**Università di Pavia**

**Computer Engineering**

***Web and Multimedia Technologies* – Project Report**

|  |  |
| --- | --- |
| **Surname** | Marinelli |
| **Name** | Francesco |
| **Student ID** | 507675 |
| **E-mail** | francesco.marinelli01@universitadipavia.it |
| **Date of written test** | 22/02/2022 |

|  |
| --- |
| **Project title** (website subject) |
| VOTES4EUROVISION: A website to freely collect Eurovision Song Contest votes from audience. |

|  |  |  |
| --- | --- | --- |
| **HTML version** | | |
| 🞎 | HTML5 | 🞎 | HTML4 | 🞎 | XHTML |

|  |
| --- |
| **Client-side technologies used** |
| HTML, CSS, JavaScript (including JavaScript's plotly library) | |

|  |
| --- |
| **Server-side technologies used** |
| php, MySQL | |

|  |
| --- |
| **Website content description** |
| Describe the website in terms of:   * *Content of the single pages* and/or of the *main sections* (e.g., “The products page lists all the…”, “The ‘News’ section presents…”) * *Graphic design* (e.g., “The navigation menu is located in the left part of the page, immediately below the header, and…”)   Specify whether, for the definition of the content and/or of the graphic structure of your site, you have “drawn inspiration” from one or more existing websites (indicating their URLs).  This section can be extended to the next page, if necessary―but possibly avoid exceeding 50 lines.  The website consists of 7 total pages: Home, Login, Registration, Participants, Statistics, Voting Page and Logout. The first element that can be found in each page is the menu, which is located on top of everything else, horizontally centered and contains only links toward main pages: Home, Participants, Statistics and Voting Page, as well as the "Vote!" button.  For accessibility a hidden link has been insert before the menu to give the chance to skip the navigation bar when visiting pages using a screen reader. The name of the website is fully understandable also if read by a screen reader. A contrast check has been performed to be sure of the possibility to read text.  In the mobile version only the logo on the left remains and all the other links are substituted by an hamburger menu icon that if clicked shows all the other options disposed vertically.  In the homepage, under the menu, there are three different sections, their order follows a linear flow. The first section contains the aim of the website, under a short explanation there is the "Vote!" button. The second one gives a brief explanation of the Eurovision Song Contest 2022 (the description is from Wikipedia), the text is on the right whilst on the left there is an image. In the third section, on the left, the textual part describes the reason why the website has been created and some information about copyrights. At the end of the page there is the footer which contains some external links that are shown as circular buttons, they aim at the Municipality of Turin, at the ESC Instagram page and at the ESC YouTube Channel, as shown by labels that appear maintaining the cursor over them.  In the mobile version images are displayed before the textual content (in the last two sections), the rest remains the same.  The second relevant page is the one that contains the list of singers participating in the show. Under the menu there is a section that contains an iframe which is a YouTube video containing slices of all songs of the competition, under that video the list of singers along with the nation they represent, and the title of their songs are presented as a sequence of anchors. Subsequent anchors are displayed in a staggered way, on click they open the official Eurovision page related to the specific singer associated to the selected anchor. In the mobile version the layout remains the same, only margins are redefined in order to better exploit the screen.  The third main page is "Statistics". The page is structured in two columns obtaining a non-linear flow, on the left there is the pie chart which represents the percentage of users from around the world that voted, the graph has been drawn using "plotly". Under that there is a summary that if clicked shows details (a table with Countries and number of users that voted from them). On the right there is a sequence of summaries containing countries' name, clicking on them details (in a table) appear and show a table with other information about total votes received. In the mobile version the page becomes single columned placing the part containing users' related information on top. The ranking is placed under that obtaining a linear flow.  The voting page has a part of description related to who can be voted by the current user and a form in which only one country can be selected. In the menu there is a difference compared to the other pages, the vote button is substituted by a Logout button.  The registration page contains a form in which email, password and country must be insert. The layout is linear. The same flow is present also in the login page but this time the country is not requested. When the button "logout" in the voting page is pressed, the user is redirected to the logout page where a successful message is shown. The registration page and login page remain the same for mobile, the only elements that change are the select and input's sizes. |
| **Technical description of the website** |
