My Online CV

Author

Anca Preoteasa

Coordinator

Andrei Dascăl

Contents

1. Objectives

Through this online CV I am putting to use the knowledge I have gathered at the IT Informal School. This is both **great practice of my programming skills** as well as an opportunity to **stand out** and be noticed in the IT market.

2. Why an online CV?

2.1. Static CV vs online CV

Unlike a static CV, an online CV is not all talk since it is a **concrete proof** of the skills and work experience that I have acquired throughout the programming course.

It is also **at our disposal** whenever needed and **easier to send to potential employers** as well as more convenient for them to find key words and look for specific pieces of information.

There is more room to add as much information as you want in it, being accessible for visually impaired users through screen readers.

Furthermore, this tailor-made online CV also displays **my specialties** and **sense of initiative** through the choice of design and ways of interacting with the user, use of color palette, images, dynamic features, etc.

2.2. Market research

According to "Why Every Job Seeker Should Have a Personal Website, And What It Should Include", **80% of job seekers want a personal website (yet, only 7% have one).**

3. Application description

3.1. Structure and Visuals

The site consists of a single page (one-page application) as I find this option **easier**, **faster** and **more natural** for the user to navigate. Moreover, all the necessary data can be stored in the browser when loaded the first time, so the user **can still navigate the page even offline if needed**.

There is also no need to duplicate HTML and CSS elements such as the header or the footer, which are always the same regardless of the page the user is on, the process being easier and faster for the web developer too.

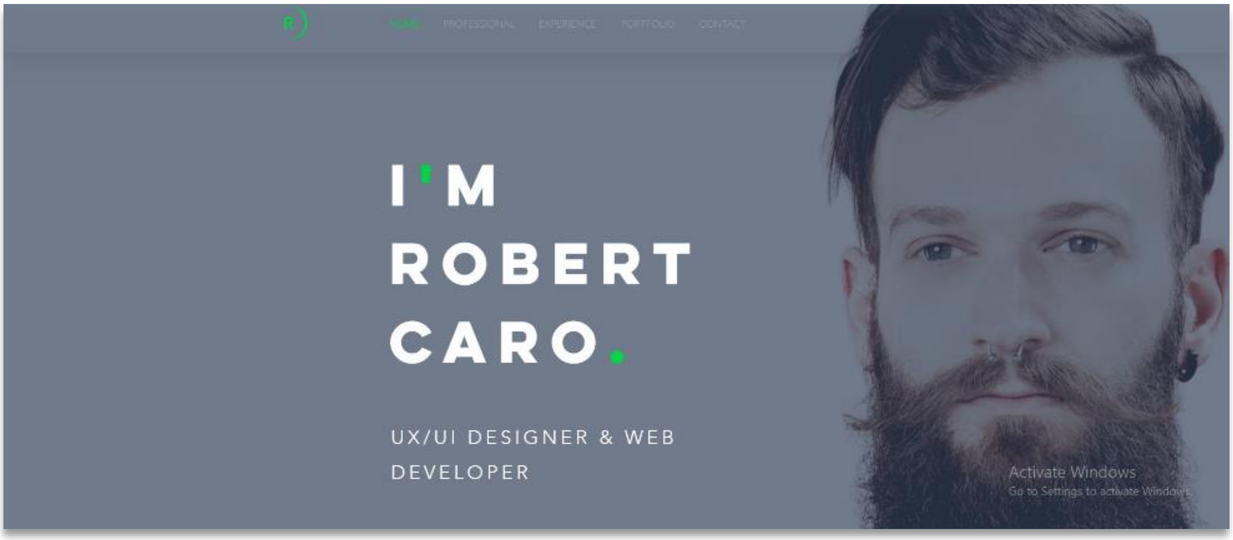


Fig. 1 – Template inspiration

3.2. Technical details

The page is made up of a header, a main section and a footer.

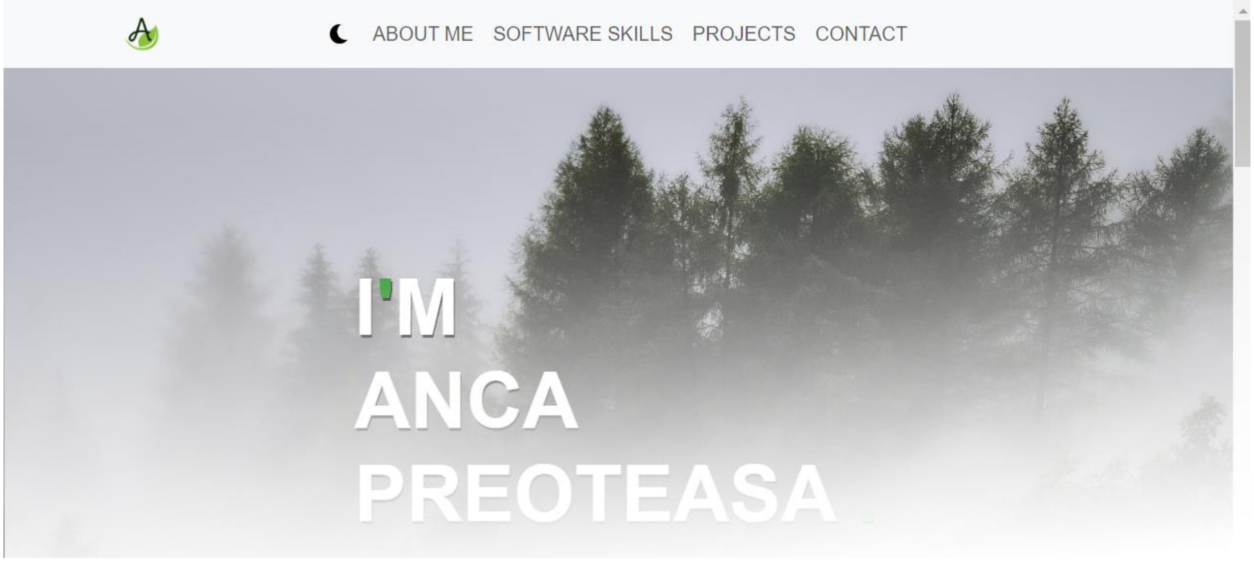


Fig. 2 – Header

The **header** includes my logo, a Dark Mode icon which changes the theme to a gray text on a dark background and the About Me, Software Skills, Projects and Contact links. It also has the fixed CSS property so that it remains on the page and the user can easily go back to it.

