









Ibrahima Diouma Diallo Génie Informatique

Examen Django

Voici l'enchaînement de quelques captures d'écran de notre plateforme Django suivi d'explication.

- ☐ Cette image correspond à la première page (sans ajout des employés)

Gestion des Employés								
<input type="checkbox"/>	Prénom	Nom	Téléphone	Email	Adresse	Département	Action	

Gestion des Employés								
<input type="checkbox"/>	Prénom	Nom	Téléphone	Email	Adresse	Département	Action	
<input type="checkbox"/>	Ibrahima	Diouma	777777777	ibara@gmail.com	Grand Dakar	Compta		
<input type="checkbox"/>	Ibrahima	Diouma	777777777	ibara@gmail.com	Grand Dakar	Informatique		
<input type="checkbox"/>	Ibrahima	Diouma	777777777	ibara@gmail.com	Grand Dakar	GRH		

- ☐ ajouts des employés avec leurs différents informations (.....)

```
1 body {
2   color: #566787;
3   background: #f5f5f5;
4   font-family: 'Varela Round', sans-serif;
5   font-size: 13px;
6 }
7 .table-responsive {
8   margin: 30px 0;
9 }
10 .table-wrapper {
11   background: #fff;
12   padding: 20px 25px;
13   border-radius: 3px;
14   min-width: 1000px;
15   box-shadow: 0 1px 1px rgba(0,0,0,.05);
16 }
17 .table-title {
18   padding-bottom: 15px;
19   background: #435d7d;
20   color: #fff;
21   padding: 16px 30px;
22   min-width: 100%;
23   margin: -20px -25px 10px;
24   border-radius: 3px 3px 0 0;
25 }
26 .table-title h2 {
27   margin: 5px 0 0;
28   font-size: 24px;
29 }
30 .table-title .btn-group {
```

□ notre code css

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Index</title>
6   {% load static %}
7   <link rel="stylesheet" href="{% static 'css/style.css' %}" />
8   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/
  bootstrap.min.css">
9 </head>
10 <body>
11 <form method="POST" class="post-form" action="/updateEmp/{{employee.pk}}">
12   {% csrf_token %}
13   <div class="container">
14 <br>
15     <div class="form-group row">
16       <label class="col-sm-1 col-form-label"></label>
17       <div class="col-sm-4">
18         <h3>Mettre à jour</h3>
19       </div>
20     </div>
21     <div class="form-group row">
22       <label class="col-sm-2 col-form-label">Prénom:</label>
23       <div class="col-sm-4">
24         <input type="text" name="" id="" required value="{{ employee.first_name }}" />
25       </div>
26     </div>
27     <div class="form-group row">
28       <label class="col-sm-2 col-form-label">Nom:</label> |
29       <div class="col-sm-4">
```

```
23 <div class="col-sm-4">
24   <input type="text" name="" id="" required value="{{ employee.first_name }}" />
25 </div>
26 </div>
27 <div class="form-group row">
28   <label class="col-sm-2 col-form-label">Nom:</label>
29   <div class="col-sm-4">
30     <input type="text" name="" id="" required value="{{ employee.last_name }}" />
31   </div>
32 </div>
33 <div class="form-group row">
34   <label class="col-sm-2 col-form-label">Téléphone:</label>
35   <div class="col-sm-4">
36     <input type="text" name="" id="" required value="{{ employee.phone_number }}" />
37   </div>
38 </div>
39 <div class="form-group row">
40   <label class="col-sm-2 col-form-label">Email:</label>
41   <div class="col-sm-4">
42     <input type="email" name="" id="" required value="{{ employee.email }}" />
43   </div>
44 </div>
45 <div class="form-group row">
46   <label class="col-sm-2 col-form-label">Adresse:</label>
47   <div class="col-sm-4">
48     <input type="text" name="" id="" required value="{{ employee.address }}" />
49   </div>
50 </div>
51 <div class="form-group row">
52   <label class="col-sm-2 col-form-label">Département:</label>
```

