**Webové stránky na straně serveru**

**1. Co znamenají webové stránky na straně serveru?**

Webové stránky na straně serveru jsou stránky, které generují svůj obsah na serveru předtím, než jsou odeslány uživateli. Server provádí veškeré zpracování (například načítání dat z databází nebo personalizaci obsahu) a odesílá uživateli hotový výstup, obvykle ve formátu HTML nebo JSON. Tyto stránky jsou **dynamické** a dokážou reagovat na různé potřeby uživatelů.

**Klíčové pojmy:**

* **Statické stránky:** Obsah je pevně uložený v HTML souborech.
* **Dynamické stránky:** Obsah je generován na základě vstupu uživatele nebo dalších dat.

**2. Jak webové stránky na straně serveru fungují?**

1. **Uživatel odešle požadavek:**
   * Například klikne na odkaz nebo odešle formulář.
2. **Server požadavek přijme:**
   * Server obdrží požadavek přes HTTP nebo HTTPS.
3. **Zpracování na serveru:**
   * Server může:
     + Načíst data z databáze.
     + Zkontrolovat uživatelské údaje.
     + Provést potřebné výpočty nebo logiku.
4. **Generování odpovědi:**
   * Server vytvoří dynamický obsah (např. HTML stránku) a odešle jej uživateli.
5. **Prohlížeč zobrazuje stránku:**
   * Uživatel vidí stránku, která odpovídá jeho požadavku.

**3. Příklady použití:**

* **E-shopy:**
  + Data o produktech, cenách a dostupnosti se načítají dynamicky.
* **Sociální sítě:**
  + Obsah (např. příspěvky, komentáře) je personalizován na základě přihlášeného uživatele.
* **Formuláře a ověřování:**
  + Při přihlášení uživatele server ověřuje zadané údaje.
* **Redakční systémy:**
  + Správa a zobrazování článků.

**4. Technologie na straně serveru:**

Pro tvorbu webových stránek na straně serveru se používají různé jazyky a frameworky. Mezi nejčastější patří:

* **PHP:**
  + Tradiční jazyk pro tvorbu webových stránek.
  + Jednoduchý na naučení, ideální pro začátečníky.
* **Python:**
  + Frameworky: **Django**, **Flask**.
  + Čitelný kód a vhodný pro robustní aplikace.
* **Java:**
  + Frameworky: **Spring**, **JSF**.
  + Vhodná pro rozsáhlé a bezpečné aplikace.
* **Node.js:**
  + Používá JavaScript i na straně serveru.
  + Rychlé a škálovatelné řešení.
* **Ruby on Rails:**
  + Rychlý a efektivní vývoj aplikací.

**5. Výhody a nevýhody:**

| **Výhody** | **Nevýhody** |
| --- | --- |
| * Dynamický a personalizovaný obsah | * Vyšší nároky na výkon serveru |
| * Možnost práce s databázemi | * Složitější nastavení |
| * Vyšší bezpečnost díky zpracování na serveru | * Potřeba programovacích znalostí |

**6. Praktický příklad:**

**Přihlašovací systém:**

1. Uživatel zadá uživatelské jméno a heslo.
2. Server zkontroluje údaje proti databázi.
3. Pokud jsou správné, zobrazí personalizovanou stránku (např. "Vítejte, Karle!").
4. Pokud jsou špatné, vrátí chybu ("Neplatné přihlašovací údaje.").

**Co je Flask?**

* Flask je **mikroframework** pro tvorbu webových aplikací v Pythonu.
* Je **lehký a flexibilní**, což znamená, že poskytuje jen základní nástroje pro webový server, a vše ostatní si můžete přidat dle potřeby.
* Flask je oblíbený pro své jednoduché použití, čitelnost a snadnost rozšíření.

