

Umjetna inteligencija kao izvor potpore nastavnicima u pripremi i izvođenju nastave

K. Maček Blažeka

Tehnička škola Ruđera Boškovića, Zagreb, Republika Hrvatska

Tehnička škola Čakovec, Zagreb, Republika Hrvatska

kristinka.blazeka@skole.hr

Sažetak - Svjedočimo vidljivo ubrzanom napretku u područjima istraživanja i implementacije umjetne inteligencije (AI) vezano uz procesiranje prirodnih jezika (NLP) i njihovog razumijevanja (NLU). U softverskom inženjerstvu uvedeni su botovi (bots), aplikacije sposobne reagirati na vanjske podražaje od strane korisnika na koje odgovaraju pokretanjem automatiziranih postupaka. Pri tome često uključuju konverzijske sposobnosti, te sa krajnjim korisnikom komuniciraju tekstualnim porukama (chatbots) ili govorom (voicebots). U radu se diskutira kako mogućnost uvođenja umjetne inteligencije, odnosno korištenje chatbota (ChatGPT) tijekom postupka izrade i razrade nastavnih sadržaja za primjer pripreme (i izvođenja) nastave u srednjoškolskom obrazovanju može pomoći kvalitetnijoj pripremi i tako poboljšati procese učenja i poučavanja. U radu se istražuje mogućnost primjene umjetne inteligencije kod oblikovanja sadržaja i tijeka nastavnog sata namijenjenog učenju programiranja. Pri tom se pokušava odgovoriti na neka značajna pitanja kao što su: kako se umjetna inteligencija može iskoristiti za olakšavanje rada ili pomoć u radu nastavnika; kako može pomoći u traženju kreativnih rješenja; kako može pomoći kod izbjegavanja čestih pogrešaka kod učenja početnicima i drugo. Prezentirane mogućnosti upućuju općenito na veliki potencijal. Konačno, u zaključku se demonstriraju preporuke i daje osvrt na moguće aspekte koji upućuju na obećavajuće rezultate sustava umjetne inteligencije da, u ulozu pomoćnika nastavnicima, utječu na povećanje kvalitete njihova rada, a time i cjelokupnog obrazovnog procesa.

Ključne riječi – planiranje i priprema nastave; učenje programiranja; umjetna inteligencija; chatbot;

I. UVOD

Umjetna inteligencija već značajno oblikuje naše živote (popunjavanje radnih mjesta, brojne primjene u vojsci, primjenjuje se otkrivanje plagijata, masovni mediji otkrivaju što volimo...), pod utjecajem smo algoritama i tehnologija sa svrhom da uzimaju i oblikuju što znamo, vidimo i kako zamišljamo svijet. Nova se tehnološka rješenja koriste za prikupljanje, sintezu i primjenu složenih oblika nadzora nad podacima koje koriste ili ih prodaju korporacije, banke, osiguravajuća društva, investitori. To su samo neki od razloga zbog kojih je važno osigurati dubinsku analizu razvoja, rizika i mogućnosti za primjenu umjetne inteligencije u obrazovanju [1] kao i istražiti kako bi to moglo oblikovati i utjecati na budućnost obrazovanja. Poznato je i u nizu

tekstova izloženo kako AI sustav uspješno rješava ispitne zadatke na raznim ispitima.

Kako bi se škole i obrazovni sustav u cjelini trebali prilagoditi širenju alata umjetne inteligencije? Za dio nastavnika OpenAI ChatGPT postao je razlogom za brigu i paniku. Od svoje pojave ovaj je alat postao uzrokom kontroverzi. S jedne strane raste bojazan vezana uz utjecaj umjetne inteligencije na lažiranje prilikom rješavanja raznovrsnih zadataka kod kuće i u školi jer taj alat može uspješno generirati eseje, rješavati probleme iz matematike ili drugih predmeta, prezentirati funkcionalan kod za računalne programe za različite jezike ili platforme, kao i niz drugih zadataka i sve to u vremenu koje se mjeri u sekundama. S druge strane, smatra se i opasnim jer konverzijski AI softver, treniran na enormnim količinama podataka, koji je u mogućnosti voditi realistične konverzije sa ljudima može u određenim situacijama manipulirati korisnicima bez njihovog znanja, npr. zbog marketinških, financijskih, političkih ili čak kriminalnih razloga.

