

## INTRODUZIONE

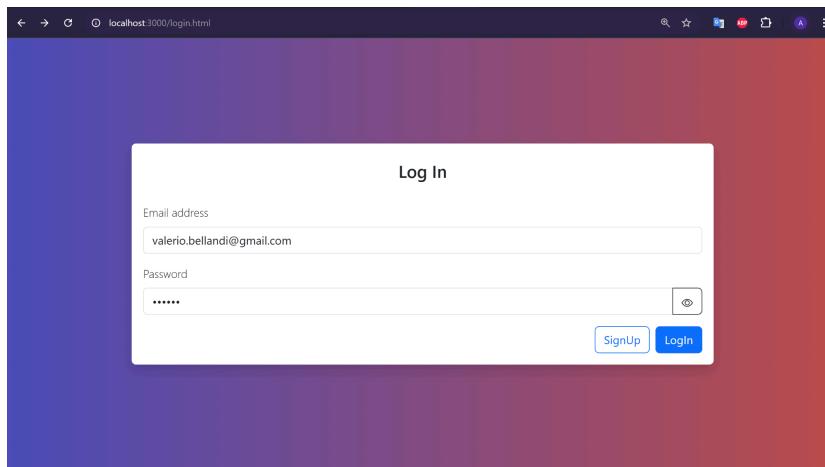
Per il back-end è stato utilizzato **Express** su **Node.js**, per gestire le operazioni lato server, con **MongoDB** come dbms per la gestione e memorizzazione dei dati.

Per il front-end ho utilizzato **HTML5**, **CSS3** e **Javascript**, utilizzando **Bootstrap** per creare un'interfaccia utente funzionale.

## ANALISI DEI REQUISITI

### Registrazione e login

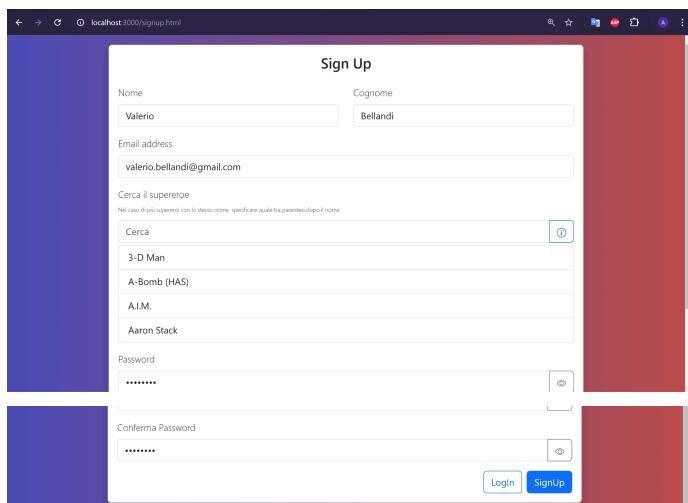
Il login permette l'accesso al portale utilizzando *mail* e *password*



The screenshot shows a 'Log In' form on a website. The form is titled 'Log In' and contains two input fields: 'Email address' with the value 'valerio.bellandi@gmail.com' and 'Password' with a redacted value. Below the password field is an 'Eye' icon for password visibility. At the bottom of the form are two buttons: 'SignUp' and 'Login'.

Se l'utente non ha ancora effettuato la registrazione, può procedere fornendo queste informazioni:

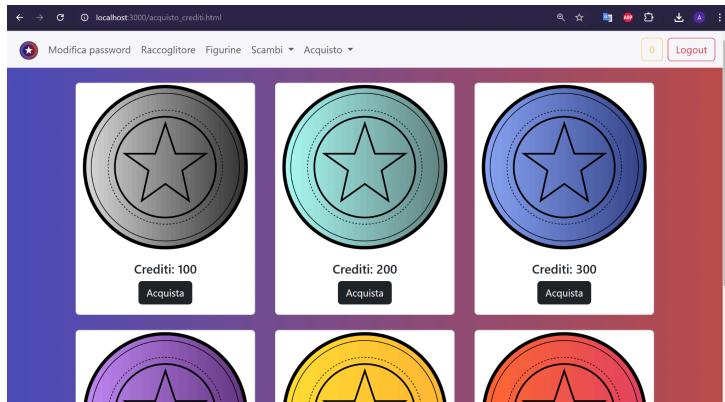
- Nome
- Cognome
- Email
- Supereroe preferito
- Password



The screenshot shows a 'Sign Up' form on a website. The form has fields for 'Nome' (Valerio), 'Cognome' (Bellandi), 'Email address' (valerio.bellandi@gmail.com), and 'Password' (redacted). Below the email field is a search bar labeled 'Cerca il supereroe' with placeholder text 'Nel caso di più supereroi con lo stesso nome, specificare quale tra parentesi dopo il nome'. A dropdown menu shows suggestions: '3-D Man', 'A-Bomb (HAS)', 'A.I.M.', and 'Aaron Stack'. At the bottom of the form are two buttons: 'Login' and 'SignUp'.

