

Modalità Esame Web Applications

(A.A. 2023/24)

NB: Se sei dai due anni in su fuori corso hai due possibilità:

1. Seguire la modalità di quest'anno che prevede l'uso di Angular;
2. Seguire la modalità degli anni precedenti;

Scorri il documento per tutti i dettagli su entrambe le modalità

Condizioni	
<p>Svolgimento: Visione e discussione progetto (domande e/o modifiche al codice) + domande generali sugli argomenti del corso.</p> <p>Due possibili modalità:</p> <ul style="list-style-type: none">• Esame congiunto con Ingegneria del Software.• Esame indipendente dall'altro corso. <p>Per gli appelli di Gennaio/Febbraio è obbligatorio che il numero dei componenti del gruppo sia uguale a 4.</p> <p>Nel caso in cui per varie necessità (giustificate) sarete meno di 4 o addirittura più di 4, concordare con il docente la modalità di svolgimento. Si noti che un numero minore di 4 non significa traccia più semplice.</p> <p>Per tutti gli appelli di recupero di Giugno/Luglio/Settembre e Sessioni Straordinarie non è più obbligatorio essere 4 persone ma essendo che avrete più tempo a disposizione l'orale potrebbe avere una durata maggiore.</p> <p>La traccia sarà la stessa dell'altro corso nel caso di congiunzione con Ingegneria del software, oppure può essere proposta dagli studenti: in ogni caso la traccia dovrà comunque essere discussa e validata con il docente.</p> <p>In alternativa è possibile svolgere una la traccia proposta dal docente che trovate più avanti in questo documento.</p>	
Requisiti tecnici principali (studenti in corso o primo anno fuori corso)	
<ul style="list-style-type: none">• Lato client:<ul style="list-style-type: none">• Implementazione della parte frontend con Angular.• Oltre ad Angular si richiede l'implementazione di qualche funzione Javascript classica, come ad esempio la validazione di un form.• Lato server:<ul style="list-style-type: none">• Implementazione di un paio di pagine tramite il modello MVC con Thymeleaf per le viste: questo ad esempio per mostrare i dati di una tabella del DB in forma tabellare. Questa parte deve essere implementata tramite Servlet classiche• Implementazione del pattern DAO per l'accesso al database Postgres. NB: è richiesto l'utilizzo di almeno un Proxy, motivando la scelta. Le relazioni del modello dei dati in Java devono essere oggetti e non chiavi esterne.• Implementazione dei @RestController che gestiranno le richieste provenienti dal frontend e quindi dalla parte Angular. E' fortemente consigliato l'uso di JSON per il trasferimento dei dati.	

