SECTION

6

USE OF WEB PAGE EDITORS

COMPUTATIONAL THINKING (PROGRAMMING LOGIC)

Web Technologies and Databases

INTRODUCTION

This section will help you learn how to use tools for creating web pages. You will explore how to design and organise web pages with sections, text, images, and multimedia. You will also learn about important topics like web page editors, Hyper Text Markup Language (HTML), Cascading Style Sheets (CSS), page layout, and formatting. These skills will help you build strong programming knowledge and solve challenging computer problems.

Key Ideas

- **Database:** A place to store and organise information.
- **Key:** A unique identifier in a database that ensures no two records are the same.
- **Queries:** Statements used to get information from a database.
- **SQL:** A special language used to query a database.
- **Relations:** Connections between tables in a database.
- **Entity:** An object or thing in a database.
- **Attributes:** A characteristic of an entity.
- **Editors:** Software used to write code, create and manage websites.
- **HTML:** The basic code used to structure and design web pages.
- **CSS:** A language used to style web pages.

WEB DEVELOPMENT

To start with web development, familiarise yourself with this simple website in **Activity 6.1**.

Activity 6.1

(Review) Visualising a website

1. Individually, observe and make notes on Ghana government's website by visiting *https://www.ghana.gov.gh/* or clicking *here*.
2. Share with the class what you have observed.

Web development is the process of making websites and web applications that people can use on the internet. It includes writing code, designing how the website looks and arranging the content to create pages that work well and are easy to use. Websites can be simple with just plain text or very advanced (like *Ghana Government Website*, *Facebook*, *X*, *Instagram* or online shopping sites like *jumia.com.gh*, *tonaton.com*, etc).

Importance of Web Development

Web development has improved our daily lives due to the following reasons:

1. **Accessibility**: Websites provide a platform for sharing information and services with people across the globe. The Ghana Education Service (GES) *website* allows students, teachers, parents and the public to access official updates and resources.
2. **Business Presence**: Having a website for your business allows the business to connect with a larger audience and showcase your products or services. *Jumia Ghana* and *Melcom online* are examples of such businesses, reaching customers beyond its physical locations.
3. **Communication**: Blogs, forums and social media on websites and web apps make it easier for people to interact and share ideas. *GhanaWeb* provides news, forums, and blogs for Ghanaians to stay informed and engage in discussions.
4. **Education**: Schools and educational organisations use websites to share resources and information with students and teachers. *WAEC Ghana* provides a website that helps BECE and WASSSCE graduates like you to check your exam results and access educational materials.
5. **E-commerce:** Online shopping platforms help businesses sell their products directly to customers via the internet. *Jumia Ghana* enables customers to shop for a wide variety of products and have them delivered to their doorsteps.

Types of Websites (Static vs. Dynamic)

It is worth knowing that we have two types of websites. These are static websites and dynamic websites. Let us look at them in details.

Static Websites

Static websites have fixed content, meaning the information on each page stays the same for everyone who uses it.

Characteristics of Static Websites

There are some characteristics a website needs to possess to be classified as a static website. These features include the following:

1. Content does not change unless manually updated by the developer: The information on the website stays the same until someone, like a web developer, changes it.
2. Easier and faster to create and host. Static websites are simpler to build and can be published on the internet more quickly.
3. Generally cheaper to develop and maintain. Static websites do not require complex features hence are often less expensive to create and keep running.
4. Suitable for small websites with limited content updates, such as portfolios or informational websites. Static websites are ideal for smaller sites that do not need frequent changes like a personal portfolio or a business information page.

An example is a small shop's website that lists products and contact details of the business without needing frequent updates.

Dynamic Websites

Unlike static websites which have fixed contents, dynamic websites are more advanced and interactive. It interacts with users and suggests or recommends products due to past experiences with the sites. Server-side technologies, such as PHP, ASP.NET, or Node.js, along with databases, are used to create content that changes depending on user interactions or other factors. An example of such a website is the JiJi Ghana website we looked at in our introduction lesson on web development, where the contents, such as product listings, prices, and recommendations change based on what a customer searches for or clicks on. This type of website is used when regular updates and personal interaction are needed to keep the content fresh and relevant for users.

Characteristics of Dynamic Websites

Also, there are some characteristics a website needs to possess to be classified as a dynamic website. These features include the following:

1. Content can change based on user inputs, time, or other variables: Dynamic websites adjust the information they show, based on what users do on the site, the time of day, or other conditions.
2. More interactive and engaging for users: Dynamic websites allow users to interact more. For instance, when you scroll through posts on a social media site or add an item to your shopping cart, the website responds to what you are doing, making it feel like you are in control.

3. Can handle large amounts of data and frequent content updates: Dynamic websites can handle a lot of information and update it easily. For instance, an e-commerce website might have thousands of products, and when an item goes out of stock or a new product is added, the website automatically updates to reflect these changes without needing someone to do it manually.

You can now distinguish between static website and dynamic website based on your understanding of the content so far.

Activity 6.2

Comparison between a Static Website and a Dynamic Website

1. Individually, based on your understanding of static and dynamic websites, compare them using the following features:
   a. content update
   b. interactivity
   c. complexity
   d. cost
   e. development time
2. Present your comparison in a table.
3. Share your response with peers for discussion in class.

Activity 6.3

Identifying websites by type

1. Individually, observe Ghana government's website by clicking *https://www.ghana.gov.gh/* or *here*.
2. In pairs, compare it with *Jumia Ghana's website* using the link *jumia.com.gh*.
3. Identify at least three key differences in functionality and interactivity.
4. Discuss whether each site is static or dynamic, with your reasons for classification.
5. Share your discussion with the class.

Role of Databases in Web Development

Databases play vital roles in web development. Typically, databases are used to store and manage data for web applications. They enable web developers to save, access and modify information whenever required. For example, on an e-commerce platform, its

database stores details such as product names, prices, and customer orders. When you browse or search for items, the website fetches this information from the database and displays it to you. If you make a purchase, the database updates to reflect the new order and adjust stock levels accordingly.

Types of Databases

Databases are grouped into two types, namely Relational Databases and NoSQL (non-relational) Databases.

1. **Relational Databases**

   Relational databases store data in tables like a spreadsheet. Each table has rows and columns. Each row is a piece of data (e.g., a customer or a product), and each column represents a property of that data (e.g., the name, price, or address). These tables can be linked together, based on shared information using keys such as a primary key or a foreign key.

   Examples of relational databases include: **MySQL** or **Microsoft SQL Server** are relational databases. For instance, an online store's database will have one table for products and another for customers. The tables are linked, so that the store can check which products each customer has purchased.

2. **NoSQL Databases**

   NoSQL databases are used for storing large amounts of data that cannot fit well into tables. These databases are more flexible and can store data in many formats, such as documents or graphs. They are good for handling big data or data that changes often.

   An example of a NoSQL database is **MongoDB**. It is often used for websites or apps that have lots of data that does not fit neatly into tables, like social media platforms or online games were data changes frequently.

   NoSQL databases are often referred to a non-relational, as the data held inside is unstructured and does not have any direct relationship with the other data held in the database.

Introduction to Web Page Editors

Web page editors are software tools that help people create, edit, and manage the content and design of websites. They are easy to use, making the process of building websites simpler for beginners, while also helping experienced developers work faster.

Purposes of Web Page Editors

There are reasons one will use a web page editor. Some of these reasons are:

1. **Makes web development easier:** Web page editors have tools that make it simple to create and edit web pages, even if you are new to web design.

2. **Streamline Workflow:** They bring together tasks like writing code, designing the layout, and fixing errors into one platform, making work faster and more organised.
3. **Enhance Productivity:** Features like colour-coded text (syntax highlighting), suggestions for writing code (code completion), and drag-and-drop tools help you work faster and complete your tasks quicker.
4. **Ensure Compatibility:** These editors allow you to easily test websites, to ensure they work properly on different browsers and devices, like mobile phones and computers.

Types of Web Page Editors

Web page editors can be broadly categorised into **WYSIWYG (What You See Is What You Get) editors** and **code editors**.

1. **WYSIWYG Editors:** Allows developers to design web pages by simply dragging and dropping elements like text, images, and buttons onto a screen, without needing to write any code. The editor automatically generates the HTML, CSS, and JavaScript code in the background (this is typically referred to as low or no code development).

   Features

   a. **Drag-and-Drop Interface:** Users can easily add and move items like text, images, or buttons on the web page by dragging and dropping them.

   b. **Real-Time Preview:** As you make changes, you can instantly see how the final web page will look, making it easier to adjust designs.

   c. **Templates and Themes:** These editors come with pre-designed templates and themes that you can customise to suit your needs, thereby saving time and effort.

   d. **Integrated Tools:** These editors often include extra tools for managing pictures, videos, and other multimedia content used on the website.

   Examples of this WYSIWYG

   a. **Adobe Dreamweaver is a** powerful editor with tools for beginners and advanced users.

   b. **WordPress is a** content management system (CMS) platform with a WYSIWYG editor for creating and managing website content.

   c. **Wix and Squarespace** are online platforms that let you design and host websites using simple drag-and-drop features.

2. **Code Editors** are tools designed for developers who want to write their own code manually. They provide a text-based platform where developers can create websites using programming languages like HTML, CSS, and JavaScript. These editors include advanced features that make it easier to write, edit, and manage code effectively. A developer might use software such as **Visual Studio Code** or

Sublime Text to manually write the code for a website, giving them full control over its structure and design.

