

## DETAILED EXPLANATORY FILE

Responsive multi-page website using HTML&CSS and a little JavaScript

### 1. Home page

To make a **responsive** (look good on all devices) web page, I used the **viewport meta tag**.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

In the **<head>** section, to connect my **CSS** file with the **html** file i used a link to an external style sheet file.

In the css file, I add to all the html elements:

```
*{  
  margin: 0;  
  padding: 0;  
  font-family: 'Poppins', sans-serif;  
  box-sizing: border-box;
```

With the **margin "0"** and **padding "0"** it is removed the default padding and margin for all the html elements.

**box-sizing: border-box** is another universal css property that allows us to include the padding and border in an element's total width and height.

The font style used (Poppins,sans-serif) are taken form Google fonts and the link is pasted on the **<head>** section of the html file and in the css file(\*{ } ).

```
<!--google fonts-->  
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>  
<link  
href="https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400;600;700&display=swap" rel="stylesheet">
```

In the **<body>** section for the header of my page is used the *section tag* with *class attribute* header.

```
<section class="header">
```

**(hero section)** In this hero section, it is set a background hero image with **gradient-linear overlay**.

```
.header{
  min-height: 100vh;
  width: 100%;
  background-image: linear-
gradient(rgba(4,9,30,0.79),rgba(4,9,30,0.7)),url(images/rose.jpg);
  background-position: center;
  background-size: cover;
  position: relative;
  overflow: hidden;
}
```

I set **min-height to 100vh** for the section so the height will be the same as any device the web users are opening the website (desktop, mobile, laptop).

The **width** of the section is **100%** which means that the width is 100% as the parent width including paddings and borders.

With the **background-position: center**, the background image is centered.

With **background-size: cover**, the image is covering complete section even if must to cut the image.

**Position-relative** sets the position of the section relative to its original position.

Inside this section(header) it is used the **<nav>** tag to add the logo and the navigation links(menu).

For the logo is used **<a href>** and **<img>** tags and for the menu it is used a **<div>** tag with **class nav-links**.

```
<div class="nav-links">
```

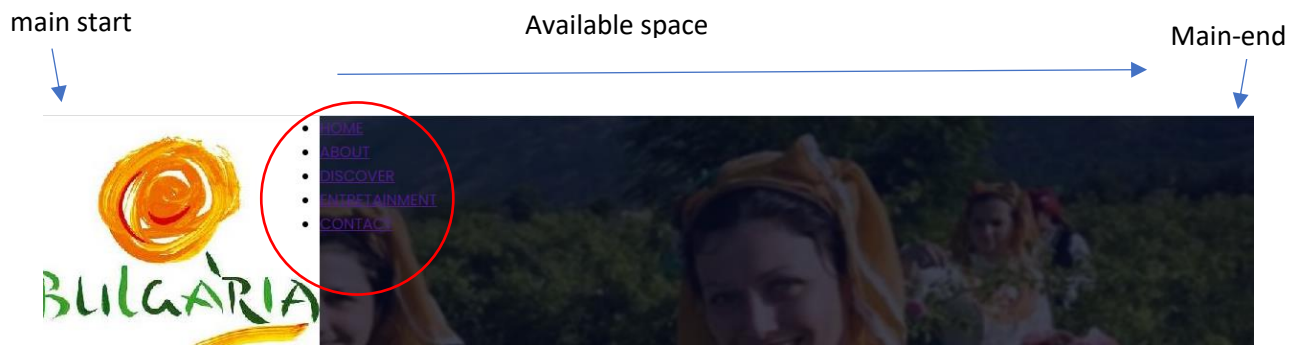
To add the links to the menu it is used **<ul>** (bulleted list) tag and **<li>** tag.

To design and align properly the logo and the nav links i used the css properties:

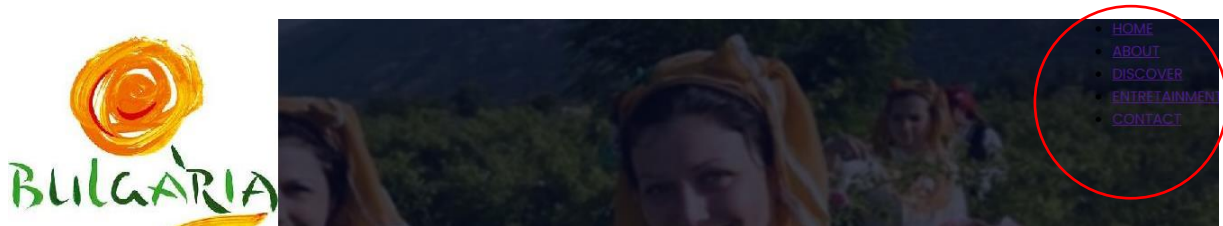
To become the **<nav>** section a flex container I set **display: flex;**

```
nav{
  display: flex;
```

After this property the logo and the nav links become flex items. The items are removed from under the logo and placed level with it toward the main start of the flex container. All the available space is pushed toward the main-end of the flex container due to the default value.



To add a space between the flex items I used the property ***justify-content: space-between;*** This property makes sure the logo and the nav-links are on the opposite ends of the nav section.



