Okay, this is a perfect next step! Creating a dedicated user\_dashboard will significantly improve the experience for your normal users/stakeholders, allowing them to track all their submitted requests in one place.

This will be distinct from the support\_dashboard (which is for staff/admins) and also more comprehensive than a single my\_complaints page, as it will unify all four request types (Complaint, ServiceRequest, Inquiry, Emergency).

Here's the plan and implementation:

### ****Step 1: Create the**** user\_dashboard ****App****

Open your terminal in your project's root directory (where manage.py is) and run:

Bash

python manage.py startapp user\_dashboard

### ****Step 2: Add the New App to**** settings.py

Open your\_project/settings.py and add 'user\_dashboard' to your INSTALLED\_APPS list:

Python

# your\_project/settings.py

INSTALLED\_APPS = [

# ... existing apps ...

'user\_dashboard', # Add this line

# ...

]

### ****Step 3: Update**** accounts/urls.py ****(If you want a common profile landing page)****

You mentioned creating a user profile page in the accounts app. It might be a good idea to rename the accounts:profile URL to accounts:user\_dashboard\_landing or something similar and have it serve as a general user landing page that then links to the requests list. Or we can just add the link directly to user\_dashboard:request\_list in the base template. Let's stick to the latter for now, as it's cleaner.

### ****Step 4: Create Views for User Request Management (****user\_dashboard/views.py****)****

This is the trickiest part because you're querying multiple models. We'll fetch requests from all four types, combine them, and then filter by the logged-in user.

Python

# user\_dashboard/views.py

from django.shortcuts import render, get\_object\_or\_404

from django.contrib.auth.decorators import login\_required

from django.core.paginator import Paginator

from django.db.models import Q # For search

# Import all your request models

from complaints.models import Complaint # Adjust as needed

from services.models import ServiceRequest # Adjust as needed

from inquiries.models import Inquiry # Adjust as needed

from emergencies.models import Emergency # Adjust as needed

@login\_required

def user\_request\_list(request):

user = request.user

all\_user\_requests = []

# Query requests from each model submitted by the current user

complaints = Complaint.objects.filter(submitted\_by=user).select\_related('category')

service\_requests = ServiceRequest.objects.filter(submitted\_by=user).select\_related('service\_type') # Assumed service\_type foreign key

inquiries = Inquiry.objects.filter(submitted\_by=user).select\_related('inquiry\_category') # Assumed inquiry\_category foreign key

emergencies = Emergency.objects.filter(submitted\_by=user).select\_related('emergency\_type') # Assumed emergency\_type foreign key

# Combine and standardize common fields for display

# Ensure all models have 'subject', 'description', 'created\_at', 'status', 'request\_type\_slug'

for req in complaints:

all\_user\_requests.append({

'pk': req.pk,

'request\_type\_slug': 'complaint',

'subject': req.subject,

'description': req.description,

'status': req.status,

'created\_at': req.created\_at,

'display\_type': 'Complaint',

'specific\_field': req.category.name if hasattr(req, 'category') and req.category else 'N/A',

'full\_object': req, # Keep reference to original object for detail view if needed

})

for req in service\_requests:

all\_user\_requests.append({

'pk': req.pk,

'request\_type\_slug': 'service\_request',

'subject': req.subject,

'description': req.description,

'status': req.status,

'created\_at': req.created\_at,

'display\_type': 'Service Request',

'specific\_field': req.service\_type.name if hasattr(req, 'service\_type') and req.service\_type else 'N/A',

'full\_object': req,

})

for req in inquiries:

all\_user\_requests.append({

'pk': req.pk,

'request\_type\_slug': 'inquiry',

'subject': req.subject, # Or req.question, adjust as per your model

'description': req.description, # Or req.question

'status': req.status,

'created\_at': req.created\_at,

'display\_type': 'Inquiry',

'specific\_field': req.inquiry\_category.name if hasattr(req, 'inquiry\_category') and req.inquiry\_category else 'N/A',

'full\_object': req,

})

for req in emergencies:

all\_user\_requests.append({

'pk': req.pk,

'request\_type\_slug': 'emergency',

'subject': req.subject,

'description': req.description,

'status': req.status,

'created\_at': req.created\_at,

'display\_type': 'Emergency Report',

'specific\_field': req.emergency\_type.name if hasattr(req, 'emergency\_type') and req.emergency\_type else 'N/A',

