

DOCUMENTAZIONE TECNICA - POKECATCH

INDICE

- [Introduzione](#)
 - [Obiettivi del progetto](#)
 - [Struttura del progetto](#)
 - [Componenti principali](#)
 - [Home Page](#)
 - [Pokémon List](#)
 - [Pokédex Personale](#)
 - [Tecnologie utilizzate](#)
 - [Funzionalità principali](#)
 - [Gestione della cattura](#)
 - [Gestione Pokédex](#)
 - [Notifiche Modal](#)
 - [Persistenza dati](#)
 - [Installazione ed esecuzione](#)
 - [Requisiti di sistema](#)
-

INTRODUZIONE

PokéCatch è un sito Web interattivo che consente agli utenti di esplorare una lista di Pokémon, catturarli e gestire il proprio Pokédex personale. Il progetto è stato realizzato per risultare semplice e basato su tecnologie front-end moderne.

Le **caratteristiche principali** di questo progetto sono:

- Interfaccia costruita con Bootstrap e stili personalizzati (style.css)
 - Persistenza dei dati tramite l'utilizzo del localStorage
 - Gestione interattiva dei Pokémon catturati
-

OBIETTIVI DEL PROGETTO

Il progetto è stato realizzato basandosi sui seguenti obiettivi:

- Fornire **un'interfaccia intuitiva** e funzionale all'utente in modo tale da poter esplorare e gestire i Pokémon
 - Progettare una **struttura chiara e modulare** per semplificare le manutenzioni future
 - Utilizzare **tecniche di storage lato client** per migliorare le performance e ridurre i tempi di caricamento
-

STRUTTURA DEL PROGETTO

Il progetto è stato organizzato in 6 file principali:

- **home.html** → pagina introduttiva con navigazione alle sezioni principali del sito
- **style.css** → stile personalizzato per completare e migliorare lo stile di Bootstrap

- **pokemon.html** → pagina per visualizzare i Pokémon disponibili e catturarli
 - **pokemon.js** → script associato a 'pokemon.html' e usato per gestire la logica della cattura
 - **pokédex.html** → pagina per visualizzare e gestire il Pokédex personale
 - **pokédex.js** → script associato a 'pokedex.html' e usato per gestire il Pokédex (visualizzazione e rilascio Pokémon)
-

COMPONENTI PRINCIPALI

1. Home Page

- Percorso: home.html
- Contenuto:
 - i. Introduzione al progetto
 - ii. Barra di navigazione e collegamenti rapidi:
 - 1. All Pokémon: lista dei Pokémon disponibili
 - 2. My Pokédex: gestione del Pokédex personale

2. Pokémon List

- Percorso: pokemon.html
- Funzionalità:
 - i. Visualizzazione dinamica della lista Pokémon tramite l'API di Pokémon
 - ii. Ogni Pokémon viene mostrato in una card con nome, immagine e bottone 'Catch' per catturarlo
- Notifiche utente per confermare la cattura o segnalare errori (es. limite raggiunto)

3. Pokédex Personale

- Percorso: pokedex.html
 - Funzionalità:
 - i. Visualizzazione dei Pokémon catturati e salvati nel localStorage
 - ii. Rimozione dei Pokémon tramite il bottone 'Release'
 - iii. Notifica utente per confermare il rilascio
-

TECNOLOGIE UTILIZZATE

In questo progetto si sono utilizzate diverse 'tecnologie':

- **HTML5** → per gestire la struttura delle pagine
 - **CSS3** → stile personalizzato e integrazione con Google Fonts
 - **Bootstrap 5** → componenti responsive come navbar, modal e layout grid
 - **JavaScript** → per gestire la logica e la manipolazione del DOM
 - **localStorage** → persistenza dei dati lato client
-