<ul style="list-style-type: none"> • L'applicazione deve offrire una interfaccia grafica responsive (per esempio attraverso il framework Bootstrap o simili oppure fatto a mano tramite CSS). Anche se si utilizza Bootstrap è necessario comunque prevedere qualche CSS personalizzato. • L'applicazione deve integrare qualche libreria esterna come grafici o API esterne quali ad esempio Google Maps, Youtube, Twitter, API per Film, Flickr, ecc.. <p>NB: è obbligatorio utilizzare le tecnologie viste a lezione. Sono accettate tecnologie aggiuntive ma queste non devono sostituire quelle viste a lezione. Si noti che l'aggiunta di ulteriori tecnologie non viste a lezione non contribuiranno a migliorare il voto</p>	
Requisiti tecnici principali (studenti a partire dal secondo anno fuori corso)	
<ul style="list-style-type: none"> • Lato client l'applicazione utilizza HTML5, CSS, e Javascript/JQuery. • Lato server l'applicazione utilizza Java Servlet+JSP in modalita' MVC, e JDBC per il database utilizzando il pattern DAO. Si consiglia l'utilizzo di Spring Boot. • Per il database è necessario utilizzare Postgres • L'applicazione dovrà utilizzare il più possibile Ajax per la comunicazione di dati e updates tra client e server. E' fortemente consigliato l'uso di JSON per il trasferimento dei dati. • L'applicazione deve offrire una interfaccia grafica responsive (per esempio attraverso il framework Bootstrap o simili oppure fatto a mano tramite CSS). Anche se si utilizza Bootstrap è necessario comunque prevedere qualche CSS personalizzato. • L'applicazione deve integrare contenuto "interessante" tramite almeno una API esterna (per esempio Google Maps, Youtube, Twitter, API per Film, Flickr, ecc.) da invocare direttamente dalle pagine html. Ad esempio, non basta integrare in modo statico una mappa di Google che mostra un indirizzo preconfigurato. 	
Traccia proposta dagli studenti	
<p>Per proporre una traccia vostra è sufficiente che compilate il form alla pagina indicando anche i componenti del gruppo</p> <p style="text-align: center;">https://forms.gle/XBXP9nBWhRSkwm7</p> <p>Fornite, nella descrizione della traccia, i requisiti in maniera sintetica indicando comunque tutte le funzionalità che intendete inserire. Per la descrizione prendete come modello la traccia proposta dal docente (si veda parte successiva). Il docente visionerà le vostre proposte e vi inoltrerà un'email indicandovi se la proposta è stata accettata: da quel momento potrete cominciare a sviluppare il progetto. Qualora la traccia richieda delle integrazioni verrà comunicato per email. Qualora la proposta è stata già fatta sull'altro modulo, incollare nel campo della traccia solamente l'email indicata nell'altro modulo in maniera tale da poter risalire alla traccia</p>	
Traccia proposta dal docente (qualora non vogliate proporre una voi)	
<p>Si realizzi una applicazione web per annunci di affitto/vendita di immobili. Gli immobili sono suddivisi per categorie: appartamenti, box auto, terreni edificabili, ville, ecc...</p> <p>Sono previste 3 tipologie di utente: amministratore, venditori, acquirenti.</p> <p>Di seguito le funzionalità previste.</p> <p>Funzionalità comuni:</p> <ul style="list-style-type: none"> • Esiste già un utente amministratore nel db • Tutti gli utenti possono registrarsi 	

<ul style="list-style-type: none"> ● In un annuncio sono presenti: foto, descrizione, prezzo, metri quadri, posizione Google Maps, recensioni da parte di possibili compratori che hanno visitato l'immobile <p>Venditori:</p> <ul style="list-style-type: none"> ● Possono creare, modificare, cancellare annunci ● Dato un annuncio, i venditori possono ribassare il prezzo: in tal caso il vecchio prezzo sarà barrato e verrà mostrato il nuovo prezzo ● Possibilità di essere contattati: si aprirà un modulo di contatto avente come oggetto il codice e il titolo dell'annuncio ● E' possibile creare delle aste per gli annunci ● Il venditore può promuovere i propri annunci in una pagina Facebook tramite API <p>Acquirenti:</p> <ul style="list-style-type: none"> ● Possono effettuare una ricerca di immobili può avvenire filtrando per: <ul style="list-style-type: none"> ○ Acquisto/Vendita ○ Tipologia di immobile: appartamento, villa, ecc... ● È possibile ordinare gli immobili per prezzo e/o metri quadri ● Se l'acquirente ha visitato l'immobile può scrivere recensioni su di esso ● Possono contattare il venditore direttamente dalla pagina compilando un form <p>Amministratore:</p> <ul style="list-style-type: none"> ● Può modificare/cancellare tutto ● Può bannare utenti ● Può nominare altri utenti come amministratori 	
Modalità di Valutazione (studenti in corso e primo anno fuori corso)	
<p>Per ambire al voto massimo del progetto, è richiesto utilizzare le seguenti metodologie mostrate durante le lezioni/esercitazioni:</p> <ul style="list-style-type: none"> ● Utilità del sito e presenza di funzionalità interessanti (max 4 punti) ● Sito Responsive (max 3 punti) ● HTML/CSS, Javascript, Typescript, Angular e REST Controller (max 9 punti) ● Metodologia MVC con Servlet/Spring Boot e Thymeleaf (max 4 punti) ● Pattern DAO con l'utilizzo di almeno un Proxy e relazioni di Join tramite oggetti (max 7 punti) ● Integrazione di tecnologie esterne: grafici, chatbot, API esterne o altro (max 3 punti) <p>Il progetto avrà un voto massimo di 30 e sarà il voto di partenza di ogni singolo componente del gruppo. In base agli interventi fatti dai singoli componenti del gruppo (sia durante la demo, che in seguito a domande fatte dal docente), il voto di ogni singolo componente potrà salire o scendere.</p>	
Modalità di Valutazione (studenti a partire dal secondo anno fuori corso)	
<p>Per ambire al voto massimo del progetto, è richiesto utilizzare le seguenti metodologie mostrate durante le lezioni/esercitazioni:</p> <ul style="list-style-type: none"> ● Utilità del sito e presenza di funzionalità interessanti (max 4 punti) ● Utilizzo di HTML5 (max 3 punti) ● Sito Responsive (max 3 punti) ● Javascript/JQuery (max 3 punti) ● Metodologia MVC (Servlet o Spring Boot) e JSP/JSTL (max 4 punti) ● Pattern DAO (max 3 punti) ● Ajax e REST Controller (max 4 punti) ● Utilizzo di API Esterne (max 3 punti) <p>Il voto massimo del progetto è di 27 secondo la scaletta di cui sopra e sarà il voto di partenza di ogni singolo studente del gruppo. Verranno poste domande ai singoli studenti, sia durante la discussione del progetto, sia dopo la discussione; le domande singole possono integrare il voto di al massimo 3+ punti (arrivando così eventualmente al 30/30L), ma anche peggiorarlo se non si dimostra di avere acquisito le competenze.</p>	

