



Manganation

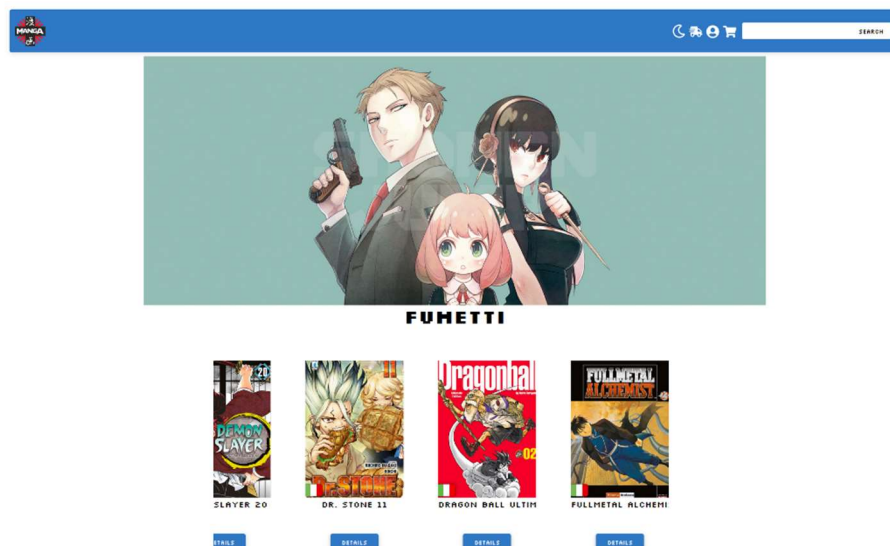
Mamone Maximiliano | 308214

Introduzione

Il fumetto giapponese, conosciuto appunto come [Manga](#), ha ormai da anni invaso la cultura occidentale. Persone di tutte le età leggono quotidianamente manga acquistati in fumetteria o al supermercato, e ancor di più ragazzi sfruttano internet per leggere edizioni ancora non pubblicate in Italia.

Con l'avvento del manga anche le pagine dedicate alla vendita e distribuzione del prodotto e di tutti gli associati come ad esempio le [Action figure](#), statuine di varie dimensioni e qualità rappresentanti personaggi presi dalle storie, e altri tipi di gadget sono aumentate esponenzialmente.

Per questo ho deciso di cimentarmi nella creazione di un e-commerce di manga e action figure, sia per fare esperienza con strumenti moderni come React nella variante TypeScript, ExpressJS e un classico database MySQL tramite XAMPP, sia perchè io stesso sono un appassionato di manga e ciò che li circonda.



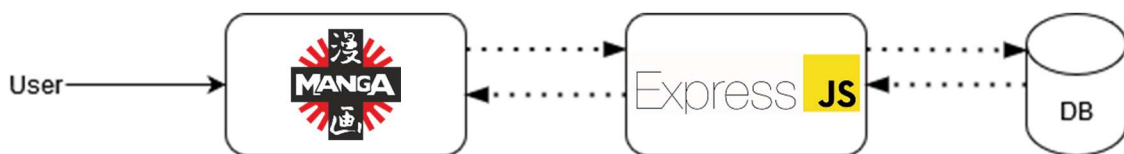
Specifica

Il progetto si pone come obiettivo quello di creare un e-commerce di fumetti e action figure con le seguenti caratteristiche:

- Frontend React in versione Typescript che garantisca un'interfaccia utente dinamica e responsiva
- API ExpressJS attraverso il quale vengono gestite tutte le richieste del client verso il database
- Database MySQL in grado di contenere tutte le informazioni necessarie al corretto funzionamento del sito

Più nello specifico il sito dovrà permettere all'utente di:

- Registrarsi e autenticarsi
- Cercare e visualizzare i prodotti
- Inserire nel carrello i prodotti e visualizzarli/comprarli in un secondo momento
- Aggiungere, rimuovere e selezionare carte di credito per l'acquisto dei prodotti
- Visualizzare lo stato dei suoi ordini



Funzionamento del sito

Registrazione e Login:

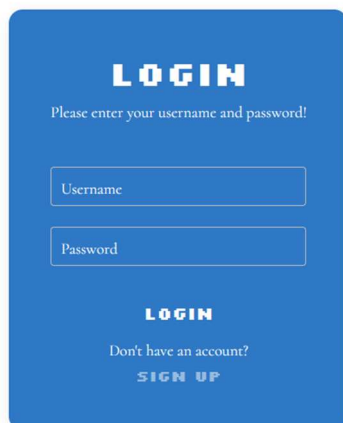
Entrambe queste funzionalità sono state implementate in modo semplice graficamente, durante la fase di registrazione all'utente vengono richiesti Nome, Cognome, Username, Indirizzo di residenza e un duplice inserimento della password.

