Transcript: Responsive Web Design – Optimal Viewing Experience
Free Webinar on May 14, 2014

Hemant Elhence: Welcome folks, welcome to today's webinar, this is our monthly webinar series. The topic today is the responsive web design. You've all I'm sure heard of the term responsive design lately. This is a way of making your application deliver or show effectively on a variety of display devices including smartphone, tablets or desktop. What we'll do in this webinar is we'll dig a little deeper to really understand this concept of responsive web design and discuss ways to effectively and efficiently deliver it in your environments. I'll introduce the speaker in a minute but before I get to that let me just make sure we go over logistics which is similar to what we do every month in these webinars. We'll have roughly 45 minutes or so of presentation that will go through to cover this topic and we'll take some questions along the way as they need to be taken. Then I'll still reserve some time towards the end, 15 minutes or so to take general questions then.

You are welcome to send in your questions on your Q and A panel and I'll keep track of these questions and as I ... and I'll make the decision based on which questions need to be taken along the way and which ones are better off holding to the end. I'm sure as usual we'll run out of time before we run out of questions but you're welcome to keep sending your questions and we can also follow up. We would record this webinar as we are right now and make the document and the recording available at the end, in a day or two after the webinar is done today. With that let me introduce our speaker, Nikhil Deshpande is our feature speaker for this topic of responsive web design. Nikhil is a user experience strategist, he is a practitioner which is always ... we welcome practitioners to these webinar series. He's a director of GeorgiaGov interactive websites and also he teaches interaction design and social game design courses at Savannah College of Art and Design in Atlanta. With that over to you Nikhil.

Nikhil Deshpande: All right, thanks Hemant, good afternoon everyone, please bear with me as I'm recovering from a pretty bad allergies to pollen here. Responsive web design, before diving into this topic I want to have a couple of disclaimers. One of them primarily is that responsive is as we all know or probably have heard, is relatively newer. Now it's been at least 4 years that the concept has been introduced, but 4 years it's still new for the fruition and the actual bulletproof ideation and execution of this technology. Within the last 4 years this concept has come a very long way. There are a bunch of people who have contributed to this and

without their contributions obviously I wouldn't be using this in my day to day work and also be talking about this.

Before we talk about responsive I want to start out by where actually responsive came from and what was the need for responsive. So, as we all know that mobile devices are multiplying, they're taking over. Some of these stats, now they actually got back to end of 2012 but in 2013 these numbers have changed just to highlight that mobile devices are actually taking over. These are literally everyday stats, and as we can see in the last 3 or 4 years, within the developing countries and developed countries, mobile devices have been getting to be the primary source of how people access the web. Western Europe or Asia Pacific regions and they literally now account 60% of the worldwide mobile data traffic and this only shows that it's only going to go up from here.

This is a terabyte per month map and as you can see the consumption for mobile compared to 2009 and 2014 is exponential. In West Africa or in India for example, there are, as per the early 2014 stats there are 100+ million Facebook users, just alone in India, and closely about 75% of them are exclusively mobile users, which means a lot of them just have not seen Facebook on a desktop device at all. What does that really say? One of the things to consider is that, yes, mobile, at least in the US, is pretty much a norm now, right. We have almost everyone who has a mobile phone but about 52% of just the laptop users who otherwise consider themselves as mobile computing users still use a smartphone. It's not just a smartphone but it's also a smartphone that is always connected. So typically being an IOS device or an Android device.

One of the myths that is around smartphones, typically when we design for smartphones is that a user using a smartphone is always on the go, they're rushed, or they might not have a good connection compared to a Wi-Fi connection or a broadband connection. Google did a study in 2013 where they found out that most of the times that people use their smartphones are when they are home, and extensive exclusive usage of smartphone. It is true that people use smartphones obviously when they're on the go but their consistent usage of a smartphone happens when they are in a covered area, typically they're home or when they're watching television. What that means is that typically a smartphone that they're using at home, they are connected to a Wi-Fi device, so the whole broadband aspect of a smartphone and designing for smartphones is not as much of a restriction, although it is always something that we need to be considering.

