

# **Progetto di Tecnologie Web**

**Sito WEB per la gestione di locazioni a studenti**

**Gruppo: 25**

**Referente: Mannini leonardo**



**Corso di Tecnologie Web  
Facoltà di Ingegneria Informatica e dell'Automazione  
Università Politecnica delle Marche  
10 Giugno 2022**

**Sommario**

Riferimenti .....4

Contributo .....5

Descrizione del sito .....6

Organization chart .....8

Soluzioni adottate .....9



# **RIFERIMENTI**

**Gruppo numero:** 25

**Lista dei componenti:**

- Di Placido Antonio
- Djouaka Kelefack Lionel
- Kamdem Defo Patrice
- Mannini Leonardo

**Referente del Gruppo:** Mannini Leonardo

**Github:** <https://github.com/leonardoman9/tecwebproject>

# **CONTRIBUTO**

- Di Placido Antonio : 30 %
- Djouaka Kelefack Lionel : 10 %
- Kamdem Defo Patrice : 10 %
- Mannini Leonardo : 50 %

# Descrizione del sito

Il sito ha la finalità di far incontrare locatori di alloggi con potenziali locatari in cerca di alloggio.

L'area pubblica del sito, nella homepage, permette la visualizzazione di informazioni utili quali: le FAQ, il regolamento e le condizioni d'uso, i contatti, la localizzazione, "Chi siamo", "Servizi", "Mission" e vari recapiti, e la visualizzazione degli ultimi cinque annunci pubblicati dai locatori nel sito.

Inoltre, un utente "ospite" non autenticato può accedere alla sezione Catalogo, con la possibilità di sfogliare l'intero catalogo paginato, ordinato per data di inserimento degli annunci, ma senza la possibilità di effettuare ricerche filtrate.

Nella navbar sono disponibili i pulsanti per la registrazione o login, per poter quindi accedere agli altri livelli di utenza.

Nel livello d'utenza del locatore, la homepage rimarrà uguale a quella pubblica, ma con la possibilità di visualizzare la pagina dei messaggi e la pagina di gestione degli annunci, oltre che alla possibilità di effettuare logout e di visualizzare le informazioni del profilo utente.

La sezione "Messaggi" (uguale a quella del locatario) presenta una prima sezione chiamata "Le mie chat", dove sarà presente una lista di tutti gli utenti con cui l'utente ha scambiato dei messaggi (sia ricevuti, sia inviati), con la possibilità, cliccando sul nome desiderato, di aprire tutto lo storico dei messaggi. Più in basso, le due colonne "Messaggi ricevuti" e "Messaggi inviati" mostrano in ordine cronologico tutti i messaggi scambiati dall'utente loggato.

Nella visualizzazione "Gestione Annunci", il locatario troverà una tabella contenente tutti gli alloggi da lui aggiunti, con elencate tutte le caratteristiche per ogni alloggio, la possibilità di aggiungerne di nuovi, di modificare quelli già esistenti, o eliminarli. Inoltre, tramite il link nell'ultima colonna, il locatore potrà visualizzare tutte le richieste di locazione (opzioni) che i locatari hanno effettuato per quell'alloggio.

Una volta visualizzate quindi le richieste di locazione, quindi, il locatario potrà scegliere a quale utente assegnare l'alloggio, visualizzando le sue caratteristiche (nome e cognome, genere, età, data di effettuazione della richiesta di locazione e date di eventuale locazione).

Il locatario, una volta loggato, si troverà nella navbar il collegamento al Catalogo del sito, come l'utente guest, ma con la possibilità di filtraggio. Infatti, nella colonna a sinistra, sarà presente un form (variabile, a seconda del tipo di alloggio ricercato), dove sarà possibile inserire i parametri ricercati dall'utente. A destra, usciranno quindi i risultati filtrati dal database del sito. Una volta scelto l'alloggio preferito, cliccando su "Vedi l'annuncio" sarà possibile visualizzare i vari dettagli dell'alloggio e un pulsante "Contatta" per inviare un messaggio al locatore dell'alloggio.

