

CURRICULUMDSL – ЈСД ЗА ОПИС НАСТАВНИХ ПЛАНОВА И ПРОГРАМА**CURRICULUMDSL – DSL FOR SPECIFYING CURRICULUMS**Stefan Apostolović, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIČKO I RAČUNARSKO INŽENJERSTVO**

Kratak sadržaj – Ovaj rad istražuje primenu jezika specifičnih za domen u oblasti obrazovanja, sa fokusom na jednostavnost korišćenja. Rad obuhvata analizu postojećih rešenja i pruža teorijsku osnovu za razumevanje problema. Uz pomoć jezika specifičnog za domen korisnicima je omogućeno da kreiraju nastavne planove i programe, a uz pomoć backend i frontend aplikacija da kreirane podatke vide na intuitivan način, uz mogućnost da njima upravljaju. Rezultati pokazuju potencijal ovakvih tehnologija da u velikoj meri pomognu prosvetnim radnicima u kreiranju edukativnih sadržaja na brz i efikasan način.

Ključne reči: jezik specifičan za domen, model, meta-model, textX, nastavni plan i program, kurs, modul, test

Abstract – This thesis explores the application of domain-specific languages in the field of education, focusing on ease of use. It includes an analysis of existing solutions and provides a theoretical basis for understanding the topic. Using the domain-specific language end-users can create curriculums, and with the help of the backend and frontend applications they can get an intuitive view of the data they created, with the ability to manage said data. The results demonstrate the potential of these technologies to significantly assist educators in creating curriculums in a quick and efficient way.

Keywords: domain-specific language, model, meta-model, textX, curriculum, course, module, test

1. UVOD

Jezici specifični za domen (JSD, eng. *Domain-Specific Languages – DSL*) predstavljaju tehniku koja omogućava definisanje formalizovanog jezika koji je prilagođen određenom domenu problema. Ovaj rad ima za cilj da istraži primenu JSD-a u kontekstu obrazovanja i predstavi razvoj jednog takvog jezika. Ciljevi rada su istraživanje postojećih softverskih rešenja koja postoje u ovoj oblasti, kao i implementacija JSD-a koji će pokušati da reši napomenute izazove. Pored jezika specifičnog za domen koji je implementiran korišćenjem jezika textX [1], u

sklopu softverskog rešenja su i *backend* server, čija je svrha da podatke dobijene pomoću pomenutog meta-jezika prepakuje u format koji je pogodan *frontend* aplikaciji. *Frontend* aplikacija služi da informacije o nastavnom planu i programu prikaže korisniku na jednostavan i razumljiv način, kao i da mu omogući da njima upravlja.

2. POSTOJEĆA REŠENJA

Najpoznatiji alati u oblasti obrazovanja su VLab [2], radni okvir za modelovanje virtuelnih učionica pomoću razvoja vođenog modelima, Moodle [3], jedan od najpopularnijih sistema za upravljanje učenjem, koji se koristi za kreiranje i upravljanje onlajn kursovima, kao i SCORM [4-5] (eng. *Sharable Content Object Reference Model*), standard za razvoj i integraciju obrazovnog sadržaja u sistemima za upravljanje učenjem.

3. KORIŠĆENI ALATI

Za razvoj aplikacije korišćen je textX [1,6] za kreiranje meta-modela, Uvicorn [7] za implementaciju *backend* servera, FastAPI [8] za implementaciju API-a (eng. *Application Programming Interface*), i React, za razvoj *frontend* aplikacije. U nastavku sledi nešto više o textX-u.

3.1. textX

textX je meta-jezik koji služi za kreiranje jezika specifičnih za domen. U suštini, on omogućava definisanje sintakse i semantike JSD-ova na jednostavan način, koristeći gramatike slične konektno-slobodnim. Zasnovan je na Python-u i koristi Arpeggio biblioteku za parsiranje, što omogućava lako generisanje JSD-ova sa minimalnim naporima. Podržava automatsko definisanje gramatike i meta-modela. Meta-model koji se generiše je zapravo graf Python klase sa vezama između njih. Pored toga, implementirana je automatska konstrukcija modela, i podrška za razrešavanje referenci koje se u njemu nalaze. Sam model je instanca meta-modela, odnosno graf Python objekata,

4. OPIS REŠENJA

Glavne komponente sistema su: jezik specifičan za domen za definisanje nastavnog plana i programa, server čija je svrha da Python objekte koji su deo textX modela prepakuje u format pogodan *frontend* aplikaciji, API čija je svrha da odgovara zahtevima te aplikacije, i sama *frontend* aplikacija, koja služi da kreiran nastavni plan i program prikaže krajnjim korisnicima na pregledan način.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Igor Dejanović, red. prof.

