

**DEVELOPMENT OF A VARIABLE-LENGTH ACCENTED CHARACTER-BASED  
CAPTCHA SYSTEM**

**BY**

**ADETUNJI ADEKUNLE OLADELE  
MATRIC NO: 232229**

**SUPERVISED BY: DR O. OSUNADE**

**A DISSERTATION SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE,  
FACULTY OF SCIENCE, UNIVERSITY OF IBADAN, IBADAN, NIGERIA**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE  
DEGREE OF MASTER OF SCIENCE (M.Sc.) IN COMPUTER SCIENCE**

**APRIL, 2024**

## **CERTIFICATION**

I certify that this research work was carried out by ADETUNJI ADEKUNLE OLADELE with Matriculation Number 232229 in the Department of Computer Science, Faculty of Science, University of Ibadan, Ibadan, Nigeria.

---

**SUPERVISOR**

**Dr O.Osunade**

**Department of Computer Science,  
University of Ibadan.**

---

**DATE**

---

**H.O.D.**

**Prof. A.B. Adeyemo**

**Head of Department,  
Computer Science,  
University of Ibadan**

---

**DATE**

## **DEDICATION**

This project work is dedicated to the Most High God for His goodness and mercy endureth forever. I also dedicated it to My dear mother Late (Mrs) Oluwatoyin Adunni Adetunji. May your soul continue to rest in the Lord.

## ACKNOWLEDGEMENTS

The prospects and success of any man in life cannot but be traced to the impact of another man. My stayed in school together with the completion of this project work was as a result of the impact of some notable personality in my life which needs to be acknowledged.

Foremost on the list is the divine intervention of God in my life for His goodness and mercy endures forever, to Him all the glory, honour, and adoration.

I appreciate the efforts of my supervisor Dr O. Osunade who have been a source of inspiration to me for his devoted time to guide, read the research workand offered useful comment and criticism which greatly help me towards the completion of this project, may God continue to bless and surprise you richly where you least expects. My appreciation also goes to my Lecturers and the entire staff members of the Department of Computer Science, Facaulty of Science, University of Ibadan, Ibadan. May the Lord continue to be with you at all days of your life.

Gold, silver I don't have, but what I have I will give. I appreciate my dear mother Mrs Toyin Adunni Adetunji though she's late but her good works on earth keeps speaking great. Her best efforts to give me the greatest legacy in life is highly manifesting; also, her cares, financial assistance, and spiritual supports rendered while alive were unquantifiable. Mum, I really miss you, may yur soul continue to rest in peace. I cannot do but appreciate as I am so much delighted for the best understanding, perseverance, efforts, and contributions of my lovely wife Mrs Adetunji Blessing Grace, and my amazing, intelligent, charming Kings Martins Daniel & Richard Phillip Adetunji. As you know, Daddy really loves you guys.

I also extend appreciation to my siblings: Mr Adewale, Mrs Adejoke, and Miss Adeola Adetunji for their level of understanding and supports during the cause of my study, may God in His unending mercy grants you all your good heart desires.

I am highly grateful for the tremendous contribution of a great brother whom God gave to me during the cause of this research Mark Izuchukwu Okechukwu. Your wealth of knowledge I cannot take for granted. I pray God connect you divinely to your destiy helper. Thank you for all you do. Also, my profound appreciation goes to my amazing student colleagues whom has turned friends, Mr Ojo Oluwaseun, Miss Akinlonu Adebisi, Josiah Olatunbosun, Fatunmbi Olatunde, Daniel Ogbomida, and all 2021/2022 M.Sc. students that I could not mention their names. God

Almighty will guide us all through the right path to greatness. You guys are indispensable. I wish us all the best that life has in store for us.

## TABLE OF CONTENTS

Title Page	i
Certification	ii
Dedication	iii
Acknowledgment	iv
Table of Contents	vi
Abstract	xii

### CHAPTER ONE: INTRODUCTION

1.1	Background of the study	1
1.2	Problem Statement	2
1.3	Aim and Objective	3
1.4	Scope of the study	3
1.5	Research Justification	3
1.6	Glossary of Terms	5
1.7	Layout of the Project	5

### CHAPTER TWO: LITERATURE REVIEW

2.1	Security	6
2.1.1	Triad of Security	6
2.1.2	Types of Security	7
2.1.3	Threats to Security	8
2.2	Authentication	9
2.2.1	Types of Authentication	9
2.2.2	Current Authentication Techniques	10
2.2.2.1	Static Authentication	10
2.2.2.2	Dynamic Authentication	10
2.2.2.2.1	One-Time Password (OTP)	10
2.2.2.2.2	Challenge-Handshake Authentication Protocol (CHAP)	10
2.2.2.2.3	CAPTCHA	11
2.3	Completely Automated Public Turing Test to Tell Computers and Humans Apart	11

2.3.1	History of CAPTCHA	12
2.3.2	Types of CAPTCHA	13
2.3.2.1	Text-Based CAPTCHA	13
2.3.2.2	Image-Based CAPTCHA	16
2.3.2.3	Sound-Based CAPTCHA	18
2.3.2.4	Video-Based CAPTCHA	20
2.3.2.5	Puzzle-Based CAPTCHA Type	22
2.3.3	Advantages and Disadvantages of CAPTCHA Types	24
2.3.4	CAPTCHA Threats	26
2.3.5	Breaking CAPTCHA System	26
2.4	Character Set	27
2.4.1	Accented Symbols	27
2.5	Language	27
2.5.1	Yoruba Language	28
2.5.1.2	Yoruba Character Set	29
2.5.2	Igbo Language	29
2.6	Review of Non-English Language CAPTCHA	30
2.6.1	NaijaCAPTCHA System	30
2.6.2	Arabic CAPTCHA	31
2.6.3	Sindhi Text-Based CAPTCHA	31
2.6.4	Advanced NAstsaliq CAPTCHA	32

### **CHAPTER THREE: METHODOLOGY**

3.1	Research Design	33
3.2	Variable-Length Accented Character-Based CAPTCHA System	33
3.2.1	CAPTCHA Generator	35
3.2.2	System Obfuscator	37
3.2.3	CAPTCHA Display	39
3.2.4	The Database	41
3.3	Performance Evaluation	41
3.3.1	Usability Tests	41

3.3.2	Security Evaluation	42
3.3.3	Evaluation Metrics	44

## **CHAPTER FOUR: RESULTS AND DISCUSSION**

4.1	System Implementation	45
4.2	The Developed System Code Generation	45
4.2.1	Feature Comparison of Generated CAPTCHA Categories	51
4.2.2	Accented User Keypad	54
4.2.3	Randomly Generated Varied-Length of Accented CAPTCHA Codesd	57
4.3	Usability Test Results	59
4.3.1	Success Rate	63
4.3.2	Response Time	67
4.3.3	Solving Time	71
4.3.4	Accuracy	75
4.4	Security Evaluation	79
4.4.1	Security Features of Variable-Length Accented CAPTCHA Categories	80
4.4.2	The CAPTCHA Solver	83
4.4.2.1	CAPTCHA Solver's Success Rate	86
4.4.2.2	Accuracy Report	89
4.5	Discussion	92
4.5.1	Comparative Study of E-NaijaCAPTCHA and NaijaCAPTCHA Systems	94

## **CHAPTER FIVE: SUMMARY, CONCLUSION, AND RECOMMENDATION**

5.1	Summary	96
5.2	Conclusion	96
5.3	Recommendation	97
5.4	Contributions to Knowledge	97
5.5	Future Recommendation	98
	REFERENCES	99
	APPENDIX	104



## **LIST OF FIGURES**

2.1	Examples of Image-Based CAPTCHA	17
2.2	Sound-Based CAPTCHA	19
2.3	Video-Based CAPTCHA	21
2.4	Puzzle-Based CAPTCHA	23
3.1	Variable-Length Accented Character-Based CAPTCHA System	34
4.1	Accented Characters Designed User Virtual Keyboard	55
4.2	User Virtual Keyboard with the Generated E-NaijaCAPTCHA Code	56
4.3	Randomly Generated Variable-Length Accented CAPTCHA Code	58
4.4	Accented CAPTCHA Category Occurrences During Usability Test	62
4.5	Successful and Unsuccessful CAPTCHA Categories Responses	66
4.6	Average Response Time	70
4.7	Average Solving Time for Participants	74
4.8	Accuracy of Responses on E-NaijaCAPTCHA System	78
4.9a	CAPTCHA Solver Interface	84
4.9b	CAPTCHA Solver with Character Squash	85

## LIST OF TABLES

2.1	Text-Based CAPTCHA	14
2.2	Advantages and Disadvantages of CAPTCHA Types	25
4.1	Components of the Generated CAPTCHA Categories	46
4.2	List of the Possible CAPTCHA Categories	47
4.3	Feature Comparison of Variable-Length Accented CAPTCHA Categories	52
4.4	Number of Occurrences of Accented CAPTCHA Categories during Usability Test	60
4.5	Success Rate of Variable-Length Accented Character CAPTCHA Categories	64
4.6	Average Response Time (Milliseconds)	68
4.7	Average Solving Time for Participants	72
4.8	Accuracy of Responses on E-NaijaCAPTCHA System	76
4.9	Security Features of Variable-Length Accented CAPTCHA Categories	81
4.10	Solver's Output on the Variable-Length Accented CAPTCHA Categories	87
4.11	CAPTCHA Solver's Accuracy Report	90
4.12	Comparison of Features of E-NaijaCAPTCHA and NaijaCAPTCHA System	95

## **LIST OF ALGORITHMS**

3.1	Variable-Length Accented Character-Based CAPTCHA System	36
3.2	Obfuscation Algorithm for Variable-Length Accented Character-Based CAPTCHA System	38
3.3	Accented Character-Based CAPTCHA Response and Matching	40
3.4	CAPTCHA Solver Algorithm	44

## ABSTRACT

CAPTCHA, aimed at distinguishing humans from computers, typically utilizes text, images, audio, or video. Text-based CAPTCHAs, the most common type, face vulnerabilities due to their limited use of Latin characters. Little is known about the usability and security of text-based CAPTCHAs, particularly those incorporating accented characters. This study aimed to address these concerns by developing the E-NaijaCAPTCHA, a Variable-Length Accented Character-Based CAPTCHA system, utilizing Latin and accented characters from two Nigerian languages to bolster online transaction security.

Implemented using Javascript, PHP, HTML, and CSS, the E-NaijaCAPTCHA comprises four modules: CAPTCHA generator, obfuscator, display unit, and database. The generator employs the Gimpy algorithm to create codes of varying lengths (4 to 7 characters) with at least two accented characters. The obfuscator manipulates the code's appearance through color, text distortion, background noise, and skewing. User presentation and authentication occur in the display unit, with code correctness verified against the obfuscated value. The usability test conducted was measured using response time, solving time, success rate, and accuracy as the metrics while the OCR-based bots were used to determine its security.

Thirty CAPTCHA categories were generated, with participants producing 510 varied-length codes. Notably, 33.00% of codes included random lines, while 67.00% utilized skewing, squash, distortion, or collapse effects. All codes incorporated accented characters and colors. Among the generated categories, Character Squash with Gradient Background (CSGB) exhibited the shortest response time of  $1.02 \times 10^2$  ms, while Character Collapse with Colored Background (CCCB) had the longest  $4.62 \times 10^2$  ms. Text with Random Lines (TRL) demonstrated the fastest solving time of  $12.44 \times 10^2$  ms. The TDRL category achieved the highest accuracy (0.77) and success rate of 76.92%. Despite these variations, all codes underwent successful usability testing.

However, the security test using the OCR-based Solver yielded concerning results. None of the 510 generated codes achieved accurate authentication, indicating robust defense against automated attacks. The developed E-NaijaCAPTCHA system presents a formidable challenge to bots, suggesting its potential utility for government websites and transactional purposes.

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

**Word Counts:** 323

## CHAPTER ONE

### INTRODUCTION

#### 1.1 BACKGROUND OF THE STUDY

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 (Olanrewaju *et al.*, 2023). 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 unauthorized access to web services. The mechanism distinguishes between legitimate human users and automated, robotic entities. An individualized 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 utilize 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 in Nigeria, have become essential in the contemporary digital age, where protecting online security 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 possessing the property of a sweet spot is always a challenge. A CAPTCHA that is easily understood by humans (usability) but not by a machine (security) at the same time possesses a sweet spot property (Kumar *et al.*, 2022). 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 utilization of Latin and accented characters to construct CAPTCHA codes, a process underpinned by a refined and customized Gimpy algorithm. Nonetheless, the current configuration of the system, characterized by the generation of CAPTCHA challenges featuring fixed text lengths, introduces thought-provoking considerations that warrant exploration and refinement.

## **1.2 PROBLEM STATEMENT**

The existing NaijaCAPTCHA system is intricately designed to generate CAPTCHA challenges with a steadfast length of precisely five characters, thoughtfully 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 work seeks to develop a variable-length accented CAPTCHA code generation system that improves the level of security of the current NaijaCAPTCHA system that has been provided against automated programs.

### **1.3 AIM AND OBJECTIVES**

This research work aims to enhance the CAPTCHA code length generated by the NaijaCAPTCHA system through the development of a dynamic accented CAPTCHA system with variable code lengths.

The objectives of this study are:

1. To Design and enhance the NaijaCAPTCHA system.
2. To implement multiple code lengths ranging from 4 to 7 characters using a modified random character generator in the developed enhanced NaijaCAPTCHA system.
3. To test the usability of the Enhanced NaijaCAPTCHA, and its security using CAPTCHA solver.
4. To evaluate the performance of the existing and the Enhanced NaijaCAPTCHA systems

### **1.4 SCOPE OF THE STUDY**

This research work focused on the enhancement of the current NaijaCAPTCHA system to produce a variable length of CAPTCHA codes including accented characters within the context of online platforms and web security, with a specific focus on the Nigerian online environments. The research also intends to evaluate the functionality and security impact of the enhanced and existing system using CAPTCHA solver. This research work did not cover user-experience issues.

### **1.5 RESEARCH JUSTIFICATION**

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 (Alsuhbany *et al.*, 2017).

The deployment of a dynamic and ever-evolving CAPTCHA system strengthens the security protocols of online platforms by reducing vulnerabilities and thwarting unwanted access, data breaches, and other malicious activities.

The legacy of the current NaijaCAPTCHA system designed to generate a fixed length of five character codes including two accented characters, which has already been shown to be successful



in improving web security, is being greatly carried further by this study. The work of Abbas *et al.*, 2020, showed how the Urdu language-based CAPTCHA generated variable lengths of four to eight characters of codes which were used for the creation of an Urdu language-based CAPTCHA for regional URDU websites. The performance evaluation of this robust Urdu CAPTCHA framework for regional websites showed significant improvement over the limitations of the existing captcha for regional Websites. This research work therefore set out to enhance the already-effective NaijaCAPTCHA system with fixed and variable code lengths to better handle the security difficulties presented by the changing digital world.

## **1.6 GLOSSARY OF TERMS**

This section describes the key terms used in this work:

1. CAPTCHA (Completely Automated Public Turing Test to Tell Humans and Computers Apart): Refers to various authentication methods that validate users as humans, and not bots, by testing users with a challenge that is simple for humans but difficult for machines.
2. Variable-Length: This means any generated words whose length can vary between 4 to 7 characters including two accented characters.
3. Accented Characters: These are letters that have variations from the standard letter. These are somewhat rare in English but are very common in Yoruba and Igbo languages.
4. Fixed-text Length: This means a set of text lengths that never varies.
5. CAPTCHA Code: This is a protection and consciousness device developed to be in a role to validate the identification of customers on the internet.
6. Language-Based CAPTCHA: This type of CAPTCHA uses characters from a specific language to generate CAPTCHA code
7. CAPTCHA Solver: This is a computer program designed to recognize and automatically solve CAPTCHAS for users.
8. Bots: This is an automated software application that performs repetitive tasks over a network. It follows specific instructions to imitate human behavior but is faster and more accurate.

## **1.7 LAYOUT OF THE PROJECT**

This chapter gives a general overview of CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart) and the NaijaCAPTCHA system. We will also discuss the Problem Statement, the Aim and Objectives of the study, the Scope of the Study, and the Research Justification.

## **CHAPTER TWO**

### **LITERATURE REVIEW**

#### **2.1 SECURITY**

Security encompasses protective measures against threats in different areas like cybersecurity and personal safety. It entails protecting assets and information from unauthorized access and ensuring confidentiality, integrity, and availability through diverse practices and technologies.

##### **2.1.1 Triad of Security**

The CIA Triad, comprising Confidentiality, Integrity, and Availability, is a fundamental principle in information security. This framework establishes three crucial goals for safeguarding sensitive information and systems (Ravi, 2019). The CIA Triad provides a comprehensive approach for designing, implementing, and assessing security measures, serving as a vital framework for organizations to protect against various cyber threats and vulnerabilities.

**Confidentiality:** Confidentiality ensures that information is accessible solely to authorized users or systems, shielding data from unauthorized access, disclosure, or theft. This is achieved through various measures like encryption, access controls, user authentication, and data masking. As an example, sensitive data may undergo encryption before being transmitted, or access to classified information can be limited according to user roles and permissions.