FUNZIONALITÀ PRINCIPALI

- **Gestione della cattura** → la funzione di cattura permette di aggiungere un Pokémon al Pokédex personale:

```
/**
 * Aggiunge un Pokémon al Pokédex
 * @param {string} name - Nome del Pokémon
 * @param {string} image - URL dell'immagine del Pokémon
 */
function catchPokemon(name, image) {
  notifyMessage.classList.forEach(Class => notifyMessage.classList.remove(Class)) //Reset della classe e immagine del messaggio
  notifyMessageImage.src = ""

  //Controllo se il Pokémon è già stato catturato
  if (!myPokemon.find(p => p.name === name)) {
    if (myPokemon.length >= 12) {
      showNotification("Uh oh!", "Hai già 12 Pokémon nel tuo Pokédex!", 'text-warning')
      return
    }

    //Aggiungo il Pokémon al Pokédex
    myPokemon.push({ name, image })
    localStorage.setItem('myPokemon', JSON.stringify(myPokemon))

    //Notifica di successo
    showNotification("Success!", `${name} è stato catturato e aggiunto al tuo Pokédex!`, 'text-success', image)
  } else {
    //Notifica di Pokémon già presente
    showNotification("Uh oh!", `${name} è già presente nel tuo Pokédex.`, 'text-warning', image)
  }
}
```

- **Gestione del Pokédex** → consente di visualizzare e rimuovere i Pokémon catturati:

```
//Funzione per rilasciare un Pokémon dal Pokédex
function releasePokemon(name, image) {
  let myPokemon = JSON.parse(localStorage.getItem('myPokemon')) || []

  //Filtro l'elenco dei Pokémon per escludere quello rilasciato
  myPokemon = myPokemon.filter(pokemon => pokemon.name !== name)

  //Aggiorno il localStorage
  localStorage.setItem('myPokemon', JSON.stringify(myPokemon))

  //Notifica per indicare che il Pokémon è stato rilasciato
  notifyMessage.classList.forEach(Class => notifyMessage.classList.remove(Class))

  notifyMessageModalLabel.textContent = "Pokémon Released"
  notifyMessage.classList.add('text-warning')
  notifyMessage.textContent = `${name} è stato rilasciato e non è più nel tuo Pokédex.`
  notifyMessageImage.src = image

  const myModal = new bootstrap.Modal(document.getElementById('notifyMessageModal'))
  myModal.show()

  //Ricarico il Pokédex
  loadPokedex()
}
```

- **Notifiche Modal** → un sistema di modali consente di notificare messaggi agli utenti

```
<!--Modal - Visualizzare i messaggi d'errore / informazioni-->
<div class="modal fade" id="notifyMessageModal" tabindex="-1" aria-labelledby="notifyMessageModalLabel" aria-hidden="true">
  <div class="modal-dialog modal-dialog-centered modal-lg">
    <div class="modal-content">
      <!--Modal Header-->
      <div class="modal-header">
        <h5 class="modal-title" id="notifyMessageModalLabel"></h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <!--Modal Body-->
      <div class="modal-body text-center">
        <img src="" alt="" id="notifyMessageImage">
        <p id="notifyMessage"></p>
      </div>
      <!--Modal Footer (il basso del Modal dove ci sta il bottone 'close')-->
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
      </div>
    </div>
  </div>
</div>
```

PERSISTENZA DEI DATI

I Pokémon catturati vengono salvati nel localStorage in formato **JSON**. Qui sotto è riportato un esempio della struttura dell'array salvato:

```
[
  {
    "name": "pikachu",
    "image": "url_immagine"
  },
]
```

INSTALLAZIONE ED ESECUZIONE

Requisiti minimi:

- Browser moderno con supporto a localStorage e Bootstrap 5

Istruzioni:

- Scaricare il file compresso del progetto ('Pokémon.zip')
- Estrarre tutto dallo zip
- Aprire 'home.html' in un browser per avviare il sito
- Naviga tra le sezioni utilizzando la barra di navigazione

REQUISITI DI SISTEMA

- **Browser supportati:** Google Chrome, Firefox, Edge
- **Librerie richieste:**
 - Bootstrap 5.3.X (già incluso)
 - Google Fonts (integrato nel CSS)

Autore: *Muhammad Shayan, Wu Yu Dong*
Ultimo aggiornamento: *22 Novembre 2024*