**Nastavenie projektového prostredia:**

**1. Inštalácia Django:**

* Ak ešte nemáte nainštalovaný Django, môžete ho nainštalovať pomocou pip (pip install django).
* Vytvorenie nového Django projektu: Použite príkaz django-admin startproject your\_project\_name na vytvorenie základnej štruktúry projektu.
* Vytvorenie Django aplikácie: V rámci projektu vytvorte aplikáciu pomocou python manage.py startapp your\_app\_name.

**2. Návrh databázy a modelov:**

* Definovanie modelov: Vytvorte modely v súbore models.py vo vašej aplikácii. Modely by mali odrážať štruktúru dát, ktoré chcete ukladať (napr. články, autori, kľúčové slová).
* Migrácie: Po definovaní modelov použite python manage.py makemigrations a python manage.py migrate na vytvorenie schémy databázy.
* Výber databázy: Django defaultne používa SQLite. Ak potrebujete, môžete zmeniť nastavenia databázy v settings.py na iný DBMS, ako je PostgreSQL, MySQL alebo iný.

**3. Vytvorenie admin rozhrania:**

* Registrácia modelov: V admin.py vašej aplikácie registrujte modely, aby boli prístupné cez Django admin rozhranie.
* Vytvorenie superužívateľa: Vytvorte superužívateľa pomocou python manage.py createsuperuser a použite ho na prístup k admin rozhraniu.

**4. Práca s Views a Templates:**

* Vytvorenie views: V views.py, definujte logiku pre rôzne funkcie vašej aplikácie (napr. zobrazenie zoznamu článkov, pridanie nového článku).
* Pripojenie URL: V urls.py vašej aplikácie a projektu nastavte cesty (URLs), ktoré budú smerovať na vaše views.
* Vytvorenie šablón: Vytvorte HTML šablóny pre rôzne views. Tieto šablóny budú zobrazovať dáta a formuláre užívateľom.

**5. Integrácia statických súborov:**

* Pridanie CSS/JS: Uložte svoje CSS a JavaScript súbory do priečinka static vo vašej aplikácii. V HTML šablónach ich môžete načítať pomocou {% static 'path/to/your/file.css' %}.
* Použitie statických súborov: V settings.py nastavte STATIC\_URL a STATICFILES\_DIRS podľa potreby.

**6. Testovanie a ladenie:**

* Vytvorenie testov: Napíšte testy pre vaše modely, views a inú logiku v tests.py.
* Spustenie testov: Použite python manage.py test na spustenie testov a overenie funkčnosti.

**7. Nasadenie:**

* Nastavenie pre produkciu: Pred nasadením nezabudnite nastaviť DEBUG = False v settings.py a konfigurovať iné produkčné nastavenia (napr. databázu, statické súbory, bezpečnosť).
* Výber hostingovej služby: Vyberte si hostingovú službu a nasadite svoj projekt podľa ich pokynov (napr. Heroku, AWS, GCP).

**8. Dokumentácia:**

* Píšte dokumentáciu: Zaznamenajte ako vaša aplikácia funguje, ako sa nastavuje a aké sú jej závislosti.

**Dodatočné tipy:**

* Virtuálne prostredie: Používajte virtuálne prostredie (napr. venv) na izoláciu závislostí vášho projektu.
* Verzovací systém: Používajte verzovací systém ako Git pre správu zmien vo vašom kóde.

**GITHUB**

**1. Vytvorenie účtu na GitHub**

* Ak ešte nemáte účet na GitHub, navštívte github.com a zaregistrujte sa.

**2. Inštalácia Git**

* Ak ešte nemáte nainštalovaný Git, stiahnite a nainštalujte ho z git-scm.com. Po inštalácii skontrolujte inštaláciu otvorením terminálu (alebo príkazového riadku) a zadajte git --version.

**3. Nastavenie Git**

* Pred prvým použitím Gitu je potrebné nastaviť vaše meno a e-mailovú adresu. Toto sa používa v každom vašom commit správe. Otvorte terminál a zadajte nasledujúce príkazy (nahraďte svojim menom a e-mailom):
* git config --global user.name "Vaše Meno"
* git config --global user.email "vášemail@príklad.com"

**4. Vytvorenie nového repozitára na GitHub**

* Prihláste sa do svojho účtu na GitHub.
* Kliknite na "+" ikonu v pravom hornom rohu a vyberte "New repository".
* Zadajte názov repozitára, prípadne krátky popis.
* Vyberte, či má byť repozitár verejný alebo súkromný.
* Môžete zaškrtnúť "Initialize this repository with a README", ak chcete.
* Kliknite na "Create repository".

**5. Klonovanie repozitára**

* Na svojom počítači otvorte terminál a prejdite do priečinka, kde chcete mať projekt. Potom použite príkaz git clone, aby ste naklonovali práve vytvorený repozitár:
* git clone https://github.com/vasemeno/vasrepozitar.git
* Nahraďte URL adresou vášho repozitára.

**6. Pridanie projektových súborov do repozitára**

* Presuňte svoje súbory projektu do naklonovaného priečinka. Potom použite nasledujúce príkazy, aby ste ich pridali do vášho repozitára:
* cd vasrepozitar
* git add .
* git commit -m "Prvý commit"

**7. Push (odoslanie) zmeny na GitHub**

* Na odoslanie vašich zmien (commitov) na GitHub použite príkaz git push:
* git push origin master

**8. Pravidelná práca s Git**

* Práca na zmene: Vytvorte novú vetvu pre každú novú funkciu alebo opravu.
* Commit: Po dokončení práce na zmene, použite git add a git commit na uloženie zmien.
* Aktualizácia: Pravidelne používajte git pull, aby ste si stiahli najnovšie zmeny z hlavnej vetvy.
* Zlúčenie: Po dokončení práce na vetve, zlúčte ju späť do hlavnej vetvy.

**9. Používanie .gitignore**

* Súbor .gitignore vám umožňuje definovať súbory alebo priečinky, ktoré by nemali byť sledované Gitom (napríklad súbory prostredia, heslá, atď.). Vytvorte súbor .gitignore vo vašom hlavnom adresári a pridajte do neho pravidlá podľa potreby.

**10. Pravidelné zálohovanie**

* Pravidelne pushujte svoje zmeny na GitHub, aby ste mali zálohu svojej práce a umožnili spoluprácu s inými.

Týmto spôsobom môžete bezpečne ukladať a spravovať svoj kód na GitHub. Nezabudnite, že Git a GitHub majú mnoho ďalších funkcií a možností, ktoré môžete využiť pre efektívnejšiu prácu na projekte.