IZRADA GAMIFICIRANE E-LEARNING PLATFORME

Biuk, Adriana

VISOKO UČILIŠTE ALGEBRA

DIPLOMSKI RAD

IZRADA GAMIFICIRANE E-LEARNING PLATFORME

Adriana Biuk

Zagreb, rujan 2018.

Predgovor

Ova stranica treba sadržavati izjavu ili zahvalu kandidata…..

Prilikom uvezivanja rada, Umjesto ove stranice ne zaboravite umetnuti original potvrde o prihvaćanju teme diplomskog rada kojeg ste preuzeli u studentskoj referadi

Sažetak

Tema ovog diplomskog rada je prikazati mogućnosti primjene *gamifikacije* u vidu *e-learning* rješenja koje prati kolegije koji se održavaju na Viskom učilištu Algebra. Hipoteza ovog rada je pokušati dokazati kako osmišljena web platforma može povećati angažman u učenju, povećati razumijevanje određenih kolegija te motivirati studente.
Rad se sastoji od uvodnog teorijskog dijela koji se bavi opisom *gamifikacije* kao i *e-learninga* te prednostima korištenja istoga. Nakon teorijskog dijela slijedi analiza ankete koja će ispitati stavove i mišljenja korisnika prije korištenja web rješenja.
Praktični dio prati izradu web aplikacije koja će implementirati sve navedeno, gdje će korisnici moći testirati aplikaciju, čime bi se potvrdila ili odbacila hipoteza o većoj motiviranosti koristeći ovakav pristup.
Kao rezultat diplomskog rada dobiva se gotova web aplikacija gdje će studenti moći rješavati testove vezane uz kolegije.

Ključne riječi: gamifikacija, e-learning, motivacija, studenti

Abstract

The topic of this master thesis is to demonstrate the possibility of applying gamification in the form of an e-learning solution that follows the courses held at the Algebra University College. The hypothesis of this paper is to try to prove how a designed web application can increase engagement in learning, increase understanding of certain courses and motivate students.
The work consists of the introductory theoretical part dealing with the description of gamification as well as e-learning and the advantages of using both. After the theoretical part comes a survey analysis that will examine users views and perspective before using the web solution.
In the practical part, a web application will be developed that implements everything listed, where users can test the app in order to confirm or reject the hypothesis of greater motivation using this approach.
As a result of this master thesis, a complete web application is provided where students will be able to solve exercises related to the course.

Keywords: gamification, e-learning, motivation, students

Sadržaj

1. Uvod

1.1. Motivacija iza projekta

Nakon završetka preddiplomskog studija na Grafičkom fakultetu, osim upisanog diplomskog studija na istome, odvažila sam se upisati i programski smjer na Algebri. Kako bih nadoknadila predznanje koje mi je potrebno kako bi se završio upisani smjer, učila sam na raznim besplatnim platformama poput HackerRank-a, Code Academy-a i slično. Ono što im je zajedničko i što ih čini dobrim jest podučavanje na zabavan način. Code Academy idealan je za početak učenja, sadrži niz interaktivnih kvizova kako bi se bolje shvatili koncepti koje podučavaju. HackerRank ima zabavan element natjecanja, budući da se zarađuju bodovi za rješavanje određenih problema. Završetkom takvih tečajeva uvidjela sam potencijal implementacije sličnog rješenja na ovom Visokom učilištu i to na način da omogućuje učenje kolegija koje studenti slušaju.

Izvanredni studenti, kakva sam i sama, imaju nastavu tri dana u tjednu umjesto pet. Što znači da za isto opterećenje od 120 ECTS bodova, koliko iznosi diplomski smjer, izvanredni studenti moraju biti puno samostalniji i aktivniji u ispunjavanju obaveza na učilištu. Također studentima koji upisuju diplomski studij sa malo ili vrlo malo predznanja vrlo brzo ponestane motivacije kada uvide koliko im znanja potencijalno 'nedostaje' naspram ostalih kolega.

Uz takvo slično iskustvo, te uz znanje naučenog na ovom Viskom učilištu vjerujem da ću moći napraviti *gamificiranu e-learning* platformu koja će pomoći prvenstveno takvim studentima, a onda i svima ostalima.

1.2. Cilj i metodologija

Ovaj diplomski rad istražuje način kako tehnike *gamifikacije* mogu pridonijeti povećavanju razumijevanja određenih kolegija, te povećati sveukupan angažman i motivaciju studenata prilikom učenja.

Metode stručnog rada koje su korištene u izradi diplomskog rada su izrada programskog rješenja te anketiranje korisnika i njihovih stavova i potreba prije i nakon upotrebe aplikacije gdje ćemo uvidjeti ukoliko je aplikacija ispunila očekivanja.

Prva metoda koja je korištena u radu je pregled literature. Vrlo je bitno poznavati teoriju koja se nalazi iza *gamifikacije* i e-*learninga* kako bi sa što više razumijevanja pristupili rješavanju problema. Sljedeća korištena metoda je anketa. Njome sam željela ispitati stavove korisnika (studenata) o korištenju *gamificiranih* rješenja u svrhu e-učenja odnosno, postoji li uopće potreba za istim.

Nakon korištenja aplikacije izrađena je još jedna anketa u svrhu potvrđivanja ili opovrgavanja hipoteze o većoj motiviranosti koristeći ovakav pristup.

2. Gamifikacija

Gamifikacija je novo kreirani termin koji se reflektira u vidu fenomena koji dolazi sa generacijom digitalno pismene populacije. *Gamifikacija* se može definirati kao korištenje mehanike, estetike i razmišljanja koje je korišteno u igrama kako bi angažirali ljude, motivirali akciju te rješavali probleme.

2.1. Uvod u gamifikaciju

Gamifikacija je možda novi termin, no sama ideja korištenja elemenata igre kao i mehanike za rješavanje problema te angažiranje publike nije sasvim nova. Ukoliko se ispravno implementira, *gamifikacija* pomaže zadržati naš interes pomoću intrinzične motivacije igrača koja se povećava sa mehanikom i nagradama koje ih privlače. Jednostavna ideja poput igre može predstavljati neke od najjačih i naj zadovoljavajućih životnih uspomena.

Uzmimo za primjer jedenje brokule. Veliki broj djece u svijetu smatra jedenje brokule velikom problemom, čak 70% ljudi ima gen koji čini brokulu gorkom. Ova genetska adaptacija (pronađena na genu Htas2r38) je vrlo vjerojatno povezana sa činjenicom da kupusasto povrće (koje uključuje brokulu i zelje između ostalog) inhibira unos joda u štitnjaču. Stoga u okruženjima sa smanjenom količinom prirodnog joda, naša percepcija gorčine u tom povrću nas je zapravo štitila (gorak okus se najvjerojatnije razvio kako bi na neki način zaštitio ljude da ne unose previše kemijski različite hrane, odnosno toksične tvari iz bilja). Trebalo je gotovo 10.000 godina kako bi 'udomaćili' brokulu da postane sigurna za jest. No, statistika pokazuje kako prosječnom djetetu treba 12 godina kako bi zavoljeli okus brokule. Također istraživanja pokazuju ako sadržimo taj gen, i dalje ćemo percipirat taj gorak okus u ustima čak i u odrasloj dobi. Ali što ako možemo promijeniti razmišljanje djeteta da ipak pojede brokulu u manje od desetak godina? Možemo ih prisiliti da jedu povrće, no vrlo je vjerojatno da im se takav pristup neće svidjeti. Ono što roditelji rade već generacijama jest jedenje kroz igru. Drugim riječima, pretvaranje iskustva u igru- koristeći nekakvu nagradu za određeno postignuće [1].

Organizacije su posudile elemente iz igre poput bodova i bedževa kako bi ih koristili za motiviranje ljudi. Vojne organizacije koriste ovakav pristup već gotovo stoljeće. Naravno, prije je taj angažman bio ograničen samo na fizički svijet. Ono što je novo u vezi *gamifikacije* je to što pomoću modela za digitalni angažman motivacija može biti zapakirana u aplikaciju ili uređaj, napravljena na način da može uključivati publike bilo koje veličine uz vrlo mali trošak.

2.2. Primjena gamifikacije

Nedostatak tjelesne aktivnosti je sve veći izazov koji stavlja milijune ljudi u opasnost, ali ne zbog nedostatka informacija ili znanja. Mnogi ljudi znaju da bi trebali vježbati i koje su sve prednosti istoga, no nažalost takvo znanje se ne odražava u njihovom ponašanju. Na primjer, mnogi ljudi koji izlaze iz podzemne željeznice zauzimaju pokretne stepenice kako bi došli do ulične razine, ali vrlo rijetko i stubišta. Čak i kada je pokretna traka zauzeta stubišta su i dalje prazne. Takvo ponašanje se uspjelo promijeniti u jednom gradu kada su pretvorili podzemnu željeznicu, odnosno njene stepenice, u skup crno-bijelih klavirskih tipki gdje bi svaki korak napravio drugačiju glazbenu notu kada bi se nagazila. Nakon instalacije 'klavira', ponašanje se promijenilo. Ljudi su se počeli koristiti stubišta kako bi izašli iz podzemne željeznice. Korištenje stubišta se povećalo na čak 66%. [9]

Slika 2.1 Piano Staircase

Da li je moguće autoceste napraviti sigurnijima? Rješenje za to je pronašlo Švedsko nacionalno društvo za sigurnost na cestama, gdje je u Stockholmu postavila kameru pod nazivom Speed Camera Lotery. Svako vozilo koje je prošlo bilo je zabilježeno od strane kamere te ušlo u nagradno izvlačenje ukoliko je vozač prilagodilo vožnju uvjetima na cesti,

odnosno poštivao ograničenja. Prosječna brzina prije postavljanja kamere je pala sa 32km/h na 25km/h nakon postavljanja. [10]

2.3. Gamifikacija i motivacija

Gamifikacija koristi prvenstveno intrinzične, a ne ekstrinzične nagrade. Razlika između intrinzičnih i ekstrinzičnih nagrada jedan je od načina na koji možemo razlikovati *gamifikaciju* od nagradnih programa. Intrinzične nagrade održavaju angažman jer se bave ljudima na emocionalnoj razini. Ekstrinzične nagrade mogu se koristiti za motiviranje ljudi, ali ta motivacija događa se na transakcijskoj razini.
Razlikujemo tri elementa motivacije (autonomija, majstorstvo i svrha) kroz objektiv *gamifikacije* [2] :

- Autonomija- želja za usmjeravanjem vlastitih života. U djelotvornim *gamificiranim* rješenjima, igrači se odlučuju za sudjelovanje, a nakon što odluče, donose odluke o tome kako će napredovati kroz izazove kako bi ostvarili svoje ciljeve. Igračima je omogućeno otkrivanje i učenje pomoću različitih putova do rješenja. Igračima se daju alati, ciljevi i pravila te prostor za 'igranje' bez usmjeravanja na sljedeće korake koje treba poduzeti.

- Majstorstvo- želja da napredujete i postanete bolji u nečemu je važno. Svi imamo duboku potrebu za poboljšanjem u aspektima našeg života. Majstorstvo nije dostupan cilj, ono je putovanje. Na putu su mnogi putokazi koji pokazuju napredak, ali ono nikada nije krajnja točka. U gotovo svim životnim pothvatima- trčanje, slikanje ili učenje novog jezika- uvijek postoji druga razina. *Gamifikacija* se radi o postizanju boljeg u nečemu.

- Svrha- čežnja za djelovanjem. Po definiciji, *gamificirana* rješenja se razlikuju od tradicionalnih igara po svrsi. *Gamifikacija* je fokusirana na jedan ili više ciljeva: promjenu ponašanja, razvijanje vještina ili poticanje inovacije. *Gamifikacija* mora započeti i završiti sa svrhom koja je usmjerena na postizanje značajnih ciljeva igrača.

Ne iznenađuje činjenica da ljudi skupljaju bodove kako bi ostvarili pravo na recimo besplatan let. Štoviše, nelogično je ostavit 'novac na stolu'. Primarna razlika između *gamifikacije* i nagradnih programa je to što *gamifikacija* angažira ljude koji je smislen za njih.

Gamifikacija, video igre i nagradni programi slični su na nekoliko načina:

- Bave se 'igračima' dobrovoljno
- Koriste mehanike igre poput bodova i razina
- Interaktivne su
- Ugrađuju napredovanje kako bi se preselili na iduću razinu

No razlike su važnije od sličnost. Video igre, nagradni programi i *gamifikacija* angažiraju ljude na različitim razinama.
Igre se uglavnom bave igračima na hirovitoj razini kako bi ih zabavili. Nagradni programi prvenstveno se bave igračima na transakcijskoj razini kako bi ih nadoknadili. Dok *gamifikacija* se bavi igračima na emocionalnoj razini da ih motivira.
Na kraju dana, *gamifikacija* zapravo pokušava stvoriti iskustvo učenja. Ako je ono dobro implementirano, kako navodi istraživanje Karla Kappa- profesora na sveučilištu Bloomsburg, ono može povećati motivaciju za čak 51,6% .[3]

2.3.1. Teorija samoodređenja

Psiholozi već jako dugo proučavaju načine kako ljude navesti da rade nešto. U drugoj polovici dvadesetog stoljeća dominantna teorija zvala se biheviorizam. Ovaj pristup temelji se na vanjskim odgovorima na podražaje. Najpoznatije studije objavili su Ivan Pavlov na poznatom istraživanju sa psima nazvan *slavering dogs* (Pavlovljevi psi), te B.F. Skinner koji je stvorio zloglasne *Skinner boxes* koje su davale hranu ili električne šokove golubovima i štakorima. Biheviorističke studije poput ovih otkrivaju jačanje učinaka nagrade i kazne na životinjama, te su takve lekcije ekstrapolirane prema ljudima. Osnovna ideja je da ljudi i životinje reagiraju na vanjske podražaje na predvidljive načine. Bihevioristička razmišljanja ukazivala su da je ekstrinzična motivacija način koji potiče ljude da rade određene stvari. Nagrada ili kazna, sustavno primijenjena, uvjetovala bi i pojačala odgovore u očekivanju daljnjih nagrada ili kazni. Doista, to se i odražava u standardnim metodama poslovnog doba poput nagrada plaće ili bonusa.
Protiv ovakvoga pristupa su zbirke kognitivističkih teorija koje se pitaju što se zapravo događa u ljudima. Možda je najutjecajnija od njih Teorija samoodređenja (eng. *Self-Determination Theory* ili SDT) čiji autori navode kako su ljudska bića inherentno proaktivna, sa velikom unutarnjom željom za rastom na način da vanjsko okruženje podupire to, inače će

ista biti zaustavljena. Teorija samoodređenja sugerira kako potrebe spadaju u tri kategorije: kompetentnost, povezanost i autonomija. Kompetencija ili majstorstvo znači biti djelotvoran u vanjskom okruženju: učenje plesanja tanga, podnošenje porezne prijave. Povezanost uključuje društvenu povezanost i univerzalnu želju za interakcijom sa obitelji, prijateljima i drugima. Ona se također može očitovat kao želja za 'višom svrhom'. Te konačno, autonomija je uređena potreba da se svi osjećaju kao zapovjednik vlastitog života i da čine ono što im je smisleno i u skladu sa vlastitim vrijednostima.