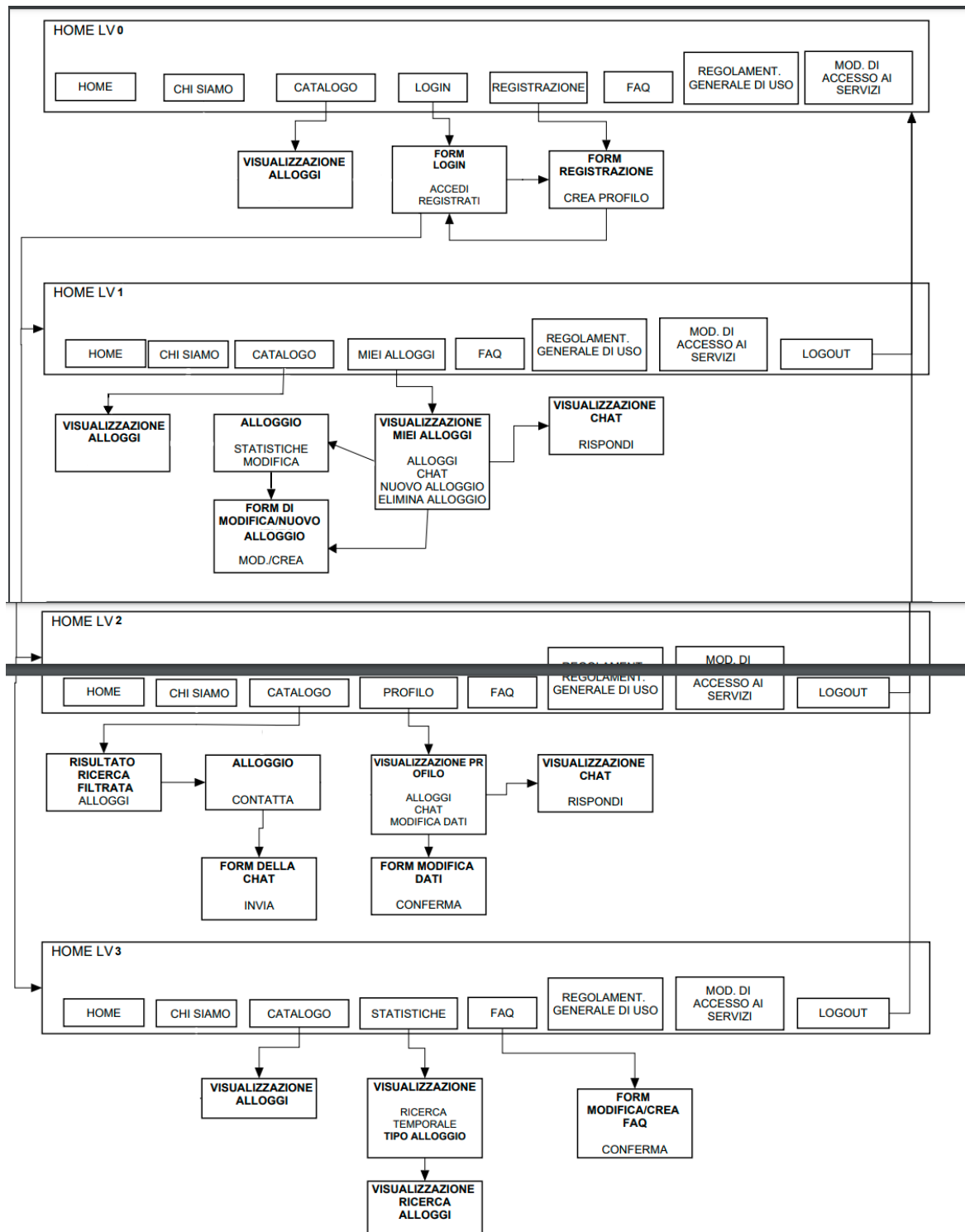
Nella form di invio messaggi del locatario, a differenza del locatore, sarà presente una spunta per scegliere se inviare, oltre al messaggio, anche una richiesta di locazione per l'alloggio, che verrà visualizzata nella precedente descritta ultima colonna della tabella "Gestione annunci" del locatore.

Nella sezione "Richieste di Opzionamento", il locatario potrà visualizzare tutte le richieste fatte per gli alloggi con una flag sullo stato della richiesta ("in pending" o "accettata"), ed eventualmente cancellare la richiesta di opzionamento.