Quest'ultima non verrà salvata in chiaro nel database ma bensì ne verrà conservato un digest prodotto dall'hashing della password, reso possibile dalla libreria [bcrypt](#)

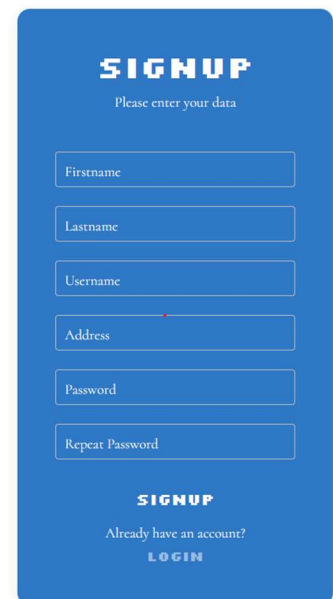
Durante la fase di login invece vengono semplicemente richiesti nome utente e password, e il server nel caso la

nostra identità venga verificata ci fornirà un token generato con una chiave segreta che ci

permetterà di accedere a tutte le funzionalità del nostro account per 24 ore senza doverci nuovamente autenticare manualmente. Il token viene generato tramite [jwt](#), ovvero JSON Web Tokens, che è per definizione: *"JSON Web Token è uno standard Internet proposto per la creazione di dati con firma opzionale e/o crittografia opzionale il cui payload contiene JSON che afferma un certo numero di attestazioni"*.



A blue rectangular form with rounded corners. At the top, the word 'LOGIN' is written in white, bold, uppercase letters. Below it, the text 'Please enter your username and password!' is in a smaller white font. There are two white input fields: the first is labeled 'Username' and the second is labeled 'Password'. At the bottom, the word 'LOGIN' is repeated in white, bold, uppercase letters. Below that, the text 'Don't have an account?' is in a smaller white font, followed by the words 'SIGN UP' in white, bold, uppercase letters.

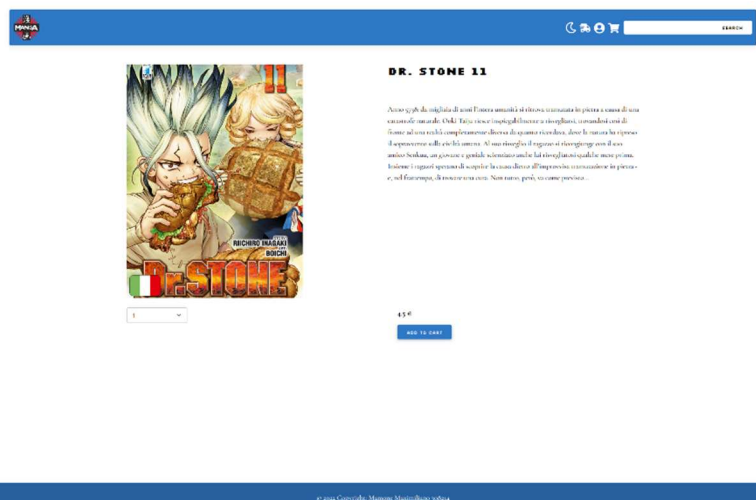
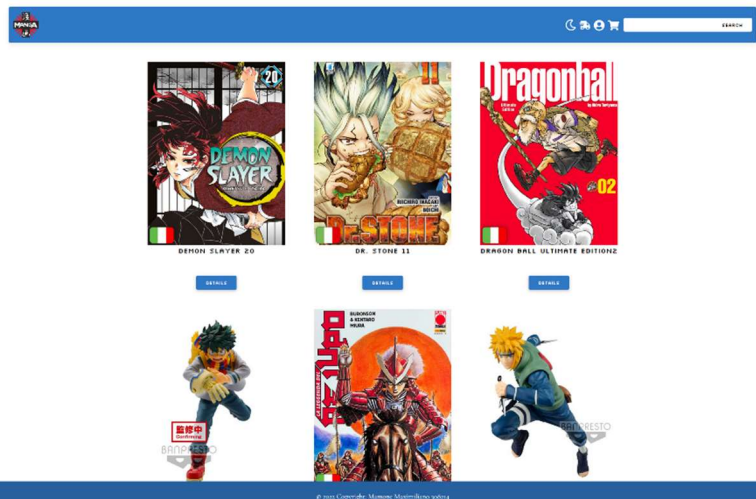
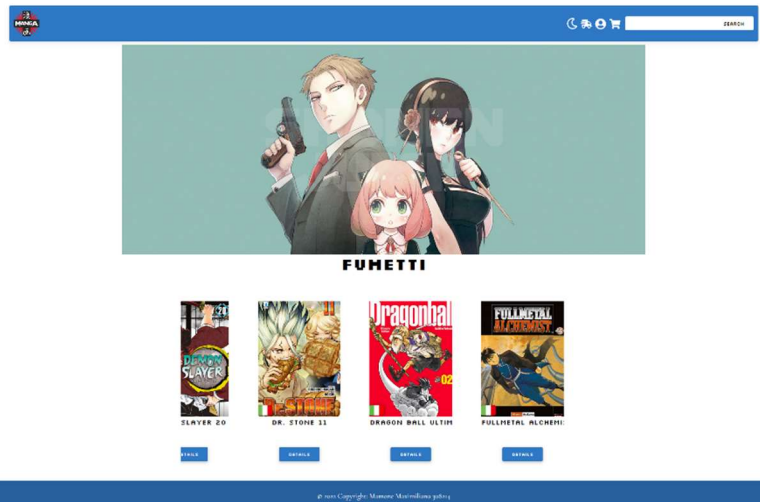


