

Projekt R

Ak. god. 2023./2024.

Pičibangoš sastančenje

Dokumentacija projekta

Sudionici: *Filip Kovač, Filip Barić*

Datum predaje: 26.1.2024.

Nastavnik: *izv. prof. dr. sc. Vladimir Čeperić*

Sadržaj

1	Opis projektnog zadatka	2
1.1	Opis aplikacije	4
2	Specifikacija programske potpore	6
2.1	FastAPI	6
2.2	Streamlit	6
2.3	BeeWare	7
2.4	Programski zahtjevi	8
3	Zaključak i budući rad	10
	Popis literature	12

1. Opis projektnog zadatka

Uvod

Osnova projekta je bilo razviti programsku podršku za *Pičibangoš*, web aplikaciju koja predstavlja online planer i nudi korisnicima opcije ugovaranja sastanaka, stvaranja evenata i timova, i online komunikaciju sa prijateljima pojedinačno ili unutar tima. Također, još jedan od zadataka je bio istražiti mogućnosti koje nudi programski jezik Python sa framework-ovima Streamlit i BeeWare u razvijanju frontend-a web aplikacije i u povezivanju sa backendom, isto tako građenim pomoću pythonovog framework-a FastAPI.

Ciljevi

- Istražiti mogućnosti razvoja programske potpore za web pomoću programskog jezika Python
- Istražiti Pythonove framework-e za razvoj web aplikacije
- Odabir Pythonovih framework-a
- Analiza primjene
- Zaključak na temelju korištenih tehnologija

Python

Python je jedan od najpopularnijih i najučinkovitijih programskih jezika koji sadrži goleme library-e i framework-e za gotovo svaku tehničku domenu. Pythonovi framework-ovi automatiziraju implementaciju mnogih zadataka i daju programerima dobru strukturu za razvoj aplikacija. Svaki framework dolazi s vlastitom kolekcijom modula ili paketa koji značajno skraćuju vrijeme razvoja.

Model razvoja

Životni ciklus razvoja aplikacija daje proces kojim se aplikacije razvijaju, a modeli razvoja aplikacija daju pristup provedbi tog procesa. Ti modeli opisuju kako se sve faze životnog ciklusa procesa razvoja softvera spajaju u softverski projekt. Postoji nekoliko uobičajenih popularnih modela razvoja i stotine, ako ne i tisuće, njihovih iteracija. Model razvoja softvera opisuje što će se raditi ali ostavlja puno prostora načinu kako će se raditi. Potrebno je odabrati odgovarajući model ovisno o veličini projekta. Za male projekte je vodopadni model bez dodatne dokumentacije i bez restrikcija iterativnog pristupa dobar izbor, te smo u ovom projektu odlučili taj i koristiti.

1.1 Opis aplikacije

Uvod

Aplikacija Pičibangoš je osmišljena kao planer gdje korisnik ima opciju ugovarati sastanke i događaje, bilo privatne ili poslovne svrhe, te na taj način voditi evidenciju obaveza i događaja. Također aplikacija nudi opciju i povezivanja, te komuniciranja se sa kolegama, pri čemu korisnik na taj način može vidjeti obaveze kolega s kojima je povezan ili kao prijatelj ili preko tima, te time može lakše upravljati i dogovarati događaje koji se tiču više ljudi. Stoga, cilj aplikacije ja olakšati svakodnevnu

komunikaciju i suradnju sa kolegama i prijateljima, te time uštediti vrijeme koje bi se neizbježno izgubilo pri dogovaranju sastanaka kroz raspitivanje o obavezama pojedinaca.

Korištenje aplikacije

Nakon svake uspješne prijave korisnik dobiva pregled svih svojih obaveza koje ima unutar bilo kojeg tima, kao i obaveza koje ima sa pojedinačnim korisnicima na osobnom kalendaru. Iznad kalendara su mu također ponuđene mogućnosti stvaranja novih događaja, timova ili pronalaženja novih prijatelja.

Pri odabiru jednog od timova u koje korisnik uključen, kalendar se prilagođava te preciznije prikazuje obaveze koje korisnik ima unutar tima, kao i obaveze koje imaju njegovi timljani. Na taj način korisnik može preciznije odabrati vrijeme u kojem bi mogao kreirati novi događaj, a kojem bi mogli prisustvovati svi timljani. Također korisnik može predložiti jednog od svojih prijatelja ko novog timljana, što odobrava admin koji je stvorio tim, te na temelju toga se i kalendar ažurira.

Korisnik pri stvaranju novog događaja daje ime događaja, odabire vrijeme u kojem će se spomenuti događaj odvijati i sudionike koje poziva na događaj, bilo pojedinačno navedene ili kroz navođenje timova u kojima su oni prisutni, a korisnik je isto tako sudionik.

Pri stvaranju novih timova, korisnik navodi ime tima, njegov opis te odabire prijatelje sa kojima želi stvoriti navedeni tim.