The nav-links are stacked on top of each other instead of lining up horizontally (due to the default styling of list items in the browser). To change this I make the ***.nav-links*** also a flex container so all the list items can be laid out according to the flexbox model.

```
.nav-links{
  flex: 1;
  text-align: right;
}
```

I use ***flex 1*** so it will occupy all the space in the row and ***text-align :right;*** so the text is on the right side.

Adding a style to the ul li nav-links:

```
.nav-links ul li{
  list-style: none;
  display: inline-block;
  padding: 8px 12px;
  position: relative;
```

```
}
```

To take away the bullets from the list it is add ***list-style: none;***

To position the nav-links ul li horizontally it is add the ***display: inline-block;***

To add some space I add ***padding*** 8px(top,bottom);12px(left, right).

And ***position relative.***

To align the flex items(logo and nav links) I used ***align-items*** property with value ***center.***

```
nav{
  display: flex;
  padding: 2% 6%;
  justify-content: space-between;
  align-items: center;
}
```

And for an internal space for the flex items I used padding 2%(top, bottom) and 6%(left, right).

I add also an underline (middle outwards) hover effect(animation) to the menu links:

```
.nav-links ul li::after{
  content: '';
  width: 0%;
  height: 2px;
  background: rgb(247, 98, 135);
  display: block;
  margin: auto;
  transition: 0.5s;
}
```

Because was used display: inline-block; for the nav-links ul li,we can make the underline span just the width of the text content.

To add a new element it is used the pseudo-element ***::after*** with width 0% to hide visually the line,;height:2px;with background color.;display: block; margin:auto; . To do the animation it is used the transition property: 0.5s;

To select the element when we mouse over we use the ***:hover*** selector;

```
.nav-links ul li:hover::after{
  width: 100%;
}
```

When the width of the line becomes 100% while hovering.

In the *hero section* is used another *div* with class “hero-text”

```
<div class="hero-text">
```

and inside this div is created a heading, a paragraph and a “*hero*” *button* (it is called “hero” button because in this section header, it is used a hero image (large image placed on the top of the page, in my case in the whole header section)).

```
.hero-text{
  width: 90%;
  color: #fff;
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  text-align: center;
}
```

The *width* of the hero text is 90% from the width of the hero section.

To position the hero text vertically and horizontally in the center of the image it is used *positioning: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%)* and text-align: center;

```
.hero-btn{
  display: inline-block;
  text-decoration: none;
  color: #fff;
  border: 1px solid #fff;
  padding: 12px 34px;
  font-size: 13px;
  background: transparent;
  position: relative;
  cursor: pointer;
}
```

For the *hero button* it is used *display: inline-block* so when we add some padding it is not break out from the hero-text.

The ***text-decoration*** is none so the underline is removed. Color of the button text is #fff; border 1 px;

I add some padding inside the button 12px top and bottom,34px left and right;font size 13 px;background:transparent;

The position is ***relative*** so it is in the flow and in the center. When the mouse is over the button the ***cursor is a pointer***.

I add a hover effect on the hero button with the ***:hover selector***; border and background while hovering and ***transition: 1s*** (the smoothly time between 2 properties).

```
.hero-btn:hover{
  border: 1px solid rgb(247, 98, 135) ;
  background: rgb(247, 98, 135);
  transition: 1s;
}
```

When we open the web page on a smaller screen it doesn't look good so to correct this I use the ***media query css*** technique.

In this media query I created the Toggle Menu as well.

If the browser window is 700px or smaller this css properties will be applied.

```
@media(max-width: 700px){
  .hero-text h1{
    font-size: 20px;
  }
  .nav-links ul li{
    display: block;
  }
  .nav-links{
    position: absolute;
    background: rgb(247, 98, 135);
    height: 100vh;
    width: 200px;
    top: 0;
    right: -200px;
    text-align: left;
    z-index: 2;
    transition: 1s;
  }
}
```

```

nav .fa{
  display: block;
  color: #fff;
  margin: 10px;
  font-size: 22px;
  cursor: pointer;
}
.nav-links ul{
  padding: 30px;
}

```

For the side menu I used these properties:

```

}
.nav-links ul li{
  display: block;
}
.nav-links{
  position: absolute;
  background: rgb(247, 98, 135);
  height: 100vh;
  width: 200px;
  top: 0;
  right: -200px;
  text-align: left;
  z-index: 2;
  transition: 1s;
}

```

For the **nav-links ul li** i have the **display: block** property so the nav links are under each other.

The position is absolute, the height is **100% viewport height**, **width: 200px**, top 0, to hide the side menu from the smaller screen it is used: **right:-200px**; the text is in the left side, **z-index: 2** means that the side menu is over the other elements, and the animation time(**transition**) between the 2 properties is 1s.

And to have some space around the menu-links I use:

```

.nav-links ul{
  padding: 30px;
}

```

To open and close the Toggle Menu I used icons from bootstrapcdn.com and copy and paste the link from the web page in the <head> tag of the html file.

```

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">

```

And after this I search for the icons in the fontawesome.com. ,I copy the close icon and the menu icon links and paste in my html document where I wanted them to be displayed.

```
<div class="nav-links">
    <i class="fa fa-times" onclick="hideMenu()"></i>

    <ul>
        <li><a href="">HOME</a></li>
        <li><a href="">ABOUT</a></li>
        <li><a href="">DISCOVER</a></li>
        <li><a href="">ENTRETAINMENT</a></li>
        <li><a href="">CONTACT</a></li>
    </ul>
</div>
<i class="fa fa-bars" onclick="showMenu()"></i>

</nav>
```

To hide the icons from the full screen, I used this property outside the media query :

```
nav .fa{
    display: none;
}
```

To add the icons to the smaller screen, inside the media query I used this properties:

```
nav .fa{
    display: block;
    color: #fff;
    margin: 10px;
    font-size: 22px;
    cursor: pointer;
```

The display: block; makes the icons visible, color:#fff; margin is 10 px from all the sides and font-size 22px; and when mouse over the cursor is a pointer.

