# Express.js Exam Preparation – Messenger

## Overview

Get familiar with the provided html. You have to create an application for chatting using express and handlebars. Run **npm install** before you start.

## Models

### User

The user model should have the following:

* **Username – string, required, unique**
* **First Name – string**
* **Last Name – string**
* **Blocked Users – collection of strings (usernames)**
* **Roles – collection of strings (either 'user' or 'Admin')**

### Thread

The thread model should have the following:

* **Users – collection of 2 users, required**
* **Date Created – a Date, required, default: current date**

### Message

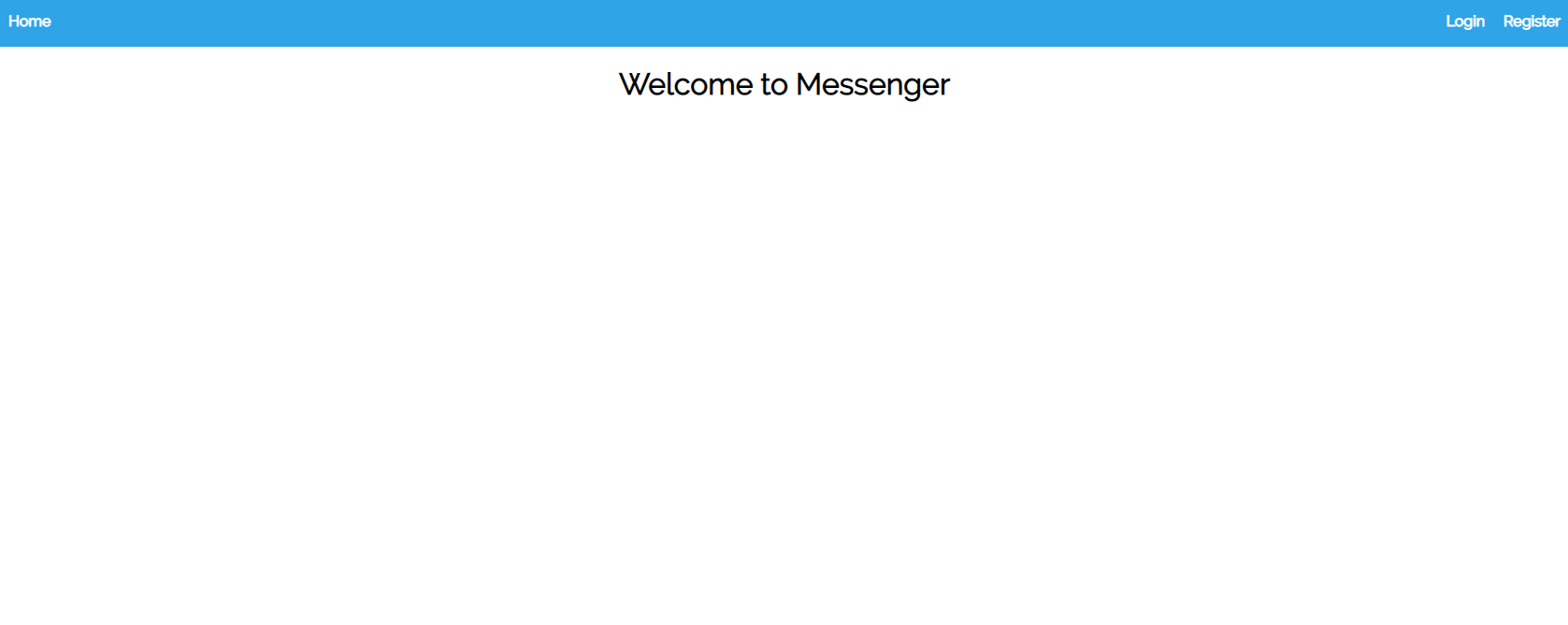
The message model should have the following:

* **Content – string, required**
* **User – ObjectId (User) – the receiver, required**
* **Thread – ObjectId (Thread), required**