'full\_object': req,

})

# Sort the combined list by creation date (newest first)

all\_user\_requests.sort(key=lambda x: x['created\_at'], reverse=True)

# --- Filtering and Search ---

status\_filter = request.GET.get('status')

if status\_filter and status\_filter != 'all':

all\_user\_requests = [req for req in all\_user\_requests if req['status'] == status\_filter]

type\_filter = request.GET.get('type')

if type\_filter and type\_filter != 'all':

all\_user\_requests = [req for req in all\_user\_requests if req['request\_type\_slug'] == type\_filter]

search\_query = request.GET.get('q')

if search\_query:

search\_query\_lower = search\_query.lower()

all\_user\_requests = [

req for req in all\_user\_requests

if search\_query\_lower in req['subject'].lower() or \

search\_query\_lower in req['description'].lower() or \

search\_query\_lower in req['display\_type'].lower()

]

# --- Pagination ---

paginator = Paginator(all\_user\_requests, 10) # Show 10 requests per page

page\_number = request.GET.get('page')

page\_obj = paginator.get\_page(page\_number)

context = {

'page\_obj': page\_obj,

'status\_filter': status\_filter,

'type\_filter': type\_filter,

'search\_query': search\_query,

'status\_choices': [('new', 'New'), ('in\_progress', 'In Progress'), ('resolved', 'Resolved'), ('closed', 'Closed'), ('rejected', 'Rejected')],

'request\_type\_choices': [

('complaint', 'Complaint'), ('service\_request', 'Service Request'),

('inquiry', 'Inquiry'), ('emergency', 'Emergency Report')

],

}

return render(request, 'user\_dashboard/user\_request\_list.html', context)

@login\_required

def user\_request\_detail(request, request\_type\_slug, pk):

user = request.user

request\_obj = None

# Dynamically determine the model based on request\_type\_slug

if request\_type\_slug == 'complaint':

model = Complaint

elif request\_type\_slug == 'service\_request':

model = ServiceRequest

elif request\_type\_slug == 'inquiry':

model = Inquiry

elif request\_type\_slug == 'emergency':

model = Emergency

else:

# If an invalid slug is provided, raise 404

raise Http404("Invalid request type.")

# Get the specific object, ensuring it belongs to the current user

# This also acts as a security check

request\_obj = get\_object\_or\_404(model, pk=pk, submitted\_by=user)

context = {

'request\_obj': request\_obj,

}

return render(request, 'user\_dashboard/user\_request\_detail.html', context)

**IMPORTANT ASSUMPTIONS FOR VIEWS:**

* All your request models (Complaint, ServiceRequest, Inquiry, Emergency) have a submitted\_by ForeignKey field pointing to settings.AUTH\_USER\_MODEL.
* All have pk, subject, description, status, created\_at, request\_type\_slug fields.
* They also have specific foreign keys like category (Complaint), service\_type (ServiceRequest), inquiry\_category (Inquiry), emergency\_type (Emergency) to allow select\_related. Adjust these if your model fields are named differently.
* status fields across all models use the same status choices for consistency in filtering.

### ****Step 5: Define URLs for the User Dashboard (****user\_dashboard/urls.py****)****

Create a new file urls.py inside your user\_dashboard app directory:

Python

# user\_dashboard/urls.py

from django.urls import path

from . import views

app\_name = 'user\_dashboard' # Important for URL namespacing

urlpatterns = [

path('', views.user\_request\_list, name='request\_list'),

path('<str:request\_type\_slug>/<int:pk>/', views.user\_request\_detail, name='request\_detail'),

]

**Important:** Include these URLs in your **main project's** urls.py:

Python

# your\_project/urls.py

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

path('admin/', admin.site.urls),

path('accounts/', include('allauth.urls')),

path('support-dashboard/', include('support\_dashboard.urls')), # Admin dashboard

path('my-dashboard/', include('user\_dashboard.urls')), # <--- Add this for user dashboard

path('', include('abode.urls')), # Your landing page app

# ... other app URLs ...

]

