

**VIŠJA STROKOVNA ŠOLA ACADEMIA**  
**MARIBOR**

# **Primerjava spletnih programskih ogrodij in izdelava spletnega vprašalnika**

Kandidat: Patrik Štraus

Vrsta študija: študent izrednega študija

Študijski program: Informatika

Mentor predavatelj: dr. Bojan Kverh

Lektor: Eva Visinski, mag. prof. slov. jez. in knj.

Maribor, 2022

# **IZJAVA O AVTORSTVU DIPLOMSKEGA DELA**

Podpisani Patrik Štraus sem avtor diplomskega dela z naslovom Primerjava spletnih programskih ogrodij in izdelava spletnega vprašalnika, ki sem ga napisal pod mentorstvom dr. Bojana Kverha.

S svojim podpisom zagotavljam, da:

- je predloženo delo izključno rezultat mojega dela,
- sem poskrbel, da so dela in mnenja drugih avtorjev, ki jih uporabljam v predloženi nalogi, navedena oz. citirana skladno s pravili Višje strokovne šole Academia Maribor,
- se zavedam, da je plagiatorstvo – predstavljanje tujih del oz. misli kot moje lastne kaznivo po Zakonu o avtorski in sorodnih pravicah (Uradni list RS, št. 16/07 – uradno prečiščeno besedilo, 68/08, 110/13, 56/15 in 63/16 – ZKUASP); prekršek pa podleže tudi ukrepom Višje strokovne šole Academia Maribor skladno z njenimi pravili,
- skladno z 32.a členom ZASP dovoljujem Višji strokovni šoli Academia Maribor objavo diplomskega dela na spletnem portalu šole.

Maribor, september 2022

Podpis študenta:

## **ZAHVALA**

Za pomoč pri izdelavi diplomskega dela se zahvaljujem mentorju, dr. Bojanu Kverhu, ki mi je hitro in korektno svetoval ter pomagal pri odpravljanju napak. Posebna zahvala gre moji puncu Tini, ki je z menoj bila potrpežljiva in me vzpodbujala ter motivirala pri pisanju, tudi ko nisem imel energije. Zahvaljujem se tudi moji in njeni družini, ki so me v času študija podpirali in vzpodbujali na svoj način, da je to diplomsko delo prišlo na dan. Zahvaliti se moram tudi mojemu direktorju, g. Benjaminu Bilaloviću, ki mi je odobril proste dneve, da sem jih lahko namenil študiju ter pisanju diplomskega dela, ter vsem sodelavcem na oddelku DWH&BI, ki so mi v času pisanja pomagali razbremeniti moj delovnik.

## POVZETEK

V diplomskem delu se bom osredotočil na izdelavo spletnih strani s pomočjo spletnih ogrodij ReactJS, VueJS in Angular, predstavil njihove prednosti in slabosti ter povzel krovno temo, ki se nanaša na delo od doma.

Že od razvoja interneta spletne strani temeljijo tako ali drugače na tehnologijah HTML, CSS, in JavaScript.

Prva poglavja diplomskega dela zajemajo kratko predstavitev zgodovine razvoja spletnih jezikov, njihovih struktur ter s kakšnimi problemi so se programerji pred 30. leti srečevali, da je nastala potreba po razvoju le-teh in kako so vplivale na razvoj najbolj popularnih spletnih ogrodij, ki jih poznamo danes.

Koda ogrodij temelji na JavaScriptu in njegovih različicah (JSX, TypeScript). Vsi gradniki temeljijo na podobni mentaliteti - ustvari čim več, s čim manj dela. To pomeni, da pri ustvarjanju spletnih strani temelji na ustvarjanju komponent.

S pomočjo tega lahko razvijalci kodo večkrat uporabijo, kar jim skrajša čas razvoja, omogoča učinkovitejše testiranje, nadgrajevanje in vzdrževanje aplikacij.

V jedrnem delu tako sledi opis razlik med enostranskimi in večstranskimi aplikacijami, njihove prednosti in slabosti, bistvene razlike med njimi ter krovna tema, v kateri opišem, kako je epidemija COVID-19 vplivala na delo od doma, ter kaj prinaša prihodnost na tem področju.

Dan danes se za razvoj z omenjenimi ogrodji odloča vse več razvijalcev pri svojih projektih. Predvsem zaradi svoje razširjene knjižnice JavaScript, delovanja virtualnega objektnega modela dokumenta, komponent ponovne uporabe ter skupnosti razvijalcev, najpopularnejši med njimi ostaja ReactJS.

V zaključku diplomskega dela so predstavljene ugotovitve hipotez in postopek izdelave projektne naloge, ki je bila izdelana z ogrodjem ReactJS.

**Ključne besede:** ReactJS, VueJS, AngularJS, zgodovina razvoja spletnih jezikov, enostranske in večstranske aplikacije

## **ABSTRACT (ali ZUSAMMENFASSUNG)**

### **Comparison of online software frameworks and development of an online questionnaire**

In my thesis, I will focus on the creation of websites with the help of ReactJS, VueJS and Angular web frameworks, present their advantages and disadvantages, and summarize the overall topic related to working from home.

Ever since the development of the Internet, websites have been based in one way or another on HTML, CSS and JavaScript technologies.

The first chapters of the thesis cover a brief presentation of the history of the development of web languages, their structure and what problems programmers faced 30 years ago, the need for their development and how they influenced the development of the most popular web frameworks that we know today.

The framework code is based on JavaScript in its variants (JSX, TypeScript). All building blocks are based on a similar mentality - create as much as possible with as little work as possible. This means that when creating web pages, it is based on the creation of components.

With the help of this, developers can reuse the code, which shortens their development time and enables more efficient testing, upgrading and maintenance of applications.

In the core part, there follows a description of the differences between one-page and multi-page applications, their advantages and disadvantages, the essential differences between them, and the overarching topic in which I describe how the COVID-19 epidemic has affected working from home, and what the future holds in this area.

Today, more and more developers decide to develop with the mentioned frameworks in their projects. ReactJS remains the most popular among them, mainly because of its extended JavaScript library, virtual document object model functionality, reusable components, and developer community.

In the conclusion of the thesis, I present the findings of the hypotheses and the process of creating the project assignment are presented.

**Keywords (ali Stichwort):** ReactJS, VueJS, AngularJS, development history of web languages, single page applications versus multi page applications.

