

Usporedba autorskih alata za izradu online tečaja za učenje programiranja

Zirdum, Nikolina

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:137:210404>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-11-19**



Repository / Repozitorij:

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli

Fakultet informatike

NIKOLINA ZIRDUM

**USPOREDBA AUTORSKIH ALATA ZA IZRADU ONLINE TEČAJA ZA
UČENJE PROGRAMIRANJA**

Diplomski rad

Pula, rujan, 2021. godine

Sveučilište Jurja Dobrile u Puli

Fakultet informatike

NIKOLINA ZIRDUM

**USPOREDBA AUTORSKIH ALATA ZA IZRADU ONLINE TEČAJA ZA
UČENJE PROGRAMIRANJA**

Diplomski rad

JMBAG: 0016111366, redoviti student

Studijski smjer: Sveučilišni diplomski studij informatika

Predmet: Sustavi elektroničkog učenja

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informacijske i komunikacijske znanosti

Znanstvena grana: Informacijski sustavi i informatologija

Mentor: doc. dr. sc. Snježana Babić

Pula, rujan, 2021. godine



IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisana *Nikolina Zirdum*, kandidat za magistra *informatike* ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mogega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

U Puli, rujan, 2021. godine



IZJAVA

o korištenju autorskog djela

Ja, *Nikolina Zirdum* dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom „*Usporedba autorskih alata za izradu online tečaja za učenje programiranja*“ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, rujan, 2021. godine

Potpis

SADRŽAJ

1. UVOD	1
2. OPĆENITO O PROGRAMIRANJU	3
2.1. Osnovni pojam programiranja	5
2.2. Proces programiranja	6
2.3. Pojam i vrste programskih jezika	9
2.3.1. Programski jezici namijenjeni učenju	13
3. ONLINE UČENJE	15
3.1. Razvoj i vrste online učenja	16
3.2. Prednosti i nedostaci online učenja	19
4. ONLINE UČENJE I PROGRAMIRANJE	21
4.1. Vrste tehnologija i alata za online učenje programiranja	22
4.2. Sustavi za upravljanje učenjem - LMS	27
4.2.1. CANVAS	32
4.2.2. Moodle	33
4.2.3. iSpring Suite	34
4.2.4. TalentLMS	35
5. PRIMJER UPOTREBE AUTORSKIH ALATA U SVRHU ONLINE PROCJENE ZNANJA IZ PROGRAMIRANJA	37
5.1. Izazovi izrade online tečaja za programiranje	38
5.2. Struktura online testa: Osnove programiranja u Pythonu	39
5.3. Kreiranje online testa u sustavu CANVAS.....	41
5.3.1. Provedba online testa	43
5.4. Kreiranje online testa u sustavu Moodle	48
5.4.1. Provedba online testa	50

5.5. Kreiranje online testa u sustavu iSpring Suite.....	57
5.5.1. Provedba online testa	59
5.6. Kreiranje online testa u sustavu TalentLMS.....	64
5.6.1. Provedba online testa	67
5.7. Usporedba autorskih alata.....	72
6. ZAKLJUČAK	74
LITERATURA	75
POPIS SLIKA	79
POPIS TABLICA.....	81
POPIS GRAFIKONA.....	81
SAŽETAK	82
ABSTRACT	83

1. UVOD

Jednostavnost i ubrzanost obavljanja poslova, učinkovitost, mobilnost, povezanost, samo su neke od prednosti korištenja tehnologije i interneta. Današnji ubrzani način života i svakodnevne obveze znatno su olakšane naglim tehnološkim razvojem. Zahvaljujući tom razvoju informacijsko komunikacijska tehnologija primjenjuje se u svim gospodarskim granama, a programer je gotovo uvijek pri vrhu najtraženijih i najbolje plaćenih zanimanja u svijetu. Naučiti programirati zahtjevan je proces koji od pojedinca iziskuje dosta učenja, truda i volje, a steći zvanje programera nije ni malo lako.

Zahvaljujući tehnologiji znanje nikada nije bilo dostupnije. Povezivanjem na internet pruža se neograničen broj mogućnosti za učenje iz bilo kojeg područja vlastitog interesa. Samim time online učenje postaje sve zastupljenije, a uz sve prednosti koje pruža postaje sve prihvaćeniji način učenja kod ljudi. Sustavi za upravljanje učenjem, razni autorski alati i tehnologije omogućavaju učenje programiranja iz udobnosti vlastitog doma. Danas postoje razni načini kako putem interneta naučiti programirati, a najpoznatiji od njih je pohađanje online tečajeva.

Izrada takvih tečajeva iznimno je zahtjevna i pred autore stavlja dosta izazova. Unapređenjem sustava za upravljanje učenjem i razvojem raznih autorskih alata, taj proces je sve jednostavniji. Cilj ovog diplomskog rada je istražiti mogućnosti različitih sustava za upravljanje učenjem kada je u pitanju jedan od najvećih izazova izrade online tečaja za učenje programiranja, a to je pisanje izvornog koda. Ovaj problem se javlja kod procjene znanja polaznika. Sustavi za upravljanje učenjem nude razne vrste pitanja koja se mogu kreirati, ali kada se radi o programiranju, autori tečaja zasigurno žele imati mogućnost da polazniku zadaju zadatak koji od njega zahtijeva da samostalno napiše izvorni kod koji će sustav odmah provjeriti i vratiti povratnu informaciju o točnosti danog odgovora. Iznimno je malo sustava koji imaju tu mogućnost. Prva pomisao je da takvi sustavi zasigurno nisu besplatni i javno dostupni, no je li to zaista tako? Ova hipoteza bit će potvrđena ili odbačena ovisno o rezultatima usporedbe na kraju rada.

Nakon uvoda, u prvom dijelu rada obrađen je pojam programiranja, procesa programiranja i programskih jezika. Drugi dio rada odnosi se na online učenje, razvoj i vrste online učenja, uz prikaz svih prednosti i nedostataka koji se pojavljuju. Spoj ova dva područja događa se u trećem dijelu rada gdje se govori o alatima i tehnologijama za online učenja programiranja te se govori o sustavima za upravljanje učenjem. Četvrti, ujedno i glavni dio rada, odnosi se na pisanje i provjeru izvornog koda - najveći izazov izrade online tečaja za učenje programiranja. U ovom dijelu prikazana je struktura probnog testa koji će biti proveden kroz svaki odabrani sustav za upravljanje učenjem na način koji ovisi o njegovim mogućnostima. Nakon toga, slijedi dio u kojem se prikazuje usporedba rezultata nakon čega slijedi zaključak ovog diplomskog rada.

S obzirom na to da ne postoji objavljeni rad koji se bavi izazovima izrade online tečaja za učenje programiranja, ovo je glavna motivacija za odabir i istraživanje ovog područja. Kako provesti procjenu znanja iz područja programiranja ovisno o različitim mogućnostima sustava, svrha je ovog rada.

2. OPĆENITO O PROGRAMIRANJU

Kako bi se moglo govoriti o programiranju, za početak ovog poglavlja potrebno je reći nešto i o samoj znanosti kojoj programiranje pripada. Informacijske znanosti jedno su od mnogih polja društvenih znanosti. U svom radu *Lipljin (1993: 67-82)* tvrdi kako se radi o relativno mladom i vrlo složenom znanstvenom području. Zašto složenom? Upravo iz razloga jer ovo područje uključuje tehničke i humanističke komponente, ali i dijelove drugih znanosti kao što su komunikologija, računarstvo, matematika, logika, inženjering i druga srodna područja. Stoga može se reći kako se radi o interdisciplinarnoj znanosti koja se bavi stvaranjem, prikupljanjem, organizacijom, pohranjivanjem, pretraživanjem, širenjem i uporabom informacija uz pomoć informacijsko komunikacijske tehnologije. Pojam „informatika“ obuhvaća područje u cjelini i predstavlja sinonim za pojam informacijske znanosti.

Bez razvoja informatike i računarstva ne bi bilo programiranja. Veliki doprinos tome imala je sve veća potreba za radom s podacima koja je probudila želju za stvaranjem raznih uređaja koji bi omogućili bržu i kvalitetniju obradu podataka. Zasiurno najstarija izrađena naprava za računanje je abak (*lat. abacus*) nastao u Kini 3000. godine pr.n.e. Na tom području nije se dogodilo ništa značajno sve do 1623. godine kada je Wilhelm Schickard izumio prvi mehanički kalkulator koji je imao mogućnost zbrajanja, oduzimanja, množenja i dijeljenja. Ipak, početkom računarstva smatra se izum francuskog matematičara Blaisea Pascala koji je 1642. godine napravio kalkulator tzv. Pascalina (Slika 1). Imala je mogućnost zbrajanja i oduzimanja velikih brojeva, ali nije bila uvijek točna. Gottfried Wilhelm Leibniz 1672. godine izradio je uređaj sličan Pascalini koji je mogao još množiti, dijeliti i računati drugi korijen. Nije se baš koristio, ali ovaj njemački matematičar utemeljio je binarni sustav koji je kostur svakog računala danas (*Kralj, 2018: 3-7*).



Slika 1 Pascalina

(Izvor: <http://w-hasselo.nl/mechn/history/Pascaline-jpg.jpg>, 26.06.2021.)

Prema *Hrvatskoj enciklopediji (2021)* programiranje je kreiranje naredbi čijim izvršavanjem dolazi do djelovanja nekog automata¹, u ovom slučaju to je računalo. Laički rečeno računalo bez programiranja je samo stroj. Pojava prvih elektronskih računala koji su baza današnjih računala započinje početkom Drugog svjetskog rata. Prvo takvo računalo ENIAC razvijeno je 1944. godine i na takvim računalima razvijeni su prvi programski jezici za programiranje računalnih programa.

Razvojem tehnologije i raznih uređaja računarstvo je iznimno napredovalo kroz vrijeme. Prije svega, zahvaljujući mnogim znanstvenicima i njihovim otkrićima kroz povijest, danas je računarstvo zasebna znanost koja uči programere umijeću programiranja. Što je programiranje i osnovni pojmovi koji su bitni za ovo područje, slijede u nastavku rada.

¹ Prema *Hrvatskoj enciklopediji* Leksikografskog zavoda Miroslav Krleža, mrežno izdanje: „*automat* je uređaj koji potpuno samostalno obavlja koristan rad prema čovjekovoj zamisli.”

2.1. Osnovni pojam programiranja

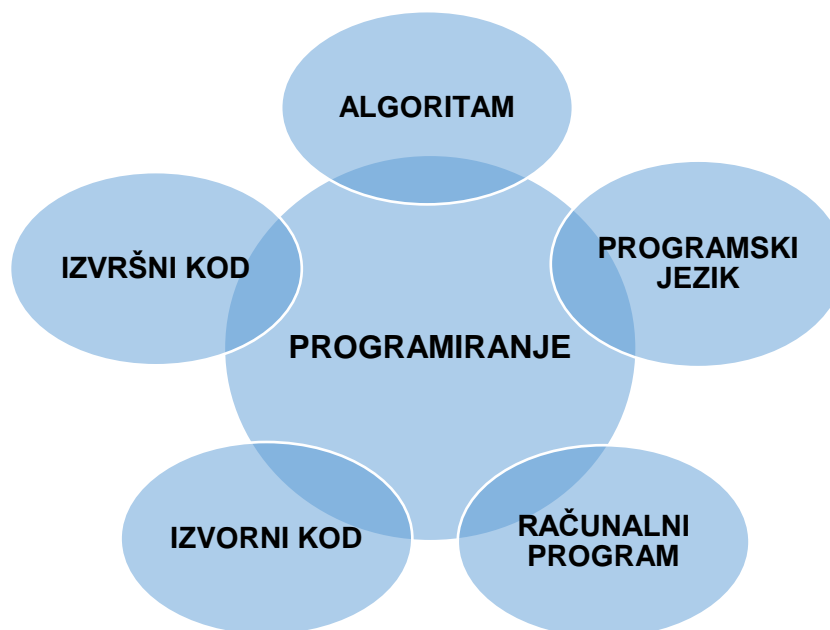
Mnogi misle kako su se temelji programiranja koji su poznati danas razvili u kasnim godinama dvadesetoga stoljeća, a je li to doista tako?

Ada Lovelace je žena koju svijet smatra prvom programerkom u povijesti jer je 1842. godine opisala prvi algoritam koji je bio prilagođen za implementaciju na računalu. Radilo se o algoritmu za analitički stroj za izračunavanje Bernoullijevih brojeva² što je izuzetno teško izračunati, a s obzirom na to da stroj nikada nije dovršen, njezin program ostao je netestiran. Nakon izuma Turingovog testa, zbog kojeg ga se smatra ocem modernog računarstva, britanski matematičar Alan Turing koristio je bilješke Ade Lovelace u radu na prvim modernim računalima. Napisao je jedan od najranijih operacijskih sustava za prva računala tada (*Parker, 2014: 20, 221-222*). Poznata Booleova algebra koja se koristi i danas nastala je čak 1854. godine, a naziv je dobila po tvorcu Georgeu Booleu (*Hrvatska enciklopedija, 2021*). Ovo su samo neki od značajnijih događaja u programerskoj povijesti koji dokazuju da su temelji ove znanosti postavljeni dosta ranije nego što se misli. Tehnološki napredak i otkriće interneta uvelike su pridonijeli naglom razvoju programiranja.

Što je zapravo programiranje? *Panian (1999: 82)* objašnjava kako je programiranje fazni postupak kojim se u računalu ugrađuje ljudsko metodološko znanje, kako bi računalno moglo izvršavati zadatke koje će mu zadati čovjek putem programa. S druge strane, *Jakupović i Šuman (2014: 44)* programiranje definiraju kao proces oblikovanja, pisanja, testiranja pronalaženja i uklanjanja greški te održavanja izvornog koda računalnog programa. *Kingsley-Hughes (2005: 4)* autori su koji kažu da je programiranje sposobnost razgovora s računalom na jeziku koji računalno razumije uz korištenje gramatike i sintakse koju može pratiti kako bi bilo sposobno izvršiti zadatke koje mu čovjek zadaje.

Na ovo pitanje ne postoji jedinstven odgovor, jer postoji mnogo definicija programiranja. Ni jedna od njih nije kriva, a svaka u sebi sadržava ključne pojmove koji se vežu uz programiranje (Slika 2).

² Prema Hrvatskoj enciklopediji Leksikografskog zavoda Miroslav Krleža, mrežno izdanje: „Bernoullijevi brojevi su niz racionalnih brojeva.“



Slika 2 Osnovni pojmovi programiranja

(Izrada autorice rada)

Računalo razumije samo binarni brojevni sustav, odnosno za njega postoje samo brojevi 0 i 1. Kako onda računalo obavlja sve te zadatke? Za početak postoji algoritam, to je opis rješenja nekog problema. Kako bi se računalu opisao algoritam koristi se programski jezik. To je jezik koji je namijenjen za opis algoritma, ljudima je razumljiv, ali računalu ne. Opisivanjem algoritma dobiva se niz instrukcija koje tvore računalni program. Računalni program sadrži izvorni kod koji je napisao programer i njega računalo ne razumije. Tu veliku ulogu imaju jezični prevoditelji (interpreteri i kompajleri) koji imaju zadatak izvorni kod pretvoriti u izvršni (strojni) kod kako bi računalo razumjelo program i instrukcije koje su mu zadane. Takav strojni kod sastoji se od niza nula i jedinica i ljudima nije čitljiv (*Jakupović i Šuman, 2014: 43-44*).

2.2. Proces programiranja

Nakon definiranja samog pojma programiranja i osnovnih pojmova koji se vežu uz programiranje, potrebno je ukratko predložiti koliko je proces programiranja zapravo zahtjevan.



Slika 3 Proces programiranja

(Izrada autorice rada)

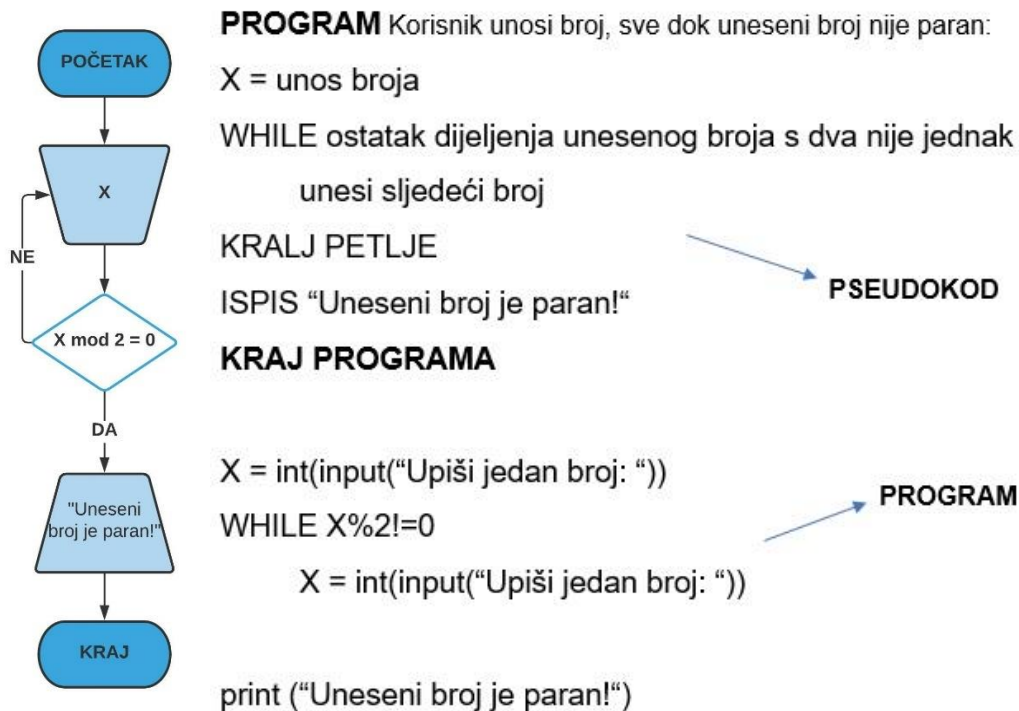
Prethodna slika ugrubo prikazuje proces programiranja. Za početak iznimno je važno definirati problem i ono što se želi postići izradom programa. Nakon toga se odabire način rješavanja problema i time započinje proces stvaranja algoritma. I za kraj, ono najbitnije i najzahtjevnije, nakon odabira programskog jezika pomoću kojeg će se opisati algoritam, stvara se računalni program.

U svojoj knjizi *Panian (1999: 82)* navodi sljedećih osam faza programiranja:

- definiranje problema,
- stvaranje algoritma,
- dokumentiranje algoritma,
- pisanje programa,
- testiranje programa,
- dokumentiranje programa,
- implementacija programa i
- korištenje i održavanje programa.

U prvoj fazi programer definira željeni rezultat programiranja s obzirom na problem, a u fazi stvaranja algoritma pronalazi način za njegovo rješavanje. Za prikaz rješenja problema koristi dijagram toka i/ili pseudokod. Nakon što je stvorio konačni algoritam, sam proces programiranja može započeti. Pisanje programa podrazumijeva stvaranje niza naredbi čijim završetkom dobivamo računalni program (Slika 4). Prije same implementacije programa isti je potrebno testirati. Ako se utvrdi da program nije ispravan nužno je provesti provjeru algoritma i koda i ispraviti pogreške. Taj se postupak ponavlja sve dok se ne dobije potpuno ispravan program koji se zatim može dokumentirati.

Pod dokumentacijom se misli na izradu tehničkog opisa programa, a ako je potrebno izrađuje se i priručnik za instalaciju i korisnike. Nakon toga, program se može implementirati, ali tu proces programiranja ne završava. Zadnja faza odnosi se na korištenje programa i njegovo održavanje. Korisnici ga koriste, a programeri su dužni program održavati, nadograđivati i mijenjati ovisno o potrebama korisnika i životnom vijeku programa.



Slika 4 Primjer dijagrama toka (lijevo) i pseudokoda i programa (desno)

(Izrada autorice rada)

Vidi se da tu postoji još bitnih koraka izuzev samog pisanja računalnog programa. Ovako kada se govori o procesu programiranja to uopće ne izgleda komplicirano, ali svatko tko se ikada susreo s programiranjem zna o koliko zahtjevnom procesu je riječ. Iz tog razloga, programer je gotovo uvijek traženo zanimanje, ali naučiti programirati nije jednostavno.

2.3. Pojam i vrste programskih jezika

Kako u svijetu postoji mnogo jezika za komunikaciju, tako i u programerskom svijetu postoji mnogo jezika pomoću kojih programeri komuniciraju s računalima. Kako je već prije rečeno, računalo razumije samo nizove nula i jedinica i pisanje instrukcija na taj način bilo bi iznimno teško za ljude. Zato postoji programski jezik koji *Jakupović i Šuman (2014: 43-44)* definiraju kao jezik koji je namijenjen za davanje instrukcija računalu. Svaki programski jezik sastoji se od sintakse (forme) i semantike (značenja) koju programer mora savladati ako želi naučiti programirati u određenom programskom jeziku. To je kao kada se uči neki strani jezik, prije svega potrebno je znati njegovu gramatiku, ali i značenje pojedinih riječi. Dobro poznata latinska poslovice: „*Koliko jezika znaš, toliko ljudi vrijediš*“, može se odraziti i na svijet programiranja, jer danas programeri koji vladaju s više programskih jezika iznimno su traženi i cijenjeni u poslovnom svijetu.

Jakupović i Šuman (2014: 46-47) kažu kako postoji pet generacija programskih jezika. Kada se govori o strojnim jezicima misli se na **programske jezike prve generacije**. Oni su bili iznimno komplicirani i od programera su zahtijevali veliku količinu znanja i mukotrpan rad jer pisanje programa se vršilo direktno u strojnome (binarnom) kodu. To znači da su naredbe ustvari bile nizovi nula i jedinica, a računalo ih je odmah izvodilo bez prevođenja jer strojni jezik računalo razumije. Ovaj način programiranja nije bio pretjerano djelotvoran, a sve većim razvojem ovog područja dolazi do pojave **programskih jezika druge generacije**. Pisanje naredbi postaje lakše jer se umjesto nizova nula i jedinica koristi prirodni ljudski jezik. S obzirom na to da je razvoj ove generacije jezika vođen u SAD, osnova nove generacije programskih jezika je engleski jezik. Ovaj način programiranja bio je znatno lakši jer čovjek bolje pamti i razumije kratke riječi (npr. save, move, load, add...), nego binarni niz (npr. 10001011). Time dolazi do pojave prvih programa prevoditelja, tzv. assembleri (*engl. assembler*). Iako se radi o jeziku niže razine, mnogo je napredniji u odnosu na strojni jezik. Problem pisanja složenijih programa i programa koji se mogu izvršavati na različitim vrstama procesora rješavaju **programski jezici treće generacije**. Za razliku od strojnog jezika i assemblera, programski jezici treće generacije su jezici više razine jer pružaju veliku količinu apstrakcije, slični su ljudskom jeziku, samim time su čitljiviji, učinkovitiji i lakše pamtljivi.

Primjeri jezika ove generacije su Fortran, Cobol, Visual Basic, Python, Java, C, C++, C#, Pascal, PHP i mnogi drugi. **Programski jezici četvrte generacije** su jezici koji su usmjereni na obradu podataka i rukovanje bazama podataka (npr. SQL). Zadnja, **peta generacija programskih jezika**, osmišljena je kako bi omogućila računalima da sami rješavaju probleme. Tu spadaju programski jezici namijenjeni programiranju umjetne inteligencije (npr. Prolog, Mercury i OPS5).

Prema prethodno prikazanom razvoju, programski jezici dijele se na:

- **programske jezike niže razine** - orijentirani prema računalu i
- **programske jezike više razine** - orijentirani prema problemu.

