

Front-End UI/UX Mini Project

- **Submitted By:**
- 2460446 Sarah Stanley sarah.stanley@btech.christuniversity.in
- 2460474 Vyshak S vyshak.s@btech.christuniversity.in
- 2460347 Chris George Thomas chris.georgethomas@btech.christuniversity.in
- **Course:** *UI/UX Design Fundamentals*
- **Instructor Name:** *Mr. Narendra Kumar*
- **Institution:** *Christ University*
- **Date of Submission:** *26/09/2025*
- **GITHUB LINK:** <https://github.com/ChrisGeorgeThomas/LNT>

TOPIC:

Photo Gallery

- **Description:** Design a photo gallery where users can browse through images and view them in a larger modal view when clicked.
- **Features:**
 - **Image Grid:** Display a grid of images in a clean, organized manner.
 - **Lightbox/Modal View:** Implement a modal feature where users can click on an image to view it in a larger size.
 - **Responsive Design:** Ensure the gallery adjusts and looks great on mobile, tablet, and desktop screens using Bootstrap.

Technologies Used: HTML, CSS, JavaScript, Bootstrap, jQuery.

2. Abstract

The Photo Gallery project is a web-based application enabling users to upload, organize, and view images within a responsive, interactive grid. By integrating features like modal (lightbox) viewing, folder and album management, cropping, and easy navigation, the solution offers seamless photo management across devices.

3. Objectives

- Create an intuitive platform for image uploading, organization, and preview.
- Enable users to group images into folders and albums.
- Provide modal image viewing with editing, zoom, crop, and download capabilities.
- Ensure mobile-first, responsive interface using Bootstrap components.
- Persist user actions (uploads, deletions, moves) using browser local storage for offline resilience.

5. Tools & Technologies Used

Tool/Technology	Purpose
HTML5	Structure and layout of the website
CSS3	Design, styling, and responsiveness
VS Code	Writing and editing code
Chrome DevTools	Testing and viewing the website
Bootstrap 5	Predefined responsive grid system and UI components
JavaScript (ES6)	Task management logic, reminders, and interactivity
jQuery	Simplified DOM manipulation and event handling

- HTML5 for structure and content.
- CSS3 (custom + Bootstrap) for styling and responsiveness.
- JavaScript & jQuery for interactivity and event handling.
- Bootstrap 4 for layout and modal management.



- Cropper.js for image crop/edit functionality.
- Font Awesome for iconography.
- Local Storage API for client-side state persistence.

6. HTML Structure Overview

- Main wrapper splits dashboard (folders & albums) from gallery display.
- Dashboard panel: Contains upload button, folder and album management sections.
- Gallery container: Presents view title and image grid.
- Modal dialogs for image view/edit, moving images, and album assignments.
- All structure emphasizes semantic tags and mobile accessibility

7. CSS Styling Strategy

- Used external stylesheet (style.css) with well-structured selectors.
- Defined CSS variables for color themes (light/dark mode).
- Applied Bootstrap grid and utility classes for responsive layout.
- Used Flexbox for alignment in topbar, sidebar, and task rows.
- Styled task priorities with colored dots (red = high, yellow = medium, green = low).
- Added hover effects and transitions for buttons and task rows.
- Ensured responsiveness with media queries (sidebar becomes horizontal on small screens).
- Maintained a minimalist and calm design using soft backgrounds, rounded cards, and subtle shadows.

- Uses CSS variables for configurable theme colors and dimensions.
- Dashboard and gallery panels styled with flat design, glassmorphism for modals, and styled scrollbars for a modern look.
- Flexbox layouts ensure consistency across breakpoints.

- Responsive adjustments made for grid items, buttons, overlays, and modal

8. Key Features

Feature	Description
Responsive Design	Adapts seamlessly to all screen sizes, from mobile to desktop.
Smooth Navigation	Fixed top navigation bar with anchor links for quick section access.
Band Member Profiles	Stylish layout with hover effects for each member's profile.
Contact Form (non-functional)	Placeholder layout for inputs and button
Accessible Fonts & Colors	High contrast and readable typography

- **Image Grid:** Automatic display of uploaded/default images in organized gallery.
- **Lightbox/Modal View:** Click any image for full-screen modal with zoom, crop, and navigation controls.
- **Folder & Album Management:** Add, rename, delete folders/albums and organize images.
- **Upload & Download:** Supports file uploads and direct downloads.
- **Cropping Functionality:** Users can crop or edit images before downloading.
- **Persistent Local Storage:** Retains state between sessions.
- **Responsive Design:** Bootstrap grid system enables seamless experience on any device.
- **Image Grid:** Automatic display of uploaded/default images in organized gallery.
- **Lightbox/Modal View:** Click any image for full-screen modal with zoom, crop, and navigation controls.
- **Folder & Album Management:** Add, rename, delete folders/albums and organize images.
- **Upload & Download:** Supports file uploads and direct downloads.
- **Cropping Functionality:** Users can crop or edit images before downloading.

