

# Sprint 02 Documentation

## Initial set up

Prior to beginning, make sure we merge our sprint01 to our main branch now that everything's working.

- Activated the virtual environment
- Create a sprint 2 branch `sprint02`
- Go into the `sprint02` branch
- Double check to see if we're on the right branch now

```
source bin/djvenv/activate
```

```
[sprint01]
[(base) naomirodriguez@Naomis-MacBook-Air sprint % source djvenv/bin/activate
[(djvenv) (base) naomirodriguez@Naomis-MacBook-Air sprint % git branch sprint02
(djvenv) (base) naomirodriguez@Naomis-MacBook-Air sprint % git checkout sprint02
M      care_app/models.py
M      db.sqlite3
Switched to branch 'sprint02'
[(djvenv) (base) naomirodriguez@Naomis-MacBook-Air sprint % git branch
  main
  sprint01
* sprint02
```

Migrate your project

```
python manage.py makemigrations
python manage.py migrate
```

## Medication Model

I wanted to add one more model to this project.

Each patient will have a medication list and a medication can be added to the list.

In `models.py` add:

```
class Medication(models.Model):
    MEDICATION_OPTIONS = (
        ('Acetaminophen', 'Acetaminophen'),
        ('Adderall', 'Adderall'),
        ('Amitriptyline', 'Amitriptyline'),
        ('Amlodipine', 'Amlodipine'),
        ('Amoxicillin', 'Amoxicillin'),
```

```

        ('Ativan', 'Ativan'),
        ('Atorvastatin', 'Atorvastatin'),
        ('Azithromycin', 'Azithromycin'),
        ('Benzonatate', 'Benzonatate'),
        ('Brilinta', 'Brilinta'),
        ('Bunavail', 'Bunavail'),
        ...
        ('Xanax', 'Xanax')
    )
    patient = models.ForeignKey(Patient, on_delete=models.CASCADE)
    name = models.CharField("Medication Name", max_length=200,
                           choices=MEDICATION_OPTIONS, blank=False)
    dosage = models.IntegerField("Dosage", blank=False)
    takenDaily = models.BooleanField("Taken Daily?", default=False)
    startDate = models.DateField("Starting Date", blank=False, null=False)
    endDate = models.DateField("Ending Date", blank=False, null=False)

    class Meta:
        verbose_name = "Medication"
        verbose_name_plural = "Medications"

    def __str__(self):
        return f"{self.patient.name}'s {self.name}"

```

- [DateField](#)

In `admin.py` add:

```

from django.contrib import admin

# Register your models here.
from django.contrib import admin
from .models import Patient, Medication

admin.site.register(Patient)
admin.site.register(Medication)

```

Migrate changes and go to your admin [page](#)

Change Medication

**Fluffy Cottentail's Xanax**

Patient: Fluffy Cottentail

Medication Name: Xanax

Dosage: 30

Taken Daily?

Starting Date: 2024-04-14 Today | Note: You are 6 hours behind server time.

Ending Date: 2024-08-08 Today | Note: You are 6 hours behind server time.

**Buttons:** Save and add another Save and continue editing

Select Medication to change

Action: Go 0 of 3 selected

MEDICATION

Fluffy Cottentail's Xanax

Fluffy Cottentail's Brilinta

Fluffy Cottentail's Naloxone

3 Medications

Right now the only way to add edit or delete medications is through the admin page. I want to focus on the UI and getting the buttons to work.

Update the `patient_detail.html` Medications button:

```
<!-- Medication Button -->
<div class="col-md">
    <div style="display: flex; flex-direction: column; align-items: center; margin-top: 40px">
        <p style="font-family: 'Monaco', monospace; margin-left: 50px; margin-bottom: 10px;"><strong>Medications</strong></p>
        <a href="{% url 'medication-list' patient.id %}" style="margin-left: 50px; justify-content: center">
            
        </a>      </div></div>
```

- We changed the `<a href="{% url 'medication-list' patient.id %}">` section to go to our `medication_list.html`

Create a template called `medication_list.html`:

```
{% extends "care_app/base_template.html" %}

{% block content %}
    <!-- Medication List -->
    <div class="container-fluid">
        <h1 class="text-center" style="font-family: 'Monaco', monospace;">{{ patient.name }}'s Medication List</h1>
        <div style="margin-top: 50px;">
            <div class="row">
                <div class="col">
                    <table class="table table-striped">
                        <thead>
                            <tr>
                                <th>Medication Name</th>
                                <th>Dosage</th>
                                <th>Start Date</th>
                                <th>End Date</th>
                            </tr>
                        </thead>
                        <tbody>
                            {% for medication in medications %}
                                <tr>
                                    <td>{{ medication.name }}</td>
                                    <td>{{ medication.dosage }}</td>
                                    <td>{{ medication.startDate|date:"m/d/Y" }}</td>
                                    <td>{{ medication.endDate|date:"m/d/Y" }}</td>
                                </tr>
                            {% endfor %}
                        </tbody>
                    </table>
                </div>
            </div>
        </div>
    </div>
    </div>
    </div>
    <% endblock %>
```

- [Tables](#) in bootstrap

Create a new view in `views.py`:

```
def medication_list(request, patient_id):
    patient = Patient.objects.get(pk=patient_id)
    medications = Medication.objects.filter(patient=patient)
    return render(request, 'care_app/medication_list.html', {'patient': patient,
    'medications': medications})
```

add a `urls.py`:

```
path('patient/<int:patient_id>/medications/', views.medication_list,
name='medication-list'),
```



# Fluffy Cottentail's Medication List

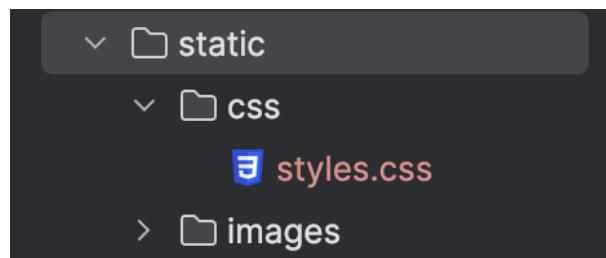
Medication Name	Dosage	Start Date	End Date
Amoxicillin	200	04/24/2024	05/25/2024
Lexapro	20	07/10/2023	02/13/2025

## Create, Read, Update, Delete a Medication

### User Interface Exploration

#### Site Color:

Previously to add an image to our navigation bar, we created a `static` directory. To add further customization, we're going to create a `css` directory under our `static` directory. Under the `css`, add a `styles.css` file.



