

Esame di Tecnologie Web - parte II

CdL in Informatica per il Management

Tempo a disposizione: 2h

INFORMAZIONI PRELIMINARI

EOL: <https://eol.unibo.it/>

VIRTUALE: <https://virtuale.unibo.it>

L'esame consiste nella scrittura di codice HTML, CSS e Javascript ed esercizi su API ed altri argomenti del corso. Scrivere il codice usando un editor di testo (jEdit o Gedit). Per testare il risultato usare il browser Firefox o Chromium.

La scelta dei nomi dei file, se non espressamente indicati, non è rilevante. I file possono essere salvati in directory diverse. E' possibile comprimere file/directory e decomprimere.

Consegnare i file su EOL. Non consegnare file temporanei e nominare i file in modo chiaro (i numeri degli esercizi devono corrispondere!).

NON è consentito l'uso di testi e appunti cartacei.


E' consentito il solo accesso a Virtuale. NON è consentito l'accesso ad altre risorse on-line.

NOTE TECNICHE

Risorse utili allo svolgimento della prova:

- Librerie jQuery, Bootstrap: <http://diiorio.nws.cs.unibo.it/twe/lib/>

Combinazioni di tasti e note sulle applicazioni:

- Tasto "Windows" : per aprire menù con le applicazioni disponibili
 - *Nautilus*: esplora file
 - *Firefox/Chromium*: browser
 - *jEdit, GEdit*: editor di testo
- ALT+TAB: per passare da un'applicazione all'altra
- CTRL+C, CTRL+V: copia&incolla
- CTRL+R: ricarica pagina nel browser
- CTRL+S: salva file
- Per cambiare le impostazioni del testo in jEdit: menù "Preferences"->Global Options->jEdit->Textarea

TESTO DEL COMPITO

Esercizio 1

Scrivere il codice HTML e CSS per ottenere la visualizzazione mostrata in Figura 1 (pagina successiva) quando il documento è caricato in un browser.

Oltre alle caratteristiche tipografiche già evidenti in figura, si tenga presente che:

- Nella versione desktop, barra di navigazione e layout occupano il 65% del viewport in larghezza (predisporre anche una versione per dispositivi mobili, dettagli sotto)
- Il layout è composto da:
 - Una barra di navigazione con diverse opzioni (Autenticazione, "Films", "Lists", "Members", "Journal").
 - Una sezione contenente un welcome text con il nome dell'utente autenticato, disposta su due righe, e con alcuni frammenti di stile diverso come mostrato in figura.
 - Una sezione principale contenente, in ordine dall'alto al basso, una riga con le label "New on Letterboxd" e "Your Activity" (quest'ultimo è un elemento cliccabile), una serie di card composte da un'immagine, un divisore che contiene informazioni sul nome utente, e un footer che contiene la valutazione espressa in stelle e una data. **NON è necessario introdurre le altre icone presenti in immagine oltre alle stelle.**
 - Al passaggio del mouse sulla locandina di un film, compare un bordo bianco intorno all'immagine. Al passaggio del mouse su un elemento cliccabile, il colore viene modificato in azzurro.
 - Cliccando su testo o descrizione di un elemento cliccabile o locandina si apre la pagina corrispondente; creare un link arbitrario; **NON** è richiesto creare anche le pagine.
 - Colori e dimensioni esatte di margini e padding non rilevanti, purché appropriate.
- Inoltre **adattare il layout** in modo che sia responsive per schermi di diverse dimensioni (desktop e mobile). La larghezza massima per la versione mobile sarà di 768px. Nella versione mobile, la barra di navigazione in alto non è presente e le locandine sono organizzate in un layout a colonna singola (una card sotto l'altra).

