**CHAPTER ONE**

1. **INTRODUCTION**

Technology has penetrated all areas of life and the use of information technology(IT) has work procedures and approaches. It has increased the global dissemination of information. Electronic resources are easily accessible in the remote. Electronic resources solve storage problems and control the flood of information. Print sources are being digitized. There is a great need to study the use of electronic resources and investigate the factors that are a hindrance to their use. An electronic resource is any information source that the library provides access to in electronic format. An electronic resource is an online raw material in which collections are stored in digital formats and accessible by the use of computers. The electronic content may be stored locally, or accessed remotely via computer networks. Technology has penetrated all areas of life and the use of information and communication technology (ICT) is common place in the 21st century. Libraries use ICT for better service and satisfying diverse user needs. Libraries have transformed into digital and virtual libraries where books, journals, and magazines have changed into e-books, e-journals, and e-zings. This has increased the global dissemination of information. Electronic resources are easily accessible in remote areas. Electronic resources solve storage problems and control the flood of information. Print sources are being digitized. The rapid growth of new technologies has changed the communication process and reduced the cost of communication for individuals. Electronic information sources can be seen as the most recent development in information technology and are among the most powerful tools ever invented in human history. Electronic information sources are becoming more and more important for the academic community (Kumar and Kumar, 2008). University academics are a unique population and rely on recent and timely information. Electronic resources are now used more often that print resources (Morse and Clintworth, 2000). There is a great need to study the use of electronic resources and investigate the level of satisfaction among academics.

An electronic resource is a type library whose collection are stored in electronic format and accessible through the use of computer network of information retrieval system. With the arrival of computers, the concept centered on large bibliographic databases, the now familiar online retrieval and public access systems that are part of any contemporary library. When computers were connected into large networks forming the Internet, the concept evolved again, and research turned to creating libraries of digital information that could be accessed by anyone from anywhere in the world. Materials in e-resource are basically electronic materials, these materials are also referred to as digital materials. An early example of a digital resource library is the Educational Resources information center (ERIC) which is an online digital library for education research and information and their mission is to provide comprehensive, east-to-use, searchable, internet-based bibliographic and full-text database research for educators, researchers and the general public. In this world of fast change, digital resource offer many facilities to education in general and to higher education in particular.

**1.1 OBJECTIVES OF THE STUDY**

The objectives of this paper are to:

* Analyze dependency of the lecturers and students on e-resource, the impact of the e-resource on their academics efficiency and problems faced by them while using the e-resources. It is particularly conducted to access the benefit of the resources over conventional sources of information.

• Ascertain the ability of academics to use electronic resources

• Discover the type of sources which are used most often among academics

• Explore the purpose for using electronic resources

• Ascertain the satisfaction level of academics with electronic resources

• Use of electronic journals increases with time.

• Age and/or academic position are inversely related to the use of electronic

media and journals.

• There is a gradual reduction in the use of printed journals as users prefer and

use the electronic format more.

• With increased use, users access the electronic format more frequently.

• The use of a journal is not necessarily an indication of the preference of users. There may be an increase in the acceptance and frequency of use of the electronic format merely because the traditional print format is no longer easily available.

• Accessibility and desktop access, home access, ease of retrieval, and hyperlinks to outside content were the arguments cited most often as the advantages of electronic journals. The disadvantages mentioned most often were the lack of back issues and problems with reading a text from the computer screen

*University Faculty Use of Electronic Resources: A Review of the Recent Literature (PDF Download Available)*. Available from: <https://www.researchgate.net/publication/263504327_University_Faculty_Use_of_Electronic_Resources_A_Review_of_the_Recent_Literature> [accessed Jun 3, 2017].

“University Faculty Use of Electronic Resources: A Review of the Recent Literature,” Adebayo Muritala

**1.2 USEFULNESS**

Advances in computer application during the past few decades have brought radical changes in the way information is gathered, stored, organized, accessed, retrieved and consumed. The application of computers in information processing has brought several products and services to the scene.