```
44 </div>
45 <div class="form-group row">
46   <label class="col-sm-2 col-form-label">Adresse:</label>
47   <div class="col-sm-4">
48     <input type="text" name="" id="" required value="{{ employee.address }}" />
49   </div>
50 </div>
51 <div class="form-group row">
52   <label class="col-sm-2 col-form-label">Département:</label>
53   <div class="col-sm-4">
54     <input type="text" name="" id="" required value="{{ employee.departement }}" />
55   </div>
56 </div>
57
58 <div class="form-group row">
59   <label class="col-sm-1 col-form-label"></label>
60   <div class="col-sm-4">
61     <button type="submit" class="btn btn-success">Valider</button>
62   </div>
63 </div>
64 </div>
65 </form>
66 </body>
67 </html>
```

- les 3 images qui se suivent il s'agit du fichier fichier permettant de modifier le les infos de l'employé

```
1 from django.contrib import admin
2 from .models import *
3
4 # Register your models here.
5
6 @admin.register(Departement)
7 class DepartementAdmin(admin.ModelAdmin):
8     list_depart = ('name')
9
10 @admin.register(Employer)
11 class EmployerAdmin(admin.ModelAdmin):
12     list_employer = ('first_name', 'last_name', 'phone_number', 'email', 'address', 'departement')
13
14
15
```

Le fichier admin enregistre les objets de modélisation

The image displays three sequential screenshots of a code editor, likely VS Code, showing the development of an HTML form for a user registration page. The left sidebar shows a project structure with folders like 'core', 'employerapp', and 'templates'. The main editor shows the 'index.html' file being edited.

The first screenshot shows the initial setup of the HTML document, including the DOCTYPE, language, charset, and title. It also includes a link to the Bootstrap CSS file and a form with a POST method and CSRF token.

The second screenshot shows the form fields being added. The first field is for the first name, followed by the last name, phone number, email, and address. Each field is wrapped in a 'form-group' and a 'col-sm-4' class.

The third screenshot shows the final form structure, including the department field and a registration button. The button is a primary button with the text 'Enregistrer'.

Ces fichiers sont pour la page de home c'est elle qu'on accède lorsqu'on se connecte à notre plateforme


```
OPEN FILES
edit.html

OLDERS
getEmployer
core
  __pycache__
  _init_.py
  asgi.py
  settings.py
  urls.py
  wsgi.py
employerapp
  __pycache__
  migrations
  _init_.py
  admin.py
  apps.py
  forms.py
  models.py
  tests.py
  urls.py
  views.py
static
templates
  edit.html
  index.html
  main.html
db.sqlite3
manage.py

15 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script>
16
17
18 </head>
19 <body>
20 <div class="container-xl">
21 <div class="table-responsive">
22 <div class="table-wrapper">
23 <div class="table-title">
24 <div class="row">
25 <div class="col-sm-6">
26 <h2>Gestion des <b>Employés</b></h2>
27 </div>
28 <div class="col-sm-6">
29 <a href="/emp" class="btn btn-success"><i class="material-icons">&#xE147;</i> <span>Ajouter</span></a>
30 <a href="/delete/{{ employee.id }}" class="btn btn-danger"><i class="material-icons">&#xE15C;</i> <span>Supprimer</span></a>
31 </div>
32 </div>
33 </div>
34 <table class="table table-striped table-hover">
35 <thead>
36 <tr>
37 <th>
38 <span class="custom-checkbox">
39 <input type="checkbox" id="selectAll">
40 <label for="selectAll"></label>
41 </span>
42 </th>
43 <th>Prénom</th>
44 <th>Nom</th>
45 <th>Téléphone</th>
46 <th>Email</th>
47 <th>Adresse</th>
48 <th>Département</th>
49 <th>Action</th>
50 </tr>
51 </thead>
52 <tbody>
53 <% for employee in employees %>
54 <tr>
55 <td>
56 <span class="custom-checkbox">
57 <input type="checkbox" id="checkbox1" name="options[]" value="1">
58 <label for="checkbox1"></label>
59 </span>
60 </td>
61 <td>{{employee.first_name}}</td>
62 <td>{{employee.last_name}}</td>
63 <td>{{employee.phone_number}}</td>
```

