**Elective : RESPONSIVE WEB DEVELOPMENT**

**UNIT - I**

**What Is Responsive Design?**

* The term responsive design is derived from the way in which the browser responds to its environment.
* Responsive design is an approach to how a web site is developed that aims to provide users of a web site with a good experience regardless of the browser, device, or screen size used.
* Web sites designed with a responsive design approach adapt their layout by using fluid grids, fluid content (e.g., images, videos, and text), and CSS3 media queries.
* Responsive design moves away from using fixed units like pixels and more toward relative units like percentages. This means that the widths of the different sections of a site are designed to be a percentage of the viewport.
* Ethn Marcotte first coined the term responsive web design in his article for A List Apart, in which he compares the web to architecture. He made a key point about how we should treat the increasing number of web devices as facets of the same experience.

**Responsive Design vs. Device-Specific Experiences:**

Mobile is a huge growth area and you might be wondering why we don’t just build device-specific experiences tailored to the platform we are targeting.

* Responsive development allows you to define the user experience that mobile devices receive, and as such, content can be adapted through hiding and showing content that is not specific to the platform.
* One of the key benefits of responsive web design is simplicity. There is no need for a separate mobile presence, as responsive allows use of the same URL and the same code base.
* With one code base, testing becomes simpler, especially useful if you work in a workplace that uses test-driven development, as two code bases could potentially lead to many more unit tests being needed.
* Part of this simplicity is that with responsive design you only have to manage one lot of content instead of essentially managing the same content on multiple sites.
* This will, of course, speed things up, resulting in both saved time and money. For the majority of web sites it is important to rank well on search engines, and Google offers guidance on how they would prefer you to build your sites.
* As part of their guidance, Google recommends responsive development with the following notes:

1. A single URL for content makes it easier for your users to interact with and share the content.

2. A single URL for content helps Google’s algorithms index your site.

3. No redirection or server side device detection is needed for users to get to the deviceoptimized view, which reduces loading time.

4. Googlebot user agents have to crawl your pages once, as opposed to crawling multiple times with different user agents, to retrieve your content.

**Getting Started with HTML5:**

Responsive design is built on top of the new technologies brought along with HTML5 and CSS3.HTML5 is the draft specification for the newest version of the HTML language specification, which is agreed upon by the World Wide Web Consortium (W3C).

The HTML5 specification is part of a larger set of technologies that the W3C terms the “open web platform,” which in a nutshell means it allows us to build web sites and web applications that work anywhere.

**Benefits of using HTML5:**

* Accessibility:

HTML5 makes it a lot simpler to make your site accessible.

The new HTML5 semantic tags allow screen readers to more easily identify types of content, and this allows them to provide a better experience for the user.

HTML5 has support for ARIA (accessible rich Internet application).

* Video and Audio Support:

HTML5 has native support for both video and audio codes.

The key benefits of HTML5 video and HTML5 audio is that they deliver strong mobile device support as they work in the browser without any plug-ins like Adobe Flash.

* Smarter Storage:

HTML5 introduced DOM (document object model) Storage, which includes localStorage (which is persistent) and sessionStorage (which is only available for the duration of the session).

The benefits of DOM Storage are the data remain client-side only, so they do not affect the size of the HTTP request and it allows you to store a lot more data.

* New Interactions:

HTML5’s new JavaScript APIs enable you to add new and improved interactions. An example of this is new APIs such as drag and drop, geolocation, and history

* Canvas:

HTML5 introduces the canvas element, which is the new HTML5 tag that can be used for drawing.

* Mobile:

HTML5 introduces a lot of improvements for mobile, new APIs such as geolocation allow the site to determine the location of the user and offer location-specific data.

Looking at HTML5 Technologies:

**The doctype:**

The doctype tells the browser how it should parse your document; as such, it is an important part of the document and it should be included in the first line of your HTML document. The previous doctype not only defined the document as HTML4, it also provided a URL to the specification document.

<!DOCTYPE HTML PUBLIC “-//W3C//DTD HTML 4.01 Transitional//EN” <http://www.w3.org/TR/html4/loose.dtd>>

The new HTML5 doctype is a lot simpler, and you no longer specify the version of HTML or the URL of the specification document.

<!DOCTYPE HTML>

New Semantic HTML Tags:

1. <article>: Defines an article.