Features of code editors

a. **Syntax Highlighting:** Code editors use different colours to highlight various parts of the code. This makes it easier to read and identify mistakes.

b. **Code Completion:** While typing, the editor suggests or completes the code for you, saving time and helping you to avoid errors.

c. **Debugging Tools:** These tools help find and fix mistakes in your code, ensuring the website works correctly.

d. **Version Control Integration:** Code editors can connect to systems like **Git**, which track changes made to your code. This allows you to manage your changes providing an audit trail and allowing you to go back to earlier versions if something goes wrong.

Examples of this Code Editors

a. **Visual Studio Code**: A popular tool by Microsoft that supports many programming languages and has useful extensions.

b. **Sublime Text**: A fast, lightweight editor known for being simple yet powerful.

c. **Atom**: An open-source editor made by GitHub, loved for its flexibility and add-ons.

Activity 6.4

WYSIWYG and Code Editors

1. In your groups of no more than five, write at least three differences between WYSIWYG and Code Editors.
2. Present your findings to your peers

Activity 6.5

Exploring WordPress

1. Visit youtube.com and search for “how to use WordPress” to build a portfolio website for free
2. Follow the instruction in the video and create a portfolio website on your own
3. Share your work with your class for discussion and feedback.

BASIC HTML AND CSS

What is HTML?

HTML stands for HyperText Markup Language. HTML is a language, which makes it possible to present information on the Internet. That is, web pages all over the world consist of HTML. However, HTML is not a programming language but a markup language. The idea behind HTML was born at the European Laboratory for High Energy Physics (CERN) in Geneva, Switzerland, as early as 1989. A year later, the World Wide Web project was also started there. The purpose was to make it easier for scientists at different universities and research laboratories to gain access to research documents of their colleagues.

The logo of HTML is shown in **Figure 6.1**.

Figure 6.1: Logo of HTML

An HTML document is simply a text file (it may also contain images, video or audio) that contains the information you want to publish and the appropriate markup instructions indicating how the browser should structure or present the document.

HTML consists of **elements** and **tags**. We will practice how to use these elements and tags in the subsequent activities.

Markup elements are made up of a start tag, such as ** **, and typically, though not always, an end tag, which is indicated by a forward slash within the tag, such as ** **. The tag pair should fully enclose any content to be affected by the element, including text and other HTML markups.

An HTML **element** consists of both the content and its markup. Elements are identified by tags in the text source. A **tag** consists of the element name (usually an abbreviation of a longer descriptive name) within angle brackets (**< >**). The element name appears in the opening or start tag and again in the closing or end tag, preceded by a forward slash (/). The tags added around the contents are referred to as the **markup**. It is important to note that an element consists of both the content and its markup (the start and end tags). Elements that have content are often referred to as **Containers**.

E.g ** ** text ** **. Not all elements have content. These elements are referred to as **Empty elements**. E.g **<input type="text" name="textbox" />**.

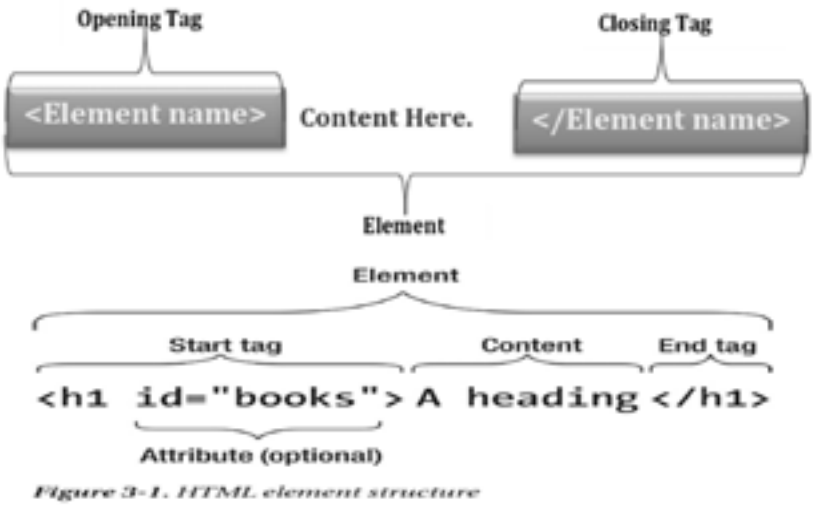


Figure 6.2: HTML Element STRUCTURE

The Basic Structure of an HTML Document

The way a document is marked up with elements and their attributes is according to a Document Type Definition (DTD). DTDs are a set of rules that govern the way in which a document can be marked up. The authoritative source for information about HTML and the HTML DTD is the World Wide Web Consortium(W3C) at *http://www.w3c.org*. An HTML document follows a specific structure with several important elements that help the browser understand how to display the content.

The basic HTML5 document structure is as follows:

<!DOCTYPE html>

<head>

<title> *Title of page goes here***</title>**

</head>

<body>

Main content of the web page goes here.

This is what is displayed in the browser window

</body>

</html>

Let us discuss the structure in more detail.

1. **Document Type Declaration:**

   The Document Type, or doctype for short is indicated by the **<!DOCTYPE>** tag in HTML documents. This specifies the rules for the document language, so the browser knows how to interpret the HTML code and display it properly.

2. **HTML Element:**

   The **<html> </html>** tags surround the entire HTML document, and it is referred to as the **root element**. The root element is the container element for all other elements in the document. The opening tag; **<html>** tells the browser that, whatever content that comes after it should be treated as an HTML content until the browser meets the closing tag **</html>.**

3. **Head Element:**

   The **<head> </head>** tags specify the header segment of any HTML document. The **<title> </title>** tags are used within the **<head> </head>** tags to specify the title of a given web page.

   Other tags such as **<meta>, <link>, <style> and <script>** are also used within the **<head>** tag of an HTML document.

4. **Body Element:**

   The **<body>…</body>** tag of a web page play an important role with regards to the page's actual content. The **body** element contains information about the page's background colour, the background image, as well as the text and link colours, and all other items that are displayed on a web page. If the **body** element is left blank, web browsers will revert to their default settings.

   The detailed HTM5 document structure looks like this:

```
<!DOCTYPE html>
<head>
<title> The title of the page goes here</title>
</head>
<body>
```

   Other HTML5 elements and tags are used here within the body tag to produce the actual content of the web page.

```
</body>
</html>
```

Activity 6.6

Creating a Web Page using HTML

1. Create a folder on the desktop named "Programming Class"
2. Create another folder in the Programming Class named "HTML_Lessons"
3. Create a text file using notepad in HTML_Lessons
4. You can name the file as "Lesson1.html"

5. Ensure you have "html" at the end of the file naming, or else it will not be recognised as an html file.
6. Launch the "Lesson1" file with any chosen text editor of your choice (e.g. notepad, VS Code or Sublime Text 3).
7. In the editing area, type the required elements of a basic HTML document as in the example below.

   <!DOCTYPE html>
   <head>
   <title> My First Web Page</title>
   </head>
   <body>
   </body>
   </html>
8. Type any text of your choice within the **<body> </body>** tags of the document structure.
9. Save the file (Ctrl+S).
10. Double click on the file in "HTML_Lesson" folder to preview your page in the browser.
11. Try to identify where the title "My First Web Page" and the content you typed in the body segment appears.
12. Share your web page with the class for discussion and feedback.

All you need to learn HTML is a simple text editor and a web browser and you are good to go. The most important thing here is to learn and understand the various tags that are used within the body segment of the entire document structure.

Common HTML Tags

The commonly used tags for formatting within HTML documents are:

1. **Headings: <h1>** Heading 1**</h1>**

   Headings are used to define titles and subtitles on a page (just like the headings in Word documents). There are six levels of headings which go from from **<h1>** through to **<h6>**, with **<h6>** being the smallest.

2. **Paragraphs: **Paragraph** **

   Paragraphs are used to define blocks of text. They are enclosed in ** **Paragraph** **, tags.

Activity 6.7

Headings and Paragraphs in HTML

1. Create a new file named “Lesson2”
2. Copy the elements below and replace the name within the tags with your name and observe the difference

   <h1> Raphael Senyo Dordoe <h1/>

   <h2> Raphael Senyo Dordoe <h2/>

   <h3> Raphael Senyo Dordoe <h3/>

   <h4> Raphael Senyo Dordoe <h4/>

   <h5> Raphael Senyo Dordoe <h5/>

   <h6> Raphael Senyo Dordoe <h6/>
3. Use the paragraph tag to write a simple message about yourself like

   <p>Hi, I’m Sir Raphael, and I’m passionate about creating safe and inclusive learning spaces for my students by using a variety of classroom management techniques.

   One way I do this is by providing more opportunities for learners from underprivileged groups.

   For example, my school doesn’t have computers for practical work, so I improvised with my personal laptop. I encouraged a girl who had never used a desktop or laptop before by making her feel safe and confident to practise her code skills in HTML before the whole class. **<p/>**
4. Share your work with the class for discussion and feedback.

Before we continue writing more code, watch this short video (*https://youtu.be/8xMnkt474q4*) about indentation in programming, to refresh your mind.

Activity 6.8

HTML Structure

1. Duplicate the “Lesson1” HTML file and name it “Lesson3”
2. Copy the code from “Lesson2” and paste it into the body segment of “Lesson3”.
3. Save the file and launch “Lesson3” in a web browser
4. Write your observations between “Lesson2” and “Lesson3”, what are do you notice that is different?
5. Show your work to your teacher

It is important to note that the structure of “Lesson3” is the best way to write your HTML code to keep your work organised.

