



Web-Based Text Editing System for Nigerian Major Languages

A. A. Kasali^{a,*}, K. O. Jimoh^b, M. A. Adeagbo^c, S. A. Bello^d

^aInnovation Technology Centre, The Federal Polytechnic Ede, Osun State, NIGERIA.

^bDepartment of Information and Communication Technology, Osun State University, Osun State, NIGERIA.

^cMathematics & Computational Sciences Department, First Technical University, Ibadan, Oyo State, NIGERIA.

^dDepartment of Computer Science and Engineering, Obafemi Awolowo University, Ile-Ife, Osun State, NIGERIA.

Abstract

This study developed a web-based text editor to eliminate the incompatibility of computer keyboard with the three major indigenous languages in Nigeria. It also aims to reduce the time taken to produce characters with diacritical marks. The editors produced valid Unicode characters and require pressing less buttons to generating all the symbols of the alphabets for the three major indigenous languages in Nigeria. Client-side technologies were used to develop these applications. Three web pages, designated for Yorùbá, Igbo and Hausa language were generated with HTML. CSS was used to define the look and feel of the HTML elements on each page. Regular Expressions implemented in JavaScript functions were used to convert selected ASCII characters into desired Unicode characters. The editors are available at <http://www.gazaliwakil.com.ng>. The editors work well on latest version of browsers like (Google Chrome, Mozilla Firefox, Opera, and Internet Explorer). They are very light, consume minimal server resources and can work offline. The system was launched Fifty-one (51) times to extract data comprising the Loading, Scripting, Rendering, Painting, System, and Idle time. The obtained result showed that on the average, it takes about 13.77ms to load the HTML DOM elements, 42.83ms to load the javaScript, 13.10ms and 1.73ms for rendering and painting the page by CSS. Additional time taken are 43.91ms and 3,045.10ms for the system and idle time respectively. A total time of 3,160.43ms (3.16s) is required when any of the editors is launched before the page can accept inputs from the users. It also takes the editors 2.66ms to add diacritical marks on a letter. This would, in effect, not reduce the typing speed of users.

Keywords: text editor, indigenous, language, language extinction, Yorùbá, Igbo, Hausa

1. INTRODUCTION

According to [1], language is tied to the culture of a group of people. No culture can thrive without language, just as no society can exist without language. Language is an important tool in the society, because man needs it to share his ideas, experiences, emotions, and interact with other people in the society or in his environment [2]. It is used by man as a means of social, cultural transmission as well as a means of knowledge advancement and educational development [3]. It is an unavoidable tool in every developmental effort [4].

Nigeria is blessed with diversities in culture, tradition, religion, colour, and language is not an exception. Nigeria has about 529 languages out of which only three languages are recognized as major languages namely: Hausa, Igbo and Yorùbá [5]. According to the Nigerian Language Policy,

these three languages belong to the class A category of the three categories of the functionality of Nigerian languages [6]. These languages are well studied and their writing systems have been standardized. They are tonal languages with three phonological contrastive tones: High, Mid and Low.

Hausa is a Chadic language predominantly spoken in the Northern Nigeria. The dialect of Kano serves as its standard. Before and immediately after independence, it was used as the official language of the Northern Region [7]. It used to be written in *ajami* (a version of the Arabic script) since 17th century. However, during the 19th century, a version of the Hausa written with the Latin alphabet (also known as *boko*) began to emerge. *Ajami* and *boko* have been in use until 1950s when *boko* eventually replaced the *ajami* and became the main alphabet for most Hausa speakers. Hausa language has three tones: low, high and falling. The low tone is indicated by a grave accent (˘). The falling tone is indicated by an upper caret (ˆ) and the high tone is marked with an acute

*Corresponding author (Tel: +234 (0)703 033 7708)

Email addresses: gabdulwakil@gmail.com (A. A. Kasali), kudirat.jimoh@uniosun.edu.ng (K. O. Jimoh), adedegy@gmail.com (M. A. Adeagbo), apinkebello@yahoo.com (S. A. Bello)

accent (˘). There are twenty-nine (29) letters and glottal stop (ʔ) in the Hausa *boko* Latin alphabet. Table 1 shows the upper and lower case of the letters.

Table 1: Hausa *Boko* latin alphabet in upper and lower case.

A a	B b	Ɓ ɓ	C c	D d	Ɗ ɗ	E e	F f	G g
H h	I i	J j	K k	Ƙ ƙ	L l	M m	N n	O o
R r	Ƴ ƴ	Ƙ ƙ	Ƙ ƙ	T t	Ƨ ɗ	U u	W w	Y y
Y y	Z z	Ƴ ƴ						

Source [8]

Yorùbá is a member of the Volta-Niger branch of the Niger-Congo family of languages [8]. It is predominantly spoken in the South-Western Nigeria. Its orthography first appeared during the mid-19th century. Today, the dialect of Oyo has been adopted as its standard after several changes. Yorùbá language has three tones: low (otherwise known as *do*), mid (otherwise known as *re*) and high (otherwise known as *mi*). The low tone is indicated by a grave accent (˘). The mid tone is not marked and the high tone is marked with an acute accent (´). There are twenty-five (25) letters in the Yorùbá alphabet. Table 2 shows the upper and lower case of the letters.

Table 2: Yorùbá alphabet in upper and lower case.

