Globant Piscine

ImageGallery

Summary: Keep calm and say 'API cheese'!

Version: 2

Contents

Chapter I

A word about this Project

Create an interactive web application that allows users to browse and search for high-quality photos using the **Unsplash** public API. Through a clean and responsive interface, users can explore an endless gallery of stunning images and search by keywords to find specific content. Additionally, users can authenticate via OAuth to unlock extra features, such as marking photos as favorites and managing their personalized collection.

This project will help you gain hands-on experience with API integration, OAuth 2.0 authentication, and modern front-end development using pure HTML, CSS, and JavaScript.

Chapter II

Introduction

What this Project will show you:

- **HTML5**, **CSS3**, and **JavaScript ES6** (Vanilla): Learn the fundamentals of building web pages, styling them with CSS, and adding interactivity using modern JavaScript without relying on frameworks or libraries.

- **TypeScript** (Recommended): Gain an understanding of how to improve code quality and maintainability by using TypeScript's static typing features.

- **OAuth2.0** Authentication: Learn how to implement secure user authentication using the OAuth 2.0 protocol, allowing users to log in to the app with third-party accounts.

- **Unsplash** API Integration: Learn how to interact with external APIs by making HTTP requests to retrieve and display data, such as high-quality photos from Unsplash.

- Managing State and User Actions: Understand how to manage user interactions, such as searching for photos and marking them as favorites, with the use of tokens and local storage for persistent data.

- Building Responsive and Accessible Web Applications: Master techniques for creating **mobile-first**, **responsive** designs that meet WCAG accessibility standards, ensuring the app is usable across devices and by all users.

- Setting up a complete development environment using using Docker and Docker Compose.

- Gaining experience in building, running, and deploying production-grade applications.

Chapter III

General instructions

Unless explicitely specified, the following rules will apply for every project of this Piscine.

- This subject is the one and only trustable source. Don't trust any rumor.
- This subject can be updated up to one hour before the turn-in deadline.
- The assignments in a subject must be done in the given order. Later assignments won't be rated unless all the previous ones are perfectly executed.
- Be careful about the access rights of your files and folders.
- Your assignments WON'T be evaluated by your Piscine peers.
- You must not leave in your turn-in your workspace any file other than the ones explicitly requested By the assignments. If the assignment don't precise them, put only the necessary ones to run your Project.
- Using some API Key or Token? Keep them for you! Do not push them on your repository.
- You have a question? Ask your left neighbor. Otherwise, try your luck with your right neighbor.
- Every technical answer you might need is available in the `man` or on the Internet.
- You must read the examples thoroughly. They can reveal requirements that are not obvious in the assignment's description.
- By Loky, by Freya! Use your brain!!!

Chapter IV

Glossary

This glossary is designed to help you quickly understand the main technologies and concepts you will work with in **ImageGallery**. In Chapter VII, you'll find extra resources and tutorials to get started confidently.

- **JavaScript (JS)**: Programming language that allows you to add logic and interactivity to your webpage.
- **TypeScript**: A programming language that extends JavaScript, used mainly in frontend and backend web development for greater safety and scalability.
- **HTML (HyperText Markup Language)**: The standard language used to structure the content of a webpage. It defines the elements you see in the browser, such as buttons, grids, or text...
- **CSS (Cascading Style Sheets)**: Language used to style and visually design web pages — controls colors, fonts, layout, and animations.
- **Tailwind**: A CSS framework for creating responsive and customizable user interfaces, commonly used in frontend design.
- **Docker**: A platform that allows you to create and run applications inside isolated containers.
- **Docker Compose**: A tool for defining and managing multiple Docker containers at once.
- **OAuth2**: A standard authorization protocol that provides secure access to resources on behalf of a user and helps protect user credentials.
- **Unsplash API**: A public API that provides access to millions of free, high-quality photos. It allows developers to search, retrieve, and display images in their applications.
- **JSON (JavaScript Object Notation)**: A lightweight data format used for exchanging information between a client (browser) and a server.
- **SPA (Single Page Application)**: A web application that loads a single HTML page and dynamically updates its content without reloading the entire page.

