Web Development Code

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="style.css">

<link href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet">

<title> Northampton Central </title>

<style>

@import url('https://fonts.googleapis.com/css?family=ZCOOL+QingKe+HuangYou');

</style>

<meta name= "viewport" content="width=device-width, inital scale=1">

</head>

<body>

<h1>

<div class="img">

<img src="NC Logo.png" >

</div>

<br>

<br>

<br>

<br>

<br>

<br>

<br>

<div class= "title">

Northampton Central

</div>

</h1>

<div class='nav wrapper2'>

<nav>

<ul>

<li class= "active"><a href= "Homepage.html">Home</a></li>

<li><a href= "Glossary.html">Glossary</a></li>

<li><a href= "Jobs.html">Jobs</a></li>

<li><a href="Tutorial.html">Tutorial</a></li>

<li><a href="Reviews.html">Reviews</a></li>

</ul>

</nav>

</div>

<div class= "wrapper">

<main class= "content"><p>Web development is based on creating and maintain website. It has different aspect such as web design, web programming and many more. Most people get confused on the term “web developer” and “web designer”. If we look closing to these two terms, web designer is often focused on designs website interfaces by using HTML and CSS. However, web developer is involved in designing a website and writing a web scripts in languages such as PHP and ASP. Nevertheless, web developer does more than web designer as a web developer maintain and update a database throughout the website. He can change the information that you looked previous and update you with the breaking news such as news around the world and depends on the website. </p>

<br>

<p> Web development have different types of web content creation. If we start giving an example on what web development can do but I will list few that can think in my head. A web developer can hand coding web pages by text editor, program such as visual code and Dreamweaver. However, Dreamweaver is no longer to be use as it is not a professional program. Visual studio is the most recently program language that are using in the industry as most company find it easy to edit and update the information on the webpages. And then anyone can use to create and edit their own website by using a web-based interface.</p>

<br>

<p>Do you know that there is different method of creating websites? it is often a trade-off between simplicity and customisation. Nowadays, businesses do not use content management systems, however they dedicated a web developer to maintain their websites. Many industries or organisation are often following the easy way by using software called WordPress and using a basic website template or editing tools. Therefore, they reckon by choosing an easy way, it will save them money and they will enable to manipulate and updating their own website, without consume a web developer or designer. Contrasted, a web developer will make their websites to function well and without any problem in the future.</p>

<br>

<div class="WebPic">

<img src="WebDevCode.jpg">

</div>

</main>

</div>

<footer class= "footer">

<div class= "copyright">

Web Development Site&copy; 2019

</div>

</footer>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="style.css">

<title> Northampton Central </title>

<style>

@import url('https://fonts.googleapis.com/css?family=ZCOOL+QingKe+HuangYou');

</style>

<meta name= "viewport" content="width=device-width, inital scale=1">

</head>

<body>

<h1>

<div class="img">

<img src="NC Logo.png" >

</div>

<br>

<br>

<br>

<br>

<br>

<br>

<br>

<div class= "title">

Northampton Central

</div>

</h1>

<div class='nav wrapper2'>

<nav>

<ul>

<li><a href= "Homepage.html">Home</a></li>

<li class= "active"><a href= "Glossary.html">Glossary</a></li>

<li><a href= "Jobs.html">Jobs</a></li>

<li><a href="Tutorial.html">Tutorial</a></li>

<li><a href="Reviews.html">Reviews</a></li>

</ul>

</nav>

</div>

<div class= "wrapper">

<sidebar1 class= "sidebar1">

<div class = "glossarypic">

<img src= "glossary.png">

</div>

</sidebar1>

<main class= "content"><b>Glossary</b>

<br>

<br>

<p>Image & multimedia

HTML supports various multimedia resources such as images, audio, and video.</p>

<br>

<p><b>"Area"</b> - The HTML "area" element defines a hot-spot region on an image, and optionally associates it with a hypertext link. This element is used only within a "map" element.</p>

<br>

<p><b>"Audio"</b> - The HTML "audio" element is used to embed sound content in documents. It may contain one or more audio sources, represented using the src attribute or the "source" element: the browser will choose the most suitable one. It can also be the destination for streamed media, using a MediaStream.</p>

<br>

<p><b>"Img"</b> - The HTML "img" element embeds an image into the document. It is a replaced element.</p>

<br>

<p><b>"Map"</b> - The HTML "map" element is used with "area" elements to define an image map (a clickable link area).</p>

<br>

<p><b>"Track"</b> - The HTML "track" element is used as a child of the media elements "audio" and "video". It lets you specify timed text tracks (or time-based data), for example to automatically handle subtitles. The tracks are formatted in WebVTT format (.vtt files) — Web Video Text Tracks or Timed Text Markup Language (TTML).</p>

<br>

<p><b>"Video"</b> - The HTML Video element ("video") embeds a media player which supports video playback into the document.</p>

<br>

<p><b>Embedded content</b>

<br> <p>In addition to regular multimedia content, HTML can include a variety of other content, even if it's not always easy to interact with.</p>

<br>

<p><b>"Applet"</b> - The obsolete HTML Applet Element ("applet") embeds a Java applet into the document; this element has been deprecated in favor of "object".</p>

<br>

<p><b>"Embed"</b> - The HTML "embed" element embeds external content at the specified point in the document. This content is provided by an external application or other source of interactive content such as a browser plug-in.</p>