A a	B b	D d	E e	Ɛ ɛ	F f	G g	Gb gb	H h
I i	J j	K k	L l	M m	N n	O o	Ọ ọ	P p
R r	S s	Ɔ ɔ	T t	U u	W w	Y y		

Source [8]

Igbo is also a member of the Volta-Niger branch of the Niger-Congo family of languages. Igbo is predominantly spoken in the South-Eastern Nigeria. The dialect of Owerri and Umuahia are its reference and has been in use since 1962. Igbo is also a tonal language though tones are sometimes marked in writing: the low tone with a grave accent (˘), and the high tone with an acute accent (´). There are thirty-six (36) letters in the Igbo alphabet. Table 3 shows the upper and lower case of the letters.

Writing, according to [8], is a method of representing language in visual or tactile form. Sets of symbols are used to represent the sounds of speech, punctuations and numerals. The system in which these symbols are used is known as the writing system. A system in which more or less permanent marks are used to represent an utterance in such a way that it can be recovered more or less exactly without the intervention of the utterer [9].

According to [10], a complete writing system should fulfill all the following criteria:

Table 3: Igbo Alphabet in Upper and Lower Case.

A a	B b	Ch ch	D d	E e	F f	G g
GB gb	Gh gh	Gw gw	H h	I i	I ị	J j
K k	Kp kp	Kw kw	L l	M m	N n	Nw nw
Ny ny	Ñ ñ	O o	O ọ	P p	R r	S s
Sh sh	T t	U u	U ụ	V v	W w	Y y
Z z						

Source [8]

- i. it must have its purpose of communication;
- ii. it must consist of artificial graphic marks on a durable or electronic surface; and
- iii. it must use marks that relate conventionally to articulate speech (the systematic arrangement of significant vocal sounds) or electronic programming in such a way that communication is achieved.

The Federal Government of Nigeria through the National Commission for Mass Literacy, Adult and Non-Formal Education has urged Nigerians to embrace the use of indigenous languages in education and literacy as key towards a more literate and sustainable society [11]. However, Research has mostly focused on English and many other languages; the same cannot be said of Nigerian and most other African languages due to the lack of online or electronic presence of these languages [12]. The use of computers, according to [13], has so far been greatly restricted only to those people who have some knowledge of the English language. This has resulted in the limited number of materials (instructional or otherwise) produced in indigenous languages that would have increased awareness and strengthened development. Also, this has resulted in the gradual extinction of the indigenous languages in Nigeria. Many people are willing to write or share their ideas in their indigenous languages (the languages in which they are most creative and innovative) but the available writing tools (Text editors or Word processors) have no facility and even if available, make it difficult to write and eventually discourage them.

Microsoft Office Word, the famous word processor in the area where the authors live, provides keyboard shortcuts for international characters. For instance, when control (ctrl) key and accent grave (˘) key are pressed simultaneously (Ctrl+˘) and released, then a letter is pressed, Word will automatically add accent grave on top of the letter. Pressing Ctrl+' (apostrophe) and followed by pressing a letter will add accent acute on top of the letter. To add caret or tilde on top of a letter, ctrl+shift+^ or ctrl+shift+~ must be pressed respectively before pressing the letter. To put dot under a letter, the letter has to be typed together

Function that Copies the Content of TextArea A to Clipboard

```

$("#copy").click(
  function(){
    $("#textEditor").select();
    document.execCommand("copy");
  }
);

```

Figure 1: Function that copies the Content of TextArea A to Clipboard.

with 0323 (Unicode for dot under) without space, then pressing Alt+X will add dot under the letter [14].

2. RELATED WORKS

Efforts have been made by researchers and developers to address the difficulties encountered while writing Nigerian indigenous languages using the available text editors and word processors. Takada, a Yorùbá text editor, was developed by [15]. Takada is a windows-based application. It was developed using Python programming language. The weakness of Takada is that to type a character with a diacritic, it requires three (3) steps: the character must be typed, it must be highlighted (this step consumes time), and the diacritic button on the toolbar must be clicked or appropriate key combination be pressed (ctrl+` for grave accent, ctrl+´ for acute accent and ctrl+. for dot under). Another one is its platform dependency. Linux and Mac users can not benefit from this application. LyTextEditor, a standalone Hausa text editor, was developed by [16]. The editor was purposely developed to test a spell corrector, LySpell, which is the meat of the work. Both applications were written in Java programming language making it deployable to Windows, Linux and other operating systems.

[17] addressed the problem of platform dependency attributed to Takada by developing web-based text editors for seventy-seven (77) of the world languages. The languages of concerned (Yorùbá, Igbo, and Hausa) are among the languages addressed by the developer. However, the learning curve of the editors needs to be addressed. For instance, in the Yorùbá editor, letter “x” is used for adding grave accent (low tone) on e and o, while “;” is used for performing the same function (adding low tone) on a, e, i, o, and u. In the same editor, “z” is used for adding high tone on e and o while “v” is used for performing the same function (adding high tone) on a, e, i, o, and u. Sharing the same function between two keys would increase the learning curve of the users and as well reduce their typing speed. Also, the angular bracket, [, is used to enable the entry of e, o, and s making the angular bracket not typeable in the editor. In the Igbo editor, the grave accent (´) is used for adding dot under u, i and o. Also, “[” is used for adding ñ making the angular bracket not typeable in the editor. In the Hausa editor, the

grave accent (´) is used to enable the entry of implosive letters (Ḅ Ḍ Ḳ Ỵ) and the ÀÈÌÒÛ. Also, “[” is used to enable the entry of ÁÉÍÓÚ making the angular bracket not typeable in the editor. The concept of [18] is similar to [17] but unfortunately, there is no information on how to use it.