## Kazalo vsebine

|          |                                                       |           |
|----------|-------------------------------------------------------|-----------|
| <b>1</b> | <b>UVOD .....</b>                                     | <b>6</b>  |
| 1.1      | OPIS PODROČJA IN OPREDELITEV PROBLEMA.....            | 6         |
| 1.2      | NAMEN, CILJI IN OSNOVNE TRDITVE.....                  | 7         |
| 1.3      | PREDPOSTAVKE IN OMEJITVE .....                        | 7         |
| 1.4      | UPORABLJENE RAZISKOVALNE METODE .....                 | 7         |
| <b>2</b> | <b>KRATKA ZGODOVINA RAZVOJA SPLETNIH OGRODIJ.....</b> | <b>8</b>  |
| 2.1      | HTML.....                                             | 8         |
| 2.1.1    | <i>Struktura HTML dokumenta.....</i>                  | <i>8</i>  |
| 2.2      | CSS .....                                             | 9         |
| 2.3      | JAVASCRIPT .....                                      | 9         |
| 2.4      | JQUERY .....                                          | 9         |
| 2.5      | ANGULAR .....                                         | 10        |
| 2.6      | REACT .....                                           | 10        |
| 2.7      | VUEJS.....                                            | 10        |
| <b>3</b> | <b>KNJIŽNICE IN OGRODJA.....</b>                      | <b>11</b> |
| 3.1      | KNJIŽNICE.....                                        | 11        |
| 3.2      | VARNOST KNJIŽNIC .....                                | 11        |
| 3.3      | SLABOSTI KNJIŽNIC .....                               | 11        |
| 3.4      | OGRODJA .....                                         | 12        |
| 3.5      | KOMPONENTE .....                                      | 13        |
| 3.6      | RAZREDNE KOMPONENTE .....                             | 13        |
| <b>4</b> | <b>ENOSTRANSKE IN VEČSTRANSKE APLIKACIJE .....</b>    | <b>14</b> |
| 4.1      | ENOSTRANSKE APLIKACIJE .....                          | 14        |
| 4.2      | VEČSTRANSKE APLIKACIJE .....                          | 14        |
| 4.3      | PREDNOSTI ENOSTRANSKIH APLIKACIJ .....                | 15        |
| 4.4      | SLABOSTI ENOSTRANSKIH APLIKACIJ .....                 | 15        |
| 4.5      | PREDNOSTI VEČ STRANSKIH APLIKACIJ.....                | 16        |
| 4.6      | SLABOSTI VEČ STRANSKIH APLIKACIJ.....                 | 16        |
| 4.7      | ALI SO SPA APLIKACIJE NOVA NORMA? .....               | 17        |
| <b>5</b> | <b>STRUKTURE OGRODIJ.....</b>                         | <b>18</b> |
| 5.1      | REACT .....                                           | 18        |
| 5.1.1    | <i>Komponente kot funkcije.....</i>                   | <i>18</i> |
| 5.1.2    | <i>Komponente kot razredi.....</i>                    | <i>19</i> |
| 5.1.3    | <i>Rekviziti (Props).....</i>                         | <i>19</i> |

|                                                                                                                                                                                                                                                                  |                                                |           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|-----------|
| <p><i>Ime komponente v tem primeru predstavlja ime razreda. Razred mora vedno izhajati iz nadrazreda React.Component. Razredi lahko vedno vrnejo samo en element, njihova vizualizacija pa mora biti zapisana v metodi render().(Javatpoint, 2022)</i> .....</p> |                                                | 19        |
| 5.1.4                                                                                                                                                                                                                                                            | Stanje .....                                   | 19        |
| 5.2                                                                                                                                                                                                                                                              | ANGULARJS.....                                 | 20        |
| 5.2.1                                                                                                                                                                                                                                                            | TypeScript .....                               | 20        |
| 5.2.2                                                                                                                                                                                                                                                            | Vezava podatkov.....                           | 21        |
| 5.2.3                                                                                                                                                                                                                                                            | Vmesnik ukazne vrstice (CLI).....              | 22        |
| 5.2.4                                                                                                                                                                                                                                                            | Usmerjanje.....                                | 22        |
| 5.3                                                                                                                                                                                                                                                              | VUEJS.....                                     | 23        |
| 5.3.1                                                                                                                                                                                                                                                            | Direktive.....                                 | 23        |
| 5.3.2                                                                                                                                                                                                                                                            | Modifikatorji .....                            | 23        |
| 5.3.3                                                                                                                                                                                                                                                            | Vezava podatkov.....                           | 23        |
| 5.3.4                                                                                                                                                                                                                                                            | Rekviziti (Props).....                         | 24        |
| 5.3.5                                                                                                                                                                                                                                                            | Vmesnik ukazne vrstice (CLI).....              | 24        |
| <b>6</b>                                                                                                                                                                                                                                                         | <b>PRIMERJAVA REACT, ANGULAR, VUEJS.....</b>   | <b>25</b> |
| 6.1                                                                                                                                                                                                                                                              | PREDNOSTI IN SLABOSTI REACT OGRODJA.....       | 25        |
| 6.1.1                                                                                                                                                                                                                                                            | Prednosti.....                                 | 25        |
| 6.1.2                                                                                                                                                                                                                                                            | Slabosti .....                                 | 26        |
| 6.2                                                                                                                                                                                                                                                              | PREDNOSTI IN SLABOSTI ANGULAR OGRODJA .....    | 26        |
| 6.2.1                                                                                                                                                                                                                                                            | Prednosti.....                                 | 26        |
| 6.2.2                                                                                                                                                                                                                                                            | Slabosti .....                                 | 26        |
| 6.3                                                                                                                                                                                                                                                              | PREDNOSTI IN SLABOSTI VUEJS OGRODJA.....       | 27        |
| 6.3.1                                                                                                                                                                                                                                                            | Prednosti.....                                 | 27        |
| 6.3.2                                                                                                                                                                                                                                                            | Slabosti .....                                 | 28        |
| <b>7</b>                                                                                                                                                                                                                                                         | <b>ALI SE BO DELO OD DOMA ZAKLJUČILO?.....</b> | <b>29</b> |
| <b>8</b>                                                                                                                                                                                                                                                         | <b>IZDELAVA PROJEKTNE NALOGE .....</b>         | <b>31</b> |
| 8.1                                                                                                                                                                                                                                                              | OPIS PROJEKTNE NALOGE .....                    | 31        |
| 8.2                                                                                                                                                                                                                                                              | POSTOPEK IZDELAVE PROJEKTNE NALOGE .....       | 31        |
| <b>9</b>                                                                                                                                                                                                                                                         | <b>SKLEP .....</b>                             | <b>41</b> |
| <b>10</b>                                                                                                                                                                                                                                                        | <b>VIRI IN LITERATURA .....</b>                | <b>43</b> |
| <b>11</b>                                                                                                                                                                                                                                                        | <b>PRILOGE .....</b>                           | <b>46</b> |