- **Persistent Local Storage:** Retains state between sessions.[scribd](#)
- **Responsive Design:** Bootstrap grid system enables seamless experience on any device.
- **Challenges Faced & Solutions**

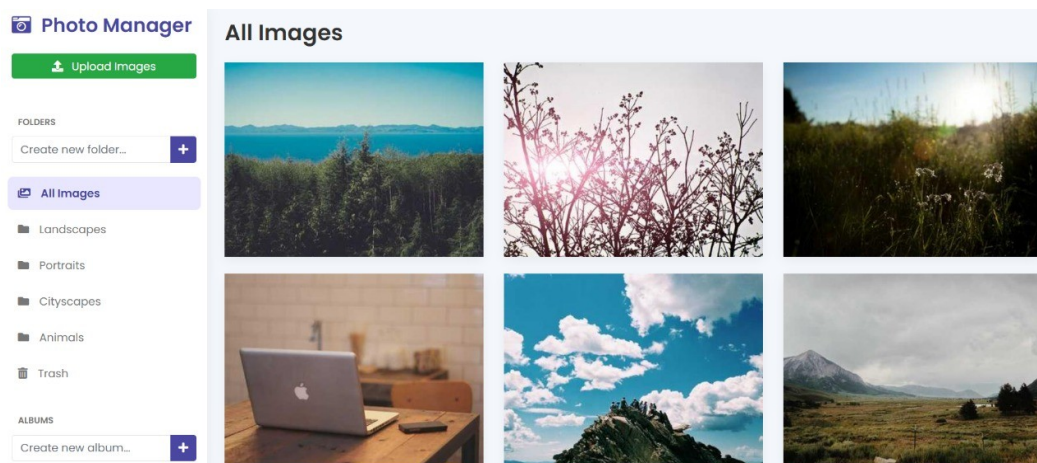
9. Challenges Faced & Solutions

- **File Management:** Handling multiple user uploads and preventing duplicates was managed by timestamped IDs and thumbnail/full distinctions.
- **Responsive Layout:** Ensured images and UI scale fluidly with Bootstrap and custom CSS variables.
- **Modal Interactivity:** Maintaining cropper state and cross-navigation between images required robust event-driven JavaScript.
- **State Persistence:** Used JSON structures in local storage to save all user-generated folders, albums, and image edits.
- **Security:** Validated file types and handled browser download restrictions with fallback alerts and error handling.[scribd+1](#)

10. Outcome

- A feature-rich, organized, and responsive photo management system that supports both novice and power users.
- Highly interactive and visually appealing UI with fast client-side performance.

• OUTPUT SCREEN





11. Future Enhancements Future

Enhancements:

- Integrate user authentication for private galleries.
- Use cloud storage or server-side backend for multi-user support.
- Add image tagging and advanced search.

- Enable batch operations and drag-and-drop organization.
- Implement lazy-loading and progressive image optimization for scalability

12. Sample Code

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta

charset="UTF-8">

  <meta

name="viewport"

content="width=device-

width, initial-scale=1.0">

  <title>Polished Photo

Manager</title>

  <link

href="https://fonts.google

apis.com/css2?
```

family=Poppins:wght@3

00;400;600&display=swa

p" rel="stylesheet">

<link rel="stylesheet"

href="https://stackpath.b

ootstrapcdn.com/bootstra

p/4.5.2/css/bootstrap.min.

css">

<link rel="stylesheet"

href="https://cdnjs.cloud

flare.com/ajax/libs/font-

awesome/5.15.4/css/

all.min.css">

<link rel="stylesheet"

href="https://cdnjs.cloud

flare.com/ajax/libs/cropp

erjs/1.5.12/cropper.min.cs

s">

<link rel="stylesheet"

href="style.css">

</head>

<body>


```
<div class="main-  
wrapper">  
  <div  
    id="dashboard">  
      <h3  
        class="p-3"><i  
          class="fas fa-camera-  
            retro mr-2"></i> Photo  
            Manager</h3>  
      <div class="px-3  
        mb-4">  
        <button  
          id="upload-btn"  
          class="btn btn-success  
            btn-block mb-3"><i  
            class="fas fa-upload  
              mr-2"></i> Upload  
              Images</button>  
        <input  
          type="file" id="file-  
            input" multiple hidden>  
      </div>
```

```
<h5
class="dashboard-
heading">Folders</h5>

<div class="px-3
mb-3">

    <div
class="input-group">

        <input
type="text" id="new-
folder-name"
class="form-control"
placeholder="Create new
folder...">

        <div
class="input-group-
append"><button
id="add-folder-btn"
class="btn btn-primary"
title="Add Folder"><i
class="fas
fa-plus"></i></button></
div>

    </div>
```

```
</div>

<ul id="folder-
list" class="list-
unstyled"></ul>

<h5
class="dashboard-
heading">Albums</h5>

<div class="px-3
mb-3">

  <div
class="input-group">

    <input
type="text" id="new-
album-name"
class="form-control"
placeholder="Create new
album...">

    <div
class="input-group-
append"><button
id="add-album-btn"
class="btn btn-primary"
title="Add Album"><i
```