Many speakers, because of their fluency, of the languages prefer to type the letters and words before adding diacritics on the letters. Unfortunately, the reviewed editors (except Takada) support only immediate entry of letters with their diacritics and do not support adding diacritic after the entry of letters. However, the above reviewed works have high learning curve except Takada (but Takada requires many steps to add diacritic marks which, in turn, introduce delay in typing). This study, therefore, developed web-based text editors for the three major indigenous languages in Nigeria to reduce the learning curve of users, increase the typing speed of users, address platform dependency and keyboard compatibility issue. The tool, being a web-based, will increase the public profile of these languages because the World Wide Web (WWW) is accessible all over the world.

3. MATERIALS AND METHODOLOGY

The material and methods used for the implementation of the text editor is discussed in this section. Client-side technologies like HTML, CSS, and JavaScript were used to develop this application. Three web pages were developed with HTML. Each page is designated for Yorùbá, Igbo and Hausa language. Attribute *lang* of the html tag in each page was set to “yo”, “ha”, and “ig” for Yorùbá, Hausa and Igbo page respectively. Also, attribute *charset* of meta tag was set to Unicode “utf-8”. Each page contains two TextArea (**TextArea A** and **TextArea B**) boxes, two buttons and a div tag. The div tag is used to hold the instructions on how to use the language editor. **TextArea A** is used to accept input from the users and also display output to the users. **TextArea B** is a hidden box used to carry out background processes. One of the buttons is used to copy the content of **TextArea A** to clipboard (code sample is shown in Fig. 1), while the second button is used to clear the content of **TextArea A** (code sample is shown in Fig. 2).

The look and feel of the pages were achieved using Cascading Style Sheet (CSS). The business logic of this application was written in JavaScript.

Function that Clears the Content of TextArea A

```

$("#new").click(
  function(){
    $("#textEditor").val("");
  }
);
    
```

Figure 2: Function that Clears the Content of TextArea A.

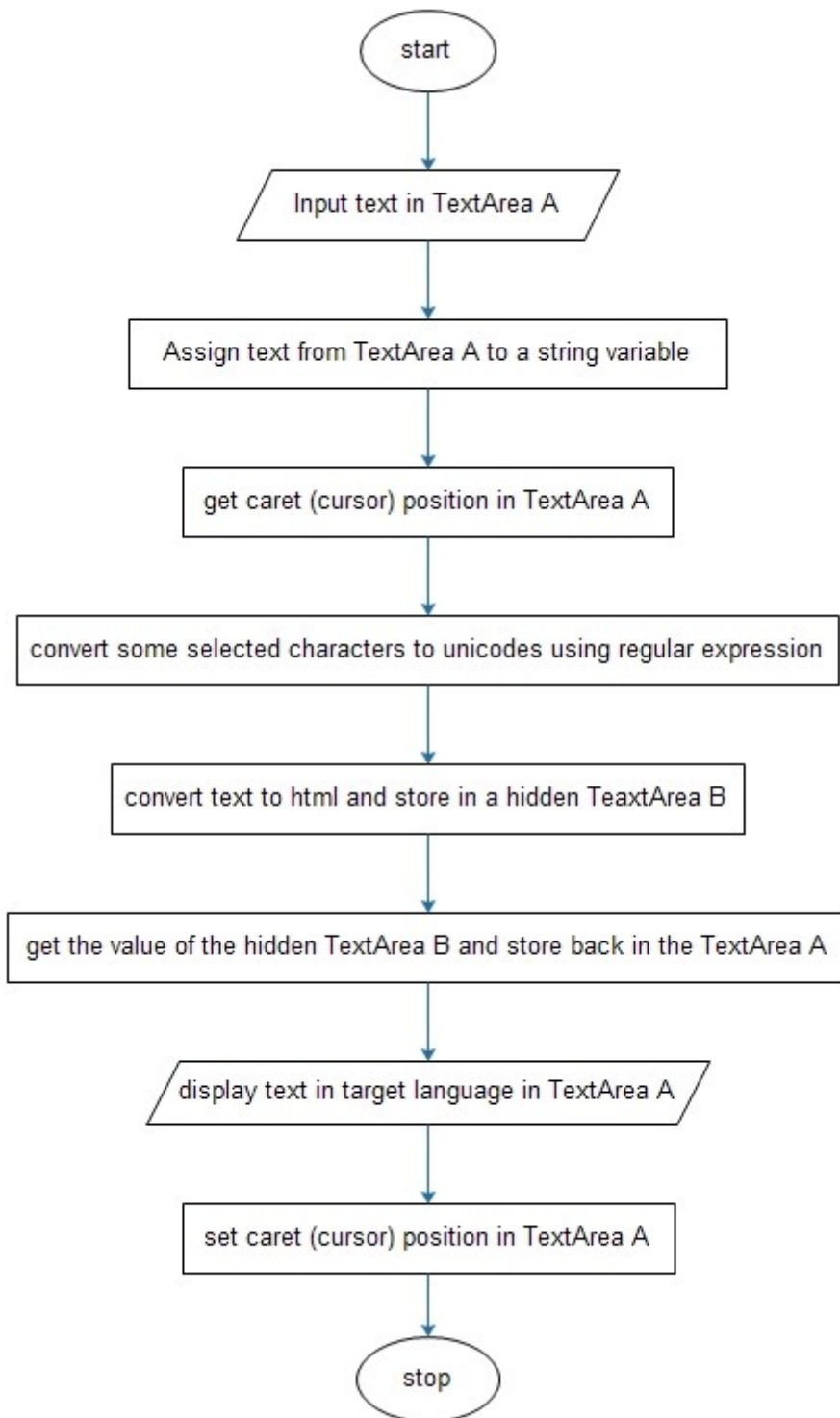


Figure 3: Flowchart for Inputting and generating Yorùbá, Igbo or Hausa Texts.

