SW Engineering CSC648/848 Summer 2020

House-a-Gator

Team 02

Milestone 2

07/08/2020

Team members:

Team Lead	Raviteja Guttula (rguttula@mail.sfsu.edu)
Front End Lead	Swetha Govindu
Front End Developer	Henry Meier
Front End Developer	Kevin Zhou
Front End Developer	Troy Turner
Back End Lead	Ashwini Uthirakumar
Back End Developer & Github Master	Fiona Senchyna

History:

Date Submitted:	07/08/2020
Date Revised:	07/14/2020

Table of Contents:

1. Functional Requirements - Prioritized

1. Priority 1:

Guest User:

1.1 Guest users shall be able to search based on a street name and zip code
1.2 Guest users shall be able to filter listings based on various prices, beds, baths, home types, and amenities
1.3 The media of the house posted by the homeowner shall be visible to the guest users
1.4 The location of the house on an interactive map for each listing shall be visible to the guest users
1.5 Guest users shall be able to register
1.6 Guest users shall be able to see the distance from the house to the university for every listing

Registered User:

1.8 Registered users shall be able to experience all the functionalities experienced by the guest users
1.9 Registered users shall be verified using their @mail.sfsu.edu or sfsu.edu email id
1.10 Registered users shall be able to post new listings
1.11 Registered users shall be able to see the contact details of landlords renting or selling their property and students/faculty looking for a roommate and message them
1.12 Registered users shall be able to reset forgotten password
1.13 Registered users shall have a personalized dashboard to check postings and messages

Admin:

1.14 Admin shall be required to approve the registered landlords listing in 24 hours, before it goes live. Approval shall be marked by a “verified” icon
1.15 Admin shall be able to delete flagged listings

1.16 Admin shall be able to delete flagged accounts and users

1.17 Admin shall be able to enforce terms and conditions for the website

2. Priority 2:

Guest User:

2.1 Guest users shall be able to sort based on price (highest/lowest), newest listing/ date posted

Registered User:

2.2 Registered users shall be able to create a short profile about themselves including their contact information

2.3 Registered users shall be able to save the filter, sort and search options to avoid repetitions

2.4 Registered users shall be able to see a personalized home page with recommendations

2.5 Registered users shall be able to update their password

3. Priority 3:

Guest User:

3.1 Guest users shall be able to see the nearby Restaurants, Laundromat, Bart station, Grocery and other essentials for every listing on the website

Registered User:

3.2 Registered users shall be able to save their favorite listings to view later

3.3 “Number of views” for every listing shall be visible only to registered users

2. List of main data items and entities

User: Someone who interacts with the website.

- Guest: Any user with/without an account can browse the listings on the website.
- Registered: User with an account who can post a listing (landlord) or reply to a listing (SFSU student/faculty).
- Admin: User who is registered and has special privileges for approving a listing to be shown on the website.

User registration record: Form for any guest user to register

- Name: Full name of the user
- Email: Email id to create an account
- Password: Password associated with the email id which will be encrypted
- Phone number: Contact detail of the user or landlord
- Profile pic: Optional profile picture of the user

Listing record:

- Title: Unique title given to the apartment/housing/room listed
- Description: Short description of the apartment/housing/room
- Price: Cost of the listing available to buy/rent
- House Type : Houses, Condos, Apartments, Town Houses
- Number of bed
- Number of bath
- Number of parking spots?
- Is the space furnished ? (Yes/No)
- Is the space for rent or sale ? (Rent/Sale)
- Are pets allowed ? (Yes/No)
- Is smoking allowed ? (Yes/No)
- Listing media (can be a list of pictures, not just a single picture)
- Listing status (Posted -> Verified -> Not available)

- Posted when the tenant/seller posts the listing
- Approved once the admin approves the listing
- Not available once the listing is occupied/sold

Address components:

- Building number
- Apartment
- Street name
- Zip code
- City
- State

Messages:

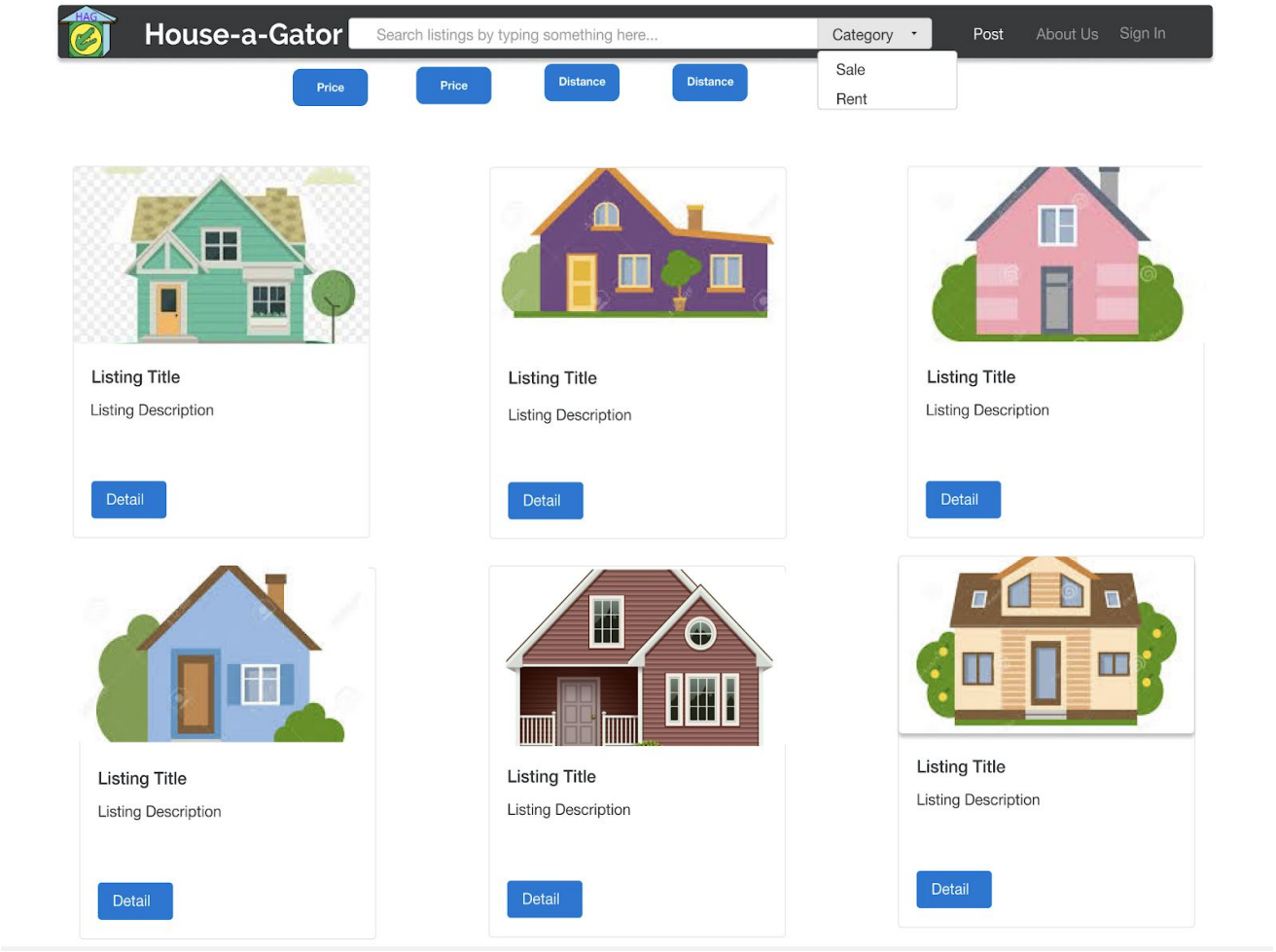
- Message ID
- Receiver ID: Current users ID
- Sender ID: User ID of the registered user
- Listing ID: Reference to property
- Date: Send/received date of the message
- Text: Content of the message