To close the side menu I add click functions so when click on the closing icon it will close the side menu. For that I used JavaScript. I add click event to the html file in the closing icon.

```
<i class="fa fa-times" onclick="hideMenu()"></i>
```



And to open the side menu I add click event to the menu icon.

```
<i class="fa fa-bars" onclick="showMenu()"></i>
```

And after I add an id to the class “nav-links”

```
<div class="nav-links" id="navLinks">
```

And before the closing <body> tag I add one <script> where first I create a variable :

```
var navLinks = document.getElementById("navLinks");
```

And next i created the function:

```
function showMenu(){
    navLinks.style.right = "0";
}
function hideMenu(){
    navLinks.style.right = "-200px";
}
```

When we must to show the menu the right value must be “0” and when to hide the menu the value must be “-200px”.

The **second** section of my home page is the section with the class name places.

```
<section class="places">
```

In the css file for this section I have these properties:

```
.places{
    width: 80%;
    margin: auto;
    text-align: center;
    padding-top: 100px;
```

The **width** is 80% of the of the total width of the <body>;

The **margin** is auto so the section is horizontally centered;

The **text-align: center** aligns the text horizontally in the center of this section.

And inside space **padding-top: 100px;**

I have created content in 3 different columns and for that I used the **Bootstrap Grid System**.

For this i created `<div class="row">` and inside this div I added the 3 columns with the `<div class="places-col">`

```
<div class="row">  
    <div class="places-col">
```

For the class **.row** I have this css properties:

```
.row{  
    margin-top: 5%;  
    display: flex;  
    justify-content: space-between;  
}
```

I set **margin-top: 5%** (5% from the width of the section class="places")

**display: flex;** to make the **.row** flexible and all the div inside this row will be in the column.

**justify-content: space-between;** to add space between the flex items (the 3 columns)

For the 3 columns I have these css properties:

```
.places-col{  
    flex-basis: 31%;  
    background: radial-gradient( circle farthest-corner at 10%  
20%,  rgba(255,249,255,1) 11.1%, rgba(255,241,255,1) 90% );  
    border-radius: 10px;  
    margin-bottom: 5%;  
    padding: 20px 12px;  
    box-sizing: border-box;  
    transition: 0.5s;  
}
```

I set the width of the flex items by **flex-basis** property, before any available space is distributed in the row according to the flex factors.

I add a radial-gradient color for the background; radius for the corners; and margin at the bottom; padding inside the column- top, bottom(20px), left, right(12px);border-box property, so all the padding, borders and content from the column are including; and transition of 0.5s between the 2 properties.

I added a hover effect on the 3 columns with these css properties:

```
.places-col:hover{
  box-shadow: 0 0 20px 0px rgba(0,0,0,0.2)
}
```

With the box-shadow property, I add shadow effect around the columns frame with black color and opacity 20%.

Because the 3 columns don't look good on the smaller device, I add css properties in the media query so the flex items goes under each other:

```
@media(max-width: 700px){
  .row{
    flex-direction: column;
  }
}
```

The **third** section of my index.html document is the section with the class name "cities"

```
<section class="cities">
```

In this section again I use the Bootstrap grid system:

I have a div with class "row" inside this section

```
<div class="row">
```

And inside this div it is another div with class name "cities-col"

```
<div class="cities-col">
```

For the section "cities" I have these properties:

```
.cities{
  width: 80%;
  margin:auto;
  text-align: center;
  padding-top: 50px;
```

```
}
```

The **width** is 80% relative to the browser window's width; the **margin** is auto, so horizontally is in the center; the **text** is also centered and some inside space with **padding-top** 50px.

For the columns inside the div "row", I have these properties:

```
.cities-col{  
  flex-basis: 32%;  
  border-radius: 10px;  
  margin-bottom: 30px;  
  position: relative;  
  overflow: hidden;  
}
```

The width of the flex items I set with the flex-basis; I add border-radius property for the corners; 30px outside space in the bottom;; position: relative; so I is relative to its own size; and overflow: hidden, to hide any content out from the column; it is margin-bottom: 30px; position relative;

On each column, I have an image hover overlay effect with the class name "layer":

```
<div class="cities-col">  
  <div class="layer">
```

In the css file I have these properties for the image and the overlay effect.

```
.cities-col img{  
  width: 100%;  
  height: 100%;  
}  
.layer{  
  background: transparent;  
  height: 100%;  
  width: 100%;  
  position: absolute;  
  top: 0;  
  left: 0;  
  transition: 0.5s;  
}
```

The image is with 100% width and height of its container;

To place an overlay ( **.layer**) over the image I use **position: absolute;**

And to cover the complete image I set width and height 100%, top and left to 0; and the transition between the 2 properties to 0.5s.