```
class="fas
fa-plus"></i></button></
div>

</div>

</div>

<ul id="album-
list" class="list-
unstyled"></ul>

</div>

<div id="gallery-
container" class="p-4">

  <h2 id="current-
view-title"
class="mb-4">All
Images</h2>

  <div id="image-
grid"
class="row"></div>

</div>

</div>
```

```
<div class="modal
fade" id="imageModal"
tabindex="-1">
```

```
<div class="modal-
dialog modal-dialog-
centered modal-xl"><div
class="modal-
content"><div
class="modal-
body"><div class="crop-
container"><img
id="modalImage"
alt="Large
Image"></div></div></di
v></div>
```

```
<div class="modal-
controls-container">
```

```
<button
class="modal-control"
id="zoom-in"
title="Zoom In"><i
class="fas fa-search-
plus"></i></button>
```

```
<button  
class="modal-control"  
id="zoom-out"  
title="Zoom Out"><i  
class="fas fa-search-  
minus"></i></button>
```

```
<button  
class="modal-control"  
id="downloadBtn"  
title="Download  
Image"><i class="fas fa-  
download"></i></button
```

```
>
```

```
<button  
class="modal-control"  
id="cropBtn"  
title="Crop &  
Download"><i class="fas  
fa-crop-alt"></i></button
```

```
>
```

```
<button  
class="modal-control"  
data-dismiss="modal"
```

```
aria-label="Close"

title="Close"><i

class="fas

fa-times"></i></button>
```

```
</div>
```

```
<button

class="slider-control

prev" id="prevBtn"><i

class="fas fa-chevron-

left"></i></button>
```

```
<button

class="slider-control

next" id="nextBtn"><i

class="fas fa-chevron-

right"></i></button>
```

```
</div>
```

```
<div class="modal

fade" id="moveModal"

tabindex="-1">
```

```
<div class="modal-

dialog modal-dialog-

centered"><div
```

```
class="modal-  
content"><div  
class="modal-  
header"><h5  
class="modal-  
title">Move  
Image</h5><button  
type="button"  
class="close" data-  
dismiss="modal">&time  
s;</button></div><div  
class="modal-  
body"><p>Move this  
image to a different  
folder:</p><select  
id="folder-select"  
class="form-control"></s  
elect></div><div  
class="modal-  
footer"><button  
type="button"  
class="btn btn-  
secondary" data-
```


dismiss="modal">Cancel

</button><button

type="button"

class="btn btn-primary"

id="confirm-move-

btn">Move

Image</button></div></d

iv></div>

</div>

**<div class="modal
fade" id="albumModal"
tabindex="-1">**

**<div class="modal-
dialog modal-dialog-
centered">**

**<div
class="modal-content">**

**<div
class="modal-
header"><h5
class="modal-title">Add
to Album(s)</h5><button**

```
type="button"
class="close" data-
dismiss="modal">&time
s;</button></div>
```

```
<div
class="modal-body">
    <p>Select
albums to add this image
to:</p>
```

```
<div
id="album-selection-
list"></div>
```

```
</div>
```

```
<div
class="modal-footer">
```

```
<button
type="button"
class="btn btn-
secondary" data-
dismiss="modal">Cancel
</button>
```

```
<button
type="button"
```

`class="btn btn-primary"`

`id="confirm-album-add-`

`btn">Save to`

`Albums</button>`

`</div>`

`</div>`

`</div>`

`</div>`

`<script`

`src="https://code.jquery.`

`com/jquery-3.5.1.min.js"`

`></script>`

`<script`

`src="https://cdn.jsdelivrivr.`

`net/npm/@popperjs/core`

`@2.5.4/dist/umd/popper.`

`min.js"></script>`

`<script`

`src="https://stackpath.bo`

`otstrapcdn.com/bootstrap`

`/4.5.2/js/bootstrap.min.js`

`"></script>`



```
<script  
src="https://cdnjs.cloudflare.com/ajax/libs/cropper  
js/1.5.12/cropper.min.js">  
</script>
```

```
<script  
src="script.js"></script>  
</body>  
</html>
```

```
:root {  
  --dashboard-width:  
300px;  
  --bg-color: #f4f7fc;  
  --dashboard-bg: #ffffff;  
  --primary-color:  
#4a47a3;  
  --primary-light:  
#e8e7ff;  
  --text-dark: #333;  
  --text-light: #777;  
  --border-color:  
#e6e6e6;
```

```
--shadow: 0 4px 20px
-5px rgba(0,0,0,0.1);
}

body {

    background: var(--bg-
color);

    font-family: 'Poppins',
sans-serif;

    overflow: hidden;

    color: var(--text-dark);
}

/* --- Styled Scrollbars
(WebKit Browsers) --- */

::-webkit-scrollbar
{ width: 6px; }

::-webkit-scrollbar-track
{ background: #f1f1f1; }

::-webkit-scrollbar-
thumb { background:
#ccc; border-radius: 6px;
}
```

::-webkit-scrollbar-

thumb:hover

{ background: #aaa; }

/* --- Dashboard --- */

#dashboard {

width: var(--

dashboard-width);

background: var(--

dashboard-bg);

box-shadow: var(--

shadow);

flex-shrink: 0;

display: flex;

flex-direction: column;

border-right: 1px solid

var(--border-color);

}

#dashboard h3 { font-

weight: 600; color: var(--

primary-color); }

.dashboard-heading {