Ovaj rad nastao je kako bi se ispitalo i prezentirale mogućnosti ChatGPT-a u ulozu pomoćnika nastavnicima skraćivanjem vremena potrebnog za pripremu nastave i nastavnih materijala. Dijelevi stranu koja je zauzela stav kako ga (ipak) ne bi trebalo izbacivati iz škola i učionica, u nastavku se iznosi iskustvo da se kroz primjer korištenja ovog alata u kreiranju plana i sadržaja poučavanja za odabrane nastavne sadržaje kod učenja programiranja utvrde i iznesu neka opažanja i iskustvo.

II. ISKUSTVA I (NE)PRIHVATLJIVOST

Kao i drugi napredni sustavi strojnog učenja, ChatGPT pokazuje potencijalne sigurnosne probleme, npr. postoji rizik od napada oglašavanja pri kojem napadač pokušava manipulirati modelom ubacujući zlonamjerne ulaze koji će producirati neispravne ili neželjene izlaze. Zabrinutost se može vezati i uz njegov potencijal da ga se koristi za širenje dezinformacija ili propagande, posebno ako se integrira sa platformama širokog dosega kao što su to društvene mreže [2].

Zbog tih i sličnih razloga, neke su javne škole u SAD promptno početkom 2023. godine blokirale pristup do ChatGPT platforme na školskim računalima i mrežama izbacivši ga iz škola i učionica s obrazloženjem „brige zbog negativnog utjecaja na učenje i zabrinutosti vezane uz sigurnost i točnost sadržaja“. U dijelu škola pristup je onemogućen samo učenicima.

Pozitivna iskustva nastavnika sa ovim sustavom iznijeta su u članku "Don't Ban ChatGPT in Schools, Teach With It" (NYT, 12.1.2023.). Jedan je nastavnik za evaluaciju učeničkih radova od AI sustava u vrlo kratkom vremenu dobio upotrebljiviji i detaljniji odgovor učenicima od onog kojeg bi on sam kao mogao pružiti, a drugi je utvrdio kako je većina pitanja koje mu je AI sustav generirao za analizu zadanog teksta bila upotrebljiva. Ipak, ne vidjevši u AI sustavu pomoćnika, nego konkurenciju, neki su se nastavnici zabrinuli za svoju egzistenciju.

Zbog različitih koristi (o jednoj se govori i u ovom radu) niz nastavnika ipak pokazuje spremnost da prigrli ovaj AI sustav kao novinu, u vremenskom slijedu niza tehnologija i alata za učenje, svjesni da je njihov zadatak na početku 21. stoljeća pripremiti učenike za svijet prožet umjetnom inteligencijom: "kao edukatori, još nismo otkrili koji je najbolji način upotrebe umjetne inteligencije u školama, no ona dolazi, htjeli mi to ili ne." [3]. Ti su nastavnici oduševljeni mogućnostima ChatGPT-a pri uključivanju u njihov svakodnevni rad, kao i njegovoj sposobnosti da lekcije nastavnih predmeta učini kreativnijima i primjerenijima svakodnevnom životu.

Neki od primjera kako se ChatGPT može koristiti u razredu su: personalizirani planovi učenja, pomoć prilikom učenja pisanja u engleskom jeziku, za generiranje ideja za razredne aktivnosti, kao pomoć u učenju nakon nastave ili kao partner u debati (niz primjera je u [4]). Kako sustav (ipak) ponekad griješi, može pomoći i u vježbama kritičkog mišljenja ili poticanju evaluacije samog sustava od strane korisnika, odnosno učenika koji ga koriste.

Očekivano, nastavnici sada predlažu i potiču da im se omogućiti stjecanje vještina u korištenju ove platforme umjetne inteligencije kako bi na odgovarajući način stimulirali kreativnost učenika i njihove vještine u rješavanju problema: "Imamo mogućnost povesti naše učenike uz nešto što će biti važan dio njihovih života. Nemojmo to blokirati i izbacivati." [3]

Ne manje važno, već i učenici osnovnih škola potvrđuju kako im ovaj nov način rada uz umjetnu inteligenciju učenje čini ugodnijim i zabavnijim, impresionirani su i smatraju ga generatorom novih, dobrih i kreativnih ideja. Uz ispravan pristup, današnji učenici koji će, kako je izgledno, rasti kao prva generacija nativnih svjetskih AI korisnika, prihvatit će da umjetnu inteligenciju koriste za sugestije, ali da proizvode svoj vlastiti rad.