### ****Step 6: Create Templates for User Dashboard****

Create these files in user\_dashboard/templates/user\_dashboard/:

#### user\_dashboard\_base.html (A base for all user dashboard pages)

HTML

{% extends 'sfrp/sfrp\_base.html' %}

{% load i18n %}

{% block title %}

{% trans "My Dashboard" %} - {% block user\_dashboard\_title %}{% endblock user\_dashboard\_title %}

{% endblock title %}

{% block content %}

<div class="container-fluid mt-4">

<div class="row">

{# Sidebar Navigation for User Dashboard #}

<nav id="sidebarMenu" class="col-md-3 col-lg-2 d-md-block bg-light sidebar collapse">

<div class="position-sticky pt-3">

<h6 class="sidebar-heading d-flex justify-content-between align-items-center px-3 mt-4 mb-1 text-muted">

<span>{% trans "My Requests" %}</span>

</h6>

<ul class="nav flex-column mb-2">

<li class="nav-item">

<a class="nav-link {% if request.resolver\_match.view\_name == 'user\_dashboard:request\_list' %}active{% endif %}" href="{% url 'user\_dashboard:request\_list' %}">

<i class="fas fa-list-alt mr-2"></i> {% trans "All My Requests" %}

</a>

</li>

{# You can add links here for specific request types if desired #}

{# <li class="nav-item">

<a class="nav-link" href="{% url 'user\_dashboard:request\_list' %}?type=complaint">

<i class="fas fa-exclamation-triangle mr-2"></i> Complaints

</a>

</li> #}

{# Add more specific type links as needed #}

</ul>

<h6 class="sidebar-heading d-flex justify-content-between align-items-center px-3 mt-4 mb-1 text-muted">

<span>{% trans "My Account" %}</span>

</h6>

<ul class="nav flex-column mb-2">

<li class="nav-item">

<a class="nav-link" href="{% url 'accounts:profile' %}">

<i class="fas fa-user-circle mr-2"></i> {% trans "Profile Settings" %}

</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'account\_email' %}">

<i class="fas fa-envelope mr-2"></i> {% trans "Email Settings" %}

</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'account\_change\_password' %}">

<i class="fas fa-lock mr-2"></i> {% trans "Change Password" %}

</a>

</li>

</ul>

</div>

</nav>

{# Main Content Area #}

<main class="col-md-9 ml-sm-auto col-lg-10 px-md-4">

<div class="d-flex justify-content-between flex-wrap flex-md-nowrap align-items-center pt-3 pb-2 mb-3 border-bottom">

<h1 class="h2">{% block main\_user\_dashboard\_heading %}{% endblock main\_user\_dashboard\_heading %}</h1>

{% block user\_dashboard\_actions %}{% endblock user\_dashboard\_actions %}

</div>

{% block user\_dashboard\_content %}

{# Specific page content (list/detail views) will go here #}

{% endblock user\_dashboard\_content %}

</main>

</div>

</div>

{% endblock content %}

#### user\_request\_list.html

HTML

{% extends 'user\_dashboard/user\_dashboard\_base.html' %}

{% load i18n widget\_tweaks %}

{% block user\_dashboard\_title %}{% trans "My Submitted Requests" %}{% endblock user\_dashboard\_title %}

{% block main\_user\_dashboard\_heading %}{% trans "All My Submitted Requests" %}{% endblock main\_user\_dashboard\_heading %}

{% block user\_dashboard\_actions %}

<a href="{% url 'unified\_requests:submit\_request' %}" class="btn btn-primary btn-sm">

<i class="fas fa-plus mr-2"></i> {% trans "Submit New Request" %}

</a>

{% endblock user\_dashboard\_actions %}

{% block user\_dashboard\_content %}

<div class="row mb-3 align-items-center">

<div class="col-md-12">

<form method="GET" class="form-inline justify-content-start">

<div class="form-group mr-3 mb-2">

<label for="statusFilter" class="sr-only">{% trans "Filter by Status" %}</label>

<select class="form-control custom-select" id="statusFilter" name="status" onchange="this.form.submit()">

<option value="all" {% if not status\_filter or status\_filter == 'all' %}selected{% endif %}>{% trans "All Statuses" %}</option>

