Empezando proyecto

Nota: Si se te joden las tablas en phpmyadmin/localhost, te vas a mysql (en el lateral izquierdo), seleccionas todas, y eliges reparar tablas.

* **Creamos el entorno:** *python -m venv entornoNombre*
* **Entramos al proyecto:***Scripts\activate.bat*
* **Instalamos Django:** *pip install Django*
* **Creamos proyecto:** *django-admin startproject proyectoNombre*
* **Instalamos mysql:** *pip install mysqlclient*
* **Creamos la app:** *python manage.py startapp nombreApp*
* **Para poner en marcha el servidor:** *python manage.py runserver*
* **Crear base de datos en localhost/phpmyAdmin y cambiamos los settings.py (Linea 76):**

*DATABASES = {*

*'default': {*

*'ENGINE': 'django.db.backends.mysql',*

*'NAME': 'agendaEmpresas',*

*'USER': 'agendaEmpresas',*

*'PASSWORD': 'agendaEmpresas',*

*'HOST': 'localhost',*

*'PORT': '3306',*

*}*

*}*

**En la línea 33 añadimos nuestra app**

*INSTALLED\_APPS = [*

*'django.contrib.admin',*

*'django.contrib.auth',*

*'django.contrib.contenttypes',*

*'django.contrib.sessions',*

*'django.contrib.messages',*

*'django.contrib.staticfiles', #Esta fila es para que salga bonito admin*

*'principal.apps.PrincipalConfig'*

*]*

**nota: también cambiar en linea 111 el idioma a ‘es’**

* **Aunque no hayamos creado ningún modelo, hacemos las migraciones:** *python manage.py makemigrations*

*python manage.py migrate*

* **Modificamos los modelos: Lo mejor es fijarse en un proyecto ya creado, este es un ejemplo.**

*from django.db import models*

*# Create your models here.*

*class ModeloViajes(models.Model):*

*viaje = models.CharField(max\_length=255)*

*fechaObjetivo = models.DateField()*

*conseguido = models.BooleanField()*

*fechaConseguido = models.DateField(null=True)*

**Y volvemos a hacer las migraciones.**

* **Creamos el archivo urls.py en la carpeta de la aplicación**

*from django.urls import path*

*from . import views*

*urlpatterns = [*

*path('',views.listaViajes, name='listaViajes')*

*]*

* **Creamos la carpeta templates, y dentro de la carpeta otra carpeta llamada nombreApp**
* **Modificamos el urls.py de proyectoNombre**

*from django.contrib import admin*

*from django.urls import path,include*

*urlpatterns = [*

*path('',include('nombreApp.urls')),*

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

*]*

Listar

* **Modificamos las views, Ahora será solo para listar.**

*from datetime import datetime, timedelta*

*from django.shortcuts import render*

*from .models import ModeloViajes*

*from django.template import loader*

*from django.http import HttpResponse,HttpResponseRedirect*

*from django.urls import reverse*

*import datetime*

*# Create your views here.*

*def listaViajes(request):*

*fechaActual = datetime.datetime.now().date()*

*viaje = ModeloViajes.objects.all()*

*context = {*

*'viajes':viaje,*

*'fechaActual':fechaActual,*

*}*

*plantilla = loader.get\_template('viajes/viajes.html')*

*return HttpResponse(plantilla.render(context,request))*

* **Creamos el html viajes.html, lo puedes llamar como quieras, solo fijate que es al que estás llamando en views**

*<!DOCTYPE html>*

*<html lang="en">*

*<head>*

*<meta charset="UTF-8">*

*<meta http-equiv="X-UA-Compatible" content="IE=edge">*

*<meta name="viewport" content="width=device-width, initial-scale=1.0">*

*<title>Listado</title>*

*</head>*

*<body>*

*<h1>Viajes</h1>*

*<table>*

*<tr>*

*<th>Viaje</th>*

*<th>fechaObjetivo</th>*

*<th>Conseguido</th>*

*<th>fechaConseguido</th>*

*</tr>*

*<!--En este caso viajes es como lo hemos llamado en nuestro context en las views-->*

*{% for c in viajes %}*

*<tr>*

*<td>{{c.viaje}}</td>*

*<td>{{c.fechaObjetivo}}</td>*

*{% if c.conseguido %}*

*<td>Si</td>*

*{%else%}*

*<td>No</td>*

*{%endif%}*

*{% if c.fechaConseguido is NULL %}*

*<td></td>*

*{%else%}*

*<td>{{c.fechaConseguido}}</td>*

*{%endif%}*

*</tr>*

*{% endfor %}*

*</table>*

*</body>*

*</html>*

Añadir a la base de datos

* **Añadimos un formulario en el html para añadir a la base de datos y por ende a la tabla que hemos creado.**