Slika 2.2 Elementi samoodređenja

Zadaci koji impliciraju jednu ili više od urođenih ljudskih potreba biti će više motivirani. Drugim riječima, ljudi će ih činiti za sebe. Neki su primjeri očiti: svaki hobi u kojem uživamo kada god imamo slobodnog vremena, kreativne aktivnosti kao što su pisanje ili crtanje, odlazak na večeru sa prijateljima, rješavanje teške križaljke i slično. Drugi možda neće biti poput održavanja velikog sastanka ili uspješno obavljanje operacije.
Igre su savršene ilustracije lekcija iz teorije samoodređenja. Zašto ljudi igraju? Nitko ih ne prisiljava na to. Čak i jednostavna igra poput Sudoku-a aktivira intrinzične potrebe za autonomijom (što i kako ću riješiti je na meni), kompetencije (shvatio sam kako riješiti) i povezanosti (mogu uspjeh podijeliti sa prijateljima). [2]

2.4. Komponente igre

Komponente igre možemo definirati kao elemente koje susrećemo u većini igara (no ne nužno i u svima), koje su jako povezane sa igrom te imaju značajnu ulogu prilikom igranja igre. Club Psych, pokrenut 2010., popularna je web stranica koja se temelji na uspješnom televizijskom programu naziva Psych, a korisnicima nudi niz izazova poput gledanja videozapisa, odgovaranja na pitanja i pridruživanje fan klubu. Mobilna aplikacija Pshych Vision omogućuje fanovima da otkriju nagrade i međusobno razgovaraju dok gledaju emisiju na TV-u. Sve ove radnje zarađuju bodove koje se mogu iskoristiti za virtualnu ili fizičku robu. Stavke se često dodaju i povremeno uklanjaju kako bi potaknuli stalni interes. Najočitije značajke *gamifikacijskih* elemenata su bodovi (eng. *points*), bedževi (eng. *badges*) i ploče sa najboljim rezultatima (eng. *leaderboard*). Club Psych je ukomponirao te elemente vrlo efektivno: nakon prezentacije ovakve *gamificirane* stranice sveukupna prodaja na mreži se povećala za 30%, dok se pregled takve stranice povećao za 130%. [4]
Slijedi pregled *gamifikacijskih* elemenata. Ne možemo izraditi uspješan sustav igranja bez razumijevanja njihovih prednosti i mana.

2.4.1. Bodovi

Često vidimo bodove za poticanje ljudi da rade stvari prikupljajući ih. Pretpostavka je da će ljudi kupiti neko sredstvo ili raditi teže u zamjenu za bodove. Ovo je jednostavan pristup koji povremeno radi kao motivacija za ljude koji vole skupljati stvari ili za one koji se vole natjecati. Ali bodovi se mogu koristiti i na mnoge druge načine te moramo shvatiti kako ta skromna točka može poslužiti mnogim drugim funkcijama. Slijedi šest različitih načina koji se koriste u igranju [4]:

- Bodovi učinkovito 'drže' rezultat. Ovo je tipičan način na koji se koriste u sustavima igre. Bodovi kažu koliko dobro igrač igra. Netko tko je zaradio 32.768 bodova je igrao ili duže ili uspješnije od onoga tko je zaradio 24.813 bodova. Bodovi također mogu odrediti razinu (eng. *level*), primjerice morate 10.000 bodova da dostignete razinu 5 u kojem trenutku otključavate postignuće i dobivate pristup novom sadržaju.
- Bodovi mogu odrediti pobjedu u *gamificiranom* procesu, pod pretpostavkom da postoji takvo što. Ponekad želite koristiti bodove kako bi kreirali uvjet za pobjedu ako želite dati nagradu recimo.

- Bodovi stvaraju vezu između napredovanja u igri i ekstrinzičnih nagrada. Mnogi *gamificirani* sustavi nude neke fizičke nagrade za postizanje virtualnih bodova: za 1,000 bodova dobivate set noževa a za 1,000.000 bodova dobivate karte za Tahiti.
- Bodovi daju povratne informacije. Izričita i česta povratna informacija ključni su element u većini igara. Svaki bod daje korisnicima malu količinu povratnih informacija, koja govori o time koliko dobro napreduje u igri.
- Bodovi mogu biti vanjski prikaz napretka. U *multiplayer* igri, ili u okruženju u kojem članovi zajednice ili radnog mjesta mogu vidjeti međusobne rezultate, bodovi pokazuju drugima kako napredujete.
- Bodovi pružaju podatke za dizajnere igara. Bodovi koje korisnici osvoje se vrlo lako prate i pohranjuju. Na primjer, koliko brzo korisnici napreduju kroz sadržaj?

Razumijevajući prirodu bodova, možemo ih koristiti na način koji udovoljavaju ciljevima našeg gamificiarnog sustava. Imajmo na umu da su bodovi vrlo ograničeni. Oni su jedinstveni, apstraktni, međusobno izmjenjivi. Svaki dodatan bod jednostavno označava veću veličinu i ništa više.

2.4.2. Bedževi

Bedževi su vizualna reprezentacija nekih postignuća u gamificiranom procesu (naziv bedž i postignuća često se koriste kao sinonimi). Neki bedževi jednostavno označavaju određenu razinu bodova, dok neki mogu označavati različite oblike aktivnosti.
Istraživači Judd Antin i Elizabeth Churchill sugeriraju da dobro osmišljen sustav bedževa ima pet motivacijskih karakteristika [4]:

- Bedževi mogu pružiti cilj kojemu korisnici trebaju težiti, što je pokazalo da ima pozitivan učinak na motivaciju
- Bedževi pružaju smjernice o tome što je moguće učiniti unutar sustava. Ovo je važna značajka za uključivanje korisnika u sustav
- Bedževi su signal o tome što korisnika zanima. One su na neki način vizualan marker korisnikove reputacije, te će korisnici često pokušati osvojiti bedževe kako bi dokazali drugima za što su sve sposobni.
- Bedževi funkcioniraju kao virtualni statusni simbol te kao afirmacija korisničkog iskustva kroz *gamificirani* sustav.

- Bedževi funkcioniraju kao pripadni markeri. Korisnik koji želi isti bedž kao ostali korisnici imati će osjećaj pripadnosti toj određenoj grupi.

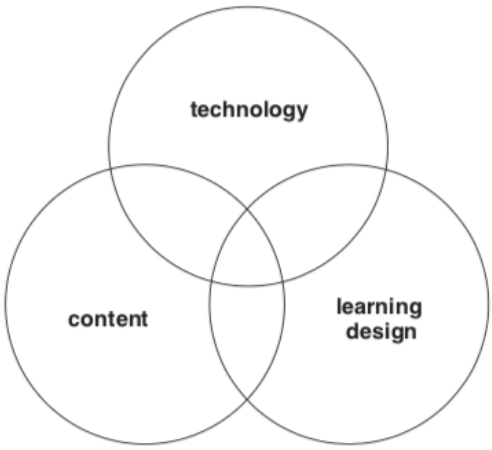
Jedna od najvažnijih atributa bedževa je njihova fleksibilnost. Mnogo različitih vrsta bedževa može se naći u različitim vrstama aktivnosti, a raspon bedževa ograničen je na kreativnost dizajnera kao i potrebe poslovanja.
U kontekstu *gamifikacije*, osvajanje bedža može biti način na koji korisnici demonstriraju određenu vještinu.

2.4.3. Ploča sa najboljim rezultatima

Ploča sa najboljim rezultatima (eng. *leaderboard*) su zadnji dio glavnih elemenata *gamifikacije*, a možda i najzahtjevniji. S jedne strane, igrači često žele znati gdje stoje u odnosu na svoje kolege. Ploča sa rezultatima daje kontekst napredovanja na način na koji bodovi ili bedževi ne mogu. U nekim situacijama, ploče sa najboljim rezultatima mogu biti snažni motivatori. Znajući da je samo još nekoliko bodova preostalo da se pojave na vrhu i to može biti snažan poticaj za korisnike. S druge strane, ploče s najboljim rezultatima mogu snažno demotivirati. Ako vidite koliko ste daleko od najboljih igrača, to može uzrokovati da se prestane truditi. Ploče na takav način mogu smanjiti bogatstvo igre. Nekoliko je studija čak pokazalo kako uvođenje ploče sa rezultatima može uvelike smanjiti performanse, a ne ih poboljšati. [4]
Postoje razni načini za izradu ploča sa najboljim rezultatom za *gamificirani* sustav. Ploča ne mora biti statičan semafor, i ne treba pratiti samo jedan atribut. U *gamifikaciji*, ploče mogu pratiti sve značajke. Ne postoji ništa krivo sa više ploča s najboljim rezultatima koje nisu univerzalne za sve sudionike.

3. E-learning

E-učenjem (u nastavku *e-learning*) možemo smatrati nešto naučeno iz YouTube videa, pronađen odgovor na pitanje sa Wikipedia-e ili sa foruma. Svaki put kada nešto naučimo sa elektroničkog izvora se smatra *e-learningom*.
Postoje tri vrlo važna koncepta *e-learninga* a ona su: omogućavanje tehnologije, učenje sadržaja te dizajn učenja. Ljudi se obično fokusiraju na prvi, na tehnologiju, jer je ona nova i nepoznata komponenta, no preostale dvije su jednako važne. Koncept učenja je proces prenošenja znanja od točke A do točke B. Učinkovito e-učenje nije samo spajanje tehnologije i sadržaja. Iako je ovo dobra formula za pružanje informacija online, samo učenje je puno više od toga.[6]



Slika 3.1 Komponente e-learninga

Možda ono najvažnije što bi trebalo imati na umu je kako je *e-learning* više od samo još jedne metode ili tehnike, poput udaljenog učenja. Ono je pristup- agregacija različitih metoda koje omogućuju tehnologije.

Organizacije su se ugledale na *e-learning* kako bi uštedjeli vrijeme obuke i troškove putovanja koje su inače povezane sa učenjem lice u lice. Međutim, uštede troškova su samo iluzija ukoliko *e-learning* ne gradi znanja i vještine učinkovito. Dio ovisi o kvalitetnom sadržaju koji je ugrađen unutar *e-learninga*. U nastavku se nalazi pet jedinstvenih značajki koje potiču učenje putem *e-learning* aplikacije.[7]

- **Prilagođena obuka**: samostalno učenje na e-learning platformi ima potencijal prilagodbe učenja za jedinstvene potrebe svakog učenika. Pod tim jedinstvenim potrebama ne misli se samo na stilove učenja, već krojenje sadržaja, metoda podučavanja i navigacije na temelju potreba pojedinih studenata.

- **Angažiranje u učenju**: bez obzira na isporučeni medij, učenje zahtjeva angažman. Pod angažmanom u ponašanju podrazumjevamo svaku akciju koju student poduzme tokom učenja na aplikaciji. Neki primjeri ponašanja u e-learningu uključuju pritisak strelice prema naprijed, upisivanjem odgovora u okvir za odgovor, klikom na opciju iz izbornika i slično. Psihološkim angažmanom podrazumjevamo kognitivnu obradu sadržaja na način koji vodi do stjecanja novih znanja i vještina. Neki kognitivni procesi koji dovode do učenja uključuju obraćanje pažnje na relevantan materijal, mentalno organiziranje te intergraciju sa odgovarajućim predhodnim znanjem. Neki primjeri metoda u e-learningu koji imaju za cilj unaprijediti psihološko angažiranje uključuju dodavanje relevantinih vizuala i prikaza na zaslonu.

- **Multimedija**: u e-learningu možemo koristiti kombinaciju teksta, zvuka kao i raznih vizuala kako bi kreirali sadržaj i pomogli studentima da dobiju adekvatno znanje i vještine.

- **Učinak stručnosti kroz scenarije**: studije stručnjaka širokog raspona domena dokazuju da je potrebni desetak godina iskustva kako bi se postigla visoka razina stručnosti. U nekim radnim postavkama, to može trajati godina jer situacije koje zahtjevaju određene vještine rijetko se predstavljaju. E-learning pruža priliku da stavlja studente u realno okruženje za posao koje zahtjeva da riješe ne tolko učestale probleme ili dovrše zadatak u nekoliko minuta, što bi moglo trajat satima ili čak danima u realnom svijetu. Simulacija na računalu primjerice može prikazati takve probleme i pružiti studentima priliku da ih rješe u stvarnom radnom okruženju.

- **Učenje kroz digitalnu igru**: pristup poznat kao *gamification*. To su simultani sustavi temeljeni na pravilima, odgovarajući na igrače, izazovni, kumulativni, omogućujući

procjenu napretka prema ciljevima i slično. Cilj gamifikacije je pružiti iskustvo učenja koje je motivirajuće, zanimljivo i učinkovito.

