

# Album delle Figurine dei Super Eroi (AFSE)

- **Anno Accademico:** 2023/2024
- **Corso:** Programmazione Web e Mobile
- **Autore:** Lorenzo Signorelli
- **Matricola:** 10135A

## Indice

1. [Introduzione](#)
2. [Funzionalità dell'Applicazione](#)
  - [Registrazione e Login](#)
  - [Acquisto di Crediti](#)
  - [Acquisto Pacchetti di Figurine](#)
  - [Scambio di Figurine](#)
3. [Architettura e Tecnologie Utilizzate](#)
4. [Organizzazione del Codice](#)
  - [Frontend](#)
  - [Backend](#)
  - [Database](#)
5. [Scelte Implementative](#)
  - [API Marvel e Raccolta Dati](#)
  - [Gestione dei Pacchetti di Figurine](#)
  - [Scambi Complessi](#)
6. [Lavori Futuri](#)

## Introduzione

L'Album delle Figurine dei Super Eroi (AFSE) è una piattaforma web che consente agli utenti di collezionare figurine virtuali di supereroi e di scambiarle con altri utenti. Gli utenti possono acquistare pacchetti di figurine, visualizzare le informazioni su ogni supereroe e proporre scambi. Il progetto si basa su un'integrazione con l'API Marvel per raccogliere i dati relativi ai supereroi.

## Funzionalità dell'Applicazione

### Registrazione e Login

- Gli utenti possono registrarsi con un nome utente, email, password e poi successivamente selezionare il loro supereroe preferito.
- È possibile effettuare il login con le credenziali registrate.
- Gestione della sessione utente tramite JWT, i cookie di sessione vengono salvati in modo sicuro in js-cookies, le informazioni dell'utente invece in localStorage.
- alla cancellazione dell'utente vengono cancellate tutte le sue copie e i suoi trade non ancora accettati per quelli accettati invece viene messo a null l'ownerId e vengono mantenuti

### Acquisto di Crediti

- Gli utenti possono acquistare crediti virtuali da utilizzare per acquistare pacchetti di figurine, l'acquisto è simulato.
- Ogni pacchetto di figurine costa un credito e contiene 5 figurine casuali.

### Acquisto Pacchetti di Figurine

- Gli utenti possono acquistare pacchetti di figurine con i crediti disponibili.
- Ogni pacchetto contiene figurine casuali di supereroi prelevati dall'api in backend.

### Scambio di Figurine

- Gli utenti possono proporre scambi di figurine con altri utenti.
- È possibile scambiare figurine singole o effettuare scambi complessi (es. due figurine per una).
- Allo scambio le copie delle figurine vengono messe senza proprietario e vengono restituite in caso il trade venga cancellato
- Agli scambi possono essere allegati i crediti

## Architettura e Tecnologie Utilizzate

- **Frontend:** React + Material-UI (MUI) per l'interfaccia utente interattiva e responsive.
- **Backend:** Node.js con Express per la gestione delle API e della logica applicativa.
- **Database:** MongoDB per la gestione degli utenti, delle figurine, dei pacchetti e degli scambi.
- **API:** Integrazione con le API Marvel per ottenere i dati dei supereroi.
- **Swagger:** Documentazione API tramite Swagger per facilitare il testing e la validazione dei servizi disponibile a `/api-docs/`.

## Organizzazione del Codice

### Frontend

- **React:** Il frontend è sviluppato con React per la gestione dello stato e l'interfaccia utente.
- **Material-UI (MUI):** Utilizzato per creare componenti UI moderni e responsivi.
- **Routing:** Il progetto utilizza React Router per la navigazione tra le pagine (login, acquisto, scambio, ecc.).

### Backend

- **Node.js & Express:** Il backend espone API REST per la gestione degli utenti, dei pacchetti, delle figurine e degli scambi.
- **Autenticazione:** Gestita tramite JSON Web Token (JWT) per proteggere le rotte e autenticare le richieste.
- **Swagger:** Documentazione API creata con Swagger per facilitare la validazione delle chiamate backend.

### Database

- **MongoDB:** Il database contiene 4 collezioni principali:
  - **Users:** Dati dell'utente (username, email, password, supereroe preferito).
  - **Figurine:** Informazioni su ciascuna figurina (nome, descrizione, immagine).
  - **Pacchetti:** Storico dei pacchetti acquistati dagli utenti.
  - **Scambi:** Proposte di scambio tra utenti.