3. UI Mockups and Storyboards

Guest User

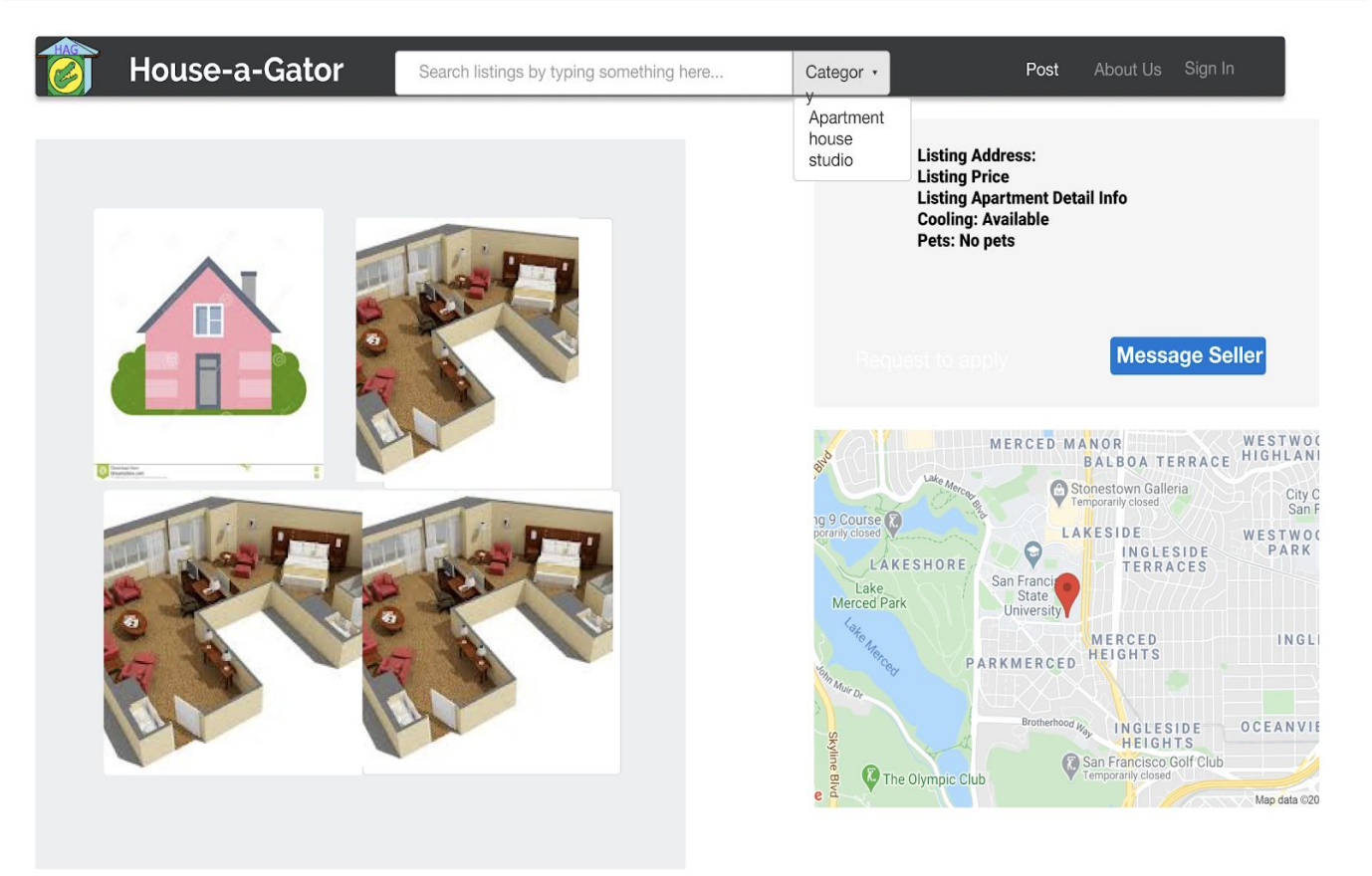
Abraham is a guest user to the House-a-Gator website. As a guest user he can view the different listings on the home page and also search for different categories in the search bar and also be able to choose different filters to refine the listing results on the home page.