{% for status\_code, status\_display in status\_choices %}

<option value="{{ status\_code }}" {% if status\_filter == status\_code %}selected{% endif %}>{{ status\_display }}</option>

{% endfor %}

</select>

</div>

<div class="form-group mr-3 mb-2">

<label for="typeFilter" class="sr-only">{% trans "Filter by Type" %}</label>

<select class="form-control custom-select" id="typeFilter" name="type" onchange="this.form.submit()">

<option value="all" {% if not type\_filter or type\_filter == 'all' %}selected{% endif %}>{% trans "All Types" %}</option>

{% for type\_code, type\_display in request\_type\_choices %}

<option value="{{ type\_code }}" {% if type\_filter == type\_code %}selected{% endif %}>{{ type\_display }}</option>

{% endfor %}

</select>

</div>

<div class="input-group mb-2">

<input type="text" name="q" class="form-control" placeholder="{% trans "Search by subject or description" %}" value="{{ search\_query|default\_if\_none:'' }}">

<div class="input-group-append">

<button class="btn btn-outline-secondary" type="submit">

<i class="fas fa-search"></i> {% trans "Search" %}

</button>

</div>

</div>

{% if status\_filter or type\_filter or search\_query %}

<a href="{% url 'user\_dashboard:request\_list' %}" class="btn btn-outline-danger ml-2 mb-2">{% trans "Clear Filters" %}</a>

{% endif %}

</form>

</div>

</div>

{% if page\_obj %}

<div class="table-responsive">

<table class="table table-striped table-hover">

<thead>

<tr>

<th>{% trans "ID" %}</th>

<th>{% trans "Type" %}</th>

<th>{% trans "Subject" %}</th>

<th>{% trans "Category/Type" %}</th> {# Dynamic label #}

<th>{% trans "Status" %}</th>

<th>{% trans "Submitted At" %}</th>

<th>{% trans "Actions" %}</th>

</tr>

</thead>

<tbody>

{% for request\_item in page\_obj %}

<tr>

<td>{{ request\_item.pk }}</td>

<td><span class="badge badge-info">{{ request\_item.display\_type }}</span></td>

<td>{{ request\_item.subject|truncatechars:50 }}</td>

<td>{{ request\_item.specific\_field }}</td>

<td>

<span class="badge badge-pill badge-{% if request\_item.status == 'new' %}primary{% elif request\_item.status == 'in\_progress' %}warning{% elif request\_item.status == 'resolved' %}success{% elif request\_item.status == 'closed' %}secondary{% elif request\_item.status == 'rejected' %}danger{% else %}dark{% endif %}">{{ request\_item.full\_object.get\_status\_display }}</span>

</td>

<td>{{ request\_item.created\_at|date:"M d, Y H:i" }}</td>

<td>

<a href="{% url 'user\_dashboard:request\_detail' request\_type\_slug=request\_item.request\_type\_slug pk=request\_item.pk %}" class="btn btn-sm btn-info">

<i class="fas fa-eye"></i> {% trans "View" %}

</a>

</td>

</tr>

{% endfor %}

</tbody>

</table>

</div>

{# Pagination #}

<nav aria-label="Page navigation">

<ul class="pagination justify-content-center">

{% if page\_obj.has\_previous %}

<li class="page-item"><a class="page-link" href="?page=1{% if status\_filter %}&status={{ status\_filter }}{% endif %}{% if type\_filter %}&type={{ type\_filter }}{% endif %}{% if search\_query %}&q={{ search\_query }}{% endif %}">{% trans "First" %}</a></li>

<li class="page-item"><a class="page-link" href="?page={{ page\_obj.previous\_page\_number }}{% if status\_filter %}&status={{ status\_filter }}{% endif %}{% if type\_filter %}&type={{ type\_filter }}{% endif %}{% if search\_query %}&q={{ search\_query }}{% endif %}">{% trans "Previous" %}</a></li>

{% endif %}

{% for num in page\_obj.paginator.page\_range %}

{% if page\_obj.number == num %}

<li class="page-item active"><span class="page-link">{{ num }}</span></li>

{% elif num > page\_obj.number|add:'-3' and num < page\_obj.number|add:'3' %}

<li class="page-item"><a class="page-link" href="?page={{ num }}{% if status\_filter %}&status={{ status\_filter }}{% endif %}{% if type\_filter %}&type={{ type\_filter }}{% endif %}{% if search\_query %}&q={{ search\_query }}{% endif %}">{{ num }}</a></li>