Another stat is that we don't just need to rely on someone having a smartphone, we are a multi gadget generation now. We have laptops, we have tablets, we have phones and a lot of times we have more than one phone. Personally me I have my work phone, have my personal phone and then just for the sake of testing I have Android devices. So just if you take some power users, I know a lot of people who use more than 5 devices. Having said that the landscape is exponentially growing and we don't know where it's going because we're always talking about variables. Without spending too much time on talking about what the future of devices is going to be, something we know for a fact that whatever that is that we are using, it's always going to be connected, with 4G, 3G, availability not just on our devices but also in our cars. The whole aspect of mobile and the definition of mobile is changing.

Now how does that relate to responsive design? As designers, which I am, it's a challenge for me to design something that I don't know how it's going to be used. Well, it was 6 or 7 years ago, I would just assume that someone's going to be using whatever I'm designing on a desktop or a decent laptop. Today, somebody just might be in the car trying to pull up something, trying to look for something that you have to design and it's our responsibility to make sure that that particular whoever that we are serving is actually doing justice to the device that the users are using. This brings us to the concept of One Web, all right. This is a big thing in the users experience community which is one web for all things.

What this really means is all websites need to work everywhere, they need to adapt to the environment, you make sure that the content is easily available, it's readily accessible and it doesn't matter how they use it, they could be using an iPhone and an Android device or a little Netbook, a tablet, regardless of which tablet or they just could be using one of those fancy iMacs or a huge 27 or 32 inch monitors. Still whatever it is that you are serving we need to be able to cater to that particular device. So what does this mean from a technical perspective? What it means is our technology that whatever we are using to serve content needs to be able to use some sort of a feature detection, mostly on the client side to get a sense of what is the device? Now it's pretty hard to actually sense devices now that truly devices could come in any shape and sizes.

What is the one thing that we saw earlier that every device has in common is all of them have a screen and if we have a screen on every device, that is pretty much the common thread of how we can actually start at least designing and considering the variations. Responsive web design is mostly a combination of multiple factors; technology, design

ideas that come together. What it lets you do is it allows you to use a single url, use a single domain name of a site and it takes away the need to design for a specific device out of the picture. That's what used to happen. We used to be into designing for devices, for mobile devices, for desktop devices. Today really there is no concept as mobile, everything can be mobile. There are tablets that are 20 inch big. What responsive design really introduced was we need to make web design future proof because we just can't predict what's coming.

We can keep changing our web presence, so that's an answer to what we used to see something like this where you had a mobile specific site and then you also had a desktop site. A lot of people started using mobile sites on desktops because it had a very specific design and a lot of times you just tend to forward urls via email. That was a lot of mix and match but the biggest issue here was that there was a duplication of effort. Mobile sites being its own entity, it had to be maintained separate from a desktop site and again, there's a lot of content that's not actually duplicated on the mobile site, now it's a different discussion if those pictures and happy people need to be on a mobile site. Considering the fact that there is some loss of data, that is significant point to note that when something goes mobile in the past, things were actually being filtered down. We needed something that could bridge the gap.

This is where responsive came in but before we actually jump into the depths of responsive, let's also take a look at the concept of responsive. Responsive web design always was coined about 4 years ago, it's actually not a concept that is just 4 years old. We had responsive design. Essentially, what is responsive design? It's design that responds. So let's take a look at this. I don't know if any of you still remember this. This was Amazon's website when they launched, this was back in 1995. This site was responsive, believe it or not because it didn't really matter what size screen you were using but when you actually pulled the browser window, it shifted the text and it adjusted the alignment, otherwise the text just went on and on all through the screen width.

I have an example here that can feature on some of the sites even today where we still see some the old school fluid layout as we called them, where for example, some like this that stretches all the way with the width of the window. If you take this and if you start moving this in, things automatically adjust and as you can see things just at some point start wrapping around like in this table here. This is actually just an example of what it used to be. Now we didn't have to do all the extra coding to make this responsive but this is a native browser property. So, how did responsive actually go from here? The change that happened

here was that because everything was so fluid, as a designer we had to put some structure around it. It's not just something that is specific to designers, everybody...