2. <aside>: Defines content alongside the main content.

3. <figure>: Defines related content, an example of use is photos or code listings.

4. <figcaption>: Defines the caption for your element.

5. <header>: Defines a header for the document or section.

6. <footer>: Defines a footer for the document or section.

7. <nav>: Defines a series of links used for navigation around the site.

8. <section>: Defines a section of content.

A simple example of how an HTML5 document may be laid out follows:

<!DOCTYPE html>

<html>

<head>

<title>title</title>

</head>

<body>

<header>

<h1>Hello World</h1>

</header>

<div class=”content”>

</div>

<footer>

</footer>

</body>

</html>

**New Meta Tags:**

Viewport Meta Tag:

The viewport meta tag content consists of a comma-separated list of key value pairs, the values that can be used are:

1. width:–The width of the viewport.

2. initial-scale: The scale of the site when it initially loads.

3. user-scalable: By default, the user can zoom the site, setting “user-scalable” to “no” disables this. This is bad for the accessability of the site so it is discouraged.

4. maximum-scale: Allows you to define a maximum level that the user can zoom the site. Although not as bad as user-scalable, this can still be harmful to accessability.

<meta name=”viewport” content=”width=1024, initial-scale=1”>

Apple Touch Icons:

Another new meta tag that has been introduced is the Apple touch icon meta tag, which allows you to define icons that will be used on iOS when the user saves a web page to the home screen.

<link rel=”apple-apple-icon” href=”apple-iconn-iphone.png”>

<link rel=”apple-apple-icon” sizes=”76\*76” href=”apple-iconn-ipad.png”>

**Web Forms:**

Input form fields have been upgraded with the HTML5 specification. Previously, we have been limited to radio, check box, and text fields, however, we now have a much larger range of input types:

1. search

2. email

3. url

4. tel

5. number

6. range

7. date

8. month

9. week

10. time

11. datetime

12. datetime-local

13. color

**Polyfilling:**

A polyfill is a bit of code that simply adds the missing functionality to the browser, which is normally achieved using JavaScript. The term is not meant to implicate older browsers as often newer browsers also need to be polyfilled with the latest features.

There are already a significant number of polyfills available for HTML5 technologies, some of the popular ones are:

1. Respond.js
2. HTML5 Shiv
3. CupCake
4. FlashCanvas

**Validating the HTML5 Page:**

To validate your HTML, visit the W3C validator (validator.w3.org) and either validate by entering the direct URL of your site or paste the HTML of your site into the text area provided.

Validation of HTML can be used:

1. As a debugging tool: The simplest bug to fix in HTML are those caused by writing invalid code. A simple validation should highlight problems with your HTML, which you can promptly fix.

2. To maintain quality of code: By ensuring code always passes the W3C validation, it maintains a level of quality in the code.

3. Ensuring ease of maintenance: Although invalid code may not break your site in the short term, unexpected bugs can crop up when you later amend that code, and validating helps minimize this.

**What’s New in CSS3:**

CSS3 is the third incarnation of CSS, which expands CSS to allow us to build deeper and richer user interfaces.

Browser Vendor Prefixes:

As the CSS3 spec is still a working draft, browser vendors often implement the new features behind vendor-specific prefixes. This means that they are able to implement features where the specification has not been completely agreed upon by all browsers. Each of the vendors has its own prefix, and the prefixes for the most popular vendors are:

1. -moz- Firefox and browsers using Mozilla’s Gecko engine

2. -webkit- Safari, Chrome, and WebKit

3. -o- Opera

4. -ms- Internet Explorer

**Buttons:**

Web sites use buttons for a variety of purposes, with common examples being call to actions, form submit buttons, and action buttons. Historically, styling buttons could be quite tricky as you would need to use images for things like gradients, none web safe fonts, and shadows. When we had different sized buttons, we would then need a different set of images. With CSS3, you are able to achieve all this simply by writing code. Not only is this simpler, it also means you can scale your buttons to different sizes without redefining the general button styles.

Eg: a single button size with a gradient.

**RGBA:**

In CSS, colors have generally been defined as hexadecimal or RGB (red, green blue). If we had wanted a translucent background, it would have required using a 1×1px 24-bit .png; however, with CSS3, we can now do this with RGBA colors. A RGBA color is an RGB color with alpha transp e this site is responsive, the browser can be resized to test the mobile view (arency applied to it. The benefit of this is that you no longer need to include an image.

