



University of Rzeszów

Internet Applications 1

MY ERASMUS EXPERIENCE

Lecturer

Piotr Lasek

First Name - Last Name

Elif Nur Tabaklı

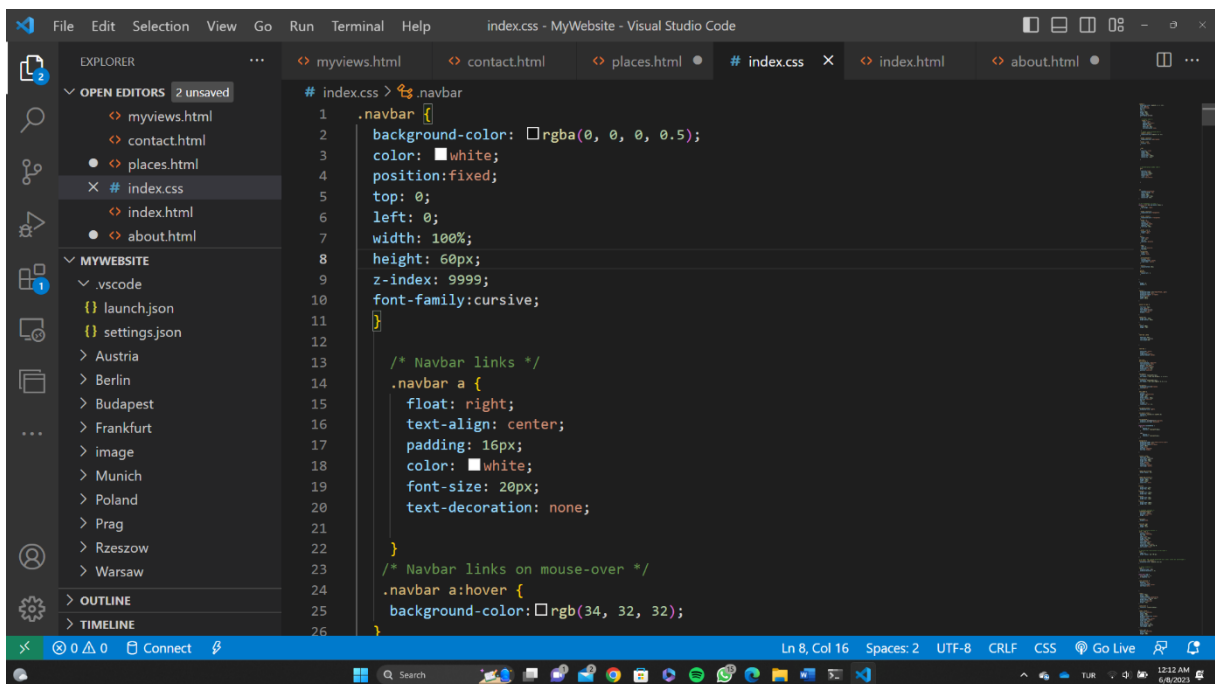
HTML Navigation Bar with Page Links and Logo Image

The HTML code snippet provided is a navigation bar (navbar) with links to Contact, MyViews, Places, About, and Home pages. It also includes a logo image for the website or application. It serves as a user-friendly way to navigate through different sections of the site.

```
<div class="navbar" id="myTopnav" >

  <a id="nav" href="#">DiaryOfErasmus</a>
  <a href="contact.html"> Contact</a>
  <a href="myviews.html"> MyViews</a>
  <a href="places.html"> Places</a>
  <a href="about.html"> About</a>
  <a class="active" href="#index">Home</a>

</div>
```



Introduction to My Erasmus Experience: Get Ready for an Exciting Journey!

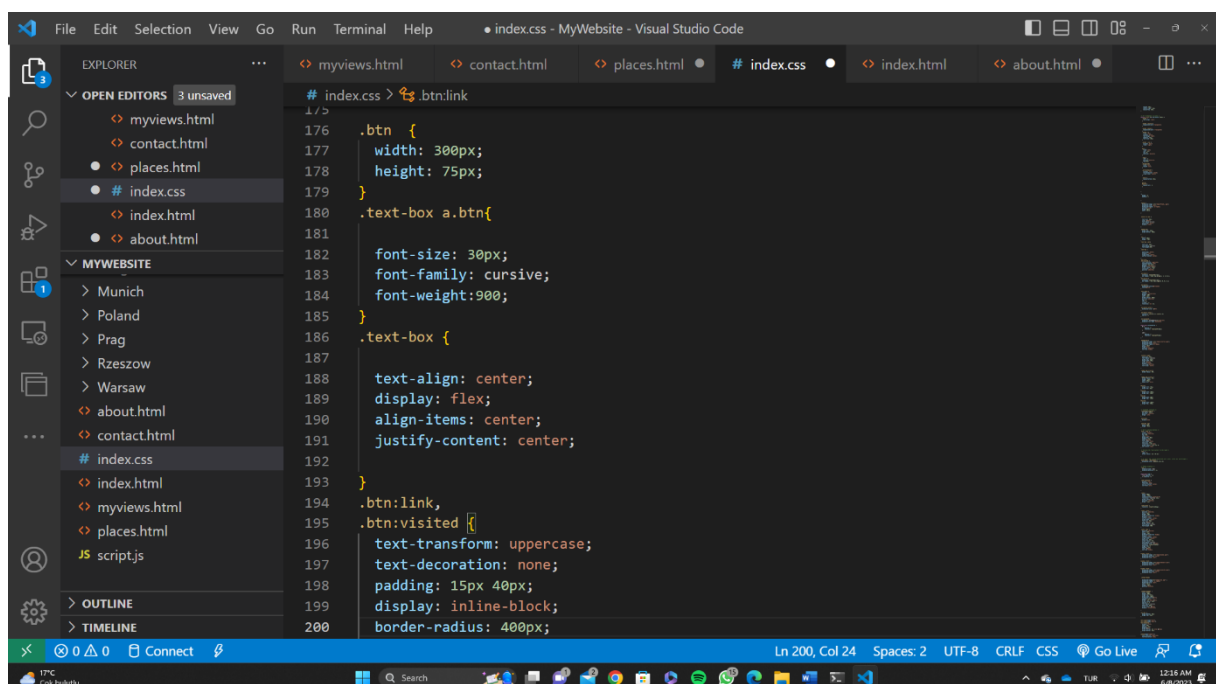
The provided code snippet represents an HTML markup for a banner section on a webpage. The banner includes a heading and a text box containing a button. The heading consists of two spans: the first span saying "Hi Everyone!" and the second span asking if the readers are ready to hear about the author's Erasmus experience. The text box contains a button with the label "LET'S START" and a white background. Clicking on the button will redirect the user to the "about.html" page.

```
<div class="banner">

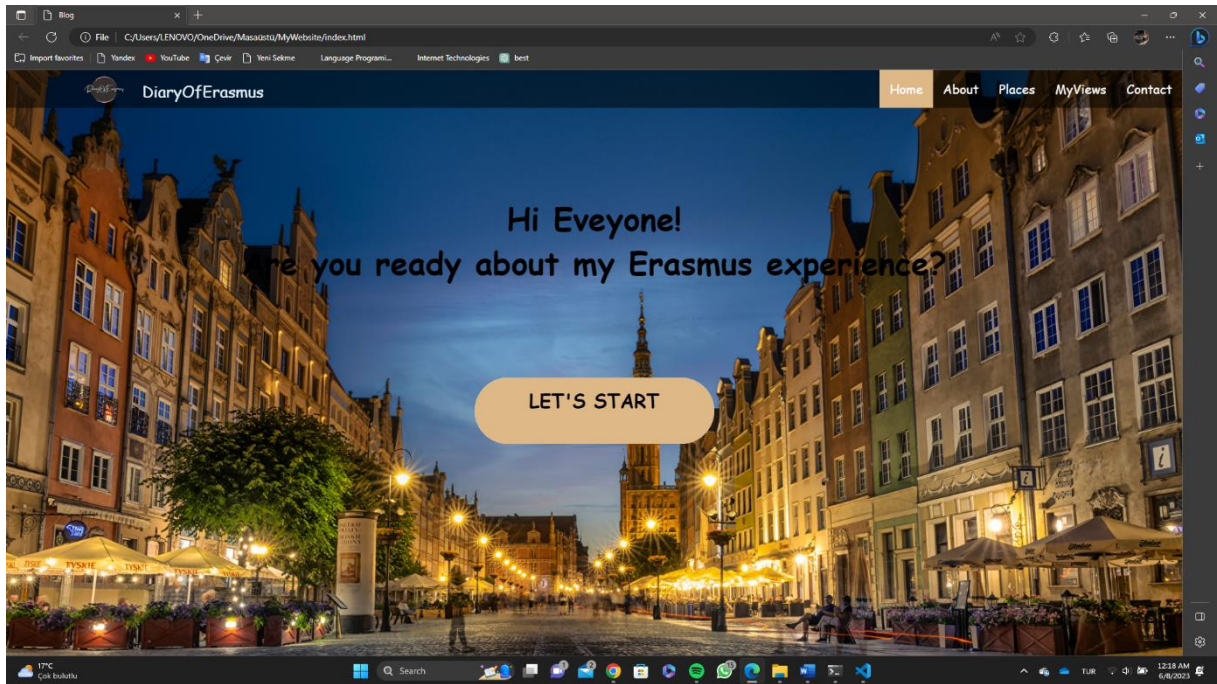
    <h1>
        <span>Hi Everyone! </span>
        <span>Are you ready about my Erasmus experience?</span>
    </h1>

    <div class="text-box">
        <a href="about.html" class="btn btn-white btn-animate">LET'S START</a>
    </div>

</div>
```

