**Technology used**

Angular 2+, HTML, CSS, Bootstrap framework

**Time taken**

11 hours – 10 hours of development and bug fixing (over 2 days) and 1 hour of documentation (3rd day)

**Code explanation**

I have not provided any comments in the code and thought it prudent to include the explanation of the code here.

1. The top header of the application is provided with a font and a background defined in CSS.
2. The next level, which is the menus, is basically done using the bootstrap framework. I downloaded the bootstrap framework version 3.3.7, which provided me with a readymade code to build the menus, which look good in almost all kinds of browser sizes. The mobile version shows the button on the right hand side which shows menu items vertically when clicked.
3. The menus are dynamic in nature and I use an angular method to return an array containing the menu item strings, which are rendered on the browser using ngFor directive from Angular.
4. In a real life application, these menus can also be returned from a service in the form of a json object that contains the menu item string, level number and the page that it points to.
5. The next set of menus (or that is how I would like to address it) are fixed in nature. I have two divs defined to show the same menu items, one for width < 750px which aligns these menu items vertically and another one for sizes >= 750px which aligns these menus horizontally.
6. Next is just a text.
7. The next paragraph or div uses panels from bootstrap framework for the look and feel. I created an angular method which returned an array of objects containing header, main, footer and the body from the angular method. Again, I used an ngIf to show/hide the body based on whether the user clicked Close or Compare Benefits.
8. The next set is the button – Compare Benefits. I tied the click event of this button to an angular method, which manipulates the text of the button. Also, the body in the above panel is shown/ hidden accordingly.
9. Next is just some simple HTML to show the TRUSTED PARTNER div.
10. The next text – workday is divided into five equal sized columns using the bootstrap class – col-lg-XX. It arranges itself gracefully in a vertical manner in the mobile version. I liked it somehow more than the template provided to me, so stuck to it.
11. The next is the simple footer, which contains black background and a couple of vertically arranged horizontally centered text.

**HTML code** - The main file to look at for the HTML code is – HitechWorld.html (inside the app folder)

**Angular code** - The main file to look at for the Angular code is – hitechworld.component.ts (inside the app folder). This file gets converted to hitechworld.component.js by the typescript transpiler.

**Installation details for the application**

Here are the steps in installing the application for which you sent me the mock-up:

1. Please create a folder in your machine and download the files from the following location: <https://github.com/kdatt2017/HRWorld>
2. Go to nodejs.org and download the version 6.11.2 LTS onto your machine.
3. Go to command prompt in Windows and traverse to the folder that you created in Step 1 and run the following command: npm install. This installs the node\_modules folder inside this folder you created in Step 1. This is essential for the application to function as this is an Angular 2+ based application.
4. In my case, I had IISExpress installed on my machine and I tested the application using IISExpress. If you do not have it, any other web server will do. However, if you choose to install IISExpress or already have IISExpress installed, then you can run the following command to start the web server: “C:\Program Files (x86)\IIS Express\IISExpress.exe” /path:”<path\_of\_the\_folder\_on\_youir\_machine\_that\_you\_created\_in\_step\_1>” /port:<port\_number\_of\_your\_choice\_i\_gave\_12345>
5. Now, you can type [http://localhost:<port\_number>/index.html](http://localhost:%3cport_number%3e/index.html). Please note that step 4 should be followed after traversing to the folder you created in step 1. An example is: <http://localhost:12345/index.html>.

**Considerations while coding**

* The main focus of my coding was to showcase my skills in Angular 2.0+, HTML and CSS and try to match the templates provided by you as much as I could.
* I have not emphasized too much on the color provided by you as I can easily get these colors if I have the image with me.
* I have not created any image as in a real-life situation I am expecting to get the images from a graphic designer. That could have enhanced the overall look and feel of the application. There is a line of images in the footer I did not provide, because of the same reason.
* I am basically a .NET programmer and my technology stack is Angular, HTML, CSS, ASP.NET MVC and SQL Server and I will be interested in a full stack position rather than only a web designer.