The **main section** contains the About Me, Software Skills, Projects and Contact sub-sections.

- **About Me** includes a photo of me, a welcoming message and a download button for the static version of the CV.

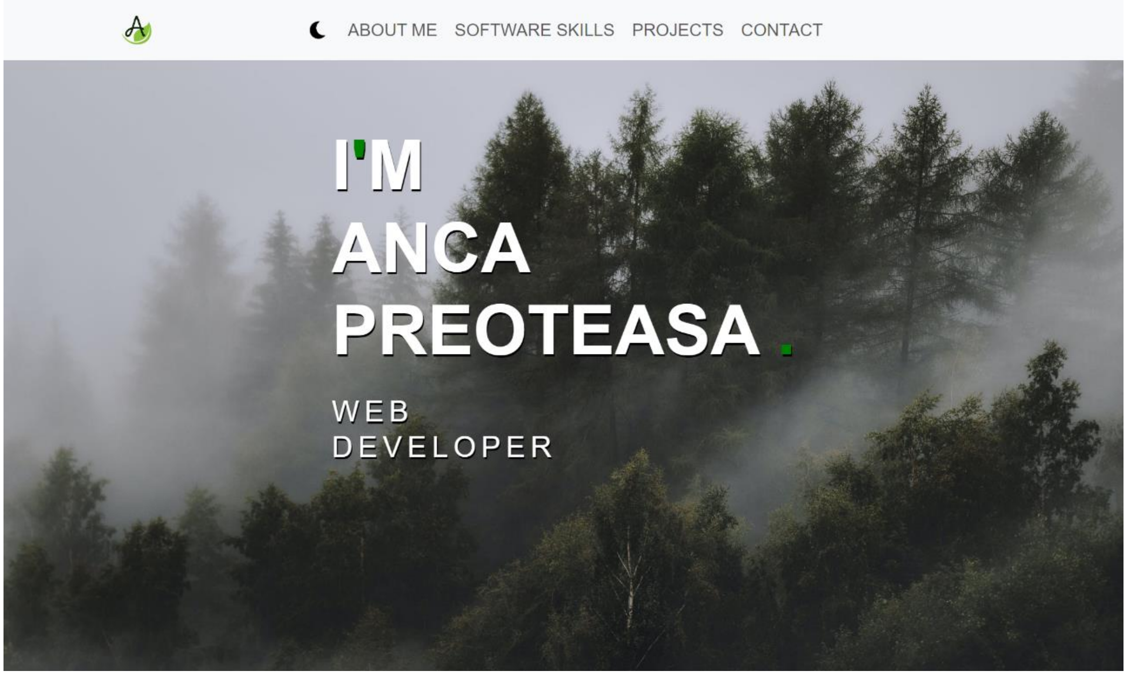
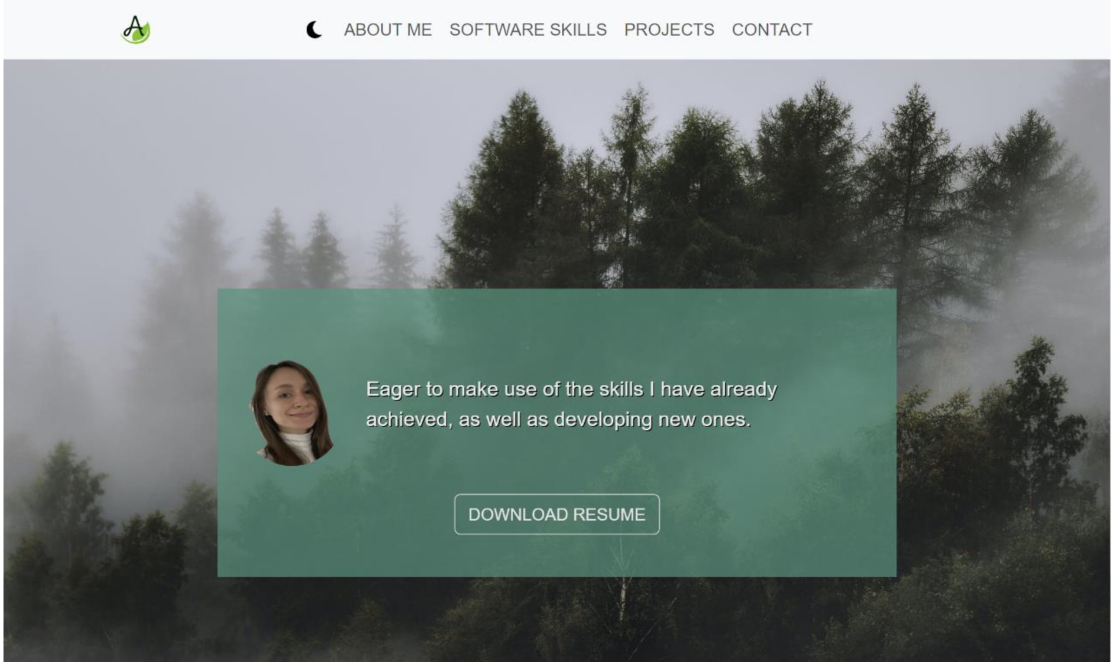


Fig. 3 – About me section



Eager to make use of the skills I have already achieved, as well as developing new ones.

Fig. 4 – Short description section

- **Software Skills** includes bars which show my current level of HTML, CSS, JavaScript, TypeScript, Angular, Bootstrap and SASS.

