations may be easily anticipated. However, if a random amount of characters is used, this prediction could become difficult [3].

2. Related Works

Non-English language CAPTCHAs offer a beneficial solution for websites and platforms serving diverse linguistic communities. By presenting challenges in languages besides English, these CAPTCHAs guarantee accessibility and ease of use for users who are not fluent in English or prefer their native language.

A significant advantage of non-English language CAPTCHA’s is their inclusiveness. They accommodate users from various linguistic backgrounds, enhancing their experience by presenting challenges in languages they understand proficiently. This inclusivity cultivates a sense of belonging and engagement among users, resulting in increased participation rates and enhanced interaction with online platforms.

Moreover, non-English language CAPTCHAs enhance security by introducing an additional layer of complexity for automated bots attempting to circumvent security measures. Because these CAPTCHAs rely on language-specific challenges, they create additional obstacles for bots programmed to recognise only English-based CAPTCHAs. This aids in deterring automated attacks and safeguarding against spam, fraud, and other malicious activities which is the key to why several researchers proposed and developed a working

non-native language CAPTCHA system that is easier for human usability and difficult for bots or automated programs to break.

Analysis of recent advancements in CAPTCHA development underscores the challenge of creating a CAPTCHA that effectively integrates security with usability. Numerous CAPTCHAs have been breached, underscoring the challenge of attaining top-notch security while maintaining ease of use. Studies suggest that CAPTCHAs utilising non-native languages could provide improved usability for internet users fluent in those languages [2].

Olanrewaju and Osunade [4] proposed NaijaCAPTCHA to strengthen security and increase accessibility for non-native English users, accented characters were added to the NaijaCAPTCHA system. This system aimed to broaden the user base by accommodating individuals who are more familiar with languages featuring accented characters, thus enhancing inclusivity and usability. Integrating accented characters not only strengthens the security of the CAPTCHA system but also reflects a commitment to catering to diverse linguistic communities Testing of the accented CAPTCHA system in a controlled setting yielded favorable outcomes. These accented characters are prevalent in various human languages, including Yoruba, Igbo, Latin, and French. Their integration into the system enhanced the acceptability and diversity of CAPTCHA authentication.

Alsuhibany and Parvez [12] presented a method for creating Arabic handwritten CAPTCHA images. These CAPTCHAs were generated using a subset of 123,200 prewritten Arabic character images, extracted from the KHATT database. The process outlined in the paper involved generating Arabic handwritten CAPTCHA images using prewritten Arabic character images. The steps included selecting a prewritten Arabic character image, transforming it into a binary image, segmenting the binary image, estimating the baseline of the character, and distorting the segmentation locations. The steps aimed to create CAPTCHAs that challenge OCR algorithms by intentionally distorting the characters' segmentation locations, estimating baselines, introducing color and noise distortion, and applying rotations. This approach enhanced security by making CAPTCHA recognition challenging for bots while maintaining human usability accuracy above 88%.

Kehar *et. al.* [5] proposed the creation of Sindhi text-based CAPTCHAs tailored for regional websites. The CAPTCHA design mirrored the Arabic language, written right to left, and featured 62 characters, incorporating coloured ellipses and clutter while avoiding noise by using dots in characters. The Sindhi Text CAPTCHA image was developed using the C# programming language. Deliberate overlapping of characters was employed to challenge CAPTCHA OCR programs in segmenting the string effectively. The character string ranged randomly from 3 to 8 characters. The implementation was tested on web pages created using ASP and JSP, with evaluation conducted by users.

Shirali-Shahreza and Shirali-Shahreza [6] introduced the "Advanced Nastaliq CAPTCHA" method, which utilises Arabian and Persian characters of varying lengths, typically between three to eight characters. This approach employed the Nastaliq font and generates images in PNG format. Since the letters are interconnected and more than half feature dots, no additional noise or distortion is necessary in the image. The right-to-left orientation of the text adds complexity to recognition by OCR programs. Users view the image and their input is compared to the string to determine if the test result is a pass or fail. This approach was executed utilising the JAVA programming language.