Ako korisnik želi dodati novog prijatelja, to radi kroz upisivanje imena korisnika u aplikaciji, te kroz slanje zahtjeva prijateljstva, kojeg druga strana mora potvrditi. Na taj način korisnici postaju prijatelji, te mogu stvarati i gledati zajedničke obaveze, kao i imati uvid u obaveze prijatelja.

2. Specifikacija programske potpore

Python framework je zbirka Python modula koja pruža skup zajedničkih funkcija koje se mogu koristiti kao struktura za izgradnju aplikacija bilo koje vrste. Framework-ovi su osmišljeni kako bi pojednostavili razvojni proces dajući općenite smjernice o tome kako bismo trebali graditi softver i apstrahirali neke od složenijih ili ponavljajućih zadataka, što stoga omogućuje jedinstvenu i prilagođenu logiku za aplikacije.

Stoga, nakon istraživanja, odlučili smo koristiti navedene Python framework-e u izgradnji aplikacije:

1. FastAPI
2. Streamlit
3. BeeWare

2.1 FastAPI

FastAPI je moderan web framework koji je kreirao Sebastián Ramírez za izgradnju RESTful API-ja u Pythonu. Ubrzo je stekao popularnost među programerima zbog svoje jednostavne upotrebe, brzine koja je usporediva sa NodeJS-om i Go-om, i robusnosti. Temelj se zasniva na Pydantic-u, te omogućuje validaciju, serijalizaciju i deserijalizaciju podataka. Također automatski generira OpenAPI dokumentaciju za API-je izgrađene s njim.

Upravo radi navedenih razloga, FastApi je postao najbolji odabir za izgradnju backend-a aplikacije.

2.2 Streamlit

Streamlit, kao otvoreni Python framework, predstavlja snažan alat koji može transformirati projekte u upečatljive web aplikacije. Ovaj okvir posebno privlači znanstvenike koji se bave podacima i inženjere strojnog učenja, pružajući im mogućnost

da svoje radove brzo i jednostavno prenesu na web platformu.

Streamlit se ističe svojom sposobnošću stvaranja atraktivnih web aplikacija s minimalnim naporom. U nekoliko redaka koda, korisnici mogu transformirati svoje analize i modele u sučelja koja su intuitivna i lako razumljiva. Ova karakteristika čini Streamlit iznimno korisnim za brze iteracije, testiranje ideja i dijeljenje rezultata sa širom publikom.

Također, raznolikost widgeta i grafičkih elemenata unutar Streamlita omogućuje korisnicima da prilagode svoje aplikacije prema specifičnim potrebama, čineći ih pristupačnima čak i početnicima u Pythonu. Aktivna zajednica Streamlit korisnika pruža bogatstvo resursa, od korisničkih vodiča do gotovih rješenja, čime olakšava učenje i implementaciju novih funkcionalnosti. Sve ove prednosti čine Streamlit ne samo alatom za brzi razvoj, već i resursom koji promiče laku prilagodbu i inovaciju u svijetu web razvoja.

Unatoč tim prednostima, Streamlit nije uvijek najfleksibilniji alat. Njegova snaga leži u jednostavnosti i brzini razvoja, ali to može rezultirati ograničenjima za složenije scenarije ili specifične zahtjeve. Programeri koji traže dublju prilagodbu u dizajnu aplikacija brzo će naići na ograničenja koja neće moći zaobići.

Dodatno, jedan od izazova je prilagodba pojedinih komponenti i njihovih međusobnih interakcija. Kada se koriste složenije komponente može doći do sukoba i neusklađenosti. Različiti widgeti ili interaktivni elementi stvoreni od strane različitih korisnika mogu imati različite zahtjeve i uvjete, što rezultira kompatibilnošću koja nije uvijek zajamčena.

Usorkos navedenim izazovima, Streamlit se izdvaja kao iznimno koristan alat za brzi razvoj web aplikacija u Pythonu. Njegova jednostavnost i brzina omogućuju korisnicima transformaciju analiza i modela u privlačna korisnička sučelja s minimalnim naporom, dok bi zahtjevi za visokom prilagodljivošću i složenim interakcijama možda zahtijevali razmatranje drugih alata. Stoga je ključno pažljivo odabrati Streamlit kao alat, uzimajući u obzir karakteristike projekta.

2.3 BeeWare

Python, kao programski jezik, ostvario je značajan uspjeh zbog svoje pristupačnosti za početnike, ali i zbog moćnih mogućnosti koje pruža iskusnim programerima. U tom kontekstu, projekt BeeWare izdvaja se kao ambiciozan pokušaj iskorištavanja snage Pythona kako bi omogućio razvoj aplikacija s izvornim korisničkim sučeljima, prilagođenih korisnicima svih razina vještina.

Jedinstvenost BeeWare projekta ogleda se u njegovom sveobuhvatnom pristupu razvoju mobilnih i web aplikacija. Cilj mu je stvoriti ekosustav alata i biblioteka koji će omogućiti korisnicima korištenje Pythona na različitim platformama bez složenih prilagodbi koda. Ovaj projekt ističe se u nekoliko ključnih aspekata.