III. OSNOVNO O CHATGPT

U jednom od važnih područja umjetne inteligencije, procesiranju prirodnih jezika (NLP, Natural Language Processing) i njihovom razumijevanju (NLU, Natural Language Understanding), u novije se vrijeme javljaju sve uspješniji proizvodi softverskog inženjerstva, a mogu se opisati kao botovi.

Botovi (bots) su općenito računalni programi sposobni reagirati na vanjske podražaje (događaje, poruke) od strane korisnika na koje odgovaraju pokretanjem automatiziranih postupaka, djelujući kao sučelje između

korisnika i odgovarajućih usluga. Pri tome često uključuju konverzijske sposobnosti, te se krajnjem korisniku obraćaju (komuniciraju) tekstualnim porukama (chatbots) ili govorom (voicebots). Chatbot (ponekad se naziva i virtualnim pomoćnikom) je alat koji ima svrhu pojednostavniti komunikaciju između ljudi i računala. Takvi AI programi su složeni obećavajući oblik interakcije između čovjeka i stroja i mogu oponašati humani razgovor sa korisnicima.

2022. godina biti će obilježena kao godina u kojoj je javnosti obznanjena nova generacija konverzijskih sučelja te vrste: OpenAI ChatGPT i Google LaMDA. Oba sustava koriste generativnu tehniku umjetne inteligencije poznatu kao Large Language Models (LLMs) i predstavljaju značajan skok u napretku konverzijskih sposobnosti softvera. Izgleda kako smo konačno ušli u doba prirodnog računalstva u kojem ljudi mogu voditi smislene i organski tekuće konverzacije sa programskim alatima i aplikacijama.[5]

ChatGPT (Generative Pre-trained Transformer) objavljen krajem studenog 2022. kao dio šireg skupa tehnologija razvijenog od strane OpenAI (startup iz San Franciscas), blisko je povezan sa Microsoft-om. Predstavlja dio nove generacije AI sustava koji je u mogućnosti razgovarati, odnosno na zahtjev generirati čitljiv tekst, pa čak i producirati slike i video temeljem naučenog iz ogromnih baza digitalnih knjiga, tekstova i drugih medija. U mogućnosti je razumjeti kontekst konverzacije i generirati prikladne odgovore u različitim programskim jezicima.

ChatGPT-ov Improved Accuracy (CGA) model ima različita svojstva koja ga čine snažnim NLP sustavom koji koristi AI arhitekturu baziranu na dubokom učenju kako bi producirao točne i smislene konverzacije. Duboko učenje omogućuje mu odlučivanje na temelju podataka, pronalaženjem i izdvajanjem uzoraka podataka iz velikih skupova podataka, koji se precizno preslikavaju u valjane izlazne odluke [6]. Primjenom predtreniranog modela GPT-3, CGA je tako u mogućnosti generirati realistične i poticajne konverzacije usavršene sa mogućnošću da uči iz vlastitih grešaka, dopuštajući mu prilagodbu novim kontekstima kako bi generirao bolje rezultate. Trenutna istraživanja pokazala su kako je CGA dostigao impresivnu razinu točnosti i generativnih sposobnosti. [2]

Dizajniran je da radi kao pisani dijalog (chatbot) između AI sustava i osobe koja postavlja pitanja. Sva pitanja koja dobije pomažu mu i čine ga „pametnijim“ [7] te je tako ChatGPT u mogućnosti učiti i poboljšavati se.

IV. ODABIR NASTAVNOG PREDMETA, CJELINE I TEME

Kao izazov ovom AI sustavu, u strukovnom kurikulumu za stjecanje kvalifikacije tehničar za računalstvo [8] odabran je predmet Algoritmi i programiranje koji se predaje u drugom razredu srednje škole i pripadajuća nastavna cjelina Pokazivači. Kurikulumom ova cjelina sastoji se od nastavnih tema: deklaracija pokazivača, funkcije i pokazivači, polja i pokazivači, te pokazivači na niz znakova.