Danas kada se govori o programskim jezicima uglavnom se misli na programske jezike više razine, a za programere to znači manja potrebna količina znanja o građi računala. Međutim, od toliko programskih jezika koji postoje danas na tržištu, koji jezik odabrati? Mnogi programski jezici dizajnirani su da budu jezici opće namjene. Premda, ovisno o tome što osoba želi programirati i stvarati, postoje određeni programski jezici za svaku od namjena. Upravo zbog toga razlikuje se više vrsta programera (Tablica 1).

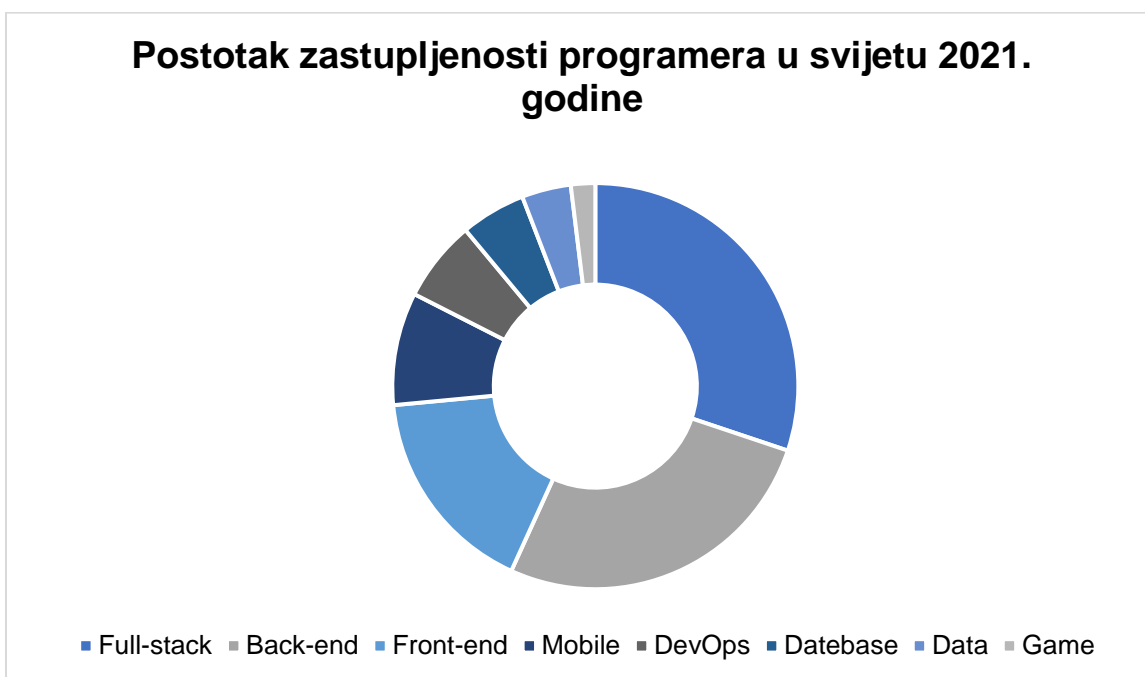
Tablica 1 Vrste programera i programski jezici

PROGRAMIRANJE	OPIS	PROGRAMSKI JEZICI
Front-end developer	Izgled	HTML/CSS, JavaScript, React, Vue, TypeScript
Back-end developer	Funkcionalnost	PHP, Java, C, C++, Python, Go, Ruby
Full-stack developer	Front-end + Back-end	HTML/CSS, JavaScript, PHP, Ruby, Python
Mobile developer	Mobilne aplikacije	HTML/CSS, JavaScript, Java, C, C++, C#, Python, Swift
Game developer	Video igre	JavaScript, HTML, C, C++, Java, Python
Database developer	Baze podataka	SQL, C, C++, C#, Java
Data developer	Analiza podataka	R, Python, SQL
DevOps developer	Razvojni proces	Python, C, C++, Scala, Ruby, Go

(Izrada autorice rada prema: <https://syndicode.com/blog/types-of-developers/>, 30.06.2021.)

U prethodnoj tablici izdvojene su najpoznatije vrste programera i programski jezici s obzirom na područje njihova stvaranja. Određeni programski jezici karakteristični su za pojedino područje, kao recimo SQL koji koriste programeri baza podataka, programeri strojnog učenja i statističke analize podataka. Dok je većina jezika opće namjene, odnosno može se koristiti za više područja, kao primjerice Java, Python, C++, što je vidljivo iz tablice.

Prema istraživanju iz svibnja 2021. godine koje je provela vodeća svjetska mreža za programere *Stack Overflow*, 66.484 programera iz cijelog svijeta dalo je svoj odgovor u kojem području programiranja djeluju. Sljedeći grafikon prikazuje prethodno navedenih osam područja i njihov postotak zastupljenosti u svijetu.

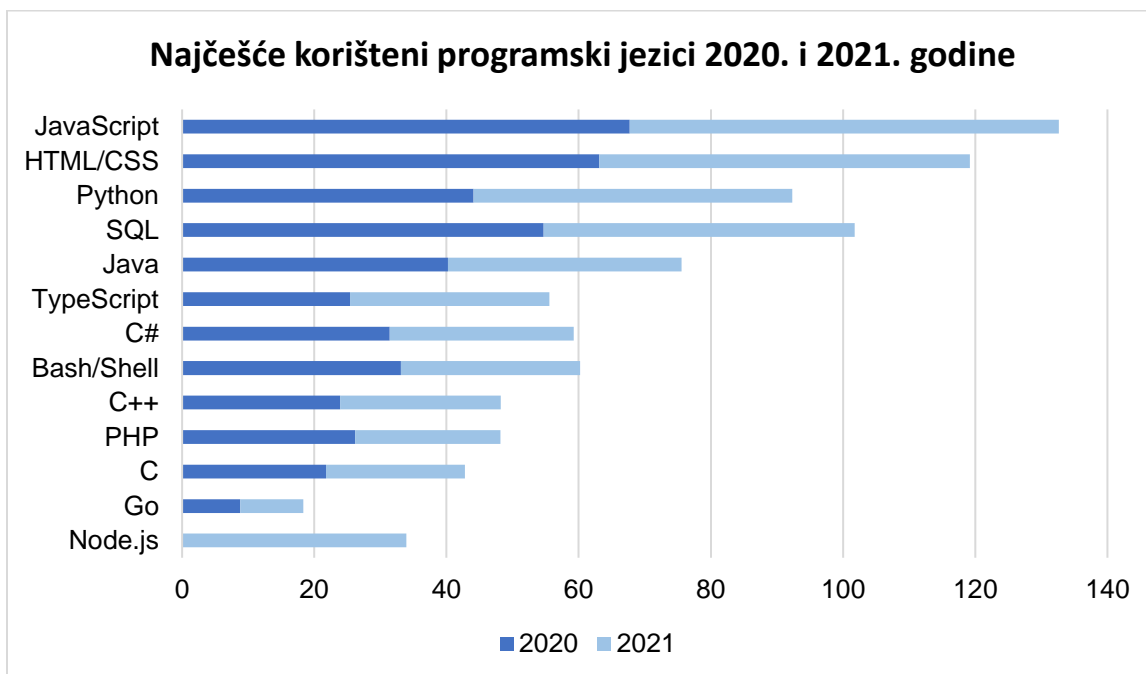


Grafikon 1 Postotak zastupljenosti programera u svijetu 2021. godine

(Izrada autorice rada prema: <https://insights.stackoverflow.com/survey/2021#developer-profile>,
30.06.2021.)

Iz grafikona je vidljivo kako su danas full-stack programeri sa 49,47% najzastupljeniji u svijetu, a nakon njih odmah slijede back-end i front-end područja programiranja. Na zadnjem mjestu nalaze se programeri video igara sa 3,18% zastupljenosti.

Sukladno rezultatima prethodnog istraživanja, sljedeći grafikon prikazuje najpopularnije jezike kod programera. Za usporedbu su prikazani podaci 2020. i 2021. godine. Iz grafikona je vidljivo kako JavaScript, već devetu (2021.) godinu za redom, slavi kao najčešće korišten programski jezik. Prvih pet pozicija je nepromijenjeno osim što je Python 2021. godine ispred SQL-a. Iznenađenje je Node.js koji se pojavio u 2021. godini sa 33,91% korištenja.



Grafikon 2 Najčešće korišteni programski jezici 2020. i 2021. godine

(Izrada autorice rada prema: <https://insights.stackoverflow.com/survey/2020#technology>,
<https://insights.stackoverflow.com/survey/2021#technology>, 30.06.2021.)

Iz priloženih podataka vidljivo je kako danas postoji mnogo programskih jezika i vrsta programera, ali kada se radi o početniku u svijetu programiranja iznimno je važno odabrati programski jezik koji će olakšati proces učenja osnova programiranja. Stoga slijedi poglavlje u kojem će se govoriti o programskim jezicima namijenjenim učenju.

2.3.1. Programski jezici namijenjeni učenju

Programske jezike može se podijeliti prema mnogo kriterija, a jedan od njih je njihova namjena. Kada se govori o izradi tečajeva programiranja, zasigurno se radi o namjeni učenja. Programski jezici namijenjeni učenju moraju biti sintaksom jednostavniji i lakši za razumijevanje, a cilj svakog od njih je da pružaju usvajanje osnovnih znanja programiranja koja se kasnije mogu primijeniti u ostalim programskim jezicima.

Istraživanjem različitih izvora Python slovi kao najbolji odabir programskog jezika kada je učenje osnova programiranja u pitanju. To dokazuje i indeks popularnosti programskog jezika (*PYPL Indeks, kolovoz 2021. godine*), prema kojem se Python nalazi na prvom mjestu popularnosti s udjelom od 29,93% kada je posrijedi pretraživanje vodiča (*engl. tutorial*) u Googleu (Grafikon 3).



Grafikon 3 Popularnost programskih jezika prema pretraživanju vodiča na Googleu

(Izrada autorice rada prema: <https://pypl.github.io/PYPL.html>, 04.08.2021.)

Python je programski jezik koji je razvio Guido Van Rossum u kasnim osamdesetim godinama. Koristi se za web programiranje, računalnu grafiku, vizualizaciju, matematiku, obrazovanje i mnoga druga područja računalne znanosti (*Lee, 2017: 14*). Jednostavnost i lako korištenje glavne su karakteristike ovog programskog jezika.

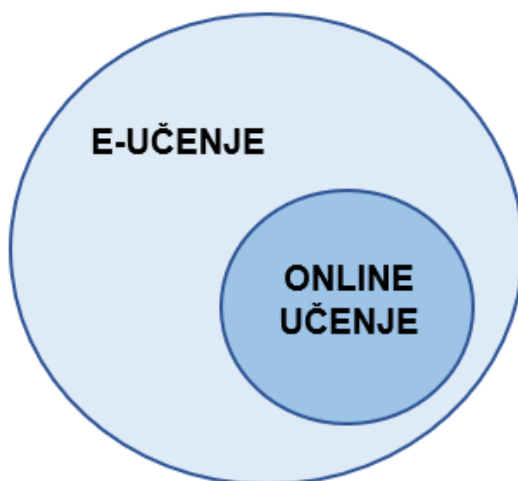
Samim time pruža laku čitljivost koda čime je shvaćanje sintakse znatno olakšano. Sve verzije Pythona su besplatne, a danas postoje i mnoge knjige, priručnici i materijali o ovom programskom jeziku, što je za početnike zasigurno dodatni plus. Već duži niz godina mnogi stručnjaci u ovom području kao preporuku programskog jezika za učenje programiranja upravo daju Python. Sve ovo prednosti su koje programski jezik Python stavljaju na prvo mjesto kada se govori o programskim jezicima namijenjenim učenju. Iz tog razloga, u glavnom dijelu ovog rada, test će se sastojati od pitanja koja se odnose na osnove programiranja u Pythonu.

3. ONLINE UČENJE

Obrazovanje je neizostavan proces u razvoju svakog čovjeka i predstavlja jedno od osnovnih ljudskih prava. Kako kaže slavni Nelson Mandela: „*Obrazovanje je najmoćnije oružje koje možete koristiti kako bi mijenjali svijet.*“ Ovom izrekom dovoljno se govori o važnosti i moći obrazovanja. Temelji obrazovanja postavljeni su kroz povijest, a razvoj ovog područja ne prestaje.

Pojam “e-obrazovanje” danas je već u širokoj upotrebi. Veliki značaj za to ima IKT (informacijsko komunikacijska tehnologija), čijim razvojem dolazi do sve jačeg prihvaćanja prefiksa “e-“. Tako su danas pojmovi poput “e-trgovina”, “e-poslovanje”, “e-bankarstvo”, ali i razni sustavi kao što su u Hrvatskoj poznati “e-Građani”, “e-Porezna”, “e-Dnevnik”, postali dio svakodnevnog života. To je dokaz kako informacijsko komunikacijska tehnologija polako ulazi u sva područja ljudskog djelovanja.

Prije same definicije online učenja, potrebno je reći kako je ono manji dio elektroničkog učenja ili e-učenja (*engl. e-learning*). Odnosno, e-učenje je širi pojam od online učenja (Slika 5).



Slika 5 Odnos e-učenja i online učenja

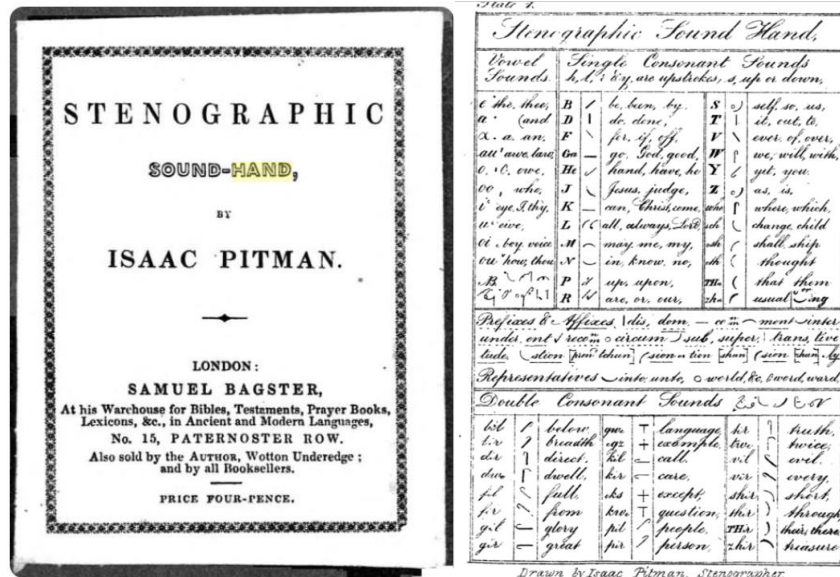
(Izrada autorice rada)

Boer i Collis (2002) kako je navedeno u knjizi Ćukušić i Jadrić (2012: 13), e-učenje definiraju kao potporu učenju primjenom mrežne tehnologije, a posebice web tehnologije. Dok online učenje omogućuje polaznicima pristup svim sadržajima i komunikaciju putem interneta (Karadeniz, 2009 navedeno u Ćukušić i Jadrić, 2012: 13). Prethodno definirane pojmove e-učenje i online učenje mnogi danas upotrebljavaju kao sinonime. To i nije velika greška, iako neki autori ističu razliku između njih. Naime, proces e-učenja ne mora se nužno odvijati putem interneta, npr. klasična nastava uz pomoć IKT-e. Dok se na online učenje gleda kao na e-učenje na daljinu, odnosno učenje koje se u potpunosti odvija putem interneta.

3.1. Razvoj i vrste online učenja

Nakon definicije online učenja potrebno je reći nešto i o razvoju cjelokupnog sustava e-učenja kroz povijest. Iako današnje e-učenje iziskuje primjenu različite tehnologije, povijest ovog područja seže u doba kada ni računala nisu bila razvijena. Izraz "e-učenje" postoji tek od 1999. godine, ali njegovi temelji postavljeni su znatno ranije. Danas je to teško zamisliti, ali tako je Isaac Pitman 1837. godine objavio svoj izum, *Stenographic Sound-Hand*, sustav stenografije³ za engleski jezik (Slika 6). Smatra ga se ocem učenja na daljinu jer je 1840. godine, uspostavom poštanske usluge u Engleskoj, počeo provoditi prvi tečaj obrazovanja na daljinu. Podučavao je vlastiti sustav slanjem tekstova prepisanih u stenografiju na razglednicama, a od svojih je učenika primao prijepise u zamjenu za ispravak (*Pitman Shorthand, & The First "Correspondence Course"*, [url](#)). To je uveliko doprinijelo razvoju dopisnih škola, od kojih je prva osnovana 1840. godine baš na tom području (Ujedinjeno Kraljevstvo). Do masovne upotrebe učenja na daljinu došlo je 1969. godine kada je osnovano otvoreno sveučilište (*engl. Open University*) također u Velikoj Britaniji. Time se 19. stoljeće smatra početkom pojave učenja na daljinu u svijetu (Ćukušić i Jadrić, 2012: 17-18).

³ Prema Hrvatskoj enciklopediji Leksikografskog zavoda Miroslav Krleža, mrežno izdanje: „*stenografija* je brzo pismo kojim se može zapisati i najbrži govor tako što za svako slovo, odnosno glas postoji poseban kratki abecedni znak.“



Slika 6 Stenographic Sound-Hand by Isaac Pitman

(Izvor: <https://www.historyofinformation.com/detail.php?id=2050>, 04.07.2021.)

Keengwe (2005) kako je navedeno u knjizi Ćukušić i Jadrić (2012: 17-19) razvoj e-učenja promatra kroz četiri razdoblja:

- obrazovanje u obliku treninga koji vodi instruktor,
- razdoblje multimedije,
- prvi val e-učenja i
- drugi val e-učenja.

Sve od 1840. do 1983. godine na obrazovanje se gledalo kao na organizirane treninge koje su vodili nastavnici i tako prenosili znanje polaznicima. Pojavom IBM PC računala dolazi do veće upotrebe osobnih računala, što je obilježilo razdoblje od 1984. do 1993. godine, odnosno razdoblje multimedije. Pojava CD-ROM medija rezultirala je da se tečajevi nalaze na njima, čime je obrazovanje postalo dostupnije i modernije. Prvi val e-učenja (od 1994. do 1999. godine) karakterizira pojava World Wide Weba (WWW, Web). Dolazi do sve većeg korištenja elektroničke pošte i internetskih preglednika, a učenje na daljinu provodilo se putem e-maila, intraneta i weba. Razdoblje nakon 1999. godine donosi mnoge tehnološke promjene i napredak.

Zahvaljujući tome, u drugom valu e-učenja dolazi do velikih promjena u načinu organiziranja i provođenja učenja što je postavilo standarde za budući suvremeni razvoj e-učenja.

Ovisno o načinu korištenja informacijsko komunikacijske tehnologije postoje sljedeći oblici e-učenja (*Sveučilišni računski centar (srce), url*):

- **klasična nastava** - nastava u tradicionalnoj učionici,
- **nastava uz pomoć informacijsko komunikacijske tehnologije** - klasična nastava u kojoj se IKT koristi u svrhu poboljšanja nastavničkog procesa,
- **hibridna ili mješovita nastava** - kombinacija nastave u učionici i nastave uz pomoć IKT-e i
- **online nastava** - nastava uz pomoć IKT-e, ali u potpunosti na daljinu.

Kako je već rečeno, online učenje se u potpunosti provodi putem interneta. Tada se s obzirom na odnos prostora i vremena razlikuju (*Čukušić i Jadrić, 2012: 23-24*):

- **sinkrono online učenje** - podrazumijeva istovremeno prisustvo polaznika i predavača na određenom virtualnom mjestu i u određeno vrijeme putem određenog internetskog medija, iako su geografski dislocirani. Tu spadaju videokonferencije, telekonferencije, razni programi i aplikacije za komunikaciju, virtualne učionice itd.,
- **asinkrono online učenje** - ne zahtijeva uključivanje predavača, a polaznici ne moraju nužno biti prisutni u isto vrijeme kako bi pratili sadržaj i učili. On je dostupan uvijek, a time si polaznik sam pravi svoj raspored i tempo učenja. Komunikaciju polaznici i nastavnici ostvaruju putem elektroničke pošte ili ugrađenih foruma. Ovdje spadaju razni tečajevi na internetu, web stranice edukativnog karaktera, razni vodiči za učenje itd.

Nakon prikaza razvoja i vrsta online učenja može se zaključiti kako se ovaj moderni način učenja temelji na učenju koje je potpomognuto informacijsko komunikacijskom tehnologijom, vlastitim tempom i učenju koje ne ovisi o vremenu i prostoru.

3.2. Prednosti i nedostaci online učenja

Tehnologija sama po sebi ima mnogo prednosti i unosi ih svako područje u kojem se pojavljuje pa tako i u proces e-učenja. Iako za sobom donosi i određene nedostatke, količina prednosti je ipak veća. Bez tehnologije, e-učenje ne bi bilo razvijeno do ove mjere i samim time ne bi imalo sve mogućnosti koje danas pruža. Popularnost e-učenja neizmjereno raste, a posebice dolaskom pandemije koja je zahvatila cijeli svijet. Pandemija je na neki način doprinijela razvoju e-učenja, ali je i prisilila određene zemlje da omoguće e-obrazovanje, čime je potaknula njihov osobni razvoj. Iako tržište e-learninga neizmjereno raste, još uvijek e-učenje nije u potpunosti zamijenilo tradicionalno obrazovanje. U nastavku je prikazana tablica s prednostima i nedostacima e-učenja u odnosu na tradicionalno učenje (Tablica 2).

Tablica 2 Prednosti i nedostaci e-učenja u odnosu na tradicionalno učenje

PREDNOSTI	NEDOSTACI
samostalno i personalizirano učenje	osnovna razina računalne pismenosti
bez geografskih ograničenja	manjak socijalizacije
brži i širi pristup informacijama	velika količina bespotrebnih informacija
veći osjećaj slobode	niska motivacija za učenje
korištenje tehnologije	moguće smetnje i prekid procesa učenja

(Izrada autorice rada)

Jedna od prednosti e-učenja je da nema geografskih ograničenja, odnosno nije bitno gdje se osoba nalazi, dok manjak socijalizacije uzrokuje nedostatak e-učenja. Prednost tradicionalnog učenja je da postoji zadovoljavajuća razina socijalizacije jer je osoba okružena drugim ljudima, ali nedostatak je što je vezana za određeno mjesto. Time se može zaključiti, ono što su prednosti e-učenja ujedno su nedostaci tradicionalnog obrazovanja, a nedostaci e-učenja su prednosti tradicionalnog obrazovanja.

Usporedba sinkronog i asinkronog online učenja prikazana je u sljedećoj tablici (Tablica 3).

Tablica 3 Usporedba sinkronog i asinkronog online učenja

	SINKRONO UČENJE	ASINKRONO UČENJE
KADA?	Točno određeno vrijeme	Vlastiti odabir vremena učenja
GDJE?	Geografska lokacija nije bitna	Geografska lokacija nije bitna
TEHNOLOGIJA?	Potrebna računalna pismenost	Potrebna računalna pismenost
KOMUNIKACIJA?	Povratni odgovor stiže odmah	Povratni odgovor treba čekati
SOCIJALIZACIJA?	Veća razina socijalizacije	Niža razina socijalizacije
SLOBODA?	Definiran raspored učenja	Vlastiti raspored učenja

(Izrada autorice rada)

Ono što je zajedničko ovim načinima online učenja je da geografska lokacija nije bitna i da mora postojati određena razina računalne pismenosti kod polaznika. Dovoljno je da postoji mogućnost povezivanja na internet i proces e-učenja može započeti.

