Razvoj aplikacije primjenom programskog okivra Angular.js

Marčac, Sanja

Sveučilište Jurja Dobrile u Puli
Fakultet informatike

SANJA MARČAC

RAZVOJ APLIKACIJE PRIMJENOM PROGRAMSKOG OKVIRA ANGULAR.JS

Završni rad

Pula, rujan 2018.

Sveučilište Jurja Dobrile u Puli
Fakultet informatike

SANJA MARČAC

RAZVOJ APLIKACIJE PRIMJENOM PROGRAMSKOG OKVIRA ANGULAR.JS

Završni rad

JMBAG: 2411037617, izvanredni student
Studijski smjer: Informatika

Predmet: Napredne tehnike programiranja
Znanstveno područje: Društvene znanosti
Znanstveno polje: Informacijske i komunikacijske znanosti
Znanstvena grana: Informacijski sustavi i informatologija
Mentor: doc. dr. sc. Tihomir Orehovački

Pula, rujan 2018.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisana Sanja Marčac, kandidatkinja za prvostupnicu (baccalaureus, odnosno baccalaurea) informatike ovime izjavljujem da je ovaj Završni rad rezultat isključivo mojega 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 Završnog 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.

Studentica

_____

U Puli, 18. rujna 2018. godine

IZJAVA

o korištenju autorskog djela

Ja, Sanja Marčac dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom Razvoj aplikacije primjenom programskog okrvira Angular.js 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, 18. rujna 2018. (datum)

Potpis

_____

Sadržaj

1 UVOD 1
1.1 Struktura rada 2
2 ANALIZA POSTOJEĆIH RJEŠENJA 3
2.1 Aplikacije za učenje stranih abecede 3
2.1.1 LetraKid PRO 3
2.1.2 Alphamonster 4
2.1.3 iWriteWords 5
2.1.4 LetterSchool – learn to write letters and numbers 6
2.1.5 Endless Alphabet 7
2.2 Aplikacije za učenje u hrvatske abecede 8
2.2.1 Sunčica 9
2.2.2 Učilica 9
2.2.3 CD Čitajmo zajedno 12
2.3 Aplikacija Učionica 12
3 PROGRAMSKE TEHNOLOGIJE KORIŠTENE ZA IZGRADNJU APLIKACIJE „Učionica“ 19
3.1 HTML 19
3.2 CSS 20
3.3 JavaScript 20
3.4 Angular.js 21
3.5 Elementi Angular.js aplikacije 23
3.5.1 Module 24
3.5.2 Service 25
3.5.3 Controller 26
3.5.4 Directive 28
3.5.5 Angular.js Scope 35

3.5.6 Predložak 37
3.5.7 jQuery 38
3.5.8 Alat za izradu apliakcije 39
3.5.9 Dizajn 40
4 PREDNOSTI I NEDOSTACI 42
5 ZAKLJUČAK 44
POPIS LITERATURE 45
POPIS SLIKA 46
PRILOZI 48
SAŽETAK 49
SUMMARY 50

1 UVOD

Kada u životu imate priliku živjeti u okruženju osobe koja ima teškoća u razvoju te sudjelujete u obrazovanju, razvoju i socijalizaciji takve osobe, a kroz vlastito fakultetsko obrazovanje dobijete znanje za razvoj aplikacija koje osobama s motoričkim poremećajima mogu pomoći pri učenju, razvoju i socijalizaciji, to vas motivira da napravite aplikaciju koja će im u tome pomoći.

Naime, moja brat ima disleksiju. „Disleksija je prvenstveno specifičan poremećaj razvoja govora koji smeta pri učenju riječi i čitanju unatoč normalnoj ili natprosječnoj inteligenciji, odgovarajućoj motivaciji i edukaciji, te normalnom vidu i sluhu (Medicinski priručnik za pacijente, 2014.)."

Gledajući mamu, koja je učiteljica razredne nastave, kako mom bratu pomaže u njegovom obrazovanju, motiviralo me da istražim kako se educiraju osobe, ne samo s disleksijom nego i s teškoćama u razvoju te sam prošle godine posjetila Kabinet za asistivne tehnologije Dnevnog Centra za rehabilitaciju Veruda u Puli. Tamo sam upoznala specijalisticu za asistivne tehnologije Ines Delzotto, koja mi je pojasnila što su asistivne tehnologije te pokazala postojeće asistivne tehnologije za djecu s cerebralnom paralizom.

Asistivne tehnologije predstavljaju zajednički naziv za bilo koji pomoćni ili prilagodljivi rehabilitacijski uređaj ili sustav koji pojedincu omogućuje izvođenje zadataka koje inače ne bi mogli izvesti ili povećava jednostavnost i sigurnost izvođenja zadataka. Asistivne tehnologije upotrebljavaju se kao sredstvo za pomoć korisnicima u nadilaženju infrastrukturnih prepreka, za omogućavanje potpune društvene uključenosti uz lako i sigurno izvršavanje aktivnosti.

Stoga je zadatak ovog završnog rada razvoj aplikacije primjenom programskog okvira Angular.js. Aplikacija koju sam napravila zove se „Učionica". Iako je motivacija za izradu te aplikacije bila osoba s teškoćama u razvoju, ona nije namijenjena samo djeci s motoričkim poremećajima i specijalistima za rad u edukacijskoj rehabilitaciji, tu aplikaciju mogu koristi sva djeca i učitelji prilikom obrade slova, te roditelji.

U „Učionici" obrađujemo abecedu, odnosno svako slovo zasebno. Učimo kako izgledaju velika i mala tiskana slova hrvatske abecede, kako se ona pišu te nakon što naučimo

slova kroz igru prepoznavanja ponavljamo naučeno.

Da bi se ova aplikacija mogla koristiti potrebno je prisustvo roditelja, učitelja ili specijalista za rad u edukacijskoj rehabilitaciji.

1.1 Struktura rada

Nakon uvodnog poglavlja u kojem je navedena motivacija nastanka ovog završnog radu u nastavku slijedi analiza postojećih rješenja, odnosno opis aplikacija za učenje hrvatske i engleske abecede, te njihove prednosti i nedostaci. Prikazano je i opisano sučelje aplikacije „Učionica“.

U drugom poglavlju opisane su tehnologije korištene za izradu završnog rada. Opisane tehnologije su HTML, CSS, JavaScript, Angular.js, jQuery. Također, opisan je i alat koji je korišten za izradu web aplikacije, Webstor, te tehnologija koja je osim CSS-a korištena za izradu dizajna, a to je Bootstrap. Dok je opisivan Angular.js korišteni su primjeri iz aplikacije „Učionica“.

U trećem poglavlju navedene su prednosti i nedostaci korištenja Angular.js okvira, a na samom kraju završnog rada iznesen je zaključak te naveden popis korištene literature, popis slika i prilozi.

2 ANALIZA POSTOJEĆIH RJEŠENJA

Od najranije dobi djeca se susreću s pametnim telefona i tabletima. Iako neki smatraju da djeci ne bi trebalo dopustiti da previše vremena provode na takvim uređajima, znanstvenici ne razmišljaju na takav način. Oni sve češće ističu dobre strane upotrebe tableta i pametnih telefona uz roditeljsku kontrolu koja predstavlja postavljanje ograničenja korištenja takvih uređaja, od najranije dobi jer kroz igru mogu jednostavno i lako učiti.

Na tržištu se mogu pronaći brojne aplikacije koje su osmišljene tako da ih mogu razumjeti i koristi djeca od vrtičke dobi pa sve do viših razreda osnovne škole, a i dalje. Osim toga postoje i aplikacije koje su osmišljene i za rad s djecom koja imaju određenih poteškoća u razvoju.

U nastavku su navedene aplikacije za učenje stranih abeceda i hrvatske abecede.

2.1 Aplikacije za učenje stranih abecede

Za učenje stranih abeceda, pogotovo engleske abecede ima nebrojeno puno aplikacija, te ih je bilo nemoguće sve opisati. U nastavku donosimo neke o njih koje su nam se dovoljno zanimljivima.

2.1.1 LetraKid PRO

LetraKid PRO je obrazovna igra za djecu koja sadrži deset fontova pisanja koji se obično koriste u školama, a uključuje 3 najpopularnija pisma koja se koriste u Sjedinim Američkim Državama, Kanadi i Velikoj Britaniji. Aplikacija je namijenjena djeci od 4 do osam godina. Upotreba aplikacija pomoći će djetetu da prepozna pisma, izgovor i ispravno pismo. Igra ima različite razine težine kako bi pomogla djeci da uče u malim koracima (LetraKid PRO – Handwriting ABC Review | Educational App Store, 2018.).

Prednost je ove aplikacije što je besplatna, moguće ju je instalirati na pametni telefon ili tablet koji podržavaju iOS ili Android operativni sustav. Kad dijete počne igrati ima izbor između četiri igre, a to su velika slova, mala slova, brojevi i opća forma oblika.

Nedostatak je ove aplikacije što nije dostupna za operativni sustav Windows te ju ne možete instalirati na računalo, prijenosno računalo, pametni telefon ili tablet koji ima taj operativni sustav. Osim toga nije dostupna na hrvatskom jeziku tako da je nemoguće kroz tu aplikaciju naučiti hrvatska slova.

Slika 1 - Sučelje aplikacije LetraKid PRO

2.1.2 Alphamonster

Alphamonster je aplikacija koja pomaže kod učenja slova, njihovog zvuka i riječi koje se upotrebljavaju korištenjem određenog slova. U ovoj aplikaciji učenje je moguće neovisno o odrasloj osobi jer različite aktivnosti podučavaju i pružaju praksu za djecu (Alphamonster Review | Educational App Store, 2018.).

Prednost je ove aplikacije da se može pokrenuti izgovor svakog slova, vidjeti oblik velikog i malog tiskanog i pisanog slova. Time se testira i razvija vlastito razumijevanje. U aplikaciji postoji niz aktivnosti poput podudaranja slova s riječima, pisma sa zvukovima, te

malih i velikih slova. Aplikacija je dostupna za tablete i pametne telefone sa operativnim sustavom iOS.

Nedostatak je ove aplikacije što pokriva abecedu na engleskom, francuskom, njemačkom i španjolskom jeziku, dok na hrvatskom jeziku nije dostupna. Dok smo pokušavali naći ovu aplikaciju u Google Storu kako bi istu bilo moguće isprobati dobili smo informaciju da uređaj koji koristimo nije kompatibilan s ovom verzijom.

Slika 2 - Sučelje aplikacije Alphamonster

2.1.3 iWriteWords

iWriteWords je aplikacija rukopisa u kojoj korisnica prate riječi, slova i brojeve vlastitim tempom kako bi se bolje upoznali s abecedom i brojevnim sustavom. Ta aplikacija namijenjena je rješavanju sitnih motoričkih nedostataka, pogotovo onih vidljivih kod autistične djece (iWriteWords Review | Education App Store, 2018.).

Prednost je aplikacije što je besplatna te je namijenjena mlađoj djeci te im pomaže u pisanju slova i brojeva vlastitim tempom. Crtani filmovi i čudesni zvučni efekti, kao i jednostavnost riječi, čine aplikaciju sjajnom za mlađe dobne skupine. Svrha je također jednostavna, a to je pomaganje djeci da se bolje upoznaju s osnovnim zvukovima i strukturama pojedinačnih pisama. Sliku pravi i svako slovo koje pomaže u povezivanju slova s riječima i riječi sa slikama.

Nedostatak je što takva aplikacija ne postoji za hrvatski jezik te je dostupna samo za pametne telefone, prijenosna računala ili tablete koji koriste iOS operativni sustav, dok za Android i Windows operativni sustav podrške nema.

Slika 3 - Sučelje aplikacije "iWriteWords"

2.1.4 LetterSchool – learn to write letters and numbers

LetterSchool je aplikacija koja na zabavan i zanimljiv način pomaže djeci da uče pisati slova i brojeve. Ta aplikacija vodi djecu kroz četiri koraka za svako veliko i malo slovo abecede, kao i za brojeve od nula do devet (LetterSchool – learn to write letters and numbers

Review | Educational App Store, 2018.).

Prednost je ove aplikacije ima vrlo zanimljive animacije i zvučne efekte koji potiču djecu da oponašaju način pisanja slova i brojeva. Koristi se postupak korak po korak koji je stvarno jasan i pružna izvrsnu podršku djeci. Na početku djecu uvodi lik koji će im pomoći kako naučiti pisati slova, dok je kod brojeva prikazan broj te je rečeno ime. Djeci se zatim prikazuju animacije objekata koji se broje dok ne dođu do broja koji je naznačen. Kod slova djeci je prikazan izgled i izgovor slova.

Nedostatak je ove aplikacije što je namijenjena samo za pametne telefona, tablete i prijenosna računala koja podržavaju iOS i Android operativne sustave, dok za Windows operativni sustav aplikacija nije kreirana. Također ove aplikacije u verziji za učenje hrvatske abecede nema.