L'admin, una volta loggato, potrà accedere alla pagina "Gestisci FAQ" dalla navbar: da qui, potrà visualizzare, modificare, eliminare od aggiungere nuove FAQ che verranno poi visualizzate nella pagina visualizzabile da ogni livello di utenza.

Inoltre, l'admin potrà accedere alla sezione Statistiche, dove potrà, selezionando un tipo di alloggio e un certo span di date, filtrare il numero di "Offerte di alloggio presenti nel sito", "Offerte di locazione fatte da potenziali locatari" e di "Alloggi locati".

# Organization chart



Schema Dei Link



# Soluzioni adottate

## Strumenti di sviluppo:

- Framework:
  - Laravel 6
- Linguaggi:
  - HTML+CSS
  - JavaScript+jQuery
  - PHP 7.2.11
- Bundle Apache-PHP-MySQL
  - XAMPP
- Ide:
  - Netbeans / Visual Studio
- Tools:
  - FileZilla
  - Putty
- Versioning:
  - GitHub

Di seguito sono riportati alcuni screenshot che illustrano sezioni di codice rilevanti per il funzionamento del sito o particolarmente articolati.

## view/welcome.blade.php

```
<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" type="text/css" href="{{ asset('css/stylehome.css') }}">
<link rel="preconnect" href="https://fonts.gstatic.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@500;700&display=swap" rel="stylesheet">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-free@5.15.4/css/fontawesome.min.css">
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-free@5.15.4/css/all.min.css">

<title>HomeAway | @yield('title', 'Home')</title>

</head>
<body>
<header>
    @include('layouts/_navpublic')
</header>
    @include('layouts/_cover')
    @yield('content')
<footer>
    @include('layouts/_footer')
</footer>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js"></script>

</body>
</html>
```

## WELCOME.BLADE: LO SCHELETRO DI OGNI VIEW

[https://tweb2.dii.univpm.it/phpmyadmin/db\\_structure.php?server=1&db=grp\\_25\\_db](https://tweb2.dii.univpm.it/phpmyadmin/db_structure.php?server=1&db=grp_25_db)

| Tabella          | Azione  | Righe     | Tipo          | Codifica caratteri       | Dimensione     | Overhead   |
|------------------|---|-----------|---------------|--------------------------|----------------|------------|
| alloggios        | Mostra Struttura Cerca Inserisci Svuota Elimina | 13        | InnoDB        | utf8mb4_unicode_ci       | 16 Kib         | -          |
| fotos            | Mostra Struttura Cerca Inserisci Svuota Elimina | 12        | InnoDB        | utf8mb4_unicode_ci       | 16 Kib         | -          |
| f_a_q_s          | Mostra Struttura Cerca Inserisci Svuota Elimina | 3         | InnoDB        | utf8mb4_unicode_ci       | 16 Kib         | -          |
| messaggis        | Mostra Struttura Cerca Inserisci Svuota Elimina | 11        | InnoDB        | utf8mb4_unicode_ci       | 16 Kib         | -          |
| migrations       | Mostra Struttura Cerca Inserisci Svuota Elimina | 7         | InnoDB        | utf8mb4_unicode_ci       | 16 Kib         | -          |
| opzionamenti     | Mostra Struttura Cerca Inserisci Svuota Elimina | 6         | InnoDB        | utf8mb4_unicode_ci       | 16 Kib         | -          |
| servizios        | Mostra Struttura Cerca Inserisci Svuota Elimina | 13        | InnoDB        | utf8mb4_unicode_ci       | 16 Kib         | -          |
| users            | Mostra Struttura Cerca Inserisci Svuota Elimina | 11        | InnoDB        | utf8mb4_unicode_ci       | 48 Kib         | -          |
| <b>8 tabelle</b> | <b>Totali</b>                                   | <b>76</b> | <b>InnoDB</b> | <b>latin1_swedish_ci</b> | <b>160 Kib</b> | <b>0 B</b> |