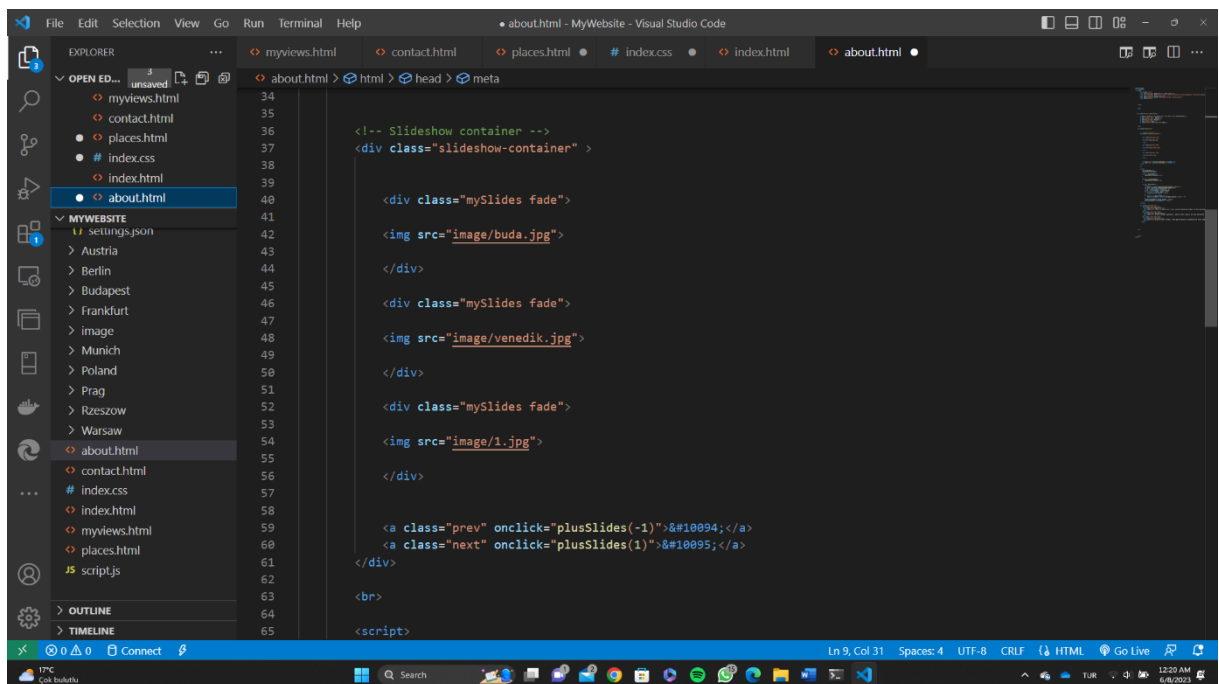


Index.html page



Captivating Slideshow: Explore Stunning Images in this Dynamic Presentation

The given code snippet represents a slideshow container section in HTML. It includes multiple slides with fading transitions. Each slide contains an image sourced from different files within the "image" folder. The slideshow has navigation arrows, a previous button with the symbol "<" and a next button with the symbol ">". These buttons are assigned the functions "plusSlides(-1)" and "plusSlides(1)" respectively, which allow users to navigate through the slideshow by moving to the previous or next slide.



```
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

<!-- Slideshow container -->
<div class="slideshow-container">

  <div class="mySlides fade">

  </div>

  <div class="mySlides fade">

  </div>

  <div class="mySlides fade">

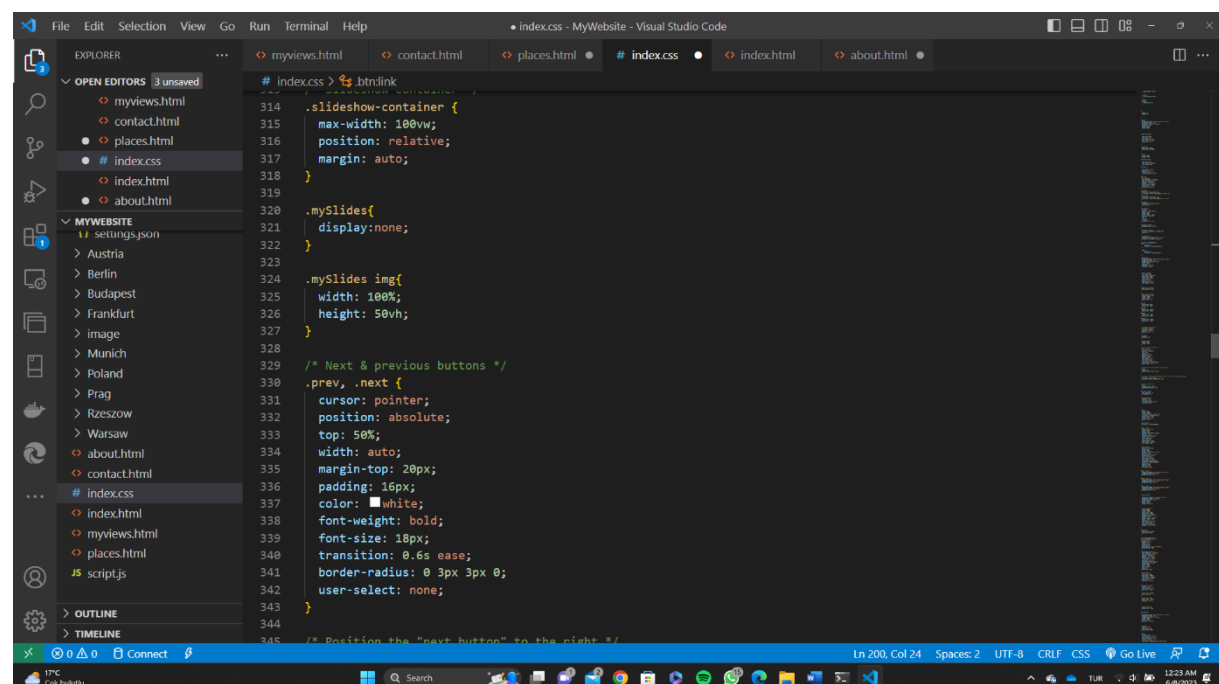
  </div>

  <a class="prev" onclick="plusSlides(-1)">&#10094;</a>
  <a class="next" onclick="plusSlides(1)">&#10095;</a>

</div>

<br>

<script>
```



```
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345

/* Slideshow container */
.slideshow-container {
  max-width: 100vw;
  position: relative;
  margin: auto;
}

.mySlides {
  display: none;
}

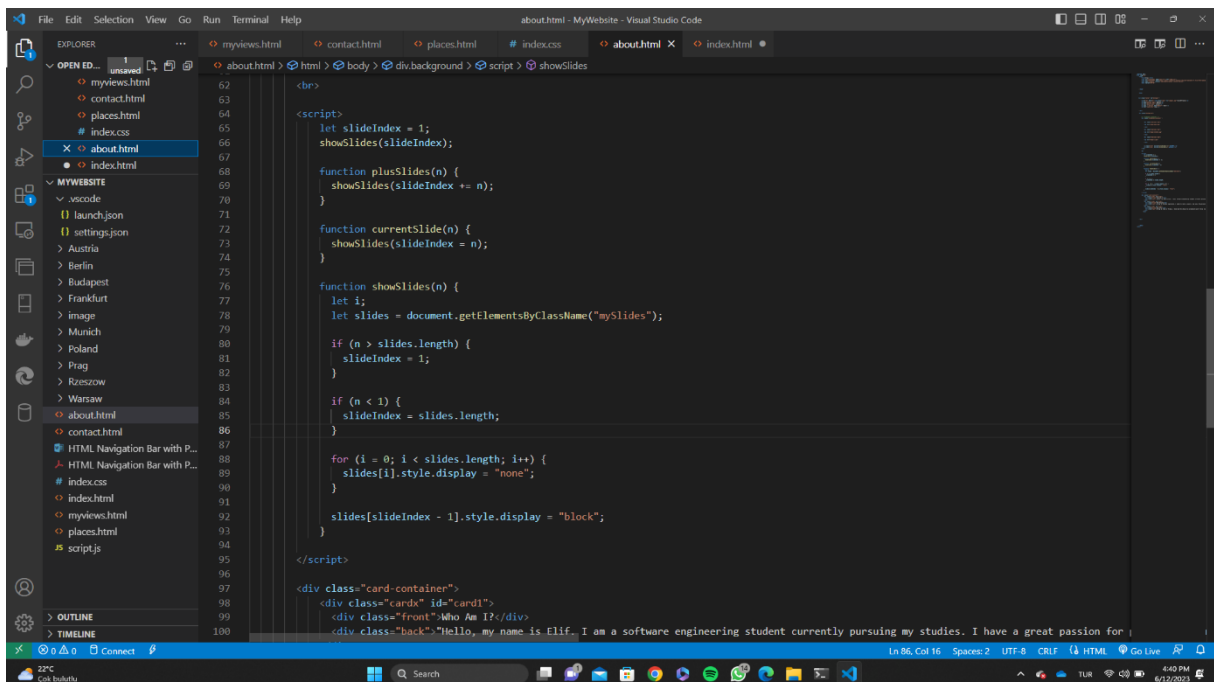
.mySlides img {
  width: 100%;
  height: 50vh;
}

/* Next & previous buttons */
.prev, .next {
  cursor: pointer;
  position: absolute;
  top: 50%;
  width: auto;
  margin-top: 20px;
  padding: 16px;
  color: white;
  font-weight: bold;
  font-size: 18px;
  transition: 0.6s ease;
  border-radius: 0 3px 3px 0;
  user-select: none;
}

/* Position the "next button" to the right */
```