3.1. Tipovi e-learninga

E-learning se može podijeliti u tri glavne vrste. Sve tri vrste temelje se na korištenju instruktora, vremenu tečaja te uključenost sa drugima. Odabir odgovarajuće vrste uzima u obzir prethodno znanje učenika, brzinu učenja kao i raspoloživo vrijeme i geografsko odvojenost. Te tri glavne vrste su: sinkrono učenje, asinkrono učenje te kohortno učenje. [8]

- Sinkrono učenje se događa kada su instruktor i učenik zajedno u isto vrijeme, no ne nužno i u istom prostoru. Tradicionalne učionice su tipičan primjer sinkroniziranog učenja- učenici se nalaze u dano vrijeme, vode razgovor i uče. Sinkroni *e-learning* koristi sličan pristup. U danom periodu instruktor, te jedan ili više učenika sudjeluju putem platforme poput primjerice GoToMeeting. Ovakav format može se nazvati *webcast, webinar* ili virtualna učionica.

- Asinkrono učenje smatra se suprotnosti sinkronom e-learningu na način da učenik ima svoj ritam učenja. A ono se događa kada instruktor i učenik ne participiraju u isto vrijeme. Riječima tradicionalne edukacije, pisanje zadaće možemo smatrati asinkronim učenjem. Učenicima je dana određena aktivnost koju mogu riješiti u bilo koje vrijeme.

- Kohortno učenje ima instruktora i učenike koje izvršavaju aktivnosti poput čitanja, projekata i zadaća. Određeno je vrijeme početka i kraja, no unutar tog vremenskog okvire učenici uče i komuniciraju svojim tempom. Primjerice, u sinkronom *webinar*-u svi sudionici se priključuju u isto vrijeme, recimo da je to u 14:00 sati i sudjeluju u prezentaciji dok nije gotova do 16:00. U kohortnom modelu, učenici se prijave početkom tjedna te od tada mogu čitati materijale, određivati aktivnosti ili pričati sa drugim polaznicima u bilo koje vrijeme tokom tog tjedna.

3.2. Prednosti e-learninga

Današnji učenici žele relevantni, mobilni, samostalni i prilagođeni sadržaj. Ovakva potreba se razvila u ideju *e-learninga*. A neke od tih prednosti su sljedeće:

- *Online* učenje prilagođava se svim potrebama: online način učenja najprikladniji je za svakoga. Digitalna revolucija dovela je do značajnih promjena u pristupu, konzumiranju, raspravljanju i dijeljenju sadržaja. *Online* edukacijske tečajeve mogu proći ljudi svih profila, u vrijeme koje im odgovara.
- Lekcije se mogu proći nebrojeno puta: za razliku od nastave u učionici, sa online učenjem možemo pristupiti sadržaju neograničen broj puta. To je posebno potrebno u vrijeme revizije prilikom pripreme za ispit. U tradicionalnom obliku učenja, ukoliko ne možete prisustvovati predavanju, za određenu temu se morate pripremati sami.
- Nudi pristup ažuriranom sadržaju: glavna prednost učenja na internetu je da osigurava da ste u sinkronizaciji sa sadržajem. To omogućuje učeniku pristup ažuriranom sadržaju.
- Brza isporuka lekcija: *e-learning* je način pružanja brze dostave lekcija. U usporedbi sa tradicionalnom metodom nastave u učionici, ovakav način rada ima relativno brze cikluse isporuke, te indicira da je vrijeme učenja na takav način smanjeno na 25%-60%. Razlozi zašto se vrijeme učenja smanjuje *e-learningom* su sljedeći:
    - Lekcije započinju brzo te se nalaze u jednoj sesiji učenja. To omogućuje programe izobrazbe u roku od nekoliko tjedana, ili čak nekoliko dana
    - Učenici mogu definirati vlastitu brzinu učenja umjesto da prate brzinu cijele grupe
    - Štedi vrijeme jer učenik ne treba putovati na mjesto učenja.
    - Učenici mogu odabrati da proučavaju specifična i relevantna područja koja ih zanimaju, bez fokusiranja na svako područje
- Skalabilnost: *e-learning* pomaže u stvaranju i komuniciranju novih treninga, koncepata i ideja. Bez obzira rad li se o formalnom obrazovanju ili zabavi, *e-learning* je brz način učenja
- Konzistentnost*: e-learning* omogućuje nastavnicima da dobiju veći stupanj pokrivenosti te da komuniciraju na dosljedan način za svoju ciljnu publiku. To osigurava da svi učenici dobiju istu vrstu obuke sa ovakvim načinom učenja.

- Smanjeni troškovi: *e-learning* je isplativ u usporedbi s tradicionalnim oblicima učenja. Razlog za smanjene cijene je to što učenje na ovakav način se odvija brzo i jednostavno. Puno vremena obuke je smanjeno s obzirom na instruktore, putovanja, materijale za tečajeve i slično. Ovakva isplativost također pomaže poboljšati profitabilnost organizacije.
- Učinkovitost: *e-learning* ima pozitivan utjecaj na profitabilnost organizacije. Olakšava shvaćanje sadržaja.
- Manje utjecaja na okoliš: budući da *e-learning* ne koristi papirnati način učenja, štiti okoliš u velikoj mjeri. Prema nekim istraživanjima utvrđeno je da programi učenja na daljinu troše oko 90% manje snage i ostvaraju 85% manje emisije CO2 u usporedbi s tradicionalnim učenjem.

3.3. Elementi e-learninga

E-learning aplikacije mogu doći u raznim oblicima i formatima. No, postoji siguran broj elemenata koji su uobičajeni u većini aplikacija. Razumijevanje ovakvih elemenata pomoći će pri planiranju i analizi aplikacije. [8]

- Sučelje
    - Sučelje je vizualni okvir za svaki zaslon. To uključuje identitet (eng. *brand*), naslove, gumbe, značajke i navigaciju koja se koristi tokom korištenja aplikacije.
- Tekst
    - Tekst može biti primarni način za komuniciranje sadržaja ili može biti kao podrška za audio naraciju
- Navigacija
    - Navigacija za tečajeve omogućava studentu da se kreće kroz aplikaciju. Gumbi za navigaciju kao što su strelice, hiperveze i izbornici, usmjeravaju studente kroz tečaj. Navigacija može biti fiksna (gdje student mora nastaviti linearno od početka do kraja) ili fleksibilna (gdje student može odabrati gdje se želi kretati)

- Interakcija
    - Interakcijom smatramo svaku akciju koja zahtjeva od studenta da odgovori na neki način. Primjer toga može biti mjesto gdje student klikne kako bi mu se prikazale dodatne informacije, pitanje na koje mora odgovoriti ili slično. Interakcije pomažu u jačanju ključnih bodova i zadržavanju studenta zainteresiranog i angažiranog.
- Testovi
    - Pitanja za testove mogu biti u raznim formatima poput višestrukog odgovora, točno/netočno, popuni bjelinu, esej i slično. Neki od ovih formata (poput višestrukog odgovora) mogu biti ocjenjeni direktno unutar aplikacije, dok drugi, kao što je to esej, ne mogu. Test se može koristiti na početku tečaja, na kraju tečaja, na kraju individualnog modula ili može biti raspršeno tokom tečaja.
- Media
    - Tehnički, *e-learning* aplikacija se može sastojati samo od teksta na zaslonu. No ipak nešto zanimljivija aplikacija bi bila kada bi se koristio niz medijskih elemenata poput zvuka, videa, grafike i animacija.

4. Istraživanje prije implementacije programskog rješenja

Postoji veliki broj *gamificiranih* elemenata, te je fokus ovdje pronaći one koji će potaknuti pozitivan efekt na *e-learning* aplikaciji. Kako bi se istražila perspektiva korisnika, odnosno njihovih potreba i stavova te da li uopće postoji potreba za takvom aplikacijom, korištena je anketa. Anketa je započela sa kratkim uvodom u aplikaciju kako bi korisnici sa više razumijevanja ispunili tražene odgovore. Ciljana skupina za ovakvu anketu su studenti svih godina.

4.1. Analiza ankete i rezultati

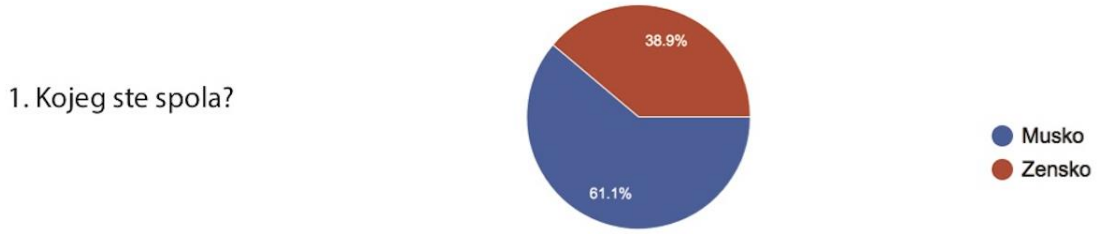
Anketa se sastoji od šest navedenih pitanja:

1. Kojeg ste spola?
2. Razina studija
3. Jeste li upoznati sa terminom *'gamification'* ?
4. Jeste li ikada koristili stranice poput HackerRank, Pluralsight, Udemy i slično? Ako da, da li su elementi poput skupljanje bodova, praćenje napretka i slično imali nekog utjecaja?
5. Koje elemente *gamifikacije* preferirate?
6. Da li biste bili produktivniji da vaše učilište ima *gamificiranu* aplikaciju za učenje?

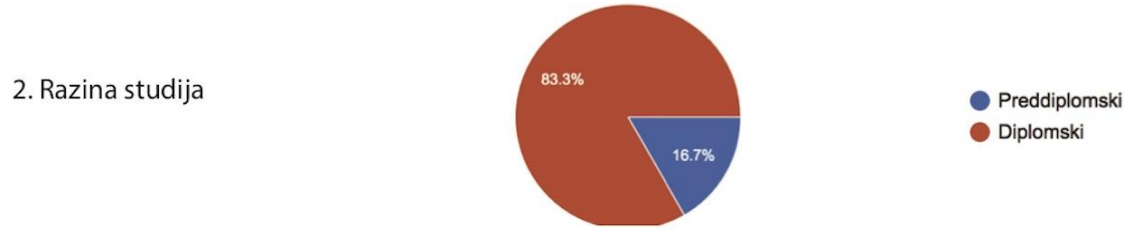
Prvo i drugo pitanje nisu obavezni za ispuniti, no oni su tu kako bismo otkrili potencijalnu korelaciju između uvriježenog mišljenja kako mlađi muškarci više igraju igrice te samim time smatraju ideju *gamifikacije* privlačnijom. Mogući odgovori na pitanje o razini studija su 'Preddiplomski' ili 'Diplomski studij'.
Treće pitanje ima moguće odgovore 'Da' ili 'Ne'. Ono nam govori o tome koliko su studenti upoznati sa terminom gamifikacija, te će rezultati toga uvelike pomoći, ukoliko je odgovor 'Da', u snalaženju u samoj aplikaciji.
Četvrto pitanje se odnosi na to ukoliko su velike aplikacije, poput iznad navedenih, imale ikakvog utjecaja na samoga korisnika. Pretpostavka je da su upravo tako velike aplikacije

primjenu *gamifikacije* napravile iznimno dobro. Odgovor na ovo pitanje je složeno na način da korisnik može napisati kratak odgovor.
Peto pitanje nam ukazuje koje elemente *gamifikacije* korisnici najviše vole vidjeti i koristiti. Mogući odgovori su: 'napredovanje na veću razinu', 'bodovi', povratni rezultati u realnom vremenu', 'traka napretka', 'bedževi', 'tablica sa rezultatima', 'avatar', 'vremenski pritisak', 'rezultat', 'biti dio priče/pripovjedački način'. Te kao posljednje šesto pitanje, potrebno je odgovoriti 'Da', 'Ne' ili 'Možda'.