<br>

<p><b>"IFrame"</b> - The HTML Inline Frame element ("iframe") represents a nested browsing context, effectively embedding another HTML page into the current page.</p>

<br>

<p><b>"Noembed"</b> - The "noembed" element is an obsolete, non-standard way to provide alternative, or "fallback", content for browsers that do not support the "embed" element or do not support the type of embedded content an author wishes to use.</p>

<br>

<p><b>"Object"</b> - The HTML "object" element represents an external resource, which can be treated as an image, a nested browsing context, or a resource to be handled by a plugin.</p>

<br>

<p><b>"Param"</b> - The HTML "param" element defines parameters for an "object" element.</p>

<br>

<p><b>"Picture"</b> - The HTML "picture" element contains zero or more "source" elements and one "img" element to provide versions of an image for different display/device scenarios.</p>

<br>

<p><b>"Source"</b> - The HTML "source" element specifies multiple media resources for the "picture", the "audio" element, or the "video" element. It is an empty element. It is commonly used to serve the same media content in multiple formats supported by different browsers.</p>

<br>

<p><b>Scripting</b></p>

<br>

In order to create dynamic content and Web applications, HTML supports the use of scripting languages, most prominently JavaScript. Certain elements support this capability.

<br>

<p><b>"Canvas"</b> - Use the HTML "canvas" element with either the canvas scripting API or the WebGL API to draw graphics and animations.</p>

<br>

<p><b>"Noscript"</b> - The HTML "noscript" element defines a section of HTML to be inserted if a script type on the page is unsupported or if scripting is currently turned off in the browser.</p>

<br>

<p><b>"Script"</b>- The HTML "script" element is used to embed or reference executable code; this is typically used to embed or refer to JavaScript code.</p>

<br>

<p><b>Demarcating edits</b></p>

<br>

These elements let you provide indications that specific parts of the text have been altered.

<br>

<p><b>"Del"</b> - The HTML "del" element represents a range of text that has been deleted from a document.</p>

<br>

<p><b>"Ins"</b> - The HTML "ins" element represents a range of text that has been added to a document.</p>

<br>

<p><b>Table Content</b></p>

<br>

The elements here are used to create and handle tabular data.

<br>

<p><b>"Caption"</b> - The HTML Table Caption element ("caption") specifies the caption (or title) of a table, and if used is always the first child of a "table".</p>

<br>

<p><b>"Col"</b> - The HTML "col" element defines a column within a table and is used for defining common semantics on all common cells. It is generally found within a "colgroup" element.</p>

<br>

<p><b>"Colgroup"</b> - The HTML "colgroup" element defines a group of columns within a table.</p>

<br>

<p><b>"Table"</b> - The HTML "table" element represents tabular data — that is, information presented in a two-dimensional table comprised of rows and columns of cells containing data.</p>

<br>

<p><b>"Tbody"</b> - The HTML Table Body element ("tbody") encapsulates a set of table row ("tr" elements), indicating that they comprise the body of the table ("table").</p>

<br>

<p><b>"Td"</b> - The HTML "td" element defines a cell of a table that contains data. It participates in the table model.</p>

<br>

<p><b>"Tfoot"</b>- The HTML "tfoot" element defines a set of rows summarizing the columns of the table.</p>

<br>

<p><b>Th"</b> - The HTML "th" element defines a cell as header of a group of table cells. The exact nature of this group is defined by the scope and headers attributes.</p>

<br>

<p><b>"Thead"</b> - The HTML "thead" element defines a set of rows defining the head of the columns of the table.</p>

<br>

<p><b>"Tr"</b>- The HTML "tr" element defines a row of cells in a table. The row's cells can then be established using a mix of "td" (data cell) and "th" (header cell) elements.The HTML "tr" element specifies that the markup contained inside the "tr" block comprises one row of a table, inside which the "th" and "td" elements create header and data cells, respectively, within the row.</p>

<br>

<p><b>Forms</b></p>

<br>

<p>HTML provides a number of elements which can be used together to create forms which the user can fill out and submit to the

Web site or application. There's a great deal of further information about this available in the HTML forms guide.</p>

<br>

<p><b>"Button"</b> - The HTML "button" element represents a clickable button, which can be used in forms or anywhere in a document that needs simple, standard button functionality.</p>

<br>

<p><b>"Datalist"</b> - The HTML "datalist" element contains a set of "option" elements that represent the values available for other controls.</p>

<br>

<p><b>"Fieldset"</b> - The HTML "fieldset" element is used to group several controls as well as labels ("label") within a web form.</p>

<br>

<p><b>"Form"</b> - The HTML "form" element represents a document section that contains interactive controls for submitting information to a web server.</p>

<br>

<p><b>"Input"</b> - The HTML "input" element is used to create interactive controls for web-based forms in order to accept data from the user; a wide variety of types of input data and control widgets are available, depending on the device and user agent.</p>

<br>

<p><b>"Label"</b> - The HTML "label" element represents a caption for an item in a user interface.</p>

<br>

<p><b>"Legend"</b> - The HTML "legend" element represents a caption for the content of its parent "fieldset".</p>

<br>

<p><b>"Meter"</b> - The HTML "meter" element represents either a scalar value within a known range or a fractional value.</p>

<br>

<p><b>"Optgroup"</b> - The HTML "optgroup" element creates a grouping of options within a "select" element.</p>