## Kazalo slik

|                                                                                                                                        |    |
|----------------------------------------------------------------------------------------------------------------------------------------|----|
| SLIKA 1: PRIMER HTML KODE. ....                                                                                                        | 8  |
| SLIKA 2: PRIMER RAZLAGE IN PREUSMERITVE VIR: (LASTNI VIR) .....                                                                        | 12 |
| SLIKA 3: PRIMERJAVA MPA IN SPA.....                                                                                                    | 14 |
| SLIKA 4: PRIKAZ DELOVANJA RAZLIK MED SPA IN MPA.....                                                                                   | 16 |
| SLIKA 5: PRIMERJAVA ZANIMANJA PROGRAMSKIH JEZIKOV ANGULAR, VUE.JS, REACT TER HTML SKOZI ČAS. ....                                      | 17 |
| SLIKA 6: PRIMER KOMPONENTE KOT FUNKCIJE. ....                                                                                          | 18 |
| SLIKA 7: PRIMER KOMPONENTE KOT RAZREDA. ....                                                                                           | 19 |
| SLIKA 8: DELOVANJE DVOSMERNE POVEZAVE. ....                                                                                            | 21 |
| SLIKA 9: PRIMER USMERJANJA V ANGULAR. VIR: <a href="https://angular.io/tutorial/toh-pt5">HTTPS://ANGULAR.IO/TUTORIAL/TOH-PT5</a> ..... | 22 |
| SLIKA 10: PODATKI UPORABIKOV IN RAZVIJALCEV REACT NA GITHUB.....                                                                       | 25 |
| SLIKA 11: NAMESTITEV POTREBNIH PAKETOV ZA IZDELAVO PROJEKTNE NALOGE. ....                                                              | 31 |
| SLIKA 12: PRIPRAVA MAP ZA DELJENJE BACKEND IN FRONT-END NALOGE. VIR:(LASTNI VIR) .....                                                 | 32 |
| SLIKA 13: NAMESTITEV PAKETKOV ZA ZALEDNI DEL APLIKACIJE. VIR: (LASTNI VIR) .....                                                       | 32 |
| SLIKA 14: ZAGON STREŽNIKA VIR: (LASTNI VIR) .....                                                                                      | 33 |
| SLIKA 15: IZGLED VSTOPNE SPLETNE STRANI. VIR: (LASTNI VIR).....                                                                        | 34 |
| SLIKA 16: USTVARI AN-KETO VIR: (LASTNI VIR) .....                                                                                      | 34 |
| SLIKA 17:PRIMER SESTAVE ANKETE ZA COVID-19 VPRAŠALNIK 1/2. VIR: (LASTNI VIR).....                                                      | 35 |
| SLIKA 18: PRIMER SESTAVE ANKETE ZA COVID-19 VPRAŠALNIK 2/2. VIR: (LASTNI VIR).....                                                     | 35 |
| SLIKA 19: PRIKAZ STRANI Z OBSTOJEČIMI ANKETAMI. VIR: (LASTNI VIR).....                                                                 | 37 |
| SLIKA 20: STRAN, NA KATERI SE PRIKAŽE ANKETA ZA REŠEVANJE. VIR: (LASTNI VIR).....                                                      | 37 |
| SLIKA 21: GUMB ZA ODDAJO ANKETE. VIR: (LASTNI VIR) .....                                                                               | 38 |
| SLIKA 22: POJAVNO OKNO PO ODDAJI ANKETE 1/2. VIR: (LASTNI VIR) .....                                                                   | 38 |
| SLIKA 23: POJAVNO OKNO PO ODDAJI ANKETE 2/2. VIR: (LASTNI VIR) .....                                                                   | 40 |

# 1 UVOD

Z eksponentnim napredkom razvoja tehnologije vsakih 14 mesecev človeštvo uspe podvojiti število tranzistorjev v računalnikih. To je odkril Gordon Moore leta 1965 in po 56 letih ta zakon še vedno drži (E. Moore, 1965).

Od začetka razvoja svetovnega spleta s strani ameriške vojske, ki so ga uporabljali za prenos podatkov, do popolne digitalizacije poslovanja podjetjih, ki uporabljajo spletne strani za širjenje poslovanja, spletnih konferenc ter v večini primerov digitalnega marketinga, spletne strani omogočajo večjim, pa tudi manjšim podjetjem prodor na trg. Digitalizacija spletnega poslovanja je bila v procesu izpeljave pri večini podjetij, vendar je epidemija virusa COVID-19 to samo posplošila. Zmanjšala se je uporaba prevozov na konference, zaradi digitalnega poslovanja se je prav tako zmanjšala uporaba dokumentov v papirnati obliki.

Obdobje digitalizacije nam je dostopno že na vseh elektronskih napravah, zato je spletna stran za podjetja že skoraj obvezna, saj je brez spletne strani in pravičnega oglaševanja podjetje skorajda neobstoječe na trgu. Spletna stran je pomembna za širjenje kredibilnosti in znamke, lažjega dostopa do ciljne publike s pomočjo oglaševanja umetne inteligence ter hitrejšega in učinkovitejšega obveščanja uporabnikov (Kaplan, 2020).

Tehnologij za razvoj spletnih strani imamo že precej, vendar so v zadnjem desetletju na vzpon prišla spletna ogrodja, ki delujejo na princip sprotnega osveževanja, kar pripomore pri hitrejši odzivnosti spletne strani, brez nepotrebnih klicev drugih datotek, saj ogrodje samo preveri, kje se bile spremembe narejene in jih prepíše.

## ***1.1 Opis področja in opredelitev problema***

Pri diplomskem delu se bom osredotočil na tehnologije spletnih ogrodij, jih med seboj primerjal ter v izbranem ogrodju izdelal spletni vprašalnik, ki ga bodo lahko uporabniki sestavili sami. Vprašanja in odgovori se bodo zapisovali v podatkovno bazo, ki jo bo uporabnik imel možnost izvoziti v dokument.

## ***1.2 Namen, cilji in osnovne trditve***

Namen izdelave spletnega vprašalnika je izboljšanje obstoječih podatkov, predvsem pa možnost izvažanja podatkov v Excelovo preglednico, kjer bodo lahko statistično obdelani. S pomočjo diplomskega dela želim odkriti, katero ogrodje je najprimernejše za tovrstno nalogo, kaj so prednosti in slabosti ostalih ogrodij, ter ugotoviti, katera tehnologija je najprimernejša za razvoj spletnih strani za nove razvijalce.

Pri raziskovalnem delu diplomskega dela sem si zadal sledeče hipoteze:

- ogrodje Angular je najpopularnejše ogrodje leta 2021,
- ogrodja, kot so Angular, Vue in React bodo nadomestila splošne spletne jezike,
- sintaksa JavaScript ogrodij je prenosljiva,
- uporaba orodij za oddaljeno komunikacijo lahko nadomesti delo v službi.

## ***1.3 Predpostavke in omejitve***

Omejitve, s katerimi se bom soočal pri pisanju diplomskega dela in izdelavi izdelka, so:

- statistični podatki uporabe sledečih spletnih vmesnikov so stari nekaj let in niso posodobljeni.
- Zaradi varovanja podatkov poslovne skrivnosti bodo odgovori na moja vprašanja omejeni.
- Pri intervjuju bom po vsej verjetnosti dobil subjektivne odgovore, saj se intervjuvanec pri svojem delu srečuje samo z vmesnikom Angular.

## ***1.4 Uporabljene raziskovalne metode***

Pri izdelavi diplomskega dela so bile uporabljene sledeče raziskovalne metode:

- intervju,
- spletni vprašalnik,
- analiza podatkov.

## 2 KRATKA ZGODOVINA RAZVOJA SPLETNIH OGRODIJ

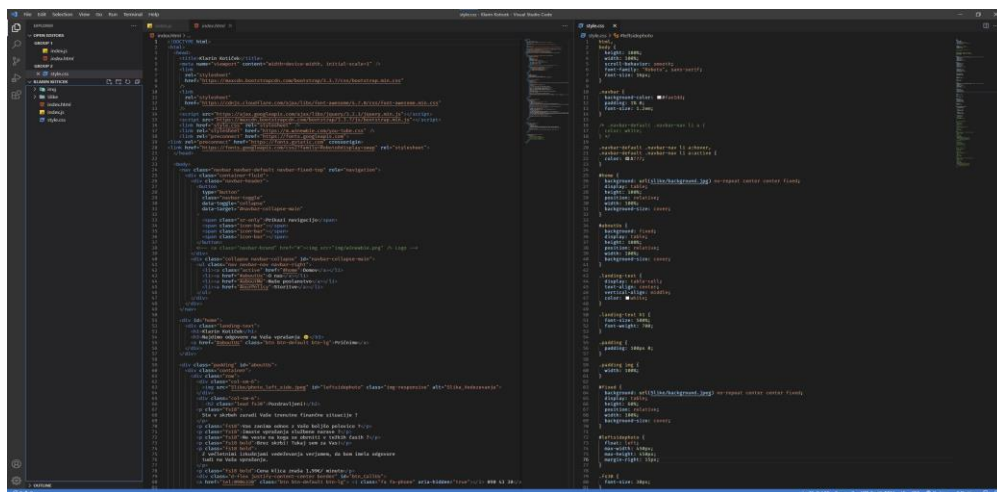
### 2.1 HTML

Zgodovina spletnih strani sega nazaj vse v leto 1991, ko je Tim Berners-Lee javno objavil prvotno specifikacijo HTML (Hyper-Text Markup Language) spletnega jezika. HTML 1.0 je na začetku vseboval samo 18 etiket (ang. Tag). Na podlagi sprememb in trendov na trgu spletnih strani so programerji ugotovili, da trenutne strani niso dovolj dinamične. Za zadostitev tem potrebam je v razvoj prišel CSS (Cascading Style Sheets – kaskadne stilske podloge), ki ga je predlagal Håkon Wium Lie oktobra 1994.

