

Anno accademico 2022/2023

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 studente: il tema scelto dovrà rispettare i vincoli indicati in questo documento.

I requisiti descritti di seguito non cambieranno e saranno validi fino alla sessione di Febbraio 2023 (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 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:

- 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 (almeno) sulle ultime versioni di Chrome (108+) e Firefox (113+).
- 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, Bootstrap per la grafica). Scelte alternative devono essere discusse con il docente e approvate dal docente.

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**. Di seguito, per dare un'idea delle funzionalità richieste, si presenta un'applicazione in particolare (per il crowdfunding) a **scopo esclusivamente illustrativo**. Ad esempio, si supponga di voler realizzare un'applicazione web

per la gestione di crowdfunding. In particolare, l'applicazione web deve supportare due tipi di utenti registrati: i creatori e i finanziatori. I creatori possono creare, modificare o cancellare un progetto, nonché di aggiungere, modificare singoli documenti. Un progetto è composto da un titolo, una breve descrizione, una categoria, un'immagine, il nome dell'autore (il creatore) e una eventuale lista dei suoi documenti, ordinati per data. Alla creazione del progetto, tutti i campi eccetto la lista dei documenti, sono obbligatori. I documenti sono rappresentati da un titolo, da una descrizione testuale, da una data. I singoli documenti possono essere gratuiti oppure a pagamento. Tutti i campi sono obbligatori alla creazione di un documento. Un documento può appartenere a un solo progetto e solo il creatore del progetto può modificarlo (per esempio, aggiungendo un documento). Prima di poter agire da creatore, un utente del sito deve registrarsi ed eseguire il login. Utilizzando l'applicazione web, i finanziatori possono visionare uno o più documenti, seguire (o smettere di seguire) una o più progetti e salvare uno o più documenti come "preferiti" (o rimuoverlo dai preferiti). Inoltre, i finanziatori possono lasciare un commento testuale sotto ogni documento. Il commento sarà visibile a tutti, ma modificabile e cancellabile solo dal finanziatore che l'ha scritto. Nel caso di documenti a pagamento, l'applicazione web chiederà i dati della sua carta di credito (nome, cognome, tipo, numero, CCV) e verificherà che i dati siano nel formato corretto. Una volta acquistato, il documento sarà disponibile al suo acquirente per sempre (fino all'eventuale cancellazione da parte del suo creatore). Prima di poter agire da finanziatore, un utente del sito deve registrarsi ed eseguire il login. Un creatore è anche un finanziatore, ma non viceversa (non è necessario gestire un eventuale "upgrade"). Sia ascoltatore che creatore avranno una pagina profilo con le informazioni utili. Un utente non registrato può, registrarsi come creatore o finanziatore, navigare nell'intero sito liberamente e per categoria. Inoltre, può cercare un progetto o un documento con una ricerca testuale (nei titoli e nelle descrizioni). Si dovrà fornire una funzionalità di ricerca avanzata dove la ricerca potrà essere raffinata per categoria e/o scegliendo tra progetti/documenti. Per esempio, un utente potrà cercare "JavaScript" tra i soli titoli e descrizioni dei progetti nella categoria "Tecnologia". Questa funzionalità è ovviamente offerta a tutti gli utenti del sito, anche a quelli non registrati.

I finanziatori posso fare donazioni a un progetto (gestita in modo simile all'acquisto di un documento). Deve quindi risultare per ogni progetto la lista dei donatori e il totale delle donazioni (tale informazioni riguardano solo le donazioni e non gli acquisti di documenti).

Si pensi che molte applicazioni Web possono seguire uno schema simile, eventualmente con funzionalità diverse, ma con un uguale livello di completezza. Altri possibili esempi possono essere: una piattaforma di social media semplificata che consente la creazione, la condivisione e la ricerca di contenuti utente, un sito di commercio elettronico con tanto di carrello della spesa, e così via.

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, 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.

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

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.