Further, the distribution time between the product publication and its delivery has been drastically reduced, the internet can be used for efficient retrieval and meeting information needs. This is very important for university libraries since most of them call for more and more research work and the digital recourses is found to be less expensive and more useful for ease of access. This is especially helpful o distant learners who have limited time to access the libraries from outside by dial-up to commonly available electronic resource, mainly CD-ROM, OPACs and internet which are replacing the print media.

Digital library would improve the quality of teaching and research through the provision of current e-books, journals, and other library resources. It will also enhance scholarship, research and lifelong learning through the establishment of access to shared global virtual archival collections.

<http://southernlibrarianship.icaap.org/content/v10n01/sharma_c01.html>

<https://www.researchgate.net/publication/263504327_University_Faculty_Use_of_Electronic_Resources_A_Review_of_the_Recent_Literature> [accessed Jun 3, 2017]

**1.3 RELEVANCE OF THE STUDY**

The emergence of electronic information resource has tremendously transformed information –handling and management in Nigerian academics environments, and universities libraries in particular(Ani and Ahiauzu,(2008) ).Its relevance to this study is to know the quality of information retrieved through electronic resources. To study the impact of electronic resources and services on the academic work of lecturers and students and also to allow both lecturers and students to be able to upload their thesis for reference purposes and also for people to review and leave report about their work.

E-resource have been the keepers and distributors of books, journals, maps and other materials that are used by students in the learning process. They have also been the legal deposit of part of the products of scholarly publications – theses & dissertations, articles, technical reports, etc.

In general, students have been patrons of the libraries of their institutions. In order to make more contents available and thus benefit students and faculty, pools of institutions have engaged in commuting items and/or their copies.

There is no reason for digital libraries not to have the same functions of traditional libraries, except that they can add functions and value due to their digital and networked nature. On the other hand, class notes, simulators, spread sheets and other materials created by faculty have traditionally been made available by their authors through copies in an informal distribution situation.

The use of ICT – Information and Communication tools has changed the informal distribution to computer and networked based solutions. As consequence, a great amount of contents became available from computers lacking the necessary identification and access control. Identification (description of the digital contents) is important for the search and retrieved actions by users. Digital libraries are suitable tools to manage courseware and additional reference items used in class. Some reasons for this use are:

* Management of documents in all formats in a unified way – texts, animations interactive exercises, audio files, video streams, e-books, e-journals and online tests can be stored, described and distributed through computers and networks. The management is independent of the type of information, as long as it can be stored in digital files. It also can be shared without human intervention making the whole process faster and cheaper.
* Access control – contents can be assigned different types of access according to the classes of users that are entitled to them. Authors can decide if their works are to be used by their students only, by any student of a given institution or the public in general.

Content sharing – authors can make their contents available for other faculty to aggregate into their courseware. This can be done without duplication, simply by ‘pointing’ to the contents with the suitable set of metadata elements.

* Interactivity – contents that are managed by digital libraries can be interactive and based on multimedia. Students can listen to soundtracks, view animated images, solve exercises and have them checked online, write and send comments to authors and/or tutors.
* Customization – some users may require special characteristics of the contents and the system. This is true when people with special needs are involved, for example, persons who are blind or visually impaired. System interfaces and contents in digital formats can be customized to fulfill these necessities.
* Reuse – courseware can be developed with a granularity that makes it flexible to combine and support multiple syllabus. Reuse is important because developing course ware is expensive and takes time, so increasing reuse improves efficiency.

Digital libraries are suitable to hold and distribute open access materials – they can manage contents

**CHAPTER TWO**

1. **LITERATURE REVIEW**

Borrego, et al. (2007) observes that there have been many studies of users of electronic resources in the professional literature in the last few years. In a recent exhaustive review of the literature on the subject, Tenopir (2003) analyzed the results of over 200 studies of the use of electronic resources in libraries published between 1995 and 2003.