3. Methodology

This study was conducted through an experiment, employing a quanlitative data. The initial phase involved the design and testing of the Variable-Length Accented Character-Based CAPTCHA system on a local web server. The conceptualisation of this variable-length CAPTCHA system, centered on accented characters, draws its foundation from the Gimpy scheme, a proven and established model. Detailed insights into the system's architecture are visually presented in Figure 1. At its core, this model is structured around four pivotal modules, each playing a distinct role in the system's functionality. These modules are the CAPTCHA Generator, Obfuscator, CAPTCHA Display, and Database.

CAPTCHA GENERATOR is designed with sets of defined algorithms. This algorithm randomly generates varied lengths of characters at random from the stored characters to form a string ranging from 4 to 7 characters that form the Accented CAPTCHA code. This module uses the algorithm shown in Algorithm 1. The algorithm uses Latin characters, and Accented characters (both in uppercase and lowercase) to formulate the variable length CAPTCHA code. These set of fifty-one (51) accented characters are used within the system: Á, á, À, à, Â, â, Ă, ă, Ä, ä, Å, å, Ã, ã, Ç, ç, É, é, È, è, Ẹ, ẹ, Ê, ê, Ĕ, ĕ, Ệ, ệ, Ë, ë, Ė, ė, Ĩ, ĩ, İ, i, Í, í, Ì, ì, Î, î, Ǐ, ǐ, Ï, ï, Ḿ, ḿ, Ṃ, ṃ, Ṁ, ṁ, Ń, ń, Ǹ, ǹ,, Ñ, ñ, Ṅ, ṅ, Ó, ó, Ò, ò, Ọ, ọ, Ô, ô, Ǒ, ǒ, Ö, ö, Õ, õ, Ō, ō, Ú, ú, Ù, ù, Û, û, Ü, ü, Ů, ů, Ũ, ũ, Ụ, ụ, Ṡ, ṡ, Ỹ, ỹ, Ŷ, ŷ, Ṣ, ṣ, Ý, ý, Ÿ, ÿ.

The CAPTCHA generator is configured to randomly generate varied lengths of CAPTCHA code ranging from four (4) to seven (7) characters in the following specified order:

- Minimum length of four characters including two (2) number of accented characters.
- Length of five (5) characters with two (2) accented characters included.
- Length of six (6) characters with two (2) accented characters included.
- Length of seven (7) characters with two (2) accented characters included.

Step 1**:** Begin the session
Step 2a: Generate two (2) accented characters at random from the characters stored –
Latin and Á, á, À, à, Â, â, Ă, ă, Ä, ä, Å, å, Ã, ã, Ç, ç, É, é, È, è, Ẹ, ẹ, Ê, ê, Ĕ, ĕ, Ệ, ệ, Ë, ë, Ė, ė, Ĩ, ĩ, İ, i, Í, í, Ì, ì, Î, î, Ǐ, ǐ, Ï, ï, Ḿ, ḿ, Ṃ, ṃ, Ṁ, ṁ, Ń, ń, Ǹ, ǹ, Ñ, ñ, Ṅ, ṅ, Ó, ó, Ò, ò, Ọ, ọ, Ô, ô, Ǒ, ǒ, Ö, ö, Õ, õ, Ō, ō, Ú, ú, Ù, ù, Û, û, Ü, ü, Ů, ů, Ũ, ũ, Ụ, ụ, Ṡ, ṡ, Ỹ, ỹ, Ŷ, ŷ, Ṣ, ṣ, Ý, ý, Ÿ, ÿ
Step 2b: Store the result
Step 3a: At random, generate varied string lengths between 2 to 5 characters from the stored characters
Step 3b: Store the result Step 4a: Fetch the result in Step 2b
Step 4b: Fetch result in Step 3b
Step 4c: Join content in Step 4a and 4b Step 5: Randomize the Generated String
Step 6: Randomly select a varied-length CAPTCHA type
Step 7: Initialize an empty canvas for drawing CAPTCHA
Step 8: Display the randomly generated characters on the canvas
Step 9: Randomly generate the canvas background and image
Step 10a: Accept user input from the keyboard
Step 10b: If the user input equals the generated displayed characters:
Step 10bi: Save the user response into the database
Step 10bii: Display the "Your Input Matched Correctly" message to the user Step 10c: Else,
Step 10ci: Save the user response into the database
Step 10cii: Display **"**Your Input Doesn**'**t Match" message to the user
Step 10ciii: Go to Step 1 (Begins the session)