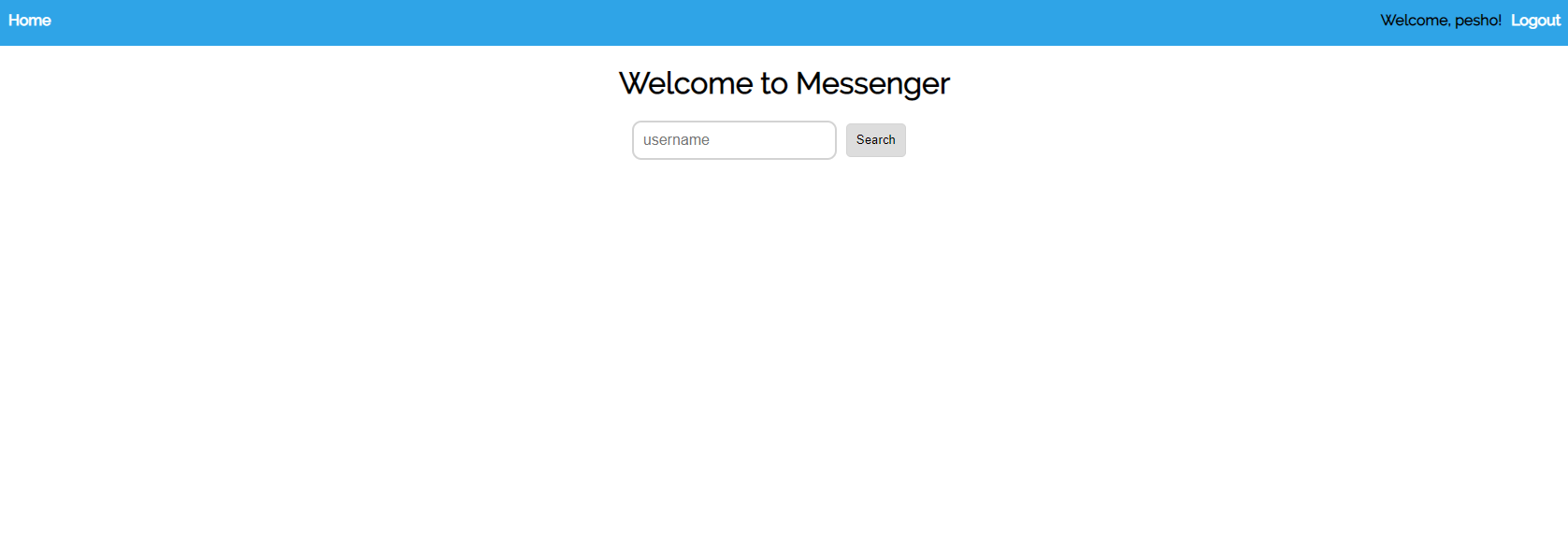
## Routes

### Home page – '/'

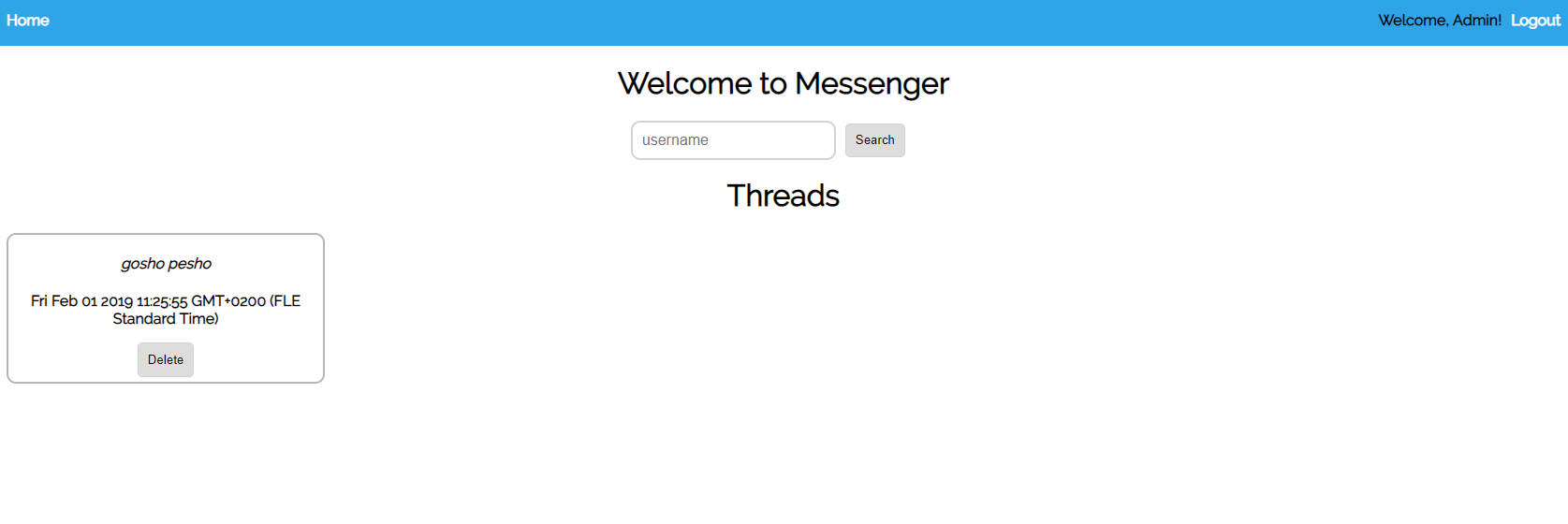