Nikhil Deshpande: So the layout aspect of this is that we needed to put everything in a structure, put everything in an order. This is where we started as designers and developers, started using tables where we gave the browser a certain width and we inserted our content, even though the content was fluid, we set the table width to be something that was fixed. So as you can see on the left hand side here, the 960 pixel is a fixed table width and the designers and developers have to work with it. Using tables was just a bad idea, for all of you people who are aware of the sensibility concerns of using tables and also the complexity of the code. It just added way too much code, it was very complex to manage and modify pages. So again, that was just a big fix solution but that was not a future proof solution.

Thankfully to over rescue we had the introduction of cascading style sheets which helped us with way less code achieve the same layout and similar control that we had around code. Another thing that the programming again was limited was that it did give us a different way of presenting but at the same time it didn't really give us an option from the fixed layout. If you were to still do in a 960 pixel fix, if you do a 90% relative width it still would span all across the 27 inch monitors. So there was still some way of control that we needed to hold onto because when we did things fluid they just went way, way out of our control. One other thing that contributed to this fact was the monitors didn't really change, when I say the monitors, the actual screen sizes. We either went from 800 x 600 to maybe going for something that was 1024 x 768 or 1200 going forward. Still, the variation between them was not that much, which drastically changed when these tablets and when hand held devices were introduced in the market.

The other aspect of this was, how did we adapt from a design perspective? It's not just about technical aspects of being able to design or being able to display something but how does it affect our design? Clearly , every time there's a new technology, there is a certain learning curve, so the same thing that happened going from print to digital that we were just trying to emulate our print designs on digital screens, same thing happened with hand held devices. We were just trying to take over digital properties the way they look on the screen and just make them show up the way they were on our hand held devices. Going forward, a good thing that happened here, that hand held devices really got defined by two early phases, one was the iPad and the other one was the iPhone

that was launched. We actually got a truncated version of what our screen experience was, we actually had to learn how to use it, we had to pinch in, zoom in, and then just move around the screen. As much as the usability aspects were a concern out there but at least we were able to get to what was now something in your hands available.

So we were still looking for that perfect solution of what could be possible and it actually got in front of us to just one simple article, which was written by a person called Ethan Marcotte. This is the first time the concept of responsive web design, it was introduced. Ethan Marcotte put down his thoughts about, this is what I know from a technical aspect and this is how we can actually make things different with a combination of certain components. We can change the layout to fit a certain size of a device, a view port. A view port is essentially the part of the screen that displays your web properties. So being able to keep in mind that, this article really, really caught fire. Everyone was just gravitated towards this concept and it's out there today and then maybe later I can share all the resources that I'm showing you and maybe I can Tweet all those resources. This is responsive web design article on a website called A List Apart.

As you can see, this is where he demonstrated how responsive design can work. So this is the article and as you can see, if we change the screen, things change, even the illustration, the way people are standing are different from here and as you keep moving, it keeps changing. That's exactly what his idea was that as your screen changes, your website or your content needs to adapt to it and not just scale it, like the iPhone used to do. So going back to our presentations, I would absolutely recommend reading this article and if this is something that you truly find interesting, Ethan also has an entire book on responsive web design which honestly if you truly want to do a deeper dive in responsive web design, you need to read both of these. Anyway, for me to distill this concept, let's look at some of the components of responsive web design that Ethan put across in that article and in his book.

The first thing that Ethan mentioned was, look, we need a certain grid system which he mentioned as fluid grid. Grids need to change, up until now we have only known grids that create structure, give us some sort of consistency. Once we switch devices, that grid needs to adapt to that device. He introduced this concept of a fluid grid. The next thing he introduced was flexible media, which we just saw an example of, when I changed that browser width, how the illustration changed. So our media needs to adapt to what our screen size is and be seen accordingly. The next important component of what Ethan considered as a very core

component of responsive is what he called as media queries, well he didn't really name it, media queries always existed in CSS but he highlighted them, this is why we need to be using media queries.

I'll be speaking to essentially all 3 of these as we go forward. Just to give you an idea of what media queries do is that media queries are essentially the actual code within a style sheet that is looking at what the screen ... or a certain dimension change that happens. And when that change happens, these media queries point to a separate set of style sheets within your CSS that say, okay now, this screen is this wide then now we should be looking at these styles to render the content. This whole conversation is assuming you have a high level idea of how web pages work. So in your HTML you have the structure of your page, you have something that defines your title, something that defines your paragraph, but with CSS that content is actually styled, that layout is actually implemented in your cascading style sheets.