Interactive Slideshow Controller: Effortlessly Navigate and Display Captivating Images

The provided code snippet serves as an interactive slideshow controller in JavaScript. It enables smooth navigation and display of captivating images within a slideshow. The script initializes a variable, `slideIndex`, which represents the current slide being shown. The `showSlides()` function is responsible for handling the logic of displaying the appropriate slide based on the given index. The `plusSlides(n)` function allows users to move forward or backward through the slideshow by incrementing or decrementing the `slideIndex`. Additionally, the `currentSlide(n)` function allows users to directly jump to a specific slide by setting the `slideIndex` to the given value. The script dynamically updates the display by manipulating the visibility of the slides using CSS properties. It also manages the active state of navigation dots corresponding to each slide. Overall, this script provides an efficient and interactive way to control and showcase captivating images within a slideshow.



```
62 <br>
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
<script>
  let slideIndex = 1;
  showSlides(slideIndex);

  function plusSlides(n) {
    showSlides(slideIndex += n);
  }

  function currentSlide(n) {
    showSlides(slideIndex = n);
  }

  function showSlides(n) {
    let i;
    let slides = document.getElementsByClassName("mySlides");

    if (n > slides.length) {
      slideIndex = 1;
    }

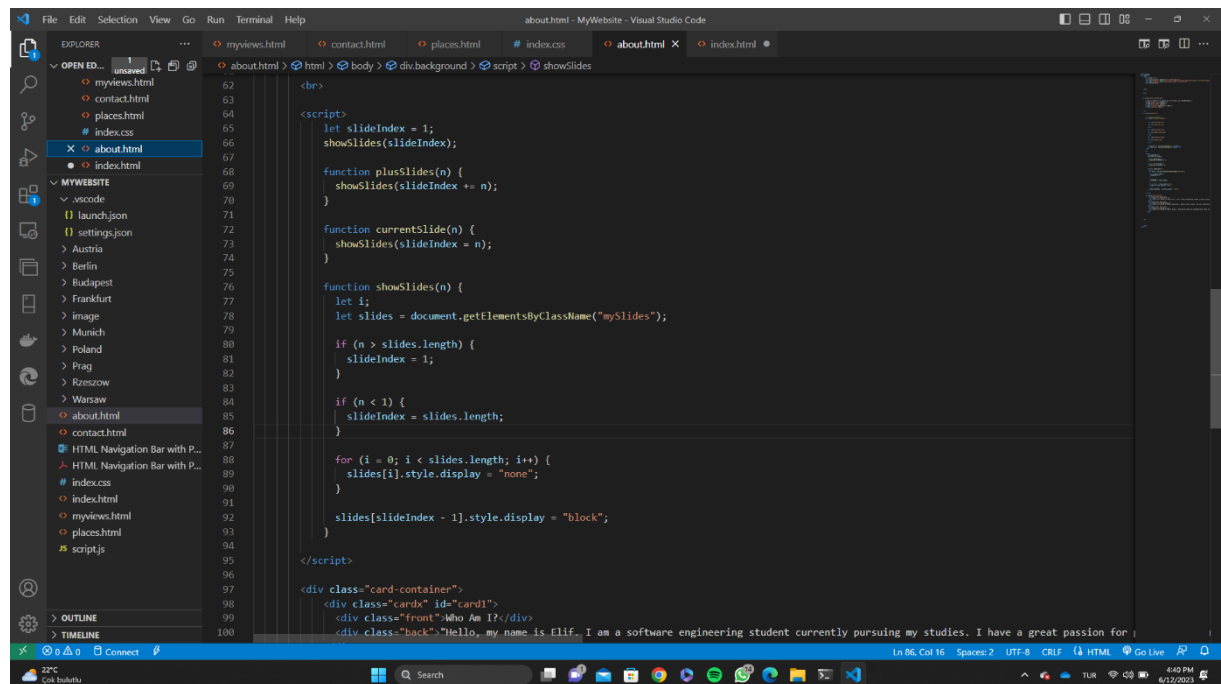
    if (n < 1) {
      slideIndex = slides.length;
    }

    for (i = 0; i < slides.length; i++) {
      slides[i].style.display = "none";
    }

    slides[slideIndex - 1].style.display = "block";
  }
</script>

<div class="card-container">
  <div class="card">
    <div class="front">Who Am I?</div>
    <div class="back">Hello, my name is Elif. I am a software engineering student currently pursuing my studies. I have a great passion for
```

Engaging Card Container: Exploring Personal Identity, Mission, and Erasmus Experience