<br>

<p><b>"Option"</b> - The HTML "option" element is used to define an item contained in a "select", an "optgroup", or a "datalist" element. As such, "option" can represent menu items in popups and other lists of items in an HTML document.</p>

<br>

<p><b>"Output"</b> - The HTML Output element ("output") is a container element into which a site or app can inject the results of a calculation or the outcome of a user action.</p>

<br>

<p><b>"Progress"</b> - The HTML "progress" element displays an indicator showing the completion progress of a task, typically displayed as a progress bar.</p>

<br>

<p><b>"Select"</b> - The HTML "select" element represents a control that provides a menu of options</p>

<br>

<p><b>"Textarea"</b> - The HTML "textarea" element represents a multi-line plain-text editing control, useful when you want to allow users to enter a sizeable amount of free-form text, for example a comment on a review or feedback form.</p>

<br>

<p><b>Interactive elements</b></p>

<br>

HTML offers a selection of elements which help to create interactive user interface objects.

<br>

<p><b>"Details"</b> - The HTML Details Element ("details") creates a disclosure widget in which information is visible only when the widget is toggled into an "open" state.</p>

<br>

<p><b>"Dialog"</b> - The HTML "dialog" element represents a dialog box or other interactive component, such as an inspector or window.</p>

<br>

<p><b>"Menu"</b> - The HTML "menu" element represents a group of commands that a user can perform or activate. This includes both list menus, which might appear across the top of a screen, as well as context menus, such as those that might appear underneath a button after it has been clicked.</p>

<br>

<p><b>"Menuitem"</b> - The HTML "menuitem" element represents a command that a user is able to invoke through a popup menu. This includes context menus, as well as menus that might be attached to a menu button.</p>

<br>

<p><b>"Summary"</b> - The HTML Disclosure Summary element ("summary") element specifies a summary, caption, or legend for a "details" element's disclosure box.</p>

<br>

<p><b>Web Components</b></p>

<br>

<p>>Web Components is an HTML-related technology which makes it possible to, essentially, create and use custom elements as if it were regular HTML. In addition, you can create custom versions of standard HTML elements.</p>

<br>

<p><b>"Content"</b> - The HTML "content" element - an obsolete part of the Web Components suite of technologies - was used inside of Shadow DOM as an insertion point, and wasn't meant to be used in ordinary HTML.</p>

<br>

<p><b>"Element"</b> - The obsolete HTML "element" element was part of the Web Components specification; it was intended to be used to define new custom DOM elements.</p>

<br>

<p><b>"Shadow"</b> - The HTML "shadow" element - an obsolete part of the Web Components technology suite - was intended to be used as a shadow DOM insertion point.</p>

<br>

<p><b>"Slot"</b> - The HTML "slot" element - part of the Web Components technology suite - is a placeholder inside a web component that you can fill with your own markup, which lets you create separate DOM trees and present them together.</p>

<br>

<p><b>"Template"</b> - The HTML Content Template ("template") element is a mechanism for holding client-side content that is not to be rendered when a page is loaded but may subsequently be instantiated during runtime using JavaScript.</p>

<br>

</main>

</div>

<footer class= "footer">

<div class= "copyright">

Web Development Site&copy; 2019

</div>

</footer>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="style.css">

<title> Northampton Central </title>

<style>

@import url('https://fonts.googleapis.com/css?family=ZCOOL+QingKe+HuangYou');

</style>

<meta name= "viewport" content="width=device-width, inital scale=1">

</head>

<body>

<h1>

<div class="img">

<img src="NC Logo.png" >

</div>

<br>

<br>

<br>

<br>

<br>

<br>

<br>

<div class= "title">

Northampton Central

</div>

</h1>

<div class='nav wrapper2'>

<nav>

<ul>

<li><a href= "Homepage.html">Home</a></li>

<li><a href= "Glossary.html">Glossary</a></li>

<li class= "active"><a href= "Jobs.html">Jobs</a></li>

<li><a href="Tutorial.html">Tutorial</a></li>

<li><a href="Reviews.html">Reviews</a></li>

</ul>

</nav>

</div>

<div class= "wrapper">

<main class= "content">

Jobs

<br>

<br>

<br>

Job Description one

<br>

<br>

START DATES FOR THIS POSITION ARE FEBRUARY 2019 AND LATER.

<br>

At IBM we have an amazing opportunity to transform the world with cognitive technology. <br> By using the vast amounts of information available today to identify new patterns and make new discoveries, we are helping cities become smarter, hospitals transform patient care, financial institutions minimize risk, and pharmaceuticals find cures for rare diseases.<br> Join the forward-thinking teams at IBM solving some of the world’s most complex problems – there is no better place to grow your career!

Software Developers at IBM can work in a variety of areas. As such, you might be involved in any of these specialties:

<br>

<br>

<ul>

<li>

Backend, Server Systems, or Cloud-Based Development – Design and code servers, services, applications and databases that are reusable, scalable and meet critical architecture goals. Create Application Programming Interfaces (APIs) that are clean, well-documented, and easy to use. Create and configure Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) applications. Design and implement large scale systems and Service Oriented Architectures (SOA) that enable continuous delivery.

</li>

<br>

<br>

<br>

<br>

<br>

<br>

<br>

<br>

<li>

Client-Facing Assistance and Solution Development – Own the customer experience by solving complex business problems for the world's leading clients.<br> Work directly with the customer to understand and troubleshoot issues, collect data, elicit detailed requirements, then design and implement innovative solutions across multiple industries and sectors, as well as governments and agencies.