- **JWT (JSON Web Token)**: A method for securely transmitting information between two parties in JSON format.
- **API (Application Programming Interface)**: An interface that allows two applications to communicate and share data.
- **WCAG 2.1 Level AA**: A set of web accessibility guidelines designed to make websites more usable for people with disabilities.
- **Insomnia / Postman**: API testing tools used during development and debugging.

Chapter V

Mandatory part

Globant	Exercise 00
Image Gallery	
Turn-in directory : *ex*00/	
Files to turn in : `All needed files to run your Project and nothing else`	
Allowed functions : `None`	

- **Technologies to Use**
  - **HTML5**
  - **CSS3**
  - JavaScript (ES6, Vanilla — no libraries or frameworks). **TypeScript** is optional but recommended.
  - **Docker** is mandatory to deliver the project. Please provide a Dockerfile and a docker-compose.yml file to run the project.
  - You must include a **README.md** file in your repository explaining briefly the project and how to run it.
- **Main Functions**
  - **OAuth** Authentication: Implement the authentication process using OAuth 2.0 so users can log in with their Unsplash account.
  - Photo Display: Show a gallery of photos obtained from the Unsplash API.
  - Photo Search: Allow users to search for photos by keywords.
  - Favorites: Once authenticated, users can mark photos as favorites.
- **API to Use**
  - Unsplash API: Provides access to a wide collection of high-quality photos.

- **Best Practices**
  - Ensure the application is usable on both mobile and desktop devices.
  - Write clean, well-commented code.
  - Properly handle errors and input validation.

`You may want to take a look at every technologies referenced in the project description before starting.`

`In any of the projects that require the consumption of an API you can leverage the use of a library to help you with it like swe or TanStack Query`