The screenshot shows the Visual Studio Code editor with a project named 'MyWebsite'. The Explorer sidebar on the left shows a file structure with folders like 'Austria', 'Berlin', 'Budapest', 'Frankfurt', 'image', 'Munich', 'Poland', 'Prag', 'Rzeszow', and 'Warsaw', and files like 'about.html', 'contact.html', 'index.css', 'index.html', 'myviews.html', 'places.html', 'script.js', and 'settings.json'. The main editor area displays the 'about.html' file, which contains a JavaScript script for a card container. The script defines a 'showSlides' function that iterates through slides and updates their display status. The HTML structure includes a 'card-container' with three cards, each with a 'front' and 'back' face. The first card's back face contains the text: 'Hello, my name is Elif. I am a software engineering student currently pursuing my studies. I have a great passion for'.

```
<script>
let slideIndex = 1;
showSlides(slideIndex);

function plusSlides(n) {
  showSlides(slideIndex += n);
}

function currentSlide(n) {
  showSlides(slideIndex = n);
}

function showSlides(n) {
  let i;
  let slides = document.getElementsByClassName("mySlides");

  if (n > slides.length) {
    slideIndex = 1;
  }

  if (n < 1) {
    slideIndex = slides.length;
  }

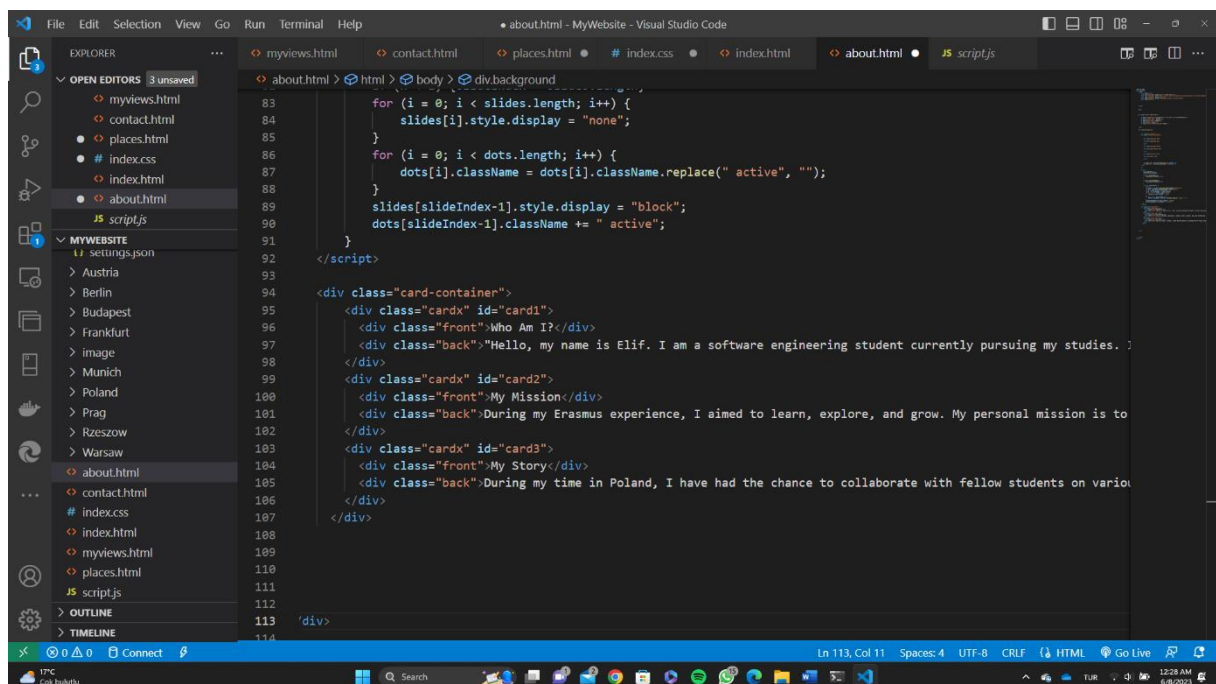
  for (i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
  }

  slides[slideIndex - 1].style.display = "block";
}

</script>

<div class="card-container">
  <div class="card" id="card1">
    <div class="front">Who Am I?</div>
    <div class="back">Hello, my name is Elif. I am a software engineering student currently pursuing my studies. I have a great passion for
  </div>
  <div class="card" id="card2">
    <div class="front">My Mission</div>
    <div class="back">During my Erasmus experience, I aimed to learn, explore, and grow. My personal mission is to
  </div>
  <div class="card" id="card3">
    <div class="front">My Story</div>
    <div class="back">During my time in Poland, I have had the chance to collaborate with fellow students on various
  </div>
</div>
```

This code snippet represents an HTML structure for a card container. The container consists of three cards, each having a front and back face. The purpose of these cards is to present information about the author's identity, mission, and Erasmus experience.



The screenshot shows the Visual Studio Code editor with the same project. The Explorer sidebar shows the 'script.js' file selected. The main editor area displays the 'script.js' file, which contains the JavaScript code for the card container. The code defines a 'showSlides' function that iterates through slides and updates their display status. The HTML structure includes a 'card-container' with three cards, each with a 'front' and 'back' face. The first card's back face contains the text: 'Hello, my name is Elif. I am a software engineering student currently pursuing my studies. I have a great passion for'.

```
for (i = 0; i < slides.length; i++) {
  slides[i].style.display = "none";
}

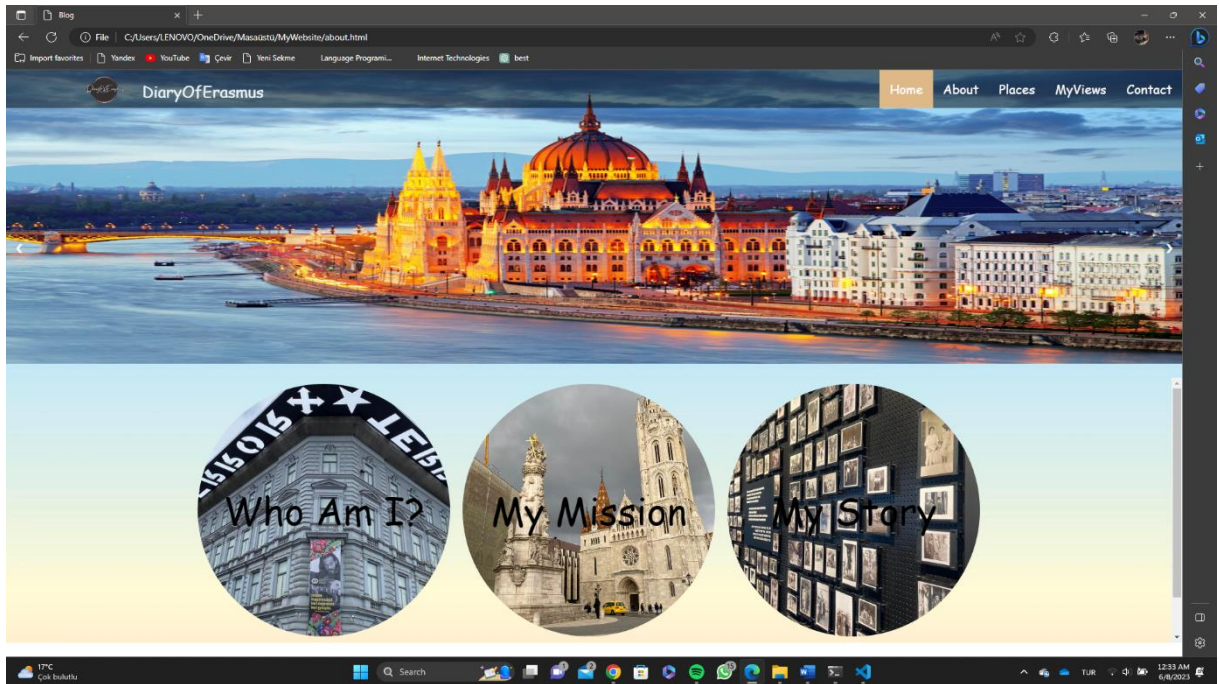
for (i = 0; i < dots.length; i++) {
  dots[i].className = dots[i].className.replace(" active", "");
}

slides[slideIndex-1].style.display = "block";
dots[slideIndex-1].className += " active";
}

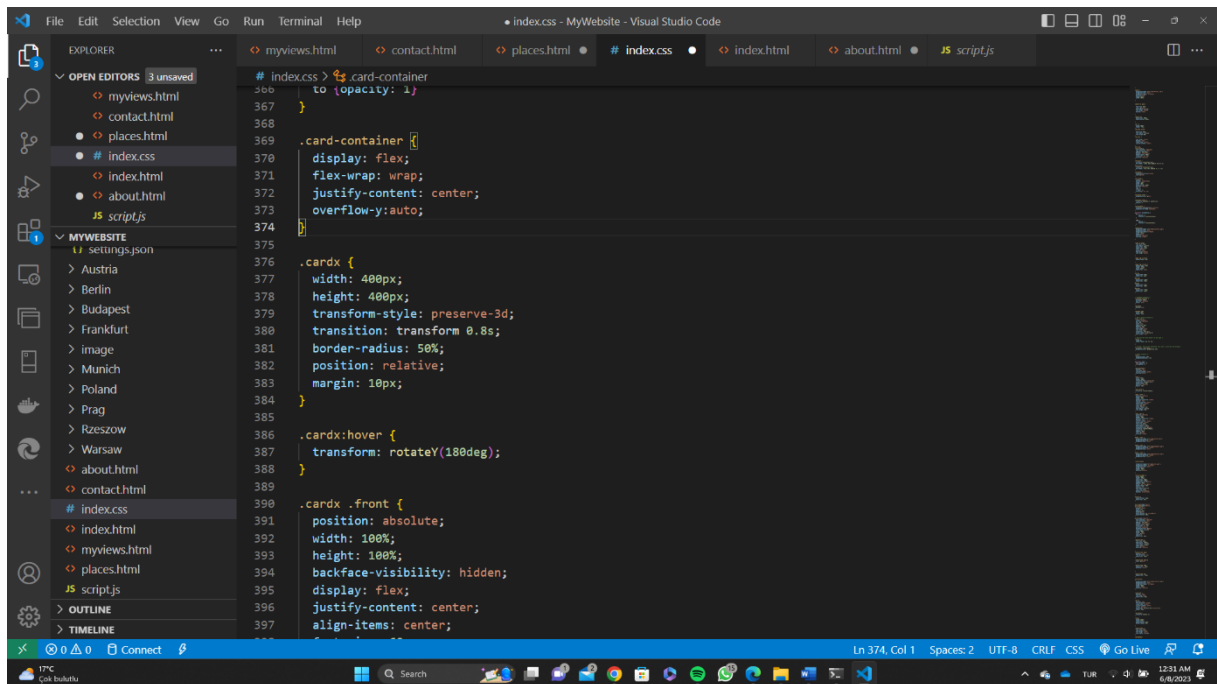
</script>

<div class="card-container">
  <div class="card" id="card1">
    <div class="front">Who Am I?</div>
    <div class="back">Hello, my name is Elif. I am a software engineering student currently pursuing my studies. I have a great passion for
  </div>
  <div class="card" id="card2">
    <div class="front">My Mission</div>
    <div class="back">During my Erasmus experience, I aimed to learn, explore, and grow. My personal mission is to
  </div>
  <div class="card" id="card3">
    <div class="front">My Story</div>
    <div class="back">During my time in Poland, I have had the chance to collaborate with fellow students on various
  </div>
</div>
```


