

# 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 circa **3** giorni prima dell'appello (saranno pubblicate sul corso dir appositi contenitori).

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. Il tema dell'applicazione web da realizzare è a scelta dello e della studente: il tema scelto dovrà rispettare i vincoli indicati in questo documento. **Inoltre, è fondamentale che lo o la studente comunichi il tema scelto prima di iniziare a svolgere il progetto: questo consentirà al docente di inviare una versione personalizzata delle specifiche, che lo o la studente dovrà realizzare.**

Il tema del vostro sito dovrete comunicarlo tramite dir (ci sarà un sondaggio apposito nella **sezione 'Progetto finale'**) **PRIMA** di iniziare a sviluppare il vostro progetto. Qualsiasi richiesta fatta a ridosso dell'esame in questo senso, non sarà presa in considerazione.

I requisiti descritti di seguito non cambieranno e saranno validi fino alla sessione di Febbraio 2024 (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 o un'altra studente del corso.
- Il progetto deve essere consegnato secondo le tempistiche riportate nell'ultima pagina di questo documento e comunque entro 3 giorni 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 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, in base alle specifiche personalizzate assegnate dal docente, la o lo studente dovrà adeguarsi e pertanto:

- Realizzare l'applicazione web 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)
- Realizzare un'applicazione web con un target di dispositivi ben preciso, eventualmente supportando la modalità responsive: mobile-only/first vs. desktop-only/first.
- Il progetto consegnato deve essere interamente testabile dal docente e deve funzionare (almeno) sulle ultime versioni di Chrome (108+) e Firefox (113+).
- Il codice sorgente deve essere ben scritto e corredato di opportuni commenti laddove necessario. L'adozione di standard di documentazione JSDoc e/o swagger YAML per la descrizione delle API è auspicabile
- 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, Bootstrap per la grafica). Scelte alternative devono essere discusse e approvate dal docente.
- In deroga al punto precedente, alcune librerie/middleware specifici possono essere richiesti dal docente come **personalizzazione** del vostro progetto. Ad esempio, invece di Page.js il docente potrebbe chiedervi di usare Navigo per la gestione del routing client side; oppure potrebbe essere necessario implementare l'interazione con il database con strumenti alternativi all'accesso diretto, come knex o sequelize, etc.

## 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 (metodologia REST).

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

## Requisiti funzionali

Funzionalità (in generale):

- Creazione di utenti: l'applicazione deve supportare almeno differenti tipi di utente, con registrazione e login;
- Funzionalità: l'applicazione deve prevedere sia funzionalità aperte a tutti, senza registrazione, sia funzionalità accessibili solamente agli utenti registrati (alcune funzionalità sono 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;
- 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.

Alla consegna, l'applicazione web dovrà contenere almeno un utente e un database popolato da dati, anche fittizi, che consentano di testare tutte le funzionalità implementate.

## Descrizione del progetto

Il tema del sito è **totalmente a scelta della o dello studente**. Un esempio di progetto completo è il **Movie Wishlist** che è stato sviluppato passo passo durante le lezioni, con particolare riferimento alla soluzione fornita a seguito dell'ultimo esercizio **lab08**.

Ovviamente, per quanto sia consentito rifarsi a quel codice, non è consentito copiarne le parti senza essere in grado di modificare consapevolmente il codice. Ricordate che durante l'esame vi potrebbe venir chiesto di modificare buona parte dello stile, o di aggiungere/modificare un'API, di cambiare alcuni route, di aggiungere e gestire variabili di sessione o ancora di applicare una logica di autenticazione alternativa a quella che presenterete. Se mostrate di non sapere dove mettere le mani, difficilmente potrete superare l'esame.

## Istruzioni per la consegna del progetto

La consegna del progetto deve avvenire tramite l'opportuno contenitore “Consegna progetto” nella pagina del corso su DIR, entro la data specificata.

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

- Codice sorgente dell'applicazione web realizzata, incluse:
  - eventuali dipendenze (grazie al file package.json), immagini, ...
  - dump del database utilizzato
- Un documento di testo (txt) contenente tutte le istruzioni per provare l'applicazione web. Ad esempio, le credenziali degli utenti di prova. Nel caso si carichino più archivi deve essere descritto come ricostruire il progetto.

La consegna **NON** deve contenere la cartella **node\_modules** e **package-lock.json** (il docente installerà il tutto grazie ad 'npm install')

# Valutazione del progetto

La valutazione del progetto terrà conto:

- del rispetto dei requisiti tecnici, stilistici e funzionali 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.)
- Il progetto viene valutato in base a **completezza e correttezza**
- **È assolutamente necessario che lo e la studente si adegui alle richieste specifiche e personalizzate che verranno richieste dal docente al momento della scelta del tema del progetto.**

Molti di questi aspetti emergeranno durante la discussione orale del progetto. Lo e la studente deve sapere giustificare, soprattutto da un punto di vista tecnico, la motivazione dietro le scelte di design ed implementative.

Infine, non preoccupatevi di eventuali bug o glitch che possono emergere durante la presentazione: sono occasioni per poter dimostrare al docente come sapete usare gli strumenti a disposizione dell sviluppatore per individuare e correggere l'errore. Si consiglia quindi di sviluppare usando sempre il debugger (sia client-side a livello browser che server-side all'interno del vostro IDE) per impararlo al meglio.