Activity 6.9

HTML tags

1. Search for "HTML tags" on the internet
2. Explore how they are used
3. Take key notes of the following, we will use them later to create a personal portfolio.
   a.
   b. <hr>
   c. <img>
   d. <video>
   e. ...
   f. <header>...<header/>
   g. <footer>...<footer/>
4. You can practise your findings by creating in a new file named "Lesson4" and using some of the HTML tags that you have researched

Links

HTML links are hyperlinks, meaning that when you click on a link, it takes you to another document, page or web resource and when you move the mouse over a link, the mouse arrow will turn into a little hand. Links are defined with the **<a>** tag and require an **href** attribute to specify the target URL.

The syntax for inserting links is *link text* .

For example: Visit T-TEL.

In this link, the target URL is *https://t-tel.org/* and the text visible to users of the site to click on is "Visit T-TEL". Let's put this into practise for a better understanding.

Activity 6.10

HTML links

1. Duplicate "Lesson2" and rename it as "Lesson4"
2. Add a short paragraph about education in Ghana using the <p> tags.
3. Write links to the websites of the following institutions using <a> tags.
   a. *https://moe.gov.gh/* (The user should see "visit MoE").
   b. *https://ges.gov.gh/* (The user should see "visit GES").
   c. *https://t-tel.org/* (The user should see "visit T-TEL").

4. Open one of the links in a new browser window.
5. Share your result with your peers.

Images

Images or pictures can be inserted into HTML documents. It is done using the ***<img >*** tag.

a. The ***src*** (Source) attribute is the only required attribute of the ***<img >*** tag. This attribute tells the browser where to find the image that can be inserted into the web page.

b. The ***alt*** attribute provides a text-based description of the image (alt is important because it helps visually impaired users understand what the image is about, making the webpage more accessible).

c. Other attributes of the ***<img >*** tag include ***height, width*** *and* ***align*** *which will be explored later.*

The <img> tag in HTML is used to embed images into the web pages content. It is a self-closing tag, so there is not need to provide a closing tag at the end. Below is an example of the <img> tag in use:

<img src=”URL” alt=”description” width=”width” height=”height”>

The above code contains the following attributes:

- src: this specifies the location of the image. It can be a relative path (e.g. ‘image/photi.jpg) or it can be an absolute URL (e.g. ‘example.com/photo.jpg)
- alt: this provides alternative text for the image if it cannot be display on the web page. This is important for both accessibility and SEO (search engine optimisation)
- width: this defines the width dimension for the image, it can be defined as either pixels or a percentage value
- height: this defines the height dimension for the image, it can be defined as either pixels or a percentage value

Activity 6.11

HTML images

1. Duplicate “Lesson4” and rename it “Lesson5”
2. Download four pictures of your schools into a new folder named “images” in the “HTML_Lessons” folder
3. Write the header as <h1> Pictures of my school </h1>
4. Write the image code as <img src=” “ alt=” “>
5. Right click on the first image and click on copy as path

6. Paste the path into the quotes of src.
7. Write a short description of the image in the alt attribute.
8. Name the image using the ... tags
9. Repeat steps 4-9 for the other three pictures
10. Share your result with your peers.

Lists

HTML supports **ordered**, **unordered** and **definition** lists. It is also possible to nest one type of list within another in HTML documents.

1. Ordered List

An ordered list is a list in which the list items are numbered sequentially. Ordered lists are enclosed in the ** ...
** tags. Each list item is placed within the ** ...
** tags using the ** ...
** tags.

For example

<h3>My favourite dancehall icons in Ghana:</h3>

 Stonebwoy

 Shatta Wale

 Samini

 Rocky Dawuni

 Shasha Marley

2. Unordered List

An unordered list is used when the list of items does not have any necessary sequence of appearance. It is also called a **bulleted list**. Unordered lists are enclosed in the **
... ** tags. Each list item is placed within the **
...
** tags using the **
... ** tags.

For example

<h1> The following are my favourite foods:</h1>

<ul>

 Fufu

 Akple

 Kenkey

Red Red

TZ

**
**

3. Definition Lists

This is used to present a term and its definition formatted like a glossary or a dictionary. It is the ideal format to present lists of words or phrases with their meanings. The tags used for a definition list are <dl> … </dl> with both the term defined as the <dt> tag and the description as defined as the <dd> tag, note that both the term and the definition are needed for each list item in the <dl> … </dl>

<dl>

<dt> Definition Term </dt>

<dd> Definition Description </dd>

</dl>

Activity 6.12

Creating a List of Favourite Meals

In groups of no more than five,

1. Create a new file named "Lesson6"
2. Create an ordered list of five local foods that you enjoy the most.
3. Create an unordered list of six local football clubs that you know in Ghana.
4. Create a definition list of three words
5. Share your list with your peers.
6.

Activity 6.13

Glossaries

In groups of no more than five,

1. Create a glossary of five terms you have learned in HTML in a new file named "Lesson7".

2. Share your glossary with your peers.

Table 6.1: Summary of List Elements

Tag	Description
	Defines an unordered list
	Defines an ordered list
	Defines a list item
<dl> </dl>	Defines a description list
<dt> </dt>	Defines the term in a description list
<dd> </dd>	Defines the description in a definition list

Tables

Tables provide an excellent way to organise and display information on web pages. Tables are defined using the **<table>…</table>** tags. The HTML **<table> element** contains the table information, which consists of **table header elements (<th>)**, **table row elements (<tr>)**, and individual **table data cells (<td>)**. The letters **td** stand for "table data" which is the content of a data cell. A data cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables, and so on.

These are the three elements that are used most frequently when you are building tables.

Table 6.2: HTML Table Elements

Element	Description
<table>…</table>	Establishes the table. It contains all other elements that specify captions, rows and the content.
<tr>…</tr>	Table row. It contains rows of table cells.
<td>…</td>	Table data cell. This contains the actual table data held in each cell.
<th>……</th>	Table header cell: It contains the header information for a column of data.
<caption>...</caption>	Provides a short description of the table's contents.
<thead>…<thead>	Signifies table header.
<tbody>…<tbody>	Signifies table body

Element	Description
<tfoot>...<tfoot>	Signifies table footer
<col />	Specifies column properties
<colgroup>...</colgroup>	Specifies multiple column properties

Activity 6.14

Tables in HTML

1. Copy and run the following code in a new file named "Lesson8".

```
<table>
<tr>
<th>Title of Music </th>
<th> Artiste </th>
</tr>
<tr>
<td> Forward to Africa </td>
<td> Joseph Hill </td>
</tr>
<tr>
<td> China Roses </td>
<td> Enya </td>
</tr>
<tr>
<td> Floral Street </td>
<td> Enya </td>
</tr>
<tr>
<td> Humble Africa </td>
<td> Joseph Hill </td>
</tr>
<tr>
<td> Nah Stay Inna Babylon</td>
<td> Joseph Hill </td>
</tr>
<tr>
<td> Miss Independent </td>
```

```
<td> Neyo </td>
</tr>
<tr>
<td> Storm in Africa</td>
<td> Enya </td>
</tr>
<tr>
<td> Count on Me</td>
<td> Whitney Houston</td>
</tr>
</table>
```

2. Edit the opening tag **<table>** to **<table width= "50%", border = "2">**
3. Observe what changes this makes and write down your observations

Activity 6.15

Creating a Personal Timetable

In groups of no more than five,

1. Create a personal study timetable with the days of the week as the header row in a new file named "Lesson9".
2. Add three rows for subjects.
3. Specify the subject and the time for each of the three subject rows.
4. Share your table with your peers.

Introduction to CSS

What is CSS?

CSS is a styling language that defines how the website should look like. CSS is an acronym which means Cascading Style Sheets. *CSS* is the recommended way to control the presentation layer in a web document. The best way this is done is by creating effective and efficient style rules. A style sheet is a grouping of formatting instructions that can control the appearance of many HTML pages at once.

CSS is a language that defines the style constructs such as fonts, colours and positions, which are used to describe how information on a web page is formatted and displayed.

Benefits of CSS

CSS offers several significant benefits. These benefits include:

1. Apply the same styles across multiple pages.
2. Separate styling rules, which can be used for different output media.
3. Easily change the appearance of your website without modifying HTML content.
4. Keep the structure (HTML) and styling (CSS) separate for better organisation.
5. All styling is kept in a limited number of style sheets.
6. There is some saving on bandwidth, therefore making the webpage(s) load faster.
7. The separation of content from presentation makes it easier for site owners to reuse the content for other purposes, such as RSS feeds or text-to-speech conversions.

CSS Syntax and Selectors

A rule or "rule set" is a statement that tells browsers how to render elements on an HTML page and it consists of a **selector** followed by a **declaration** block. Each declaration consists of a property and a value. A CSS declaration always ends with a semicolon, and declaration groups are surrounded by curly brackets.

The syntax rule looks like the image below.

