**SISTEMA DE GERENCIAMENTO PSC – 2024**

1. **Objetivo:**
   * Criar um sistema de gerenciamento de tarefas, processos, serviços administrativos, com uma base de dados MySql.
2. **Identifique os Requisitos:**
   * Requisitos funcionais - Armazenar dados
   * Requisitos não funcionais - Segurança e backup.
3. **Escopo do Projeto:**
   * Determine quais funcionalidades específicas estarão incluídas no projeto.
   * Defina os limites do que será desenvolvido (por exemplo, se o projeto incluirá apenas a parte de autenticação ou também a parte de administração).
4. **Tecnologias e Ferramentas:**
   * IDE - Visual Studio;
   * Framekork – Django;
   * Liguagem de Programação – Python;
   * Banco de Dados – My Sql;
   * Hospedagem – Hostgator
   * Cloud - > GitHub
5. **Plano de Desenvolvimento:**
   * Divida o projeto em tarefas menores e crie um cronograma para cada etapa.

Nome do projeto: datapsc

Local de desenvolvimento: C:\Users\pscse\Documents\datapsc

**Fase 1**

Início do desenvolvimento do projeto:

**Definição do local:**

Documents

C:\Users\pscse\Documents

**Criação pasta pelo VsCode**

Nome da Pasta: datapsc

C:\Users\pscse\Documents\datapsc

**Comando no terminal do Visual Studio**

1. Cd Documents - > para entrar na pasta documentos;
2. Django-admin -> Entrar no administrador do Django;

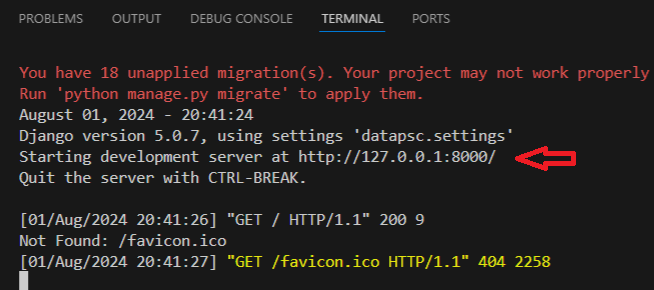


1. Django-admin startproject datapsc (nome do projeto);
2. Cd datapsc -> Entra na pasta do projeto datapsc;

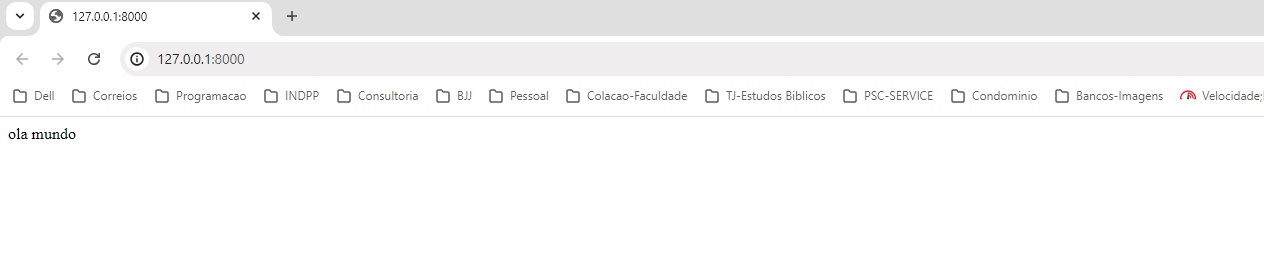


1. Abrir o projeto no VsCode – Open Folder escolher o caminho onde está o projeto
2. C:\Users\pscse\Documents\datapsc
3. Abrir a pasta manage.py
4. Subir o projeto para o GitHub
5. No VsCode digitar CTRL + SHIFT + P;
6. Na guia do VsCode digitar sinal de >;
7. Escolher Publish to Github;
8. Escolher Publish to GitHub private repository;
9. Dar styart no projeto, definindo um nome (website)
10. No terminal digitar: python manage.py startapp website;
11. Testar se o projeto está ok – no terminal -> python manage.py runserver