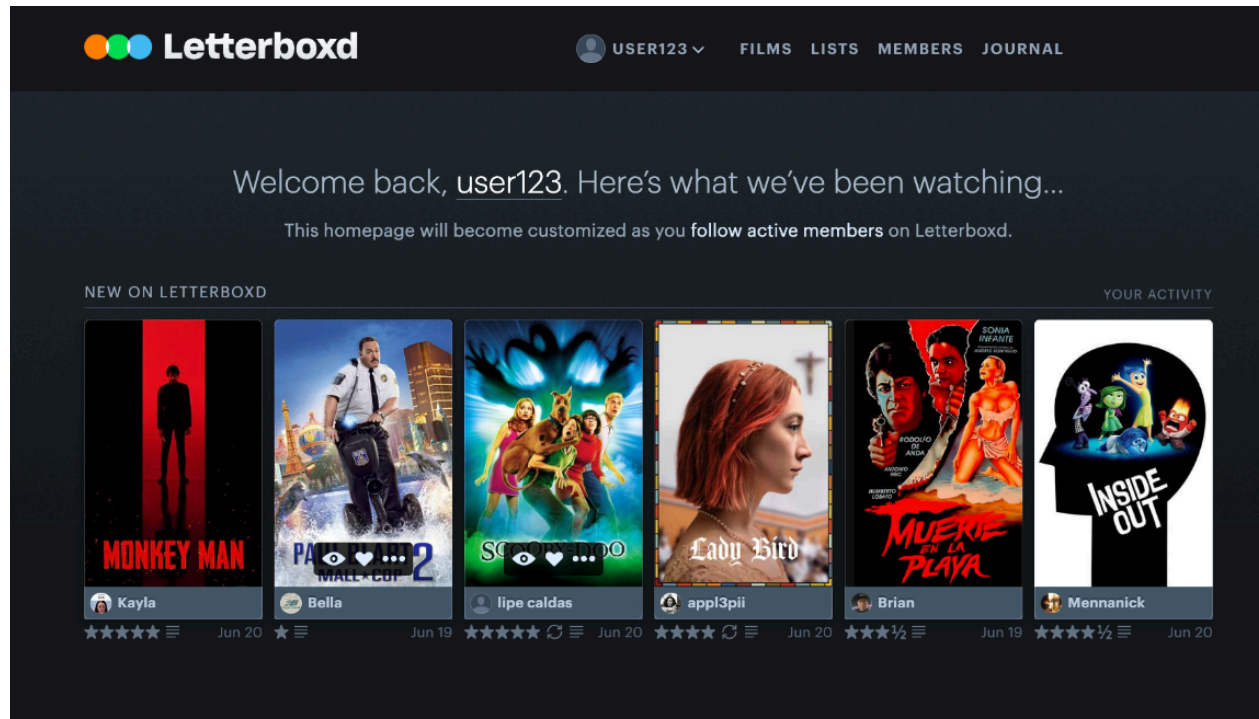
Vincoli:

- non è ammesso usare attributi @id e @class nel sorgente HTML;
- non è ammesso usare tabelle per organizzare il layout
- non è possibile usare Javascript

Risorse: Immagini disponibili su EOL

Per gli studenti DSA: non è richiesto includere la barra di navigazione in alto

Figura 1 - Resa della pagina in un browser:



Esercizio 2

Scrivere il codice HTML, CSS e Javascript per realizzare un'applicazione di rubrica di contatti. Gli utenti devono poter aggiungere, visualizzare e rimuovere contatti dalla rubrica.

Note:

- L'interfaccia deve avere campi di input per inserire nome, numero di telefono e indirizzo email di un contatto. Per semplicità la rubrica contiene solo il nome dei contatti (non è richiesto gestire cognomi o nomi doppi).
- Esiste un unico bottone per aggiungere un contatto alla rubrica.
- **NON sono richieste funzionalità per modificare un contatto esistente.**
- La rubrica mostra una lista ordinata alfabeticamente sul nome per visualizzare i contatti (sempre visibile)
- Quando un utente clicca sul bottone "Aggiungi Contatto", il contatto deve essere aggiunto alla lista nella posizione corretta.
- Ogni contatto nella lista deve avere un bottone "Rimuovi" per eliminarlo dalla rubrica.
- Implementare una validazione di base per assicurarsi che TUTTI i campi siano riempiti prima di aggiungere un contatto e che i dati siano del tipo corretto (es. indirizzo mail sintatticamente corretto o numero di telefono come sequenza di cifre).
- Inoltre non è possibile aggiungere due contatti con lo stesso nome. In questo caso viene visualizzato un messaggio di errore.
- I dettagli di presentazione (posizione elementi, margini, padding, etc.) non sono rilevanti.

Vincoli:

- È ammesso l'uso di attributi `@id` e `@class`.
- È ammesso l'uso di jQuery.
- Bisogna scrivere separatamente il codice HTML, CSS e Javascript.

Per gli studenti DSA: non è richiesto ordinare alfabeticamente la lista per visualizzare i contatti.

Esercizio 3

Progettare un API REST (parziale) per gestire informazioni su una metropolitana.

La metro è organizzata in linee, a loro volta organizzate in fermate. Ogni linea è caratterizzata da un identificativo (di tipo stringa per semplicità) e da una data di apertura (di tipo *date*). Ogni fermata è identificata da un codice numerico (di tipo intero), da un nome (stringa) e da uno stato che può assumere valori *attiva*, *sospesa*, *soppressa*.

Scrivere un file in formato JSON o YAML.

L'API permette di:

1. elencare tutte le fermate *attive* di una linea X
2. aggiungere un insieme di fermate ad una linea X (è possibile aggiungere più fermate nella stessa richiesta)
3. modificare lo stato di una linea in *sospesa*

Scrivere un file in formato JSON o YAML.

Specificare: URL di accesso, metodi HTTP, parametri e risposte con esempi.

L'API restituisce un errore, con codice 400, se i parametri in input non sono corretti.

Note:

- Non è richiesto includere le sezioni host, schemes, servers, tags
- Non è richiesto gestire autenticazione

Per gli studenti DSA: non è richiesto svolgere il punto 2