**Integrity:** Integrity guarantees that information maintains accuracy, consistency, and reliability from creation to deletion. It prevents unauthorized alterations, tampering, or corruption of data (Mitchell *et al.*, 2023). Ensuring integrity requires implementing methods like data validation, checksums, digital signatures, and version control. For example, checksum algorithms can identify file alterations, while digital signatures authenticate documents.

**Availability:** Ensures that information and resources are readily accessible and usable for authorized users whenever necessary. This encompasses measures to prevent and address disruptions or downtime that might impact system or service availability. Strategies to uphold availability comprise redundancy, fault tolerance, disaster recovery planning, and defense against denial-of-service (DoS) attacks (Cabric, 2015). For instance, implementing backup systems for

seamless operation during hardware failures or employing network traffic filtering to counteract DoS attacks.

To enhance the outlined security principle, it's vital to effectively incorporate security mechanisms that optimize internal and external resources. These mechanisms serve to bolster information security, thereby enhancing the value of the CIA Triad. Security measures can be sorted based on their goals, encompassing prevention, education, and recovery. Each of these plays a crucial role in fortifying security measures.

### **2.1.2 Types of Security**

Security encompasses a range of categories, each tailored to address distinct facets of safeguarding and fortification. These categories comprise:

**Information Security:** This is the protection of digital resources from unauthorized activities such as alteration, disclosure, or destruction as it involves the deployment of firewalls, intrusion detection, encryption, and access control to bolster defenses against potential threats. Within organizations, it plays a pivotal role within an organization in operational resilience, upholding trust, and compliance by safeguarding the confidentiality, integrity, and availability of digital assets (Svanadze & Gnatyuk, 2024).

**Cyberwarfare:** It pertains to conflicts involving the use of computers, online control systems, and networks. It encompasses both offensive and defensive measures undertaken in reaction to cyber-attacks, espionage, and sabotage threats (Olanrewaju & Osunade, 2017).

**Network security:** Serves as a protective mechanism for computer networks and their infrastructure, aiming to thwart data interception, unauthorized access, and misuse of resources. It includes deploying encryption protocols, firewalls, access controls, and intrusion detection systems as measures for the creation of a secure environment for data communication and transmission.

**Internet security:** Focuses on maintaining the confidentiality, integrity, and availability of information and resources exchanged online, thus fostering secure online interactions and transactions. To protect against data breaches, malware, phishing, and other online threats, it comprises deploying a range of strategies and technologies (Agrawal & Baniya, 2024).

### 2.1.3 Threats to Security

The discussion below pertains to security threats that can lead to system malfunction due to unauthorized access.

**Social Engineering:** It's a manipulative strategy used by attackers to exploit human psychology and trust, aiming to gain unauthorized access to sensitive data or systems. Unlike a technical vulnerability, social engineering is based on the manipulation of emotions such as curiosity, fear, or willingness to help (S. A. Khan, 2023). Offering security awareness training is crucial to mitigate the risk of falling prey to a social engineering attack. This training covers common tactics that attackers use and how to effectively identify and respond to suspicious requests or communications.

**Phishing:** is a cyber-attack tactic aimed at acquiring personal information from victims through various forms or methods. The targets of such attacks can range from individual users to multiple organizations or institutions (Madleňák & Kampová, 2022).

**Worms:** Worms are unleashed by infiltrating a network and autonomously replicating from one system to another. Unlike viruses, they propagate without needing user interaction.

**Denial-of-Service (DoS) Attacks:** Denial of service represents a significant challenge in information security as it obstructs the timely availability of information. Security protocols might be susceptible to denial of service vulnerabilities due to resource-intensive verification processes, potentially allowing attackers to exhaust legitimate user resources (Toluwalope *et al.*, 2021).

**Virus:** A form of harmful software crafted to infect and propagate throughout computer networks by embedding itself into other software or files. It possesses the ability to reproduce and activate independently, frequently resulting in damage such as data corruption, system disruption, or security breaches.

## **2.2 AUTHENTICATION**

Authentication serves as the virtual gatekeeper, akin to a bouncer at a digital nightclub, meticulously checking IDs before granting entry. Its essence lies in verifying identities within the online domain. Similar to presenting a driver's license or passport for identity confirmation in the physical world, digital authentication entails providing credentials or proof to validate one's asserted identity. This stringent process ensures that only authorized individuals or entities gain entry to confidential data, systems, or services, thereby excluding unauthorized users (Yan & Ahmad, 2009). A wide variety of methods are included in the category of authentication techniques. These encompass advanced options such as cryptographic keys and biometric scans (like fingerprint or facial recognition), alongside traditional methods like passwords and PINs. Ultimately, authentication plays a pivotal role in upholding security and trust in digital interactions by authenticating the identities of involved parties (Althamary & El-Alfy, 2017).

### **2.2.1 Types of Authentication**

Three types of authentication are discussed below:

**Proof of identity:** involves accepting documentation or evidence provided by a reliable entity that directly verifies the authenticity of an individual's identity. This form of authentication relies on trustworthy sources or reputable individuals who can provide firsthand confirmation of the identity's legitimacy (Kim, 2017).

**Original Document:** In the second authentication technique, the attributes of the original object are compared to the known traits of items from that particular origin (Kim, 2017).

**Documentation:** Meeting the standard of proof in legal proceedings often requires judicial scrutiny of the presented evidence. This can be achieved through written documentation such as evidence logs or testimonies from law enforcement officers and forensic experts involved in the case. Additionally, certain antiques are accompanied by certificates verifying their authenticity. In the realm of computing and mobile technology, authentication methods vary widely. For instance, CAPTCHA stands out as a secure authentication method that integrates elements from different categories (Olanrewaju & Osunade, 2017).

## **2.2.2 Current Authentication Techniques**

Authentication is the process or action of showing something to be true, genuine, or valid. In today's digital age, the predominant electronic authentication method typically relies on using both a username and password. Two distinct types of authentication mechanisms are accessible: static and dynamic.

### **2.2.2.1 Static Authentication**

Static authentication employs a predetermined authenticator, like a password or PIN. It is called static because the authenticator is reused multiple times and stays the same until you change it.

### **2.2.2.2 Dynamic Authentication**

In contrast, in dynamic authentication, a separate authenticator is generated for every session and nothing is ever reused. Below are the methods of dynamic authentication:

#### **2.2.2.2.1 One-Time-Password (OTP)**

A one-time password (OTP) acts as a digital key that's usable only once, serving as an additional security measure during authentication procedures. When you're logging into an online account or making a sensitive transaction, the system sends you a unique OTP, typically via SMS, email, or a specialized authentication app. You enter this OTP along with your regular login credentials to prove it's you trying to access the account. Since the OTP is single-use and has a brief validity duration, it substantially lowers the chance of unauthorized entry, enhancing the safety and security of online transactions.

#### **2.2.2.2.2 Challenge-Handshake Authentication Protocol (CHAP)**

Introduces cryptographic mechanisms to verify user identities and prevent unauthorized access (Jasper *et al.*, 2023). Is like the bouncer at a digital club, ensuring only the right guests get in. It works by verifying a user or network entity's identity through a three-step handshake process. This protocol acts as a trusted guardian within computer networks, ensuring only authorized users or devices gain access.

#### **2.2.2.2.3 Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA)**

This is like a gatekeeper for websites, distinguishing between human users and automated bots. It presents challenges, like distorted text or image recognition tasks, that humans can typically solve but are difficult for machines. This helps prevent automated bots from accessing websites or services, ensuring a more secure and reliable online experience for users. The rise of automated programs, commonly referred to as bots, capable of accessing and engaging with online services just like humans, prompted the adoption of CAPTCHA as an authentication method. Many business websites implement CAPTCHA to deter automated bot attacks and ensure secure access to their services (Olanrewaju & Osunade, 2017).

### **2.3 COMPLETELY AUTOMATED PUBLIC TURING TEST TO TELL COMPUTERS AND HUMAN APART (CAPTCHA)**

CAPTCHAs serve as challenge puzzles designed to discern human users from automated ones. Essentially, they are programs capable of creating and evaluating tests that are typically solvable by humans but are challenging for current computer programs (Yang *et al.*, 2009). The adoption of CAPTCHA for authentication stemmed from the rise of automated programs, or bots, capable of mimicking human interactions with online services. To deter automated bot attacks, many commercial websites implement some form of CAPTCHA for authentication when gathering user data (Olanrewaju & Osunade, 2017).

Various elements like text, audio, numbers, images, and videos, along with their combinations, have been employed in CAPTCHA methods to pose challenges for bots attempting to solve them. Text-based CAPTCHA, particularly using the Latin character set, has been widely adopted. These text-based CAPTCHA schemes generate characters from the English alphabet, comprising 26 letters. Despite the development of numerous text-based CAPTCHA schemes with anti-segmentation mechanisms over the past two decades, many of them have been successfully compromised (Algwil, 2023).



### 2.3.1 History of CAPTCHA

Web services are important functions made possible by the internet. The widespread automated access to web resources via robots has made it crucial for web service providers to distinguish between human users and robots. Human Interaction Proofs (HIP), such as Completely Automated Public Turing tests to tell Computers and Humans Apart (CAPTCHA), provide a means to make this distinction (Banday & Shah, 2011). The deployment of CAPTCHA systems is an essential preventive measure against possible attacks by automated bots directed at websites and the network resources that support them. By efficiently differentiating between human users and automated scripts, these CAPTCHA systems play a crucial role in protecting online services and reducing the likelihood of unauthorized access or malevolent entity exploitation (Kaur & Behal, 2014).

Research indicates that CAPTCHA was initially developed by a research team at Carnegie Mellon University and has subsequently garnered global scholarly attention. Previously, the prevailing system relied solely on text passwords. Yet, the advent of CAPTCHA resulted in the fusion of text passwords with diverse CAPTCHA formats. Additionally, an alphanumeric password was devised, comprising a mix of letters and numbers (Yang & Hung, 2013). The use of CAPTCHA for authentication became more widespread as a result of the emergence of automated programs, or "bots," which can mimic human behavior while interacting with online services. The security and operation of online platforms were significantly in danger due to this technological advancement. As such, most for-profit websites engaged in data gathering realized how critical it was to properly combat automated bot attacks. To lessen the threat posed by these automated bot attacks, CAPTCHA was implemented as a preventative defensive measure. Completely Automated Public Turing Test to Tell Computers and Humans Apart, or CAPTCHA offered obstacles made especially to discriminate between automated scripts and human users. These problems usually entail tasks that are much harder for automated bots to do than for humans. Commercial websites strengthened their security measures by adding CAPTCHA to their authentication processes (B. Khan *et al.*, 2013). This prevented unauthorized access and potential exploitation of their systems by bad actors.

This deliberate application of CAPTCHA strengthens consumer trust in the security of their personal information while also improving the security posture of websites that gather data. Adopting CAPTCHA continues to be an essential part of a complete defensive plan as online threats change, protecting user data's integrity and confidentiality in an increasingly linked digital world.



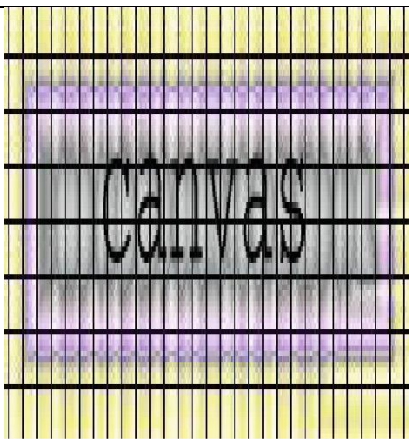
























## **2.3.2 Types of CAPTCHA**



### **2.3.2.1 Text-Based CAPTCHA**

Text-based CAPTCHA, the most prevalent form, typically presents distorted text comprising letters and numbers, often without case sensitivity. This type of CAPTCHA is accessible even to individuals with color blindness. The characters are intentionally distorted to prevent recognition by computer programs, ensuring that only humans can decipher and input the correct sequence (Alammar *et al.*, 2022).

Text-based CAPTCHA stands as the most frequently employed form owing to its widespread use across the internet. However, if not implemented with caution, it can pose significant security risks, susceptible to various attacks (Ling-Zi & Yi-Chun, 2012). To counter this vulnerability, meticulous design considerations and security measures are incorporated into text-based CAPTCHA systems to thwart potential assaults. There are other examples of Text-Based CAPTCHA which are briefly explained in Table 2.1 page 14

Table 2.1: Text-Based CAPTCHA (Pate & Ramteke, 2023)

ReCAPTCHA: The reCAPTCHA service asks users to complete on-screen words shown in distorted text pictures and click on "I'm Not a Robot" using the CAPTCHA interface.									
Gimpy: Instead of using automated technologies, Gimpy is founded on the idea that individuals read damaged and corrupted words. It functions by choosing words from a dictionary and showing them to the user in a distorted and corrupted visual form. The user is then prompted to type the words they saw on the screen. works in conjunction with Yahoo.									
EZ-Gimpy: Instead of automated programs that operate by selecting a single word from a dictionary and then making it appear in a corrupted and distorted image format before asking the user to type the term presented in the distorted image format, EZGimpy CAPTCHA bases its operation on the idea that humans can read distorted, textured backgrounds and overwhelmed text.									
Bongo: Bongo is a CAPTCHA designed to address the problem of visual pattern recognition in humans. It displays two separate block series (left and right). The user is tasked with identifying the feature that indicates the difference between two blocks.	<p>Figure 2. Sample Bongo Captcha.</p> <table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>								
									
									

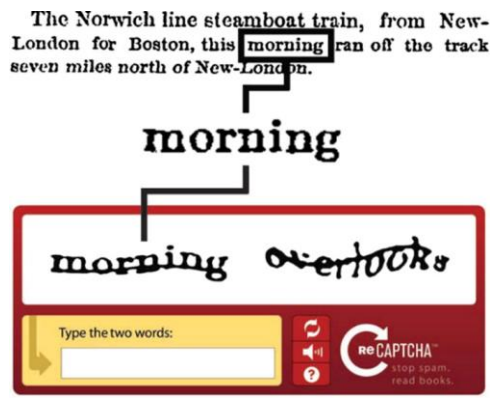
<p>MSN-CAPTCHA: Eight (upper case) characters and digits are used. Dark blue makes up the foreground, while grey makes up the background. It is employed to produce the ripple effect and bend the characters.</p>	
<p>Baffle -Text: At California University in Berkeley, Henry Baird creates the design. It is an altered form of Gimpy. In the case of Baffle text, alphabets or characters are chosen at random and combined to create a pronounceable text. After that, the user is prompted to type the right term</p>	

### **2.3.2.2 Image-Based CAPTCHA**

This type primarily capitalizes on the disparity in semantic interpretation capabilities between humans and computers in image recognition. Users are presented with an image-based CAPTCHA test, requiring them to differentiate or categorize certain images or objects based on their attributes. In recent times, diverse image-based CAPTCHA schemes have emerged, aiming to enhance security features and usability across various devices (Algwil, 2023). This includes Style Matching CAPTCHA, Google's reCAPTCHA, and PiSHi CAPTCHA as illustrated in Figure 2.1 page 17



(a) PiSHi CAPTCHA



(b) Google CAPTCHA



(c) Style MatchingCAPTCHA

Figure 2.1: Examples of Image-Based CAPTCHA (Mehrnezhad *et al.*, 2017), (Abubaker *et al.*, 2017), (Algwil, 2023)

### 2.3.2.3 Sound-Based CAPTCHA

The Audio-Based CAPTCHA offers an alternative to traditional visual CAPTCHAs, catering especially to individuals with visual impairments or accessibility concerns. This type of CAPTCHA in Figure 2.2 page 19, presents users with audio challenges instead of visual ones. Users listen to a series of spoken letters, numbers, or words and accurately respond based on the auditory cues alone (Alnefaie, 2020). In these systems, audio hardware is used to produce clear sound, but sometimes, localization issues can make it challenging for users to perceive the audio accurately. However, tackling an audio-based CAPTCHA employs a comparable strategy to text-based CAPTCHAs, entailing the extraction of features and the identification of letters. Consequently, the efficacy of audio-based CAPTCHA in terms of user-friendliness or resistance against automated bots can differ based on the unique implementation and user interaction.

A prevalent illustration of a sound-based CAPTCHA is reCAPTCHA, originally crafted by Carnegie Mellon University and subsequently acquired by Google. Alongside its text counterpart, this system offers users an audio clip featuring spoken numbers by various individuals. These sound-based CAPTCHAs have been implemented on platforms like google.com and dig.com (Gao *et al.*, 2010). Also, other developers introduced a CAPTCHA called Match-the-Sound CAPTCHA (MS-CAPTCHA), harnessing the cognitive capabilities of human users. In this setup, individuals listen to a sound and subsequently choose the image that matches the sound they heard to verify their human status. Studies reveal that MS-CAPTCHA enhances user experience and effectiveness, demonstrating higher success rates and quicker completion times compared to reCAPTCHA (Tariq & Khan, 2018).