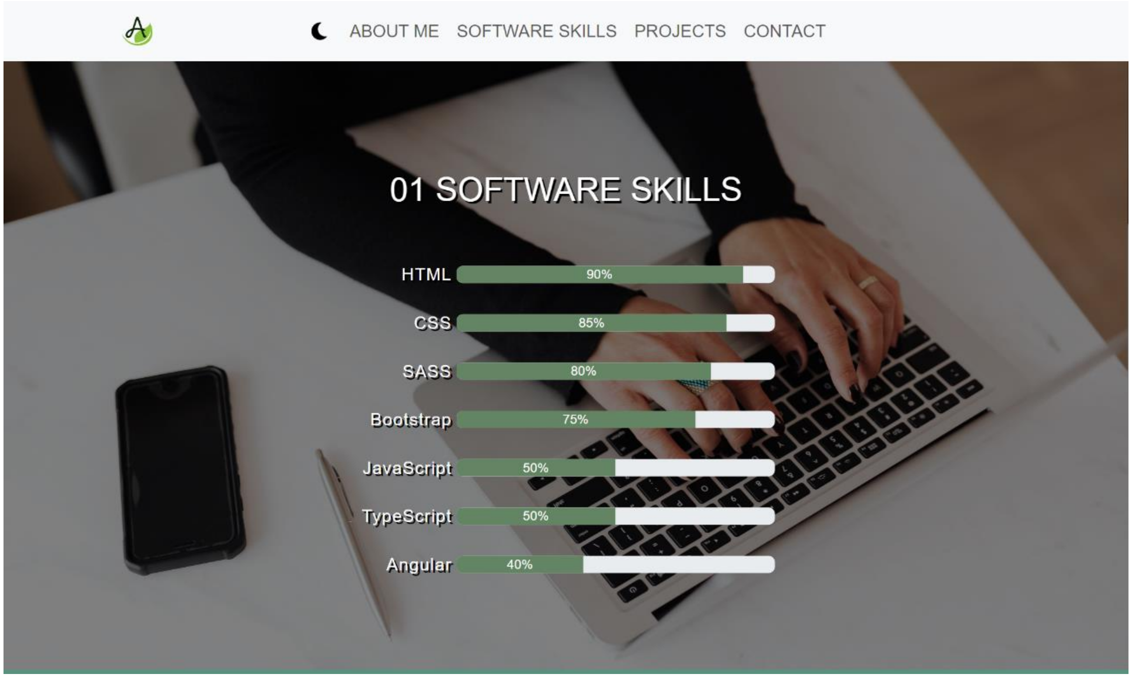


Fig. 5 – Software skills section

- **Projects** includes some of my best projects finished during the course.

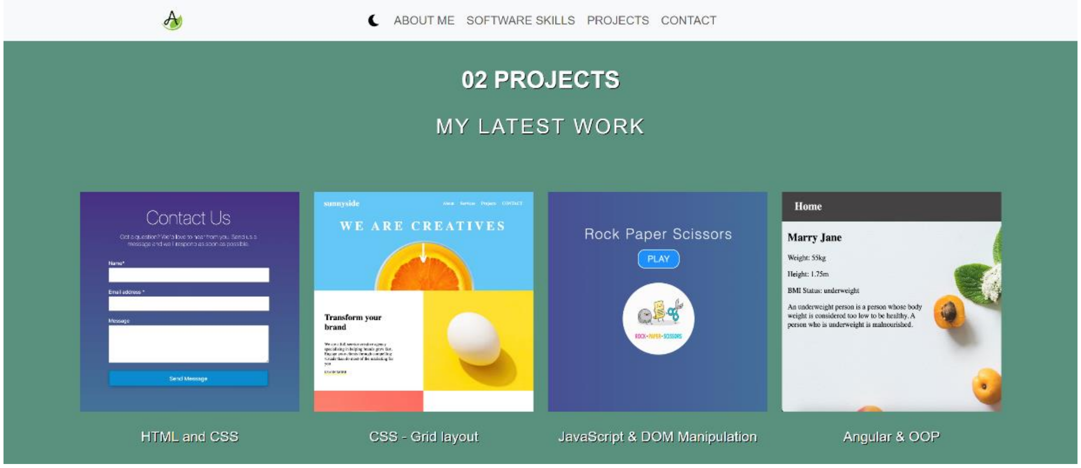
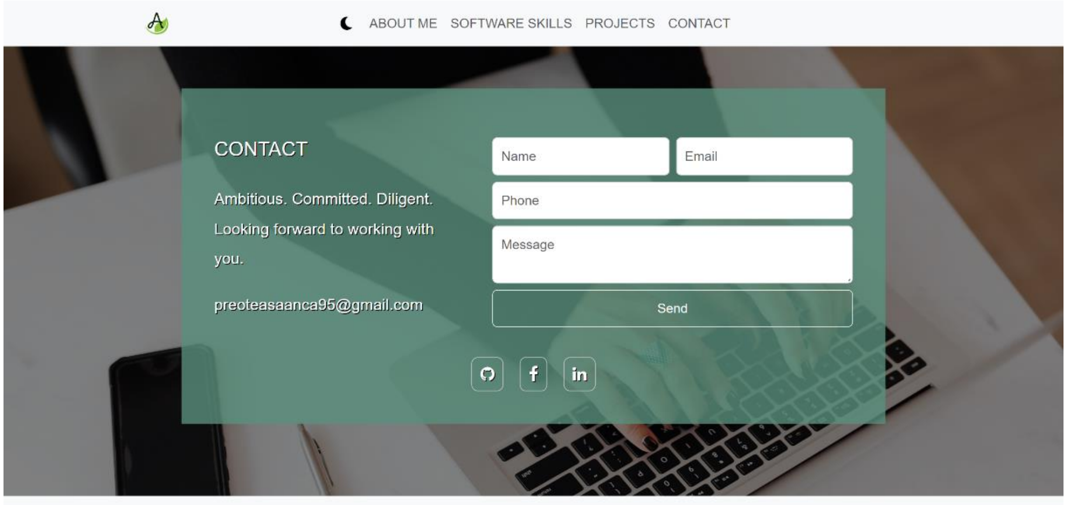


Fig. 6 – Projects section

- **Contact** includes a contact form with my email address and GitHub, Facebook and LinkedIn links to my accounts, for potential employers to send messages directly.



