PointerX

Armin / Zahra

FrontEnd / LO Agency

2021-08-15

1. Introduction

PointerX is the new name for ”neshan.org”. This project has some differences from original version. We tried to fix and improve main UI and we believe this is one of the strengths of our project. We also tried to use our creativity to build this project and we do not consider this project just a clone.

2. Team members / Tasks (Sort by tasks)

Armin:

1. Main files for master branch / Time spent: 1hr
2. Section 1 / Time spent: 1:30hr
3. Section 3 / Time spent: 1hr
4. Footer (Two steps) / Time spent: 4hr
5. Responsive for 1024 / Time spent: 3hr
6. Add JavaScript into the project (Two steps) / Time spent: 6hr
7. Refactor project / Time spent: 2hr
8. Project document / Time spent: 4hr

Zahra:

1. Header / Time spent: 12hr
2. Section 2 / Time spent: 9hr
3. Responsive for 1024 / Time spent: 4hr
4. Responsive for 767 / Time spent: 5hr
5. Responsive for 319 / Time spent: //

3. Team goals

1. Complete the project on time
2. Improved the project compared to the original version
3. Use the best practices

4. Support each other in problems

Subjects (Sort by tasks/Written by task owner)

- Main files (Armin)
- Section 1 (Armin)
- Section 3 (Armin)
- Footer (Armin)
- Responsive for 1024 (Armin)
- Add JavaScript into the project(Armin)
- Refactor project (Armin)
- Header (Zahra)
- Section 2 (Zahra)
- Responsive for 767 (Zahra)
- Responsive for 319
  - ☑ ~~Complete the project on time~~
  - ☑ ~~Improved the project compared to the original version~~
  - ☑ ~~Use the best practices~~
  - ☑ ~~Support each other in problems~~

Main files

To begin, we needed the main files on the master branch to do our tasks. So I (Armin) created the main files and then pushed on the master branch. Our main files contain one HTML file for all sections and one CSS file for main styles.
I created tags for header, main and footer in HTML files. Of course, the main tag also includes three section tags too. I did this to block conflicts in the merge step.
In addition to HTML and CSS files, i created font files and images directory for easier development. Later i added a JavaScript file too.

Section 1

In the original version of the project, Section 1 has two parts but we decided to create this section in one part. For part one I used a background image to manage the image better and use some properties like background size and background position. I used cover size and center position for the Tehran image.
And for the text, i used margin auto for left and right and some padding and margins for up and down.
For part two i used one img tag for image and h2 and p for texts and one button. To put this items

in the middle i used a flex display. Justify content center to put items middle to width and align items to put items middle to height.
I created a class for button styles and i used this class for all buttons. For the text i did the same.

Section 3

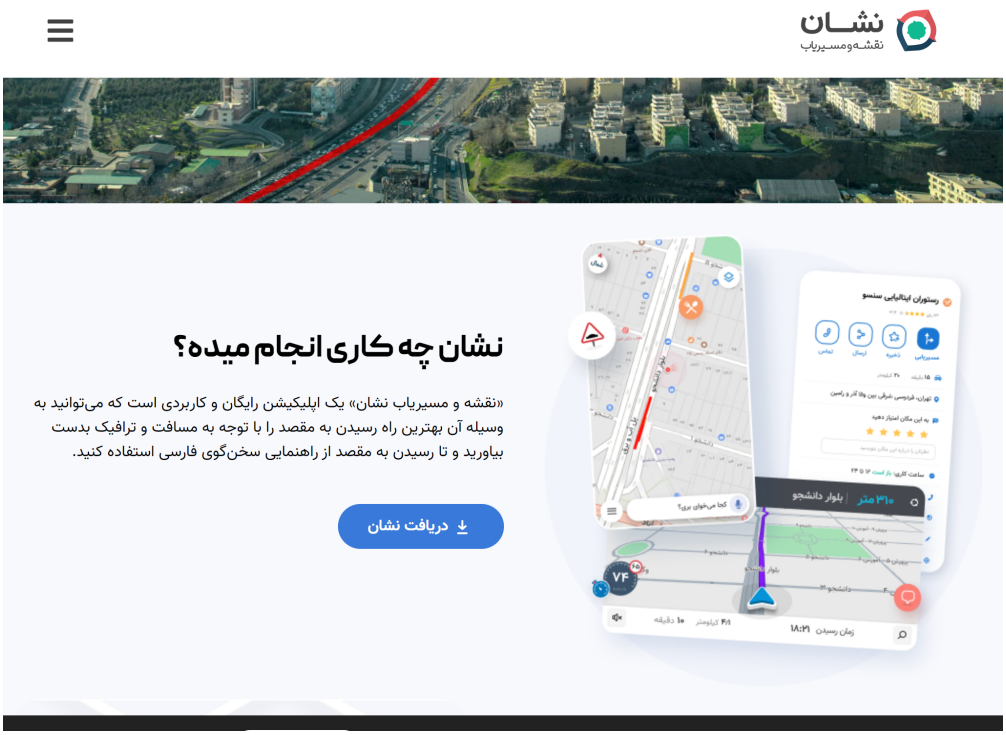
Exactly like section 1, Section 2 has two parts. I used a flex display to put these parts in the middle.
For part one i used a container div and in this div i used more div for other parts.
At the top i used h tag for the title. For the search input like the original version, i used div to create search input. In fact this search input is not working; it's just decorative. So i used div too. For the search icon and others i used SVG. SVG icons are better than png and other image formats, you can edit and style your image better.
For part two i used one container div and child div for two boxes. I used flex display for container with row direction and the same classes for child divs. I tried before the responsive steps make these sections responsive and i think i was good at it.