Home Page



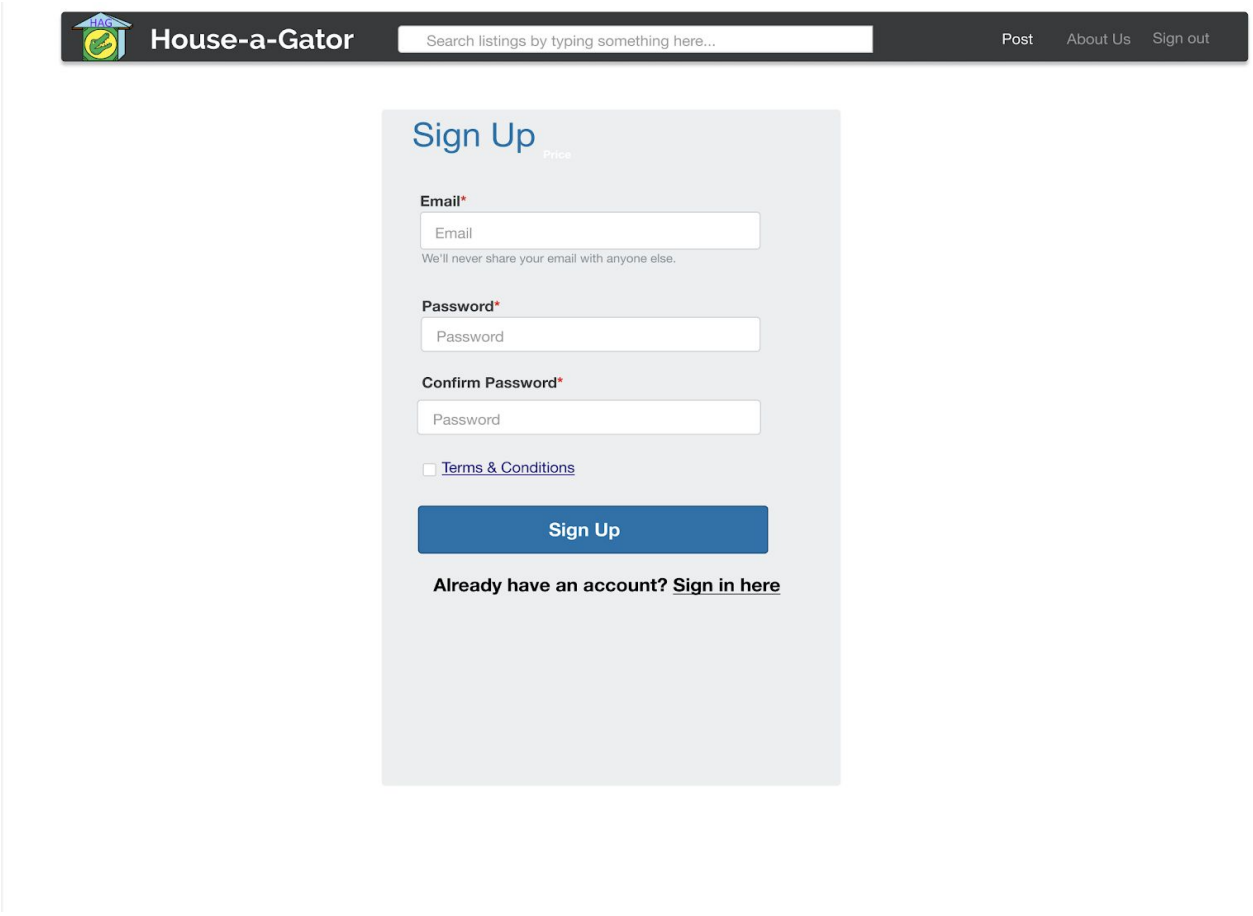
Listing Detail Page for Guest User

If Abraham wants to view the detailed version of the listing. Once he clicks on the listing card on the home page, he is redirected to the listing detail page where he can view more pictures ,detailed description of listing with map view.



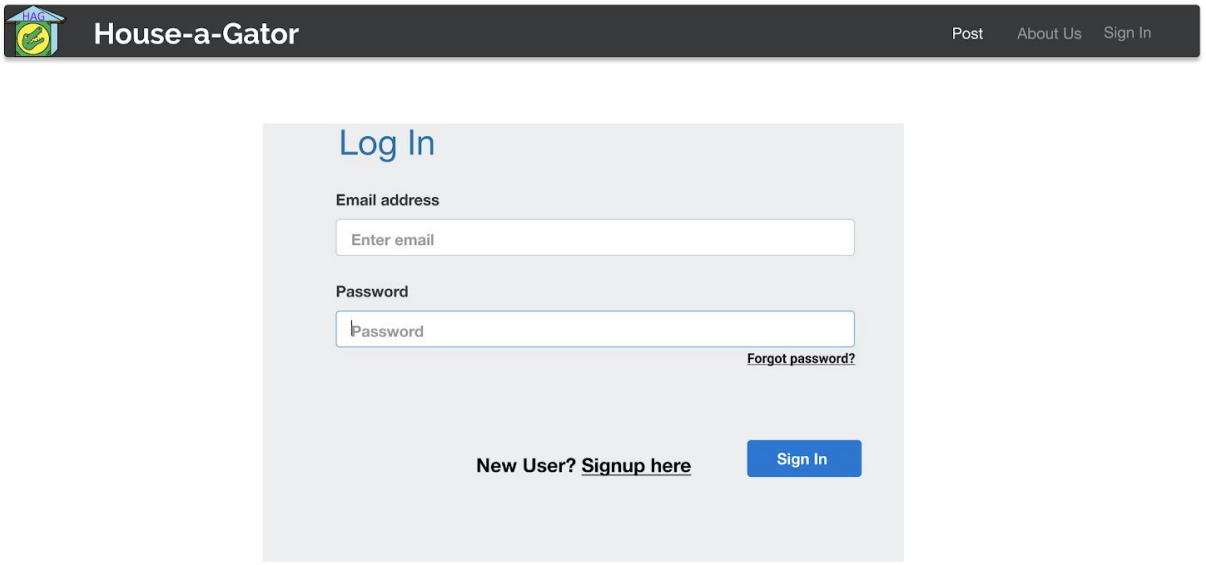
Sign Up Page

When Abraham tries to contact the seller from the listing detail page, House-a-Gator application is redirected to login/signup screen . He is able to register by giving email and password.