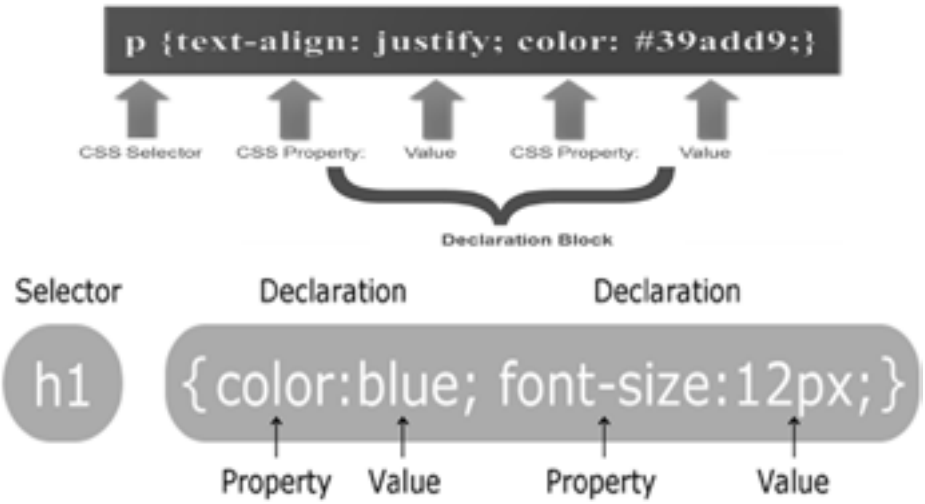


Figure 6.3: The syntax rule

The selector is used to 'select' or 'target' the HTML element that you want the style to apply to. This is how we tell the web browser to apply a particular style to a particular html element. For instance, when you want to style all the paragraphs on your web pages.

- The letter **"p"** would be our selector and it targets (selects) all paragraphs.
- The **declaration block** is everything between the curly brackets. What you see between these curly brackets is referred to as the ***Property*** *and the* ***Value***. This is how we define the properties we want to apply to our web pages.

Combining HTML and CSS

To work in CSS, one must be familiar with some markup languages like HTML or XHTML, so that you can add personalised styles on the markup documents. This is because CSS rules are applied to HTML or XHTML elements. There are three ways to apply CSS to HTML elements.

1. **Using inline CSS**

   An inline style can be used if a unique style is to be applied to one single occurrence of an element. To use inline styles, use the style attribute in the start tag of the relevant element. The style attribute can contain any CSS property. The example below shows how to change the text colour of a paragraph:

   This is a paragraph.

   NOTE CSS use American English spellings, for example 'colour' is spelt 'color'

2. **Using internal CSS**

   An internal style sheet can be used if one single document has a unique style. Internal styles are defined in the <head> section of an HTML page, by using the **<style>** tag, like this:

   <head>

   <style type="text/css">

   body {background-color:yellow}

   p {color:blue}

   </style>

   </head>

3. **Using external CSS**

   An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire web site by using one CSS file. Each page must link to the style sheet using the **<link>** tag. The **<link>** tag goes inside the **<head>** section as shown in the example below:

   <head>

   <link rel="stylesheet" type="text/css" href="mystyle.css" />

   </head>

Activity 6.16

Creating My Portfolio

Using basic HTML and CSS, complete the following

1. Create a new folder named "**CSS Lessons**".
2. Create a new HTML file named "Portfolio.html" in the folder

3. Create a static web page titled “About Me” in your HTML file
4. On your web page, include the following
   a. Your name,
   b. A description about you,
   c. A picture of you,
   d. Your favourite subject, food, teacher, hobbies,
   e. The name of your best friends,
   f. The name and description of your school,
   g. A picture of your school,
   h. A google map link to your school.
5. Show your work to your teacher to ensure it has the required details in step 4.

Activity 6.17

Styling My Portfolio

1. Follow the instruction in this video
   https://drive.google.com/file/d/1mcVQzsUA3Vr2aedmT5zNSTK4yWd2-H-g/view?usp=drive_link
2. Style your portfolio website.
3. Share your websites with your peers.

Note: The source code is attached in this link, you can download, edit and improve it.

a. Html
   https://drive.google.com/file/d/15vRumOyrnizJsfd1cAXlLBDIZV8qMC/view?usp=drive_link
b. CSS:
   https://drive.google.com/file/d/1jDsdnKHoQPkUJggz03JfUxPFERwbCRSv/view?usp=drive_link

IMAGE AND VIDEO INTEGRATION

Videos

The <video> tag lets you include video files directly from your server. It supports various video formats like MP4, WebM, and Ogg.

<video controls width=”width” height=”height”>

<source src=”URL” type=”video/format”>

Your browser does not support the video tag.

</video>

The above code contains the following attributes:

a. **controls:** this adds playback controls (play, pause, volume) to the video player
b. **width:** this defines the width dimension for the video player, it can be defined as either pixels or a percentage value
c. **height:** this defines the height dimension for the video player, it can be defined as either
d. **source:** this specifies the video file's path and format

Integration of Video and Audio files to a website

1. Video files are a collection of images usually with a related sound file and can be inserted into the HTML. Some common video formats are: MPEG (.mpeg/.mpg), AVI for Audio/Video Interleave (.avi), and QuickTime (.qt/.mov), etc.
2. Integrating audio into your web pages can enhance the user experience by providing sound effects, background music, or narrated content.

Activity 6.18

Adding Video and Audio Files

1. Search online for how to add video and audio files respectively to an HTML page
2. Follow to instruction or demonstration to add a video file to your portfolio page
3. Share your results with the class for discussion and feedback.

Note: there are other interactive multimedia files like, slideshows and maps, which can be integrated in a website using specific plugins.

MULTIMEDIA INTERGRATION

Best Practices for Multimedia

There are things that need to be done and considered when developing a website. It is prudent to follow such standards to write syntactically correct code and get websites

that offer the best in terms of the content loading efficiently, looking good, and being accessible to all users. These best practices include:

Optimising Media for the Web

Optimising media means making sure that the images and videos on your website load quickly and look good, no matter what device or internet speed people may be using. This is important because slow websites can frustrate users and make them leave. Good optimisation improves the user experience and makes websites accessible to everyone. The following need to be optimised for a website.

1. Image Optimisation

Images often take up the most space on a website, so optimising them is essential to make your website load faster. This involves choosing the right format and reducing the file size, while keeping the quality high. Examples of file types for images include:

a. **Joint Photographic Expert Group (.JPEG)**: Best for photos and images with lots of colours and detail. Compresses the image to make the file smaller, which helps the website load faster.

b. **Portable Network Graphic (.PNG):** Ideal for images with transparent backgrounds or high-quality graphics, like logos or illustrations. Does not lose quality when compressed but creates larger file sizes compared to JPEG.

Compressing Images

Image compression reduces file size without making the image look blurry. Smaller files load faster on websites. Tools to compress images include:

i. *TinyPNG*: Reduces the size of PNG and JPEG images.

ii. *JPEG-Optimizer*: Helps you compress JPEG images while keeping them clear.

2. Video Optimisation

Videos are another type of media that can slow down a website. Optimising videos means choosing the right format and reducing the file size while, keeping good quality.

a. **Choosing the Right Format:**

i. MP4: This is the most popular video format because it balances file size and quality. MP4 works on almost all devices and browsers, so it is a safe choice for websites.

ii. WebM and Ogg: These formats offer better compression than MP4, making the file sizes smaller. However, they are not supported by all browsers, so you might need to provide an alternative format.

3. Compressing Videos:

For video compression, use tools like *HandBrake* **and** *Clipchamp* to reduce video sizes, while keeping high quality for web use.

INTRODUCTION TO RESPONSIVE DESIGN

Responsive design means creating websites that can change their look and layout to fit different devices such as computers, tablets, or smartphones. This makes sure that anyone can use the website easily, no matter the size of their screen. Websites that adapt to the device so that it fits the size of the screen are called responsive websites.

Importance of Responsive Design

Below are some of the benefits that users get from responsive website design:

1. **Improved User Experience:** A responsive website makes it easier for users to find information or complete tasks, whether on a big screen or a small one.
2. **Increased Mobile Traffic:** Many people use mobile phones to browse the internet. Responsive websites make sure mobile users can access everything easily, without distortion or excessive scrolling.
3. **Cost-Effective:** Instead of creating separate websites for phones and computers, a responsive design works across all devices
4. **Search Engine Optimisation (SEO) Benefits:** Search engines like Google prefer responsive websites, so they appear higher in search results. Google recommends responsive web design as the best practice for mobile configuration.
5. **Futureproofing:** A responsive website will still work as new devices with different screen sizes are developed. This makes sure that the website will fit any future device that may be developed. This makes responsive design beneficial to users.

Key Principles of Responsive Design

1. **Fluid Grid Layouts:** A website's layout should adjust based on the screen size. For example, instead of fixing a section to 300 pixels wide, make it take up 50% of the screen. If a webpage is divided into two halves, each half will resize depending on the screen size.

Figure 6.4: Responsive Grid

From **Figure 6.4**. You will notice that the display layout on a desktop, laptop screen is different to that of the mobile screen. This is achieved by setting the display width in percentages and not keeping the widths fixed with the use of divs.

Activity 6.19

Exploring Responsive Grid

1. Copy and paste the code in a code editor and RUN it.
2. Reduce the page width by dragging the left or right side to make it smaller
3. Observe how the items rearrange because of the layout created.

```
<!DOCTYPE html>
<html lang=”en”>
<head>
<title>Fluid Layout Grid Example</title>
<style>
    .container {
      display: grid;
       grid-template-columns: repeat(auto-fit, minmax(100px, 1fr));
        gap: 10px;
        padding: 10px;
    }
    .item {
        padding: 20px;
        text-align: center;
        border: 1px solid #ccc;
```

```
    }
</style>
</head>
<body>
  <div class=”container”>
    <div class=”item” style=”background-color: yellow”>Item 1</div>
    <div class=”item” style=”background-color: green”>Item 2</div>
    <div class=”item” style=”background-color: orange”>Item 3</div>
    <div class=”item” style=”background-color: blue”>Item 4</div>
    <div class=”item” style=”background-color: pink”>Item 5</div>
    <div class=”item” style=”background-color: gray”>Item 6</div>
    <div class=”item” style=”background-color: purple”>Item 7</div>
    <div class=”item” style=”background-color: red”>Item 8</div>
    <div class=”item” style=”background-color: lightSeaGreen”>Item 9</
div>
    <div class=”item” style=”background-color: tomato”>Item 10</div>
   </div>
</body>
</html>
```

2. **Flexible Images:** Pictures on a website should shrink or expand to fit the screen without breaking the design. Use CSS properties like “max-width: 100%;” to ensure the images resize within their containing elements.