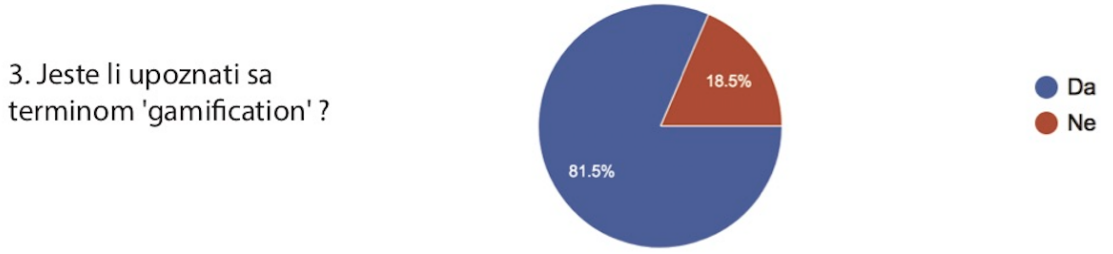
Istraživanje je obavljeno u rujnu 2018.godine, među kolegama studentima, njih sveukupno 54. Te su rezultati istraživanja sljedeći:



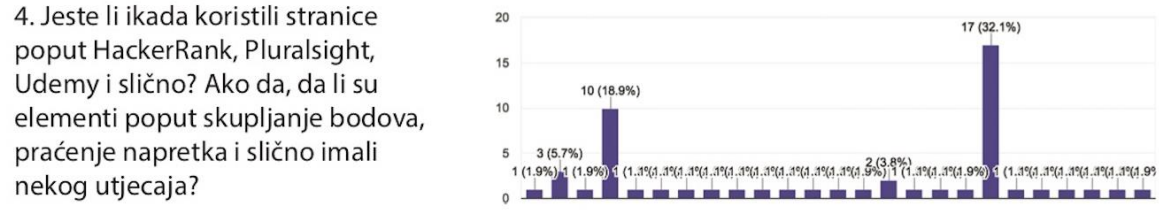
Slika 4.1 Odgovor na pitanje 'Kojeg ste spola'



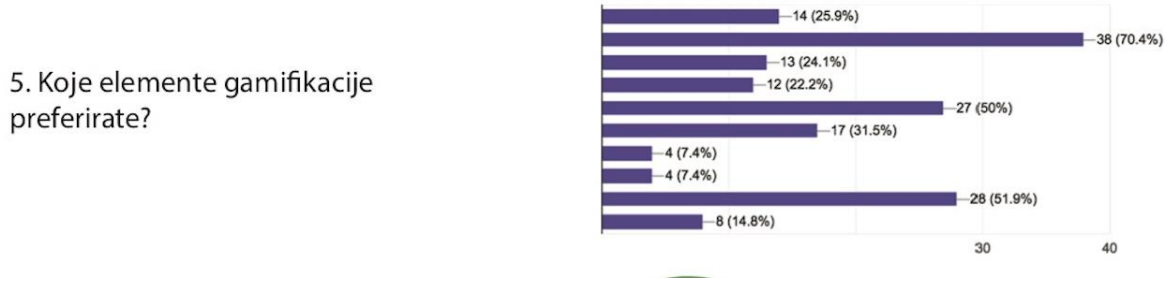
Slika 4.2 Odgovor na pitanje 'Razina studija'



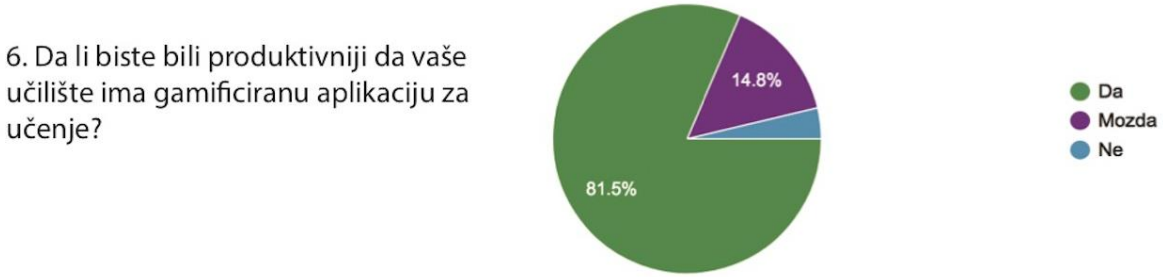
Slika 4.3 Odgovor na pitanje 'Jeste li upoznati sa terminom 'gamification"



Slika 4.4 Odgovor na pitanje 'Jeste li ikada koristili stranice poput HackerRank, Pluralsight, Udemy i slično? Ako da, da li su elementi poput skupljanje bodova, praćenje napretka i slično imali nekog utjecaja?'



Slika 4.5 Odgovor na pitanje ' Koje elemente gamifikacije preferirate?'



Slika 4.6 Odgovor na pitanje ' Da li biste bili produktivniji da vaše učilište ima gamificiranu aplikaciju za učenje?'

Iz slike 4.1 vidljivo je kako je veći dio ispitanika, njih čak 61.1% muška populacija, no na slici 4.2 nemamo dovoljan broj studenata preddiplomskog studija kako bismo mogli potvrditi korelaciju iznad navedenu između mlađih muškaraca i igranja igrica.

Na slici 4.3 potvrđujemo upoznatost studenata sa terminom *gamifikacije*. 81.5% njih je upoznato s tim terminom, ostatak od 18.5% nije.
Sa slike 4.4. nije vidljivo rješenje ankete, stoga je potrebno individualno analizirati rezultate. Osam korisnika je odgovorilo negativno na postavljeno pitanje, jedan korisnik nije odgovorio, te 45 ih je pozitivno odgovorilo. Od 45 pozitivno odgovorenih, njih 16 ih je odgovorilo sa više riječi od samo 'Da', od kojih su neka 'Da, Pluralsight, Udemy i Allison', 'Da, Trailhead sa Salesforca', 'Da, motivirajuće je', 'Da, daje mi samopouzdanje', 'Da, ima pozitivan utjecaj'.

Slika 4.5 nam daje uvid u postotak zastupljenosti određenih elemenata koje korisnici preferiraju prikazani na način od najzastupljenije komponente iz ankete:
'bodovi': 70.4%
'rezultat': 51.9%
'bedževi': 50%
'tablica sa rezultatima': 31.5%
'napredovanje na veću razinu': 25.9%
povratni rezultati u realnom vremenu': 24.1%
'traka napretka': 22.2 %
'biti dio priče/pripovjedački način': 14.8%
'avatar':7.4%
'vremenski pritisak': 7.4%

Te kao posljednje pitanje, sa slike 4.6, vidljivo je kako odgovor 'Da', bili bi produktivniji ukoliko bi učilište imalo *gamificiranu* aplikaciju za učenje, odgovorilo visokih 81.5% studenata, 'Ne' 14.8% te 'Možda' 3.7%

Iz ankete možemo zaključiti kako su studenti upoznati sa *gamifikacijom* na *e-learning* rješenjima te ih aktivno i koriste. Iz čega proizlazi zaključak, koji je ujedno i odgovor na postavljeno 6. pitanje, da bi studenti bili produktivniji ukoliko bi njihovo učilište imalo slično rješenje.

5. Implementacija programskog rješenja

5.1. Klijentski dio sustava

Za klijentski dio sustava (eng. *frontend*) koristila sam JavaScript *framework* Angular 6, te biblioteku (eng. *library*) Ngrx. Njihova sama funkcionalnost kao i primjeri korištenja iz kreirane aplikacije vidljivi su u nastavku.

5.1.1. Angular

Angular je JavaScript biblioteka koja omogućava programerima izradu aplikacija. Angular omogućava veliki broj mogućnosti koje čine implementaciju kompleksnih zahtjeva modernih aplikacija trivijalnim.
Prije najave Angulara, zaposlenik Google-a pod imenom Miško Hevery razvija projekt koji je trebao omogućiti lakšu izradu web aplikacija za par projekata na kojima je radio. Taj projekt kasnije je postao poznat kao AngularJS. Naziv Angular je dobio po zagradama < > u HTML-u. Miško i nekolicina ostalih počeli su raditi na još par aplikacija sa AngularJS-om što je dovelo do toga da ga naprave kao otvoreni projekt 2010. godine. Kada su se pojavili standardi i napredovanja u JavaScript-u, AngularJS je tu dostigao svoje ograničenje. Tim Google-a nastojao je isporučiti verziju 2.0 koja neće biti okovana na prethodni dizajn AngularJS, a to je značilo potpuno pisanje biblioteke ispočetka. [11]
Ono što je novo uvedeno od verzija 2.0+ je podrška za TypeScript[1], bazirano je na komponentama, dolazi sa Angular CLI za ubrzavanje procesa kreiranja aplikacija, sintatske promjene iz npr. ng-class u ngClass, pojednostavljenja je upotreba injekcija ovisnosti[2] (engl. *dependency injection* ili skraćeno DI), podatkovno povezivanje (eng. *data binding*) i slično [12].

1. TypeScript je superset JavaScripta
2. Injekcija ovisnosti ili DI je tehnika gdje jedan objekt opskrbljuje ovisnosti drugog objekta

Angular CLI

Angular CLI dolazi od engleske skraćenice za Command Line Interface, te kao što mu samo ime kaže ono je komandna linija za kreiranje aplikacija. Ubrzava sam proces kreiranja i konfiguriranja. Kako bi se Angular CLI mogao koristiti potrebno je instalirati Node te upisati sljedeću komandu:

```
npm install –g @angular/cli
```

-g zastava na komandnoj liniji označava da ga instaliramo u globalnom dohvatu.
Za kreiranje novog projekta koji sadržava dio koda (eng. *boilerplate*) naredba je:

```
ng new imeProjekta
```

Koristeći sličnu strukturu pisanja, možemo kreirati i ostale gradivne blokove aplikacije poput komponenti, direktiva, servisa, *interface*-a[3] i slično.

Komponente

Komponente su odgovorne za prikazivanje određenih dijelova. One definiraju logiku aplikacije. Komponente imaju @Component dekoraciju koja služi za specificiranje metapodataka. Komponenta može izgledati na sljedeći način:

```
@Component({
      selector: ‘app-landing-page’,
      templateUrl: ‘./landing-page.component.html’,
      styleUrls: [‘./landing-page.component.css’]
})
```

Ova komponenta sadrži logiku za prikaz početne stranice, odnosno dio koda koji vidimo iza poziva HTML predloška te odgovarajući CSS.

---

3. Interface ili sučelja su korisna kada trebamo provoditi neka pravila o tome kako se gradi klasa

Bitno za napomenuti je to da prilikom kreiranja komponenta pomoću Angular CLI dobivamo odgovarajuće komponente za stiliziranje odnosno ekstenzije .css, .html, spec.ts koji nam služi za pisanje testova te .ts što predstavlja TypeScript ekstenziju, koja je ustvari glavni dio u kojoj pišemo navedenu logiku. Kako za izradu ovog rada nije bilo potrebno kreiranje testnog dokumenta, komponente radimo na sljedeći način:

```
ng g c landing-page --spec false
```

zastava --spec false ne generira testni dokument, dok su g i c skraćenice za *generate component*.

Ugrađene direktive

NgFor je strukturalna direktiva, što znači da mijenja strukturu DOM-a. Njegova svrha je da ponavlja svaki dani HTML predložak jednom za svaku vrijednost u polju. Primjer korištenja nalazimo u course komponenti čija je svrha prikazati sve moguće testove za određeni kolegij.

```
*ngFor=”let test of courseTests”
```

na ovakav način prolazimo kroz svaki `test` koji smo definirali kao varijablu u `courseTests` polju, te ih ispisujemo dobiveno na način:

`[test]= “test”` ili `{{test}}`

NgIf nam služi za kondicionalno uklanjanje ili stavljanje elementa iz ili u DOM.

```
<ng-container *ngIf=”authService.isAuthenitcated()”>
```

Navedena linija iznad će pokazati sadržaj *containera* ukoliko je izraz točan. Ovaj dio koda koristi se kako bi provjerili da li je korisnik autoriziran, te ovisno o tome na drugačiji način pokazuje navigacijsku traku.

Forme

HTML forme su važni dio gotovo svake aplikacije. Korisnik ima mogućnost upisati podatke koristeći određene elemente. Kako bismo bili u mogućnosti koristiti forme potrebno ih je uključiti u aplikaciju u app.module.ts

```
Import { FormsModule } from '@angular/forms';
```

Forme su korištene prilikom izrade ekrana za *Login* i *Sign up* te na taj način dohvaćamo podatke o korisniku koje je upisao.

```
<form (ngSubmit)="onSignIn(f)" #f="ngForm">
```

koristeći `#f="ngForm"` Angular automatski stavlja ngForm direktivu na HTML elemente forme, te na ovakav način dajemo formi ime reference kako bi lakše pristupili njegovim vrijednostima. Na formi smo definirali `onSignIn` metodu kako bi klikom na gumb koji je tipa submit pozvali ngSubmit događaj.

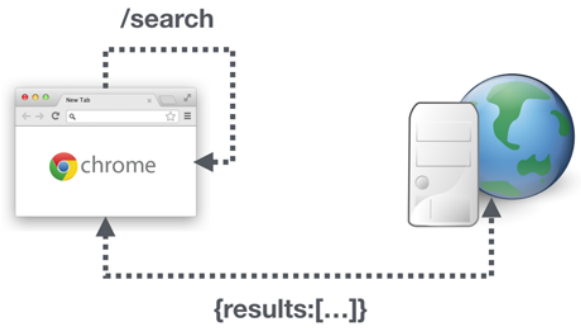
```
<button type="submit">Sign in</button>
```

Također svaki element forme mora na sebi imati ngModel direktivu:

```
<input type="email" id="email" name="email" ngModel class="form-control">
```

Kreiranje ruta

Kod *Single Page* aplikacija kao što je to Angular, implementacija ruta je nešto drugačija. Koristimo klijentske rute. Kada se URL promjeni u pretraživaču, želimo da naša lokalna aplikacija koja se vrti u pretraživaču, odnosno na klijentu, obavlja potrebnu promjenu. Ne želimo slati zahtjev na poslužitelja. Kada prvi put navigiramo na stranicu, server vrati html, javascript i css potreban za renderiranje sadržaja, sve ostale promjene u URL-u su održavane lokalno pomoću klijentske aplikacije. Tipično, aplikacija će napraviti jedan ili više API zahtjeva kako bi dobio informacije potrebne za prikaz stranice. Uvijek je jedna stranica vraćena od strane servera, sve daljnje modifikacije na stranici radi klijent i zato se zove Single Page Application.



Slika 5.1 Prikaz poziva na rutu /search u Single Page Aplikacijama

Kako bismo mogli korisiti rute prvo moramo uključiti Routes i RoutesModule

```
import {Routes, RoutesModule} from '@angular/core'
```

Mapiranje URL-a na komponente radi se pomoću tzv. Route Configuation-a, u srži to je samo polje koje definiramo na način:

```
export const appRoutes:Routes= [
      { path: '', component: LandingPageComponent, pathMatch: 'full' },
      { path: 'courses', component: CoursesComponent }
      { path: 'course/:courseName', component: CourseComponent }
];
```

Path je svojstvo koje opisuje izgled URL-a, odnosno kako želimo da se ruta obavi dok je *component* svojstvo ime komponente koje želimo prikazati kada dođemo na navedenu rutu definiranu u *path*-u.
Direktiva koju moramo pozvati za mijenjanje ruta je

```
<router-outlet></router-outlet>
```

Route Guards

Angular dolazi sa brojnim ugrađenim značajkama, od koji su jedne korisne za rukovanje sa autorizacijom. *Route guards* su sučelja[3] (eng. interface) koja moguću reći *router*-u da li trebaju dopustiti navigaciju do određene rute. Oni donose odluku tako što traže povratnu vrijednost, istinu ili laž, koja se implementira u zadano sučelje *guard*-a.