Slika 4 - Sučelje aplikacije "LetterSchool"

2.1.5 Endless Alphabet

Endless Alphabet je aplikacija koja je osmišljena za učenje zvuka svakog slova te novog rječnika. Dijete povlači slovo u riječ, slovo tada dobiva osobnost te se pokrene zvuk

kada dijete točno slovo na tom mjestu u riječi (Endless Alphabet , 2018.).

Aplikacija ima izvrsnu grafiku i atraktivnu pozadinsku glazbu. Prikladna je za djecu u dobi između tri i jedanaest godina zbog grafike i sadržaja. Ta aplikacija je korisna jer se bavi fonetikom dok učite više razine rječnika, može se koristiti za rad s malom djecom koja rad na odgovarajućem slovu i njegovom zvuku.

Nedostatak je što je što nije dostupna za hrvatska slova, niti za slova bilo kojeg drugog jezika osim engleskog, te što ju je moguće koristi samo na pametnim telefonima, tabletima i prijenosnim računalima koja podržavaju iOS i Android operativne sustave, dok za Windows operativni sustav nema podrške.

Slika 5 - Sučelje aplikacije "Endless Alphabet"

2.2 Aplikacije za učenje u hrvatske abecede

Traženjem edukativnih igara i aplikacija na hrvatskom tržištu uvidjeli smo kako ih nema mnogo. Naime, hrvatsko tržište još nije toliko razvijeno po pitanju edukativni igara, te u nastavku donosimo samo neke edukativne igre.

2.2.1 Sunčica

Sunčica je obrazovna igra za djecu u dobi od četiri do osam godina koju je 1999. godine razvila hrvatska tvrtka „32bita“. „Lik Sunčice tijekom igre komunicira sa igračem i potiče na rješavanja zadataka. Uz Sunčicu djeca započinju osnove čitanja, rješavanja osnovnih matematičkih operacija te uče o prirodi, glazbi, svemiru, prometu, Hrvatskoj i Europi“ (Sunčica, 2013.).

Sunčica ima nekoliko igara, a to su: Sunčica Promet, Sunčica po Hrvatskoj, Sunčica Zbrajalica, Sunčica Sat, Sunčica Hvataj, Sunčica Točkalica, Sunčica Osmosmjerka, Sunčica Labirint, Sunčica Spremalica, Sunčica Brojalica, Sunčica Slovkanje, Sunčica Slagalica i Sunčica Memo.

Moglo se ju je pronaći na Google Playu, te je postojala i inačica za Smart TV uređaje. Međutim pretragom na Google Play-u ovu obrazovnu igru nije bilo moguće pronaći što i jest nedostatak.

Slika 6 - Prikaz sučelja „Sunčica“

2.2.2 Učilica

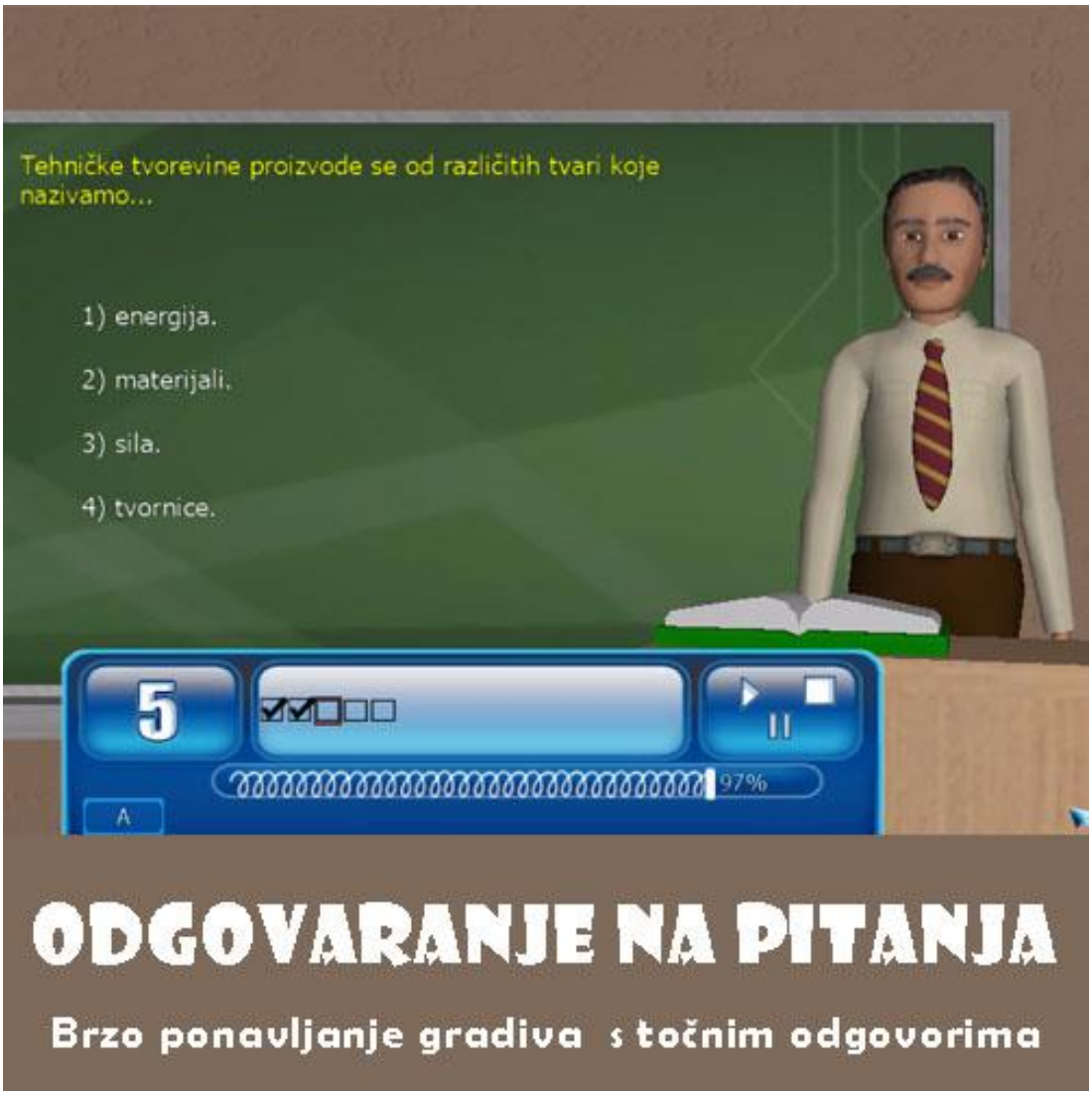
„Učilica je obiteljsko edukativni projekt čije je misija podržavati osvajanje životnih

uporišta te uklanjanje zapreka, gluposti i dosade. (Učilica, 2012.)“ To je servis koji podržava zajedništvo roditelja, djece i nastavnika kroz uzbudljivu avanturu dječjeg odrastanja i realizacije kvalitetne edukacije te adekvatnog odgoja. „Vizija Učilice je postati 'one stop shop' za djecu, roditelje i nastavnike za sve njihove sadašnje i buduće potrebe“ (Učilica, 2012.). Postoji Mala Učilica za predškolce i CD Učilica za osnovnoškolce.

Mala Učilica za predškolce namijenjena je djeci predškolske dobi od četiri do sedam godina. Sastoji se od sedam edukativnih igara na dvije težinske razine i dvije zabavne razine. Edukativne igre su: Igra Slova, Igra Brojevi, Igra Društvo, Igra Priroda, Igra Boja, Igra Promet i Igra Odnosi Među Predmetima.

Slika 7 - Prikaz sučelja "Mala Učilica" za predškolce

CD Učilica za osnovnoškolce edukativni i računalni je softver za djecu od prvog do osmog razrada, a djeca njome kroz pitanja ponavljaju najvažnije iz gradiva svih predmeta svog razreda. Akademik Josip Bratulić rekao je za ovaj edukativni softver kako se danas osnovnoškolcima nudi jedna sasvim nova metoda učenja, učenje kroz igru. Naveo je kako prema nedavnom istraživanju, čak 80% roditelja navodi da njihova djeca uče uz pomoć novi tehnologija – edukativnih CD-a i Interneta. Tu spada i Učilica, edukativni CD uz čiju pomoć učenje više nije mrsko i dosadno, već zabavno i zanimljivo. Jer ono znanje koje stječemo u igri, sa slikom, sa iskazima koji su vizualni, to znanje ostaje trajno prisutno (Učilica, 2012.).



Slika 8 - Prikaz sučelja "CD Učilica" za osnovnoškolce

2.2.3 CD Čitajmo zajedno

Hrvatska udruga za disleksiju izradila je prvi interaktivni CD Čitajmo zajedno za čitanje i pisanje na hrvatskom jeziku. Namijenjen je djeci starije vrtićke dobi te učenicima prvog i drugog razreda osnovne škole (Izdanja HUD-a – Hrvatska udruga za disleksiju, 2014.).

Prednost je ove aplikacije da učenicu kroz nju vježbaju čitanje, razumijevanje i pamćenje pročitanog, razvijaju jezične sposobnosti, a posebno se razvija motivacija za čitanje i jezično izražavanje.

Slika 9 - Prikaz naslovne strane aplikacije "CD Čitajmo zajedno"

2.3 Aplikacija Učionica

Pretražujući hrvatsko tržite, shvatili smo da je ono relativno malo te je stoga prilično ograničen izbor aplikacija na hrvatskom jeziku naspram onih na engleskom jeziku. Stoga smo odlučili izraditi aplikaciju po uzoru na one za učenje engleskih slova i abecede, kako bi nadopunili takav nedostatak. Osim toga većina aplikacija je namijenjena za iOS i Android

operativne sustave dok za Windows operativni sustav takvih aplikacija je relativno malo.

Dok smo izrađivali aplikaciju krenuli smo s pretpostavkom da dijete nikada neće samo krenuti s učenjem slova, nego će uz njega uvijek biti odrasla osoba (roditelj, učitelj, specijalist za rad u edukacijskoj rehabilitaciji) koja će mu pomoći u tome. Naime, djetetu je za učenje potreban učitelj, asistent, a posebice djeci s poteškoćama u razvoju kojima je potreban specijalist za rad u edukacijskoj rehabilitaciji te ta je aplikacija namijenjena za takav način učenja. Kada učitelj, asistent ili specijalist prikaže djetetu određeno slovo, ono vidi na ekranu kako slovo izgleda te kojim se redoslijedom slovo piše. Naime, slova se uče od jednostavnijih prema složenijim, a plan i program učenja propisuje da se u približno isto vrijeme uče slova i brojevi. Stoga ako se kreće od velikog i malog slova "I i" dovoljno je povlačenje samo jedne linije te u međuvremenu se podrazumijeva da je dijete istovremeno naučilo, kako se piše i kako izgleda broj jedan. Zato smo kod pisanja slova koristili strelice i brojeve te na taj način demonstrirali kako se slova pišu. Obrazovno je postignuće te aplikacije prepoznati velika i mala tiskana slova i naučiti pisati velika i mala tiskana slova. Dijete na taj način razvija pažnju, pamćenje, preciznost, finu motoriku, grafomotoriku te maštu.

Osim učenja, ova aplikacija može se koristi i za ponavljanje naučenog kod kuće. Dijete nakon što u obrazovnoj instituciji nauči kako određeno slovo izgleda, kako se piše te izgovara ima mogućnost uz tu aplikaciju ponoviti naučeno, odnosno kod kuće isprobati napisati slovo.

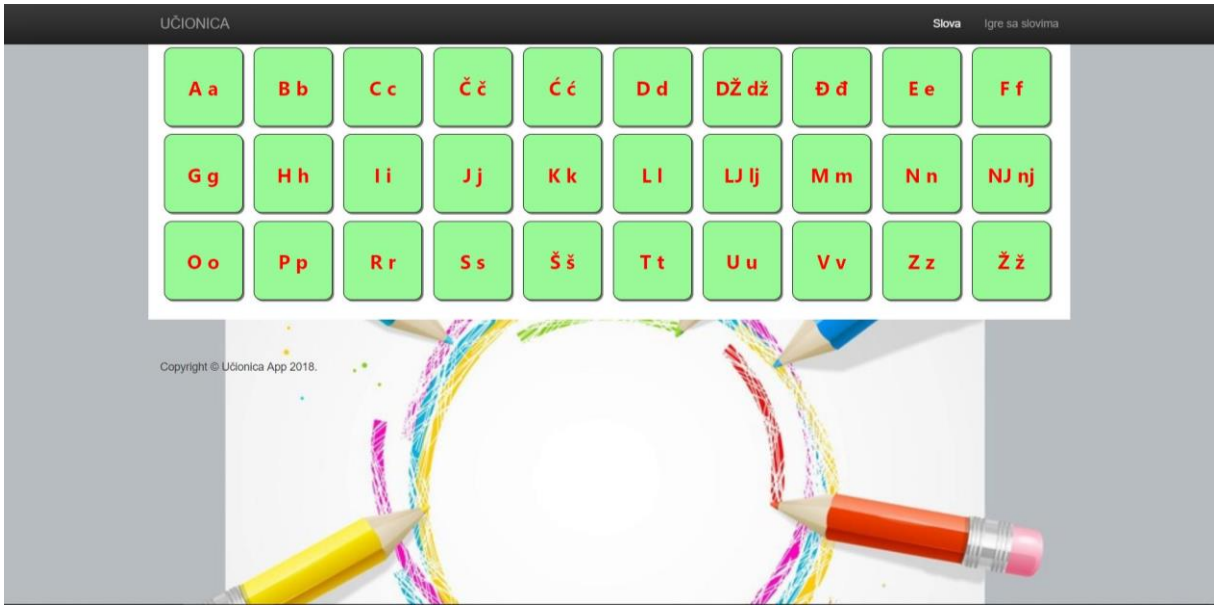
Nakon što po nastavnom planu i programu dijete nauči sva slova, djetetu je u ovoj aplikaciji omogućeno da kroz igru prepoznavanja, kod kuće ili u obrazovnoj instituciji, ponovi naučena slova.