#### 2.1.1 Struktura HTML dokumenta

Vsak HTML dokument je osnovno strukturiran z značkami po istem principu:

- **<!DOCTYPE html>** - definira dokument tipa HTML;
- **<head>** - glava dokumenta. Vključuje metapodatke (tj. Podatke, ki niso vidni uporabnikom). Primer teh podatkov so naslovi spletnih strani, povezave do knjižnic, stil strani in podobno;
- **<body>** - značka, ki definira telo dokumenta. Tukaj se nahajajo vsi vidni deli spletne strani. Zajema vse od naslovov, podnaslovov, tekstov, hiperpovezav, tabel, slik in marsičesa drugega.



Slika 1: Primer HTML kode.

Vir: (Lastni vir).

## 2.2 CSS

Za razvoj CSS so potrebovali 2 leti po predlogu potreb po le-tem. HTML 4.0 je bila prva verzija, ki je podpirala CSS, Internet Explorer 3 pa je postal prvi komercialni brskalnik s podporo CSSa. Sintaksa CSS je preprosta, saj uporablja veliko večino ključnih besed angleškega slovarja, ki specificirajo stilske lastnosti. CSS deluje na princip selektorjev, ki se sklicujejo na ujemajoče se etikete v datoteki HTML. Med razvojem in publikacijo stilskih predlog pa je v ozadju leta 1995 pričel potekati razvoj skriptnega jezika JavaScript (MDN contributors, 2022).

## 2.3 JavaScript

Po potrebi razvoja dinamičnih spletnih strani je v svet vstopil skriptni jezik Mocha, ki je bil kasneje preimenovan v LiveScript, nato pa v JavaScript. Razvoj skriptnega jezika JavaScript se je končal septembra 1995 v samo desetih dneh, s strani Netscape. Čeprav se jezik imenuje JavaScript, je bil ta skriptni jezik razvit neodvisno od programskega jezika Java, kvečjemu ima večjo podobnost z programskim jezikom C, saj JavaScript, kot C, nima vgrajenih vhodno izhodnih funkcij, kar pomeni, da je njegova izvedba neodvisna od gostitelja. Njegova pomembnost v razvoju spletnih strani je postavila standarde skriptnih jezikov in bila predana ECMI (European Computer Manufacturers Association) (MDN contributors, 2022).

## 2.4 jQuery

jQuery je knjižnica elementov z dokumentom objektnega modela (ang. DOM – Document Object Model). Knjižnica jQuery poenostavlja sintakso iskanja izbiranja in upravljanja domen. jQuery je prišel v razvoj leta 2005 in postal uradno objavljen leta 2006 na dogodku BarCampNYC. jQuery ponuja zmogljive metode za izbiranje in manipuliranje elementov. V zelo kratkem času je jQuery dosegel visoko št. uporabnikov in po trenutnih predvidevanjih jQuery uporablja kar 74 % spletnih strani (Guo, 2019).

## **2.5 Angular**

Angular je JavaScript ogrodje, ki je napisano v jeziku TypeScript in je razvito ter vzdrževano s strani podjetja Google. Tekom svojega obstoja je Angular prešel mnogo sprememb. Prvotna verzija je bila objavljena leta 2010, prvo večjo nadgradnjo pa je prejel leta 2016, ko se je Angular preimenoval v AngularJS, ki se sklicuje na verzijo 2 ali več. Ker se AngularJS uporablja pri Google AdWords, eden izmed pomembnejših projektov Googla za oglaševanje, lahko spekuliramo, da bo Angular še nekaj časa redno vzdrževan (Dizowki, 2022).

## **2.6 React**

React, ki je v svetu razvijalcev poznan tudi kot ReactJS ali React.js, je brezplačno odprtokodno začetno JavaScript ogrodje, ki se uporablja pri gradnji spletnih uporabniških vmesnikov. Vzdrževan je s strani podjetja Meta (v preteklosti poznan kot Facebook) ter raznih posameznikov in podjetij. React je eden izmed gradnikov, ki se uporablja za gradnjo enostranskih aplikacij (angl. SPA – Single-page application). Ker pri izdelavi spletnih strani z ReactJS upravlja samo upravljanje stanj (angl. state management), za spreminjanje DOM-a, je po navadi potrebno uporabljati različne knjižnice za doseg željenih funkcionalnosti (Rodriguez Martinez, 2017).

## **2.7 VueJS**

VueJS je prišel v fazo razvoja leta 2013 s strani programerja Evan You, ki je bil v preteklosti zaposlen v podjetju Google. VueJS je uradno postal objavljen februarja 2014 in ga od takrat aktivno uporablja veliko večjih podjetij, kot na primer Alibaba, Baidu, Xiaomi, Sina Weibo, Gitlab, ... VueJS je umeščen tudi v jedra ogrodij, kot sta Laravel in PageKit. Leta 2016 je bila objavljena verzija 2.0, kateri so svoji uporabniki pravili "Vue na steroidih", zaradi izboljšave stabilnosti in učinkovitosti. Danes, zaradi rednih nadgradenj in izboljšav, VueJS konkurira ReactJS ter AngularJS ogrodju (Grybniak, 2021).

## 3 KNJIŽNICE IN OGRODJA

### 3.1 *Knjižnice*

Knjižnice v programerskem svetu predstavljajo vnaprej spisane kode, ki so lahko dodane v projekt, z namenom olajšanja razvoja. Vsebujejo vnaprej pripravljene razrede, procedure, skripte ali izračune, ki skrajšajo razvoj kompleksnejših programerskih problemov. V kolikor te ne obstajajo, so programerji prepuščeni samostojnemu razvoju in iskanju rešitve njihovega problema. V kolikor pa se strokovnjaki, ki so rešili nek problem, odločijo, da to delijo s svetom (torej ustvarijo odprtokoden projekt), olajšajo delo ostalim programerjem, ki se soočajo s podobnim problemom, in lahko uporabijo njihovo rešitev. V osnovi knjižnice vsebujejo veliko število osnovnih in/ali kompleksnejših algoritmov ter funkcij, ki so bile že razvite s strani ostalih programerjev v tisti skupnosti. Tukaj je potreben poudarek na besedi skupnost, saj se vsak programer srečuje s svojim jezikom in svojimi problemi. Ni rečeno, da bo nekdo, ki se ukvarja z razvojem v .NET, imel knjižnice z istimi ali podobnimi funkcionalnostmi kot programer, ki pri svojem delu dela z Javo ali React-om.