## DATABASE

routes/web.php

| Method   | URI   | Name                    |
|----------|---|-------------------------|
| GET HEAD | /   | Home                    |
| GET HEAD | api/user  |                         |
| GET HEAD | cancelli_alloggio/{id_alloggio}                             | cancellaAlloggio        |
| GET HEAD | catalogo  | Catalogo                |
| POST     | catalogo  | RicercaCatalogo         |
| GET HEAD | catalogo/{num}  | Ann                     |
| GET HEAD | elimina_faq/{id}  | elimina_faq             |
| GET HEAD | faq   | FaqPage                 |
| GET HEAD | form_faq  | form_faq                |
| GET HEAD | gestionefaq   | gestioneFaq             |
| POST     | gestionefaq   | crea_faq                |
| GET HEAD | gestionefaq/{num}   | modificaFaq             |
| GET HEAD | home  | home                    |
| POST     | login   |                         |
| GET HEAD | login   | login                   |
| POST     | logout  | logout                  |
| GET HEAD | messaggi  | msgPage                 |
| POST     | messaggi  | inviaMessaggio          |
| GET HEAD | messaggi/{ann}  | msgConDest              |
| GET HEAD | messaggio/{annId}   | messaggio               |
| POST     | mieialloggi   | creazioneAlloggio       |
| GET HEAD | mieialloggi   | alloggiLocatore         |
| GET HEAD | mieialloggi/accettata/{opzId}/{id_alloggio}/{id_opzionante} | AccettaRichiesta        |
| GET HEAD | mieialloggi/nuovo   | nuovoAlloggio           |
| GET HEAD | mieialloggi/richieste/{annId}                               | RichiesteLocazione      |
| GET HEAD | mieialloggi/{id}  | visualizzaAlloggio      |
| GET HEAD | modifica_alloggio/{id}                                      | modificaAlloggio        |
| POST     | modifica_alloggio/{id}                                      | updateAlloggio          |
| GET HEAD | modifica_faq/{id}   | modifica_faq            |
| POST     | modifica_faq/{id}   | modifica_faq            |
| GET HEAD | opzionamenti  | opzionamenti            |
| GET HEAD | opzionamenti/cancella/{id}                                  | cancellaOpzionamento    |
| GET HEAD | password/confirm  | password.confirm        |
| POST     | password/confirm  |                         |
| POST     | password/email  | password.email          |
| POST     | password/reset  | password.update         |
| GET HEAD | password/reset  | password.request        |
| GET HEAD | password/reset/{token}                                      | password.reset          |
| GET HEAD | profilo   | profilo                 |
| GET HEAD | profilo/modifica  | mostra_modifica_profilo |
| POST     | profilo/modifica  | modifica_profilo        |
| POST     | register  |                         |
| GET HEAD | register  | register                |
| GET HEAD | regolamento   | RegolamentoPage         |
| GET HEAD | rispondi/{id}   | rispondi                |
| GET HEAD | stats   | statsPage               |
| POST     | stats   | StatFiltrate            |
| GET HEAD | user  | user                    |

TUTTE LE ROUTE DISPONIBILI

```

26  class UserController extends Controller {
28
29
30      protected $_faqs;
31      protected $FAQ;
32      private $alloggio;
33      protected $servizi;
34      protected $opzionamenti;
35
36      public function construct() {...8 lines }
44      public function index() {...3 lines }
47      public function showProfile() {...7 lines }
54      public function showModificaProfilo() {...6 lines }
61      public function modificaProfilo(UpdateProfileRequest $request) {...31 lines }
91      public function gestioneFaq() {...5 lines }
96      public function creaFaq(FaqCreateRequest $request) {...7 lines }
103      public function showModificaFaq($id) {...9 lines }
112      public function modificaFaq(FaqUpdateRequest $request, $id) {...16 lines }
128      public function eliminaFaq($id) {...14 lines }
142      public function showStats() {...11 lines }
149      public function showStatsFilt(StatisticheRequest $request) {...51 lines }
204      public function showMsg() {...21 lines }
225      public function showAnn($ann) {...23 lines }
248      public function showFaqPage($faq) {...6 lines }
254      public function findPosterByAnnId($annId) {...8 lines }
262      public function inviaMessaggio($annId) {...14 lines }
276      public function messInviato(MessaggioRequest $request) {...22 lines }
283      public function getSentMessagesByUserId($sender) {...6 lines }
290      public function getReceivedMessagesByUserId($receiver) {...6 lines }
310      public function showDestMsgPage($dest) {...15 lines }
325      public function risponDiMsg($msg) {...16 lines }
341      public function showOpzionamenti() {...9 lines }
350      public function cancellaOpzionamento($opzId) {...6 lines }
356  }