**Resultado:**

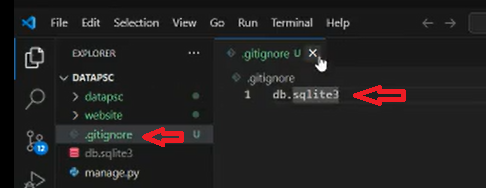


Cancelar o comando do runserver- digitar Ctrl + C

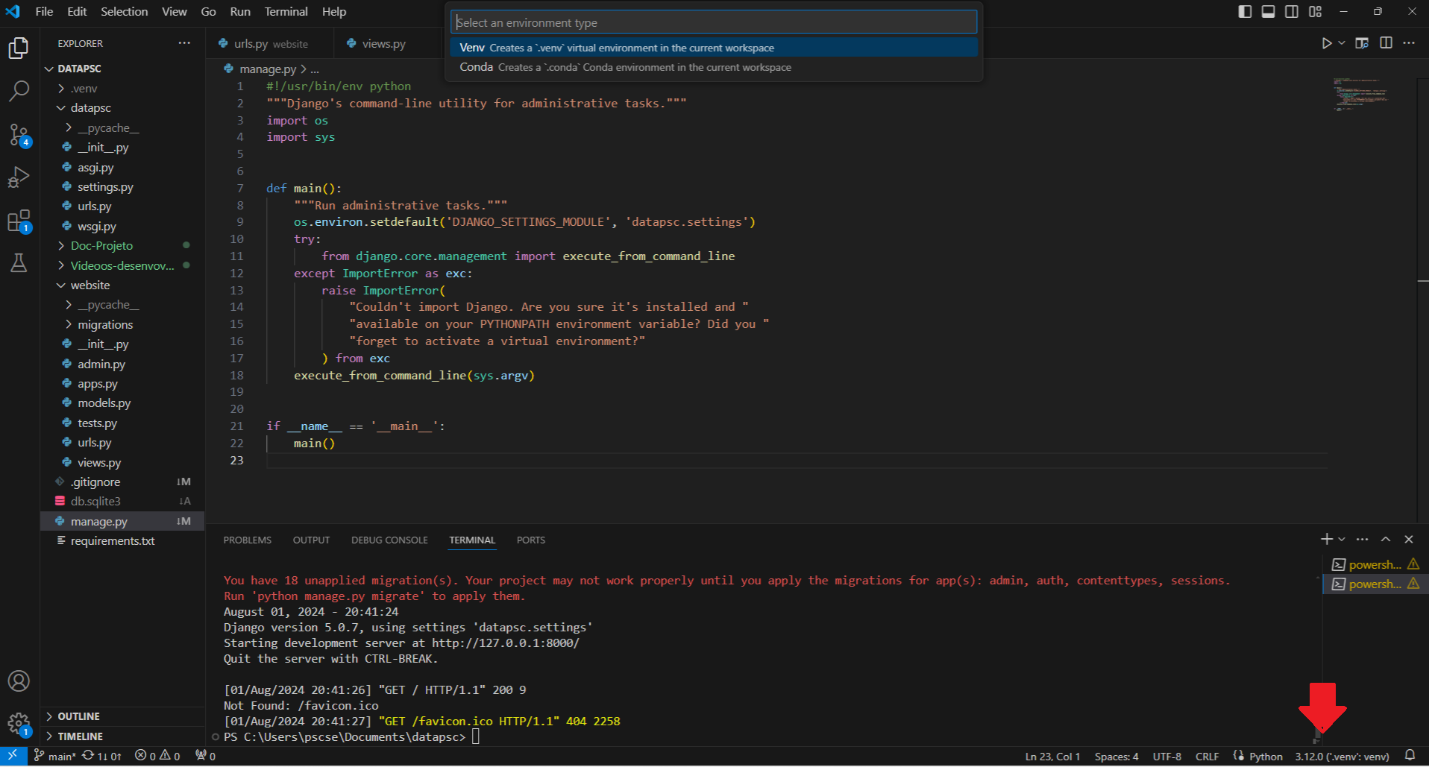
**Ajustes no sistema:**

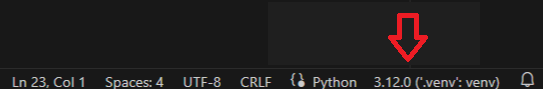
O banco de dados não pode fazer parte do repositório, não permitir que ele vá para o GitHub