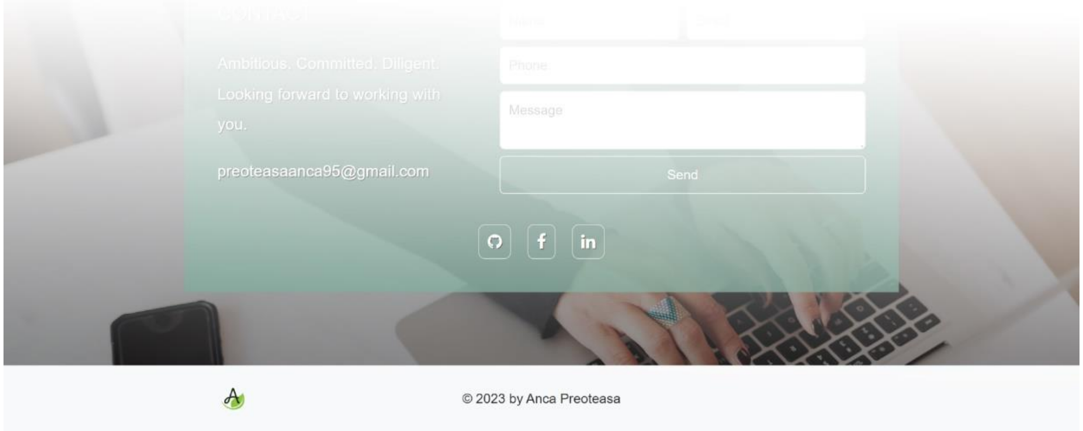
CONTACT

Ambitious. Committed. Diligent. Looking forward to working with you.

preoteasaanca95@gmail.com

Fig. 7 – Contact section

The **footer** contains a logo and a copyright notice.



Ambitious. Committed. Diligent. Looking forward to working with you.

