

# REPORT

ASHISH NAYAK – MINH TUAN NGUYEN

Citations:

- Front-Page background-gif : <https://gifer.com/en/v4>
- FONTS: <https://allbestfonts.com/derelict-typeface/>  
<https://allbestfonts.com/voyager-typeface/>

## **OUTLINE:**

We have constructed a web-based gallery that incorporates the main features of any gallery. You start at a front-page from where you get an introduction to what the site is about. You then proceed to the main page. You can either set up an account or log-in if you already have. From there on in, you can see your photos stored in multiple albums. Features such as 'Add image', 'Delete Image', 'Zoom on image once clicked', and others have been coded.

## **FRONT PAGE:**

We wanted the front-page to be simple but not too simple. Our focus was on making sure we got the display technicalities right to perfection. We did not want any misalignments or out-of-place texts. So we made use of CSS to the maximum and particularly explored the 'Keyframes' framework available in CSS to get our motions as well using '@Font-face' to make use of the great fonts available. We made extensive use of the div operators and found it very useful in structuring our code.

## **MAIN PAGE:**

### **I. Features**

As our goal was to create a user-friendly and fully functional gallery and inspired by the functionalities of Google Drive and Samsung Galaxy's Gallery, we tried to push JavaScript to the limit and to replicate and implement as many features, interactions as possible, just with plain JavaScript and no other external library. Here are all the features that we have been able to implement:

#### a) Login and Signup

- On the top right corner, there are buttons, that will create popup forms when clicked, allow users to create account and access their personal albums and images.
- There is a default account with username: "admin" and password: "admin" for the demo purpose.

#### b) Interaction with albums

- On the left-side menu is the list of albums.
- There is a button with "+" sign that will create a new album on click.
- It is possible to click on different albums to open the content of each album, the currently open album is the album with white background.
- Next to each album are 2 buttons that allow rename or delete albums.

#### c) Interaction with images

- When an album is opened, we can see all the images contained in that album on the right side.
- Add images: On the top left of the page, next to the logo, there is a button "Add image" to import new image into the currently open album.
- Preview: Users can either double click on an image or right click and choose "Preview" to open a popup with that image zoomed in.
- Select images: There are a few ways to select single or multiple images:
  - o Single left click or right click and choose "Select" to select a single image.
  - o Hold "ctrl" and left click on different images to select multiples image.
  - o Press "ctrl" + "A" to select all the images of the current album.
- Cut and copy images: To cut and copy all the selected images:
  - o Right click on selected image or images and choose "Cut" or "Copy".
  - o Press "ctrl" + "X" or "ctrl" + "C".
- Paste images: To paste cut or copied images, just right click and choose "Paste" or press "ctrl" + "V".
- Delete images: To delete selected images, right click on selected image(s) and click on "Delete" or just press "Delete" on the keyboard.

## II. Difficulties

When doing this part, we encounter some difficulties on how to catch different events like "right-click", "double click" or a combination of key press like "ctrl" + "C", "ctr" + "V", "ctrl" + "click" ... Which we found the solutions and implement them.

We also have some difficulties in storing the data (username, password and image files) and we came up with 2 solutions: We use Local Storage of the web browser to store those data and we also created a very basic local server by using NodeJS and ExpressJS and store our data in ".json" files.

## III. Brief explanation of files and codes

We have some folders to store files:

- "Assets" is used to store images and fonts that are used by our website.
- "Data" is used to store all the images for the premade albums and the images that are sent to the server.
- "Server" is where NodeJS and ExpressJS is implemented. The "Server" contains a file name "library.js", which have the same functionality as the file "library.js" in the main folder but is used for interaction with the server.

Functionalities of all the JavaScript files:

- "library.js": There are 2 "library.js". Both of them are used for interactions between the web page and the data. They contain function like "login", "signup", "addAlbum", "deleteAlbum", "addImage" ... However, the file in the main folder is used to interact with data in Local Storage while "library.js" in "Server" folder is used to interact with the local server. On local storage version, the login process will try to read the two ".json" files for initial data, if it fails to do so, it will have only one default account username "admin" password "admin" with no album or image.
- "signup\_login.js" contains all the functions necessary for the login and signup process like controlling the login and signup buttons, forms.

- “pre\_process.js” contains most of all the process that needed to be done as soon as the page is loaded like some global variables, control of the preview panel and popups...
- “script.js” contains most of the important functions like “initiate” to load all the data after login, “getContent” and “wipeContent” for display or clean all the images showing on the website. There are also functions to handle right click, left click, double click events or select, copy, delete, preview image