Postoji pet različitih tipova *guard*-a, od kojih se svaki zove u određenoj sekvenciji. Ponašanje *router*-a se mijenja ovisno koji guard se koristi, a oni su: CanActivate, CanActivateChild, CanDeactivate, CanLoad i Resolve.
Na mjestu gdje provjeravamo da li je korisnik registriran, možemo staviti logiku koja će vratiti *true* ili *false* za autorizaciju.

```
public isAuthenticated(): boolean {}
```

Kada znamo da li je korisnik autoriziran radimo servis koji će implementirati *route guard*, najčešća konvencija je da se ono zove auth-guard.service.

```
import { Injectable } from '@angular/core';
import { Router, CanActivate } from '@angular/router';
import { AuthService } from './auth.service';

@Injectable()
export class AuthGuardService implements CanActivate {
      canActivate() : boolean {
            if (!this.auth.isAuthenticated()) {
            this.router.navigate(['login']);
                  return false;
            }
                  return true;
      }
}
```

Kôd 5.1 Kreiranje servisa AuthGuardService koji provjerava autorizaciju korisnika

Na ovakav način, *guard* se može primijeniti na bilo koju rutu koju planiramo zaštiti od neovlaštenog dolaska na nju:

```
{ path: 'courses', component: CourseComponent, canActivate: [AuthGuard] }
```

Http API

Http klijent je servis kojeg možemo staviti (eng. *inject*) u klase unutar Angulara. On se nalazi unutar @angular/http modula zajedno sa ostalim klasama, poput Response, koje su korisne prilikom rada sa samim Http-om.

```
import { Http, Response } from '@anuglar/htpp';
```

HTTP definira set metoda kojima šalje zahtjeve kako bi se indicirala određena akcija za određeni resurs. Ti zahtjevi se zovu HTTP glagoli (eng. HTTP *verbs*) i oni su GET, PUT, POST, DELETE, HEAD i OPTIONS. Kako bismo dohvatili podatke iz baze koristimo:

```
fetchData() {
      return this.http.get(`${DATABASE_LINK}/data.json`)
            .map((response: Response) =>
extractDataFromResponse(response));
}
```

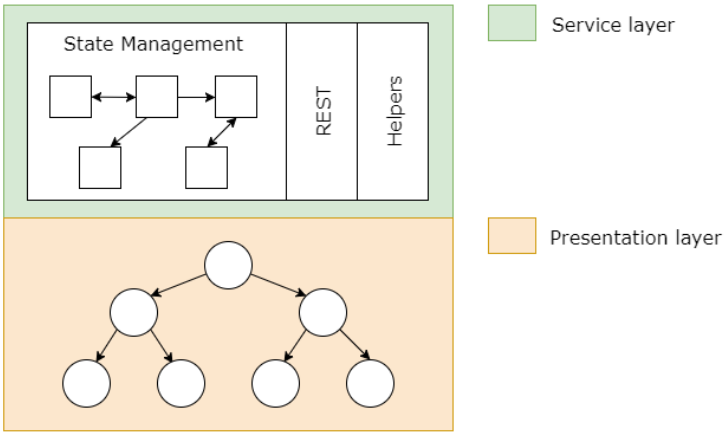
Zovemo `get` funkciju kojoj dajemo adresu baze na koju mora ići i dohvatiti podatke. Ono vraća *Observable* koji mapira dalje dobiveni odgovor u format koji nama odgovara. U ovom slučaju, funkcija `extractDataFromResponse` vraća JSON format. Na sličan način zovemo i `put` funkciju, koja sprema podatke nazad na bazu, koja ima još jedan dodatan parametar kako bi znala koje podatke treba spremiti:

```
storeData(data) {
      return this.http.put(`${DATABASE_LINK}/data.json`, data};
}
```

5.1.2. Ngrx biblioteka

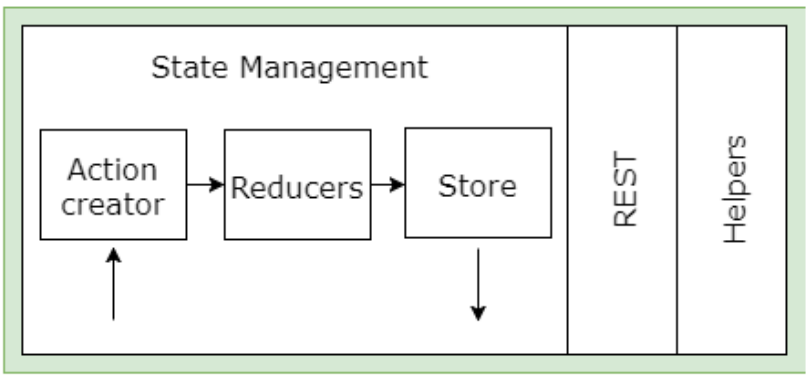
U suprotnosti sa već spomenutim AngularJS-om, Angular provodi jednosmjerni tok podataka. Na takav način se pomaže u održavanju jednostavnijih i predvidljivijih tokova podataka u aplikacijama uz značajna poboljšanja performansi. Ovakva promjena je značajna jer je uzrokovala probleme sa AngularJS-om. Angular je to poboljšao na način koji se mogu koristiti sa konvencionalnim MV* obrascima poput Reduxa, Fluxa ili MVI-a.

Spomenuti jednosmjerni tok podataka vezan je uz servisni sloj a ne prezentacijski. Na slici broj 11 je prikazana raspodjela tih slojeva. Prezentacijski sloj sadrži funkcionalnosti za prikaz podataka korisniku kroz DOM. U Angular-u takav sloj je implementiran kroz komponente. Servisni sloj implementira funkcionalnosti procesa i pohranu relevantnih podataka. Jednosmjerni tok podataka definira arhitekturu vezivanja koja se obrađuje tijekom otkrivanja promjena.



Slika 5.2 Arhitekturni slojevi, servisni i prezentacijski dio

Obrada izlaznih veza (eng. Output bindning) se izvodi van okvira otkrivanja promjena (eng. *change detection*) i time ne transformira jednosmjerni tok podatka u dvosmjerno vezanje podataka (eng. *two way binding*). Drugim riječima, promjena svojstva na roditelju koji koristi *output* mehanizam vezanja nije izveden kao dio otkrivanja promjena. Uvođenjem ngrx koji implementira redux za menadžment stanja nije relevantan za prezentacijski sloj, on je vezan na servisni sloj te arhitekturu sa slike 5.2 transformira kao sto je prikazano na slici 5.3.



Slika 5.3 State menadžment kod Redux-a

Ngrx je grupa Angular biblioteka za reaktivne ekstenzije. Ngrx/store implementira Redux obrazac koristeći RxJS Observables. Ono pruža nekoliko prednosti pojednostavljujući stanje aplikacije u obične objekte, provođenje jednosmjernog protoka podataka i još mnogo toga. Redux koncepti su sljedeći [12]:

- *Store*: na store se može gledati kao na klijentsku bazu podataka, ali ono što je bitnije, to se odražava na stanje aplikacije. Možemo ga gledati kao jedinstvenim izvorom istine.
- *Reducer*: reduktori su funkcije koje znaju što učiniti sa određenom radnjom i prethodnim stanjem vaše aplikacije. Reduktori će preuzeti prethodno stanje iz *store*-a i na nju primijeniti čistu funkciju. Čista funkcija znači da uvijek vraća istu vrijednost za isti unos bez ikakvih posljedica.
- *Actions*: akcije su potrebni podaci koji mijenjaju *store*. U osnovi, akcija ima vrstu i količinu.
- *Dispatcher*: dispečer ili pošiljatelj su jednostavna polazna točka za slanje vaše akcije.
- *Middleware*: su neke funkcije koje će presresti svaku akciju koja se šalje kako bi kreirala neku nuspojavu. Oni se implementiraju u biblioteci Ngrx/Effect.

Primjer iz aplikacije koji koristi navedeno je sljedeći: komponenta se pretplaćuje na podatke koji se nalaze u *store*-u, te promjene koje se dogode putem *reducer*-a će stići do ove komponente i onda će se učitati:

```
this.store.select<any>(fromStore.getTests).subscribe(
      tests => { this.courseTests= tests [this.courseName]; }
);
```

Spremanje na server se događa u `submitTest` funkciji, pa je to onda asinkrona akcija

```
submitTest() {
      this.store.dispatch(new
TestActions.TestMarkedAsFinished(this.activeTest));
      this.store.dispatch(new
TestActions.SubmitTestSolution(this.activeTest));
```

```
}
```

Što znači da zovemo Effect koji se nalazi u biblioteci ngrx/effect, koji nakon što spremi test u bazu šalje akciju koja označava da je test spremljen.
Effect radi na sličan način kao i Angular Service gdje koristimo @Injectable dekorator.

```
@Effect()
submitTestSolution= this.actions$
      .ofType(TestActions.SUBMIT_TEST_SOLUTION)
      .switchMap((action: TestActions.SubmitTestSolution) =>
            return this.databaseService.storeTestSolution(action.payload);
      })
```

Akcije su sinkrone, ova akcija samo miče test koji se trenutno rješava iz stanja, pošto je taj test završen

```
case TestActions.TEST_SOLUTION_SUBMITED:
      return {
            …state,
            testCurrentlyBeingSolved:[]
      };
```

JavaScript operator … zove se *spread* i koristi nam kako bi kopirali staro stanje u novo.

5.1.3. RxJs i Observables

RxJS je alat za reaktivno programiranje, dolazi od riječi Reactive Extension for JavaScipt a služi nam za implementaciju Observable-a.

Počnimo sa vrlo jednostavnim primjerom reaktivnog sustava: tablice. Svi smo ih koristili ali rijetko razmišljamo koliko su intuitivne. Pretpostavimo da imamo vrijednost u ćeliji A1. Možemo ga referencirati na druge ćelije u tablici te kada god promijenimo A1, svaka ćelija koja ovisi o A1 automatski će ažurirati svoju vrijednost. Ovakvo ponašanje ćelije je prirodno za nas. Nismo morali reći računalu da ažurira stranice koje su ovisne o A1 ili kako to učiniti, ćelije su same reagirale na promjenu. I to je ono na što cilja reaktivno programiranje. U reaktivnom programiranju, klik mišem vidimo kao kontinuirani tok događaja koji možemo pronaći i manipulirati njime. Razmišljanjem o toku događaja umjesto izoliranim vrijednostima otvara

novi način programiranja- onaj u kojem možemo manipulirati čitavim sekvencama vrijednosti koje još nisu stvorene. To se uvelike razlikuje od onoga što smo koristili, kao pohranjene vrijednosti u bazi ili polju koje čekamo da postanu aktivne prije nego li ih upotrijebimo. [13]

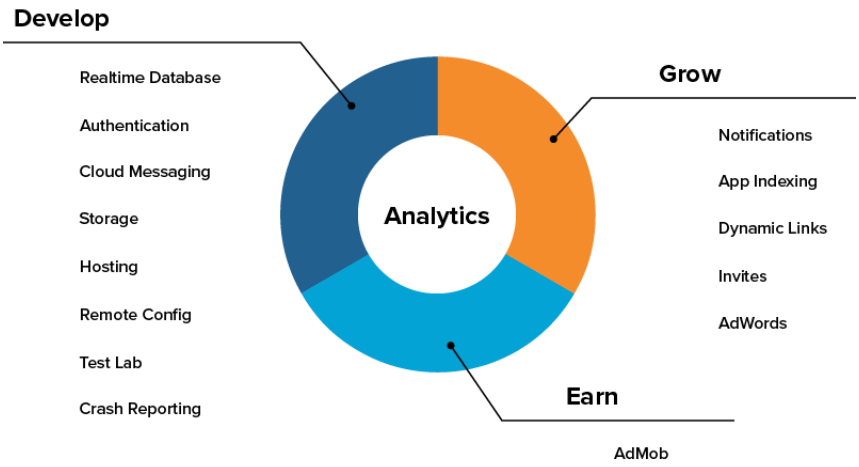
Tokovi su polja tokom vremena. Može emitirati tri različite stvari: vrijednost nekog tipa, grešku ili signal da je završeno. Navedene emitirane događaje 'snimamo' (eng. *capture*) samo asinkrono, definiranjem funkcije koja će se izvršiti kada se emitira neka vrijednost, druge funkcije za grešku i treće kada se emitira završeno stanje. Ponekad se posljednja dva mogu i izostaviti. Slušanje toka nazivamo pretplatom (eng. *subscribing*), funkcije koje definiramo su promatrači (eng. *observers*), dok je tok predmet ili *observable* koji se promatra.

5.2. Poslužiteljski dio sustava

Za poslužiteljski dio aplikacije koristilo se Google-ovo rješenje Firebase. Ono nam omogućuje pohranu i dohvaćanje podataka kao i autorizaciju korisnika.