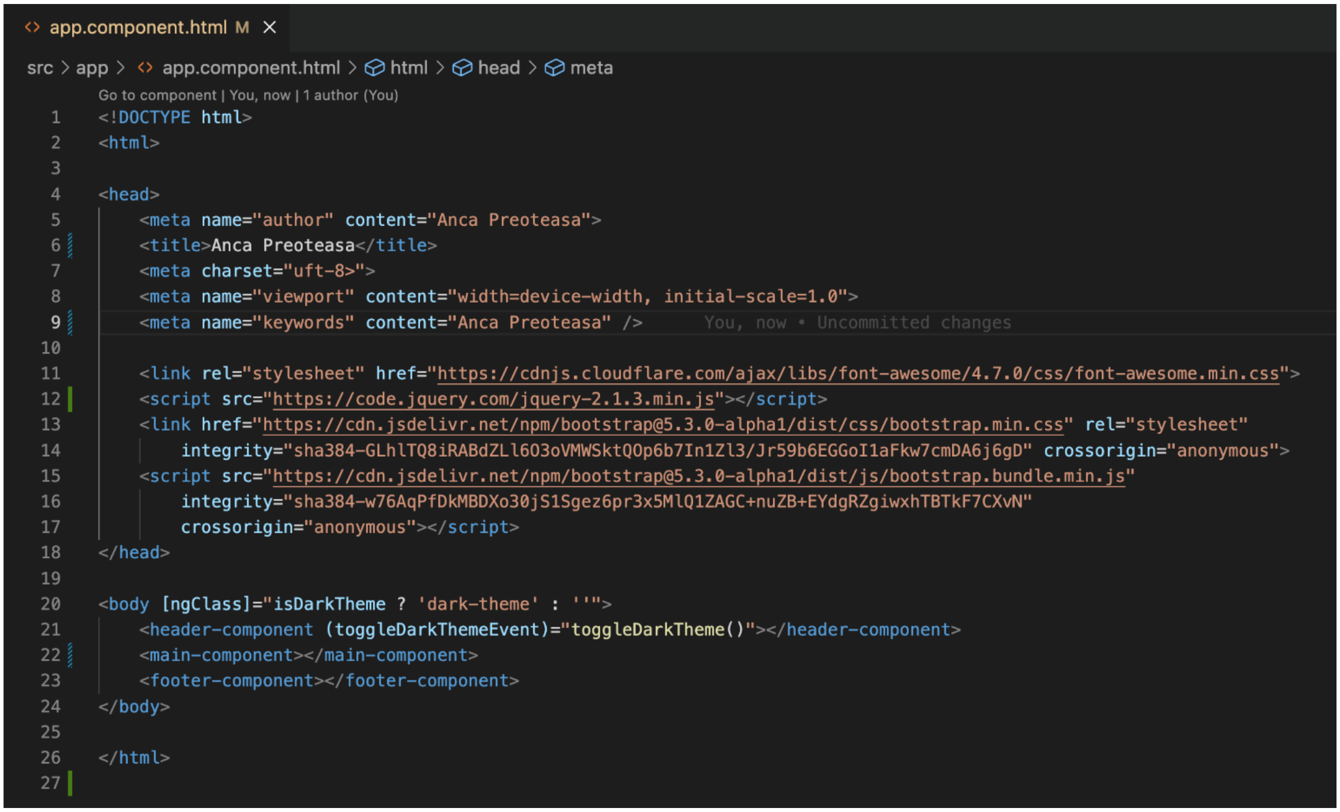
preoteasaanca95@gmail.com

© 2023 by Anca Preoteasa

Fig. 8 – Footer

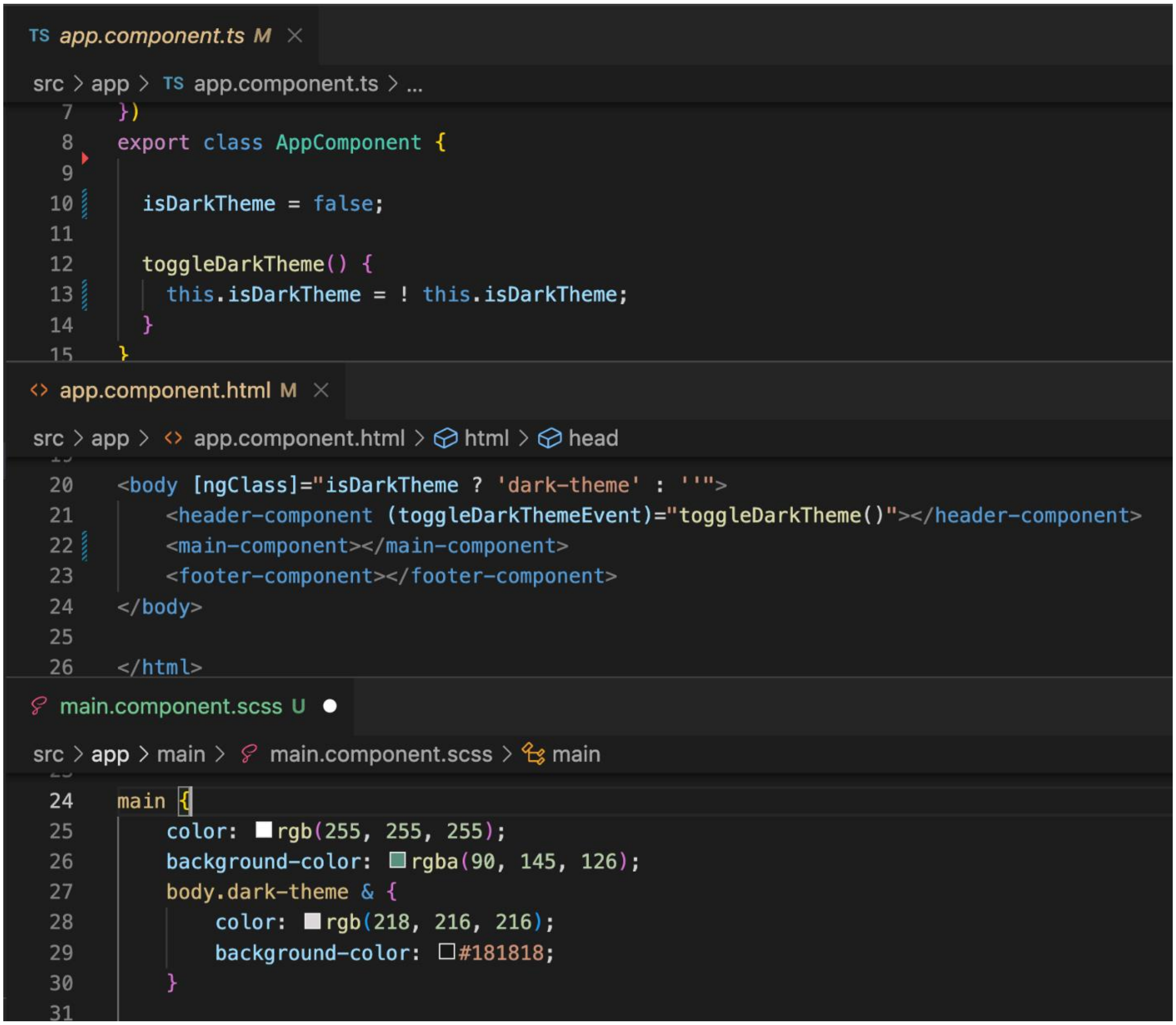
The **navigation** to different sections on the page can be done manually by using the up and down arrows on the keyboard or by scrolling the mouse or swiping if on mobile. It can also be done through the About Me, Software Skills, Projects and Contact links in the header.

3.3. Pieces of code


app.component.html M ×
src > app > app.component.html > html > head > meta
Go to component | You, now | 1 author (You)
You, now • Uncommitted changes

```html
<!DOCTYPE html>
<html>

<head>
    <meta name="author" content="Anca Preoteasa">
    <title>Anca Preoteasa</title>
    <meta charset="uft-8>">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="keywords" content="Anca Preoteasa" />

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
    <script src="https://code.jquery.com/jquery-2.1.3.min.js"></script>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-GLhlTQ8iRABdZLl6O3oVMWSktQOp6b7In1Zl3/Jr59b6EGGoI1aFkw7cmDA6j6gD" crossorigin="anonymous">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-w76AqPfDkMBDXo30jS1Sgez6pr3x5MlQ1ZAGC+nuZB+EYdgRZgiwxhTBTkF7CXvN"
        crossorigin="anonymous"></script>
</head>

<body [ngClass]="isDarkTheme ? 'dark-theme' : ''">
    <header-component (toggleDarkThemeEvent)="toggleDarkTheme()"></header-component>
    <main-component></main-component>
    <footer-component></footer-component>
</body>

</html>
```

Fig. 9 – HTML structure



TS app.component.ts M
src > app > TS app.component.ts > ...

```typescript
})
export class AppComponent {

  isDarkTheme = false;

  toggleDarkTheme() {
    this.isDarkTheme = ! this.isDarkTheme;
  }
}
```