Ovisno o karakteristikama i željama polaznika i nastavnika, području učenja, mogućnostima itd., odabire se ona vrsta e-učenja koja će pružiti najkvalitetniji proces učenja. Kada govorimo o sinkronom i asinkronom online učenju tu postoje iznimno različite značajke koje mogu utjecati na kvalitetu učenja kod pojedinca. Nekome odgovara da ima više slobode i da si samostalno određuje raspored, tempo učenja i usvajanja gradiva, dok za nekoga to može biti velika zamka. Stoga je danas sve više prisutna kombinacija ova dva modela.

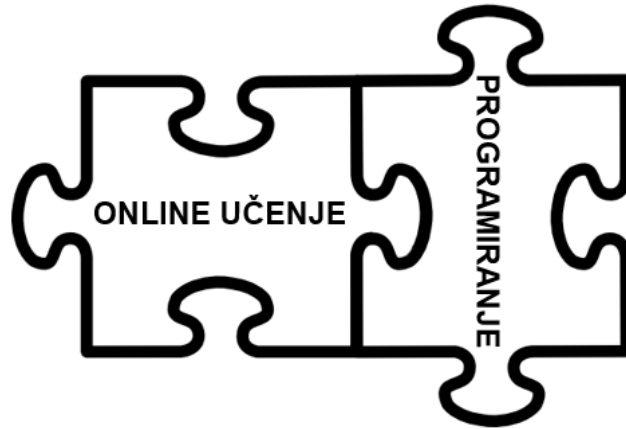
4. ONLINE UČENJE I PROGRAMIRANJE

Nakon prijašnjih odlomaka gdje se govorilo o programiranju i online učenju, u ovom dijelu rada dolazi do spajanja navedenih područja. Danas je velika šteta ne iskoristiti znanje koje se pruža nadohvat ruke. Naime, znanje nikada nije bilo bliže. Računalo, internet i volja jedini su potrebni resursi kako bi započeli proces online učenja.

Mnogi ljudi, izuzev formalnog, svoje slobodno vrijeme posvećuju neformalnom obrazovanju. Razlika ova dva pojma je što formalno obrazovanje provode obrazovne ustanove i pojedinac završetkom stječe priznate diplome i kvalifikacije, dok neformalno obrazovanje je svaki oblik obrazovanja koji ne dovodi do stjecanja novih kvalifikacija nego nudi unaprjeđenje profesionalnih i osobnih vještina i stjecanje novih kompetencija. Nakon završetka srednjoškolskog obrazovanja, u Hrvatskoj postoje mnogi fakulteti za one koji žele formalno obrazovanje posvetiti informatici i području programiranja.

Naučiti programirati nije ni malo jednostavan proces, a kako bi se steklo zvanje programera nije dovoljno samo formalno obrazovanje. Ako se osoba odluči baviti ovim područjem, uz formalno, neformalno obrazovanje je nužno za daljnji razvoj u IT sektoru. Barem je danas, zahvaljujući tehnološkom razvoju, neformalno obrazovanje nikad dostupnije. Bilo da se radi o početniku ili već postojećem programeru koji želi unaprijediti svoje znanje, na internetu postoji iznimno mnogo izvora i načina učenja kada se radi o programiranju. Uz to, jedini uvjet koji se traži od osobe za online učenje programiranja je osnovna razina računalne pismenosti i poznavanje rada na računalu. Upravo se iz ovih razloga programiranje izvrsno uklapa u proces online učenja (Slika 7).

U nastavku rada govori se o alatima i tehnologijama za online učenje programiranja.



Slika 7 Online učenje i programiranje

(Izrada autorice rada)

4.1. Vrste tehnologija i alata za online učenje programiranja

Učiti programirati putem interneta iz udobnosti vlastitog doma prije nije bilo moguće niti zamisliti. Tu do izražaja dolazi značaj tehnologije koji je za razvoj online učenja iznimno velik. Upravo zbog toga, danas postoji veliki broj alata koji su namijenjeni učenju programiranja putem interneta.

S obzirom na njihove sličnosti i razlike, alati za online učenje programiranja mogu se svrstati u četiri osnovne skupine, a to su (*Miličević, 2020*):

- vizualni alati za učenje programiranja,
- alati za učenje programiranja putem video igara,
- alati za učenje programiranja u okruženju Web 2.0 i
- alati podrške za učenje programiranja u višim programskim jezicima.

Vizualni alati za učenje programiranja omogućavaju kreiranje programske logike pomoću slaganja i povezivanja vizualnih elemenata. Za početnike u svijetu programiranja ovo je znatno lakši način početka učenja umjesto pisanja koda. Zašto? Ako se početnik nije do sada susreo s programiranjem, pomoću vizualizacije može znatno lakše razumjeti koncept i srž samog programiranja.

Uz to, svaka promjena objekta u alatu rezultira vizualnim prikazom što uvelike olakšava shvaćanje rješavanja zadataka i lakše ispravljanje nastalih pogrešaka. Neki od najpopularnijih alata za vizualno učenje programiranja su (*Miličević, 2020*):

- alati za učenje kroz dijagram toka (*Miro, Lucidchart, RAPTOR, FlowGorithm, Gliffy i dr.*),
- alati za vizualizaciju algoritma (*ViSA, PITON i dr.*) i
- mini-jezici (*MSWLogo, Guido van Robot, Pascal, Basic i dr.*).

Jedan od najkreativnijih načina danas za učenja programiranja je putem **video igara**. One su zasigurno najveći interes mlađih dobnih skupina i ta činjenica se može iskoristiti kako bi vrijeme provedeno za igricama uz zabavu bilo ujedno i korisno. Ovaj način učenja nije namijenjen samo djeci već i starijim uzrastima. Video igre predstavljaju zanimljiv i interaktivan način kako početnicima približiti svijet programiranja i da na programiranje ne gledaju sa strahom, nego kao nešto zabavno, jednostavno i impresivno. Cilj učenja programiranja putem video igara je naučiti osobe logičkom razmišljanju koje je neophodno ako se žele baviti programiranjem. Ovisno o dobnom uzrastu razlikuju se:

- video igre za djecu (*CodeMonkey, Daisy the Dinosaur, LightBot, Algorithm City: Coding Game i dr.*) i
- video igre za stariji uzrast (*CodinGame, CodeCombat, Screeeps i dr.*).

Ovime se dokazuje kako video igre nisu nužno samo nešto loše, imaju one i svoje prednosti, a onima koji ih iskoriste na pravi način pruža se velika količina znanja koje kasnije mogu nadograđivati kroz druge alate.

Pojava Weba 1989. godine donijela je velike promjene u računalnom svijetu. Radi se o najčešće korištenom internetskom servisu koji korisnicima omogućuje pregledavanje mnoštva digitalnih dokumenata putem interneta. Web se smatra glavnim razlogom za veliku popularnost interneta zbog čega se pojmovi internet i web često koriste kao sinonimi. Izumitelj Weba Tim Berners-Lee, također je izumio poznati jezik oznaka HTML (*HyperText Markup Language*) i komunikacijski protokol HTTP (*Hypertext Transfer Protocol*) koji se koristi za slanje informacija putem weba (*Deitel, 2008: 11*).

Od samog osnivanja, web se kroz godine nastavio razvijati i tako danas postoje tri generacije weba. U nastavku je tablica koja prikazuje njihove ključne razlike (Tablica 4).

Tablica 4 Razlika Web 1.0, Web 2.0 i Web 3.0

OBILJEŽJE	Web 1.0	Web 2.0	Web 3.0
Pojava	1996-2004	2004- 2016	od 2016
Korištenje	Čitanje	Čitanje, pisanje	Čitanje, pisanje, izvršavanje
Web	Statičan	Dinamičan	Semantički web
Sadržaj	Posjedovanje	Dijeljenje	Objedinjavanje
Povezivanje	Informacija	Ljudi	Znanja
Interakcija	Web obrasci	Web aplikacije	Pametne aplikacije
Pretraživanje	Imenici	Ključne riječi	Ponašanje korisnika
Tehnologija	HTML	XML/RSS	RDF/RDFS/OWL

(Izrada autorice rada prema: <https://www.geeksforgeeks.org/web-1-0-web-2-0-and-web-3-0-with-their-difference/>, 10.07.2021.)

Dolaskom druge generacije 2004. godine, web se podiže na jednu novu razinu. Korisnici više nisu samo čitatelji sadržaja i informacija, već sudjeluju u njihovom stvaranju, izmjenjivanju i prenošenju. Time se ostvarila dvosmjerna komunikacija između korisnika i poslužitelja što je ujedno i velika prekretnica u razvoju weba. U svom radu *Putica (2018: 100-102)* tvrdi kako su prve dvije generacije weba namijenjene za rad s fizičkim osobama koje mogu znatno lakše upravljati informacijama. Web sadržaj dobiva na značenju tek nakon ljudske interpretacije, dok je za strojeve on i dalje nerazumljiv. Upravo treća generacija, tzv. web budućnosti želi omogućiti strojevima da razumiju informacije na webu tako što bi se informacijama dodijelilo značenje - semantički web. Uz elemente umjetne inteligencije cilj je stvoriti još više povezan, otvoren i inteligentan web za korisnike.

Upravo **alati Web 2.0 okruženja** omogućavaju učenje raznih područja putem interneta. Razvojem ove generacije weba korisnicima se pruža mogućnost da kreiraju sadržaj koji mora biti podijeljen. Zbog toga je došlo do razvijanja velikog broja internetskih platformi koje služe za distribuciju sadržaja.

Najpopularnije od njih su: blogovi, wiki sustav, podcasti i društvene mreže. S obzirom na to da sadržaj može kreirati svatko, njegovu točnost treba provjeriti. Ipak, kada je riječ o učenju programiranja potrebno je da se radi o pouzdanim izvorima učenja. Danas postoje razni načini za učenje programiranja pomoću Web 2.0 alata, a neki od njih su:

- korištenje YouTube vodiča,
- interaktivne web stranice (*W3Schools, SoloLearn, Codecademy i dr.*),
- online platforme (*freeCodeCamp, Coursera, Udemy, edX, Campster i dr.*),
- LMS (*engl. Learning Management System*) sustavi,
- korištenje e-knjiga i drugih materijala dostupnih na internetu i
- korištenje pouzdanih web stranica i foruma (*Stack Overflow i dr.*).

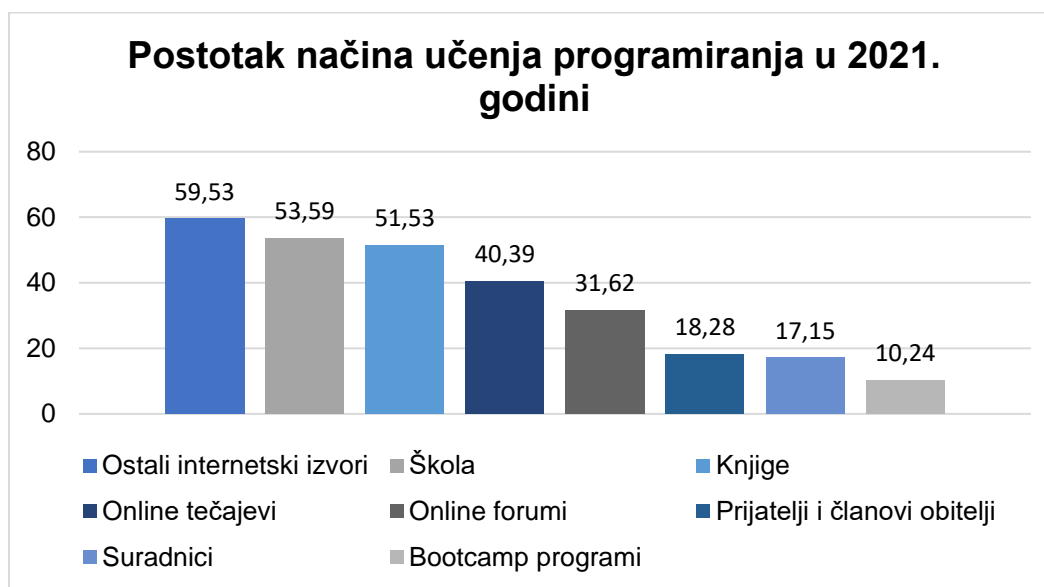
Svaki prethodno navedeni način ima neizmjereno puno izvora za učenje programiranja. Primjerice, velika većina interaktivnih web stranica nudi učenje raznih programskih jezika, ne samo jednog. YouTube ima neizmjereno velik broj video materijala kada je u pitanju programiranje. Mnogi od ovih načina učenja su besplatni, a ako osoba želi, izdvajanjem manje svote novca, na raznim online platformama može kupiti tečaj čijim polaganjem dobiva određeni certifikat. Danas je učenje programiranja zaista lako dostupno, pristupačno i svatko, početnici pa i iskusni programeri, mogu pronaći nešto za sebe.

Alati podrške za učenje programiranja u višim programskim jezicima su alati koji omogućavaju, tzv. “pravo“ programiranje, odnosno pisanje koda u određenom programskom jeziku. Kada je osoba savladala srž i opću logiku programiranja (kroz vizualne alate, video igre, web alate...), poželjno je nastaviti učiti uz neki alat iz ove skupine. Radi se o alatima koji pružaju pomoć tako što daju prijedloge kada je pravilno pisanje sintakse u pitanju i pronalaze greške u kodu. To iznimno olakšava početak programiranja i omogućuje ostvarenje konačnog cilja učenja, a to je stjecanje znanja i vještina koje nam daju mogućnost da programiramo u višim programskim jezicima, odnosno jezicima treće generacije (C, Java, PHP, HTML, JavaScript, C++, C#, Python i dr.). Koji programski jezik odabrati kao početnik definirano je u poglavlju 2.3.1. ovog rada, a ako se radi o iskusnom programeru koji želi proširiti svoje znanje izbor programskih jezika je zaista velik.

Kada se odabere programski jezik potrebno je potražiti odgovarajući alat jer danas ih ima zaista puno. U nastavku su navedeni neki od popularnih programskih jezika i njima pripadajući alati (Miličević, 2020):

- **Python** - *Prospector, Pylint*,
- **Java** - *QAPlug, CheckStyle*,
- **JavaScript** - *Atom, Sublime Text*,
- **PHP** - *Zen Studio, Rapid PHP Editor* i
- **C++** - *Verificator, Cppcheck*.

Učenje programiranja putem interneta zajednička je namjena svih vrsta alata. Ovisno o trenutnoj razini znanja, potrebama i željama, osoba može izabrati hoće li učiti programirati pomoću vizualnih alata, igranjem različitih video igara, korištenjem web alata ili će koristiti različite alate podrške. Unutar svake vrste alata postoji nebrojeno mnogo izvora kada je programiranje u pitanju, tako da svatko može pronaći ono što mu treba, a samim time se broj korisnika online učenja povećava čime u konačnici ovo tržište raste.



Grafikon 4 Postotak načina učenja programiranja u 2021. godini

(Izrada autorice rada prema: <https://insights.stackoverflow.com/survey/2021#developer-profile-experience>, 10.07.2021.)

Prema podacima *Stack Overflow* organizacije (Grafikon 4), skoro 60% ispitanika od njih 82.963 reklo je kako je učilo programirati iz internetskih izvora. Mlađi ispitanici uglavnom uče pohađanjem online tečajeva, dok su stariji skloniji tradicionalnijim načinima učenja (knjige, škole).

Danas mnogi početnici odabiru pohađanje online tečajeva kada je učenje osnova programiranja u pitanju. Ovaj način učenja oduvijek je pri vrhu kada se uspoređuju web alati za učenje programiranja, a u doba pandemije tržište online tečajeva ostvaruje značajan rast. Osobe su više sklone odabiru ovog načina učenja jer im on pruža:

- **sigurnost** - tečajeve pružaju poznate organizacije,
- **povezanost** - veliki broj stručnjaka i drugih polaznika,
- **slobodu** - vlastiti tempo učenja i usvajanja gradiva i
- **zadovoljstvo** - dobivanje certifikata što je za početnike dodatna motivacija.

Koliko je zapravo teško izraditi jedan online tečaj programiranja, bit će prikazano u glavnom dijelu rada gdje se govori o izazovima s kojima se susreću njihovi autori. Kako bi se prikazalo rješenje jednog od najvećih izazova, koristit će se sustavi za upravljanje učenjem jer oni omogućavaju kreiranje vlastitih tečajeva. Prije toga, potrebno je reći nešto općenito o tim sustavima.

4.2. Sustavi za upravljanje učenjem - LMS

Razvoj tehnologije doveo je do pojave virtualnih okruženja za učenje. Danas postoje četiri vrste sustava (Ćukušić i Jadrić, 2012: 78-82):

- **LMS** (*engl. Learning Management System*) - sustav za upravljanje učenjem služi za izradu, izvođenje i upravljanje online tečajevima,
- **CMS** (*engl. Content Management System*) - sustav za upravljanje sadržajem omogućuje kreiranje različitih sadržaja (slika, reportaža, članaka itd.),
- **LCMS** (*engl. Learning Content Management System*) - sustav za upravljanje sadržajem i učenjem omogućuje stvaranje, čuvanje i korištenje sadržaja za učenje,
- **LCAS** (*engl. Learning Content Authoring System*) - podrazumijeva skup raznih autorskih alata za izradu sadržaja učenja.

Uvođenjem informacijsko komunikacijske tehnologije u proces obrazovanja dolazi do sve većeg korištenja sustava za upravljanje učenjem. Stoga se LMS sustavi smatraju podrškom e-obrazovanju. *Morrison (2003) kako je navedeno u knjizi Ćukušić i Jadrić (2012: 78) tvrdi kako je sustav za upravljanje učenjem (LMS) aplikacija utemeljena na webu koja sadržava niz alata koji omogućuju centralizaciju i automatizaciju različitih segmenta učenja pomoću sljedećih funkcija:*

- registracija polaznika,
- održavanje profila polaznika,
- održavanje kataloga e-tečajeva,
- spremanje i isporuka e-tečajeva za samostalno učenje,
- preuzimanje modula i alata potrebnih za e-učenje,
- praćenje i snimanje napretka polaznika,
- ocjenjivanje polaznika,
- spremanje rezultata procjena i kreiranje potrebnih izvještaja.

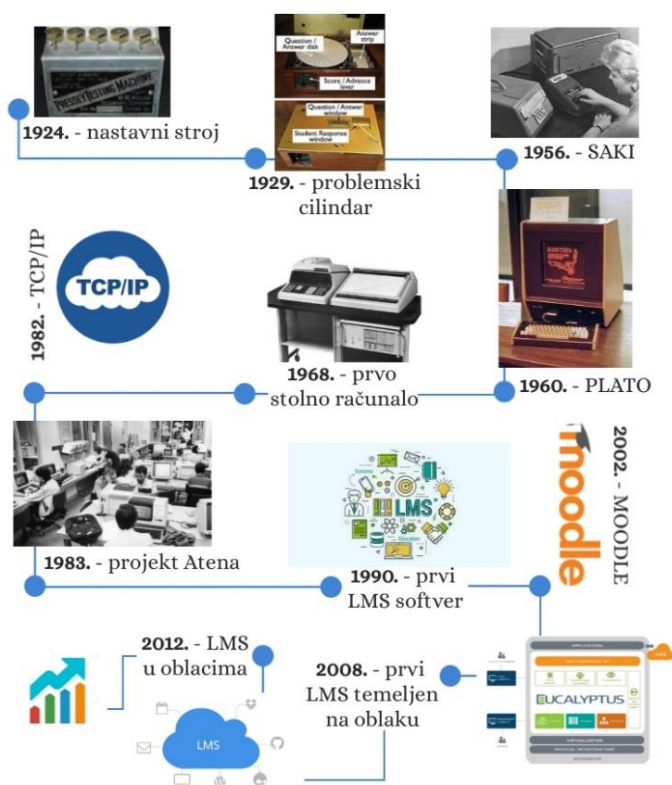
Postoji mnogo različitih definicija, ali ključne značajke sustava za upravljanje učenjem mogu se pronaći u njegovoj skraćenici:

- **L** Learning Učenje - pružanje učenja,
- **M** Management Upravljanje - organizacija i upravljanje tečajevima,
- **S** System Sustav - sve na jednom mjestu.

Povijest LMS sustava je kratka (Slika 8). Značajna godina je 1924. kada je sve počelo izumom nastavnog stroja. Stroj je ličio na pisaću mašinu s prozorom u kojem su se postavljala pitanja s više odabira. Umjesto tipkanja, osoba bi pritiskom na odabrani gumb bušila papir koji se nalazio na stražnjoj strani stroja i tako označavala svoj odabrani odgovor. Izumi su se počeli samo nizati i 1929. godine razvijen je problemski cilindar koji osim što je provjeravao znanje osobe, provjeravao je i ispravnost danih odgovora. SAKI je prvi stroj koji je ponudio personalizirana pitanja iz prakse, a kako je učinak osobe rastao tako se povećavala složenost sljedećih pitanja. PLATO je računalni program nastao 1960. godine, a smatra se izumom koji je najviše sličio današnjim LMS sustavima. Polaznicima je omogućio učenje vlastitim tempom i komunikaciju s drugima.

Pojava prvog stolnog računala omogućila je rješavanje matematičkih i inženjerskih problema deset puta brže od drugih strojeva. Razvojem TCP/IP (engl. *Transmission Control Protocol/Internet Protocol*) protokola 1982. godine razvilo se online učenje. Godinu dana kasnije pokrenut je projekt Atena čija je namjena bila proširiti upotrebu računalnih alata i na druga područja. Prvi LMS softver razvijen je 1990. godine, a 2002. nastao je poznati Moodle. Nedugo zatim dolazi do pojave mnogih sustava za upravljanje učenjem, a 2008. godine nastao je Eucalyptus, prvi LMS temeljen na oblaku. Ubrzo dolazi do sve veće pojave LMS sustava u oblacima čime se tvrtke oslobađaju dodatnih troškova. Zahvaljujući svim ovim izumima, danas postoje moderni LMS sustavi čije su mogućnosti sve veće, a daljnji razvoj sustava za upravljanje učenjem je neupitan.

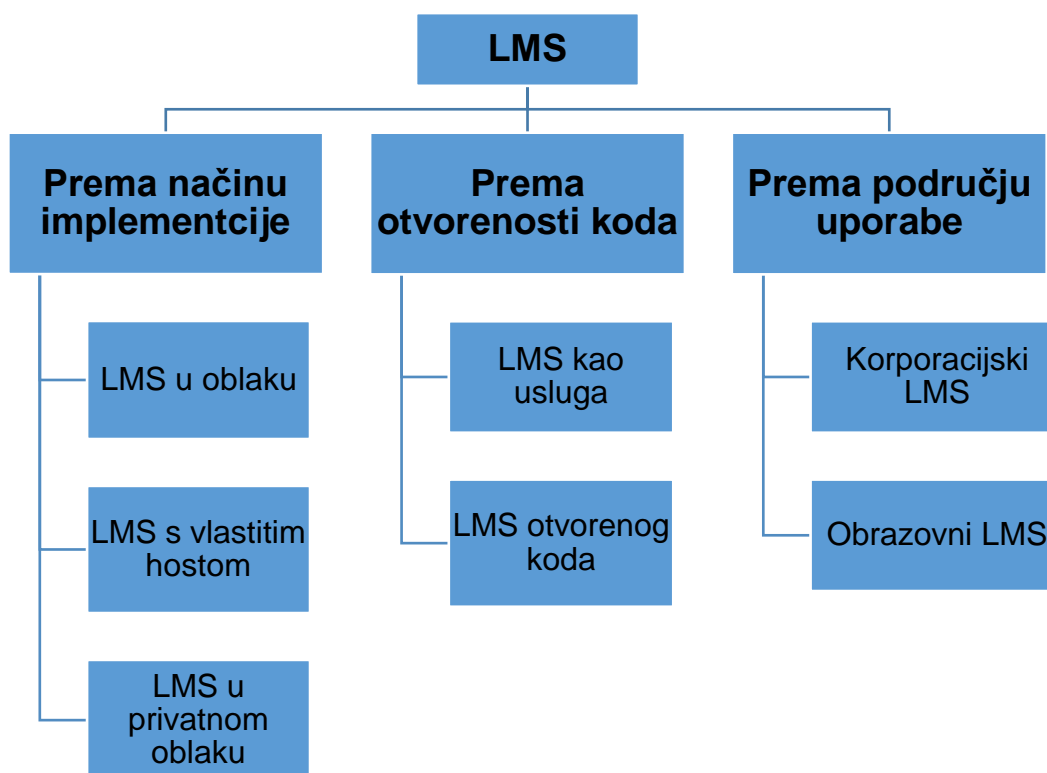
RAZVOJ LMS SUSTAVA



Slika 8 Razvoj LMS sustava

(Izrada autorice rada prema: <https://www.timetoast.com/timelines/history-of-lms-19f41c38-48f6-4f8d-b8f3-a3ebc4ebed15>, <https://www.worldmanager.com/resources/history-of-lms/>, 12.07.2021.)