#### Anonymous



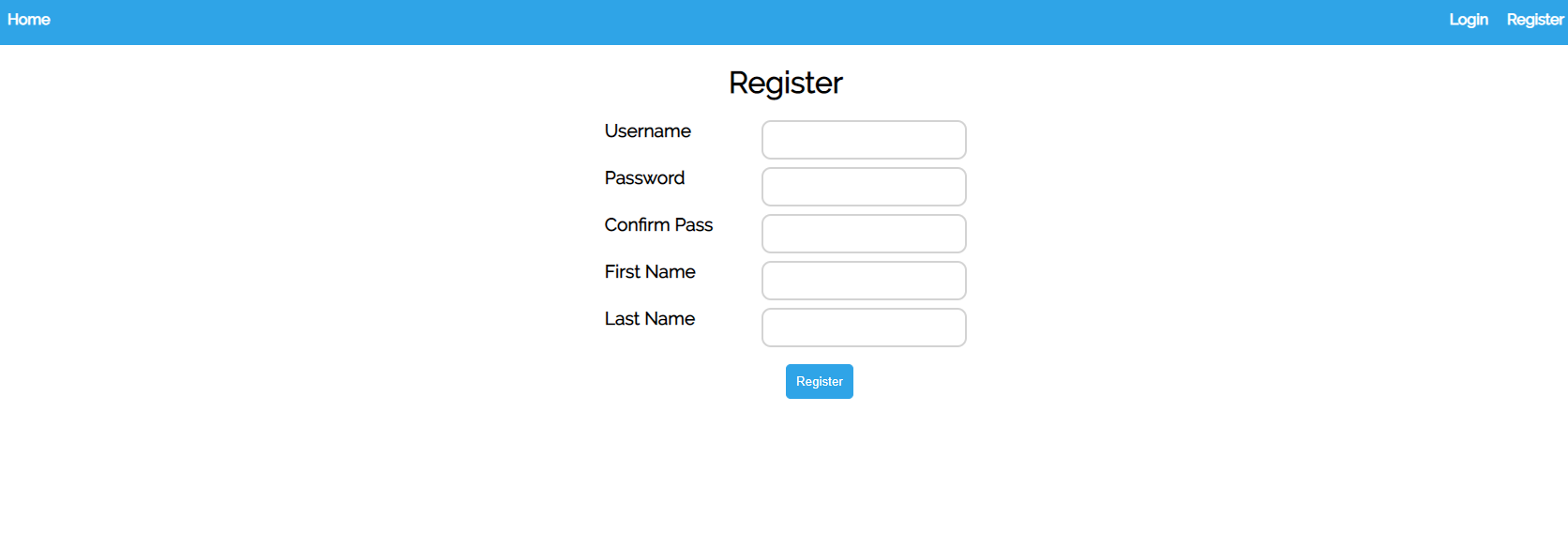
#### Logged in user



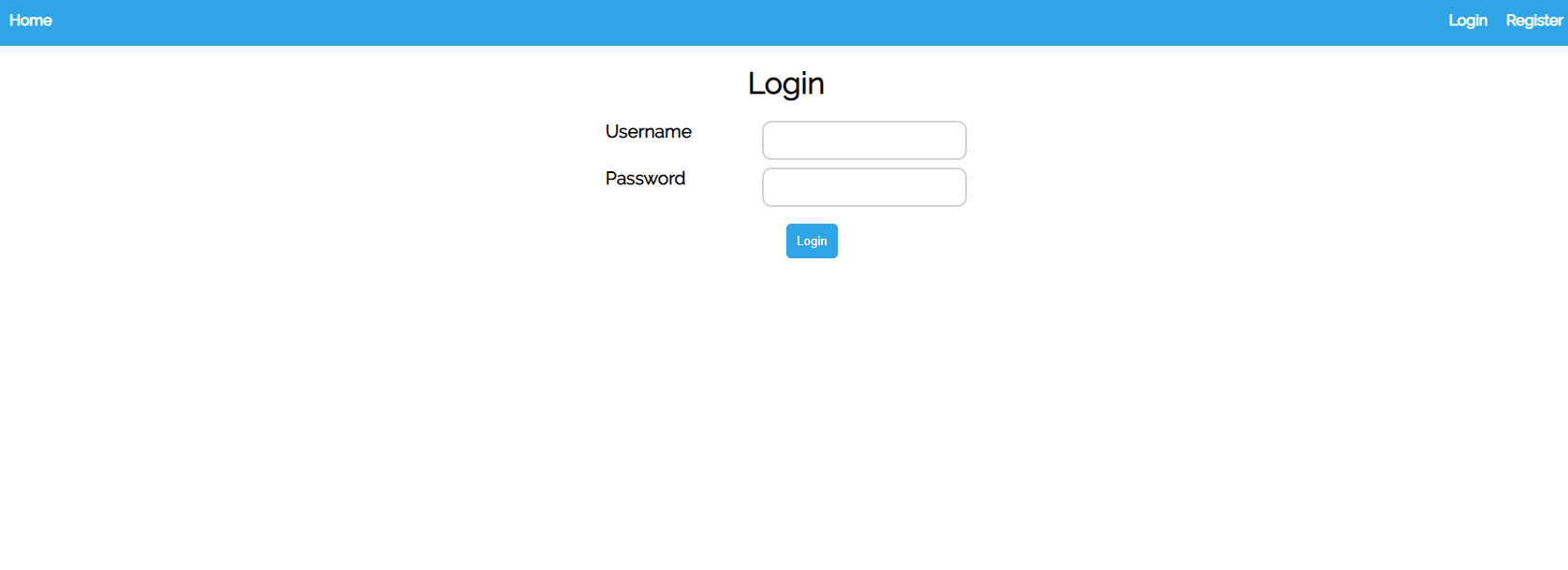
#### Logged in Admin