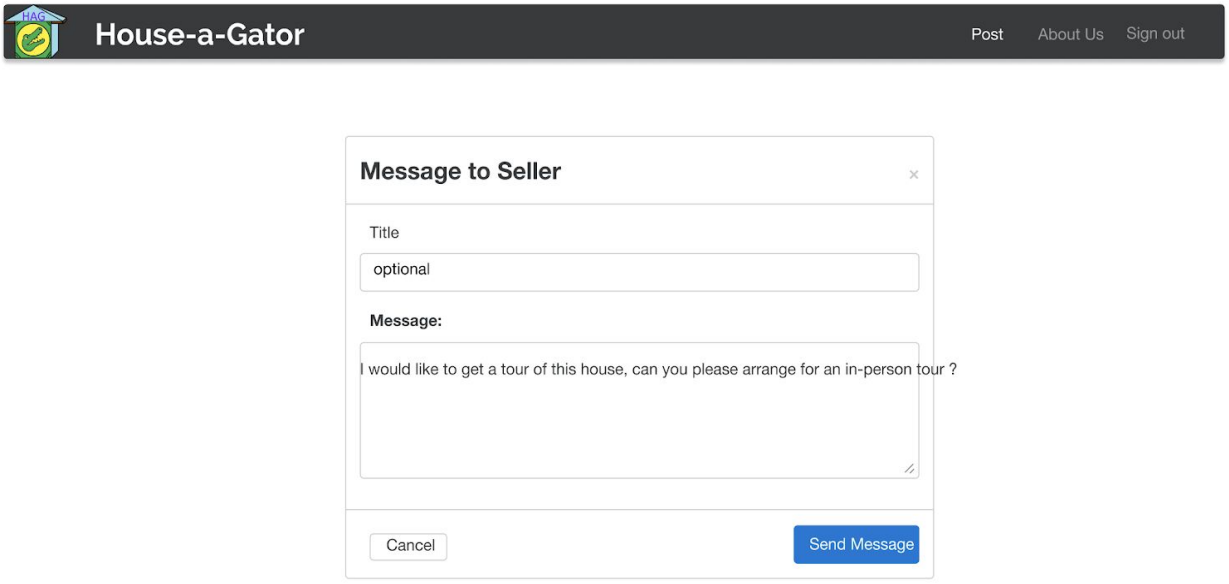
Login Page

Once Abraham is registered , he should be able to login with an email address and password into the House-a-Gator website.



Message Seller

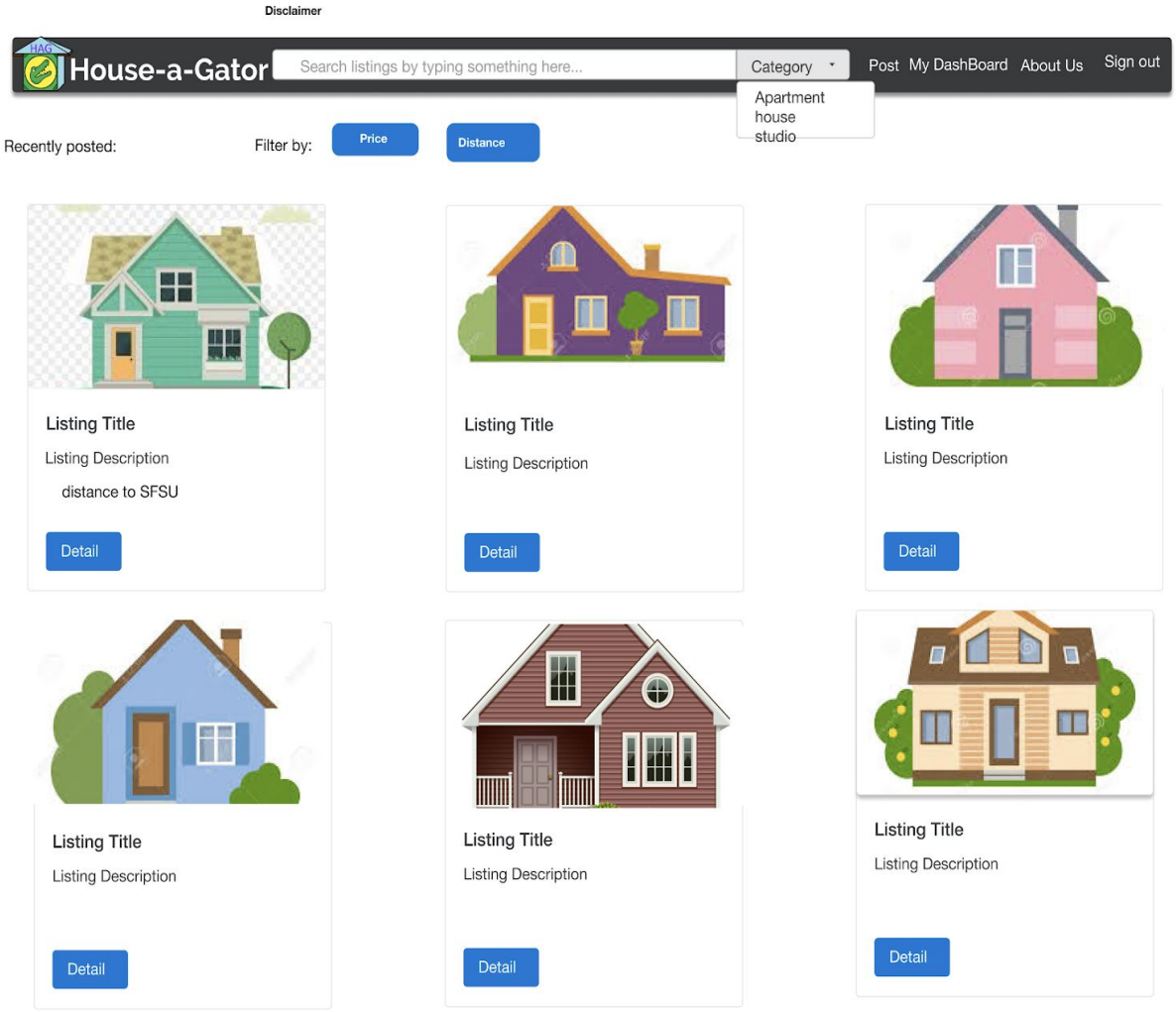
Once Abraham login into the application, he will be able to contact the seller from the listing detail page. When he clicks on message seller , message to seller modal will be prompted asking about the message