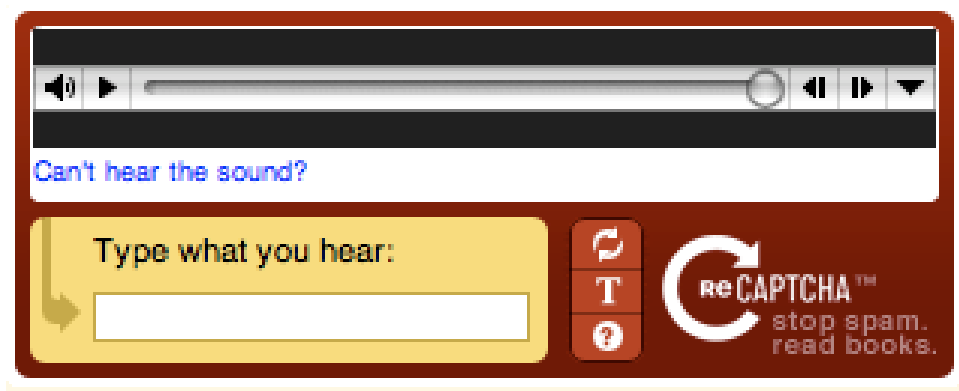


Figure 2.2: Sound-Based CAPTCHA (Shivani & Challa, 2020)



#### **2.3.2.4 Video-Based CAPTCHA**

This CAPTCHA variant employs videos as the challenge medium instead of traditional images or text. In this approach, users are presented with videos and tasked with labeling descriptive keywords. The user is tasked to watch a designated challenge video and then accurately annotate it with appropriate tags or keywords. The system evaluates users' responses by comparing them to a database containing predefined tags associated with the video, grading based on the accuracy of the annotations. This method leverages users' ability to comprehend and categorize visual content, enhancing security by introducing a dynamic and engaging challenge (Rao *et al.*, 2016). They are shown in Figure 2.3 page 21.



Type 3 words that best describe this video:

Submit

Figure 2.3: Video-Based CAPTCHA Sample (Kluever & Zanibbi, 2009)

### **2.3.2.5 Puzzle-Based CAPTCHA Type**

A puzzle-based CAPTCHA engages users with visual or interactive challenges, prompting them to complete tasks like arranging objects, solving puzzles, or navigating mazes. These tasks are designed to be straightforward for humans but challenging for automated bots. By requiring cognitive engagement rather than simple recognition, puzzle-based CAPTCHAs offer heightened security measures to verify human users and thwart automated bot attacks. The authentication process using puzzle-based CAPTCHA begins with users interacting with the interface. Users are required to drag and drop images into designated positions, arranging them in the correct sequence as demonstrated in Figure 2.4 page 23. Successful arrangement of all images confirms the user as human (Ali & Karim, 2014). However, any error in arrangement results in the user being prompted with a new set of images for solving. The architecture of puzzle-based CAPTCHA comprises three primary components: image collection, image processing engine, and user interface.



Figure 2.4: Puzzle-Based CAPTCHA (Gao *et al.*, 2010)

### **2.3.3 Advantages and Disadvantages of CAPTCHA Types**

While CAPTCHA aims primarily to block bot access to web services, a commendable goal, it does possess several limitations that could potentially dissuade users from its use. The choice of CAPTCHA type depends on factors such as security needs, user accessibility, and implementation feasibility. Each CAPTCHA variation carries its own set of pros and cons, emphasizing the need for a thorough assessment to select the most suitable option. A few of these are shown in Table 2.2 page 25.

Table 2.2: Advantages and Disadvantages of CAPTCHA Types (Pate & Ramteke, 2023)

S.No	Type of CAPTCHA	Advantages	Disadvantages
1	Text-Based CAPTCHA	<ul style="list-style-type: none"> <li>1. Implementation is easy.</li> <li>2. Baffle Text-based CAPTCHA is used to defeat dictionary attacks.</li> <li>3. Re-CAPTCHA uses new words in the dictionary that cannot be read by OCR.</li> </ul>	<ul style="list-style-type: none"> <li>1. The user has some problem identifying the correct text or characters i. Multiple fonts. ii. Font size. iii. Blurred Letters iv. Wave Motion</li> <li>2. OCR methods can easily identify it.</li> </ul>
2	Image-Based CAPTCHA	<ul style="list-style-type: none"> <li>1. Increases Security than text-based CAPTCHA.</li> <li>2. An easy system based on clicks, so no need to type.</li> <li>3. Image recognition pattern is a challenging AI program.</li> </ul>	The problem of image identification for those who have low vision or are due to the blurring of images
3	Audio-Based CAPTCHA	<ul style="list-style-type: none"> <li>1. It is used for people who have a visual impairment.</li> <li>2. Friendly to people.</li> </ul>	<ul style="list-style-type: none"> <li>1. Language support</li> <li>2. The character that has a similar sound.</li> </ul>
4	Video-Based CAPTCHA	<ul style="list-style-type: none"> <li>1. Using OCR (Optical Character Recognition) it cannot crack.</li> <li>2. Provides greater security.</li> </ul>	Because of the big size of the file, users have trouble downloading video and finding the right CAPTCHA
5	Puzzle-Based CAPTCHA	<ul style="list-style-type: none"> <li>1. It appears to be enjoyable.</li> <li>2. It helps monitor the brain of the user.</li> </ul>	The job is not simple for users because it takes longer to fix the puzzle-based CAPTCHA.

### **2.3.4 CAPTCHA Threats**

CAPTCHA systems, while effective in mitigating automated bot attacks, are not without their vulnerabilities (Korakakis *et al.*, 2014). One significant threat is CAPTCHA bypass techniques, where attackers find ways to circumvent the CAPTCHA mechanism, allowing automated bots to pass through undetected. Another issue is CAPTCHA fatigue, which occurs when genuine users become frustrated with the complexity of CAPTCHA challenges, resulting in a negative user experience or abandonment of the task altogether. Furthermore, CAPTCHA systems may encounter accessibility challenges, creating obstacles for users with disabilities who may find it challenging to fulfill the requirements (Kumar, 2017). Furthermore, as CAPTCHA methods advance to become more intricate, adversaries develop corresponding techniques to bypass them, leading to a continuous battle between CAPTCHA designers and attackers (Al-Fannah, 2017). These threats underscore the importance of continually evaluating and enhancing CAPTCHA systems to maintain their effectiveness in thwarting automated attacks while minimizing user friction.

### **2.3.5 Breaking CAPTCHA System**

Breaking CAPTCHA systems entails discovering techniques to evade or overcome the security measures implemented by these systems, enabling unauthorized access or automated operations. It's the process where clever tricks or technologies are used to outsmart those little puzzles designed to separate humans from bots. This could involve things like using special software to decipher distorted text, training computers to recognize patterns, or even hiring real people to solve Captchas on demand (Al-Taie, 2014). In essence, it involves discovering methods to automate activities originally intended for humans, such as creating accounts or inundating websites with spam. It's an ongoing challenge between defenders safeguarding online environments and adversaries seeking to exploit them.

Compared to other security solutions, a system's ability to achieve its intended security goals may be compromised by flaws in its implementation or design. Numerous CAPTCHA implementations, particularly those lacking expert design and review in the security domain, are susceptible to common attack vectors (Khawandi *et al.*, 2019).

## **2.4 CHARACTER SET**

A character set represents the comprehensive range of unique characters utilized and supported by computer software and hardware. It involves using codes, bit sequences, or numeric forms to represent different characters, as demonstrated by standards such as ASCII and EBCDIC. These standards define how characters are encoded in computer systems. The ASCII character set, known for its extensive usage, is especially notable for presenting text and numbers on computer displays. Ranging from 0 to 127, it encompasses uppercase and lowercase English letters, numbers, mathematical operators, and various symbols. This encompasses characters employed in representing text in computer systems, including letters, numbers, symbols, and other displayable or processable characters like 0 - 9, A - Z, a - z, as well as special symbols.

### **2.4.1 Accented Symbols**

Accented characters feature diacritical marks used in languages such as French, Spanish, German, and Portuguese to indicate subtle pronunciation distinctions or emphasize stress (for example: like é, è, â, ö, ñ, and ç). In computing, they're integral to character sets like Unicode and ASCII, ensuring accurate text representation across languages. While enhancing linguistic clarity, they can complicate data processing, particularly in systems lacking Unicode support. Proper handling is crucial to maintain seamless communication and user experience across diverse linguistic environments.

## **2.5 LANGUAGE**

Language is an organized method of communication comprising grammar and vocabulary. It serves as the fundamental tool through which humans express meaning, whether orally or in writing, and can also be communicated through sign languages. Human language displays cultural and historical diversity, showcasing notable differences among cultures and over different periods.



### 2.5.1 Yoruba Language

Yorùbá, also known as èdè Yorùbá, belongs to the Benue-Congo branch of the Niger-Congo language family, spoken by a vast community of about 40 million people primarily residing in Nigeria. The language is also spoken in areas of the United Kingdom, the United States of America, Sierra Leone, Benin, and Togo (Oyeniran *et al.*, 2021).

Even though English holds the status of the official language in Nigeria, Yorùbá, along with Igbo and Hausa, plays a quasi-official role, acting as a common language for communication among speakers of the numerous languages spoken across Nigeria. Particularly prevalent in southwest Nigeria where most Yorùbá speakers reside, the language finds utility in various domains, including government administration, media, education, literature, and entertainment.

Code-switching between Yorùbá and English is common among bilingual Yorùbá-English speakers, occurring interchangeably in different contexts. While Yorùbá is often used in familial and informal settings, standard English is preferred in formal or official situations. Informally, a hybrid form known as Yoruglish emerges, blending elements of both languages in grammar and vocabulary.

Yorùbá encompasses a continuum of dialects with distinct variations in pronunciation, grammar, and vocabulary, categorized into three main geographic areas: Central Yoruba, Northwest Yoruba, and Southeast. The translation of the Bible in 1884 by Bishop Samuel Ajayi Crowther, a Yorùbá speaker, facilitated the standardization of written Yorùbá, which has since been adopted widely across dialects. This standardized form, known as Standard Yorùbá, serves as the literary and educational norm, incorporating features from various dialects and adopting a simplified vowel harmony system. Yorùbá syllables typically consist of a vowel with or without a preceding consonant, devoid of consonant clusters. Derivation and reduplication are primary mechanisms for word formation in Yorùbá, supplemented by borrowings from neighboring languages like Hausa.

Yorùbá transitioned from an unwritten language to a written one in the early 19th century, catalyzed by Bishop Crowther's translation efforts. Since then, there has been a steady flow of Yorùbá literature, including books, newspapers, and magazines, enriching the language's written corpus. Yorùbá is typically written using the Latin alphabet, with adaptations to represent Yorùbá

sounds, including diacritics and digraphs. Tonal distinctions are indicated using acute and grave accent marks.

In essence, the Yorùbá language and its various dialects embody a rich cultural heritage, serving as a vehicle for communication, expression, and cultural preservation among its speakers across the globe.

### **2.5.1.2 Yoruba Character Set**

Yorùbá, a language spoken in Nigeria, employs a character set comprising twenty-five letters, some resembling Roman characters. Of these, seven represent vowels and eighteen represent consonants (Olanrewaju & Osunade, 2017). The characters can be written in upper and lower case. The Yoruba character set includes uppercase and lowercase ((A B D E Ě F G GB I H J K L M N O Ọ P R S Ș T U W Y), and (a b d e ẹ f g gb i h j k l m n o ọ p r s ș t u w y)). Notably, the combination 'GB' or 'gb' is a unique case where two consonants are allowed to follow each other in Yoruba orthography (Ojumah *et al.*, 2018).

### **2.5.2 Igbo Language**

The Igbo language is spoken by an estimated 31 million people and comprises a diverse range of dialects, potentially numbering around 35 distinct Igboid languages depending on classification criteria. While regarded as a single language, there is limited mutual intelligibility among the core Igbo cluster's various groupings in regions like the north, west, south, and east.

Igbo is tonal, with tone variations playing a significant role in distinguishing words across dialects. Typically, there are observed three register tones and three contour tones, illustrated by examples like "ákwa" (cry), "àkwà" (bed), "àkwá" (egg), and "ákwa" (cloth). Although tone representation in writing is inconsistent, it holds essential meaning-conveying functions.

The standardization of Igbo orthography has been the subject of historical discourse, leading to the formulation of a literary language known as "Igbo izugbe" or "general Igbo." This standardized variant, adopted around 1972, amalgamates elements from dialects such as Orlu (Isu), Anambra (Awka), and Umuahia (Ohuhu), with adjustments to nasalization and aspiration.

In 1961, the Government of Eastern Nigeria established a committee chaired by Mr. S.E. Onwu to tackle Igbo orthographic issues. The committee proposed a set of twenty-eight consonant and eight

vowel letters (a b c h d e f g gb gh gw h i j k kp kw l m n ñ nw ny o o p r s sh t u u v w y and z), laying the foundation for a standardized Igbo writing system (Ohiri-Aniche, 2007).

## **2.6 REVIEW OF NON-ENGLISH LANGUAGE CAPTCHA**

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 CAPTCHAs 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 recognize 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 utilizing non-native languages could provide improved usability for internet users fluent in those languages (M. Kumar *et al.*, 2022).

### **2.6.1 NaijaCAPTCHA System**

Olanrewaju & Osunade, Proposed the “Development of an Accented Character-Based CAPTCHA System called NaijaCAPTCHA”. To strengthen security and increase accessibility for non-native English users, accented characters were added to the NaijaCAPTCHA system. This addition aims

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 enhances the acceptability and diversity of CAPTCHA authentication.

### **2.6.2 Arabic CAPTCHA**

In their paper titled "Secure Arabic Handwritten CAPTCHA Generation Using OCR Operations," authored by S. A. Alsuhibany and M. T. Parvez, the authors present a method for creating Arabic handwritten CAPTCHA images. These CAPTCHAs are generated using a subset of 123,200 prewritten Arabic character images, extracted from the KHATT database. The process outlined in the paper involves generating Arabic handwritten CAPTCHA images using prewritten Arabic character images. The steps include 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. These steps aim 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 aims to enhance security by making CAPTCHA recognition challenging for automated systems while maintaining usability accuracy above 88%.

### **2.6.3 Sindhi Text-Based CAPTCHA**

Kehar *et al.*, (2021) 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 colored 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.

#### **2.6.4 Advanced Nastaliq CAPTCHA**

In 2008, Shirali-Shahreza & Shirali-Shahreza introduced the "Advanced Nastaliq CAPTCHA" method, which utilizes Arabian and Persian characters of varying lengths, typically between three to eight characters. This approach employs 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 utilizing the JAVA programming language.

## **CHAPTER THREE**

### **METHODOLOGY**

#### **3.1 RESEARCH DESIGN**

This study was conducted through an experimental framework, employing a quantitative methodology. The research unfolded in three distinct phases. The initial phase involved the development and deployment of the Variable-Length Accented Character-Based CAPTCHA system on a web server. Subsequently, the second phase centered on the usability assessment of the developed CAPTCHA system, focusing on user interactions and experiences. Finally, the third phase entailed evaluating the security aspects of the variable-length accented character-based CAPTCHA scheme by subjecting it to analysis using a CAPTCHA solver.

#### **3.2 VARIABLE-LENGTH ACCENTED CHARACTER-BASED CAPTCHA SYSTEM**

The conceptualization 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 3.1 page 34, offering a comprehensive overview of the system. 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.

These four modules' intricate relationships and coordinated operations form the backbone of the variable-length accented character-based CAPTCHA system. Each module's specific role, combined with the synergistic collaboration between them, contributes to the effectiveness of this CAPTCHA design. Further elucidation on these relationships and operations is provided to offer a comprehensive understanding of the system's inner workings and its capacity to provide a secure and user-friendly experience.