1. Criar um arquivo – new file nome gitignore
2. Dentro do arquivo git ignore escrever o nome do banco db.sqlite3

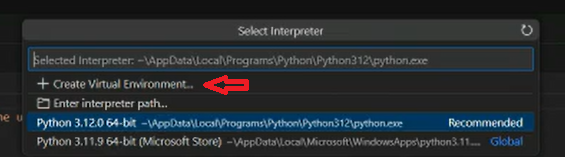


1. Selecionar o ambiente virtual VENV
2. No VsCode na parte inferior direita selecionar o número da versão do Python





1. Escolher + Create Virtual Environment

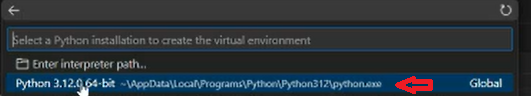


1. Escolher -> Venv

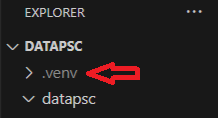
Interface gráfica do usuário, Texto

Descrição gerada automaticamente

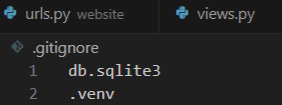
1. Escolher Python 3.12.064.bit



1. O VsCode já cria o ambiente virtual



1. O Venv também não deve subir para o GitHub então deve estar adicionado no gitinore, coloca o nome .venv e salva Ctrl + S

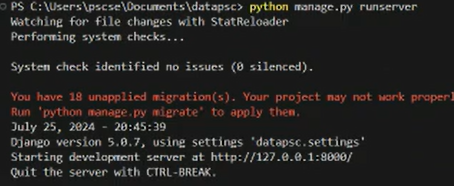


1. Abrir um terminal novo e digitar: python manage.py



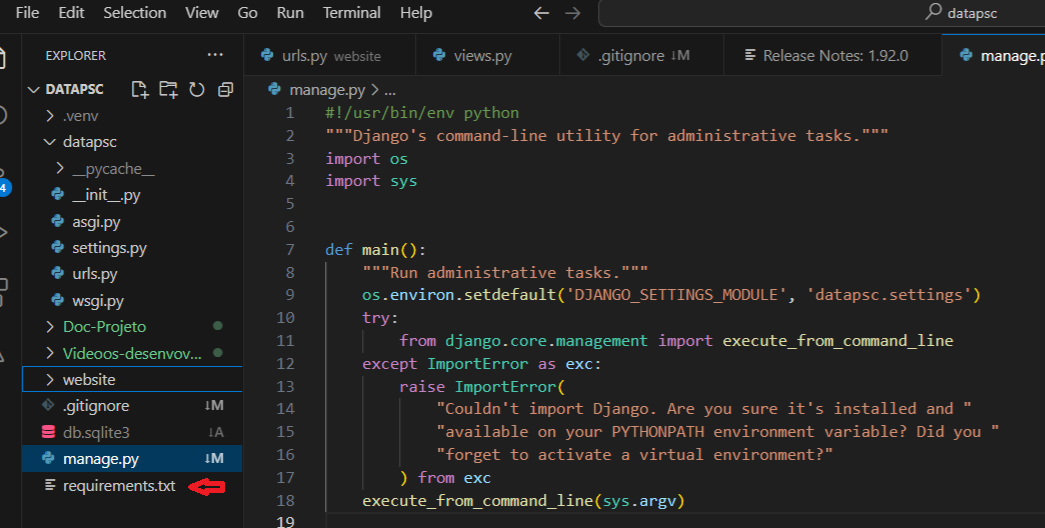
Provavelmente dará um erro porque o Env está vazio, próximo passo:

1. Instalar o Django no virtual ENV: pip install Django
2. Teste: No terminar digitar python manage.py runserver

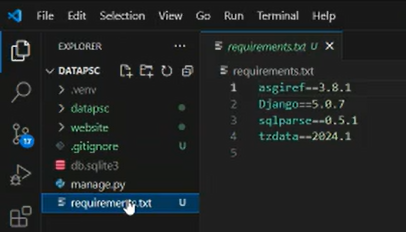


1. Instalar o freeze pelo terminal
2. pip freeze > requiriments.txt



****

1. os arquivos **bibliotecas** (instaladas que o Django utiliza) estão descritas dentro do requirements