And to hide the overlay and be only visible when hovering on the image I set these properties:

```
.layer:hover{
  background: hsl(39, 100%, 50%,0.5)
}
```

To animate the title which is inside the div with the class layer I set these properties:

```
<div class="layer">
  <h3>TARNOVO</h3>
</div>
```

```
.layer h3{
  width: 100%;
  font-weight: 500;
  color: #fff;
  font-size: 26px;
  bottom: 0;
  left: 50%;
  transform: translateX(-50%);
  position: absolute;
  opacity: 0;
  transition: 0.5s;
}
```

I add to this heading **width: 100%**, so it takes all the width of the layer; I changed the **font-weight to 500**; the size to 26px so it is big;

To put the text at the bottom and center it from the X axis, i is used **bottom: 0; left: 50%; transform: translate X(-50%); position: absolute** so it is over the layer; **opacity: 0** to hide the text; and transition between the 2 properties 0.5s.

To show the text animation when hover:

```
.layer:hover h3{
  bottom: 49%;
  opacity: 1;
}
```

The **fourth** section of my index.html file is the section with the class name "ski"

```
<section class="ski">
```

For this section I have the bootstrap grid system as well.

And the css properties for this section are:

```
.ski{
  width: 80%;
  margin: auto;
  text-align: center;
  padding-top: 100px;
```

The width is 80% from the browser width; margin: auto; so it is centered horizontally; the text is in the center and it is some inside space on top (100px).

In this section, it is a div with class name “row” and inside this div are 3 columns with class “col” and inside every column is image, heading and paragraph.

```
<div class="ski-col">
  
  <h3>Borovets</h3>
  <p>You will have a lot of joy in this ski resort as never
before</p>
</div>
```

And the css properties for these are:

```
.ski-col{
  flex-basis: 31%;
  border-radius: 10px;
  margin-bottom: 5%;
  text-align: left;
}
.ski-col img{
  width: 100%;
  border-radius: 10px;
}
.ski-col p{
  padding: 0;
}
.ski-col h3{
  margin-top: 16px;
  margin-bottom: 15px;
  text-align: left;
}
```

For the column, I have the width **flex basis: 31%**, before any available space is distributed in the row according to the flex factors; border **with radius 10px**;

***margin-bottom: 5%***(outside space with 5 % of the row width); and text to the left.

And the image is **100%** so the image expand to fit the column to any size; border with 10px; the paragraph it is set no padding; and for the heading it is : outside space from the top (***margin-top: 16px;*** ), outside space from the bottom (***margin-bottom: 15px;*** ) and text is on the left.

The ***fifth*** section of the index.html file is the section with the class name ***“gallery”***, inside this section I have 1 row,4 columns and images in each column.

```
<section class="gallery">
  <h1>Our Gallery</h1>
  <p>A journey of a thousand miles begins with a single step</p>
  <div class="row-gallery">
    <div class="column">
      
```

The css properties for this gallery are:

```
.gallery{
  width: 80%;
  margin: auto;
  text-align: center;
  padding-top: 100px;
}
```

The section “gallery” has 80% of the browser width, ***margin: auto*** so the section is centered, the heading and the paragraph are horizontally centered, and there is inside space from the top(100px).

```
.row-gallery {
  display: flex;
  flex-wrap: wrap;
  padding: 0 4px;
  margin-top: 5%;
  justify-content: space-between;
}
```

***The display: flex*** property makes the row flexible and all the div inside will be in the column. I used ***flex-wrap: wrap***, so the flex items break into multiple lines. The padding inside the row is 0px from top and bottom, 4px from left and

right; the ***margin from the top*** is 5% of the section gallery width. And to add a space between the flex items I used ***justify-content : space between***.

To create equal columns that sit next to each other, I add the following css properties:

```
.column {  
  flex: 25%;  
  max-width: 25%;  
  padding: 0 4px;  
  float: left;  
}
```

The ***sixth*** section of my home page is with the class name ***“testimonials”***.

```
<section class="testimonials">  
  <h1>Testimonials</h1>  
  <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Sint delectus  
error saepe dicta cupi</p>
```

In this section I have title and description as well and the css properties for this section are:

```
.testimonials{  
  width: 80%;  
  margin: auto;  
  padding-top: 100px;  
  text-align: center;  
}
```

The ***width*** of the section is ***80%*** of the browser width, the ***margin*** is ***auto***, so it is positioned in the center; and I add some inside space from the top: ***padding-top: 100px; text-align: center***, so the heading and the paragraph are horizontally centered.

To create 2 columns in this section I add a div with the class name ***“row”***. In this row, I add another div with the class name ***“testimonial-col”*** for the columns



and I add a user image `<img>` and some text `<p>` ,for the name of the user I used the `<h3>` tag.

```
<div class="testimonial-col">
  
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Alias
vitae dolore placeat nam modi repellendus consequatur ad, provident blanditiis
minus illum fugit facilis ut repudiandae quia, architecto eveniet ullam
sint.</p>
  <h3>Kristina Popova</h3>
</div>
```

The second column has the same html properties but different name and image.

