PYSTACK WEEK 3.0 | AULA 02

Link direto para o Notion:

```
https://grizzly-amaranthus-f6a.notion.site/PYSTACK-WEEK-3-0-AULA-02-c37498c750164bc3a9428058fda445f8
```

Vamos começar com o reset de senha

Primeiro vamos realizar a configuração de envio de e-mails:

```
EMAIL_BACKEND = 'django.core.mail.backends.console.EmailBackend'
```

Crie as URL's:

```
path('password_reset/', auth_views.PasswordResetView.as_view(template_name="password_reset.html"), name="password_reset"),
path('password_reset_done/', auth_views.PasswordResetDoneView.as_view(template_name="password_reset_done.html"), name="password_reset_done"),
path('reset/<uidb64>/<token>/', auth_views.PasswordResetConfirmView.as_view(template_name="password_reset_confirm_view.html"), name="password_reset_confirm"),
path('password_reset_complete/', auth_views.PasswordResetCompleteView.as_view(template_name="password_reset_complete.html"), name="password_reset_complete"),
```

Crie todos os templates:

password_reset.html

```
<form action="{% url 'password_reset' %}" method="POST">{% csrf_token %}

    {{form}}

    <input type="submit" value="Recuperar">
    {% if form.errors %}
        {{form.errors}}
    {% endif %}


</form>
```

password_reset_done.html:

```
Enviamos para o seu email um link para alteração da senha.
```

password_reset_complete.html

```
Sua senha foi resetada com sucesso

<a href="{% url 'logar' %}">CLIQUE AQUI</a> para acessar sua conta.
```

password_reset_confirm_view.html

```
<form action="" method="POST">{%csrf_token%}
    {{form}}
    <input type="submit" value="enviar">
    {% if form.errors %}
        {{form.errors}}
    {% endif %}

</form>
```

Em logar.html redirecione para reset_password:

```
<a href="{% url 'password_reset' %}">Esqueci minha senha</a>
```

Crie um novo APP:

```
python3 manage.py startapp jobs
```

Crie as models:

```
from django.db import models
from django.contrib.auth.models import User

class Referencias(models.Model):
    arquivo = models.FileField(upload_to='referencias')

    def __str__(self) -> str:
        return self.arquivo.url

class Jobs(models.Model):
    categoria_choices = (('D', 'Design'),
                         ('EV', 'Edição de Vídeo'))

    status_choices = (('C', 'Em criação'),
                      ('AA', 'Aguardando aprovação'),
                      ('F', 'Finalizado'))

    titulo = models.CharField(max_length=200)
    descricao = models.TextField()
    categoria = models.CharField(max_length=2, choices=categoria_choices, default="D")
    prazo_entrega = models.DateTimeField()
    preco = models.FloatField()
    referencias = models.ManyToManyField(Referencias)
    profissional = models.ForeignKey(User, on_delete=models.DO_NOTHING, null=True, blank=True)
    reservado = models.BooleanField(default=False)
    status = models.CharField(max_length=2, default='AA')



    def __str__(self) -> str:
        return self.titulo
```

Crie as migrações:

```
python3 manage.py makemigrations
```

Faça as migrações:

```
python3 manage.py migrate
```

Inclua as models no admin

```
admin.site.register(Jobs)
admin.site.register(Referencias)
```

Crie a URL para os arquivos de media:

```
from django.conf import settings
from django.conf.urls.static import static

...
urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Crie uma URL para JOBS:

```
path('jobs/', include('jobs.urls'))
```

Crie uma URL para encontrar jobs:

```
path('encontrar_jobs/', views.encontrar_jobs, name="encontrar_jobs")
```

Crie a view encontrar_jobs:

```
def encontrar_jobs(request):
    return render(request, 'encontrar_jobs.html')
```

Crie o HTML encontrar_jobs

```
{% extends 'base.html' %}
{% load static %}

{% block 'body' %}

    <div class="container">
    <br>
    <br>
        <div class="row">

            <div class="col-sm">
            <form action="/teste" method="POST">
                <input type="text" class="form-control" placeholder="Preço mínimo" name="preco_minimo">
                <br>
                <input type="text" class="form-control" placeholder="Preço máximo" name="preco_maximo">
            </div>

