

# Relazione del Progetto di Programmazione Web e Mobile

Francesco Franguelli Emanuele

May 13, 2025

## Introduzione

Il progetto "Album delle Figurine di Harry Potter (AFHP)" è un'applicazione web per la gestione di un album elettronico di figurine di super eroi. Il progetto prevede la gestione del profilo utente, l'acquisto (simulato) di crediti e pacchetti di figurine, e la gestione dello scambio di figurine tra giocatori.

## 1 Configurazione e avvio dell'applicazione

Per eseguire l'app bisogna seguire i seguenti passaggi:

- scaricare il codice sorgente
- aver installato node js e npm
- installare le dipendenze con il comando 'npm install'
- configurare correttamente il file .env (il file .env è già presente nel codice sorgente ed è già impostato, ma se da problemi occorre riconfiguralo adeguatamente)
- avviare il progetto con il comando 'npm run start'
- a questo punto si può accedere al sito web al link "<https://127.0.0.1:3000>" (attenzione questo è il link di default se è stato modificato il file .env il link potrebbe differire)
- una volta effettuato l'accesso il browser potrebbe segnalare il sito come non sicuro in quanto i certificati per l'https sono stati autofirmati, basterà chiudere il pop up e si potrà accedere al sito, in alternativa si possono cambiare i certificati con dei certificati validi.

## 2 Struttura

Il progetto è suddiviso in 3 cartelle principali:

- **FrontEnd:** questa cartella contiene tutto il codice del sito web che viene eseguito lato client. Al suo interno troviamo 3 cartelle: `html`, `css`, `javascript`, che contengono i vari file del sito web.
- **BackEnd:** in questa cartella è presente tutto il codice del backend, ovvero le API. Al suo interno troviamo il file `routes.js`, in cui sono scritti i vari endpoint delle API, ciascuno associato al controller, cioè la funzione che gestisce la richiesta. La cartella `middleware` contiene il file `auth.js`. I middleware sono funzioni che gestiscono alcune richieste, ma non forniscono la risposta. Piuttosto, effettuano una prima parte di elaborazione. In questo caso, `auth.js` si occupa di verificare se la richiesta proviene da un utente autenticato e, successivamente, passa la richiesta al controller di competenza. Infine, c'è la cartella `controllers`, che contiene i file dei vari controller, suddivisi in `user` (controller che gestiscono le operazioni sugli utenti), `trade` (controller che gestiscono le operazioni di scambio) e `marvel` (controller che utilizzano le API esterne di Marvel).
- **Database:** questa cartella contiene il file `database.js`, che si occupa della connessione al database, e la cartella `schemas`, che contiene gli schemi del progetto.

Oltre a queste 3 cartelle, ci sono altri file e cartelle importanti:

- Il file `server.js`, che viene eseguito all'avvio del progetto e si occupa di connettersi al database e successivamente di avviare il server.
- La cartella `docs`, che contiene la documentazione del progetto.
- La cartella `certificate`, che contiene i certificati.
- Altri file, come la cartella `node_modules`, che contiene le dipendenze del progetto, e file di configurazione come `.gitignore`.

## 3 Progettazione

Il progetto è stato sviluppato come un server HTTPS che ospita un sito web dedicato alle figurine, con le relative API disponibili alla root `/api/....`. La connessione al database avviene prima dell'avvio del server. Il database scelto è MongoDB, un database NoSQL, e la sua gestione avviene nel cloud tramite MongoDB Atlas.

Per ottenere informazioni sulle figurine, sono utilizzate le API pubbliche. L'autenticazione degli utenti avviene tramite cookie, che vengono inviati con le richieste quando necessario. In particolare, viene utilizzato un token JWT (JSON Web Token) per gestire l'autenticazione degli utenti.

# Implementazione

## Sito Web

La parte relativa al sito web è stata sviluppata utilizzando il framework *Bootstrap*, che fornisce una griglia flessibile e componenti predefiniti per la creazione di interfacce utente reattive. Questo framework consente di adattare automaticamente l'interfaccia a diverse risoluzioni e dispositivi, garantendo un'esperienza utente ottimale su desktop, tablet e dispositivi mobili.

La struttura del sito è stata progettata per essere semplice, chiara e intuitiva, con una navigazione fluida e facilmente accessibile. Per la visualizzazione delle figurine e delle informazioni correlate, sono stati utilizzati vari componenti di Bootstrap, come modali, pulsanti e tavole, che semplificano l'interazione con il contenuto.

Il sito è ospitato su un server principale che utilizza Express per gestire le richieste HTTP e servire i contenuti dinamici, consentendo un'interazione rapida ed efficiente con l'utente.

## Interazione con le API

Il sito web interagisce con il backend tramite API RESTful, che permettono di recuperare i dati relativi alle figurine e di gestire le operazioni di autenticazione. Le API sono state sviluppate utilizzando Express e il modulo *swagger-ui-express*, che consente di visualizzare facilmente la documentazione delle API in modo chiaro e intuitivo per gli sviluppatori.

Il backend accede al database tramite *Mongoose*, una libreria che facilita l'interazione con MongoDB utilizzando schemi e modelli. Questo approccio semplifica lo sviluppo e consente una gestione più organizzata e strutturata dei dati, garantendo al contempo un'integrazione fluida tra il database e le API.

## Autenticazione e Sicurezza

L'autenticazione degli utenti è gestita tramite token JWT (JSON Web Token), che vengono inviati con ogni richiesta al backend per garantire l'accesso alle risorse protette. I cookie contenenti i token JWT vengono generati dal backend e utilizzati per validare l'identità dell'utente in modo sicuro, mantenendo la sessione attiva durante la navigazione.

Per proteggere il sito web da vulnerabilità comuni, come Cross-Site Scripting (XSS) e Cross-Site Request Forgery (CSRF), sono state adottate le necessarie misure di sicurezza. Tra queste, la gestione sicura dei cookie e l'uso di HTTPS per crittografare le comunicazioni tra client e server, garantendo così la protezione dei dati sensibili e la privacy degli utenti.

## 4 Funzionalità Implementate (tra quelle a scelta)

Ho deciso di implementare le seguenti funzionalità:

- **Estensione del baratto:** è stata introdotta la possibilità di effettuare scambi più complessi, come ad esempio scambiare due figurine per una, ampliando così le opzioni di scambio disponibili.
- **Vendita di figurine:** gli utenti possono ora vendere le figurine in loro possesso in cambio di crediti, che possono essere utilizzati per acquisire altre figurine o per altre funzionalità del sistema.
- **Controlli di integrità:** sono stati implementati dei controlli per evitare che gli utenti ricevano figurine già in loro possesso o propongano scambi che includano due o più figurine uguali, garantendo così una gestione più equa e logica degli scambi.

## Test e Validazione

Di seguito alcuni screenshot del sito web in esecuzione. Le immagini mostrano le varie schermate dell'applicazione, inclusa la home page, la pagina di login, la registrazione, la visualizzazione dei crediti, l'acquisto di pacchetti di figurine, la visualizzazione dell'album e la gestione degli scambi.