```

UserController: PER UTENTI LOGGATI

```

public function search(Ricerca $request) {
    $alloggio = new alloggio();
    $foto = new foto();

    $results = $alloggio::query()
        ->where('citta', 'LIKE', "%{$request['citta']}%")
        ->where('tipologia', '=', $request['tipologia'])
        ->whereDate('data_inizio_locazione', '>=', $request['data_inizio_locazione'])
        ->whereDate('data_fine_locazione', '<=', $request['data_fine_locazione'])->get();

    if ($request['dimensionemax'] !=null ){
        $results=$results->where('dimensione', '<=', $request['dimensionemax']);
    }
    if ($request['dimensionemin'] !=null ){
        $results=$results->where('dimensione', '>=', $request['dimensionemin']);
    }
    if ($request['prezzomax'] !=null ){
        $results=$results->where('canone', '<=', $request['prezzomax']);
    }
    if ($request['prezzomin'] !=null ){
        $results=$results->where('canone', '>=', $request['prezzomin']);
    }
    if ($request['numeropostiletto'] !=null){
        $results=$results->where('numero_posto_letto_totale', '=', $request['numeropostiletto']);
    }
    if ($request['lettinellacamera'] !=null){
        $results=$results->where('numero_letti_nella_camera', '=', $request['lettinellacamera']);
    }
    if ($request['numerocamere'] !=null){
        $results=$results->where('numero_camere', '=', $request['numerocamere']);
    }
    return view('catalogo')
        ->with('results', $results)
        ->with('request', $request)
        ->with('allFotos', $foto->returnAllFotos());
}
}

```

## FILTRAGGIO DEL CATALOGO

## Http/Controllers/UserController.php

```
public function showStatsFilt(StatisticheRequest $request){
    switch($request['statistica']){          //a seconda della statistica selezionata
        case 'offerteDiAlloggio':
            //filtro tutti gli alloggi in base al tipo
            switch($request['scelta']){
                case 1: $alloggi = alloggio::where('tipologia', 1)->get(); break;
                case 2: $alloggi = alloggio::where('tipologia', 2)->get(); break;
                case 3: $alloggi = DB::table('alloggios')->get(); break; }
            //filtro tutti gli alloggi in base alla data
            $result = $alloggi->where('data_inserimento', '>=', $request['data_inizio'])
                                ->where('data_inserimento', '<=', $request['data_fine'])
                                ->count();break;
        case 'offerteLocazioneDaLocat':
            $joinedTable = DB::table('opzionamenti')
                                ->join('alloggios', 'opzionamenti.id_alloggio', '=', 'alloggios.id_alloggio')
                                ->where('data', '>=', $request['data_inizio'])
                                ->where('data', '<=', $request['data_fine']) ;
            switch($request['scelta']){
                case 1: $result = $joinedTable->where('tipologia', 1)->get(); break;
                case 2: $result = $joinedTable->where('tipologia', 2)->get(); break;
                case 3: $result = $joinedTable->get(); break;
            }
            $result = $result->count();
            break;
        case 'alloggiLocati':
            $joinedTable = DB::table('opzionamenti')
                                ->join('alloggios', 'opzionamenti.id_alloggio', '=', 'alloggios.id_alloggio')
                                ->where('data', '>=', $request['data_inizio'])
                                ->where('data', '<=', $request['data_fine'])
                                ->where('accettata', '=', 1)
                                ;
            switch($request['scelta']){
                case 1: $result = $joinedTable->where('tipologia', 1)->get(); break;
                case 2: $result = $joinedTable->where('tipologia', 2)->get(); break;
                case 3: $result = $joinedTable->get(); break;
            }
            $result = $result->count(); break;
    } return view('statsPage')
        ->with('result', $result)
        ->with('request', $request);
}
```

