# Views.py

# Function to load pickle file

def load\_pickle\_file(file\_path):

    with open(file\_path, 'rb') as f:

        data = pickle.load(f)

    return data

# Function to convert images to Base64 for HTML display

def convert\_images\_to\_base64(data):

    images\_base64 = []

    for i in range(min(10, len(data))):  # Limit to first 10 images

        image = data[i]

        # Convert image to PNG format

        fig, ax = plt.subplots()

        ax.imshow(image)

        ax.axis('off')  # Hide axis

        # Save the image in memory buffer

        buf = io.BytesIO()

        plt.savefig(buf, format="png", bbox\_inches='tight', pad\_inches=0)

        plt.close(fig)

        # Convert to base64

        image\_base64 = base64.b64encode(buf.getvalue()).decode('utf-8')

        images\_base64.append(image\_base64)

    return images\_base64

# View function for demographics page

def demographics\_view(request):

    file\_path = "visits/X\_9\_RGB.pickle"  # Update with actual path

    data = load\_pickle\_file(file\_path)

    # Convert image data to base64 for HTML display

    images\_base64 = convert\_images\_to\_base64(data)

    return render(request, "users/dashboard\_demographics.html", {"images": images\_base64})

# urls.py

    path('demographics/', views.dashboard\_demographics, name='dashboard\_demographics'),

# models.py

def visit\_file\_path(instance, filename):

    """Rename file to match format: visitID-fileNumber.pickle"""

    visit\_id = f"{instance.visit.visitID:03d}"  # Format as 3-digit (e.g., 005)

    # Count existing files for this visit to determine file number

    file\_count = VisitData.objects.filter(visit=instance.visit).count() + 1

    new\_filename = f"{visit\_id}-{file\_count:02d}.pickle"  # 005-01.pickle, 005-02.pickle, etc.

    return os.path.join('visit\_data/', new\_filename)

class VisitData(models.Model):

    visit = models.ForeignKey(Visit, on\_delete=models.CASCADE, related\_name='visit\_files')

    file = models.FileField(upload\_to=visit\_file\_path)  # Custom upload path

    uploaded\_at = models.DateTimeField(auto\_now\_add=True)

# Dashboard\_demograpahics

{% extends 'base.html' %}

{% block title %}Demographics{% endblock %}

{% block content %}

<h1>Demographics</h1>

<p>Content for demographics page.</p>

<h2>Pickle File Images</h2>

<div style="display: flex; flex-wrap: wrap; gap: 10px;">

    {% for image in images %}

        <div>

            <img src="data:image/png;base64,{{ image }}" alt="Pickle Image" width="200">

        </div>

    {% endfor %}

</div>

{% endblock %}