

“Noleggio veicoli”

VERSIONE FINALE – Aggiunte e correzioni sono mostrate in **rosso**

Progettare ed implementare un’applicazione web per gestire un’azienda che noleggia veicoli.

L’azienda è proprietaria di un insieme di veicoli. I veicoli sono suddivisi in 5 categorie (A...E), secondo la loro dimensione. Ogni auto è anche caratterizzata da una marca (per es. Audi, Ford, Fiat, Mercedes, ...) e un modello (una stringa).

Gli utenti non autenticati possono navigare l’intero insieme dei veicoli, filtrandoli per categoria e/o marca (l’utente può contemporaneamente selezionare una o più categorie e una o più marche). **La lista dei veicoli si aggiorna in tempo reale nel momento in cui i filtri sono modificati.**

Una volta autenticati, gli utenti hanno accesso a una *pagina di configurazione interattiva* che rimpiazza la pagina di navigazione originale. Nel configuratore gli utenti possono impostare i parametri per una nuova richiesta di noleggio. I parametri **del noleggio sono**: giorno di inizio, giorno di fine, categoria dell’auto, età del guidatore, numero di guidatori aggiuntivi, numero stimato di km da percorrere **al giorno**, richiesta di assicurazione extra (**si/no**).

Per ogni combinazione, **se i parametri sono correttamente specificati**, la pagina dovrà mostrare interattivamente il numero di auto disponibili (che soddisfano tutti i criteri) e il *prezzo di noleggio* per quella soluzione. **Il numero di auto disponibili deve coincidere con quante auto della stessa categoria sono disponibili per tutta la durata del noleggio.** Il prezzo del noleggio è calcolato considerando differenti fattori (si veda la Tabella Prezzi), che sono anche dipendenti dal numero di veicoli disponibili nei giorni del noleggio (quando rimangono pochi veicoli il prezzo aumenta). Inoltre, dopo aver **terminato** 3 noleggi con successo, il cliente ottiene il 10% di sconto sul suo prossimo noleggio.

Se l’utente accetta la proposta del configuratore interattivo (**l’auto specifica è selezionata dal sistema e non è necessario mostrarla**), l’utente deve pagare immediatamente, non appena conferma la prenotazione (si deve sviluppare un’API “stub” per finti pagamenti, che riceve l’informazione della carta di credito¹ e il valore da addebitare, **e ritorna un risultato di tipo si/no; questo servizio deve solo validare la presenza di tutti i campi, e non il loro valore**).

Gli utenti autenticati, in aggiunta, possono controllare la lista delle loro prenotazioni future (e potenzialmente cancellarle), e anche controllare lo storico **dei noleggi correnti o passati** in una pagina dedicata.

Tabella Prezzi

Elemento	Prezzo
Auto di categoria A	80 €/giorno
Auto di categoria B	70 €/giorno
Auto di categoria C	60 €/giorno

¹ Nome completo, numero della carta, codice CVV

Auto di categoria D	50 €/giorno
Auto di categoria E	40 €/giorno
Meno di 50 km/giorno	-5%
Meno di 150 km/giorno	0%
km/giorno illimitati	+5%
Età del guidatore inferiore a 25	+5%
Età del guidatore superiore a 65	+10%
Guidatori addizionali (si applica una sola volta per 1 o più guidatori extra)	+15%
Assicurazione Extra	+20%
Meno di 10% di veicoli rimasti in garage nella stessa categoria nei giorni del noleggio	+10%
Cliente frequente (almeno 3 noleggi terminati)	-10%

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". Uno scheletro delle directory di progetto sarà fornito.
- 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.
- 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 esempio (la procedura di registrazione non è richiesta), e almeno 20 auto di almeno 5 marche differenti.

Contenuto del file README.md

Il file README.md deve contenere le seguenti informazioni (un template sarà 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 (se è utilizzato il routing), 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 del configuratore interattivo (embeddando una immagine messa nel repository)
6. Username e password dei 5 utenti di test. Identificare un utente che è un cliente frequente.