Gradient Panels:

The benefit of this for a responsive site is that it will scale with the width of the block.

**Testing a Responsive Site:**

**Testing Responsive Design in the Browser:**

During the development life cycle of a web site, it is likely that the first place you turn to test your site is the same browser you would use to surf the web. With responsive design, as long as your browser supports media queries (IE9+, Chrome, Firefox, Opera, Safari), you can continue to do this, turning to the other browsers when you reach the point where you need to cross-browser test.

To get started testing your responsive site, the first step is to load the URL of a responsive site into your browser.

For example, let’s look at my blog at www.jonathanfielding.com

this site is responsive, the browser can be resized to test the mobile view

**Browser Specific Testing Features:**

**Google Chrome:**

In Chrome you are able to emulate some of the most popular devices by following these steps:

1. Open the Developer tools, there are two ways in which you can do this, the first is to right-click your page and click Inspect element. The second method is to click on the menu button found to the right of the url field and select Tools ➤ Developer Tools.

2. Click the Show console icon to the right of the Developer Tools or press the Esc key on your keyboard. This will open the Console drawer

3. With the Console drawer open, you can now select the Emulation tab, the Device line indicates it will default to Google Nexus 4.

4. Using the drop-down menu, you have the option to emulate a specific device. Select the Apple iPhone 4 and click Emulate

5. Having selected the device to emulate, you will see the viewport has now automatically narrowed, and when you move your mouse cursor over the viewport, it will turn into a circle to signify your finger.

**Firefox:**

Firefox takes a slightly different approach to this. Rather than emulating specific devices, you can enter responsive design mode, which allows you to test common viewport widths.

1. Open the developer tools. The easiest way to do this is to right-click the web page and click Inspect element.

2. Enter responsive design view; this is achieved using the icon tab on the far right.

3. Now that you are in responsive design view, you can see the site in a smaller viewport. You have the option to switch to landscape view, enable touch events, or change the viewport size to other commonly used viewport sizes. You can also resize the viewport and then save the new viewport size as a preset.

**Testing on a Device Simulator:**

Before you get started with testing on real mobile browsers, you need to determine which browsers you are going to support. This may be different for each web site you build based on your site’s target audience, so if you already have some analytics on the browsers your users are using, this is a good start. Additionally, if your web site is targeting a specific country, it is easy to find statistics on what percentage of users use each platform by looking at the StatCounter GlobalStats site (<http://gs.statcounter.com/>).

There are currently five main smartphone operating systems:

1. iOS

2. Android

3. Windows Mobile

4. BlackBerry OS

5. Firefox OS

**Simulators :**

Simulators provide a fantastic start to testing your responsive site. They are able to run on your local machine, often with the capability to access all your local files and files on the network you are on.

iOS Simulator:

If you are using a Mac you can download the Mac developer toolkit, Xcode, from the Apple App Store. Included with this is the iOS Simulator, which can be used to test responsive sites. To get started with the iOS Simulator, you need to follow the following steps:

1. Download Xcode from the Apple App Store.

2. Open Xcode.

3. Use the menu bar to navigate to Xcode ➤ Open Developer Tool ➤ iOS Simulator.

4. Open Safari by clicking the Safari icon in the dock

**Android :**

The Android emulator is available as part of the Android SDK (software development kit) on both Windows and Mac.

**Mac:**

1. Download the Android SDK from <http://developer.android.com/sdk>.

2. Extract the zip file to ~/bin/Android (~is the Unix shorthand for your user directory, so if your username is Jonathan, the full folder path would be /Users/jonathan/bin/Android).

3. Open your terminal.

4. Navigate to the SDK, which is located at ~/bin/Android/sdk/tools. You can navigate to this path using the change directory Unix command cd, so the full command you enter into your terminal would be ~/bin/Android/sdk/tools.

5. Run the SDK Manager by running the ./android command in your terminal. This will load another application (be aware this application may take a while to load and unfortunately does not give any indication of the status of it loading).

**Windows :**

1. Download Android SDK from <http://developer.android.com/sdk>.

2. Extract the zip file to C:\Program Files\Android.

3. Launch the SDK Manager.exe.

**Firefox OS Simulator :**

The Firefox OS Simulator runs as an add on for Firefox, and as of Firefox 26.0 it is really simple to install.