4.1. JEZIK SPECIFIČAN ZA DOMEN

Jedan od glavnih ciljeva pri razvijanju ovog jezika jeste jednostavnost korišćenja. Sledi prikaz i objašnjenje jednog pravila koje se nalazi u gramatici jezika. Pravilo **Kurs** (eng. *Course*) služi za opisivanje kursa koji pripada nastavnom planu i programu. Sastoji se iz sledećih atributa: **Ime** (eng. *Name*), **Opis** (eng. *Description*), **Moduli** (eng. *Modules*), **Test**, **Savet** (eng. *Advice*), **Preduslovi** (eng. *Prerequisites*) – spisak kurseva ili modula koje korisnik mora da pređe pre nego što stekne pravo da pristupi kursu. Pravilo je prikazano na listingu 1.

```
Course:
  'C:'
  'Name:' name=STRING
  'Description:' description=STRING
  'Modules:' modules+=Module
  'Test:' test=Test
  ('Advice:' advice=STRING)?
  ('Prerequisites:' prerequisites*=Prerequisite)?
;
```

Listing 1 – Course pravilo

4.2. BACKEND SERVER

Backend server implementiran je pomoću Uvicorn veb servera. Uvicorn je brz, lagan i efikasan ASGI server (eng. *Asynchronous Server Gateway Interface*) za Python, dizajniran da pokreće asinhronu veb aplikaciju i servere. Na listingu 2 prikazan je kod koji je odgovoran za pokretanje servera.

```
def main():
    logging.info("Application started...")
    uvicorn.run("main:app", log_level="info", reload=True)
```

Listing 2 – Kod za pokretanje servera

Komanda za pokretanje servera može imati više parametara, u ovom slučaju prvi parametar je lokacija aplikacije koju želimo da pokrenemo, drugi postavlja nivo beleženja logova na *info*, a treći govori aplikaciji da ponovo učita server svaki put kada primeti neku promenu u kodu.

4.3. API

API koji služi za upravljanje resursima u aplikaciji implementiran je pomoću FastAPI radnog okruženja. FastAPI je moderan, brz i visoko performantan radni okvir zasnovan na Python-u. Dizajniran je da omogući razvoj aplikacija sa visokim performansama koje su gotovo uporedive sa najbržim okvirima, kao što je Go. Na listingu 3 prikazan je kod odgovoran za definisanje i podešavanje API-ja.

```
origins = [
    "http://localhost:3000"
]

middleware = [
    Middleware(cors.CORSMiddleware, allow_origins=origins,
               allow_methods=["*"], allow_headers=["*"],)
]

app = FastAPI(middleware=middleware)
app.include_router(api.router)
```

Listing 3 – Kod za podešavanje API-ja

Na listingu 4 prikazan je jedan *endpoint*, koji je odgovoran za dobavljanje statistika o korisniku.

```
@router.get("/testResult/userStatistics/{course_name}", response_model=None)
def getUserStatisticsForCourse(course_name: str):
    course = curric.get_course_by_name(course_name)
    if course is None:
        raise HTTPException(status_code=404, detail=f"Course with the name {course_name} not found.")

    response = course.get_user_statistics()

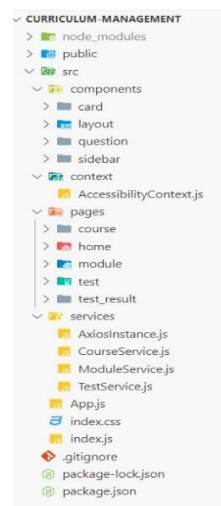
    try:
        json_response = jsonable_encoder(response)
    except Exception as e:
        logging.exception("Exception: {}".format(type(e).__name__))
        logging.exception("Exception message: {}".format(type(e)))
        return

    return JSONResponse(content=json_response)
```

Listing 4 – FastAPI endpoint

4.4. FRONTEND

Frontend aplikacija napisana je uz pomoć React JavaScript biblioteke. Sastoji se od pet stranica, a svaka stranica je sačinjena od više komponenti. Struktura projekta prikazana je na slici 1, a izgled samih stranica biće pokazan u nastavku poglavlja.