Danas na tržištu postoji jako puno sustava za upravljanje učenjem. Sljedeća slika prikazuje njihovu podjelu prema tri kriterija.



Slika 9 Vrste sustava za upravljanje učenjem (LMS-a)

(Izrada autorice rada prema: <https://www.talentlms.com/what-is-an-lms#definition>, 12.07.2021.)

Korporacijski LMS uglavnom odabiru tvrtke koje žele obučiti svoje zaposlenike, dok obrazovni koriste škole, fakulteti, sveučilišta ili druge obrazovne ustanove. Ovisno o mogućnostima i potrebama, postoje besplatni i sustavi koji se plaćaju. Većina sustava koji nisu besplatni ima mogućnost odabira više paketa tako da korisnik ne treba bespotrebno plaćati značajke koje ne misli koristiti, dok besplatni sustavi zahtijevaju samostalnu instalaciju i održavanje sustava. S obzirom na način implementacije razlikuju se LMS u oblaku koji je u potpunosti pod nadzorom poslužitelja (model pretplate), LMS s vlastitim hostom u osnovi je web aplikacija koja se prodaje kao licencirani proizvod koju instalira i održava sam korisnik i LMS u privatnom oblaku koji je kombinacija prethodna dva modela. Na kraju, izbor ovisi o mogućnostima i potrebama korisnika, a cilj je odabrati onaj sustav koji će omogućiti što bolji proces online učenja.

Kada se govori o LMS tržištu zacijelo se radi o rastućem tržištu koje je 2020. godine eksplodiralo i svoj rast nastavlja i dalje. Veliki doprinos tome dale su obrazovne ustanove koje su zbog dolaska pandemije svoj obrazovni proces morale nastaviti online. Prema podacima svjetske organizacije veličina tržišta sustava za upravljanje učenjem u 2020. godini iznosila je 10,84 milijardi USD. Za 2021. godinu predviđa se rast na 13,38 milijardi USD, a za 2028. godinu na čak 44,49 milijardi USD (*Fortune Business Insights, 2021, url*).

Od svih postojećih softvera na tržištu nemoguće je izdvojiti najbolju LMS platformu. Vodeća svjetska platforma za pregled i odabir softvera *Capterra* u kategoriji “LMS Software“ nudi 1055 proizvoda. S obzirom na to da je u ovom radu prvenstveno naglasak na sustave otvorenog koda, korišten je filter opcije cijena gdje je odabrana opcija “Free“, filter obilježja gdje je odabrana opcija “izrada tečaja“ i rezultati su sortirani prema broju recenzija. Pronađeno je 145 proizvoda, a odabrana su prva četiri rezultata: CANVAS, Moodle, iSpring Suite i TalentLMS (*Capterra, url*). U nastavku slijedi tablica koja prikazuje općenite informacije o svakom odabranom sustavu za upravljanje učenjem.

Tablica 5 Usporedba odabranih LMS sustava

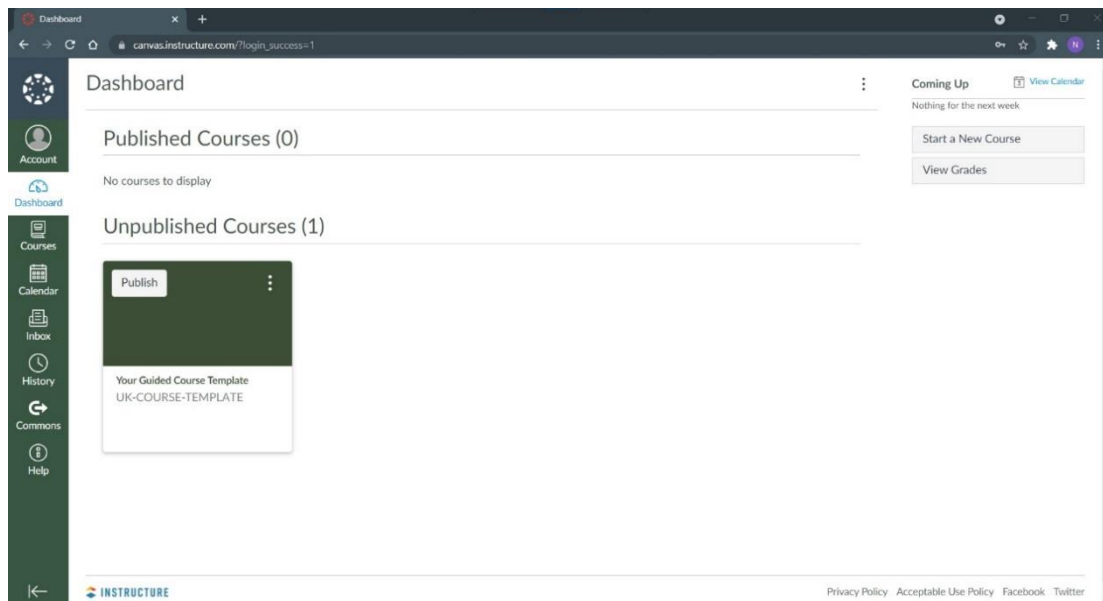
OBILJEŽJE	CANVAS	Moodle	iSpring Suite	TalentLMS
Godina	2011	2002	2001	2012
Mjesto	SAD	Australija	SAD	SAD
Broj korisnika	+30M	+200M	+55k	4M
Obrazovni alat	✓	✓	✓	✓
Free	Free-for-Teacher account	Free	Free Trial	Free Version
Web	✓	✓	✓	✓
Desktop	✓	✓	✓	✓

(Izrada autorice rada prema: <https://www.capterra.com/learning-management-system-software/compare/127214-80691-218265-132935/CANVAS-vs-Moodle-vs-iSpring-Suite-Max-vs-TalentLMS>, 13.07.2021.)

4.2.1. CANVAS

U ovom odlomku pobliže će se prikazati prvi odabrani LMS sustav. CANVAS je nastao 2008. godine od strane dva poduzetna studenta, dok je 2011. godine službeno pokrenut istoimeni sustav za upravljanje učenjem koji danas broji preko 30 milijuna korisnika diljem svijeta (*Canvas, url*). On omogućuje kreiranje tečajeva iz raznih područja, a zbog svoje jednostavnosti i popularnosti korištenja uvijek je među prvima kada je izbor LMS sustava u pitanju. Besplatan je za pojedinačne nastavnike, dok ga institucije moraju platiti. Kako bi se prikazale njegove mogućnosti u daljnjem radu, potrebno je bilo kreirati besplatan račun za učitelje.

Prijavom se otvara nadzorna ploča (*engl. dashboard*) na kojoj se nalazi predložak tečaja (Slika 10). Ono što uvelike olakšava kreiranje tečaja za početnike koji koriste ovaj sustav su uputstva koja se pojavljuju s desne strane ekrana. CANVAS ima dobro razrađen sustav korisničke podrške, a svojim korisnicima nudi fleksibilan pristup bez obzira na vrijeme, mjesto i vrstu uređaja. Sustav osigurava visoku razinu interakcije između nastavnika i polaznika, razne diskusije, nudi mogućnost procjene znanja polaznika, ali i implementaciju raznih autorskih alata pomoću kojih se sadržaji tečajeva mogu podići na višu razinu.

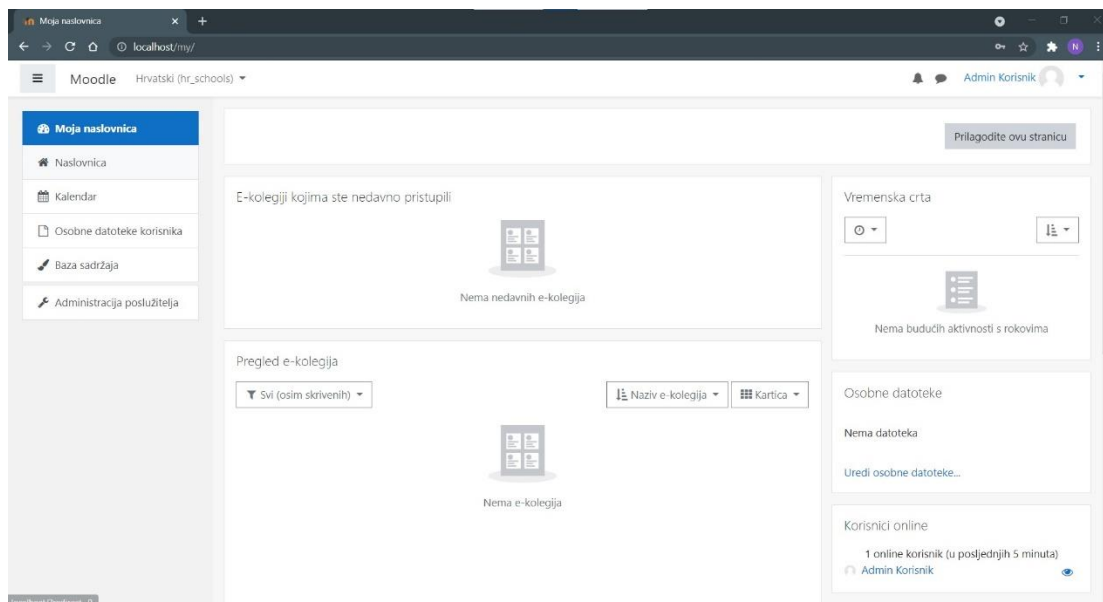


Slika 10 Izgled korisničkog sučelja - CANVAS

(Izrada autorice rada)

4.2.2. Moodle

Naziv Moodle kratica je koja potječe od „*Modular Object-Oriented Dynamic Learning Environment*“ što u prijevodu znači Modularno objektno-orijentirano dinamičko okruženje za učenje. Moodle je zasigurno najpopularniji svjetski sustav za upravljanje učenjem. Razvio ga je Martin Dougiamas, a prva verzija objavljena je 2002. godine. Od onda Moodle raste neizmjernom brzinom, a danas ova platforma ima više od 200 milijuna zadovoljnih korisnika u 243 zemlje diljem svijeta. Moodle je osmišljen kao snažan i integriran sustav koji svojim korisnicima pruža sigurno okruženje za učenje. Njegovo jednostavno sučelje, drag-and-drop značajka i kvalitetna popratna dokumentacija razlozi su za njegovo lako korištenje. Sve veći broj korisnika odlučuje se upravo za Moodle jer je besplatan, a uz mnoge dodatke koje posjeduje korisnik može proširiti njegove mogućnosti. Moodle nastoji otkloniti jezična ograničenja, tako je ova platforma danas dostupna na više od 120 jezika. Mnoge verzije rezultat su stalnih poboljšanja, a u trenutku pisanja ovog rada aktualna je verzija 3.11.1+ koja će biti korištena u daljnjoj izradi rada. Prateći YouTube vodič (<https://www.youtube.com/watch?v=nd4dytem4DA>), instalacija ovog LMS-a na računalo nije zahtjevna (*Moodle, url*). Nakon instalacije, Moodle je spreman za korištenje (Slika 11).

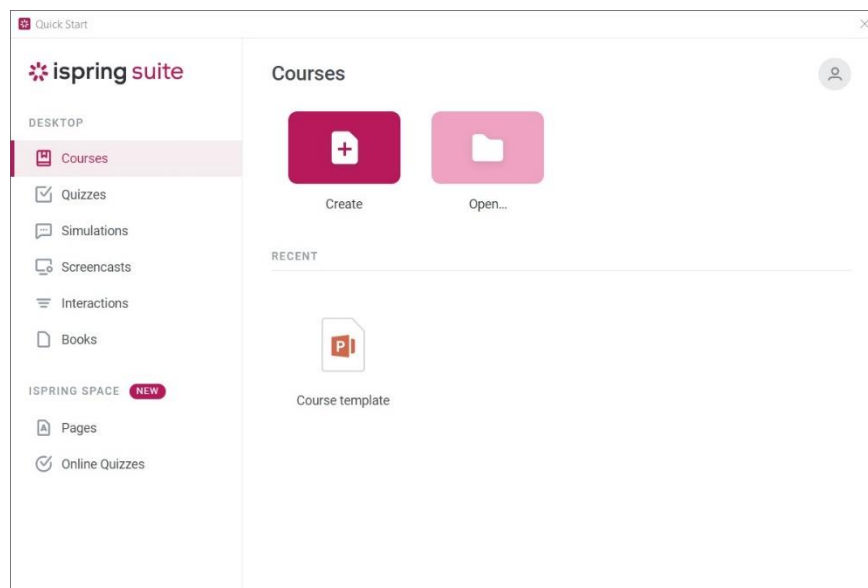


Slika 11 Izgled korisničkog sučelja - Moodle

(Izrada autorice rada)

4.2.3. iSpring Suite

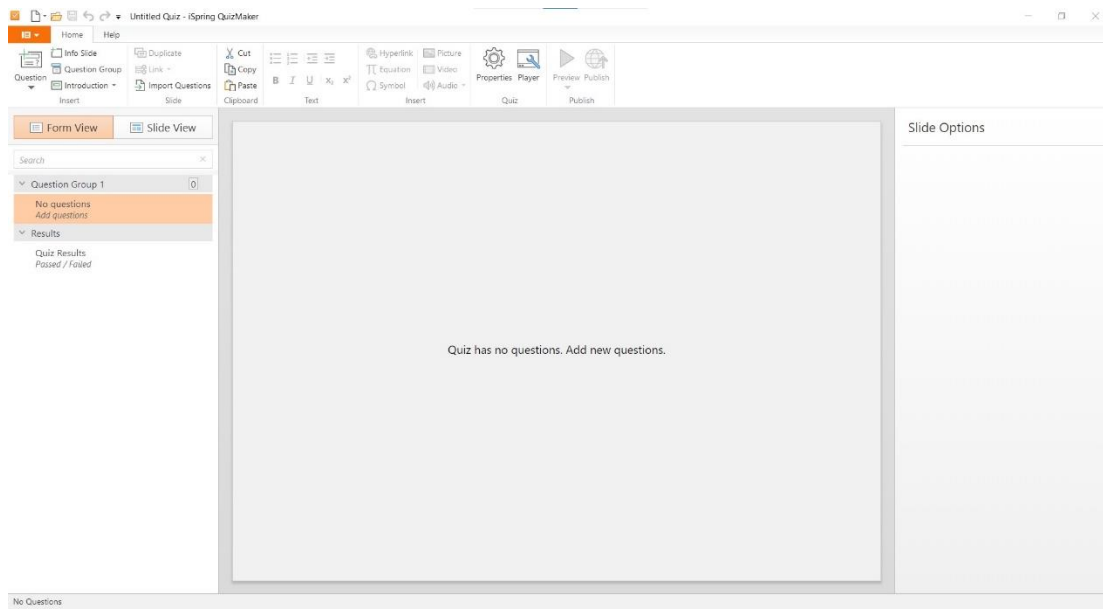
Brzi set alata za stvaranje online učenja, tako se može definirati iSpring Suite. Ovaj alat temeljen je na PowerPointu što omogućuje olakšano kreiranje interaktivnog sadržaja i kvizova posebice za osobe koje dobro poznaju PowerPoint alat. Jedan je od proizvoda svjetske tvrtke iSpring Solutions koja je osnovana 2001. godine. Tvrtka je usmjerena na pružanje rješenja u području e-učenja, a broji oko 59.000 korisnika diljem svijeta. Imaju i svoj LMS sustav poznatiji kao iSpring Learn, ali on ne nudi mogućnost kreiranja sadržaja i kvizova, nego služi za upis tečajeva i upravljanje sadržajem. Zato postoji iSpring Suite gdje se sadržaj izrađuje, dok se na iSpring Learn on samo učitava. Odabirom na iSpring Suite pruža se mogućnost besplatno skinuti alat na svoje računalo, ali samo na probno razdoblje od 30 dana (*iSpring, url*). Nakon instalacije otvara se početni zaslon alata (Slika 12).



Slika 12 Početni zaslon - iSpring Suite

(Izrada autorice rada)

Iz slike je vidljivo kako korisnik može kreirati tečajeve, kvizove, simulacije, interakcije i knjige. S obzirom na to da će u daljnjem radu fokus biti na procjeni znanja, sljedeća slika prikazuje izgled korisničkog sučelja iSpring QuizMaker alata koji je dio iSpring Suite rješenja.

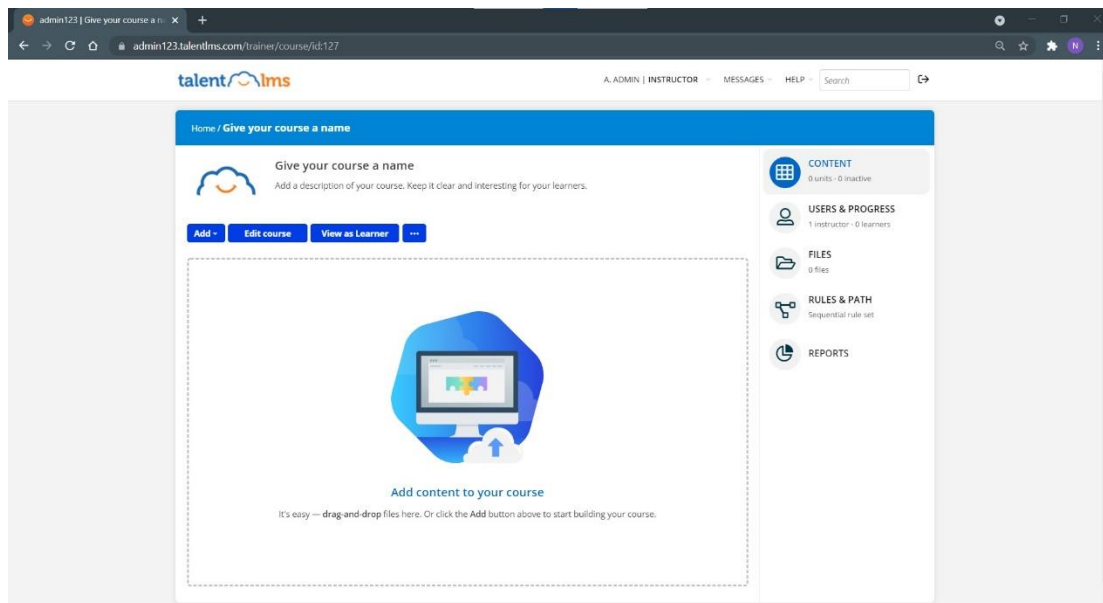


Slika 13 Izgled korisničkog sučelja - iSpring QuizMaker

(Izrada autorice rada)

4.2.4. TalentLMS

Najmlađi od četiri odabrana alata upravo je TalentLMS koji je 2012. godine osnovala softverska tvrtka Epignosis, a danas broji već oko 4 milijuna korisnika. Omogućuje izradu online tečajeva uz mogućnost uvoza sadržaja različitih izvora i formata. Zahvaljujući jednostavnom korisničkom sučelju koje ne zahtijeva napredna tehnička znanja lako se služiti ovim alatom (Slika 14). TalentLMS nudi različite pakete kada je pretplata u pitanju, a besplatna verzija nudi do 5 korisnika, izradu 10 tečajeva i neograničenu podršku e-poštom, što je iznimno odlično. Dovoljno je registrirati se na stranici kako bi se moglo koristiti ovaj LMS sustav. Prilikom prijave uvijek se može pogledati kratak tečaj o samom načinu kako TalentLMS radi, iako sustav nudi bazu znanja kojoj se može uvijek pristupiti. U njoj se mogu pronaći sve potrebne informacije i instrukcije kako doći do nečega i kako nešto napraviti. Korisnička podrška je izvrsna, a sustav je dostupan gdje god se nalazili, u bilo koje vrijeme, pa čak i preko mobilne aplikacije (*TalentLMS, url*).



Slika 14 Izgled korisničkog sučelja - TalentLMS

(Izrada autorice rada)

5. PRIMJER UPOTREBE AUTORSKIH ALATA U SVRHU ONLINE PROCJENE ZNANJA IZ PROGRAMIRANJA

Nakon teorijskog dijela slijedi glavno poglavlje rada. Ono je posvećeno istraživanju mogućnosti prethodno odabranih sustava kada se želi provesti online procjena znanja iz programiranja. Zašto se za izradu praktičnog dijela ovog rada koriste upravo sustavi za upravljanje učenjem? Iz tog razloga, jer oni omogućuju kreiranje tečajeva odnosno pružaju mogućnost stvaranja provjera (testova) pomoću kojih se zapravo vrši procjena znanja polaznika na nekom tečaju.

Prije svega, potrebno je reći kako se tečaj definira kao kratka obuka za stjecanje znanja i umijeća iz nekog područja (*Hrvatska enciklopedija, 2021*). Danas pomoću online učenja ljudi imaju priliku usvajati nova znanja i usavršavati svoje vještine, a dolaskom pandemije i formalno obrazovanje provodi se putem interneta. Samim time je važnost online obrazovanja narasla i više nije toliko neobična pojava u društvu. Zahvaljujući napretku tehnologije danas postoji mnogo načina i izvora online učenja. Iako je veća vjerojatnost da će pohađanje tečaja odabrati početnik u nekom području, ovaj način online učenja nije nepoznat ni osobama koje žele proširiti već postojeće znanje i unaprijediti postojeće vještine. Stoga, danas na internetu postoji nebrojeno mnogo vrsta tečajeva iz raznih područja.

Autori na raspolaganju imaju mnogo sustava koji omogućuju kreiranje online tečajeva, a uz razne dodatne alate sadržaj za učenje mogu učiniti još interaktivnijim i privlačnijim. Isto vrijedi i za nastavnike koji sustave za upravljanje učenjem koriste kako bi provodili formalno obrazovanje učenika. Svako pohađanje tečaja i učenje podrazumijeva procjenu znanja. Razni sustavi imaju različite mogućnosti kada je provjera znanja u pitanju i uglavnom su oni dovoljni kada se radi o područjima gdje prevladava uglavnom teorija. Međutim, što kada se osobu želi ispitati programersko znanje koje zahtjeva pisanje izvornog koda kao odgovor na pitanje? Postoji li sustav koji omogućuje ovu značajku? Kako prilagoditi pitanja ako nemamo ovu mogućnost? Upravo je ovo svrha izrade ovog diplomskog rada.

5.1. Izazovi izrade online tečaja za programiranje

Online tečajevi programiranja uvijek su među najtraženijim tečajevima na tržištu, a razlog je taj što je programer uvijek pri vrhu najtraženijih i dobro plaćenih zanimanja u svijetu. Zbog toga se mnogi danas odlučuju za razvoj u ovom području. Bilo da je osoba početnik ili već iskusni programer koji želi savladati novi programski jezik, na internetu zaista postoji mnogo različitih vrsta tečajeva koji se tiču programiranja.