| Describe the website implementation in terms of:   * *Folders (directories) and files*: list the folders that are part of the project, describing their content (e.g., “The folder img contains all the images of the website, while the folder…”). Clearly specify the name of the Home Page file * *HTML structure of pages*: indicate whether there are “structures” shared by all or many pages of the website, such as headers (e.g., <header> tag), generic containers (<div>, <section>), navigation sections (<nav>), main content (<article>), footer (<footer>), etc. * *CSS*: indicate the main selectors for which you have defined styles (shortly describing them), including possible classes or pseudo-classes (e.g., “The style for the #main selector defines an absolute positioning and…”; “The generic class .ital specifies that…”) * *JavaScript (and other possible client-side technologies used)*: illustrate the purpose of the employed JavaScript code (as well as of other possible client-side codes), shortly describing it (without necessarily providing all technical details, however); e.g., “The JavaScript function slideSh() allows to display a sequence of images which are placed in the folder …”). Indicate the URL(s) of the possible page(s) from which the JavaScript (or other technology) code has been “copied” and adapted to your own needs * *Server-side technologies*: indicate the purpose of the employed server-side code (e.g., PHP, ASP, Node.js, etc.), shortly describing it (without necessarily providing all technical details, however); e.g., “The PHP code at the beginning of the <body> of the page is used to…” * *Development tools employed*: HTML/CSS/JavaScript/etc. editors, possible image editing tools (e.g., Gimp, Photoshop, …), etc.   Specify whether, for the creation of the website, you have “drawn inspiration” from existing websites or templates (indicating their URLs), and whether you have used specific frameworks (such as Bootstrap). Also indicate anything you deem useful to make your implementation choices clear.  This section can be extended to the next page, if necessary―but possibly avoid exceeding 90 lines.  Inside the root directory there are two subdirectories:   1. "img" which contains images:    1. learning.jpeg which represents a guy browsing a book and some Math's formulas;    2. bgwave.png which is used as background for h1 and h2 titles;    3. logo2.png which is used as icon of the website, the image has been realized using Canva;    4. maneskin.jpeg which represents Maneskin after their win in Rotterdam. 2. "fonts" which contains the files associated to the two fonts used throughout all the website:    1. Anton-Regular.ttf used for the logo;    2. Raleway-Thin.ttf, Raleway-Thin.eot, Raleway-Thin.svg, Raleway-Thin.woff used for everything else.   The remaining files are not contained into subdirectories.  <https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css>: has been used as a support for displaying anchors in the footer.  mycss.css: this file defines the style of the whole website. The following selectors have been used:   * "body" to define the gradient for the background font and dimension of text. * ".skip-to-main-content" to hide the anchor necessary for the accessibility. * ".menu" to set the menu in the foreground in comparison with all the rest in the page, the background color, the position (centered at the top of the screen and sticky so that it is always displayed there also scrolling down the page). Contained links are distributed horizontally. * ".menu a" to define the color of links which are displayed in bold with an uppercase transformation. * ".menu .icon" to hide the hamburger menu's icon when the screen is bigger than 900px. * ".menu a:hover" to set to white the color of links inside the menu when the mouse is over them. * "a" to set to white the color of other links on the website. * ".menu a.currentpage" to highlight with color, font weight and effects the page of the website the user is navigating. The color is set to white, the weight is incremented from bold to bolder and the shadow effect is added. * "a:hover" to set the behavior of links outside the menu when the pointer is over them. Their dimension increases and the animation lasts 0.5 seconds. * "a.fa", ".fa-instagram", ".fa-location-arrow", ".fa-youtube" to set the style of links in the footer. * "button" to set the color of text inside buttons to white, the color of the background to a shade of fucsia, to redefine the edges of borders and to define the size of buttons. * ".aim p.button-container" to set the button in index.php (home) in the middle of the paragraph of the first section contained on the page. * "img[id="maneskin"]" and "img[id="learning"]" to set dimensions, position and rounded edges for the two images of the two sections of the homepage. * "button:hover" to set that when the "hover" event happens the button must enlarge and display a shadow. * "form.centred \*" to set the width and spacing values for all items contained in login and registration forms. * "hr.half-row" to define an horizontal row with a width half of its container wide. * "article.button-container" to define alignment and margin size of the article containing the button at the end of the body of login and sign up pages. * "header" to set the wave which recalls the ESC logo of this year as background for the header, to specify the font and other position's values. * ".logo" to set the font of the logo, the size of text, that it has to be displayed as uppercase text an to apply the shadow effect to the text. * "header h1.logo" to increase the dimension of h1 inside header by threefold. * "header h2" to change color of h2s inside header, to change their size and to apply the shadow effect. * ".artist" to define the