Applicazioni Web I – Esame # 4 (scadenza 2021-01-20 alle 23:59)

"Pizza"

VERSIONE FINALE – Aggiunte e correzioni sono mostrate in rosso.

Progettare e implementare un'applicazione web per gestire gli ordini di un negozio di pizza take-away. Si noti che, per semplicità, non è necessario gestire le date: si assume che tutti gli ordini siano effettuati nello stesso giorno (non specificato).

L'applicazione deve soddisfare i seguenti requisiti:

- **Tipi di pizza**
  - Le dimensioni della pizza possono essere di 3 tipi: Small, Medium, Large (S, M, L).
  - Ogni pizza può essere ordinata con un insieme di condimenti: fino a 2 per la Small, fino a 3 per la Medium, e fino a tre per ognuna delle 2 parti per la pizza Large. Nel caso di pizza Large con una sola parte il massimo sono 6 condimenti.
  - I condimenti sono: olive, prosciutto, bacon, funghi, uova, carciofi, frutti di mare, patatine, verdure. I condimenti possono essere combinati come desiderato rimanendo nel limite numerico indicato per ogni dimensione di pizza, ma con i seguenti vincoli addizionali: i frutti di mare possono essere richiesti solo per la pizza Large e, se presenti, devono essere presenti in entrambe le parti. In aggiunta, è possibile richiedere la pizza (di qualsiasi dimensione) completamente senza pomodoro. Il pomodoro non è considerato nel numero totale dei condimenti.
  - Più pizze della stessa dimensione e con gli stessi condimenti possono essere selezionate modificando un apposito contatore durante la fase di ordinazione: per esempio, deve essere possibile ordinare 3 pizze S con olive e funghi senza reimmettere tre volte gli stessi condimenti ma semplicemente selezionando 3 come numero delle pizze.
- Ogni giorno il negozio può produrre solamente un numero massimo di pizze di ciascuna dimensione: rispettivamente, 10 S, 8 M, 6 L. Questa informazione deve essere immagazzinata nel database, insieme ai prezzi delle pizze discussi nel seguito.
- **Un generico utente (autenticato o no) può visionare liberamente la disponibilità** delle 3 dimensioni di pizza, così come la lista dei condimenti che possono essere usati per ogni dimensione di pizza.
- Un utente autenticato può creare il suo ordine di pizze creando una lista di pizze specificando le caratteristiche in una **pagina di configurazione interattiva**
  - Tutta l'interazione per la configurazione delle pizze deve essere gestita sul client, ad eccezione del controllo delle disponibilità delle dimensioni delle pizze che deve essere effettuato in tempo reale ~~ogni volta che nuove pizze sono richieste~~ quando l'ordine è completo e sta per essere inviato, per evitare di lasciar configurare all'utente delle pizze che poi non sarebbero disponibili.
  - Mentre l'utente configura l'ordine, il prezzo totale è aggiornato in tempo reale a ogni cambiamento (aggiunta/rimozione, o modifica dei condimenti o delle quantità). Il prezzo totale è la somma del costo di tutte le pizze nell'ordine. Ogni dimensione di pizza (S, M, L) ha un prezzo che è immagazzinato nel database: S costa 4 euro, M costa 6 euro, L costa

10 euro. I frutti di mare aumentano del 20% il prezzo totale delle pizze sulle quali sono presenti. Si applica uno sconto del 10% sul totale dell'ordine se più di 3 pizze sono incluse nell'ordine.

- Successivamente, quando l'utente invia l'ordine da eseguire al negozio, il negozio verifica che un numero sufficiente di pizze delle dimensioni richieste siano ancora disponibili e conferma l'ordine, altrimenti un opportuno messaggio informativo viene mostrato (per es. non ci sono sufficienti pizze L), e l'utente è reinviato al configuratore che deve consentire di correggere l'ordine ~~appena inviato~~.
- 5 secondi dopo la conferma dell'ordine, un messaggio deve apparire sullo schermo riguardo al fatto che le pizze sono pronte per essere ritirate dall'utente identificato tramite la sua email. Il messaggio deve stare sullo schermo fino a che il bottone di chiusura è cliccato da qualcuno che usa l'applicazione.
- Gli utenti devono poter sempre controllare la lista dei propri ordini passati: la lista deve mostrare una riga per ogni ordine. La riga mostra il numero totale di pizze e il prezzo totale pagato. Una vista dettagliata dell'ordine che include le stesse informazioni presentate nel configuratore deve essere mostrata (e nascosta) quando la riga viene cliccata.
  Suggerimento: una vista immutabile del configuratore può essere riciclata a questo scopo se si ritiene appropriato.

