**pyDatabase connectivity**

PostgreSQL

**This is the youtube video for Installing PostgreSQL on Windows:** <https://www.youtube.com/watch?v=AEZg-sTxxmw>

After install search in window pgAdmin4

Create the database

Go to the server/database/create\_database

Password is Mysql1212.

Register our database in our project.

Settings.py

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.postgresql',

        'NAME': 'carzone\_db',

        'USER': 'postgres',

        'PASSWORD': 'Mysql1212',

        'HOST': 'localhost',

    }

}

**Here user is the owner (when you create a database so you can see the owner name.)**

**Create a superuser**

Before creating the super user we have to do migrations

Python manage.py makemigrations

Python manage.py migrate

**Ans also install the psycopg2.**

Pip install psycopg2

**Now create the super user**

Normal (cmd)command

Python manage.py createsuperuser

**Gitbase command for super user**

Winpty Python manage.py createsuperuser

Now login in the admin panel.

**Now create the model for our project.**

class Team(models.Model):

    first\_name = models.CharField(max\_length=255)

    last\_name = models.CharField(max\_length=255)

    designation = models.CharField(max\_length=255)

    photo = models.ImageField(upload\_to='photos/%y/%m/%d/')

    facebook\_link = models.URLField(max\_length=500)

    twitter\_link = models.URLField(max\_length=500)

    googlr\_plus\_link = models.URLField(max\_length=500)

    created\_date = models.DateTimeField(auto\_now\_add=True)

    def \_\_str\_\_(self):

        return self.first\_name

**install the pillow bcz we are working with photos/media.**

**Pip install pillow**

**Now do the migrations**

Run both the commands

**Now run the server and also register the admin.**

Admin.py

From .models import Team

Admin.site.register(Team)

Whenever you change in models.py always run the migration commands.

**Media file configuration**

Whenever you add new member in team using admin

You will not get the photo

Settings.py

MEDIA\_ROOT = os.path.join(BASE\_DIR,'media')

MEDIA\_URL = '/media/'

Go to urls.py (main urls.py)

Import this 2 packages.

from django.conf.urls.static import static

from django.conf import settings

urlpatterns = [

    path('admin/', admin.site.urls),

    path('',include('pages.urls')),

] + static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

**Now fetch all the data from database and display on the browser.**

Now collect all the object.

**Views.py**

def home(request):

teams = Team.objects.all()

data = {

'teams': teams,

}

return render(request,'pages/home.html',data)

**home.html**

here I removed all the static content of html I mean all the div tags except one div bcz we have to fetch data dynamically

here I used {{}} double bcz we are fetching from the database.

{{ team.photo.url }}

Here team object collect the photo in the form of url and search in the static if find than display on screen.

I used this for the social media links and you can also use target=”\_blank”

{{ team.googlr\_plus\_link}}

<!-- Our team start -->

<div class="our-team content-area bg-grea-3">

    <div class="container">

        <!-- Main title -->

        <div class="main-title">

            <h1>Executive <span>Team</span></h1>

            <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod</p>

        </div>

        <!-- Slick slider area start -->

        <div class="slick-slider-area clearfix">

            <div class="row slick-carousel" data-slick='{"slidesToShow": 4, "responsive":[{"breakpoint": 1024,"settings":{"slidesToShow": 2}}, {"breakpoint": 768,"settings":{"slidesToShow": 1}}]}'>

              {% for team in teams %}

                <div class="slick-slide-item">

                    <div class="team-1">

                        <div class="photo">

                            <a href="#">

                                <img src="{{ team.photo.url }}" alt="team-1" class="img-fluid">

                            </a>

                            <div class="social-list clearfix">

                                <a href=" {{ team.facebook\_link}}" class="facebook-bg"><i class="fa fa-facebook"></i></a>

                                <a href=" {{ team.twitter\_link}}" class="twitter-bg"><i class="fa fa-twitter"></i></a>

                                <a href=" {{ team.googlr\_plus\_link}}" class="google-bg"><i class="fa fa-google"></i></a>

                            </div>

                        </div>

                        <div class="details">

                            <h4><a href="team-detail.html">{{ team.first\_name }} {{ team.last\_name }}</a></h4>

                            <h5>{{ team.designation }}</h5>

                        </div>

                    </div>

                </div>

              {% endfor %}

            </div>

        </div>

    </div>

</div>

<!-- Our team end -->

Do the same for contact.

Do same changes in contact view and contact html .

And also redirect the home and logo on home page

**Team Admin Customization**

**I want to display team members as a list format where id photo and name should be clickable.**

**Admin.py**

First create a class where it will be inherited from admin.ModelAdmin

Than register this class in admin

from django.contrib import admin

from .models import Team

class TeamAdmin(admin.ModelAdmin):

list\_display = ('id','first\_name','last\_name','designation')

admin.site.register(Team, TeamAdmin)

Here we created a class and tuple of the field which we want to display at admin site.

Than we register this class. (*See the last line*)

You can only use the list\_display name only

Only id is clickable but we want to name also be clickable and also add the photo (thumbnail)

list\_display\_links = ('id','first\_name')

**Add photo** - > we have create a function and pass the dynamic values(images)

First import the html format

Than create the function after that we do the short\_description of our function

from django.utils.html import format\_html

class TeamAdmin(admin.ModelAdmin):

def thumbNail(self,object):

return format\_html('<img src="{}" width="40" style="border-radius:50%" />'.format(object.photo.url))

thumbNail.short\_description = 'photo'

list\_display = ('id','thumbNail','first\_name','last\_name','designation','created\_date')

list\_display\_links = ('id','first\_name')

Here the function contain the object (object contain all the team data)

**Create a Search bar and filter**

We have to use search\_fields and inside the tuple we have to pass the field name. (By what field name you want to search.)

# Search field

search\_fields = ('id','first\_name','designation')

# filter the team member by their designation

list\_filter = ('designation',)

List\_filter = (‘designation’,) after comma is compulsory

Complete Class

from django.contrib import admin

from .models import Team

from django.utils.html import format\_html

# Register your models here.

class TeamAdmin(admin.ModelAdmin):

def thumbNail(self,object):

return format\_html('<img src="{}" width="40" style="border-radius:50%" />'.format(object.photo.url))

thumbNail.short\_description = 'photo'

list\_display = ('id','thumbNail','first\_name','last\_name','designation','created\_date')

list\_display\_links = ('id','thumbNail','first\_name')

# Search field

search\_fields = ('id','first\_name','designation')

# filter the team member by their designation

list\_filter = ('designation',)

admin.site.register(Team, TeamAdmin)