Na taj način riješen je problem učenja, odnosno ponavljanja naučenog kod kuće. Iako djeca za učenje imaju knjige, početnice, radne bilježnice i slično, smatramo da računalo, odnosno web aplikacije predstavljaju zanimljiviji način učenja koji je djeci naprosto neodoljiv bila to djeca bez ili sa poteškoćama u razvoju.

U nastavku na slici 10 prikazana je početna stranica koja se otvara kada pokrenete aplikaciju.

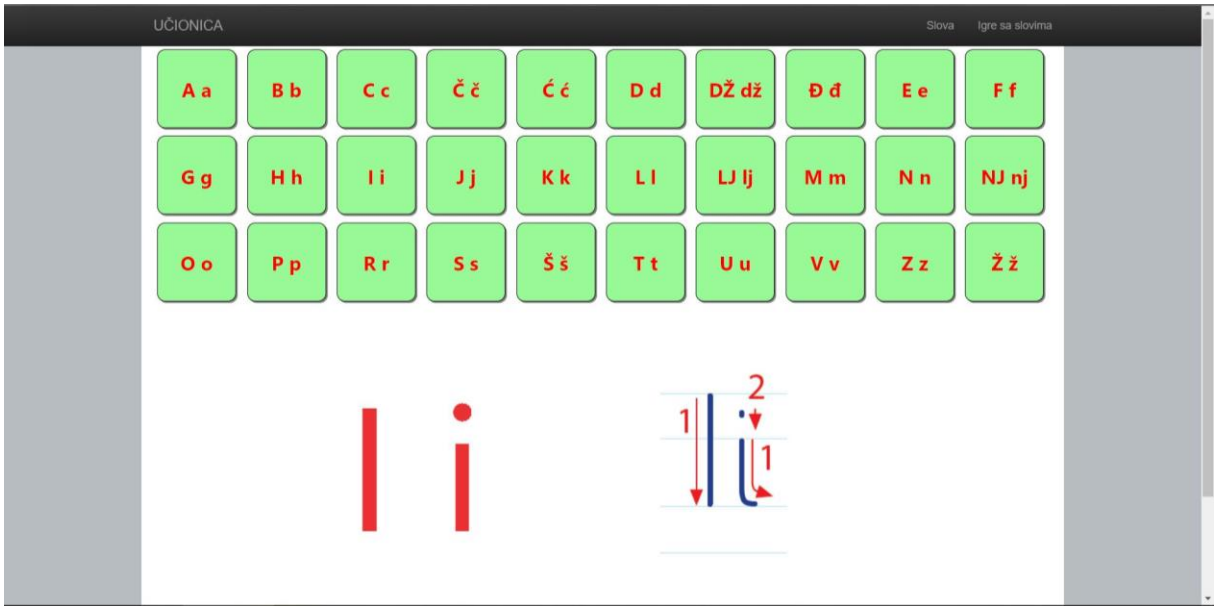
Slika 10 - Početna stranica aplikacije "Učionica"

Nakon pokretanja aplikacije kako bi se moglo započeti učenje slova potrebno je u desnom gornjem kutu u navigacijskoj traci kliknuti na gumb „Slova“ te se tada otvara prikaz koji vidite na slici 11 u nastavku.

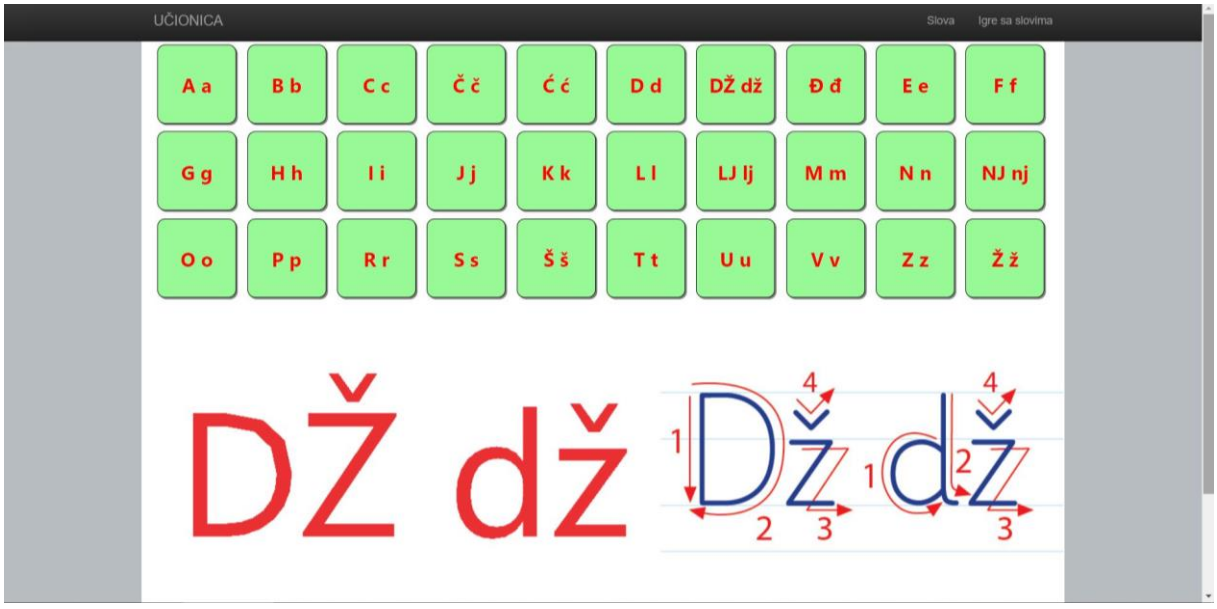


Slika 11 - Prikaz stranice za učenje izgleda slova u aplikaciji "Učionica"

Kako biste krenuli sa učenje potrebno je kliknuti na određeno slovo koje želite naučiti te se nakon toga prikazuju slike sa izgledom tog slova, te načinom pisanja kao što je vidljivo na slikama 12 i 13 . U nastavku su prikazani način pisanja jednostavnog i kompleksnog slova.



Slika 12 - Prikaz izgleda i pisanja jednostavnog slova aplikacije "Učionica"



Slika 13 - Prikaz izgleda i pisanja kompleksnog slova aplikacije "Učionica"

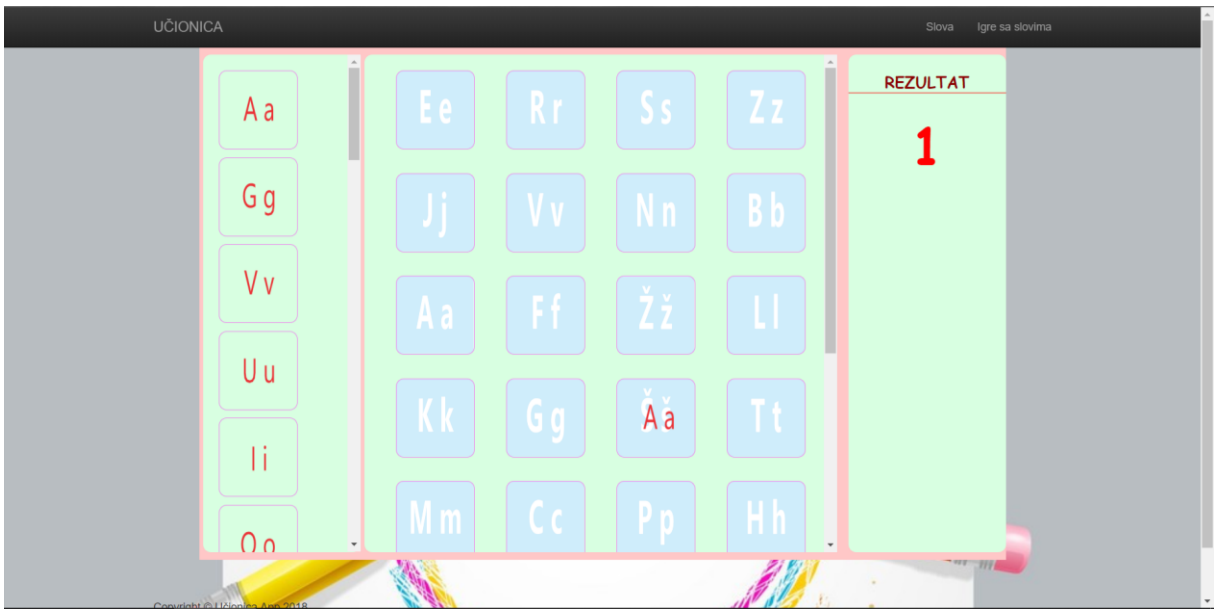
Osim gumba „Slova“ u gornjem desnom kutu na navigacijskoj traci nalazi se gumb „Igre sa slovima“. Ovaj gumb vas vodi na web stranicu u kojoj se prikazuje igra prepoznavanja slova kao što možete vidjeti na slici 14.

Slika 14 - Prikaz stranice za igru sa slovima aplikacije "Učionica"

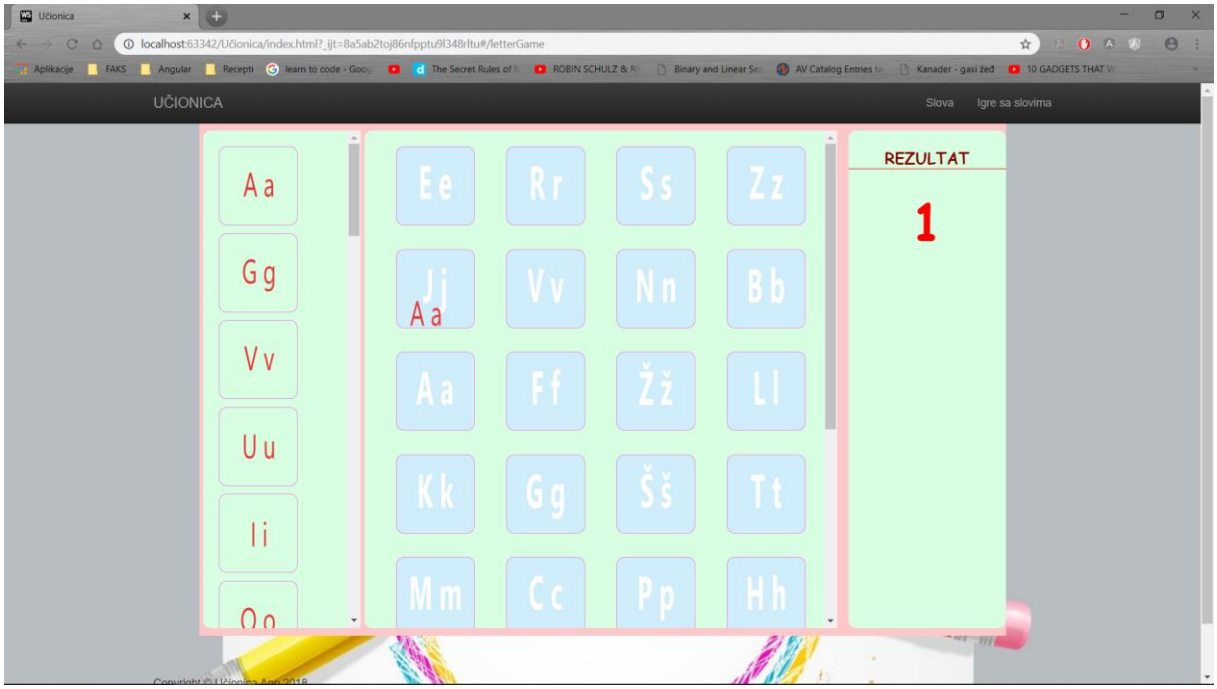
Igra radi na način da slovo koje se nalazi u lijevom stupcu povučete u srednji stupac na prozorčić u kojemu se nalazi prikaz slova koje bi odgovaralo slovu iz lijevog stupca. Ako ste pogodili slovo ostane zapamćeno, a u desnom stupcu prikazuje se povećanje rezultata sa 0 na 1 kao što je vidljivo na slici 15. Igru igrate sve dok sva slova s lijevog stupca se ne nalaze u srednjem stupcu.

