

Prova Esame di Laboratorio

15 Luglio 2025

Il progetto prevede lo sviluppo di un portale per la creazione automatica di quiz, ognuno dei quali contenente un insieme di domande a risposta chiusa, formati da domanda e quattro possibili risposte.

Di seguito le specifiche di sviluppo.

La prova dovrà essere svolta a partire dal progetto Flask di base contenuto nel file

- Una route `/index` con un template HTML di base.
- Un file `quizzes.csv` sotto la cartella data, contenente i dati dei quiz generati.
- Un file `questions.csv` nella cartella data, contenente le possibili domande da inserire in un quiz.
- Un file `style.css` per lo stile.

Il file `questions.csv` contiene le seguenti colonne: `question_code`, `statement`, `answer_1`, `answer_2`, `answer_3`, `answer_4`, `correct_answers` (lista degli indici delle risposte corrette).

Il file `quizzes.csv` contiene le seguenti colonne: `quiz_code`, `name`, `question_list` (contiene gli `question_code` delle domande contenute nel quiz).

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

A partire da questo progetto si chiede:

1. **(3pt)** Scrivere due funzioni per leggere e restituire:

- Tutti i quiz presenti nel file `quizzes.csv`.
- Tutte le domande presenti nel file `questions.csv`.

Le funzioni devono essere richiamate, dove necessario, in `app.py` per ottenere l'elenco dei videogiochi e delle valutazioni. Il caricamento dei file deve essere implementato in un'unica parte del codice; in caso contrario, verranno sottratti 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 di tutti i quiz visualizzando il codice, il nome del quiz e il numero di domande.
3. **(6pt)** Nella pagina `index.html`, sopra la tabella deve essere presente un form con due campi e un bottone per aggiungere un nuovo quiz. Il form deve contenere due campi di input, uno per il nome del quiz e l'altro per il numero di domande che dovranno essere presenti nel quiz. Una volta premuto il bottone, l'applicazione dovrà selezionare automaticamente le domande tra quelle disponibili, senza ripetizioni. Inserire i dati del nuovo quiz nel file `quizzes.csv`. Nel caso il numero di domande inserito fosse più grande del numero di domande presenti, dovrà essere mandato un alert di errore.

4. **(6pt)** Rendere cliccabile le righe della tabella. Quando un utente clicca su un videogioco, reindirizzarlo a una pagina dedicata `/quiz/<quiz_code>` che mostri i dettagli del quiz. Dovranno essere visualizzate tutte le domande presenti nel quiz, per ognuna delle quali dovranno essere mostrate le seguenti proprietà: l'enunciato della domanda e le quattro risposte evidenziando quali sono le risposte corrette. Nella pagina dovrà anche essere presente un pulsante per tornare a `index.html`.
5. **(5 pt)** Creare due API:
- Una per restituire tutte le possibili domande in JSON.
 - Una per restituire le domande legate a un quiz in JSON.
- Non è richiesto che le API restituiscano anche le colonne del file, ma solo un dizionario contenente i dati delle valutazioni.
6. **(2 pt)** 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/questions`, `/react/question/id`, ecc.
7. **(4 pt)** Creare un componente React `QuestionList` per visualizzare l'elenco di tutte le possibili domande in una tabella che mostri il codice e l'enunciato di ogni domanda. La componente deve contenere un bottone “Aggiungi domanda”. Associare questo componente alla route `/react`.
8. **(5 pt)** Il bottone aggiungi domanda deve portare alla visualizzazione di un componente `NewQuestion`, contenente un form per aggiungere una nuova domanda. Il form deve permettere l'inserimento dell'enunciato della domanda, le quattro possibili risposte e specificare quali risposte sono giuste. Aggiungere la nuova domanda nel file `questions.csv`. La componente deve stare sotto la route `/react/new_questions`.

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 1,2,3,4,5,...”)