{% endif %}

{% endfor %}

{% if page\_obj.has\_next %}

<li class="page-item"><a class="page-link" href="?page={{ page\_obj.next\_page\_number }}{% if status\_filter %}&status={{ status\_filter }}{% endif %}{% if type\_filter %}&type={{ type\_filter }}{% endif %}{% if search\_query %}&q={{ search\_query }}{% endif %}">{% trans "Next" %}</a></li>

<li class="page-item"><a class="page-link" href="?page={{ page\_obj.paginator.num\_pages }}{% if status\_filter %}&status={{ status\_filter }}{% endif %}{% if type\_filter %}&type={{ type\_filter }}{% endif %}{% if search\_query %}&q={{ search\_query }}{% endif %}">{% trans "Last" %}</a></li>

{% endif %}

</ul>

</nav>

{% else %}

<div class="alert alert-info text-center" role="alert">

{% trans "You have not submitted any requests yet." %}

</div>

{% endif %}

{% endblock user\_dashboard\_content %}

#### user\_request\_detail.html

HTML

{% extends 'user\_dashboard/user\_dashboard\_base.html' %}

{% load i18n %}

{% block user\_dashboard\_title %}

{{ request\_obj.request\_type\_slug|title|replace\_chars:"\_"," " }} #{{ request\_obj.pk }}

{% endblock user\_dashboard\_title %}

{% block main\_user\_dashboard\_heading %}

{{ request\_obj.request\_type\_slug|title|replace\_chars:"\_"," " }} #{{ request\_obj.pk }}

{% endblock main\_user\_dashboard\_heading %}

{% block user\_dashboard\_actions %}

<a href="{% url 'user\_dashboard:request\_list' %}" class="btn btn-secondary btn-sm">

<i class="fas fa-arrow-left mr-2"></i> {% trans "Back to All My Requests" %}

</a>

{% endblock user\_dashboard\_actions %}

{% block user\_dashboard\_content %}

<div class="card mb-4">

<div class="card-header bg-primary text-white">

<h5 class="mb-0">{{ request\_obj.subject }}</h5>

</div>

<div class="card-body">

<div class="row">

<div class="col-md-8">

<p><strong>{% trans "Description:" %}</strong></p>

<p>{{ request\_obj.description }}</p>

{# Type-Specific Details - Re-use the logic from Admin Email Template #}

{% if request\_obj.request\_type\_slug == 'complaint' %}

<p><strong>{% trans "Category:" %}</strong> {{ request\_obj.category.name|default:"N/A" }}</p>

{% elif request\_obj.request\_type\_slug == 'service\_request' %}

<p><strong>{% trans "Service Type:" %}</strong> {{ request\_obj.get\_service\_type\_display|default:"N/A" }}</p>

{% if request\_obj.preferred\_datetime %}<p><strong>{% trans "Preferred Date/Time:" %}</strong> {{ request\_obj.preferred\_datetime|date:"M d, Y H:i A" }}</p>{% endif %}

{% if request\_obj.location\_details %}<p><strong>{% trans "Location Details:" %}</strong> {{ request\_obj.location\_details }}</p>{% endif %}

{% elif request\_obj.request\_type\_slug == 'inquiry' %}

<p><strong>{% trans "Inquiry Category:" %}</strong> {{ request\_obj.get\_inquiry\_category\_display|default:"N/A" }}</p>

<p><strong>{% trans "Urgency Level:" %}</strong> {{ request\_obj.get\_urgency\_level\_display|default:"N/A" }}</p>

{% elif request\_obj.request\_type\_slug == 'emergency' %}

<p><strong>{% trans "Emergency Type:" %}</strong> {{ request\_obj.get\_emergency\_type\_display|default:"N/A" }}</p>

<p><strong>{% trans "Impact Level:" %}</strong> {{ request\_obj.get\_impact\_level\_display|default:"N/A" }}</p>