Slika 15 - Prikaz pogođenog slova i povećanja rezultata u aplikaciji "Učionica"

U slučaju da pokušate slovo staviti na krivi prozorčić sa prikazom slova, on ne ostaje memorirano već se vraća u lijevi stupac sve dok ga ne postavite na točan prozorčić. Prikaz pokušaja stavljanja slova na krivi prozorčić prikazan je na slici 16, dok je prikaz vraćanja slova u lijevi stupac vidljiv na slici 17.



Slika 16 - Prikaz postavljanja slova na krivi prozorčić u aplikaciji "Učionica"



Slika 17 - Prikaz vraćanja slova u lijevi stupac aplikacije "Učionica"

Ta igra ima jednu zanimljivost, svaki puta kada ažurirate stranicu ili izađete iz igre i ponovo se vratite promijeni se i redoslijed slova u lijevom stupcu, kao i redoslijed prozorčića u srednjem stupcu. Na taj način se izbjegla mogućnost da dijete upamti gdje se koje slovo nalazi i uzastopce ponavlja isto, a postignuto je da dijete iznova mora razmišljati i tražiti sličnost kod slova u lijevom i srednjem stupcu.

3 PROGRAMSKE TEHNOLOGIJE KORIŠTENE ZA IZGRADNJU APLIKACIJE „Učionica“

Da bi mogao izrađivati web aplikacije svaki programer mora naučiti tri osnovne stvari: HTML (eng. *HyperTextMarkupLanguage*) za postavljanje sadržaja, CSS (eng. *Cascading Style Sheet*) za postavljanje izgleda te JavaScript za određivanje na koji će se način ponašati web stranica ili web aplikacija koje će biti objašnjene u daljnjem tekstu.

3.1 HTML

HTML (eng. *HyperTextMarkupLanguage*) opisni je jezik za stvaranje web sadržaja. HTML jezikom oblikuje se sadržaj, stvaraju se hiperveze (hipertekst) dokumenta. Prva verzija HTML jezika objavljena je 1993. godine, a napisao ju je Tim Berners-Lee. Od tada se HTML drastično razvio. Najkorištenije verzija tijekom 2000.-ih godina bila je HTML 4.01 verzija te je ta verzija postala službeni ISO standar. Verzija koja se danas koristi, a ujedno i posljednja verzija je HTML 5.

Svaki HTML dokument sastoji se od blokova takozvanih HTML elemenata, dok se svaki HTML element sastoji od HTML oznaka (eng. *tag*). Neki od HTML elemenata su: <body>, <button>, <tittle>, <form> i slično. Postoje dvije verzije HTML elemenata, one koje imaju otvarajuće i one koje imaju zatvarajuće oznake. Otvarajuće oznake započinju oznakom "<" (manje od), a završavaju znakom ">" (veće od), npr. <body>, dok zatvarajuće oznake započinju znakom "</"(manje od te kosa crta), a završavaju znakom ">" (veće od), npr. </body>. Također postoje i samozatvarajuće oznake kojima nije potrebna zatvarajuća oznaka (npr. <link ... />). Svaki element može imati atribute kojima se definira taj element, npr. za definiranje imena (name="..."), za definiranje klasa (class="..."), za definiranje ID-a (id="...") i slično.

HTML-om se ne izvršavaju nikakvi zadaci, jer nije programski jezik, već se samo opisuje sadržaj web stranice te se na taj način mogu stvarati tablice, mijenjati boje, fontovi pojedinih elemenata i odvajati odlomci.

3.2 CSS

CSS (eng. *Cascading Style Sheet*) stilski je jezik koji se koristi za opis prezentacije dokumenta napisanog pomoću HTML jezika. CSS opisuje kako elementi trebaju biti organizirani na ekranu, papiru ili na drugim medijima. Poput HTML-a, CSS zapravo nije programski jezik, to je jezik stila. To znači da omogućuje selektivno primjenjivanje stilova elemenata u HTML dokumentima. Kako se razvijao HTML u početku su u njega ubacivane oznake (eng. *tagovi*), no uočena je potreba za stilskim jezikom na web stranici, odnosno HTML dokumentu, te je na taj način nastao CSS.

Korištenjem CSS-a postalo je moguće odvojiti prezentacijski dio podataka i dizajn od same strukture podataka. Sintaksa CSS je da su stilski obrasci sačinjeni od stilskih pravila. Svako pravilo ima svoja dva dijela, a to su selektor koji određuje element na koji se stilsko pravilo odnosi te deklaraciju koja određuje kako izgleda sadržaj opisan CSS-om. Deklaraciju dijelimo još i na dvije stavke, a to su propertis, odnosno aspekti kako da računalo prikaže tekst i grafiku te values, tj. podaci koji određuju kako želimo da tekst i slike izgledaju na našoj stranici (Introduction to CSS - Learn web development | MDN, 2018.).

3.3 JavaScript

JavaScript skriptni je jezik koji omogućuje implementaciju složenih stvari na web stranicama. Svaki puta kad web stranica ne prikazuje statične informacije, već prikazuje pravodobno ažuriranje sadržaja, interaktivne karte, animirane 2D/3D grafike i slično, tada se može zaključiti kako je uključen JavaScript.

Jezgra JavaScript jezika sastoji se od nekih uobičajenih značajki programiranja koje vam omogućuju sljedeće:

- pohranjivanje korisnih vrijednosti unutar varijabli,
- operacije nad dijelovima teksta (poznato kao "nizovi" u programiranju),
- kod koji neprestano radi kao odgovor na određene događaje koji se pojavljuju na stranici, ...

Kada učitate web stranicu u pregledniku, pokreće kod (HTML, CSS i JavaScript)

unutar izvršne okoline (kartica preglednika), što sliči na nekakvu tvornicu koja uzima sirovine (kod) i izlaže proizvod (web stranicu). JavaScrip provodi JavaScriptov motor preglednika, nakon što su HTML i CSS sklopljeni i postavljeni na web stranicu. To osigurava da struktura i stil stranice već budu na mjestu do trenutka kada se JavaScript počne prikazivati. Ako se JavaScript učita i pokuša pokrenuti prije nego što su HTML i CSS učitani, onda se javljaju greške (What is JavaScript? - Learn web development | MDN, 2018.). Također bitno je napomenuti da postoji razlika između JavaScripta i Jave. Naime, JavaScript skriptni je jezik, dok je Java programski, objektno-orijentirani jezik.

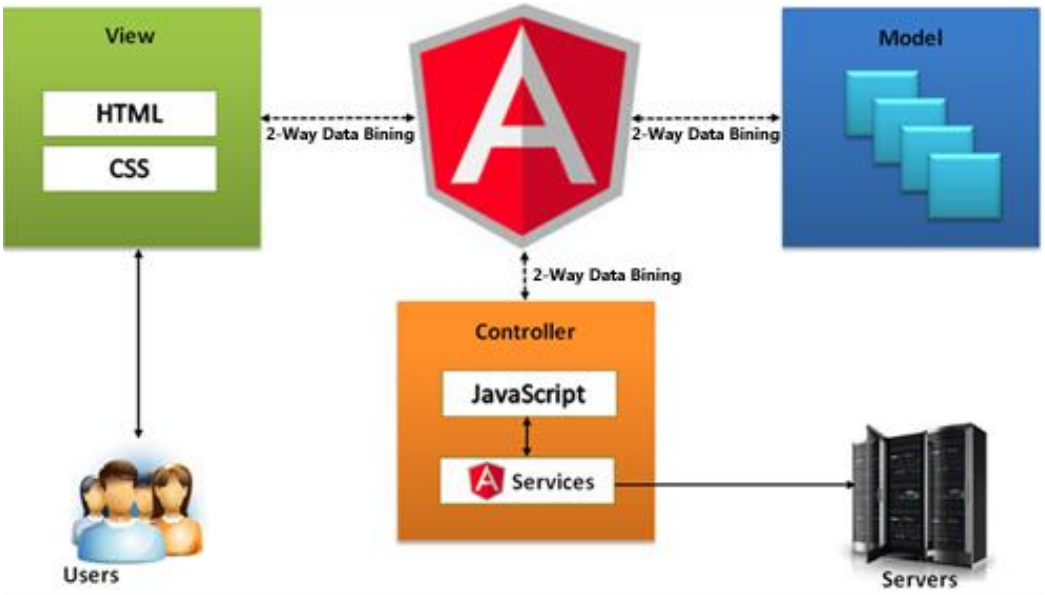
3.4 Angular.js

Angular.js je okvir (eng. *framework*) za razvoj aplikacija. Razvijen je 2009. godine od strane Google-a, ali je sad *open-source* (softver čiji je izvorni kod dostupan i može se redistribuirati i mijenjat) okvir. Sufiks „JS“ znači da se radi o JavaScript okviru za razvoje klijentskih weba aplikacija. Angular.js prva je verzije te se počevši od druge verzije naziva samo Angular.

Osim što je *open-source* okvir, također je i *client-side* i *server-side* web aplikacijski okvir. *Client-side* okvir znači da se JavaScript kod izvršava u web pregledniku. Takav se kod može pisati u zasebnom dokumentu sa sufiksom .js, te možemo ga uključiti unutar dokumenta u kojem se nalazi naš HTML kod ili pak ga možemo pisati direktno u sami HTML dokument. U slučaju da ga pišemo direktno u HTML dokument koristimo otvorenu „<service>“ i zatvorenu „</service>“ oznaku, a kod pišemo unutar tih oznaka. Takav način korištenja okvira nam omogućuje kontrolu nad ponašanjem web stranice. *Server-side* okvir za razliku od *client-side* okvira, znači da se JavaScript kod izvršava izvan web preglednika. Takav kod na serverskoj strani ima pristup serverima, bazama podataka i file sistemima.

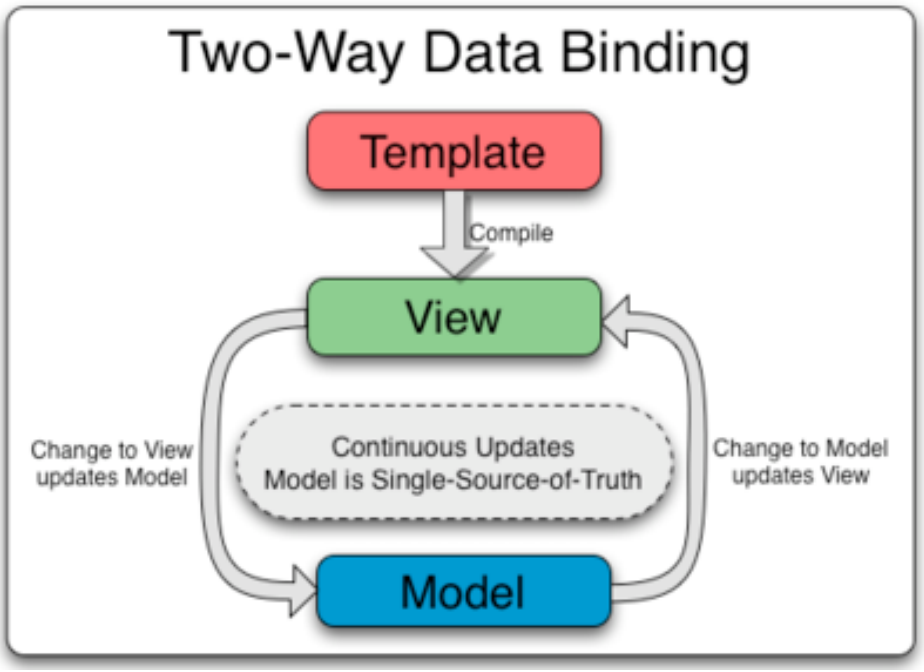
Osnovna je namjena Angular.js okvira razvoj aplikacija koje se sastoje od samo jedne stranice (eng. *single-peage - SPA*). Svojstva su mu direktive, ubrizgavanje ovisnosti i slično, koji će biti kasnije objašnjeni u ovom radu. Zasniva se na model-pogled-upravitelj arhitekturi (eng. *Model – View – Controller - MVC*) koja pomaže pravilnom uređivanju i organiziranju web aplikacija. Model su podaci i poslovna logika aplikacije, pogled je prikaz prethodno modeliranih podataka, a upravitelj upravlja korisničkim zahtjevima. Moglo bi se reći i da je

upravitelj kao neka vrsta ljepila između pogleda i modela koja ih povezuje.