## STATISTICHE

## Http/Controllers/UserController.php

```
public function inviaMessaggio($annId){...10 lines }
public function messInviato(MessaggioRequest $request){ //funzione attivata dopo pressione pulsante "invia" nella form del messaggio
    if($request['opziona']==true){ //se il locatario decide di opzionare, verrà creata una nuova opzione
        $opzionamenti = new Opzionamento();
        $opzionamenti->id_alloggio = $request['annId'];
        $opzionamenti->id_proprietario = User::getIdByUsername($request['receiver']);
        $opzionamenti->id_opzionante = Auth::user()->id;
        $opzionamenti->data = Carbon::now()->toDateTimeString();
        $opzionamenti->accettata = false;
        $opzionamenti->save();
    } //inserimento messaggio nel db
    $messaggi = new messaggi();
    $messaggi->mittente = $request['sender'];
    $messaggi->destinatario = $request['receiver'];
    $messaggi->testo = $request['mess'];
    $messaggi->timestamp= Carbon::now()->toDateTimeString();
    $messaggi->id_alloggio = $request['annId'];
    $messaggi->save();
    return $this->showMsg();
} //varie funzioni utili a ottenere messaggi ricevuti e inviati
public function getSentMessagesByUserId($sender){...6 lines }
public function getReceivedMessagesByUserId($receiver){...6 lines }
public function showDestMsgPage($dest){ //mostra la pagina contenente messaggi da/per un particolare utente
    $auth = Auth::user();
    $messaggi = new messaggi();
    $selectedMessaggiA = $messaggi
        ->where('mittente', '=', $auth->username) ->where('destinatario', '=', $dest);
    $selectedMessaggiB = $messaggi //unione fra messaggi ricevuti e inviati
        ->where('mittente', '=', $dest)->where('destinatario', '=', $auth->username)
        ->union($selectedMessaggiA) ->orderBy('timestamp', 'desc') ->get();
    return view('msgPageDest')
        ->with('msgs', $selectedMessaggiB);
}
public function rispondiMsg($msg){ //dato un messaggio, si risponde estraendo il destinatario: se sono il locatario,
    // scrivo al locatore, altrimenti viceversa
    $messaggi = new messaggi();
    $selectedAnn = $messaggi::where('id', $msg)->first();
    if (Auth::user()->username == $selectedAnn->destinatario){
        $destinatario = $selectedAnn->mittente;
    }else { $destinatario = $selectedAnn->destinatario;}
    $madeOpzionamento = Opzionamento::where('id_alloggio', $selectedAnn->id_alloggio)
        ->where('id_opzionante', Auth::user()->id)
        ->first();
    $madeOpzionamento!=null ? $flag=1 : $flag=0;
    return view('messaggio')
```

## VARIE F. RELATIVE AI MESSAGGI

```

20  class PostController extends Controller
21  {
22      protected $_faqs;
23      protected $FAQ;
24      private $alloggio;
25      protected $servizi;
26      protected $opzionamenti;
27
28      public function __construct() {
29          $this->middleware('auth');
30          $this->FAQ = new FAQ;
31          $this->alloggio = new alloggio;
32          $this->servizi = new servizio();
33          $this->opzionamenti = new Opzionamento();
34      }
35      public function inserisciAlloggio() {...3 lines }
36      public function createAlloggio(InserimentoRequest $request) {...48 lines }
37      public function getalloggio() {...4 lines }
38      public function getalloggioById($id_alloggio) {...15 lines }
39
40      public function cancelliAlloggio($id_alloggio) {...17 lines }
41      public function modificaAlloggio($id_alloggio) {...15 lines }
42      public function updateAlloggio(InserimentoRequest $request, $idAlloggio) {...32 lines }
43      public function showRichieste($id_alloggio) {...16 lines }
44
45      public function accettaRichiesta($opzId, $id_alloggio, $id_opzionante) {...17 lines }
46  }