In `styles.css` add the following:

```
body {
    background-color: #EEF0FF;
}
```

Back in our `base_template.html` at the beginning of the file under `<head>` add:

```
{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>  <title>Care-a-lot</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link
```

```

    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css"
    rel="stylesheet" integrity="sha384-
4bw+/aepP/YC94hEpVNViZdgIC5+VKNBQNGCHeKRQN+PtmoHDEXuppvnDJzQIU9"
    crossorigin="anonymous">
    <!-- Add link to your custom CSS file -->
    <link rel="stylesheet" href="{% static 'css/styles.css' %}">
</head>

```

In `settings.py` under the `static files` section make sure you have the following, (you may already have this)

```

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.2/howto/static-files/

STATIC_URL = "/static/"

STATICFILES_DIRS = [
    os.path.join(BASE_DIR, 'static')
]

```

## Home Page Layout

### Add a Logo

I only want my home screen to have a logo so we will be editing `home.html`

- I added another `.png` file to `static/images`  
Add the following:

```

{% extends "care_app/base_template.html" %}

{% load static %}

{% block content %}


<!-- Logo -->
    <div class="container-fluid d-flex justify-content-center align-items-center" style="height: 50vh; padding: 0px;">
        
    </div>
    {% if patients %}
        <ul>
            {% for patient in patients %}
                <li style="padding-bottom:10px;">
                    <strong>{{ patient.name }}'s Profile:</strong>
                    <a href="{% url 'patient-detail' patient.id %}" class="btn btn-primary">View Patient</a>
                </li>
            {% endfor %}
        </ul>
    {% endif %}


```

```

        </li>          {% endfor %}
    </ul>
    {% else %}
        <p>No active patients found.</p>
    {% endif %}
        <a href="{% url 'patient-create' %}" class="btn btn-success">Create
Patient</a>
</div>
{% endblock %}

```

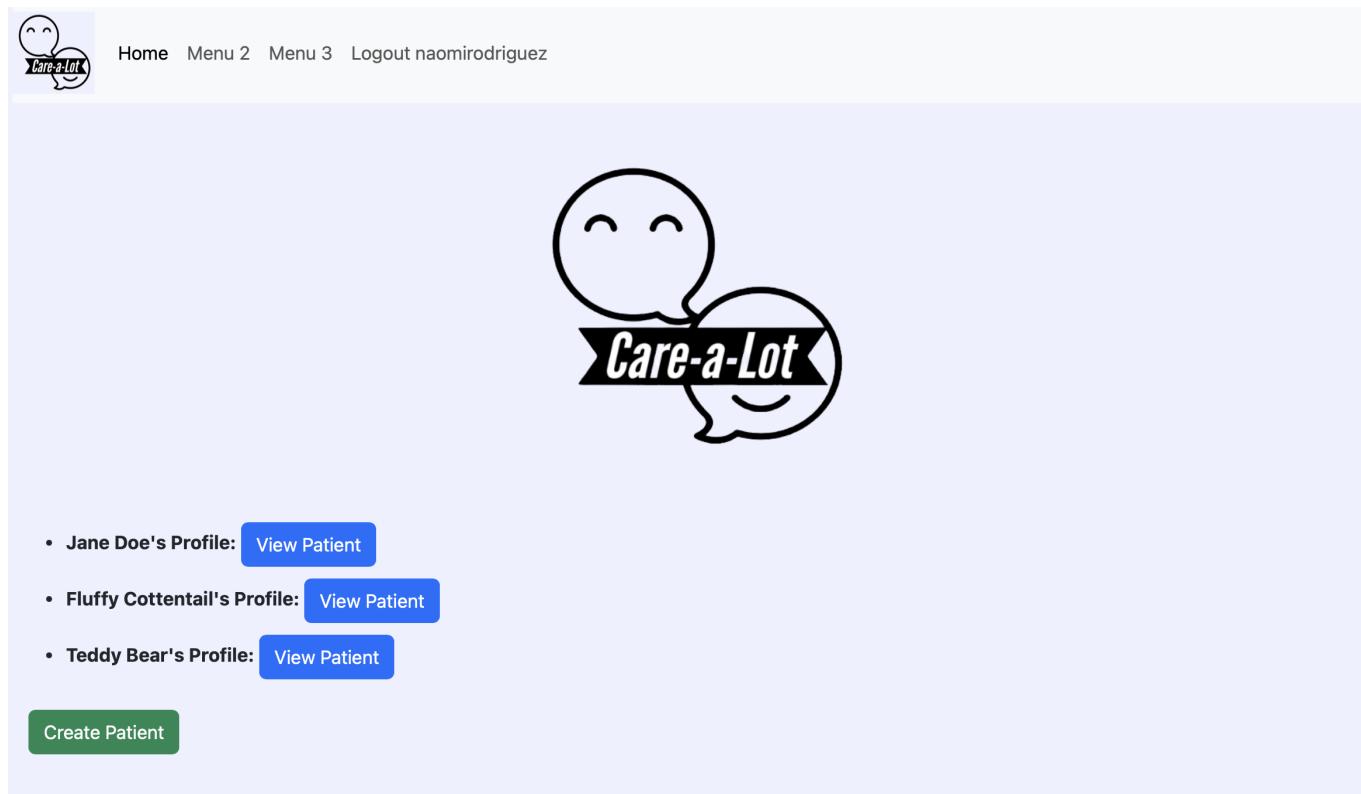
The portion we changed is:

- [container-fluid](#) which moves images from side to side (I had to do a lot of trial and error to get it all correct)

```

<div class="container-fluid d-flex justify-content-center align-items-center"
style="height: 50vh; padding: 0px;">
    
</div>

```



## Avatars

First I wanted to figure to space the patients on the screen

```
<div class="row justify-content-center">
    <div style="margin-bottom: 50px;">
```

- The second line moves the Create Patient button lower on the page. We will work on that button later.

If a Patient upload's a profile picture, their avatar should appear and be clickable.

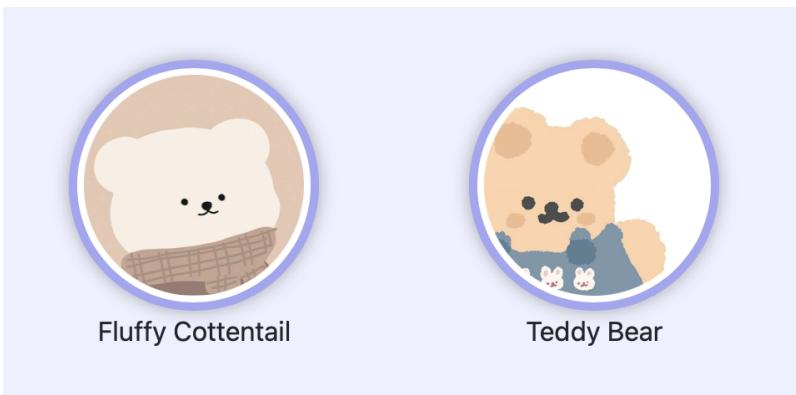
```
{% if patient.profile_picture %}
<a href="{% url 'patient-detail' patient.id %}">
    
</a>
```

- The img html element allows us to use the profile pictures as a clickable image.
- alt is used if images cannot be uploaded, increasing accessibility for all devices.
- class can let us use a rounded avatar picture.
- borders can be customized as well.