5.2.1. Firebase

Firebase je web i mobilna razvojna platforma koja omogućava programerima razne alate i servise koji im omogućavaju kako bi razvili kvalitetne aplikacije. 2011.godine, Firebase je započeo kao razvojna tvrka (eng. *startup*) pod nazivom Envolve. Oni su omogućavali API koji se mogao koristiti za integraciju online razgovora (eng. *chat*) unutar web stranice. Nakon nekog vremena, ljudi su počeli koristiti Envolve kako bi proslijedili aplikaciji podatke koji su bili više od običnog razgovora. Programeri su to koristili za sinkronizaciju aplikacijskih podataka u stvarnom vremenu. Takvi događaji doveli su osnivače Envolva da razdvoje sustav za razgovore od arhitekture u stvarnom vremenu, te je tako u travnju 2012 nastao Firebase koji je nudio Backend-as-a-Service sa funkcionalnostima u stvarnom vremenu. Nakon što ih je kupio Google 2014.godine, Firebase je naglo narastao sa svojim multifunkcionalnostima kako za mobilne tako i za web platforme. [14]



Slika 5.4 Moguće Firebase usluge

U samome projektu se koristio Realtime Database, te će o tome biti više riječi u nastavku. Za kreiranje baze ili bilo koje usluge koju Firebase nudi potrebno je kreirati korisnički račun na web stranicama Firebase-a te odabrati opciju *Add project* koja će potom kreirati projekt. Na Firebase konzoli sa lijeve strane u izborniku vidljive su dvije zanimljive opcije vezane za ovaj rad a to su *Authentication* i *Database*.
Authentication sadrži popis svih registriranih korisnika sa datumom kada su kreirani, kada su zadnji put bili prijavljeni na aplikaciji te *User* UID koji je jedinstven za svakoga korisnika.
Database nam nudi dvije opcije: Cloud Firestore i Realtime Database. Neke od osnovnih razlika su:

- Cloud Firestore sprema podatke u formatu koji je vrlo sličan JSON-u. U Cloud Firestore-u dokumenti mogu sadržavati pod kolekcije i ugniježđene objekte što ga čini lakšim za rad u kompleksnim hijerarhijama modela
- Različiti modeli plaćanja. Realtime Database primarno naplaćuje prema zauzetom kapacitetu dok Cloud Firestore naplaćuje prema broju operacija koje napravimo

Realtime database

Firebase realtime database je NoSQL baza u oblaku. Služi za pohranu podataka u JSON obliku koji su sinkronizirani u stvarnom vremenu prema svakom spojenom klijentu. Intuitivno, baza u stvarnom vremenu se gleda kao klasična baza koja je sposobna obaviti zadatke čija se stanja trajno mijenjaju korištenjem procesiranja u stvarnom vremenu. Ona se razlikuje od tradicionalnih baza podataka po svojim performansama, vremenskim

ograničenjem koja su mikro sekundama ili čak u nano sekundama i mogućnosti da izvrši procjenu propuštenih transakcija i troškove nastale u tim propuštenim transakcijama.

Slika 5.5 Struktura Firebase baze

Slika 5.5 prikazuje strukturu baze podataka. Ona je u JSON obliku te prikazuje podatke koji se dohvaćaju i potom prikazuju unutar aplikacije.

Struktura baze je podijeljena u dva glavna čvora: *data* i *solutions*. Data nadalje ima dva svoja čvora *courses* koji predstavlja predmete koje možemo odabrati, te *tests* gdje se nalaze pitanja kada odemo na rutu recimo /test/oop/Introduction. *Solutions* bilježi odgovore, klikom na *submit* gumb i korisnika koji je autoriziran na aplikaciji.

Firebase autorizacija

Firebase nudi svoj poslužiteljski servis kako bi dopustio autorizaciju kako bismo na siguran način pohranili korisnika. Ta autorizacija se može provesti na razne načine, preko Google-a, Facebook-a, Twitter-a i slično, no isto tako nudi identifikaciju korisnika preko email-a i lozinke ili broja mobitela. Kako bismo omogućili ovu stavku možemo koristiti FirebaseUI ili Firebase Authentication SDK.

Unutar auth.actions komonente, zapisujemo akcije koje korisnik može napraviti poput:

```
export const TRY_SIGNUP = 'TRY_SIGNUP'
export const SET_TOKEN= 'SET_TOKEN'
```

također korisniku želimo dati opciju da se odjavi:

```
export const LOGOUT= 'LOGOUT'
```

svaka od tih akcija ima svoju akciju koja implementira *interface* akcije

```
export class Logout implements Action {
      readonly type= LOGOUT;
}
```

te kao posljednji korak je *export* tih tipova kako bi ih mogli jednostavno koristiti u drugim komponentama.

```
export type AuthActions = Signup | SignIn | Logout | SetToken | TrySignup | TrySignin;
```

Kada smo završili sa akcijama, prelazimo na *reducer* koji će ažurirati stanje nakon svakog događaja. U *reducer* funkciji prosljeđujemo stanje, koje je u ovom slučaju inicijalno stanje te

akciju. Unutar te funkcije radimo *switch* logiku koja je bazirana na akciji. Kada recimo pokrenemo akciju za odjavu korisnika, sa *spread* operatorom ažuriramo trenutno stanje i postavljamo vrijednosti na *null*.

```
export function reducer (state= initialState, action: AuthActions.AuthActions) {
	switch(actions.type) {
		case(AuthAction.LOGOUT):
		...state,
		user: null,
		token: null,
		authenticated: false
	}
}
```

Dolazimo do dijela koji koristi Effects koji rukuje sa asinkronom prirodom akcija. Kako bi koristili Effect stavljamo dekorator @Effect(). Efekte koje koristimo su authSignup, authSignin te authLogout.

Ngrx Effect slušaju akcije koje su poslane iz Ngrx store-a, naprave neku logiku te potom šalju novu akciju.

```
@Effect()
authSignup= this.actions$
	.ofType(AuthActions.TRY_SIGNUP)
	.map((action: AuthActions.TrySignup)=> {
		return action.payload;
	})
```

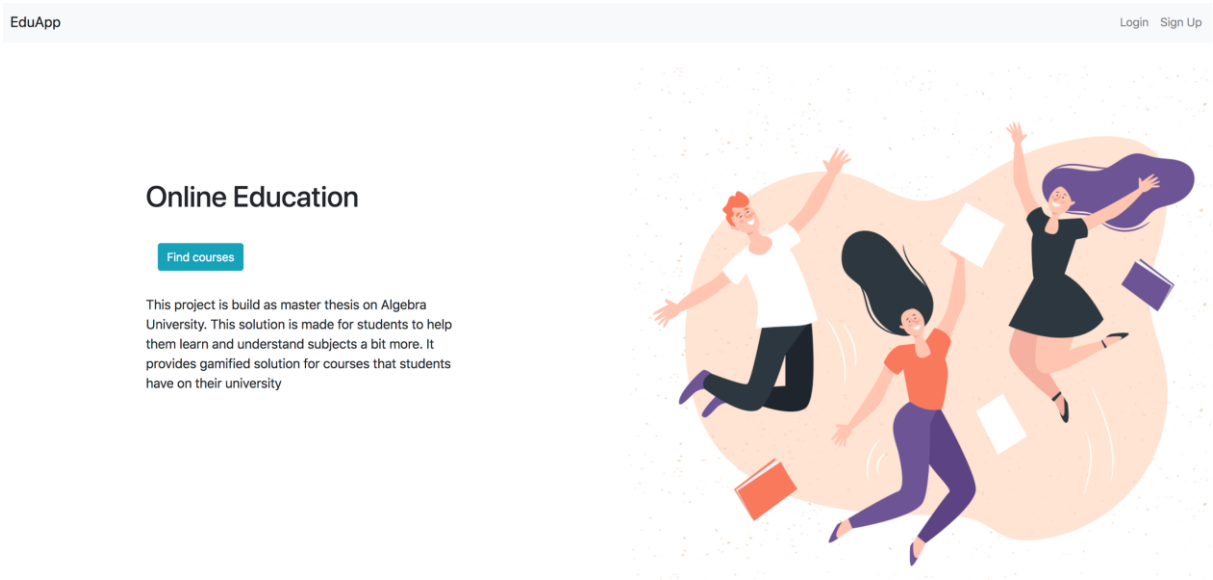
`ofType` operator filtrira akcije po tipu. Može prihvatiti više akcija, pa tako jedan efekt može imati neki broj akcija. Zatim sa `map` mapiramo akciju za njegov *payload*. Te će ona ustvari vratiti *observable*-a samo sa *payload*-om.

5.3. Grafičko sučelje i opis funkcionalnosti

Aplikacija je zamišljena na način da prati kolegije koji se izvode na Visokom učilištu Algebra. Trenutno su implementirani kolegiji objektno orijentirano programiranje te poduzetništvo. *Gamifikacijski* elementi korišteni u izradi ove aplikacije su skupljanje bodova, prikaz najboljih rezultata te povratni rezultati koji su ujedno bili među 4 najzastupljenija elementa *gamifikacije* koje studenti preferiraju. Student ima mogućnost pregleda kolegija po poglavljima koja se izvode na nastavnom planu kao i mogućnost polaganja testa. U ovome dijelu se očituje *e-learning*, odnosno učenje sadržaja kolegija, gdje kroz povratnu informaciju dobivamo odgovore na postavljena pitanja testa uz obrazloženje.

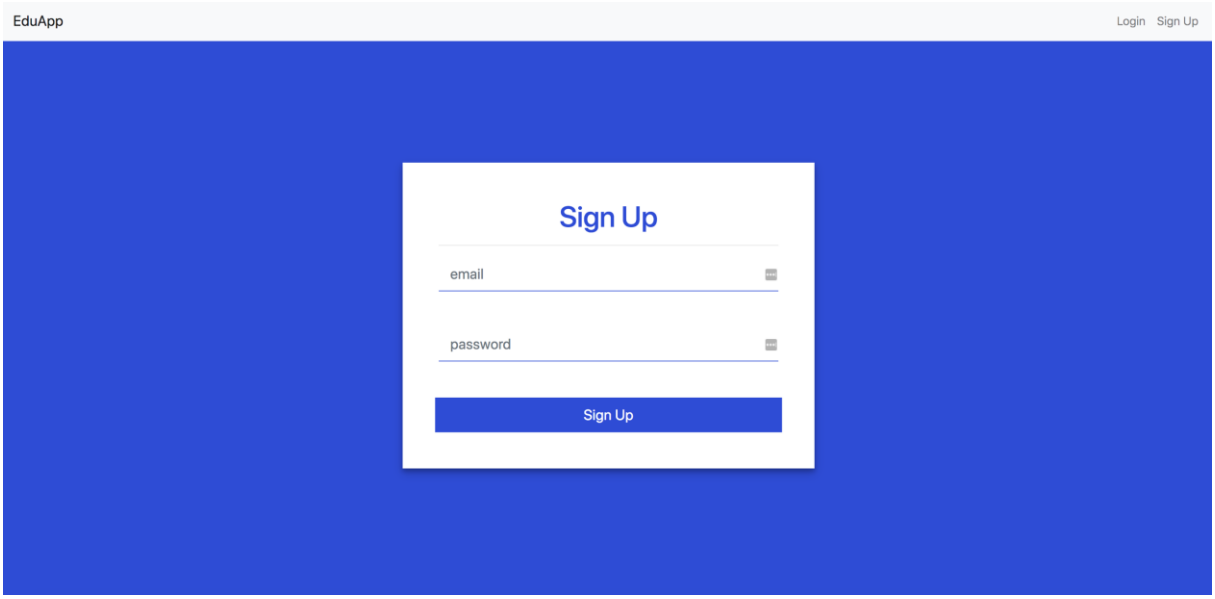
Prilikom izrade ove aplikacije vodila sam se idejom *e-learning* platforme pod nazivom Duolingo. Ona na izvrstan način kombinira učenje stranih jezika na *gamificirani* način, gdje na sličan način koji je proveden u ovoj aplikaciji, znanje se dobiva rješavanjem testova.

Odlaskom na aplikaciju pod nazivom EduApp pronalazimo početni ekran sa slike broj 5.6 koji nudi opciju *Login* i *Sign Up*.
Gumb *Find courses* je onemogućen dok se korisnik ne registrira u aplikaciju ili prijavi.



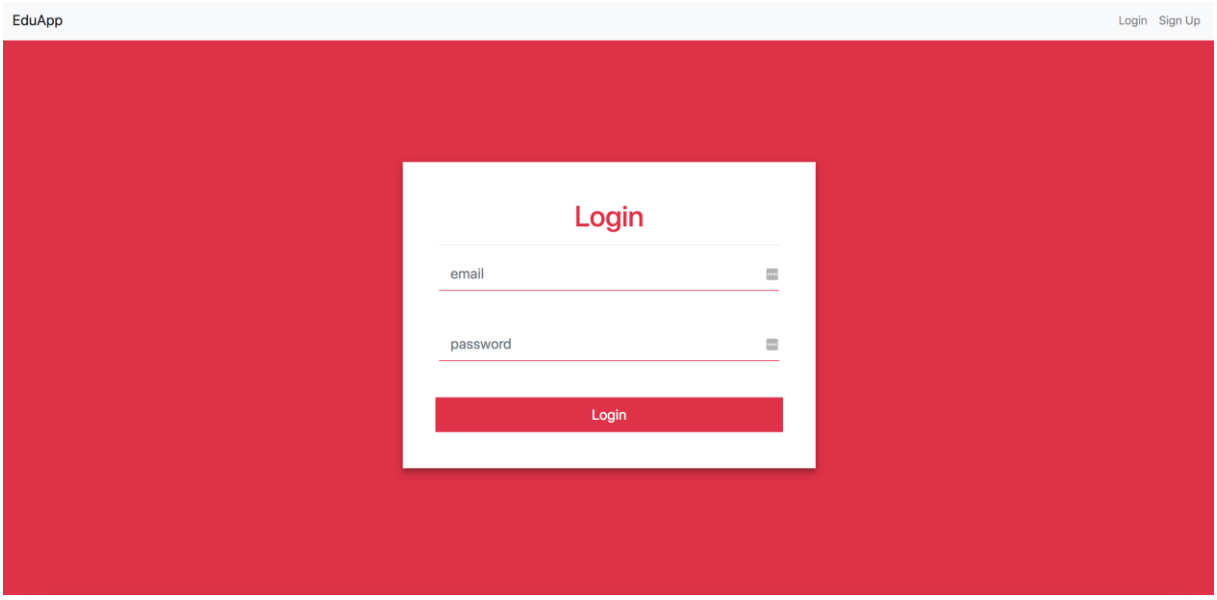
Slika 5.6 Prikaz početnog ekrana aplikacije

Prije korištenja aplikacije potrebno je upisati email i lozinku kako bismo bili autorizirani za korištenje aplikacije. Za to potrebno otići na *Sign Up* poveznicu gdje upišemo tražene podatke prikazano na slici 5.7.