</li>

<br>

<br>

<br>

<br>

<br>

<br>

<br>

<li>

Automation Development and Testing – Enable continuous deployment and availability by designing, developing and creating automated frameworks, <br> processes and test cases based on functional and non-functional requirements. Build and manage testing environments, assisting in debugging application issues.

</li>

<br>

<br>

<br>

<br>

<br>

<br>

<li>

Firmware Development – Work with industry experts to design, develop and test fast and efficient software that directly manages and controls the hardware. Firmware may exist anywhere in a computing system from the central computing complex to management consoles, power supplies, and I/O adapters.

</li>

<br>

<br>

<br>

<br>

<br>

<br>

<li>

Front End Development – Create high-quality web and mobile apps containing rich content and user interface components by working closely with user experience designers to take wireframes and mockups from conception to implementation. Identify specific issues in the user interface, recommending and implementing solutions that influence and improve the design of products that provide a delightful user experience along with high performance, security, quality, and stability.

</li>

<br>

<br>

<br>

<br>

<br>

<br>

<br>

<br>

<li>

Full Stack Development – Work across the entire system architecture, from backend to front end, to design, develop, and support high quality and scalable products. Create everything from mockups and UI components to algorithms and data structures as you deliver a complete minimally viable product.

</li>

<br>

<br>

<br>

<br>

<br>

<br>

<br>

<br>

<p>Entry-Level Software Developers are in demand across IBM's growth areas. You'll be matched and deployed to a development team in a strategic business, based on your offered location and fit. These are office-based positions in IBM locations including:</p>

<br>

<br>

<b>Opportunities in these locations will vary based on business demand.</b>

<br>

<br>

<p>What You’ll Do:</p>

<li>

You’ll work in an Agile, collaborative environment to understand requirements, design, code and test innovative applications, and support those applications for our highly valued customers.

</li>

<br>

<br>

<br>

<br>

<br>

<li>

You’ll employ IBM’s Design Thinking and Agile methodologies to create products that provide a great user experience along with high performance, security, quality, and stability.

</li>

<br>

<br>

<br>

<br>

<br>

<br>

<li>

You’ll work with a variety of relational databases (SQL, Postgres, DB2, MongoDB), operating systems (Linux, Windows, iOS, Android), and modern UI frameworks (Backbone.js, AngularJS, React, Ember.js, Bootstrap, Node.js, and JQuery).

</li>

<br>

<br>

<br>

<br>

<br>

<br>

<p><b>Who You Are:</b></p>

<br>

<br>

<li>

You are highly motivated and have a passion for creating and supporting great products.

</li>

<br>

<br>

<br>

<br>

<li>

You thrive on collaboration, working side by side with people of all backgrounds and disciplines, and you have very strong verbal and written communication skills.

</li>

<br>

<br>

<br>

<br>

<li>

You are great at solving problems, debugging, troubleshooting, designing and implementing solutions to complex technical issues.

</li>

<br>

<br>

<br>

<li>

You have a basic understanding of software development and programming in a language such as Java, Javascript, C, C++, or Python.

</li>

<br>

<br>

<br>

<br>

<p><b>Required Technical and Professional Expertise</b></p>

<br>

<li>

Must have basic knowledge in one of the following technology areas: Java, Javascript, C, C++, or Python

</li>

<br>

<br>

<br>

<p><b>Preferred Tech and Prof Experience</b></p>

<br>

<br>

<li>

Willing to travel up to 10%

</li>

<br>

<br>

<li>

Bachelors or Masters Degree

</li>

<br>

<p><b>EO Statement</b></p>

<br>

<br>

IBM is committed to creating a diverse environment and is proud to be an equal opportunity employer. All qualified applicants will receive consideration for employment without regard to race, color, religion, gender, gender identity or expression, sexual orientation, national origin, genetics, disability, age, or veteran status. IBM is also committed to compliance with all fair employment practices regarding citizenship and immigration status.

<p><b>Job Number 2:</b></p>

<br>

<br>

<p>

System1 is looking for a creative and talented Front End Developer. Our platform interacts with tens of million of users per month, and our mission is to develop user interfaces that provide an intuitive user interaction and dramatically enhance the overall user experience. In this role, your contributions will be extremely visible and have a direct impact on a large user base. We experiment with and optimize all our user interfaces, and we actively find the most effective layouts via extensive A/B testing. Our design life-cycle is consistently driven by quantitative data analysis in order to measure and maximize user impact. You will utilize core software development skills to create user-facing applications and tools that will assist partners in optimizing their use of our innovative analytics platform.

</p>

<br>

<br>

<br>

<br>

<p><b>The Roles You Will Have</b></p>

<br>

<li>

Responsible for delivering high quality features on new and existing web applications using HTML, CSS, and JavaScript (Node.js, React, and vanilla)

</li>

<br>

<br>

<br>

<li>

Create intuitive, robust, and reusable user interfaces with JavaScript and associated frameworks

</li>

<br>

<br>

<br>

<li>

Convert UI specifications into clean and easy to maintain responsive web pages

</li>

<br>

<br>

<br>

<li>

Assist in documenting, debugging, evaluating, and unit testing projects

</li>

<br>

<br>

<br>

<li>

Participate in code reviews and application design discussions

</li>

<br>

<br>

<br>

<li>

Collaborate and communicate effectively, working in a team environment with other experienced engineers

</li>

<br>

<br>

<br>

<li>

Find solutions for cross-browser issues in both HTML/CSS and JavaScript

</li>

<br>

<br>

<br>

<li>

Implement A/B tests

</li>

<p><b>What You Will Bring:</b></p>

<br>

<br>

<br>

<li>

5+ years of professional web development experience.