Jumping forward to highlighting what the fluid grid system does for you. Grids essentially give you structure, consistency, some sort of predictability and they define the relationship between your pages. We all have been to web pages that even though it's one site, one page doesn't look like the other. In that case, you start doubting the credibility of the website. What grids do for you, they put things in order. At some point they do create some constraints, but at the same time it allows the user to have some sort of predictability, some sort of consistency. But what Ethan did was he predicted that this grid system can be made flexible for different screen sizes. So in this case, if a screen is of this particular width, you can have this layout, but the same grid system can be tweaked a bit if the screen is of this particular width. The same thing with something happens on the phone. This by itself is a good start but we need the rest of the elements to actually act with it.

We saw that the text actually aligns itself, which is a native property of every browser, but we needed the rest of the media to actually also work with it. Now in the world of web design I should say, flexible media, flexible images, it's a little thorny issue, it's still being debated because there's a tie to performance when it comes to images because a lot of times what happens is, you have multiple ways that you can actually do this. You can choose to scale an image from something that is huge. I can, for example, I can take an image on the extreme right that we are seeing, something that fits our laptops, so an image that's 1140 x 960 pixels, so let's say that image size is 500k. Now if I use the exact same image and just scale it down to my smartphone which is probably from an image size perspective, I need about 320 pixels x 200 pixels or whatever, if it's scaled

down diagonally: First, I will probably lose part of my image details because my laptop image is horizontal, my mobile screen is vertical. Secondly, if I just scale down the same image, I'm going to be having to render a 500k image, which is meant for my laptop, maybe a broadband connection, on my mobile phone.

Now okay, I did talk about mobile phones now being connected to Wi-Fi but that is one of the use cases but there are instances where mobile phones are on the run, obviously, and that's what makes it mobile. So we don't want to be using the same image for multiple devices. This is where flexible images, the concept of creating different images for different screen sizes and changing that editorial content comes in. This is where media queries again come in. Where media queries can say, well change this particular image based on a particular size of the screen. Media queries typically deal with background images but there are many ways where now the community, the responsive community is trying to figure out, what's the best way to treat images as content or responsive? There is dissent. The entire responsive images community group, and they have a website and they keep experimenting and coming up with new solutions.

The picture element is surely something that we should be considering when we talk about responsive. There are many ways that this is today done in a very efficient way. If you're part of a conversation where there are some concerns about images being used in responsive design, a lot of times in my experience what I've seen is, a lot of people probably don't have the entire picture of what it truly takes to make something not just responsive but also optimal. Nothing like this can actually happen without using our really futuristic technology. Responsive, although it doesn't truly need HTML5 syntax, it is very, very helpful to use HTML5 from a technical perspective. There is surely an expectation from a standards viewpoint that somebody, as a front end developer or a designer should really know HTML5.

If your web property at this point is not using HTML5, you should be aware of that because going forward, HTML5, is going to be the norm, especially now that we know Adobe Flash is pretty much getting sub-setted from the production value of it. It is still probably used extensively for mock-ups but a lot of our rich media interaction that used to hinge on Flash being able to deliver it now is all moving to HTML5. If you see all the Google illustrations and their daily animations on Google's website, they're all done in HTML5. So HTML5 has some really, really good advantages. The first advantage is that it comes up with its own new defined tags. HTML5 gives you that structure for a page that in the

previous versions of HTML we had to create our own structure, we had to name our own structure. HTML5 took away that weightiness of the coding and now they gave us these tags that just make sense, like a header tag, a footer tag, a navigation tag, a video tag.

Earlier if you were to show video on your website, you would have to resort to Flash or resort to some other middle ware to render that on the client side, whereas HTML5 you just point to the video source and it just renders it. What HTML5 really brings along with this is it has a better semantic value because when HTML5 was architected, these names, these tag names to these new elements, they weren't just made up, obviously. There was a big community of these professional designers, developers, who came together and they exchanged notes. When I develop something, what is it that I use? So hey, if all of us use a header tag, why not actually have that in HTML where it just becomes a standard? So there's a lot of semantic value of using HTML for the best tag for getting the job done.