The main conclusion of this review is that electronic resources have been rapidly adopted in academic spheres, though the behavior varies according to the discipline. Heterick (2002) reports that more than 60% of faculty studied are comfortable using electronic resources. They believe that a variety of electronic resources is important to their research, and they consider electronic databases to be invaluable. In addition, 62% expect that they will become increasingly dependent on electronic resources in the future. The resources they use most often are online catalogs, full-text electronic journal databases, and abstracting and indexing databases. More than 70% of all respondents consider their library's online catalogue to be "very important" to their research. However, the importance of this resource varies significantly by field. Just over 60% of the economists consider their library's online catalogue to be "very important", while nearly 90% of humanists regarded it as such. In fact, the home library catalogue is the most important electronic resource for humanists, by a large margin. Based on their replies, it is as important to their research as personal computer.

This study tends to mastery of the use of application software such as creating a new word processor document, printing of documents, ability to use the World Wide Web (WWW), sending e-mail messages, taking part in an on-line discussion or chatting (teleconferencing), sending e-mail attachments etc. by adopting the following types of approach: 1. content-based approach (e-library as a collection of data and metadata), 2. technical-based approach (e-library as a software system), 3.service-based approach (e-library as an organization providing a set of intangible goods, i.e. bene ts), 4. user-based approach (e-library as a personal and social environment).Kim (2011, 64) identified three perspectives of e-library usage: User perspective(the users perceive online resources helpful and more likely to use the resources). Website design perspective (by making a website design that is simple, so that people are more likely to use it) and Library service quality perspective.

Notes that evaluation might be concerned with usability, looking at the efficiency, effectiveness, and satisfaction with which users can achieve specified goals. Electronic resources are widely used in universities. There is a direct relationship between computer literacy and use of electronic resources. Mostly academics are computer literate; however, they need to develop their searching skills.

*University Faculty Use of Electronic Resources: A Review of the Recent Literature (PDF Download Available)*. Available from: <https://www.researchgate.net/publication/263504327_University_Faculty_Use_of_Electronic_Resources_A_Review_of_the_Recent_Literature> [accessed Jun 3, 2017].

Ugwuanyi, F. C. (2009). Information communication technologies (ICT) literacy among academic librarians in Enugu State. International Journal of Information and Communication Technology, 6(1), 123-132

**CHAPTER THREE**

1. **RESEARCH METHODOLODY**

**3.1 PROBLEM STATEMENT**

The project work is designed to solve the various challenges encountered by students in doing their research, reduce the stress of going to library to finding materials for their thesis. And also to make it easier for the student and lecturers to upload their writing on the internet so as for other to review and comment on it.

* 1. **FRAME WORK**

The frame works used to develop this project are:

* 1. **CODE IGNITER.**

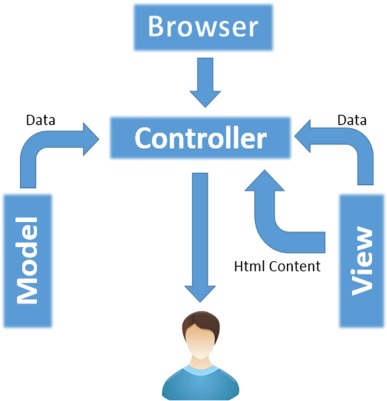
Reference: https://en.wikipedia.org/wiki/CodeIgniter

**Code Igniter**is an open-source software rapid development web framework, for use in building dynamic web sites with PHP.

Reference: https://www.tutorialspoint.com/codeigniter/codeigniter\_mvc\_framework.htm

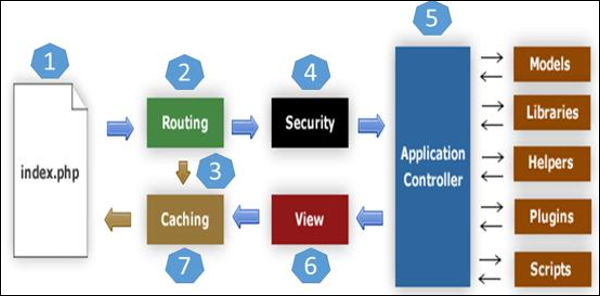
**Code Igniter - MVC Framework**

Code Igniter is based on the **Model-View-Controller (MVC) development pattern**. MVC is a software approach that separates application logic from presentation. In practice, it permits your web pages to contain minimal scripting since the presentation is separate from the PHP scripting.