### 3.2 *Varnost knjižnic*

Seveda pa lahko vse te odprtokodne knjižnice predstavljajo za podjetja ali posameznike varnostno tveganje pri njihovem projektu. V kolikor se programer odloči za uporabo zunanje knjižnice, je priporočljivo, da pred integracijo v projekt sam preveri verodostojnost knjižnice, bodisi sam ali preko pregleda mnenj ostalih uporabnikov. Dober primer predstavlja popularna stran za prenos ilegalnih vsebin preko protokola P2P Pirate Bay. Leta 2017 je uporabnik na svojo datoteko naložil skripto, ki je na posameznikovem računalniku rudarila kripto valute, ob prenosu datoteke. Podobno (ali celo hujše) varnostno tveganje predstavlja uporaba neznanih knjižnic pri projektu.

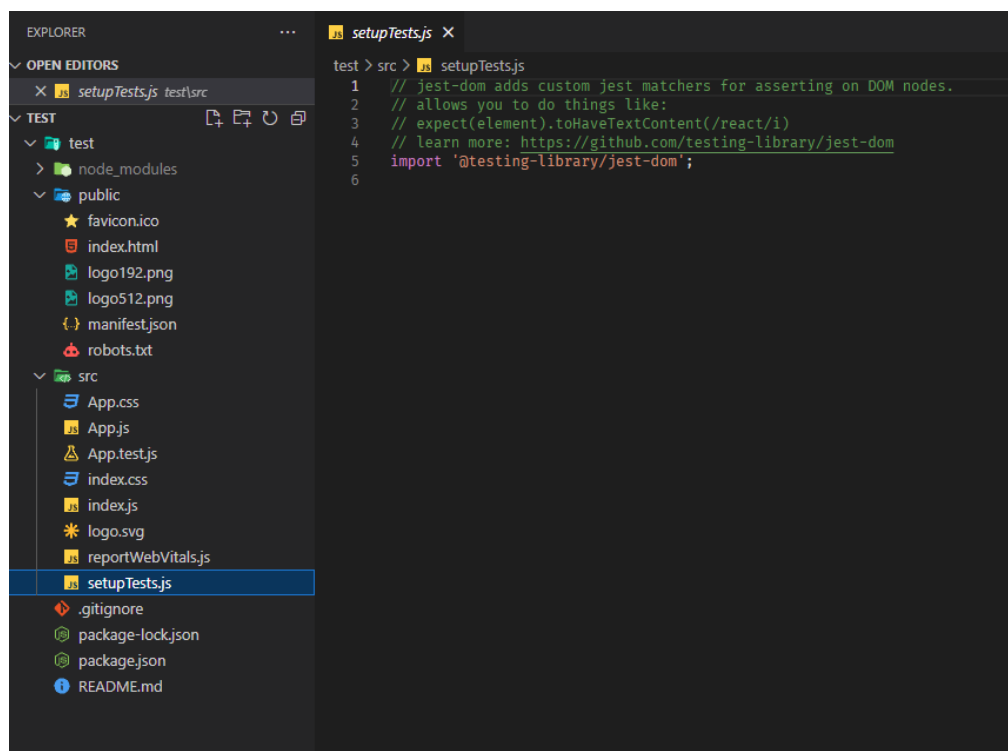
### 3.3 *Slabosti knjižnic*

Čeprav se sprva sliši, kot da lahko knjižnice rešijo vse probleme, temu žal ni tako. V določenih primerih lahko povzročijo več škode kot dobrega. Načeloma lahko, ob (skoraj) vsaki posodobitvi ogrodja, vmesnika ali jezika, povzroča težave kompatibilnosti, ki lahko povzročijo nedelovanje programa. Zato, v kolikor se gre za kompleksnejše zadeve in ima uporabnik/podjetje možnost razvoja svoje rešitve, raje posveti čas temu, saj je to kodo potem

lažje popravljati in nadgrajevati skupaj s projektom, kot pa čakati (ali v nekaterih primerih celo upati), da se bo knjižnica posodobila, ki bo omogočila ponovno normalno delovanje programa. Pogost primer tega je popularna knjižnica Bootstrap, ki obstaja v različnih verzijah. V takratni službi sem se soočal ravno s tem problemom in verjamem, da nisem edini, saj so na novem projektu, ki je izviral iz obstoječega starejšega projekta, želeli novo verzijo knjižnice. Na koncu se je izšlo tako, da mi je vzelo več časa, da sem posodobil kodo in nadgradil knjižnico, kot pa da bi projekt ustvaril od začetka in postopoma gradil na njem.

### 3.4 Ogradja

V primerjavi s knjižnicami predstavljajo ogrodja uporabne funkcije, ki programerju omogočajo izdelavo projekta s klici le-teh. V kolikor te ne bi obstajale, bi bili programerji bili sami spisati aplikacijsko kodo. To zajema vse od URL poti, logike obnašanja programa, povezovanja zalednega in začetnega dela ... Večina današnjih ogrodij ima tako vnaprej spisano podrobno dokumentacijo delovanja ter ob ustanovitvi projekta s komentarji v kodi razloži programerju osnovno delovanje programa in usmeri programerja na spletno stran dokumentacije, kjer si lahko prebere več o specifični zadevi.



Slika 2: Primer razlage in preusmeritve

Vir: (Lastni vir)

### **3.5 Komponente**

V primerjavi s knjižnicami predstavljajo ogrodja uporabne funkcije, ki programerju omogočajo izdelavo projekta s klici le-teh. V kolikor te ne bi obstajale, bi bili programerji bili sami spisati aplikacijsko kodo. To zajema vse od URL poti, logike obnašanja programa, povezovanja zalednega in začetnega dela ... Večina današnjih ogrodij ima tako vnaprej spisano podrobno dokumentacijo delovanja ter ob ustanovitvi projekta s komentarji v kodi razloži programerju osnovno delovanje programa in usmeri programerja na spletno stran dokumentacije, kjer si lahko prebere več o specifični zadevi.

### **3.6 Razredne komponente**

Razredne komponente so ključnega pomena pri ogrodjih, saj predstavljajo gradnike in naredijo ogrodje modularno. Z uporabo pravih komponent lahko programerji zagotovijo, da je začetna koda ponovno uporabna in razširljiva. Uporabljajo se, ko je potrebno voditi stanje aplikacije, potrebno pridobiti podatke iz strežnika. Te skrbijo za sproščanje virov v aplikaciji in pospešijo delovanje.

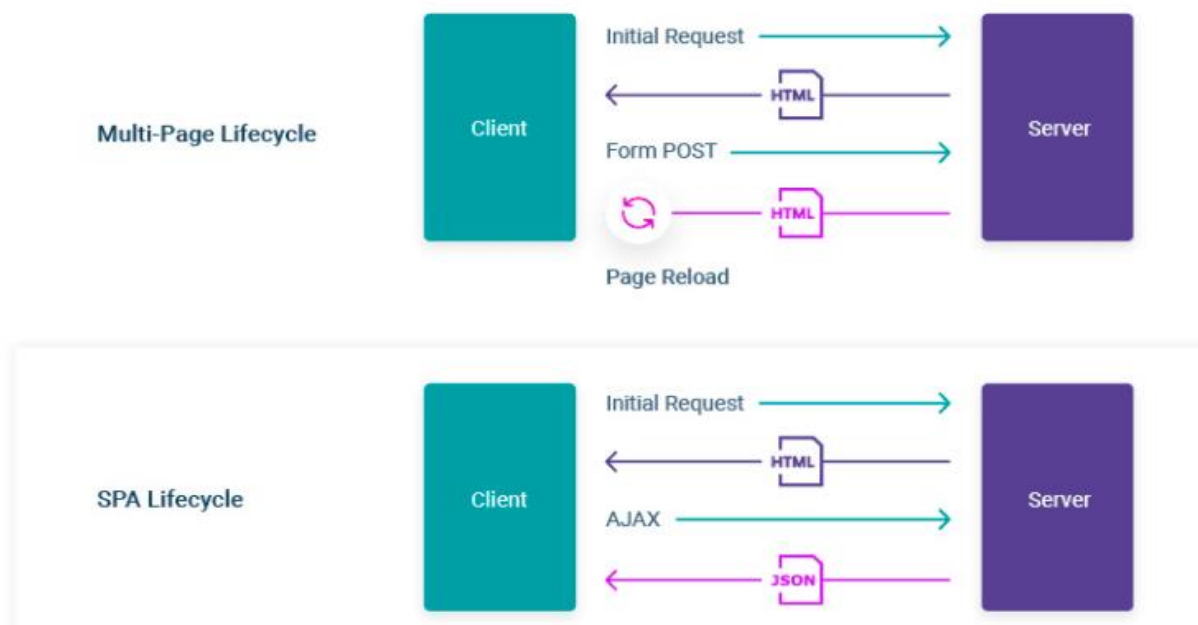
## 4 ENOSTRANSKE IN VEČSTRANSKE APLIKACIJE