   For example, the CSS code img {max-width: 100%; height: auto;} sets the maximum width of the image to 100% and sets the height to automatic for every device.

Activity 6.20

Exploring Image flexibility

1. Download the index file (*https://drive.google.com/file/d/1kXzcLruOhN8QFH0QbMLvXUrzw3lbV4yQ/view?usp=sharing*)
2. Follow the instructions in the **TODO**
3. **Media Queries:** Media queries are special styles (rules) used to change the look of a website for different screen sizes, like the screen width, height, orientation and resolution. This allows you to create specific styles for different devices. Media Queries will be explored later.

4. **Responsive Typography:** Text should be easy to read on all devices. Use relative sizes like em or rem instead of fixed sizes like px. For instance, the CSS style: body {font-size: 2em; line-height: 1.8;}.

   Note that Responsive Typography will be explored later.

5. **Touch-Friendly Design:** Buttons and links should be large enough for easy tapping on touchscreens. Avoid hover effects, as they do not work well on mobile devices.

Extension Activity 1: Comparing Responsive and Non-Responsive Websites

Instructions

1. **Visit Two Sample Websites**

   Responsive Site: Visit a website known for its responsive design, such as [*https://www.example.com](https://www.example.com*). Resize the browser window to see how the layout adjusts to different screen sizes.

   Non-Responsive Site: Visit a website that is not designed to be responsive. Use [*https://www.nonresponsive.com](https://www.nonresponsive.com*) as an example. Resize the browser window and observe how the site does not adapt to different screen sizes.

2. **Analyse the Differences**

   Layout Changes: How does the layout of the responsive site adjust as you resize the browser window? How does the layout of the non-responsive site change?

   Image Scaling: Observe how images are handled in both sites. Are images scaling properly on the responsive site?

   Text Readability: Compare text readability on both sites as you adjust the screen size. Is text readable on the responsive site across different sizes?

   Navigation: How does navigation change on smaller screens? Is the responsive site's navigation user-friendly on mobile devices?

3. **Document Your Findings**

   Write a summary of your observations. Include specific examples of how the responsive site's design benefits users compared to the non-responsive site.

   Engage in a class discussion to discuss any potential improvements that could be made to the non-responsive site.

WEB ACCESSIBILITY

Web accessibility is about making websites easy to use for everyone, including people with disabilities. For instance, someone with poor eyesight should be able to read the text on a website and someone who cannot use a mouse should be able to navigate using a keyboard. The goal is to create an inclusive digital environment where all users can interact with and benefit from web content, regardless of their abilities.

Key Aspects of Web Accessibility

1. **Perceivable:** All users should be able to see or hear the information on a website. This comprises of providing text alternatives for non-text content, making content adaptable and ensuring that the content can be both seen and heard. For example, images should have text descriptions for people who cannot see them, this will often be read out by a screen reader.
2. **Operable:** Users should be able to navigate the website in different ways, like using a keyboard instead of a mouse. For instance, we should be able to use the tab key on the keyboard to move around the different elements of the webpage.
3. **Understandable:** The website should be easy to read and use. Avoid complicated designs or unpredictable behaviour.
4. **Robust:** Websites should work well with different devices and assistive tools like screen readers and future technologies that will emerge. To do this, you need to use valid HTML and follow best practices in developing your website to ensure compatibility and accessibility.

Importance of Web Accessibility

1. **Inclusivity:** This allows everyone, including people with disabilities, to access and use websites. Accessibility ensures that people with disabilities have equal access to information and services, promoting inclusivity and equal opportunity.
2. **Legal Compliance:** Some countries have laws requiring websites to be accessible. such as the Americans with Disabilities Act (ADA) in the U.S. or the Web Content Accessibility Guidelines (WCAG) internationally. Compliance helps avoid legal issues and ensures that your website meets legal standards and does not violate the fundamental right of the users.
3. **Enhanced User Experience:** Accessibility improvements, like clear navigation, make the website easier to use for everyone and not just those with disabilities.
4. **Broader Audience:** Accessible websites can be used by more people, including those with challenges such as a broken arm. By doing this and making your website accessible, you could reach a larger audience.
5. **SEO Benefits:** Search engines reward accessible websites with higher rankings. They are awarded because they follow best practices for structure and content, making it easier for search engines to index and rank them.

Activity 6.21

Evaluating a Website for Accessibility

1. Copy the GES website URL *https://ges.gov.gh/* and paste it into the address box of your web browser
2. Press **Enter** or click on the forward arrow to check the website's accessibility
3. Observe the result on the left panel
4. Scroll down to see where the accessibility issues where highlighted
5. Write a brief report, descripting the issues you find and suggest ways to make the website more accessible
6. Share your report with your peers

Activity 6.22

Research Assignment on Website

In groups of no more than five, using the internet or online resources and your understanding of concepts of website, complete the following

1. Discuss and document what you think of the following questions.
   a. Why do you think Google prefers responsive designs? How could this impact your business?
   b. Why is it cost-effective to use responsive design and ensure accessibility together for your business? Discuss with examples from local or global websites.
   c. How can you convince a local business owner to invest in a responsive and accessible website? Write a persuasive paragraph.
2. Create a responsive website to show your responses to the above questions.
3. Share the link to your website with your teacher and peers, to get feedback.

CREATING A SHOPPING CART

A **shopping cart** is like a basket you use when shopping online. It helps you pick the items you want to buy, shows you what you have chosen, and lets you pay for them when you are ready. Think of it as your helper when shopping on websites like kromonline.com.

It is an essential component of any e-commerce website. It allows users to select products, view their selected items and proceed to the checkout just like Jumia. See **Figure 6.5** for a typical image of a shopping cart.

Figure 6.5: Shopping Cart

Basic Structure of a Shopping Cart

The structure of a shopping cart typically includes the following:

1. **Cart Container:** This is the main box where all your selected items are placed. Take it as a basket on a website, where you can see all the things you have picked. This is just like the normal basket where you keep your chosen items when you visit Kumasi City Mall, Melcom or Shoprite. The cart is virtual, while baskets are physical to the user.
2. **Item List:** This shows you details about the items you have added. The details may include the name, quantity and price of the item(s).
3. **Cart Controls:** These are buttons that you use to modify your item(s). You can increase or decrease the quantities, remove items and proceed to checkout.
4. **Summary:** A summary section displays the total price and other relevant information. This is the final section where you see the total amount of money you need to pay for everything in your cart. For instance, if your cart has two T-shirts at GH¢580.99 each and a pair of shoes for GH¢358.99, the summary will show: Total: GH¢1,520.97.

Activity 6.23

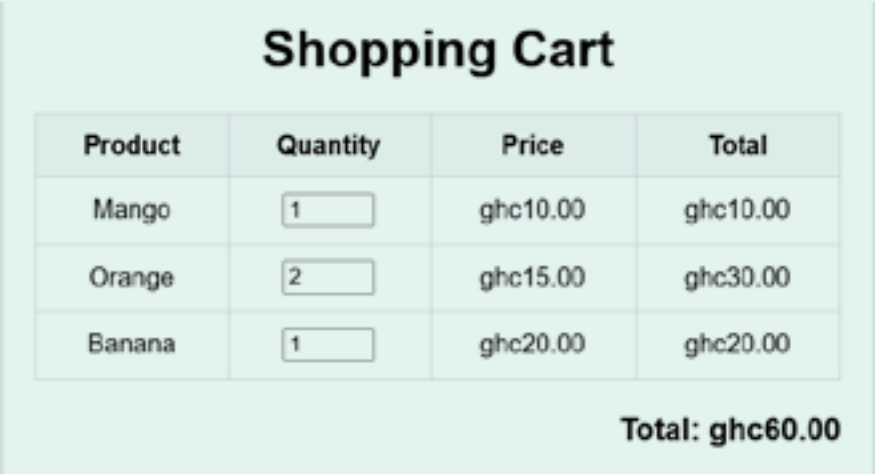
Creating a Shopping Cart Structure

In groups of no more than five, chose a local company (or create a fictional brand) and create an online e-commerce website. You will be competing with the other teams in your class for who had created the best-designed website.

You must use HTML and CSS to design your website, and you must include the following elements on your website:

- Headers
- Paragraphs
- At least one table
- At least one image
- At least one video
- An e-commerce shopping cart to allow customers to purchase items

Below is an example of what a shopping cart looks like, you can choose to model that layout, or you can create your own



Present your website to your teacher and peers for discussion and feedback.

PAYMENT GATEWAY INTEGRATION

A payment gateway is like a digital cashier. When you buy something online, the gateway takes your payment details, sends this information securely to the seller's bank and makes sure the payment is accepted before telling both you and the seller. The simple way to understand gateway is to think of it as a bridge between the buyer (you) and the seller to ensure the payment is safe, successful and fast.

