

Projekt z przedmiotu **Technologie internetowe**

Strona internetowa responsywna

Autor: Łukasz Jasiński

Prowadzący: dr Katarzyna Garwol

Rok akademicki: 2022/2023

Wprowadzenie

Nazwa projektu

Portfolio

Kodowanie znaków

UTF-8

Wykorzystane technologie

HTML , SASS/SCSS , JavaScript

Wykorzystane biblioteki

KUTE.js - animacja ścieżek SVG

Fontawesome- ikony

GoogleFonts-czcionka

Struktura plików

Portfolio




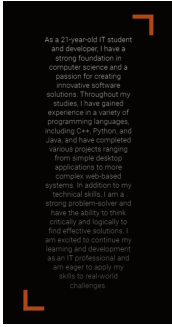
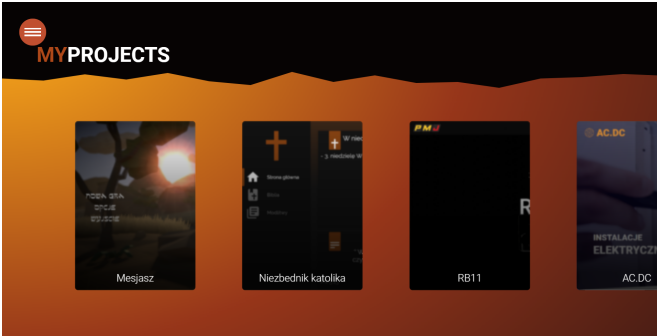
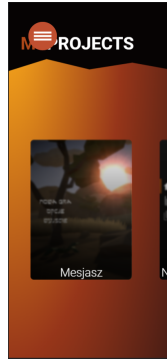
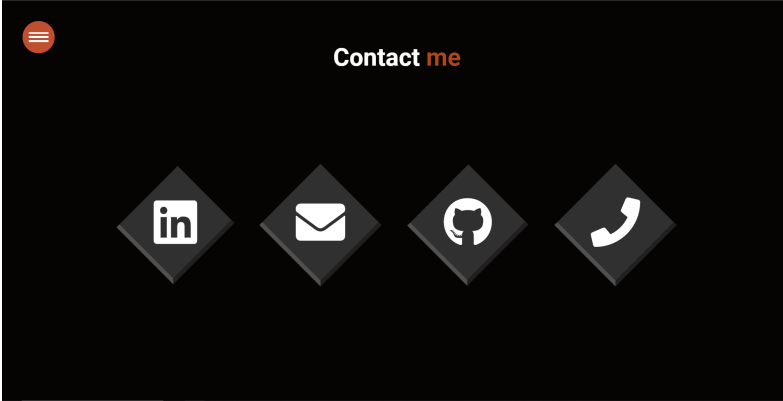

- \img - folder grafikami
 - \layers
- \styles - folder ze plikami stylów scss
 - about.scss
 - animations.scss
 - contact.scss
 - hero.scss
 - main.scss
 - media-queries.scss
 - mixins.scss
 - project.scss
 - navigation.scss
 - typography.scss
- index.html
- style.css
- script.js

Opis projektu

Projekt jest to responsywna strona portfolio zawierająca mój opis a także listę moich projektów godnych uwagi. Składa się z 4 sekcji których wygląd i budowa zostaną przedstawione niżej.

Uwaga przeglądanie listy projektów odbywa się poprzez przeciąganie.

Wygląd strony

Desktop	Mobile
	
	
	
	

Budowa strony

Nawigacja

```
<body>
  <nav class="nav" id="nav">
    <div class="nav__button" id="nav-button">
      <div class="nav__button_icon"></div>
    </div>
    <div class="nav__container" id="nav-container">
      <div class="nav__list">
        <a class="nav__list_item">Home</a>
        <a class="nav__list_item">About</a>
        <a class="nav__list_item">Projects</a>
        <a class="nav__list_item">Contact</a>
      </div>
    </div>
    <div class="nav__icons"></div>
    <div class="nav__background"></div>
  </nav>
</body>
```

Sekcja Główna

```
<main class="parallax">
  <div class="parallax__container">
    <svg class="hero__layer" id="main-layer" id="Warstwa_1" preserveAspectRatio="xMidYMid slice" width="1920" ...
    </svg>
    <header class="hero__header"></header>
    
    
    
    
    
  </div>
</main>
```

Znacznik svg jest zwinięty ponieważ znajduje się tam kilkaset linii wygenerowanego kodu SVG (musiał być umieszczony w postaci kodu by potem go zaanimować)