About.html Page



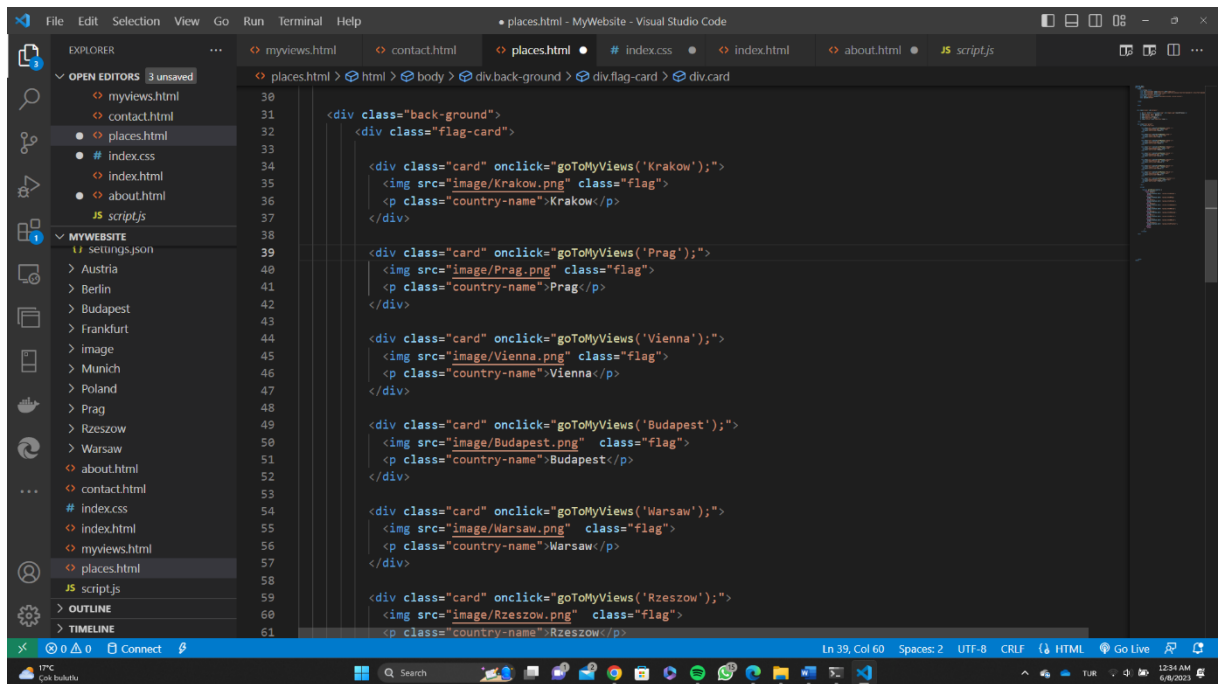
Interactive Flag Card Selection: Explore Various Destinations



The given code snippet represents an interactive flag card selection section in HTML. It consists of a container called "flag-card" containing multiple cards, each representing a different destination. Each card contains an image of the country's flag, along with the name of the city or country below it.

Clicking on a card triggers a JavaScript function called "goToMyViews()" with the corresponding destination as a parameter. This function likely navigates the user to specific views or pages associated with the selected destination.

The cards provide an engaging and visually appealing way for users to explore different destinations. By clicking on a card, users can access more information or content related to that particular location. This code snippet allows users to easily discover and navigate through various cities and countries for further exploration and engagement.



```
30
31
32 <div class="back-ground">
33   <div class="flag-card">
34     <div class="card" onclick="goToMyViews('Krakow');">
35       
36       <p class="country-name">Krakow</p>
37     </div>
38
39     <div class="card" onclick="goToMyViews('Prag');">
40       
41       <p class="country-name">Prag</p>
42     </div>
43
44     <div class="card" onclick="goToMyViews('Vienna');">
45       
46       <p class="country-name">Vienna</p>
47     </div>
48
49     <div class="card" onclick="goToMyViews('Budapest');">
50       
51       <p class="country-name">Budapest</p>
52     </div>
53
54     <div class="card" onclick="goToMyViews('Warsaw');">
55       
56       <p class="country-name">Warsaw</p>
57     </div>
58
59     <div class="card" onclick="goToMyViews('Rzeszow');">
60       
61       <p class="country-name">Rzeszow</p>
```

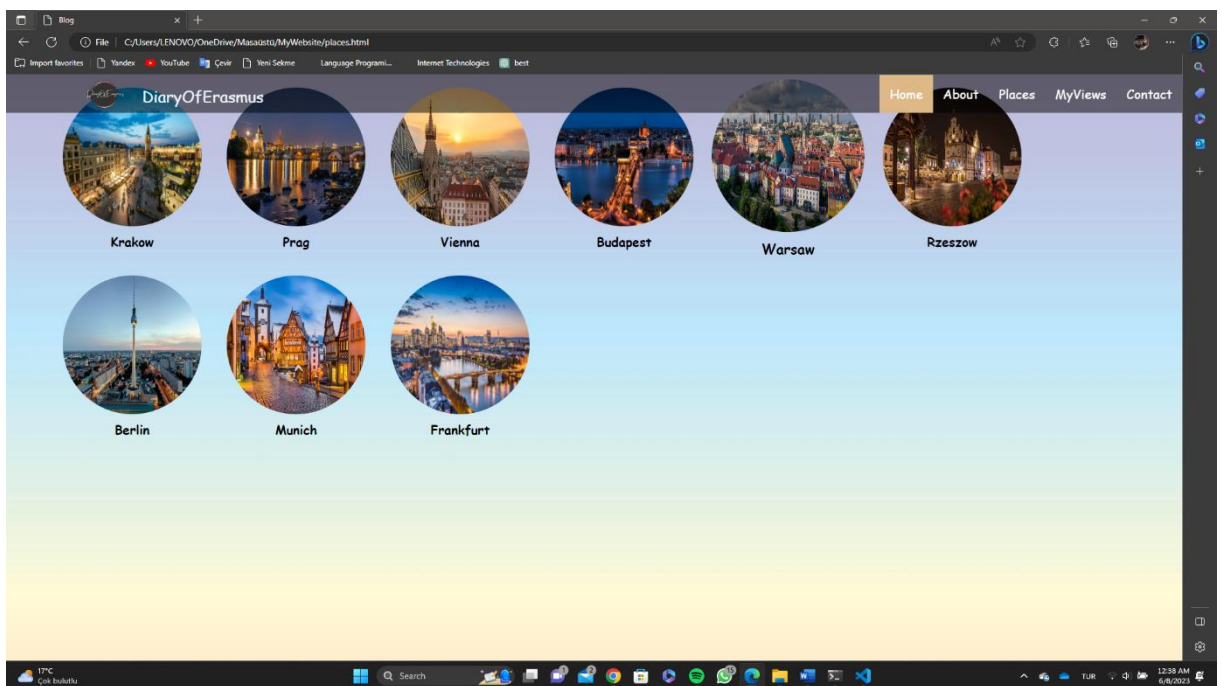
Interactive Flag Card Selection: Navigate to Destination Views

The provided code snippet introduces a JavaScript function called `goToMyViews(country)`, which facilitates seamless navigation to different destination views. The function is designed to be triggered when a flag card is clicked within an HTML structure.