Function that Submits Text and Calls generateYoruba Function

```

$(("#textEditor").keyup(
  function(){
    var txt = $("#textEditor").val(); //Assign text from TextArea A to variable txt
    generateYoruba(txt);           //call generateYoruba(txt)function
  }
);

```

Figure 4: Function that Submits Text and Calls generateYorùbá Function.

Function that Generates Yoruba Texts

```

function generateYoruba(txt){

  //get caret(cursor) position in the textEditor
  var editor = document.getElementById("textEditor");
  var pos = 0;
  if("selectionStart" in editor){
    pos = editor.selectionStart;
  } else if("selection" in document){
    editor.focus();
    var Sel = document.selection.createRange();
    var SelLength = document.selection.createRange().text.length;
    Sel.moveStart("character", -editor.value.length);
    pos = Sel.text.length - SelLength;
  }

  //converts the selected characters to unicodes using regular expressions
  txt = txt.replace(/Q/g, "&#768;");
  txt = txt.replace(/X/g, "&#769;");
  txt = txt.replace(/c/g, "&#803;");
  txt = txt.replace(/v/g, "");
  txt = txt.replace(/z/g, "");

  // convert text to html and store in a hidden TextArea B(id = content)
  $("#content").html(txt);
  txt = $("#content").val();

  // get the value of the hidden TextArea B (content) and store back in the TextArea A (textEditor)
  $("#textEditor").val(txt);

  // set caret(cursor) position in TextArea A (textEditor)
  setCaretToPos(editor, pos);
}

```

Figure 5: Function that Generates Yorùbá texts.

Table 4: Evaluation Performance for Retrieved Data from Web Page Testing.

S/N	Range (ms)	Load (ms)	Script(ms)	Render (ms)	Paint(ms)	System (ms)	Idle (ms)
1	3141	10.8	40.2	15.2	1.8	36.3	3037.2
2	3131	11.1	35	10.4	1.3	30.8	3042.5
3	3138	18.4	43.2	13.8	1.7	33.5	3027.9
4	3138	12.6	44.1	13.3	1.2	39	3027.3
5	3124	8.2	32.9	14	4.4	26.4	3037.9
6	3261	26.4	53	12.8	1.9	45.1	3122.1
7	3196	23.1	36.5	10.3	1	52.3	3072.6
8	3180	18.6	44.4	23.1	2.1	50.8	3040.6
9	3160	12	36.3	10.8	1	40.8	3058.8
10	3164	10.9	44.7	14.6	3.1	44.1	3047.1
11	3161	11.4	45	15.1	2.9	42.5	3044.6
12	3179	27.9	37.3	12.3	1.1	33.8	3066.6
13	3148	19.8	31.2	12.3	1.3	53.5	3030.3
14	3153	12	42.1	12.3	1.3	50.5	3034.3
15	3193	15.4	72.4	12.2	13.8	34.2	3045.6
16	3187	18.9	34.2	13	1.2	41.3	3078.5
17	3149	14.9	33.4	11.9	1	43.3	3044.3
18	3156	8.5	37.6	12.7	1.7	51.6	3043.5
19	3141	8.7	35.5	13.4	1.3	41.8	3040.1
20	3170	8.4	35.6	11.1	1.3	63.3	3050.2
21	3139	10.3	48.1	10.6	1.2	37.2	3031.1
22	3139	7	45.3	15.9	1	29.5	3040.1
23	3133	9.1	48.4	13.9	1.2	73.7	2986.6
24	3155	11.2	41.4	11.2	1.4	41.5	3048.6
25	3145	9.3	43.8	13.7	1.3	37.8	3038.8
26	3154	18.3	38.3	12.8	1	40.2	3043.9
27	3145	10.4	42.6	11.2	1.1	38.1	3041.4
28	3143	14.8	41.9	15	1	42.9	3026.9
29	3152	14.6	41.9	13.1	1.1	43.3	3037.9
30	3165	24.7	32.4	9.8	1.1	41.2	3055.8
31	3167	10.9	56	8.9	1.1	55	3035.5
32	3155	10.5	38.6	16.7	2.2	47.8	3039.5
33	3155	12.6	42.2	10.7	1.1	40.2	3047.9
34	3170	9.9	37.7	12.2	1.4	43.7	3065.4
35	3184	13.5	42.4	13.4	1.5	69.3	3043.8
36	3162	14	44.4	10.6	1.1	49.1	3042.6
37	3160	9.8	40.5	14.6	1.7	46.2	3047.8
38	3222	10.8	37.4	11.5	1.1	49	3112.2
39	3153	14.8	39.9	12.7	1.3	41.3	3043
40	3156	14	39.6	15.7	1.5	48.8	3036.7
41	3187	25	96.7	10.3	1.1	21	3033.4
42	3160	8.9	42.9	11.9	1.2	48.9	3046.1
43	3159	18.8	37	12.2	1	46.9	3043
44	3149	12.4	40.6	16	1.3	39	3039.9
45	3152	10.9	38.2	15	2.3	47.6	3038.2
46	3165	20.2	44.5	13.6	1	37.7	3047.5
47	3157	10.3	47.5	14.2	1.3	50.1	3033.3
48	3163	11.9	49.3	14	1.7	41.4	3044.5
49	3177	15	55.5	15.3	3	53.9	3034.7
50	3151	10.1	40.3	13.7	1.6	41.1	3043.8
51	3138	10.3	34.5	13.3	1	40.9	3038.1
Average	3,160.43	13.77	42.83	13.10	1.73	43.91	3,045.10

Table 5: Computational Time for each Character in the Sentence considered.