Svijet treba veliki broj stručnjaka za programiranje jer IT sektor neprestano raste i razvija se, ali postati programer težak je i zahtjevan proces. Nije dovoljno samo savladati vještinu programiranja već je potrebno usvojiti određeni način razmišljanja koji uvelike olakšava kreiranje programskih rješenja. Zbog složenosti ovog područja nije ni malo jednostavno izraditi online tečaj programiranja, posebice kada se radi o osnovama programiranja koje su potrebne početnicima. Neki od izazova koji se stavljaju pred autore tečajeva programiranja su:

- shvaćanje programske logike,
- svladavanje sintakse programskog jezika,
- održavanje motivacije polaznika i
- usvajanje „programerskog“ načina razmišljanja.

Prethodno navedeni izazovi zahtijevaju veliki angažman autora da početnik uspješno završi online tečaj. Za početak nužno je da se programska logika što bolje predstavi polazniku kako bi on shvatio samu srž programiranja. Svladavanje sintakse programskog jezika najmanje je izazovno od svega navedenog, jer pravi problemi započinju prilikom rješavanja konkretnih programskih zadataka. U tom trenutku, posebice za početnike u svijetu programiranja, nužno je održati razinu motivaciju, jer za očekivati je da neće sve ići iz prve kako treba, da će biti grešaka i problema. Mnogi odustanu, ali programiranje je vještina koja zahtijeva mnogo uloženog truda. Ono najteže je usaditi polaznicima „programerski“ način razmišljanja, ali do njega se dolazi dugotrajnim učenjem i vježbom.

Danas zaista postoje različiti LMS sustavi, autorski alati, dodaci itd., pomoću kojih se sadržaj online tečaja programiranja može olakšano prikazati polaznicima. To nije ni malo lagan posao, a na autorima tečajeva je da osmisle način na koji to napraviti.

Nakon kreiranja sadržaja za učenje autori tečajeva moraju osmisliti i online testove pomoću kojih će provjeriti znanje svojih polaznika. Mnogi sustavi za upravljanje učenjem nude mogućnost kreiranja različitih vrsta pitanja. Neki sustavi imaju zaista širok izbor dok neki pružaju samo osnovne oblike pitanja. Međutim, ono što svi autori žele kada se radi o usvajanju znanja iz područja programiranja je mogućnost kreiranja pitanja koja od polaznika zahtijevaju da samostalno napiše programski kod kao odgovor na njih. Ovo je jedan od najvećih izazova za autore online tečajeva programiranja, jer danas još uvijek ne postoji mnogo autorskih alata koji pružaju tu mogućnost.

Kako bi se istražilo ovo područje u sljedećem poglavlju prikazat će se struktura online testa koji će biti proveden kroz četiri odabrana sustava za upravljanje učenjem.

5.2. Struktura online testa: Osnove programiranja u Pythonu

Mnogi početnici koji kreću s učenjem programiranja biraju Python kao programski jezik za učenje osnova. Pomoću Pythona početnici mogu lakše savladati programsku logiku, jer sintaksa jezika omogućava pisanje vrlo preglednih programa. Mnogi stručnjaci preporučuju Python kao prvi programski jezik zbog njegove jednostavnosti, jer naučene osnove kasnije je lakše primijeniti u drugim programskim jezicima. S obzirom na to da se radi o jeziku opće namjene, učenjem Pythona osobi su otvorena različita područja programiranja.

Cilj ovog rada je istražiti postoji li mogućnost da prilikom izrade online testa kojim se želi provjeriti znanje polaznika online tečaja programiranja, kao odgovor na postavljeno pitanje polaznik piše izvorni kod. Kako bi se provjerila ova mogućnost u četiri odabrana LMS sustava prije svega potrebno je osmisliti potrebna pitanja. S obzirom na to da Python slovi kao najbolji izbor programskog jezika za početnike online test će se sastojati od tri pitanja koja se odnose na osnove programiranja u Pythonu (Slika 15).

Osnove programiranja u Pythonu



1 PITANJE

Napišite program u kojem su definirane varijable $x = 5$ i $y = 10$. Kreirajte varijablu z čija će vrijednost biti zbroj varijabli x i y . Ispišite vrijednost varijable z .

2 PITANJE

Kreirajte varijablu x čija je vrijednost 3. Napišite program koji provjerava parnost varijable x . Ukoliko je broj paran program ispisuje poruku: "Broj je paran", u suprotnom se ispisuje: "Broj je neparan".

3 PITANJE

Napišite funkciju *sumaBrojeva* koja vraća zbroj brojeva x i y . Pozovite funkciju s vrijednostima $x = 1$ i $y = 3$.

Slika 15 Struktura online testa: Osnove programiranja u Pythonu

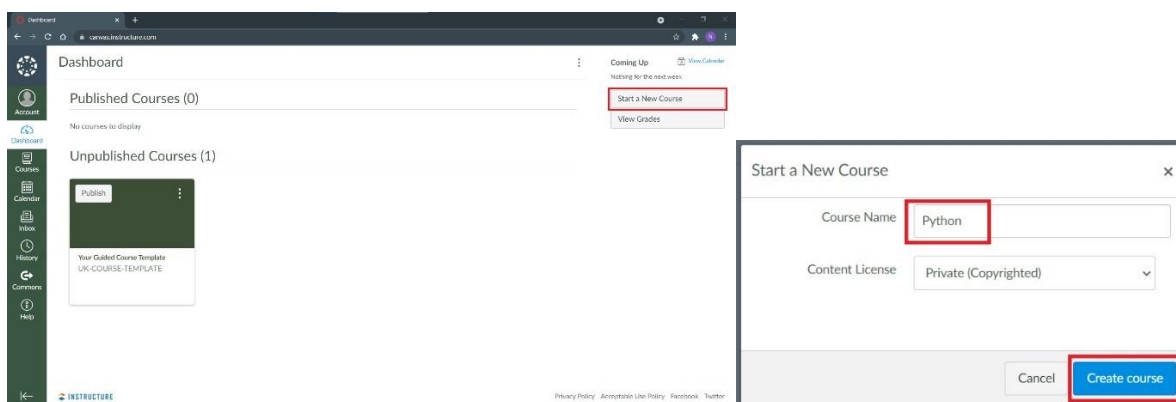
(Izrada autorice rada)

Prvo pitanje provjerava polaznikov rad s varijablama. U sljedećem pitanju postavljanjem uvjeta za parnost broja provjerava se logičko zaključivanje polaznika i usvajanje korištenja IF izjave. Treće pitanje traži od polaznika da kreiranjem jednostavne funkcije pokaže da je usvojio sintaksu pisanja funkcija.

S obzirom na to da Python ima više verzija, u radu će biti korištena verzija Python3. Nakon prikaza strukture online testa slijedi njegova provedba u prvom odabranom sustavu.

5.3. Kreiranje online testa u sustavu CANVAS

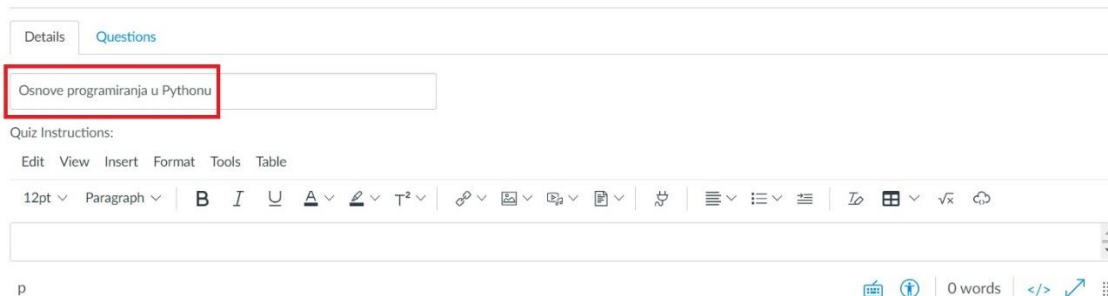
Nakon uspješne registracije, ako se želi pristupiti CANVAS-u dovoljno je otvoriti internetski preglednik, otići na stranicu i prijaviti se s valjanim podacima. Otvara se korisničko sučelje koje je prethodno prikazano u radu (Slika 10). Uz lijevi rub ekrana nalazi se izbornik, a na sredini je otvorena nadzorna ploča na kojoj se nalaze objavljeni i neobjavljeni tečajevi. S obzirom na to da nema kreiranih tečaja trenutno se samo prikazuje primjer tečaja od strane sustava. Kako bi se kreirao test potrebno je kreirati tečaj, a to se radi tako što se na početnoj stranici u desnom kutu odabere opcija *Start a New Course*. U prozor koji se otvori upisuje se željeni naziv tečaja *Python* i klikom na *Create course* tečaj je kreiran (Slika 16).



Slika 16 Kreiranje tečaja - CANVAS

(Izrada autorice rada)

Automatski se otvara sučelje kreiranog tečaja ili mu se može pristupiti preko glavnog izbornika klikom na opciju *Courses*. Kada se otvori tečaj *Python* u izborniku s lijeve strane bira se opcija *Quizzes*. Time se otvara popis testova, a u gornjem desnom kutu klikom na gumb *+Quiz* kreira se novi test. Otvara se prozor *Details* u kojem se unose podaci testa (Slika 17). Za početak se upisuje naziv testa - *Osnove programiranja u Pythonu*. Iza toga u uređivaču bogatog sadržaja može se detaljnije predstaviti test, a nakon toga slijede ostale postavke kviza od kojih se opcija *višestrukog rješavanja* omogućila radi pokušaja rješavanja našeg izazova. Kada su se upisali željeni podaci, klik na gumb *Save* na kraju forme pohranjuje test.

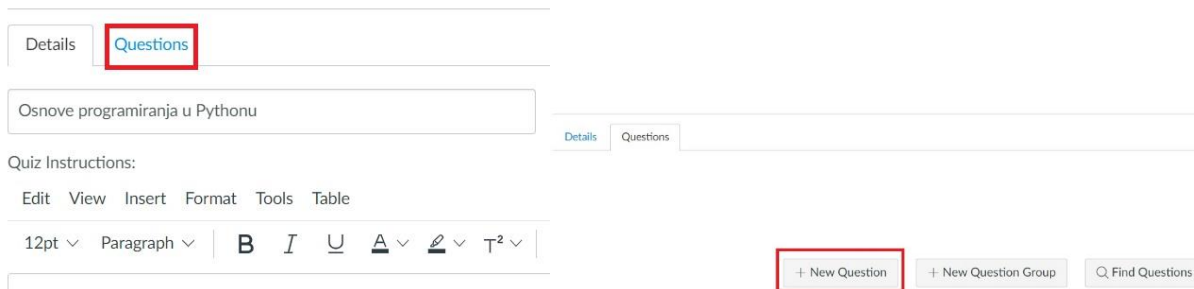


Slika 17 Podaci testa - CANVAS

(Izrada autorice rada)

S obzirom na to da CANVAS nema mogućnost ugradnje dodatka koji bi omogućio da na kreirano programersko pitanje polaznik odgovara pisanjem izvornog koda, autori tečajeva u ovakvim situacijama moraju biti snalažljivi s vrstama pitanja koje sustav nudi. Upravo na ovaj način budu kreirana pitanja na većini online tečajeva programiranja. Cilj je da autori budu kreativni kako bi pitanja zadržala određenu razinu zahtjevnosti za polaznika.

Otvaranjem kreiranog tečaja *Python*, iz izbornika se bira opcija *Quizzes* i otvara se kreirani test *Osnove programiranja u Pythonu*. U desnom kutu klikom na opciju *Edit*, otvara se prozor s podacima testa. Ovaj puta bira se kartica *Questions*, a zatim *+New Question* (Slika 18).



Slika 18 Dodavanje pitanja - CANVAS

(Izrada autorice rada)

Otvara se prozor gdje se unosi naziv pitanja, vrsta pitanja, tekst pitanja i odgovor (Slika 19). Naravno, ovisno o vrsti pitanja koje se odabere mogućnosti kreiranja odgovora se razlikuju. U nastavku slijedi provedba online testa u sustavu CANVAS.

Details Questions

Question True/False

Enter your question text, then select if True or False is the correct answer.

Question:

Edit View Insert Format Tools Table

12pt Paragraph B I U A

True

False

Cancel Update Question

Slika 19 Kreiranje pitanja - CANVAS

(Izrada autorice rada)

5.3.1. Provedba online testa

Nakon što se prikazalo kako se kreira test i pitanje u sustavu CANVAS, u ovom dijelu može se provesti online test koji se osmislio u svrhu izrade ovog rada. Kreiranje svakog pitanja prikazat će se kroz tri koraka: **zadatak**, **opis problema** i **rješenje**. Zadatak će sadržavati sliku koja prikazuje postavke kreiranja pitanja. Opis problema za sva pitanja je isti, a on je **pisanje izvornog koda kao odgovor na pitanje**, stoga se opis problema neće ponavljati za svako pitanje. Dok rješenje prikazuje uspješno kreirano i provedeno pitanje u sustavu CANVAS.

Vrsta pitanja koja će se koristiti je *Fill In Multiple Blanks*, odnosno *ispunite više praznina*. Na ovaj način polaznik samostalno ispisuje odgovor bez ponuđenih riječi.

Zadatak - prvo pitanje

Rad s varijablama Fill In Multiple Blanks pts: 1

Enter your question, specifying where each blank should go. Then define the possible correct answer for each blank. Students must type correct answers into text boxes at each blank.

Question:
In the box below, every place you want to show an answer box, type a reference word (no spaces) surrounded by brackets (i.e. "Roses are [color1], violets are [color2]")

Edit View Insert Format Tools Table

10pt Paragraph B I U A T² | | | | | | | | | |

Napišite program u kojem su definirane varijable $x = 5$ i $y = 10$. Kreirajte varijablu z čija će vrijednost biti zbroj varijabli x i y . Ispišite vrijednost varijable z .

[varijabla_x]
[varijabla_y]
[varijabla_z]
[ispis]

p ▶ span ▶ strong ▶ span 34 words </> ↗ ⋮

Answers:
Show Possible Answers for varijabla_x

Possible Answer: x=5
...
Possible Answer: x = 5
...

Slika 20 Kreiranje prvog pitanja - CANVAS

(Izrada autorice rada)

Rješenje

Napišite program u kojem su definirane varijable $x = 5$ i $y = 10$. Kreirajte varijablu z čija će vrijednost biti zbroj varijabli x i y . Ispišite vrijednost varijable z .

x=5
y=10
z = x+y
print(z)

Answer 1:
Correct! x=5
Correct Answer x = 5

Answer 2:
Correct! y=10
Correct Answer y = 10

Answer 3:
Correct! z = x+y
Correct Answer z=x+y
Correct Answer z = x + y

Answer 4:
Correct! print(z)
Correct Answer print (z)

Slika 21 Rješenje prvog pitanja - CANVAS

(Izrada autorice rada)

Zadatak - drugo pitanje

IF izjava Fill In Multiple Blanks pts: 1

Enter your question, specifying where each blank should go. Then define the possible correct answer for each blank. Students must type correct answers into text boxes at each blank.

Question:
In the box below, every place you want to show an answer box, type a reference word (no spaces) surrounded by brackets (i.e. "Roses are [color1], violets are [color2]")

Edit View Insert Format Tools Table

10pt Paragraph B I U A T² | | | | | | | | | |

Kreirajte varijablu x čija je vrijednost 3. Napišite program koji provjerava parnost varijable x. Ukoliko je broj paran program ispisuje poruku: "Broj je paran", u suprotnom se ispisuje: "Broj je neparan".

[varijabla_x]
[if_uvjet]
[ispis1]
[drugo]
[ispis2]

p ▶ span 36 words </>

Answers:
Show Possible Answers for drugo

Possible Answer: else:

Slika 22 Kreiranje drugog pitanja - CANVAS

(Izrada autorice rada)

Rješenje

Kreirajte varijablu x čija je vrijednost 3. Napišite program koji provjerava parnost varijable x. Ukoliko je broj paran program ispisuje poruku: "Broj je paran", u suprotnom se ispisuje: "Broj je neparan".

x=3

if x%2 == 0:

print("Broj je paran")

else:

print("Broj je neparan")

Answer 1:
Correct! x=3
Correct Answer x = 3

Answer 2:
Correct! if x%2 == 0:
Correct Answer if x%2==0:

Answer 3:
Correct! print("Broj je paran")
Correct Answer print ("Broj je paran")

Answer 4:
Correct! else:

Answer 5:
Correct! print("Broj je neparan")
Correct Answer print ("Broj je neparan")

Slika 23 Rješenje drugog pitanja - CANVAS

(Izrada autorice rada)

Slika u nastavku prikazuje cjelokupni izgled testa. Ova vrsta pitanja *popuni praznine* najbolja je moguća vrsta pitanja u sustavu CANVAS kada se želi provjeriti programersko znanje, jer na ovaj način polaznik samostalno piše kod u obliku teksta kao odgovor na pitanje. Velika mana ovog načina ispitivanja je što autor testa mora pokriti sve mogućnosti odgovora za svako pitanje. Primjerice: ako autor pod točan odgovor upiše $x=5$, a polaznik prilikom odgovaranja na pitanje napiše $x = 5$, sustav će odgovor registrirati kao netočan. U sustavu CANVAS moguće je kreirati pitanje gdje polaznik samostalno upisuje kod kao odgovor i automatski dobiva povratni odgovor o ispravnosti rješenja, ali to od autora testa zahtjeva izniman trud i kreativnost.

The image shows three separate question boxes from the CANVAS system. Each box has a title bar with the question number and '1 pts'. The questions are as follows:

- Question 1:** 'Napišite program u kojem su definirane varijable $x = 5$ i $y = 10$. Kreirajte varijablu z čija će vrijednost biti zbroj varijabli x i y . Ispišite vrijednost varijable z .' Below the text are four empty text input fields.
- Question 2:** 'Kreirajte varijablu x čija je vrijednost 3. Napišite program koji provjerava parnost varijable x . Ukoliko je broj paran program ispisuje poruku: "Broj je paran", u suprotnom se ispisuje: "Broj je neparan".' Below the text are five empty text input fields.
- Question 3:** 'Napišite funkciju `sumaBojevo` koja vraća zbroj brojeva x i y . Pozovite funkciju s vrijednostima $x = 1$ i $y = 3$.' Below the text are three empty text input fields.

Slika 26 Izgled testa - CANVAS

(Izrada autorice rada)

5.4. Kreiranje online testa u sustavu Moodle

Nakon instalacije sustava Moodle na vlastito računalo isti se pokreće klikom na ikonu "Start Moodle". Time se pokreće Apache server i MySQL baza podataka koji zajedno sa programskim jezikom PHP-om omogućuju rad sustava. U tražilicu internetskog preglednika upisuje se *localhost* i pritiskom tipke enter otvara se Moodle. Ako je potrebno korisnik se mora prijaviti s podacima koje je odabrao prilikom instalacije sustava i zatim se otvara korisničko sučelje koje je prethodno prikazano u radu (Slika 11). Na početnoj stranici u gornjem desnom kutu nalaze se podaci prijavljenog korisnika. Prilikom prijave korisnik ima ulogu administratora, a ulogu prema potrebi može promijeniti (nastavnik, student, gost itd.). U suprotnom kutu nalazi se odabrani jezik (hrvatski) na kojem se prikazuje sustav uz mogućnost promjene na engleski. Na samoj sredini nalazi se popis kolegija, s desne strane nalaze se zanimljive informacije (vremenska crta, korisnici online, kalendar) i s lijeve strane prikazuje se glavni izbornik.

Ono bitno je kreiranje testa. Kako bi se mogla izraditi ova aktivnost najprije se mora kreirati kolegij (*engl. course*). U izborniku s lijeve strane odabire se opcija *Administracija poslužitelja*. Otvara se izbornik u kojem je prvo potrebno otvoriti karticu *E-kolegiji*, zatim izabrati opciju *Dodajte novi e-kolegij* (Slika 27).



Slika 27 Kreiranje e-kolegija - Moodle

(Izrada autorice rada)

Nakon toga unose se podaci novog e-kolegija. Naziv, kratica i kategorija e-kolegija su obvezna polja, a opis i slika dodali su se proizvoljno (Slika 28).

Postoje još dodatne opcije za uređenje kolegija kao što su oblik e-kolegija, izgled, praćenje dovršenosti itd. Klikom na gumb *Pohrani i prikaži* na kraju forme uspješno je kreiran kolegij *Python*.

Dodajte novi e-kolegij

▼ Opće postavke

Naziv e-kolegija

Kratice e-kolegija

Kategorija e-kolegija

Pretraži

Vidljivost e-kolegija

E-kolegij počinje

Datum završetka e-kolegija Omogući

ID broj e-kolegija

▼ Opis

Sažetak e-kolegija

Tečaj programiranja u Pythonu za početnike.

Slika 28 Podaci e-kolegija - Moodle

(Izrada autorice rada)

U glavnom izborniku bira se opcija *Naslovnica* i klikom na kreirani kolegij *Python* otvaramo isti. U desnom kutu klik na gumb *Omogući izmjene* nudi mogućnost *dodavanja aktivnosti ili resursa*. Od svega ponuđenog bira se *test* (Slika 29).

Onemogući izmjene

Uredi

Uredi

+ Dodajte aktivnost ili resurs

+ Dodaj teme / sekcije

Dodajte aktivnost ili resurs

Pretraži

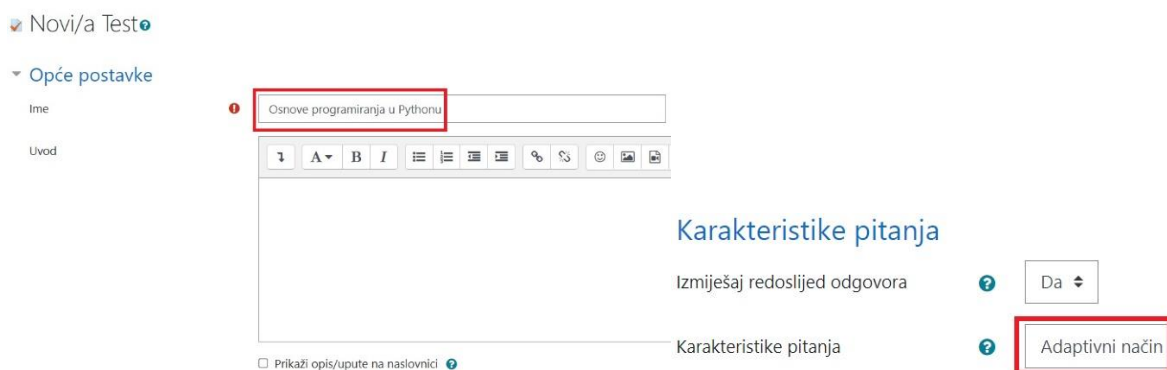
Sve Aktivnosti Resursi

Anketa	Baza podataka	Čuži	Datoteka	Forum	HSP
IMS paket	Knjiga	Lekcija	Mapa	Natpis	Odabir
Poveznica	Radionica	Iječnik	SCORM paket	Stranica	Test
Uprtnik	Venjski alat	Wii	Zadaca		

Slika 29 Dodavanje aktivnosti (test) - Moodle

(Izrada autorice rada)

Otvara se prozor gdje se unosi podaci testa. Polje s imenom je obvezno, a nakon opisa koji se može napisati postoji niz drugih opcija kojima se test može unaprijediti. Ono što je bitno je pod *Karakteristike pitanja* odabrati opciju *Adaptivni način*. On omogućuje višestruko odgovaranje na pitanje uz dodjeljivanje kaznenih bodova (Slika 30). Klikom na gumb *Pohrani i prikaži test* je kreiran.



