

Anno accademico 2024/2025

Progetto d'esame

MF0438 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 **lavorativi** prima dell'appello. La dimostrazione e la breve discussione orale inerente al progetto è da svolgersi il giorno dell'appello, **previa registrazione su ESSE3**.

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. Non è consentito l'uso di altri strumenti/tecniche alternative se non concordati preventivamente con i docenti. Il tema dell'applicazione web da realizzare è a scelta dello studente: il tema scelto dovrà rispettare i vincoli indicati in questo documento, e andrà preventivamente concordato con i docenti del corso. Una volta fissato il tema, i requisiti non cambieranno e saranno validi fino alla sessione di Febbraio 2026 (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 di seguito in questo documento e comunque entro 3 giorni lavorativi prima della discussione.
- 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 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, page per la gestione di pagine multiple) oppure con pagine web generate dinamicamente dal server (utilizzando un sistema di templating appropriato), a scelta degli studenti.
- 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 (133+) e Firefox (135+).
- 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.
- È espressamente richiesto di utilizzare le librerie e tecnologie viste all'interno del corso (page.js per routing lato client, EJS per templating lato server, passport per autenticazione). Scelte alternative devono essere discusse e approvate dal docente.

Extra:

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

- Fino a 2 punti aggiuntivi possono essere assegnati per l'utilizzo di librerie/funzionalità aggiuntive (e non alternative) rispetto a quanto visto nei laboratori/slide del corso, o integrazione con API esterne. Eventuali integrazioni devono essere comunicate e concordate con i docenti.

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 deprecati.
- 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 - vincoli

Il tema e lo scopo dell'applicazione web realizzata è a scelta dello studente. L'applicazione deve implementare almeno le seguenti funzionalità:

- Creazione di utenti: l'applicazione deve supportare **almeno un tipo di utente**, con *registrazione* e *login*, e **preferibilmente almeno due tipi diversi di utenti** (per il pieno punteggio);
- Funzionalità: l'applicazione deve prevedere sia funzionalità aperte a tutti, senza registrazione, sia **funzionalità accessibili solamente agli utenti registrati** (laddove funzione all'applicazione, in alternativa è possibile avere funzionalità accessibili solo ad alcuni tipi di utenti). L'applicazione deve tenere conto di quale utente è correntemente loggato: a titolo di esempio, un'applicazione per la *gestione di contenuti* web non può permettere la modifica dei contenuti stessi ad utenti diversi dai *creatori*;
- Funzionalità: l'applicazione deve prevedere **funzionalità di ricerca**, testuale, per keyword, geolocalizzazione, o altra modalità a scelta dello studente, funzionale all'applicazione stessa;
- Gestione dati: l'applicazione deve prevedere **l'utilizzo di un database** per l'inserimento, cancellazione e ricerca di contenuti; non è accettabile un'applicazione che utilizzi un database unicamente per memorizzare le credenziali di accesso degli utenti.

Scadenze

Non rispettare le scadenze delle seguenti consegne può causerà una penalizzazione fino ad un massimo di 3 punti sul voto finale.

- Le specifiche devono essere comunicate al docente compilando il form disponibile al link <https://forms.gle/stPYTk4gDMyJunj68> entro e non oltre l'08/04/2025.
- L'interfaccia grafica realizzata in HTML e CSS deve essere consegnata tramite caricamento sulla cartella condivisa GDrive (https://drive.google.com/drive/folders/15E_gHDvSU1OG0ftU8IdOq5QUdCSc5Yw?usp=sharing) entro e non oltre il 06/05/2025.
 - Per questa consegna, ogni studente dovrà caricare una cartella nominata usando il seguente formato: Cognome-Nome-matricola (es., Rossi-Mario-111234). La cartella deve contenere i file .html, .css e le risorse necessari (es., immagini, video) per una corretta visualizzazione delle pagine.
- L'elaborato finale deve essere consegnato entro i 3 giorni lavorativi precedenti alla data dell'esame tramite caricamento su DIR.

Nota bene

Alla consegna, l'applicazione web dovrà contenere almeno un utente e un database popolato da dati, anche fintizi, che consentano di testare tutte le funzionalità implementate. I nomi utente e le password di tali utenti devono essere forniti al docente (vedi dopo).

ISTRUZIONI PER LA CONSEGNA DEL PROGETTO

La consegna del progetto deve avvenire tramite il link “[Consegna progetto](#)” nella sezione *Esame* della pagina del corso su DIR, entro 3 giorni lavorativi 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 installare il progetto (es., step per installare librerie aggiuntive e/o requisiti del progetto) e provare l'applicazione web (es., le credenziali degli utenti di prova) e il link al video.

Una volta effettuato il caricamento accertarsi che il materiale sia stata caricato correttamente e che non si trovi più salvato come bozza. Ogni caricamento fatto dopo la data di consegna non sarà considerato.

VALUTAZIONE DEL PROGETTO

La valutazione del progetto terrà conto:

- del rispetto dei requisiti tecnici e stilistici indicati
- della correttezza dell'implementazione (assenza di bug, crash, comportamenti inattesi)
- dell'appropriatezza delle scelte progettuali effettuate
- dell'accuratezza dell'implementazione (accuratezza dei controlli, personalizzazione del front end, usabilità, etc.)
- della complessità tecnica dei requisiti scelti

La valutazione del progetto sarà espressa in 30/30. Saranno previsti fino a 3 punti aggiuntivi che tengono conto della complessità dell'applicazione implementata e dell'accuratezza dell'implementazione sia dal punto di vista funzionale che tecnico/stilistico

ESEMPIO DI PROGETTO

Nota bene: Quello che segue è un eSEMPIO di progetto che funge da traccia per l'elaborazione dei propri requisiti; non è da intendersi come un suggerimento, e pertanto l'applicazione realizzata dovrà essere diversa.

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 *obbligatori*.

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 *obbligatori* 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 "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.

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.