## Acquisto dei crediti

Per simulare l'acquisto dei crediti bisogna recarsi nella sezione acquisto(quella dedicata ai crediti):

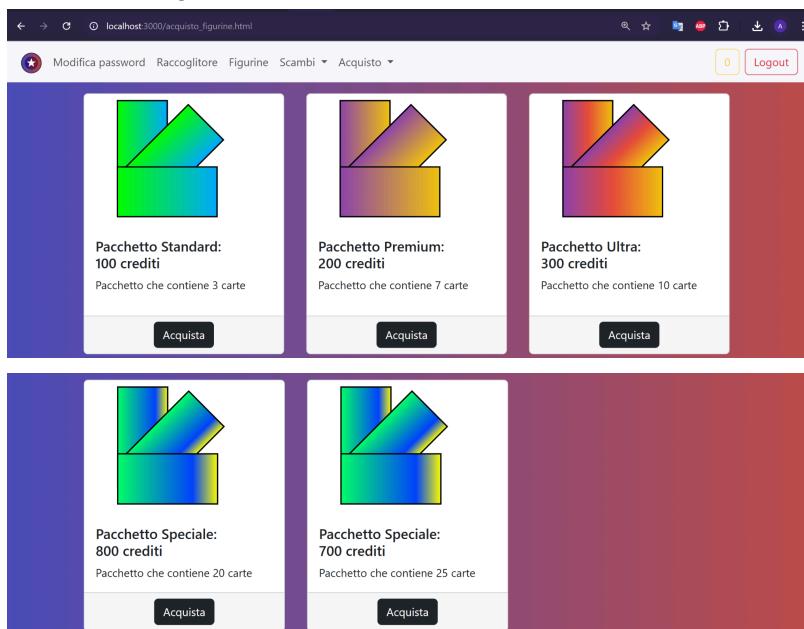


I tagli disponibili sono:

- 100 crediti
- 200 crediti
- 300 crediti
- 400 crediti
- 500 crediti
- 600 crediti

## Acquisto dei pacchetti di figurine

Per simulare l'acquisto dei crediti bisogna recarsi nella sezione acquisto(quella dedicata ai pacchetti di figurine):



Ci sono due scenari possibili: la prima riga riguarda i pacchetti default, la seconda(in questo caso essendoci non più di tre pacchetti *speciali*) riguarda i pacchetti speciali, ovvero le offerte gestite da un utente amministratore.

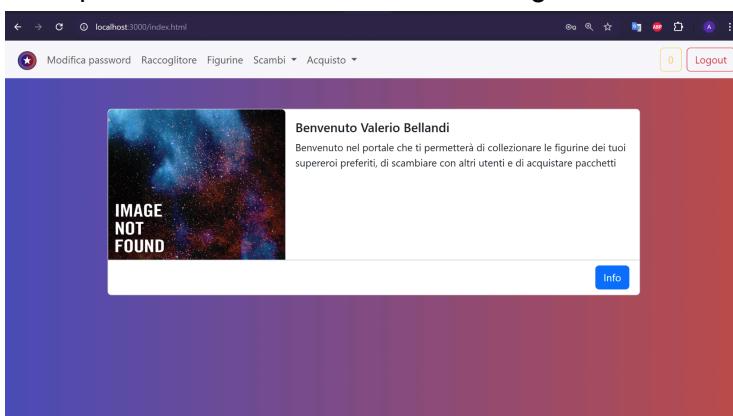
Per gestire l'**estrazione casuale** di figurine nel back-end è stato fatto:

- creazione di una funzione che genera tutte le figurine possibili e le salva in un file **tutti\_eroi.json**;
- questa funzione viene usata *temporizzata*, ossia viene eseguita ogni 12 ore
- per l'estrazione effettiva delle carte si prende l'array completo delle carte, si usa la funzione shuffle(che mischia l'ordine delle carte dell'array) e poi la funzione slice(che prende un sottoinsieme delle carte in precedenza mischiate)

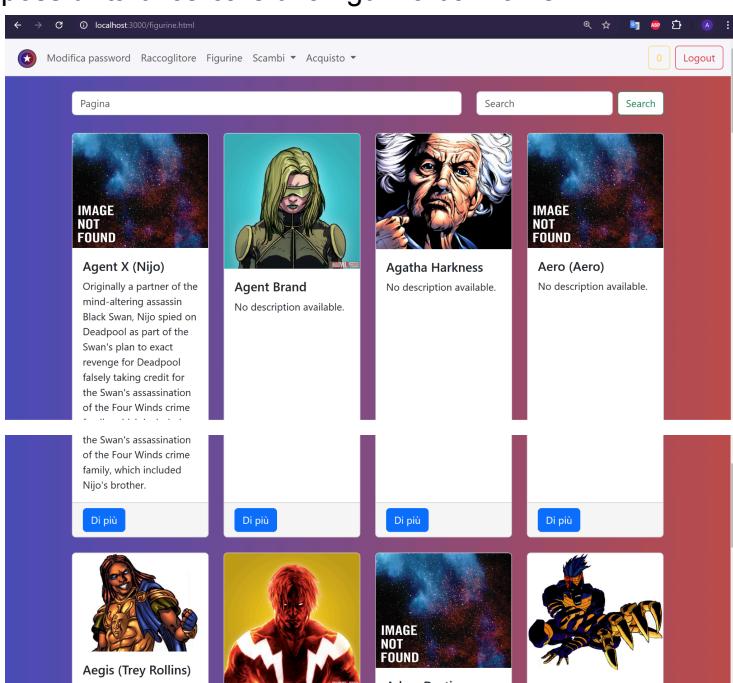
### Visualizzazione delle informazioni dei supereroi

Ci sono due/tre pagine dedicate alla visualizzazione dei supereroi e dalle quali posso risalire alle loro informazioni.

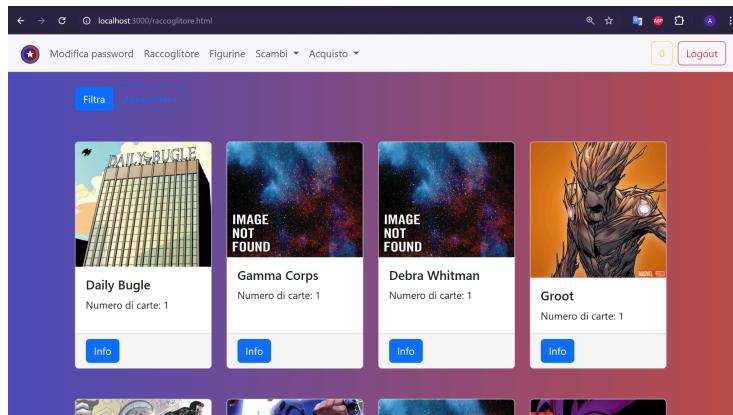
- 1) home page, schiacciando il bottone **Info** vado alla pagina dedicata alle informazioni del supereroe scelto al momento della registrazione



- 2) figurine, la pagina dedicata alla visualizzazione di tutte le possibili figurine, schiacciando il bottone **Di più** posso andare alla pagina che mi fa vedere le informazioni principali delle figurine; in questa pagina si vedono 16 figurine per volta, posso scegliere che pagina vedere(sostanzialmente da un offset per la generazione delle figurine), in più da la possibilità di cercare una figurina dal nome:



- 3) raccoglitore, la pagina dedicata alla visualizzazione delle figurine nel raccoglitore dell'utente, schiacciando il bottone **Info** posso andare alla pagina che mi fa vedere le informazioni principali delle figurine;  
 i bottoni **Filtra** e **Azzera filtro** rispettivamente fanno vedere le figurine scambiabili e tutte le figurine:



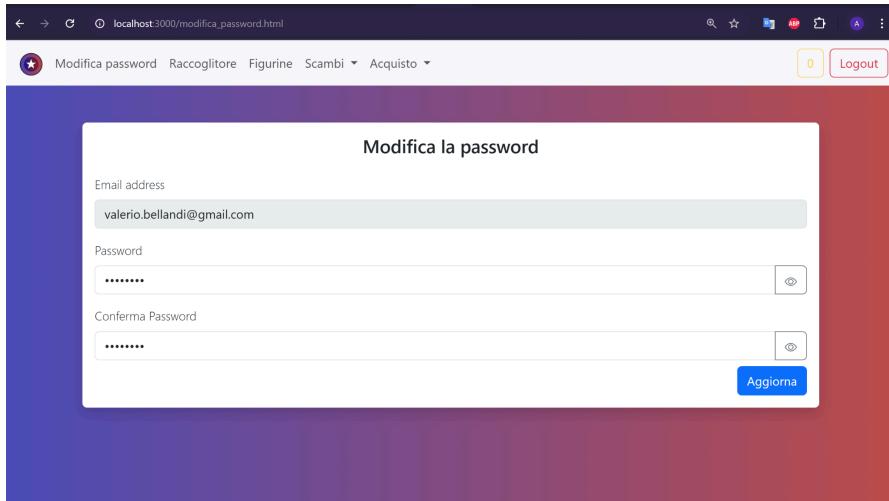
### La pagina delle informazioni delle figurine

The screenshots illustrate the detailed information page for a superhero figure. The top section displays a large portrait of Agent Brand and basic details like 'Agent Brand' and 'No description available'. Below this, there are sections for comic books ('FUMETTI') and TV shows ('SERIE'). The comic book section shows covers for issues #21, #24, #23, #22, and #32 of 'Astonishing X-Men' (2004). The TV show section shows covers for 'Men' (2004) episodes #40 and #38, and 'Dead X-Men' (2024 - Present) with the tagline 'DEATH WAS ONLY THE BEGINNING!'. At the bottom, it says 'Non sono presenti eventi'.

Arrivati alla pagina delle informazioni, quello che vediamo è la prima figura, schiacciando poi sul bottone compaiono tutte le informazioni principali dei supereroi.

## Modifica della password

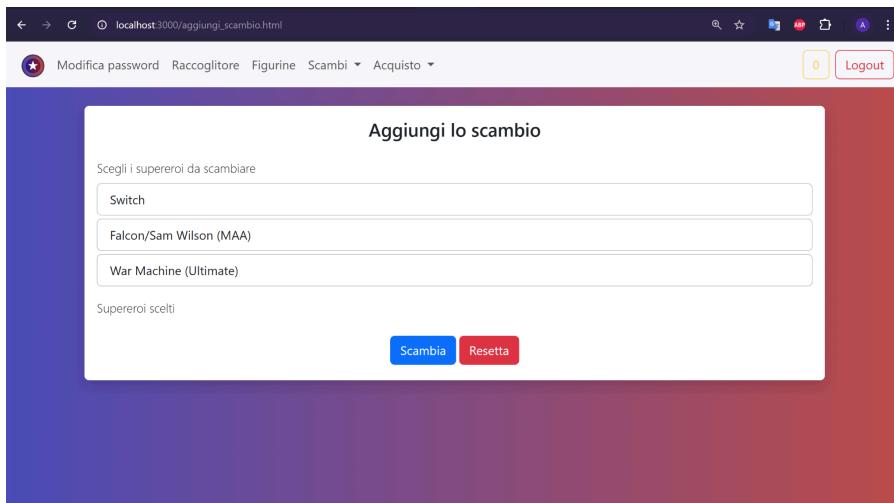
È presente una pagina dedicata alla modifica della password



The screenshot shows a web browser window with the URL `localhost:3000/modifica_password.html`. The page title is "Modifica la password". It contains three input fields: "Email address" with the value "valerio.bellandi@gmail.com", "Password" with the value "\*\*\*\*\*", and "Conferma Password" with the value "\*\*\*\*\*". There are "Aggiorna" and "Logout" buttons at the bottom right.

## Nella sezione scambi sono presenti più pagine

### Inserimento degli scambi



The screenshot shows a web browser window with the URL `localhost:3000/aggiungi_scambio.html`. The page title is "Aggiungi lo scambio". It has a dropdown menu "Scegli i supereroi da scambiare" containing "Switch", "Falcon/Sam Wilson (MAA)", and "War Machine (Ultimate)". Below the dropdown is a button "Scambia".

l'utente una volta cliccato sul nome di un supereroe, lo sceglie per essere scambiato. Vengono visualizzati solo i supereroi che possono essere scelti dall'utente per essere scambiati(⇒ quelli per cui l'utente ha almeno due figurine).