Termini e modalità di consegna	
<p>Oltre a registrarsi su ESSE3, occorre compilare un modulo (che sarà presto disponibile) con i nominativi del gruppo, la descrizione del progetto proposto e già validato dal docente, e l'appello in cui si prevede di discutere (la scelta dell'appello è facoltativa e serve solo per organizzarmi con i tempi). Qualora si intenda svolgere una delle tracce proposte dal docente è sufficiente incollare nella descrizione la stessa traccia.</p> <p>Il progetto dovrà essere consegnato via GitHub (è possibile creare un proprio repository e condividere il link oppure chiedere ai docenti un nuovo repository) ENTRO 3 giorni prima lo svolgimento dell'appello. Nel repository GitHub dovrà essere inserito, oltre ai sorgenti dell'intero progetto, anche il dump del database Postgres. Occorre quindi mandare una mail a kristian.reale@unical.it con il link al repository git.</p> <p>NB: prima di consegnare il progetto, assicurarsi che il tutto funzioni. Si consiglia quindi di passare ad un altro computer e provare a fare la pull, fare il restore del dump e lanciare il progetto.</p> <p>Il progetto può comunque essere completato fino al giorno dell'esame e discusso poi dal vivo. Il codice consegnato deve contenere tutto il necessario per essere eseguibile, ed il dump del database dovrà essere incluso in tale consegna. Includere un Readme file con le istruzioni ed eventuali note.</p> <p>In aggiunta, nella stessa email, occorre fornire un numero non superiore a 10 slides powerpoint (o pagine google docs) che descriva SOMMARIAMENTE l'applicazione realizzata, meglio usando UML class diagrams per le classi principali, e specifichi quale dei casi d'uso sono supportati e non, eventuali bugs, lo schema delle entità e del database.</p> <p>Sottolineate le funzionalità' aggiuntive implementate ed in generale ciò che pensate sia la parte migliore dell'applicazione, i framework usati, ecc.</p>	
Modalità di svolgimento della prova orale	
<p>Alla singola prova orale parteciperà l'intero gruppo. Uno dei componenti del gruppo (a scelta vostra) sarà il relatore che farà una demo dell'applicazione realizzata. In particolare, il relatore dovrà descrivere cosa fa l'applicazione interagendo con l'applicazione stessa tramite browser Chrome. Durante la demo non è richiesto spiegare il codice in quanto sarà già stato visionato in seguito alla pre-consegna. Ogni componente del gruppo potrà intervenire durante la demo ove necessario. Terminata la demo ogni componente del gruppo spiegherà brevemente di quale aspetto del progetto si è occupato. Le domande ai singoli componenti del gruppo possono avvenire sia durante la demo che subito dopo.</p>	