Sekcja O mnie

```
<section class="about trigger">
  <div class="about__container">
    <h1 class="about__header"><span class="text--orange"> </span> <span class="text--white"></span>
    </h1>
    <p class="about__description corners">As a 21-year-old IT student and developer, I have a strong
      foundation in computer science and a passion for creating innovative software solutions.
      Throughout my studies, I have gained experience in a variety of programming languages, including
      C#,JS,JAVA and more and have completed various projects ranging from simple desktop
      applications to more complex web-based systems. In addition to my technical skills, I am a
      strong problem-solver and have the ability to think critically and logically to find effective
      solutions. I am excited to continue my learning and development as an IT professional and am
      eager to apply my skills to real-world challenges.

    <div class="about__head">
      <div class="head">
        <div class="head__word">C#</div>
        <div class="head__word">Unity</div>
        <div class="head__word">.Net Core</div>
        <div class="head__word">Blender</div>
        <div class="head__word">JavaScript</div>
        <div class="head__word">Angular</div>
        <div class="head__word">PHP</div>
        <div class="head__word">WPF</div>
        <div class="head__eyes">
          <div class="head__eye">
            <div class="head__eye__middle"></div>
          </div>
          <div class="head__eye">
            <div class="head__eye__middle"></div>
          </div>
        </div>
        
      </div>
    </div>
  </div>
```

Sekcja projektów

```
</section>
<section class="projects trigger">
  <div class="projects__container">
    <div class="card" draggable="false" unselectable="on">
      <div class="card__image" src="" alt="" srcset=""></div>
      <div class="card__container">
        <h3 class="card__header">Mesjasz</h3>
        <div class="card__technologies">
          Unity C# Blender
        </div>
        <p class="card__description">
          Demo gry fabularnej stworzone w silniku unity. <br>
          Opowiada historie Jezusa w czasie i przed spotkania z Janem Chrzcicielem <br>
          II miejsce w Ogólnopolskim Konkursie GEEK <br>
          I miejsce w Konkursie Katechetycznych Aplikacji Multimedialnych
        </p>
        <div class="card__links">
          <a href="" class="card-link"></a>
        </div>
      </div>
    </div>
    <div class="card" draggable="false" unselectable="on">
      <div class="card__image" src="" alt="" srcset=""></div>
      <div class="card__container">
        <h3 class="card__header">Niezbędnik katolika</h3>
        <div class="card__technologies">
          C# WPF .NET SQL
        </div>
        <p class="card__description">
          Desktopowa aplikacja wyświetlająca różne informacje katolickie <br>
          Pozwala czytać biblię, wyszukiwać modlitwy a także dodawać własne i zapisywać ulubione
          Backend to .NetCore RestApi <br>
          II miejsce w Konkursie Katechetycznych Aplikacji Multimedialnych
        </p>
        <div class="card__links">
          <a href="" class="card-link"></a>
        </div>
      </div>
    </div>
    <div class="card" draggable="false" unselectable="on">
      <div class="card__image" src="" alt="" srcset=""></div>
      <div class="card__container">
        <h3 class="card__header">RB11</h3>
        <div class="card__technologies">
          HTML CSS JS GSAP BLENDER
        </div>
        <p class="card__description">
          Strona prezentująca produkt automatycznej śrutownicy PMJ RB11
          Wyświetlane animacje 3D w tle zostały stworzone w programie Blender
        </p>
        <div class="card__links">
          <a href="" class="card-link"></a>
        </div>
      </div>
    </div>
  </div>
</section>
```

itd reszta projektów

Sekcja kontaktu

```
</section>
<section class="contact trigger">
  

  <h1 class="contact__header"><span class="text--white">Contact</span> <span class="text--orange">me</span>
</h1>
  <ul class="contact__container">
    <li class="contact__option"><i class="contact__icon fa-brands fa-linkedin"></i></li>
    <li class="contact__option"><i href="" class="contact__icon fas fa-regular fa-envelope"></i></li>
    <li class="contact__option"><i class="contact__icon fa-brands fa-github"></i></li>
    <li class="contact__option"><i href="" class="contact__icon fas fa-phone"></i></li>
  </ul>
</section>
```