### Register – '/users/register'



### Login – '/users/login'



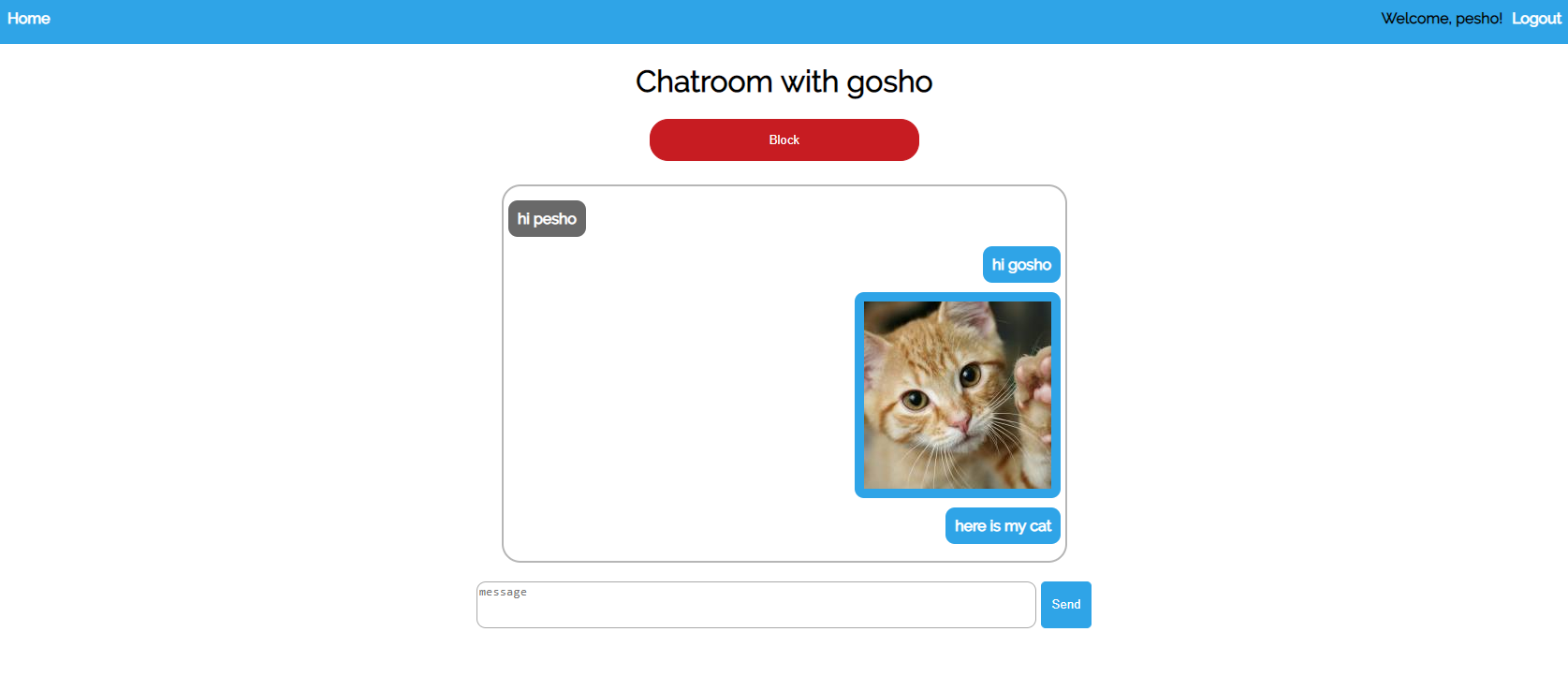
### Logout – '/users/logout'