</li>

<br>

<br>

<br>

<li>

Experience in HTML, CSS, and vanilla JavaScript, cross-browser compatible and mobile-friendly environments.

</li>

<br>

<br>

<br>

<li>

Understanding of CSS and experience with preprocessors (Less, Sass).

</li>

<br>

<br>

<br>

<li>

Experience with PostCSS a plus.

</li>

<br>

<br>

<br>

<li>

Excellent communication and time management skills.

</li>

<br>

<br>

<br>

<li>

Attention to detail for tasks both big and small.

</li>

<br>

<br>

<br>

<li>

Ability to manage time effectively and complete work by given deadlines.

</li>

<br>

<br>

<br>

<li>

Proactive instead of reactive: looking for ways to improve self and projects.

</li>

<br>

<br>

<br>

<li>

Ability to adapt and integrate quickly into a fast-paced environment.

</li>

<br>

<br>

<br>

<li>

Experience with GitExperience with Webpack, Gulp, Babel, and BrowserStack preferred.

</li>

<br>

<br>

<br>

<li>

Understanding of Python and Jinja a plus.

</li>

<br>

<br>

<br>

<p><b>What we have to offer:</b></p>

<br>

<br>

<li>

Free Uber/Lyft to and from work every day!

</li>

<br>

<br>

<br>

<li>

Collegial and collaborative team with highly intelligent and motivated co-workers.

</li>

<br>

<br>

<br>

<li>

Cross-team lunches and demos to foster learning.

</li>

<br>

<br>

<br>

<li>

Unlimited Paid Sick Time, Competitive PTO and Benefits package.

</li>

<br>

<br>

<br>

<li>

Daily catered meals and fully stocked kitchen.

</li>

<br>

<br>

<br>

<li>

Bi-weekly happy hour at various bars, restaurants, and venues across Los Angeles.

</li>

<br>

<br>

<br>

<li>

Bi-weekly onsite happy hour.

</li>

<br>

<br>

<br>

<li>

Catered dinner on Tuesdays and Thursdays.

</li>

<br>

<br>

<br>

<li>

Weekly fitness class with private trainer: high intensity training, yoga, beach volleyball, beach soccer, ultimate frisbee

</li>

<br>

<br>

<br>

<br>

<li>

Company parties and outings: Skyzone indoor skydiving, Medieval Times, Karaoke, etc.

</li>

<br>

<br>

<br>

</ul>

<ul>

<p><b>Links for the information for the job:</b></p>

<br>

<li><a href= "https://www.indeed.com/q-Software-Engineer-Web-Developer-jobs.html?vjk=5be94bb13c766176"></a></li>

<li><a href= "https://www.indeed.com/rc/clk?jk=84cd4de58c5d4e0d&from=vj&pos=bottom"></a></li>

</ul>

</main>

</div>

<footer class= "footer">

<div class= "copyright">

Web Development Site&copy; 2019

</div>

</footer>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="style.css">

<title> Northampton Central </title>

<style>

@import url('https://fonts.googleapis.com/css?family=ZCOOL+QingKe+HuangYou');

</style>

<meta name= "viewport" content="width=device-width, inital scale=1">

</head>

<body>

<h1>

<div class="img">

<img src="NC Logo.png" >

</div>

<br>

<br>

<br>

<br>

<br>

<br>

<br>

<div class= "title">

Northampton Central

</div>

</h1>

<div class='nav wrapper2'>

<nav>

<ul>

<li><a href= "Homepage.html">Home</a></li>

<li><a href= "Glossary.html">Glossary</a></li>

<li><a href= "Jobs.html">Jobs</a></li>

<li class= "active"><a href="Tutorial.html">Tutorial</a></li>

<li><a href="Reviews.html">Reviews</a></li>

</ul>

</nav>

</div>

<div class= "wrapper">

<main class= "content">

<p><b>Turorials</b></p>

<br>

<p>These are the link we are going to use In our website </p>

<br>

<!---This is the video link used for the video, the code is embedded from YouTube and Vimeo-->

<iframe width="250" height="250" src="https://www.youtube.com/embed/Zftx68K-1D4" frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

<br>

<iframe src="https://player.vimeo.com/video/272304775" width="250" height="250" frameborder="0" webkitallowfullscreen mozallowfullscreen allowfullscreen></iframe>

<br>

<iframe width="250" height="250" src="https://www.youtube.com/embed/W68ztp8Wzuk" frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

<br>

<iframe width="250" height="250" src="https://www.youtube.com/embed/9CqDaMXCmoA" frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

<br>

</main>

<sidebar2 class= "sidebar2">

Video 1 is embedded from YouTube, Titled - <b>Web Development in 2018 - A Practical Guide</b>

<br>

<br>

Video 2 is embedded from Vimeo, Titled - <b>8 Secret Techniques To Improve Web Development</b>

<br>

<br>

Video 3 is embedded from YouTube, Titled - <b>Modern Web & Mobile Design Techniques (Live Streamed)</b>

