
Anno accademico 2019/2020

Progetto d'esame

Metodologie di programmazione per il Web

MODALITÀ D'ESAME

L'esame di Metodologie di programmazione per il Web consta di due parti, strettamente collegate ed entrambe obbligatorie:

1. Progettazione e realizzazione individuale di un'applicazione web
2. Dimostrazione e discussione orale sul progetto

La valutazione dei progetti e la dimostrazione/orale riguarderà il materiale consegnato entro 3 giorni prima dell'appello. La dimostrazione e la breve discussione orale inerente il progetto è da svolgersi il giorno dell'appello, previa registrazione su ESSE3.

Per la sessione autunnale 2020 (in presenza): La valutazione dei progetti e la dimostrazione/orale riguarderà il materiale consegnato entro il **1 Settembre**, tramite il link predisposto su DIR.

La discussione riguarderà le scelte di progettazione (layout, componenti utilizzati, struttura del DB, ecc.) nonché le scelte implementative e funzionali adottate.

DESCRIZIONE E REQUISITI

Il progetto consiste in un'applicazione web che deve soddisfare alcuni requisiti tecnici, stilistici e funzionali, dettagliati in seguito.

L'applicazione web dovrà utilizzare le tecnologie illustrate e sperimentate durante il corso. Il "tema" dell'applicazione web da realizzare è riportato di seguito e la sua realizzazione deve comunque rispettare i requisiti.

Per quest'anno accademico, i requisiti e il tema dell'applicazione web non cambieranno e saranno validi fino alla sessione di Febbraio 2021 (*inclusa*).

Requisiti logistici

- Il progetto deve essere realizzato individualmente, senza eccessiva "collaborazione" tra gli studenti.
- Non è prevista né accettabile una consegna (parzialmente o totalmente) in comune con un altro studente del corso.

-
- Il progetto deve essere consegnato secondo le tempistiche riportate nell'ultima pagina di questo documento, ~~comunque entro 3 giorni prima di ogni appello~~.
 - Non è prevista né accettabile una consegna in ritardo.

Requisiti tecnici

L'applicazione web deve rispettare i seguenti requisiti tecnici:

- Utilizzo di JavaScript ad oggetti (classi ES6), sia nel back-end che nel front-end.
- Utilizzo di HTML5 e CSS3, avvalendosi se necessario di framework esterni come Bootstrap e/o di template HTML pre-esistenti, ma personalizzandone lo stile tramite regole create ad-hoc.
- Utilizzo di Node.js ed Express, *nonché* di un database relazionale come SQLite, per il back-end.
- Utilizzo di async/await (e Promise, ovviamente) quando appropriato, nel back-end, nel front-end, o in entrambi.

Inoltre:

- L'applicazione web può essere realizzata come client + server (fetch + API) oppure con pagine web generate dinamicamente dal server, a scelta degli studenti (**tranne che per gli studenti in debito dell'esame da 9 crediti, vedi dopo**).
- L'applicazione web deve avere un target di dispositivi ben preciso, a scelta degli studenti, eventualmente supportando la modalità responsive: mobile-only/first vs. desktop-only/first.
- Il progetto consegnato deve essere interamente testabile dal docente e deve funzionare sulle ultime versioni di Chrome (80+) e Firefox (74+).
- Il codice sorgente deve essere ben scritto e corredato di opportuni commenti laddove necessario.
- Tutte le tecnologie elencate in precedenza devono essere integrate in maniera coesa e uniforme all'interno di un'unica applicazione web.

Extra:

- Fare il deploy dell'applicazione web su qualche servizio online, per esempio su Heroku (<https://www.heroku.com>).

Requisiti stilistici

L'applicazione web deve rispettare i seguenti requisiti stilistici:

- Utilizzo di tag HTML in maniera semantica (per esempio, non tutto è un <div>).
- No tag HTML deprecate.
- Evitare variabili JavaScript globali.

-
- Non utilizzare dichiarazioni CSS/JS in-line, nel front-end.
 - In JavaScript (nel front-end), modificare/aggiungere, togliere classi CSS agendo su `classList` e non manipolando direttamente `style`, quando possibile
 - I metodi HTTP, se e quando utilizzati, devono essere usati in maniera appropriata, per es., GET per recuperare informazioni e POST per salvare dati.

Inoltre, l'applicazione web deve essere “usabile” e rispettare i principi fondamentali di visual design.

Requisiti funzionali - descrizione

Si vuole realizzare un'applicazione web per la gestione di **podcast**. In particolare, l'applicazione web deve supportare due tipi di utenti registrati: i creatori e gli ascoltatori.

I **creatori** possono creare, modificare o cancellare una *serie di podcast*, nonché di aggiungere, modificare e cancellare i singoli *episodi* di una loro serie.

Una serie di podcast è composta da un *titolo*, una breve *descrizione*, una *categoria*, un'*immagine*, il nome dell'*autore* (il creatore) e una eventuale *lista degli episodi* in essa contenuti, ordinati per *data*. Alla creazione della serie, tutti i campi eccetto la lista degli episodi, sono *obbligatorie*.