1. Launch Firefox and open the URL about:app-manager. You will be presented with the built-in App Manager
2. You can now click Start Simulator. Because there is no simulator installed, you will be given the option to install the simulator, so click Install Simulator to continue
3. You will now be taken to a web page to install a simulator, simply follow the instructions to install the latest version.
4. Once installed, go back to about:app-manager and refresh the page. Click Start Simulator and then select the version of the simulator you wish to start. If it still says Install Simulator, you should try to install a different version
5. After a short loading time, you will find yourself at the Firefox OS home screen, where you can select the web browser
6. Upon loading the browser, you can enter your web site URL and start testing your site.

**BlackBerry OS Simulator :**

BlackBerry Limited provides emulators for BlackBerry OS that run on both Windows and Mac, and they are really simple to install. However, you will need to also install VMware Player on Windows or VMware Fusion on Mac if you do not already have these applications, which can be found at [www.vmware.com](http://www.vmware.com).

1. Download the emulator at http://uk.blackberry.com/sites/developers/resources/ simulators.html.

2. Run the Installer.

3. Run the Virtual Machine in VMware

**Windows Phone Simulator:**

The Windows Phone Simulator is part of the Windows Phone SDK, which is currently only available for Windows 8+. Unfortunately, running the Windows Phone Simulator also has specific hardware requirements, which can be found at msdn.microsoft.com/en-us/library/windowsphone/develop/ff626524(v=vs.105).aspx.

1. Download the Windows Phone SDK from http://dev.windowsphone.com/en-us/downloadsdk. 2. Run the SDK installer (this requires 6.5GB of hard drive space).

3. Upon completion of the installation, you will be told you need a license for Visual Studio, so you can either cancel and use the trial or register for a free license at http://www.visualstudio.com/en-US/products/visual-studio-express-vs.

4. Check for available updates to the SDK.

5. Open the command line (cmd) by right-clicking and clicking Run as administrator.

6. Run the command:

"C:\Program Files (x86)\Microsoft XDE\8.0\XDE.exe" /vhd "C:\Program Files (x86)\Microsoft SDKs\Windows Phone\v8.0\Emulation\Images\Flash.vhd" /name WP8SDK

1. With the emulator now open, you should now be able to run Internet Explorer in the Simulator.
2. To make opening the simulator easier in the future, create a .bat file with the command.

**Debugging Sites on Android :**

* For remote debugging on Android, the preferable browser is mainly Chrome. Since it is pre-installed, and countless customers are already familiar with it.
* In short, Chrome is dominating the mobile browser segment. Besides, the leading browsers now offer tools to aid the test and debug a website on Android. Remote debugging web apps and sites meant for mobile devices like Android can help fix problems and improve UX on handheld devices.
* Remote debugging in Chrome permits you to connect your Chrome’s developer tools on a Computer to your device (Android device) running mobile Chrome. This one can have the Chrome developer tools’ complete access for your browser (in the device)!
* The process of debugging a website on Android isn’t simple work. Most browsers are designed to be rapid and fluid, and to accomplish this, debug functions are generally not integrated. The following section takes you through a chronological guide on remote debugging on Android with Chrome, Mozilla, and Opera.

**Open Device Labs :**

If you don’t have the budget to buy any devices of your own, you can look if there are any Open Device Labs (http://opendevicelab.com/) near where you work or live. Open Device Labs is a community movement that establishes pools of devices that anyone can go and test on. They typically have a range of different devices, which will enable you to get a good picture of how your site works across them.

**Online Solutions :**

You might be in the situation where you are not able to get your hands on devices and do not live close to an Open Device Lab. In this case there are also online solutions that will allow you to test across various devices.

* The first of these online solutions is Perfecto Mobile (www.perfectomobile.com/). Perfecto Mobile is not a simulator but instead you are remotely controlling real devices. What this means is that you get a truer likeness of what the users of your site will experience when using your site.
* A second solution is BrowserStack (www.browserstack.com), which allows you to test your web site across a wide number of different browsers. In addition to allowing you to test and debug your responsive site, BrowserStack also allows you to quickly generate screenshots of your phone across different browsers and devices. It is also worth mentioning that you do not need to make your site publically available to use BrowserStack, because it installs a browser plug-in that will proxy your local sites through to their servers securely.