Izbor je pao na drugu godinu učenja, s obzirom da omogućava širi obuhvat znanja o algoritmima i osnovama

programiranja koje su učenici do tada već usvojili. Vremenski ova tema ulazi u prvo polugodište drugog razreda srednje škole, a u objašnjavnje se može uključiti imenovanje i deklaraciju varijabli, aritmetičke i logičke operacije, naredbe za određivanje tijeka programa i naredbe za ponavljanje, funkcije, polja i rad sa nizovima znakova jer su to učenici trebali savladati već ranije. Uz to, učenici su već trebali upoznati i algoritme za pretraživanje i sortiranje.

Nakon te nastavne cjeline uče se još samo strukture podataka i datoteke, te grafika. Dosta logično se strukture podataka mogu uvoditi i već kroz teme u odabranoj cjelini Pokazivači. Sve su to razlozi zbog kojih je izbor pao upravo na ovu cjelinu.

V. KONVERZACIJA SA SUSTAVOM PRILIKOM PRIPREME MATERIJALA ZA NASTAVNU CJELINU POKAZIVAČI

S ciljem ispitivanja mogućnosti i reakcija ChatGPT sustava za pripremu i razradu nastavne cjeline Pokazivači provedena je konverzacija u nastavku. Odgovori su većinom skraćeni bez utjecaja na njihovu bit.

P1: “Koji bi bio najprimjereniji obuhvat i poredak tema za objašnjavanje pokazivača početnicima u C++?”

O1. Logičan i razumljiv poredak bio bi:

1. Uvod u pokazivače: što su pokazivači, za što se koriste, kako se razlikuju od „običnih“ varijabli.
2. Deklaracija i inicijalizacija pokazivača kako bi pokazivali na specifične memorijske lokacije.
3. Pristup memoriji korištenjem pokazivača uz postupak dereferenciranja kako bi se dohvatilo vrijednost spremljenu na specifičnoj memorijskoj lokaciji.
4. Aritmetika s pokazivačima uz korištenje aritmetičkih operacija kod prolaska članovima polja i drugim podatkovnim strukturama.
5. Dinamička memorijska alokacija upotrebom new i delete i kako pokazivače koristiti za upravljanje dinamičkom memorijom u C++.
6. Pokazivači i funkcije: predaja pokazivača kao argumenta funkciji i vraćanje pokazivača iz funkcija.
7. Pokazivači i polja: kako pokazivači služe za manipulaciju poljima i kako se polja mogu predati funkciji kao pokazivači.
8. Pokazivači na pokazivače: deklaracija i korištenje, posebno kao korištenje kod implementacije višedimenzionalnih polja i drugih naprednih podatkovnih struktura.
9. Uobičajene pogreške prilikom upotrebe pokazivača te kako ih spriječiti.

Komentar: prihvatljivo i podudarno uobičajenom tijeku obrade tih sadržaja u priručnicima. Prijedlog prati sugestivan i koristan zaključak sustava: „ovakav outline osigurava solidno razumijevanje osnovnih koncepata pokazivača u C++ početnicima prije nego što prijeđu na naprednije teme“.

P2: “Koje su razlike između pokazivača i uobičajenih varijabli?”

O2: Radi se o dva različita tipa varijabli u programiranju koji se primarno razlikuju s obzirom na

definiciju, memorijsku alokaciju, pristup podacima, dinamičku memorijsku alokaciju, mogućnost poprivanja null vrijednosti, mogućnost indirektnog pristupa te mogućnost konverzije.

Komentar: sažeto i korektno.

P3: „Koje su prednosti upotrebe pokazivača kod programiranja u C++?“

O3: Pokazivači su moćan alat koji, ako se koristi učinkovito, osigurava različite prednosti: mogućnost dinamičke memorijske alokacije, povećanu učinkovitost zbog izravne manipulacije memorijom, poboljšanu fleksibilnost funkcija, te unaprijeđeno objektno orijentirano programiranje kod programiranja koncepata poput nasljeđivanja i polimorfizma, kao i mogućnost aritmetike s pokazivačima te poboljšanu kontrolu nad memorijskim resursima.

Ipak, nekorektno korištenje pokazivača može voditi ka teškim slučajevima otklanjanja pogrešaka (memory leaks, pogreške segmentacije i druge pogreške kod izvođenja) zbog čega je esencijalno razumjeti osnove te zadržati dobre programerske prakse prilikom njihovog korištenja.

Komentar: vrlo dobra analiza zaključena sa korisnom primjedbom.