```
font-size: 0.8rem;

font-weight: 600;

text-transform:

uppercase;

color: var(--text-light);

padding: 15px 25px

5px;

}

#dashboard .form-control

{ border-color: var(--

border-color); }

#dashboard .form-

control:focus { border-

color: var(--primary-

color); box-shadow: 0 0 0

0.2rem rgba(74, 71, 163,

0.25); }

#dashboard .btn-primary

{ background-color:

var(--primary-color);

border: none; }
```

```
#dashboard .btn-success {
```

```
background-color:
```

```
#28a745; border: none; }
```

```
#folder-list, #album-list
```

```
{ overflow-y: auto;
```

```
padding: 0 10px; flex-
```

```
grow: 1; }
```

```
.folder-item {
```

```
padding: 12px 15px;
```

```
margin: 4px 0;
```

```
cursor: pointer;
```

```
color: var(--text-light);
```

```
transition: all 0.2s ease-
```

```
in-out;
```

```
border-radius: 8px;
```

```
display: flex;
```

```
align-items: center;
```

```
justify-content: space-
```

```
between;
```

```
}
```


.folder-item:hover

{ background: var(--bg-color); color: var(--text-dark); }

.folder-item.active

{ background: var(--primary-light); color: var(--primary-color); font-weight: 600; }

.folder-item .folder-name

{ display: flex; align-items: center; overflow: hidden; text-overflow: ellipsis; white-space: nowrap; }

.folder-item .folder-name

i { margin-right: 15px; }

.delete-folder-icon

{ color: #dc3545; opacity: 0; transition: opacity 0.2s; font-size: 0.9em; }

.folder-

item:hover .delete-folder-

icon { opacity: 0.6; }

.delete-folder-icon:hover

{ opacity: 1; }

/* --- Album Modal

Specific --- */

#album-selection-

list .form-check { margin-

bottom: 10px; }

#album-selection-

list .form-check-label

{ font-weight: 500; }

/* --- Gallery --- */

#gallery-container { flex-

grow: 1; overflow-y:

auto; }

#current-view-title { font-

weight: 600; color: var(--

text-dark); }

.gallery-item {

```
border-radius: 12px;  
  
box-shadow: 0 5px  
15px rgba(0,0,0,0.05);  
  
transition: all 0.3s ease;  
  
}
```

```
.gallery-item:hover  
  
{ transform:  
  
translateY(-5px); box-  
  
shadow: 0 12px 25px  
rgba(0,0,0,0.1); }  
  
.gallery-item .overlay  
  
{ position: absolute; top:  
  
0; left: 0; width: 100%;  
  
height: 100%;  
  
background:  
  
rgba(0,0,0,0.5); display:  
  
flex; justify-content:  
  
center; align-items:  
  
center; gap: 15px;  
  
opacity: 0; transition:  
  
opacity 0.3s ease; }
```

.gallery-

item:hover .overlay

{ opacity: 1; }

.overlay-icon { font-size:

1.5rem; color: white;

cursor: pointer;

transition: transform 0.2s

ease; }

.overlay-icon:hover

{ transform: scale(1.2); }

/* --- Modals --- */

.modal-header, .modal-

footer { border-color:

var(--border-color); }

.modal-content { border-

radius: 12px; border:

none; box-shadow: var(--

shadow); }

.modal-backdrop.show

{ opacity: 0.7; }

/* Glassmorphism modal

controls */

```
.modal-controls-  
container, .slider-control  
{ z-index: 1055;  
background: rgba(0, 0, 0,  
0.25); border: 1px solid  
rgba(255, 255, 255, 0.2);  
backdrop-filter:  
blur(5px); }  
.modal-controls-container  
{ position: fixed; top:  
20px; right: 30px;  
display: flex; gap: 10px;  
border-radius: 50px;  
padding: 5px; }  
.modal-control, .slider-  
control { color: white;  
width: 45px; height:  
45px; border-radius:  
50%; font-size: 1.1rem;  
cursor: pointer;  
transition: all 0.3s ease;  
display: flex; justify-  
content: center; align-
```

```
items: center; border:
none; background:
transparent; }

.modal-
control:hover, .slider-
control:hover
{ background: rgba(0, 0,
0, 0.5); transform:
scale(1.1); }

.slider-control { position:
fixed; top: 50%;
transform:
translateY(-50%); }

.slider-control.prev { left:
20px; }

.slider-control.next
{ right: 20px; }

.main-wrapper {
display: flex;
height: 100vh;
}

#dashboard {
```

```
width: var(--  
dashboard-width); /*  
Uses the width defined at  
the top of the CSS */  
flex-shrink: 0; /*  
Prevents the dashboard  
from shrinking */  
}
```