### 4.1 Enostranske aplikacije

Enostranske aplikacije (angl. SPA – Single Page Application) so spletne aplikacije (tj. spletne strani), ki komunicirajo z uporabnikom tako, da dinamično prepisujejo kodo trenutne spletne strani, z novimi podatki spletnega strežnika. Enostranske aplikacije delujejo na princip "ne osvežuj". To pomeni, da se spletna stran nikoli ne osveži, ampak ob kakšni interakciji uporabnika pogleda spremembe v kodi (kakšne spremembe ta interakcija sproži) in prepiše določen del spletne strani. V preteklosti je bilo mogoče izdelati samo tako imenovane večstranske aplikacije (angl. MPA – Multi-page Application) (Flanagan, 2011).

### 4.2 Večstranske aplikacije

Večstranske aplikacije se od enostranskih aplikacij razlikujejo v delovanju. Medtem ko se enostranske aplikacije ne osvežujejo, je pri večstranskih aplikacijah pri vsaki interakciji potrebna osvežitev spletne strani, da se posodobijo podatki. Tehnično to pomeni, da se s strani odjemalca (uporabnika) pošlje zahteva na strežnik, strežnik nazaj pošlje odgovor (angl. Response) – v tem primeru nove dokumente in ko je zahteva uspešno zaključena, se osveži spletna stran za prikaz podatkov.



Slika 3: Primerjava MPA in SPA.

Vir: (<https://lvivcity.com/single-page-app-vs-multi-page-app>).

### 4.3 *Prednosti enostranskih aplikacij*

- **Delovanje:** vsa potrebna sredstva za delovanje spletne strani so naložene ob prvotnem obisku spletne strani. Vse potrebne spremembe se izvedejo dinamično ob interakciji.
- **Boljša uporabniška izkušnja:** Enostranske aplikacije uporabnikom zagotavljajo bolj razumljivo linearno izkušnjo. Z pomočjo ogrodij (kot npr. AJAX) je možno ustvariti tudi bolj fleksibilno in odzivno spletno stran.
- **Podatkovno predpolnjenje:** po prvotni poizvedbi strežnika se vsi potrebni lokalni podatki shranijo v skladišče, ki omogoča uporabnikom delo brez povezave (primer takšne aplikacije je Google Docs).
- **Hitrejši razvoj:** zaradi manjše količine datotek, potrebnih za delovanje spletne strani, nam pohitri razvoj, saj moramo posledično razviti in testirati manj stvari.
- **Kompatibilnost:** prav tako je potrebno omeniti, da z razvojem aplikacij s takšnimi ogrodji (React, AngularJS, Vue.JS) ni potrebno posebej prilagajati funkcionalnosti za delovanje na mobilnih napravah, saj so avtohtoni za le-te.

### 4.4 *Slabosti enostranskih aplikacij*

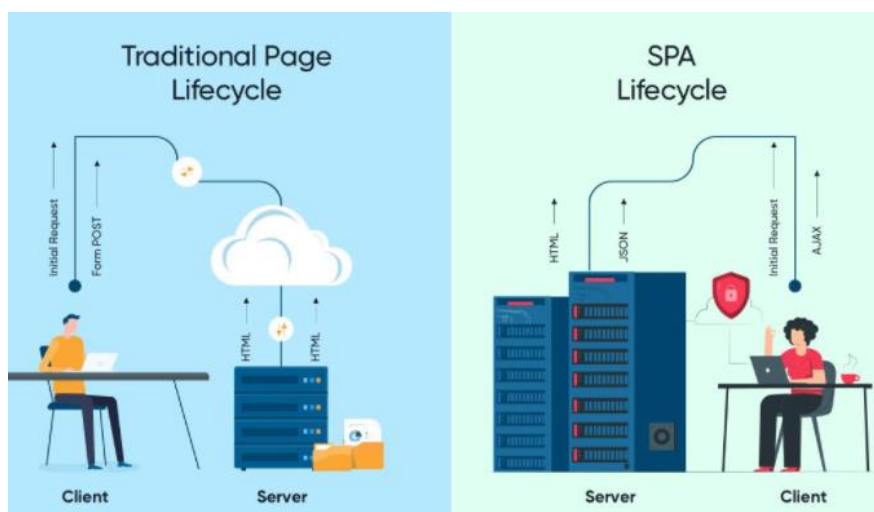
- **SEO optimizacija:** SEO optimizacija je na splošno povedano optimizacija spletnih strani pri uporabi iskalnikov. Večino spletnih strani lahko optimiziramo s kraticami URL naslovov, ampak pri enostranskih aplikacijah to ni mogoče, saj se podatki naložijo brez nalaganja in samo ob zahtevi uporabnika.
- **Nalagalni čas:** če je aplikacija kompleksna ali velika in slabo optimizirana, bo trajalo dlje časa, da bo brskalnik prikazal vsebino.
- **Obvezna podpora JavaScripta:** brez omogočenega JavaScript vmesnika v brskalniku je lahko uporaba spletne strani omejena.

## 4.5 Prednosti več stranskih aplikacij

- **SEO Optimizacija:** ker ima aplikacija več strani in vsaka od njih predstavlja ključno besedo v URL naslovu, lahko iskalnik (npr. Google) avtomatično prilagodi iskanju vsebine, ko oseba išče po ključnih besedah.
- **Enostavnost skaliranja:** ta tip arhitekture omogoča ustvarjanje kolikor hočete novih strani za vsak izdelek ali storitev in izvajanje kakršnih koli sprememb na njih.
- **Na voljo že pripravljene rešitve:** razvoj MPA praviloma zahteva uporabo manj tehnologij, poleg tega pa je na voljo široka paleta že pripravljenih rešitev (CMS).

## 4.6 Slabosti več stranskih aplikacij

- **Možne težave pri učinkovitosti aplikacije:** v primeru velikega števila zahtev in potrebe po ponovnem nalaganju velikega števila strani bosta hitrost in zmožljivost neizogibno padli. To še posebej velja za spletne strani z večjim številom obiskovalcev.
- **Začetna in zaledna integracija:** komponente teh spletnih aplikacij so praviloma globoko integrirane, zato lahko traja dlje, da jih razvijemo in preizkusimo.
- **Vzdrževanje in posodobitve:** v določenih primerih spletnih strani, ki uporabljajo več dokumentov, lahko posodabljanje spletnih strani predstavlja zaskrbljujočo nalogo tehnične pomoči, saj če nisi sam postavil strani, si jih ustrezno komentiral ali ne veš, kako deluje, porabiš več časa, da preučiš proces delovanja in pomagaš pri težavah.