appearance of the singers in the list in Participants page defining the gradient, the radius of borders, the padding (horizontal and vertical), vertical margins, color and size of text and the shape of cursor over this type of items. * "section.artists-container iframe" to set the dimension and position of the iframe containing all the 40 songs. * "section.artists-container a.column1" and "section.artists-container a.column2" to obtain the staggered effect on the list of participants. * ".artist .country" to set the content related to the country inside the .artist element on the right. * ".countries\_container" and "artists\_container" to display the Statistics page in two columns when the display is big enough. Otherwise, they are used to place the two sections one under the other. * "summary" to define the appearance of items that if clicked show the values of users that voted, or votes gained by an artist. * ".menu.responsive" to show the 'hamburger menu icon" when the screen is small enough. * @media screen and (max-width: 900px) and @media screen and (max-width: 600px) are used to change the page appearance when the width of the screen is minor than 900px or even 600px.   Javascript.js contains an "exists()" function to check input fields validity (it returns false if the field is empty or contains only spaces). "showMenu()" to change menu class name in order to make it appear in a different way according to screen size. "check\_login()" to check that fields of login form are not left empty, similar thing happens with "check\_reg()". The function "display\_totals()" is the one used to display the pie chart using the library "plotly-2.9.0.min.js"; the function collects data printed in the html thanks to php and stores them into arrays which are the used as parameters for the pie chart display.  PHP has been used for all pages given that they all need some information stored server side. Only the index could have been an html file, but MAMP expects to have a file called "index.php" so the extension has been changed into that.  The pages that make the website up are:   * artist.php is used server side to create a class "Artist" in order to make easier retrieving data and displaying them. * index.php is the main page. After the menu (in a div element) inside which there are all the <a> towards other pages, there is a main section made of the <header> that contains title <h1> and subtitle <h2> and three <article> with the main contents. The last two articles also contain an <img> element. The footer contains an <hr> to distinguish the rest of the page from it and useful external links <a>. * login.php at the beginning presents some code: a function which is executed when the user tries to log in and a function that only checks if the user is logged or not. Then there is the menu, and a section which contains two articles. The first article contains the <form> for logging in and the second article contains the <button> that sends to the registration page. * logout.php removes the variables set in the login phase and destroys the session. As well as the menu it contains a section with some <p> to communicate the successful logout. * participants.php collects all participants from database storing them into an array. The only <section> of the page contains two articles, the first one hosts the <iframe>: a YouTube video. The second <article> uses php to dynamically generate the content. It takes all the items of the array preloaded before and uses them to create an appropriate <a> with a <span> in it to display the country of the artist (on the right). * password.php is a utility file which if executed allows to collect the association between emails and passwords and contains the function check\_logged() used in votes.php for instance. * registration.php checks if the user has sent the registration form filled to the server, in that case it checks values and if the email is already present in the database. If everything is ok, it adds a row in the users' table. The structure is similar to the page of login. The main difference is that the form also contains a <select> element to let the user choose which country they come from. The second article contains a <button> which sends the user to the login page. * statistics.php is the most complex page of the website. In the first part contains php code used to collect data form database related to the artists, users that voted along with their provenience and votes received by each artist. Secondly there is the page built on the basis of previous data. The <section> countries\_container contains the <div> which hosts the pie chart, under that a <summary> invites to click on it to see data and a <table> is shown. About the <section> artists\_container, it has the current ranking of singers. Thanks to php artists with the same number of votes are in the same position. Clicking on the <summary> (containing only the nation) all the details (artist, song name and votes) are displayed inside a <table>. * votes.php in the first part has a code which retrieves all participants with all related data. The following code checks in the voting phase if the user is trying to vote for their country (not allowed) and in case the vote is legit it adds a row or modifies the corresponding existing one. The rest is simple, there are some explaining paragraphs and then a form with a <select> whose options are generated dynamically.   The most used technological tool is Atom for text editing along with GitHub which helped maintaining an history of the website. |