```

## CONTROLLER PER ALLOGGI PER LOCATORE



http/Controllers/PostController.php

```
public function createAlloggio(InserimentoRequest $request){
    $Authenticated = Auth::user();
    $post = new alloggio();
    $serv = new servizio();
    $post->tipologia = $request->tipologia;
    $post->data_inserimento = Carbon::now()->format('Y-m-d');
    $post->canone = $request->canone;
    {$post->dimensione = $request->dimensione;
    $post->citta = $request->citta;
    $post->indirizzo = $request->indirizzo;
    $post->data_inizio_locazione = $request->data_inizio_locazione;
    $post->data_fine_locazione = $request->data_fine_locazione;
    $post->numero_camere = $request->numero_camere;
    $post->numero_posto_letto_totale = $request->numero_posto_letto_totale;
    $post->numero_letti_nella_camera = $request->numero_letti_nella_camera;
    $post->descrizione = $request->descrizione;
    $post->added_by = $Authenticated->username;}
    $post->save();
    $id = DB::table('alloggios')
        ->orderBy('timestamp', 'desc')
        ->first()
        ->id_alloggio;
    $serv->id_alloggio = $id;
    if($request['cucina']!=null){
        $serv->cucina = $request['cucina'];
    } else {$serv->cucina = false;}
    if($request['locRicr']!=null){
        $serv->localeRicreativo = $request['locRicr'];
    } else {$serv->localeRicreativo = false;}
    if($request['angoloStudio']!=null){
        $serv->angoloStudio = $request['angoloStudio'];
    } else {$serv->angoloStudio = false;}
    $serv->save();

    $foto = new foto();
    $file = $request->file('image');
    if($file!=null){
        $path = $file->store('\public\foto');
        $foto->path = $path ;
        $foto->id_alloggio = $post->id;
        $foto->save();
    } $alloggios = alloggio::where('added_by', '=', Auth::user()->username)->get();
    return view('alloggios')
        ->with('alloggios', $alloggios);
}
```

Creazione di un nuovo alloggio, creando nuovi record nelle tabelle 'alloggios', 'fotos, e 'servizios'

Http/Providers/AuthServiceProvider.php

```
public function boot()
{
    $this->registerPolicies();

    Gate::define('isLocatario', function($user){
        return $user->hasRole("locatario");
    });

    Gate::define('isLocatore', function($user){
        return $user->hasRole("locatore");
    });

    Gate::define('isAdmin', function($user){
        return $user->hasRole("admin");
    });

    Gate::define('isLocatarioOrLocatore', function($user){
        return ($user->hasRole("locatario") || $user->hasRole("locatore"));
    });

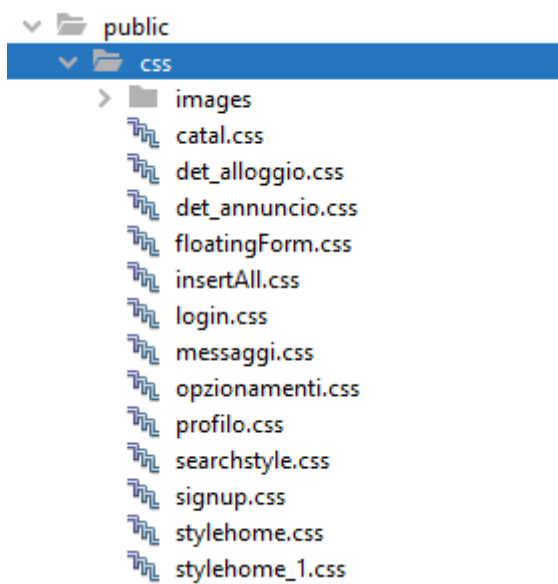
    Gate::define('isAdmin', function($user){
        return $user->hasRole("admin");
    });
}
```

## GATES PER I LIVELLI D'UTENZA

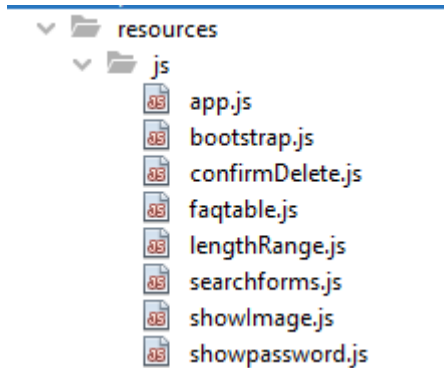
Resources/views/helpers/annImg.php (utilizzato nelle view home.blade e catalogo.blade)

```
<?php
$stampa = false;
foreach($allFotos as $foto){
    if($ann->id_alloggio == $foto->id_alloggio) {
        echo "<img src=../storage/app/{ $foto->path} class='catfoto' onmouseover='bigImg(this)' onmouseout='normalImg(this)'>";
        $stampa = true;
        break;
    }
}
if($stampa==false) echo "<img src=../storage/app/public/foto/default.jpg class='catfoto'>";
```

## HELPER PER LA VISUALIZZAZIONE FOTO CAT.



DIRECTORY CONTENENTE IMG UTILI + CSS



SCRIPT JS/JQUERY