```
#gallery-container {  
flex-grow: 1; /* Allows  
the gallery to take up all  
remaining space */  
overflow-y: auto; /*  
Lets only the gallery  
scroll */  
}
```

```
$  
(document).ready(function()  
{
```

// --- DATA STORE & STATE MANAGEMENT

```
---  
  
let images = [];  
  
let folders = [];  
  
let albums = [];  
  
let activeView = { type:  
'folder', name: 'All  
Images' };  
  
let  
currentModalImages =  
[];  
  
let cropper = null;  
  
const modalImage =  
document.getElementById('modalImage');  
  
  
// *DEFINITIVE FIX*:  
  
This default data is now  
correctly included to  
ensure images display on  
first load.  
  
const defaultImages = [
```



```
{ id: 1, thumb:
'https://picsum.photos/id/
10/400/300', full:
'https://picsum.photos/id/
10/1920/1080', folder:
'Landscapes', isDeleted:
false },
{ id: 2, thumb:
'https://picsum.photos/id/
25/400/300', full:
'https://picsum.photos/id/
25/1920/1080', folder:
'Landscapes', isDeleted:
false },
{ id: 3, thumb:
'https://picsum.photos/id/
33/400/300', full:
'https://picsum.photos/id/
33/1920/1080', folder:
'Portraits', isDeleted:
false },
{ id: 4, thumb:
'https://picsum.photos/id/
```



48/400/300', full:

'https://picsum.photos/id/

48/1920/1080', folder:

'Cityscapes', isDeleted:

false },

{ id: 5, thumb:

'https://picsum.photos/id/

54/400/300', full:

'https://picsum.photos/id/

54/1920/1080', folder:

'Landscapes', isDeleted:

false },

{ id: 6, thumb:

'https://picsum.photos/id/

66/400/300', full:

'https://picsum.photos/id/

66/1920/1080', folder:

'Portraits', isDeleted:

false },

{ id: 7, thumb:

'https://picsum.photos/id/

71/400/300', full:

'https://picsum.photos/id/



```
71/1920/1080', folder:
'Animals', isDeleted: false
},
  { id: 8, thumb:
'https://picsum.photos/id/
88/400/300', full:
'https://picsum.photos/id/
88/1920/1080', folder:
'Cityscapes', isDeleted:
false },
  { id: 9, thumb:
'https://picsum.photos/id/
96/400/300', full:
'https://picsum.photos/id/
96/1920/1080', folder:
'Animals', isDeleted: false
}
];
const defaultFolders =
['Landscapes', 'Portraits',
'Cityscapes', 'Animals'];

function saveState() {
```

```
    try {  
  
        const appState =  
  
        { images, folders,  
  
        albums };  
  
  
        localStorage.setItem('photoAppState',  
  
        JSON.stringify(appState)  
  
        );  
  
        } catch (e)  
  
        { console.error("Error  
saving state to  
localStorage", e); }  
  
    }  
  
  
    function loadState() {  
  
        try {  
  
            const savedState =  
  
            localStorage.getItem('photoAppState');  
  
            if (savedState) {  
  
                const appState =  
  
                JSON.parse(savedState);
```

```
        if (!appState || !
appState.images || !
appState.folders ||
appState.images.length
=== 0) {
            images =
defaultImages;
            folders =
defaultFolders;
            albums = [];
        } else {
            images =
appState.images;
            folders =
appState.folders;
            albums =
appState.albums || [];
        }
    } else {
        images =
defaultImages;
        folders =
defaultFolders;
```

```
        albums = [];  
    }  
    } catch (e) {  
  
        console.error("Error  
loading state from  
localStorage", e);  
        images =  
defaultImages; folders =  
defaultFolders; albums =  
[];  
    }  
}  
  
// --- RENDER  
FUNCTIONS ---  
  
function  
renderFolders() {  
    const folderList = $  
('#folder-list');  
    folderList.empty();  
    ['All  
Images', ...folders,
```

```

'Trash'].forEach(folder
=> {

    const isDeletable =

    ![ 'All Images',
    'Trash'].includes(folder);

    const deleteIcon =

    isDeletable ? <i

    class="fas fa-times

    delete-folder-icon"

    title="Delete

    Folder"></i> : "";

    const icon = folder

    === 'Trash' ? 'fa-trash-

    alt' : (folder === 'All

    Images' ? 'fa-images' :

    'fa-folder');

    const item = $(<li

    class="folder-item $

    {activeView.type ===

    'folder' &&

    activeView.name ===

    folder ? 'active' : ''}

    data-folder="$

```

```

{folder}"><span
class="folder-name"><i
class="fas ${icon}"></i>
${folder}</span>$
{deleteIcon}</li>);

```

```

folderList.append(item);

});

}

```

```

function

renderAlbums() {

    const albumList = $

('#album-list');

    albumList.empty();

```

```

albums.forEach(album

=> {

    const deleteIcon =

<i class="fas fa-times

delete-folder-icon delete-

album-icon"

```



```
title="Delete

Album"></i>;

    const item = $(<li

class="folder-item $

{activeView.type ===

'album' &&

activeView.name ===

album.name ? 'active' :