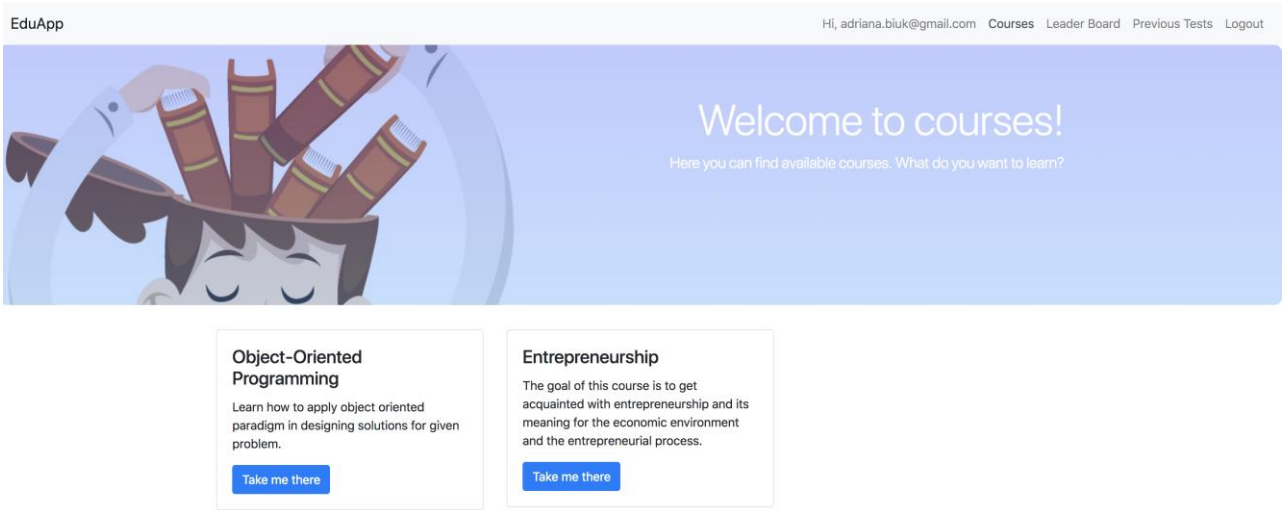
Slika 5.7 Prikaz stranice za Sign Up

Nakon uspješne registracije, te iste podatke možemo koristiti kod svake prijave prilikom korištenja aplikacije.



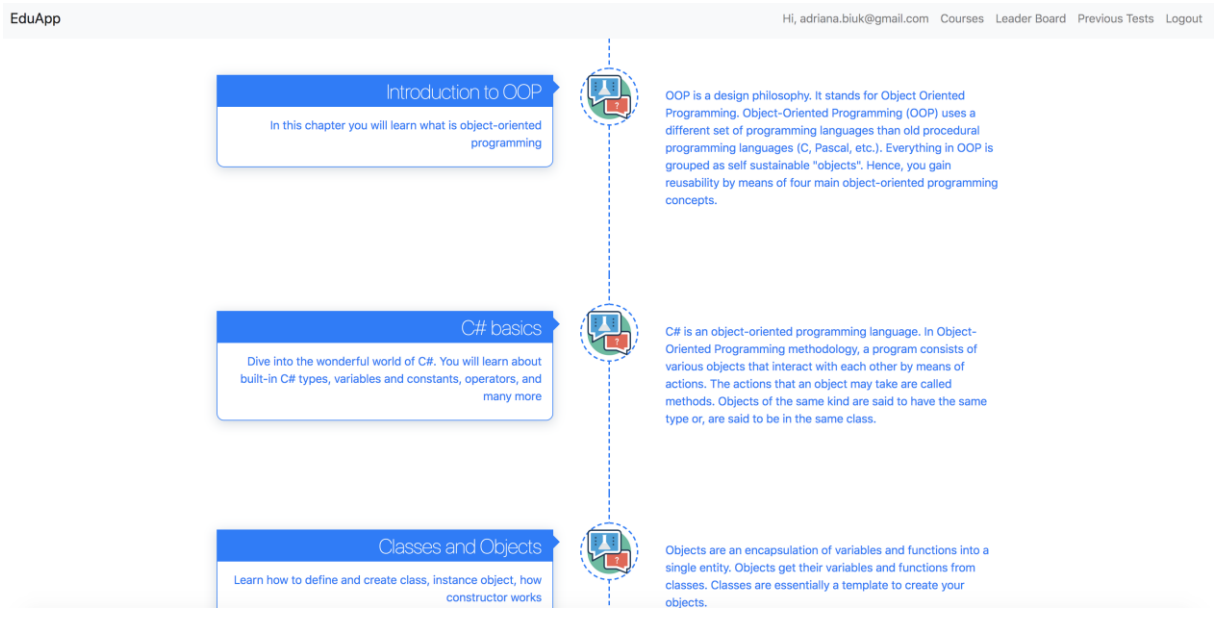
Slika 5.8 Prikaz stranice za Login

Nakon uspješne registracije, stranica vas automatski preusmjerava na *courses*. Ovdje se nalazi popis dostupnih predmeta prikazanih na slici 5.9.



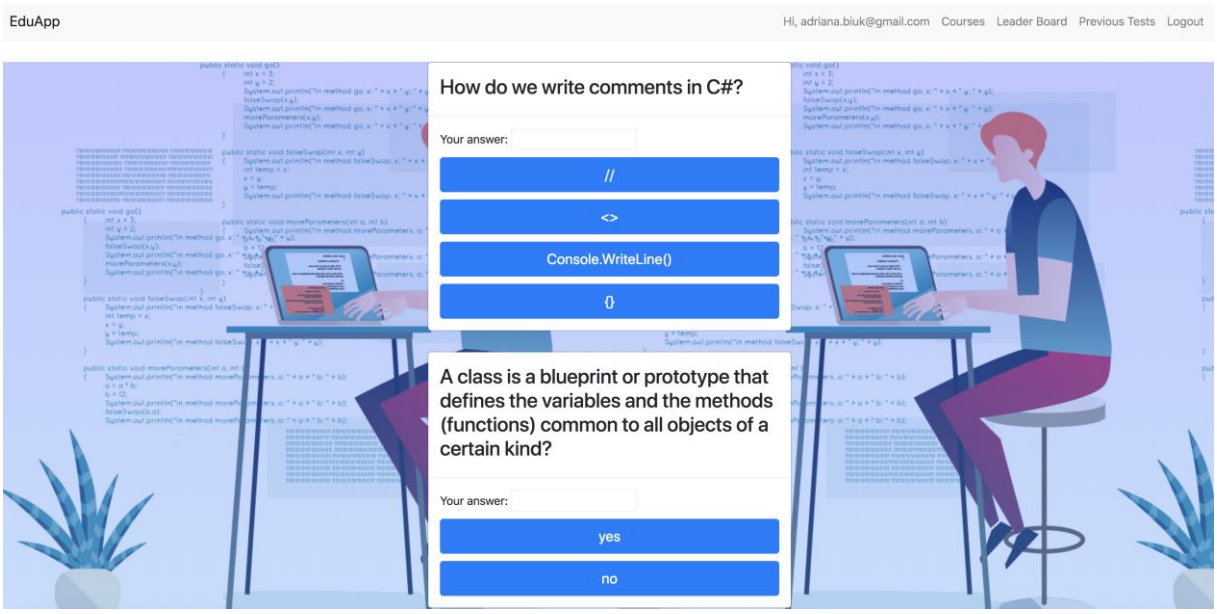
Slika 5.9 Prikaz stranice sa dostupnim predmetima

Odabirom nekog od predmeta, recimo da smo odabrali Object-Oriented Programming dolazimo na URL /course/oop čiji sadržaj je prikazan na slici 5.10.



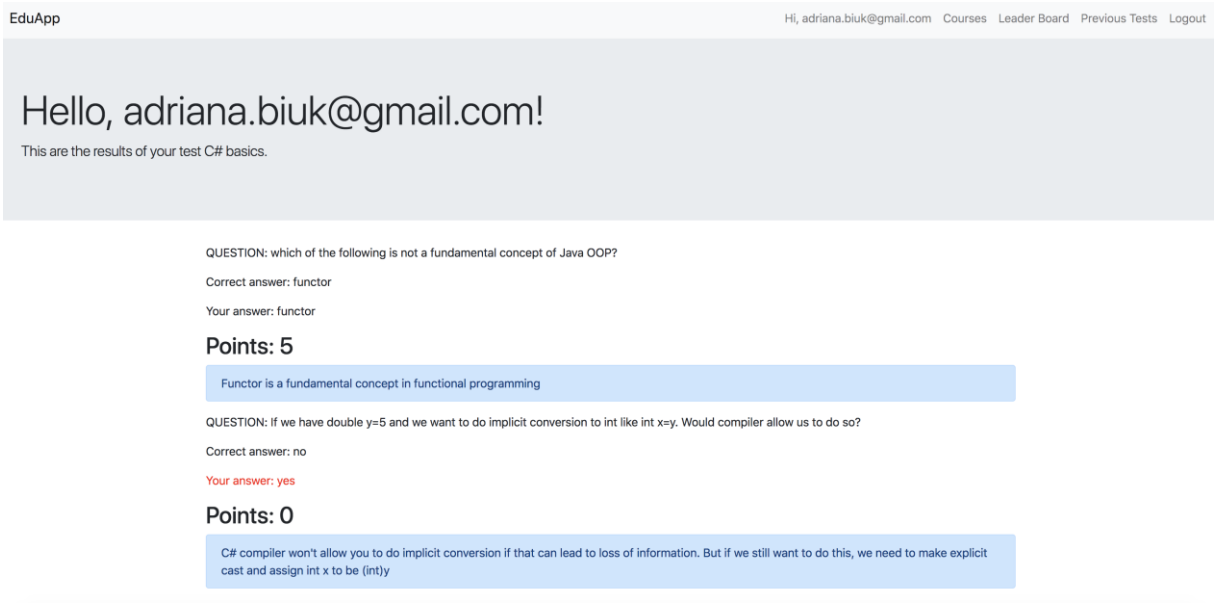
Slika 5.10 Prikaz stranice sa dostupnim poglavljima određenog predmeta

Na slici 5.10 su prikazana poglavlja koja se uče na tom kolegiju. Svako poglavlje sa lijeve strane ima popis najvažnijih pojmova koji se u njemu spominju, te sa desne strane je kratki opis naslova i što nas očekuje odabirom pojedinog poglavlja. Odabirom na neka od poglavlja dolazimo na kviz koji ima dva moguća oblika pitanja, a oni su- nadopunjavanje te odabir ponuđenih opcija.



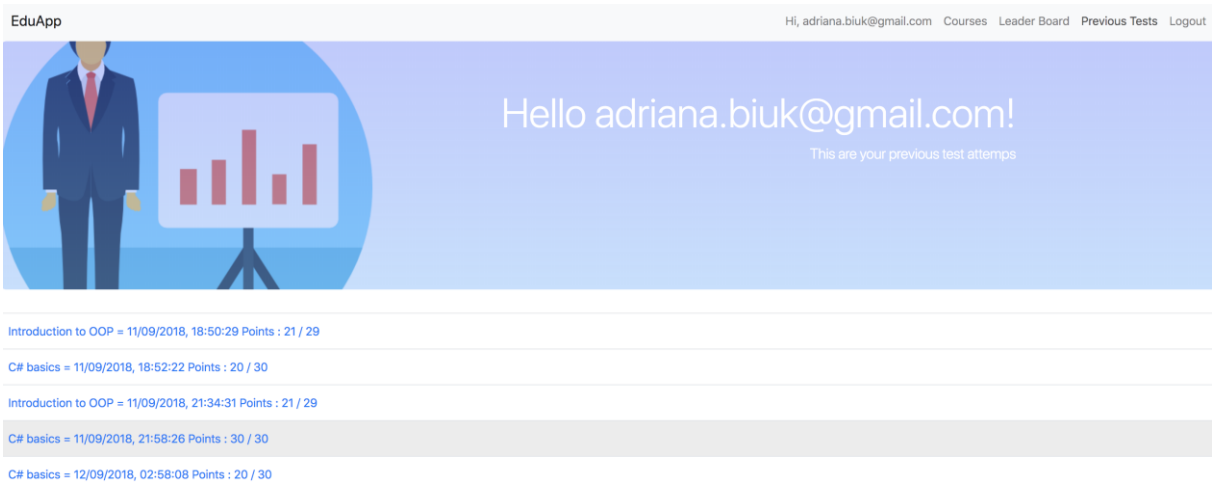
Slika 5.11 Prikaz testa aplikacije

Završetkom testa, kliknemo na *Submit Test* koji se nalazi u podnožju stranice, odnosno na kraju testa, koji nas dalje vodi na osvojene bodove te rezultate testa prikazanih na slici 5.12 Na stranici sa rezultatima prikazuju se pitanja na koja smo odgovarali, točan odgovor te odgovor korisnika koji, ukoliko nije točan, se ispisuje crvenim tekstom. Također prikazuju se osvojeni bodovi i obrazloženje točnog odgovora koje se nalazi u plavome polju.
Svako točno odgovoreno pitanje donosi određen broj bodova koji je definiran u bazi podataka, koji su u rasponu od 1 do 15 ovisno o težini pitanja. Netočno odgovoreno pitanje donosi nula bodova.



