

RELAZIONE DI LAVORO - PROGETTO WEB APP CON CHIAMATE API RESTful

1. NOME DELLA WEB APP

AnimeMangaExplorer

2. INTRODUZIONE - DESCRIZIONE SINTETICA DELLA WEB APP

L'applicazione web "AnimeMangaExplorer" permette agli utenti di scoprire un'ampia varietà di anime e manga attraverso chiamate API.

Gli utenti possono cercare anime o manga per titolo o autore e visualizzare informazioni dettagliate, come la trama e le valutazioni, per ciascun anime o manga.

Questa app è pensata soprattutto per gli appassionati di anime e manga che vogliono ottenere informazioni in modo semplice e veloce.

3. STRUMENTI UTILIZZATI

Linguaggi di sviluppo:

- HTML (per la struttura delle pagine web)
- CSS (per la gestione dell'aspetto visivo)
- JavaScript (per la gestione delle chiamate API)

Framework:

- [Bootstrap](#) (per alcuni componenti grafici)

Web Service utilizzati:

- [Jikan API](#) (API open source che fornisce informazioni su anime e manga da [MyAnimeList](#)).

4. DESCRIZIONE DELLE API UTILIZZATE

La Web App utilizza Jikan API, un'API che permette di effettuare chiamate per ottenere informazioni su anime e manga.

Alcuni esempi di chiamate:

Ricerca per titolo di anime:

- Metodo: GET
- URL: <https://api.jikan.moe/v4/anime?q={titolo}>
- Esempio: <https://api.jikan.moe/v4/anime?q=Naruto>

Ricerca per autore:

- Metodo: GET
- URL: <https://api.jikan.moe/v4/manga?q={autore}>
- Esempio: <https://api.jikan.moe/v4/manga?q=Eiichiro Oda>

5. DESCRIZIONE DELLA STRUTTURA JSON DELLE RISPOSTE

Le risposte JSON dalle chiamate API contengono informazioni strutturate in oggetti e array. Ad esempio, per la ricerca di un anime:

```
1  {
2    "data": [
3      {
4        "title": "Naruto",
5        "synopsis": "Naruto è un giovane ninja che sogna di diventare...",
6        "episodes": 220,
7        "score": 8.2,
8        "image_url": "https://url_immagine.jpg"
9      }
10   ]
11 }
```

I dati principali includono:

- title: Il titolo dell'anime
- synopsis: La sinossi dell'anime
- episodes: Il numero di episodi
- score: La valutazione media
- image_url: L'URL dell'immagine dell'anime

6. OBIETTIVI RAGGIUNTI

OBIETTIVO RICHIESTO	INDICARE UNA DELLE SEGUENTI VOCI: RAGGIUNTO, IN PARTE RAGGIUNTO, NON RAGGIUNTO/NON REALIZZATO
Comprensione della documentazione del Web Service	Raggiunto
Chiamata API tramite fetch JS e stampa sulla console della risposta	Raggiunto
Utilizzo in JS dei dati prelevati dalla risposta JSON di una chiamata API	Raggiunto
Realizzazione della pagina web con almeno una chiamata API e relativa visualizzazione a video del risultato.	Raggiunto
Realizzazione della pagina web con almeno una chiamata API e relativa visualizzazione a video del risultato curando anche l'estetica della visualizzazione.	Raggiunto
Realizzazione della pagina web con più di una chiamata API e relativa visualizzazione a video del risultato.	In parte raggiunto
Realizzazione della pagina web con più di una chiamata API e relativa visualizzazione a video del risultato curando anche l'estetica della visualizzazione.	Raggiunto
Realizzazione delle tre pagine richieste.	Raggiunto

7. DESCRIZIONE DI UNA PORZIONE DI CODICE DELLA WEB APP

```
1 // Funzione per ottenere i dati da un'API e visualizzarli
2 async function fetchAnimeData(title){
3     try{
4         const response = await fetch(`https://api.jikan.moe/v4/anime?q=${title}`);
5         const data = await response.json();
6         displayAnimeData(data);
7     }
8     catch(error){
9         console.error('Error fetching data: ', error);
10    }
11 }
12
13 // Funzione per visualizzare i dati dell'anime
14 function displayAnimeData(data) {
15     const anime = data.data[0];
16     document.getElementById('anime-title').innerText = anime.title;
17     document.getElementById('anime-synopsis').innerText = anime.synopsis;
18     document.getElementById('anime-episodes').innerText = `Episodes: ${anime.episodes}`;
19     document.getElementById('anime-score').innerText = `Score: ${anime.score}`;
20     document.getElementById('anime-image').src = anime.image_url;
21 }
```

La funzione `fetchAnimeData` esegue una chiamata API per cercare un anime per titolo. I dati vengono poi estratti e inviati alla funzione `displayAnimeData` che aggiorna il DOM per visualizzare le informazioni sull'anime.

8. TROUBLESHOOTING DI PROBLEMI RISCONTRATI

Durante lo sviluppo, mi sono imbattuto in un problema con la gestione delle risposte errate dell'API, che restituivano errori 404 quando il titolo non esisteva.

Per risolvere questa situazione, ho aggiunto un controllo per verificare se l'array dei risultati era vuoto e ho fatto in modo di mostrare un messaggio di errore all'utente.

9. RISULTATI E CONCLUSIONI

Ho portato a termine con successo la creazione della Web App, riuscendo a implementare tutte le funzionalità richieste, come la ricerca per titolo e autore e la visualizzazione dei dettagli dell'anime.

Questa esperienza mi ha dato l'opportunità di approfondire la mia comprensione delle API RESTful, delle chiamate asincrone in JavaScript e dell'interazione con il DOM per aggiornare dinamicamente i contenuti.

10. ALLEGATI

1. *Allegato 1: Struttura JSON delle richieste.*
2. *Allegato 2: Porzione di codice della web API.*