

Modifiche nella richiesta del progetto

Per comodità, ho utilizzato un'immagine trovata su Internet e ho implementato il login tramite JSON Web Tokens (JWT). Inizialmente, avevo pensato di usare un cookie con il valore `loggedIn: <bool>`, ma ho poi deciso che fosse più pratico inviare il codice fiscale dell'utente in ogni richiesta. La scelta di usare JWT è motivata dalla necessità di garantire una maggiore sicurezza. Ho inoltre deciso di eliminare la pagina `about-us`.

Penso anche che realizzare tutte le pagine (tranne la home o landing page) tramite richieste AJAX sia una cattiva idea per motivi legati alla SEO. Una scelta migliore potrebbe essere quella di non caricare l'intera pagina tramite AJAX, ma solo i contenuti dinamici. Per esempio, nella pagina dei piani, i titoli e le descrizioni potrebbero essere già presenti nel file `.html`, mentre i dettagli come numero di persone e prezzi potrebbero essere caricati dinamicamente.

Inoltre, ho migliorato la gestione degli errori nel backend, restituendo codici di stato HTTP più appropriati (ad esempio 401 per accesso non autorizzato o 404 per risorse mancanti). Questo aiuta a mantenere una comunicazione più chiara tra frontend e backend e migliora l'esperienza utente in caso di problemi.

Riepilogo

- **Login JWT:** invia in ciascuna richiesta il codice fiscale del membro in modo sicuro.
- **Pagina "About us"** rimossa.
- **AJAX:** utilizzato per contenuti dinamici (ma attenzione al problema SEO).
- **Gestione degli errori migliorata:** codici HTTP coerenti.
- **Configurazione** tramite file `.env`.

Utilizzo

Prima di tutto, è necessario che il server *PostgreSQL* sia in esecuzione. Per comodità durante il deploy, **non** utilizzo password per l'utente `postgres` (vedi `Makefile`, variabile `CLEAN_PSQLE_FLAGS`).

Successivamente, è necessario generare un file `.env`. Io ho utilizzato la seguente configurazione:

```
1 DB_PASS=
2 DB_NAME=palestra_pwm
3 DB_HOST=localhost
4 DB_USER=postgres
5 JWT_SECRET='NWNK pbagrkgghny_gbxrafyvqrf 15 sbyqre_mvcrfrepvmv 51
6 Ernyvmmnmvbar r Pbafrtan Cebtrggb pbagrkgghny_gbxrafyvqrf 81'
```

Per usare il progetto, basta creare il database (`make clean_db`), eseguire il deploy del database (`make db`) e avviare il server con `node`:

```
1 npm install      # installa le dipendenze definite in ./package.json
2 make clean_db db # crea il database, tutte le tabelle e aggiunge i dati
3 node server.js   # avvia il server
```

Esempi di utenti

Qui ci sono alcuni esempi di codice fiscale di utenti. Le password sono uguali al codice fiscale dell'utente, con `_123` aggiunto alla fine.

- RSSMRC80A01F205X
- BNCGLA85B42F205Y
- FRRLSS90C03F205Z
- RMNSFA88D44F205A