For the 2 columns I add the following css properties:

```
testimonial-col{
  flex-basis: 44%;
  border-radius: 10px;
  margin-bottom: 5%;
  text-align: left;
  background: #fff3f3;
  padding: 25px;
  cursor: pointer;
  display: flex;
}
```

**flex-basis: 44%** because I have only 2 columns in the row; the **radius** of the columns is 10px; some **outside space** for the columns from the **bottom 5%** of the width of the row; the **text** is on the left side; some inside space it is added: padding: 25px; the cursor when hover is a pointer; and **display: flex** so the columns are displayed as a bloc- level flex containers.

In each column, there is a user image and the css properties for that are:

```
.testimonials-col img{
  height: 40px;
  width: 40px;
  margin-left: 5px;
  margin-right: 30px;
  border-radius: 50%;
}
```

The **height and weight** of the image are equal because with the **border radius=50%** it I possible to make the image as a circle. Also, it is some **margin** (outside space) from the left and right sides of the image.

For the text and the name inside the columns I set these css properties:

```
.testimonials-col p{
  padding: 0;
}
.testimonials-col h3{
  margin-top: 15px;
  text-align: left;
  color: black;
}
```

At the bottom of the 2 columns, I add **star icons** and for that, I copy and paste into my html file a link from google fontawesome icons.

```
<i class="fa fa-star"></i>
    <i class="fa fa-star"></i>
    <i class="fa fa-star"></i>
    <i class="fa fa-star"></i>
    <i class="fa fa-half-o"></i>
```

And the last icon class is ***"fa fa-half-o"*** to add an half star icon, and for empty star icon it is the class=***"fa-star-o"***.

```
<i class="fa fa-star-o"></i>
```

To change the color of the star icons I set the following css properties:

```
.testimonials-col .fa{
  color: orangered;
}
```

The ***seventh*** section of my index.html file is the ***call to action section*** with the class name ***"cta"***:

```
section class="cta">
  <h1>Contact Us to Share Your Experience<br> with Us and join our team<h1>
    <a href="" class="hero-btn">CONTACT US</a>
</section>
```

In this section, it is a heading and a "hero" button.

And the css properties for this section are:

```
.cta{
  width: 80%;
  margin: 100px auto;
```

```

    background-image:linear-gradient(rgba(0, 0, 0, 0.7), rgba(0, 0, 0, 0.7)),url(images/contactus.jpg);
    background-position: center;
    background-size: cover;
    border-radius: 10px;
    text-align: center;
    padding: 100px 0;
}
.cta h1{
    margin-bottom: 40px;
    padding: 0;
}

```

80% width from the browser width; margin 100px from top and bottom; and auto to put it horizontally centered; background image with linear-gradient color; background image is in the center and covers entire section; the text is horizontally centered; the padding 100px from top and bottom and 0 from the right and left;

The heading of this section doesn't look good on a smaller device so I use the media query technique:

```

@media(max-width: 700){
    .cta h1{
        font-size: 24px;
    }
}

```

In the ***eighth*** section of the index.html file, I have the “***footer***” section.

```

<section class="footer">
    <h4>About Us</h4>
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Pariatur impedit nobis, quidem dolor sunt autem ullam mollitia,<br> praesentium placeat animi et porro ratione ipsa reprehenderit nam vero alias provident laudantium.</p>
    <div class="icons">
        <i class="fa fa-facebook"></i>
        <i class="fa fa-twitter"></i>
        <i class="fa fa-instagram"></i>
        <i class="fa fa-linkedin"></i>
    </div>
    <p>All rights reserved.</p>
</section>

```

In this section, there is one heading, one paragraph, one div with icons, and another paragraph.

And the css properties for this section are:

```
.footer{
  width: 100%;
  text-align: center;
  padding: 30 px 0;
}
.footer h4{
  margin-bottom: 25px;
  margin-top: 20px;
  font-weight: 600;
}
.icons .fa{
  color: orangered;
  margin: 0 13px;
  cursor: pointer;
  padding: 18px 0;
}
```

The footer is 100% of the browser width; text-in the center; inside space top, bottom 30px left and right 0px;

The heading of this footer section has outside space from the bottom 25px; and from the top 20px; font-weight 600 (almost bold);

The icons are with color orange; space from outside 0px from top and bottom and 13 px from the right and left; when hover over them, the cursor is a pointer, and some inside space is added 18px top and bottom, and 0px left and right.

## **2. About Us page**

To create the about page from my project, first I duplicate the html file, and rename the new html file as **about.html** file.

After that, the first section of this page is renamed to a section with the class name **“sub-header”**.

```
<section class="sub-header">
```

For this section, I add the following css properties:

```
.sub-header{
  height: 50vh;
  width: 100%;
  background-image: linear-
gradient(rgba(4,9,30,0.7),rgba(4,9,30,0.7)),url(images/aboutus.jpg);
  background-position: center;
  background-size: cover;
  text-align: center;
}
```

The height of this section is **50% of the viewport height** and the **width** is **100%** of the browser width.