A blue rectangular form with rounded corners. At the top, the word 'SIGNUP' is written in white, bold, uppercase letters. Below it, the text 'Please enter your data' is in a smaller white font. There are five white input fields: 'Firstname', 'Lastname', 'Username', 'Address', and 'Password'. Below the 'Password' field is a 'Repeat Password' field. At the bottom, the word 'SIGNUP' is repeated in white, bold, uppercase letters. Below that, the text 'Already have an account?' is in a smaller white font, followed by the word 'LOGIN' in white, bold, uppercase letters.

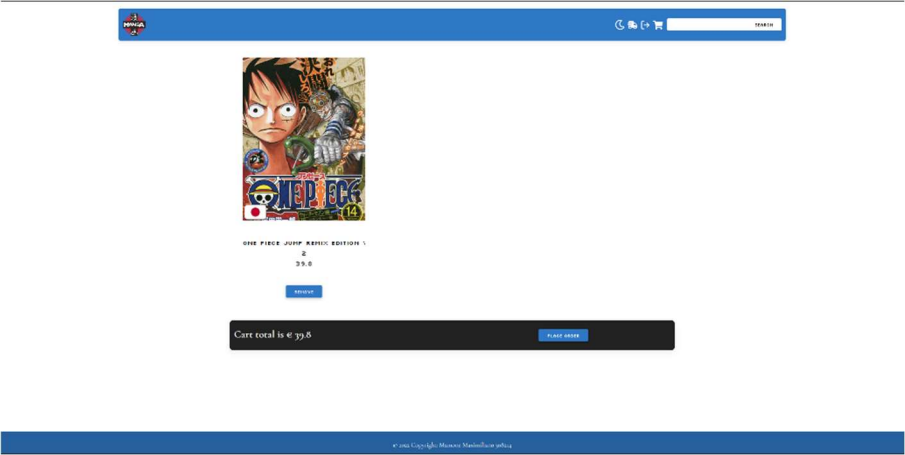
Navigazione utente:

Tutti gli utenti,
registrati o meno
possono accedere alla
homepage,
visualizzare le pagine
dei singoli prodotti ed
effettuare ricerche.