Style Strony

Przekonwertowany kod scss na kod css

```
.nav {
  background: none;
  transition: opacity 0.3s ease;
}

.nav__container {
  min-height: 100vh;
  min-width: 100vw;
  position: fixed;
  z-index: 20;
  align-items: center;
  visibility: hidden;
  display: flex;
  opacity: 0;
  transition: transform 0.5s;
}

.nav[data-active=false] .nav__button:hover .nav__button__icon::before {
  transform: translateY(0.2vh);
  transition: 0.1s;
}

.nav[data-active=false] .nav__button:hover .nav__button__icon::after {
  transform: translateY(-0.2vh);
  transition: 0.1s;
}

.nav[data-active=true] .nav__container {
  visibility: visible;
  opacity: 1;
  transition-delay: 1s;
  transition: opacity 0.5s;
}

.nav[data-active=true] .nav__background {
  transform: scale(60);
  transition: transform 0.5s;
}

.nav[data-active=true] .nav__icons {
  visibility: visible;
  opacity: 0.1;
  transition-delay: 1s;
}
```

```
    transition: opacity 0.5s;
}
.nav[data-active=true] .nav__button__icon {
    background-color: #00000000;
}
.nav[data-active=true] .nav__button__icon::after {
    top: 0;
    left: 0%;
    transform: rotate(225deg);
}
.nav[data-active=true] .nav__button__icon::before {
    top: 0%;
    left: 0;
    transform: rotate(135deg);
}
.nav__icons {
    background-image: url(img/Background.png);
    position: absolute;
    left: 0;
    top: 0;
    min-height: 100%;
    min-width: 100%;
    z-index: 12;
    opacity: 0;
    background-repeat: no-repeat;
    background-position: 0% -10%;
    background-size: 100vh;
    visibility: hidden;
}
.nav__background {
    position: fixed;
    left: 50px;
    top: 50px;
    border-radius: 50%;
    height: 8vh;
    width: 8vh;
    z-index: 11;
    background: #97361a;
    background: radial-gradient(#ffa11b, #97361a, #2c1413 100%);
    transition: transform 0.5s;
}
.nav__button {
    position: fixed;
```

```
    left: 50px;
    top: 50px;
    border-radius: 50%;
    background-color: #c04f30;
    z-index: 100;
    height: 8vh;
    width: 8vh;
    display: flex;
    justify-content: center;
    align-items: center;
    cursor: pointer;
}

.nav__button__icon {
    width: 5vh;
    height: 0.5vh;
    background-color: white;
    position: relative;
}

.nav__button__icon::after, .nav__button__icon::before {
    position: absolute;
    content: "";
    width: 5vh;
    height: 100%;
    background-color: white;
    display: inline-block;
    transition: 0.1s;
}

.nav__button__icon::after {
    top: -1vh;
}

.nav__button__icon::before {
    top: 1vh;
}

.nav__list {
    margin-left: clamp(2rem, 13vw, 32rem);
    z-index: 15;
}

.nav__list__item {
    color: white;
    font-size: clamp(2rem, 7vw, 7rem);
    display: block;
    font-weight: 300;
    padding: clamp(0.25rem, 0.5vw, 1rem) 0rem;
```

```
}  
.nav__list:hover .nav__list__item {  
  opacity: 0.3;  
}  
.nav__list:hover .nav__list__item:hover {  
  opacity: 1;  
  cursor: pointer;  
}  
.nav[data-active-index="0"] .nav__icons {  
  background-position: 0% -10%;  
  transition: background-position 0.3s ease;  
}  
.nav[data-active-index="1"] .nav__icons {  
  background-position: 0% 25%;  
  transition: background-position 0.3s ease;  
}  
.nav[data-active-index="2"] .nav__icons {  
  background-position: 0% 60%;  
  transition: background-position 0.3s ease;  
}  
.nav[data-active-index="3"] .nav__icons {  
  background-position: 0% 95%;  
  transition: background-position 0.3s ease;  
}  
  
#layer1 {  
  transform: translateZ(-50px) scale(6);  
}  
  
#layer2 {  
  transform: translateZ(-40px) scale(5);  
  left: -15%;  
  animation-delay: 0ms;  
}  
  
#layer3 {  
  transform: translateZ(-30px) scale(4);  
  animation: 1s popfromUp forwards ease-out;  
  animation-delay: 0ms;  
}  
  
#layer4 {  
  transform: translateZ(-20px) scale(3);
```

```
    animation: 1s popfromUp forwards ease-out;
    animation-delay: 200ms;
}

#layer5 {
    transform: translateZ(-10px) scale(2) scaleY(1.1);
    animation: 1s popfromUp forwards ease-out;
    animation-delay: 400ms;
}

#layer6 {
    animation: 1s popfromUp forwards ease-out;
    animation-delay: 600ms;
    width: 100vw;
    height: 100vh;
    top: 300vh;
}

.parallax {
    width: 100vw;
    height: 100vh;
    position: relative;
    perspective: 10px;
    top: 0;
    left: 0;
    overflow-x: hidden;
    overflow-y: scroll;
}

.parallax__container {
    transform-style: preserve-3d;
}

#main-layer {
    overflow: visible;
    position: absolute;
    transform: translateX(50%);
    left: -54%;
}

.hero__layer {
    position: absolute;
    width: 110%;
    height: 100vh;
```

```
z-index: -1;
object-fit: cover;
top: 0;
left: -2%;
perspective: 10px;
}