At the bottom of the page, we can add a box-shadow

```
<!-- Add CSS styles here -->
<style>
    .with-shadow {
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);
    }
</style>
```



If a Patient does not upload a profile picture, their first initial should appear in the avatar's space (similar to google).

```
{% else %}  
<a href="{% url 'patient-detail' patient.id %}">  
  <div class="initials-avatar" style="width: 150px; height: 150px; line-height: 150px; border-radius: 50%; background-color: #A2A6F2; color: #EEF0FF; font-size: 50px; font-weight: bold; box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);">{{  
    patient.name|slice:"1"|upper }}</div>  
</a>  
{% endif %}
```

- This time we're just setting the colors of the avatar's with no profile picture.



Currently, this is how I'm displaying the name under the avatar:

```
<p>{{ patient.name }}</p>
```

I wanted to change the fonts and move the name down slightly:

```
<p style="font-family: 'Monaco', monospace; margin-top: 10px;">{{ patient.name }}</p>
```



Jane Doe



Fluffy Cottentail



Teddy Bear

**Warning**

I tried to center the entire name of Patient's who do not upload profile pictures but even though I attempted to align the name in several different spaces, I couldn't figure out how to fix it.

This is what my entire screen looks like. I would like to move the avatars closer to the center of the page.

The screenshot shows a web interface for a medical or pet care application. At the top left is a logo for "Care-a-Lot" featuring two cartoon faces above a banner. To the right are navigation links: Home, Menu 2, Menu 3, and Logout naomirodriguez. Below the navigation is a large, stylized Care-a-Lot logo consisting of two smiling speech bubbles joined at the bottom, with the text "Care-a-Lot" in a bold, sans-serif font on a black banner. At the bottom of the screen are three patient profiles arranged horizontally. Each profile consists of a circular icon (purple for Jane Doe, brown for Fluffy Cottentail, and blue for Teddy Bear), the patient's name, and a small description below it. A green button labeled "Create Patient" is located at the bottom left.

Home Menu 2 Menu 3 Logout naomirodriguez

Care-a-Lot

Jane Doe

Fluffy Cottentail

Teddy Bear

Create Patient

When moving the margins, it squished the profile pictures.



Jane  
Doe



Fluffy  
Cottentail



Teddy  
Bear

Create Patient

Next time, I would figure out the margins prior to placing the avatars, I could not figure out how to completely center the avatars but it looks better!



Jane Doe



Fluffy  
Cottentail



Teddy Bear

Create Patient

**Create Patient Button**

Bootstrap has a huge [icon library](#) that we can use to make the website feel more professional!

- This [video](#) gives a simple and quick tutorial on how to use this library.
- Click the `install` section and under the CDN section, you can copy the first command.

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-  
icons@1.11.3/font/bootstrap-icons.min.css"> downloads the bootstrap icon library.
```

In `base_template.html`, add this line to your `head` section. It should look something like this:

```
<!DOCTYPE html>  
<html lang="en">  
  <head>    <title>Care-a-lot</title>  
    <meta charset="utf-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1">  
    <title>Bootstrap demo</title>  
    <link  
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css"  
      rel="stylesheet" integrity="sha384-  
      4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+PtmoHDEXuppvnDJzQIU9"  
      crossorigin="anonymous">  
      <!-- Add link to your custom CSS file -->  
      <link rel="stylesheet" href="{% static 'css/styles.css' %}">  
      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-  
      icons@1.11.3/font/bootstrap-icons.min.css">  
  </head>
```

To add the icon I added the following line:

```
<a href="{% url 'patient-create' %}" class="btn btn-outline-dark"  
  style="position: absolute; top: 100px; right: 40px;"><i class="bi bi-plus-lg">  
</i> </a>
```

- [buttons](#) in Bootstrap can be customized in several different manners.
- You can look through the bootstrap library and choose an icon you want to add to your page. I chose the following:



# Plus lg

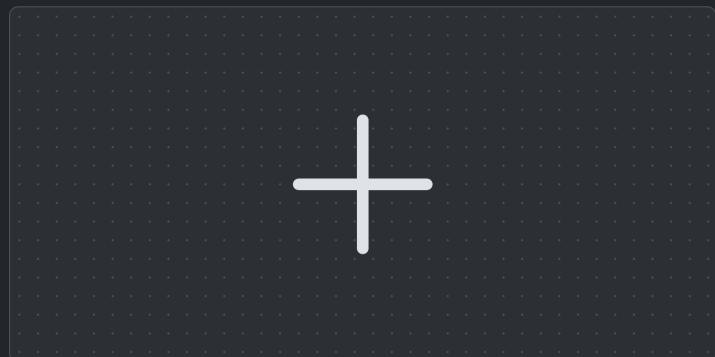
Tags: add, new

Category: Alerts, warnings, and signs



Want to go static?  
Deploy static sites,  
SPAs, and PHP  
Apps on Git Push  
with Deploy Now.

[ads via Carbon](#)



## Examples

- Below the `Icon font` section the class is displayed.



## Download

Download the SVG to use or edit.

[Download SVG](#)

## Icon font

Using the web font? Copy, paste, and go.

```
<i class="bi bi-plus-lg"> 
```

## Code point

The following displays the final customized home page:



Jane Doe

Fluffy  
Cottentail

Teddy Bear

home.html:

```
{% extends "care_app/base_template.html" %}

{% load static %}

{% block content %}
<div class="container">
    <!-- Logo -->
    <div class="container-fluid d-flex justify-content-center align-items-center" style="height: 50vh; padding: 0px;">
        
    </div>    {% if patients %}
    <div class="row justify-content-center">
        {% for patient in patients %}
        <div class="col-md-3 text-center">
            <div style="margin-bottom: 20px; margin-left: 60px;">
                {% if patient.profile_picture %}
                <a href="{% url 'patient-detail' patient.id %}">
                    
                </a>                {% else %}
                <a href="{% url 'patient-detail' patient.id %}">
```

```

        <div class="initials-avatar" style="width: 145px; height: 145px; line-height: 145px; border-radius: 50%; background-color: #A2A6F2; color: #EEF0FF; font-size: 70px; font-weight: bold; box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);">{{ patient.name|slice:"1"|upper }}</div>
        </a>                                {% endif %}
        <p style="font-family: 'Monaco', monospace; margin-top: 10px;">{{ patient.name }}</p>
        </div>      </div>      {% endfor %}
    </div>
    {% else %}
    <p>No active patients found.</p>
    {% endif %}
    <a href="{% url 'patient-create' %}" class="btn btn-outline-dark" style="position: absolute; top: 100px; right: 40px;"><i class="bi bi-plus-lg">
    </i> </a>
</div>