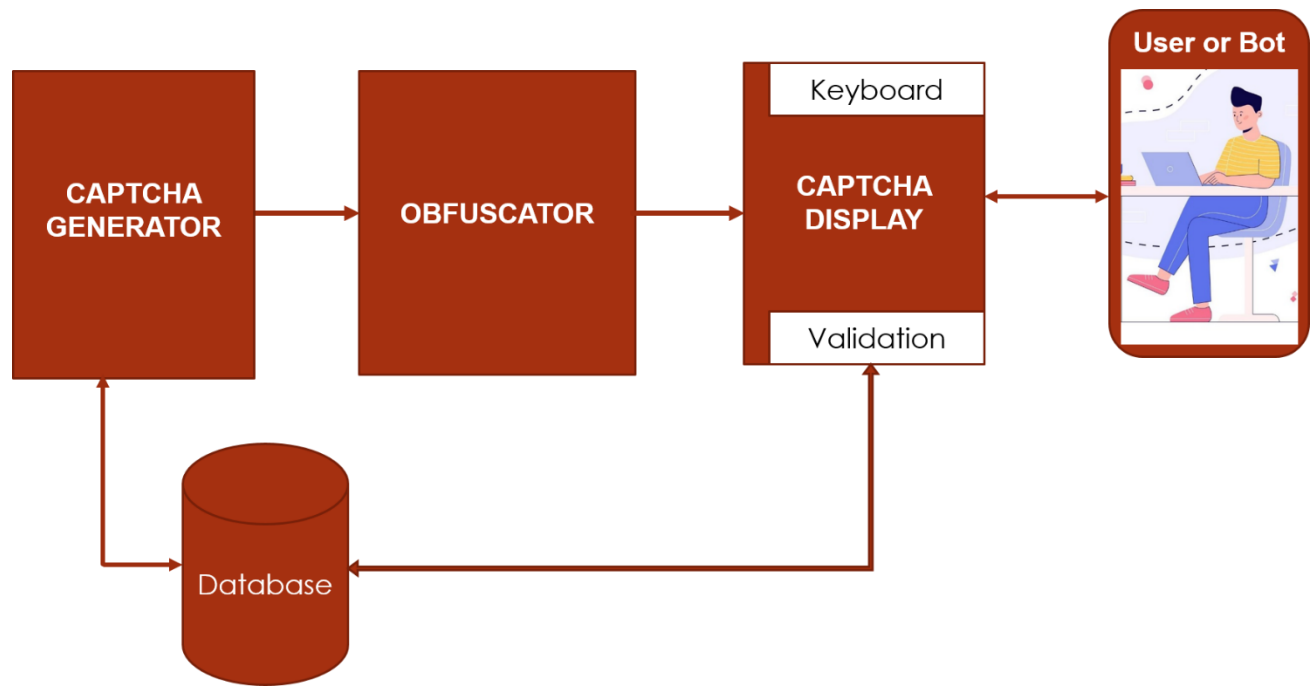


Figure 3.1: Variable-Length Accented Character-Based CAPTCHA System

### 3.2.1 CAPTCHA GENERATOR

This module 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 3.1, page. 36. 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, Ã, ã, Ä, ä, Å, å, Æ, æ, Ç, ç, É, é, Ê, è, Ë, ë, Ò, ò, Ó, ó, Ô, ô, Õ, õ, Ö, ö, Ø, ø, Ù, ú, Ú, ù, Û, û, Ü, ü, 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, Ã, ã, Ä, ä, Å, å, Æ, æ, Ç, ç, É, é, Ê, è, Ë, ë, Ò, ò, Ó, ó, Ô, ô, Õ, õ, Ö, ö, Ù, ú, Ú, ù, Û, û, Ü, ü, Ý, ý, 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)

Step 11: STOP

Algorithm 3.1: Variable-Length Accented Character-Based CAPTCHA System

### **3.2.2 SYSTEM OBFUSCATOR**

This part of the designed system is a sophisticated security mechanism that generates complex Variable-Length Accented CAPTCHA codes to make it difficult for automated programs or bots to decode. It uses a dynamic Varied Length Accented CAPTCHA code with various deformities which include but are not limited to 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. The intellectual cornerstone of this system is Algorithm 3.2 page 38, which orchestrates these deformities and ensures the desired performance.

**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 3.2: Obfuscation Algorithm for Variable-Length Accented Character-Based CAPTCHA System

### 3.2.3 CAPTCHA DISPLAY

This module presents to the user the varied length accented CAPTCHA code generated on screen and accepts their input. Within this module, three processes are implemented which are to be carried out such as:

**The Keyboard:** This unit within the system is equipped with a purposefully designed Virtual Keyboard, providing users with convenient access to accented characters for simplified input. 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:** This unit is implemented to carry out two major operations that include response checking and response matching:

**Response Checking Unit:** This specialized 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. This proactive approach aligns with the system's commitment to maintaining security, accuracy, and user-friendly interaction throughout the variable-length accented CAPTCHA validation process.

**Response Matching Unit:** This module occurs upon receiving user input. The user's input undergoes scrutiny by comparing it against the record in the database to ascertain equality. There's a possibility that the user's input might resemble the pre-encoded string generated by the variable length accented CAPTCHA code generator. The execution of this procedure is facilitated by the implementation of Algorithm 3.3, as documented on page 40.

```
Step 1a: Display the generated variable-length CAPTCHA code
Step 2a: Determine if the user has started to input
    Step 2b: If yes, initiate the Start Time
    Step 2c: If no, wait for user input
    Step 2d: If the user presses Enter/Submit key, take as the End Time of input
Step 3: Check if the user input is equal to the generated CAPTCHA code
Step 4: If yes,
    Step 4a: Compute the Response and Solving Time
    Step 4b: Save time in the database
    Step 4c: Display the “Your Input Match Correctly” message

//If No//

Step 5: Else:
    Step 5a: Calculate the Response and Solving Time
    Step 5b: Display the “Your Input Doesn’t Match” message to the user
    Step 5c: End loop
Step 6: Go to Step 1
```

Algorithm 3.3: Accented Character-Based CAPCTHA Response and Matching Algorithm

### **3.2.4 THE DATABASE**

This module 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 responses provided by users or solvers.

## **3.3 PERFORMANCE EVALUATION**

The assessment of the newly created Variable-Length CAPTCHA system, which utilizes accented characters, comprised two distinct evaluations. Firstly, a usability test was conducted to gauge the acceptability of the generated CAPTCHA codes by human users. This test aimed to assess the system's user-friendliness and ease of interaction. Secondly, a security evaluation was carried out using the Tesseract OCR (Optical Character Recognition) Engine. This assessment sought to determine the system's susceptibility to attacks from automated programs operated by computers. Through these evaluations, the effectiveness of the CAPTCHA system in terms of both user acceptance and resilience against automated attacks was thoroughly examined and scrutinized.

### **3.3.1 USABILITY TESTS**

The usability tests were meticulously conducted within an uncontrolled environment, involving fifty-one (51) participants of four (4) different ethnicities including Yoruba, Igbo, Hausa, and Fulani with the ages ranging from 12 to 45 years and above of both Males and Females to gauge the acceptability of the generated CAPTCHA codes by human users. The participants were provided with step-by-step instructions on how the developed accented character-based captcha system works. Each participant was tasked to attempt Ten (10) CAPTCHA challenges that would be generated randomly by the system via a user-friendly web interface. After each attempt, the user was asked to respond to the CAPTCHA challenge posed on screen and hit “Verify” button to submit their response at each attempt.

Following each submission, both the solving time and response time for the solved and unsolved CAPTCHA codes were meticulously recorded with their corresponding categories of CAPTCHA code generated. The webpage seamlessly updated after each submission, presenting participants with the subsequent CAPTCHA code to be solved. This methodical approach allowed for a

comprehensive evaluation of the system's usability, providing valuable insights into participants' interactions and responses to the CAPTCHA challenges presented.

### **3.3.2 SECURITY EVALUATION**

The automated application based on the open-source OCR Engine Tesseract was extensively tested to determine whether the variable-length accented character-based CAPTCHA system could be solved. With the aid of Unicode Transformation Format (UTF-8) support, Tesseract, an OCR text recognition engine, can recognize more than 100 languages. When presented in Latin format, it has been trained to identify characters from a variety of languages, including Japanese, German, and Korean. To conduct the test, we created a graphical user interface that would make it simple for us to combine the Tesseract library with the accented CAPTCHA codes that our system generated. To determine how reliable and strong the language-based CAPTCHA system was, an automated program was specifically created to decode the CAPTCHA codes that the system produced. Every attempt made by the automated program was meticulously documented to determine its success or failure and the accuracy of every CAPTCHA code attempted.

The procedure that occurs during the evaluation phase is outlined in Algorithm 3.4 page 43. The generator, or portion of the system that generates the CAPTCHA code, completes certain tasks to produce the code; the solver, or portion of the system that solves the CAPTCHA, completes the remaining steps to decrypt and break the generated codes. We were able to assess the system's resistance to automated attacks and obtain important knowledge about its functionality and security protocols thanks to this methodical approach.

```
Step 1: Start
Step 2: Send the Generated CAPTCHA code to the solver
Step 3: Display Generated CAPTCHA code
Step 4: Solve the Displayed Generated CAPTCHA code
Step 5: Return result (on-screen display)
Step 6: If result is equal to the generated Variable-length accented
        characters:
        Step 6a: Save the result to database

//If No//

        Step 6b: Else:
                Step 6bi: Return result (Display on screen)
                Step 6bii: Save result into the database
Step 7: Stop
```

Algorithm 3.4: CAPTCHA Solver Algorithm



### **3.3.3 EVALUATION METRICS**

The effectiveness of the variable-length accented character-based CAPTCHA system that was developed was assessed through a comprehensive examination of various performance metrics listed below. This evaluation aimed to gauge the system's proficiency and reliability in effectively distinguishing between human users and automated bots. Through meticulous scrutiny of these performance metrics, critical insights were garnered regarding the system's overall efficacy and its capacity to fulfill its intended purpose in securing online platforms and preventing unauthorized access.

#### **Solving Time**

This is the amount of time the user takes to answer the CAPTCHA system's challenge. The time it takes to solve the CAPTCHA is measured from the moment the code appears on the screen until the user begins to type a response.

#### **Response Time**

The time it takes a user to respond to the CAPTCHA challenge is this. It begins when the user hits the Enter or Return key to indicate that they are finished entering a CAPTCHA code and ends when the user sees a response, either successful or unsuccessful.

#### **Accuracy**

This indicates the proportion of CAPTCHA codes that have been accurately solved by either humans or bots, in comparison to the total number of CAPTCHA codes generated.

#### **Success Rate**

This represents the ratio of correctly entered CAPTCHA codes to the total number of CAPTCHA codes generated for a specific CAPTCHA type.

## **CHAPTER FOUR**

### **RESULTS AND DISCUSSION**

#### **4.1 SYSTEM IMPLEMENTATION**

The findings of the study are systematically presented, aligning them with the predefined research objectives and delineating the three distinct phases of the employed methodology. Different visual elements such as tables, charts, images, and screenshots were utilized to clarify and enhance understanding by illustrating the results.

The research inquiry was conducted 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 utilized a variety of software components such as Windows 10, JavaScript, PHP, HTML, CSS, MYSQL, and XAMPP Server. The assessment of the created CAPTCHA system, named E-NaijaCAPTCHA, centered on variable-length accented characters-based, 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.

#### **4.2 THE DEVELOPED SYSTEM CODE GENERATION**

The E-NaijaCAPTCHA system yields 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 4.1 on page 46. The combination of different backgrounds and character formats results in thirty (30) unique CAPTCHA categories, as illustrated in Table 4.2 on page 47. The program code for generating the variable-length accented character-based CAPTCHA system codes is provided in Appendix 3.1 on page 104.

Table 4.1: Components of the Generated CAPTCHA Categories

<b>Random Background</b>	<b>Randomly Generated Characters</b>
Gradient	Colored
Random Line	Squashed
Background with Noise	Fragmented
Colored Background	Distorted
Plain Background	Skewed
	Collapsed

Table 4.2: List of the Possible CAPTCHA Categories

S/N	CAPTCHA Category	Acronym
1	Character Fragmentation with Random Lines	CFRL
2	Text No Background	TNB
3	Character Squash with Colored Background	CSCB
4	Character Squash with Background Noise	CSBN
5	Character Collapse Background Noise	CCNB
6	Text with Gradient Background	TGB
7	Text Distortion Background Noise	TDBN
8	Character Fragmentation with Background Noise	CFBN
9	Text with Colored Background	TCB
10	Text Distortion with Colored Background	TDCB
11	Character Squash with Gradient Background	CSGB
12	Character Squash No Background	CSNB
13	Character Collapse with Random Lines	CCRL
14	Text Distortion with No Background	TDNB
15	Text Distortion with Gradient Background	TDGB
16	Character Fragmentation No Background	CFNB
17	Colored Text with Random Lines	CTRL
18	Character Collapse with Colored Background	CCCB
19	Character Fragmentation Colored Background	CFCB
20	Text with Background Noise	TBN
21	Colored Text No Background	CTNB
22	Character Fragmentation with Gradient Background	CFGB
23	Colored Text Colored Background	CTCB
24	Character Collapse Background Noise	CCBN
25	Text with Random Lines	TRL
26	Character Squash with Random Lines	CSRL
27	Text Distortion with Random Lines	TDRL
28	Character Collapse Gradient Background	CCGB

29	Colored Text Background Noise	CTBN
30	Colored Text with Gradient Background	CTGB

## **Description of each CAPTCHA Categories**

1. Character Fragmentation with Random Lines (CFRL): The CAPTCHA features colored characters that are fragmented into units, with random lines incorporated.
2. Text No Background (TNB): This CAPTCHA Category has colored text with no background.
3. Character Squash with Colored Background (CSCB): This CAPTCHA category has its characters colored and the squash feature and colored background are applied
4. Character Squash with Background Noise (CSBN): This Category of CAPTCHA has color applied to its characters and a squash feature with background noise is applied to them.
5. Character Collapsed Background Noise (CCBN): This CAPTCHA category has the characters colored and collapsed with background noise applied.
6. Text with Gradient Background (TGB): The CAPTCHA category comprises colored characters with gradient backgrounds included.
7. Text Distortion with Background Noise (TDBN): This CAPTCHA category has its characters distorted, and colored with background noise added.
8. Character Fragmentation with Background Noise (CFBN): This category of CAPTCHA has the characters colored, broken into units with colored a backgrounds added.
9. Text with Colored Background (TCB): This CAPTCHA has color applied to the characters with colored background added at random.
10. Text Distortion with Colored Background (TDCB): This category of CAPTCHA has its characters colored, distorted, and with colored backgrounds applied randomly.
11. Character Squash with Gradient Background (CSGB): This Category of CAPTCHA has color applied to its characters and a squash feature with gradient background applied to them.
12. Character Squash with No Background (CSNB): This Category of CAPTCHA has its characters colored with a squash feature applied with no background.
13. Character Collapse with Random Lines (CCRL): This CAPTCHA has its characters colored with random lines in the background.
14. Text Distortion with No Background (TDNB): This category of CAPTCHA has its characters colored, distorted, with no background applied.

15. Text Distortion with Gradient Background (TDGB): This category of CAPTCHA has its characters colored, distorted, and with gradient backgrounds.
16. Character Fragmentation with No Background (CFNB): This CAPTCHA category has the characters colored, broken into units with no background added.
17. Colored Text with Random Line (CTRL): This category has its characters colored with random lines applied as the background.
18. Character Collapse with Colored Background (CCCB): The CAPTCHA category has its characters colored but distorted with colored background added.
19. Character Fragmentation Colored Background (CFCB): The CAPTCHA category has the characters colored, but broken into units, and with colored backgrounds.
20. Text with Background Noise (TBN): This CAPTCHA category has the characters colored with noise background applied.
21. Colored Text No Background (CTNB): The CAPTCHA category has colored texts and a plain background is applied.
22. Character Fragmentation with Gradient Background (CFGB): This CAPTCHA has the characters fragmented, colored, and gradient-type of background is applied.
23. Colored Text Colored Background (CTCB): This CAPTCHA type has colored text with colored background.
24. Character Collapse Background Noise (CCBN): This CAPTCHA has the characters colored, collapsed, and with background noise.
25. Text with Random Lines (TRL): This CAPTCHA has colored characters with random lines background type.
26. Character Squash with Random Lines (CSRL): This CAPTCHA category has its characters squashed with random lines background.
27. Text Distortion with Random Lines (TDRL): This CAPTCHA category has its characters distorted with random lines added as the background.
28. Character Collapse Gradient Background (CCGB): This category of CAPTCHA has colored characters but is distorted with a gradient background.
29. Colored Text Background Noise (CTBN): This CAPTCHA category has colored characters with background noise.

30. Colored Text with Gradient Background (CTGB): This CAPTCHA colored characters with a gradient background are applied.

#### **4.2.1 Feature Comparison of Generated CAPTCHA Categories**

The primary characteristics introduced in the obfuscation module determine the names of the generated CAPTCHA types. Table 4.3 on page 52 enumerates the various CAPTCHA types and specifies the features employed by each type. Although incorporating multiple features could imply numerous CAPTCHA types, some generated CAPTCHAs share the same features as other types.



Table 4.3: Feature Comparison of Variable-Length Accented CAPTCHA Categories

<b>CAPTCHA Category</b>	<b>Color</b>	<b>Squash</b>	<b>Fragmentation</b>	<b>Distortion</b>	<b>Skew</b>	<b>Collapse</b>	<b>Lines</b>	<b>Background</b>
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
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

### 4.2.2 Accented User Keyboard

The E-NaijaCAPTCHA system utilizes accented characters that are not typically found on standard keyboards available in the market. To address this limitation, the keyboard depicted in Figure 4.1 on page 55 was customized for character input specific to the E-NaijaCAPTCHA system. Fifty-one (51) additional characters, including "Á, á, À, à, Â, â, Ã, ã, Ä, ä, Å, å, Æ, æ, Ç, ç, É, é, Ê, è, Ë, ë, Ò, ò, Ó, ó, Ô, ô, Ö, ö, Ø, ø, Ù, ú, Ú, ù, Û, û, Ü, ü, Ý, ý, ÿ," were integrated into the keyboard layout to facilitate user input. The modified keyboard maintains the standard 106-key format, with special character keys adjusted to accommodate the accented characters required for the developed CAPTCHA system.

Figure 4.2 on page 56 illustrates the utilization of the accented user keyboard while responding to an accented CAPTCHA challenge. The graphical interface displays the accented CAPTCHA code, allowing users to respond directly to the challenge without the need for translator assistance.

The implementation of the user keyboard not only reduces the response time for users but also makes variable-length accented character-based CAPTCHA challenges a practical reality. Moreover, the user keyboard serves as the exclusive input method for the developed CAPTCHA system and was implemented using JavaScript for testing purposes.