## Scelte Implementative

### API Marvel e Raccolta Dati

I dati relativi ai supereroi sono stati prelevati tramite script di integrazione con le API Marvel. Lo script raccoglie e memorizza le informazioni essenziali sui supereroi nel database, comprese le serie, gli eventi e i fumetti in cui compaiono.

```
import requests
import hashlib
import pymongo
from datetime import datetime
import time
import logging

# Setup logging
logging.basicConfig(level=logging.INFO)

# Define the Marvel API keys
public_key = '953981c51893300fca926fc6bc973764'
private_key = '9b1659c15c2d58941aa575a6f59e9eba2acb482f'

# Define the MongoDB connection
mongo_client = pymongo.MongoClient('localhost', 27017)
db = mongo_client['MarvelDb']
collection = db['cards']

# Generate the MD5 hash
def generate_hash(ts, private_key, public_key):
    to_hash = ts + private_key + public_key
    return hashlib.md5(to_hash.encode()).hexdigest()

# Fetch data from the Marvel API
def fetch_marvel_characters(offset, limit):
    ts = str(datetime.now().timestamp())
    hash_value = generate_hash(ts, private_key, public_key)
    url = f"http://gateway.marvel.com/v1/public/characters?offset={offset}&limit={limit}&ts={ts}&apikey={hash_value}"

    response = requests.get(url) # Disable SSL verification
    if response.status_code == 200:
        return response.json()['data']['results']
    else:
        response.raise_for_status()

# Transform and load data into MongoDB
def load_data_to_mongo(data):
    for character in data:
        document = {
            "Name": character.get("name"),
            "Desc": character.get("description", False) if character.get("description") else False,
            "Img": character['thumbnail']['path'] + '.' + character['thumbnail']['extension'],
            "Details": {
                "series": {
                    "names": [item["name"] for item in character['series']['items']]
                },
                "events": {
                    "names": [item["name"] for item in character['events']['items']]
                },
                "comics": {
                    "names": [item["name"] for item in character['comics']['items']]
                }
            }
        }
        collection.insert_one(document)

# Main function to fetch and load data
def main():
    total_entries = 200
    batch_size = 10
    offset = 0
    max_retries = 5
    retry_delay = 2 # Initial retry delay in seconds

    while offset < total_entries:
        retries = 0
        while retries < max_retries:
            try:
                marvel_characters = fetch_marvel_characters(offset, batch_size)
                load_data_to_mongo(marvel_characters)
                offset += batch_size
                retry_delay = 2 # Reset retry delay after successful request
                break
            except Exception as e:
                retries += 1
                logging.error(f"Attempt {retries}: An error occurred: {e}. Retrying in {retry_delay} seconds")
                time.sleep(retry_delay)
                retry_delay *= 2 # Exponential backoff
                if retries == max_retries:
                    logging.error("Max retries exceeded. Skipping this batch.")
                    offset += batch_size # Skip this batch and move to the next

        time.sleep(1) # To avoid hitting the API rate limit

    logging.info("Data successfully inserted into MongoDB")

if __name__ == "__main__":
    main()
```

### Gestione dei Pacchetti di Figurine

Ogni utente può acquistare pacchetti di figurine utilizzando i crediti virtuali. Ogni pacchetto contiene 5 figurine casuali. Il backend effettua una chiamata per selezionare casualmente i supereroi da includere nel pacchetto, ne crea delle copie e le copie vengono associate all'utente.

### Scambi Complessi

Gli utenti possono proporre scambi di figurine semplici (1 figurina per 1) o scambi complessi (ad esempio, due figurine per una), inoltre si possono inserire dei costi in valuta virtuale. Il backend gestisce la validazione degli scambi e assicura che l'utente non possa ricevere figurine duplicate, quando uno scambio viene effettuato esso viene segnato come accettato e non viene eliminato, tutti gli scambi accettati possono essere visti se si seleziona show accepted.

### Prove di Funzionamento

[VIDEO](#)

## Lavori Futuri

- Miglioramento delle notifiche in tempo reale per gli scambi di figurine.
- Implementazione dei modal per una migliore interfaccia
- Implementazione di offerte speciali per l'acquisto di crediti.