```
35 <thead>
36 <tr>
37 <th>
38 <span class="custom-checkbox">
39 <input type="checkbox" id="selectAll">
40 <label for="selectAll"></label>
41 </span>
42 </th>
43 <th>Prénom</th>
44 <th>Nom</th>
45 <th>Téléphone</th>
46 <th>Email</th>
47 <th>Adresse</th>
48 <th>Département</th>
49 <th>Action</th>
50 </tr>
51 </thead>
52 <tbody>
53 <% for employee in employees %>
54 <tr>
55 <td>
56 <span class="custom-checkbox">
57 <input type="checkbox" id="checkbox1" name="options[]" value="1">
58 <label for="checkbox1"></label>
59 </span>
60 </td>
61 <td>{{employee.first_name}}</td>
62 <td>{{employee.last_name}}</td>
63 <td>{{employee.phone_number}}</td>
```

```
57 <input type="checkbox" id="checkbox1" name="options[]" value="1">
58 <label for="checkbox1"></label>
59 </span>
60 </td>
61 <td>{{employee.first_name}}</td>
62 <td>{{employee.last_name}}</td>
63 <td>{{employee.phone_number}}</td>
64 <td>{{employee.email}}</td>
65 <td>{{employee.address}}</td>
66 <td>{{employee.departement}}</td>
67 <td>
68 <button class="btn-danger btn-sm" onclick="location.href='/deleteEmp/{{employee.pk}}'"><i class="material-icons" data-toggle="tooltip" title="Delete">&#xE872;</i></button>
69 <button class="btn-success btn-sm" onclick="location.href='/editEmp/{{employee.pk}}'"><i class="material-icons" data-toggle="tooltip" title="Edit">&#xE254;</i></a></button>
70 </td>
71 </tr>
72 <% endfor %>
73 </tbody>
74 </table>
75
76 </div>
77 </div>
78 </body>
79 </html>
```

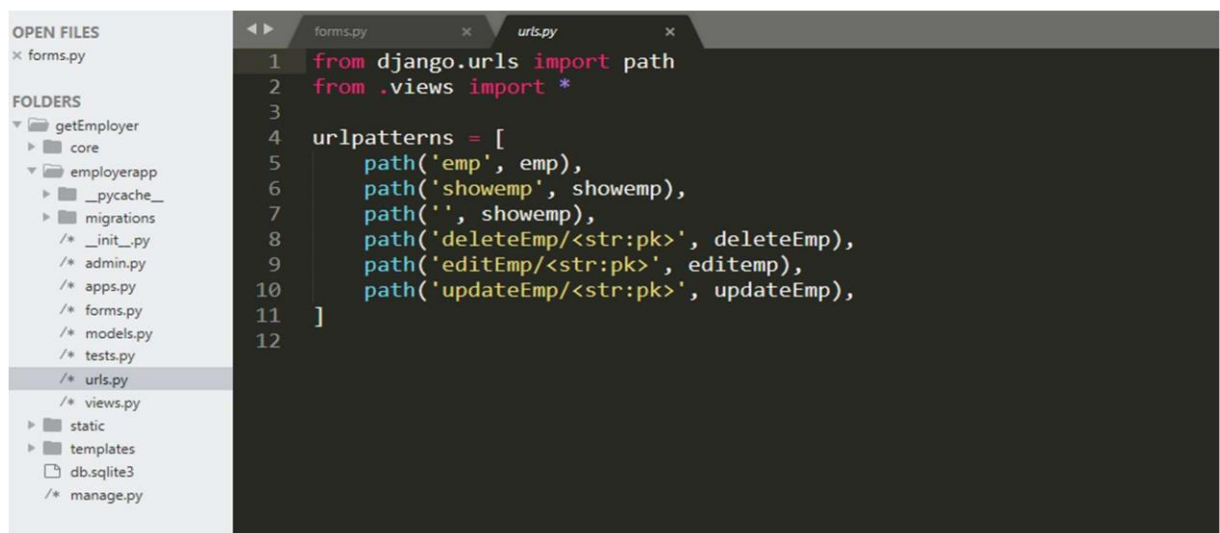


cette fichier permet aux employées de s'inscrire