Slika 18 - Prikaz MVC arhitekture

Jedna je od prednosti Angulara.js njegovo vezivanje podataka (eng. *data binding*) i to dvostruko vezivanje podataka (eng. *two-way data binding*). Naime, kod većine sustava za prikazivanje povezuju se podaci samo u jednom smjeru, i to na način da spojite predložak i model komponente zajedno u prikaz. Nakon takvog spajanja promjene na modelu ili srodnim odjeljcima pogleda ne prikazuju se automatski u tom pogledu. Što je još gore, svaka promjena koju korisnik napravi na pogledu nije reflektirana u modelu. To znači da programer mora napisati kod koji konstanto sinkronizira pogled s modelom i obrnuto. Kao što je već navedeno kod Angular.js postoji dvostruko vezivanje podataka što znači da svaka promjena koja se događa na pogledu automatski se prikazuje na modelu i suprotno.



Slika 19 - Dvostruko vezivanje podatka

3.5 Elementi Angular.js aplikacije

Kako bismo omogućili korištenje Angular.js okvira na web stranici potrebno je učitati angular kod. Potom treba povezati Angular.js aplikaciju s html ili body oznakom, a to napravimo na način da nakon html ili body oznake navedemo Angular.js direktivu npr.:

<body ng-app="imeAplikacije"> ... <body>

odnosno pod „imeAplikacije" upisujemo naziv koji smo dali modul kada smo ga kreirali. U nastavku bit će prikazano povezivanje html dokumenta i Angulara.js aplikacije na primjeru aplikacije „Učionica".

Slika 20 - Povezivanje Angular.js okvira i HTML dokumenta

Bitno je napomenuti da kada povezujete Angular.js okvir i HTML dokument to možete napraviti samo jednom, odnosno direktivu ng-app možete koristi samo jednom, u slučaju da direktivu ng-app navedete više puta Angular.js će u obzir uzeti samo onu koja je prvi puta navedena i ona će imati kontrolu nad dokumentom. Naravno, postoji mogućnost da se aplikacija podijeli na više smislenih cjelina, pa se u tu svrhu koriste moduli. U sljedećem odlomku biti će objašnjeno što je to modul.

3.5.1 Module

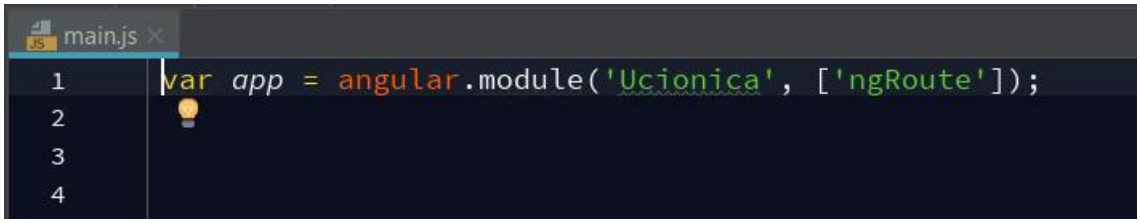
Angular.js modul je kao neka vrsta spremnika, odnosno cjelina koja se sastoji od nekoliko komponenti, a to mogu biti controller, service, filter, directive, i slično (AngularJS - Superheroic JavaScript MVW Framwork, 2018.).

Većina aplikacija ima glavnu metodu koja se instalira i povezuje različite dijelove aplikacije. Angular.js aplikacije nemaju glavnu metodu, umjesto toga moduli deklarativno određuju kako se aplikacija treba podizati. U takvom pristupu postoji nekoliko prednosti, a to su:

- deklarativni proces je lakše razumjeti,
- kod se može pakirati kao modul koji je iznova iskoristiv,
- moduli se mogu učitati bilo kojim redoslijedom, čak i paralelno, jer odgađaju izvršenje,

- unite testovi moraju učitati relevantne module, što ih čini brzima,
- end-to-end testovi mogu koristiti module za nadjačavanje konfiguracije

Modul se deklarira tako da upišete: „angular.module ('myApp', []);“. Na taj način stvara se modul naziva „myApp“, a u uglatim zagradama pišu se nazivi modula o kojima on ovisi. U ovoj aplikaciji nismo koristili svoje module, koji se mogu također upisati, već smo koristili „ngRoute“ modul što možete vidjeti na slici u nastavku.



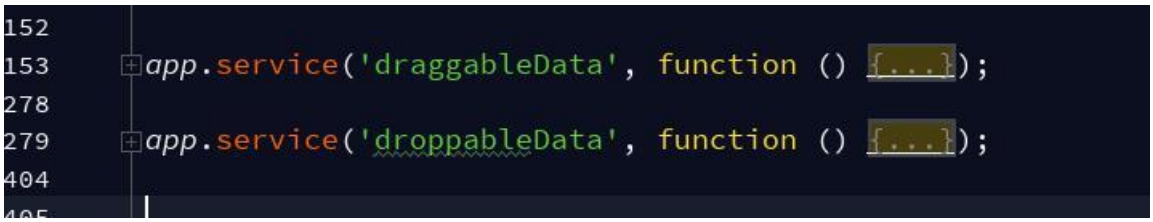
Slika 21 - Kreiranje modula

Kako bi aplikacija postala SPA aplikacija, u tome nam pomaže ngRoute. On vodi aplikaciju kroz različite stranice bez da uvijek iznova očitava cijelu aplikaciju. No o ngRoute ćemo nešto kasnije kad budu objašnjavani pogledi.

Osim toga kreirali smo varijablu „app“, koju smo svaki put pozivali kada smo trebali dohvatiti modul.

3.5.2 Service

Angular.js service su objekti ili funkcije koje obavljaju određene zadatke, uglavnom sadrže neku poslovnu logiku. Dijele se na Lazily instantiated i Singelton. Angular.js prikazuje service samo onda kada aplikacijska komponenta ovisi o tome, tada imamo Lazy instanted, a kod Singletones svaka komponenta koja ovisi o usluzi dobiva referencu na pojedinu instancu koju generira service factory (AngularJS - Superheroic JavaScript MVW Framwork, 2018.). U aplikaciji koristili smo dva Angular.js service, i to kod igrice. Jedan je bio za dohvaćanje, a drugi za postavljanje podataka, odnosno slova. Igrica koju smo izradili funkcionira na način da dohvatite slovo te postavite ga na objekt koji vam sliči tom slovu.



Slika 22- Kreiranje service

3.5.3 Controller

Angular.js kontroler kontrolira podatke u Angular.js aplikaciji, a definira se korištenjem direktive ng-controller (AngularJS - Superheroic JavaScript MVW Framwork, 2018.). Na slici koja slijedi u nastavku vidljiv je primjer na koji način se ng-controller deklarira u HTML dokumentu, a primjer je uzet iz aplikacije "Učionica".

```
<div ng-controller="showLetter" class="container">
    <div class="container">

        <div class="row">
            <div class="col letter" ng-click="changeView('A a')">A a</div>
            <div class="col letter" ng-click="changeView('B b')">B b</div>
            <div class="col letter" ng-click="changeView('C c')">C c</div>
            <div class="col letter" ng-click="changeView('Č č')">Č č</div>
            <div class="col letter" ng-click="changeView('Ć ć')">Ć ć</div>
            <div class="col letter" ng-click="changeView('D d')">D d</div>
            <div class="col letter" ng-click="changeView('DŽ dž')">DŽ dž</div>
            <div class="col letter" ng-click="changeView('Đ đ')">Đ đ</div>
            <div class="col letter" ng-click="changeView('E e')">E e</div>
            <div class="col letter" ng-click="changeView('F f')">F f</div>

        </div>
        <div class="row"...>
        <div class="row"...>
        <div ng-include='"templates/picLetters.html"'></div>
    </div>
</div>
```

Slika 23 - HTML ng-controller za prikaz slova

Slika 24 - HTML ng-controller za prikaz slika slova

Slika 25- HTML ng-controller korišten kod izrađene igre sa slovima

Na takav način kontroler povezujemo s HTML dokumentom. Da bismo kontroler kreirali u dokumentu sa .js sufksom te potom ga povezali s HTML dokumentom, kao što je prikazano u prethodnoj slici, deklariramo ga na način da metodi .controller () proslijedimo ime kontrolera, servise o kojima ovisi te funkciju za kreiranje kontrolera objekta. Budući da smo ovu metodu spremili u varijablu, pri deklaraciji nismo pisali metodu, nego smo svaki put

koristili naziv varijable u kojoj je spremljena metoda. Primjer možete vidjeti na slici koja slijedi u nastavku.

Slika 26 - controller u dokumentu sa .js sufiksom

Kada želimo manipulirati DOM (*Document Object Model*) elementima ne bismo trebali koristi kontrolere za manipulaciju već direktive, a što su direktive bit će objašnjeno u nastavku.

3.5.4 Directive

Angular.js dolazi sa skupom direktiva koje su već izgrađene unutar samog Angular.js, a neke od njih smo već prethodno navodili npr. ng-app, ng-controller i slično. Ispred svake takve direktive dodajemo prefiks "ng-" te potom naziv direktive. Neke od najpoznatijih direktivne koju su već implementirane u Angular.js su:

- ngApp – ova direktiva služi za automatsko pokretanje aplikacije Angular.js, označava korijenskog elementa zahtjeva jer se obično nalazi blizu korijena elemenata stranice npr. <body> ili <html> oznaka,
- ngBind – ovaj atribut govori Angular.js da zamijeni tekstualni sadržaj određenog HTML elementa sa vrijednost danog izraza i da ažurira sadržaj teksta kada je vrijednost tog izraza promijenjena,
- ngClass – omogućuje dinamičko postavljanje CSS klasa na HTML element kroz dvostruko vezivanje izraza koji predstavlja sve klase koje moraju biti dodane,
- ngController – ova direktiva dodaje kontroler klasu na pogled. Ovo je ključni aspekt kako Angular.js podržava načela iz arhitekture model-pogled-upravitelj,

- ngDisabled – ova direktiva postavlja oznaku „onemogućen“ atribut na elementu (obično kontrolni oblik, npr. input, button, select, itd.), ako se izraz unutar ngDisabled vrednuje istinitim,
- ngForm – pomoćna direktiva koja omogućuje da se unutar kontrolne grupe kreira form direktiva. Budući da HTMLL ne dopušta gniježđenje form elemenata, ova direktiva se može koristiti za to,
- ngHide – prikazuje ili skriva dani HTML element na temelju izraza uklanjanjem ili dodavanjem ngHide CSS klase na element,
- ngHref – korištenje Angular.js oznake poput atributa href učinit će vezom pogrešan URL ako ga korisnik klikne prije nego Angular.js ima priliku zamijeniti oznaku s njegovom vrijednost. Sve dok Angular.js ne zamijeni oznaku, veza će se slomiti i najvjerojatnije će vratiti pogrešku 404. Ova direktiva rješava takav problem,
- ngIf - uklanja ili obnavlja dio stabla DOM na temelju {izraz}. Ako se izraz pogrešno procijeni, element se uklanja iz DOM-a, inače se klon elementa ponovno unese u DOM,
- ngInclude – dohvaća, kompajlira i uključuje vanjski HTML dokument na unutarnji HTML dokument,
- ngModel – vezuje <input>, <select>, <textarea> (ili prilagođenu vrstu obrasca) u entitet u opsegu,
- ngReadonly – postavlja samo read-only (samo za čitanje) HTML atribut na elementu uvjetno,
- ngRepeat – instantira predložak jednom po stavci iz zbirke,
- ngShow – prikazuje ili skriva HTML element na temelju navedenog izraza,
- ngSrc – korištenje angular.js oznake {{hash}} kao src atributa ne funkcionira: Preglednik će dohvatiti URL s doslovnim tekstom {{hash}} dok Angular.js ne zamijeni izraz unutar {{hash}}. Ova direktiva rješava taj problem,
- ngStayle – ova direktive omogućuje da postavite CSS stil na HTML element uvjetno,
- ngSwitch – uvjetno zamjenjuje DOM strukturu baziranu na scope izrazu.