**Challenges faced during coding**

* The main challenge I faced during coding was to provide the text box for the last panel (Type it here and then print Day 01). Now it looks OK, but I was not getting any idea as to how to return the text box from the angular method. The best I could come up is to hardcode the text box in HTML and show that div based on the criteria defined in the application. (In this case, I returned a text with &inputText& from the angular method)

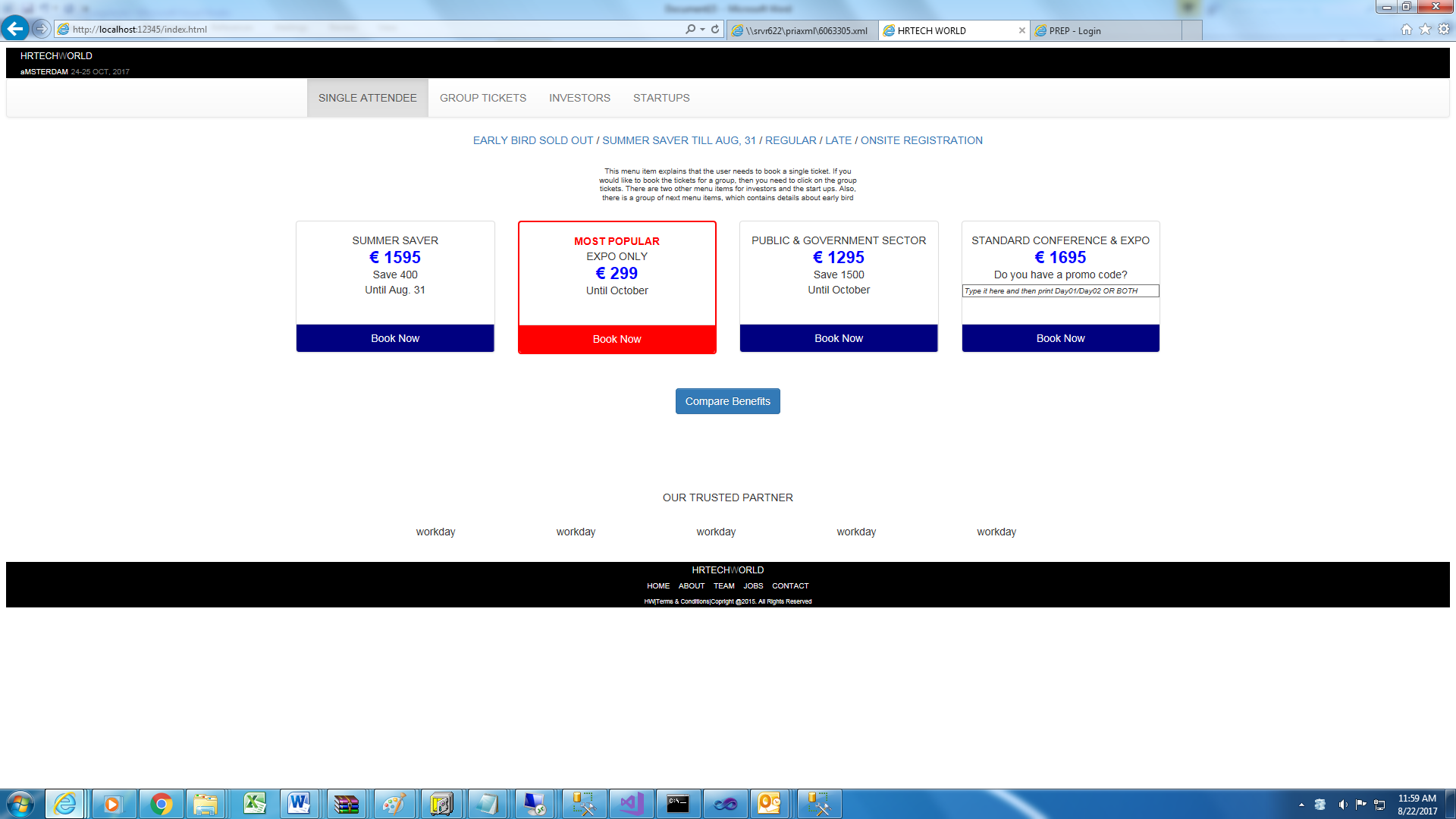


* My second challenge was to show the second menu items in a graceful format in both mobile and desktop application, but I could not do that using bootstrap only, so I had to take an approach where I have two divs for mobile and tab and desktop version.