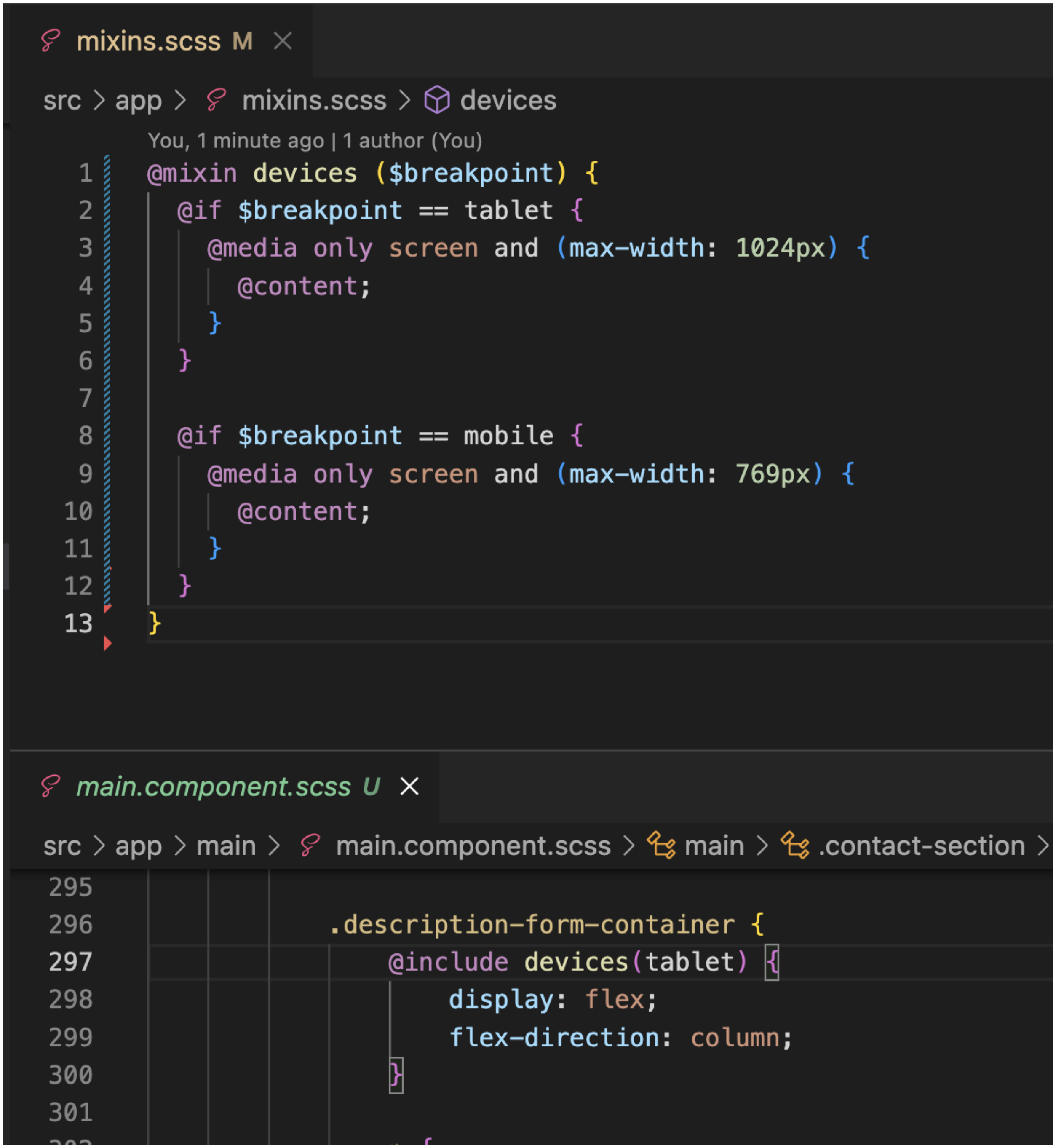
app.component.html M
src > app > app.component.html > html > head

```html
<body [ngClass]="isDarkTheme ? 'dark-theme' : ''">
    <header-component (toggleDarkThemeEvent)="toggleDarkTheme()"></header-component>
    <main-component></main-component>
    <footer-component></footer-component>
</body>

</html>
```

main.component.scss U
src > app > main > main.component.scss > main

```scss
main {
    color: rgb(255, 255, 255);
    background-color: rgba(90, 145, 126);
    body.dark-theme & {
        color: rgb(218, 216, 216);
        background-color: #181818;
    }
```

