

# Progetto d'esame

*Versione FINALE. Le modifiche al testo sono evidenziate in rosso.*

## MODALITÀ D'ESAME

L'esame di Introduzione alle Applicazioni 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 prima dell'appello. La dimostrazione e la breve discussione orale inerente il progetto è da svolgersi secondo il calendario che sarà reso disponibile qualche giorno prima dell'appello, previa registrazione all'appello stesso.

La discussione riguarderà le scelte di progettazione (layout, struttura del codice, 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.

### 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.
- Non è prevista né accettabile una consegna in ritardo.

### Requisiti tecnici

L'applicazione web deve rispettare i seguenti requisiti tecnici:

- 
- Utilizzo di HTML5 e CSS3, avvalendosi se necessario di framework esterni come Bootstrap, ma personalizzandone lo stile tramite regole create ad-hoc.
  - Utilizzo di Flask e di SQLite (come database relazionale), per il back-end.
  - Utilizzo di Flask-Login per la gestione dell'autenticazione.

Inoltre:

- L'applicazione web deve avere un target di dispositivi ben preciso (**desktop, mobile oppure tablet**), a scelta degli studenti, ~~eventualmente supportando la modalità responsive~~.
- Eventuali form utilizzati nel progetto devono essere **validati** sia nel front-end che nel back-end, come spiegato durante il corso, almeno per i dati e le informazioni essenziali.
- Il progetto consegnato deve essere interamente testabile dal docente e deve funzionare sulle ultime versioni di Chrome (111+) e Firefox (110+).
- 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.

#### Extra:

- Fare il deploy dell'applicazione web su PythonAnywhere (<https://www.pythonanywhere.com/>).

### 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.
- Non utilizzare dichiarazioni CSS inline, mantenendo cioè le regole CSS separate dalla struttura semantica del HTML.

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

### Descrizione (requisiti funzionali)

Si vuole creare un'applicazione web per **trovare case da affittare** in una determinata città. Per semplicità, l'applicazione sarà dedicata a una sola città, a scelta degli studenti/esse.

L'applicazione web deve supportare due tipi di utenti registrati: i *clienti* e i *locatori*.

I **clienti** possono esplorare gli annunci relativi alle case attualmente disponibili per l'affitto e, se interessati, prenotare una visita. Prima di poter agire da cliente, l'utente deve registrarsi e fare

---

login. Il login/registrazione richiede un campo univoco che sarà utilizzato per riconoscere l'utente nel sito (per esempio, la mail).

I **locatori**, invece, potranno gestire gli annunci delle proprie case in affitto, anch'essi attraverso login/registrazione. In particolare, i locatori potranno inserire e modificare annunci di abitazioni nonché gestire le prenotazioni di visite per le loro case.

Ogni **annuncio** presente nel sito avrà associate le seguenti informazioni (obbligatorie):

- *Titolo* dell'annuncio
- *Indirizzo* dell'abitazione
- *Tipo* di casa (casa indipendente, appartamento, loft, villa)
- Numero di *locali* (1, 2, ... 5+)
- Breve *descrizione* dell'appartamento in affitto
- *Prezzo* mensile
- Se la casa è *arredata* o meno
- 1-5 *fotografie* dell'appartamento
- Identificativo del *locatore*
- Se è *disponibile* (= visibile a tutti) oppure no (= visibile solo al locatore)

Nella homepage dell'applicazione, i clienti vedranno una *versione breve* degli annunci delle case disponibili per l'affitto, ordinati per prezzo decrescente. Tramite un apposito filtro, i clienti potranno anche ordinare gli annunci per numero di locali (crescenti).

Cliccando su ogni annuncio, i clienti potranno vedere la versione completa dello stesso che riporterà in un layout adeguato tutte le informazioni relative a un annuncio (vedi sopra). Inoltre, i clienti potranno **prenotare** una visita con il locatore tramite un apposito bottone. Al momento della prenotazione di una visita, l'applicazione web chiederà *come si vuole visitare la casa* (di persona o da remoto), una *data di visita* (nei 7 giorni seguenti la data di richiesta prenotazione) e una *fascia oraria* in cui effettuare la visita (9-12, 12-14, 14-17, 17-20). **Se esiste già una prenotazione accettata per quella casa in un dato orario/giorno, la fascia oraria corrispondente nel dato giorno non sarà più richiedibile. Una volta che una prenotazione è stata richiesta, il cliente che ne ha fatto domanda non potrà più chiedere un'altra visita per la stessa casa a meno che la richiesta non venga rifiutata.**

L'elenco delle prenotazioni, con il loro stato (*richiesta, accettata, rifiutata*) e l'eventuale motivazione del rifiuto sarà visibile nella pagina profilo del cliente.

Il locatore, in maniera simile, avrà una pagina profilo in cui saranno mostrati i suoi annunci, divisi in disponibili e non, e l'elenco delle richieste di prenotazione che ha ricevuto, con il loro stato (richiesta, accettata, rifiutata) insieme all'eventuale motivazione del rifiuto. Per le prenotazioni ricevute ma non ancora risposte (stato: richiesta), il locatore potrà accettarle o rifiutarle. In caso di rifiuto, dovrà *obbligatoriamente* specificare il motivo di tale rifiuto in un campo testuale, che sarà visibile anche ai clienti nell'elenco delle prenotazioni che appare nella pagina profilo.

---

I locatori potranno, inoltre, aggiungere annunci di abitazioni (inserendo le informazioni relative a un annuncio descritte sopra) e modificare i propri annunci, con la possibilità di cambiare tutte le informazioni eccetto l'indirizzo dell'abitazione in affitto.

I locatori potranno usare l'applicazione web come i clienti ma non potranno richiedere la prenotazione di una visita di una loro casa. **Un cliente non può diventare un locatore.**

Infine, un **utente non registrato** può, al di là di registrarsi come cliente o locatore, navigare nell'intero sito liberamente (ma non inserire annunci o richiedere prenotazioni di visite).

**Nota bene:** alla consegna, l'applicazione web dovrà contenere almeno 4 utenti registrati al sito (due locatori e due clienti), alcuni (3-4) annunci inseriti, una richiesta di visita accettata, una rifiutata e 2 richieste di visite non ancora accettate o rifiutate. I nomi utente e le password di tali utenti devono essere forniti ai docenti (vedi dopo).

## ISTRUZIONI PER LA CONSEGNA DEL PROGETTO

La consegna del progetto deve avvenire tramite lo strumento “**Consegna Elaborati**” presente nella pagina del corso sul *Portale della Didattica*, entro le **23:59 del 08 Febbraio 2024**.

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

- Codice sorgente dell'applicazione web realizzata, incluse:
  - eventuali dipendenze (in un file come requirements.txt), immagini, ...
  - file SQLite contenente il database
- Un documento di testo (.md o .txt) contenente:
  - le credenziali degli utenti;
  - eventuali istruzioni per provare l'applicazione web;
  - se è stato fatto, l'indirizzo a cui l'applicazione web è visibile dopo il deploy.