P4: “U čiju se pomoć najviše možemo pouzdati prilikom traženja pogrešaka u složenim situacijama upotrebe pokazivača?”

O4: Iskusni programeri, alati za debugiranje (gdb, lldb ili Visual Studio debugger), online forumi i zajednice (npr. Stack Overflow) te profesionalni servisi za debugiranje.

Komentar: Korektno uz opasku kako sustav ipak ne navodi mogućnost da sam pomaže u takvim situacijama.

P5: „Mogu li dobiti nekoliko dobrih primjera deklaracije i inicijalizacije pokazivača.“

O5:

```

css
int *ptr; // declare a pointer to an integer
char *cptr; // declare a pointer to a character
float *fptr; // declare a pointer to a float

cpp
int x = 10;
int *ptr = &x; // initialize a pointer to an integer with the address of x

char ch = 'A';
char *cptr = &ch; // initialize a pointer to a character with the address of ch

float f = 3.14;
float *fptr = &f; // initialize a pointer to a float with the address of f

int *ptr2 = new int[10]; // dynamically allocate an array of 10 integers and init
  
```

Slika 1 Primjeri deklaracije i inicijalizacije pokazivača

Komentar: Sustav ispisu dodaje napomenu kako je “važno da se kod inicijalizacije osigura da pokazivač pokazuje na validnu memorijsku lokaciju”. Također se ističe kako je “prilikom dinamičke alokacije memorije sa new, važno i dealocirati memoriju sa delete”. Korektno, uz opasku kako je u naslovu ispisa krivo navedeno css.

P6: „Postoje li u programskom jeziku C++ pokazivači na pokazivače?“

O6:

```
java
int x = 42;
int *ptr = &x;
int **ptrptr = &ptr; // ptrptr is a pointer to a pointer to an integer

c
cout << **ptrptr << endl; // outputs 42
```

Slika 2 Primjer deklaracije pokazivača na pokazivač

Komentar: Generirani primjer slijedi opis funkcioniranja prilikom indirektnog pristupa varijabli u memoriji. U nastavku se naglašava kako “pokazivači na pokazivače mogu biti korisni npr. u slučaju rada sa dinamičkim višedimenzionalnim poljima, odnosno općenito kod dinamičke manipulacije pokazivačima.” Ovdje se također u naslovu javlja krivi naziv programskog jezika.

P7: U pokušaju povezivanja naučenog o manipulaciji sa nizovima znakova upotrebom klase string i korištenja pokazivača za rješavanje zadataka takvog tipa pomoću pokazivača, slijedi pitanje: “Koji bi bio dobar primjer za usporedbu dva pristupa u programiranju?”

O7:

```
c
#include <iostream>
#include <string>

using namespace std;

int main()
{
    // Approach 1: using string class
    string str1 = "Hello";
    string str2 = " World";
    string str3 = str1 + str2;
    cout << str3 << endl;

    // Approach 2: using character pointers
    char cstr1[] = "Hello";
    char cstr2[] = " World";
    char cstr3[11];
    int i;
    for (i = 0; cstr1[i]; i++) {
        cstr3[i] = cstr1[i];
    }
    for (int j = 0; cstr2[j]; j++, i++) {
        cstr3[i] = cstr2[j];
    }
    cstr3[i] = '\0';
    cout << cstr3 << endl;

    return 0;
}
```

Slika 3 Primjer korištenja pokazivača kod spajanja nizova

Navedeni programski kod prati i iscrpno objašnjenje te usporedba s obzirom na očekivano vrijeme izvođenja ovog programskog koda na velikim (dugim) stringovima koje je dulje prilikom korištenja klase string.