Another aspect that it actually has the better semantics put out in place where it helps us is this is a more organized code that comes with it and this is very important when it comes to responsive. Responsive because it's so code driven, you have to have a really very organized code and I'm a designer and I'm also a front end developer, and a lot of times I'm in these positions where I have to work on something that is developed by somebody else and it took me a lot of time just to understand what they're trying to do, just to interpret that code. With HTML5 because the tags are now standardized, it's much easier to read and understand. The wonderful benefit of HTML5 is accessibility, the syntax is similar because it creates such a meaningful document outline, it automatically enhances the accessibility aspect of your page especially with devices that use screen readers.

Lastly but not least, and this is something that is so near to heart for everyone is the search engine optimization. HTML5 improves your SEO because search engines love things that are categorized, things that are in order, things that give your content weight and preference. The tags and the overall semantic aspects of it really help that. Also a few things that we really need to understand with HTML5, the main issue with that is the browser support. Most of the older browsers, they don't support HTML5. When I say most of the older browsers I'm primarily talking about Internet Explorer. IE9 and 10, they do but if you look at the overall footprint of IE9 and 10, it's still significantly lower compared to the previous versions of IE. Most of the other browsers, like the Chromes and the Firefoxes, they support HTML5 from a lot of their previous versions.

There are many tools out there that can check and they can help you see if a certain aspect of your site is supported. If there is something that is not supported, there is a concept called SHIMs where we can put in place these hacks where, hey, IE8 if you can't render this, no problem, you can just pretty much emulate this particular thing that we're trying to do here. So there are ways to go around that as well. The thing that actually makes this happen is CSS3. Now CSS3 it has quite a similar history to HTML5, right, they have a lot of great features, it has a lot of things that it can do, but again going back to can browsers support it?

One of the biggest features, other than the stress on media queries and we've been talking ... we're going to be talking about media queries separate, I'm just highlighting some of the other features, is I love this whole fallback option that has in CSS3, for example you can now use any kinds of fonts in CSS3. We were restricted to font usage and then somehow there were web fonts that were being made available by Google, but you can actually choose what fonts you want and something needs to be rendered on a phone, you probably don't want a font that takes a long time to render, you probably want a font that is easier to render. See, then I can switch fonts using CSS3. This fallback where, for example like I was talking earlier where, hey, is HTML5 not being able to play video because browsers don't recognize the video tag, no problem, we can just come up with a fallback option for Flash to render it.

Talked about how we can embed fonts, that's something that CSS can do for us, they can actually use licensed web fonts. It can be local or it can be remote and all we have to do is use a font face reference in our style sheet and we can use whatever that is that we aspire to. I can tell you as a designer or a front-end developer, what a huge benefit that is just from a usability perspective. There are many other features like polyfill, like the HTML5 SHIM that I was talking about, it allows you to use HTML file elements in your code in the previous versions of IE and render them so you're not actually missing out on features, you're providing support for your older (browsers)... so the point being here is don't let the fact that older browsers don't support responsive stop you from doing responsive because again, you're not doing this for computers who are using IE6, you are using this for the phones, you are using this for the tablets. Anyway, Microsoft themselves are actually trying to get rid of IE6 and they're trying to reduce the overall footprint.

The next aspect is media queries and there is a really good resource up here, it's called media queries and I can share this again with you. This lifts out some of the really good responsive sites that use media queries. One of the examples that I would really like to show you which I feel is a

very good example is the WorldWildlife.org website. This is a website, they have done such a good job with the responsive. As you can see, I start moving this and just like you saw in Ethan's article, the images change from a four column, now they're going to go ... notice our navigation changed, earlier we had navigation, now they just kept what was absolutely premium, what was important was donate button. As you go down, this just adapts to what he screen size is and this is exactly what you need in your website because someone using a phone for the first time to look what you have on your website, you don't want to lose them because you just didn't really plan for them to be using their phone.