I add a **background linear-gradient color** and image **positioned in the center** and **cover** all the section space; and the heading in this section is centered horizontally: **text-align: center;**

To add some content to this page I created another section with 2 columns, with the basic structure of the Bootstrap's grid system.

```
<section class="about-us">
  <div class="row">
    <div class="about-col">
      <h1>We are a small team of  adventure lovers</h1>
      <p>Exploring every part of Bulgaria is a big pleasure for us and
we are always ready for taking new roads</p>
      <a href="">EXPLORE MORE</a>
    </div>
    <div class="about-col">
      
    </div>
  </div>
</section>
```

And the css properties for this section are:

```
.row{
  margin-top: 5%;
  display: flex;
  justify-content: space-between;
}
```

The class name “row” properties were defined in the sections before.

```
.about-us{
  width: 80%;
  margin: auto;
  padding-top: 80px;
  padding-bottom: 50px;
}
```

This section has **80% of the width** of the browser width; **margin: auto** puts the section in the center; and some inside space from top and bottom.

The css properties for the columns in this section are:

```
.about-col{
  flex-basis: 48%;
  padding: 30px 2px;
}
```

The width of the flex items(columns) I set with the **flex-basis**; and because there are only 2 columns I set it to **48%**. Also is added some inside space from **the top and bottom 30px, left and right 2px**;

For the image in the columns I have the following css properties:

```
.about-col img{
  width: 100%;
}
```

The image is set 100% so it is covering the entire column.

For the heading and the paragraph in this column I have these css properties:

```
.about-col h1{
  padding: 0;
}
.about-col p{
  padding: 15px 0 25px;
}
```

The padding is 0 for both left and right, so the text and paragraph are aligned to left.

The bottom in the first column is already created in the css file, I have only changed the color for it and add another class name, so it is visible without hover and when hover the text color is changed to white.

```
<a href="" class="hero-btn red-btn">EXPLORE MORE</a>
```

These are the css properties for the button:

```
.red-btn{
  border: 1px solid rgb(247, 98, 135);
  background: transparent;
  color: rgb(247, 98, 135);
}
.red-btn:hover{
  color: #fff;
}
```

The same color, for the border and the text of the button, and I add white color when hovering on it.

The footer of this page is the same as the home page.

### **3. Activities Page**

On this page I have used almost the same html and css properties as the ***about us*** page and with the difference that on this page I have changed the background image for the header section:

```
<section class="sub-header1"
```

```
height: 50vh;
width: 100%;
background-image: linear-
gradient(rgba(4,9,30,0.7),rgba(4,9,30,0.7)),url(images/backgroundmountain.jpg)
;
background-position: center;
background-size: cover;
text-align: center;
```

The second section of this page has the same properties as the section ***"cities"*** from the home page with some small changes. In this section I have 1 row with 6 columns and the same css properties as the class name ***"cities"*** from the home page but different images (***"cities1"*** on the activities page)

And for the third section of the *activities page* I set the same html and css properties as the class name "*about-us*" from the about us page but different image and different text content ("*about-us1*").

In the next part of the activity page, I add a video banner with personalized play and pause icon buttons:

For this, I created a div with the class name "*banner*".

Inside the `<video>` tag is placed the video. To make the video playable I add a **Boolean** and **Loop** attributes: **controls(Boolean)** it specifies that audio/video controls should be displayed; **loop(loop)** to restart the video again and again after finishing it; **autoplay(Boolean)** the video automatically starts to play; **muted(Boolean)** it specifies that audio output is muted when video starts. In the `<source>` tag is placed the path of the video in the same way as the source of an image.

In the same div I put 2 more divs with icon buttons: pause and audio, the **class** attribute I will use for the css properties, and the **id** for the javascript properties.

```
<div id="banner">
  <div class="btn-pause" id="btnPause">
    <i class="fa fa-pause-circle-o"></i>
  </div>
  <div class="btn-audio" id="btnAudio">
    <i class="fa fa-volume-up"></i>
  </div>

  <video controls loop muted autoplay preload="auto" id="video">
    <source src="video.mp4" />
  </video>
</div>
```

For this video banner I have include 2 *javascript external files*, before the closing tag `<body>`.

```
<script src="js/object-fit-videos.js"></script>
<script src="js/app.js"></script>
```



The css properties for this video banner are:

```
#banner {  
  width: 80%;  
  margin: auto;  
  padding-top: 80px;  
  padding-bottom: 50px;  
  display: flex;  
  position: relative;  
}
```

The container of the **banner is 80%** of the browser width, **margin: auto** so is the same space from left and right; some inside space from top and bottom added; **display: flex**, so the video inside is flexible and **position relative**.

The css for the video are:

```
#banner video {  
  width: 100%;  
  height: 100%;  
  position: relative;  
  z-index: 1;  
  object-fit: cover;  
  font-family: "object-fit: cover;";  
}
```

The **width and height** of the video is 100% of his parent's container width and height, position relative to his parent's position; **z-index: 1** so I can position the icon buttons on it; I set **object-fit** property so the video is resized to fit its container when the video change resolution and with value **cover** so the video maintains its aspect ratio in the container.