S/N	Character	Computational Time (ms)	Computational Time of Selected Characters (q, c, and x) (ms)	Computational Time of Other Characters (ms)
1	A	1.045		1.045
2	d	0.49		0.49
3	e	0.495		0.495
4	x	4.175	4.175	
5	w	1.77		1.77
6	a	0.48		0.48
7	x	2.745	2.745	
8	l	0.595		0.595
9	e	0.485		0.485
10	x	2	2	
11	Space	0.56		0.56
12	l	0.54		0.54
13	o	0.5		0.5
14	c	2.65	2.65	
15	Space	0.43		0.43
16	s	0.51		0.51
17	i	0.515		0.515
18	x	2.46	2.46	
19	Space	0.51		0.51
20	o	0.495		0.495
21	c	1.945	1.945	
22	j	0.52		0.52
23	a	0.505		0.505
24	q	2.665	2.665	
		Average	2.66	0.61

The flowchart of the text editors for the languages is as shown in Fig. 3. Using Yorùbá editor as an example, id attribute of *TextArea A* was set to “textEditor”. A *keyup* function (an event that triggers whenever a key is released after being pressed) was attached to *textEditor* as shown in Fig. 4. When a character is entered in the textEditor, the *keyup* function is fired.

Consequently, the value of the *textEditor* is assigned to a variable *txt* and then function *generateYorùbá(txt)* is called. Variable *txt* is passed as an argument to the function *generateYorùbá(txt)*. Fig. 5 shows the *generateYorùbá(txt)* function. When this function is called, it gets the caret(cursor) position in *TeaxtArea A*; converts some selected characters to unicodes using regular expression; converts text to html and store in a hidden *TextArea B*; gets the value of the hidden *TextArea B* and store back in the *TextArea A*; and finally sets the caret(cursor) position in *TeaxtArea A*.

Operation performed at each step in the flowchart is the same for the three languages except the step that *converts some selected characters to Unicode using regular expressions*. Letter Q, X, C, V, and Z are not in the alphabet of Yorùbá language. They must not appear in the editor. Therefore, letter Q, X, and C are used to add accent “do”, “mi” and put dot under preceding letter respectively. This is done by calling *replace* method of variable *txt*. The method requires two parameters, the *text to be replaced* and the *replacing text*. The text to be replaced is placed between two forward slash characters. Letter g and i are modifiers. Letter g perform a global match (find

all matches rather than stopping after the first) while letter i perform case-insensitive matching. Unicode and null are used as the replacing texts. Letter Q is replaced with “&# 768;”, X with “&# 769;”, C with “&# 803;”, V with “”, and Z with “” to display accent “do”, “mi”, put dot under preceding letter, remove letter V and Z respectively.

In the case of Igbo language, letter Q and X are not in the alphabet. Therefore, letter Q is replaced with “&# 771;”, X with “&# 803;” to put dot on top and under the preceding letter respectively. Character ‘ is replaced with “&# 768;”, apostrophe () with “́” to display accent “low tone”, and “high tone” on the preceding letter respectively.

In the case of Hausa language, letter Q, X, P, and V are not in the alphabet. Therefore, letter Q is replaced with “,”&# 768;”, X with “&# 769;”, P with “&# 770;”, and V with “&# 771;” to display accent “low tone”, “high tone”, “falling tone”, and put tilde character on top of the preceding letter respectively. There are special letters in Hausa language. These letters are implosive Bb () implosive Dd(), implosive Kk(), and implosive Yy(). They are taken care of by replacing the sequence of ‘B with “&# 385;”, ‘b with “&# 595;”, ‘D with “&# 394;”, ‘d with “&# 599;”, ‘K with “&# 408;”, ‘k with “&# 409;”, ‘Y with “&# 435;”, and ‘y with “&# 436;”.

4. RESULTS AND DISCUSSION

The editors were tested using Profiling tool in Google Chrome Version 89.0.4389.90 (Official Build) (64-bit) running on a Windows 7 Ultimate Service Pack 1. The Windows Operating System runs on a Dell latitude E6330 core i5 with 4-CPU's each of 2.50GHz speed and RAM of 8.00GB.