.hero__header {
  font-weight: bold;
  position: absolute;
  text-align: center;
  height: 20vh;
  width: 100vw;
  left: 0;
  font-size: 13vmin;
  color: white;
  opacity: 0.3;
  transform: translateZ(-50px) translateY(-1400px) scale(6);
  text-transform: uppercase;
}

.about {
  width: 100vw;
  position: relative;
  padding-top: 100px;
  z-index: 1;
  min-height: 100vh;
  top: 99.9vh;
  background-color: #060303;
}

.about__container {
  display: flex;
  flex-flow: row-reverse;
  justify-content: center;
  z-index: 300;
  padding: 100px;
  padding-bottom: 400px;
  gap: 100px;
  transform-origin: bottom;
  margin: auto;
}

.about__header {
  color: white;
  font-size: 56px;
```

```
text-transform: uppercase;
animation: fade-in 1s forwards;
animation-delay: 2s;
opacity: 0;
}

.about__description {
  font-size: 24px;
  color: white;
  text-align: center;
  padding: 50px;
  background-image: linear-gradient(#ffffff 0%, #707070 100%);
  background-clip: text;
  -webkit-background-clip: text;
  -webkit-text-fill-color: transparent;
  margin-top: 100px;
  opacity: 0;
}

.about__head {
  padding-left: 50px;
  min-width: 525px;
  min-height: 400px;
  transform-origin: bottom;
}

.corners {
  position: relative;
}

.corners::before, .corners::after {
  content: "";
  display: block;
  position: absolute;
  width: 50px;
  height: 50px;
  visibility: hidden;
  box-sizing: border-box;
}

.corners::after {
  right: 0px;
  top: 0px;
  border-top: #ae4819 solid 10px;
  border-right: #ae4819 solid 10px;
}

.corners::before {
```

```
    bottom: 0px;
    border-left: #ae4819 solid 10px;
    border-bottom: #ae4819 solid 10px;
}

.animated .corners::before {
    animation: corner2 2s forwards;
    animation-delay: 1s;
}

.animated .corners::after {
    animation: corner1 2s forwards;
    animation-delay: 1s;
}

.animated .about__description {
    animation: pop-up 1s forwards;
}

.animated .head {
    animation: jump-in 1s forwards;
}

.head {
    width: 100%;
    position: relative;
    opacity: 0;
    transform-origin: center;
}

.head::after {
    display: block;
    content: "";
    width: 450px;
    position: absolute;
    height: 450px;
    border-radius: 50%;
    background: radial-gradient(at 0% 0%, #ffaf1b, #97361a, #2c1413
100%);
    top: 150px;
    left: 0px;
    z-index: -1;
}

.head__image {
    position: absolute;
    left: 50px;
    top: 0px;
```



```
    height: 600px;
    z-index: 150;
}
.head__eyes {
    position: absolute;
    left: 150px;
    top: 208px;
    z-index: 350;
    display: flex;
    width: 500px;
    gap: 100px;
}
.head__eye {
    position: relative;
    border-radius: 50%;
    background-color: #c7c7c7;
    position: relative;
    width: 35px;
    height: 35px;
    z-index: 400;
}
.head__eye:nth-child(2) {
    top: -9px;
}
.head__eye__middle {
    position: absolute;
    border-radius: 50%;
    top: 50%;
    left: 10px;
    width: 25px;
    height: 25px;
    background-color: black;
    transform: translate(-50%, -50%);
    border: 5px solid #005500;
    box-sizing: border-box;
}
.head__words {
    height: 100vh;
    transform-origin: center;
    overflow: visible !important;
    perspective: 10px;
    perspective-origin: center;
    transform: rotateX(70deg);
```

```
}  
.head__word {  
  font-size: 25px;  
  position: absolute;  
  display: inline-block;  
  width: 120px;  
  font-weight: 600;  
  color: #d55114;  
}  
  
.projects {  
  height: 140vh;  
  width: 100vw;  
  background: radial-gradient(at 0% 0%, #ffa1b, #97361a, #2c1413  
100%);  
  background-position: top;  
  transform: translateZ(-10px) scale(2);  
  position: relative;  
  top: 110vh;  
}  
  
.projects__container {  
  position: absolute;  
  left: 10%;  
  top: 40%;  
  transform: translate(0%, -50%) translateZ(5px);  
  display: flex;  
  transform-origin: left;  
  gap: 10vmin;  
}  
  
.projects__header {  
  font-size: 60px;  
  position: absolute;  
  bottom: 0;  
  visibility: hidden;  
  left: 100px;  
  text-transform: uppercase;  
}  
  
.projects__header[class*=animated] {  
  animation: slideFromRight 1s forwards;  
  animation-delay: 0.5s;  
}  
  
.card {
```

```
height: 56vmin;
width: 40vmin;
transform: scale(0.9);
object-fit: cover;
-webkit-backdrop-filter: blur(2px);
overflow: hidden;
color: white;
font-size: 30px;
user-select: none;
font-weight: bold;
position: relative;
border-radius: 10px;
}
.card__container {
  position: absolute;
  top: 0;
  width: 100%;
  height: 180%;
  background: linear-gradient(0deg, rgba(0, 0, 0, 0.8) 50%, rgba(0, 0, 0, 0) 100%);
  transition: 0.5s ease-out;
  overflow: hidden;
  backdrop-filter: blur(2px);
}
.card__container:hover {