Prvo, BeeWare pruža alate za univerzalnost, omogućavajući rad Pythona na različitim uređajima, što širi spektar primjena od mobilnih do stolnih uređaja. Osim toga, projekt nudi alate za pakiranje Python projekata, pojednostavljujući proces izvođenja na različitim uređajima bez potrebe za kompleksnim postupcima prilagodbe.

Važan aspekt BeeWare projekta je i pristup izvornim widgetima. Razvoj aplikacija s izvornim korisničkim sučeljem zahtijeva pristup specifičnim widgetima i mogućnostima uređaja, a BeeWare pruža odgovarajuće knjižnice koje olakšavaju ovaj proces.

Nadalje, BeeWare se posvećuje podršci za razvoj i analizu projekata. Osim alata za razvoj, projekt se fokusira na otklanjanje pogrešaka, analizu i implementaciju kako bi korisnicima pružio cjelovito iskustvo u razvoju aplikacija.

BeeWare ne samo da predstavlja značajan doprinos Python ekosustavu, već i raste u popularnosti unutar Python zajednice. Njegova svestranost i cilj omogućavanja jednostavnog razvoja aplikacija na različitim platformama čine ga obećavajućim projektom koji ima potencijal utjecati na način na koji razmišljamo o mobilnom i stolnom korisničkom softveru. Kroz suradnju i podršku zajednice, BeeWare gradi most između Pythona i izvornih korisničkih sučelja, postavljajući temelje za budućnost aplikacijskog razvoja.

2.4 Programski zahtjevi

Neovisno o framework-ovima, aplikacija mora ispunjavati sljedeće programske zahtjeve:

- Sustav nužno mora podržavati rad više korisnika u stvarnom vremenu
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu
- Sustav treba biti implementiran kao web aplikacija koristeći objektno-orijentirane jezike
- Neispravno korištenje korisničkog sučelja ne smije narušiti sustav
- Sustav treba imati jasno i intuitivno sučelje
- Sustavu se može pristupiti iz javne mreže
- Zaporke korisnika ne smiju se zapisivati u bazu u plain text formatu

3. Zaključak i budući rad

Ideja projektnog zadatka je bila napraviti web aplikaciju u Pythonu koristeći moderni web framework FastAPI te Open Source biblioteke poput Streamlita i BeeWare kako bi ih proanalizirali i upoznali se s njima. Aplikaciju smo kreirali koristeći Streamlit, a ostale biblioteke smo detaljno proučili.

FastAPI se istaknuo kao odličan framework za sastavljanje temelja aplikacije. Jednostavan je za korištenje, posebno za one koji su već upoznati Pythonom, dizajniran je kako bi pružao brze i efikasne obrade HTTP zahtjeva te uključuje ugrađene mehanizme sigurnosti.

BeeWare nije samo jedan alat, već skup alata i biblioteka koji omogućuju pisanje mobilnih, web i stolnih aplikacija koristeći Python. Ističe se time što nudi mogućnost korištenja Pythona na različitim platformama bez potrebe za mijenjanjem koda. Unatoč tomu, Streamlit se sa svojom jednostavnošću, brzinom razvoja i većom zajednicom koja mu je posvećena jasno ističe kao bolji odabir za rad na web aplikaciji.

Streamlit se pokazao kao iznimno sposoban alat za stvaranje web aplikacija. Praktičan je, pruža razne widgete za lakšu interakciju s podacima, lako se implementira te ima aktivnu i rastuću zajednicu koja doprinosi njegovom razvoju. Osim toga, Streamlit je jedina biblioteka, od svih koje smo analizirali, prilagođena za stvaranje web aplikacija za podatkovnu znanost i strojno učenje.

Radom na aplikaciji otkrile su se i mnogobrojne mane navedenih open source biblioteka. Što su alati jednostavniji za koristiti, to su ograničeniji i nestabilniji. Komunikacija s bazom ili nekim drugim frameworkom se teško postiže. Kompliciraniji elementi i widgeti rijetko funkcioniraju zajedno bez grešaka, zato što su mnogi kreirani od različitih korisnika, pa njihova kvaliteta ovisi o vještini programiranja korisnika koji ih je napravio. Streamlit je pogotovo pogođen tim problemima, no zbog velike zajednice ljudi koja radi na njemu, oni su ograničeni samo na njegove

najsloženije dijelove.

Uzevši u obzir sve vrline i mane, možemo zaključiti da su open-source biblioteke poput Streamlita i BeeWarea korak u dobrom smjeru za razvoj mobilnih i web aplikacija u Pythonu. Iako imaju svoje mane koje ih sprječavaju da se koriste u kompliciranijim aplikacijama, predstavljaju se kao izrazito sposobni alati za brzo i jednostavno stvaranje aplikacija, vizualiziranje i dijeljenje podataka te prikazivanje modela i rezultata strojnog učenja.

Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. FastAPI dokumentacija, <https://fastapi.tiangolo.com/>
5. Streamlit dokumentacija, <https://docs.streamlit.io/>
6. BeeWare dokumentacija, <https://beeware.org/>