

## WEB APPLICATION DEVELOPMENT PROJECT SUBMISSION 2022

STUDENT NAME: ANTE DUJIC

STUDENT ID: G00398814

DATE: 21.05.2022.

START PAGE: login.html (username: user | password: password)

---

ITEM NO.	REQUIREMENTS
1	Allow the customer to enter their login details and have login details validated (via a login screen)
2	Perform form validation through JavaScript or HTML to ensure that: a. text fields are not empty b. a valid email address is entered
3	Access and change HTML on the web page through the DOM
4	Access and change styling through the DOM
5	Demonstrate the use of events
6	Contain two D3 data visualisations (e.g., Bar Chart) of your choosing a. One from a CSV file b. One from an array
7	Both visualisations should allow the user to specify display settings, including an option to change colour, display size and animations
8	Have a minimum of 3 linked pages

The table below shows which of the given requirements are implemented on which page.

PAGE	ITEMS									
	1	2		3	4	5	6		7	8
		a	b				a	b		
Login	•	•								
Home						•				•
Artwork				•	•	•				•
Insights						•	•	•	•	•
Contact		•	•			•				•
About				•	•	•				•

The tables below show detailed overview of each page and the brief explanation how and where the requirements were implemented.

login.html
On page access user is presented with the “WELCOME” message and a simple login form. Form consists of “User name” and “Password” field and the “Login” button. User is prompted to enter the details.
ITEM 1 & ITEM 2a
Form is created in HTML containing <i>input</i> elements of types <i>text</i> for username and <i>password</i> for password. Both input types are set to be <i>required</i> , meaning they have to be inputted. Form also contains a <i>button</i> element of type <i>reset</i> . Note that reset type was used, not the submit. [1]
JS <i>function</i> login() was used for verification. First, valid login details are defined. <ul style="list-style-type: none"> <li>• USERNAME: user</li> <li>• PASSWORD: password</li> </ul> The function is run when the user clicks the <i>login</i> button. Through DOM, it extracts the user input and checks against the defined (valid) login details. If the input is valid user proceeds to <i>home.html</i> page. If the input is invalid an alert message pops-up.
ITEM 1: login form, function login()
ITEM 2a: input text (for username) set to be required

home.html
After entering valid input details, user is taken to the home page. Home page (and all the other pages on this website) contain the navigation bar and the main content of the page. Navigation bar consists of a logo on the left corner, and links to all the other pages on the top right side. The main content on this page is a welcoming message with the “animation” text. [2]
ITEM 5 & ITEM 8
Navigation bar contains links to all pages. This was done using <code>&lt;a&gt;</code> tag. Each of the page name also triggers an event when hovered over, and change the style when being opened (active).
ITEM 5: Navigation bar
ITEM 8: Navigation bar

## artwork.html

User is presented with the options bar on the left and the set of images displayed right from it. The images displayed are from the realistic style.

### ITEM 4

The images are initially set to be hidden (their *display* property is set to *none*). There are two functions used to show images on this page. Function *showMini()* that shows 4 drawings done in minimalistic style and function *showReal()* that shows drawings done in realistic style. Running any of these functions will, through DOM, first set the style *display* property of the other set of images to *none* (to avoid displaying multiple sets of images on the page), and making the wanted images *display* to *block*, making them visible. *showReal()* function is run on opening the artwork page and also by clicking on REALISM button on the side menu. *showMini()* is run by clicking on MINIMALISM button on the side menu. (e.g. entering to Artwork page will show realistic images (*display=block*), clicking on the MINIMALISM will remove the realistic images (*display=none*) and display minimalistic images (*display=block*).

### ITEM 3

Under each image there is a heart icon with a number next to it. Those represent the number of likes(clicks) on the given image and this number is initialized in JS. The container with the number is accessed through DOM and changed – initialized numbers are inputted.