Komentar: Točnost prezentiranog koda provjerena je korištenjem programskog okruženja Microsoft Visual Studio Community 2022 (64-bit), pa s obzirom da je kod

drugog pristupa bila potrebna korekcija programskog koda (radi određenih upozorenja i/ili pogrešaka vezanih uz ograničenja korištenih funkcija), od sustava je zatraženo novo rješenje što je rezultiralo slijedećim prijedlogom:

```
c
#include <iostream>
#include <cstring>

int main()
{
    char str1[] = "Hello";
    char str2[] = " World";
    int length = strlen(str1) + strlen(str2);
    char *str3 = new char[length + 1];

    strcpy(str3, str1);
    strcat(str3, str2);

    std::cout << str3 << std::endl;

    delete[] str3;
    return 0;
}
```

```
c
#include <iostream>

int main()
{
    char str1[] = "Hello";
    char str2[] = " World";
    int length1 = 0;
    int length2 = 0;
    while (str1[length1] != '\0')
        length1++;
    while (str2[length2] != '\0')
        length2++;
    int length = length1 + length2;
    char *str3 = new char[length + 1];

    for (int i = 0; i < length1; i++)
        str3[i] = str1[i];
    for (int i = 0; i < length2; i++)
        str3[length1 + i] = str2[i];
    str3[length] = '\0';

    std::cout << str3 << std::endl;

    delete[] str3;
    return 0;
}
```

Slika 4 Dodatni primjeri rješenja za problem spajanja nizova

Komentar: Kako je drugi prijedlog nakon provjere generirao pogreške u prevodenju, na zahtjev za novim, trećim prijedlogom, sustav ispisuje treći primjer programskog koda koji kao ispis daje očekivani tekst. Oduševljava što su svi primjeri popraćeni korisnim detaljnim tumačenjem prezentiranog koda, a impresionira i mogućnost sustava da s lakoćom prikazuje alternativna rješenja za zadani problem. Pokazuje se poželjnim da u novijim inačicama sustav bude u mogućnosti predvidjeti ili upozoriti i na ograničenja ili specifičnosti korištenja primijenjenih funkcija za rad sa nizovima znakova.

P8: „Koji bi bili primjeri ili slučajevi kada, u postupcima manipulacije nizovima znakova, pristup upotrebom pokazivača zahtjeva značajno manje vremena kod izvođenja i ima prednosti u performansama nego odgovarajući pristup upotrebom klase string?“

O8: Neki primjeri su:

- Vrlo dugi stringovi zbog toga što klasa string svoje podatke sprema dinamički na heap

memoriju tražeći dodatne operacije memorijske alokacije i dealokacije što usporava postupak.

- Performance-critical code kad se manipulacija stringovima izvodi učestalo (npr. high-performance game engine).
- Operacije spajanja stringova kod spajanja mnogo manjih stringova u jedan veliki i to zbog višestrukih memorijskih alokacija i dealokacija potrebnih kako bi se promijenila veličina interne kratkoročne memorije prilikom rasta stringa u slučaju rada sa string klasom. Upotrebom pokazivača može se prealocirati velika kratkoročna memorija i spajati stringove izravno u toj memoriji.
- Manipulacija C-style stringovima za sučelja sa C programskim kodom ili bibliotekama.

Komentar: Prilikom ovog pitanja sustav također generira obuhvatan i korektan odgovor.

P9: „Koji bi bio primjer za programski kod funkcije koja će bez i uz pomoć pokazivača sortirati riječi u rečenici prema njihovoj duljini?“

O9:

```
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>

using namespace std;

bool compareLength(const string &a, const string &b)
{
    return a.length() < b.length();
}

void sortWordsByLength(const string &sentence)
{
    vector<string> words;
    string word;
    for (char c : sentence) {
        if (c == ' ') {
            words.push_back(word);
            word.clear();
        }
        else {
            word += c;
        }
    }
    words.push_back(word);

    sort(words.begin(), words.end(), compareLength);

    for (const string &word : words) {
        cout << word << " ";
    }
    cout << endl;
}

int main()
{
    string sentence = "This is a sentence that we want to sort";
    sortWordsByLength(sentence);

    return 0;
}
```

Slika 5 Primjer programskog koda sa sortiranjem bez pomoći pokazivača

Komentar: Korektni primjeri koda, sa i bez upotrebe pokazivača, detaljnije opisani sa objašnjenjima korištenih funkcija i čitavog programa i izvode se na očekivani način.

```
c
#include <iostream>
#include <cstring>
#include <algorithm>

using namespace std;

bool compareLength(const char* a, const char* b)
{
    return strlen(a) < strlen(b);
}

void sortWordsByLength(char* sentence)
{
    int n = 0;
    char* words[100];
    char* p = strtok(sentence, " ");
    while (p) {
        words[n++] = p;
        p = strtok(nullptr, " ");
    }

    sort(words, words + n, compareLength);

    for (int i = 0; i < n; i++) {
        cout << words[i] << " ";
    }
    cout << endl;
}

int main()
{
    char sentence[] = "This is a sentence that we want to sort";
    sortWordsByLength(sentence);

    return 0;
}
```