The core functionality of the function involves utilizing a switch statement to determine the selected country. Based on the value of `country`, the function redirects the user to a specific URL using the `window.location.href` property. The URL

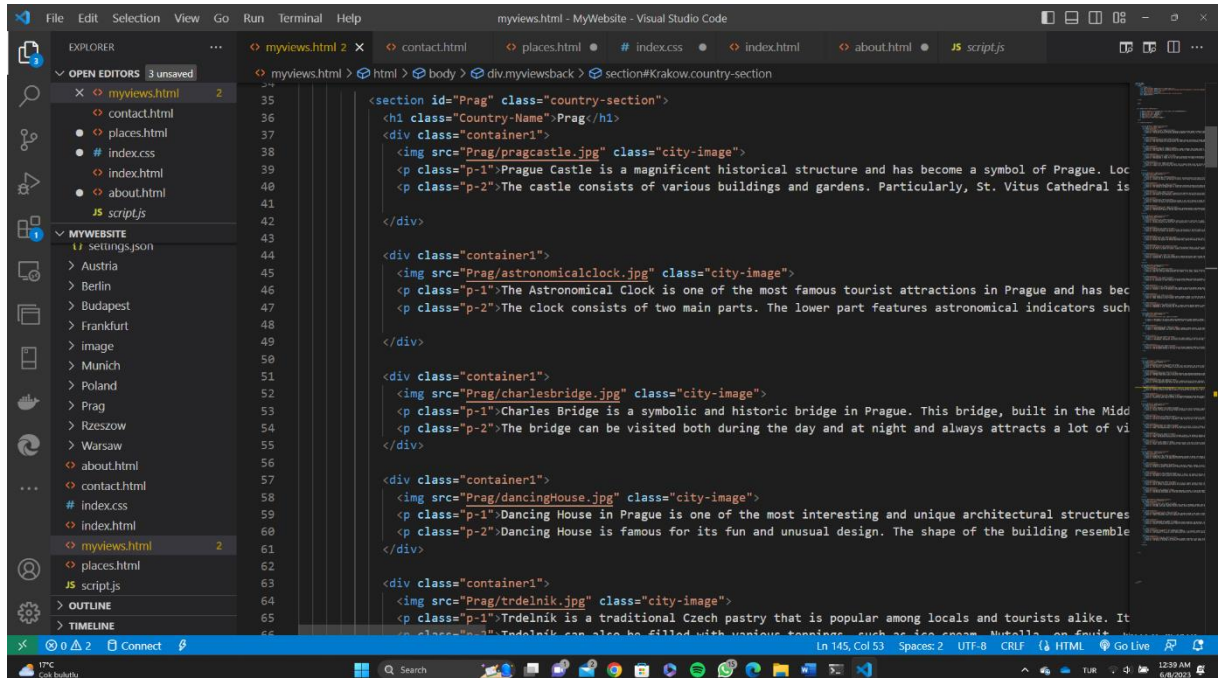
incorporates an anchor fragment corresponding to the selected country, allowing the user to access relevant destination-specific content or information.

This code snippet enhances the interactivity of the flag card selection by enabling users to effortlessly explore different destinations. By clicking on a card, users are promptly directed to the respective view associated with their chosen country. Such an implementation offers a user-friendly and intuitive means to navigate and discover compelling content related to various destinations.

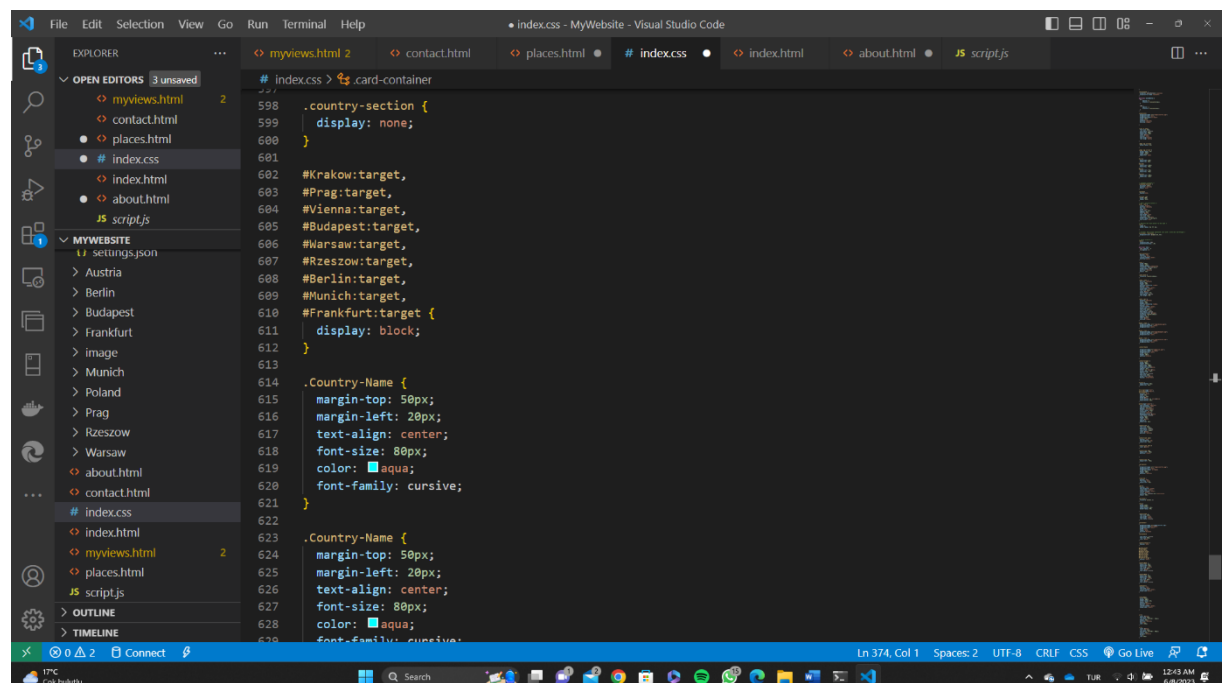


The purpose of the provided code is to create a section dedicated to Prague and showcase various aspects of the city, including its landmarks, attractions, and cultural highlights. The code utilizes HTML elements such as headings, paragraphs, and images to present the information in a structured manner.

Within the section, multiple containers (div elements with the class "container1") are used to group related content together. Each container consists of an image representing a specific aspect of Prague, followed by one or two paragraphs of text providing details and descriptions.



```
<section id="Prag" class="country-section">
  <h1 class="Country-Name">Prag</h1>
  <div class="container1">
    
    <p class="p-1">Prague Castle is a magnificent historical structure and has become a symbol of Prague. Loc
    <p class="p-2">The castle consists of various buildings and gardens. Particularly, St. Vitus Cathedral is
  </div>
  <div class="container1">
    
    <p class="p-1">The Astronomical Clock is one of the most famous tourist attractions in Prague and has bec
    <p class="p-2">The clock consists of two main parts. The lower part features astronomical indicators such
  </div>
  <div class="container1">
    
    <p class="p-1">Charles Bridge is a symbolic and historic bridge in Prague. This bridge, built in the Midd
    <p class="p-2">The bridge can be visited both during the day and at night and always attracts a lot of vi
  </div>
  <div class="container1">
    
    <p class="p-1">Dancing House in Prague is one of the most interesting and unique architectural structures
    <p class="p-2">Dancing House is famous for its fun and unusual design. The shape of the building resemble
  </div>
  <div class="container1">
    
    <p class="p-1">Trdelnik is a traditional Czech pastry that is popular among locals and tourists alike. It
    <p class="p-2">Trdelnik can also be filled with various toppings such as ice cream, Nutella, or fruit.
```