### ITEM 5 & ITEM 8

Hovering over side menu will trigger an event(underline). Also, hovering over the images will lower their *opacity*, and increase the *opacity* of the heart icons, revealing them to the user. Those icons can also be hovered over and are clickable. Hovering and clicking them both trigger events. If hovered over they change the *color*, and if clicked they change their size for the duration of the click. Clicking on the heart also runs a functions *clickOneM()*, *clickTwoM()*, *clickThreeM()*, etc. (see script.js) Those functions increase the initialized number by one and updating HTML through DOM, now showing number of likes + 1. Clicking on the heart will keep increasing the number. If the user leaves the page, numbers are set back to the first initialize values.

This page contains the navigation bar with above mentioned properties.

ITEM 3: function *clickOneM()*, function *clickTwoM()*, function *clickThreeM()*, etc. (see script.js)

ITEM 4: function *showMini()*, function *showReal()*

ITEM 5: hover over left menu (REALISM, MINIMALISM); hover over images; hover over heart icons; navigation bar

ITEM 8: Navigation bar

## insights.html

User is presented with the form to draw a chart on the left side upon entering the page. The form consists of 4 select elements with drop down options, 1 input of type color for color selection and a button to draw chart. The other part of the page is blank – this is where the chart gets drawn, using D3.

### ITEM 5 & ITEM 8

Hovering over the Draw chart *button* changes *background* and *font color*, making it clear it's clickable. This is done using *.hover* in CSS.

This page contains the navigation bar with above mentioned properties.

### ITEM 6a & ITEM 6b

*Select data* – Gives two options to select, *Sales by Month* and *Sales by Artwork*. If *Select by Month* is selected it will draw a chart using *yearlySales.csv* file. If *Sales by Artwork* is selected it will draw a chart using and 2-dimensional *array*. When *button* Draw Chart is clicked, it runs the function *drawChart()*. It extracts the selected value, and depending on the selection runs the appropriate code. This was done using *if else* statement.

### ITEM 7

*Select Height* – Gives options to select the height of the chart.

*Select Width* – Gives option to select the width of the chart.

*Select Draw Speed* – Gives option to select draw speed (transition duration)

*Select Color* – Gives options to select chart bars color.

All values are accessed through DOM and inserted in the relevant part of the code.

ITEM 5: hover over Draw Chart button; Navigation Bar

ITEMS 6a & 6b: form + function drawChart()

ITEM 7: form + function drawChart()

ITEM 8: Navigation bar

## contact.html

On page access user is presented with a contact form. It prompts the user to fill the details, write a message and send it.

### ITEM 2a & ITEM 2b

Form is created in *html* containing *input* forms of type *text*, *submit* and *email*. A *textarea* is also included in this form. All the forms are set as *required* and have to be filled (can't be left empty). Also, *email* form by default checks if an email address is inputted (by checking usual email format – e.g. if “@” is included).

### ITEM 5 & ITEM 8

Hovering over the Send Message *button* changes *background* and *font color*, making it clear it's clickable. This is done using *.hover* in CSS.

This page contains the navigation bar with above mentioned properties.

ITEM 2a: Input text (for first and last name),

ITEM 2b: Input email

ITEM 5: Navigation bar

ITEM 8: Navigation bar

about.html
On page access user is presented with a brief description of the page. There is a red dot located under the text.
ITEM 3 & ITEM 4
<p>Clicking the red dot runs a function <i>colorRed()</i>. The function accesses the description text through DOM and changes its style (<i>color</i> becomes red, all letters <i>lowercase</i>). It also reveals the hidden heart above the description, by setting its <i>opacity</i> to 1. This is also done through DOM.</p> <p>Clicking the red heart runs a function <i>project()</i>. This function changes the HTML using DOM. It accesses the description text and changes the content. Function also changes the style. It sets the <i>color</i> of the text to white, removes the dot by setting the <i>display</i> to <i>none</i> and changes the text back to <i>uppercase</i>.</p>
ITEM 5 & 8
This page contains the navigation bar with above mentioned properties.
ITEM 3: function project() ITEM 4: function project(); function colorRed() ITEM 5: Navigation bar ITEM 8: Navigation bar

## REFERENCES:

---

1. <https://stackoverflow.com/questions/15759020/window-location-href-doesnt-redirect>
2. <https://codepen.io/lefoy/pen/oXadRa>
3. <https://www.w3schools.com/html/default.asp>