What media queries do is they're all about conditions. So what I mean by that is you use media query or media type or the expressions to check for a certain condition. This condition could be anything, this could be what screen size they are using, which is typically used in the media queries. It would create a certain condition that well, if this is a certain screen size, and we use screen sizes because devices tend to use fixed resolutions, so at this point, you can't really resize a web browser on your phone, at least not right not. What this uses is well, it gives us at least a good starting point to understand where the users are coming from and then accordingly point them to a certain media query. Now I don't want to get too much in the weeds of the technical aspects of it, but this is just an example of what a media query looks like.

This is the title of a media query, where here it says the media type being a screen and the maximum device with the 480 pixels, what it tells the style is that well if your screen is anything that's under 480 pixels, use this particular style sheet. Now what 480 pixel otherwise is known as is a break point. A break point is a point where if you don't adapt your layout, it breaks going forward. So this is what media queries do for you.

Hemant Elhence: Let's take one quick question on the media queries.

Nikhil Deshpande: Yes.

Hemant Elhence: Does the media queries also support other display or media attributes like say availability of a touch sense on screen aside from the size?

Nikhil Deshpande: Repeat that question, Hemant, again.

Hemant Elhence: So does media queries also support things other than size of the display device, so for example touch.

Nikhil Deshpande: Yes absolutely. Well not so much touch, but media queries can also point to a lot of other things, not just the screen. For example, you can have a media query to sense if someone is trying to print a page. So in that case what media query does is it understands that someone is trying to print the page and then it just switches to the style sheet that we actually have something that is a print friendly style sheet. There are so many media queries. Media queries again is not very ... not a CSS3 issue, it actually was back in CSS2 as well but now with CSS3 having the screen width aspect to it, actually is being used prominently.

Hemant Elhence: Okay, thanks.

Nikhil Deshpande: Sure. So moving on, from a usability perspective, if our design has to change it has to actually also consider some of the actual native aspects of it, right. In this case, navigation. Navigation is always hard, navigation is a challenge because users don't look for navigation unless they're absolutely lost. Lately we're seeing that people just love to just search things, but that also has a huge dependency on how well your search is. Navigation obviously it has to be something that is clear, usable, also affordable, well it needs to look like a navigation. What happens when navigation changes from something that's supposed to be on desktop to something like this where it goes down to a tablet down to maybe a tablet that's vertical or something all the way down to a phone? There are multiple ways that we can actually treat navigation, one is, depending on how much time and budget you want to spend, you can quickly style navigation to just stack up like this on the left is a typical desktop site, on the right, this is what navigation looks on the phone.

This is a lightweight solution obviously because what the media queries do is they just point to styles that just stack these links, that's pretty easy to do from a styling perspective. What this does is it eats up your space, right because right now you have 6 links, but if you have 10 or 15 that's not going to be a good thing. In other cases you might have to resort to some other strategies where, well, it takes a little more time to develop it but this is like a menu where unless you touch that menu button, which most of us have seen now being implemented, it doesn't actually show us what the menu is. For this we need to understand how users also use the phone. We know that about 75% of our users use their thumb now to actually browse their phones. As they've been on the device, diversity increases, honestly but the way we interact with our phone essentially is the way we interact with the web is not changed, so it's not just about understanding how to make things change and make things happen we also need to constantly keep up with the user aspect of it.

For example for this to happen, we need to understand the touch aspect of it. One of the things that we can't really use the touch assumption because now tablets that use a larger browser are also touch enabled. So if there was a media query that only sensed that, if touch is a criteria and render it this way, for a tablet, like a Sony tablet that is a 22 inch tablet, it still will be confusing because it will probably render something that looks like a phone but then it's not used like a phone. In this case, a quick look at the touch aspect of it is that we need to understand that when we use a finger as compared to a mouse or a pointer, we have to obviously consider the fact that there are going to be some errors, which otherwise we call it the fat finger.

Because Apple and Microsoft have done some extensive research and Apple recommends a target size of about 44 x 44 pixels for a touch target. Then later when they also introduced retina display they changed it to a points aspect which I think is about 326 points per inches. Microsoft also suggest that well maybe you could do this around 9 mm x 9 mm. They introduced a physical aspect of width in that instead of relying on pixels or points because screen densities vary. We see that okay first if our target should be about 9 mm wide, and maybe between 2 targets we need at least 2 mm distance so that we don't end up using more finger in the wrong place. This changes our design as well, right. So it's not just about adapting one design to the other, it truly needs to have its own way of thought.