For the **icons (pause and audio)** I have the following css properties:

```
.btn-pause,  
.btn-audio {  
  position: absolute;  
  display: flex;
```

```

align-items: center;
justify-content: center;
background: black;
border: 1px solid black;
color: #fff;
width: 40px;
height: 40px;
z-index: 2;
cursor: pointer;
border-radius: 20px;
transition: 0.2s;
opacity: 0.8;
left: 30px;
top: 180px;
}

```

**position: absolute** relative to the video container; **display: flex** so the icons are flexible and responsive; **z-index: 2** so the icons are over the video; To position the icons I used left and top properties(left: 30px; top: 180px; and to position one of the icons under the other one I used:

```

.btn-audio {
  top: 225px;
}

```

And while hover on the icons I used:

```

.btn-pause:hover,
.btn-audio:hover {
  background: gray;
  border: 1px solid #000;
  opacity: 1;
}

```

For smaller devices I add a media query:

```

@media (max-width: 700px) {
  .btn-audio,
  .btn-pause {
    width: 30px;
    height: 30px;
    top: 130px;
    left: 8px;
  }
  .btn-audio {
    top: 165px;
  }
}

```

```
}
```

Some browsers do not support object-fit property, for this problem I use a javascript polyfill **object-fit videos**, by searching them from google (object-fit-videos-npm), and to download the scripts we go to the repository [github.com/TricomB2B/object-fit-videos](https://github.com/TricomB2B/object-fit-videos) and we copy the script **object-fit videos.js** and paste it in my javascript folder.

And after adding a special font-family css property **font-family: "object-fit: cover;"**;

To function better the property **font-family: 'object-fit: cover'**; we must to make the JavaScript call to initialize the video with the special css property **objectFitVideos();** (in app.js)

```
objectFitVideos();
```

To function the icon buttons(audio and pause) I add some variables(constants), with the document method **querySelector** (find the first element with an id (#video); (#btnPause);(#btnAudio);

```
const video = document.querySelector("#video");
const btnPause = document.querySelector("#btnPause");
const btnAudio = document.querySelector("#btnAudio");
```

When the video starts, I set it to start with low volume:

For this, I go to the constant video and set the volume to 0.3

```
video.volume = 0.3;
```

To set the configuration for the **button pause**, I add an event listener that pause and play the video when click on the button:

```
btnPause.addEventListener("click", () => {
```

And to happend this, Inside this event I create one **if**:

```
if (video.paused) {
  video.play();
} else {
```

```
    video.pause();  
  }  
});
```

To add configuration for the **audio button**, I add an event listener that starts and stops the sound of the video when click on the audio button:

```
btnAudio.addEventListener("click", () => {  
  if (video.muted == true) {  
    video.muted = false;  
  } else {  
    video.muted = true;  
  }  
});
```

The **footer** of this page is the same as the home page.

#### **4. Blog page**

On this page, I have the same html and css properties for the **header section** as the activities and about us page with the difference that here I changed the background image.

```
<section class="sub-header2">
```

```
.sub-header2{  
  height: 50vh;  
  width: 100%;  
  background-image:linear-  
gradient(rgba(4,9,30,0.7),rgba(4,9,30,0.7)),url(images/blog-writing.jpg);  
  background-repeat: no-repeat;  
  background-position: center;  
  background-size: cover;  
  text-align: center;  
}
```

For the second section of the blog page I have the class name ***“news-items”***.

One row with 6 columns. In each column, I added a date with the `<small>` tag and the class name ***“text-muted”***; ***heading*** and ***paragraph***. For the bottom of the column, it is a horizontal line `<hr>`.

```
<section class="news-items">
  <div class="row3">
    <div class="news-col">
      <p>
        <small class="text-muted">01.02.2022 17:42</small>
      </p>
      <h4>
        <a href="">Bulgarian sight views now days.</a>
      </h4>
      <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit.
Sed hic perspiciatis quo, quam quis esse quae sint enim corrupti est
repudiandae dolores velit facilis harum labore repellat earum dolorem
fuga?</p>
    </div>
  </div>
</div>
<hr>
```

For this section are the following css properties:

```
.news-items{
  width: 65%;
  margin:auto;
  padding: 10px 60px;
  border: 0.3px solid rgb(238, 237, 237);
  margin-top: 50px;
  border-radius: 15px;
  background-color: rgba(255,241,255,1);
}
```

The ***width is 65%*** of the browser width; ***margin: auto*** to center the section; some ***inside padding*** ; border: 0.3px; and outside space from the top(***margin-top: 50px;***) ;***border radius*** and ***background color***.

For the row and the columns are the following css properties:

```
.row3{
  display: flex;
  justify-content: space-between;
}
```

```
.news-col{
  flex-basis: 100%;
  text-align: left;
}
```

**display: flex;** to make the **.row** flexible and all the div inside this row will be in the column.

**justify-content: space-between;** to add space between the flex items (the 3 columns)

For the column flex-basis: 100%; to fill all the available space in the row and the text is to the left in the column.

And for the date in the column are the following css properties:

```
small{
  font-size: 70%;
  font-weight: 400;
}
.text-muted{
  color: #777;
}
```

And to add some meaning to the text it is used the css bootstrap helper class **.text-muted**.

And the **footer** for this page is the same as the home page.

