

# Valido per il 2021/22

<b>Condizioni</b>	
<p><b>Valido solo gli appelli di Gen/Sett 2022</b></p> <p><b>Svolgimento:</b> Visione progetto (domande e/o modifiche al codice) + domande generali sugli argomenti del corso.</p> <p><b>Studenti dal 2015/16 in poi:</b> esame congiunto con Ingegneria del Software, oppure gruppi di max 4 studenti o individualmente. Traccia proposta dagli studenti e discussa con il docente oppure svolgimento traccia proposta dal docente</p> <p><b>NB:</b> nel caso in cui il gruppo sia composto da 4 persone, è necessario dividere il lavoro in due parti: una parte verrà svolta da 2 persone, mentre l'altra parte dalle altre 2, fermo restando che il progetto sarà unico e di conseguenza ci sarà un momento in cui le due parti dovranno essere integrate. Ad esempio, si potrebbe prevedere che una tipologia di utente possa svolgere alcuni compiti dentro la web app (es. compiti di amministrazione del sito) mentre un'altra tipologia svolga altri compiti.</p>	
<b>Requisiti tecnici principali</b>	
<ul style="list-style-type: none"><li>• Lato client l'applicazione utilizza HTML5, CSS, e Javascript/JQuery.</li><li>• 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.</li><li>• Per il database è necessario utilizzare Postgres</li><li>• 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.</li><li>• 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.</li><li>• L'applicazione deve integrare contenuto "interessante" tramite <b>almeno</b> 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.</li></ul> <p><b>NB:</b> fermo restando che il progetto deve contenere le tecnologie di base viste a lezione, è possibile utilizzare ulteriori framework (Angular, React, JPA, ...). Si noti che queste sono scelte tecnologiche personali e la presenza di questi framework <b>non contribuirà a migliorare il voto</b></p>	
<b>Traccia proposta dagli studenti</b>	
<p>Per proporre una traccia vostra è sufficiente che compilate il form alla pagina</p> <p><a href="https://forms.gle/FWBj7TQdCdXM8XKNA">https://forms.gle/FWBj7TQdCdXM8XKNA</a></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</p>	

<b>Traccia proposta dal docente (qualora non vogliate proporre una voi)</b>	
<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"> <li>• Esiste già un utente amministratore nel db</li> <li>• Tutti gli utenti possono registrarsi</li> <li>• In un annuncio sono presenti: foto, descrizione, prezzo, metri quadri, posizione Google Maps, recensioni da parte di possibili compratori che hanno visitato l'immobile</li> </ul> <p>Venditori:</p> <ul style="list-style-type: none"> <li>• Possono creare, modificare, cancellare annunci</li> <li>• Dato un annuncio, i venditori possono ribassare il prezzo: in tal caso il vecchio prezzo sarà barrato e verrà mostrato il nuovo prezzo</li> <li>• Possibilità di essere contattati: si aprirà un modulo di contatto avente come oggetto il codice e il titolo dell'annuncio</li> <li>• E' possibile creare delle aste per gli annunci</li> <li>• Il venditore può promuovere i propri annunci in una pagina Facebook tramite API</li> </ul> <p>Acquirenti:</p> <ul style="list-style-type: none"> <li>• Possono effettuare una ricerca di immobili può avvenire filtrando per: <ul style="list-style-type: none"> <li>◦ Acquisto/Vendita</li> <li>◦ Tipologia di immobile: appartamento, villa, ecc...</li> </ul> </li> <li>• È possibile ordinare gli immobili per prezzo e/o metri quadri</li> <li>• Se l'acquirente ha visitato l'immobile può scrivere recensioni su di esso</li> <li>• Possono contattare il venditore direttamente dalla pagina compilando un form</li> </ul> <p>Amministratore:</p> <ul style="list-style-type: none"> <li>• Può modificare/cancellare tutto</li> <li>• Può bannare utenti</li> <li>• Può nominare altri utenti come amministratori</li> </ul>	
<b>Modalità di Valutazione</b>	
<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"> <li>• Utilità del sito e presenza di funzionalità interessanti (max 4 punti)</li> <li>• Utilizzo di HTML5 (max 3 punti)</li> <li>• Sito Responsive (max 3 punti)</li> <li>• Javascript/JQuery (max 3 punti)</li> <li>• Metodologia MVC (Servlet o Spring Boot) e JSP/JSTL (max 4 punti)</li> <li>• Pattern DAO (max 3 punti)</li> <li>• Ajax e REST Controller (max 4 punti)</li> <li>• Utilizzo di API Esterne (max 3 punti)</li> </ul> <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>	
<b>Termini e modalità di consegna</b>	
<p>Oltre a registrarsi su ESSE3, occorre compilare al più presto il seguente foglio</p> <p style="text-align: center;"><a href="https://forms.gle/FWBj7TQdCdXM8XKNA">https://forms.gle/FWBj7TQdCdXM8XKNA</a></p>	

<p>con i nominativi del gruppo, la descrizione del progetto proposto 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 la traccia proposta 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) <b>ENTRO 3 giorni prima lo svolgimento dell'appello</b>. Nel repository GitHub dovrà essere inserito, oltre ai sorgenti dell'intero progetto, anche il dump del database Postgres. Occorre quindi mandare una mail a <a href="mailto:reale@mat.unical.it">reale@mat.unical.it</a> con il link al repository git.</p> <p><b>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.</b></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 <b>SOMMARIAMENTE</b> 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>	
<p><b>Modalità di svolgimento della prova orale</b></p>	
<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>	