Slika 6. Primjer programskog koda sa sortiranjem uz pomoć pokazivača

P10: „Koji bi bio primjer korištenja pokazivača kod programiranja rekurzije?“

O10: Uz ispis primjera, sustav odgovara da su pokazivači vrlo korisni kod rekurzije:

```
c
#include <iostream>

void printList(int* arr, int size) {
    if (size == 0) return;
    std::cout << *arr << " ";
    printList(arr + 1, size - 1);
}

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int size = sizeof(arr) / sizeof(arr[0]);
    printList(arr, size);
    return 0;
}
```

Slika 7. Primjer upotreba pokazivača kod rekurzivne funkcije

Komentar: Nakon iznošenja primjera, u objašnjenje je uključen poučan savjet kako “korištenjem pokazivača umjesto varijable koja bi predstavljala trenutnu poziciju u polju možemo spriječiti overhead kopiranja čitavog polja tijekom svakog rekurzivnog poziva. To može biti vrlo korisno kod rada sa vrlo velikim poljima, jer reducira upotrebu memorije te popravlja performanse programa.“

P11: “Koji bi bili primjeri uobičajenih pogrešaka koje se javljaju prilikom korištenja pokazivača i kako ih izbjeći?”

O11:

- Dereferenciranje null pokazivača: pojavljuje se prilikom pokušaja pristupa vrijednosti spremljenoj u pokazivač koji nije bio inicijaliziran ili je bio postavljen u null.
- Pristup memoriji izvan granica polja
- Memory leak: pojavljuje se prilikom dinamičnog alociranja memorije pomoću new ili malloc, ali je propušteno dealocirati ili osloboditi memoriju.
- Korištenje/pristupanje memoriji nakon oslobađanja ili dealociranja sa delete ili free.
- Dvostruko oslobađanje: dealokacija već dealocirane memorije

VI. KONVERZACIJA PRILIKOM OBLIKOVANJA PREZENTACIJE ZA NASTAVNU CJELINU POKAZIVAČI

P12: „Molim za prijedlog i razradu sadržaja za prezentaciju koja bi se koristila za poučavanje i objašnjavanje pokazivača početnicima.“

O12:

- I. Uvod (objašnjenje pokazivača, važnost pokazivača u C++, zbog čega je važno učiti o pokazivačima, preduvjeti za učenje pokazivača)
- II. Memorija (objašnjenje memorije u C++, alokacija i upravljanje memorijom, stack i heap memorija)
- III. Pokazivači i adrese (Što je pokazivač? Kako radi pokazivač? Sintaksa i deklaracija pokazivača, aritmetika pokazivača, adresni operator (&), operator dereferenciranja (*))
- IV. Pokazivači i funkcije (Korištenje pokazivača kao funkcijskih parametara, predaja pokazivača preko adrese – by reference, vraćanje pokazivača iz funkcija, Dangling pointers)
- V. Dinamička memorijska alokacija (Objašnjenje, operatori new i delete, memory leaks, pametni pokazivači)
- VI. Uobičajene upotrebe pokazivača (polja i pokazivači, strukture i pokazivači, pokazivači i klase, pokazivači i stringovi)
- VII. Savjeti i dobre prakse (Izbjegavanje uobičajenih pogrešaka kod upotrebe pokazivača, traženje pogrešaka u stvarima vezanim uz pokazivače, najbolje prakse upravljanja memorijom)
- VIII. Zaključak (Sažetak ključnih koncepata, Izvori za daljnja učenja, rekapitulacija važnosti pokazivača)

P14: „Gdje se mogu naći dobre ilustracije i animacije koje mogu pomoći u tumačenju ovih koncepata učenicima?“

O14: Mnogo je raspoloživih online resursa koji mogu u tome pomoći, npr.:

YouTube – neki popularni kanali su "The Cherno" "CodeVault," i "Programming with Mosh."

Online Learning Platforms – Udemy, Coursera i edX Codecademy - interaktivna platforma

Priručnici – npr. na Amazon ili u lokalnoj knjižnici

Online Forumi i zajednice - Stack Overflow i Reddit.