The importance of a Payment Gateway

Payment gateways are very important components of online payment systems. They securely transmit transaction information between a customer's and merchant's banks. Some the importances of payment gateway include:

1. **Security:** Your card details are kept private and safe using special techniques like encryption.
2. **Convenience**: You can use many payment methods like bank cards and mobile money.
3. **Efficiency:** Payments are processed in seconds with quick alert.
4. **Compliance:** They follow strict rules to protect both the buyer and seller from fraud. They follow standards such as PCI-DSS (Payment Card Industry Data Security Standard).

Examples of popular Payment Gateways in Ghana

1. **Hubtel** is a locally built payment platform that supports mobile money (MTN MoMo, AT Money, Telecel Cash), bank cards, and QR codes. Many businesses like online shops use it for seamless integration with their online stores and physical shops.
2. **ExpressPay** is a Ghanaian payment gateway for mobile money, card payments, and bank transfers. It provides an app for utility bill payments, such as school fees and online shopping.
3. **Slydepay** supports payments via mobile money, bank cards, and QR codes. It offers simple integration for small businesses and online stores in Ghana.
4. **iPay** focuses on simplifying mobile money and card payments for businesses. Businesses can both receive payments and manage transactions using its platform.
5. **Paystack** is a popular payment gateway across Africa, including Ghana, which allows businesses to accept card payments, mobile money, and bank transfers. It is known for its smooth onboarding process and developer-friendly tools.
6. **Flutterwave** operates in multiple African countries, including Ghana. It supports mobile money, card payments, and bank transfers. It offers businesses the ability to accept payments from customers across Africa, as well as globally.
7. **MTN Mobile Money API** specifically designed for MTN MoMo users, allowing merchants to directly integrate mobile money payments into their platforms. MTN MoMo is widely used in Ghana making this API highly relevant for local businesses.
8. **Zeepay** enables businesses to accept remittances, mobile money, and card payments. It focuses on linking international remittances to local mobile money wallets.
9. **Global Accelerex** provides secure payment solutions for online businesses, including card and mobile money integrations. It offers tools to streamline payments and reconciliation.

Activity 6.24

Integrating a mock payment gateway

In your, extend your e-commerce website from **Activity 6.23** to include a payment gateway

1. Follow the instructions in the tutorial provided by your teacher (Link to tutorial for integrating a mock payment gateway)
2. Integrate a mock payment gateway, using a sandbox environment available here (Link to either paypal.com or stripe.com)
3. Extend your shopping cart so that you can:
   a. Add items

b. Remove items
c. Change the quantities
d. Calculate the totals
e. Add security measures to your e-commerce website, including HTTPS and SSL certificates, to protect the customers data

USER ACCOUNT MANAGEMENT

Enhancing User Experience

User accounts let you personalise your online shopping experience. For example, when you log into an account on a shopping site like Jumia, Melcom Online or portfoliogh.com, you can see your order history, inbox, vouchers, saved items, recently viewed items, recent searches, and quickly check out, without entering your details every time.

Order Tracking

With an account, you can check where your items are after you buy them. For instance, if you order a phone on the Jumia online shop, from the comfort of your house, the website can tell you whether it's "Being Prepared," "Shipped," or "Delivered. E-commerce provides real-time updates on the status of your orders, thereby improving transparency and your satisfaction.

Customer Support

If something goes wrong, your account lets you contact customer support, track your complaints, and get quicker solutions from customer service representatives who can access your profiles, to provide support and resolve issues more efficiently.

Security

Accounts store private information like your email, address, and bank details. With implementation of secured login mechanisms using authentication, there is less risk of unauthorised access and fraud.

Basic User Account Features

Let us look at some basic account features of online system, especially those that deal with sensitive data and finances.

1. Account Creation

Before you can use the features of an account on an e-commerce site, you first need to create one. In creating an account, you must fill a registration form. The registration typically requests your personal details such as name, email address, and a password. To prove you are a human and not a robot, websites often use CAPTCHA tools, such as identifying pictures of cars, fire hydrants, buses, bicycles, bridges, animals or typing distorted letters. You then need to confirm you are not a robot by providing accurate responses to the CAPTCHA questions.

CAPTCHA

CAPTCHA stands for **Completely Automated Public Turing Test to Tell Computers and Humans Apart**. It is a test that websites use to make sure the person accessing the site is a real human and not a robot or computer program (bot). Bots often try to create fake accounts or spam websites, and CAPTCHA prevents this by asking questions that are easy for humans but hard for bots. By stopping bots, CAPTCHA prevents cyberattacks, like overwhelming the website with too many requests (DDoS attacks) and ensures that only real humans can access sensitive areas of a website, like registration forms or payment sites. **Figure 6.6** below shows an example of a Login page with CAPTCHA.

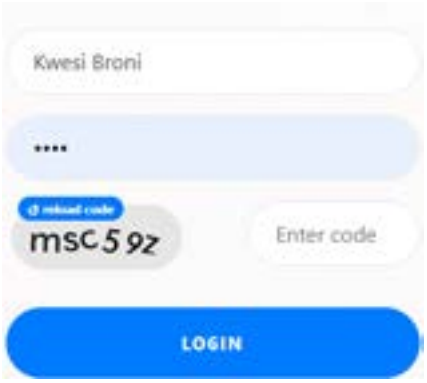


Figure 6.6: Login page with CAPTCHA

2. Login

After creating your account, you log in with your email and password. Some websites add extra security by sending a code to your phone for you to enter after your password. This type of login has more than one layer of verification. They have two-way or three-way verification mechanism, generally called either two-factor authentication (2FA) or multi-factor authentication (MFA). They are designed to identify the authorised accessor of the account. If you provide the required login credentials, you are given access, otherwise, you will be locked out. The secure login process includes password hashing/salting and encryption.

Form Structure

A registration form is like a paper form; however, it is online. It allows you to enter your details, such as your username, email and password.

Every registration form has a structure which make it clear and easy to use.

1. **Form Elements**: These are the parts of the form that you fill out during registration. They typically include text fields, email fields, password fields, and submit buttons.
2. **Labels and Inputs**: A label is the name written next to or above a field (e.g., "Email Address") to show what you need to enter and the input is where you type the required information.
3. **Submit Button**: Once you have entered all required information, you need to click the Submit button to send your details to the website, so that it can create your account.

Figure 6.7 shows an example structure of a registration form.

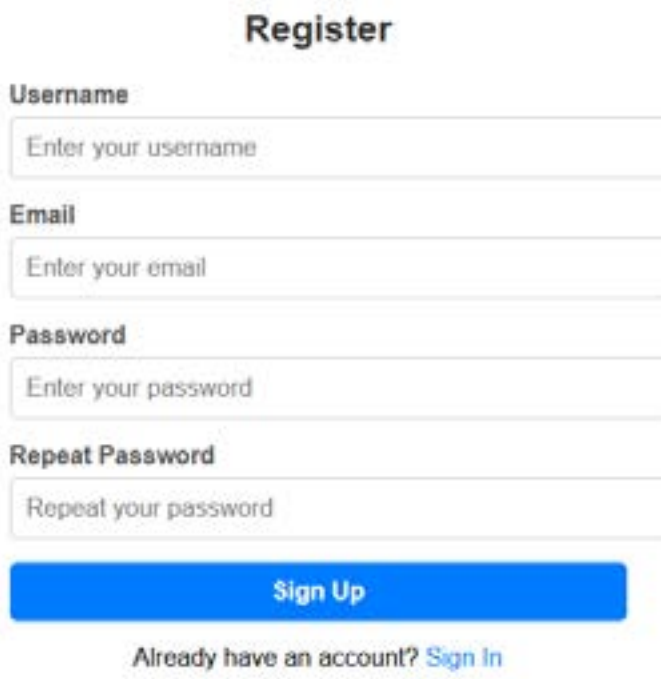


Figure 6.7: Structure of a registration form

Activity 6.25

Design and Fill Out a Registration Form

In groups of no more than five,

1. Search online for "how to create forms using HTML"
2. Pay key attention to the use of
   a. Labels
   b. Names
   c. Requirements
   d. Difference between placeholders and values

3. Create a simple form using the following fields:
   a. Username
   b. Email Address
   c. Password
   d. Confirm Password (to retype and confirm your password)
4. Add a Submit button at the bottom of your form.
5. Do not worry about styling your form at this point but ensure your items are in vertical order.

Managing User Accounts

When you create an account on a website, you also get a **user profile**. This is a special section where you can see and change your personal information.

Profile Management

Profile management enables you to see your username, email, and other details you added when creating your account. If there are changes to your details, like a new email address or password, you can modify your profile and save the changes. This action allows you to manage your profile by updating it in the database.

SECURITY AND TRUSTWORTHINESS

Security is important for websites because they usually collect and store sensitive information, such as name, email, payment information and order history. If this information is not protected, it can lead to serious issues such as identity theft and fraud. To protect one's data on a website, they include integrated layers of security mechanisms that make you trust the site and feel comfortable using it. If a website has poor security, people stop using it, and the company will lose money or even face legal actions.

In Ghana, the Data Protection Act ensures that companies manage personal information safely. If they do not follow these rules, they will face **fines** or other **legal consequences**.

Activity 6.26

Security in E-Commerce

1. Organise yourselves into groups of no more than five. In your groups, discuss and document the following
   a. Why it is important to have a strong password?
   b. What would make you trust an online store?
   c. What can happen if someone else gets access to your personal details?
2. Share your findings with your peers.

When creating an e-commerce website, it is important to follow these three key security principles, to ensure that users' data and the website itself are safe and reliable.