*<!-- en action es la funcion que has creado en el views-->*

*<form action="add" method="get">*

*<label for="">Proposito:</label>*

*<input type="text" name="proposito" id="" required>*

*<label for="" >Fecha del objetivo:</label>*

*<input type="date" name="fechaObjetivo" id="" required>*

*<button type="submit" >Añadir</button>*

*</form>*

* **Modificamos el views.py**

*def add(request):*

*v = request.GET['viaje'] #Es lo que hayamos puesto en el name del input*

*fechaOb = request.GET['fechaObjetivo'] #Es lo que hayamos puesto en el name del input*

*viaje = ModeloViajes(viaje=v,fechaObjetivo=fechaOb,conseguido=False)*

*viaje.save()*

*return HttpResponseRedirect(reverse('listaViajes')) #Para volver a la pagina donde listamos todo*

* **Modificamos el urls.py de la aplicacion y añadimos:**

**path('add', views.add, name='add'),**

Borrar

* **Modificamos el views.py**

*def borrar(request,identificador):*

*prop = ModeloViajes.objects.get(id=identificador)*

*prop.delete()*

*return HttpResponseRedirect(reverse('listaViajes'))*

* **Modificar urls.py**

*path('borrar/<int:identificador>', views.borrar, name='borrar'),*

* **Añadimos un enlace en el html**

*<a href="borrar/{{c.id}}">Borrar</a>*

Modificar

**En modificar hay dos pasos, uno que te lleva al form, y el otro que te manda las cosas a la base de datos. Por lo que tendremos dos funciones en el views, y dos urls. Tendremos que crear un html para el formulario.**

* **Modificamos el views.py**

*#Nos lleva al fomrulario*

*def modificar(request, identificador):*

*viaje = ModeloViajes.objects.get(id=identificador)*

*context = {*

*'modificarViaje':viaje*

*}*

*plantilla = loader.get\_template('modificar.html')*

*return HttpResponse(plantilla.render(context,request))*

*#Manda los datos del formulario a la base de datos y redirige a la pagina principal*

*def actualizar (request,identificador):*

*viajeModif = ModeloViajes.objects.get(id=identificador)*

*viajeModif.viaje = request.GET['viaje']*

*viajeModif.save()*

*return HttpResponseRedirect(reverse('listaViajes'))*

* **Modificamos el urls.py**

**path('modificar/<int:identificador>', views.modificar, name = 'modificar'),**

**path('modificar/actualizar/<int:identificador>',views.actualizar, name='actualizar'),**

* **Modificamos nuestro html principal, en mi caso viajes.html para añadir**

*<a href="modificar/{{c.id}}">Borrar</a>*

* **creamos el html modificar.html**

*<!DOCTYPE html>*

*<html lang="en">*

*<head>*

*<meta charset="UTF-8">*

*<meta http-equiv="X-UA-Compatible" content="IE=edge">*

*<meta name="viewport" content="width=device-width, initial-scale=1.0">*

*<title>Modificar</title>*

*</head>*

*<body>*

*<h1>Modificando : {{modificarViaje.viaje}}</h1>*

*<form action="actualizar/{{modificarViaje.id}}" method="get">*

*<div>*

*<label for="">Viaje:</label>*

*<input type="text" name="viaje" id="" required>*

*</div>*

*<button type="submit">Modificar</button>*

*</form>*

*</div>*

*</body>*

*</html>*

Marcar como conseguido y resetear.

* **Modificamos el viajes.html**

**{% if p.conseguido %}**

**<td> <a href="resetear/{{p.id}}">Reset</a> </td>**

**{%else%}**

**<td><a href="conseguido/{{p.id}}">Conseguir</a></td>**

**{%endif%}**

**\**

* **Modificamos las views.py**

*def conseguido(request,identificador):*

*viaje = ModeloViajes.objects.get(id=identificador)*

*viaje.fechaConseguido = datetime.datetime.now()*

*viaje.conseguido = True*

*viaje.save()*

*return HttpResponseRedirect(reverse('listaViajes'))*

*def resetear(request,identificador):*

*viaje = ModeloViajes.objects.get(id=identificador)*

*viaje.fechaConseguido = None*

*viaje.conseguido = False*

*viaje.save()*

*return HttpResponseRedirect(reverse('listaViajes'))*

* **Añadimos en las urls.py**

*path('conseguido/<int:identificador>',views.conseguido,name='conseguido'),*

*path('resetear/<int:identificador>',views.resetear,name='resetear'),*