



Prirodoslovno-matematički fakultet
Matematički odsjek
Sveučilište u Zagrebu

RAZVOJ WEB-APLIKACIJA

Predavanje 03 - Uvod u Flask

3. listopada 2025.

Sastavio: Zvonimir Bujanović



Za razvoj web-aplikacija na ovom kolegiju ćemo koristiti **Flask**.

- Flask je razvojni okvir za web-aplikacije napisan u Pythonu.
- Nastao 2010. godine, uz **Django** najpopularniji okvir za Python.
- Flask je **minimalistički** razvojni okvir:
 - Web-aplikacija se može sastojati već od samo jedne datoteke.
 - Jedine ugrađene komponente su **router** i HTML predlošci u formatu **Jinja2**.
 - Za sve ostalo (npr. autentifikacija, interakcija s bazom podataka, session) Flask daje potpunu slobodu: samostalna implementacija ili korištenje gotovih ekstenzija.
- Literatura:
 - **Službena** web-stranica projekta.
 - **Flask Mega-Tutorial**, autora Miguela Grinberga.
 - **GeeksforGeeks Tutorial**.

Oprez

Cilj ovog kolegija je naučiti **općenite koncepte** web-programiranja, a ne jednu specifičnu tehnologiju!

- Zbog toga nećemo uvijek koristiti gotove postojeće ekstenzije za Flask, nego ćemo sami implementirati neku funkcionalnost.
- Zbog toga nećemo uvijek slijediti preporuke službene dokumentacije ako su specifične samo za Flask.

Python

Web-aplikacije u Flasku pišemo u Pythonu.

- Za ovaj kolegij je potrebno minimalno predznanje Pythona.
- Ekspresni pregled možete pronaći na web-stranicama kolegija.

Flask - Instalacija i virtualno okruženje

Projekti u Pythonu se obično razvijaju pomoću virtualnog okruženja.

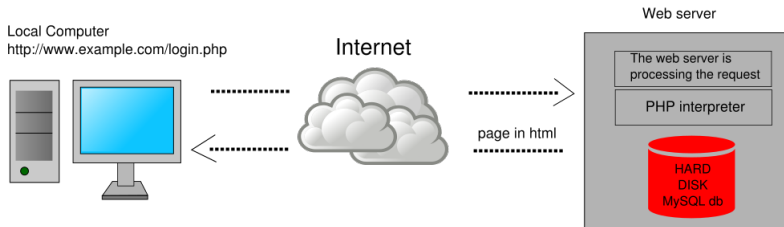
- Okruženje za ovaj kolegij je pripremljeno u praktikumima pod Linuxom.
- Pokrenuti ga iz konzole možete ovako:

```
anaconda_setup  
mamba activate rwa
```

- Identično okruženje možete instalirati na svoje računalo prema **ovim uputama**.
- Za svaku našu aplikaciju ćemo koristiti isto okruženje. U njemu je instalirano i nekoliko popularnih ekstenzija za Flask.

Na ovom kolegiju podrazumijevamo korištenje Linuxa, što je i inače standard za razvoj web-aplikacija.

Uloga web-servera (ponovno)



- Klijent se spaja na server; može slati i izvjesne parametre (na primjer, korisničko ime ili lokaciju).
- Server šalje klijentu tražene datoteke:
 - pronalazi ih u svom datotečnom sustavu ("statičke datoteke");
 - generira ih na temelju dobivenih parametara ("dinamičke web-stranice")
 - radi upite na bazu podataka; pokreće aplikacije i skripte.
- Web-browser na klijentu prikaže "downloadane" datoteke.
- Komunikacija se odvija putem HTTP protokola.

Web-server za aplikacije u Flasku

Flask dolazi s vlastitim web-serverom.

- Namijenjen je samo za razvoj web-aplikacija.
- Omogućava debugiranje, prijavljuje greške.
- Nije namijenjen za produkciju.
- Koristit ćemo ga za pokretanje naših web-aplikacija.

U produkciji, aplikacije u Flasku koriste druge web-servere.

- Apache + `mod_wsgi`.
- Nginx + `gunicorn`.
- Amazon AWS, Microsoft Azure, Google Cloud podržavaju aplikacije u Flasku.

Popularni web-serveri

- Apache (Apache Software Foundation)
 - najrašireniji web-server
 - prvo izdanje 1995. godine; aktualna verzija 2.4
 - open-source, dostupan za velik broj platformi
 - koristi sustav modula za proširenje funkcionalnosti - `mod_php`, `mod_wsgi` (za web-aplikacije u Pythonu), `mod_ssl`, ...
 - za svakog klijenta stvara se novi CPU proces
↪ ograničen broj simultanih klijenata, za svakog se troši nezanemariva količina CPU/RAM
- Nginx
 - noviji open-source web-server, prvo izdanje 2004.
 - bolje performanse od Apache-a
 - dizajniran za tzv. **C10K problem** - 10000 simultanih konekcija
 - koristi asinkroni "event-driven" pristup
↪ broj procesa je jednak broju jezgri procesora, a komunikacija s klijentima se odvija kroz sustav događaja.
 - Za Flask aplikacije se obično koristi kao **reverse proxy** u kombinaciji s **gunicorn-om**.

Flask - Minimalna aplikacija

Svaki zadatak/aplikaciju u Flasku ćemo staviti u svoj direktorij.

- Glavna datoteka aplikacije se uvijek zove `app.py`.
- Minimalni sadržaj te datoteke je sljedeći:

```
1 from flask import Flask;
2
3 app = Flask( __name__ );
```

- Ovim je stvorena nova web-aplikacija koja nema niti jednu (dinamičku) pod-stranicu.
- Flaskov razvojni web-server pokrećemo iz konzole u istom direktoriju:

```
flask run --debug
```

- Ovo pokreće web-aplikaciju na adresi <http://127.0.0.1:5000>
- Imena datoteke `app.py` i varijable `app` su konvencije koje olakšavaju pokretanje servera bez dodatnih parametara. [Može i drugačije.](#)

Po konvenciji, poddirektorij `static` sadrži statične dijelove aplikacije.

- Tu ćemo smještati slike, CSS, JavaScript datoteke.
- Tu možemo staviti i statične HTML dokumente - one koje nisu rezultat programa u Pythonu i ne ovise o varijablama u Pythonu.
- Bilo kojoj datoteci (npr. `slika.jpg`) iz tog direktorija možemo direktno pristupiti preko adrese oblika `http://127.0.0.1:5000/static/slika.jpg`.
- Datotekama iz drugih direktorija **ne možemo pristupiti** preko URL.

Napravite statičnu Flask aplikaciju:

- U HTML datoteci prikažite neku sliku te jedan paragraf teksta.
- Font prilagodite pomoću CSS datoteke.

Pokrenite Flaskov web-server i prikažite HTML datoteku u browseru.

Pratite što se događa u konzoli s pokrenutim web-serverom kada pristupate datotekama u browseru.