

Prova di Laboratorio - Tecnologie Web

12 febbraio 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 `video.csv` nella cartella data, contenente i video presenti vendita.
- Un file `comments.csv` nella cartella data, contenente i dati di commenti associati ad ogni video.
- Un file `style.css` per lo stile.

Il file `video.csv` contiene le seguenti colonne: codice video, nome video, link video.

Il file `comments.csv` contiene le seguenti colonne: codice video, commento.

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

A partire da questo progetto, si chiede:

1. **(3 pt)** Scrivere due funzioni che leggano e restituiscano tutti i video presenti nel file `video.csv` emmenti i presenti nel file `comments.csv`. Le funzioni devono essere richiamate, dove necessario, in `app.py` per ritornare l'elenco dei video e commenti. **Ci si aspetta che il caricamento dei file venga sviluppato in una sola parte del codice**, in caso non sia così verranno tolti dei punti anche se il caricamento è corretto in tutte le parti sviluppate.
2. **(2pt)** 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 video disponibili, con nome del video e numero di commenti associati al video.
3. **(5pt)** Rendere cliccabile il nome del video nella tabella. Quando un utente clicca su un video, reindirizzarlo a una pagina dedicata `/video/<video_code>` che mostri i dettagli del video corrispondente, formattati in modo chiaro e leggibile. Nella pagina di dettaglio deve essere presente il video incorporato (embedded player) che può essere direttamente visualizzato dalla pagina, insieme al nome

del video e la lista dei commenti. Inoltre, inserire nella pagina di dettaglio un pulsante per tornare a `index.html`.

4. **(6pt)** Nella pagina del video, aggiungere un *textbox* e un *bottone* ‘Aggiungi’. Il bottone deve chiamare un’API Flask tramite il metodo POST per aggiungere un commento nuovo al video. Aggiungere nel file `comments.csv` il nuovo commento associato al codice del video.
5. **(6pt)** Creare due API:
 - a. Una per restituire tutti i commenti in formato JSON.
 - b. Una per restituire i commenti legati ad un solo video.
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/videos`, `/react/detail`, ecc.
7. **(4pt)** Creare una componente React `VideoList` per visualizzare l’elenco dei prodotti in una tabella che visualizzi codice e nome di ogni video. Associarla alla route `/react/videos`.
8. **(5pt)** Nella componente `VideoList`, inserire sopra la tabella un form per l’aggiunta di un nuovo video. Il form deve prendere in input un nome e il link del video. Il codice deve essere generato automaticamente lato backend all’interno di un’API Flask creata apposta per aggiungere il video nel file `video.csv`.

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, ...”)