  transition: 0.5s ease-out;
  height: 100%;
}
.card__container:hover .card__header {
  transition: 0.5s ease-out;
  margin-top: 40%;
}
.card__header {
  font-size: 33px;
  color: white;
  margin-top: 125%;
  font-weight: 400;
  margin-bottom: 30px;
  text-align: center;
  transition: 0.5s ease-out;
}
.card__image {
  height: 100%;
```

```
background-position: 0% center;
background-size: cover;
filter: drop-shadow(2);
}

.card__description {
margin: 20px;
font-size: 20px;
color: white;
font-weight: 400;
text-align: center;
}

.card__technologies {
text-align: center;
}

.wave {
width: 100vw;
height: 10vh;
position: absolute;
object-fit: cover;
overflow: hidden;
pointer-events: none;
}

.wave--top {
bottom: -9vh;
}

.wave--bottom {
transform: rotate(180deg);
top: -9vh;
}

.contact {
width: 100vw;
position: relative;
padding-top: 100px;
z-index: 1;
top: 50vh;
width: 100vw;
min-height: 100vh;
background-color: #060303;
}
```

```
.contact__container {
  min-height: 60vh;
  padding: 80px;
  display: flex;
  flex-flow: row;
  gap: 150px;
  justify-content: center;
  align-items: center;
  flex-wrap: wrap;
}

.contact__icon {
  font-size: 120px;
  color: white;
  transform: rotate(-45deg);
  text-align: center;
}

.contact__header {
  text-align: center;
  font-size: 60px;
}

.contact__option {
  position: relative;
  transform: rotate(45deg);
  padding: 40px;
  background-color: #303030;
  display: flex;
  align-items: center;
  justify-content: center;
  border-right: 10px #202020 solid;
  border-bottom: 10px #505050 solid;
  transition: 0.5s;
  cursor: pointer;
}

.contact__option:hover {
  transition: 0.5s;
  transform: translateY(-50px) rotate(45deg);
  box-shadow: 50px 50px 100px 0px #ae4819aa;
  background-color: #ae4819;
  border-right: 10px #c04f30 solid;
  border-bottom: 10px #97361a solid;
}

@media (max-width: 1200px) {
```

```
.about {
  flex-direction: column;
}
}
@media only screen and (max-width: 1000px) {
  #main-layer {
    left: -46%;
  }

  .about {
    padding-top: 0px;
  }
}
@media only screen and (max-width: 1600px) {
  .about__container {
    flex-flow: column;
    align-items: center;
    gap: 0px;
    padding-top: 0px;
    padding: 50px;
  }

  .about__head {
    margin-bottom: 300px;
  }

  .about__head {
    width: 420px;
  }

  .about__description {
    margin-top: 0px;
  }
}
@media only screen and (max-width: 1650px) {
  .layer7 {
    transform: translateY(3vh);
  }
}
@media only screen and (max-width: 600px) {
  #main-layer {
    left: -33%;
  }
}
```

```
.contact__container {
  padding-top: 100px;
  flex-flow: column;
}

.projects__header {
  font-size: 45px;
  left: 30px;
}

.about__description {
  font-size: 18px;
}

.about__head {
  transform: scale(0.6) !important;
  margin-top: 50px;
}

.about-middle {
  margin-bottom: 50px;
}
}
@media only screen and (max-width: 750px) {
  .card {
    height: 97vmin;
    width: 70vmin;
  }
}
.text--orange {
  color: #ae4819;
}

.text--white {
  color: white;
}

@keyframes jump-in {
  0% {
    transform: translateX(150px);
    opacity: 0;
  }
}
```

```
100% {  
    transform: translateX(0px);  
    opacity: 1;  
}  
}  
@keyframes corner1 {  
    0% {  
        visibility: visible;  
        width: 0%;  
        left: 0vw;  
        height: 10px;  
    }  
    45% {  
        width: 100%;  
        left: 0%;  
        height: 10px;  
    }  
    90% {  
        width: 50px;  
        left: calc(100% - 50px);  
    }  
    100% {  
        height: 50px;  
        visibility: visible;  
    }  
}  
@keyframes corner2 {  
    0% {  
        visibility: visible;  
        width: 0%;  
        right: 0vw;  
        height: 10px;  
    }  
    45% {  
        width: 100%;  
        right: 0%;  
        height: 10px;  
    }  
    90% {  
        width: 50px;  
        right: calc(100% - 50px);  
    }  
    100% {
```



```
    height: 50px;
    right: calc(100% - 50px);
    visibility: visible;
  }
}
@keyframes pop-up {
  0% {
    opacity: 0;
    transform: translateX(-150px);
  }
  100% {
    opacity: 1;
    transform: translateX(0px);
  }
}
@keyframes fade-in {
  0% {
    opacity: 0;
  }
  100% {
    opacity: 1;
  }
}
@keyframes jump-in {
  0% {
    transform: translateX(150px);
    opacity: 0;
  }
  100% {
    transform: translateX(0px);
    opacity: 1;
  }
}
@keyframes slideFromRight {
  0% {
    visibility: visible;
    transform: translateX(100vw);
  }
  100% {
    visibility: visible;
    transform: translateX(0);
  }
}
```

```
html {  
  background-color: black;  
  overflow: hidden;  
}  
  
* {  
  padding: 0;  
  margin: 0;  
  font-family: "Roboto", sans-serif;  
}  
  
/*# sourceMappingURL=style.css.map */
```