### Find a thread – '/threads/find'

When a **logged-in** user wants to chat with another user, he submits the **search form** in the home page. If there is **already a thread with these two users**, you have to **open it**, otherwise **create a new thread**

### Open a thread – '/thread/:otherUser'

After the **'/threads/find'** route, **redirect** to this route. Find all the **messages that have this thread** (if any) and visualize them.



* Check if there is an **image among the messages**. If so, display it.

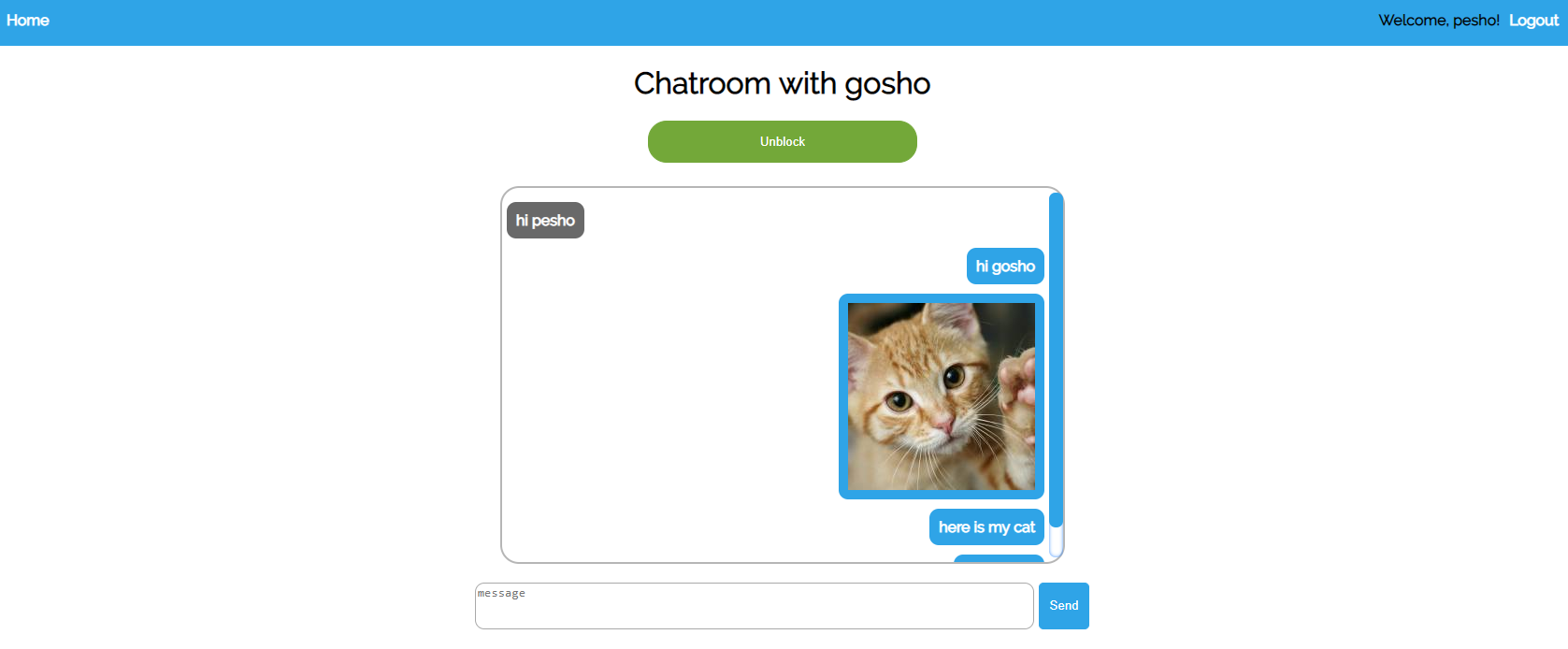
### Send a message – '/thread/:otherUser'