Osim najpoznatijih postoje i one koje služe za kreiranje događaja. Općenito rukovanje događajima u Angular.js obavlja se kroz Angular.js događajne direktive. U nastavku će biti navedene neke od najpoznatijih direktiva za rukovanje s događajima, a to su:

- ngChange – procjenjuje dani izraz kada korisnik promijeni ulaz. Izraz je odmah procijenjen, za razliku od JavaScript onchange događaja koji se aktivira na kraju promjena (obično kada korisnik napusti obrazac ili pritisne tipku za povratak),
- ngClick – omogućuje određivanje prilagođenog ponašanja kada se element klikne,
- ngDblclick – omogućuje određivanje prilagođenog ponašanja prilikom dvostrukog klika na događaj,
- ngMousedown – omogućuje određivanje prilagođenog ponašanja prilikom povlačenja miša za dolje na događaj,
- ngMouseup – omogućuje određivanje prilagođenog ponašanja prilikom povlačenja miša za gore na događaj,
- ngMouseover – omogućuje određivanje prilagođenog ponašanja prilikom povlačenja miša preko događaja,
- ngMouseenter – omogućuje određivanje prilagođenog ponašanja prilikom postavljanja miša na događaj,
- ngMouseleave – omogućuje određivanje prilagođenog ponašanja prilikom povlačenja miša s događaja,
- ngMousemove – omogućuje određivanje prilagođenog ponašanja prilikom povlačenja miša na događaj,
- ngKeydown – omogućuje određivanje prilagođenog ponašanja prilikom pritiska tipke za dolje na događaj,
- ngKeyup – omogućuje određivanje prilagođenog ponašanja prilikom pritiska tipke za gore na događaj,
- ngKeypress - omogućuje određivanje prilagođenog ponašanja prilikom pritiska tipke na događaj,
- ngFocus - omogućuje fokusiranje na događaj,
- ngBlur – omogućuje zamagljivanje događaja,
- ngCopy – omogućuje kopiranje događaja,
- ngCut – omogućuje rezanje događaja,
- ngPaste – omogućuje lijepljenje događaja.

Također postoje i HTML povezne direktive. U nekim slučajevima Angular.js modificira ponašanje postojećih HTML elemenata stvaranjem direktive koja nadjačava

postojeće ponašanje. U nastavku će biti navedene neke od tih direktiva, a to su:

- a – izmijeni zadano ponašanja HTML oznake <a>, tako da je zadana radnja spriječena kada je atribut href prazan,
- form – direktiva koja instancira FormController,
- input – kontrola HTML elementa <input>. Kada se upotrebljava zajedno s ngModel, on pruža vezivanje podataka, kontrolu ulaznog stanja i provjeru valjanosti,
- input[checkbox] – HTML <input [checkbox]> kontrola elemenata,
- input[date] – HTML <input [date]> kontrola elemenata,
- input[datetime-local] – HTML <input [datetime-local]> kontrola elemenata,
- input[email] – HTML <input [email]> kontrola elemenata,
- input[month] – HTML <input [month]> kontrola elemenata,
- input[number] – HTML <input [number]> kontrola elemenata,
- input[radio] – HTML <input [radio]> kontrola elemenata,
- input[text] – HTML <input [text]> kontrola elemenata,
- input[time] – HTML <input [time]> kontrola elemenata,
- input[url] – HTML <input [url]> kontrola elemenata,
- input[week] – HTML <input [week]> kontrola elemenata,
- select – HTML <select> element s Angular.js vezivanjem podataka,
- textarea - HTML <textarea> kontrola elemenata s Angular.js vezivanjem podataka.

Direktive koje su korištene u aplikaciji "Učionica" su: ng-click, ng-controller, ng-app, ng-include te ng-repeat. Direktiva ng-app, kao što je navedeno u odlomku „Elementi Angular.js aplikacije“, korištena je kako bismo omogućili korištenje Angular.js okvira na web stranici, odnosno učitali web angular kod, a primjer je pokazan na slici 20.

Osim navedenih direktiva u svojoj aplikaciji koristili smo i svoje direktive, odnosno one koje smo sami kreirali, a koje slijede na slikama u nastavku.

```
app.directive('dragme', ['$timeout', function ($timeout) {
    return {
        restrict: 'A',
        replace: true,
        scope: {
            myindex:'='
        },
        link: function ($scope, $elem, $attr) {
            var backgraundImage = $attr.backgroundimage;
            var answerData = $attr.answerdata;
            var myBgcolor = $attr.bgcolor;
            var myLeft = parseInt($attr.left);

            $elem.addClass('draggable');
            $elem.attr('data-answerimage', backgraundImage);
            $elem.attr('data-answerdata', answerData);
            $elem.attr('data-myindex', $scope.myindex);

            $elem.css({
                left: myLeft,
                backgroundImage: 'url(img/slikeZaIgre/' + backgraundImage + ')'
            });
            $elem.draggable({
                helper: 'clone',
                revert:true,
                appendTo:'body',
                zIndex: 100,
                drag: function (event, ui) {
                    $(ui.helper).css('border', '0px');
                }
            });
        }
    }
}]);
```

Slika 27 - dragme direktiva

Ta direktiva kreirana je kako bi se slova mogla iz dohvaćajućeg retka premještati na otpuštajući redak, dok je direktiva koja slijedi u nastavku kreirana kako bi se slova koja su ista, a nakon što ih povučemo s dohvaćajućeg retka mogla zapamtiti u otpuštajućem retku.

```
app.directive('dropme', ['$timeout', function ($timeout){
    return{
        restrict: 'A',
        replace: true,
        scope: {
            setScore: "&",
            removeArray: "&"
        },
        link: function ($scope, $elem, $attr) {
            var backgroundImage = $attr.backgroundimage;
            var answerData = $attr.lettername;

            $elem.addClass('droppable');
            $elem.attr("data-answerimage", backgroundImage);
            $elem.attr("data-answerdata", answerData);
            $elem.css({
                backgroundImage: 'url(img/slikeZaIgre/' + backgroundImage + ')'
            });
            $elem.droppable({
                accept: '.draggable',
                drop: function (event, ui) {
                    var droppedElem = ui.draggable;
                    var myAnswer = $(this).attr("data-answerdata");
                    if ($(droppedElem).attr("data-answerdata") == myAnswer) {
                        $(this).css("background-image", "url(img/slikeZaIgre/" + droppedElem.attr("backgroundimage") + ")");
                        $(this).attr("data-isanswered", "yes");
                        $scope.setScore({
                            value: 1
                        });
                        $scope.removeArray({
                            value:$(droppedElem).attr('data-answerdata')
                        });
                    }
                }
```

Slika 28 - dropme direktiva

ng-repeat

Direktivu ng-repeat koristimo kada želimo prikazati ponavljajuće vrijednosti definirane u našem kontroleru. Primjer možemo vidjeti na slikama koje slijede.

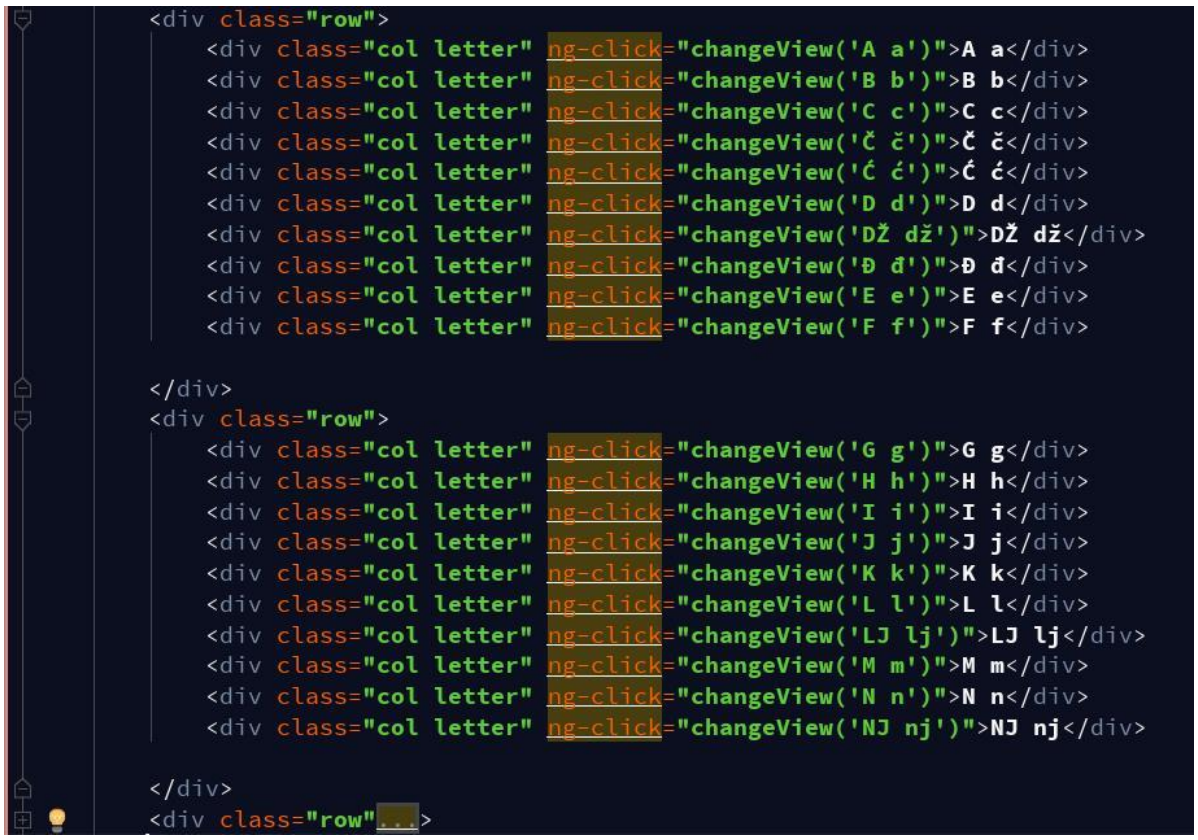
```
<div data-ng-repeat="item in draggableArray">
    <div dragme bgcolor="fdcca0"
        left="20"
        data-lettername="{{item.lettername}}"
        answerdata="{{item.lettername}}"
        backgroundimage="{{item.letterimg}}"
        myindex="$index">


    </div>
</div>
```

Slika 29 - ng-repeat direktiva za prikaz slova koji se pomiču

Slika 30 - ng-repeat direktiva za prikaz sadržaja na koji ćemo staviti slovo

ng-click

Direktiva ng-click korištena je kada smo željeli prikazati slike slova te kako se ta slova pišu. Iz razloga da bismo izbjegli pretjeran broj kreiranih pogleda, te na taj način smanjili veličinu same aplikacije koristili smo navedenu direktivu. Primjer direktive vidljiv je na slici u nastavku.



Slika 31 - ng-click direktiva za prikaz slika slova

ng-include

Diretkiva ng-include dohvaća, kompajlira i uključuje vanjski HTML dokument. Ovo je izvrsno kod podjele koda na manje fragmente HTML-a ili partials koji se mogu uključiti kada je to potrebno. Ng-include direktivu koristili smo kada smo željeli prikazati predloške na pogledima, a primjer možete vidjeti na slikama broj 34 i 35.

ng-View

Direktiva ng-View nadopunjuje uslugu $route uključivanjem isporučenog predloška trenutne rute u glavnu datoteku izgleda. Svaki put kada se mijenja trenutačna ruta, uključeni prikaz mijenja se s njom prema konfiguraciji usluge. Ta direktiva zahtijeva ngRoute modul. Primjer korištenja ngView direktive vidljiv je u nastavku na slici broj 34, a primjer definiranja ngRoute modela vidljiv je na slici broj 21.

3.5.5 Angular.js Scope

Angular.js Scope poseban je JavaScript objekt koji pomaže u pridruživanju kontrolera i pogleda. Drugim riječima Angular.js scope sadrži podatke o modelu. Unutar kontrolera, tom modelu podataka može se pristupiti koristeći $scope objekt.

Angular.js Scope omogućuje nekoj varijabli ili sadržaju da joj se pristupi u HTML dokumentu koristeći izraz:

```
<div ng-controller="MyController">
    <h1>{{message}}</h1>
</div>.
```

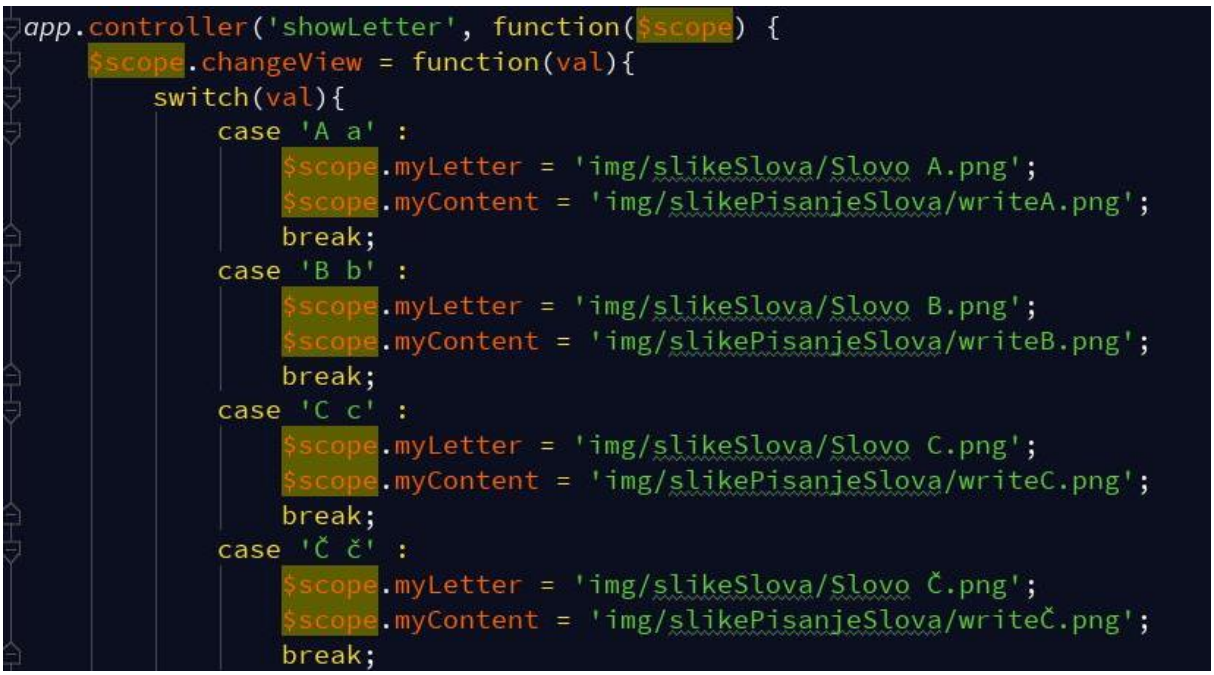
Kako bi se određeni izraz mogao ispisati na takav način potrebno ga je deklarirati u dokumentu s .js sufiksom, i to na način:

```
angular.module („myApp“, [])
.controller(„MyCotroller“, function ($scope){
```

$scope.message = „Hello Word“;

});.