Slika 30 Podaci testa - Moodle

(Izrada autorice rada)

Nakon kreiranja testa u isti je potrebno dodati pitanja koja su prethodno prikazana u poglavlju 5.2. ovog rada. Slijedi provedba online testa u sustavu Moodle.

5.4.1. Provedba online testa

Sustav Moodle ima veliki broj dodataka koji se mogu ugraditi i time se može unaprijediti kreiranje sadržaja za učenje. Jedan od dodataka je **CodeRunner**. On omogućuje kreiranje programerskih pitanja u kojima polaznik kao odgovor piše izvorni kod u nekom programskom jeziku, a ocjenjuje se njegovim pokretanjem. Sa službene stranice u direktoriju dodataka pronađe se CodeRunner i ZIP datoteka se skine na računalo (*Moodle, url*). Kako bi se instalirao ovaj dodatak u Moodle, u glavnom izborniku sustava bira se opcija *Administracija poslužitelja*. Iz izbornika koji se otvori prvo se bira kartica *Dodaci*, a zatim opcija *Instalacija dodataka* (Slika 31).

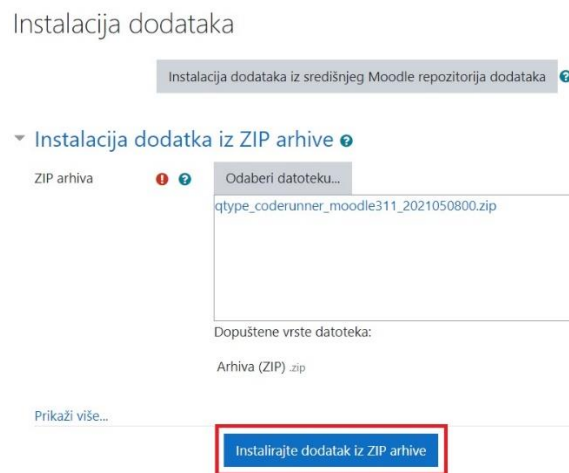
Administracija poslužitelja



Slika 31 Instalacija dodataka - Moodle

(Izrada autorice rada)

Otvara se prozor (Slika 32) gdje je potrebno učitati ZIP datoteku dodatka kojeg se želi instalirati, a koja se prethodno skinula na računalo. Klikom na gumb *Instalirajte dodatak iz ZIP arhive* potrebno je pratiti korake instalacije. Nakon podešavanja i pohrane postavki, dodatak CodeRunner je uspješno instaliran i spreman za korištenje.

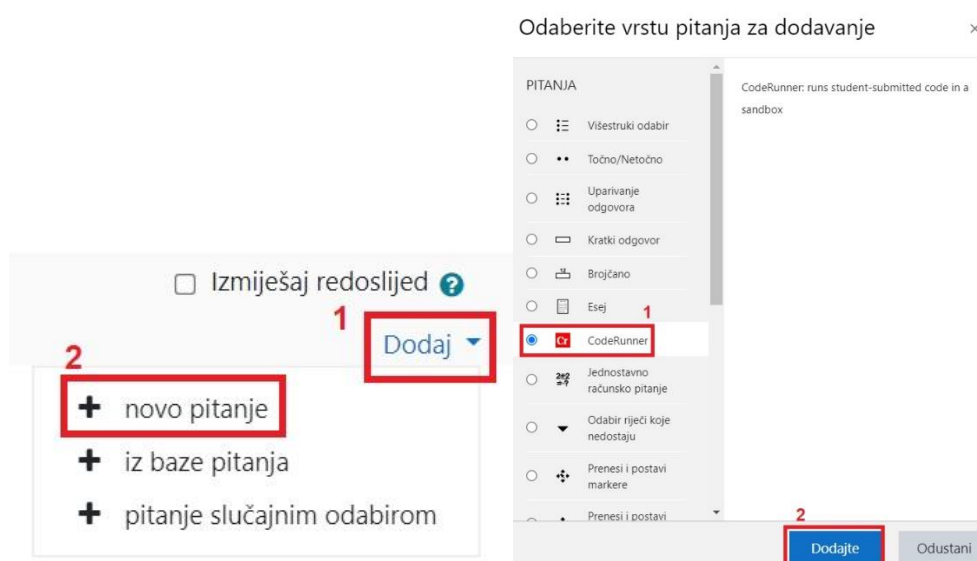


Slika 32 Instalacija dodatka CodeRunner - Moodle

(Izrada autorice rada)

Otvara se kolegij *Python* zatim test koji je prethodno kreiran - *Osnove programiranja u Pythonu*. Klikom na gumb *Uredi test* otvara se mogućnost dodavanja pitanja. Prvo se bira opcija *Dodaj*, zatim se iz padajućeg izbornika odabire *ново pitanje*. Otvara se prozor gdje se odabire vrsta pitanja koja se želi dodati. U popisu pitanja nalazi se i CodeRunner i to je upravo ono što je potrebno za rješavanje izazova.

Označi se CodeRunner i klikom na gumb *Dodajte* otvara se prozor u kojem se kreira željeno pitanje (Slika 33).



Slika 33 Dodavanje pitanja - CodeRunner - Moodle

(Izrada autorice rada)

Kreiranje svakog pitanja prikazat će se kroz tri koraka: **zadatak**, **opis problema** i **rješenje**. Zadatak će sadržavati tablicu koja prikazuje postavke kreiranja pitanja. Opis problema za sva pitanja je isti, a on je **pisanje izvornog koda kao odgovor na pitanje**, stoga se opis problema neće ponavljati za svako pitanje. Dok rješenje prikazuje uspješno kreirano i provedeno pitanje u sustavu Moodle.

Zadatak - prvo pitanje

Tablica 6 Postavke kreiranja prvog pitanja - CodeRunner

ZNAČAJKA	OPIS
Question type	python3
Answer box	Rows 5
Marking	<input checked="" type="checkbox"/> All-or-nothing grading Penalty regime: 0, 10, 20, ...
Naziv pitanja	Rad s varijablama
Tekst pitanja	Napišite program u kojem su definirane varijable $x = 5$ i $y = 10$. Kreirajte varijablu z čija će vrijednost biti zbroj varijabli x i y . Ispišite vrijednost varijable z .
Odgovor	<pre>1 x=5 2 y=10 3 z=x+y 4 5 print(z)</pre>
Test case 1	z
Expected output	15

(Izrada autorice rada)

Rješenje

Pitanje 1
Točno
Broj bodova:
1.00 od 1.00
Označi
pitanje

Napišite program u kojem su definirane varijable $x = 5$ i $y = 10$. Kreirajte varijablu z čija će vrijednost biti zbroj varijabli x i y . Ispišite vrijednost varijable z .

Answer: (penalty regime: 0, 10, 20, ... %)

```
1 x=5
2 y=10
3 z=x+y
4 print(z)
5
```

Provjeri

Test	Expected	Got
✓ z	15	15 ✓

Run using the University of Canterbury's Jobe server. This is for initial testing only. Please set up your own Jobe server as soon as possible. See here.

Passed all tests! ✓

Točno
Broj bodova za ovaj odgovor: 1.00/1.00.

Slika 34 Rješenje prvog pitanja - Moodle

(Izrada autorice rada)

Zadatak - drugo pitanje

Tablica 7 Postavke kreiranja drugog pitanja - CodeRunner

ZNAČAJKA	OPIS
Question type	python3
Answer box	Rows 5
Marking	<input checked="" type="checkbox"/> All-or-nothing grading Penalty regime: 0, 10, 20, ...
Naziv pitanja	IF izjava
Tekst pitanja	Kreirajte varijablu x čija je vrijednost 3. Napišite program koji provjerava parnost varijable x. Ukoliko je broj paran program ispisuje poruku: "Broj je paran", u suprotnom se ispisuje: "Broj je neparan".
Odgovor	<pre> 1 x=3 2 3 if x%2 == 0: 4 print("Broj je paran") 5 else: 6 print("Broj je neparan") </pre>
Test case 1	x
Expected output	Broj je neparan

(Izrada autorice rada)

Rješenje

Pitanje 2

Točno

Broj bodova: 1.00 od 1.00

Označi pitanje

Kreirajte varijablu x čija je vrijednost 3. Napišite program koji provjerava parnost varijable x. Ukoliko je broj paran program ispisuje poruku: "Broj je paran", u suprotnom se ispisuje: "Broj je neparan".

Answer: (penalty regime: 0, 10, 20, ... %)

```

1 x=3
2
3 if x%2==0:
4     print("Broj je paran")
5 else:
6     print("Broj je neparan")

```

Provjeri

Test	Expected	Got	
✓	x	Broj je neparan	Broj je neparan ✓

Run using the University of Canterbury's Jobe server. This is for initial testing only. Please set up your own Jobe server as soon as possible. See here.

Passed all tests! ✓

Točno

Broj bodova za ovaj odgovor: 1.00/1.00.

Slika 35 Rješenje drugog pitanja - Moodle

(Izrada autorice rada)

54

Zadatak - treće pitanje

Tablica 8 Postavke kreiranja trećeg pitanja - CodeRunner

ZNAČAJKA	OPIS
Question type	python3
Answer box	Rows 5
Marking	<input checked="" type="checkbox"/> All-or-nothing grading Penalty regime: 0, 10, 20, ...
Naziv pitanja	Funkcija
Tekst pitanja	Napišite funkciju <i>sumaBrojeva</i> koja vraća zbroj brojeva x i y. Pozovite funkciju s vrijednostima x = 1 i y = 3.
Odgovor	<pre>1 def sumaBrojeva(x,y): 2 return x+y 3 4 sumaBrojeva(1,3)</pre>
Test case 1	sumaBrojeva(1,3)
Expected output	4

(Izrada autorice rada)

Rješenje

Pitanje 3
Točno
Broj bodova:
1.00 od 1.00
Označi
pitanje

Napišite funkciju *sumaBrojeva* koja vraća zbroj brojeva x i y. Pozovite funkciju s vrijednostima x = 1 i y = 3.

Answer: (penalty regime: 0, 10, 20, ... %)

```
1 def sumaBrojeva(x,y):
2     return x+y
3 sumaBrojeva(1,3)
4
```

Provjeri

Test
✓ sumaBrojeva(1,3)

Run using the University of Canterbury's Jobe server. This is for initial testing only. Please set up your own Jobe server as soon as possible. See here.

Passed all tests! ✓

Točno
Broj bodova za ovaj odgovor: 1.00/1.00.

Slika 36 Rješenje trećeg pitanja - Moodle

(Izrada autorice rada)

Sljedeća slika prikazuje izgled cijelog testa. CodeRunner zasta nudi mogućnost da polaznik kao odgovor na pitanje upisuje izvorni kod koji se automatski provjerava njegovim pokretanjem. Kada unese odgovor klikom na gumb *Provjeri* vraća se povratna informacija o točnosti odgovora. Postoji mogućnost kreiranja kaznenih bodova tako što se prilikom svake provjere (izuzev prve) smanjuje broj bodova za 10% od ukupnog broja bodova za pitanje. Autor pitanja može ponuditi i primjer polazniku ako se radi o kompliciranijem zadatku, a sve to i još mnogo toga moguće je postaviti prilikom kreiranja pitanja.

The screenshot displays three Moodle CodeRunner questions, each with a metadata sidebar on the left and a main question area on the right. The sidebar for each question includes: 'Pitanje' (Question), 'Nije dovršeno' (Not completed), 'Broj bodova od 1.00' (Number of points out of 1.00), and 'Označi pitanje' (Mark question). The main area contains the question text, an 'Answer:' field with a penalty regime '(penalty regime: 0, 10, 20, ... %)', a code editor with a line number '1', and a 'Provjeri' (Check) button. A gear icon is visible in the top right corner of the first question's area.

Pitanje 1
Napišite program u kojem su definirane varijable $x = 5$ i $y = 10$. Kreirajte varijablu z čija će vrijednost biti zbroj varijabli x i y . Ispišite vrijednost varijable z .

Answer: (penalty regime: 0, 10, 20, ... %)

Pitanje 2
Kreirajte varijablu x čija je vrijednost 3. Napišite program koji provjerava parnost varijable x . Ukoliko je broj paran program ispisuje poruku: "Broj je paran", u suprotnom se ispisuje: "Broj je neparan".

Answer: (penalty regime: 0, 10, 20, ... %)

Pitanje 3
Napišite funkciju *sumaBrojeva* koja vraća zbroj brojeva x i y . Pozovite funkciju s vrijednostima $x = 1$ i $y = 3$.

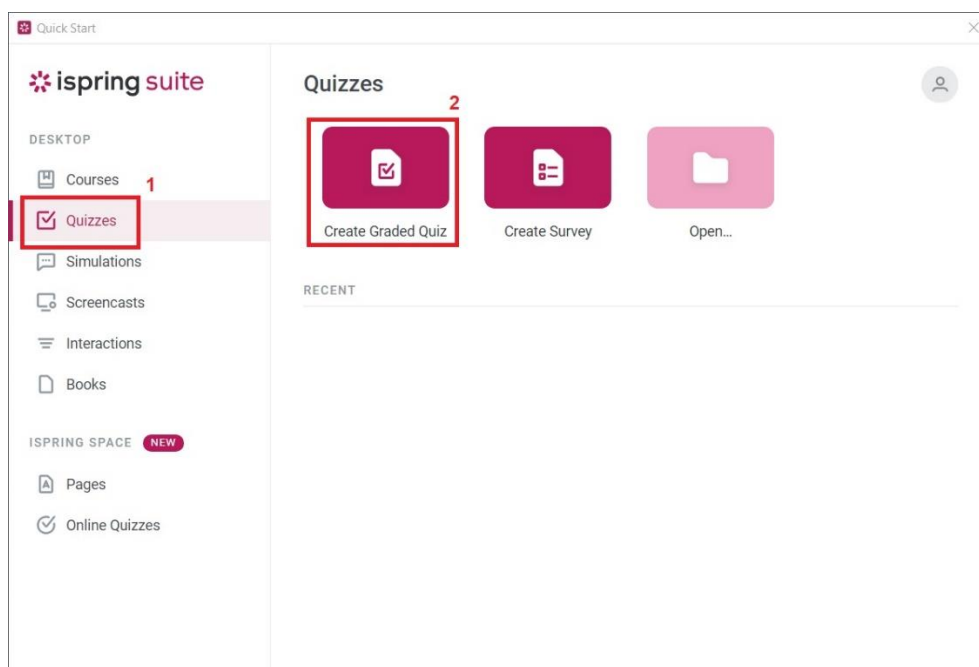
Answer: (penalty regime: 0, 10, 20, ... %)

Slika 37 Izgled testa - Moodle

(Izrada autorice rada)

5.5. Kreiranje online testa u sustavu iSpring Suite

Ovaj alat služi za kreiranje raznih sadržaja za e-učenje koji se onda kasnije mogu koristiti i dodati na sustave za upravljanje učenjem (iSpring Learn i dr.). Nakon instalacije stvara se ikona aplikacije “iSpring Suite 10“ na radnoj površini računala. Klikom na nju otvara se početni zaslon alata, prikazan prije u radu (Slika 12). Ovo je jedini alat koji ne zahtijeva kreiranje tečaja ako se želi kreirati samo test, jer postoje odvojeni moduli pomoću kojih se kreira tečaj, a pomoću kojih test. Stoga se odabire opcija *Quizzes* čime se nudi kreiranje ocjenjivačkog testa ili ankete. Odabire se opcija *Create Graded Quiz* (Slika 38). Otvara se korisničko sučelje *iSpring QuizMaker* u kojem će se izraditi online test.

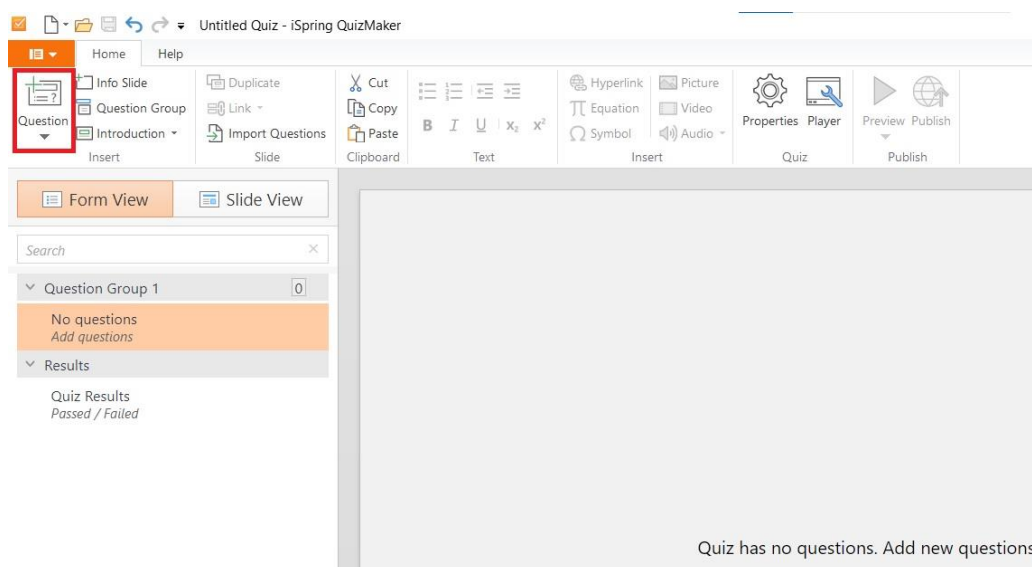


Slika 38 Kreiranje testa - iSpring Suite

(Izrada autorice rada)

Sučelje je zaista slično alatu Powerpoint. S obzirom na to da alat služi isključivo za kreiranje testova iznimno je lako koristiti se njime. Korisnička podrška tvrtke iSpring Solutions zaista je učinkovita, jer se putem online razgovora s agentom korisničke službe uspješno saznalo kako ovaj alat nema mogućnost ugradnje dodatka koji bi riješio najveći izazov autora online tečajeva programiranja. Autorima preostaje koristiti vrste pitanja koje alat nudi.

Kako bi se kreiralo pitanje iz izbornika je potrebno odabrati prvu opciju *Question* (Slika 39). Time se otvara padajući izbornik koji nudi vrste pitanja koja se mogu kreirati.

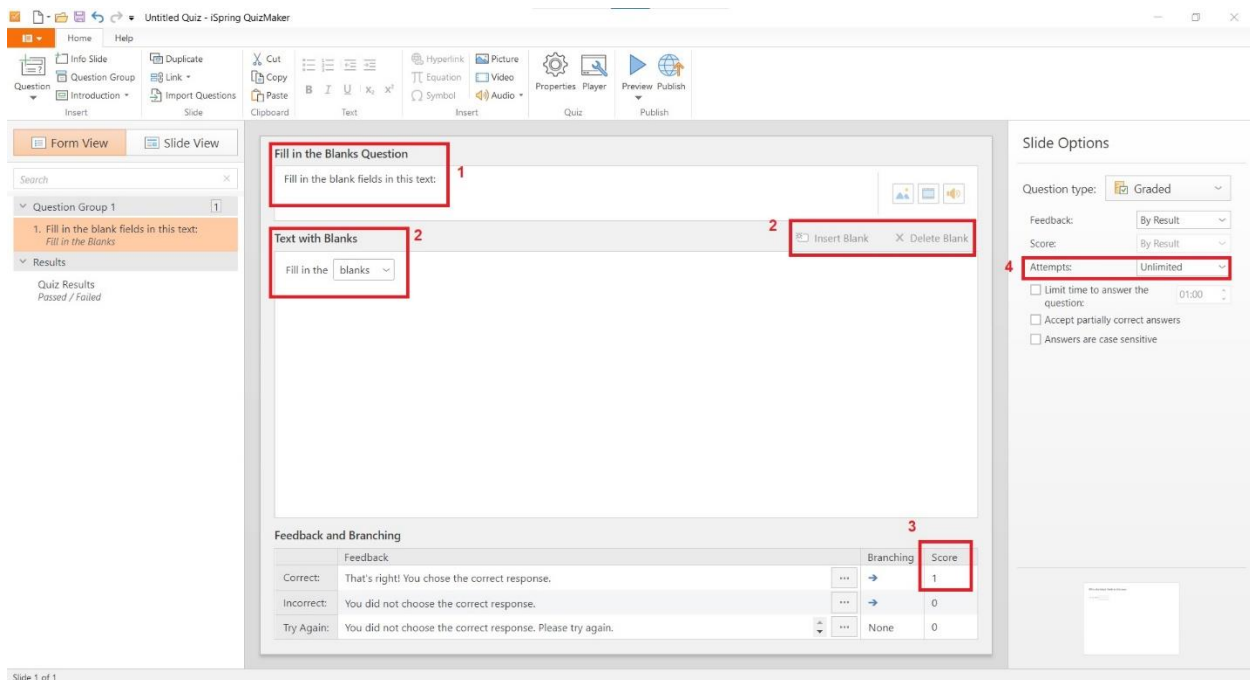


Slika 39 Dodavanje pitanja - iSpring Suite

(Izrada autorice rada)

Kao i u prvom sustavu, bira se opcija pitanja *Fill in the Blanks* koja pruža opciju da polaznik popuni jednu ili više praznina. Time se omogućuje da polaznik samostalno piše kod u obliku teksta kao odgovor. Provjera točnosti odgovora opet ovisi o kreatoru pitanja koji mora uzeti u obzir sve moguće načine pisanja odgovora, jer “ $x=5$ ” i “ $x = 5$ ” sustav razlikuje. Sučelje kreiranja ove vrste pitanja prikazano je u nastavku (Slika 40). Na slici su označeni dijelovi kreiranja pitanja:

- 1) tekst pitanja,
- 2) kreiranje odgovora,
- 3) bodovanje pitanja i
- 4) broj pokušaja rješavanja testa.



Slika 40 Kreiranje pitanja - iSpring Suite

(Izrada autorice rada)



5.5.1. Provedba online testa

Kao i do sada, kreiranje svakog pitanja prikazat će se kroz tri koraka: **zadatak**, **opis problema** i **rješenje**. Zadatak će sadržavati sliku koja prikazuje postavke kreiranja pitanja. Opis problema za sva pitanja je isti, a on je **pisanje izvornog koda kao odgovor na pitanje**, stoga se opis problema neće ponavljati za svako pitanje. Dok rješenje prikazuje uspješno kreirano i provedeno pitanje u alatu iSpring Suite.

Zadatak - prvo pitanje

Fill in the Blanks Question

Napišite program u kojem su definirane varijable $x = 5$ i $y = 10$. Kreirajte varijablu z čija će vrijednost biti zbroj varijabli x i y . Ispišite vrijednost varijable z .

Text with Blanks Insert Blank Delete Blank


Slika 41 Kreiranje prvog pitanja - iSpring Suite


(Izrada autorice rada)

Rješenje

Question List Question 1 of 3 Your Score: 1 of 3

Napišite program u kojem su definirane varijable $x = 5$ i $y = 10$. Kreirajte varijablu z čija će vrijednost biti zbroj varijabli x i y . Ispišite vrijednost varijable z .

 Created with iSpring Suite evaluation version [Learn more](#)



Correct

That's right! You chose the correct response.

[CONTINUE >](#)



Slika 42 Rješenje prvog pitanja - iSpring Suite

(Izrada autorice rada)

Zadatak - drugo pitanje

Fill in the Blanks Question

Kreirajte varijablu x čija je vrijednost 3. Napišite program koji provjerava parnost varijable x . Ukoliko je broj paran program ispisuje poruku: "Broj je paran", u suprotnom se ispisuje: "Broj je neparan".

