

Pour restreindre l'accès aux pages en fonction des rôles des utilisateurs, nous devons utiliser le système d'authentification de Django pour déterminer le rôle de chaque utilisateur connecté et afficher les liens correspondants dans la barre de navigation. Nous allons modifier le template `base.html` pour inclure des vérifications de rôle, ce qui permettra de personnaliser l'affichage de la navigation.

Modifications du Template `base.html`

Voici le code modifié du template `base.html` avec des conditions basées sur les rôles des utilisateurs :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>HopitalApp</title>
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
  <div class="container mt-4">
    <nav class="navbar navbar-expand-lg navbar-light bg-light mb-4">
      <a class="navbar-brand" href="#">HopitalApp</a>
      <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">

        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">

          <!-- Liens visibles par tous les utilisateurs -->
          <li class="nav-item">
            <a class="nav-link" href="{% url 'liste_patients' %}">Patients</a>
          </li>

          <!-- Liens spécifiques à l'Administrateur -->
          {% if user.is_authenticated and user.role == 'ADMIN' %}
          <li class="nav-item">
            <a class="nav-link" href="{% url 'admin:index' %}">Administration</a>
          </li>
          {% endif %}

          <!-- Liens spécifiques à l'Infirmier -->
          {% if user.is_authenticated and user.role == 'INFIRMIER' %}
          <li class="nav-item">
            <a class="nav-link" href="{% url 'liste_mesures' %}">Infirmierie</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="{% url 'attribuer_chambre' %}">Attribution
Chambre</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="{% url 'enregistrer_rendezvous' %}">Rendez-
vous</a>
          </li>
          {% endif %}
        </ul>
      </div>
    </nav>
  </div>
</body>
</html>
```

```

<!-- Liens spécifiques au Médecin -->
{% if user.is_authenticated and user.role == 'MEDECIN' %}
<li class="nav-item">
    <a class="nav-link" href="{% url 'liste_rendezvous' %}">Mes Rendez-
vous</a>

</li>
<li class="nav-item">
    <a class="nav-link" href="{% url 'liste_prescriptions'
%}">Prescriptions</a>

</li>
{% endif %}

<!-- Liens spécifiques au Caissier -->
{% if user.is_authenticated and user.role == 'CAISSIER' %}
<li class="nav-item">
    <a class="nav-link" href="{% url 'liste_factures' %}">Factures</a>
</li>
<li class="nav-item">
    <a class="nav-link" href="{% url 'liste_analyses' %}">Analyses à
Payer</a>

</li>
<li class="nav-item">
    <a class="nav-link" href="{% url 'analyse_financiere' %}">Analyse
Financière</a>

</li>
{% endif %}

<!-- Liens spécifiques au Laborantin -->
{% if user.is_authenticated and user.role == 'LABORANTIN' %}
<li class="nav-item">
    <a class="nav-link" href="{% url 'liste_analyses' %}">Analyses</a>
</li>
<li class="nav-item">
    <a class="nav-link" href="{% url 'enregistrer_analyse' %}">Ajouter
Analyse</a>

</li>
{% endif %}

<!-- Liens spécifiques au Pharmacien -->
{% if user.is_authenticated and user.role == 'PHARMACIEN' %}
<li class="nav-item">
    <a class="nav-link" href="{% url 'liste_prescriptions'
%}">Prescriptions</a>

</li>
<li class="nav-item">
    <a class="nav-link" href="{% url 'enregistrer_prescription' %}">Ajout
Médicament</a>

</li>
{% endif %}

<!-- Lien de déconnexion visible par tous les utilisateurs authentifiés -->
{% if user.is_authenticated %}
<li class="nav-item">
    <a class="nav-link" href="{% url 'logout' %}">Déconnexion</a>
</li>
{% else %}
<li class="nav-item">
    <a class="nav-link" href="{% url 'login' %}">Connexion</a>

```

```

        </li>
        {% endif %}
    </ul>
</div>
</nav>

    {% block content %}
    {% endblock %}
</div>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.3/dist/umd/popper.min.js">
</script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js">
</script>
</body>
</html>

```

Explications des Modifications

1. Vérification du Rôle de l'Utilisateur :

- Les balises `{% if user.is_authenticated and user.role == '<ROLE>' %}` sont utilisées pour vérifier si l'utilisateur est connecté et a un rôle spécifique.
- Les rôles pris en compte sont `ADMIN`, `INFIRMIER`, `MEDECIN`, `CAISSIER`, `LABORANTIN`, et `PHARMACIEN`.

2. Liens de Navigation :

- Chaque utilisateur voit les liens de navigation pertinents à son rôle.
- L'administrateur a accès à l'administration de Django.
- Les infirmiers peuvent accéder aux sections de patients, infirmerie, et chambres.
- Les médecins peuvent accéder à leurs rendez-vous et prescriptions.
- Les caissiers peuvent gérer les factures et les analyses à payer.
- Les laborantins peuvent gérer les analyses.
- Les pharmaciens peuvent gérer les prescriptions et les médicaments.

3. Affichage Conditionnel :

- Les utilisateurs non authentifiés voient uniquement le lien de connexion.
- Une fois authentifié, le lien de déconnexion est visible.

Protection des Vues avec des Décorateurs

Pour renforcer la sécurité et l'intégrité de votre application, vous devez également restreindre l'accès aux vues spécifiques en fonction des rôles. Cela peut être fait en ajoutant des décorateurs aux vues concernées.

Exemple d'une Vue Restreinte

Supposons que nous voulons restreindre l'accès à la vue des prescriptions uniquement aux médecins :

```

from django.contrib.auth.decorators import user_passes_test
from django.utils.decorators import method_decorator
from django.views import View
from django.shortcuts import render

# Fonction pour vérifier le rôle
def is_medecin(user):
    return user.is_authenticated and user.role == 'MEDECIN'

```

```

@method_decorator(user_passes_test(is_medecin), name='dispatch')
class ListePrescriptions(View):
    def get(self, request):
        prescriptions = Prescription.objects.all()
        return render(request, 'pharmacie/liste_prescriptions.html', {'prescriptions':
prescriptions})

```

Dans cet exemple, la fonction `is_medecin` vérifie que l'utilisateur est authentifié et a le rôle `MEDECIN`. Le décorateur `@method_decorator(user_passes_test(is_medecin), name='dispatch')` est utilisé pour restreindre l'accès à cette vue.

Ajout des Tests Unitaires

Pour assurer le bon fonctionnement de votre application, il est essentiel d'ajouter des tests unitaires. Voici un exemple de test pour vérifier que l'accès est bien restreint aux rôles appropriés :

Exemple de Test

```

from django.test import TestCase, Client
from django.urls import reverse
from auth.models import CustomUser

class MedecinAccessTest(TestCase):
    def setUp(self):
        self.client = Client()
        self.medecin_user = CustomUser.objects.create_user(username='medecin',
password='password', role='MEDECIN')

    def test_medecin_access(self):
        # Test accès sans authentication
        response = self.client.get(reverse('liste_prescriptions'))
        self.assertEqual(response.status_code, 302) # Redirection vers la page de connexion

        # Test accès avec authentication
        self.client.login(username='medecin', password='password')
        response = self.client.get(reverse('liste_prescriptions'))
        self.assertEqual(response.status_code, 200) # Accès autorisé

```

Ce test assure que :

- Les utilisateurs non authentifiés sont redirigés vers la page de connexion.
- Les médecins authentifiés ont accès à la page des prescriptions.

Conclusion