U svojoj aplikaciji „Učionica“ koristili smo Angular.js Scope kada smo željeli promijeniti izgled slike slova, i to na način prikazan na slikama u nastavku. Prva slika prikazuje kako izgleda kreiranje Angular.js Scope, a druga kako se taj $scope, odnosno u ovom slučaju njegov sadržaj poziva u HTML dokumentu. Naime, ovaj HTML dokument je template, a što su template-ovi bit će objašnjeno u sljedećem poglavlju.



Slika 32 - Kreiranje Angular.js Scope kod prikaza slika slova (u main.js dokumentu)

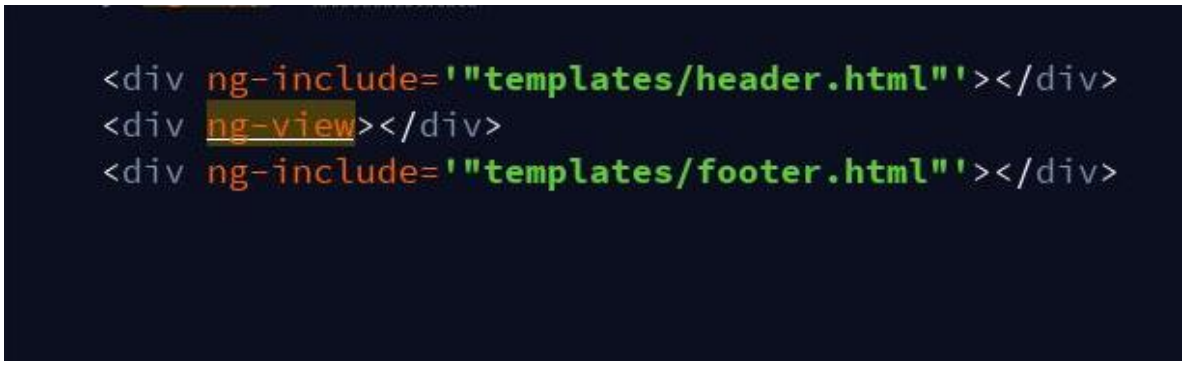
Slika 33 - Korištenje Angular.js Scope kod prikaza sadržaja na template-u (u pickLetter.html) dokumentu

3.5.6 Predložak

U Angularu.js predlošci (eng. *template*) su napisani u HTML-u te sadrže elemente i atribute specifične za Angular.js. Angular.js kombinira predložak s podacima iz modela i kontrolera kako bi prikazao dinamički prikaz koji korisnik vidi u pregledniku. U jednostavnim aplikacijama, predložak se sastoji od HTML-a, CSS-a i Angular.js direktiva sadržanih u samo jednoj HTML datoteci, obično je to index.html. Dok u složenim aplikacijama možete prikazati više pogleda, odnosno predložaka koristeći „partials“. Partials su segmenti predložaka locirani u razdvojenim HTML dokumentima. Da bi se „partial“ temeljen na konfiguraciji proslijeđenoj servisu $route ažurirao, potrebno je koristiti ngView.

U svojoj aplikaciji koristili smo zasebno predložak i pogled. Pogledi su korišteni za prikaz cjelokupne stranice na kojoj se nalaze slova te prikaza igre sa slovima, kao i početne stranice, dok su predlošci korišteni za prikaz zaglavlja (header.html), podnožja (footer.html) te predloška za pregled slika slova (picLetters.html).

Kako biste predložak prikazali na nekom pogledu on se mora deklarirati na način naveden slikom u nastavku.



Slika 34 - template (header i footer)

Kako bi se na pogledu letters.html mogle prikazati slike slova, i to bez da smo za svako slovo radili poseban pogled koristili smo ng-include koji nam je služio kako bismo dohvatili, kompajlirali i uključili vanjski template (picLetters.html), te pomoću njega omogućili mijenjanje slika slova.

Slika 35 – uključivanje template (picLetters.html)

3.5.7 jQuery

Ako malo bolje pogledate slike 11 i 12 vidjet ćete da smo kod izrade direktiva koristila elemente i atribute jQuery. jQuery je JavaScript biblioteka. On omogućava manipulaciju HTML dokumentom, animacijom i AJAX-om. Dizajneri ga koriste kako bi olakšali sebi posao oko dizajniranja interaktivne web stranice.

Prednost je jQuery-ja jednostavno korištenje. U odnosu na ostale JavaScript biblioteke, a i sami JavaScript ima jednostavnu sintaksu te također zahtijeva puno manje linija koda kako bi se postigla ista značajka. U odnosu na ostale JavaScript biblioteke jQuery ima ogromnu biblioteku te pruža puno više funkcija. Za poboljšanje jQuery biblioteke postoji open-surce zajednica koja svoje vrijeme posvećuje njezinom razvoju i poboljšanju, a postoji i

mnogo već napisanih datoteka (eng. *plugins*) dostupnih za preuzimanje, kako bi se proces razvoja ubrzao. Web stranice imaju sveobuhvatnu dokumentaciju i tutorijale kako bi se olakšao rad s bibliotekama. Također, jQuery omogućava razvoj Ajax obrazaca, a Ajax omogućuje sučelje gdje se mogu obaviti radnje na stranicama, bez potrebe da se učitava ponovo cijelu stranicu. Za pokretanja jQuery naredbi potrebna je jQuery JavaScript datoteka premada je veličina takve datoteke mala, ona opterećuje klijentsko računalo te web poslužitelja ako se jQuery skripte pohranjuju na web poslužitelju (jQuery, 2018.).

Kao što se moglo vidjeti u direktivama smo koristili oznaku "$" koja označava da se radi o jQuery.

Da bi spojili jQuery s našim HTML dokumentom uključili smo jQuery od strane Contact Delivery Networka (CDN), u ovom slučaju Google, i to na način kako je prikazan na slici u nastavku.

Slika 36 - Uključivanje jQuery u HTML dokument

3.5.8 Alat za izradu apliakcije

Alat koji smo koristili prilikom izrade aplikacije je Webstorm. To je alakt koji je kreirao JetBrains. Ovaj alat je stručni IDE (Integrated Development Enviorment). Pomaže kod jednostavnog kreiranja web stranica tako da omogućava uređivanje HTML, CSS i JavaScript dokumenata, korištenje automatskog dovršetka za sve što je u kodu koji kreiramo te jednostavno kretanje kroz datoteke. Webstorm također prati sve izmjene na izvornim datotekama te na taj način štiti od slučajnih gubitaka ili izmjena. To je i program za ispravljanje pogrešaka. Radi na način da otklanja poteškoće s aplikacijama Node.js na stranicama klijenata i IDE (WebStorm: The Smartest JavaScript IDE by JetBrains, 2018.)

Na temelju poznavanja HTML i XML koda Webstorm dovršava stilove, reference o

datotekama, nazive oznaka, zatvaranje oznaka, atribute, također u CSS kodu završava klase, HTML ID-ove, ključne riječi, vrijednosti i slično. Kao što smo već i naveli ovo je program i za ispravljanje pogrešaka, i to na način da otkriva i predlaže automatske ispravke za probleme kao što su nevažeći format izbornika CSS, nevažeća CSS svojstva, neiskorištene definicije CSS razreda, u slučaju da nedostaju atributi, kod nevažećih atributa ili nezakonitih vrijednosti, te duplikata atributa.

Osim svih tih prednosti WebStorm ima jednu manu, a to je da se plaća. Međutim, to nije slučaj kod studenata. Naime, student se može prijaviti na stranicu JetBrains s korisničkom elektroničkom poštom fakulteta te na taj način mu je omogućeno da WebStorm koristi besplatno.

3.5.9 Dizajn

Za izradu ove web aplikacije korišten je Bootstrap okvir koji se koristi za stvaranje bržih web stranica. Ima već napravljene gumbe, tablice, forme, navigacijske trake i slično. Osim toga korišten je i stayle.css dokument koji sadrži naredbe za veličinu slova, pozadina itd. Kako bi se koristio Bootstrap okvir, korišteni su „class“ atributi svakog HTML elementa (Bootstrap - The most popular HTML, CSS, and JS library ind the world, 2018.).

Da biste mogli koristi Bootstrap okvir potrebno je u HTML dokument dodati link koji se nalazi na njihovim stranicama. Dovoljno ga je kopirati i dodati u svoj projekt. Osim toga potrebno je dodati i JavaScript za funkcioniranje jer mnoge Bootstrap komponente to zahtijevaju. Točnije zahtijeva se jQuery, Popper.js i vlastiti JavaScript dodaci. Način na koji dodajete JavaScript na svoju stranicu je isti kao i s dodavanjem Bootstrapa.

```
<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-theme.min.css">
<link href="http://maxcdn.bootstrapcdn.com/font-awesome/4.1.0/css/font-awesome.min.css" rel="stylesheet">
<link rel="stylesheet" href="css/main.css">
```

Slika 37 - Dodavanje Bootstrapa

Bootstrap dolazi s dodatkom globalnih CSS stilova i stiliziranih HTML elementima

koji su unaprijed dodati klasama. Radi na način da se „class“ atributu elementa pridoda određeno ime koje je ključna riječ unutar Bootsrapa te onda se tim imenom određuje kako će taj element izgledati. Od Bootstrap elemenata koristili smo „navbar“, „navbar-collapse,“ „col“, „row“ i slično.

```
<nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
    <div class="container">
        <div class="navbar-header">
            <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-ex1-collapse">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" href="#/">UČIONICA</a>
        </div>
        <div class="collapse navbar-collapse navbar-ex1-collapse">
            <ul class="nav navbar-nav navbar-right">
                <li>
                    <a href="#/letters">Slova</a>
                </li>
                <li>
                    <a href="#/letterGame">Igre sa slovima</a>
                </li>
            </ul>
        </div>
    </div>
</nav>
```

Slika 38 - Primjer korištenja Bootstrap class

Na slici 22 prikazan je primjer korištenja Bootstrap klasa kod izrade navigacijskog izbornika koji se nalazi u aplikaciji.

4 PREDNOSTI I NEDOSTACI

JavaScript već je dobro uspostavljena tehnologija poznata po prikazivanju digitalnih svojstava. Dinamički skriptni jezik na strani klijenta spojen je s HTML-om i CSS-om kako bi proširio i unaprijedio funkcionalnost na webu. Nije ni čudo da brojni razvojni okviri za web razvoje zasnovani na JavaScriptu su i dalje vodeći u većem broj web stranica i web aplikacija. Angular.js je jedan od tih okvira i omiljen je među programerima bogatih internetskih aplikacija.

Prednosti Angular.js dvosmjerno je vezivanje podataka što znači da Angular.js olakšava bržu i lakšu vezu podataka koja ne zahtijeva da programer intervenira na način da kreira program koji će konstantno sinkronizirati pogled s modelom i obrnuto. Osim toga prednost je i DOM manipulacija. Za razliku od ostalih popularnih JavaScript okvira, prikladno olakšava programeru aktivno manipuliranje DOM-om, a sve zahvaljujući dvosmjernom pristupu vezivanja podataka. To znači da programer štedi vrijeme i napore za kodiranje, prevođenje i ažuriranje DOM elemenata. Budući da podržava predmemoriranje i mnoge druge procese, Angular.js smanjuje opterećenje na procesoru poslužitelja. To znači da poslužitelj izvodi iznimno dobro zahvaljujući smanjenom prometu i zato što služi samo statičnim datotekama i odgovara na API pozive, na taj način poboljšana je izvedba poslužitelja. Jedna od prednosti je i to da je Angular.js ima mogućnost lakog i brzog testirani proizvoda. Angular.js temelji se na model-pogled-upravitelj (MVC) arhitekturi koja dodatno povećava brzinu testiranja. U potpunosti je baziran na HTML-u i JavaScript-u, stoga nema potrebe da se uči neka druga sintaksa ili jezik. On mijenja statički HTML u dinamički, proširuje mogućnosti HTML-a dodavanjem ugrađenih atributa i komponenti, a također pruža mogućnosti stvaranja prilagođenih atributa jednostavnim JavaScript-om.