Fig. 10 – Dark theme

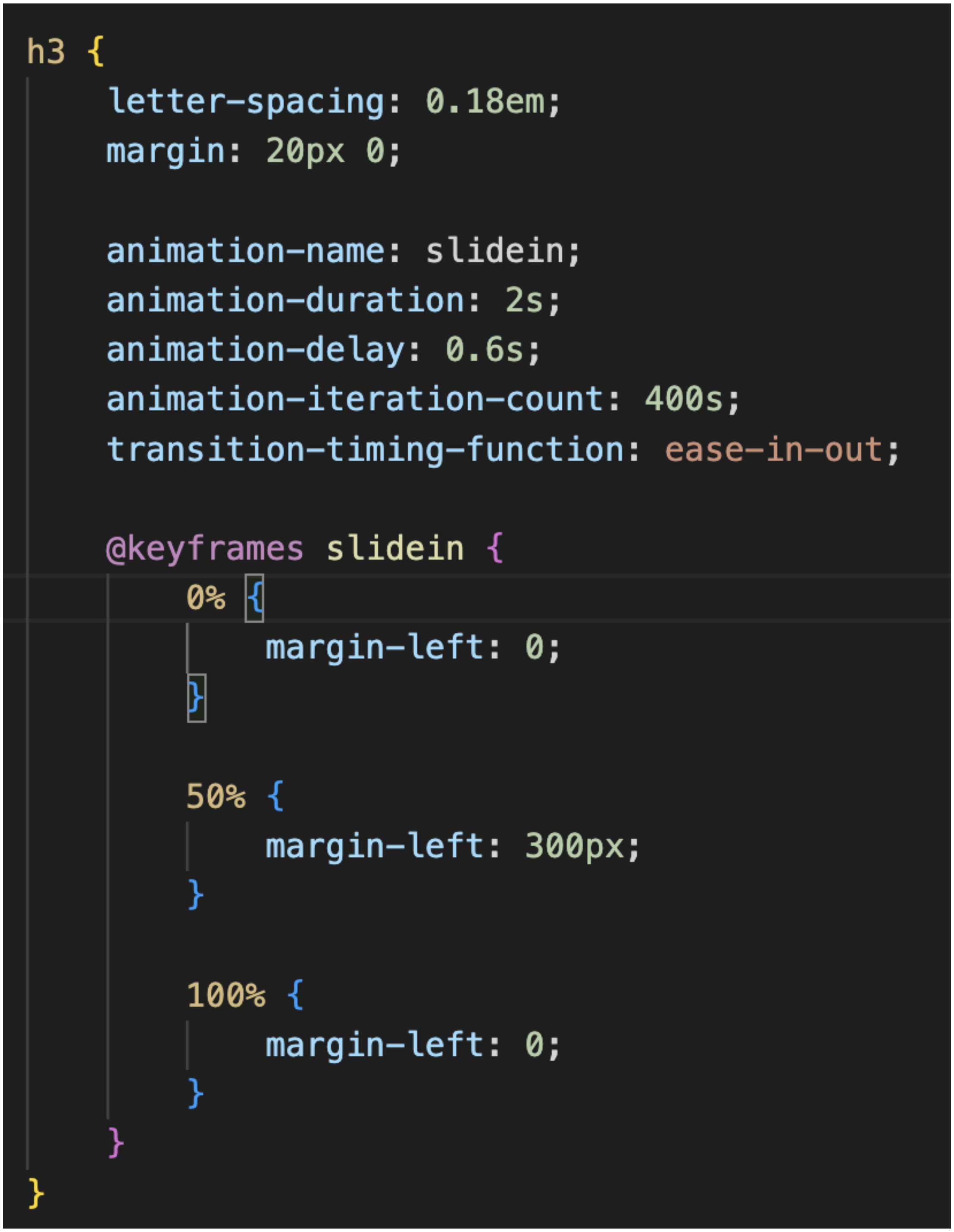

mixins.scss M
src > app > mixins.scss > devices
You, 1 minute ago | 1 author (You)

```scss
@mixin devices ($breakpoint) {
  @if $breakpoint == tablet {
    @media only screen and (max-width: 1024px) {
      @content;
    }
  }

  @if $breakpoint == mobile {
    @media only screen and (max-width: 769px) {
      @content;
    }
  }
}
```

main.component.scss U
src > app > main > main.component.scss > main > .contact-section >

```scss

            .description-form-container {
                @include devices(tablet) {
                    display: flex;
                    flex-direction: column;
                }

```

Fig. 11 – Responsive design for tablet and mobile



```
h3 {
    letter-spacing: 0.18em;
    margin: 20px 0;

    animation-name: slidein;
    animation-duration: 2s;
    animation-delay: 0.6s;
    animation-iteration-count: 400s;
    transition-timing-function: ease-in-out;

    @keyframes slidein {
        0% {
            margin-left: 0;
        }

        50% {
            margin-left: 300px;
        }

        100% {
            margin-left: 0;
        }
    }
}
```

Fig. 12 – CSS animation

4. Technologies & Tools

4.1. Technologies:

Fig. 13 – HTML, CSS and JavaScript logos

- HTML
- CSS
- JavaScript

HTML (HyperText Markup Language) defines the meaning and structure of the web content, CSS describes a web page's **appearance** and **JavaScript** describes its behavior.

Fig. 14 – Angular, TypeScript, Bootstrap and SASS logos

- Angular

Angular is a development platform, built on TypeScript. It is designed to make updating as easy as possible.

- TypeScript

TypeScript is a strongly typed programming language that builds on JavaScript.

- Bootstrap

Bootstrap is a free, open-source front-end development framework which provides a collection of syntax for template designs.

- SASS

Sass is the most mature, stable, and powerful professional grade CSS extension language in the world.

4.2. Tools:

Fig. 15 – Git and GitHub logos

Git

Git allows developers to collaborate with other developers on a project without the danger of them overwriting each other's work, and roll back to previous versions of the code base if a problem is discovered later on.

GitHub

GitHub is a site that provides hosting for your repositories and several tools for working with them.

Fig. 16 – Visual Studio Code and Google Chrome Developer Tools logos

Visual Studio Code

Visual Studio Code is a code editor redefined and optimized for building and debugging modern web and cloud applications.

Google Chrome Developer Tools

Google Chrome Developer Tools is a set of web developer tools built directly into the Google Chrome browser. These tools let you inspect the rendered HTML (DOM) and network activity of your pages.

4.3. Programming languages market share

The programming languages market share helped me choose the most used programming languages when building the website, so that I could easily collaborate with other developers in the future.

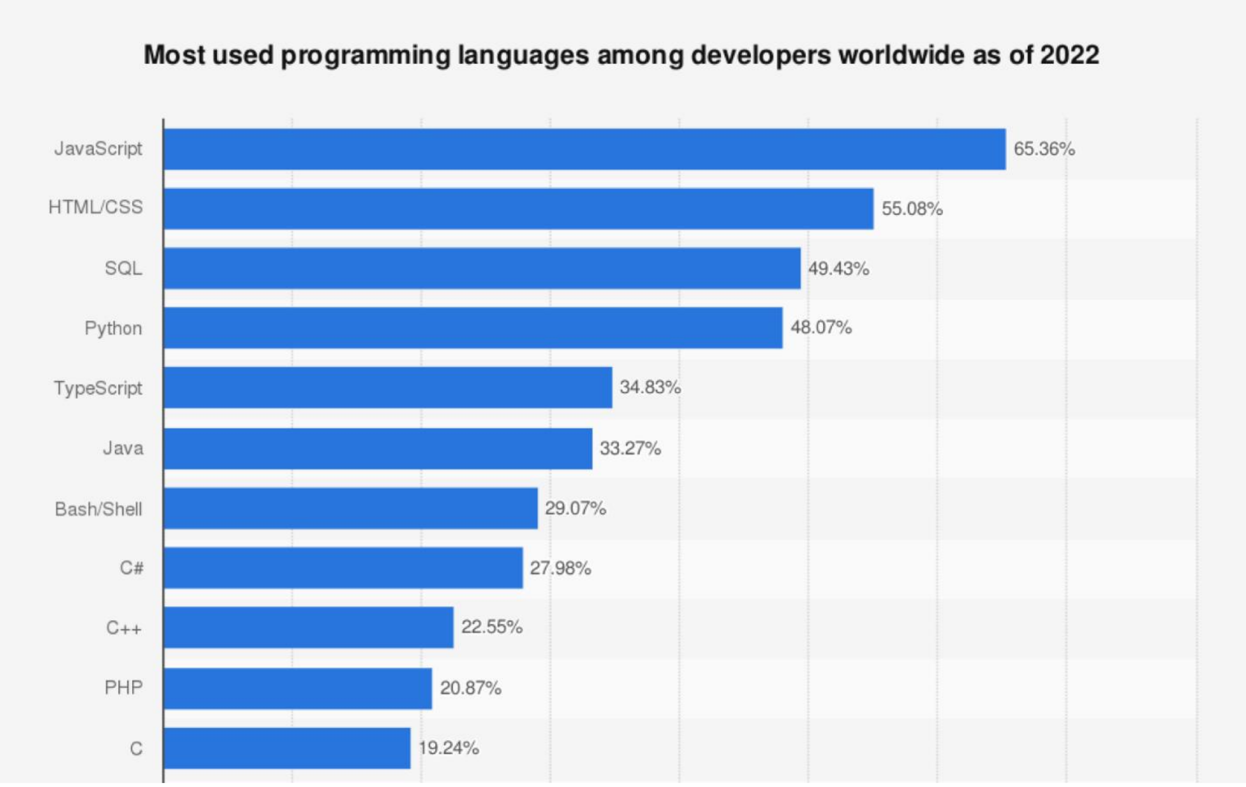


Fig. 17 - Statistics most used programming languages in 2022

5. Tasks planning

5.1. Researching similar projects - 1 day

Let the pros do their thing. As proud as I am of my artistic skills, an UX designer has the proper training and experience to create a more efficient web page, so researching is the smart idea.