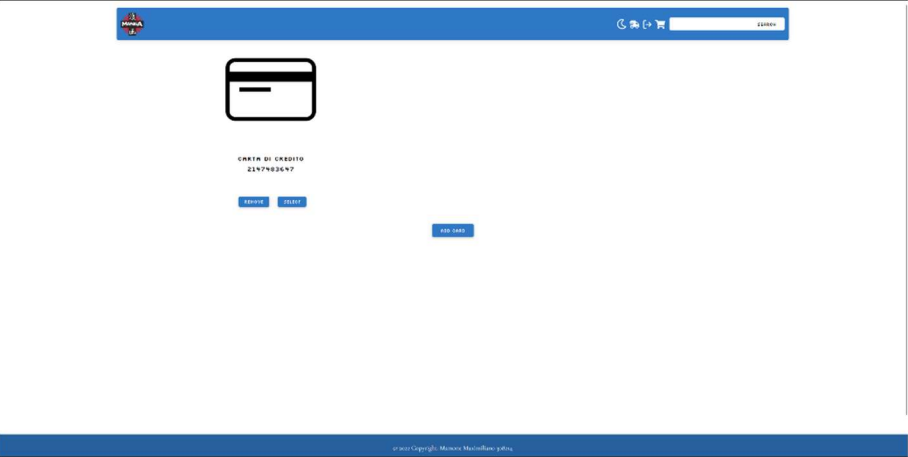
Tramite la barra di
ricerca in alto a destra
e i pulsanti “Details”
che danno accesso
alle pagine di ogni
prodotto possono
infatti navigare
liberaamente nel sito



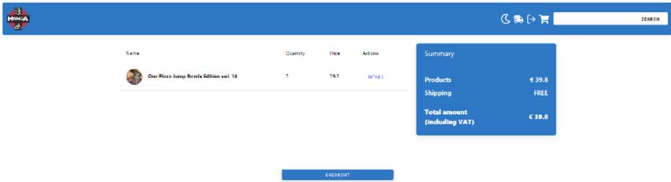
Gli utenti registrati inoltre hanno accesso al carrello



Possono aggiungere, rimuovere e selezionare carte di credito per gli acquisti



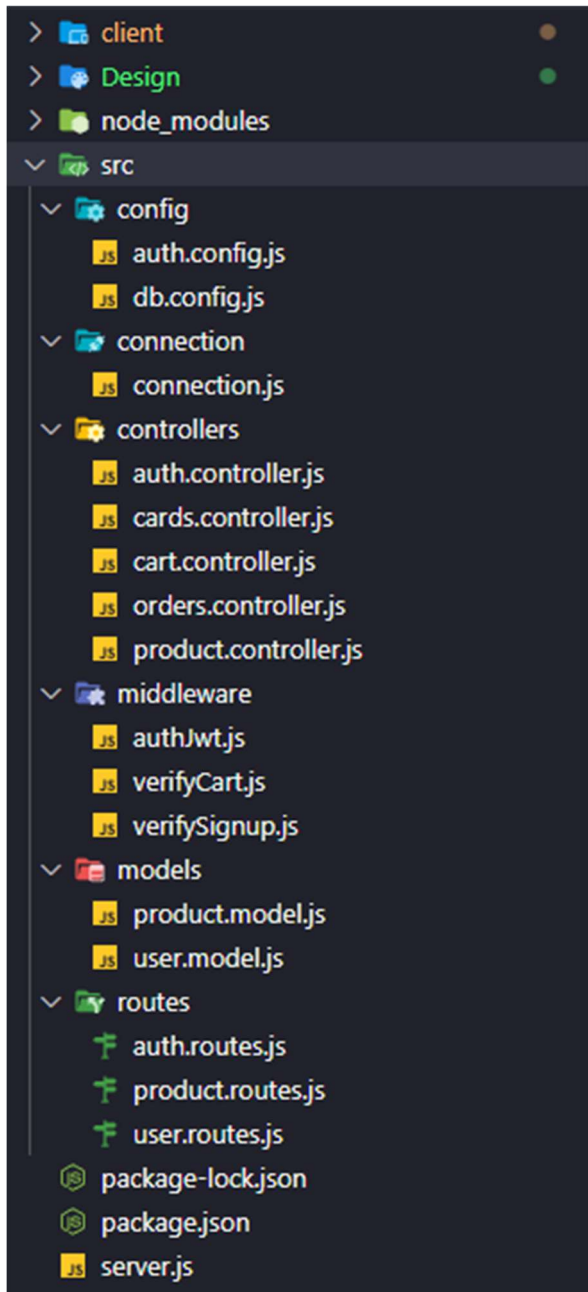
Confermare l'acquisto nella pagina di riepilogo dell'ordine