* The **Model** represents your data structures. Typically, your model classes will contain functions that help you retrieve, insert and update information in your database.
* The **View** is information that is being presented to a user. A View will normally be a web page, but in Code Igniter, a view can also be a page fragment like a header or footer. It can also be an RSS page, or any other type of “page”.
* The **Controller** serves as an intermediary between the Model, the View, and any other resources needed to process the HTTP request and generate a web page.

Reference: https://www.tutorialspoint.com/codeigniter/codeigniter\_application\_architecture.htm

The architecture of Code Igniter application is shown below.

* As shown in the figure, whenever a request comes to Code Igniter, it will first go to **index.php** page.
* In the second step, **Routing** will decide whether to pass this request to step-3 for caching or to pass this request to step-4 for security check.
* If the requested page is already in **Caching**, then **Routing** will pass the request to step-3 and the response will go back to the user.
* If the requested page does not exist in **Caching**, then **Routing** will pass the requested page to step-4 for **Security** checks.
* Before passing the request to **Application Controller**, the **Security** of the submitted data is checked. After the **Security** check, the **Application Controller** loadsnecessary **Models, Libraries,Helpers, Plug-in** and **Scripts** and pass it on to **View**.
* The **View** will render the page with available data and pass it on for **Caching**. As the requested page was not cached before so this time it will be cached in **Caching**, to process this page quickly for future requests.

**Directory Structure**

The image given below shows the directory structure of the Code Igniter.

Code Igniter directory structure is divided into 2 folders −

* Application
* System

**Application**

As the name indicates the Application folder contains all the code of your application that you are building. This is the folder where you will develop your project. The Application folder contains several other folders, which are explained below −

* **Cache** − This folder contains all the cached pages of your application. These cached pages will increase the overall speed of accessing the pages.
* **Config** − This folder contains various files to configure the application. With the help of **config.php** file, user can configure the application. Using**database.php** file, user can configure the database of the application.
* **Controllers** − This folder holds the controllers of your application. It is the basic part of your application.
* **Core** − This folder will contain base class of your application.
* **Helpers** − In this folder, you can put helper class of your application.
* **Hooks** − The files in this folder provide a means to tap into and modify the inner workings of the framework without hacking the core files.
* **Language** − This folder contains language related files.
* **Libraries** − This folder contains files of the libraries developed for your application.
* **Logs** − This folder contains files related to the log of the system.
* **Models** − The database login will be placed in this folder.
* **Third\_party** − In this folder, you can place any plugins, which will be used for your application.
* **Views** − Application’s HTML files will be placed in this folder.

**System**

This folder contains Code Igniter core codes, libraries, helpers and other files, which help make the coding easy. These libraries and helpers are loaded and used in web app development.

This folder contains all the Code Igniter code of consequence, organized into various folders −

* **Core** − This folder contains Code Igniter’s core class. Do not modify anything here. All of your work will take place in the application folder. Even if your intent is to extend the Code Igniter core, you have to do it with hooks, and hooks live in the application folder.
* **Database** − The database folder contains core database drivers and other database utilities.
* **Fonts** − The fonts folder contains font related information and utilities.
* **Helpers** − The helper’s folder contains standard CodeIgniter helpers (such as date, cookie, and URL helpers).
* **Language** − The language folder contains language files. You can ignore it for now.
* **Libraries** − The libraries folder contains standard CodeIgniter libraries (to help you with e-mail, calendars, file uploads, and more). You can create your own libraries or extend (and even replace) standard ones, but those will be saved in the **application/libraries** directory to keep them separate from the standard Code Igniter libraries saved in this particular folder.
  1. **JQUERY**

Reference: https://en.wikipedia.org/wiki/JQuery

**JQuery** is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. It is free, open-source software using the permissive MIT license. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin. JQuery is a lightweight, "write less, do more", JavaScript library. The purpose of jQuery is to make it much easier to use JavaScript on your website. jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.

JQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation jquery’s syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, the meable widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and Web applications.

The set of jQuery core features—DOM element selections, traversal and manipulation—enabled by its *selector engine* (named "Sizzle" from v1.3), created a new "programming style", fusing algorithms and DOM data structures. This style influenced the architecture of other JavaScript frameworks like YUI v3 and Dojo, later stimulating the creation of the standard *Selectors API*.

Microsoft and Nokia bundle jQuery on their platforms. Microsoft includes it with Visual Studio for use within Microsoft's ASP.NET AJAX and ASP.NET MVC frameworks while Nokia has integrated it into the Web Run-Time widget development platform.

Reference: https://www.w3schools.com/jquery/jquery\_intro.asp

.The jQuery library contains the following features:

* HTML/DOM manipulation
* CSS manipulation
* HTML event methods
* Effects and animations
* AJAX
* Utilities

Tip: In addition, jQuery has plug-in for almost any task out there.

Reference: https://www.digitalocean.com/community/tutorials/an-introduction-to-jquery

**Introduction**

HTML, CSS, and JavaScript are the three fundamental languages of the web. We structure our websites with HTML, style them with CSS, and add interactive functionality with JavaScript. In fact, most animations and any action that happens as a result of a user clicking, hovering, or scrolling are constructed with JavaScript.

jQuery is the "Write Less, Do More" JavaScript library. It is not a programming language, but rather a tool used to make writing common JavaScript tasks more concise. jQuery has the added benefit of being cross-browser compatible, meaning you can be certain the output of your code will render as intended in any modern browser.

By comparing a simple "Hello, World!" program in both JavaScript and jQuery, we can see the difference of how they're both written.

JavaScript

document.getElementById("demo").innerHTML = "Hello, World!";

jQuery

$("#demo").html ("Hello, World!");

This short example demonstrates how jQuery can achieve the same end result as plain JavaScript in a succinct manner.

**Using jQuery**

At its core, jQuery is used to connect with HTML elements in the browser via the DOM.

The **Document Object Model** (DOM) is the method by which JavaScript (and jQuery) interact with the HTML in a browser. To view exactly what the DOM is, in your web browser, right click on the current web page select **Inspect**. This will open up Developer Tools. The HTML code you see here is the DOM.

Each HTML element is considered a **node** in the DOM - an object that JavaScript can touch. These objects are arranged in a tree structure, with <html> being closer to the root, and each nested element being a branch further along the tree. JavaScript can add, remove, and change any of these elements.

If you right click on the site again and click **View Page Source**, you will see the raw HTML output of the website. It's easy at first to confuse the DOM with the HTML source, but they're different - the page source is exactly what is written in the HTML file. It is static and will not change, and will not be affected by JavaScript. The DOM is dynamic, and can change.

The outermost layer of the DOM, the layer that wraps the entire <html> node, is the **document** object. To begin manipulating the page with jQuery, we need to ensure the document is "ready" first.

Create the file scripts.js in your js/ directory, and type the following code:

js/scripts.js

$(document).ready(function() {

// all custom jQuery will go here

});

All jQuery code you write will be wrapped in the above code. jQuery will detect this state of readiness so that code included inside this function will only run once the DOM is ready for JavaScript code to execute. Even if in some cases JavaScript won't be loaded until elements are rendered, including this block is considered to be best practice.

In the introduction of this article, you saw a simple "Hello, World!" script. To initiate this script and print text to the browser with jQuery, first we'll create an empty block-level paragraph element with the ID demo applied to it.

index.html

...