<!-- Add CSS styles here --&gt;
&lt;style&gt;
    .with-shadow {
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);
    }
&lt;/style&gt;

{% endblock %}
</pre>

```

## Patient Profiles

### Avatar on Profile Page

This is currently the layout of the Patient Profile page. I would like for the profile picture to look the same as the home page.



Home Menu 2 Menu 3 Logout naomirodriguez

## Fluffy Cottentail's Details



Email: fluffy@yahoo.com

Phone: 123-456-7899

Date of Birth: 01/01/2001

[Update Information](#) [Delete](#)

Using the same code we used in `home.html`, we want to create consistency throughout the screens and have the same buttons and layout as that page.

Home Menu 2 Menu 3 Logout naomirodriguez

## Fluffy Cottentail



[+]

Email: fluffy@yahoo.com

Phone: 123-456-7899

Date of Birth: 01/01/2001

Delete

## Buttons

Now I want to add my own 'buttons' which are images that I've created.

We are once again basing a lot of this portion from our `home.html`:

```

<!-- Custom Buttons -->
<div class="col-sm">
    <div style="display: flex; flex-direction: column; align-items: center;">
        <p style="font-family: 'Monaco', monospace; margin-left:50px; margin-bottom: 10px;"><strong>Calander</strong></p>
        <a href="#" style="margin-left: 50px; justify-content: center;">
            
        </a>    </div></div>

```

The screenshot shows a patient profile for "Fluffy Cottentail". At the top left is the Care-a-lot logo. The top navigation bar includes links for Home, Menu 2, Menu 3, and Logout, along with the user name naomirodriguez. The main title "Fluffy Cottentail" is centered above a circular profile picture of a white fluffy animal. Below the title is the text "Date of Birth: 01/01/2001". A grid of six circular icons represents different features: Calander (calendar icon), To-Do (list icon), Message (speech bubble icon), Medications (first aid kit icon), Documents (file with asterisk icon), and Settings (gear icon). A "Delete" button is located at the bottom center of the grid.

patient\_detail.html

```

{% extends "care_app/base_template.html" %}

{% load static %}

{% block content %}
    <h1 class="text-center" style="font-family: 'Monaco', monospace; margin-left:

```

```

190px">>{{ patient.name }}</h1>
<div class="container-fluid d-flex justify-content-center align-items-center"
style="height: 20vh; margin-left: 100px">
    {% if patient.profile_picture %}
        
    {% else %}
        <p>No profile picture available.</p>
    {% endif %}
</div>
<p class="text-center" style="font-family: 'Monaco', monospace; margin-left:
190px; margin-top: 10px"><strong> Date of Birth:</strong> {{ patient.dob }}</p>

<!-- Buttons -->
<a href="{% url 'patient-update' patient.id %}" class="btn btn-outline-dark"
style="position: absolute; top: 100px; right: 40px;"><i class="bi bi-plus-lg">
</i></a>

<!-- Custom Buttons -->
<div class="container-fluid">
    <div class="row justify-content-center" style="margin-left: 110px; margin-
top: 50px;">

        <div class="col-sm">
            <div style="display: flex; flex-direction: column; align-items:
center;">
                <p style="font-family: 'Monaco', monospace; margin-left:50px;
margin-bottom: 10px;"><strong>Calander</strong></p>
                <a href="#" style="margin-left: 50px; justify-content: center">
                    
                </a> </div> </div>
            <div class="col-sm">
                <div style="display: flex; flex-direction: column; align-items:
center;">
                    <p style="font-family: 'Monaco', monospace; margin-left:50px;
margin-bottom: 10px;"><strong>To-Do</strong></p>
                    <a href="#" style="margin-left: 50px; justify-content: center">
                        
                    </a> </div> </div>
            <div class="col-sm">
                <div style="display: flex; flex-direction: column; align-items:
center;">
                    <p style="font-family: 'Monaco', monospace; margin-left:50px;
margin-bottom: 10px;">
```

```

margin-bottom: 10px;">><strong>Message</strong></p>
            <a href="#" style="margin-left: 50px; justify-content: center">
                
            </a>           </div>           </div>
<div class="col-md">
    <div style="display: flex; flex-direction: column; align-items:
center; margin-top: 40px">
        <p style="font-family: 'Monaco', monospace; margin-left:50px;
margin-bottom: 10px;"><strong>Medications</strong></p>
        <a href="#" style="margin-left: 50px; justify-content: center">
            
        </a>           </div>           </div>
<div class="col-md">
    <div style="display: flex; flex-direction: column; align-items:
center; margin-top: 40px">
        <p style="font-family: 'Monaco', monospace; margin-left:50px;
margin-bottom: 10px;"><strong>Documents</strong></p>
        <a href="#" style="margin-left: 50px; justify-content: center">
            
        </a>           </div>           </div>
<div class="col-md">
    <div style="display: flex; flex-direction: column; align-items:
center; margin-top: 40px">
        <p style="font-family: 'Monaco', monospace; margin-left:50px;
margin-bottom: 10px;"><strong>Settings</strong></p>
        <a href="#" style="margin-left: 50px; justify-content: center">
            
        </a>           </div>           </div>
        <form method="post" action="{% url 'patient-delete' patient.id %}"
class="d-flex justify-content-center" style="margin-top: 40px; margin-left: 40px"
onsubmit="return confirm('Are you sure you want to delete {{ patient.name }} from
the care app?');">
            {% csrf_token %}
            <button type="submit" class="btn btn-dark">Delete</button>
        </form>
    </div></div>

<!-- Add CSS styles here -->
<style>
```

```

.with-shadow {
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);
}
</style>

{% endblock %}

```

## Navbar

I'd like for the Navbar to look like the rest of the website.

Start by choosing the same color of the background color and adding it to your `styles.css`:

```

.custom-navbar{
    background-color: #E6F0FF;
}

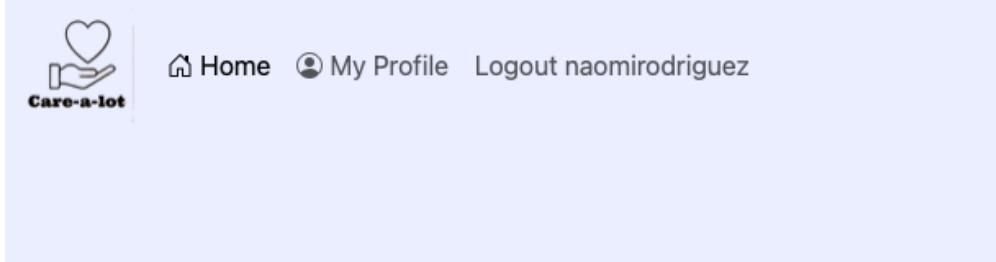
```

In `base_teemplate.html` add the icons from the bootstrap icon library:

```

<!-- Navbar-->
<nav class="navbar navbar-expand-lg custom-navbar">
    ...
<a class="nav-link active" aria-current="page" href="{% url 'index' %}"><i
    class="bi bi-house-door"></i> Home</a>

```



## Updating the 'Add' and 'Update' pages

All I changed on these pages were the headings and button colors!



[Home](#) [My Profile](#) Logout naomirodriguez

## Add Patient

Name:

Email:

Phone Number:

Date of Birth:

Is active:

Profile picture:  no file selected



[Home](#) [My Profile](#) Logout naomirodriguez

## Update Patient Information

Name: Fluffy Cottontail

Email:  fluffy@yahoo.com

Phone Number:  123-456-7899

Date of Birth:  01/01/2001

# Register

First, I wanted to get the base portion done without user authentication.

Create a `UserRegistrationForm` in `forms.py`

```
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth.models import User

class UserRegisterForm(UserCreationForm):
    email = forms.EmailField()

    class Meta:
        model = User
        fields = ['username', 'email', 'password1', 'password2']
```

- Django's [UserCreationForm](#) is very useful for all this registration portion!

Create a view in `views.py`:

```
def register(request):
    if request.method == 'POST':
        form = UserRegisterForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('login')
```

```

    else:
        form = UserRegisterForm()
    return render(request, 'care_app/register.html', {'form': form})

```

Create a template called `register.html`:

```

{% extends "care_app/base_template.html" %}

{% block content %}
<h1 class="text-center" style="font-family: 'Monaco', monospace; margin-left: 190px; margin-top: 20px;">Register</h1>
<form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Register</button>
</form>
{% endblock %}

```

in `urls.py` add a path:

```
path('register/', views.register, name='register'),
```

The screenshot shows the 'Care-a-lot' website's registration page. At the top, there is a navigation bar with links for Home, My Profile, Register, Login, and Logout. The 'Care-a-lot' logo, which features a hand holding a heart, is also present. The main content area has a large title 'Register'. Below it are three input fields: 'Username' (with a note: 'Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.'), 'Email', and 'Password' (with a note: 'Your password can't be too similar to your other personal information. Your password must contain at least 8 characters. Your password can't be a commonly used password. Your password can't be entirely numeric.'). Below the password field is a note: 'Enter the same password as before, for verification.' At the bottom is a 'Register' button.

## Login and Logout

In `views.py` add:

```

def login_view(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']

```

```

        user = authenticate(request, username=username, password=password)
        if user is not None:
            login(request, user)
            return redirect('index')
        else:
            # Handle invalid login
            pass
    return render(request, 'registration/login.html')

def logout_view(request):
    logout(request)
    return redirect('index')

@login_required
def profile(request):
    user = request.user
    return render(request, 'care_app/profile.html' , {'user': user})

```

- The profile function is used so that when we click the My Profile section on the navbar, it shows that the user is logged in.

Create a template in `profile.html`:

```

<!-- profile.html -->

{% extends "care_app/base_template.html" %}

{% block content %}
    <h1 class="text-center" style="font-family: 'Monaco', monospace; margin-left: 190px; margin-bottom: 40px;">Profile</h1>
    <p>Username: {{ user.username }}</p>
{% endblock %}

```

In `urls.py` add the following:

```

path('login/', views.login_view, name='login'),
path('logout/', views.logout_view, name='logout'),
path('accounts/profile/', views.profile, name='profile'),

```



[Home](#) [My Profile](#) [Register](#) [Login](#) [Logout](#)

# Login

Username  
Password



[Home](#) [My Profile](#) [Logout myName](#)

# Profile

Username: myName

## User Authentication

Now that we have that working, I'll be focusing on [Role-Based Access Control](#)

I want for there to be three options:

- Admin: Allowed access to the entire site, all portions of CRUD
- Manager: Allowed to Create, Update, and Read. Used for patients who may have disabilities and need someone to help them manage their schedules, etc.
- User: Although it would be nice for anyone to view the site, there is sensitive information on the site.

I need to add a model to `models.py`:

```
class CustomUser(AbstractUser):  
    ROLE_CHOICES = (  
        ('admin', 'Admin'),  
        ('manager', 'Manager'),  
        ('user', 'User'),  
    )  
    role = models.CharField(max_length=10, choices=ROLE_CHOICES)  
    groups = models.ManyToManyField('auth.Group',  
        related_name='custom_user_groups')  
    user_permissions = models.ManyToManyField('auth.Permission',  
        related_name='custom_user_permissions')
```

- I kept getting the following error prior to adding the last two lines `groups` and `user_permissions`.

```

django.core.management.base.SystemCheckError: SystemCheckError: System check identified some issues:

ERRORS:
auth.User.groups: (fields.E304) Reverse accessor 'Group.user_set' for 'auth.User.groups' clashes with reverse accessor for 'care_app.CustomUser.groups'.
    HINT: Add or change a related_name argument to the definition for 'auth.User.groups' or 'care_app.CustomUser.groups'.
auth.User.user_permissions: (fields.E304) Reverse accessor 'Permission.user_set' for 'auth.User.user_permissions' clashes with reverse accessor for 'care_app.Cu
stomUser.user_permissions'.
    HINT: Add or change a related_name argument to the definition for 'auth.User.user_permissions' or 'care_app.CustomUser.user_permissions'.
care_app.CustomUser.groups: (fields.E304) Reverse accessor 'Group.user_set' for 'care_app.CustomUser.groups' clashes with reverse accessor for 'auth.User.groups'.
    HINT: Add or change a related_name argument to the definition for 'care_
app.CustomUser.groups' or 'auth.User.groups'.
care_app.CustomUser.user_permissions: (fields.E304) Reverse accessor 'Permission
.user_set' for 'care_app.CustomUser.user_permissions' clashes with reverse acces
sor for 'auth.User.user_permissions'.
    HINT: Add or change a related_name argument to the definition for 'care_
app.CustomUser.user_permissions' or 'auth.User.user_permissions'.

```

- It seemed that there was an issue with my CustomUser model.
- I was led me down the path of [related\\_name](#) and [django.contrib.auth](#)

Create a new python file called `decorators.py` under `care_app/decorators.py`:

```

from functools import wraps
from django.http import HttpResponseRedirect

def admin_required(view_func):
    @wraps(view_func)
    def wrapped_view(request, *args, **kwargs):
        if request.user.is_authenticated and request.user.role == 'admin':
            return view_func(request, *args, **kwargs)
        else:
            return HttpResponseRedirect("You are not authorized to access this
page.")
    return wrapped_view

def manager_required(view_func):
    @wraps(view_func)
    def wrapped_view(request, *args, **kwargs):
        if request.user.is_authenticated and request.user.role in ['admin',
'manager']:
            return view_func(request, *args, **kwargs)
        else:
            return HttpResponseRedirect("You are not authorized to access this
page.")
    return wrapped_view

def user_required(view_func):
    @wraps(view_func)

```

```

def wrapped_view(request, *args, **kwargs):
    if request.user.is_authenticated:
        return view_func(request, *args, **kwargs)
    else:
        return HttpResponseRedirect("You are not authorized to access this
page.")
    return wrapped_view

```

- [Django decorators](#)
- [Django authentication](#)

In `views.py` add:

```

from .decorators import admin_required, manager_required, user_required

# User Authentication
@admin_required
def admin_view(request):
    return HttpResponseRedirect("Admin View")

@manager_required
def manager_view(request):
    return HttpResponseRedirect("Manager View")

@user_required
def user_view(request):
    return HttpResponseRedirect("User View")

@login_required
@admin_required
def admin_dashboard(request):
    return render(request, 'profile.html', {'user': request.user})

@login_required
@manager_required
def manager_dashboard(request):
    return render(request, 'profile.html', {'user': request.user})

@login_required
@user_required
def user_dashboard(request):
    return render(request, 'profile.html', {'user': request.user})

```

- make sure to import your decorators otherwise an error will occur!

Modify `forms.py`:

```

# Registration
class UserRegisterForm(UserCreationForm):
    email = forms.EmailField()
    ROLE_CHOICES = (
        ('admin', 'Admin'),
        ('manager', 'Manager'),
        ('user', 'User'),
    )
    role = forms.ChoiceField(choices=ROLE_CHOICES)

    class Meta:
        model = User
        fields = ['username', 'email', 'password1', 'password2', 'role']

```

In urls.py:

```

# User Authentication
path('admin/', views.admin_view, name='admin'),
path('manager/', views.manager_view, name='manager'),
path('user/', views.user_view, name='user'),
path('admin/dashboard/', views.admin_dashboard, name='admin_dashboard'),
path('manager/dashboard/', views.manager_dashboard, name='manager_dashboard'),
path('user/dashboard/', views.user_dashboard, name='user_dashboard'),

```



Home My Profile Register Login Logout

## Register

Username:  Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

Email:

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation:  Enter the same password as before, for verification.

Role:

Now when users register they can choose whether they're admin, managers, or users!  
*But that's the only change, the admin, manager and user can all do the same thing still.*



[Home](#) [My Profile](#) [Register](#) [Login](#) [Logout](#)

# Register

Username:  Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

Email:

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation:  Enter the same password as before, for verification.

Role:  [Change](#)

[Register](#)

## AttributeError at /register/

'User' object has no attribute 'role'

Request Method: POST  
Request URL: http://127.0.0.1:8000/register/  
Django Version: 4.2  
Exception Type: AttributeError  
Exception Value: 'User' object has no attribute 'role'  
Exception Location: /Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 Intro to Software Engineering/cs3300/sprint/care\_app/views.py, line 79, in register  
Raised during: care\_app.views.register  
Python Executable: /Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 Intro to Software Engineering/cs3300/sprint/djvenv/bin/python  
Python Version: 3.11.7  
Python Path: ['/Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 Intro',  
 'to Software Engineering/cs3300/sprint',  
 '/opt/anaconda3/lib/python311.zip',  
 '/opt/anaconda3/lib/python3.11',  
 '/opt/anaconda3/lib/python3.11/lib-dynload',  
 '/Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 Intro',  
 'to Software Engineering/cs3300/sprint/djvenv/lib/python3.11/site-packages']  
Server time: Fri, 26 Apr 2024 20:56:01 +0000

### Traceback [Switch to copy-and-paste view](#)

/Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 Intro to Software Engineering/cs3300/sprint/djvenv/lib/python3.11/site-packages/django/core/handlers/exception.py, line 55, in inner

55. response = get\_response(request)

▶ Local vars

/Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 Intro to Software Engineering/cs3300/sprint/djvenv/lib/python3.11/site-packages/django/core/handlers/base.py, line 197, in \_get\_response

197. response = wrapped\_callback(request, \*callback\_args, \*\*callback\_kwargs)

▶ Local vars

# Testing

## Resources:

- [Testing in Django](#)
- [Testing tools](#)
- [Writing and running tests](#)
- [Advanced testing topics](#)

Download selenium to your system

In settings.py add selenium.

```
asgiref==3.8.1
Django==4.2
django-bootstrap5==23.4
seleniumparse==0.4.4
selenium~=4.20.0
```

In order to get these tests to run, in the terminal enter:

```
python manage.py test
```

At this point, I got the following message:

```
(djvenv) (base) naomirodriguez@Naomis-MacBook-Air sprint % python manage.py test
Found 4 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
FnE.E
=====
ERROR: test_user_registration (care_app.tests.IntegrationTestCase.test_user_registration)
-----
Traceback (most recent call last):
  File "/Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 Intro to Software Engineering/cs3300/sprint/care_app/tests.py", line 42, in test_user_registration
    self.selenium.get('http://localhost:8000/register/')
  File "/Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 Intro to Software Engineering/cs3300/sprint/djenv/lib/python3.11/site-packages/selenium/webdriver/remote/webdriver.py", line 356, in get
    self.execute(Command.GET, {"url": url})
  File "/Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 Intro to Software Engineering/cs3300/sprint/djenv/lib/python3.11/site-packages/selenium/webdriver/remote/webdriver.py", line 347, in execute
    self.error_handler.check_response(response)
  File "/Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 Intro to Software Engineering/cs3300/sprint/djenv/lib/python3.11/site-packages/selenium/webdriver/remote/errorhandler.py", line 229, in check_response
    raise exception_class(message, screen, stacktrace)
selenium.common.exceptions.WebDriverException: Message: unknown error: net::ERR_CONNECTION_REFUSED
(Session info: chrome=124.0.6367.91)
Stacktrace:
0  chromedriver          0x00000000102b728cc chromedriver + 436858
8
1  chromedriver          0x00000000102b6ad60 chromedriver + 433699
2
2  chromedriver          0x0000000010278ec04 chromedriver + 289796
3  chromedriver          0x00000000102787af8 chromedriver + 260856
4  chromedriver          0x000000001027795e0 chromedriver + 202208
5  chromedriver          0x0000000010277a8d4 chromedriver + 207060
6  chromedriver          0x000000001027798fc chromedriver + 203004
7  chromedriver          0x00000000102779294 chromedriver + 201364
8  chromedriver          0x00000000102778f5c chromedriver + 200540
9  chromedriver          0x00000000102776d20 chromedriver + 191776
10 chromedriver          0x000000001027776bc chromedriver + 194236
11 chromedriver          0x0000000010279116c chromedriver + 299372
12 chromedriver          0x00000000102809c08 chromedriver + 793608
13 chromedriver          0x000000001028095ec chromedriver + 792044
```