Algorithm 1: Variable-Length Accented Character-Based CAPTCHA

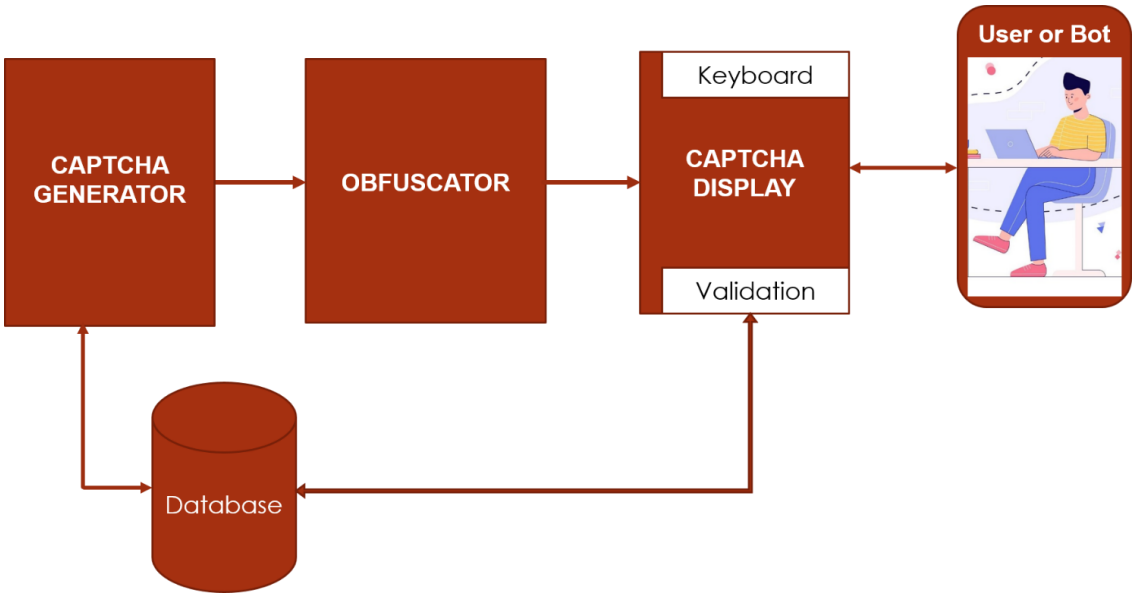

Figure 1: Variable-Length Accented Character-Based CAPTCHA System

OBFUSCATOR is a security mechanism that generates complex variable length accented CAPTCHA codes to make it difficult for automated programs or bots to decode. Algorithm 2 produces the dynamic varied length accented CAPTCHA code with various deformities which include text color manipulation, text trimming, text skewing, text squash, background noise, and manipulation techniques to create a barrier against automated decryption attempts. The module's strategic indifference to the sequence of deformity application adds an extra layer of complexity, making it difficult for potential attackers to predict or adapt to specific patterns.

CAPTCHA Display presents to the user the varied length accented CAPTCHA code generated on screen and accepts their input. Within this module, three processes are implemented: The Keyboard is embedded with the on-screen display of the CAPTCHA code generated by the system. This virtual keyboard adheres to the familiar QWERTY standard keyboard format. Integrated into the display module encompasses a comprehensive character set that includes Latin characters, Extended ASCII characters, and specifically, accented characters representing Nigerian languages.

This design emphasizes user-friendly input while catering to the linguistic nuances of Nigerian languages through the inclusion of accented characters. The Validation Unit is implemented to carry out two major operations that include response checking and response matching. Response Checking Unit initiates a meticulous scanning process, focusing on the CAPTCHA input field, and anticipating user responses. It employs a sophisticated scanning protocol to detect and capture user responses, contributing to the system's dynamic engagement with user interactions.

Response Matching Unit receives the user's input and undergoes scrutiny by comparing it against the record in the database to ascertain equality. The Database is responsible for storing the character sets used by the Variable Length Accented Character-Based CAPTCHA Code Generator, as well as the generated CAPTCHA codes and their corresponding response times by the users or bots. The Validation unit accesses this module to check, verify, and match the response provided by users or bots.