Slika 1 – Struktura frontend projekta

4.5. SPECIFIKACIJA PLANA I PROGRAMA PUTEM JEZIKA SPECIFIČNOG ZA DOMEN

Na listingu 5 prikazana je specifikacija nastavnog plana i programa, tačnije samo jedan njen deo, radi čitljivosti.

Videos:

Url: "https://www.youtube.com/watch?v=6WUjbJEJwM"

Description: "Lorem ipsum"

Url: "https://www.youtube.com/watch?v=rCW0dfQ3cwQ"

Description: "Lorem ipsum"

Test:

Name: "Kurs1 Modul1 Test"

Questions:

Q:

text: "Modul tekst1"

type: open_ended

Answer: - "modul odgovor 1"

points: 4

PassCriteria: percentage_required: 85

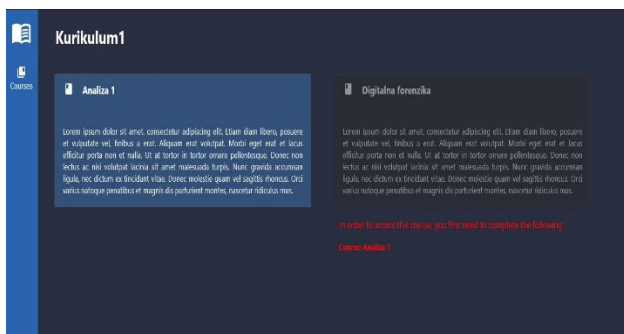
Advice: "Advice module 1"

Listing 5 – Specifikacija dela plana i programa

Kao što se može videti sa listinga, definisan je test čije je ime "Kurs1 Modul1 Test", ima jedno pitanje koje nosi četiri boda, i kao kriterijum za prolaz zahteva da je osvojen procenat tačnih odgovora veći od 85. Pored toga, definisana su i dva video snimka. Svaki snimak ima link, kao i kratak opis.

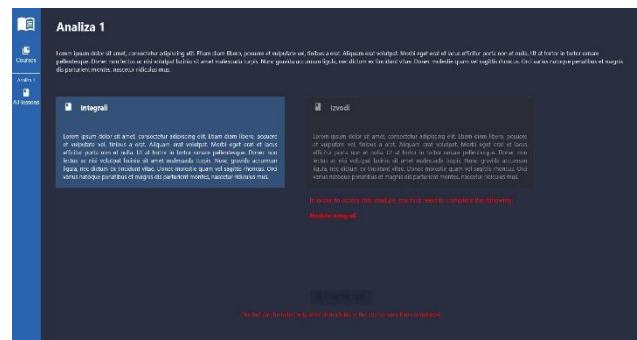
4.6. KORISNIČKI INTERFEJS I FUNKCIONALNOSTI APLIKACIJE

Kada korisnik pokrene aplikaciju, prvo što vidi jeste početnu stranicu, na kojoj su izlistani svi dostupni kursevi. Izgled stranice prikazan je na slici 2.



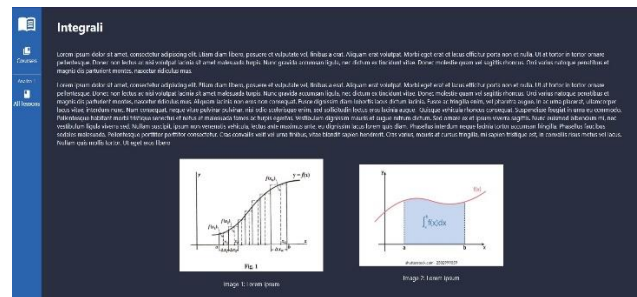
Slika 2 – Početna stranica

Kursu "Digitalna forenzika" se ne može pristupiti, jer zahteva da korisnik prvo završi kurs "Analiza 1", što je naznačeno crvenim slovima ispod kartice za kurs. Klikom na karticu za prvi kurs, prelazimo na stranicu Course, na kojoj su izlistani svi moduli koji mu pripadaju. Slika 3 prikazuje stranicu.