<br>

<br>

Video 4 is embedded from YouTube, Titled - <b>Web Development Techniques</b>

</sidebar2>

</div>

<footer class= "footer">

<div class= "copyright">

Web Development Site&copy; 2019

</div>

</footer>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="style.css">

<title> Northampton Central </title>

<style>

@import url('https://fonts.googleapis.com/css?family=ZCOOL+QingKe+HuangYou');

</style>

<meta name= "viewport" content="width=device-width, inital scale=1">

</head>

<body>

<h1>

<div class="img">

<img src="NC Logo.png" >

</div>

<br>

<br>

<br>

<br>

<br>

<br>

<br>

<div class= "title">

Northampton Central

</div>

</h1>

<div class='nav wrapper2'>

<nav>

<ul>

<li><a href= "Homepage.html">Home</a></li>

<li><a href= "Glossary.html">Glossary</a></li>

<li><a href= "Jobs.html">Jobs</a></li>

<li><a href="Tutorial.html">Tutorial</a></li>

<li class= "active"><a href="Reviews.html">Reviews</a></li>

</ul>

</nav>

</div>

<div class= "wrapper">

<main class= "content">

<p><b>Reviews (reviews.html) (written) or Podcast (audio) One or two reviews of ‘current’ web related resources.

Contained was gained from: <b><i>whichcmstochoose.com/wordpress.html</i></b></b></b></p>

<br>

<br>

<p>If WordPress had a show in Vegas it would be Celine Dion. WordPress is by far the most popular CMS platform around—having reached Bieber-like status. Referring to WordPress as a CMS is debatable though as technically it is closer to a blogging platform then CMS. Only recently has it upgraded and added a enough plugins that it can mimic a CMS.

The Open Source WordPress platform is the platform of choice for most bloggers and can be used successfully for simple website design. It does have some stiff competition though, especially with the release of Drupal 7 and with Joomla’s increasing popularity.

Each platform has its advantages and disadvantages, and users need to evaluate what is important for them in a blogging platform/CMS:

</p>

<br>

<br>

<p><b>Advantages</b></p>

<br>

<p><b>Easy to Use -</b>

<br>

While most products advertise themselves as being “easy to use” WordPress doesn’t advertise this, this is the reputation it has earned. Unlike Drupal and Joomla, WordPress doesn’t require knowledge of PHP or HTML coding. WordPress comes with rich preinstalled features that are literally point and click, installing plugins is simple, and their templates make designing easy.

WordPress is an out-of-the-box turnkey solution that just works. It comes loaded with rich features without any tweaking, coding or other bumbling around. If you are a beginner user then seriously consider using WordPress.

</p>

<br>

<br>

<p><b>Menus</b></p>

WP 3.0 has some great menu options, making it prettier and simpler than Joomla due to its custom menus that can be rearranged to include categories, pages, etc.

<br>

<p><b>Community</b></p>

Having a large community to draw on for support and tricks is important to many people and WP has one of the largest online communities around.

<p><b>Themes</b></p>

There are thousands of free themes available online to use to help you choose a design for your site. There are also premium themes available for purchase for those who want an added touch of professionalism to their site.

<p><b>Plugins</b></p>

There are almost 13,000 plugins available for the WP platform. It is through these plugins that WP gained its CMS title as plugins opened up WP to a world of possibilities. Most of these plugins are free which is cool.

<p><b>Custom Fields</b></p>

Through the use of custom fields you can turn WP into a CMS by going beyond the typical blogging activities like posting, categorizing and tagging. They have made this process easy for beginners as well with the addition of custom field plugins like ‘Custom Fields Template,’ ‘Flutter’, and ‘More Fields’.

<p><b>Pods CMS</b></p>

This plugin allows you to create and manage different content types without using custom fields.

<br>

<br>

<p><b>Disadvantages:</b></p>

<br>

<br>

<p><b>WP Scripting</b></p>

For advanced users who are familiar with more advanced techniques like scripting, WP uses its own script. And knowledge of the WP script is necessary for things like adding or removing post dates.

<br>

<br>

<p><b>Design Knowledge Required</b></p>

While WP does have plenty of nice-looking templates many people want to design a unique website as WP themes often look too similar to one another. To make unique designs in WP you need knowledge of the CSS style sheet language, and odds are that if you have this sort of designing knowledge you will want a more powerful toolset like those available in Drupal-like CMSs.

<br>

<br>

<p><b>Too Many Plugins</b></p>

WP needs a lot of plugins to be able to do the things that Joomla and Drupal can do out of the box. While the plugins are available, they can slow your page down if you install too many of them. The double-edged sword on WP plugins is that by the time your page has been slowed because of the plugins you have already put a significant number of man-hours into the project and would be reluctant to migrate to another CMS. Plan for this ahead of time by looking at how many plugins you would need to create the functionality you want.

<br>

<br>

<p><b>PHP Security</b></p>

Unfortunately the PHP security for WP has a shoddy track record and security isn’t as strong as some of the other platforms. This will only be a factor depending on how sensitive the content of your site is.

<br>

<br>

<p><b>Documentation</b></p>

While there is a large online community of users to help and support you there is little to no offline documentation. WP has been criticized for not creating a MS SDK document, especially now that it calls itself a CMS.

<br>

<br>

<p><b>Tables and Graphics</b></p>

Modifying and formatting tables and graphic images is more difficult in WP than in its competitors.

<br>

<br>

<p><b>Needs Regular Updates </b></p>

WP is constantly changing and growing and it needs regular updates. This is not a big deal unless you are looking for a set-it-and-forget-it solution, in which case this might not be the solution for you as it needs periodic updates.

