

# Galleria - Smart Photo Gallery Documentation

## Project Overview

Galleria is a modern, feature-rich photo gallery web application that provides an intuitive interface for discovering, organizing, and sharing stunning photography<sup>[1]</sup>. Built with cutting-edge web technologies, it offers a seamless user experience with advanced features like private collections, album management, and social sharing capabilities<sup>[1]</sup>.

## Table of Contents

1. [Features](#)
2. [Technology Stack](#)
3. [Project Structure](#)
4. [Installation & Setup](#)
5. [API Configuration](#)
6. [Core Components](#)
7. [User Interface](#)
8. [Data Management](#)
9. [Responsive Design](#)
10. [Browser Compatibility](#)

## Features

### Core Functionality

- **Dynamic Image Loading:** Integrates with Unsplash API for high-quality photo discovery<sup>[1]</sup>
- **Advanced Search:** Real-time search with intelligent suggestions and filtering<sup>[1]</sup>
- **Multiple View Modes:** Grid and list layouts with adaptive sizing<sup>[1]</sup>
- **Fullscreen Lightbox:** Immersive viewing experience with keyboard navigation<sup>[1]</sup>

### User Management

- **Private Gallery:** Secure user authentication with 4-digit passcode system<sup>[1]</sup>
- **Personal Collections:** Save favorite images to private galleries<sup>[1]</sup>
- **User Statistics:** Track activities including searches, downloads, and saved images<sup>[1]</sup>

## Organization Features

- **Album Management:** Create, organize, and manage custom photo albums<sup>[1]</sup>
- **Search History:** Automatic tracking and management of search queries<sup>[1]</sup>
- **Smart Categorization:** Intelligent image organization and tagging<sup>[1]</sup>

## Social Features

- **Multi-Platform Sharing:** Share to Facebook, Twitter, LinkedIn, and Pinterest<sup>[1]</sup>
- **Direct Downloads:** High-resolution image downloading capabilities<sup>[1]</sup>
- **Link Sharing:** Generate shareable links for individual images<sup>[1]</sup>

## Visual Experience

- **Neon Glass Theme:** Modern UI with dynamic glass morphism effects<sup>[1]</sup>
- **Smooth Animations:** CSS-powered transitions and hover effects<sup>[1]</sup>
- **Context Menu:** Right-click navigation for quick actions<sup>[1]</sup>

## Technology Stack

### Frontend Technologies

Technology	Version	Purpose
HTML5	Latest	Semantic markup and structure <sup>[1]</sup>
CSS3	Latest	Styling, animations, and responsive design <sup>[1]</sup>
JavaScript	ES6+	Application logic and interactivity <sup>[1]</sup>

### External APIs

- **Unsplash API:** High-quality photo search and retrieval<sup>[1]</sup>
- **Browser APIs:** Fullscreen, Intersection Observer, Local Storage<sup>[1]</sup>

### Architecture Patterns

- **Modular Design:** Class-based JavaScript architecture<sup>[1]</sup>
- **Component-Based:** Reusable UI components and utilities<sup>[1]</sup>
- **MVC Pattern:** Separation of concerns with clear data flow<sup>[1]</sup>

## Project Structure

```
galleria/
├── index.html      # Main application structure (525 lines)
├── style.css       # Styling and animations (2,166 lines)
├── script.js       # Application logic (1,680 lines)
└── README.md       # Project documentation
```

## File Breakdown

### index.html

The main HTML file contains the complete application structure including<sup>[1]</sup>:

- Navigation bar with five main sections
- Hero section with search functionality
- Adaptive gallery grid layout
- Modal components for user interactions
- Fullscreen lightbox viewer
- About section with project information

### style.css

Enhanced CSS file featuring<sup>[1]</sup>:

- Neon glass visual effects with custom animations
- Responsive grid layout system
- Dark theme with gradient backgrounds
- Mobile-first responsive design (768px, 480px breakpoints)
- 10 custom CSS animations (neon-glow, shimmer, pulse-glow, etc.)
- 176 unique CSS custom properties

### script.js

Modular JavaScript application with nine core classes<sup>[1]</sup>:

- **Utils**: Utility functions for formatting and animations
- **StatsManager**: User statistics and activity tracking
- **Galleria**: Main gallery functionality and image management
- **PrivateGallery**: Private collection with authentication
- **AlbumManager**: Album creation and management system
- **ShareManager**: Social media sharing functionality

- **SearchHistoryManager**: Search history tracking
- **NavigationHandler**: Section navigation and routing
- **ContextMenuHandler**: Right-click context menu system

## Installation & Setup

### Prerequisites

- Modern web browser (Chrome 90+, Firefox 88+, Safari 14+)
- Internet connection for API access
- Local web server (optional, recommended for development)

### Quick Start

#### 1. Download Project Files

```
# Clone or download the project files
# Ensure all three files are in the same directory
```

#### 2. API Key Configuration

```
// In script.js, update the CONFIG object
const CONFIG = {
  API_KEY: 'your-unsplash-api-key',
  BASE_URL: 'https://api.unsplash.com',
  // ... other configuration options
};
```

#### 3. Local Development

```
# Option 1: Simple HTTP server (Python)
python -m http.server 8000

# Option 2: Node.js http-server
npm install http-server
npx http-server

# Option 3: Live Server (VS Code extension)
# Right-click index.html → Open with Live Server
```