Registered User

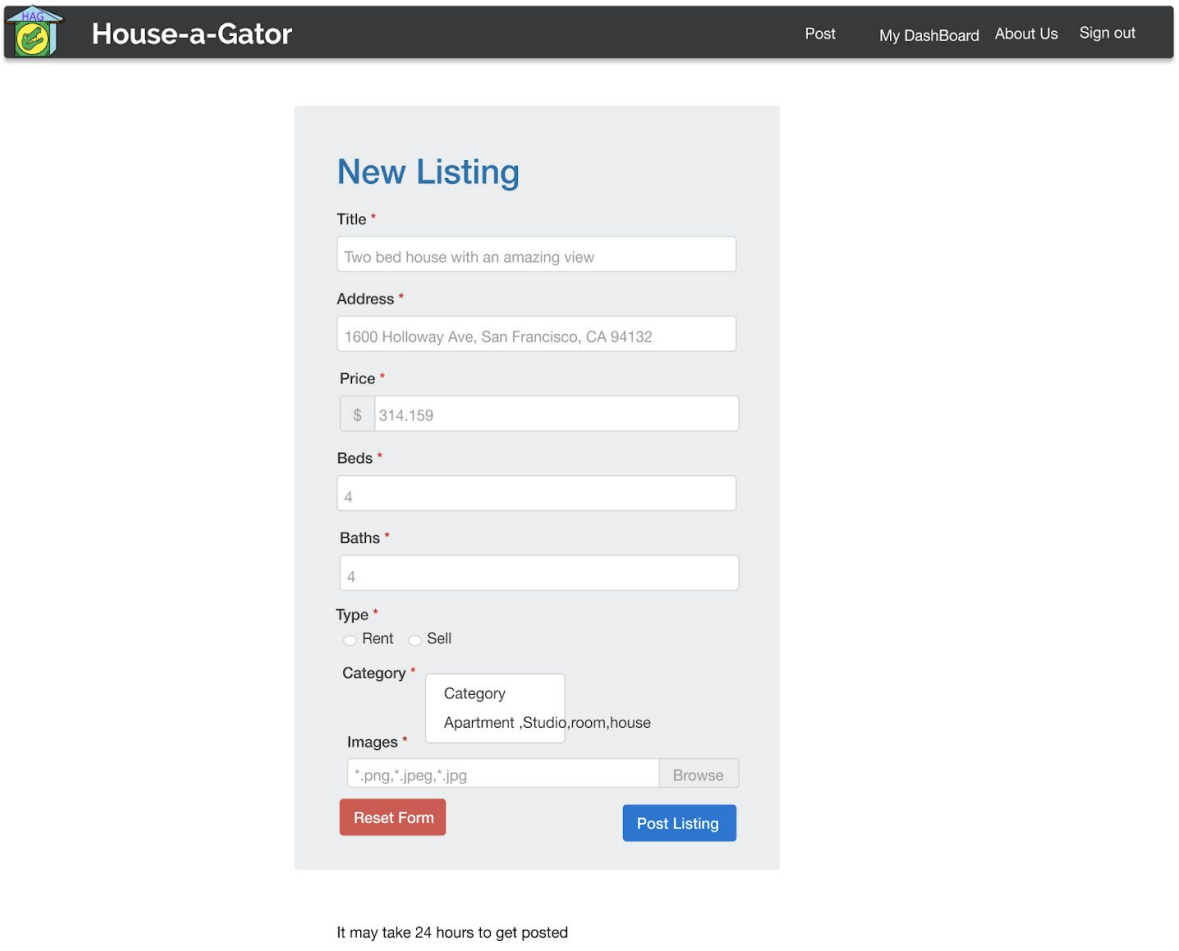
Vanessa is a registered user of the House-a-Gator application. As a registered user , she is able to post the new listing by clicking on the post link on navbar.