## Pagina per proporre uno scambio

Se si clicca su **Proponi** si viene reindirizzati a una pagina dove si può fare la proposta di scambio: funziona allo stesso modo dell'aggiunta, ma non mi permette di inserire figurine che l'utente che ha proposto lo scambio ha già nel suo raccoglitore.

The screenshot shows a web browser window with the URL `localhost:3000/scambi_disponibili.html`. The page has a dark header bar with a user icon, a search bar, and navigation links: 'Modifica password', 'Raccoglitore', 'Figurine', 'Scambi ▾', and 'Acquisto ▾'. On the right side of the header are '0' notifications and a 'Logout' button. The main content area has a purple-to-red gradient background. It displays a table with four rows of swap proposals:

ID dello scambio	Carte proposte	Informazioni
677e547d8483b2d00c4b4e83	Mister Sinister (Ultimate), Zaladane	<button>Proponi</button>
677e53e28483b2d00c4b4e82	Colossus	<button>Proponi</button>
677e4f61c67d917038f95b5e	Zaladane	<button>Proponi</button>

## Visualizzazione degli scambi in sospeso

Mi permette di accettare uno scambio, tra quelli che ho proposto io e che hanno ricevuto delle offerte.

The screenshot shows a web browser window with the URL `localhost:3000/scambi_sospesi.html`. The header is identical to the previous screenshot. The main content area displays a table with one row of information:

Cosa hai offerto	Cosa ti propongono	Accetta
Pixie	Sabretooth Heroes For Hire	<button>Accetta</button>

## Utente amministratore

L'utente amministratore può inserire(o eliminare) delle offerte sull'acquisto di pacchetti di figurine

The screenshot shows a web application interface with two main sections:

- PACCHETTI SPECIALI**: A table displaying two rows of package information. The first row has "Crediti" (800) and "Numero di figurine" (20), with a red "Elimina" button. The second row has "Crediti" (700) and "Numero di figurine" (25), also with a red "Elimina" button.
- AGGIUNGI PACCHETTO SPECIALE**: A form for adding new packages. It has two input fields: "Crediti" (set to 100) and "Figurine" (set to 10). Below the inputs is a green "Aggiungi" button.

At the bottom of each section are buttons for "Aggiungi un pacchetto" and "Logout".

## BACKEND

Come specificato nell'introduzione, per lo sviluppo del server non sono stati utilizzati particolari framework se non quelli visti durante il corso.

Per la gestione dei vari endpoint ho optato per dividerli in diverse routes a seconda della funzionalità, per avere un codice più modulare, scalabile e mantenibile.

Tutti gli endpoints tramite swagger sono documentati e consultabili all'indirizzo [localhost:3000/api-docs](http://localhost:3000/api-docs).

## INSTALLAZIONE E PROVE DI FUNZIONAMENTO

Per l'installazione è necessario:

- entrare nella cartella **Backend**
- eseguire da terminale **npm install**(in modo tale che vengano installate tutte le dipendenze)
- creare un file **.env** per il setup delle variabili d'ambiente

```
PORT = 3000
MONGO_URI =
mongodb+srv://andrea2002:password@cluster0.daiqpbl.mongodb.net/?r
etryWrites=true&w=majority&appName=Cluster0
PUBLIC_API_KEY = 8525e9ec8bcb62a34e6ed8ff14a55936
PRIVATE_API_KEY = 8a458cd9cd99475730a9b98f5f15a216f88a9161
```

una cosa di questo genere

viene fornito nella cartella del codice sorgente

- eseguire su linea di comando (sempre dalla cartella Backend) **node server.js**
- nel browser basta poi, nella barra di ricerca, inserire **localhost:3000/** e da qui è possibile fare tutte le operazioni scritte in precedenza

Nel database sono presenti tre utenti:

- email: [user1.spiderman@afse.com](mailto:user1.spiderman@afse.com), password: password
- email: [user2.hulk@afse.com](mailto:user2.hulk@afse.com), password: password
- email [admin\\_afse@admin.com](mailto:admin_afse@admin.com), password:1234

Rispettivamente due utenti per *testare* le operazioni “base” che un utente può fare all’interno del sito web e un utente amministratore, che ha i privilegi di amministratore, inserisce le offerte per i pacchetti di figurine.