            <div class="col-sm">
                <input type="date" class="form-control" name="prazo_minimo">
                <br>
                <input type="date" class="form-control" name="prazo_maximo">
            </div>

            <div class="col-sm">
                <select class="form-control" name="categoria">
                    <option value="D">Design</option>
                    <option value="EV">Edição de vídeo</option>
                </select>
                <br>
                <input type="submit" class="btn btn-success btn-lg" value="FILTRAR">
            </form>
            </div>

        </div>
        <br>
        <br>
        <h2 class="titulo">JOB’S EM ABERTO:</h2>
        <div class="row justify-content-around">
            <div class="col-6 row-card">
                <div class="card-job">
                    <div class="header-job">
                       <h4 class="titulo-job">Criação de thumb</h4>
                    </div>
                    <div class="body-job">
                        <p class="titulo-body">Descrição:</p>
                        <div style="width: 70%">
                            <p>What is Lorem Ipsum?Lorem Ipsum is simply
                                dummy text of the printing and
                                typesettingindustry. Lorem Ipsum has been</p>
                        </div>

                        <div class="row">
                            <div class="col-4">
                                <p class="titulo-body">Preço:</p>
                                <p>R$ 25,00</p>
                            </div>

                            <div class="col-4">
                                <p class="titulo-body">Prazo:</p>
                                <p>22 de Abril</p>
                            </div>

                            <div class="col-4">
                                <p class="titulo-body">Categoria:</p>
                                <p>Design</p>
                            </div>

                        </div>

                        <button class="btn btn-success">Ver detalhes</button>
                        <br>
                        <br>

                    </div>
                </div>
            </div>
        </div>
        <br>
        <br>
        <h2 class="titulo">JOB’S EM ANDAMENTO:</h2>
        <div class="row justify-content-around">
            <div class="col-6 row-card">
                <div class="card-job">
                    <div class="header-job">
                       <h4 class="titulo-job">Criação de thumb</h4>
                    </div>
                    <div class="body-job">
                        <p class="titulo-body">Descrição:</p>
                        <div style="width: 70%">
                            <p>What is Lorem Ipsum?Lorem Ipsum is simply
                                dummy text of the printing and
                                typesettingindustry. Lorem Ipsum has been</p>
                        </div>

                        <div class="row">
                            <div class="col-4">
                                <p class="titulo-body">Preço:</p>
                                <p>R$ 25,00</p>
                            </div>

                            <div class="col-4">
                                <p class="titulo-body">Prazo:</p>
                                <p>22 de Abril</p>
                            </div>

                            <div class="col-4">
                                <p class="titulo-body">Categoria:</p>
                                <p>Design</p>
                            </div>

                        </div>

                        <button class="btn btn-success">Ver detalhes</button>
                        <br>
                        <br>

                    </div>
                </div>
            </div>
        </div>
    </div>

{% endblock %}
```

Crie o CSS encontrar_jobs.css

```
.fundo{

    background-color: #222;
    color: white;

}

.card-job{

    background-color: #444;
    box-shadow: 0 0px 20px 0 rgba( 255, 207, 0, 0.3 );
```

```
    padding: 0;
    margin-top: 50px;



}

.header-job{

    background-color: #333;
    width: 100% !important;
    padding: 5px;
    padding-left: 15px;


}

.row-card{
    margin-top: 40px;
}

.titulo-job{

    color: #FFCF00;
    font-weight: bold;
    margin-top: 10px;

}

.body-job{

    padding: 5px;
    padding-left: 15px;

}

.titulo-body{

    color: #E747DC;
    margin-top: 10px;

}

.titulo{
    font-family: Arial, Helvetica, sans-serif;
    font-weight: bold;
    font-size: 52px;
    display: inline;
    background-image: linear-gradient(90deg, rgba(255,207,0,1) 0%, rgba(231,71,220,1) 100%);
    background-clip: text;
    -webkit-background-clip: text;
    color: transparent;
    padding-bottom: 10px;
}