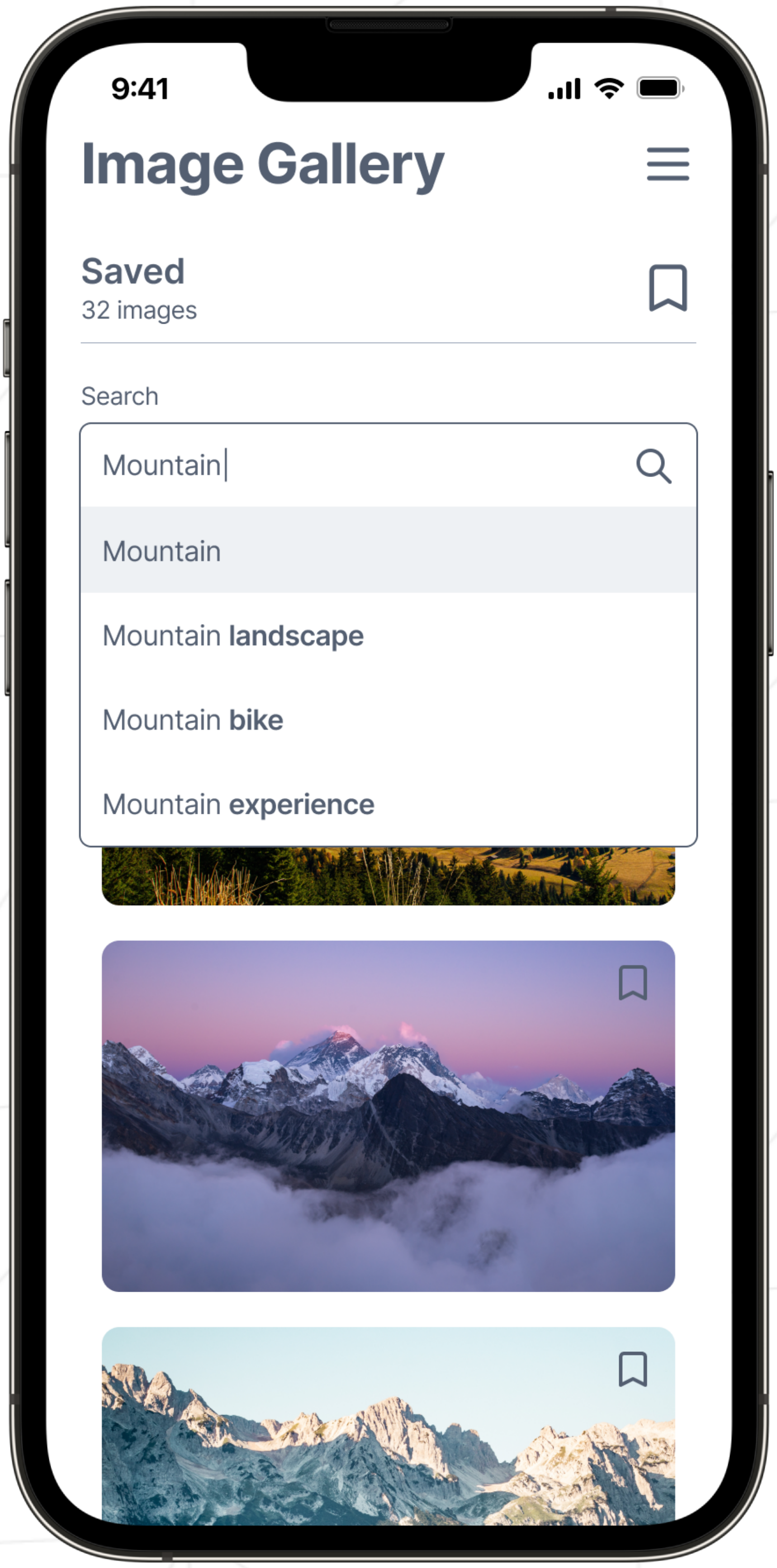

Figure V.1: Image Gallery

Chapter VI

Bonus Part

Once mandatory requirements are implemented in the application, you can add any additional features you see fit. The only limit is your imagination, but be respectful of your time.

Chapter VII

Resources

This chapter provides a collection of useful resources to help you get started with the technologies used in this project.

You'll find links to articles, tutorials, and videos — both in English and Spanish — as well as simple code examples to help you understand key concepts and start experimenting on your own.

These materials are meant to guide you a little if you get stuck, but we highly recommend that you keep exploring on your own.

- **Docker**
  - Article 1
  - Article 2
  - Article 3
  - Video
- **APIs**
  - Article 1
  - Article 2
  - Article 3
  - Article 4
  - Video
- **Typescript**
  - Article
  - Video 1
  - Video 2
  - Video 3

We provide you an **example** below as a basic example to illustrate **TypeScript** usage. It does not include the full project setup. You still need to:

- Create an `index.html` file that loads your compiled JavaScript file.
- Set up your `tsconfig.json` for TypeScript compilation.
- Install TypeScript globally or in your project (`npm install typescript -save-dev`).
- Compile your code with `npx tsc`.

```
const button = document.getElementById("clickMe") as HTMLButtonElement;
const message = document.getElementById("message") as HTMLParagraphElement;

let count: number = 0;

button.addEventListener("click", (): void => {
  count++;
  message.textContent = ‘You clicked ${count} times!‘;
});
```

Need some extra help? Check out these resources: OAuth 2.0 Documentation, Passport.js OAuth Documentation, Google OAuth 2.0 Documentation.

Chapter VIII

Submission

- Create a git repo (Github, Gitlab, Bitbucket, etc) and add your project files to it.
- Copy the link to your repository and paste it in the project submission form.
- Project submission form: TYPEFORM

Please note, no modifications made on the repo after the form is sent will be taken into account for the evaluation.

No Peer evaluation for this Piscine, but we strongly recommend reviewing and using the evaluation sheet we give you to check if your project meets all the requirements.