Slika 4: Prikaz delovanja razlik med SPA in MPA.

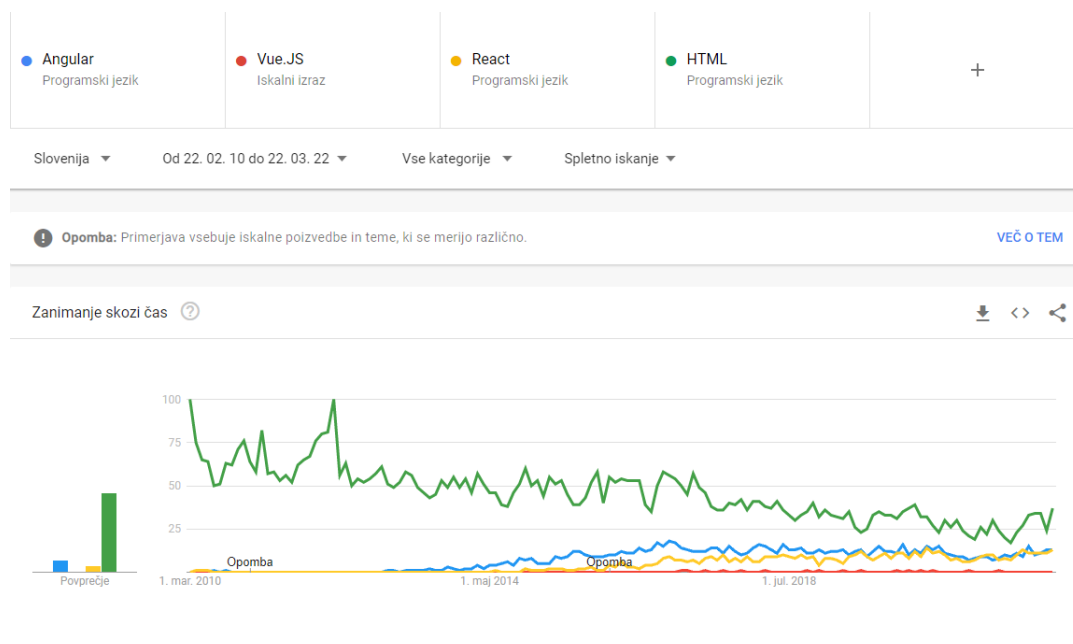
Vir: ([https://miro.medium.com/max/1400/1\\*vzv4-iUuuwgRWLAlcxzJKA.jpeg](https://miro.medium.com/max/1400/1*vzv4-iUuuwgRWLAlcxzJKA.jpeg)).

## 4.7 Ali so SPA aplikacije nova norma?

Kot omenjeno v preteklih poglavjih, opisujem veliko dobrih stvari glede SPA aplikacij.

Dosti programerjev se tudi srečuje s vprašanji, kot so »Ali so SPA aplikacije nova norma?«, »Ali bo potrebno v podjetju predelati obstoječe strani z novo tehnologijo?«, »Ali tehnologija podpira naše zahteve spletne strani?«. Odgovor na vsa vprašanja je na kratko povedano ne. Vsako podjetje mora uporabljati tehnologijo, ki ustreza specifično njihovim potrebam. Čeprav sem sam že videl tehnologijo, uporabljeno pri marsikaterem podjetju, in uporabljam njihove aplikacije na dnevni bazi (Google Maps, Netflix, Facebook, YouTube, ...), še ne pomeni, da so MPA aplikacije zastarele. Pri intervjuju z g. Robertom Španom sem med drugim postavil tudi to vprašanje.

Za primerjavo zanimanja specifičnih jezikov sem uporabil orodje Google Trends, ki prikaže zanimanje trendov skozi čas. Primerjal sem Angular, Vue.JS, React ter HTML.



Slika 5: Primerjava zanimanja programskih jezikov Angular, Vue.JS, React ter HTML skozi čas.

Vir: (Lastni vir).

Kot je razvidno iz slike, je zanimanje za HTML postopoma začelo padati, ob prihodu tehnologij React, Angular ter Vue.JS. Osebnostno sem mnenja, da nas tehnologija HTML ne bo zapustila v kratkem času. Ne samo, da ima še vedno svoje prednosti (omenjene v začetku poglavja), ampak tudi strošek, ki bi ga predstavljalo podjetju, da predela celotno arhitekturo spletne strani na novo tehnologijo, bi bil zelo velika investicija, ki je marsikatero podjetje ne podpre.

## 5 STRUKTURE OGRODIJ

### 5.1 *React*

Jedro ReactJS predstavljajo komponente in njihova sestava. Splošni cilj je preoblikovati določeno stanje aplikacije v pogled, ki ga je možno prikazati v brskalniku.

Z ReactJS je možno ustvariti komponente z dvema različnima pristopoma:

- komponente kot funkcije,
- komponente kot razredi.

#### 5.1.1 Komponente kot funkcije

Komponente kot funkcije, kot ime pove, predstavljajo funkcije, ki vrnejo točno eno vrednost - `ReactElement` (element, ki se prikaže na zaslonu). Funkcije lahko prejmejo `Prop` (izhaja iz besede `properties` - lastnosti), s katerimi lahko programerji pošiljajo podatke med komponentami. Ime komponente predstavlja ime funkcije.

```
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>;  
}
```

Slika 6: Primer komponente kot funkcije.

Vir: <https://reactjs.org/docs/components-and-props.html>

### 5.1.2 Komponente kot razredi

Ime komponente v tem primeru predstavlja ime razreda. Razred mora vedno izhajati iz nadrazreda `React.Component`. Razredi lahko vedno vrnejo samo en element, njihova vizualizacija pa mora biti zapisana v metodi `render()`. (W3Schools, 2022)

```
class Welcome extends React.Component {  
  render() {  
    return <h1>Hello, {this.props.name}</h1>;  
  }  
}
```

Slika 7: Primer komponente kot razreda.

Vir: <https://reactjs.org/docs/components-and-props.html>

### 5.1.3 Rekviziti (Props)

Ime komponente v tem primeru predstavlja ime razreda. Razred mora vedno izhajati iz nadrazreda `React.Component`. Razredi lahko vedno vrnejo samo en element, njihova vizualizacija pa mora biti zapisana v metodi `render()`. (Javatpoint, 2022)

### 5.1.4 Stanje

Stanje (ang. State) je struktura, ki jo je mogoče posodobiti in se uporablja za shranjevanje podatkov ali informacij o komponenti. Stanje v komponenti se lahko sčasoma spremeni. Sprememba stanja skozi čas se lahko zgodi kot odziv na dejanje uporabnika ali sistemski dogodek.

Dogodek mora biti čimbolj preprost. Nastavi se z uporabo metode `setState()`, katere metoda sproži posodobitve uporabniškega vmesnika. Za nastavitev začetnega stanja, preden pride do kakršne koli interakcije, moramo uporabiti metodo `getInitialState()`.

Na primer, če imamo pet komponent, ki potrebujejo podatke ali informacije iz stanja, potem moramo ustvariti eno komponento vsebnika, ki bo ohranila stanje za vse. (Meta Platforms, Inc., 2022)

## 5.2 *AngularJS*

Angular je bil eden izmed prvih ogrodij za razvoj enostranskih aplikacij, s funkcijami, kot so dvosmerna vezava podatkov in možnosti organiziranja modulov za uvoz zunanjih skript. Prav tako je njegova skupina razvijalcev zagotovila zelo izčrpno dokumentacijo, ki je bila med razvijalci zelo dobro sprejeta.