5.2. Design resources - 1 day

The functionality of the website is what gives the user a pleasant experience and makes him not leave your page and later on come back to it. Combine it with a nice, clean design and the user will be more likely to not leave your site due to ambiguities and chaos.

5.3. HTML structure - 2 days

Before doing anything else, the HTML structure is essential to the website. **Do it right, and it will save you so much time in the long run**. I started by defining the HTML structure because it is like establishing the bare walls of a building before adding the paint, the wooden floors and everything else.

5.4. CSS styling & Responsiveness - 3 days

I continued with the CSS, thus styling the house.

Styling with CSS for various types of screens, starting from **mobile** (under 769px), continuing with a **tablet** format (between 769 and 1024px) and finishing with a **web browser** format (1025px or more).

Fig. 18 – Screen resolutions on mobile, tablet and web browser

Why mobile-first?

- The number of mobile users (60.53%) has surpassed desktop users (39.57%)
- Graceful Degradation vs. Progressive Enhancement – Start mobile-first, and you can easily adapt the web page for tablets and desktops

5.5. JavaScript implementation - 2 days

Next, I finished the house by adding the practical, useful features, such as the door, the windows, the lights and everything we need in order to make the house which in this case is my website, functional. In this era of technology, also known as the **Computer Age** or **Digital Age**, adding the **dynamism** to the page is no longer optional, but a fundamental must.

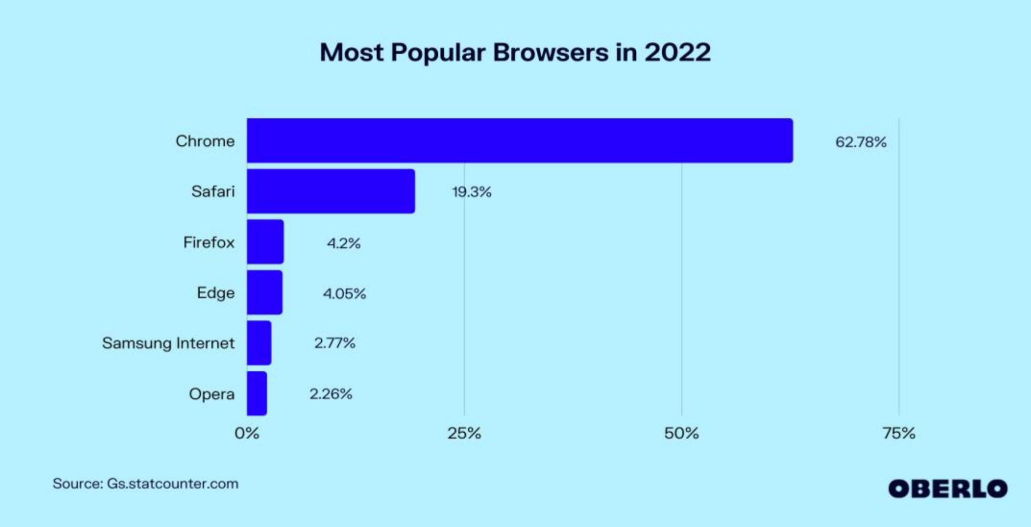
Fig. 19 – HTML, CSS and JavaScript visuals

5.6. Manual Testing - 1 day

Finding and fixing potential bugs by testing on mobile, tablet and desktop, as well as on different types of browsers: Chrome/Microsoft Edge, Safari and Mozilla.

Why Chrome/Microsoft Edge?

Because both Chrome and Microsoft Edge share the same core named Chromium and **they are currently the most used, according to statistics:**



Source: Gs.statcounter.com

Fig. 20 – Statistics popular browsers in 2022

6. Further development

In the future, I would like to add **translations into Romanian and Spanish**, Romania and Spain being the two countries in which I am seeking to find a job.

7. Bibliography

[1] – Marcus - *Google Chrome vs. Microsoft Edge: Which is better? -* https://www.expressvpn.com/blog/google-chrome-vs-microsoft-edge-which-is-better/

[2] – https://developer.mozilla.org/en-US/docs/Web/HTML

[3] – https://developer.mozilla.org/enUS/docs/Learn/Tools_and_testing/GitHub

[4] – *Most popular web browsers in 2022* – https://www.oberlo.com/statistics/browser-market-share

[5] – Satendra Bhadoria – *Future of Javascript In 2022: What to expect next?* – https://solguruz.com/blog/future-of-javascript/

[6] – Jash Unadkat – *Mobile First Design: What It Is and How to Implement It* – https://www.browserstack.com/guide/how-to-implement-mobile-first-design

[7] – https://developers.google.com/web/fundamentals/design-and-ui/responsive/

[8] – Jacquelyn Smith – *Why Every Job Seeker Should Have a Personal Website, And What It Should Include* – https://www.forbes.com/sites/jacquelynsmith/2013/04/26/why-every-job-seeker-should-have-a-personal-website-and-what-it-should-include/

[9] – https://developer.mozilla.org/enUS/docs/Learn/Tools_and_testing/Clit-side_JavaScript_frameworks/Angular_getting_started

[10] – https://code.visualstudio.com/

[11] - https://developer.chrome.com/docs/devtools/

[12] - https://www.techtarget.com/whatis/definition/bootstrap

[13] - https://sass-lang.com/

[14] - https://www.typescriptlang.org/