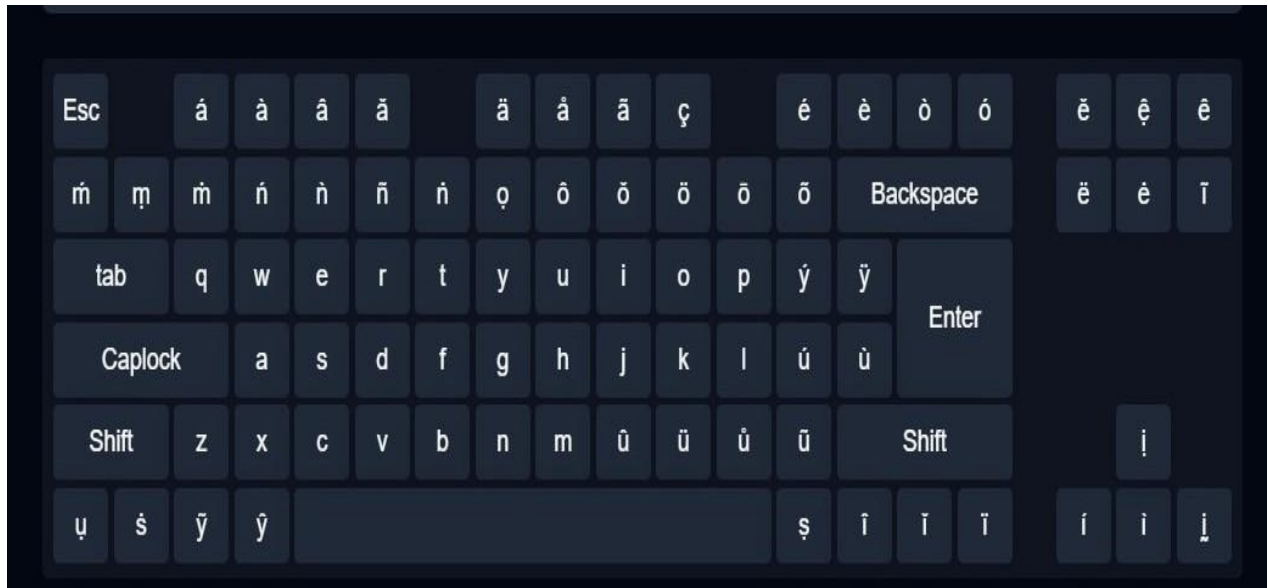


Figure 4.1: Accented Characters Designed User Virtual Keyboard

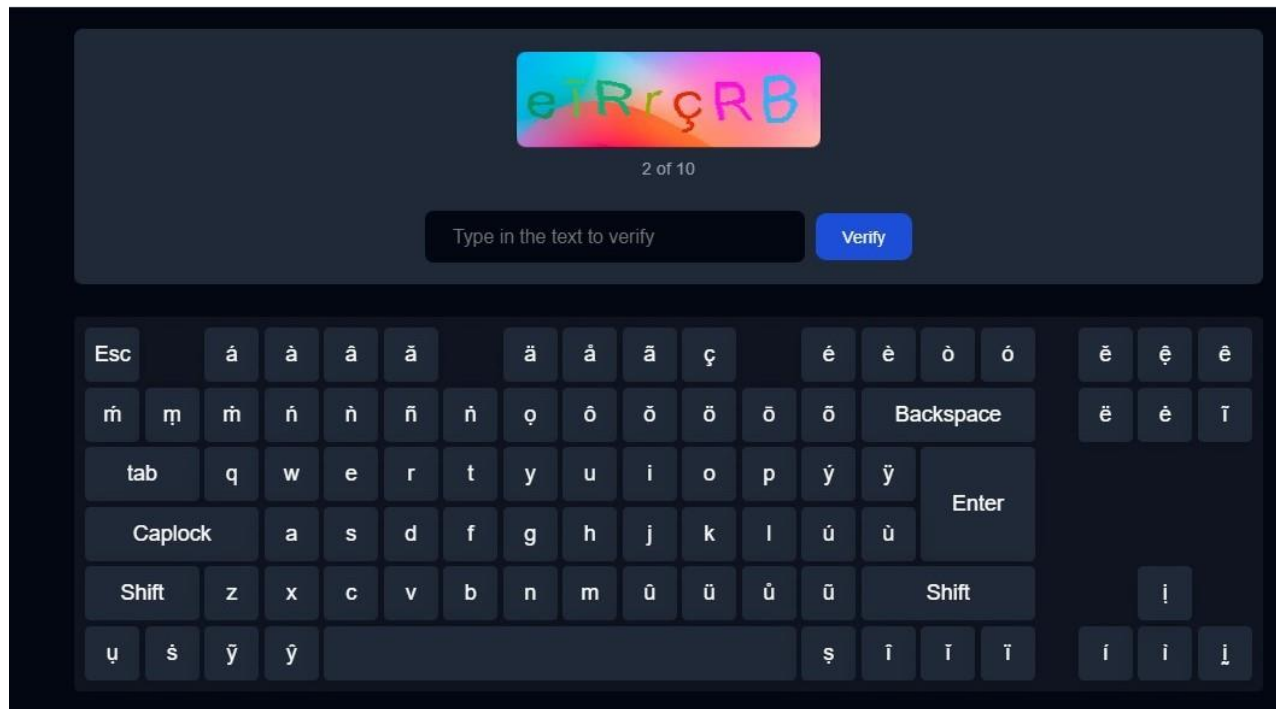


Figure 4.2: User Virtual Keyboard with the Generated E-NaijaCAPTCHA Code

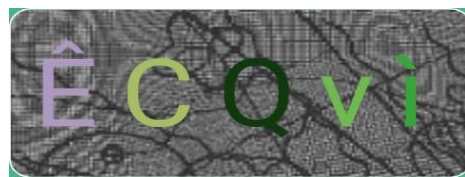
### **4.2.3 Randomly Generated Varied-Length of Accented CAPTCHA Codes**

The length of characters the E-NaijaCAPTCHA generates varies. The system is designed to generate at random a varied length ranging between 4 to 7 characters with two accented characters included in any of the lengths it generates at random after all the specified deformity has been applied to the text. The CAPTCHA code is presented as an image to the user to attempt in line with the identified CAPTCHA categories. Figure 4.3, page 58 shows the variable length of the accented characters randomly generated without duplication of codes.

The CAPTCHA code generator employs predefined rules to generate accented codes, ensuring their security through techniques such as fragmentation, squashing, background noise, color variation, line integration, and character collapse during the obfuscation phase.



**6 Lengths**



**5 Lengths**



**4 Lengths**



**7 Lengths**

Figure 4.3: Randomly Generated Variable-Length Accented CAPTCHA Codes

### **4.3 USABILITY TEST RESULTS**

The outcomes of the experiments conducted to assess the usability of the E-NaijaCAPTCHA system are summarized as follows: a total of five hundred and ten (510) CAPTCHA codes with varied text lengths ranging from 4 to 7 characters were randomly generated, with Texts with No Background (TNB) being the most frequently occurring category, totaling twenty-seven (27) occurrences, while Character Collapse with Random Lines (CCRL) had the lowest occurrence, totaling six (6). The detailed findings are showcased in Table 4.4 on page 60 and Figure 4.4 on page 62.

Performance metrics, including solving time and response time, were recorded and stored in the system's database using embedded code snippets within the usability testing website. The test was conducted among a selected fifty-one group of individuals, both male and female, ranging in age from 12 to 45 years and above of different ethnicity; Yoruba, Igbo, Hausa, and Fulani with each required to respond to ten (10) different CAPTCHA challenges as generated randomly by the system. Participants were provided with instructional guides outlining the system's operation flow to ensure familiarity and smooth interaction with the developed variable-length accented character-based CAPTCHA system (E-NaijaCAPTCHA).



Table 4.4: Number of Occurrences of Accented CAPTCHA Categories during Usability Test

s/n	CAPTCHA Category	Category Acronyms	Number of Occurrences	%
1	Character Fragmentation with Random Lines	CFRL	19	3.73
2	Text No Background	TNB	27	5.29
3	Character Squash with Colored Background	CSCB	15	2.94
4	Character Squash with Background Noise	CSBN	18	3.53
5	Character Collapse Background Noise	CCNB	15	2.94
6	Text with Gradient Background	TGB	16	3.14
7	Text Distortion Background Noise	TDBN	21	4.12
8	Character Fragmentation with Background Noise	CFBN	16	3.14
9	Text with Colored Background	TCB	15	2.94
10	Text Distortion with Colored Background	TDCB	21	4.12
11	Character Squash with Gradient Background	CSGB	21	4.12
12	Character Squash No Background	CSNB	24	4.71
13	Character Collapse with Random Lines	CCRL	6	1.18
14	Text Distortion with No Background	TDNB	22	4.31
15	Text Distortion with Gradient Background	TDGB	15	2.94
16	Character Fragmentation No Background	CFNB	13	2.55
17	Colored Text with Random Lines	CTRL	21	4.12
18	Character Collapse with Colored Background	CCCB	17	3.33
19	Character Fragmentation Colored Background	CFCB	15	2.94
20	Text with Background Noise	TBN	23	4.51
21	Colored Text No Background	CTNB	11	2.16
22	Character Fragmentation with Gradient Background	CFGB	12	2.35
23	Colored Text Colored Background	CTCB	15	2.94
24	Character Collapse Background Noise	CCBN	24	4.71

25	Text with Random Lines	TRL	18	3.53
26	Character Squash with Random Lines	CSRL	14	2.75
27	Text Distortion with Random Lines	TDRL	13	2.55
28	Character Collapse Gradient Background	CCGB	18	3.53
29	Colored Text Background Noise	CTBN	12	2.35
30	Colored Text with Gradient Background	CTGB	13	2.55
	<b>TOTAL</b>		<b>510</b>	<b>100</b>

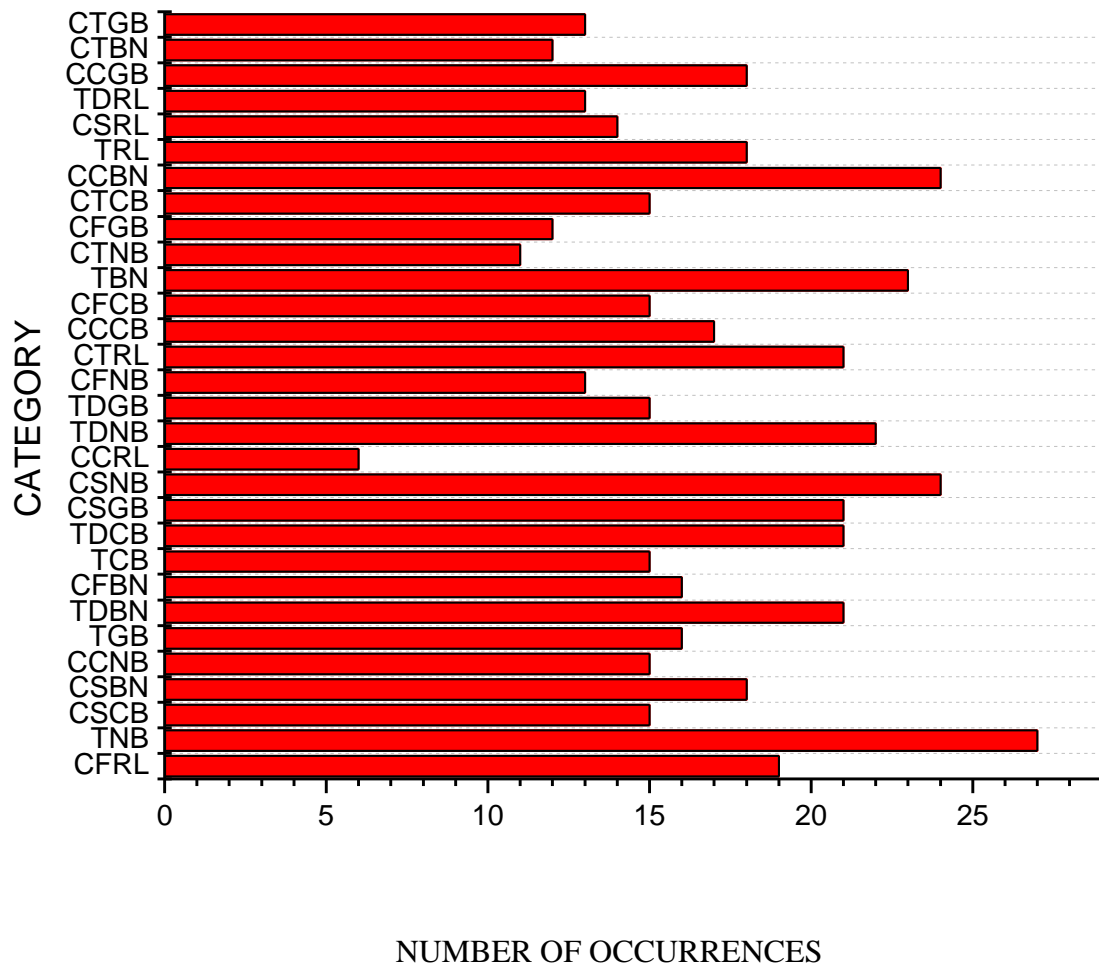


Figure 4.4: Accented CAPTCHA Category Occurrences During Usability Test

### 4.3.1 Success Rate

The success is the proportion of successful attempts relative to the total number of attempts conducted. The results on Table show 4.5 page 64 that the Text Distortion with Random Lines (TDRL) CAPTCHA category had the highest Success Rate at 76.92% followed by Character Fragmentation with Gradient Background (CFGB) at 75.00% while the Character Squash with Gradient Background (CSGB) had the lowest success rate of 28.57%.

Figure 4.5 on page 66 illustrates that the variable-length accented character-based CAPTCHA system achieved a higher number of successful attempts, consequently resulting in a higher overall success rate. CAPTCHA categories such as CCNB, CCBN, and CFBN demonstrated nearly equal proportions of successful and unsuccessful attempts.

Table 4.5: Success Rate of Variable-Length Accented Character CAPTCHA Categories

s/n	CAPTCHA Category	Acronym	Successful	Unsuccessful	Success Rate (%)
1	Character Fragmentation with Random Lines	CFRL	9	10	47.37
2	Text No Background	TNB	17	10	62.96
3	Character Squash with Colored Background	CSCB	11	4	73.33
4	Character Squash with Background Noise	CSBN	13	5	72.22
5	Character Collapse Background Noise	CCNB	8	7	53.33
6	Text with Gradient Background	TGB	11	5	68.75
7	Text Distortion Background Noise	TDBN	11	10	52.38
8	Character Fragmentation with Background Noise	CFBN	6	10	37.50
9	Text with Colored Background	TCB	9	6	60.00
10	Text Distortion with Colored Background	TDCB	13	8	61.90
11	Character Squash with Gradient Background	CSGB	6	15	28.57
12	Character Squash No Background	CSNB	16	8	66.67
13	Character Collapse with Random Lines	CCRL	3	3	50.00
14	Text Distortion with No Background	TDNB	13	9	59.09
15	Text Distortion with Gradient Background	TDGB	7	8	46.67
16	Character Fragmentation No Background	CFNB	7	6	53.85
17	Colored Text with Random Lines	CTRL	11	10	52.38
18	Character Collapse with Colored Background	CCCB	12	5	70.59
19	Character Fragmentation Colored Background	CFCB	7	8	46.67
20	Text with Background Noise	TBN	15	8	65.22
21	Colored Text No Background	CTNB	5	6	45.45
22	Character Fragmentation with Gradient Background	CFGB	9	3	75.00

23	Colored Text Colored Background	CTCB	7	8	46.67
24	Character Collapse Background Noise	CCBN	16	8	66.67
25	Text with Random Lines	TRL	9	9	50.00
26	Character Squash with Random Lines	CSRL	9	5	64.29
27	Text Distortion with Random Lines	TDRL	10	3	76.92
28	Character Collapse Gradient Background	CCGB	12	6	66.67
29	Colored Text Background Noise	CTBN	5	7	41.67
30	Colored Text with Gradient Background	CTGB	5	8	38.46

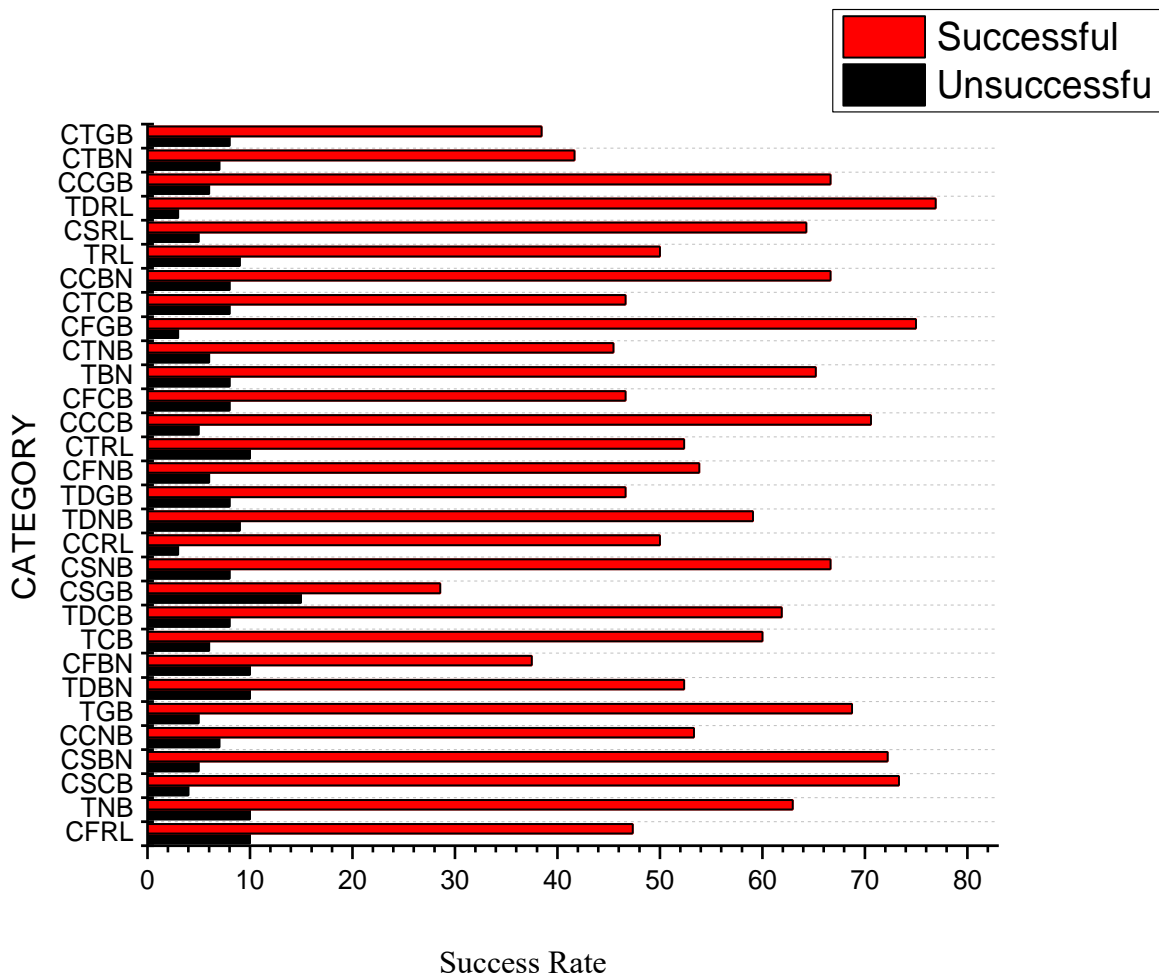


Figure 4.5: Successful and Unsuccessful CAPTCHA Categories Responses

### **4.3.2 Response Time**

Table 4.6 on page 68 and Figure 4.6 page 70 reveals that Character Squash with Gradient Background (CSGB) had the lowest average response time while Character Collapse with Colored Background (CCCB) had the highest average response time of all the responses from the participants on the randomly generated variable-length accented character-based CAPTCHA category.



