# NOLO QUIDDITCH

## Progetto di Irene Lassanutti, Federica Lolli e Martina E. Quaranta per il corso di Tecnologie Web 2020/2021 - 41731

### Presentazioni

Nolo Quidditch è un sito ideato per tutti coloro che hanno voglia di giocare a Quidditch perché dà la possibilità di noleggiare per un intero giorno tutto ciò che necessita una partita: dalle scope fino all’abbigliamento, dalle pluffe fino alle porte!

La richiesta di noleggio è semplice, veloce e intuitiva: basterà eseguire un login con le credenziali dei clienti già registrati nel database, inserire la quantità degli oggetti che si desiderano e procedere al pagamento ricevendo la fattura in pdf (scaricabile anche dalla pagina “I tuoi ordini”), che avverrà in galeoni alla cassa al momento del ritiro della merce.

Per quanto riguarda la restituzione, invece, basterà che il cliente riconsegni la merce e l’amministratore, a quel punto, potrà concludere il noleggio attraverso una semplice modifica sul portale.

Per quanto riguarda invece la parte back-end, quindi per i dipendenti, il sito offre una comodissima finestra su tutto ciò che viene noleggiato attraverso uno storico ben dettagliato.

Dalla pagina dell’amministratore, quindi, sarà possibile tenere sotto controllo lo stato del noleggio, visionare la data, la merce e l’utente. Inoltre, sarà semplicissimo modificare o aggiungere o eliminare i prodotti:

per aggiungere un nuovo prodotto dovrà semplicemente inserire il nome, una breve descrizione, la quantità massima, il prezzo per giornata e la disponibilità dell’oggetto;

per modificare un prodotto basterà scrivere le modifiche negli appositi campi e cliccare in seguito su “modifica” per salvare;

per eliminare un prodotto sarà necessario cliccare su “elimina”.

### Installazione

Per l’installazione saranno necessari:

l’applicazione Node;

il framework Express;

la libreria Puppeteer;

la cartella con i codici per poter dare inizio alla magia.

### Utilizzo

Una volta installati i programmi, andare sul terminale, posizionarsi nella cartella, dare i comandi “npm i” per l’installazione dei moduli node e “npm i puppeteer” per integrare la libreria necessaria al pdf, poi procedere col comando “npm start” che darà accesso al server sulla porta 3000 del localhost, essendo locale. Aprire un browser (noi abbiamo utilizzato per lo più Chrome) e digitare localhost:3000, si aprirà la pagina relativa al file index.ejs (che contiene l’html) e index.js (che contiene il javascript). Da qui, si potrà cliccare su “ACCEDI QUI” che indirizzerà alla pagina del login (login.ejs e login.js) per poter accedere o come utente o come amministratore.

#### Front-end: utente

Le credenziali di accesso degli utenti sono inserite nel file user.json da cui possono essere modificate, eliminate, aggiunte di nuove. Dal client non è possibile aggiungere nuovi utenti.

Inserite le credenziali, il portale si trasferirà sul file home.ejs e home.js.

La pagina contiene:

un link (“Storico”) che porta allo storico degli utenti nel file storico.ejs e storico.js;

una tabella (dotata di metodo POST per poter comunicare con il server che registra tutti i movimenti del sistema) con i prodotti (inseriti nel file ITEMS.json che compaiono grazie al metodo GET, inserito nel home.js) da cui si può selezionare la quantità desiderata.

es. se utente x ordina 3 Nimbus 2000 su 12, utente y troverà una disponibilità di sole 9 Nimbus 2000 totali. Al termine della disponibilità, il ciclo farà apparire la dicitura “Temporaneamente non disponibile” (‘’’html ‘’’). Una volta scelti i prodotti, il server riceve una richiesta POST con i dati inviati dalla form che vengono elaborati in un ciclo per generare l’ordine tramite cui viene riscritto il file ITEMS.json e vengono generati o riscritti i file del cliente;

un altro link (“Procedi al pagamento”) che indirizza alla pagina finale del file pagamento.ejs, mentre per la fattura vengono richiamati i file ordine.ejs e ordine.js. Da qui, l’utente potrà visionare il riepilogo di ciò che ha noleggiato con il totale dell’acquisto e potrà scaricare il pdf della fattura che avviene attraverso il metodo GET.

#### Back-end: amministratore

Le credenziali di accesso dell’amministratore sono inserite nel file user.json da cui possono essere

modificate, eliminate, aggiunte di nuove. Dal client non è possibile aggiungere nuovi utenti.

Inserite le credenziali, il portale si trasferirà sul file admin.ejs e admin.js. La pagina contiene:

un link (“Storico”) che porta allo storico degli utenti nel file storico.ejs e storico.js;

una tabella (dotata di metodo POST) con stringhe di testo per poter aggiungere nuovi prodotti, che andranno a inserirsi automaticamente nel file ITEMS.json (tramite metodo GET), e che sono codificati con un ID. Con una richiesta POST vengono aggiunti o modificati gli items cliccando si tasti “Aggiungi” o “Modifica” (che si trova nella tabella sottostante) che dicono al sistema di collegarsi al file json per poter aggiungere o modificare quel particolare prodotto;

una tabella (dotata di metodo POST) che è collegata al file ITEMS.json e che restituisce tutti i prodotti all’interno della home, a cui si accompagnano due tasti: modifica e elimina. Il funzionamento del tasto elimina è il seguente: si ha una richiesta con metodo POST e avviene una verifica tra gli ordini già effettuati tramite il blocco try-catch, se non ci sono stati noleggi in corso o anche passati, allora avviene la conferma dell’eliminazione. Si aprirà un alarm attraverso una res.send che conferma l’eliminazione del prodotto. L’eliminazione vera e prpria avviene, però, attraverso un’unica riga: *prodotti = prodotti.filter(p => p.ID != req.body.id);*

### Supporto

Per qualsiasi chiarimento rivolgersi a Irene Lassanutti, Federica Lolli, Martina E. Quaranta.

### Autori

Irene Lassanutti, Federica Lolli, Martina E. Quaranta.

### Stato del progetto

Attivo fino al 08/09/2021 escluso.

### Licenze

I programmi sono tutti OpenSource.