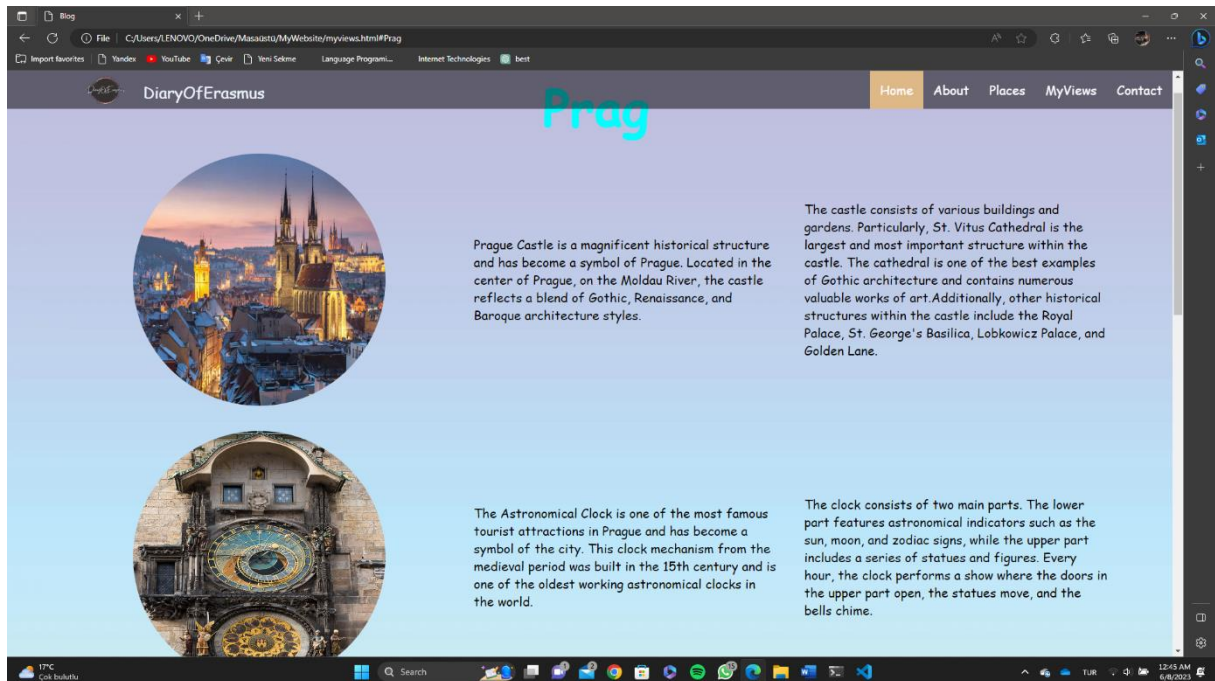
```
.country-section {
  display: none;
}

#Krakow:target,
#Prag:target,
#Vienna:target,
#Budapest:target,
#Warsaw:target,
#Rzeszow:target,
#Berlin:target,
#Munich:target,
#Frankfurt:target {
  display: block;
}

.Country-Name {
  margin-top: 50px;
  margin-left: 20px;
  text-align: center;
  font-size: 80px;
  color: aqua;
  font-family: cursive;
}

.Country-Name {
  margin-top: 50px;
  margin-left: 20px;
  text-align: center;
  font-size: 80px;
  color: aqua;
  font-family: cursive;
}
```

Myviews.html Page



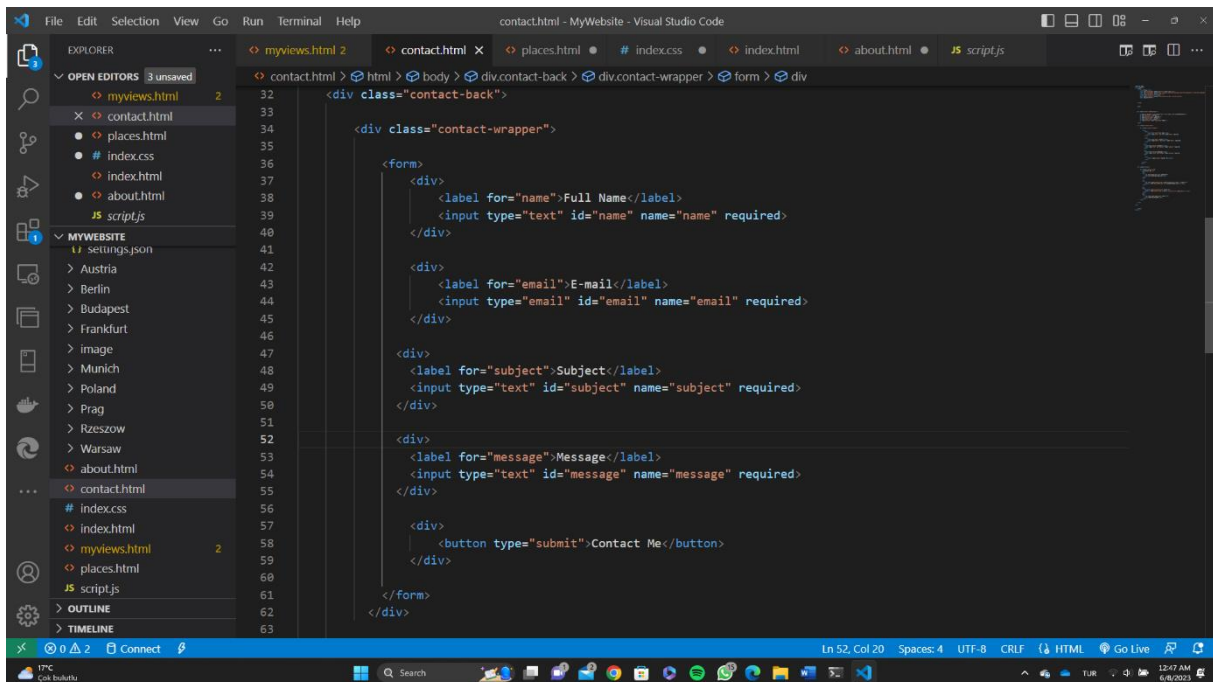
Contact Form for Seamless Communication

The provided code represents a contact form designed to facilitate seamless communication between website visitors and the website owner. The form is enclosed within a container with the class "contact-wrapper" for styling purposes.

The form consists of several input fields for users to enter their information. The "Full Name" field requires users to provide their complete name, while the "E-mail" field expects users to enter a valid email address. These fields are marked as required, ensuring that users provide the necessary information before submitting the form.

Additionally, the form includes a "Subject" field, where users can specify the purpose or topic of their message. There is also a "Message" field, allowing users to write their inquiries, comments, or any other relevant information they wish to convey.

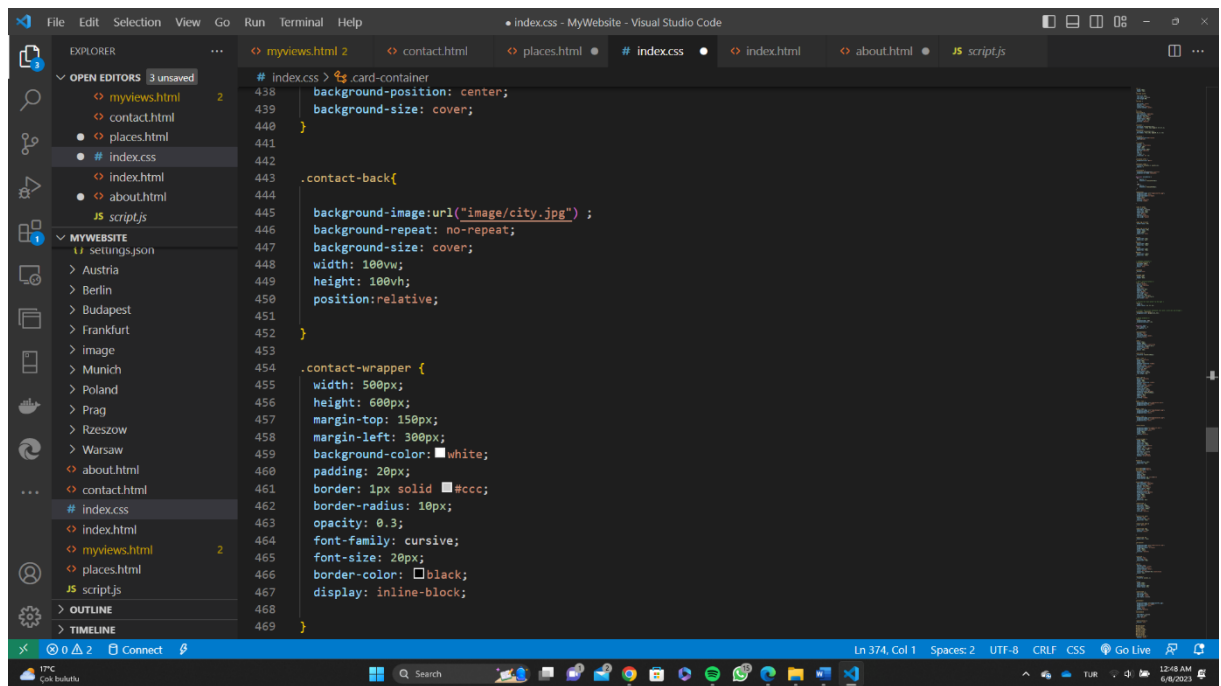
At the bottom of the form, there is a "Contact Me" button, which users can click to submit their form. Once submitted, the form data can be processed on the server-side to initiate appropriate actions, such as sending an email or storing the received messages.



```
File Edit Selection View Go Run Terminal Help
contact.html - MyWebsite - Visual Studio Code

EXPLORER
  OPEN EDITORS 3 unsaved
    myviews.html 2
    contact.html
    places.html
    # index.css
    index.html
    about.html
    JS script.js
  MYWEBSITE
    settings.json
    Austria
    Berlin
    Budapest
    Frankfurt
    image
    Munich
    Poland
    Prag
    Rzeszow
    Warsaw
    about.html
    contact.html
    # index.css
    index.html
    myviews.html 2
    places.html
    JS script.js
  OUTLINE
  TIMELINE

32 <div class="contact-back">
33
34
35
36
37
38 <div class="contact-wrapper">
39
40   <form>
41     <div>
42       <label for="name">Full Name</label>
43       <input type="text" id="name" name="name" required>
44     </div>
45
46     <div>
47       <label for="email">E-mail</label>
48       <input type="email" id="email" name="email" required>
49     </div>
50
51     <div>
52       <label for="subject">Subject</label>
53       <input type="text" id="subject" name="subject" required>
54     </div>
55
56     <div>
57       <label for="message">Message</label>
58       <input type="text" id="message" name="message" required>
59     </div>
60
61     <div>
62       <button type="submit">Contact Me</button>
63     </div>
64   </form>
65 </div>
```