Skrypty

```
Array.from(document.getElementsByClassName("nav__list__item"))
  .forEach((item, index) => {
    item.onmouseover = () => {
      nav.dataset.activeIndex = index;
    }
  })

navButton.onclick = () => {
  if (!activeNav) {
    nav.dataset.active = true;
    activeNav = true;
  } else {
    nav.dataset.active = false;
    activeNav = false;
  }
}
```

Powyższy fragment kodu odpowiada za obsługę nawigacji czyli wyświetlenie jej oraz schowanie a także przypisanie data setów w celu późniejszego zaanimowania przesuwania się ikon w tle

```
const x = 250;
const y = 150;
const r = 75;
let a = 0;

const rotate = (a, obj) => {
  const px = x + r * 3.5 * Math.cos(a);
  const py = y + r * Math.sin(a);

  obj.style.left = px - obj.offsetWidth / 2 + 'px';
  obj.style.top = py - obj.offsetHeight / 2 + 'px';
  obj.style.transform = `scale(${(y + 150) / (py + 150)})`;
  obj.style.zIndex = py > y ? 100 : 200;
};

const wordWidth = words[0].offsetWidth;
const distance = wordWidth / (2 * Math.PI * r);

const updateAnimation = () => {
  a = (a + distance/10) % (2 * Math.PI);
  for (let i = 0; i < words.length; i++) {
    rotate(i * 2 * Math.PI / words.length - a, words[i]);
  }
  requestAnimationFrame(updateAnimation);
};

updateAnimation();
```

Ten fragment kodu odpowiada zaanimowanie słów krążących dookoła głowy po elipsie. Wykorzystujący funkcje matematyczne i szerokość słów obliczana jest pozycja danego słowa względem punktu (x,y)

```

document.addEventListener('mousemove', eyeball);

function eyeball(event) {
  const eyes = document.querySelectorAll('.head__eye');
  eyes.forEach(eye => {
    const { left, top, width, height } = eye.getBoundingClientRect();
    const x = left + width / 2;
    const y = top + height / 2;

    const radian = Math.atan2(event.pageX - x, event.pageY - y);
    const rotate = radian * (180 / Math.PI) * -1 + 270;
    eye.style.transform = `rotate(${rotate}deg)`;
  });
}

```

Powyższy kod odpowiada za animacje oczu. Pobiera pozycje oczu i kursora na stronie następnie poprzez funkcje `atan2` wylicza kąt o jaki należy obrócić dane oko

```

73
74 let cardsContainer = document.querySelector(".projects__container");
75 let projects = document.querySelector(".projects");
76 cardsContainer.dataset.prevPercentage = "0";
77
78
79 Array.from(cardsContainer.getElementsByClassName("card__image"))
80   .forEach((item, index) => {
81     item.style.backgroundColor = `url(img/project${index}.png)`;
82   })
83
84
85 projects.onmousedown = e => {
86   cardsContainer.dataset.mouseDownAt = e.clientX;
87 }
88 projects.onmouseup = e => {
89   cardsContainer.dataset.mouseDownAt = "0";
90   cardsContainer.dataset.prevPercentage = cardsContainer.dataset.percentage;
91 }
92
93 projects.onmousemove = e => {
94
95   if (cardsContainer.dataset.mouseDownAt === "0") return;
96
97   const mouseDelta = parseFloat(cardsContainer.dataset.mouseDownAt) - e.clientX;
98   maxDelta = window.innerWidth / 2;
99
100   const percentage = (mouseDelta / maxDelta) * 100;
101   const nextPercentage = Math.min(Math.max(parseFloat(cardsContainer.dataset.prevPercentage) + percentage, 0), 50);
102
103   cardsContainer.dataset.percentage = nextPercentage;
104
105   cardsContainer.style.transform = `translate(${nextPercentage}%, -50%)`;
106
107   cardsContainer.animate([
108     { transform: `translate(${nextPercentage}%, -50%)` },
109   ], { duration: 1000, fill: 'forwards' });
110
111   for (const image of cardsContainer.getElementsByClassName('card__image')) {
112     image.animate([
113       { backgroundPosition: `${nextPercentage}% center` },
114     ], { duration: 1000, fill: 'forwards' });
115   }
116 }
117
118
119