```
75 # Database
76 # https://docs.djangoproject.com/en/3.1/ref/settings/#databases
77
78 DATABASES = {
79     'default': {
80         'ENGINE': 'django.db.backends.mysql',
81         'NAME': 'django_unipro',
82         'USER': 'root',
83         'PASSWORD': '',
84         'HOST': 'localhost',
85         'PORT': '3306' ,
86     }
87 }
88
89
```

Configuration de la base de données



```
1 from django.urls import path
2 from .views import *
3
4 urlpatterns = [
5     path('emp', emp),
6     path('showemp', showemp),
7     path('', showemp),
8     path('deleteEmp/<str:pk>', deleteEmp),
9     path('editEmp/<str:pk>', editemp),
10    path('updateEmp/<str:pk>', updateEmp),
11 ]
12
```

☐ cette image permet la configuration des urls qui se trouve dans le dossier du projet

FOLDERS

getEmployer

core

__pycache__

/* _init_.py

asgi.py

settings.py

urls.py

wsgi.py

employerapp

__pycache__

migrations

/* _init_.py

admin.py

apps.py

forms.py

models.py

tests.py

urls.py

views.py

static

css

/* main.css

img

js

templates

2

3 The `urlpatterns` list routes URLs to views. For more information please see:

4 | <https://docs.djangoproject.com/en/3.1/topics/http/urls/>

5 Examples:

6 Function views

7 | 1. Add an import: from my_app import views

8 | 2. Add a URL to urlpatterns: path('', views.home, name='home')

9 Class-based views

10 | 1. Add an import: from other_app.views import Home

11 | 2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')

12 Including another URLconf

13 | 1. Import the include() function: from django.urls import include, path

14 | 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))

15 """

16 from django.contrib import admin

17 from django.urls import path, include

18

19 urlpatterns = [

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

21 | path('', include('employerapp.urls')),

22]

23

url dans le projet


```
OPEN FILES
x forms.py

FOLDERS
v getEmployer
  p core
  v employerapp
    p _pycache_
    p migrations
    /* _init_.py
    /* admin.py
    /* apps.py
    /* forms.py
    /* models.py
    /* tests.py
    /* urls.py
    /* views.py
  p static
  p templates
  db.sqlite3
  /* manage.py

forms.py
views.py

1 from django.shortcuts import render, redirect
2 from .models import Employer, Departement
3 from .forms import Form_employer
4
5
6 # Create your views here.
7
8 # To create employee
9 def emp(request):
10     if request.method == "POST":
11
12         form = Form_employer(request.POST)
13         if form.is_valid():
14             try:
15
16                 form.save()
17                 return redirect("/showemp")
18             except:
19                 pass
20         else:
21             form = Form_employer()
22             return render(request, "index.html", {'form':form})
23
24 # To show employee details
25 def showemp(request):
26     employees = Employer.objects.all()
27     return render(request, "main.html", {'employees':employees})
28
29
30
31
32
33
34
35 # To delete employee details
36 def deleteEmp(request, pk):
37     employee = Employer.objects.get(pk=pk)
38     employee.delete()
39     return redirect("/showemp")
40
41 # To edit employee details
42 def editemp(request, pk):
43     employee = Employer.objects.get(pk=pk)
44     return render(request, "edit.html", {'employee':employee})
45
46 # To update employee details
47 def updateEmp(request, pk):
48     employee = Employer.objects.get(pk=pk)
49     form = Form_employer(request.POST, instance= employee)
50     if form.is_valid():
51         form.save()
52         return redirect("/showemp")
53     return render(request, "main.html", {'employee': employee})
```

Ce fichier permet l'affichage des interfaces pour l'utilisateur