<body>

<p id="demo"></p>

...

jQuery is called with and represented by the dollar sign ($). We access the DOM with jQuery using mostly CSS syntax, and apply an action with a method. A basic jQuery example follows this format.

$("selector").method();

Since an ID is represented by a hash symbol (#) in CSS, we will access the demo ID with the selector #demo. html() is a method that changes the HTML within an element.

We're now going to put our custom "Hello, World!" program inside the jQuery ready() wrapper. Add this line to your scripts.js file within the existing function:

js/scripts.js

$(document).ready(function() {

$("#demo").html("Hello, World!");

});

Once you've saved the file, you can open your index.html file in your browser. If everything worked properly, you will see the output Hello, World!.

If you were confused by the DOM before, you can see it in action now. Right click on the "Hello, World!" text on the page and choose **Inspect Element**. The DOM will now display <p id="demo">Hello, World!</p>. If you **View Page Source**, you will only see <p id="demo"></p>, the raw HTML we wrote.

https://www.w3schools.com/jquery/jquery\_syntax.asp

**jQuery Syntax**

The jQuery syntax is tailor-made for **selecting** HTML elements and performing some **action** on the element(s).

Basic syntax is: $(*selector*).*action*()

* A $ sign to define/access jQuery
* A (*selector*) to "query (or find)" HTML elements
* A jQuery *action*() to be performed on the element(s)

Examples:

$(this).hide() - hides the current element.

$("p").hide() - hides all <p> elements.

$(".test").hide() - hides all elements with class="test".

$("#test").hide() - hides the element with id="test".

Reference: <https://www.w3schools.com/jquery/jquery_selectors.asp>

jQuery selectors are one of the most important parts of the jQuery library.

**jQuery Selectors**

jQuery selectors allow you to select and manipulate HTML element(s).

jQuery selectors are used to "find" (or select) HTML elements based on their name, id, classes, types, attributes, values of attributes and much more. It's based on the existing CSS Selectors, and in addition, it has some own custom selectors.

All selectors in jQuery start with the dollar sign and parentheses: $().

**The element Selector**

The jQuery element selector selects elements based on the element name.

You can select all <p> elements on a page like this:

"color:black">

$("color:brown">"p")

**The #id Selector**

The jQuery #id selector uses the id attribute of an HTML tag to find the specific element.

An id should be unique within a page, so you should use the #id selector when you want to find a single, unique element.

To find an element with a specific id, write a hash character, followed by the id of the HTML element:

"color:black">

$("color:brown">"#test")

**Example**

When a user clicks on a button, the element with id="test" will be hidden:

Example

"color:black"> $(document)."color:black">ready("color:mediumblue">function(){  
"color:red">     $("color:brown">"button")."color:black">click("color:mediumblue">function(){  
"color:red">    "color:red">     $("color:brown">"#test")."color:black">hide();  
"color:red">     });  
"color:red"> });"color:red">

**The .class Selector**

The jQuery class selector finds elements with a specific class.

To find elements with a specific class, write a period character, followed by the name of the class:

"color:black">

$("color:brown">".test")

**Example**

When a user clicks on a button, the elements with class="test" will be hidden:

Example

"color:black"> $(document)."color:black">ready("color:mediumblue">function(){  
"color:red">     $("color:brown">"button")."color:black">click("color:mediumblue">function(){  
"color:red">    "color:red">     $("color:brown">".test")."color:black">hide();  
"color:red">     });  
"color:red"> });"color:red">

More Examples of jQuery Selectors

|  |  |
| --- | --- |
| **Syntax** | **Description** |
| $("\*") | Selects all elements |
| $(this) | Selects the current HTML element |
| $("p.intro") | Selects all <p> elements with class="intro" |
| $("p:first") | Selects the first <p> element |
| $("ulli:first") | Selects the first <li> element of the first <ul> |
| $("ulli:first-child") | Selects the first <li> element of every <ul> |
| $("[href]") | Selects all elements with an href attribute |
| $("a[target='\_blank']") | Selects all <a> elements with a target attribute value equal to "\_blank" |
| $("a[target!='\_blank']") | Selects all <a> elements with a target attribute value NOT equal to "\_blank" |
| $(":button") | Selects all <button> elements and <input> elements of type="button" |
| $("tr:even") | Selects all even <tr> elements |
| $("tr:odd") | Selects all odd <tr> elements |

**Functions In a Separate File**

If your website contains a lot of pages, and you want your jQuery functions to be easy to maintain, you can put your jQuery functions in a separate .js file.