```

Ten fragment kodu odpowiada za animacje projektów podczas ich przesuwania. po kliknięciu myszki zapisuje pozycje a następnie podczas jej ruchu oblicza o ile i w która stronę się przesunęła aby zaktualizować pozycje list projektów a także i obrazy w tle.

```
//FIRECAMP ANIMATION

for (let i = 1; i <= 3; i++) {
  let tween = KUTE.fromTo(
    '#fire' + i + '-1',
    { path: '#fire' + i + '-1' },
    { path: '#fire' + i + '-3' },
    { duration: i == 3 ? 1000 : 100, yoyo: true, }
  )

  let tween2 = KUTE.fromTo(
    '#fire' + i + '-1',
    { path: '#fire' + i + '-3' },
    { path: '#fire' + i + '-2' },
    { duration: i == 3 ? 1000 : 100, }
  )

  let tween3 = KUTE.fromTo(
    '#fire' + i + '-1',
    { path: '#fire' + i + '-2' },
    { path: '#fire' + i + '-1' },
    { duration: i == 3 ? 1000 : 100, }
  )
  tween.chain(tween2);
  tween2.chain(tween3);
  tween3.chain(tween);

  tween.start();
}
let tweenLight = KUTE.fromTo(
  '#fire4-1',
  { opacity: 0.75 },
  { opacity: 1 },
  { duration: 150, yoyo: true, repeat: 9999 }
)
tweenLight.start();
```

```
let typingSpeed = 75;

let leftHandAnimation = KUTE.fromTo(
  '#hand1-1',
  { path: '#hand1-1' },
  { path: '#hand1-2' },
  { duration: typingSpeed, yoyo: true, repeat: 999 }
)

let rightHandAnimation = KUTE.fromTo(
  '#hand2-1',
  { path: '#hand2-1' },
  { path: '#hand2-2' },
  { duration: typingSpeed, yoyo: true, repeat: 999 }
)

let headAnimation = KUTE.fromTo(
  '#head1-1',
  { path: '#head1-1' },
  { path: '#head1-2' },
  { duration: 500, yoyo: true }
)

let headAnimation2 = KUTE.fromTo(
  '#head2-1',
  { path: '#head2-1' },
  { path: '#head2-2' },
  { duration: 500, yoyo: true }
)
```

Te fragmenty kodu odpowiadają za stworzenie animacji w rąk, głowy oraz ogniska. Do ich stworzenia wykorzystana została biblioteka KUTE.JS. Biblioteka ta pozwala na płynne przejście kształtu svg do innego i w ten oto sposób kształt ognia zmienia się z 1 na 3, z 3 na 2 a następnie z 2 na 1 i tak w kółko dając pozór skaczącego ognia. Podobnie z pozostałymi animacjami

```

✓ async function typingAnimation(className, texts, speed) {
  const element = document.querySelector(className);
  ✓ for (let i = 0; i < texts.length - 1; i++) {
    const text = texts[i];

    leftHandAnimation.start();
    rightHandAnimation.start();
    console.log("start typing")
  ✓ for (const char of text) {
    element.innerHTML += char;
    await new Promise(resolve => setTimeout(resolve, speed));
  }
  leftHandAnimation.stop();
  rightHandAnimation.stop();
  await new Promise(resolve => setTimeout(resolve, 500));
  rightHandAnimation.start();
  ✓ for (let j = element.innerHTML.length - 1; j >= 0; j--) {
    element.innerHTML = element.innerHTML.substring(0, j);
    await new Promise(resolve => setTimeout(resolve, speed));
  }
  rightHandAnimation.stop();

}

const finalText = texts[texts.length - 1];
leftHandAnimation.start();
rightHandAnimation.start();
✓ for (const char of finalText) {
  element.innerHTML += char;
  await new Promise(resolve => setTimeout(resolve, speed));
}
leftHandAnimation.stop();
rightHandAnimation.stop();
headAnimation.reverse();
headAnimation2.reverse();
headAnimation.start();
headAnimation2.start();

}

✓ async function animations() {
  await new Promise(resolve => setTimeout(resolve, 2000));

  headAnimation.start();
  headAnimation2.start();
  await new Promise(resolve => setTimeout(resolve, 1000));
  typingAnimation(".hero__header", ["Hi I'm Łukasz Jasiński", "And this is my", "Portfolio"], typingSpeed);
}

animations();

