

Relazione Progetto FPW (A. A. 2021/2022)

Titolo del progetto	FakeFlix
Nome del gruppo	EaSbronz
Componenti del gruppo	Stefano Rocca 60/61/65971 Mirko Contini 60/61/66049 Pietro Musiu 60/61/66041 Marco Monni 60/61/65967

Compilare, per ogni modulo del progetto implementato, la relativa descrizione. Cancellare il testo presente dentro i blocchi e sostituirlo con la relativa descrizione. Ogni descrizione deve contenere massimo 10 righe (font Arial, dimensione 11).

I blocchi relativi ai moduli che NON avete implementato devono essere cancellati dalla relazione.

Consegnare questa relazione in formato PDF (non doc/docx/odt o altro).

Descrizione del tema del progetto
Piattaforma per l'acquisto di film online

Modulo 1: HTML
Il modulo comprende nove pagine. Di queste, sei pagine sono quelle principali (in cui è possibile navigare in determinate condizioni), due vengono incluse in ogni pagina e contengono header e footer e l'ultima è una pagina di output dei messaggi di errore o successo. Ogni pagina contiene i sopra citati header e footer, una navbar, un contenuto principale e uno secondario. Il contenuto secondario comprende una toggle light/dark mode, tre bottoni per la gestione della dimensione del font, una locandina con un film in evidenza e due pubblicità. Il contenuto delle pagine principali è suddiviso in due zone: main, rappresentante il contenuto primario della pagina e aside rappresentante il contenuto secondario.

Modulo 2: CSS
All'interno del modulo le regole sono state divise in quattro breakpoint, rispettivamente per la visualizzazione desktop, smartphone, tablet e uno ad hoc per la visualizzazione corretta della pagina di login e registrazione. Alla fine della pagina sono invece presenti le regole più generali. Grazie ai breakpoint è stato possibile ottenere un layout a dodici colonne, di cui otto occupate dal contenuto principale e le restanti quattro dal contenuto secondario nella visualizzazione desktop; nella visualizzazione tablet e smartphone gli elementi vanno invece ad occupare la totalità delle colonne disponendosi uno sotto l'altro. Un'eccezione è stata fatta per l'area personale, in cui i film inseriti si dispongono sotto il contenuto per una questione estetica. Sono inoltre state utilizzate le flexbox per la gestione del posizionamento degli elementi sullo schermo.

Modulo 3: Server
L'implementazione lato server è piuttosto vasta. In primis abbiamo il package contenente

le classi Utente e Prodotto e le relative factory che si occupano della creazione degli oggetti e che contengono i metodi relativi alle operazioni sul DB come, ad esempio, l'inserimento di un nuovo prodotto. Oltre queste, abbiamo il package contenente le servlet, una servlet per ogni pagina visualizzata e una extra che gestisce il logout. Le servlet si occupano, oltre che del suddetto rendering, anche di richiamare i metodi delle factory in modo da utilizzare i dati inviati o ricevuti dal frontend. Infine, ci siamo avvalsi dell'utilizzo di una classe Utils, in cui sono contenute varie funzioni di utilità come ad esempio una fondamentale per l'implementazione del modulo 6. E' stata gestita la visualizzazione condizionale in base alla sessione con verifiche sia nelle jsp che nelle servlet.

Modulo 4: Database

Il database è organizzato in due tabelle: prodotti e utenti. La prima tabella presenta le seguenti colonne: Titolo (PK), Trama, Genere, Durata, Regista, Locandina (che contiene il path relativo dell'immagine scelta come locandina), Prezzo, Username_aggiunta (contiene il nome utente di chi ha aggiunto il film al database). La tabella utenti presenta invece le colonne: Username (PK), Password, Nome, Cognome, E-mail, Città, Foto (contiene il path locale dell'immagine scelta come foto del profilo).

Inserire username e password degli utenti già registrati da utilizzare durante per la correzione (come descritto nelle specifiche).

rocca	65971
contini	66049
musiu	66041
monni	65967

Modulo 5: Client

Sono stati creati tre script (carousel.js è relativo al prossimo modulo). Il primo, dark-mode-handler, si occupa dello switch tra dark e light mode attraverso un apposito tasto nell'aside delle pagine e l'utilizzo delle funzioni addClass e toggleClass. Abbiamo deciso di dare persistenza tra una pagina e l'altra alla scelta salvando la stessa in localStorage. Avremmo voluto fare la stessa cosa nel font-size-handler, ma le specifiche del progetto ce l'hanno impedito. Il funzionamento è semplice: a seconda del tasto cliccato variano le dimensioni del testo del main content modificando il css da jQuery stesso. Il terzo script è il più corposo client-side-validation. Gestisce sia i contatori dei caratteri che la validazione lato client dei form, evitando l'invio al server fino al raggiungimento di tutte le condizioni. Il mancato raggiungimento viene notificato da messaggi sottostanti il campo interessato.

Modulo 6: Ajax + Upload immagini

Tramite Ajax abbiamo gestito il carosello nella pagina Catalogo, le funzioni utilizzate si possono trovare in carousel.js. Per quanto riguarda l'upload immagini abbiamo cercato di fare in modo che il path fosse dinamico in maniera tale da non dover modificare manualmente la riga, comportamento reso funzionante, dai test effettuati, sia su Windows che sulla VM del corso. Se ciò non dovesse verificarsi (ma sottolineo il successo dei test), sarà sufficiente andare nella classe Utils e modificare il metodo getPathImg, all'interno del

quale si dovranno commentare le righe 54-55, decommentare le righe 52-53 e modificare il path avendo cura di includere fino a /img/, in quanto poi avviene il rilevamento dinamico della cartella prodotti o utenti a seconda del form utilizzato.

Suddivisione del lavoro	
Descrivete, per ogni membro del gruppo, il contributo nella realizzazione del progetto.	
Rocca	Moduli 1 - 2
Contini	Moduli 1 - 2 - 4
Musiu	Moduli 1 - 2
Monni	Moduli 3 - 5 - 6