1. **Confidentiality:** Confidentiality ensures that private information, such as usernames, passwords, or payment details is only accessible to the people who are authorised to see it. If you enter your payment details when buying something online, only you and the website should be able to see them — not hackers or unauthorised users. To ensure confidentiality, websites should use secure logins and encrypt sensitive data, so even if it is stolen, it cannot be read.
2. **Integrity:** Integrity means making sure the information on the website is accurate and has not been tampered with by unauthorised people. For instance, if you buy an item worth 100 cedis, someone else should not be able to change it to 500 cedis in the system.
3. **Availability**: Availability ensures that authorised users can access the website and its services whenever they need them. If a customer wants to place an order, the website should be functional and not down due to cyberattacks or technical issues. This also makes the website more reliable.

Secure Authentication Mechanisms

To protect users and sensitive data, e-commerce websites implement secure ways of handling passwords and user logins. They include password hashing and salting. Let us discuss them.

1. **Password Hashing**

   Password hashing is like converting a password into a scrambled code that looks random. This "code" cannot be reversed to reveal the original password. Even if someone steals the hashed passwords, they cannot easily figure out the actual passwords. Imagine you create a password, "Abu1957", the website hashes it into something like "bua75@91dfg#". Each time you log in, the system compares the new hashed version of your password with the stored one to confirm it matches.

2. **Password Salting**

   Salting adds a unique, random string (called a "salt") to your password before hashing it. If two people use the same password, the hashes will still look different because of the unique salt. For instance, for "Abu1957", one user might get "saltA@Uba91#dfg" while the other gets "saltB@xy75#hij".

Implementing Two-Factor Authentication

Two-factor authentication (2FA) adds another layer of security by requiring users to provide two forms of authentication.

In real life, 2FA works this way: the user enters their username and password. After the password is confirmed, the user must provide a second verification code. This code is then sent to their phone, email or generated by an app. If the code is correct, the user gains access. You must note that even if someone steals your password, they cannot log in without the second factor.

Activity 6.27

Securing your password

Organise yourselves into groups of no more than five. In your groups,

1. Discuss the following questions:
   a. Why do you think websites ask for your password twice?
   b. What would happen if you forgot your password after creating an account?
   c. How can you make your password secure?
2. Discuss common issues in form validation and how to handle them.
3. Explore additional validation options, such as server-side validation or more complex user feedback mechanisms.
4. Create a PowerPoint presentation to show your findings on additional validation options

Present your slides to your class for review and feedback.

DATABASE CONCEPTS

A **relational database** is a way of organising data into tables. This is just like spreadsheets, where information is stored in rows and columns. These tables are connected (or "related") using shared values, making it easy to manage and retrieve data.

Importance of a Database

A **well-designed relational database** is vital for an e-commerce website because it ensures smooth operations, accurate data, and a better customer experience. The following are some reasons why relational databases are important:

1. **Efficiently Manages Large Volumes of Data** - In an e-commerce system, you need to manage large amounts of data. This includes products, customers, and orders, which are all stored in separate tables that are connected. Each table is connected to the others through shared values, allowing the website to easily manage and update product listings, customer information, and orders, without duplicating data.

   For instance, when a customer places an order on Jumia, Melcom Ghana or Jiji, the database automatically links the customer with their order details, and updates the product stock levels.

2. **Ensures Data Integrity and Reduces Redundancy** - A well-designed relational database ensures that when a piece of information changes, like a customer's address, it updates across all relevant tables automatically, keeping everything accurate. Without a database, you might end up storing the same information in multiple places, creating unnecessary repetition, like people do in paper-based storage. With a relational database, the customer's information is stored once in the "Customers" table and linked to each order they place.

3. **Facilitates Complex Queries and Data Analysis** – Relational databases support complex queries that help the e-commerce business find specific products by name, category, or price range. Queries allow the e-commerce business to check a customer's purchase history or see how often a particular product is bought and they can generate sales reports for a specific period or by product category; allowing the business to analyse the data that it is storing.

4. **Dynamic and Responsive Functioning** — This function handles multiple aspects of e-commerce, such as product listings, customer details, orders, and transactions, making the website dynamic and responsive. Relational databases respond to requests in real-time (e.g., updating product stock or showing up-to-date customer info) and keep the site's operations smooth and quick even when multiple customers are interacting with the website simultaneously.

Activity 6.28

Simple Database Structure

1. Individually, create the following tables, using either paper or excel
   a. product
   b. customer
   c. employee
2. Discuss with your peers how these three tables are related to each other.

Key Components of a Relational Database

There are key components of databases that you must consider. These include the following:

1. **Tables (Relations)**

   A **table** is a collection of related data entries in a structured format consisting of **columns** and **rows**. Each table stores data about a specific entity. For example:

   a. Product Table: Holds details about items available for sale (e.g., product name, price, description).

   b. Customer Table: Stores information about customers who buy products (e.g., name, email, address).

   c. Order Table: Keeps records of transactions between customers and the store (e.g., order ID, order date).

   d. Transaction Table: Contains details about payments made by customers (e.g., payment method, transaction amount, etc).

Table 6.3 below shows an example of a product table

ProductID	ProductName	Price	Quantity
1	Wristwatch	GH¢ 373.30	4
2	Tecno Camon 30	GH¢ 3494.80	1
3	Toshiba Laptop	GH¢ 5536.00	1

2. **Columns (Fields/Attributes)**

   Each **column** in a table represents a specific **attribute** or property of the data being stored. For example, in our **Customers Table** (table 6.4), the columns include CustomerID, Name, and Email. These columns define the characteristics of the data stored in each row.

Table 6.4 below shows an example of a customer table

CustomerID	Name	Email
1	Daniel Miheso	miheso@example.com
2	Raphael Senyo Dordoe	dordoe@example.com
3	Mark Anibrika	anibrika@example.com

3. **Rows (Records/Tuples)**

   Each **row** (also called a **record** or **tuple**) represents an individual entry in the table. A row contains a set of values corresponding to each column in the table.

For example, in Table 6.4 above, each row represents one customer, with details like their Customer ID, name, and email.

4. **Entities and Attributes**
   a. **Entities** refer to the objects about which data is stored in the database. For instance, a product, a customer, or an order can all be considered entities in an e-commerce website.
   b. **Attributes** are the specific details or properties that describe an entity for example:
      i. **Product** - ProductID, Name, Description, Price, StockQuantity.
      ii. **Customer** - CustomerID, Name, Email, Address, PhoneNumber.
      iii. **Order** - OrderID, CustomerID, OrderDate, OrderTotal.
      iv. **Transaction** - TransactionID, OrderID, PaymentMethod, TransactionDate, Amount.

Relationships between Entities

Relationships link the different tables together using shared data. There are different types of relationships that can be established between tables, such as:

1. **One-to-Many (1:M):** This relationship means one entity in a table can be associated with many entities in another table. For example, a customer can place many orders, but each order is linked to only one customer.
2. **Many-to-Many (M:N**): This relationship means that multiple entities in one table can be associated with multiple entities in another table. For example, a product can appear in multiple orders, and an order can contain many products.

Activity 6.29

Designing Tables

1. Individually, on paper, design tables for customers, orders, products and transactions.
2. Define the relationships between these tables (e.g., one customer can place many orders).
3. Share your design with your peers in the class.

Keys in Databases

Primary Key

A **primary key** is a unique identifier for each record in a database table. It ensures that no two rows in a table have the same value for this field, to maintain order and integrity.

Key Characteristics

1. **Uniqueness** - Each value in the primary key column(s) must be unique. For instance, in a "Customers" table, CustomerID will be a primary key.
2. **Non-null** - A primary key cannot be empty. It means that a primary key cannot contain NULL (empty) values. This requirement ensures that every record has a valid and identifiable primary key value.
3. **Immutability** - The value of a primary key should remain unchanged throughout the record's lifecycle. Thus, the value should not change over time. If the value needs to change then the whole record needs to be deleted and a new record created, with a new unique identifier e.g. customerID.

Importance of Primary Keys

1. **Ensuring Data Integrity** — The primary key enforces uniqueness, preventing duplicate records from existing within the table and therefore maintaining the integrity of the data.
2. **Efficient Data Retrieval** - Database systems use primary keys to locate and retrieve data quickly. This efficiency improves query performance.
3. **Establishing Relationships** - Primary keys are often referenced by foreign keys in other tables, to create relationships between the tables. This referencing helps maintain referential integrity.

Foreign Key

A **foreign key** is a column in one table that references the primary key in another table. It creates a link between two tables, thereby enforcing a relationship.

Key Characteristics

1. **Referential Integrity** - Foreign keys help maintain referential integrity by ensuring that the value in the foreign key column corresponds to a valid value in the referenced table's primary key column. This ensures that the value in the foreign key column exists in the primary key column of the referenced table. For instance, if a foreign key in the "Orders Table" refers to a "CustomerID" in the "Customers Table," the system will ensure that every "CustomerID" in the "Orders Table" exists in the "Customers Table."

2. **Nullable** - Foreign key columns can be empty (NULL). This means that some records may not need to be linked to another table. This allows flexibility in relationships. For instance, a "Product Table" may include a column for OrderID as a foreign key. If a product has not yet been ordered, the OrderID column can remain NULL (empty).
3. **Consistency** – Foreign keys enforce consistency, by ensuring data in the related tables stay aligned and are valid. This prevents errors, like having an order that points to a deleted customer, or a payment linked to a non-existent order. If a customer is deleted from the "Customers Table", the database can either prevent the deletion or delete all related orders in the "Orders Table," thereby ensuring consistency.

