GITHUB LINK - <https://github.com/m-stavrakov/Recipes_project>

The website that I created is a recipe website where people can find their favourite recipes. I decided to use blue colors to make it easy on the eyes of the customer. The body is light blue, so all pictures and text are easy to see and read. Headings, and important links (such as the ones for social media) are darker blue so they contrast with the body color and are easily seen and read.

The development process took a lot of time and effort, and many things were changed throughout the process. I started with the header section where I used a pattern as a background so it could stand out from the rest of the page. In the header I placed icons with links to the most famous social medias. On the other side of the header is a log in button, which when clicked it opens a small container. The container allows the user to enter their log in details if they have registered before. When the user clicks on the email or password tap it changes the color of the icon and the bottom border to darker blue so it can be clear to the user which tab, they are on. This was made using JavaScript with an event listener and if statements. Furthermore, the user can toggle between visible and invisible password and depending on which option they chose the icon of an open and closed eye changes accordingly. After that there is a log in button, or the user can choose to log in using Facebook or Google.

The next part of the main page is the title and the navigation bar. The title is displayed using Lobster font and darker blue color, so it is easy to see and differentiate from the rest of the content on the website. The navigation bar consists of home, categories, and search. When clicking on categories it will take the user to the categories section within the page.

The carousel is positioned after the navigation bar and takes the whole width of the display. I tried to make it smaller in terms of width, however I faced many challenges doing that. Most of the issues were with the positioning of the arrows and the carousel itself and their responsiveness when resizing the window. In the end I fixed most issues, however I decided to go with full width carousel as it looks better, and the responsiveness is much better. There are five slides each portraying different type of food and there is text description on each slide. Using JavaScript, I made it so when the arrows are clicked the user can browse through the slides and can go to the next and previous slide with ease. The initial transition between the slides was not smooth at all, but with the use of CSS I managed to make it much smoother.

The next section of the page is the categories. They are separated into six circle containers with pictures of the category in it. When the user hovers on top of it the image zooms in, and the color of the text underneath changes to dark blue so the user can be aware on which category they are hovering with no problem. When the user clicks on a category three containers will appear. This was made using JavaScript, when the user clicks once the recipe links will appear, when they click on another category the recipes will change according to the selected category. If the user clicks the same circle again the recipes will disappear. This was done using for each function and then add event listener on click for each category. It was difficult to write the code so it recognises when to open the specific category so for this reason I used data-list so I could match the category with the recipes. Using a switch case, I change the titles of each category depending on which one is chosen. When the client chooses a category, they will see three recipes. Each one consists of a picture, which like the circles when hoovered over it will zoom in, star ratings, name of recipe, picture and name of creator, date created and number of comments. Each recipe has a star rating, the user can rate the recipe only once. I did this using a for loop which selects all the elements within the parent, and then for each star element I assigned an event listener on click and the function checks if the star has already been clicked on and if it has it returns false. If it has not been clicked it sets an attribute to true for the specific star that has been clicked which colors all the stars before and including the one clicked. To color the stars before the one which was clicked, I used CSS that selects the selected star and all children after it and because I used row-reverse in the parent it will color the stars before the clicked one. After the user has rated, I made the rating be stored in the local storage so even after reloading the page the rating still remains. I had to do a lot of research on how to do this, and I faced a lot of errors until I actually managed to do it. I stored the specific star and the specific recipe that have been rated into and array and then stored the array in the local storage as a string. Then I check if there is an item stored locally with a specific name that I have assigned and retrieve it and convert it into an array. Then using a for loop, I go through each rating and each product and check if the product id from the current rating object matches the data-product id attribute of the current product element.

The final section in the home page is the footer. It has a slightly lighter color, so it is differentiated from the body. It contains the logo of the website, a quote and information links. Underneath all that is the copyrights and icons which are links to variable social medias.

The other page can be accessed by clicking on any recipe from the category section. The header of the new page has the same background pattern, and it contains the logo of the website and navigation link to the home page and the log in button. The log in button has the same functionality as the one from the main page. One issue I had when creating this new page and using the same JavaScript file as that there were errors saying it cannot read null data. This was due to the fact that this new page was lacking the elements that were in the home page and the JavaScript file was trying to find them. The solution I found is to wrap each such JavaScript code in an if statement checking if the section that corresponds to the interactivity code is present on the current page. This fixed all the errors.

The main section starts with the name of the recipe and two icons above it for share and flagging the recipe. This design is a standard for a recipe website. Below the title is the star rating which would display the same rating if the user rated it on the home page. If they have not the user can rate it here. Then there is the picture and names of the creator of the recipe, the date it was created, and the number of comments it has. After that is the image of the recipe dish and underneath it there are the preparation and cooking times, number of servings and a print icon. Using JavaScript, I added interactivity to the print button so when the user clicks on it will print the whole page. I also had to put it in an if statement to check if the print button is present on the page, so it does not give error when the home page is open.

Next part is the ingredients which are positioned in two columns with empty circles next to them. My plan was to make a check list using JavaScript. So, when the user clicks on the ingredient it will add a line through the text and it will change the circle image to a circle with a tick that way the user can tick the ingredients they have. The same was done for the next part which is the instructions for the recipe. One thing I found difficult is to toggle both the crossing of the text and the change of the image. This was fixed by putting both elements within the same variable and then use the for each function. And then create two different functions one to change the image and then another which toggles the text line and calls the function for the image. Then I create two variables within the for each function which are specific for the image and the text and use two event listeners for both variables and assign the functions.

After that comes the form which can be accessed by clicking on the button ‘Share your feedback’. Clicking on it will make the form appear. The form consists of inputs for the names of the user, their age, radio button which indicated if the recipe was delicious and then a text area for a longer comment. After clicking the submit button a comment will appear underneath. This was done by creating the html elements within the JavaScript and assigning them the pre-created classes and as a content assigning them the input from the form. After the submit button is clicked the input fields will be emptied. When the user chooses whether to click or not click the radio input it will display an icon in the comment showing a happy or sad emoji .Using a similar method as the one for the stars I store the comments on the local storage so that even after closing or reloading the page they will still be there. Furthermore, next to the comment title there is a counter which will show the number of comments up until now and will update when a new comment is added.

This page has the same footer as the one on the home page.

Bot pages are responsive and will work on computer, tablet, and phone screen sizes.