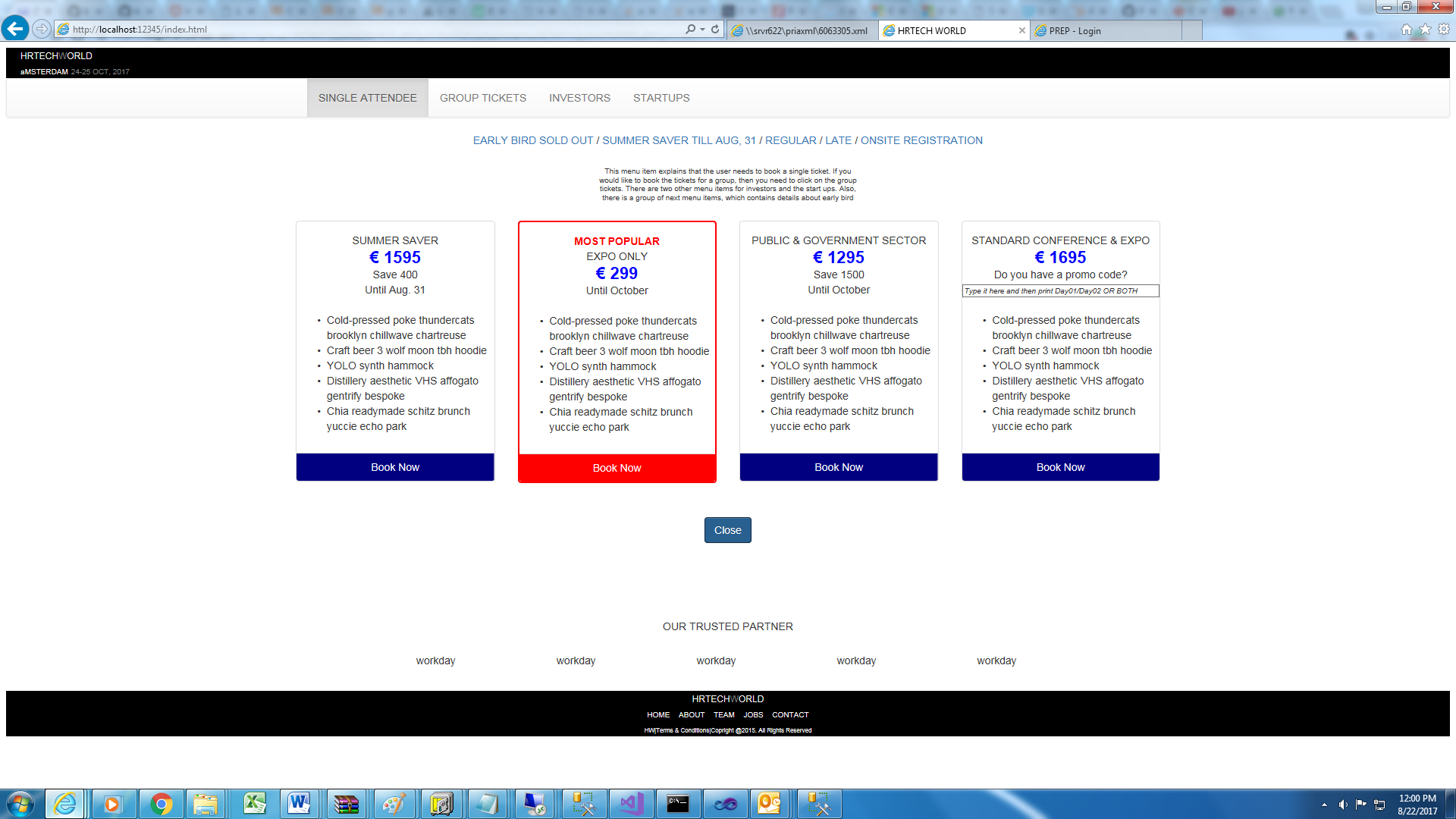


* The third challenge or rather my point of view was to show the menus in mobile which could gracefully be shown arranged vertically when required and could be hidden when not needed. This is easily accomplished with the use of bootstrap framework and is a standard practice in all mobile websites. That is why I digressed from the