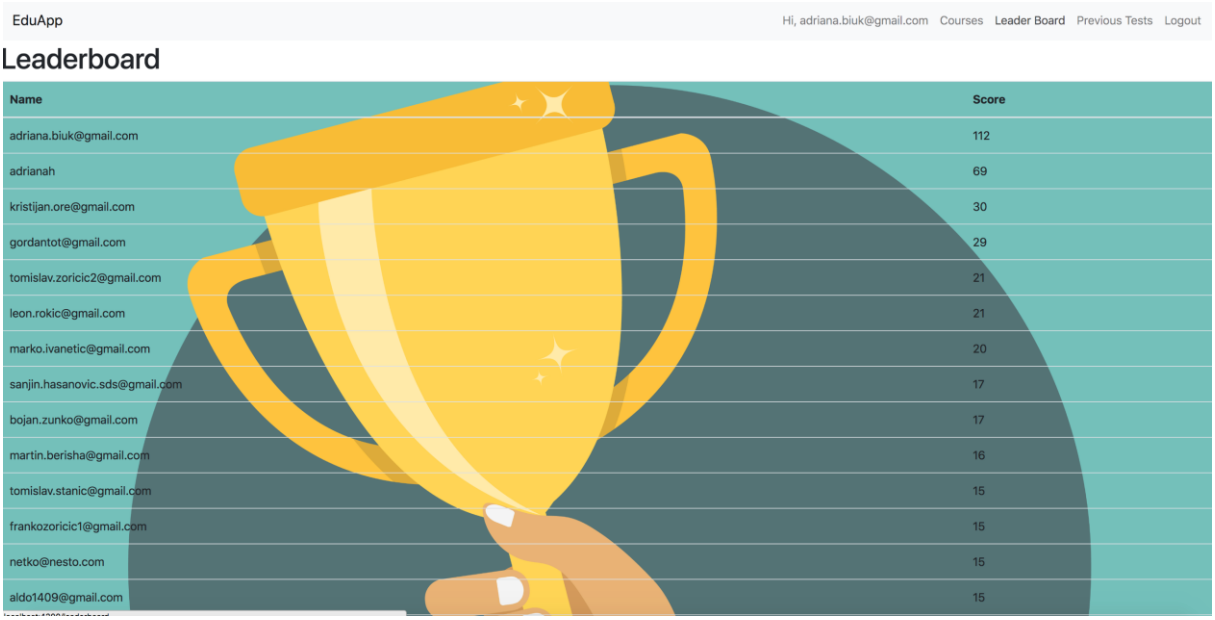
Slika 5.12 Prikaz rezultata- osvojeni bodovi te pitanja, točni odgovor i objašnjenje

U navigacijskom izborniku imamo link za *Previous Tests* (slika 5.13).
Ona pokazuje koje testove je korisnik polagao sa točnim datumom i vremenom te koliko je bodova u tom pokušaju osvojio.



Slika 5.13 Prikaz prijašnjih testova koje je korisnik polagao

Također u navigacijskom izborniku imamo link za *Leaderboard* što predstavlja ploču sa najboljim rezultatima. Ovdje se nalaze svi registrirani korisnici aplikacije sortirani po bodovima, od najviše skupljenih do najmanje.



Slika 5.14 Prikaz ploče sa najboljim rezultatima

Ploča sa najboljim rezultatima ili Leaderboard omogućava korisnicima da se usporede sa ostalim kolegama. Ploča sa najboljim rezultatima je prepoznat način natjecanja gdje je cilj kontinuirano se poboljšavat kako bi dostigao na vrh liste.

Te naposljetku imamo opciju *Logout* koja nas preusmjeri na početni ekran te odjavi iz aplikacije.

6. Analiza nakon implementacije programskog rješenja

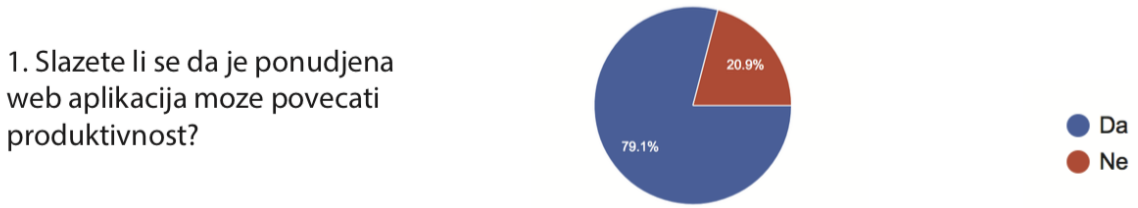
Nakon što je aplikacija u cijelosti izrađena provedena je izlazna anketa koja će nam reći o uspješnosti implementiranog rješenja.

6.1. Analiza ankete i rezultati

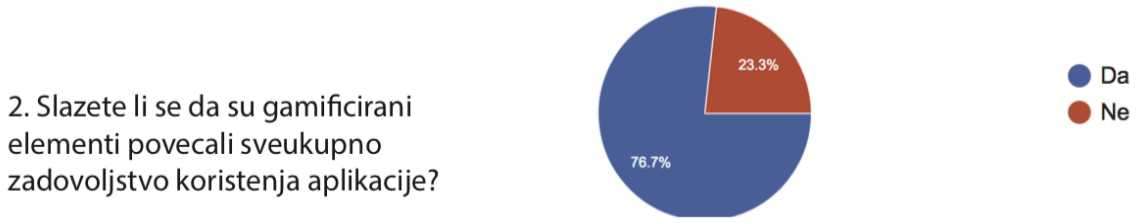
1. Slažete li se da je ponuđena web aplikacija može povećati produktivnost?
2. Slažete li se da su *gamificirani* elementi povećali sveukupno zadovoljstvo korištenja aplikacije?
3. Da li biste preporučili kolegama ovakvu aplikaciju?
4. Smatrate li da se na ovakav način bolje razumiju koncepti koji se podučavaju na učilištima?
5. Smatrate li da vam je učenje kroz ovakvu aplikacija dalo razumljive povratne informacije?

Izlazna anketa koju su studenti ispunili nakon korištenja aplikacije strukturirana je na način da su mogući odgovori 'Da' ili 'Ne'. Podučena iskustvom prijašnje ankete koja je imala prazno polje za upisati odgovor, oni su velikim djelom bili odgovoreni sa 'Da' i 'Ne'. Stoga sam se odlučila na ovakav tip izlazne ankete sa samo dva moguća odgovora gdje će se vrlo jednostavno vidjeti stavovi korisnika.

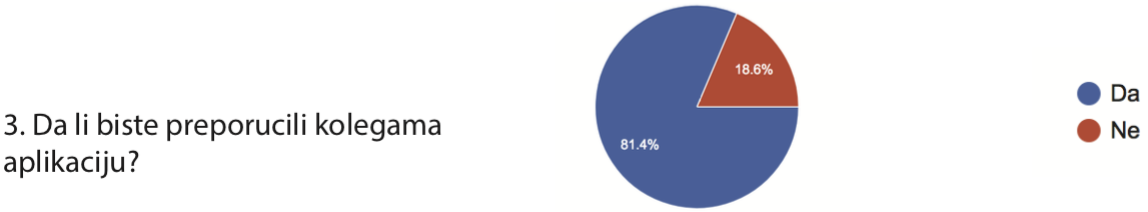
Ova anketa provedena je u rujnu također među kolegama studentima i to njih sveukupno 43. Rezultati anketa prikazani su na slikama 6.1-6.5.



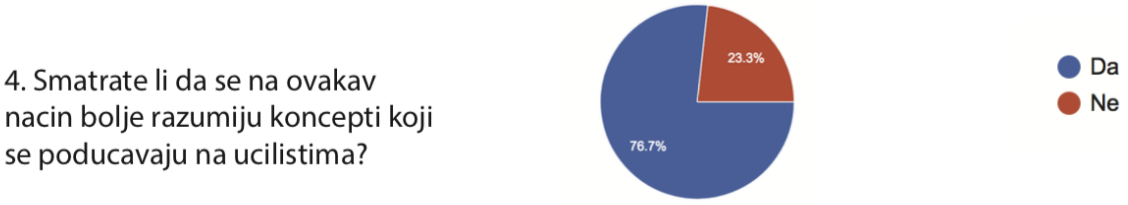
Slika 6.1 Odgovor na pitanje 'Slažete li se da ponuđena web aplikacija može povećati produktivnost?'



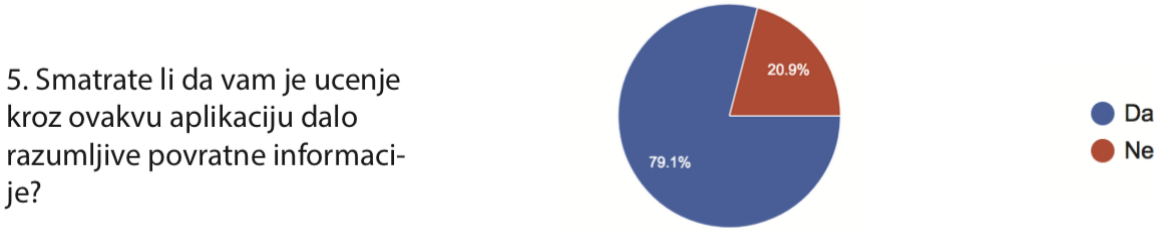
Slika 6.2 Odgovor na pitanje 'Slažete li se da su gamificirani elementi povećali sveukupno zadovoljstvo aplikacije?'



Slika 6.3 'Da li biste preporučili kolegama aplikaciju?'



Slika 6.4 Odgovor na pitanje 'Smatrate li da se na ovakav način bolje razumiju koncepti koji se podučavaju na učilištima?'



Slika 6.5 Odgovor na pitanje 'Smatrate li da je učenje kroz ovakvu aplikaciju dalo razumljive povratne informacije?'

Iz rezultata ankete vidljivo je kako su sudionici generalno zadovoljni aplikacijom.

79.1% odgovorilo je kako im ponuđena web aplikacija može povećati produktivnost.

Na pitanje da li se slažu da su *gamificirani* elementi povećali sveukupno zadovoljstvo korištenja aplikacije, njih 76.7% odgovorilo je pozitivno. Ovo je indikator kako sam *gamificirane* elemente složila smisleno, te da su upravo oni ti koji povećavaju produktivnost i motivaciju prilikom učenja za određeni kolegij.

81.4% studenata bi ovu aplikaciju preporučili kolegama. Kroz izrađenu tablicu sa najboljim rezultatima studenti se mogu međusobno natjecati u stjecanju bodova.

76.7% studenata smatra da se na ovakav način bolje razumiju koncepti koji se podučavaju na učilištima. Ovakav način je izvrstan za provjeru znanja nakon slušanja kolegija te čak i prije kako bi se studenti mogli na vrijeme pripremiti za nastavu.

79.1% studenata smatra da mu je učenje osiguralo razumljive povratne informacije. Ovaj dio je jako bitan pošto povratne informacije unutar aplikacije imaju ulogu učenja odnosno *e-learninga*.

7. Zaključak

U ovome radu obrađene su teme *gamifikacija* i e-*learning*. Obrađena je motivacija, odnosno ono što potiče ljude da odrade određenu aktivnost. Ono proizlazi iz intrinzičnih osjećaja koji vode osobu da djeluje zbog zabave ili izazova umjesto vanjskih pritisaka ili nagrada. *E-learning* se tu uklopio kao način učenja upotrebom elektroničkih medija gdje, primjenom gamifikacije, se može olakšati proces učenja.

Izrađena aplikacija lako je nadogradiva ukoliko se pokaže potreba za tim. Također manjim preinakama ove aplikacije može se omogućiti, recimo, potrebna edukacija za zaposlenike unutar neke tvrtke.

Angažiranost, uključenost i motiviranost možda i nisu toliko postignute sa *gamifikacijom* (iako rezultati ankete pokazuju suprotno) već sa oduševljenošću i zabavom koju su studenti imali dok su testirali aplikaciju. No važno je imati na umu veliko ograničenje u ovakvom rješenju, a to su- studentske obaveze. Korištenje ovakve aplikacije je na volonterskoj razini. Pojedinačne zadaće i obaveze na razini studijskog programa nisu uključene u aplikaciju.

Smatram da je budućnost edukacije i općenito učenja na ovakav način ono što nam slijedi. Studenti će imat više mogućnosti, personalizirani sadržaj, biti će manji pritisak na nastavnike i slično.

Popis kratica

SDT	*Self Determination Theory*	teorija motivacija
CLI	*Command Line Interface*	interpreter
API	*Application programming interface*	aplikacijsko programsko sučelje
SPA	*Single-page application*	vrsta web aplikacija
DI	*Dependency Injection*	programska tehnika
URL	*Uniform Resource Locator*	lokator resursa
JSON	*JavaScript Object Notation*	format podataka
DOM	*Document Object Model*	vrsta aplikacijskog programskog sučelja
RxJS	*Reactive Extensions for JavaScript*	ekstenzija, nastavak
NgRX	*Reactive Libraries for Angular*	biblioteka

Popis slika

Popis kôdova

Literatura

[1] Z.ZICHERMAN, C. CUNNINGHAM, *Gamification By Design: Implementing Game Mechanics in Web and Mobile Apps*, O'Reilly, 2011.

[2] ANDREJ MARCEWSKI, *Gamification: A Simple Introduction,* Andrej Marczewski, 2013.

[3] KERSTIN SONTS, *Gamificatin in Higher Education,* diplomski rad, Tallin University, 2013.

[4] KEVIN WEBACH, *For the Win*, Wharton Digital Press, 2012.

[5] KARL M.KAPP, *The Gamification of Learning and Instruction*, Pfeiffer, 2012.

[6] KENNETH FEE, *Delivering E-Learning*, Kogan Page Limited, 2009.

[7] RICHARD E.MAYER, RUTH C. CLARK, *e-Learning and the Science Of Instruction*, Wiley, 2016.

[8] DIANE ELKINS, DESIREE PINDER, *E-Learning Fundamentals,* Association for Talent Development, 2015.

[9] http://www.dailymail.co.uk/sciencetech/article-1218944/Scaling-new-heights-Piano-stairway-encourages-commuters-ditch-escalators.html; kolovoz 2019.

[10] https://www.wired.com/2010/12/swedish-speed-camera-pays-drivers-to-slow-down/; kolovoz 2018.

[11] https://medium.com/the-startup-lab-blog/the-history-of-angular-3e36f7e828c7; rujan 2019.

[12] http://blog.montrosesoftware.com/2017/11/30/the-history-of-angular-evolution-or-revolution/; rujan 2019.

[13] https://www.toptal.com/angular-js/ngrx-angular-reaction-application; rujan 2019.

[14] https://hackernoon.com/introduction-to-firebase-218a23186cd7; rujan 2019.