Home Page



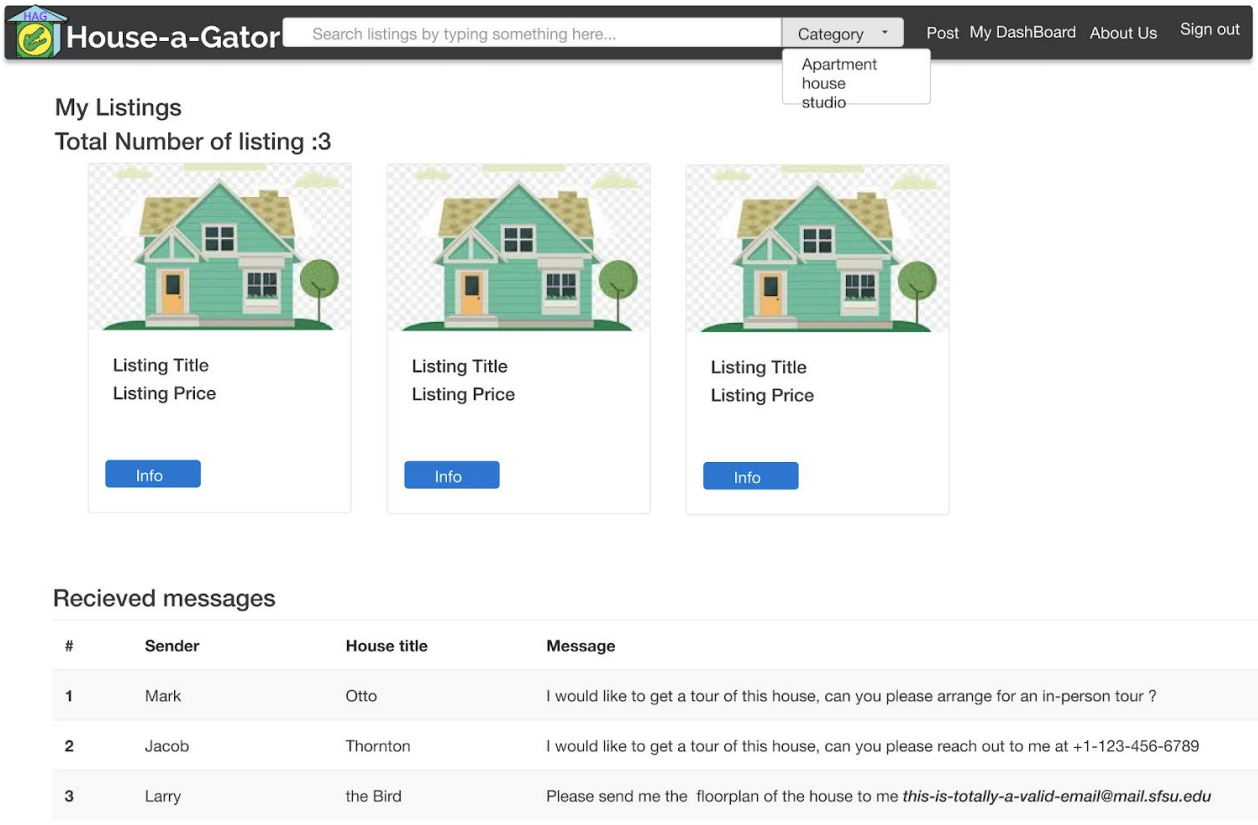
New Listing

In the new Listing Page where Vanessa can post listing information and images about the house she wanted to sell.