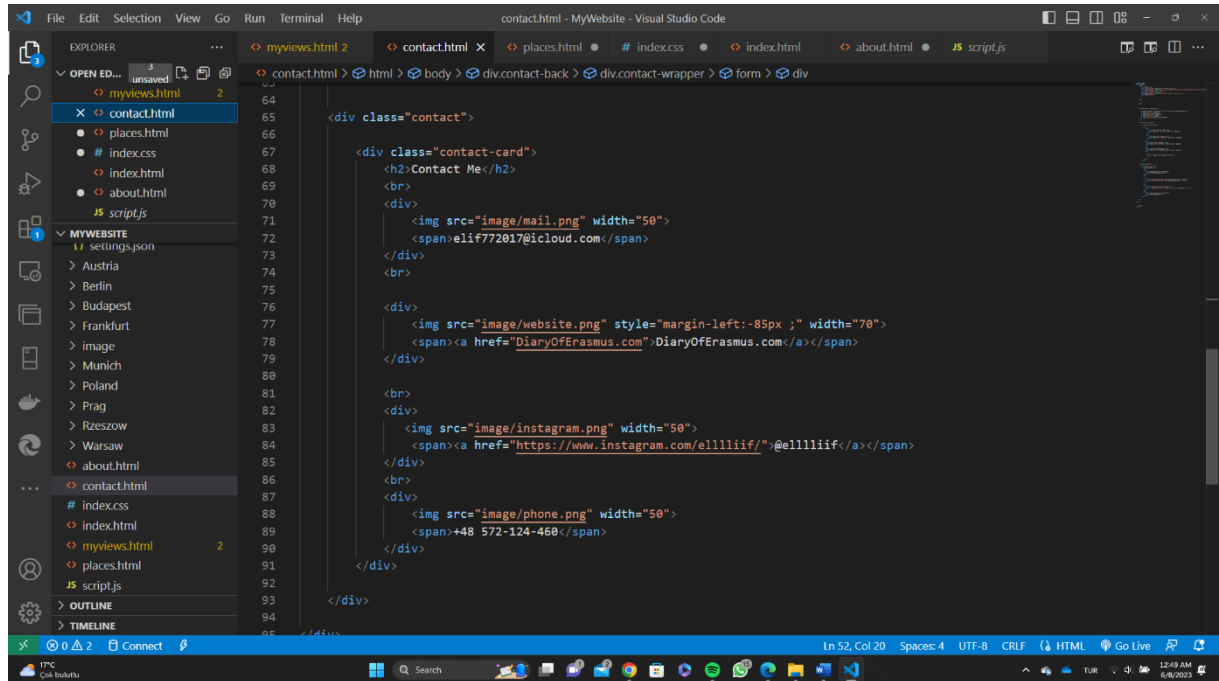


Contact Information

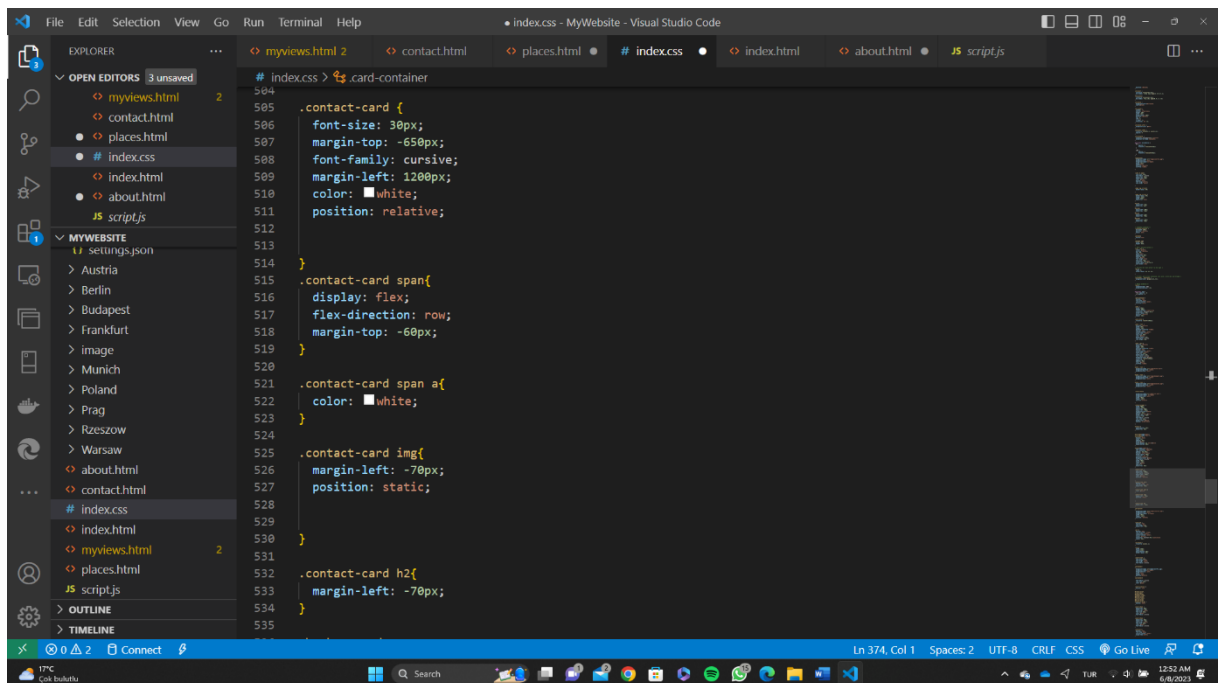
The provided code represents a contact section that allows users to get in touch with the person or organization. The section is enclosed within a container with the class "contact."

Within the section, there is a contact card denoted by the "contact-card" class. The card includes a heading "Contact Me" to emphasize its purpose.

This contact section offers multiple means of contact, including email, website, Instagram, and phone, making it convenient for users to reach out using their preferred communication method.



```
64 </div>
65
66 <div class="contact">
67   <div class="contact-card">
68     <h2>Contact Me</h2>
69     <br>
70     <div>
71       
72       <span>elif772017@icloud.com</span>
73     </div>
74     <br>
75     <div>
76       
77       <span><a href="DiaryOfErasmus.com">DiaryOfErasmus.com</a></span>
78     </div>
79     <br>
80     <div>
81       
82       <span><a href="https://www.instagram.com/ellllliif/">@ellllliif</a></span>
83     </div>
84     <br>
85     <div>
86       
87       <span>+48 572-124-460</span>
88     </div>
89   </div>
90 </div>
91
92
93
94
95 </div>
```



```
504 # index.css > card-container
505
506 .contact-card {
507   font-size: 30px;
508   margin-top: -650px;
509   font-family: cursive;
510   margin-left: 1200px;
511   color: white;
512   position: relative;
513 }
514
515 .contact-card span{
516   display: flex;
517   flex-direction: row;
518   margin-top: -60px;
519 }
520
521 .contact-card span a{
522   color: white;
523 }
524
525 .contact-card img{
526   margin-left: -70px;
527   position: static;
528 }
529
530
531
532 .contact-card h2{
533   margin-left: -70px;
534 }
535
```

Contact.html Page