**Hlavní vlastnosti Flasku:**

1. **Lehký a modulární:**
   * Obsahuje pouze základní komponenty, což usnadňuje začátečníkům práci.
2. **Routing:**
   * Umožňuje přesměrování URL na konkrétní funkce v Pythonu.
3. **Šablonovací systém Jinja2:**
   * Umožňuje dynamicky vytvářet HTML stránky na základě dat.
4. **Podpora rozšíření:**
   * Například pro práci s databázemi (SQLAlchemy), autentizaci a další.

**Jak Flask funguje?**

1. **Server přijímá požadavky:**
   * Flask naslouchá HTTP požadavkům (GET, POST, atd.).
2. **Router zpracovává požadavky:**
   * Každá URL je přesměrována na konkrétní funkci v Pythonu.
3. **Funkce vrací odpověď:**
   * Odpověď může být HTML stránka, JSON data, nebo jen textový výstup.
4. **Prohlížeč vykreslí výsledek:**
   * Prohlížeč zobrazí odpověď, kterou Flask vygeneroval.

**V čem se Flask dá vytvořit?**

Flask můžete vytvořit v libovolném textovém editoru nebo IDE, které podporuje Python. Mezi oblíbené možnosti patří:

* **Visual Studio Code (VS Code)**: Díky rozšířením je ideální pro práci s Pythonem.
* **PyCharm**: Výkonné IDE, které usnadňuje práci na větších projektech.
* **Jupyter Notebook**: Pro rychlé testování funkcí Flasku.

**Postup nastavení virtuálního prostředí pro Flask**

**Krok 1: Instalace Pythonu**

1. Stáhněte a nainstalujte Python z <https://www.python.org/downloads/>.
2. Během instalace zaškrtněte možnost **„Add Python to PATH“**, aby byl Python dostupný z příkazového řádku.

**Krok 2: Vytvoření složky pro projekt**

1. Vytvořte složku, kde budete mít svůj projekt Flask, například:

*C:\Users\VašeJméno\FlaskProject*

**Krok 3: Otevření terminálu**

1. Otevřete příkazový řádek nebo terminál (např. ve VS Code pomocí **Ctrl + ;**).
2. Přesuňte se do složky projektu pomocí příkazu:

*cd C:\Users\VašeJméno\FlaskProject*

**Krok 4: Vytvoření virtuálního prostředí**

1. V terminálu zadejte příkaz:

*python -m venv venv*

Tento příkaz vytvoří složku venv, která obsahuje virtuální prostředí.

**Krok 5: Aktivace virtuálního prostředí**

1. **Windows**:
   * Zadejte:

*venv\Scripts\activate*

1. **Mac/Linux**:
   * Zadejte:

*source venv/bin/activate*

1. **Po aktivaci by se v terminálu mělo objevit něco jako:**

*(venv) C:\Users\VašeJméno\FlaskProject>*

**Krok 6: Instalace Flasku**

1. Ujistěte se, že je aktivní virtuální prostředí (vidíte „(venv)“ na začátku řádku v terminálu).
2. Nainstalujte Flask příkazem:

*pip install flask*

1. Ověřte, že Flask je nainstalován:

*pip show flask*

Tento příkaz zobrazí verzi Flasku a jeho umístění.

**Krok 7: Vytvoření základní Flask aplikace**

1. V kořenové složce projektu vytvořte nový soubor app.py.
2. Do souboru vložte tento základní kód:

*from flask import Flask*

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

return "Vítejte na Flask aplikaci!"

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**Krok 8: Spuštění Flask aplikace**

1. Ujistěte se, že jste stále v aktivovaném virtuálním prostředí.
2. Spusťte aplikaci příkazem:

*python app.py*

1. Flask vypíše informaci:

*\* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)*

1. Otevřete prohlížeč a přejděte na:

*http://127.0.0.1:5000*

Zobrazí se zpráva: **"Vítejte na Flask aplikaci!"**

**Krok 9: Ukončení virtuálního prostředí**

1. Po dokončení práce ukončete virtuální prostředí příkazem:

*deactivate*