Necessário para rodar no mac precisa instalar no terminal do Mac

Comando -> pip install -r requirements.txt

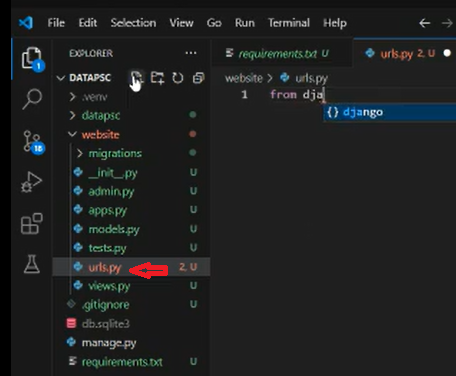
**ESSA ETAPA E O MINIMO PARA INICIAR UM PROJETO**

**Fase 2**

Criando página views

14) Dentro do website criar URL

a) New file – urls.py



1. dentro da **url** escrever os códigos
2. from django.urls import path
3. from website import views
4. urlpatterns = [
5. path('', views.index)
6. ]

15) dentro do datapsc tem a url que contem todas as urls do site, necessário incluir tudo dentro desta lista.

from django.contrib import admin

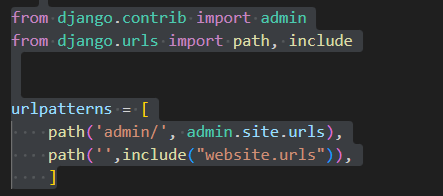
from django.urls import path, include

urlpatterns = [

    path('admin/', admin.site.urls),

    path('',include("website.urls")),

]



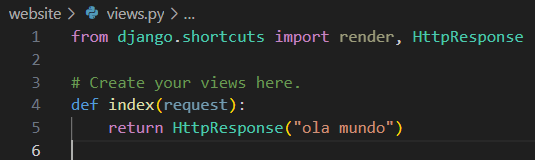
16) Dentro da views.py na pasta migrations colocar o código a abaixo:

from django.shortcuts import render, HttpResponse

# Create your views here.

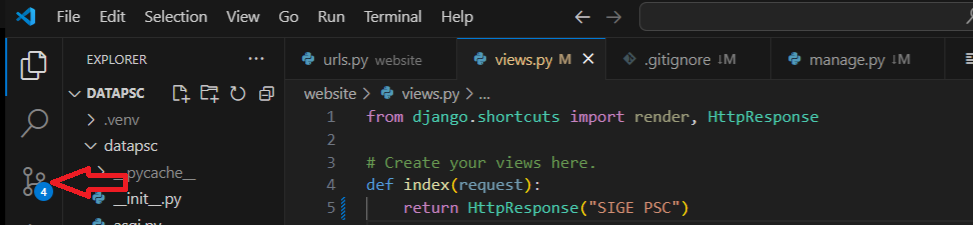
def index(request):

    return HttpResponse("ola mundo")



**17) Fazer o commit:**

a) acessar o ícone do commit

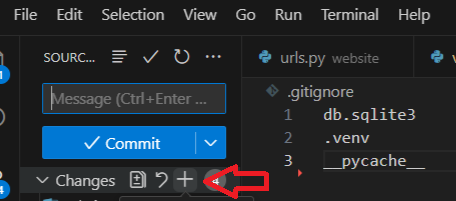


1. Mostra quantos arquivos para serem salvos

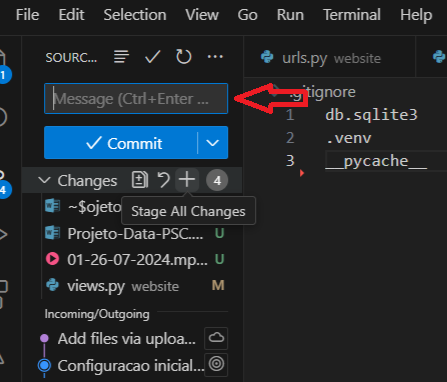
Tela de telefone celular com letras

Descrição gerada automaticamente

1. Clica no botão de + na caixa de texto para salvar tudo;



1. Colocar uma mensagem



1. Após a mensagem clicar no commit e depois no SybcChanges para salvar no GitHub