{% if request\_obj.contact\_during\_emergency %}<p><strong>{% trans "Contact During Emergency:" %}</strong> {{ request\_obj.contact\_during\_emergency }}</p>{% endif %}

{% endif %}

{# You can add display for attachments here if applicable #}

{# {% if request\_obj.attachments %} #}

{# <p><strong>Attachments:</strong> <a href="{{ request\_obj.attachments.url }}" target="\_blank">View Attachment</a></p> #}

{# {% endif %} #}

</div>

<div class="col-md-4">

<p><strong>{% trans "Submitted By:" %}</strong>

{% if request\_obj.submitted\_by %}

{{ request\_obj.submitted\_by.get\_full\_name|default:request\_obj.submitted\_by.username }}

{% else %}

Anonymous

{% if request\_obj.full\_name %} ({{ request\_obj.full\_name }}){% endif %}

{% if request\_obj.email %} ({{ request\_obj.email }}){% endif %}

{% endif %}

</p>

<p><strong>{% trans "Current Status:" %}</strong> <span class="badge badge-pill badge-{% if request\_obj.status == 'new' %}primary{% elif request\_obj.status == 'in\_progress' %}warning{% elif request\_obj.status == 'resolved' %}success{% elif request\_obj.status == 'closed' %}secondary{% elif request\_obj.status == 'rejected' %}danger{% else %}dark{% endif %}">{{ request\_obj.get\_status\_display }}</span></p>

<p><strong>{% trans "Assigned To:" %}</strong> {{ request\_obj.assigned\_to.username|default:"N/A" }}</p>

<p><strong>{% trans "Submitted On:" %}</strong> {{ request\_obj.created\_at|date:"M d, Y H:i" }}</p>

<p><strong>{% trans "Last Updated:" %}</strong> {{ request\_obj.updated\_at|date:"M d, Y H:i" }}</p>

</div>

</div>

</div>

<div class="card-footer text-right">

{# Add any user-specific actions here, e.g., "Cancel Request" (requires form/view logic) #}

{# <button type="button" class="btn btn-outline-danger">Cancel Request</button> #}

</div>

</div>

{% endblock user\_dashboard\_content %}

### ****Step 7: Update Navigation in**** sfrp/sfrp\_base.html

You already have a "My Request" link. Let's update it to point to this new unified user dashboard list.

**Find this section:**

HTML

<li class="nav-item">

<a class="nav-link" href="{% url 'complaints:my\_complaints' %}">My Request</a>

</li>

**Change it to:**

HTML

<li class="nav-item">

<a class="nav-link" href="{% url 'user\_dashboard:request\_list' %}">My Requests</a> {# Updated to plural and new URL #}

</li>

This new link will now take the user to their unified request list.

### ****Final Checklist Before Testing:****

1. user\_dashboard **App in** INSTALLED\_APPS**:** Verified in Step 2.
2. user\_dashboard/views.py **created:** Verified in Step 4.
3. user\_dashboard/urls.py **created:** Verified in Step 5.
4. **Main Project** urls.py **updated:** Ensure path('my-dashboard/', include('user\_dashboard.urls')) is present.
5. **Template Directory Structure:** Make sure user\_dashboard/templates/user\_dashboard/ exists and contains user\_dashboard\_base.html, user\_request\_list.html, and user\_request\_detail.html.
6. **Model Fields Consistency:** Crucially, ensure all your request models (Complaint, ServiceRequest, Inquiry, Emergency) have the following:
   * submitted\_by (ForeignKey to User)
   * subject (CharField)
   * description (TextField)
   * status (CharField with choices, with a get\_status\_display() method)
   * created\_at (DateTimeField)
   * updated\_at (DateTimeField)
   * request\_type\_slug (CharField, set in save method as we did)
   * And their respective unique ForeignKeys (category, service\_type, inquiry\_category, emergency\_type).
7. Http404 **Import:** Add from django.http import Http404 at the top of user\_dashboard/views.py.
8. notifications\_extras **filter:** Ensure replace\_chars is correctly loaded/defined for user\_request\_detail.html (it should be if you fixed the previous error).

After verifying these points, **restart your Django development server** and log in as a regular user who has submitted requests. Navigate to /my-dashboard/ (or click "My Requests" in the navbar), and you should see their comprehensive request list!