



Nottingham Trent
University

Department of Computer Science

Year 2 – Term 2

SOFT20181: Internet Application Programming

Documentation

SOFT 20181

Internet Application Programming

Supporting Documentation

N1025571



Declaration of ownership:

I hereby declare that I am the sole author of this report and that all third-party items including code have been adequately acknowledge and referenced.

Fonti Jennifer.



INDEX

Introduction:	4
Success criteria:	4
Design process:	7
Initial Research:	7
Initial Design:	7
Implementation:	10
General:	10
Style sheet:	10
Home page:	10
Shop all, flowers, blue, places:	10
Product page:	10
Cart:	11
Artist:	11
Account, Sign Up:	11
Challenges encountered:	14
Summary and conclusion:	14
References:	14



LINK to Website on the UDON server:

<https://udon.ads.ntu.ac.uk/web/soft20181/N1025571/ecommerce/>

Introduction:

This is the supporting documentation for the ecommerce application created for SOFT20181 Internet Applications Programming.

This ecommerce application is intended to promote an artist's gallery. It's composed by 12 pages from the home one generally promoting the art, to a shop all section, a collection section, a specific page on singular paintings, an artist biography article and, lastly, the user space area.

The application has been developed mainly for pc/laptop but is completely responsive and works perfectly on phones and iPads as well. In fact, it has been tested with google responsive design, iPhone 11, iPhone 12 pro, iPad air and various laptops.

The Website has been written mainly in HTML5, the graphic part has been done with a separate CSS3 style file, and all the client side backend has been developed using JavaScript and jQuery.

Success criteria:

Everything on the applications match the requirements stated in the assessment brief. Some parts are slightly different from what asked, these will be discussed in the appropriate sections.

Requirements	Implemented?	Notes
Home (index.html) consisting of section/article elements describing some applications or benefit of the product to your audience. These should contain one or more images each.	YES	Home sections shows the most famous paintings and painting technique (video)
Product pages generally promoting each product. This should have a relevant video. (Note that you are not required to create a video file).	YES	I created a product page for each painting. I used one single HTML page (product) and changed the content using JavaScript. All the informations of the painting are stored in the URL to retrieved by the JS code. No video is present in the product page as a style choice. In fact, I wanted to give more visibility to the paintings without obscuring them with a video.

		Although, a video is present in the home page.
A registration page for account creation utilising appropriate form controls. It should not be possible to submit a malformed email address and name fields should only contain letters and spaces. Note that you are not required to conduct server-side processing of the form's data in this assignment.	YES	All the forms in SIGN UP are validated using JavaScript regex. The server side programming has not been implemented but the important login informations are stored in the local storage API, to give the chance to the user to experience the website both as a simple viewer and buyer. The sign in forms don't have any check because not working. This was a personal choice, since the server side processing was not done and I didn't want to store a password in the local storage for security reasons.
A customised presentation that allows the user to customise the basic font size and the background colour of the header elements used across all pages of the website. Their preference should be stored in Local Storage (using the Local Storage API) and used to control presentation the next time that they visit the site.	YES	In the account page between many options you can find two forms that allow the user to manipulate the style according to their own preferences. Although, some checks have been applied, e.g. the text size can't go over 4rem. This was done to give the possibility to the user to modify the website but still have an enjoyable user experience. All the data are stored in the Local Storage API, can be deleted logging out and modified back to original using a button.
A thumbnail image of each product should animate into a large version when it is clicked. This should be coded in JavaScript using the window.setInterval() and window.clearInterval ()methods or similar methods.	YES	All the image are thumbnail and if clicked they open a bigger version of themselves in anew page(done using CSS). On the other hand I decided to animate them using 'onmouseover', In fact, when the mouse is over a picture this one becomes slightly bigger to get back to its original size once the time



		interval declared with Timeout() is expired. I decided to use timeout instead of setInterval() because the action is not repeating in time but happens just once when triggered.
--	--	--

Design process:

Initial Research:

Not being an expert in the art sector the research for the initial design was vital to me. I explored numerous websites to brainstorm ideas and create a perfect mood board for this brand. Between other websites I visited and took inspiration from London Tate Modern (<https://www.tate.org.uk/visit/tate-modern>) and Jenny Saville (<https://gagosian.com/artists/jenny-saville/>) .

The design of this ecommerce us aimed for adult audiences, art lovers and critics. The colour palette is neutral and cold, to let the paintings be the main protagonist of the whole website. The layout is minimal and slim to give the page an elegant appearance.

