

Prova di Laboratorio - Tecnologie Web

23 gennaio 2025

La prova dovrà essere svolta a partire dal progetto Flask di base contenuto nel file **.zip di Moodle**. Il progetto contiene:

- Una route `/index` con un template HTML di base `base.html`.
- Un file `products.csv` nella cartella `data`, contenente i dati di prodotti in vendita.
- Un file `style.css` per lo stile.

Il file `products.csv` contiene le seguenti colonne: codice prodotto, nome prodotto, categoria, prezzo, disponibilità e descrizione.

Il codice aggiunto deve essere commentato con **ESERCIZIO #<numero esercizio>**.

A partire da questo progetto, si chiede:

1. **(2pt)** Scrivere una funzione che legga e restituisca tutti i prodotti presenti nel file `products.csv`. La funzione deve essere richiamata dove necessario in `app.py` per ritornare l'elenco dei prodotti.
2. **(3pt)** Modificare la route `/index` per renderizzare un template `index.html`, che estenda il template di base. Il template deve mostrare una tabella con l'elenco dei prodotti disponibili. Se non ci sono prodotti, visualizzare il messaggio “Nessun prodotto disponibile”. Nella tabella deve essere riportato solo il nome del prodotto, il prezzo e la dicitura “*Esaurito*” in una colonna a parte nel caso il prodotto non sia disponibile.
3. **(5pt)** Rendere cliccabile il nome di ogni prodotto nella tabella. Quando un utente clicca su un prodotto, reindirizzarlo a una pagina dedicata `/product/<product_code>` che mostri i dettagli del prodotto corrispondente, formattati in modo chiaro e leggibile. Nella pagina di dettaglio devono essere presenti tutte le informazioni del prodotto. Inoltre, inserire nella pagina di dettaglio un pulsante per tornare a `index.html`.

4. **(6pt)** Nella pagina del prodotto, aggiungere un pulsante per acquistare il prodotto. Il pulsante deve chiamare un'API Flask `/api/buy/<product_code>` tramite il metodo POST. Ridurre di uno la disponibilità del prodotto nel file `products.csv`. Gestire eventuali errori, come la mancanza di disponibilità, mostrando messaggi appropriati all'utente.
5. **(6pt)** Creare due API:
 - a. Una per restituire tutti i prodotti in formato JSON.
 - b. Una per restituire i dettagli di un singolo prodotto basandosi sul codice del prodotto.

Non è richiesto che ritorni anche la lista con le colonne del file, solo il dizionario contente i dati dei prodotti.
6. **(2pt)** Creare una route `/react` lato Flask per renderizzare il template `index_react.html`, dove costruire una Single Page Application (SPA) con React. Tutte le route impostate con React devono seguire il prefisso `/react`, ad esempio `/react/product_detail`, `/react/buy`, ecc.
7. **(5pt)** Creare una componente React `ProductList` per visualizzare l'elenco dei prodotti in una tabella analoga a quella del punto 3, senza il requisito della cliccabilità del nome.
8. **(4pt)** Nella componente `ProductList`, inserire un form con un campo di input e un pulsante che implementi un filtro dei prodotti con le seguenti specifiche: se il campo contiene una stringa, la componente deve renderizzare solo i prodotti il cui nome contiene quella stringa; se il campo è vuoto, la tabella deve tornare ad elencare tutti i prodotti presenti.

Per consegnare la prova di laboratorio :

- Creare un'unica cartella compressa (ZIP, o RAR), denominata `Cognome_Nome_Matricola.estensione` (sostituire Cognome, Nome e Matricola con i propri dati e sostituire estensione con l'estensione utilizzata, es. zip o rar)
- La cartella compressa deve contenere le cartelle iniziali (`data`, `static`, `templates`), e i file richiesti nel testo dell'esame posizionati nelle cartelle apposite.
- Caricare su MOODLE il file zip
- Nella sezione “Commenti” su Moodle inserire il numero degli esercizi svolti (es. “Svolti gli esercizi numero 1, 3, 5, ...”)