template provided to me. Please note that this is also good for future enhancement purposes as the users may requests for more menus in future.

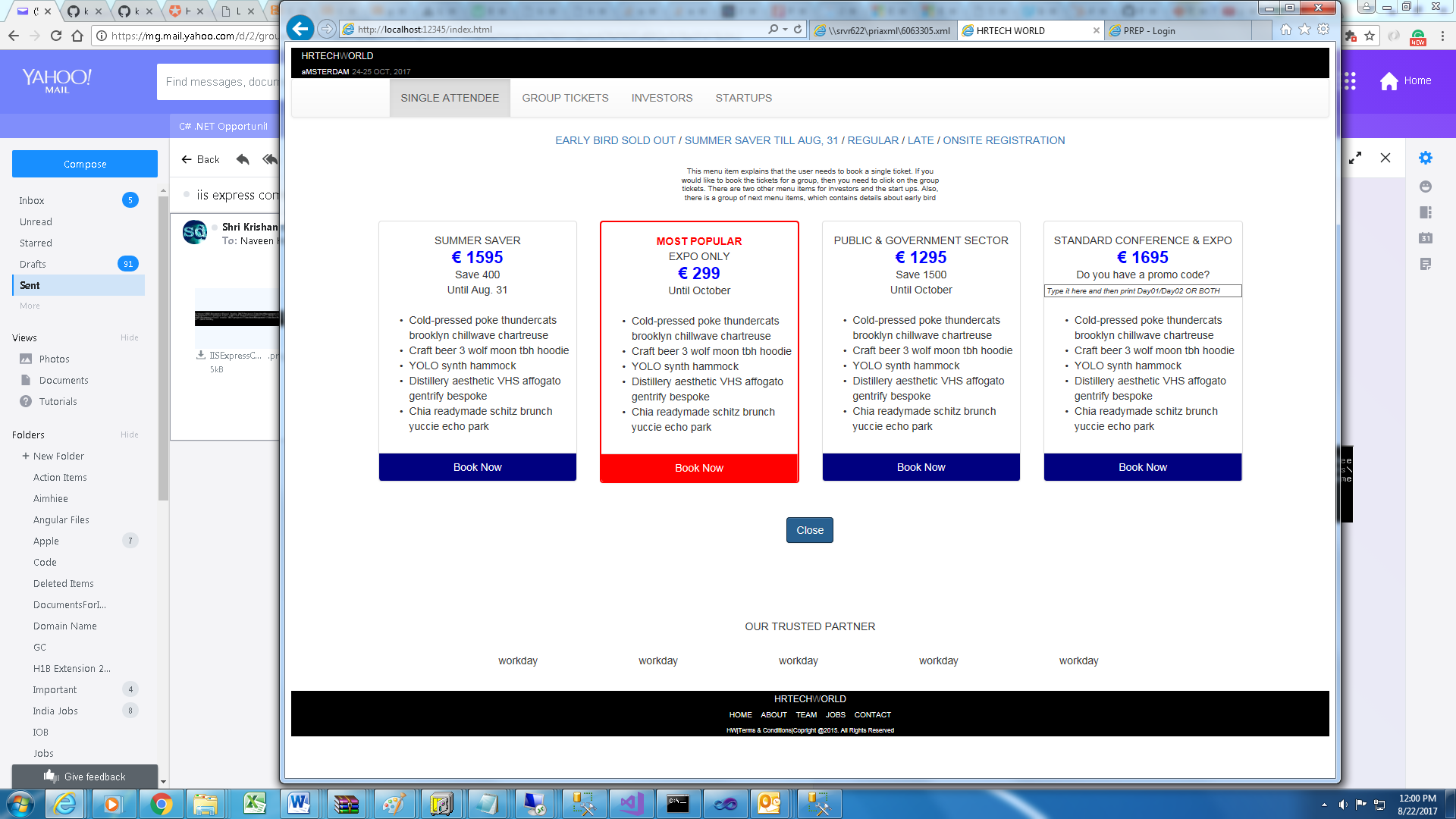
You should be able to view the sample that you provided me. I have tested the application with different screen sizes and it looks reasonable good in all of them. Here are the pictures:



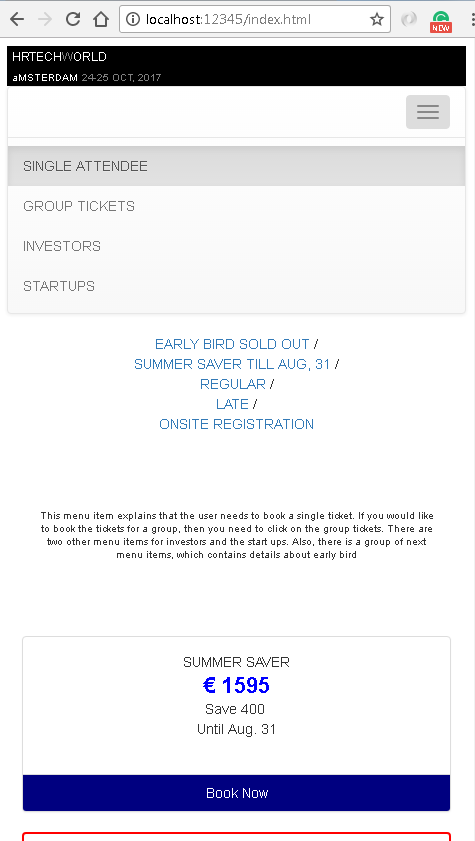
I) Desktop version (width: 1920px) with no body shown in panels (above)



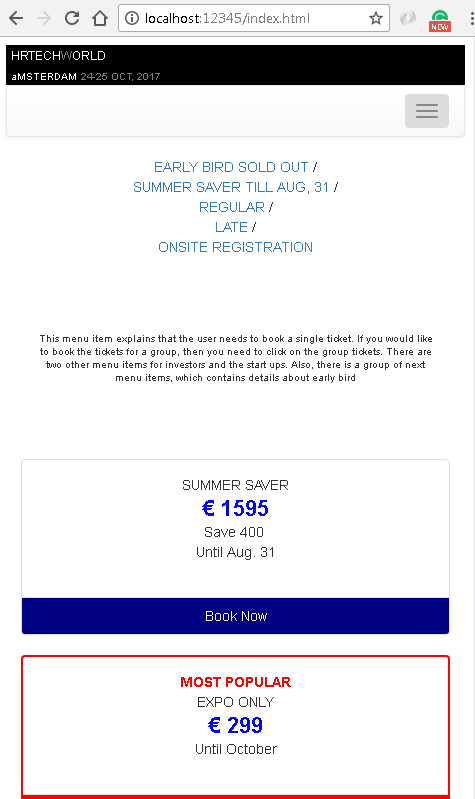
II) The desktop version (width: 1920px) with body shown in panels (above)



III) The tablet version (above)



IV) The mobile version with vertically arranged menus (above).



V) The mobile version with hidden menus (above).