#### 4. Access Application

Open <http://localhost:8000> in your web browser<sup>[1]</sup>

## API Configuration

## Unsplash API Setup

The application requires an Unsplash API key for full functionality<sup>[1]</sup>:

### 1. Register at Unsplash Developers

- Visit <https://unsplash.com/developers>
- Create a developer account
- Register a new application

### 2. Configuration Parameters

```
const CONFIG = {  
  API_KEY: 'your-access-key-here',  
  BASE_URL: 'https://api.unsplash.com',  
  DEMO_MODE: false,  
  IMAGES_PER_PAGE: 30,  
  SEARCH_DELAY: 500,  
  ANIMATION_DELAY: 100,  
  MAX_HISTORY_ITEMS: 50  
};
```

## Demo Mode

If no API key is available, the application includes a demo mode with sample images<sup>[1]</sup>:

```
CONFIG.DEMO_MODE = true; // Enables offline demonstration
```

## Core Components

### Navigation System

The application features a fixed navigation bar with five main sections<sup>[1]</sup>:

- **Home:** Main gallery and search interface
- **About:** Project information and technical details
- **Private:** Personal gallery with authentication
- **Albums:** Custom album management
- **History:** Search history tracking

### Search Functionality

Advanced search system with multiple features<sup>[1]</sup>:

- Real-time search as you type (500ms debounce)
- Search suggestions for popular terms
- Sort options: relevance, latest, most popular

- Search history with timestamp tracking
- Clear and repeat search functionality

## Image Display

Responsive gallery system with<sup>[1]</sup>:

- Adaptive grid layout (350px minimum column width)
- Lazy loading with Intersection Observer
- Hover overlays with image statistics
- Multiple view modes (grid/list)
- Smart image sizing based on aspect ratios

## Lightbox Viewer

Full-featured image viewer including<sup>[1]</sup>:

- Fullscreen display with navigation
- Keyboard shortcuts (arrows, escape, f, d, s)
- Image metadata and statistics
- Download and sharing options
- Previous/next navigation

## User Interface

### Visual Design

The application features a modern neon glass theme with<sup>[1]</sup>:

- Dark background with gradient overlays
- Glowing neon borders and effects
- Smooth CSS animations and transitions
- Glass morphism effects with backdrop blur
- Responsive color scheme with CSS custom properties

### Interaction Patterns

- **Hover Effects:** Dynamic overlays and scaling animations
- **Click Actions:** Image selection, navigation, and modal triggers
- **Keyboard Shortcuts:** Quick navigation and actions
- **Context Menu:** Right-click for additional options
- **Touch Gestures:** Mobile-friendly touch interactions

## Modal System

Four main modal types for user interactions<sup>[1]</sup>:

- **Create Album Modal:** Album creation with name and description
- **Add to Album Modal:** Image-to-album assignment
- **Share Modal:** Social media sharing options
- **Authentication:** Private gallery login interface

## Data Management

### Local Storage

The application uses browser localStorage for data persistence<sup>[1]</sup>:

Storage Key	Purpose	Data Structure
galleriaStats	User statistics	JSON object with counters
galleryUsers	Private gallery users	User profiles with credentials
galleriaAlbums	Album collections	Album metadata and image lists
galleriaSearchHistory	Search queries	Array of search objects with timestamps

## Data Models

### User Statistics

```
{
  totalImages: number,
  totalSearches: number,
  totalDownloads: number,
  totalAlbums: number,
  totalFavorites: number,
  lastActivity: ISO timestamp
}
```

### Album Structure

```
{
  id: string,
  name: string,
  description: string,
  images: array,
  createdAt: ISO timestamp,
  updatedAt: ISO timestamp
}
```

## Search History

```
{
  query: string,
  timestamp: ISO timestamp,
  id: unique identifier
}
```

## Responsive Design

### Breakpoints

The application implements a mobile-first responsive design with two main breakpoints<sup>[1]</sup>:

Breakpoint	Target Devices	Layout Changes
768px	Tablets and below	Simplified navigation, stacked layouts
480px	Mobile phones	Single column grid, condensed UI

### Responsive Features

- **Flexible Grid:** Auto-adjusting column counts based on screen size
- **Scalable Typography:** Clamp() functions for responsive text sizing
- **Touch-Friendly:** Larger touch targets for mobile devices
- **Optimized Navigation:** Collapsible menu items on smaller screens
- **Adaptive Modals:** Full-screen modals on mobile devices

### Browser Compatibility

#### Minimum Requirements

- **Chrome:** Version 90+ (April 2021)
- **Firefox:** Version 88+ (April 2021)
- **Safari:** Version 14+ (September 2020)
- **Edge:** Version 90+ (April 2021)

#### Required Browser APIs

- CSS Custom Properties (CSS Variables)
- ES6+ JavaScript features (Classes, Arrow Functions, Template Literals)
- Intersection Observer API
- Local Storage API
- Fullscreen API



- Fetch API for network requests

## Progressive Enhancement

The application gracefully degrades for older browsers<sup>[1]</sup>:

- Fallback to demo mode if API fails
- Basic functionality without advanced CSS features
- Alternative layouts for unsupported grid properties

**Project Statistics:** 4,371 total lines of code across HTML (525), CSS (2,166), and JavaScript (1,680) files<sup>[1]</sup>.

**Last Updated:** Current version includes all modern web standards and best practices for 2024-2025—