After the button is clicked, **create a new message** and set its **thread to be the current one** and set the **user to be 'otherUser'**

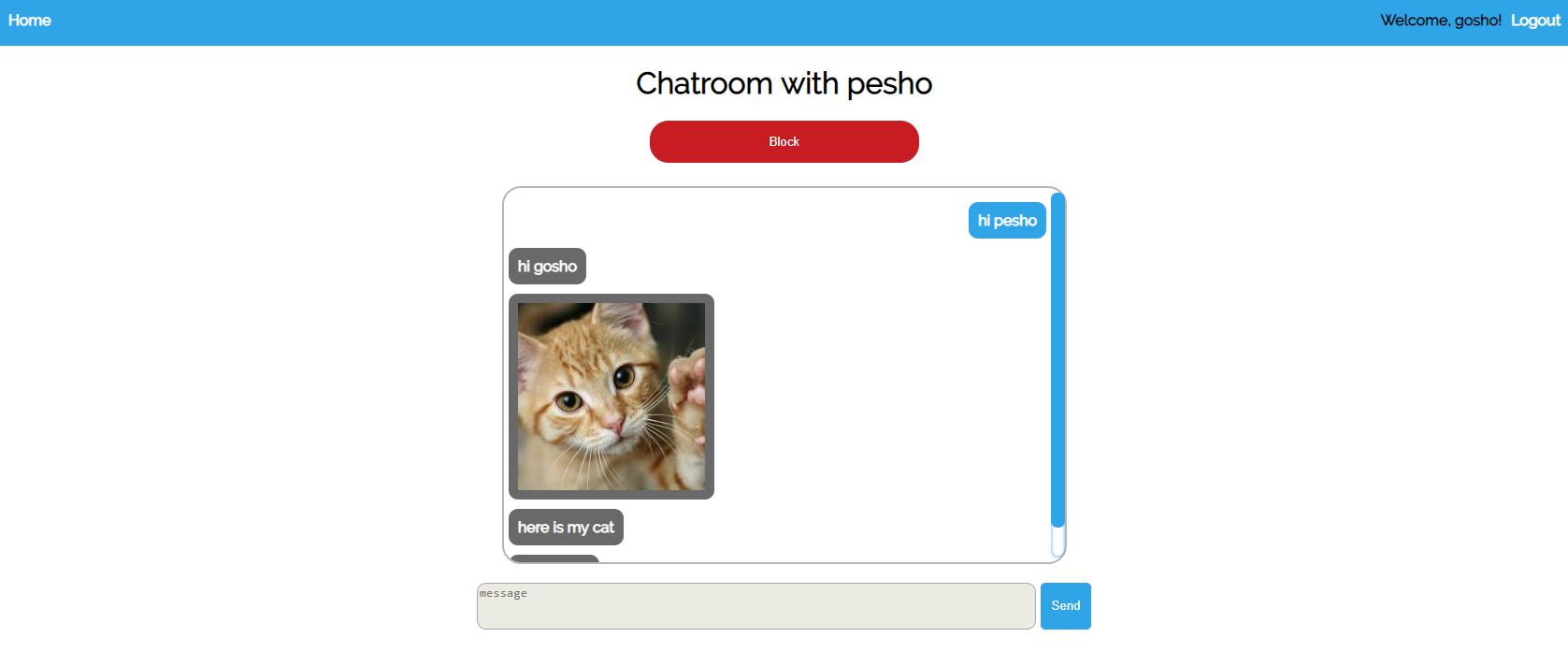
### Block user – '/block/:username'

When this button is clicked, you have to **add** the username to the **blockedUser** field of the current user. Blocked **users can no longer send messages** (the textarea is disabled). However, the **current user** **can** still **send messages to the blocked ones**. Redirect to the same page and **change the button**