Requisiti del progetto

- L'architettura dell'applicazione e il codice sorgente devono essere sviluppati adottando le migliori pratiche (best practices) di sviluppo del software, in particolare quelle per le single page applications (SPA) che usano React e REST.
- Il progetto deve essere realizzato come applicazione React, che interagisce con un'API REST implementata in Node+Express. Il database deve essere memorizzato in un file SQLite.
- La comunicazione tra il client ed il server deve seguire il pattern "React Development Proxy" e React deve girare in modalità "development".
- La directory radice del progetto deve contenere un file `README.md` e contenere due subdirectories (`client` e `server`). Il progetto deve poter essere lanciato con i comandi: "`cd server; nodemon server.js`" and "`cd client; npm start`". Viene fornito uno scheletro delle directory del progetto.
- L'intero progetto deve essere consegnato tramite GitHub, nel repository creato da GitHub Classroom.
- Il progetto **non deve includere** le directories in `node_modules`. Esse devono essere ricreabili tramite il comando "`npm install`", subito dopo "`git clone`".
- Il progetto può usare librerie popolari e comunemente adottate (come per esempio `moment.js`, `react-bootstrap`, ecc.), se applicabili e utili. Tali librerie devono essere correttamente dichiarate nel file `package.json` cosicché il comando `npm install` le possa scaricare.
- L'autenticazione dell'utente e l'accesso devono essere realizzati tramite token JWT, memorizzato in un cookie http-only. Non è richiesto alcun ulteriore meccanismo di protezione.

- Il database del progetto deve essere definito dallo studente e deve essere precaricato con almeno 5 utenti di test (la procedura di registrazione non è richiesta), con almeno un utente che ha effettuato 2 ordini, uno che ha effettuato 1 ordine, e un totale di 10 pizze ordinate.

Contenuto del file README.md

Il file README.md deve contenere le seguenti informazioni (un template è disponibile nello scheletro del progetto – in generale, ogni spiegazione non dovrebbe essere più lunga di 1-2 righe):

1. Una lista delle Route dell'Applicazione React, con una breve descrizione dello scopo di ogni route
2. Una lista delle API REST offerte dal server, con una breve descrizione dei parametri e delle entità scambiate
3. Una lista delle tabelle del database, con il loro scopo
4. Una lista dei principali componenti React usati nell'applicazione
5. Uno screenshot della **pagina per configurare una pizza** (embeddando una immagine messa nel repository)
6. Username e password degli utenti di test, che indichi anche quali utenti hanno ordinato qualcosa e quanti ordini ognuno ha effettuato.

Procedura di sottomissione (importante!)

Per sottomettere correttamente il progetto è necessario:

- essere **iscritti** all'appello
- fare il **push del progetto** nel **branch `main`** del repository che GitHub Classroom ha generato per lo studente. L'ultimo commit (quello che si vuole venga valutato) deve essere **taggato** con il tag `final`.
  NB: recentemente github *ha cambiato il nome del branch di default da master a main*, porre attenzione al nome del branch utilizzato, specialmente se si parte/riutilizza/modifica una soluzione precedentemente caricata su un sistema git.

Nota: per taggare un commit, si può usare (dal terminale) i seguenti comandi:

```
# ensure the latest version is committed
git commit -m "...comment..."
git push

# add the 'final' tag and push it
git tag final
git push origin --tags
```

In alternative, è possibile inserire un tag dall'interfaccia web di GitHub (seguire il link 'Create a new release').

Per testare la propria sottomissione, questi sono i comandi che useremo per scaricare il progetto... potreste volerli provare in una directory vuota:

```
git clone ...yourCloneURL...
cd ...yourProjectDir...
git pull origin main  # just in case the default branch is not main
git checkout -b evaluation final # check out the version tagged with
'final' and create a new branch 'evaluation'
```

```
(cd client ; npm install)
(cd server ; npm install)
```

Assicurarsi che tutti i pacchetti (package) necessari siano scaricati tramite i comandi `npm install` . Fare attenzione se alcuni pacchetti sono stati installati a livello globale perché potrebbero non apparire come dipendenze necessarie: potreste voler testare la procedura su un'installazione completamente nuova (per es. in una VM).

Il progetto sarà testato sotto Linux (si faccia attenzione al fatto che Linux è case-sensitive nei nomi dei files, mentre MacOS-X e Windows non lo sono).