Footer

At first it was not supposed to i make the footer but because of the time conditions i was forced to do this. I divided the footer into three parts. For part one i use a div and in the div i used p tag and button. For the background i used background color property. For part two because of the links numbers i had to use container div.
For putting items into the middle, i used max width and margin auto for the left and right. And for styling links i used the same class and row direction for putting items inline.
This part was a little challenging. When you switch the resolution to 1024, award image moves and links order make this challenge harder. So I made a new div for image and i used negative margin to the top for fixing this problem.
At the end, part three it's a simple paragraph with an inside link. I used p tag and span tag inside paragraph.

Responsive for 1024

Like footer At first it was not supposed to i do this task but because of the time conditions i was forced to do this. This part compared the original version has some differences. In the original version section 1 its not responsive for 1024 but i did it right!



Another part of responsiveness for 1024 is the footer. I tried to improve font sizes and Sections width to footer be more readable.

Add JavaScript into the project

I think this is the best part of the project, at least for me. In this project i found two features that need JavaScript. The first one is the black part of the footer. Thet need to move with the mouse scroll and when it arrived to the footer change the position to static. I did this challenge with the window.scrollY option.
The second one is something like a hamburger menu for mobile versions. I did This with some functions and classes and i didn't have any challenges.

Refactor project

Because of the time conditions I didn't spend much time on refactoring.
I checked and fixed the codes and sections for about 2 hours. For example some parts of the header could have been better. Like Margins between links and company logo positions. Overall i believe our codes really don't need a refactor.

Header

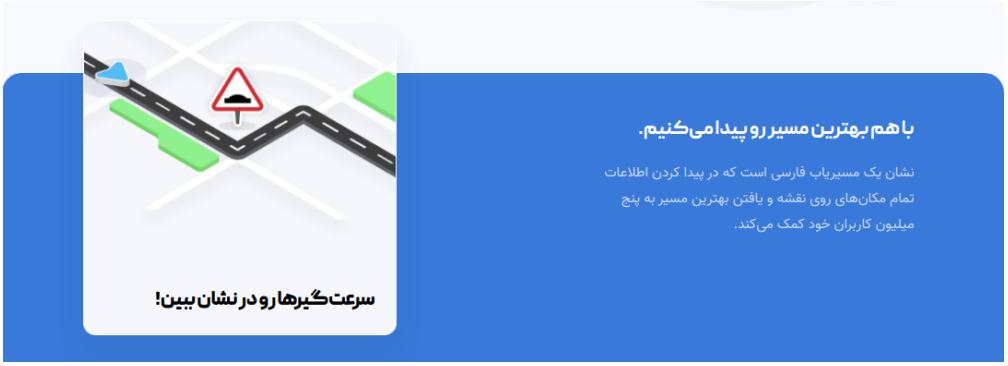
I coded the site header section in two different codes because the first code had problem with responsiveness. The challenges I faced were initially to fix the header at the top of the site and other elements. I was able to solve this problem with z-index:10. The next challenge was related to the download button. At first, I coded it with< a > tag, but during the response, there was a problem. In changing the code, I used the button tag, and the problems that arose were solved. The first time I coded the site header code, I did not pay attention to the fact that my code should be responsive in different sizes, and this caused me to have to look at all the codes in this section again and change them. I had to delete a series of extra sections and this made my code more efficient and responsive in different sizes. The other thing I learned in this header section was that we can use some of the elements in our code, but in some sizes, we can hide the display so that we can display it in the size we need.

Section 2

Section two, which was related to me, I had to type the code twice again, The first code I coded again had trouble responding in different sizes ,I split the code in Section 3 into three main sections, each containing a picture and text, and because it had duplicate elements, I used a common class.My main challenge for this section was to fix a text on a photo, which I was able to solve through Figcaption code.
I used the figure tag for the photos in each part two section.The main section has three similar sections.

Responsive 1024 (Zahra)

For responsive size 1024, I did the code of the header and section 2. One of the elements of section 2 (menu) is hidden in normal size, but has a display in smaller sizes.Section 2 also has the photos they need in 1024 size
Find a larger size.



Responsive 767

The logo and the menu had to be a little bigger Section 1 I had to change the margin a bit to keep its distance from the other sections.Section Two of its elements should all be in column state, and Part 2 Section 2, which was reversed, I used the flex-direction: column-reverse; code.