#### Blocking a user view



#### Blocked user view

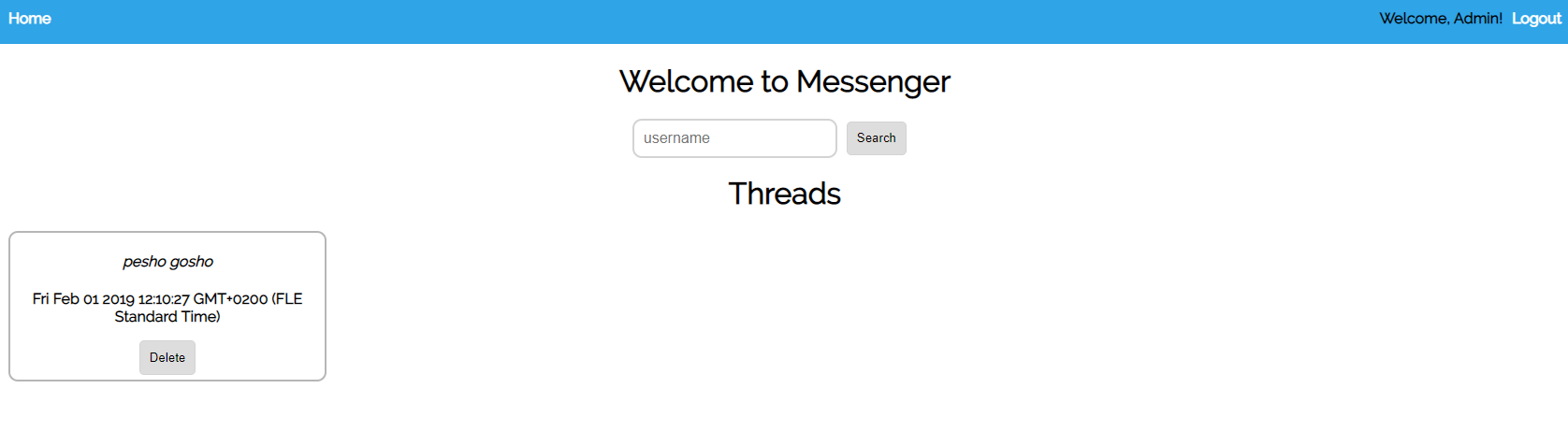


### Unblock user – '/unblock/:otherUser'

Does the opposite: remove the other user from the blocked users, and change the button, redirect to the same page

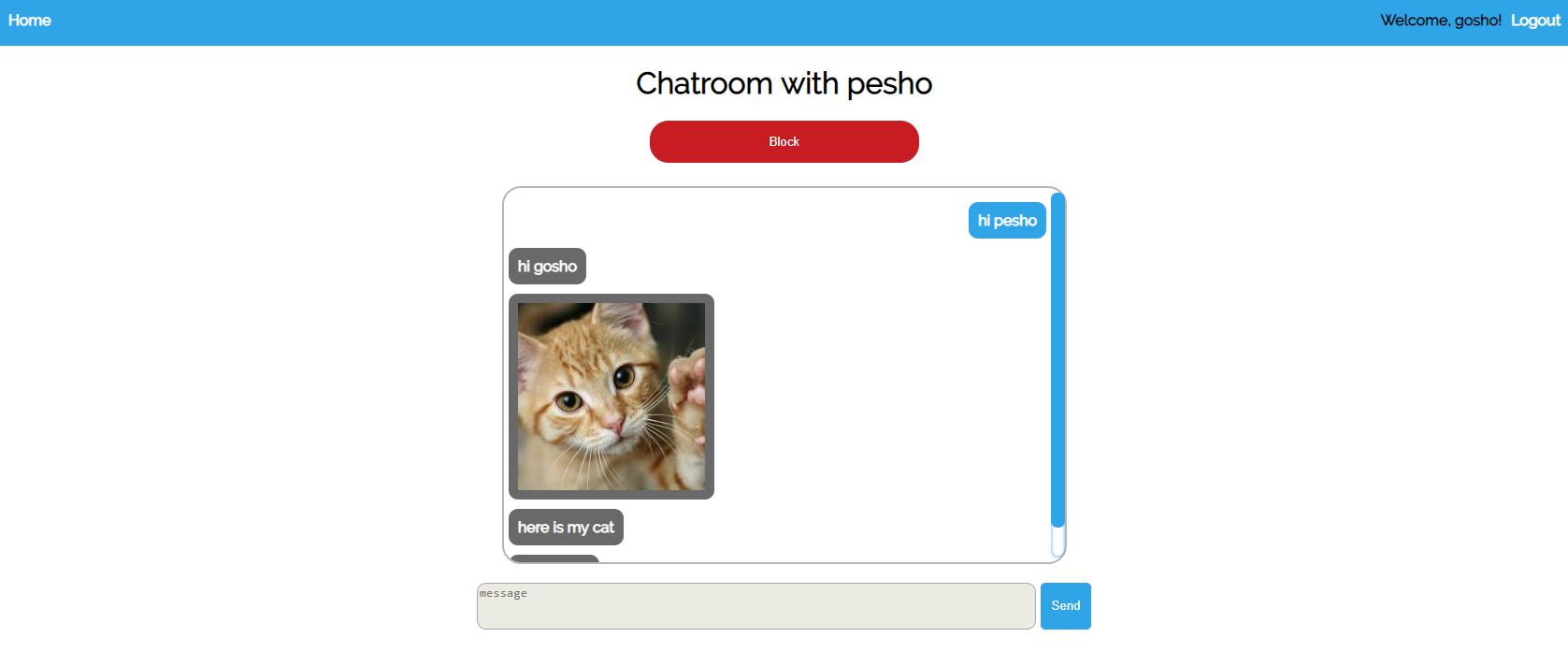
### (Admin only) Remove threads – '/threads/remove/:threadId'