This is where the mobile first approach comes in. Earlier when the responsive band wagon was getting filled with early adopters people were moving back from their desktop design sites to a tablet site to a mobile site. Then they saw that there were a lot of problems because they were not doing it the right way. What they did was, a lot of them redesigned their sites and are going mobile first. First design for mobile, then for tablet and then for desktop, because that's really where the users are coming from. While we are doing this, while we do think about responsive is that it takes away from the whole waterfall mindset of you need to first do mock-ups and you need to first do prototypes and then you develop them and then you deploy them, test them, this is an iterative route, so you need to concentrate and you need to develop real time.

Hemant Elhence: Nikhil before you go to the next page ...

Nikhil Deshpande: Yeah.

Hemant Elhence: In the page there where you're just showing transition to mobile first, can you give an example of an alternative design that one would do if they did not do mobile first versus if they did in the top half of the page where you say, if you did it for desktop first and then made it work on mobile versus if we did it mobile first, is there an example you can use to illustrate this difference?

Nikhil Deshpande: There is, actually let me see if I have it open. There is a website called HappyCog.com, and what they did was they literally spent one weekend in redesigning their entire site because what they had done in the past was ... and this is the new design, and what they had done in the past was they actually went back from their existing design to make it mobile and then they realized that they really needed to change strategies. You can see a really good other example would be BostonGlobe.com, where ... this is what BostonGlobe.com used to look like. It still looks just like that but unlike New York Times, now if you move this, and this is a heavy content website, notice how the columns change and as you move down again, the navigation changes and this becomes truly responsive. This is an example where they actually held onto their look and feel of what their site used to look like but they made the rest of their site responsive for mobile devices.

Hemant Elhence: The New York Times is another example.

Nikhil Deshpande: So the last I had checked New York Times, it's a very similar layout but last I had checked they had ... look they're still not responsive. This is what the users have to deal with is a scaled down version of their site compared to the Boston Globe. This is a huge usability problem. Does that answer your question Hemant?

Hemant Elhence: Yeah, good, thanks.

Nikhil Deshpande: Okay. So to make sure that you are on the right path, you really have to test it. You have to test essentially the responsive aspects of it, the responsive aspect being the viewpoint or the breakpoint testing. The section of every device differs. There are some testing tools that can help you. This is a tool that I use, this is called Screenfly, it's by his service called Quirktools and if we just put in, for example, let's say BostonGlobe.com in here, it actually gives us what this site will look like. So this is it... you can toggle, this is what the tablet, say Kindle Fire will render it for you. So yeah, this is a good place to start but I wouldn't completely rely on an online tool to do this. What you really need to do is make sure you use actual device testing.

For your analytics, look up what devices are being used to access your website and then come up with that particular device list. This is a website dmolsen, Dan Olsen, who is a huge thought leader in this area, he lists out this is like at least the devices that you need to have to check your website. Again I will be sharing these resources later with Hemant. Going forward, another thing you could actually rely on that again is a really foolproof way of doing it is Adobe's software tool, Inspect. What they do is they let you connect all these devices together and so you can just use your laptop to browse a certain site and then all the rest of the devices actually show you the exact same thing, what you are using your laptop for. So you don't have to spend ... and this is if you don't want to spend time testing 25 different devices, if you just have 1 or 2 people testing it, so this is a really good tool.

The last but not the least is make sure you optimize your website. There are tools out there that validate your HTML code, that validate your CSS code, but treat your performance and I believe Scott Gill talked about this in the past, treat your performance as a milestone in your design process because without performance, your responsive investment is nothing. This is where some of the bad responsive examples give responsive a bad rep because they don't do performance testing. Like your household budget, you really have to put a certain budget on your responsive and I'm not saying budget as far as how much you spend on your optimization but how much content can you allow for a certain page? Based on if you have mobile users that are using actual 4G or 3G then limit your page to a certain ... maybe 100K or 200K, whatever that is. For your desktop users, limit it to 500K. So fix your budget on things.