Design approaches:

- The navigation bar on the top shows all the main part of the website to the user, making it easier to find any page.
- The first banner in the home screen takes the user directly to the newest and most important collection
- The video in the home page catches the user's eye and gets them to keep on scrolling through the website

Initial Design:

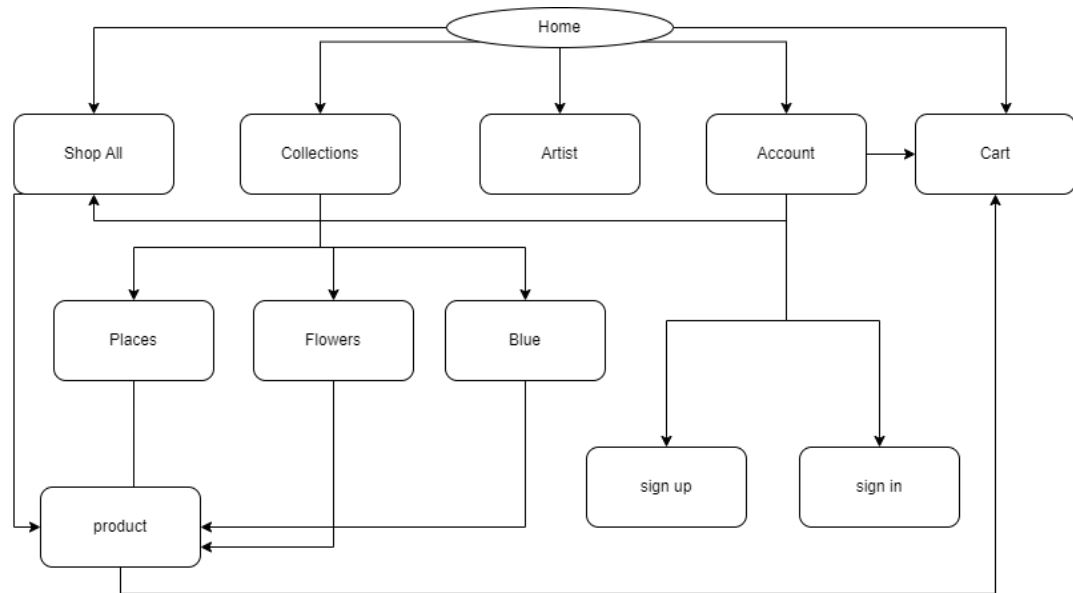
- Mood board:



- Wireframe:



- SiteMap:



Implementation:

General:

Every page of this website contains the same header and footer, the header is implemented using the tag <header>, it contains the image logo, links to the main pages and icons to the user space. The footer is delimited by the div <div class="footer">, contains basic information such as support contacts, social media name and copyright.

Every image of the paintings is a Thumbnail image. If clicked opens a new window with a larger image (implemented with HTML and CSS) and if the mouse goes over it becomes slightly bigger to emphasize the painting (coded in JavaScript using the setTimeout() function).

Style sheet:

The website uses just one single style sheet CSS3, that applies to every page. Most divs are display flex because I found them the best option to work with a responsive environment since they can stretch and compress without giving a bad user experiences. For some components such as the header was necessary implementing different properties when the screen size get <830px.

Home page:

The home page contains 3 main parts all delimited by section. The first one is a banner, uses an image as background and it redirects to the newest collection. The second one consists in a gallery of pictures, the most important paintings, each of this is a singular div, contains a picture and a description. The last section is composed of a significant video and the explanation of the painting technique

Shop all, flowers, blue, places:

The structure of these four pages is quite similar, they are galleries of images, delimited using a section and a div <div class='element'> for each picture. As mentioned in 'general' all pictures are thumbnails. When a picture gets clicked on in these pages all the meaningful informations get send to the product page using the header.

```
e.g. <a  
href="product.html?title=Sour&img=apple_lemon.jpeg&price=300"><p><b><i>Fiammetta  
<br> "Sour" </b></i></p></a>
```

Product page:

Since my website contains a lot of different painting the best approach seemed to be create a dynamic product page. In fact, the JavaScript page 'product.js' retrieves the information for each product from the URL using the following code:

```
const queryString = window.location.search;  
const urlParams = new URLSearchParams(queryString);
```

Using the informations to create dynamically the page.

It is also possible to add the Item to the cart by clicking the 'add to cart' button. The button will call the JS function add_cart(). This function is going to check if the Item is already

present in the cart and in this case it won't let the user adding it another time since any piece is supposed to be unique. When a product gets added to the cart its infos get stored in the Local Storage with a numerical key (generated incrementing a variable) and an array of information (name, picture, price). In fact, to check if the item is already present in the cart the function is going read through the local storage.

Code to add to cart:

```
function add_cart(){
    // puts all the info about the product on an array
    let element = new Array(title, img, price);
    // in case the item is not already in the cart it records a key(generated
    sequentially) and the array in local storage
    if(alreadypresent(element)==false){
        let n = localStorage.length +1;
        localStorage.setItem(n, element);
        products.push(img);
    }else{
        // if the item is already present it gives error (why explained in the
        report)
        alert("item already present in your cart");
    }
}
```

Cart:

As just stated every item is saved to the local storage, to retrieve it the JS function reads through the API till it finds a numerical Key (identifies a product) and it uses this informations to print on screen the title, image and price of the painting. It also keep count of the prices to give the user their total. The checkout is not implemented since there is not any server side part.