```
12 chromedriver          0x00000000102809c08 chromedriver + 793608
13 chromedriver          0x000000001028095ec chromedriver + 792044
14 chromedriver          0x000000001027c5ab4 chromedriver + 514740
15 chromedriver          0x000000001027c650c chromedriver + 517388
16 chromedriver          0x00000000102b36de8 chromedriver + 412413
17
18 chromedriver          0x00000000102b3bbdb8 chromedriver + 414408
19 chromedriver          0x00000000102b1c7b0 chromedriver + 401604
20 chromedriver          0x00000000102b3c508 chromedriver + 414644
21 chromedriver          0x00000000102b0e264 chromedriver + 395734
22 chromedriver          0x00000000102b5be50 chromedriver + 427579
23 chromedriver          0x00000000102b5bfcc chromedriver + 427617
24 libsystem_pthread.dylib 0x000000001978f3fa8 _pthread_start + 148
25 libsystem_pthread.dylib 0x000000001978eeda0 thread_start + 8

=====
ERROR: test_admin_dashboard_view (care_app.tests.ViewTestCase.test_admin_dashboard_view)
-----
Traceback (most recent call last):
  File "/Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 Intro to Software Engineering/cs3300/sprint/care_app/tests.py", line 32, in test_admin_dashboard_view
    url = reverse('admin_dashboard')
          ^^^^^^^^^^^^^^^^^^^^^^
  File "/Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 Intro to Software Engineering/cs3300/sprint/djvenv/lib/python3.11/site-packages/django/urls/base.py", line 88, in reverse
    return resolver._reverse_with_prefix(view, prefix, *args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^^
  File "/Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 Intro to Software Engineering/cs3300/sprint/djvenv/lib/python3.11/site-packages/django/urls/resolvers.py", line 828, in _reverse_with_prefix
    raise NoReverseMatch(msg)
django.urls.exceptions.NoReverseMatch: Reverse for 'admin_dashboard' not found. 'admin_dashboard' is not a valid view function or pattern name.

=====
FAIL: test_user_register_form (care_app.tests.FormTestCase.test_user_register_form)
-----
Traceback (most recent call last):
  File "/Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 Intro to Software Engineering/cs3300/sprint/care_app/tests.py", line 28, in test_user_register_form
    self.assertTrue(form.is_valid())
AssertionError: False is not true

-----
Ran 4 tests in 18.522s

FAILED (failures=1, errors=2)
Destroying test database for alias 'default' ...
```

When I looked up further information, I created a second tab where I am running the server in one, then testing in the other. Prior to doing this, I stopped running the server then entered `python manage.py test`.

That got rid of a good amount of the errors!

test.py:

```
from django.test import TestCase
from django.urls import reverse
from selenium import webdriver
from django.contrib.staticfiles.testing import StaticLiveServerTestCase
from selenium.webdriver.common.by import By

from .models import CustomUser, Patient
from .forms import UserRegisterForm
```

```
class ModelTestCase(TestCase):
    def setUp(self):
        self.user = CustomUser.objects.create(username='testuser', role='admin')

    def test_user_string_representation(self):
        self.assertEqual(str(self.user), 'testuser')

    def test_user_creation(self):
        self.assertEqual(self.user.username, 'testuser')
        self.assertEqual(self.user.role, 'admin')


class FormTestCase(TestCase):
    def test_user_register_form(self):
        form_data = {
            'username': 'testuser',
            'email': 'test@example.com',
            'password1': 'password123',
            'password2': 'password123',
            'role': 'admin'
        }
        form = UserRegisterForm(data=form_data)
        self.assertTrue(form.is_valid())


class ViewTestCase(TestCase):
    def test_admin_dashboard_view(self):
        url = reverse('admin_dashboard')
        response = self.client.get(url)
        self.assertEqual(response.status_code, 200)

    def test_user_login_view(self):
        url = reverse('login')
        response = self.client.get(url)
        self.assertEqual(response.status_code, 200)

    def test_user_logout_view(self):
        url = reverse('logout')
        response = self.client.get(url)
        self.assertEqual(response.status_code, 200)


class IntegrationTestCase(StaticLiveServerTestCase):
    @classmethod
    def setUpClass(cls):
        super().setUpClass()
        cls.selenium = webdriver.Chrome()
```

```
cls.selenium.implicitly_wait(10)

@classmethod
def tearDownClass(cls):
    cls.selenium.quit()
    super().tearDownClass()

def test_user_registration(self):
    # Go to registration page
    self.selenium.get(self.live_server_url + '/register/')

    # Find the username input and input testuser
    username_input = self.selenium.find_element(By.NAME, 'username')
    username_input.send_keys('testuser')

def test_user_login(self):
    # Go to login page
    self.selenium.get(self.live_server_url + '/login/')

    # Find the username and password inputs and fill them
    username_input = self.selenium.find_element(By.NAME, 'username')
    password_input = self.selenium.find_element(By.NAME, 'password')
    username_input.send_keys('testuser')
    password_input.send_keys('password123')

    # Submit the form
    self.selenium.find_element(By.XPATH, "//button[text()='Login']").click()

def test_user_logout(self):
    # Log in first
    self.selenium.get(self.live_server_url + '/login/')
    username_input = self.selenium.find_element(By.NAME, 'username')
    password_input = self.selenium.find_element(By.NAME, 'password')
    username_input.send_keys('testuser')
    password_input.send_keys('password123')
    self.selenium.find_element(By.XPATH, "//button[text()='Login']").click()

    # Go to logout page
    self.selenium.get(self.live_server_url + '/logout/')

def test_update_patient_name(self):
    # Create a patient object
    patient = Patient.objects.create(name='Test Patient',
email='test@example.com', phone='1234567890',
                                      dob='2000-01-01')

    # Go to the patient detail page
    self.selenium.get(self.live_server_url + f'/patient/{patient.id}/')
```

```
# Find the edit button and click it
edit_button = self.selenium.find_element(By.XPATH,
"//button[text()='Edit']")
edit_button.click()

# Find the input field for updating the name and fill it
name_input = self.selenium.find_element(By.NAME, 'name')
name_input.clear()
name_input.send_keys('Updated Name')

# Find the save button and click it
save_button = self.selenium.find_element(By.XPATH,
"//button[text()='Save']")
save_button.click()

# Check if the patient's name is updated
updated_patient = Patient.objects.get(id=patient.id)
self.assertEqual(updated_patient.name, 'Updated Name')
```

- [Testing in Django Examples](#)

Using this test.py :