Step 1a: Accept the generated string

Step 1b: Deform the characters generated in the string at random

(Trimming, Scaling, Rotate, Skew, Squash)

Step 2: Implement color feature for text and background at random

Step 3: Select a background image at random (patterns, gradient, transparent, lines, and noise)

Step 4: Combined the randomly deformed string and background image generated

Step 5: Produce the variable-length accented CAPTCHA image

Step 6: Save a copy of the produced variable-length accented CAPTCHA image to the Database

Algorithm 2: Obfuscation Algorithm for Variable-Length Accented Character-based CAPTCHA System

4. Results and Discussion

The enhanced NaijaCAPTCHA system, E-NaijaCAPTCHA, was designed on a laptop system equipped with hardware specifications including an Intel(R) Core(TM) i5 with (Processor: 2.50GHz, and 1TB HDD storage). The investigation utilised a variety of software components such as Windows 10, JavaScript, PHP, HTML, CSS, MYSQL, and XAMPP Server. The assessment of the created CAPTCHA system centered on variable-length accented characters, was conducted across various web browsers on both Windows and Android Phones. These browsers encompass Google Chrome, Mozilla Firefox, Opera Mini, Phoenix, Hola Browser, Safari and Microsoft Edge.

The E-NaijaCAPTCHA system yielded thirty (30) distinct categories of CAPTCHA codes upon implementation. These codes consist of two primary components: characters and background. The background comprises bitmap images or canvases where the generated characters are embedded. There are five (5) types of backgrounds: colored background, random lines, background with noise, gradient background, and plain background (no background). The randomly generated characters can undergo various modifications such as squashing, coloring, fragmentation, distortion, skewing, or collapsing. A detailed breakdown of these components and their distinguishing features is presented in Table 1. The combination of different backgrounds and character formats resulted in thirty (30) unique CAPTCHA categories as illustrated in Table 2.

Table 1: Components of the Generated CAPTCHA Categories

Random Background	**Randomly Generated Characters**
Gradient	Colored
Random Line	Squashed
Background with Noise	Fragmented
Colored Background	Distorted
Plain Background	Skewed
	Collapsed

Table 2: List of the Possible CAPTCHA Categories

S/N	**CAPTCHA Category**	**Acronym**	**S/N**	**CAPTCHA Category**	**Acronym**
1	Character Fragmentation with Random Lines	CFRL	16	Character Fragmentation No Background	CFNB
2	Text No Background	TNB	17	Colored Text with Random Lines	CTRL
3	Character Squash with Colored Background	CSCB	18	Character Collapse with Colored Background	CCCB
4	Character Squash with Background Noise	CSBN	19	Character Fragmentation Colored Background	CFCB
5	Character Collapse Background Noise	CCNB	20	Text with Background Noise	TBN
6	Text with Gradient Background	TGB	21	Colored Text No Background	CTNB
7	Text Distortion Background Noise	TDBN	22	Character Fragmentation with Gradient Background	CFGB
8	Character Fragmentation with Background Noise	CFBN	23	Colored Text Colored Background	CTCB
9	Text with Colored Background	TCB	24	Character Collapse Background Noise	CCBN
10	Text Distortion with Colored Background	TDCB	25	Text with Random Lines	TRL

11	Character Squash with Gradient Background	CSGB	26	Character Squash with Random Lines	CSRL
12	Character Squash No Background	CSNB	27	Text Distortion with Random Lines	TDRL
13	Character Collapse with Random Lines	CCRL	28	Character Collapse Gradient Background	CCGB
14	Text Distortion with No Background	TDNB	29	Colored Text Background Noise	CTBN
15	Text Distortion with Gradient Background	TDGB	30	Colored Text with Gradient Background	CTGB

Table 3a: Feature Comparison of Variable-Length Accented CAPTCHA Categories

CAPTCHA Category	**Color**	**Squash**	**Fragme-ntation**	**Distortion**	**Skew**	**Collapse**	**Lines**	**Background**
CFRL	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TNB	Yes	Yes	Yes	Yes	Yes	Yes	No	No
CSCB	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
CSBN	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
CCNB	Yes	Yes	Yes	Yes	Yes	Yes	No	No
TGB	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
TDBN	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
CFBN	Yes	Yes	Yes	No	No	Yes	No	Yes
TCB	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
TDCB	Yes	Yes	Yes	Yes	No	Yes	No	Yes
CSGB	Yes	Yes	Yes	No	Yes	Yes	No	Yes
CSNB	Yes	Yes	Yes	Yes	Yes	Yes	No	No
CCRL	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TDNB	Yes	Yes	Yes	Yes	Yes	Yes	No	No
TDGB	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes

Table 3b: Feature Comparison of Variable-Length Accented CAPTCHA Categories

CAPTCHA Category	Color	Squash	Fragme-ntation	Distortion	Skew	Collapse	Lines	Background
CFNB	Yes	Yes	Yes	Yes	Yes	Yes	No	No
CTRL	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
CCCB	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
CFCB	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
TBN	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
CTNB	Yes	Yes	Yes	Yes	Yes	Yes	No	No
CFGB	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
CTCB	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
CCBN	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
TRL	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
CSRL	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TDRL	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
CCGB	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
CTBN	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
CTGB	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes

Figure 2: Accented Characters Designed User Virtual Keyboard

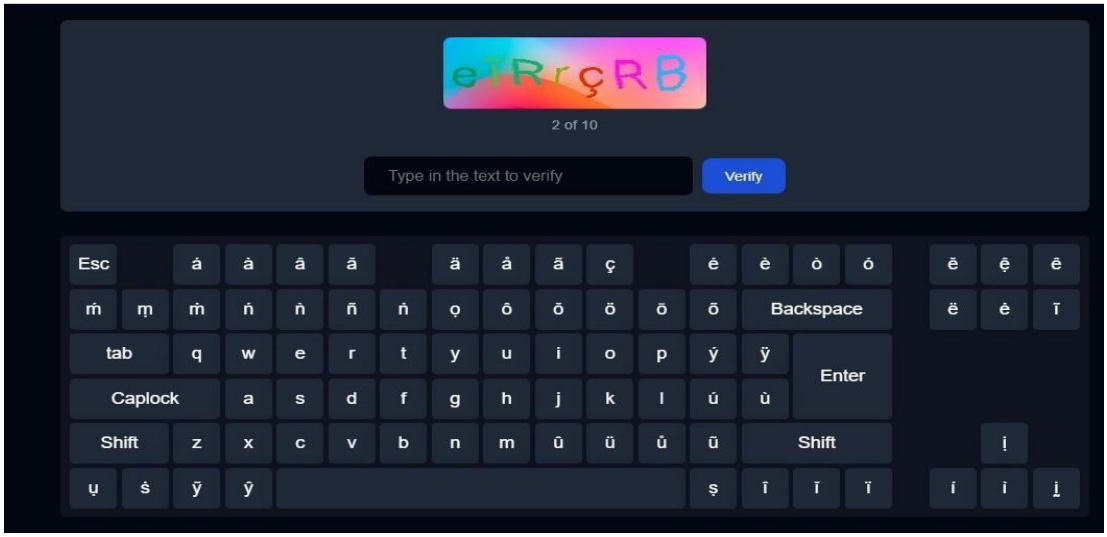

Figure 3: User Virtual Keyboard with the Generated E-NaijaCAPTCHA Code Randomly Generated Varied-Length of Accented Codes

	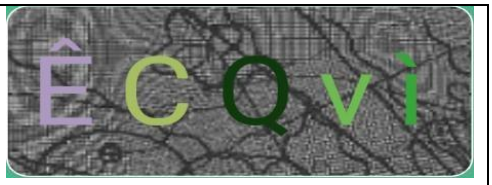
4 Lengths	**5 Lengths**
	
6 Lengths	**7 Lengths**

Figure 4: Randomly Generated Variable-Length Accented CAPTCHA Code

4.1 Discussion

E-NaijaCAPTCHA is compatible with Windows, Linux, and Android operating systems thus offering versatile implementation options. The source code for E-NaijaCAPTCHA is accessible and compatible with various web browsers, including Google Chrome, Mozilla Firefox, Opera Mini, Phoenix, Hola, Safari and Microsoft Edge, ensuring widespread usability and accessibility.