Artist:

The about the artist page is just a really easy presentation page composed of a single section with an image and a text area.

Account, Sign Up:

Sign in is a page where the user can insert their informations to login and shopping on the website. Every form is required (done using the HTML tag) and validate using JS and REGEX. If all the informations are correct the submit button will call a JavaScript function that saves the name and email address of the user in the local storage (so it keeps on being logged in even if it closes the website) and show them the Account page.

The account page is a quite simple page as well, it greets the user and give them the option to browse the main page or modify the website. As said in section 'Success Criteria' of this report the forms are still being checked in order to provide the best user experience possible. Another option given to the user is to logout, doing this all the informations stored in the local storage are going to be ereased.

Form validation code:

```
function check_forms(name,surname,age,phone,email,psw,psw_repeat){
    // regex
    let nameRegex = /^[a-z][a-z '-.]{0,31}$|^$/i;
    let pswRegex = /^[a-z][a-z '-.]{0,31}$|^$/i;
    let emailRegex = /^[a-z0-9_\. -]+@([\da-z\.-]+)\.([a-z\.-]{2,6})$/;
    let phoneRegex =
    /^(+44\s?\d{4}|\(?0\d{4}\)?)\s?\d{3}\s?\d{3}|(+44\s?\d{3}|\(?0\d{3}\)?)\s?\d{
3}\s?\d{4}|(+44\s?\d{2}|\(?0\d{2}\)?)\s?\d{4}\s?\d{4})\s?\#(\d{4}|\d{3})?$/;

    // test each value and if it's wrong gives alert and return
    if (nameRegex.test(name) == false){
        alert("insert a valid name");
        document.getElementById('name').focus();
        return false;
    }else if(nameRegex.test(surname) == false){
        alert("insert a valid last name");
        document.getElementById('surname').focus();
        return false;
    }else if(pswRegex.test(psw) == false){
        alert("insert a valid password");
        document.getElementById('psw').focus();
        return false;
    }else if(psw != psw_repeat){
        alert("the password must be the same");
        document.getElementById('psw-repeat').focus();
        return false;
    }else if(emailRegex.test(email) == false){
        alert("insert a valid email");
        document.getElementById('email').focus();
        return false;
    }else if(phoneRegex.test(phone) == false){
        alert("insert a valid phone");
        document.getElementById('phone').focus();
        return false;
    }else if(age<15 || age>100){
        alert("insert a valid age");
        document.getElementById('age').focus();
        return false;
    }else{
        save_data(name, email);
        return true;
    }
}
```

Customized header code:

```
function update_header(){
    // save values of preferences
    let size = document.getElementById("hsize").value;
    let color = document.getElementById("hcolor").value;
    // validation form on the parameters
    if(size != 0){
        if(size<1 || size>5){
            // return error
            alert("size must be >0 and <6");
            return false;
        }else{
            // store informations
            localStorage.setItem('hsize',size);
            localStorage.setItem('hcolor',color);
            set_header();
            return true;
        }
    }else{
        localStorage.setItem('hcolor',color);
        set_header();
        return true;
    }
}
function set_header(){
    let header = document.getElementById("header");

    // if values are stored it update the header
    // if it doesn't found any record gives a console error
    if(localStorage.getItem("hcolor") === null){
        console.log("size not found")
    }else{
        header.style.backgroundColor = localStorage.getItem('hcolor');
    }

    if(localStorage.getItem("hsize") === null){
        console.log("size not found")
    }else{
        let size = localStorage.getItem('hsize')+"rem";
        console.log(size)
        document.querySelectorAll("div.header > a").forEach((a) => {
            console.log(a);
            a.style.fontSize = size;
        });
    }
}
```

Challenges encountered:

The main challenge I encountered during the development of this website was the management of the cart, I tried different solutions before using an array to store the informations and when I did I found it difficult to retrieve the information inside it because it was being converted to string in the process. At the end, after some research I found the function `split()` to cast a string into array and I managed to make it work.

Summary and conclusion:

The development of this website was a bit challenging in particular for the design and planning part. I have experience in the development of internet applications but I never had do project one from scratch so I felt overwhelmed by the design choices. At the end following each diagram became gradually easier.

References:

- Images: All paintings have been done by Fiammetta Gianfanti and have no copyright on.
- Video: <https://pixabay.com/videos/>
- Oil painting technique text: https://en.wikipedia.org/wiki/Oil_painting
- Icons: <https://www.flaticon.com/>
- Platform used: Canva, Diagram, Visual Studio, Word.