Table 4.6: Average Response Time (Milliseconds)

s/n	CAPTCHA Category	Acronyms	Number of Generated CAPTCHA Code	Response Time (Milliseconds)
1	Character Fragmentation with Random Lines	CFRL	19	359.62
2	Text No Background	TNB	27	205.50
3	Character Squash with Colored Background	CSCB	15	411.00
4	Character Squash with Background Noise	CSBN	18	259.63
5	Character Collapse Background Noise	CCNB	15	308.25
6	Text with Gradient Background	TGB	16	308.25
7	Text Distortion Background Noise	TDBN	21	256.87
8	Character Fragmentation with Background Noise	CFBN	16	274.00
9	Text with Colored Background	TCB	15	205.50
10	Text Distortion with Colored Background	TDCB	21	411.00
11	Character Squash with Gradient Background	CSGB	21	102.75
12	Character Squash No Background	CSNB	24	376.75
13	Character Collapse with Random Lines	CCRL	6	256.87
14	Text Distortion with No Background	TDNB	22	239.75
15	Text Distortion with Gradient Background	TDGB	15	308.73
16	Character Fragmentation No Background	CFNB	13	256.88
17	Colored Text with Random Lines	CTRL	21	393.88
18	Character Collapse with Colored Background	CCCB	17	462.38
19	Character Fragmentation Colored Background	CFCB	15	359.63
20	Text with Background Noise	TBN	23	188.38
21	Colored Text No Background	CTNB	11	256.88
22	Character Fragmentation with Gradient Background	CFGB	12	274.00
23	Colored Text Colored Background	CTCB	15	359.63

24	Character Collapse Background Noise	CCBN	24	325.38
25	Text with Random Lines	TRL	18	222.62
26	Character Squash with Random Lines	CSRL	14	256.88
27	Text Distortion with Random Lines	TDRL	13	222.63
28	Character Collapse Gradient Background	CCGB	18	256.88
29	Colored Text Background Noise	CTBN	12	222.63
30	Colored Text with Gradient Background	CTGB	13	291.13

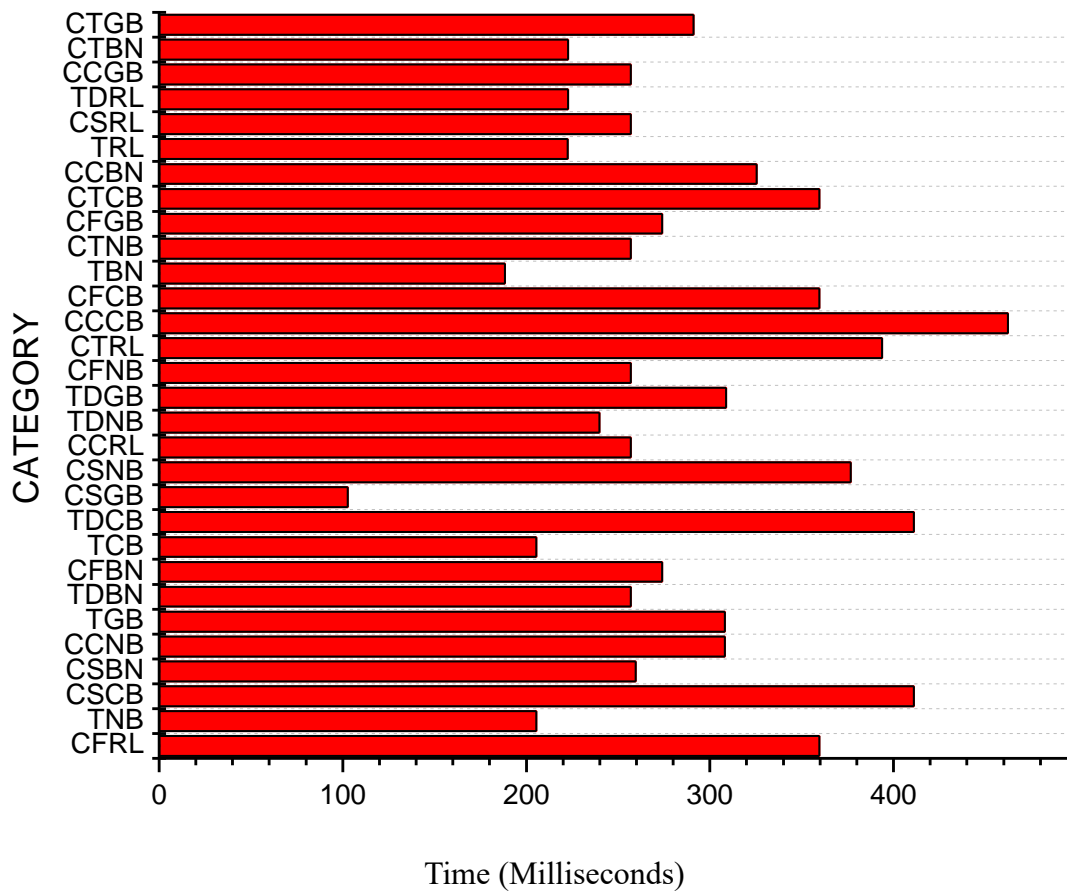


Figure 4.6: Average Response Time

### **4.3.3 Solving Time**

During the usability test, the selected participants demonstrated swift response times, typically solving the presented accented CAPTCHA codes in milliseconds. Referencing Table 4.7 on page 72 and Figure 4.7 on page 74, the average solving time for each CAPTCHA category was documented. Notably, Text with Random Lines (TRL) had the lowest average solving time while Text Distortion with Colored Background (TDCB) had the highest average solving time of all the responses from the participants on the generated CAPTCHA category.

Table 4.7: Average Solving Time for Participants

s/n	CAPTCHA Category	Acronyms	Number of Generated CAPTCHA Code	Solving Time (Milliseconds)
1	Character Fragmentation with Random Lines	CFRL	19	3701.20
2	Text No Background	TNB	27	1263.56
3	Character Squash with Colored Background	CSCB	15	3147.90
4	Character Squash with Background Noise	CSBN	18	3722.71
5	Character Collapse Background Noise	CCNB	15	3420.24
6	Text with Gradient Background	TGB	16	3416.59
7	Text Distortion Background Noise	TDBN	21	3247.47
8	Character Fragmentation with Background Noise	CFBN	16	2820.72
9	Text with Colored Background	TCB	15	1783.42
10	Text Distortion with Colored Background	TDCB	21	6232.31
11	Character Squash with Gradient Background	CSGB	21	1253.27
12	Character Squash No Background	CSNB	24	3829.51
13	Character Collapse with Random Lines	CCRL	6	3032.13
14	Text Distortion with No Background	TDNB	22	1868.66
15	Text Distortion with Gradient Background	TDGB	15	2320.73
16	Character Fragmentation No Background	CFNB	13	2809.17
17	Colored Text with Random Lines	CTRL	21	4627.88
18	Character Collapse with Colored Background	CCCB	17	3703.32
19	Character Fragmentation Colored Background	CFCB	15	4940.12
20	Text with Background Noise	TBN	23	1277.05
21	Colored Text No Background	CTNB	11	2918.29
22	Character Fragmentation with Gradient Background	CFGB	12	4348.79
23	Colored Text Colored Background	CTCB	15	4568.06
24	Character Collapse Background Noise	CCBN	24	2349.85

25	Text with Random Lines	TRL	18	1244.85
26	Character Squash with Random Lines	CSRL	14	4986.78
27	Text Distortion with Random Lines	TDRL	13	2599.80
28	Character Collapse Gradient Background	CCGB	18	3353.643
29	Colored Text Background Noise	CTBN	12	2861.17
30	Colored Text with Gradient Background	CTGB	13	2874.58

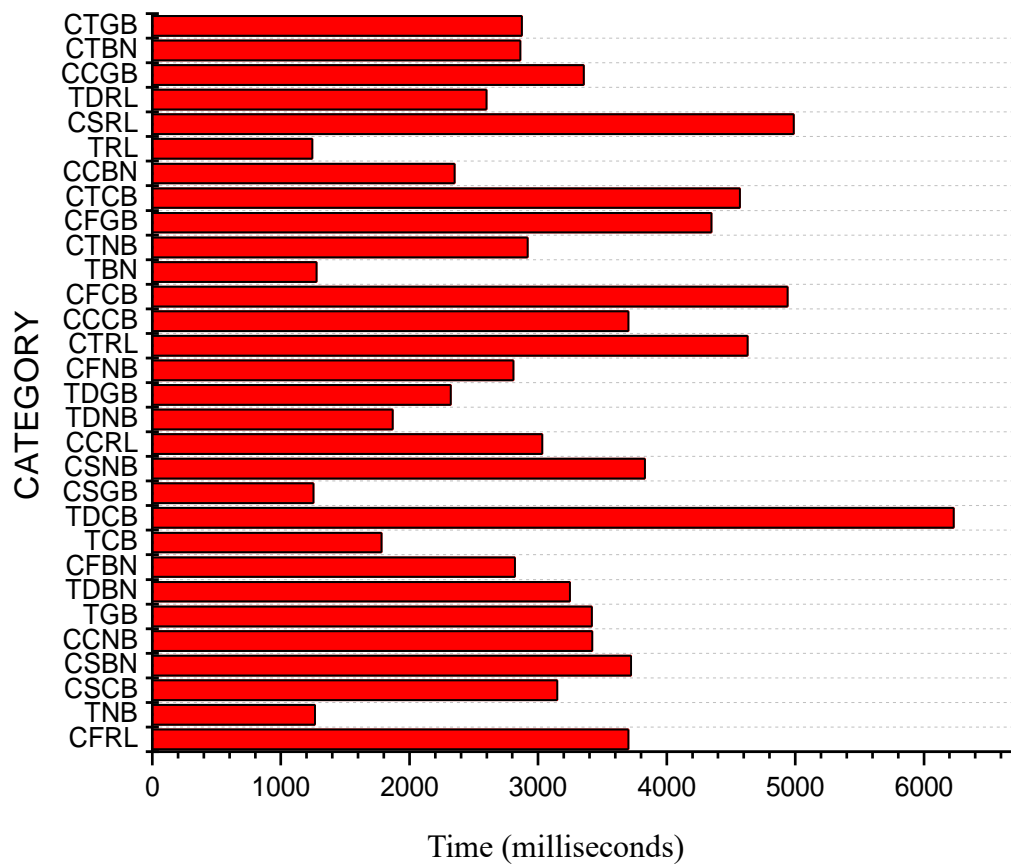


Figure 4.7: Average Solving Time for Participants

#### **4.3.4 Accuracy**

The accuracy of the provided responses by the selected participants for the usability test conducted was collected from the system's developed database through the implemented algorithm within the Enhanced NaijaCAPTCHA system as shown in Table 4.8 page 76 and Figure 4.8 page 78. The report shows that Text Distortion with Random Lines (TDRL) CAPTCHA Category had the highest accuracy of 0.77. Character Squash with Gradient Background (CSGB) had the lowest accuracy of 0.29. This implies that the TDRL Captcha category with the higher accuracy values is more suitable for human authentication.



Table 4.8: Accuracy of Responses on E-NaijaCAPTCHA System

s/n	CAPTCHA Category	Acronym	Correct	Incorrect	Accuracy
1	Character Fragmentation with Random Lines	CFRL	9	10	0.47
2	Text No Background	TNB	17	10	0.63
3	Character Squash with Colored Background	CSCB	11	4	0.73
4	Character Squash with Background Noise	CSBN	13	5	0.72
5	Character Collapse Background Noise	CCNB	8	7	0.53
6	Text with Gradient Background	TGB	11	5	0.69
7	Text Distortion Background Noise	TDBN	11	10	0.52
8	Character Fragmentation with Background Noise	CFBN	6	10	0.38
9	Text with Colored Background	TCB	9	6	0.60
10	Text Distortion with Colored Background	TDCB	13	8	0.62
11	Character Squash with Gradient Background	CSGB	6	15	0.29
12	Character Squash No Background	CSNB	16	8	0.67
13	Character Collapse with Random Lines	CCRL	3	3	0.50
14	Text Distortion with No Background	TDNB	13	9	0.59
15	Text Distortion with Gradient Background	TDGB	7	8	0.47
16	Character Fragmentation No Background	CFNB	7	6	0.54
17	Colored Text with Random Lines	CTRL	11	10	0.52
18	Character Collapse with Colored Background	CCCB	12	5	0.71
19	Character Fragmentation Colored Background	CFCB	7	8	0.47
20	Text with Background Noise	TBN	15	8	0.65
21	Colored Text No Background	CTNB	5	6	0.45
22	Character Fragmentation with Gradient Background	CFGB	9	3	0.75
23	Colored Text Colored Background	CTCB	7	8	0.47
24	Character Collapse Background Noise	CCBN	16	8	0.67
25	Text with Random Lines	TRL	9	9	0.50
26	Character Squash with Random Lines	CSRL	9	5	0.64

27	Text Distortion with Random Lines	TDRL	10	3	0.77
28	Character Collapse Gradient Background	CCGB	12	6	0.67
29	Colored Text Background Noise	CTBN	5	7	0.42
30	Colored Text with Gradient Background	CTGB	5	8	0.38

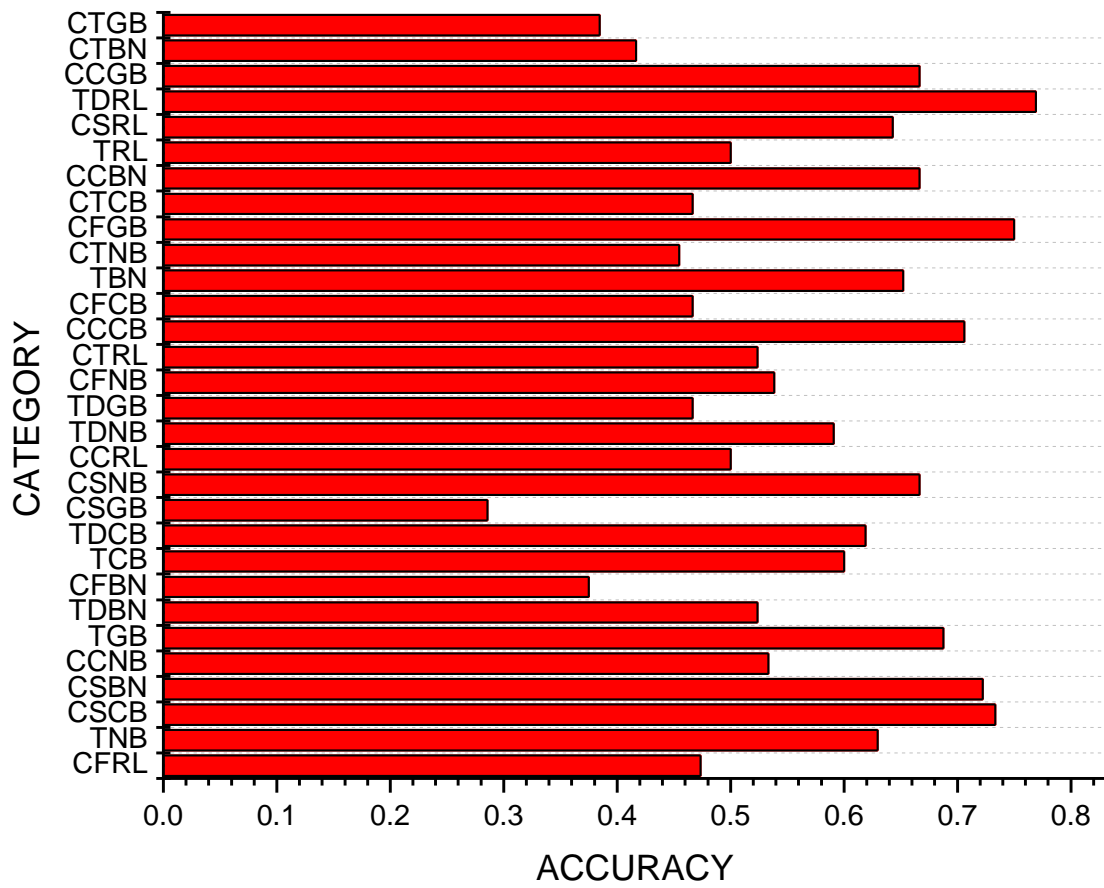


Figure 4.8: Accuracy of Responses on E-NaijaCAPTCHA System

#### **4.4 SECURITY EVALUATION**

The effectiveness of the developed E-NaijaCAPTCHA system, based on variable-length accented characters, against automated programs or bots, was evaluated using a CAPTCHA solver reliant on the Tesseract library. The Variable-Length Accented CAPTCHA codes generated by the system during the Usability Test were employed as input for a simulated bot, and the success rate and accuracy of deciphering the characters on these codes were assessed. A total of 510 varied-length accented CAPTCHA codes from the database were utilized as input for the solver during this evaluation. The distribution of occurrences for each CAPTCHA category is presented in Table 4.4 on page 61.