E visualizzare lo stato degli ordini effettuati



Backend

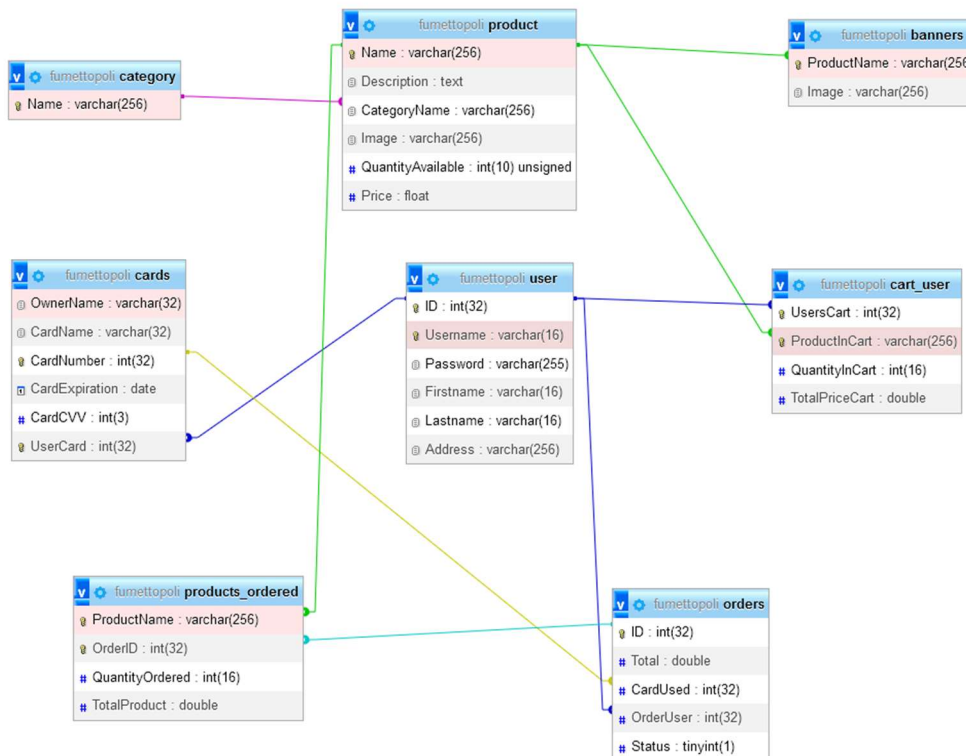


Il backend è invece un semplice server **ExpressJS + NodeJS** che tramite l'utilizzo di modelli, middleware, controller e gestione delle routes permette di effettuare richieste **HTTP** in grado di fornire alla pagina react tutte le informazioni necessarie e aggiornare il database in base alle azioni effettuate dall'utente.

Ho mantenuto quella che è la struttura file riconosciuta come standard per un server ExpressJS come visibile nell'immagine. Importanti da notare le cartelle middleware, models e routes che contengono i principali componenti di funzionamento del backend mentre le cartelle config e connection hanno al loro interno i dati per l'accesso al database e la chiave segreta per la generazione dei token.

Database

Il database, infine, è un database relazionale MySQL creato e gestito tramite l'applicazione XAMPP e i suoi strumenti



Le tabelle utilizzate sono:

- **user**, contenente le informazioni del profilo utente e il codice ID associato internamente al profilo
- **product**, contenente le informazioni dei prodotti
- **cart_user**, contenente i prodotti nei carrelli di ogni utente
- **orders**, contenente le informazioni sugli ordini effettuati dagli utenti
- **products_ordered**, contenente le informazioni sui prodotti ordinati
- **cards**, contenente le informazioni sulle carte di credito degli utenti
- **category**, contenente le informazioni sulle categorie di prodotti disponibili
- **banners**, contenente i banner pubblicitari della homepage

Conclusioni

Lo sviluppo dell'e-commerce si è rivelato stimolante e interessante per quanto riguarda la mia crescita come sviluppatore web, sia per quelle che sono le mie capacità di design estetico sia per la quantità di cose che ho dovuto imparare per riuscire a portare a termine il progetto nell'ambito React/Express.

La piattaforma potrebbe essere ulteriormente migliorata, aggiungendo ad esempio un'interfaccia amministratore che sia in grado di aggiungere prodotti direttamente dal sito senza dover utilizzare il database.