## **5. Contact page**

The **header section** of this page has the same properties as the blog page header section with the difference that in this section is used another background image.

```
<section class="sub-header3">
```

And the css properties for this header:

```
.sub-header3{
  height: 50vh;
  width: 100%;
  background-image: linear-
gradient(rgba(4,9,30,0.7),rgba(4,9,30,0.7)),url(images/contactus.jpg);
  background-repeat: no-repeat;
  background-position: center;
```

```
background-size: cover;
text-align: center;
}
```

The **second section** of this page is with the class name **"location"**. In this section to add a location map I used google map location and after click on the share button it shows pop-up window with option **embed a map**, after selecting this option, on the new window I select **Copy Html** option and paste the code in my html file.

```
section class="location">
    <iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d2905.67915963453!2d28.02466321499213!3d43.25814558590501!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x40a45418c9764623%3A0x6c38b96215a11510!2z0JLQsNGA0L3QtdC90YHQutC4INGB0LLQvtCx0L7QtNC10L0g0YPQvdC40LLQtdGA0YHQUNGC0LXRgiAi0KfQtdGA0L3QvtGA0LjQt9C10YYg0KXRgNCw0LHRitGAig!5e0!3m2!1sbg!2sse!4v1643819657884!5m2!1sbg!2sse"
width="600" height="450" style="border:0;" allowfullscreen=""
loading="lazy"></iframe>
```

The css properties for this section are:

```
.location{
    width: 80%;
    margin: auto;
    padding: 80px 0;
}
.location iframe{
    width:100%;
}
```

The **width** of the section is 80% of the browser width; **margin: auto** so the section is in the center; **inside space(padding)** top and bottom 80px, 0px left and right.

And the **width of the iframe is 100%** so the map has the same width as the section and will be responsive.

The third section(**"contact-us"**) of this page it has one row with 2 columns. The left column has 3 divs.

```
<section class="contact-us">
    <div class="row">
        <div class="contact-col">
            <div>
```

```

        <i class="fa fa-home"></i>
        <span>
            <h5>Yanko Slav4ev 84</h5>
            <p>Varna, Bulgaria</p>
        </span>
    </div>
</div>
<div>
    <i class="fa fa-phone"></i>
    <span>
        <h5>003591215644</h5>
        <p>Monday to Saturday, 10AM to 5PM</p>
    </span>
</div>
<div>
    <i class="fa fa-envelope-o"></i>
    <span>
        <h5>elli.corleone@yahoo.com</h5>
        <p>Email Us</p>
    </span>
</div>
</div>

```

The css properties for this section are:

```

.contact-us{
    width: 80%;
    margin: auto;
}

```

The width is 80% from the browser width and margin: auto so the section is in the center.

For the row are the following css properties as it is mentioned before.

```

.row{
    margin-top: 5%;
    display: flex;
    justify-content: space-between;
}

```

And for the columns are these css properties:

```

.contact-col{
    flex-basis: 48%;
    margin-bottom: 30px;
}

```

Because I have only 2 columns so I set ***flex-basis: 48%***; and add some outside space from the bottom.



In the first column I add ***contact information***.

For the divs inside the first column are these css properties:

```
.contact-col div{
  display: flex;
  align-items: center;
  margin-bottom: 40px;
}
.contact-col div .fa{
  font-size: 28px;
  color: rgb(247, 98, 135);
  margin: 10px;
  margin-right: 30px;
}
.contact-col div p{
  padding: 0;
}
.contact-col div h5{
  font-size: 20px;
  margin-bottom: 5px;
  color: #555;
  font-weight: 400;
}
```

The div in the column is set to ***display: flex***, to make it flexible and all the items inside will be in a column. The items inside the div are aligned in the center of the div. The size of the icons ***.fa*** are 28px; the outside space is 10px from all the sides and an extra 30px on the right side. Then I have the properties for the heading and the paragraph.

In the second column, I add a ***contact form***.

```
<div class="contact-col">
  <form action="">
```

I add input fields for name, email and subject:

```
<input type="text" placeholder="Enter Your name" required>
```

***<input>*** element of type ***text***, creates basic single-line text field. The expected data in the field is defined by the attribute ***placeholder*** with the value ***Enter Your Name***, so this provides a brief hint of what kind of information is expected. The ***Boolean*** attribute ***required*** specifies the value for the input (text in this case) before the form can be submitted.

The input field for the email:

```
<input type="email" placeholder="Enter email address" required>
```

And for the subject:

```
<input type="text" placeholder="Enter your subject" required>
```

To allow users to send a message, I add a message box using the **<textarea>** element that represents a multiline plain-text editing control to enter a sizeable amount of free-form text. And the **row** attribute specifies the exact size for the **<textarea>** to take ( in my case 8 rows). And the placeholder is with value message and Boolean **required**.

```
<textarea rows="8" placeholder="Message" required></textarea>
```

To submit the form I add the **<button>** element and to specify the behavior of the button is used **type** attribute with value **"submit"**. To design this button I used the same class name for this button as the **hero-btn red-btn**.

```
<button type="submit" class="hero-btn red-btn">Send Message</button>
```

To design this form I used the following css properties:

```
.contact-col input, .contact-col textarea{  
  width: 100%;  
  padding: 15px;  
  margin-bottom: 17px;  
  outline: none;  
  border: 1px solid #ccc;  
  box-sizing: border-box;  
}
```

The **width is 100%** so it is responsive and has the same width as the column. I add some inside space (padding) inside the fields and margin at the bottom so it is space between the fields.

On the small screen, the page was scrolling horizontally and I could see the side menu, so to fix this I just changed the position absolute in the media query to position fixed.

```
.nav-links{  
  position: fixed;  
  background: rgb(247, 98, 135);  
  height: 100vh;  
  width: 200px;  
  top: 0;  
  right: -200px;  
  text-align: left;  
  z-index: 2;
```

```
    transition: 1s;  
}
```

The footer of this page is the same as the footer section in the previous pages.