There are some really good tools. There's a website called YSlow.org, like the Y as in Yahoo, slow dot org, that helps you with giving you a performance test check for your website. Brad Frost is a huge name when it comes to performance testing, he has a book called Performance As Design. I think it's a very, very important thing to keep in mind as ... we do load testing for everything else, but especially for responsive, you need to test what the performance looks like and if it's optimized to actually deliver that responsive. So having said that, this is pretty much an overview that I could distill the whole responsive design process for you.

Hemant Elhence: Excellent, thanks Nikhil, this is good, so I think now this is time for us to take questions and audience, I already have a few questions here. I'll encourage you guys if you have any other burning questions based on the content that Nikhil just covered we'll take those, we have a few more

minutes. And Nikhil if you can advance to the next page, I can do a quick introduction to Synerzip while I'm collecting the questions.

So Synerzip, for those of you are new to us in these webinar series, we are a product development partner and emphasis is on partner, for small to mid sized technology companies, we help these companies. For each of these client companies, which are small to mid sized software companies, we help them put together a dedicated team of software professionals and really build ... deliver software following agile practices and help them not only reduce costs but also reduce the risk of software development.

We have a development center in India and we are able to provide a significant, typically 50% cost advantage and at the same time provide flexibility also so our clients can work at least through a captive operation however they feel like it. That's a quick introduction to Synerzip, you're welcome to reach out to us if we can help.

And Nikhil if you can move, this is just a glimpse of some of our clients. So let's go to the next page and take questions now. Nikhil, if you're ready let's take at least a few questions in the remaining minutes. So the first one is, we went through this responsive web design can you contrast this with the need some cases still really build in native mobile app, in some cases it's something they feel like the need to have a separate mobile app versus just having a responsive web design that allows the content to be displayed properly on a mobile device, when ... any idea, any suggestions how people should think about that?

Nikhil Deshpande: Sure, yeah. Actually it's a really good question because it's a different litmus test. There is a decision that needs to be made between, should I go responsive or should I go native? This particularly comes when you are talking about applications. A certain transaction that needs to happen. Responsive surely gives you that benefit of having one central location. So with native apps what you have to do is if you truly need a native app you have to first know why you need a native app. Do you want to use any device specific features like the camera or the overall GPS or the whole aspect of your phone? Then for whatever that app actually features, which is not offered on the responsive side of it.

That truly qualifies you to have your native app per se. If that's the criteria then surely, have a native app. Native app typically is not a solution or a replacement for a website, responsive is something that you apply towards your web presence or something that you do on the web. It could be an application, that's fine, but it's still ... if it has a web

presence then you're looking at a cost effective solution because for a native app, for example, if you were to have it, you at least want to have it for IOS or for Android or maybe for a windows phone now, if not for a Blackberry. Then you're still looking at keeping up with 3 different platforms here. With responsive, regardless of your devices, device diagnostic, you just have to keep up with 1 platform and it just renders based on whatever the device you're using. So, a little different but at the same time it's something that if you don't need any of the device specific features, then by all means responsive is your friend.

Hemant Elhence: Okay excellent. I'll take one other question and maybe we can just evolve this into a comment since we're out of time, so the question was, what is a good site or approach to start learning HTML5 and CSS3 for experienced and inexperienced web designers? So I was going to request you, Nikhil, maybe could take that question and other resources that you pointed along the way that are useful for testing or as good best practice examples of responsive design, maybe we can add a page or two in this PowerPoint before we put it up on our website.

Nikhil Deshpande: There's so many of them. What we can do is, I can actually give you some of the resources that are really good resources for maybe non technical people to just understand about this technology and also somebody who is just starting out, where they can start from.

Hemant Elhence: Excellent, sounds good. We look forward to that and thank you so much Nikhil for taking a few more questions if we had the time, but it's always a challenge for us but I really appreciate, Nikhil, you sharing this in depth perspective on responsive web design and thanks all the audience for attending. Stay tuned for next webinar in the series.

Nikhil Deshpande: Sure thanks.

Hemant Elhence: Thank you.