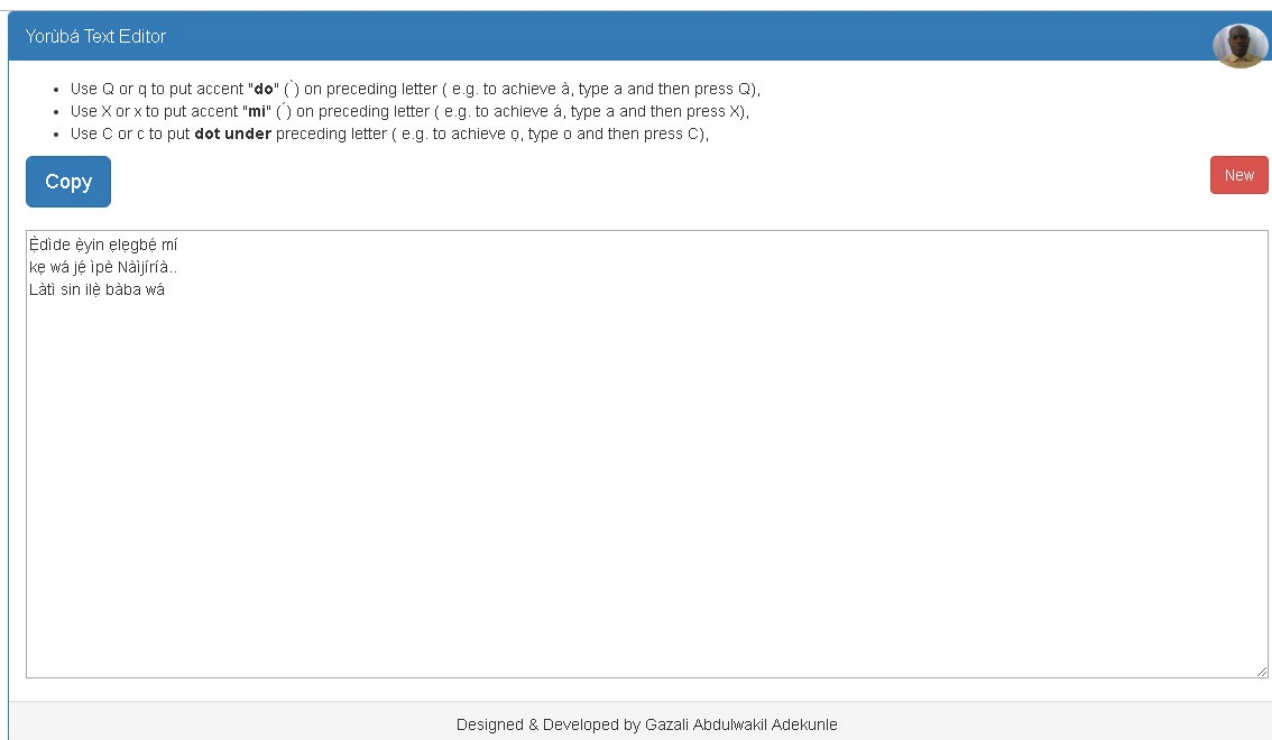


Figure 6: Yorùbá Text Editor.

Apache 2.2.21 was used to serve the pages to the web browser. Fifty-one (51) samples of data comprising the Loading, Scripting, Rendering, Painting, System, and Idle time were recorded under the Performance tab of the Developer tools. The obtained data is as shown in Table 4. On the average, it takes about 13.77ms to load the HTML DOM elements, **42.83ms** to load the JavaScript, **13.10ms** and **1.73ms** for rendering and painting the page by CSS. Additional time taken are **43.91ms** and **3,045.10ms** for the system and idle time respectively, totaling to **3,160.43ms (3.16s)** before the page can accept inputs from the users.

In order to measure the computational time to convert some selected characters to unicodes, Performance.now() function, a Windows High Resolution Timer, was added to the beginning and the end of the code segment where conversions take place in the JavaScript file. For example, when the sentence “Adéwálé ọ sí ọjà” was typed in the Yorùbá editor, the time taken by individual character is as shown in Table 5. The average time taken by some selected characters is 2.66ms while other characters have the average time of 0.66ms.

The editor pages, when save on a local disk, occupy hundreds of kilobytes. The Yorùbá editor occupies 480 KB, Igbo editor occupies 480 KB while the Hausa editor occupies 484 KB. Instructions on how to use each editor are placed at the top of each page. These can be seen on the screenshots of the editors as shown in Fig. 6, 7 and 8. The pages, if saved to local disk storage, can work without internet, making them affordable to all classes of people, because they were developed with client-side technologies (HTML, CSS,

and JavaScript). The editors work well on all latest version of browsers like (Google Chrome, Mozilla Firefox, Opera, Safari and Microsoft Edge). They were also tested on browsers of smart phones and desirable result was achieved.

A total time of 3,160.43ms (approximately 3s) is required when any of the editors is launched before the page can accept inputs from the users. This time taken before the system ready for use is small enough, despite being a web-based system, to engage the users. The computational time, 2.66ms, recorded while converting a selected ASCII character (e.g. c, q, or x, e.t.c.) to Unicode character has shown that the system will not in any way reduce the typing speed of the users.

The Editors are operating system independent because they are web-based system (web browser required). They can run offline (because of the client-side technologies like HTML, CSS, and JavaScript employed) if saved to local disk storage. This makes it to work without internet and thereby providing easy to use platform. The editors work well on all latest version of web browsers like Google Chrome, Mozilla Firefox, Opera, Safari and Microsoft Edge. The system has been able to address incompatibility of computer keyboards by converting some letters in English alphabet which are not in the alphabet of the languages considered (e.g. by converting letter c to *dot-under* in the Yorùbá editor) to produce letters in the alphabets of the languages under consideration (Hausa, Igbo, and Yorùbá).