<br>

<br>

<p><b>Queries</b></p>

Queries can be a difficult and convoluted process because of all the custom fields you need to use to go take WP beyond typical blogging actions. Often you need a Custom Select Query, which can be a royal pain in the butt.

<br>

<br>

<p><b>Pods CMS</b></p>

Pods CMS is both a benefit and a drawback, while they are handy the admin community does not support them, as they would prefer you use custom fields. This is not a problem in other CMSs.

<br>

<br>

<p><b>Conclusion</b></p>

WP is a great platform to begin blogging or simple website design. It is the easiest to use solution and offers a gre at toolset to beginners. For more advanced users, you may find that you need to install a throng of plugins to get the design capabilities you want, in which case a more advanced CMS might be in order for you.

<br>

<br>

<br>

<p><b>What is Weebly?</b></p>

Weebly is a web host, but one that caters to users who want the process of building a site to be very simple and non-technical. Weebly advertises its most popular service as “free hosting”. Indeed, you can sign up right now and have a free website online in less than an hour (or even fifteen minutes, if you work fast and don’t add too many features right away).

<p>With Weebly, you simply choose one of over 100 templates and begin dragging and dropping the elements you want to include. You can add text, images, slideshows, videos and just about anything you could add on any other type of website. There is no need to download any kind of software or plugins (which are used widely on WordPress sites). The entire process is easy to understand.</p>

<br>

<p>Many people start out with free websites on Weebly and then decide to upgrade. There are several more advanced plans, starting with Basic, which allows you to add your domain. The premium plans range from $4 to $25 per month and you can choose which premium features matter most to you. For example, if you want to build an online store and sell unlimited items, you could choose their eCommerce plan.</p>

<br>

<p>Of course, it’s possible to do all of the same things with a traditional web host as you can on Weebly. So does it really make sense to use this service? Let’s look at the advantages and disadvantages of Weebly compared to leading web hosts.</p>

<br>

<br>

<br>

<p><b>Weebly – Pros</b></p>

<p>Here are the leading benefits of using Weebly.</p>

<br>

<br>

<ul>

<li>

<b>Simplicity</b>

It would be difficult to create a more intuitive and simple user interface than you’ll find at Weebly. For this reason, it can be a good choice for newbies as well as anyone who shuns anything technical.

</li>

</ul>

<br>

<br>

<br>

<br>

<br>

<li>

<b>Lets You Build a High Quality Free Site</b>

<br>

If you’re looking to build a website for free, you have other options. Most free web hosting, however, is severely limited compared with what you can do on Weebly. You won’t, for example, have to put up with forced third-party ads on your free site.

</li>

<br>

<br>

<br>

<br>

<br>

<br>

<br>

<li>

<b>Good Selection of Templates</b>

<br>

This allows you to create a professional looking website and one that is relevant to your niche, whether your site is based on your personal interests or your business.

</li>

<br>

<br>

<br>

<br>

<br>

<li>

<b>Mobile Apps</b>

You can create and edit your site on your phone or tablet using Weebly’s mobile apps.

</li>

<br>

<br>

<br>

<li>

<b>Bonus</b>

Even with a free account at Weebly, you get a $100 coupon for Google AdWords.

</li>

<br>

<br>

<br>

<li>

<b>Easy to Upgrade</b>

Because Weebly offers several premium plans, you can start off at whatever level you’re comfortable with and upgrade later if you choose to.

</li>

<br>

<br>

<br>

<br>

<p><b>Weebly – Cons</b></p>

<p>There are certain disadvantages to using Weebly as opposed to a traditional web host. Some of these cons apply only to the free plan, but others apply across the board.</p>

<br>

<br>

<br>

<p><b>Your Site is Hosted on Weebly’s Domain</b></p>

<p>If you have a free website (using their free plan), you will be using Weebly’s domain and not your own. If you want to optimize your site for the search engines and build your brand, this is not ideal. You could, of course, upgrade to the Starter premium plan at $4 per month. On the other hand, you can find many web hosts that charge that price or even less.</p>

<br>

<br>

<p><b>Blog Options are Limited</b></p>

<p>While Weebly does have a blog tool, it’s far less customizable than using a blogging application such as WordPress. If you want to make changes to the layout, you will have to do some coding.</p>

<br>

<br>

<p><b>Premium Accounts Are Not Cheap</b></p>

<p>While Weebly’s Starter plan, where you can connect your domain, is not a bad deal at $4 per month, once you start upgrading beyond this you will be paying more for hosting than you would with most conventional web hosts.</p>

<br>

<p>For example, the eCommerce plan, at $25 per month, gives you lots of useful features, but you could also set up a free shopping cart with shared hosting for $3- $5 per month with a host such as Bluehost. On the other hand, you are paying for the convenience of using Weebly’s extremely user-friendly platform</p>

<br>

<br>

Content was gained from source:

<br>

<br>

<b><i>http://www.100webhosting.com/pros-and-cons-of-using-weebly-to-build-your-website/ </i></b>

</main>

</div>

<footer class= "footer">

<div class= "copyright">

Web Development Site&copy; 2019

</div>

</footer>

</body>

</html>

CSS Styling

/\*The CSS style here is for the body of the web page. It's used to style any code that is a part of the body\*/

body {

display: flex;

background-color: white;

flex-direction: column;

min-height: 100vh;

margin: 0;

line-height: 100%;

text-align: center;

font-family: 'ZCOOL QingKe HuangYou', cursive;

image-orientation: left;

font-size: 1em;

color: #080808;