Importance of Foreign Keys

1. **Establishing Relationships** - Foreign keys create relationships between tables, such as One-to-Many, where one customer can have many orders and Many-to-Many, where multiple orders can include multiple products. It connects data logically, enabling better organisation and retrieval. For instance, a "Customers Table" is related to an "Orders Table" by the foreign key, "CustomerID". This relationship shows which order(s) belong to which customer.
2. **Maintaining Data Integrity** - They ensure that references between tables are valid, avoiding issues like missing data and inconsistent data. This maintains trust in the database's accuracy. For instance, deleting a customer will ensure that all their orders are also deleted (cascading delete) or it prevents the deletion if orders still exist.
3. **Facilitating Joins** - Foreign keys make it easier to join tables during queries, allowing for complex data retrieval and analysis. For instance, an SQL query can join the "Customers Table" and "Orders Table" on the "CustomerID" foreign key to display customer names and their corresponding orders.

Examples

Consider an e-commerce database with the following tables

1. **Customer Table**: **In the customer table, the primary Key** is **CustomerID** (highlighted in red) as shown in **Figure 6.8**.

CustomerID	Name	Email
1	Daniel Miheso	miheso@example.com
2	Senyo Dordoe	*dordoe@example.com*
3	Mark Anibrika	anibrika@example.com

Figure 6.8: Customer Table

2. **Order Table:** In **Figure 6.9** below, the primary key is **OrderID** (highlighted in green) and the foreign key is **CustomerID** (referencing the **CustomerID** in the

Customer table, highlighted in red). This relationship ensures that each order is associated with a valid customer.

OrderID	OrderDate	CustomerID
1011	2024-11-21	1
1012	2024-11-22	1
1013	2024-11-21	2

Figure 6.9: Order Table

3. **OrderProduct Table:** In **Figure 6.10** there are two primary keys (OrderID and ProductID). This is because the OrderID as a primary key is not unique since multiple products can be linked to the OrderID.

   For example, OrderID (1013) can be linked to products (B4 and B7).

   On the other hand, the ProductID can also be linked to more than one OrderID. Therefore, to get a unique key, we combine the two primary keys, which is known as a composite key (highlighted in blue).

OrderID	ProductID	Quantity
1013	B4	3
1012	C6	6
1013	B7	2

Figure 6.10: OrderProduct Table

Activity 6.30

Identifying Relationships

1. Create the following tables: Customers, Orders, Products
2. Add attributes to the tables.
3. Identify which fields should be primary or foreign keys and explain your reasons why.
4. Share your response with your peers in class.

Relational Database Tables and Fields

Tables and fields are essential components of a relational database, enabling structured storage and easy retrieval of information. For instance:

1. **Product Table** - Fields might include ProductID, Name, Description, Price and StockQuantity.
2. **Customer Table** - Fields might include CustomerID, Name, Email, Address and PhoneNumber.
3. **Order Table** - Fields might include OrderID, CustomerID, OrderDate, OrderTotal.
4. **Transaction Table** - Fields might include TransactionID, OrderID, PaymentMethod, TransactionDate, and Amount.

Schema Design

A **database schema** is a conceptual layout that defines the structure of the database. It outlines the tables, fields, relationships and constraints to ensure that the data remains accurate and accessible.

Table 6.5 Example of Schema Design of Product Table

Field	Type	Purpose	Constraints
ProductID	INT	Unique identifier for the product	PRIMARY KEY (ensures uniqueness)
Name	VARCHAR (255)	Product name	NOT NULL (every product must have a name)
Description	TEXT	Detailed product description	NULL allowed (optional field)
Price	DECIMAL (8, 3)	Product price	NOT NULL, CHECK (Price >= 0) ensures a valid price
StockQuantity	INT	Available stock	NOT NULL, CHECK (StockQuantity >= 0) ensures non-negative stock
Brand	VARCHAR(100)	Product brand name	NULL allowed (optional field)

Queries

SQL queries are written to perform various operations, such as retrieving, updating, or deleting data. This helps users to interact with a database to retrieve, update and manage the data.

Activity 6.31

Installation of MySQL

Caution: Ensure you are connected to the Internet for this activity.

1. Click on this link to download MySQL: *https://dev.mysql.com/downloads/installer/*

2. Click the download button of the second option (mysql-installer-community)
3. Select **No thanks, just start my download**
4. Wait for the download to complete
5. Double click the application to install it
6. Under the choosing a set up type, select **Full,** then click on next
7. Under check requirements, click on **execute** and wait for the status to upload
8. Click install the Microsoft Visual C++ if prompted
9. Click on next when the status column shows as "INSTALL DONE" for all items
10. On the installation popup page, click on execute to install the items
11. Click next three times repeatedly when done
12. Set a strong password that you can remember (leave this section for your teacher if you are using the school's computer)
13. Click next three times repeatedly when done
14. Click on Execute, wait for a while for it to complete
15. Click on finish
16. Click on next again
17. Click on finish
18. Click on next
19. Enter the password you set earlier, click on check and then click on finish
20. Click on next and finish again for a new window to open (Welcome to MySQL Workbench)

You are now ready to study and practice the basic operations that can be done on data in the database, using SQL queries.

Activity 6.32

SQL practice

1. Download this file by clicking on the link: *https://drive.google.com/file/d/1KF-dC14AGi5u3T66ntEhmqRcHYnAG6U0/view?usp=drive_link*
2. Watch this video tutorial and follow the instructions

   https://drive.google.com/file/d/1yG4UKdeQ1DQ-ojaC3zYhLg44-rDUjepE/view?usp=drive_link

SQL Statements

An SQL command (statement) is used to retrieve data from one or more table(s). It specifies the columns to be retrieved and the conditions for selecting records. This means it can filter, sort and group data by using additional clauses. For example:

1. **INSERT Statement**

   An SQL command is used to add new records to a table. It specifies the table and the values for each column. With one query, it can add multiple rows.

2. **UPDATE Statement**

   An SQL command is used to modify existing records in a table. It specifies the table columns to be updated and the conditions for updating records.

3. **DELETE Statement**

   An SQL command is used to remove records from a table. It specifies the table and the conditions for deleting records. Be careful when using this statement, as it permanently deletes data from the database.

4. **WHERE Clause**

   An SQL clause which specifies the conditions that filter records in a query. It is used with SELECT, UPDATE, and DELETE statements. It can use multiple conditions with AND, OR and NOT (logic operators).

5. **ORDER BY Clause**

   An SQL clause that is used to sort data in ascending (default) or descending order. You can specify either ascending (ASC) or descending (DESC) order, if neither is specified then the data will be ordered in ascending order by default.

6. **GROUP BY Clause**

   The GROUP BY clause is used to group rows with the same values, in specified columns, into summary rows. It is commonly used with aggregate functions.

7. **JOIN Clause**

   The JOIN operation combines rows from two or more tables, based on a related column. There are different types of joins, including INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN.

8. **Aggregate Functions**

   Aggregate functions perform calculations on a set of values and return a single value. Common aggregate functions include COUNT(), SUM(), AVG(), MAX(), and MIN().

9. **Constraint**

   Constraints are rules applied to table columns, to ensure data integrity. Constraints include NOT NULL, UNIQUE, CHECK, and FOREIGN KEY.

Activity 6.33

SQL Statements

In groups of no more than five, complete the following

1. For each of the 9 SQL statements above, write down a real-world example of when that would be used in a database and the reason for using that statement
2. For each of the constraints, provide a definition and a reason why it might be important in a database:
   a. NOT NULL
   b. UNIQUE
   c. CHECK
   d. FOREIGN KEY
3. Share your responses with your peers

Activity 6.34

Organise yourselves into groups of no more than five. In your groups, using the e-commerce website that you created in **Activity 6.23** and the tutorials provided by your teacher, complete the following:

1. Create a database to hold and manage customer details (users of the website)
2. Add a feature for a new user to register for an online account, using a secure username and password
3. Add a feature to allow that new user to login
4. Add a feature to allow that new user to logout

EXTENDED READING

- Follow the link *here* or *https://www.w3schools.com/html/html_links.asp* to get more information on links in HTML.
- Refer to the URL *here* or *http://www.w3schools.com/html/html_media.asp* for more information on audio and video formats.
- The **DM ICT Centre** is an organisation that provides ICT Training and Education, Digital Literacy Programs, Software and Hardware Solutions, E-Learning Support, Internet Access and Digital Resources and Business and Entrepreneurship Support. The centre requires a website that can be designed and implemented by using HTML for structure, CSS for styling, and responsive frameworks like Bootstrap for its services. As a computing student, research and design this website that will best fit for DM ICT Centre.

Review Questions

1. List three key design elements you can add to a webpage using a web editor.
2. Why is UX (User Experience) important in web design.
3. What are the essential elements of an e-commerce web page?
4. What is a relational database model?
5. How would you add an image to a webpage using a web editor?
6. Why is it important to include a responsive design for an e-commerce webpage?
7. Write a query to retrieve all orders placed by a customer with CustomerID = 2.
8. How can you ensure secure transactions on an e-commerce website?
9. Why is accessibility an essential consideration in web design?
10. Analyse how integrating multimedia elements can improve the user interface of a webpage.
11. Compare the features of two popular e-commerce websites and identify key design elements that enhance user experience.
12. Given a relational database for your school's library system, propose a schema to include Books, Members, Loans, and Staff as best Computer Science student in the school.
13. Develop a relational database schema for Aflao Community Day Senior High School system.
14. Create a shopping product and cart tables for Jumia.
15. Create a personal web page which requests for personal information with placeholders, save and cancel buttons.