""} data-album-name="$

{album.name}"><div

class="folder-name"><i

class="fas fa-book"></i>

${album.name}</div>$

{deleteIcon}</li>);

albumList.append(item);

    });

}

function

renderImages() {

    const imageGrid = $

('#image-grid');
```

```
imageGrid.empty();

$('#current-view-
title').text(activeView.name);

let imagesToRender
= [];

if (activeView.type
=== 'folder') {

    if
(activeView.name ===
'All Images')
imagesToRender =
images.filter(img => !
img.isDeleted);

    else if
(activeView.name ===
'Trash') imagesToRender
= images.filter(img =>
img.isDeleted);

    else
imagesToRender =
images.filter(img =>
```

```
img.folder ===  
  
activeView.name && !  
  
img.isDeleted);  
  
    } else if  
  
(activeView.type ===  
  
'album') {  
  
    const album =  
  
albums.find(a => a.name  
  
=== activeView.name);  
  
    if (album)  
  
imagesToRender =  
  
images.filter(img =>  
  
album.imageIds.includes(  
  
img.id) && !  
  
img.isDeleted);  
  
    }  
  
    currentModalImages  
  
= imagesToRender;  
  
  
    if  
  
(imagesToRender.length  
  
=== 0) {
```

```

imageGrid.html('<p
class="col-12 text-muted
mt-4">This view is
empty.</p>');

return;

}

```

```

imagesToRender.forEach
(img => {

    const actions =

    activeView.name ===

    'Trash'

    ? <i class="fas
fa-undo-alt overlay-icon
restore-icon"
title="Restore"></i>

    : `<i class="fas
fa-plus-square overlay-
icon add-to-album-icon"
title="Add to
Album"></i>

```

```

<i class="fas
fa-edit overlay-icon edit-
icon" title="Move to
folder"></i>

```

```

<i class="fas
fa-trash-alt overlay-icon
delete-icon"
title="Delete"></i>`;

```

```

const itemHTML
= <div class="col-lg-4 col-
md-6 col-sm-12
mb-4"><div
class="gallery-item"
data-id="$
{img.id}"><div
class="overlay"><i
class="fas fa-expand
overlay-icon view-icon"
title="View"></i><i
class="fas fa-download
overlay-icon download-

```

icon" title="Download

Original"></i>\${actions}

</div></div></div>;

imageGrid.append(itemH

TML);

});

}

function renderAll()

{ renderFolders();

renderAlbums();

renderImages(); }

// --- EVENT

HANDLERS ---

\$('#upload-

btn').on('click', () => \$

('#file-input').click());

\$('#file-

input').on('change',

function(event) {

```
const files =  
event.target.files;  
for (const file of files)  
{  
    const reader = new  
    FileReader();  
    reader.onload =  
    function(e) {  
        const newImage  
= {  
            id:  
Date.now() +  
Math.random(),  
            thumb:  
e.target.result,  
            full:  
e.target.result,  
            folder:  
(activeView.type !==  
'folder' || ['All Images',  
'Trash'].includes(activeVi  
ew.name)) ? (folders[0] ||
```

```
'Uncategorized') :  
  
    activeView.name,  
  
        isDeleted:  
  
        false  
  
        };  
  
    images.push(newImage);  
  
        renderImages();  
  
        saveState();  
  
        };  
  
    reader.readAsDataURL(f  
  
    ile);  
  
        }  
  
        $(this).val('');  
  
        });  
  
        $('#add-folder-  
  
    btn').on('click', function()  
  
    {  
  
        const folderName =  
  
        $('#new-folder-  
  
    name').val().trim();
```



```
        if (folderName && !
folders.includes(folderNa
me) && !['All Images',
'Trash'].includes(folderN
ame)) {

folders.push(folderName)
;

        renderFolders();

        saveState();

        $('#new-folder-
name').val('');

        } else { alert("Folder
name is empty, a reserved
name, or already
exists."); }

    });

    $('#add-album-
btn').on('click', function()
{
```

```
const albumName =  
  
$('#new-album-  
name').val().trim();  
  
if (albumName && !  
albums.find(a => a.name  
=== albumName)) {  
  
    albums.push({ id:  
Date.now(), name:  
albumName, imageIds: []  
});  
  
    renderAlbums();  
  
    saveState();  
  
    $('#new-album-  
name').val('');  
  
    } else { alert("Album  
name is empty or already  
exists."); }  
  
});  
  
$('#folder-  
list').on('click', '.folder-  
item', function()  
{ activeView = { type:
```

```
'folder', name: $  
(this).data('folder') });  
  
renderAll(); });  
  
$('#album-  
list').on('click', '.folder-  
item', function()  
{ activeView = { type:  
'album', name: $  
(this).data('album-name')  
}; renderAll(); });
```