#### **4.4.1 Security Features of Variable-Length Accented CAPTCHA Categories**

The security attributes of the E-NaijaCAPTCHA system are detailed in Table 4.9 on page 81, encompassing all generated CAPTCHA categories. Each randomly generated varied length of code ranging between 4 to 7 incorporates accented characters along with backgrounds, color variations, fragmentation, skewing, squashing, collapse, and distortion. Among the CAPTCHA categories, those with the highest level of security integration feature seven distinct security attributes. These categories comprise Character Fragmentation with Random Lines (CFRL), Character Squash with Background Noise (CSBN), Character Collapse with Random Lines (CCRL), Colored Text with Random Lines (CTRL), Text with Random Lines (TRL), Character Squash with Random Lines (CSRL), and Text Distortion with Random Lines (TDRL). Conversely, Character Fragmentation with Background Noise (CFBN) exhibits the fewest security features, incorporating only four distinct attributes for security purposes.

Table 4.9: Security Features of Variable-Length Accented CAPTCHA Categories

<b>CAPTCHA Category</b>	<b>Color</b>	<b>Squash</b>	<b>Fragmentation</b>	<b>Distortion</b>	<b>Skew</b>	<b>Collapse</b>	<b>Line</b>	<b>Background</b>
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
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

#### **4.4.2 The CAPTCHA Solver**

Figure 4.9a on page 84 shows the initial interface of the adopted CAPTCHA Solver before any uploads of the generated CAPTCHA code. The variable-length accented CAPTCHA code generated from the database is presented for solving to the solver. This CAPTCHA code is identical to the ones generated during the usability tests. Users initiate the CAPTCHA Solver program by uploading the image through the "Choose File" button on the solver's interface. The program then proceeds to decipher the supplied CAPTCHA code and displays the identified characters alongside it. These identified characters, along with their corresponding accuracy results, are directly saved into the system's database as implemented. The identified characters by the bot do not correspond to the characters on the generated CAPTCHA code, this shows that the automated program could not break the developed E-NaijaCAPTCHA system. This is depicted in Figure 4.9b on page 85.





Figure 4.9a: CAPTCHA Solver Interface Before Upload

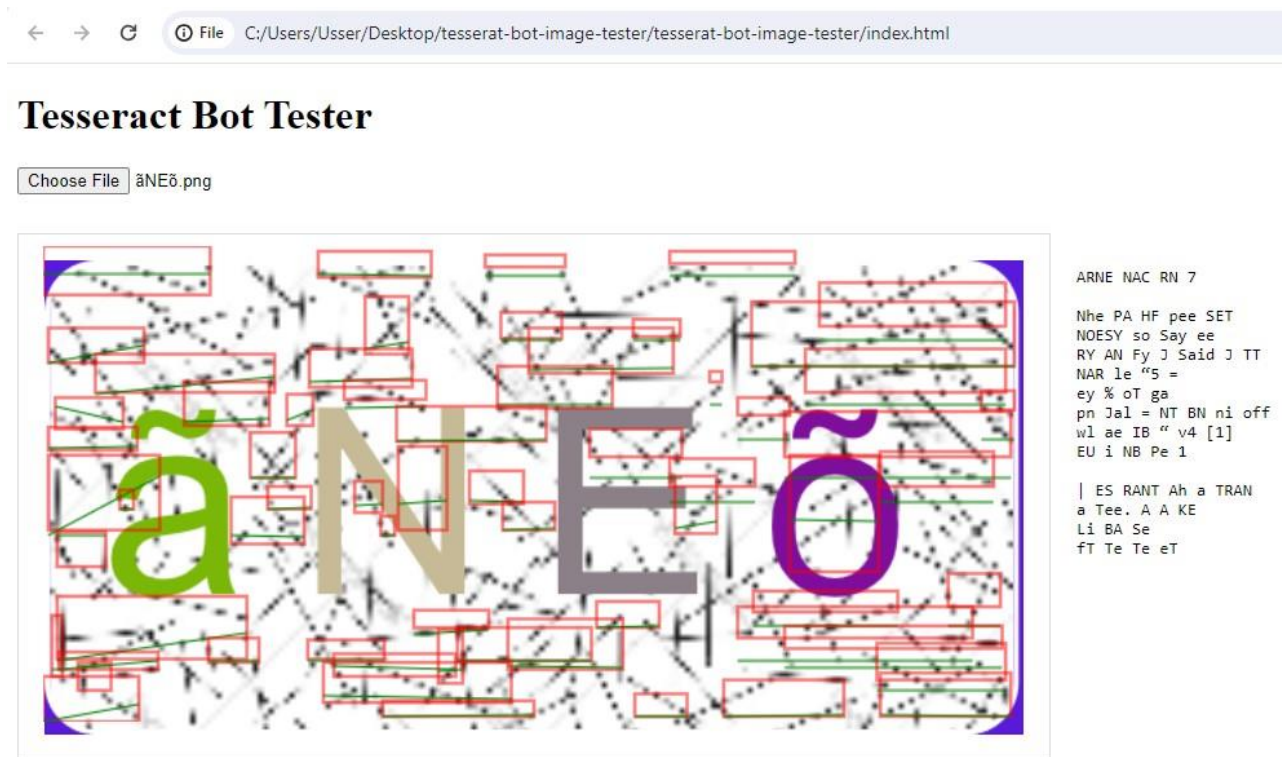


Figure 4.9b: CAPTCHA Solver with Character Squash

#### **4.4.2.1 CAPTCHA Solver's Success Rate**

Table 4.10 page 87 shows the success rate obtained from the security test conducted on the Enhanced-NaijaCAPTCHA system using a CAPTCHA solver based on Tesseract Library. The same generated accented CAPTCHA codes from the database were used as input to the simulated bot. As shown in the table, none of the 510 variable-length accented CAPTCHA codes category uploaded were broken by the solver.

Table 4.10: Solver's Output on the Variable-Length Accented CAPTCHA Categories

s/n	CAPTCHA Category	Acronym	Successful	Unsuccessful	Success Rate
1	Character Fragmentation with Random Lines	CFRL	0	19	0
2	Text No Background	TNB	0	27	0
3	Character Squash with Colored Background	CSCB	0	15	0
4	Character Squash with Background Noise	CSBN	0	18	0
5	Character Collapse Background Noise	CCNB	0	15	0
6	Text with Gradient Background	TGB	0	16	0
7	Text Distortion Background Noise	TDBN	0	21	0
8	Character Fragmentation with Background Noise	CFBN	0	16	0
9	Text with Colored Background	TCB	0	15	0
10	Text Distortion with Colored Background	TDCB	0	21	0
11	Character Squash with Gradient Background	CSGB	0	21	0
12	Character Squash No Background	CSNB	0	24	0
13	Character Collapse with Random Lines	CCRL	0	6	0
14	Text Distortion with No Background	TDNB	0	22	0
15	Text Distortion with Gradient Background	TDGB	0	15	0
16	Character Fragmentation No Background	CFNB	0	13	0
17	Colored Text with Random Lines	CTRL	0	21	0
18	Character Collapse with Colored Background	CCCB	0	17	0
19	Character Fragmentation Colored Background	CFCB	0	15	0
20	Text with Background Noise	TBN	0	23	0
21	Colored Text No Background	CTNB	0	11	0
22	Character Fragmentation with Gradient Background	CFGB	0	12	0
23	Colored Text Colored Background	CTCB	0	15	0
24	Character Collapse Background Noise	CCBN	0	24	0

25	Text with Random Lines	TRL	0	18	0
26	Character Squash with Random Lines	CSRL	0	14	0
27	Text Distortion with Random Lines	TDRL	0	13	0
28	Character Collapse Gradient Background	CCGB	0	18	0
29	Colored Text Background Noise	CTBN	0	12	0
30	Colored Text with Gradient Background	CTGB	0	13	0

#### **4.4.2.2 Accuracy Report**

Table 4.11 page 90 illustrates the accuracy outcomes obtained from the Solver. It is evident from the results that none of the inputs from the E-NaijaCAPTCHA system, when provided to the Solver, were solved accurately in comparison to the database results. This is attributed to the heightened inclusion of random variable-length generation of characters, accented characters, and the various CAPTCHA generation components integrated within the system as part of its security measures.

Table 4.11: CAPTCHA Solver's Accuracy Report

s/n	CAPTCHA Category	CAPTCHA Category	Correct	Incorrect	Accuracy
1	Character Fragmentation with Random Lines	CFRL	0	19	0.00
2	Text No Background	TNB	0	27	0.00
3	Character Squash with Colored Background	CSCB	0	15	0.00
4	Character Squash with Background Noise	CSBN	0	18	0.00
5	Character Collapse Background Noise	CCNB	0	15	0.00
6	Text with Gradient Background	TGB	0	16	0.00
7	Text Distortion Background Noise	TDBN	0	21	0.00
8	Character Fragmentation with Background Noise	CFBN	0	16	0.00
9	Text with Colored Background	TCB	0	15	0.00
10	Text Distortion with Colored Background	TDCB	0	21	0.00
11	Character Squash with Gradient Background	CSGB	0	21	0.00
12	Character Squash No Background	CSNB	0	24	0.00
13	Character Collapse with Random Lines	CCRL	0	6	0.00
14	Text Distortion with No Background	TDNB	0	22	0.00
15	Text Distortion with Gradient Background	TDGB	0	15	0.00
16	Character Fragmentation No Background	CFNB	0	13	0.00
17	Colored Text with Random Lines	CTRL	0	21	0.00
18	Character Collapse with Colored Background	CCCB	0	17	0.00
19	Character Fragmentation Colored Background	CFCB	0	15	0.00
20	Text with Background Noise	TBN	0	23	0.00
21	Colored Text No Background	CTNB	0	11	0.00
22	Character Fragmentation with Gradient Background	CFGB	0	12	0.00
23	Colored Text Colored Background	CTCB	0	15	0.00
24	Character Collapse Background Noise	CCBN	0	24	0.00

25	Text with Random Lines	TRL	0	18	0.00
26	Character Squash with Random Lines	CSRL	0	14	0.00
27	Text Distortion with Random Lines	TDRL	0	13	0.00
28	Character Collapse Gradient Background	CCGB	0	18	0.00
29	Colored Text Background Noise	CTBN	0	12	0.00
30	Colored Text with Gradient Background	CTGB	0	13	0.00



## 4.5 DISCUSSION

E-NaijaCAPTCHA is compatible with Windows, Linux operating systems, and Android Phones, 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 Browser, Safari, and Microsoft Edge, ensuring widespread usability and accessibility.

The development of the Enhanced NaijaCAPTCHA (E-NaijaCAPTCHA) system was to satisfy the curiosity of to what extent the outcomes of the existing NaijaCAPTCHA system (Olanrewaju & Osunade, 2017) that generates a fixed length of characters would impart if the length it generates is increased or decreased, and the addition of some other security features. Pate & Ramteke (2023) 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.*, (2020). 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 & Sapkal, 2012 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 values into these two primary components: the formats of generated characters and the backgrounds utilized. 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 (Chen *et al.*, 2017). 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 (Olanrewaju & Osunade, 2017).

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 standardizing the data entry process, enhancing user convenience and efficiency.

The Usability test was conducted on the E-NaijaCAPTCHA system with the selected 51 participants, for which each participant was tasked to attempt 10 variable-length accented character-based CAPTCHA challenges as it's randomly generated by the system. This is in total of 510 CAPTCHA codes randomly generated. Four performance metrics were utilized to assess the CAPTCHA categories. The recorded occurrences for each category fell within a consistent range of 6 to 19, except for seven categories that exhibited higher values. Participants' success rates in solving the CAPTCHA challenges, facilitated by the provided virtual keyboard, underscore the human-readable nature of the generated CAPTCHA codes. Remarkably, participants swiftly solved the challenges, with response times measured in milliseconds, indicative of the efficiency of E-NaijaCAPTCHA's matching and verification algorithm. Moreover, the accuracy of responses to the CAPTCHA challenges was notably higher for categories featuring Text Distortion. Findings from the usability test suggest that language proficiency did not significantly impact the accuracy level.

E-NaijaCAPTCHA's capability to thwart attacks on web transactions was evaluated using a trained CAPTCHA solver, Tesseract OCR, contrasting with the approach employed by Wang et al. (2019), who utilized Deep Convolutional Neural Network (CNN). Similarly, the Arabic CAPTCHA Scheme was subjected to testing using Arabic OCR, as demonstrated by Khan *et al.*, (2013). E-NaijaCAPTCHA enhances its security through the incorporation of nine security features in CAPTCHA code generation, with the majority of the thirty CAPTCHA categories integrating all these features. Common security attributes across all categories include the utilization of accented characters, distortion, squash, fragment, color, and collapse, whereas a few categories lack features such as lines, skew, and background.

During the security test, 510 randomly generated CAPTCHA codes during the usability test were fed into the solver and evaluated using two metrics. The results revealed a zero success rate across all CAPTCHA categories, indicating E-NaijaCAPTCHA's resilience against bots. Chen *et al.*, (2019) suggested the integration of a selective learning confusion class into text-based CAPTCHA recognition with varying accuracy enhancements of 1.4% to 39.4% specifically for Latin characters. However, this strategy might not yield the same efficacy when applied to non-Latin characters, as Rai (2020) proposed. Rai advocates for bolstering security in text-based CAPTCHA by employing Generative Adversarial Network (GAN) techniques, harnessing accented characters that extend beyond the Latin character repertoire.

#### **4.5.1 Comparative Study of E-NaijaCAPTCHA and NaijaCAPTCHA Systems**

Table 4.12 page 95 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 (Chen *et al.*, 2019).

Table 4.12: Comparison of Features of E-NaijaCAPTCHA and NaijaCAPTCHA Systems

<b>CAPTCHA System Model</b>	<b>Character Length</b>	<b>Generated CAPTCHA Category</b>	<b>Implemented Accented Characters</b>	<b>Latin Character Case</b>	<b>Accented Character Case</b>
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

## **CHAPTER FIVE**

### **SUMMARY, CONCLUSION, AND RECOMMENDATION**

#### **5.1 SUMMARY**

This study delves into bolstering web security through the development and deployment of the E-NaijaCAPTCHA system, which integrates variable-length accented characters into CAPTCHA codes. The research adopts a structured approach, spanning three distinct phases: system development, usability assessment, and security evaluation. Within this framework, the system architecture encompasses four pivotal modules: CAPTCHA Generator, Obfuscator, CAPTCHA Display, and Database, synergistically working to produce secure and user-friendly CAPTCHA codes.

The primary objective of this research is to enhance the existing NaijaCAPTCHA system by introducing the capability for variable code lengths and examining the security implications of both fixed and variable code lengths. Through systematic evaluation, the study assesses system performance utilizing a CAPTCHA solver, underscoring the significance of dynamic CAPTCHA systems in fortifying online security against automated bots and nefarious activities.

The E-NaijaCAPTCHA system presents a repertoire of thirty distinct CAPTCHA categories, each characterized by diverse backgrounds and character modifications. This diversity contributes to a more resilient and dependable web security infrastructure, mitigating the risk of unauthorized access and safeguarding sensitive online transactions.