The development of the E-NaijaCAPTCHA system was to satisfy the curiosity of what happens when the length generated is increased or decreased, and the addition of some other security features on the existing NaijaCAPTCHA system presented by Olanrewaju & Osunade [4] and Olanrewaju *et. al.* [1]. Pate and Ramteke [7] developed a new text-based CAPTCHA system called Devanagari CAPTCHA that generates varying lengths (5 to 7) of text using printed and handwritten Devanagari characters and numeral combinations. Specifically, the system was designed for use in the Urdu language by Abbas *et. al.* [8]. It creates CAPTCHA strings at random, with a character length of four to eight (4 to 8). This system is specifically designed to cater to regional Urdu websites. However, the existing NaijaCAPTCHA generates only a fixed length of five (5) characters at random. Chandavale and Sapkal [9] suggested that the lesser the character set size and the string length, the higher the chances of random guessing of the CAPTCHA code. The longer the string used in CAPTCHA system development, the more secure it will be. The E-NaijaCAPTCHA system

was developed to randomly generate variable-length accented characters within the range of 4 to 7 lengths.

The E-NaijaCAPTCHA operates on a foundation of two distinct components, each contributing unique values to produce thirty distinct CAPTCHA categories. This indicates the potential for generating a significantly larger variety of CAPTCHA types by introducing additional features into these two primary components: the formats of generated characters and the backgrounds utilised. Expanding the range of values within these components offers the system greater flexibility and diversity in creating CAPTCHA challenges, thereby enhancing its effectiveness in thwarting automated attacks. The security of text-based CAPTCHA systems primarily relies on visual interference mechanisms, such as rotation, twisting, adhesion, and overlap [10].

These techniques introduce complexities and distortions to the text elements, enhancing the system's resilience against automated attacks. The CAPTCHA categories produced are elaborated, delineating the distinctive attributes of each component amalgamated to create the category. These categories are engineered to yield character lengths ranging from 4 to 7, with two of any generated length comprising accented characters. The adoption of variable-length generation by E- NaijaCAPTCHA contrasts with the methodology employed by [4].

A comprehensive analysis of the generated CAPTCHA categories is conducted, evaluating six distinct features: squashing, color, collapse, distortion, fragment, and pattern, to ascertain the composition of each type. It is observed that not all categories exhibit identical features; rather, only six CAPTCHA categories encompass all the aforementioned attributes.

Additionally, the integration of a non-Latin virtual keyboard within the E-NaijaCAPTCHA system significantly facilitates user interaction with the CAPTCHA challenge. This keyboard remained unaltered, only the non-essential keys repurposed to accommodate the fifty-one (51) accented characters embedded on it. The keyboard is designed to respond to both the Upper Case and Lower case character functions in the same way it was programmed within the developed system which is one of the uniqueness that set E-NaijaCAPTCHA out from its predecessor. The provision of a familiar keyboard layout contributes to standardising the data entry process, enhancing user convenience and efficiency.

Table 4 compares the features of the newly developed variable-length accented character-based CAPTCHA system called E-NaijaCAPTCHA with the NaijaCAPTCHA and shows its advancement over the existing. Compared to the existing NaijaCAPTCHA system, the Enhanced-NaijaCAPTCHA system outperforms the existing one in terms of the character lengths, CAPTCHA category, the number of accented characters implemented, and the character case (Upper Case and Lower Case) that can be generated at random by the system which are the keys to a secured CAPTCHA system against any automated programs or bots [11].

Table 4: Compararing features of E-NaijaCAPTCHA with the NaijaCAPTCHA

CAPTCHA System Model	**Character Length**	**Generated CAPTCHA Category**	**Implemented Accented Characters**	**Latin Character Case**	**Accented Character Case**
E-NaijaCAPTCHA	Variable-Length (4 –7)	30	51	Upper & Lower Case	Upper & Lower Case
NaijaCAPTCHA	Fixed Length (5)	16	20	Upper & Lower Case	Lower Case Only

5. Conclusion

This research work created a variable-length accented characters system by enhancing the design of an existing accented character CAPTCHA system. The enhanced system integratesd random generation of variable-length accented characters ranging (4 to 7) to improve web security. By producing thirty unique CAPTCHA categories featuring diverse backgrounds and character modifications, the E-NaijaCAPTCHA system emerges as a robust and dependable solution for safeguarding web platforms against unauthorised access. It is recommended that the E-NaijaCAPTCHA system should be utilised by Nigerian industries and digital companies like Jumia, SLOT, and Konga for transactions conducted on their online platforms. E-NaijaCAPTCHA can be used in conjunction with Digital Rights Management (DRM) protocols to safeguard digital contents; and utilised as a verification code for commenters on social media platforms of Nigerian enterprises. Further exploration and investigation identified are development of an enhanced NaijaCAPTCHA system that incorporates handwritten characters in the generation of variable-length accented characters and enhancing E-NaijaCAPTCHA's security against evolving cyber threats with advanced measures like biometric authentication and AI-driven anomaly detection.