In the home view, the admin can see all the threads with the 2 users and the date of creation

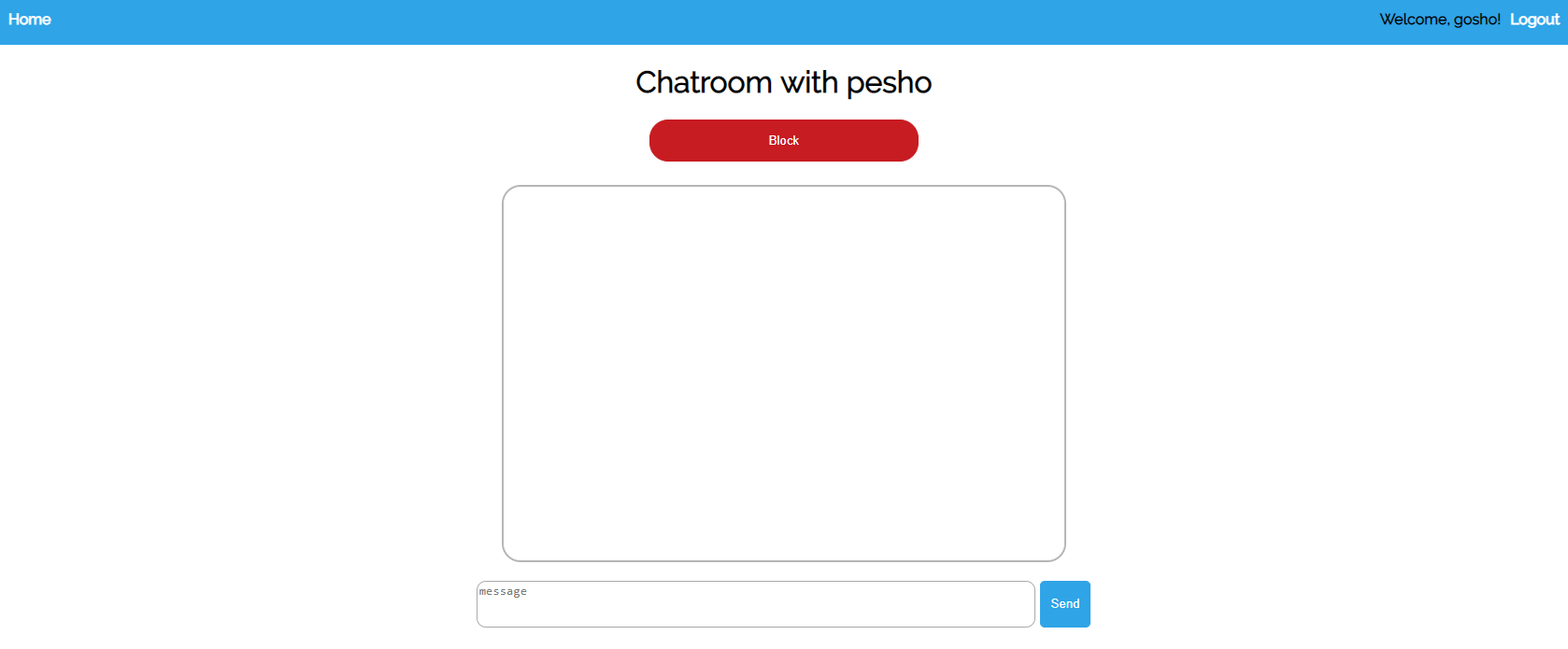


When the delete button is clicked, you have to **delete the thread** and the **messages connected to it** also. When the two **users chat again**, their **chat box should be empty**

#### Chat box before



#### Chat box after



## Restricted pages

* **Anonymous** – anonymous users should only be able to see the **home page**, should be able to **register** and **login**
* **Users** – users should be able to **start chats**, **send messages** **and logout**
* **Admin** – should be able to do **everything as the user + see all threads on the home page and delete them**

## Additional exercise

Try adding **notifications** and **validations.** The **styling** for the notifications is **included in the css**