Međutim, Angular.js ima svojih nedostataka, a ti nedostaci su da je JavaScript podrška obavezna. Zamislite računala i prijenosna računala koja pristupaju vašem poslužitelju, ali njihova podržaka za JavaScript je onemogućena. Povezani korisnici neće moći pristupiti vašoj web lokaciji ili web aplikacijama. Ako postoji nekolicina takvih korisnika, može se očekivati da će web svojstva bazirana na Anguar.js okviru biti manje korisna. To nije slučaj s običnim webom temeljenim na HTML-u. Ako ste razvojni programer koji slijedi tradicionalni pristup i upoznati ste s MVC arhitekturom, upotreba Angulara.js može biti iznimno dugotrajna. Angular.js je vrlo slojevit i raspoređen hijerarhijski, scope može biti složen entitet s kojim se

morate nositi ako tek počinjete stjecati iskustvo u korištenju Anguar.js. Jedan od najtežih zadataka može biti otklanjanje grešaka u scopeu. Direktive je teško koristiti, osim toga mnoge Angular.js značajke kao dependency injections i factories mogu biti problematične za tradicionalne programere. Pregledniku može biti potrebno više vremena za prikaz web stranica i aplikacija dizajniranih pomoću okvira. To se može dogoditi jer će preglednici biti preopterećeni za obavljanje dodatnih zadataka poput DOM manipulacije. Međutim, taj nedostatak ograničen je samo na stara računala i uređaje. Suvremeni, osobito mobilni uređaji nemaju takvih problema, odnosno nisu problematični. Koliko je učenje Angulara.js lagano toliko je i teško. Umjesto da se slijedi ravni grafikon učenja možete se suočiti s velikim teškoćama prilikom prilagodbe okviru. Štoviše, ograničena dostupna dokumentacija može dodatno utjecati na proces učenja, ali zbog zajednice koja radi na Angular.js svakodnevno olakšava se rad u njemu.

5 ZAKLJUČAK

Budući da smo imali priliku vidjeti i upoznati se sa asistivnim tehnologijama, došli smo do zaključka kako one predstavljaju veliku prednost kod poboljšanja života pojedinca. Omogućuju pojedincu lakšu socijalizaciju, učenje, te nadilaženje infrastrukturnih prepreka kako bi se što lakše uključio u društvene znanosti. Međutim, asistivne su tehnologije još uvijek jako skupe te to im je i najveća mana. Roditelji djece s poteškoćama u razvoju ne mogu si priuštiti neku od asistivnih tehnologija koje su im nužno potrebne kako bi olakšali osnovne životne potrebe svom djetetu. Osim toga i roditelji djece koja nemaju poteškoće u razvoju, ali zbog svog socijalnog i materijalnog statusa nisu u mogućnosti djeci kupiti udžbenike, aplikacije koje se naplaćuju i slično, ova aplikacija ima mogućnost nadogradnje, besplatna je i dostupna svima u svakom trenutku i to je njezina prednost.

Kod ove aplikacije ostavljeno je mjesta za nadogradnju te istoj se može dodati sučelje za prijavu, koje bi omogućavalo prijavu ili registraciju korisnika, i to na način da učitelj vidi sve korisnike, odnosno svoje učenike i može obavljati pregled zadaća, tj. kako je dijete riješilo igru ili zadaću, jer je namjera bila napraviti bazu podataka u koju bi se spremao rezultat. Kada bi dijete riješilo igru ili zadaću, učitelj bi imao mogućnost ocijeniti riješeno, odnosno vidjeti rezultate svojih učenika. Ta se aplikacija s namjerom zove "Učionica" kako bi se ju moglo nadograditi za učenje brojeva, zbrajanja, oduzimanja, množenja, dijeljenja, što i jest bio cilj, pa čak i za ostale predmete koji se uče u školama. Budući da gradivo koje se uči u škola ima tendenciju promjene, ovdje postoji mogućnost da se gradivo promijeni, a da pritom nije potrebno mijenjati cjelokupnu aplikaciju ili kupovati u potpunosti novi udžbenik te za to potrošiti neizmjerno puno materijalnih sredstava. Nije potrebno instalirati nekoliko aplikacija kako bi se moglo učiti više predmeta, nego je dovoljna samo jedna.

Današnja djece odrastaju uz pametne telefone i tablete te od malih nogu imaju „osjećaj za tehnologije“, pa bi edukativne igre i aplikacije pomogle bržem učenju slova, brojeva, čitanja, pisanja, zbrajanja, kao i logičkih koncepata, rješavanju problema te razvoju finih motoričkih vještina. Učenicima bi to bio izazov, poticaj u radu i izvor radosti u stjecanju znanja što je i cilj načina stjecanja znanja.

POPIS LITERATURE

1. Alphamonster Review | Educational App Store (2018.), https://www.educationalappstore.com/app/alphamonster (14.9.2018.)
2. AngularJS - Superheroic JavaScript MVW Framwork (2018.) https://angularjs.org/ (5. rujna 2018.)
3. Bootstrap - The most popular HTML, CSS, and JS library ind the world, (2018.) https://getbootstrap.com/ (5. rujna 2018.)
4. Endless Alphabet Reviw | Educational App Store (2018.), https://www.educationalappstore.com/images/screenshots/app8009/44.png (14.9.2018.)
5. Introduction to CSS - Learn web development | MDN, (2018.) https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS (1. rujna 2018.)
6. Izdanja HUD-a – Hrvatska udruga za disleksiju (2014.), http://hud.hr/izdanja-hud-a/ (14. rujna 2018.)
7. iWriteWords Review | Education App Store (2018.), https://www.educationalappstore.com/app/iwritewords-lite (14.9.2018.)
8. jQuery,(2018.) https://jquery.com/ (8. rujna 2018.)
9. LetraKid PRO – Handwriting ABC Review | Educational App Store (2018.), https://www.educationalappstore.com/app/letrakid-pro-handwriting-abc (14. rujna 2018.)
10. LetterSchool – learn to write letters and numbers Review | Educational App Store (2018.), https://www.educationalappstore.com/app/letterschool-learn-to-write-letters-and-numbers (14.9.2018.)
11. Medicinski priručnik za pacijente, (2014.) http://www.msd-prirucnici.placebo.hr/msd-za-pacijente/zdravlje-djece/problemi-u-razvoju-male-djece/disleksija (1. rujna 2018.)
12. Sunčica (2013.), http://www.32bita.hr/suncica (12.9.2018.)
13. Učilica (2012.), http://www.ucilica.tv/Default.aspx (12.9.2018.)
14. WebStorm: The Smartest JavaScript IDE by JetBrains, (2018.), https://www.jetbrains.com/webstorm/ (25. srpnja 2018.)
15. What is JavaScript? - Learn web development | MDN, (2018.) https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript (2. rujan 2018.)

POPIS SLIKA

Slika 1 - Sučelje aplikacije LetraKid PRO 4

Slika 2 - Sučelje aplikacije Alphamonster 5

Slika 3 - Sučelje aplikacije "iWriteWords" 6

Slika 4 - Sučelje aplikacije "LetterSchool" 7

Slika 5 - Sučelje aplikacije "Endless Alphabet" 8

Slika 6 - Prikaz sučelja „Sunčica“ 9

Slika 7 - Prikaz sučelja "Mala Učilica" za predškolce 10

Slika 8 - Prikaz sučelja "CD Učilica" za osnovnoškolce 11

Slika 9 - Prikaz naslovne strane aplikacije "CD Čitajmo zajedno" 12

Slika 10 - Početna stranica aplikacije "Učionica" 14

Slika 11 - Prikaz stranice za učenje izgleda slova u aplikaciji "Učionica" 14

Slika 12 - Prikaz izgleda i pisanja jednostavnog slova aplikacije "Učionica" 15

Slika 13 - Prikaz izgleda i pisanja kompleksnog slova aplikacije "Učionica" 15

Slika 14 - Prikaz stranice za igru sa slovima aplikacije "Učionica" 16

Slika 15 - Prikaz pogođenog slova i povećanja rezultata u aplikaciji "Učionica" 17

Slika 16 - Prikaz postavljanja slova na krivi prozorčić u aplikaciji "Učionica" 17

Slika 17 - Prikaz vraćanja slova u lijevi stupac aplikacije "Učionica" 18

Slika 18 - Prikaz MVC arhitekture 22

Slika 19 - Dvostruko vezivanje podatka 23

Slika 20 - Povezivanje Angular.js okvira i HTML dokumenta 24

Slika 21 - Kreiranje modula 25

Slika 22- Kreiranje service 26

Slika 23 - HTML ng-controller za prikaz slova 26

Slika 24 - HTML ng-controller za prikaz slika slova..... 27

Slika 25- HTML ng-controller korišten kod izrađene igre sa slovima 27

Slika 26 - controller u dokumentu sa .js sufiksom 28

Slika 27 - dragme direktiva 32

Slika 28 - dropme direktiva 33

Slika 29 - ng-repeat direktiva za prikaz slova koji se pomiču 33

Slika 30 - ng-repeat direktiva za prikaz sadržaja na koji ćemo staviti slovo 34

Slika 31 - ng-click direktiva za prikaz slika slova 34

Slika 32 - Kreiranje Angular.js Scope kod prikaza slika slova (u main.js dokumentu)..... 36

Slika 33 - Korištenje Angular.js Scope kod prikaza sadržaja na template-u (u pickLetter.html) dokumentu 37

Slika 34 - template (header i footer)..... 38

Slika 35 – uključivanje template (picLetters.html) 38

Slika 36 - Uključivanje jQuery u HTML dokument 39

Slika 37 - Dodavanje Bootstrapa..... 40

Slika 38 - Primjer korištenja Bootstrap class 41

PRILOZI

CD:

JavaScript, HTML, Angular.js i CSS dokumenti koji sadrže cjelokupni rad (main.css, slikePisanjaSlova, slikeSlova, slikeZaIgru, school_board.jpg, angular.min.js, main.js, underscore-min.js, home.html, letterGame.html, letters.html, footer.html, header.html, pickLetters.html, index.html).

Seminar u .docx i .pdf formatu.

SAŽETAK

Tema završnog rada razvoje je aplikacije korištenjem programskog okvira Angular.js. Aplikacija koju smo razvili zove se „Učionica“ te služi za prepoznavanje, učenje velikog i malog tiskanog slova, kako se pišu. Ovaj rad sastoji se od teorijskog i praktičnog dijela. U teorijskom dijelu detaljno je opisan Angular.js okvir, JavaScript kao skriptni jezkik te ostale tehnologije i alati koji su korišteni u praktičnome dijelu. Weba aplikacija rađena je skriptnim jezikom JavaScript, opisnim jezikom HTML, jezikom za postavljanje izgleda CSS-om te Angular.js okvirom. Aplikacija se sastoji od tri pogleda i tri template. Prvi pogled prikazuje početnu stranicu, drugi pogled prikazuje grid sa slovima, a na tom istom pogledu nalazi se predložak koji se otvara nakon što se klikne na određeno slovo. Ostali predlošci su zaglavlje i podnožje. Na posljednjoj, trećoj stranici nalazi se igra prepoznavanja slova. Igra funkcionira na način da iz lijevog retka vučemo slova u središnji redak i postavljamo na prozorčić u kojem se nalazi slovo koje mu sliči. U slučaju da se postavi na krivi prozorčić slovo se automatski vraća na početno mjesto. U desnom retku nalazi se podatak o postignutom rezultatu.

Ključne riječi: Angular.js, JavaScript, Učionica, HTML, CSS, Bootstrap

SUMMARY

The theme of the bachelor thesis is to develop applications using the Angular.js framework. The application I developed is called "Učionica" and serves to recognize and learn the big and small letters, as they are written. This work consists of a theoretical and practical part. In the theoretical part, the Angular.js framework, JavaScript as a scripting language and other technologies and tools that are used in the practical part are described in detail. The web application is executed with JavaScript, descriptive HTML language, CSS layout language, and Angular.js framework. The application consists of three views and three templates. The first view shows the home page, the second view shows the letter grid, and in that view there is a template that opens after clicking on a particular letter. Other templates are header and footer. On the last, third page there is a character recognition game. The game works by dragging the letters in the middle line from the left line and placing it in a window with the letter that resembles it. If it is placed in the wrong window, the letter automatically returns to the initial position. In the right line there is information about the result achieved.

Keywords: Angular.js, JavaScript, Učionica, HTML, CSS, Bootstrap