5. CONCLUSION AND FUTURE WORK

This study has developed invaluable text editors for the speakers of the Nigerian languages

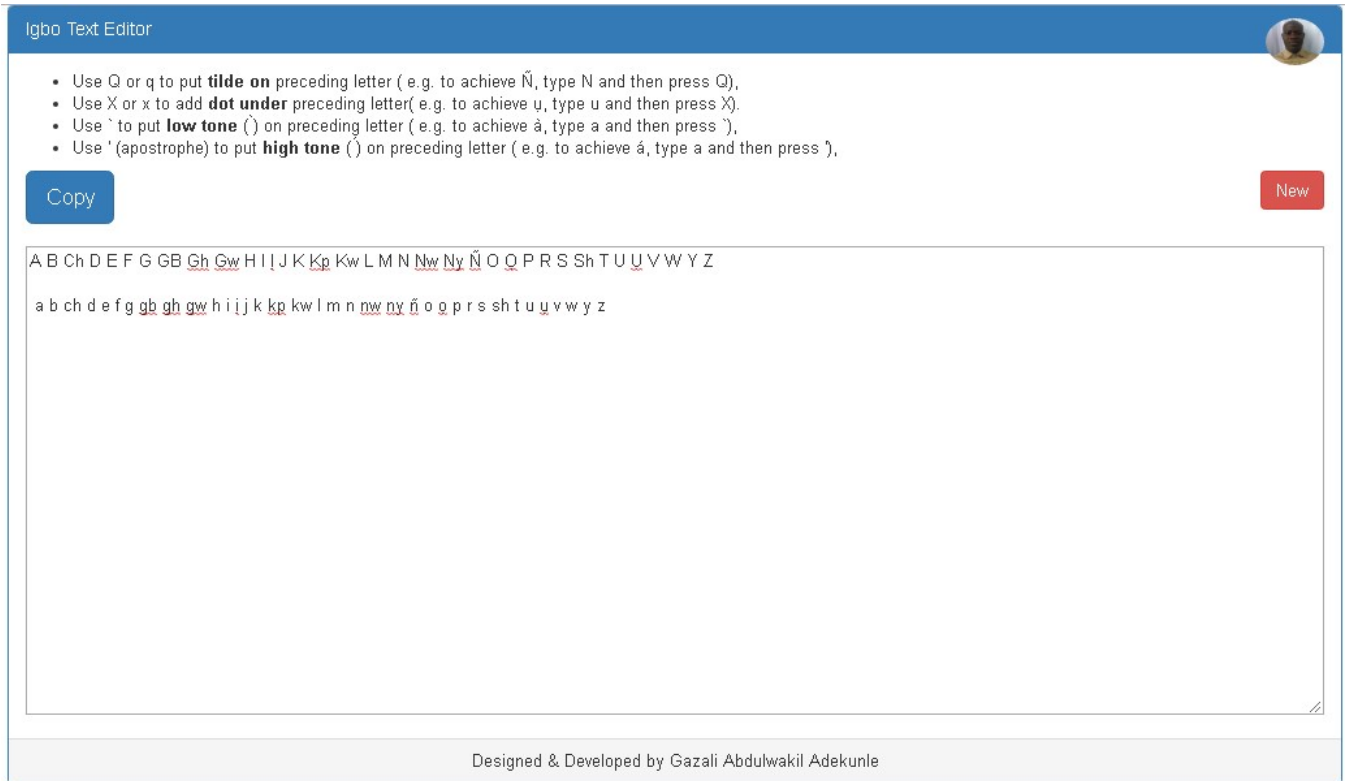


Figure 7: Igbo Text Editor.