```
$('#folder-  
list').on('click', '.delete-  
folder-icon', function(e) {  
    e.stopPropagation();  
    const folderToDelete  
= $(this).closest('.folder-  
item').data('folder');  
    if (confirm(Are you  
sure you want to delete  
the "${folderToDelete}"  
folder? All images inside
```

will be moved to the

Trash.)) {

images.forEach(img =>

{ if (img.folder ===

folderToDelete)

img.isDeleted = true; });

folders =

folders.filter(f => f !==

folderToDelete);

activeView =

{ type: 'folder', name:

'All Images' };

renderAll();

saveState();

}

});

\$('#album-

list').on('click', '.delete-

album-icon', function(e) {

e.stopPropagation();

```
const albumToDelete

= $(this).closest('.folder-
item').data('album-
name');

if (confirm(Are you
sure you want to delete
the "${albumToDelete}"
album? The images will
NOT be deleted.)) {

    albums =
albums.filter(a => a.name
!== albumToDelete);

    if (activeView.type
=== 'album' &&
activeView.name ===
albumToDelete) {

        activeView =
{ type: 'folder', name:
'All Images' };

    }

    renderAll();

    saveState();

}
```

```
});

$('#image-
grid').on('click',
'.overlay-icon',
function(e) {

    e.stopPropagation();

    const imageId =
parseFloat($
(this).closest('.gallery-
item').data('id'));

    const image =
images.find(img =>
img.id === imageId);

    if (!image) return;

    if ($
(this).hasClass('view-
icon')) {

        const startIndex =
currentModalImages.find
Index(img => img.id ===
imageId);
```

```
        if (startIndex > -1)

openModal(startIndex);

    } else if ($)

(this).hasClass('download

-icon')) {

triggerDownload(image.f

ull, image-$

{image.id}.jpg);

    } else if ($)

(this).hasClass('add-to-

album-icon')) {

        const

albumSelectionList = $

('#album-selection-list');

albumSelectionList.empty

y();

        if (albums.length

=== 0) {

albumSelectionList.html(

'<p>No albums created
```

yet. Please create an

album first.</p>');

} else {

albums.forEach(album

=> {

const

isChecked =

album.imageIds.includes(
imageId) ? 'checked' : '';

albumSelectionList.appe

nd(<div class="form-

check"><input

class="form-check-

input" type="checkbox"

value="{album.id}"

id="album-{album.id}"

{isChecked}><label

class="form-check-label"

for="album-\$

{album.id}">\$


```
{album.name}</label></d
```

```
iv>);
```

```
});
```

```
}
```

```
$
```

```
(#albumModal').data('i
```

```
mageId',
```

```
imageId).modal('show');
```

```
} else if ($
```

```
(this).hasClass('edit-
```

```
icon')) {
```

```
const folderSelect
```

```
= $('#folder-select');
```

```
folderSelect.empty();
```

```
folders.forEach(f
```

```
=> { if(f !== image.folder)
```

```
folderSelect.append(<opti
```

```
on
```

```
value="{{$f}}">{{$f}</optio
```

```
n>); });
```

```
$
```

```
(#moveModal').data('im
```

```
ageId',  
  
imageId).modal('show');  
  
    } else if ($  
  
(this).hasClass('delete-  
  
icon')) {  
  
        image.isDeleted =  
  
true;  
  
        renderImages();  
  
        saveState();  
  
    } else if ($  
  
(this).hasClass('restore-  
  
icon')) {  
  
        image.isDeleted =  
  
false;  
  
        renderImages();  
  
        saveState();  
  
    }  
  
});  
  
  
$('#confirm-move-  
  
btn').on('click', function()  
  
{
```

```
const imageId =  
parseFloat($  
('#moveModal').data('im  
ageId'));  
  
const newFolder = $  
('#folder-select').val();  
  
const image =  
images.find(img =>  
img.id === imageId);  
  
if (image &&  
newFolder) image.folder  
= newFolder;  
  
$  
('#moveModal').modal('h  
ide');  
  
renderImages();  
  
saveState();  
  
});  
  
$('#confirm-album-  
add-btn').on('click',  
function() {
```

```
const imageId =  
parseFloat($  
('#albumModal').data('i  
mageId'));  
$('#album-selection-  
list .form-check-  
input').each(function() {  
const albumId =  
parseInt($(this).val());  
const album =  
albums.find(a => a.id  
=== albumId);  
if (!album) return;  
const  
imageInAlbum =  
album.imageIds.includes(  
imageId);  
if ($  
(this).is(':checked') && !  
imageInAlbum)  
album.imageIds.push(im  
ageId);
```

```
        else if (!$  
        (this).is(':checked') &&  
        imageInAlbum)  
        album.imageIds =  
        album.imageIds.filter(id  
        => id !== imageId);  
    });  
    $  
    ('#albumModal').modal('  
    hide');  
    saveState();  
    });  
  
    // --- UTILITY &  
    MODAL LOGIC ---  
    function  
    triggerDownload(url,  
    filename) {  
        fetch(url, { mode:  
        'cors' })  
        .then(response =>  
        response.blob())  
        .then(blob => {
```

```
const link =  
  
document.createElement(  
  
"a");  
  
link.href =  
  
URL.createObjectURL(b  
  
lob);  
  
link.download =  
  
filename;  
  
link.click();  
  