#### **5.2 CONCLUSION**

Conclusively, this research has effectively demonstrated the creation and deployment of the E-NaijaCAPTCHA system, which integrates random generation of variable-length accented characters ranging from 4 to 7 lengths to bolster web security. Following a systematic approach involving system development, usability assessment, and security evaluation, the study has underscored the efficacy of the CAPTCHA system in distinguishing between legitimate human users and automated bots. It emphasizes the crucial role of dynamic CAPTCHA systems in fortifying online security protocols and mitigating vulnerabilities to malicious activities.

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 unauthorized access. Looking ahead, further research and advancements in CAPTCHA technology hold the potential to enhance online security measures and foster a safer digital environment for users worldwide.

### **5.3 RECOMMENDATION**

Based on the findings and outcomes of this research work, it is recommended that the E-NaijaCAPTCHA system should be:

Utilized by Nigerian industries and companies like Jumia, SLOT, Konga, etc. for transactions conducted on their online platforms.

Used in conjunction with Digital Rights Management (DRM) protocols to safeguard digital contents.

Utilized as a verification code for commenters on the social media platforms of Nigerian enterprises.

Implemented as a security mechanism for mobile devices similar to Personal Identification Numbers (PINs).

### **5.4 CONTRIBUTIONS TO KNOWLEDGE**

This research contributed the following to the existing body of knowledge:

An extension of the existing NaijaCAPTCHA system.

Development of an Enhanced variable-length accented character-based NaijaCAPTCHA system called E-NaijaCAPTCHA system that is secure.

Usability Test for an Enhanced NaijaCAPTCHA system.

Security evaluation of an Enhanced NaijaCAPTCHA system.

Dataset of over 500 generated Variable-Length Accented Character-Based CAPTCHA Category

## **5.5 FUTURE RECOMMENDATION**

During this research, several areas for further exploration and investigation were identified and are recommended for study:

Development of an Enhanced NaijaCAPTCHA system that incorporates handwritten characters in the generation of variable-length accented characters.

Enhancing E-NaijaCAPTCHA's security against evolving cyber threats with advanced measures like biometric authentication and AI-driven anomaly detection.

Perform user experience studies to improve E-NaijaCAPTCHA's design and usability, prioritizing accessibility, inclusivity, and user preferences to boost satisfaction and adoption rates.

Creation of systems to monitor and respond promptly to detected threats by the E-NaijaCAPTCHA, ensuring quick defense actions against security risks.

## REFERENCES

- Abbas, F., Rajput, U., & Dahar, I. (2020). *Enhancing Security of Urdu Language Websites through Urdu CAPTCHA*. 20, 142–152.
- Abubaker, H., Salah, K., Al-Muhairi, H., & Bentiba, A. (2017). Arabic reCAPTCHA Service for Enhancing Digitization of Arabic Manuscripts. *Arabian Journal for Science and Engineering*, 42. <https://doi.org/10.1007/s13369-017-2494-2>
- Agrawal, A., & Baniya, P. (2024). *The Internet of Things: Security Challenges and Opportunities*. <https://doi.org/10.1109/PARC59193.2024.10486356>
- Alammar, A., Al-Yousef, A., & Achour, I. (2022). CAPTCHA Techniques: Types, Benefits, and issues: A Review-Majmaah Kingdom of Saudi Arabia. *International Journal of Computer Network and Information Security*, 485. <https://doi.org/10.22937/IJCSNS.2022.22.5.68>
- Al-Fannah, N. (2017). *Making defeating CAPTCHAs harder for bots*. <https://doi.org/10.1109/SAI.2017.8252183>
- Algwil, A. (2023). *A Security Analysis of Text-based Captcha Schemes*. 2, 309–323.
- Ali, F., & Karim, F. (2014). Development of CAPTCHA system based on puzzle. In *I4CT 2014 - 1st International Conference on Computer, Communications, and Control Technology, Proceedings*. <https://doi.org/10.1109/I4CT.2014.6914219>
- Alnefaie, M. (2020). *A Novel Design of Audio CAPTCHA for Visually Impaired Users*. <https://doi.org/10.13140/RG.2.2.25923.22560>
- Alsuhibany, S., & Parvez, M. (2016). *Secure Arabic Handwritten CAPTCHA Generation Using OCR Operations*. <https://doi.org/10.1109/ICFHR.2016.0035>
- Alsuhibany, S., Parvez, M., Alrobah, N., Almohaimeed, F., & Alduayji, S. (2017). *Evaluating robustness of Arabic CAPTCHAs*. <https://doi.org/10.1109/Anti-Cybercrime.2017.7905268>
- Al-Taie, R. (2014). *A More Robust Text Based CAPTCHA For Security in Web Applications*. 3, 6.
- Althamary, I., & El-Alfy, E.-S. (2017). *A more Secure Scheme for CAPTCHA-Based Authentication in Cloud Environment*. <https://doi.org/10.1109/ICITECH.2017.8080034>
- Banday, M. T., & Shah, N. (2011). *A Study of CAPTCHAs for Securing Web Services*.
- Cabric, M. (2015). *Confidentiality, Integrity, and Availability* (pp. 185–200). <https://doi.org/10.1016/B978-0-12-802934-3.00011-1>



- Chandavale, A., & Sapkal, A. (2012). Security analysis of CAPTCHA. In *Communications in Computer and Information Science* (Vol. 335). [https://doi.org/10.1007/978-3-642-34135-9\\_10](https://doi.org/10.1007/978-3-642-34135-9_10)
- Chen, J., Luo, X., Guo, Y., Zhang, Y., & Gong, D. (2017). A Survey on Breaking Technique of Text-Based CAPTCHA. *Security and Communication Networks*, 2017, 1–15. <https://doi.org/10.1155/2017/6898617>
- Chen, J., Luo, X., Liu, Y., Wang, J., & Ma, Y. (2019). Selective Learning Confusion Class for Text-based CAPTCHA Recognition. *IEEE Access*, PP, 1. <https://doi.org/10.1109/ACCESS.2019.2899044>
- Gao, H., Liu, H., Yao, D., Liu, X., & Aickelin, U. (2010). An Audio CAPTCHA to Distinguish Humans from Computers. *2010 Third International Symposium on Electronic Commerce and Security*, 265–269. <https://doi.org/10.1109/ISECS.2010.65>
- Gao, H., Yao, D., Liu, H., Liu, X., & Wang, L. (2010). A novel image based CAPTCHA using jigsaw puzzle. <https://doi.org/10.1109/CSE.2010.53>
- Jasper, K. D., Raja A.S, V., Ramesh, N., Rajest, S., Rajan, R., & Senapati, B. (2023). *Secure Identity: A Comprehensive Approach to Identity and Access Management*. 171–189.
- Kaur, K., & Behal, S. (2014). Captcha and Its Techniques: A Review. *International Journal of Computer Science and Information Technologies*, 5.
- Kehar, A., Hussain, R., Shaikh, R., Shah, S., Khoso, F., Dahar, I., Jiskani, A., Fatima, S., & Shaikh, H. (2021). Design and Development of Sindhi Text Based CAPTCHAs for Regional Websites. *Sukkur IBA Journal of Emerging Technologies*, 4. <https://doi.org/10.30537/sjet.v4i1.877>
- Khan, B., Alghathbar, K., Khan, K., Alkelabi, A., & Alajaji, A. (2013). Cyber security using Arabic CAPTCHA scheme. *International Arab Journal of Information Technology*, 10.
- Khan, S. A. (2023). *Social Engineering*.
- Kim, J. (2017). Efficiency of Paid Authentication Methods for Mobile Devices. *Wireless Personal Communications*, 93. <https://doi.org/10.1007/s11277-016-3286-9>
- Cluever, K., & Zanibbi, R. (2009). Balancing usability and security in a video CAPTCHA. In *Proc. SOUPS 2009* (Vol. 14). <https://doi.org/10.1145/1572532.1572551>

- Korakakis, M., Magkos, E., & Mylonas, P. (2014). Automated CAPTCHA Solving: An Empirical Comparison of Selected Techniques. In *Proceedings - 9th International Workshop on Semantic and Social Media Adaptation and Personalization, SMAP 2014*. <https://doi.org/10.1109/SMAP.2014.29>
- Kumar, S. (2017). *Enhancing the Security of CAPTCHA based on the New Character Locations*.
- Kumar, M., Jindal, M. K., & Kumar, M. (2022). Design of innovative CAPTCHA for Hindi language. In *Neural Computing and Applications* (Vol. 34, Issue 6). Springer London. <https://doi.org/10.1007/s00521-021-06686-0>
- Ling-Zi, X., & Yi-Chun, Z. (2012). *A Case Study of Text-Based CAPTCHA Attacks*. <https://doi.org/10.1109/CyberC.2012.28>
- Madleňák, M., & Kampová, K. (2022). *Phishing as a Cyber Security Threat*. <https://doi.org/10.1109/ICETA57911.2022.9974817>
- Mehrnezhad, M., Bafghi, A., Harati, A., & Toreini, E. (2017). PiSHi: click the images and I tell if you are a human. *International Journal of Information Security*, 16. <https://doi.org/10.1007/s10207-015-0311-z>
- Mitchell, O., Osazuwa, M., Msc, Cpss, Css, & Fty. (2023). *Confidentiality, Integrity, and Availability in Network Systems: A Review of Related Literature*. 8, 10. <https://doi.org/10.5281/zenodo.10464076>
- Ohiri-Aniche, C. (2007). Stemming the tide of centrifugal forces in Igbo orthography. *Dialectical Anthropology*, 31, 423–436. <https://doi.org/10.1007/s10624-008-9037-x>
- Ojumah, S., Misra, S., & Adewumi, A. (2018). *A Database for Handwritten Yoruba Characters* (pp. 107–115). [https://doi.org/10.1007/978-981-10-8527-7\\_10](https://doi.org/10.1007/978-981-10-8527-7_10)
- Olanrewaju, O. T., Omilabu, A. A., Asoro, B. O., Nwufor, C. V., Adewale, F. O., & Osunade, O. (2023). *Design of an Accented Character-Based Text CAPTCHA System*. <https://zenodo.org/record/8254124>
- Oyeniran, O., Oyeniyi, J., Omotosho, L., & Kazeem, I. (2021). DEVELOPMENT OF AN IMPROVED DATABASE FOR YORUBA HANDWRITTEN CHARACTER. *Journal of Engineering Studies and Research*, 27, 84–89. <https://doi.org/10.29081/jesr.v27i4.302>

- Pate, S. E., & Ramteke, R. J. (2023). *Design and Generation of Devanagari Script CAPTCHA: Imaginative Technique*. Atlantis Press International BV. [https://doi.org/10.2991/978-94-6463-196-8\\_28](https://doi.org/10.2991/978-94-6463-196-8_28)
- Rai, N. (2020). CAPTCHA Recognition using Generative Adversarial Network Implementation. A Project report of Capstone Project-2, School of Computer Science and Engineering. *1613101458/16SCSE101835.13-15*
- Rao, K., Sri, K., & Sai, G. (2016). A Novel Video CAPTCHA Technique To Prevent BOT Attacks. *Procedia Computer Science*, 85, 236–240. <https://doi.org/10.1016/j.procs.2016.05.220>
- Ravi, R. (2019). *CRYPTOGRAPHY AND NETWORK SECURITY PRINCIPLES AND PRACTICE. FIRST EDITION* (p. 63).
- Shirali-Shahreza, M. H., & Shirali-Shahreza, M. (2008). *Advanced nastaliq CAPTCHA*. <https://doi.org/10.1109/UKRICIS.2008.4798962>
- Shivani, & Challa, R. K. (2020). CAPTCHA: A Systematic Review. *2020 IEEE International Conference on Advent Trends in Multidisciplinary Research and Innovation (ICATMRI)*, 1–8. <https://doi.org/10.1109/ICATMRI51801.2020.9398494>
- Svanadze, V., & Gnatyuk, S. (2024). *Challenges and Solutions for Cybersecurity and Information Security Management in Organizations*.
- Tariq, N., & Khan, F. (2018). *Match-the-Sound CAPTCHA* (pp. 803–808). [https://doi.org/10.1007/978-3-319-54978-1\\_99](https://doi.org/10.1007/978-3-319-54978-1_99)
- Toluwalope, D., Akande, & David, T. (2021). *Detection of Denial of Service Attack (DOS)*.
- Ugorji, C. U. C. (2007). A democratic IGBO orthography. *Poznan Studies in Contemporary Linguistics - POZNAN STUD CONTEMP LINGUIST*, 43, 169–180. <https://doi.org/10.2478/v10010-007-0009-0>
- Wang, J., Qin, J., Xiang, X., Tan, Y., & Pan, N. (2019). CAPTCHA recognition based on deep convolutional neural network. *Mathematical Biosciences and Engineering*, 16, 5851–5861. <https://doi.org/10.3934/mbe.2019292>
- Yan, J., & Ahmad, A. (2009). CAPTCHA security: A case study. *Security & Privacy, IEEE*, 7, 22–28. <https://doi.org/10.1109/MSP.2009.84>

- Yang, C., & Hung, J.-L. (2013). An analysis view on password patterns of Chinese internet users. *Nankai Business Review International*, 4. <https://doi.org/10.1108/20408741311303887>
- Yang, T.-C., Ince, I., & Salman, Y. (2009). A Korean CAPTCHA Study: Defeating OCRs In a New CAPTCHA Context By Using Korean Syllables. *International Journal of Contents (IJoC)*, 5. <https://doi.org/10.5392/IJoC.2009.5.3.050>

### 3.1 Code for E-NaijaCAPTCHA System

104

```

const backgroundImages = {
};

const CaptchaBackgroundType = ['NB', 'BN', 'CB', 'RL', 'GB'];
const CaptchaTextType = ['CC', 'CF', 'CT', 'TD', 'T', 'CS'];

function setRandomBackground(characterType, bgType) {
const resultDiv = Id('result');
const typeObj = {
  "type": characterType + bgType,
  "bgImg": backgroundImages[`${bgType}`],
  "characterType": characterStyles[`${characterType}`]
};
resultDiv.style.backgroundImage = `url(../public/images/${typeObj.bgImg})`;
return typeObj.type;
}

function CaptchaType(){
const randomTextType = Math.floor(Math.random() * CaptchaTextType.length);
const randomBgType = Math.floor(Math.random() * CaptchaBackgroundType.length);
const randomTextCaptchaType = CaptchaTextType[randomTextType];
const randomBgCaptchaType = CaptchaBackgroundType[randomBgType];
const JoinTextBgCaptchaType = randomTextCaptchaType + randomBgCaptchaType;
var captchaStyle = '';
switch(JoinTextBgCaptchaType){
case 'CCNB':
captchaStyle = setRandomBackground('CC', 'NB');
break;
case 'CCBN':
captchaStyle = setRandomBackground('CC', 'BN');

```

```

break;
}
return captchaStyle;
}

function getRandomAccentedCharacters() {
const myArray = AccentedCapitalCharacters.concat(AccentedSmallCharacters);
const maxCharacters = 2;
const randomCharacters = [];

for (let i = 0; i < Math.min(maxCharacters, myArray.length); i++) {
const randomIndex = Math.floor(Math.random() * myArray.length);
randomCharacters.push(myArray[randomIndex]);
myArray.splice(randomIndex, 1);
}

let randomizeString = randomCharacters.join('');
return randomizeString;
}

function getRandomEnglishCharacters() {
const minCharacters = 2;
const maxCharacters = 5;
const characters = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
const randomCharacters = [];

const numCharacters = Math.floor(Math.random() * (maxCharacters - minCharacters + 1)) + minCharacters;
for (let i = 0; i < numCharacters; i++) {
const randomIndex = Math.floor(Math.random() * characters.length);

```



```

randomCharacters.push(characters[randomIndex]);
}

return randomCharacters.join('');
}

function randomizeString(inputString) {
const characters = inputString.split('');
for (let i = characters.length - 1; i > 0; i--) {
const j = Math.floor(Math.random() * (i + 1));
[characters[i], characters[j]] = [characters[j], characters[i]];
}
return characters.join('');
}

function trimText(text, minLength, maxLength) {
const length = getRandomLength(minLength, maxLength);
const trimmedText = text.slice(0, length);
return trimmedText;
}

function getRandomIndex(arr) {
return Math.floor(Math.random() * (arr.length + 1));
}

function resizeCanvas(sourceCanvas) {
const borderSize = 10;

```



```

const borderSize = 10;
const patternSize = 20;
const resizedCanvas = document.createElement('canvas');
const resizedContext = resizedCanvas.getContext('2d');
resizedCanvas.width = sourceCanvas.width + 4 * borderSize;
resizedCanvas.height = sourceCanvas.height + 4 * borderSize;
resizedContext.drawImage(sourceCanvas, 0, 0, sourceCanvas.width, sourceCanvas.height);
return resizedCanvas;
}

function ArrayMatcher(generatedCaptcha,userText){
var combinedArray = [];
var percentage = '';
var perPercentage = 0;

for(var i = 0; i < generatedCaptcha.length, i < userText.length; i++){
combinedArray.push(`${generatedCaptcha[i]}:${userText[i]}`);
}

const iterator = combinedArray.values();
const re = new RegExp(":");
percentage = 100/generatedCaptcha.length;
for (const letter of iterator) {
var result = letter.split(re);
if(result[0] === result[1]){
perPercentage += percentage;
}
}
return `${perPercentage.toFixed(2)}%`;
}

```