Text with Blanks ✎ Insert Blank ✕ Delete Blank

```
x=3  
  
if x%2==0:  
    print("Broj je paran")  
else:  
    print("Broj je neparan")
```


Slika 43 Kreiranje drugog pitanja - iSpring Suite


(Izrada autorice rada)

Rješenje


Question List Question 2 of 3 | Your Score: 2 of 3

Kreirajte varijablu x čija je vrijednost 3. Napišite program koji provjerava parnost varijable x . Ukoliko je broj paran program ispisuje poruku: "Broj je paran", u suprotnom se ispisuje: "Broj je neparan".

 Created with iSpring Suite evaluation version [Learn more](#)



```
x=3  
  
if x%2 == 0:  
    print("Broj je paran")  
else:  
    print("Broj je neparan")
```

Correct 

That's right! You chose the correct response.

CONTINUE >

Slika 44 Rješenje drugog pitanja - iSpring Suite

(Izrada autorice rada)

Zadatak - treće pitanje

Fill in the Blanks Question

Napišite funkciju `sumaBrojeva` koja vraća zbroj brojeva `x` i `y`. Pozovite funkciju s vrijednostima `x = 1` i `y = 3`.

Text with Blanks

def sumaBrojeva(x,y):
 return x+y

sumaBrojeva(1,3)

Insert Blank Delete Blank

Slika 45 Kreiranje trećeg pitanja - iSpring Suite

(Izrada autorice rada)

Rješenje

Question List

Question 3 of 3 | Your Score: 3 of 3

Napišite funkciju `sumaBrojeva` koja vraća zbroj brojeva `x` i `y`. Pozovite funkciju s vrijednostima `x = 1` i `y = 3`.

def sumaBrojeva(x,y):
 return x+y

sumaBrojeva(1,3)

Correct

That's right! You chose the correct response.

VIEW RESULTS

Slika 46 Rješenje trećeg pitanja - iSpring Suite

(Izrada autorice rada)

Kako ovaj alat radi na principu slajdova, svako pitanje prikazuje se posebno i zato je nemoguće prikazati izgled testa sa sva tri pitanja kako je to za prijašnja dva sustava napravljeno. Time će se ovdje prikazati još jedna značajka kojom polaznik može prilikom rješavanja testa, u lijevom kutu kliknuti na opciju *Question List* i time vidjeti popis svih pitanja testa. To mu omogućuje da se kreće kroz pitanja testa, a uz to prikazan mu je ostvaren rezultat za svako pitanje. iSpring Suite nudi dosta vrsta pitanja, ali popunjavanje praznina najbolje je rješenje kada se želi ispitati programersko znanje polaznika. Kod se unosi kao običan tekst, a ispravnost odgovora slijedi odmah nakon podnošenja odgovora. Tu opet veliku ulogu imaju autori pitanja koji se moraju pobrinuti za sve moguće točne odgovore na neko pitanje. Alat nudi mnogo opcija za uređenja testa prema željama autora.

The screenshot shows the iSpring Suite test interface. At the top left, there is a 'Question List' button highlighted with a red box. The main content area displays a question in Croatian: 'Kreirajte varijablu x čija je vrijednost 3. Napišite program koji provjerava parnost varijable x. Ukoliko je broj paran program ispisuje poruku: "Broj je paran", u suprotnom se ispisuje: "Broj je neparan".' Below the question is a table with three rows of questions. The first row is completed with a score of 1 and a green checkmark. The second row is highlighted in yellow and is currently selected. At the bottom right, there is a blue 'SUBMIT' button.

Question	Awarded	Points	Result
1. Napišite program u kojem su definirane varijable $x = 5$ i $y = 10$. Kreirajte varijablu z čija će vrijednost biti zbroj varijabli x i y . Ispišite vr...	1	1	✓
2. Kreirajte varijablu x čija je vrijednost 3. Napišite program koji provjerava parnost varijable x . Ukoliko je broj paran program ispisuje...	-	1	
3. Napišite funkciju <code>sumaBrojeva</code> koja vraća zbroj brojeva x i y . Pozovite funkciju s vrijednostima $x = 1$ i $y = 3$.	-	1	

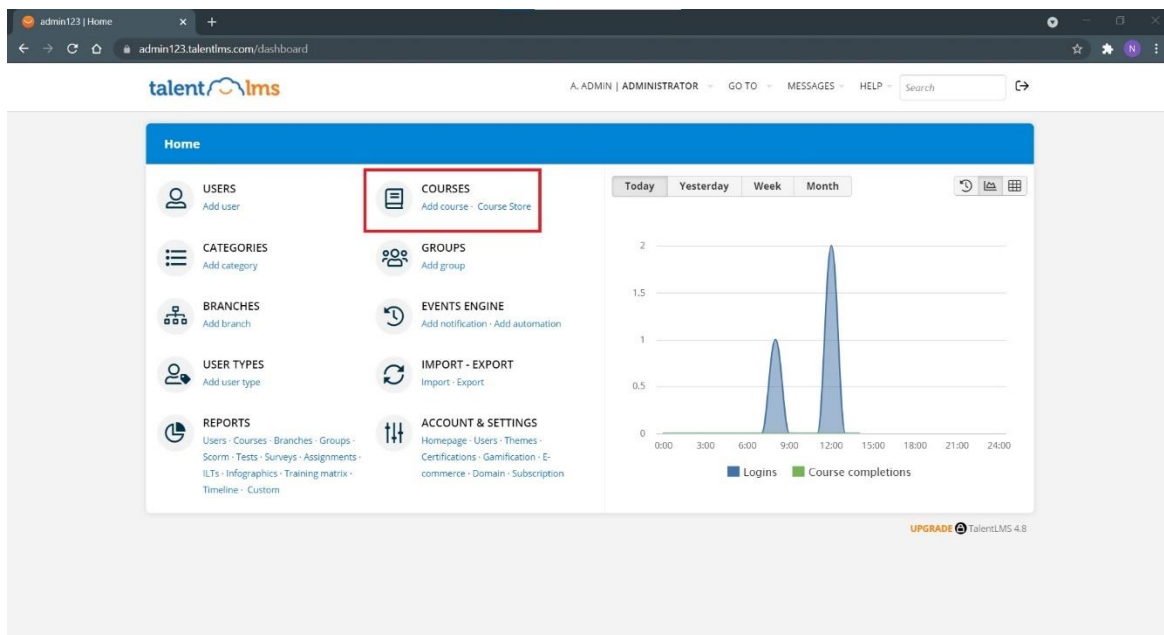
Slika 47 Izgled testa - iSpring Suite

(Izrada autorice rada)

5.6. Kreiranje online testa u sustavu TalentLMS

Na redu je zadnji, najmlađi sustav koji se želi testirati kada je u pitanju izrada testa za online provjeru znanja iz programiranja. Nakon uspješne registracije, na stranicama sustava prijavljuje se sa zadanim podacima. Otvara se početna stranica sustava TalentLMS (Slika 48). To je stranica *Home* i s lijeve strane prikazan je izbornik prema kategorijama. S desne strane nalazi se grafikon koji prikazuje statističke podatke na dnevnoj, tjednoj ili mjesečnoj bazi o prijavama i završetcima tečajeva. Iznad se nalazi traka s izbornikom gdje se može vidjeti kako je trenutna uloga korisnika *Administrator*, a postoje još instruktor i učenik. Pored uloge ima se pristup izborniku, porukama i bazi znanja koja omogućuje da korisnik potraži pomoć i slijedi instrukcije izravno iz sustava, što je zaista veliki plus za početnike koji koriste TalentLMS.

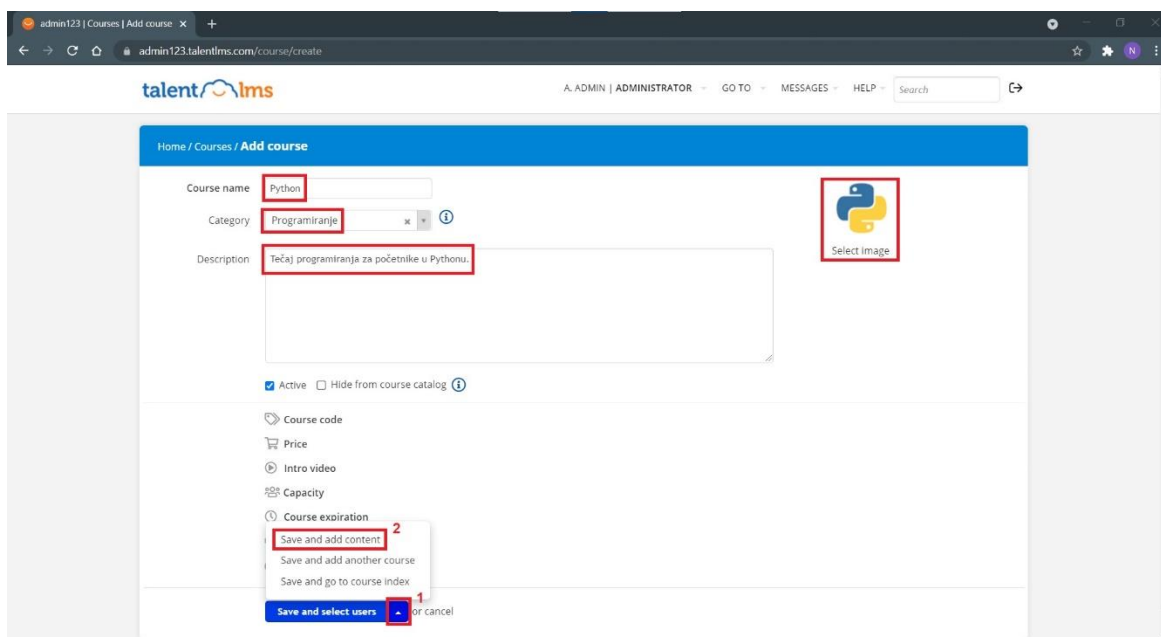
U ovom sustavu također je potrebno kreirati tečaj kako bi se moglo pristupiti opciji kreiranja testa. Na početnoj stranici u kategoriji *Courses* odabire se opcija *Add course* (Slika 48).



Slika 48 Kreiranje tečaja - TalentLMS

(Izrada autorice rada)

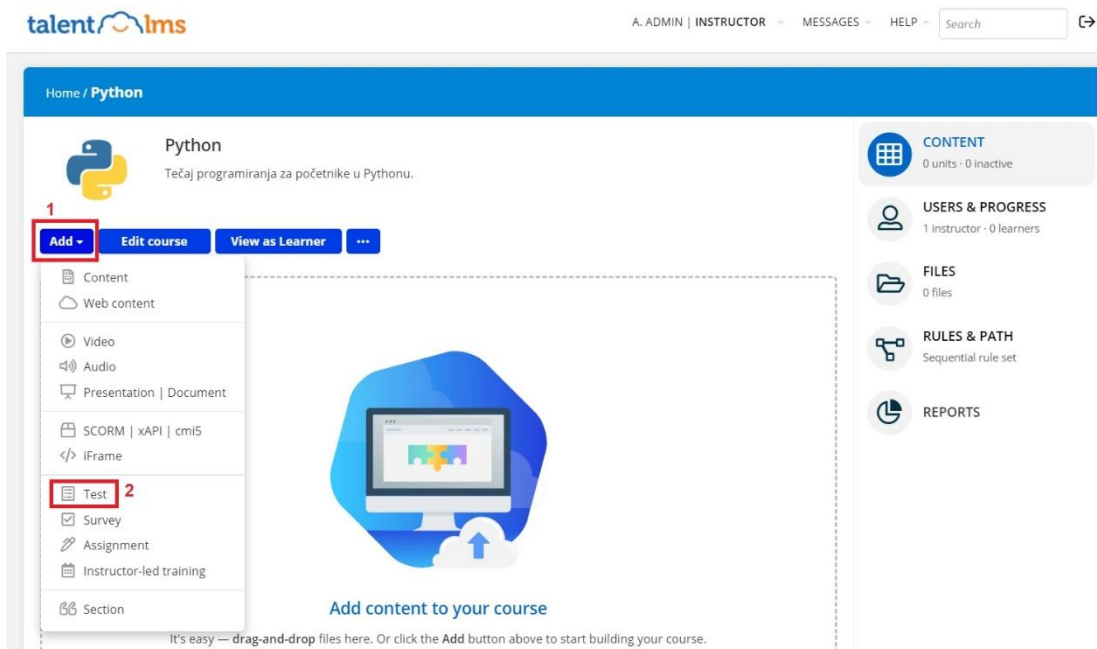
Otvora se prozor koji je prikazan na slici u nastavku. U ovom koraku unose se podaci za tečaj koji se želi kreirati. Ime, kategorija, opis i slika. Postoje još određene postavke kojima se tečaj može prilagoditi. Klikom na strelicu kod gumba *Save and select users* otvara se izbornik u kojem je potrebno odabrati opciju *Save and add content* (Slika 49). Ovime je uspješno kreiran tečaj *Python* koji se automatski otvara.



Slika 49 Podaci tečaja - TalentLMS

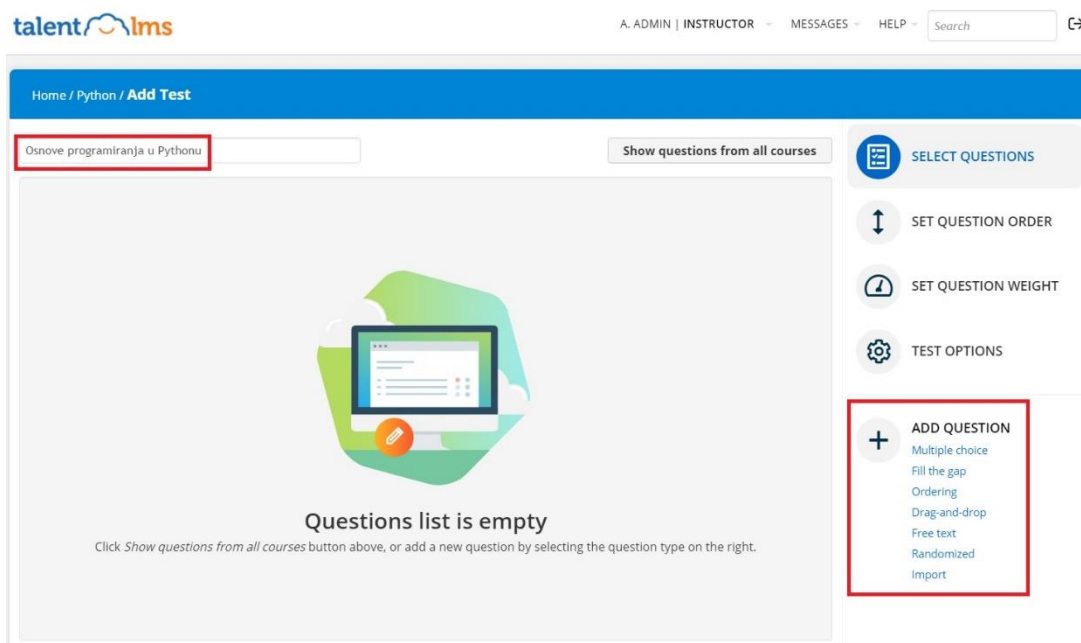
(Izrada autorice rada)

Sučelje tečaja prikazano je na sljedećoj slici. Kako bi se dodao test potrebno je kliknuti na opciju *Add*, a zatim iz padajućeg izbornika izabrati *Test* (Slika 50). Otvara se prozor u kojem se kreira test. Za početak u prazno polje unosi se naziv testa - *Osnove programiranja u Pythonu*. S desne strane nalaze se opcije kojima se može označiti pitanje, urediti redoslijed pitanja, postaviti težina pitanja ali i otvoriti postavke testa. Ispod se nalazi popis vrsta pitanja koje je moguće kreirati koristeći ovaj sustav. Vidljivo je kako ovaj sustav ima najmanje vrsta pitanja na raspolaganju od svih do sada (Slika 51).



Slika 50 Kreiranje testa - TalentLMS

(Izrada autorice rada)



Slika 51 Dodavanje pitanja - TalentLMS

(Izrada autorice rada)

Nakon što se upiše ime testa i postave željene postavke, u donjem desnom kutu odabirom na vrstu pitanja otvara se prozor u kojem se sastavlja željeno pitanje. S obzirom na mogućnosti sustava, kako bi se kreirao online test za rješavanje izazova, koristit će se vrsta pitanja *Free text*. Ova vrsta pitanja omogućuje da polaznik piše kod u obliku običnog teksta kao odgovor na pitanje koje sustav provjerava ovisno o odgovorima koje je zadao autor testa i automatski daje povratni odgovor polazniku (Slika 52). U prvo polje unosi se tekst pitanja, drugi dio služi za kreiranje odgovora i dodjeljivanje bodova i na kraju pritiskom na gumb **Save** uspješno je kreirano pitanje.

Home / Python / New test / Add question (Free text)

1

2

3

Slika 52 Kreiranje pitanja - TalentLMS

(Izrada autorice rada)

5.6.1. Provedba online testa

I za ovaj zadnji sustav u radu kreiranje svakog pitanja prikazat će se kroz tri koraka: **zadatak, opis problema i rješenje**. Zadatak će sadržavati sliku koja prikazuje postavke kreiranja pitanja. Opis problema za sva pitanja je isti, a on je **pisanje izvornog koda kao odgovor na pitanje**, stoga se opis problema neće ponavljati za svako pitanje. Dok rješenje prikazuje uspješno kreirano i provedeno pitanje u sustavu TalentLMS.

Zadatak - prvo pitanje

Home / Python / New test / Add question (Free text)

Rad s varijablama

Napišite program u kojem su definirane varijable $x = 5$ i $y = 10$. Kreirajte varijablu z čija će vrijednost biti zbroj varijabli x i y . Ispišite vrijednost varijable z .

Consider correct when accumulated points are greater or equal to

When	contains	the word	<input type="text" value="x=5 x = 5"/>	add	<input type="text" value="1"/>	points
When	contains	the word	<input type="text" value="y=10 y = 10"/>	add	<input type="text" value="1"/>	points
When	contains	the word	<input type="text" value="z=x+y z = x+y z = x + y"/>	add	<input type="text" value="1"/>	points
When	contains	the word	<input type="text" value="print(z) print (z)"/>	add	<input type="text" value="1"/>	points

Slika 53 Kreiranje prvog pitanja - TalentLMS

(Izrada autorice rada)

Rješenje

CORRECT

Rad s varijablama

Napišite program u kojem su definirane varijable $x = 5$ i $y = 10$. Kreirajte varijablu z čija će vrijednost biti zbroj varijabli x i y . Ispišite vrijednost varijable z .

```
x=5  
y=10  
z=x+y
```

```
print(z)
```

Slika 54 Rješenje prvog pitanja - TalentLMS

(Izrada autorice rada)

Zadatak - drugo pitanje

Home / Python / New test / Add question (Free text)

IF izjava

Kreirajte varijablu x čija je vrijednost 3. Napišite program koji provjerava parnost varijable x . Ukoliko je broj paran program ispisuje poruku: "Broj je paran", u suprotnom se ispisuje: "Broj je neparan".

Consider correct when accumulated points are greater or equal to

When	contains	the word	<input type="text" value="x=3 x = 3"/>	add	<input type="text" value="1"/>	points
When	contains	the word	<input type="text" value="if x%2==0: if x%2 == 0:"/>	add	<input type="text" value="1"/>	points
When	contains	the word	<input type="text" value="print('Broj je paran')"/>	add	<input type="text" value="1"/>	points
When	contains	the word	<input type="text" value="else:"/>	add	<input type="text" value="1"/>	points
When	contains	the word	<input type="text" value="print('Broj je neparan')"/>	add	<input type="text" value="1"/>	points

Slika 55 Kreiranje drugog pitanja - TalentLMS

(Izrada autorice rada)

Rješenje

CORRECT

IF izjava

Kreirajte varijablu x čija je vrijednost 3. Napišite program koji provjerava parnost varijable x . Ukoliko je broj paran program ispisuje poruku: "Broj je paran", u suprotnom se ispisuje: "Broj je neparan".

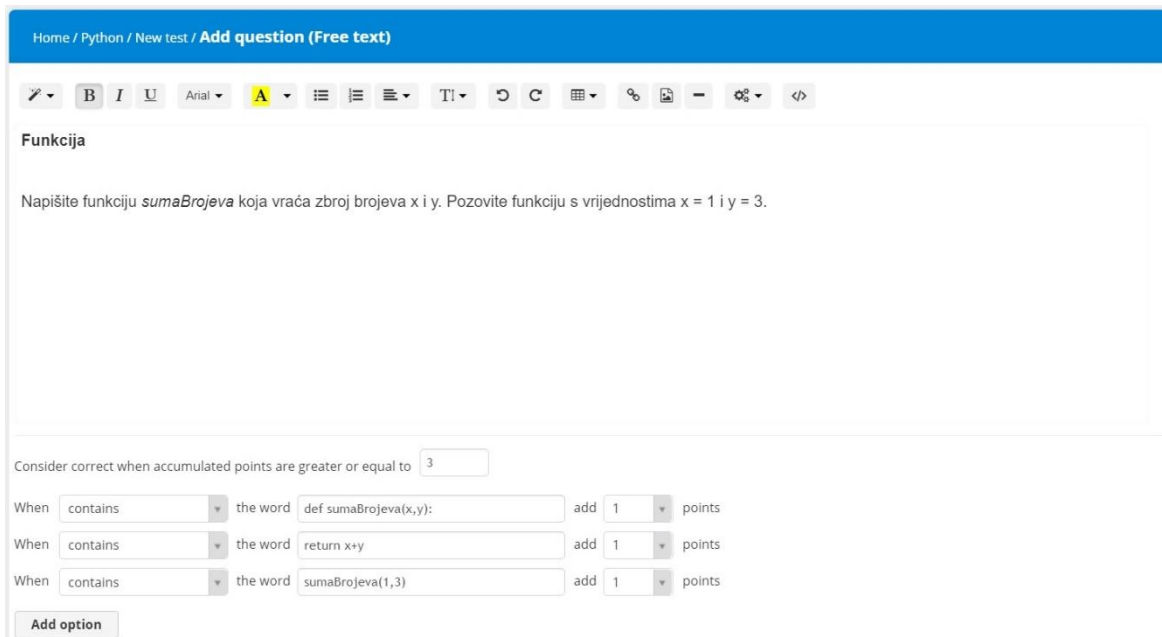
```
x=3
```

```
if x%2 == 0:  
    print("Broj je paran")  
else:  
    print("Broj je neparan")
```

Slika 56 Rješenje drugog pitanja - TalentLMS

(Izrada autorice rada)

Zadatak - treće pitanje



Home / Python / New test / Add question (Free text)

Funkcija

Napišite funkciju *sumaBrojeva* koja vraća zbroj brojeva x i y. Pozovite funkciju s vrijednostima x = 1 i y = 3.

Consider correct when accumulated points are greater or equal to

When the word add points

When the word add points

When the word add points

Slika 57 Kreiranje trećeg pitanja - TalentLMS

(Izrada autorice rada)

Rješenje

CORRECT

Funkcija

Napišite funkciju *sumaBrojeva* koja vraća zbroj brojeva x i y. Pozovite funkciju s vrijednostima x = 1 i y = 3.

```
def sumaBrojeva(x,y):  
    return x+y
```

```
sumaBrojeva(1,3)
```

Slika 58 Rješenje trećeg pitanja - TalentLMS

(Izrada autorice rada)

Čak i u ovom LMS sustavu koji ima najmanji izbor pitanja uspio se provesti test za online provjeru znanja iz programiranja. Sustav je modernog i privlačnog dizajna, a uz bazu znanja koja postoji snalaženje u njemu je zaista olakšano. Da bi se i pomoću ovog sustava riješio najveći izazov za autore tečajeva programiranja, koristila se vrsta pitanja *Free text*. Naime, ni ovaj sustav nema mogućnost ugradnje dodatka koji bi ovaj izazov riješio kao što to radi CodeRunner, ali kreiranje testa nije nemoguće. Vrsta pitanja *slobodni tekst* od autora testa zahtjeva kreativnost i mnogo truda kako bi ovako kreirani test programiranja funkcionirao.