  
URL.revokeObjectURL(l  
  
ink.href);  
  
}).catch(() =>  
  
alert('Could not  
  
download image. This  
  
may be due to browser  
  
security restrictions.'));  
  
}  
  
$  
  
('#downloadBtn').on('cl  
  
ick', () => {
```

```
const currentImage  
=  
currentModalImages[cur  
rentModalIndex];  
  
if (currentImage)  
triggerDownload(current  
Image.full, image-$  
{currentImage.id}.jpg);  
});
```

```
let currentModalIndex  
= 0;  
  
function  
openModal(startIndex) {  
currentModalIndex  
= startIndex;  
  
$  
('#imageModal').modal('s  
how');  
}
```

```
function  
updateModalImage() {
```

```
        if
        (currentModalImages.length > 0 &&
        currentModalImages[currentModalIndex]) {
            const imageUrl =
            currentModalImages[currentModalIndex].full;
            if (cropper)
            cropper.replace(imageUrl
            );
            else
            modalImage.src =
            imageUrl;
        }
    }
```

```
    $('#nextBtn').on('click',
    function() { if
    (currentModalImages.length > 0)
    { currentModalIndex =
    (currentModalIndex + 1)
```



```
%  
  
currentModalImages.leng  
th; updateModalImage();  
} });  
  
$  
( '#prevBtn').on('click',  
function() { if  
(currentModalImages.len  
gth > 0)  
{ currentModalIndex =  
(currentModalIndex - 1 +  
currentModalImages.leng  
th) %  
currentModalImages.leng  
th; updateModalImage();  
} });  
  
$('#zoom-  
in').on('click', () =>  
cropper &&  
cropper.zoom(0.1));  
  
$('#zoom-  
out').on('click', () =>
```

```
cropper &&
cropper.zoom(-0.1));

$
($('#imageModal').on('shown.bs.modal', function() {
    if (cropper)
    cropper.destroy();

    cropper = new
    Cropper(modalImage,
    { viewMode: 1,
    dragMode: 'move',
    background: false,
    autoCrop: false, ready: ()
=> $
(modalImage).css('opacity', 1) });

updateModalImage();

});

$
($('#imageModal').on('hidden.bs.modal', function() {
```

```
        if(cropper)

        { cropper.destroy();

        cropper = null; }

        $

        (modalImage).css('opacit

        y', 0).attr('src', '');

        });

        $

        ($('#cropBtn').on('click',

        function() {

                if (!cropper) return;

                if (cropper.cropped)

                {

                        cropper.getCroppedCanv

                        as({ width: 1024, height:

                        1024 }).toBlob(blob => {

                                const link =

                                document.createElement(

                                'a');
```

```
link.href =  
  
URL.createObjectURL(b  
lob);
```

```
link.download =  
  
cropped-image-$  
{Date.now()}.png;  
  
link.click();
```

```
URL.revokeObjectURL(l  
ink.href);
```

```
});  
  
cropper.clear();  
  
} else {  
  
cropper.crop();  
  
}  
  
});
```

```
// ---
```

```
INITIALIZATION ---
```

```
function init() {  
  
loadState();  
  
renderAll();  
  
}
```

```
init();  
  
});
```

13. Screenshots of Final Output

11. Conclusion

This Photo Gallery project demonstrates practical application of front-end web development concepts, offering a user-focused, responsive experience. Through strong UI/UX, robust state management, and modular architecture, the system is well-positioned for further enhancement and real-world deployment in personal, organizational, or educational contexts

- A feature-rich, organized, and responsive photo management system that supports both novice and power users.
- Highly interactive and visually appealing UI with fast client-side performance.

Future Enhancements

- Integrate user authentication for private galleries.
- Use cloud storage or server-side backend for multi-user support.
- Add image tagging and advanced search.
- Enable batch operations and drag-and-drop organization.
- Implement lazy-loading and progressive image optimization for scalability.

12. References

- General Online Photo Gallery project examples and layouts:
<https://www.slideshare.net/slideshow/online-photo-gallery/66367347>
- Simple Image Gallery Web App guides and PHP projects:
<https://www.kashipara.com/project/php/9414/simple-image-gallery-web-app>
- Responsive Image Gallery how-tos using HTML, CSS, jQuery, and Bootstrap:
<https://www.geeksforgeeks.org/jquery/how-to-create-responsive-image-gallery-using-html-css-jquery-and-bootstrap/>

- Responsive Image Galleries with best practices:
<https://www.browserstack.com/guide/responsive-image-gallery-html-css>
- Detailed Image Gallery Web App system reports:
<https://www.scribd.com/document/858869031/image-gallery-report>
- Cropper.js library documentation and usage examples:
<https://github.com/fengyuanchen/cropperjs>
- General UI/UX and front-end development principles for modals and galleries:
<https://dev.to/webdesignersusa/building-a-beginners-web-development-project-interactive-photo-gallery-i8o>
- Local Storage for Web Apps and client-side persistence guides:
<https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage>
-