References

[1]. Olanrewaju, O. T., Omilabu, A. A., Asoro, B. O., Nwufor, C. V., Adewale, F. O., and Osunade, O. (2023). Design of an Accented Character-Based Text CAPTCHA System. International Journal of Computer Science and Information Security (IJCSIS), Vol. 21, No. 7: 91-100. https://zenodo.org/record/8254124

[2]. Kumar, M., Jindal, M. K., and Kumar, M. (2022). Design of innovative CAPTCHA for Hindi language. In Neural Computing and Applications, Volume 34, Issue 6: 4957–4992. Springer London. https://doi.org/10.1007/s00521-021-06686-0

[3]. Alsuhibany, S. A., Parvez, M. T., Alrobah, N., Almohaimeed, F., and Alduayji, S. (2017). Evaluating robustness of Arabic CAPTCHAs. 2nd International Conference on Anti-Cyber Crimes (ICACC). doi:10.1109/anti-cybercrime.2017.7905268

[4]. Olanrewaju, O. T., and Osunade, O. (2017). Design of Accented Character-based CAPTCHA with Usability Test for Online Transactions. Zambia ICT Journal, 1(1), 20–24. https://doi.org/10.33260/zictjournal.v1i1.14

[5]. Kehar, A., Hussain, R., Shaikh, R., Shah, S., Khoso, F., Dahar, I., Jiskani, A., Fatima, S., and Shaikh, H. (2021). Design and Development of Sindhi Text Based CAPTCHAs for Regional Websites. Sukkur IBA Journal of Emerging Technologies, Volume 4, Number 1: 34-44. https://doi.org/10.30537/sjet.v4i1.877

[6]. Shirali-Shahreza, M. H., and Shirali-Shahreza, M. (2008). Advanced Nastaliq CAPTCHA. 7th IEEE International Conference on Cybernetic Intelligent Systems, London, UK, 2008, pp. 1-3, https://doi.org/10.1109/UKRICIS.2008.4798962

[7]. Pate, S. E., and Ramteke, R. J. (2023). Design and Generation of Devanagari Script CAPTCHA:Imaginative Technique. Proceedings of the First International Conference on Advances in Computer Vision and Artificial Intelligence Technologies (ACVAIT 2022), pp:356-381, Atlantis Press International BV. https://doi.org/10.2991/978-94-6463-196-8_28

[8]. Abbas, F., Rajput, U., and Dahar, I. (2020). Enhancing Security of Urdu Language Websites through Urdu CAPTCHA. International Journal of Computer Science and Network Security, Volume 20 Number 11, 142–152. http://paper.ijcsns.org/07_book/202011/20201118.pdf

[9]. Chandavale, A.A.,and Sapkal, A. (2012). Security Analysis of CAPTCHA. In: Thampi, S.M., Zomaya, A.Y., Strufe, T., Alcaraz Calero, J.M., Thomas, T. (eds) Recent Trends in Computer Networks and Distributed Systems Security. SNDS 2012. Communications in Computer and Information Science, vol 335. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-34135-9_10

[10]. Chen, J., Luo, X., Guo, Y., Zhang, Y., and Gong, D. (2017). A Survey on Breaking Technique of Text-Based CAPTCHA. Security and Communication Networks, 1–15. https://doi.org/10.1155/2017/6898617

[11]. Chen, J., Luo, X., Liu, Y., Wang, J., and Ma, Y. (2019). Selective Learning Confusion Class for Text-based CAPTCHA Recognition. IEEE Access, 1-1, Volume 4:1-14. https://doi.org/10.1109/ACCESS.2019.2899044

[12]. Alsuhibany, S. A., and Parvez, M. T. (2016). Secure Arabic Handwritten CAPTCHA Generation Using OCR Operations. 15th International Conference on Frontiers in Handwriting Recognition (ICFHR). doi:10.1109/icfhr.2016.0035