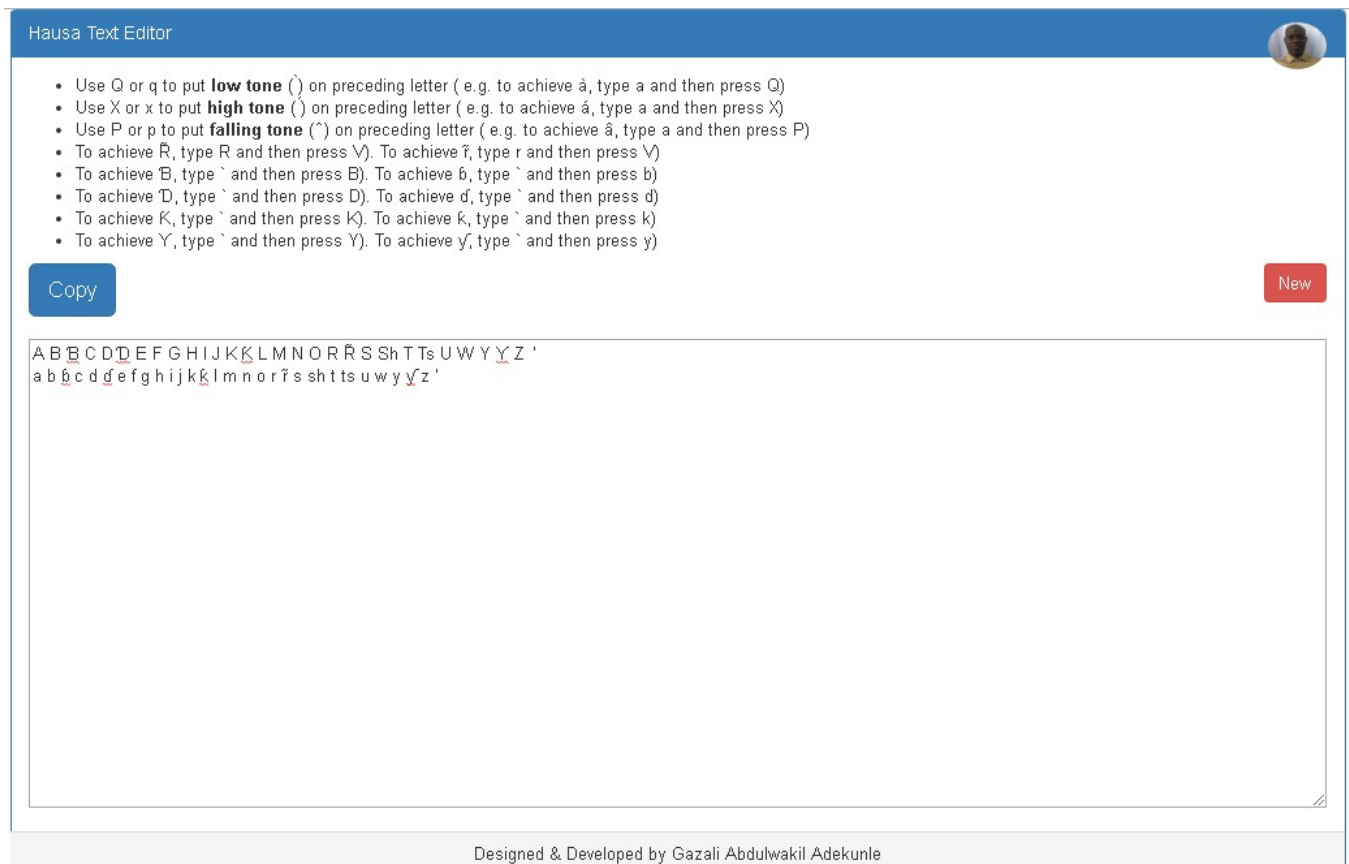


Figure 8: Hausa Text Editor.

(Yorùbá, Igbo and Hausa) and for those who are interested in the languages. The three client-side technologies employed for the construction of the web pages are Hyper-Text Markup Language (HTML), Cascading Style Sheet (CSS), and JavaScript. These web pages designated for Yorùbá, Igbo and Hausa language were generated with HTML. The CSS was used to define the look and feel of the HTML elements on each page and JavaScript was employed to put life on the pages.

The study has been able to provide solution to the incompatibility of computer keyboards (hard or soft) with major Nigeria languages (Hausa, Igbo, and Yorùbá). Thereby, providing user friendly interaction in typing text. The editors have been fully annotated, however, the web pages for the three languages and current annotations are publicly available for non-commercial use at <http://www.gazaliwakil.com.ng/yoruba>, <http://www.gazaliwakil.com.ng/igbo>, and <http://www.gazaliwakil.com.ng/hausa> respectively. These web addresses which contain the editor and annotations can be easily accessed using Google Chrome, Mozilla Firefox, Opera, Safari, and Microsoft Edge browser.

Future work includes but not limited to making these editors web-based word processors. Also, text-to-speech and speech-to-text features for these editors are in pipeline.

References

- [1] O. Benson, L. Anyalebechi, and I. Ariole, "Promoting Indigenous Language in Nigeria: Issues and Challenges for the Library and Information Professionals," *Library Philosophy and Practice*, 2017.
- [2] M. Ayeomoni, "Migration and culture: Implications on Nigerian languages," *International Journal of English and Literature*, vol. 2, no. 9, pp. 195–199, 2011.
- [3] H. Alhassan and A. Kadiri, "Reviving Indigenous Languages through Teaching and Learning: The Case of Igala Language," *International Journal of Education, Learning and Development*, vol. 3, no. 7, pp. 35–46, 2015.
- [4] E. O. Ugwu and P. O. Ogunremi, "Nigeria and the Sustainable Development Goals: Why the indigenous languages count," *European Scientific Journal*, vol. 15, no. 5, pp. 52–68, 2019.
- [5] I. Ajepe and A. Ademowo, "English Language Dominance and the Fate of Indigenous Languages in Nigeria," *International Journal of History and Cultural Studies*, vol. 2, no. 4, pp. 10–17, 2016.
- [6] M. Adekunle, "National language policy and planning: the Nigerian situation," *West African Journal of Modern Languages*, vol. 1, no. 1, pp. 23–29, 1976.
- [7] C. C. Mann, "Choosing an indigenous official language for nigeria," 1990.
- [8] S. Ager. (2020, April) The online encyclopedia of writing systems and languages. [Online]. Available: <https://www.omniglot.com/writing/>
- [9] P. Daniels and W. Bright, Eds., *The world's writing systems*. Oxford University Press on Demand, 1996.
- [10] S. Fischer, Ed., *History of writing*. Reaktion Books P12, 2003.
- [11] A. Haladu. (2020, April) Use local languages in education. [Online]. Available: <https://punchng.com/use-local-languages-in-education/>
- [12] I. Ayogu, "Exploring multinomial naïve Bayes for Yorùbá text document classification," *Nigerian Journal of Technology*, vol. 39, no. 2, pp. 528–535, 2020.
- [13] O. Abiola, A. Adetunmbi, A. Fasiku, and K. Olatunji, "A web-based English to Yoruba noun-phrases machine translation system," *International Journal of English and Literature*, vol. 5, no. 3, pp. 71–78, 2014.
- [14] Microsoft. (2020, March) Keyboard shortcuts for international characters. [Online]. Available: <https://support.office.com/en-us/article/keyboard-shortcuts-for-international-characters-108fa0c1-fb8e-4aae-9db1-d60407d13c35>
- [15] G. Akinade. (2019, June) Tákàdà: A text editor for the yorùbá language. [Online]. Available: <https://sourceforge.net/projects/takada/>
- [16] L. Salifou and H. Naroua, "Design of a spell corrector for Hausa language," *International Journal of Computational Linguistics*, vol. 5, no. 2, pp. 14–26, 2014.
- [17] Branah. (2019, June) Online keyboards - type your language online. [Online]. Available: <https://www.branah.com>
- [18] Online-keyboard. (2019, June) Us online keyboard. [Online]. Available: <http://online-keyboard.org>