Komentar: sustav „ne zaboravlja“ u odgovoru naglasiti kako je „važno da su vizualni jasni i jednostavni za razumijevanje, te ih je važno detaljno objasniti kako bi učenici razumjeli prezentirane koncepte.“

VII. ZAKLJUČAK

Unatoč određenoj sumnjičavosti pri pristupu ovoj konverzaciji, primjenjeni AI sustav demonstrirao je brze odgovore, konkretna i upotrebljiva znanja te sposobnost sintetiziranja znanja kao i konciznost u odgovorima. Na sva pitanja ili komentare AI odgovarao je prihvatljivo i s prividnim „razumijevanjem“. Programski odsječci su korektni, raznoliki i dovoljno detaljno objašnjeni.

Iskustvo izloženo kroz primjere u ovom radu navodi na stav kako bi se ChatGPT mogao koristiti kao pomagalo u pripremi materijala za poučavanje, u objašnjavanju nastavnih sadržaja i pripremi ispitnih pitanja.

Rezultati i dosadašnje iskustvo potvrđuju mišljenje koje u knjizi [1] istražuje mogućnosti i rizike upotrebe AI u obrazovanju te utvrđuje kako bi škole trebale prigrliti tu tehnologiju i koristiti ju za poboljšanje zadataka i poticanje originalnosti u idejama i razmišljanju učenika. "U mogućnosti smo razmisliti što mi u obrazovanju zapravo radimo. Zbog čega imamo zadatke koje AI s takvom lakoćom replicira?". Zadatci bi trebali biti smisleniji i ispitivati vještine kritičkog razmišljanja učenika, razvijati umove sposobne nalaziti rješenja.

Zato potencijal ovog softvera, koji je od svoje pojave krajem studenog 2022. postao globalni fenomen, ne bi trebalo ni zanemarivati ni negirati. U području obrazovanja njegove uloge pomoćnika u pripremi i izvođenju nastave (kao obrazovnog alata) mogu biti vrlo različite i trebalo bi osigurati da nadjačaju rizike njegovog korištenja. Određeni oblici „suradnje“ sa ovim alatom kao učinkovitim pomoćnikom omogućuju put ka uspješnijem učenju i dubljem razumijevanju predmetnog gradiva. Tako bi se ovaj alat u poučavanju mogao prihvatiti i kao pomoć oslobađanju učeničke kreativnosti nuđenjem personaliziranog mentorskog nadzora, ali i radi pripreme učenika za rad sa AI sustavima i njihovim modelima u budućnosti. Primjenivši ga na način da djeluje kao pomoćnik, ChatGPT može učiniti nastavu kvalitetnijom.

REFERENCE

- [1] S. Popenici: "Artificial Intelligence and Learning Futures, Critical Narratives of Technology and Imagination in Higher Education", Routledge, New York, 2023.
- [2] Jianyang D, Yijia L.: "The Benefits and Challenges of ChatGPT: An Overview Assessment in vocational upper secondary education", Frontiers in Computing and Intelligent Systems, Vol. 2, 2022.
- [3] J. Gecker: "It's coming, whether we want it or not": Teachers nationwide are using ChatGPT to prepare kids for an AI world", sa <https://fortune.com/2023/02/14/chatgpt-school-lessons-cheating-robots-ai-teachers/> (posjećeno veljača 2023.)
- [4] "ChatGPT, Chatbots and Artificial Intelligence in Education", sa <https://ditchthattextbook.com/ai> (posjećeno prosinac 2022.)
- [5] "We gave ChatGPT a college-level microbiology quiz. It aced it.", sa <https://bigthink.com/the-future/chatgpt-microbiology-quiz-aced/> (posjećeno, veljača 2023.)
- [6] J. D. Kelleher: "Duboko učenje", Mate, Zagreb, 2021.
- [7] M. O'Brien: "Explainer: What is ChatGPT and why are schools blocking it?", sa <https://apnews.com/article/what-is-chat-gpt-ac4967a4fb41fda31c4d27f015e32660> (posjećeno veljača 2023.)
- [8] Strukovni kurikulumu za stjecanje kvalifikacije tehničar za računalstvo sa https://www.asoo.hr/UserDocsImages/Kurikulumi/K_Tehnicar%20za%20racunalstvo_zaz%20odobrenje.pdf, posjećeno veljača 2023.