The image shows a screenshot of a LMS interface with three programming questions. Each question has a title, a description, a text input area, a 'Submit answer' button, and a progress indicator.

Rad s varijablama
Napišite program u kojem su definirane varijable $x = 5$ i $y = 10$. Kreirajte varijablu z čija će vrijednost biti zbroj varijabli x i y . Ispišite vrijednost varijable z .

IF izjava
Kreirajte varijablu x čija je vrijednost 3. Napišite program koji provjerava parnost varijable x . Ukoliko je broj paran program ispisuje poruku: "Broj je paran", u suprotnom se ispisuje: "Broj je neparan".

Funkcija
Napišite funkciju `sumaBrojeva` koja vraća zbroj brojeva x i y . Pozovite funkciju s vrijednostima $x = 1$ i $y = 3$.

Slika 59 Izgled testa - TalentLMS

(Izrada autorice rada)

5.7. Usporedba autorskih alata

Nakon pokušaja rješavanja najvećeg izazova za autore tečajeva programiranja u ovom zadnjem dijelu rada prikazati će se usporedba korištenih autorskih alata. Jednostavnost korištenja i korisnost tehnologije su dva ključna činitelja koja imaju najjači utjecaj na stav odnosno korištenje iste od strane korisnika (Davis, 1993). Prvi dio tablice odnosi se na vrste pitanja koja alati nude kada se želi kreirati test za online provjeru znanja, a drugi dio su kriteriji koji se odnose na samo korištenje alata.

Tablica 9 Usporedba autorskih alata

VRSTE PITANJA	CANVAS	Moodle	iSpring Suite	TalentLMS
Točno/Netočno	✓	✓	✓	✗
Višestruki izbor s jednim točnim odgovorom	✓	✓	✓	✓
Višestruki izbor s više točnih odgovora	✓	✓	✓	✓
Kratki odgovor	✗	✓	✓	✗
Brojčani odgovor	✓	✓	✓	✗
Esejsko pitanje	✓	✓	✓	✓
Popuni prazninu/praznine	✓	✓	✓	✓
Spajanje	✓	✓	✓	✓
Redoslijed	✗	✗	✓	✓
Odabir iz padajućih izbornika	✓	✓	✓	✗
Računsko pitanje	✓	✓	✗	✗
Dovuci riječi	✗	✓	✓	✗
Slikovno pitanje	✗	✓	✓	✗
Učitaj datoteku	✓	✗	✗	✗
KORIŠTENJE ALATA	CANVAS	Moodle	iSpring Suite	TalentLMS
Dizajn alata				
Jednostavnost korištenja				
Baza znanja - pomoć pri korištenju alata				
Podržan hrvatski jezik	✗	✓	✗	✗
Djelomično/potpuno rješavanje izazova ⁴	djelomično	potpuno	djelomično	djelomično

(Izrada autorice rada)

⁴ Pisanje izvornog koda kao odgovor na pitanje u obliku teksta je djelomično rješavanje izazova, dok potpuno rješavanje izazova podrazumijeva pisanje izvornog koda koji se automatski pokreće i na taj se način provjerava ispravnost odgovora.



Slika 60 Ocjene kriterija

(Izrada autorice rada)

Ocjene kriterija dizajn alata, jednostavnost korištenja i baza znanja u tablici usporedbe prikazane su pomoću boja. Svaka boja označava određenu ocjenu kako je prikazano na prethodnoj slici (Slika 60).

Ako se usporede alati s obzirom na vrste pitanja koja nude kada se želi kreirati online provjera znanja, Moodle i iSpring Suite pružaju najviše mogućnosti za autora testa, dok TalentLMS ima najmanji izbor pitanja. Tablica u nastavku prikazuje rang odabira alata kada je u pitanju izrada online provjere znanja iz programiranja. Rangiranje alata napravljeno je na temelju drugog dijela tablice usporedbe koji se temelji na kriterijima kada je u pitanju korištenje samih alata. Za očekivati je da Moodle zauzme prvo mjesto, jer uz odlične ocjene za dizajn sučelja, jednostavnost korištenja i pomoć pri izradi sadržaja, jedini je koji podržava hrvatski jezik, a ono najvažnije, jedini je alat u kojem se izazov riješio u potpunosti. Od ostala tri alata niti jedan ne podržava hrvatski jezik i sa svakim se izazov riješio tek djelomično. S obzirom na veoma dobre ocjene za ostale kriterije (dizajn sučelja, jednostavnost korištenja, pomoć pri izradi sadržaja) CANVAS se nalazi na drugom mjestu. Bez obzira na manji izbor pitanja, zbog boljih ocjena i mogućnosti, TalentLMS se nalazi ispred alata iSpring Suite.

Tablica 10 Rang odabira alata

	CANVAS	Moodle	iSpring Suite	TalentLMS
RANG ODABIRA ALATA	2.	1.	4.	3.

(Izrada autorice rada)

6. ZAKLJUČAK

Život u suvremenom „informativnom“ društvu i razne svjetske krize poput trenutne svjetske pandemije doprinose sve većoj upotrebi i razvoju online obrazovanja. Još uvijek se ne može reći kako je e-učenje u potpunosti zamijenilo tradicionalno obrazovanje, što ne znači da se to u skoroj budućnosti neće dogoditi. Da bi se to ostvarilo potreban je veći tehnološki napredak, povećanje računalne pismenosti u društvu, podizanje kvalitete autorskih alata i tehnologija za e-učenje i ono najbitnije, pronalaženje načina da rezultati online provjera znanja budu realni i pravedni.

Danas je tehnologija prisutna u svim segmentima ljudskog života stoga su stručnjaci iz tih područja iznimno traženi na tržištu rada. Programiranje i online učenje područja su koja se iznimno slažu, a kroz rad je prikazana njihova teorija i razvoj kroz povijest. Od svih vrsta alata za online učenje programiranja, u ovom radu se za izradu praktičnog dijela koriste sustavi za upravljanje učenjem. Oni nude mogućnost kreiranja tečaja, a izrada tečaja programiranja je dio teme ovog rada. Kratki test od tri pitanja proveo se u četiri odabrana alata i tako su se prikazale njihove mogućnosti kada je online provjera znanja programiranja u pitanju. Moodle je potpuno besplatan alat, a pokazao se kao najbolji odabir kada je u pitanju najveći izazov svih autora tečajeva programiranja, a to je pisanje izvornog koda kao odgovor na pitanje. Time se dokazalo da ne mora nužno značiti da su alati koji nisu besplatni i javno dostupni bolji i imaju više mogućnosti. Danas na tržištu postoji veliki izbor alata kada je kreiranje sadržaja za e-učenje u pitanju, stoga prema potrebama, željama i mogućnostima ovisi i odabir autora. Mogućnosti za online provjeru znanja iz programiranja zasigurno će s vremenom porasti, a do tada autori tečajeva moraju biti kreativni, snalažljivi i uložiti više truda ako polaznicima žele postaviti pitanje tako da od njih zahtijevaju pisanje izvornog koda, jer vidi se da to nije nemoguće.

Daljnjim razvitkom i napretkom tehnologije zasigurno se može očekivati kako će se u budućnosti na području online obrazovanja noviteti samo nizati. Upravo radi toga, ako se osoba odluči djelovati u ovom području, mora biti spremna da nikada ne može prestati učiti. Online obrazovanje se razvija i širi neizmjenom brzinom, a hoće li uopće i kada u potpunosti zamijeniti tradicionalno obrazovanje, preostaje vidjeti.

LITERATURA

Knjige:

1. Ćukušić, M. i Jadrić, M. (2012) *e-učenje: koncept i primjena*. Zagreb: Školska knjiga.
2. Deitel, P.J. i Deitel, H.M. (2008), *Internet & World Wide Web: How to Program*, 4th edition, [e-book], New Jersey: Prentice Hall. Dostupno: <https://www.auhd.site/upfiles/elibrary/Azal2020-01-22-01-38-51-40850.pdf> [pristup: 10.07.2021.]
3. Jakupović, A. i Šuman, S. (2014) *Osnove programiranja*. Rijeka: Veleučilište u Rijeci.
4. Kingsley-Hughes, A. i Kingsley-Hughes, K. (2005), *Beginning Programming*, [e-book], Indianapolis: Wiley Publishing. Scribd, dostupno: <https://www.scribd.com/doc/89546957/Wrox-Beginning-Programming-Apr-2005> [pristup: 28.06.2021.]
5. Lee, K.D. (2017), *Foundations of Programming Languages*, 2th edition, [e-book], New York: Springer. Scribd, dostupno: <https://www.scribd.com/document/462133775/2017-Book-FoundationsOfProgrammingLangua-pdf> [pristup: 01.07.2021.]
6. Panian, Ž. (1999) *Poslovna informatika: koncepti, metode i tehnologija*. Zagreb: Informator.
7. Parker, M. (2014), *Things to Make and Do in the Fourth Dimension*, [e-book], New York: Ferrar, Straus and Giroux. Dostupno: <https://www.pdfdrive.com/things-to-make-and-do-in-the-fourth-dimension-a-mathematicians-journey-through-narcissistic-numbers-optimal-dating-algorithms-at-least-two-kinds-of-infinity-and-more-e166043704.html> [pristup: 28.06.2021.]

Radovi i članci:

1. Davis, F.D. (1993) User acceptance of information technology: system characteristics, user perceptions and behavioral impacts. *International Journal of Man - Machine Studies*, 38, str. 475-487 [pristup: 21.08.2021]

2. *Hrvatska enciklopedija* (2021) Automat. Mrežno izdanje. Zagreb: Leksikografski zavod Miroslav Krleža. Dostupno:
<https://www.enciklopedija.hr/natuknica.aspx?ID=4743> [pristup: 27.06.2021.]
3. *Hrvatska enciklopedija* (2021) Bernoulli, Jakob. Mrežno izdanje. Zagreb: Leksikografski zavod Miroslav Krleža. Dostupno:
<https://www.enciklopedija.hr/natuknica.aspx?id=7183> [pristup: 28.06.2021.]
4. *Hrvatska enciklopedija* (2021) Booleova algebra. Mrežno izdanje. Zagreb: Leksikografski zavod Miroslav Krleža. Dostupno:
<https://enciklopedija.hr/natuknica.aspx?ID=70097> [pristup: 28.06.2021.]
5. *Hrvatska enciklopedija* (2021) Programiranje. Mrežno izdanje. Zagreb: Leksikografski zavod Miroslav Krleža. Dostupno:
<https://www.enciklopedija.hr/natuknica.aspx?ID=50555> [pristup: 27.06.2021.]
6. *Hrvatska enciklopedija* (2021) Stenografija. Mrežno izdanje. Zagreb: Leksikografski zavod Miroslav Krleža. Dostupno:
<https://www.enciklopedija.hr/natuknica.aspx?id=58004> [pristup: 04.07.2021.]
7. *Hrvatska enciklopedija* (2021) Tečaj. Mrežno izdanje. Zagreb: Leksikografski zavod Miroslav Krleža. Dostupno:
<https://www.enciklopedija.hr/natuknica.aspx?id=60621> [pristup: 01.08.2021.]
8. Kralj, D. (2018) Primjena računala. Karlovac: Veleučilište u Karlovcu. Dostupno:
http://www.vuka.hr/fileadmin/user_upload/knjiznica/on_line_izdanja/Damir_Kralj-Primjena_racunala.pdf [pristup: 26.06.2021.]
9. Lipjlin, N. (1993) Čime se bave informacijske znanosti?. *Journal of Information and Organizational Sciences*, (17), str. 67-97. Varaždin: Fakultet organizacije i informatike. Dostupno: <https://hrcak.srce.hr/79439> [pristup: 26.06.2021.]
10. Miličević, M. (2020) *Usporedna analiza alata za učenje programiranja*. Završni rad. Pula: Sveučilište Jurja Dobrile u Puli. Dostupno:
<https://repozitorij.unipu.hr/islandora/object/unipu%3A4941/datastream/PDF/view> [pristup: 10.07.2021.]
11. Putica, M. (2018) Semantički web. *Hum: časopis Filozofskog fakulteta Sveučilišta u Mostaru*, 13(19), str. 99-116. Mostar: Filozofski fakultet Sveučilišta u Mostaru. Dostupno: <https://hrcak.srce.hr/212440> [pristup: 10.07.2021.]

Internet izvori:

1. Canvas, *Our Company Story*. Dostupno: <https://www.instructure.com/about/our-story> [pristup: 17.07.2021.]
2. Capterra, *Comparing 4 Learning Management System Software Products*. Dostupno: <https://www.capterra.com/learning-management-system-software/compare/127214-80691-218265-132935/CANVAS-vs-Moodle-vs-iSpring-Suite-Max-vs-TalentLMS> [pristup: 13.07.2021.]
3. Capterra, *LMS Software*. Dostupno: https://www.capterra.com/learning-management-system-software/?pricing_options=%5B8%5D&feature=%5B19208%5D&sortOrder=most_reviews [pristup: 13.07.2021.]
4. Fortune Business Insights, *Learning Management System (LMS) Market*. Dostupno: <https://www.fortunebusinessinsights.com/industry-reports/learning-management-system-market-101376> [pristup: 20.08.2021.]
5. GeeksforGeeks, *Web 1.0, Web 2.0 and Web 3.0 with their difference*. Dostupno: <https://www.geeksforgeeks.org/web-1-0-web-2-0-and-web-3-0-with-their-difference/> [pristup: 10.07.2021.]
6. iSpring, *eLearning Software That Works*. Dostupno: <https://www.ispringsolutions.com/> [pristup: 20.07.2021.]
7. Moodle, *About Moodle*. Dostupno: [https://docs.moodle.org/311/en/About Moodle](https://docs.moodle.org/311/en/About_Moodle) [pristup: 18.07.2021.]
8. Moodle, *CodeRunner*. Dostupno: https://moodle.org/plugins/qtype_coderunner [pristup: 09.08.2021.]
9. Moodle, *Statistics*. Dostupno: <https://stats.moodle.org/> [pristup: 18.07.2021.]
10. Norman, J.M., *Pitman Shorthand, & The First "Correspondence Course"*. Dostupno: <https://www.historyofinformation.com/detail.php?id=2050> [pristup: 04.07.2021.]
11. PYPL - PopularitY of Programming Language, *PYPL Indeks*. Dostupno: <https://pypl.github.io/PYPL.html> [pristup: 04.08.2021.]
12. Stack Overflow, *2020 Developer Survey: Technology*. Dostupno: <https://insights.stackoverflow.com/survey/2020#technology> [pristup: 30.06.2021.]

13. Stack Overflow, *2021 Developer Survey: Developer Profile*. Dostupno: <https://insights.stackoverflow.com/survey/2021#developer-profile> [pristup: 30.06.2021.]
14. Stack Overflow, *2021 Developer Survey: Experience*. Dostupno: <https://insights.stackoverflow.com/survey/2021#developer-profile-experience> [pristup: 10.07.2021.]
15. Stack Overflow, *2021 Developer Survey: Technology*. Dostupno: <https://insights.stackoverflow.com/survey/2021#technology> [pristup: 30.06.2021.]
16. Sveučilišni računski centar (srce), *E-učenje*. Dostupno: <https://www.srce.unizg.hr/centar-za-e-ucenje/o-centru/e-ucenje> [pristup: 04.07.2021.]
17. Syndicode Team, *Types of developers*. Dostupno: <https://syndicode.com/blog/types-of-developers/> [pristup: 30.06.2021.]
18. TalentLMS, *About us*. Dostupno: <https://www.talentlms.com/about> [pristup: 22.07.2021.]
19. TalentLMS, *Types of LMS*. Dostupno: <https://www.talentlms.com/what-is-an-lms#definition> [pristup: 12.07.2021.]
20. Timetoast timelines, *History of LMS*. Dostupno: <https://www.timetoast.com/timelines/history-of-lms-19f41c38-48f6-4f8d-b8f3-a3ebc4ebed15> [pristup: 12.07.2021.]
21. World Manager, *A Timeline of LMS History*. Dostupno: <https://www.worldmanager.com/resources/history-of-lms/> [pristup: 12.07.2021.]
22. YouTube, *Install Moodle on your PC*. Dostupno: <https://www.youtube.com/watch?v=nd4dytem4DA> [pristup: 16.06.2021.]

POPIS SLIKA

Slika 1 Pascalina.....	4
Slika 2 Osnovni pojmovi programiranja	6
Slika 3 Proces programiranja	7
Slika 4 Primjer dijagrama toka (lijevo) i pseudokoda i programa (desno).....	8
Slika 5 Odnos e-učenja i online učenja	15
Slika 6 Stenographic Sound-Hand by Isaac Pitman	17
Slika 7 Online učenje i programiranje.....	22
Slika 8 Razvoj LMS sustava.....	29
Slika 9 Vrste sustava za upravljanje učenjem (LMS-a).....	30
Slika 10 Izgled korisničkog sučelja - CANVAS	32
Slika 11 Izgled korisničkog sučelja - Moodle	33
Slika 12 Početni zaslon - iSpring Suite.....	34
Slika 13 Izgled korisničkog sučelja - iSpring QuizMaker.....	35
Slika 14 Izgled korisničkog sučelja - TalentLMS.....	36
Slika 15 Struktura online testa: Osnove programiranja u Pythonu	40
Slika 16 Kreiranje tečaja - CANVAS.....	41
Slika 17 Podaci testa - CANVAS.....	42
Slika 18 Dodavanje pitanja - CANVAS	42
Slika 19 Kreiranje pitanja - CANVAS.....	43
Slika 20 Kreiranje prvog pitanja - CANVAS	44
Slika 21 Rješenje prvog pitanja - CANVAS	44
Slika 22 Kreiranje drugog pitanja - CANVAS	45
Slika 23 Rješenje drugog pitanja - CANVAS	45
Slika 24 Kreiranje trećeg pitanja - CANVAS	46
Slika 25 Rješenje trećeg pitanja - CANVAS	46
Slika 26 Izgled testa - CANVAS	47
Slika 27 Kreiranje e-kolegija - Moodle	48
Slika 28 Podaci e-kolegija - Moodle	49
Slika 29 Dodavanje aktivnosti (test) - Moodle.....	49
Slika 30 Podaci testa - Moodle	50
Slika 31 Instalacija dodatka - Moodle	51
Slika 32 Instalacija dodatka CodeRunner - Moodle	51
Slika 33 Dodavanje pitanja - CodeRunner - Moodle	52

Slika 34 Rješenje prvog pitanja - Moodle	53
Slika 35 Rješenje drugog pitanja - Moodle	54
Slika 36 Rješenje trećeg pitanja - Moodle	55
Slika 37 Izgled testa - Moodle	56
Slika 38 Kreiranje testa - iSpring Suite	57
Slika 39 Dodavanje pitanja - iSpring Suite.....	58
Slika 40 Kreiranje pitanja - iSpring Suite	59
Slika 41 Kreiranje prvog pitanja - iSpring Suite.....	60
Slika 42 Rješenje prvog pitanja - iSpring Suite.....	60
Slika 43 Kreiranje drugog pitanja - iSpring Suite	61
Slika 44 Rješenje drugog pitanja - iSpring Suite.....	61
Slika 45 Kreiranje trećeg pitanja - iSpring Suite.....	62
Slika 46 Rješenje trećeg pitanja - iSpring Suite.....	62
Slika 47 Izgled testa - iSpring Suite.....	63
Slika 48 Kreiranje tečaja - TalentLMS	64
Slika 49 Podaci tečaja - TalentLMS.....	65
Slika 50 Kreiranje testa - TalentLMS	66
Slika 51 Dodavanje pitanja - TalentLMS.....	66
Slika 52 Kreiranje pitanja - TalentLMS	67
Slika 53 Kreiranje prvog pitanja - TalentLMS	68
Slika 54 Rješenje prvog pitanja - TalentLMS.....	68
Slika 55 Kreiranje drugog pitanja - TalentLMS	69
Slika 56 Rješenje drugog pitanja - TalentLMS.....	69
Slika 57 Kreiranje trećeg pitanja - TalentLMS	70
Slika 58 Rješenje trećeg pitanja - TalentLMS.....	70
Slika 59 Izgled testa - TalentLMS.....	71
Slika 60 Ocjene kriterija	73

POPIS TABLICA

Tablica 1 Vrste programera i programski jezici.....	10
Tablica 2 Prednosti i nedostaci e-učenja u odnosu na tradicionalno učenje	19
Tablica 3 Usporedba sinkronog i asinkronog online učenja	20
Tablica 4 Razlika Web 1.0, Web 2.0 i Web 3.0.....	24
Tablica 5 Usporedba odabranih LMS sustava	31
Tablica 6 Postavke kreiranja prvog pitanja - CodeRunner	53
Tablica 7 Postavke kreiranja drugog pitanja - CodeRunner	54
Tablica 8 Postavke kreiranja trećeg pitanja - CodeRunner	55
Tablica 9 Usporedba autorskih alata	72
Tablica 10 Rang odabira alata	73

POPIS GRAFIKONA

Grafikon 1 Postotak zastupljenosti programera u svijetu 2021. godine.....	11
Grafikon 2 Najčešće korišteni programski jezici 2020. i 2021. godine	12
Grafikon 3 Popularnost programskih jezika prema pretraživanju vodiča na Googleu	13
Grafikon 4 Postotak načina učenja programiranja u 2021. godini	26

SAŽETAK

Predmet ovog diplomskog rada je primjena informacijsko komunikacijske tehnologije u području online obrazovanja. U radu se ukazuje na bitne pojmove iz područja programiranja i online učenja. Kako procijeniti znanje polaznika online tečaja izazov je za autore tečajeva programiranja i glavni problem ovog rada. Cilj je prikazati moguća rješenja u različitim sustavima za upravljanje učenjem kada je u pitanju procjena znanja iz programiranja, odnosno rješavanje najvećeg izazova za autore tečajeva, a to je pisanje izvornog koda kao odgovor na postavljeno pitanje. U praktičnom dijelu radu prikazan je proces provedbe online testa u svakom odabranom sustavu za upravljanje učenjem (CANVAS, Moodle, iSpring Suite i TalentLMS). Na kraju je prikazana usporedba odabranih sustava na temelju dobivenih rezultata koji dokazuju da postoji mogućnost kreiranja i provedbe ovakve vrste pitanja čime se omogućuje kvalitetnija procjena znanja kada je online učenje programiranja u pitanju.

Ključne riječi: online učenje, programiranje, autorski alati, sustavi za upravljanje učenjem, online tečaj, procjena znanja, izazovi online tečaja programiranja

ABSTRACT

The subject of this diploma paper is the application of Information and Communication Technology in the field of Online Education. The paper deals with the most important terms from the area of programming and online learning. The biggest challenge for the developers of online programming courses and also the main problem of this diploma paper was how to assess prior knowledge and skills of the course participators. The main aim of this paper was to demonstrate possible solutions for the assessment of programming skills derived from each of the learning management systems, i.e. the aim was to solve the biggest challenge that course developers face, which is writing code as an answer to a given question. The practical part of the paper displays the process of online tests execution in each of the chosen learning management systems (CANVAS, Moodle, iSpring, Suite, TalentLMS). The final part of the paper contains the comparison of the chosen systems based on the results of the research. The research proved that there is a possibility of creating and executing tests which enable a more effective assessment of programming skills during online learning.

Keywords: online learning, programming, developer tools, learning management systems, online course, assessment of knowledge, the challenges of online programming courses