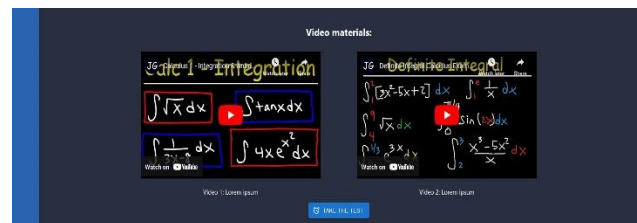


Slika 3 – Stranica Course

Slično kao i na prethodnoj stranici, pristup drugom modulu je zabranjen dok korisnik ne završi modul koji je naznačen podebljanim crvenim slovima. Ono što je novo jeste postojanje dugmeta čijim pritiskom se započinje test. Pošto korisnik nije prešao sve module, ne može da pristupi testu. Klikom na prvi modul prelazimo na stranicu Module (slike 4 i 5).

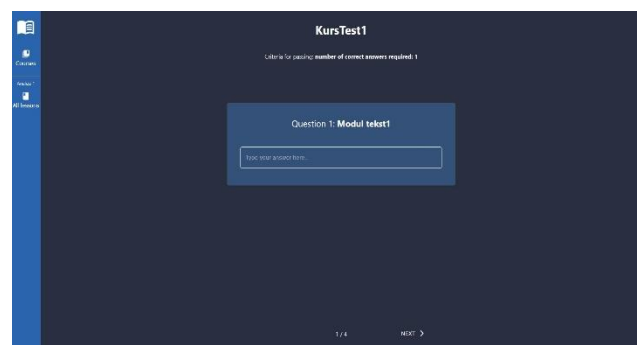


Slika 4 – Stranica Module prvi deo



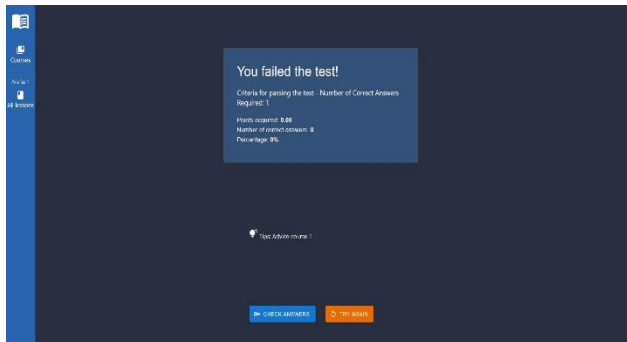
Slika 5 – Stranica Module drugi deo

Na Module stranici pored tekstualnog sadržaja prikazane su i slike [9-10], kao i video materijali [11-12]. Pored toga, svaki modul ima i test, koji se može započeti klikom na dugme "Take the test". Prelaskom na stranicu Test, korisniku je prikazan određeni broj pitanja iz "bazena" pitanja koje test poseduje. Na slici 6 može se videti izgled stranice, i pitanje slobodnog tipa.



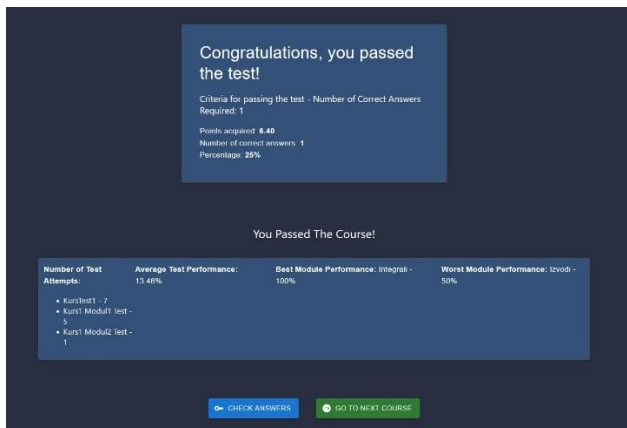
Slika 6 – Test stranica i pitanje slobodnog tipa

Pored pitanja slobodnog tipa, test može imati i pitanja tipa tačno-netačno, pitanja čiji je odgovor broj, pitanja koja mogu imati više tačnih odgovora, i pitanje sa samo jednim tačnim odgovorom. Nakon završetka testa, korisniku se prikazuje TestResult stranica. U slučaju da nije uspeo da položi test (slika 7), na stranici će se nalaziti saveti za napredak, opcija da vidi odgovore na pitanja, kao i mogućnost da pokuša ponovo. Ako se odluči da pokuša ponovo, pitanja će se nasumično izvući iz bazena, a njihov redosled, kao i redosled odgovora izmešati.