Gli episodi sono rappresentati da un *file audio*¹, da un *titolo*, da una *descrizione* testuale, da una *data* e da un eventuale *sponsor*. I singoli episodi possono essere *gratuiti* oppure a *pagamento*. Tutti i campi, eccetto lo sponsor, sono *obbligatorie* alla creazione di un episodio.

Un episodio può appartenere a una sola serie e solo il creatore della serie può modificarla (per esempio, aggiungendo un episodio).

Prima di poter agire da creatore, un utente del sito deve registrarsi ed eseguire il login.

Utilizzando l'applicazione web, gli **ascoltatori** possono *ascoltare* uno o più episodi, *seguire* (o smettere di seguire) una o più serie di podcast e *salvare* uno o più episodi come “preferiti” (o rimuoverlo dai preferiti). Inoltre, gli ascoltatori possono lasciare un *commento* testuale sotto ogni episodio. Il commento sarà visibile a tutti, ma modificabile e cancellabile solo dall'ascoltatore che l'ha scritto. Nel caso di episodi a pagamento, l'applicazione web chiederà all'ascoltatore i dati della sua *carta di credito* (nome, cognome, tipo, numero, CCV) e verificherà che i dati siano nel formato corretto, al posto di procedere alla riproduzione. Una volta acquistato, l'episodio sarà disponibile al suo acquirente per sempre (fino all'eventuale cancellazione dell'episodio da parte del suo creatore).

Prima di poter agire da ascoltatore, un utente del sito deve registrarsi ed eseguire il login. Un creatore è anche un ascoltatore, ma non viceversa (non è necessario gestire un eventuale

¹ Il caricamento di file audio può essere considerato un *extra*. Ai fini del progetto, è perfettamente accettabile che i file audio siano già disponibili nel server o siano recuperati dal Web, senza penalizzazione alcuna.

“upgrade” da utente ascoltatore a creatore, in questo momento). Sia ascoltatore che creatore avranno una pagina profilo con le informazioni utili.

Un **utente non registrato** può, al di là di registrarsi come creatore o ascoltatore, navigare nell'intero sito liberamente e per categoria.

Inoltre, può *cercare* una serie o un episodio con una ricerca testuale (nei titoli e nelle descrizioni). La ricerca testuale potrà essere *raffinata* per categoria o scegliendo tra episodi/serie. Per esempio, un utente potrà cercare “JavaScript” tra i soli titoli e descrizioni degli episodi nella categoria “Tecnologia”.

Questa funzionalità è ovviamente offerta a tutti gli utenti del sito, anche a quelli non registrati.

Nota bene

Alla consegna, l'applicazione web dovrà contenere almeno un utente “creatore” e almeno un utente “ascoltatore” nonché alcuni (3-4) podcast di esempio (serie ed episodi). I nomi utente e le password di tali utenti devono essere forniti al docente (vedi dopo).

ESAME DA 9 CREDITI

In aggiunta ai requisiti di cui sopra, gli studenti che devono sostenere l'esame da 9 crediti devono:

- Realizzare il front-end dell'applicazione web come una Single Page Application (SPA), utilizzando un framework a loro scelta (es., React, Vue.js, Angular)

Puntatori e suggerimenti verranno forniti direttamente agli studenti interessati. Si raccomanda agli studenti interessati di coordinarsi nella scelta di un framework e di supportarsi nell'apprendimento.

CONSEGNE INTERMEDIE (facoltative)

Al di là della consegna finale del progetto, gli studenti possono seguire due consegne intermedie. Tali consegne non saranno valutate, ma genereranno un feedback utile per meglio completare il progetto. L'applicazione web sarà valutata unicamente in fase d'esame.

Tali consegne intermedie sono:

1. **24 Maggio:** prima versione del front-end
2. **7 Giugno:** database e back-end (quasi completo)

Per le consegne intermedie, caricare un file “nome_cognome_1.zip” oppure “nome_cognome_2.zip”:

- per la prima consegna, <https://www.dropbox.com/request/CuRXoYxcasA9UsCryo5y>
- per la seconda consegna, <https://www.dropbox.com/request/tOBtOjkoGEYHsWpjPOn3>

Il feedback sarà dato direttamente agli studenti interessati (su Slack) nella settimana seguente la consegna.

ISTRUZIONI PER LA CONSEGNA DEL PROGETTO

La consegna del progetto deve avvenire tramite il link “[Consegna progetto finale](#)” nella sezione *Esame* della pagina del corso su DIR, entro 3 giorni prima dell'appello (date precise nella pagina del corso).

La consegna dovrà contenere in un unico archivio (.zip):

- Codice sorgente dell'applicazione web realizzata, incluse:
 - eventuali dipendenze, immagini, ...
 - dump del database utilizzato (preferibilmente in SQLite)
- Link a un video (caricato su YouTube, durata 3-5 minuti). Il video può essere realizzato a partire da una presentazione PowerPoint (o simili) e deve contenere:
 - le scelte di progettazione effettuate, in termini di layout, componenti utilizzati, struttura del DB, ecc.
 - le scelte implementative e funzionali adottate
- Un documento di testo (.md o .txt) contenente eventuali istruzioni per provare l'applicazione web (ad esempio, le credenziali degli utenti di prova) e il link al video

Ogni caricamento fatto dopo la data di consegna non sarà considerato.