```
(djvenv) (base) naomirodriguez@Naomis-MacBook-Air sprint % python manage.py test
Found 10 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
F..F..E...
=====
ERROR: test_update_patient_name (care_app.tests.IntegrationTestCase.test_update_
patient_name)
-----
Traceback (most recent call last):
  File "/Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 In
tro to Software Engineering/cs3300/sprint/care_app/tests.py", line 107, in test_
update_patient_name
    edit_button = self.selenium.find_element(By.XPATH, "//button[text()='Edit']")
  )
  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^
  File "/Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 In
tro to Software Engineering/cs3300/sprint/djvenv/lib/python3.11/site-packages/se
lenium/webdriver/remote/webdriver.py", line 741, in find_element
    return self.execute(Command.FIND_ELEMENT, {"using": by, "value": value})["va
lue"]
  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 In
tro to Software Engineering/cs3300/sprint/djvenv/lib/python3.11/site-packages/se
lenium/webdriver/remote/webdriver.py", line 347, in execute
    self.error_handler.check_response(response)
  File "/Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 In
tro to Software Engineering/cs3300/sprint/djvenv/lib/python3.11/site-packages/se
lenium/webdriver/remote/errorhandler.py", line 229, in check_response
    raise exception_class(message, screen, stacktrace)
selenium.common.exceptions NoSuchElementException: Message: no such element: Una
ble to locate element: {"method": "xpath", "selector": "//button[text()='Edit']"}
(Session info: chrome=124.0.6367.91); For documentation on this error, please
visit: https://www.selenium.dev/documentation/webdriver/troubleshooting/errors#n
o-such-element-exception
Stacktrace:
0  chromedriver                      0x0000000010252a8cc chromedriver + 436858
8
1  chromedriver                      0x00000000102522d60 chromedriver + 433699
2
2  chromedriver                      0x00000000102146c04 chromedriver + 289796
3  chromedriver                      0x00000000102188e00 chromedriver + 560640
4  chromedriver                      0x000000001021c15ec chromedriver + 792044
5  chromedriver                      0x0000000010217dab4 chromedriver + 514740
6  chromedriver                      0x0000000010217e50c chromedriver + 517388
7  chromedriver                      0x000000001024eede8 chromedriver + 412413
6
8  chromedriver                      0x000000001024f3bd8 chromedriver + 414408
8
9  chromedriver                      0x000000001024d47b0 chromedriver + 401604
8
10 chromedriver                      0x000000001024f4508 chromedriver + 414644
0
11 chromedriver                      0x000000001024c6264 chromedriver + 395734
8
```

```
12 chromedriver          0x0000000102513e50 chromedriver + 427579
2
13 chromedriver          0x0000000102513fcc chromedriver + 427617
2
14 chromedriver          0x00000001025229c0 chromedriver + 433606
4
15 libsystem_pthread.dylib 0x00000001978f3fa8 _pthread_start + 148
16 libsystem_pthread.dylib 0x00000001978eeda0 thread_start + 8

=====
FAIL: test_user_register_form (care_app.tests.FormTestCase.test_user_register_form)
-----
Traceback (most recent call last):
  File "/Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 Intro to Software Engineering/cs3300/sprint/care_app/tests.py", line 33, in test_user_register_form
    self.assertTrue(form.is_valid())
AssertionError: False is not true

=====
FAIL: test_admin_dashboard_view (care_app.tests.ViewTestCase.test_admin_dashboard_view)
-----
Traceback (most recent call last):
  File "/Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 Intro to Software Engineering/cs3300/sprint/care_app/tests.py", line 40, in test_admin_dashboard_view
    self.assertEqual(response.status_code, 200)
AssertionError: 302 != 200

-----
Ran 10 tests in 13.439s

FAILED (failures=2, errors=1)
Destroying test database for alias 'default'...
```

To help me debug, I entered this error into chatgpt and asked for it to help me understand the errors. These were offered as a starting point:

To resolve these issues, you should:

- ▶ Review and update the XPath for the "Edit" button in the `test\_update\_patient\_name` test case.
- ▶ Debug the `UserRegisterForm` validation to understand why it is failing in the `test\_user\_register\_form` test case.
- ▶ Investigate the redirection issue in the admin dashboard view and adjust the test case or view implementation as needed.

```
(djvenv) (base) naomirodriguez@Naomis-MacBook-Air sprint % python manage.py test
Found 10 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
F..F.....
=====
FAIL: test_user_register_form (care_app.tests.FormTestCase.test_user_register_form)
-----
Traceback (most recent call last):
  File "/Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 Intro to Software Engineering/cs3300/sprint/care_app/tests.py", line 33, in test_user_register_form
    self.assertTrue(form.is_valid())
AssertionError: False is not true
=====
FAIL: test_admin_dashboard_view (care_app.tests.ViewTestCase.test_admin_dashboard_view)
-----
Traceback (most recent call last):
  File "/Users/naomirodriguez/Documents/Past UCCS Classes/Spring 2024/CS 3300 Intro to Software Engineering/cs3300/sprint/care_app/tests.py", line 40, in test_admin_dashboard_view
    self.assertEqual(response.status_code, 200)
AssertionError: 302 != 200
-----
Ran 10 tests in 2.859s
FAILED (failures=2)
Destroying test database for alias 'default'...
```

I tried building some user tests of all kinds but I know my user\_authentication isn't working correctly, so I created the following model test instead:

```
class ModelTestCase(TestCase):
    def setUp(self):
        self.user = CustomUser.objects.create(username='testuser',
email='test@example.com', role='admin')

    def test_user_string_representation(self):
        self.assertEqual(str(self.user), 'testuser')

    def test_user_creation(self):
        self.assertEqual(self.user.username, 'testuser')
        self.assertEqual(self.user.role, 'admin')

    def test_user_email(self):
```

```
user = CustomUser.objects.get(username='testuser')
self.assertEqual(user.email, 'test@example.com')
```

I kept getting an error on the `test_user_register_form`, so I looked back at my registration page. One of my requirements was the following:

- Your password can't be a commonly used password.

I changed the form test to the following:

```
class FormTestCase(TestCase):
    def test_user_register_form(self):
        form_data = {
            'username': 'testuser',
            'email': 'test@example.com',
            'password1': 'passwordToTest',
            'password2': 'passwordToTest',
            'role': 'admin'
        }
        form = UserRegisterForm(data=form_data)
        self.assertTrue(form.is_valid())
```

```
(djvenv) (base) naomirodriguez@Naomis-MacBook-Air sprint % python manage.py test
Found 10 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
Ran 10 tests in 2.892s

OK
Destroying test database for alias 'default'...
(djvenv) (base) naomirodriguez@Naomis-MacBook-Air sprint %
```

They're all working!!