}

/\* This is the styling for the logo (image) \*/

img{

min-width: 50%;

max-width: 50%;

min-height: 50%;

max-height: 50%;

float: none;

flex: auto;

margin: 0 auto;

display: block;

padding: 1em;

background-color: white;

}

\* {

margin: 0;

padding: 0;

}

/\*This is the styling for the header (h1)\*/

h1{

order: -1;

padding: 1em;

background-color: white;

justify-content: right;

font-size: 25px;

}

/\*This is the styling for the copyright symbol\*/

.copyright{

size: 50em;

}

footer a{

text-decoration: none;

}

/\* This is the styling for the logo (image) \*/

img{

min-width: 20%;

max-width: 20%;

min-height: 20%;

max-height: 20%;

float: left;

flex: auto;

margin: 0 auto;

display: block;

padding: 0%;

order: -1;

}

/\* This is to style th \*/

.nav {

order: -1;

padding: 0em;

background-color: rgb(0, 0, 0);

width: 100%;

}

/\*For the social media icons\*/

li i {

list-style: none;

margin: 10px 20px;

display: inline-block;

font-size: 15px;

padding: 1px 1px;

color: black;

border: 2px solid white;

border-radius:40%;

transition: 5%;

text-decoration: none;

}

/\*This is the styling for the title\*/

.title {

Font-size: 70px;

order: -1;

float: right;

}

/\*This is the styling for the picture on the gloassary page\*/

.glossarypic{

max-width: 300px;

min-width: 300px;

max-height: 300px;

min-height: 300px;

overflow: none;

}

li a {

list-style-type: none;

}

/\*This is the styling for the navigation\*/

nav ul {

list-style-type: none;

padding: 1em;

overflow: auto;

justify-content: right;

float: right;

}

li {

height: 20px;

float: left;

margin-right: 0px;

border-right: none;

padding: 0 20px;

}

/\*This is the styling for the WebDev pic on the homepage\*/

.WebDevPic{

max-width: 60%;

min-width: 40%;

}

/\* this is the stlying made for the "a" specified text in the navigation bar \*/

nav a{

text-decoration: none;

color: white;

display: block;

font: 15px 'oswald', 'sans-serif';

text-transform: capitalize;

-webkit-transition: all 0.5s ease;

-moz-transistion: all 0.5s ease;

-o-transition: all 0.5s ease;

transition: all 0.5s ease;

}

nav a:hover{

background-color: #555;

color: #661111

}

.active{

text-transform: Uppercase;

}

/\* flex styles \*/

@media screen and (min-width: 768px) {

nav ul{

display: flex;

}

/\*nav li {\*/

}

.footer {

padding: 1em;

background-color: rgb(0, 0, 0);

text-align: center;

color: white;

width: 97.75%;

}

.wrapper {

display: flex;

flex-direction: column;

flex: 2 0 auto;

Height: 30em;

max-width: 100%;

}

.main {

padding: 1em;

background: rgb(187,179,135);

text-align: center;

overflow: auto;

}

.content {

padding: 1em;

background-color: rgb(146, 139, 121);

text-align: left;

width: 95%;

justify-content: left;

overflow: auto;

margin-right: 2px;

border-right: 2px black;

}

.sidebar1 {

padding: 1em;

text-align: left;

background-color: rgb(146, 139, 121);

overflow: auto;

margin-right: 2px;

border-right: 2px black;

}

.sidebar2 {

padding: 1em;

text-align: right;

background-color: rgb(146, 139, 121);

overflow: auto;

}

@media (max-width: 30em) {

.main {

flex: 1 0 auto;

}

}

@media (min-width:30em){

.nav {

display: block;

}

.wrapper {

flex-direction: row;

}

.content{

flex: 1 1 auto;

}

.sidebar1{

width: 5%;

}

.sidebar2{

width: 60%;

}

}

/\*This is the styling for the media queries\*/

@media (min-width: 30em) {

.content {background-color: rgb(146, 139, 121);}

@media (min-width: 31em) and (max-width: 50em) {

background-color: rgb(146, 139, 121);

@media (min-width: 51em){

.content {background-color: rgb(146, 139, 121);}

@media (min-width: 30em) {

.sidebar1 {background-color: rgb(146, 139, 121);}

@media (min-width: 31em) and (max-width: 50em) {

background-color: rgb(146, 139, 121);

@media (min-width: 51em){

.sidebar1 {background-color: rgb(146, 139, 121);}

@media (min-width: 30em) {

.sidebar2 {background-color: rgb(146, 139, 121);}

@media (min-width: 31em) and (max-width: 50em) {

background-color: rgb(146, 139, 121);

@media (min-width: 51em){

.sidebar2 {background-color: rgb(146, 139, 121);}

@media (min-width: 30em) {

.nav {background-color: rgb(146, 139, 121);}

@media (min-width: 31em) and (max-width: 50em) {

background-color: rgb(146, 139, 121);

@media (min-width: 51em){

.nav {background-color: rgb(146, 139, 121);}

@media (min-width: 30em) {

.footer {background-color: rgb(146, 139, 121);}

@media (min-width: 31em) and (max-width: 50em) {

background-color: rgb(146, 139, 121);

@media (min-width: 51em){

.footer {background-color: rgb(146, 139, 121);}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}