Slika 7 – TestResult stranica

Nakon što korisnik pređe prvi modul, moći će da pristupi sledećem. Kada se svi moduli na kursu završe, stiže se pravo na polaganje samog kursa. Kada korisnik uspešno položi kurs, prikazuju mu se osnovni statistički podaci, poput ukupnog broja pokušaja za svaki test, srednja uspešnost na testovima, koji modul je korisniku najbolje išao, a koji najlošije. Ovo je prikazano na slici 8.



Slika 8 – Prikaz statističkih podataka nakon uspešno položenog kursa

5. ZAKLJUČAK

Ovaj rad je fokusiran na razvoj i primenu jezika specifičnog za domen radi opisa nastavnih planova i programa. Generalizovani pristupi često ne zadovoljavaju konkretne potrebe obrazovnog sektora, što dovodi do komplikovanijeg i manje efikasnog procesa upravljanja nastavnim sadržajem. Jezik predstavljen u ovom radu omogućava jasnije i preciznije definisanje nastavnih planova, poboljšava njihovu preglednost i pojednostavljuje modifikacije. Primenom ovog JSD-a u obrazovanju, nastavnici i administratori dobijaju alat koji olakšava kreiranje i prilagođavanje kurikuluma, što dugoročno može uticati na unapređenje kvaliteta nastave.

6. LITERATURA

- [1] И. Дејановић, Р. Вадерна, Г. Милосављевић, Ж. Вуковић, „TextX: A Python tool for Domain-Specific Languages implementation”, Knowledge-Based Systems, vol. 115, Na mreži, 01-01-2017, pristup: 02-10-2024.
- [2] David Ganan, Ana-Elena Guerrero-Roldan, Josep Prieto-Blazquez, Jordi Conesa, “VLab: A domain-specific language for Virtual Classrooms”, Na mreži, 27-10-2014, pristup: 16-08-2024.
- [3] Ajlan Al-Ajlan, Hussein Zedan, “Why Moodle”, Na mreži, 21-11-2008, pristup: 16-08-2024
- [4] SCORM, <https://scorm.com/>, Na mreži, pristup: 16-08-2024
- [5] O. Bohl, J. Scheuhase, R. Sengler, U. Winand, “The sharable content object reference model (SCORM) - a critical review”, Na mreži, 20-03-2003, pristup: 16-08-2024
- [6] И. Дејановић, Jezici specifični za domen, Fakultet tehničkih nauka, Novi Sad, 2021.
- [7] Uvicorn, <https://www.uvicorn.org/>, Na mreži, pristup: 16-09-2024
- [8] FastAPI, <https://fastapi.tiangolo.com/>, Na mreži, pristup: 16-09-2024
- [9] “PRIMITIVE, DEFINITE INTEGRAL, FUNDAMENTAL THEOREM OF INTEGRAL CALCULUS, PROPERTIES OF DEFINITE INTEGRALS”, <https://solitaryroad.com/c364.html>, Na mreži, pristup: 20-09-2024
- [10] “Origins of Integration”, <https://www.semanticscholar.org/paper/Origins-of-Integration-Hammarstr%C3%B6m/ea302ec2107d8668b2d12e8b9fc965217f33322>, Na mreži, pristup: 20-09-2024
- [11] “Calculus 1 - Integration & Antiderivatives”, <https://www.youtube.com/watch?v=6WUjbJEJwM>, Na mreži, pristup: 20-09-2024
- [12] “Definite Integral Calculus Examples, Integration - Basic Introduction, Practice Problems”, <https://www.youtube.com/watch?v=rCWOfQ3cwQ>, Na mreži, pristup: 20-09-2024

Kratka biografija:



Stefan Apostolović rođen je u Nišu 2000. godine. Diplomirao je na Fakultetu tehničkih nauka u Novom Sadu, na smeru Računarstvo i automatika 2023. godine. Master rad na Fakultetu tehničkih nauka u Novom Sadu iz oblasti Elektrotehničko i računarsko inženjerstvo – CurriculumDSL – jezik specifičan za domen za opis nastavnih planova i programa, odbranio je 2024.godine.