.job-modal-titulo{
    background-image: linear-gradient(90deg, rgba(255,207,0,1) 0%, rgba(231,71,220,1) 100%);
    background-clip: text;
    -webkit-background-clip: text;
    color: transparent;
    font-size: 30px;
    font-weight: bold;
}
```

importe o CSS para dentro do template HTML:

```
{% block 'head' %}

<link rel="stylesheet" href="{% static 'jobs/css/encontrar_jobs.css' %}">

{% endblock %}
```

Adicione uma barra de navegação em base.html:

```
{% if user.is_authenticated %}
      <nav style="background-color: #444;" class="navbar navbar-expand-lg navbar-dark">
        <a class="navbar-brand" href="#"><span class="font-degrade-nav">FreelaWay</span></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">
            <li class="nav-item active">
              <a class="nav-link" href="#">Encontrar job's</a>
            </li>

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

Vamos adicionar o CSS da barra de navegação:

```
<style>

      .font-degrade-nav{
          background-image: linear-gradient(90deg, rgba(255,207,0,1) 0%, rgba(231,71,220,1) 100%);
          background-clip: text;
```

```
          -webkit-background-clip: text;
          color: transparent;
          padding-bottom: 10px;
          font-size: 20px;
          font-weight: bold;
      }


</style>
```

Vamos ler do banco de dados todos os JOB'S em aberto:

```
def encontrar_jobs(request):
    if request.method == "GET":
        jobs = Jobs.objects.filter(reservado=False)
        return render(request, 'encontrar_jobs.html', {'jobs': jobs})
```

No template exiba todos os JOB'S:

```
{% for job in jobs %}
                <div class="col-6 row-card">
                    <div class="card-job">
                        <div class="header-job">
                        <h4 class="titulo-job">{{job.titulo}}</h4>
                        </div>
                        <div class="body-job">
                            <p class="titulo-body">Descrição:</p>
                            <div style="width: 70%">
                        <p>{{job.descricao}}</p>
                    </div>

                    <div class="row">
                        <div class="col-4">
                            <p class="titulo-body">Preço:</p>
                            <p>{{job.preco}}</p>
                        </div>

                        <div class="col-4">
                            <p class="titulo-body">Prazo:</p>
                            <p>{{job.prazo_entrega}}</p>
                        </div>

                        <div class="col-4">
                            <p class="titulo-body">Categoria:</p>
                            <p>{% if job.categoria == 'D'%}
                                    Design
                                {% else %}
                                    Edição de vídeo
                                {% endif%}

                            </p>
                        </div>

                    </div>

                    <button class="btn btn-success">Ver detalhes</button>
                    <br>
                    <br>

                </div>
            </div>
        </div>

{% endfor %}
```

Crie os filtros:

```
def encontrar_jobs(request):
    if request.method == "GET":
        preco_minimo = request.GET.get('preco_minimo')
        preco_maximo = request.GET.get('preco_maximo')

        prazo_minimo = request.GET.get('prazo_minimo')
        prazo_maximo = request.GET.get('prazo_maximo')

        categoria = request.GET.get('categoria')

        if preco_minimo or preco_maximo or prazo_minimo or prazo_maximo or categoria:
            if not preco_minimo:
                preco_minimo = 0

            if not preco_maximo:
                preco_maximo = 999999

            if not prazo_minimo:
                prazo_minimo = datetime(year=1900, month=1, day=1)

            if not prazo_maximo:
                prazo_maximo = datetime(year=3000, month=1, day=1)

            jobs = Jobs.objects.filter(preco__gte=preco_minimo)\
                .filter(preco__lte=preco_maximo)\
                .filter(prazo_entrega__gte=prazo_minimo)\
                .filter(prazo_entrega__lte=prazo_maximo)\
                .filter(categoria__in=categoria)
        else:
            jobs = Jobs.objects.filter(reservado=False)
```

```
        return render(request, 'encontrar_jobs.html', {'jobs': jobs})
```

Altere o form

```
<form action="{% url 'encontrar_jobs' %}" method="GET">
```

E com isso finalizamos nossa segunda aula da PSW 3.0