```

Powyższy kod wykorzystując animacje stworzone wcześniej tworzy odpowiednia sekwencje animacji pisania najpierw uruchamia animację rak czeka aż słowo zostanie napisane, wyłącza animację rak, uruchamia animację prawej ręki (usuwanie) następnie literka po literce usuwa wcześniej wypisany tekst i pętla przechodzi do następnej iteracji wypisując i usuwając kolejne słowo do czasu aż dojdzie do ostatniego słowa wtedy tekst nie jest usuwany oraz animacja głowy jest odwracana dzięki czemu głowa wraca do swojej normalnej pozycji

```

249
250 const targets = document.querySelectorAll('.trigger');
251
252 const observer = new IntersectionObserver((entries) => {
253   entries.forEach(entry => {
254     if (entry.isIntersecting) {
255       entry.target.classList.add('animated');
256     }
257   });
258 }, {
259   rootMargin: '50%',
260   threshold: [0.5]
261 });
262
263 targets.forEach(target => {
264   observer.observe(target);
265 });
266
267
268
269

```

Ostatni już fragment kodu wykorzystując Intersection Observer dodaje do sekcji klasę animated gdy pojawi się ona w 50% na ekranie dzięki czemu elementy w niej zawarte zostaną zaanimowane

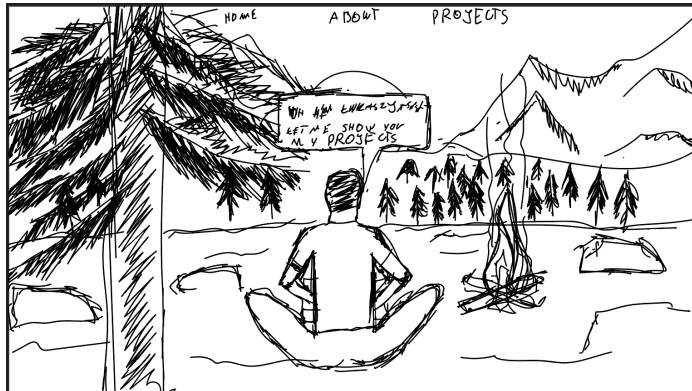
Dodatkowe informacje

Strona nie jest jeszcze w 100% ukończona i wymaga wielu poprawek oraz refaktoryzacji.

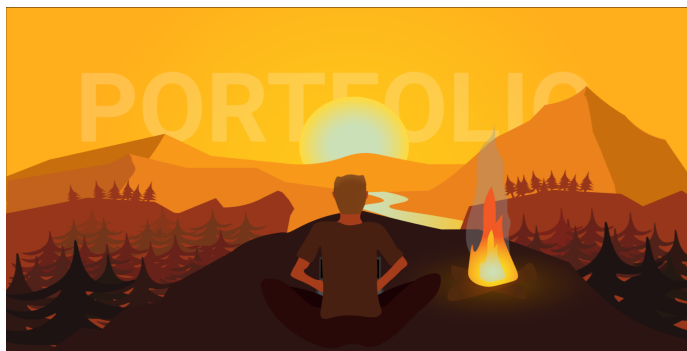
Grafiki wykorzystane na stronie(poza ikonami) zostały stworzone przeze mnie w programie Adobe Illustrator

Inspiracja Firewatch

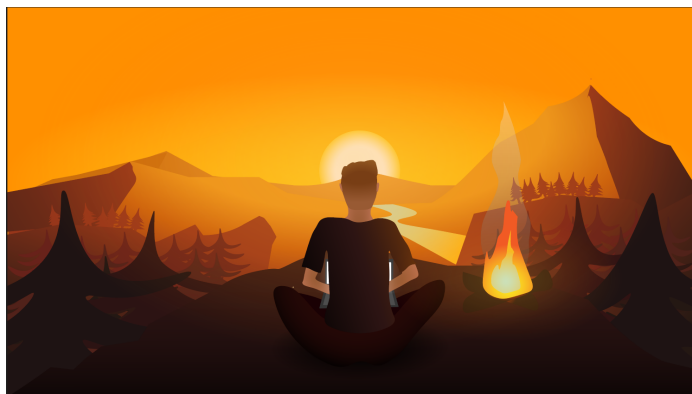
Concept Art



Pierwsza wersja



Wersja finałowa



Grafika była tworzona warstwami dzięki czemu później mógł zostać osiągnięty efekt paralaksy.