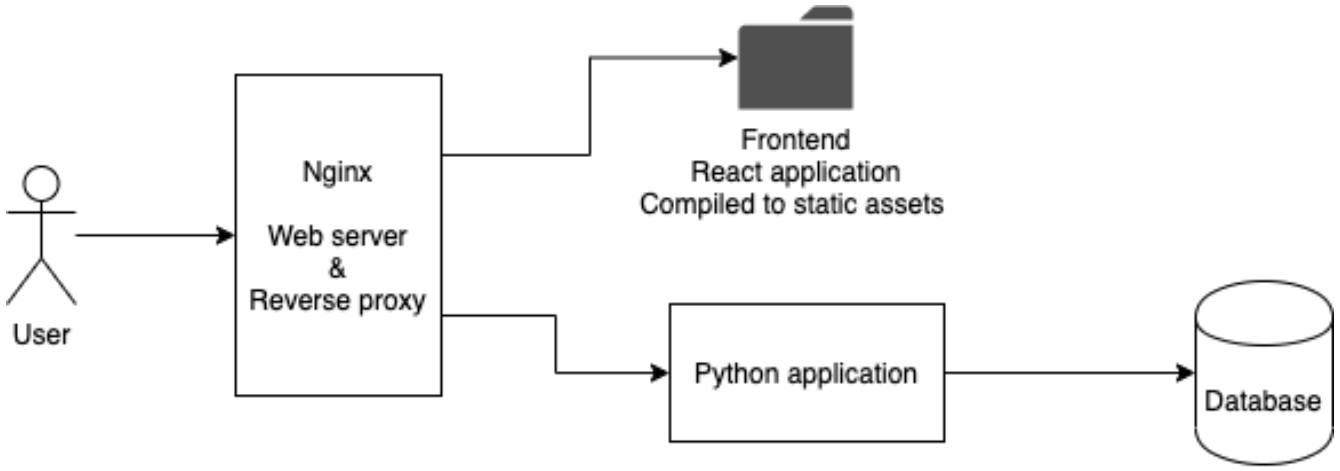
User Dashboard

Vanessa can check about her listings and messages on the user dashboard page .



4. High level Architecture, Database Organization:

4.1 Application Architecture:



4.2 DB organization:

Database name : HouseaGator

- Registration_record
  - User_id (primary key)
  - email
  - password
  - first_name
  - last_name

- Listings
  - Listing_id (primary key)
  - Listing_title
  - Listing_description
  - Street_name
  - Building_number
  - Apartment

        - City
        - State
        - Country
        - Landmark
        - Listing_price
        - Listing_type
        - Listing_status
        - Listing_views
        - Furnished
        - Square_foot
        - Number_of_bed
        - Number_of_bath
        - Parking_spots
        - Number_of_occupants
        - Pet_policy
        - Smoking_policy

    - Listings_media
        - Listings_media_id (primary key)
        - Listing_id (foreign key)
        - Listing_media

    - Listings_home_type
        - Home_type_id (primary key)
        - Listings_id (foreign key)
        - Home_type

    - Messages
        - message_id (primary key)
        - Sender_id (foreign key)
        - Receiver_id(foreign key)
        - listing_id(foreign key)
        - Date
        - Message_text

- **Media Storage**: File systems
- **Search/filter architecture and implementation**: SQL and *%like*

- **APIs**:
    - Registration -> POST /user
    - Login -> POST /session
    - Post Listing -> POST /listing
    - Get Listings (with filters and search params) -> GET /listings?<query_param>
    - Get Listing details -> GET /listings/<listing-id>
    - Send message -> POST /messages
    - Get messages sent to me -> GET /messages?receiverId=<userId>

5 .Identification of key risk factors

Risks	Resolution
Skills : Possessing little knowledge with front-end or back-end language.	Splitting the work up between two different groups, to assign personnel with more knowledge to designated tasks. Having others in the team teach either-side.
Schedule: Not always being available at the same time due to life and other classes.	Have a message board on Discord to keep everyone up to date. Can schedule video conferences when team members are available. Modifying functional requirements to fit the schedule.
Technical: Not knowing specific implementation of code for a function.	Providing necessary comments in the code. Researching online to find a way to get it working. Ask the team if they potentially know how to get a function to work properly.
Teamwork: New to teamwork type of programming with GitHub sharing	Other members possess knowledge of how to manage the Git commands. Pointing members in right direction to find the needed knowledge
Legal/content: N/A	Images to be downloaded from google images. Address of commercial properties to be used for listing

6. Project management

For M2, we split the team into two chunks.

- Frontend
- Backend

The front-end team worked on developing mockups needed for the milestone 2 document. The back-end team worked on data modeling without being blocked. Once the frontend team came up with a list of screens, the entire team held a review with the whole team and came up with a list of APIs to integrate with the frontend. The entire team had some discussion here to finalize the APIs and list of screens.

Our team also created a trello board to assign the tasks. This is a great way to track progress for the project, and it can be updated asynchronously. It will be helpful especially in an online based environment where all of us need not be working at the same time. It also saves time in our weekly meetings.