When we demonstrate jQuery in this tutorial, the functions are added directly into the <head> section. However, sometimes it is preferable to place them in a separate file, like this (use the src attribute to refer to the .js file):

Example

<head<span style="color:mediumblue"></span>  
<script<span style="color:red"> src="color:mediumblue">="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"</span="color:mediumblue">>  
</script<span style="color:mediumblue"></span>  
<script<span style="color:red"> src="color:mediumblue">="my\_jquery\_functions.js"</span="color:mediumblue">></script<span style="color:mediumblue"></span>  
</head<span style="color:mediumblue"></span>

* 1. **BOOTSTRAP (FRONT-END FRAMEWORK)**

Reference: https://en.wikipedia.org/wiki/Bootstrap\_(front-end\_framework)

**Bootstrap** is a free and open-source front-end web framework for designing websites and web applications. It contains HTML- andCSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optionalJavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

Bootstrap is the second most-starred project on GitHub, with more than 107,000 stars and 48,000 forks.

Features

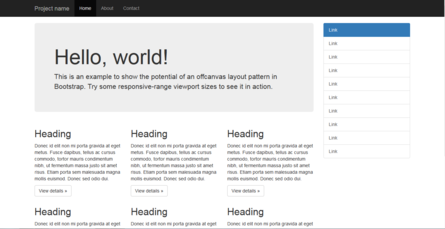
Bootstrap 3 supports the latest versions of the Google Chrome, Firefox, Internet Explorer, Opera, and Safari (except on Windows). It additionally supports back to IE8 and the latest Firefox Extended Support Release (ESR).

Since 2.0, Bootstrap supports responsive web design. This means the layout of web pages adjusts dynamically, taking into account the characteristics of the device used (desktop, tablet, mobile phone).

Starting with version 3.0, Bootstrap adopted a mobile-first design philosophy, emphasizing responsive design by default.

The version 4.0 alpha release added Sass and flexbox support.

**STRUCTURE AND FUNCTION**

Example of a webpage using Bootstrap framework rendered in Mozilla Firefox.

Bootstrap is modular and consists of a series of Less stylesheets that implement the various components of the toolkit. These stylesheets are generally compiled into a bundle and included in web pages, but individual components can be included or removed. Bootstrap provides a number of configuration variables that control things such as color and padding of various components.

Since Bootstrap 2, the Bootstrap documentation has included a customization wizard which generates a customized version of Bootstrap based on the requested components and various settings.

As of Bootstrap 4, Sass is used instead of Less for the stylesheets.

Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code.

Grid system and responsive design comes standard with an 1170 pixel wide grid layout. Alternatively, the developer can use a variable-width layout. For both cases, the toolkit has four variations to make use of different resolutions and types of devices: mobile phones, portrait and landscape, tablets and PCs with low and high resolution. Each variation adjusts the width of the columns.

**Style sheets**

Bootstrap provides a set of style sheets that provide basic style definitions for all key HTML components. These provide a uniform, modern appearance for formatting text, tables and form elements.

**Re-usable components**

In addition to the regular HTML elements, Bootstrap contains other commonly used interface elements. The components are implemented as CSS classes, which must be applied to certain HTML elements in a page.

**JavaScript components**

Bootstrap comes with several JavaScript components in the form of jQuery plug-in. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields. In version 2.0, the following JavaScript plug-in are supported: Modal, Dropdown, Scroll spy, Tab, Tooltip, Popover, Alert, Button, Collapse, Carousel and Type ahead.