### 5.2.1 TypeScript

Med pomembnejše novosti razvoja Angularja sodi TypeScript. Razvil se je kot naslednik JavaScripta ter postal primarni jezik za programiranje v ogrodju Angular. TypeScript je nadgrajena verzija ECMAScripta (poznani tudi kot ES6), katerega je razvilo in še danes vzdržuje podjetje Microsoft. Tako TypeScript kot Angular sta oba odprtokodna ter prinašata napredne programerske zmogljivosti, kot so:

- "for..of" in "for..in" zanke – omogočijo lažje sprehajanje skozi vrednosti,
- funkcije puščic – prihranijo čas in poenostavijo obseg funkcij,
- modifikatorji vmesnikov – nadzorujejo dostopnost članov razreda (javni ali zasebni).

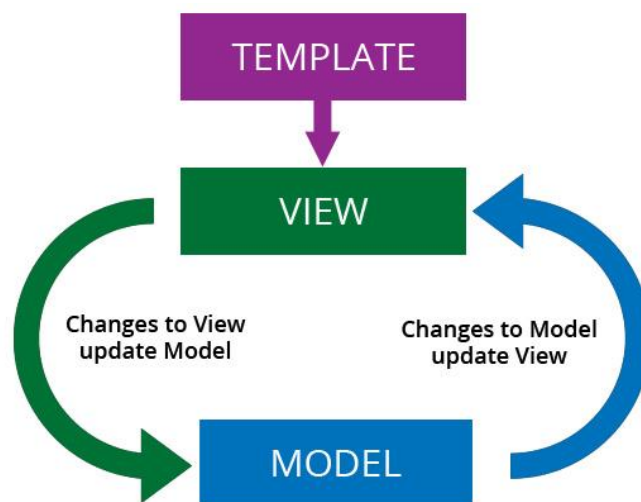
(Codecraft, 2022)

### 5.2.2 Vezava podatkov

Vezava podatkov na splošno pomeni izmenjavo podatkov med pogledom (npr. HTML dokument) in modelom (npr. TypeScript datoteka). To povezavo je mogoče uporabiti za sinhronizacijo pogleda z modelom, za obveščanje modela, ko se v pogledu zgodi dogodek ali dejanje uporabnika, ali oboje. (Google, 2022)

Angular pozna več različnih tipov vezav podatkov. To so:

- vezave atributov - omogočajo direktno nastavitev vrednosti atributov,
- razredne in stilske vezave - omogočajo uporabo razredov CSS knjižnice, kar omogoča programerjem urejanje stila znotraj razreda elementov,
- dogodkovne vezave - omogočajo odzive na uporabnikova dejanja (kot npr. premik miške, kliki ...),
- lastninska vezava - omogoča nastavitev lastnosti HTML elementov in direktiv. To zajema funkcionalnost gumbov, nastavitev poti ali delitev vrednosti med komponentami,
- dvosmerne vezave - omogočajo aplikaciji deljenje podatkov (npr. posodobitev vrednosti istočasno med komponento starša in otroka).



Slika 8: Delovanje dvosmerne povezave.

Vir: <https://stackoverflow.com/questions/13504906/what-is-two-way-binding>

### 5.2.3 Vmesnik ukazne vrstice (CLI)

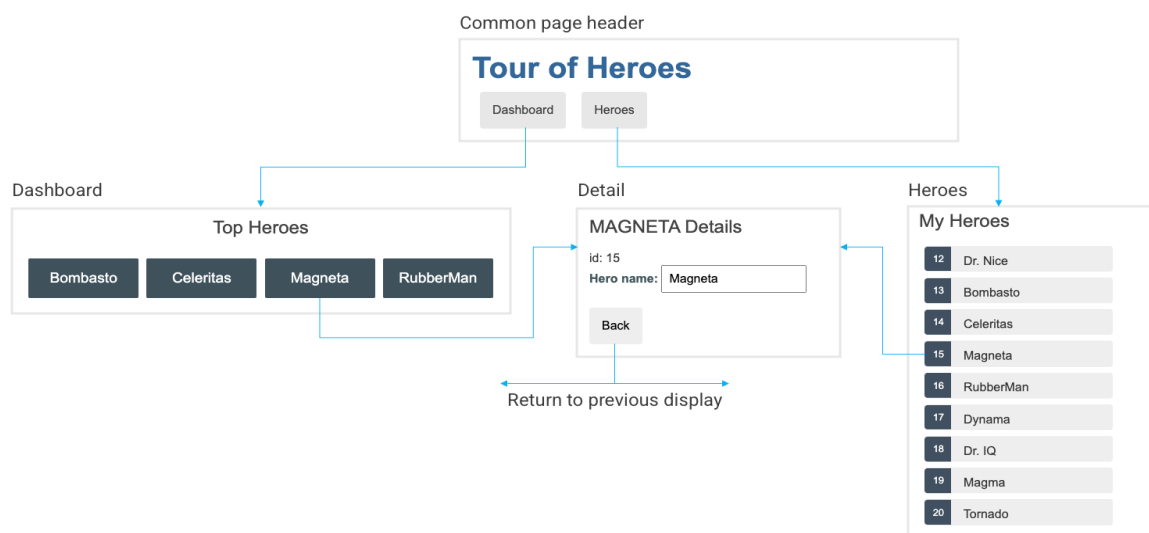
Vmesnik ukazne vrstice (ang. Command-line interface ali CLI) je orodje, katerega se uporablja za inicializacijo, razvoj, gradnjo in vzdrževanje Angular aplikacij neposredno iz ukazne vrstice. Služi kot začetna točka za vse Angular aplikacije. Prav tako zajema veliko uporabnih ukazov, ki obogatijo razvojni proces. Najpomembnejši so:

- **ng-serve** - omogoča testiranje aplikacije lokalno,
- **ng-build** - prevede aplikacijo ali knjižnico v izhodni imenik,
- **ng-generate** - generira ali modificira datoteke na podlagi sheme.

(Google, 2022)

### 5.2.4 Usmerjanje

Usmerjanje (ang. Routing) igra pri ogrodjih, kot je Angular, eno izmed najpomembnejših vlog. Ta vloga je usmerjanje po spletni strani. V modulu usmerjevalnika je definirano polje preusmeritev, ki zajema pravila, kako in kdaj se uporabnik sklicuje na določen URL. Angular s funkcionalnostjo usmerjanja obravnava logiko, katere komponente morajo biti prikazane glede na trenutno URL pot. (Google, 2022)



Slika 9: Primer usmerjanja v Angular.

Vir: <https://angular.io/tutorial/toh-pt5>

## 5.3 VueJS

Edinstvena značilnost VueJS (v nadaljevanju Vue) je, da jo je v celoti ustvarila odprtokodna skupnost in ne veliko podjetje. Evan You se je s tem začel ukvarjati kot s hobijem, na koncu pa odločil, da pusti službo, saj je prejel finančno pomoč s strani Patreona. Patreon je socialno omrežje za proizvajalce vsebin na različnih področjih, kot so razvoj iger, glasba, pisanje, ustvarjanje video vsebin ..., kjer se lahko ljudje naročijo na določen projekt, kateremu želijo mesečno prispevati sredstva. Podporniki projekta Vue so v skupno namenili že 4000 dolarjev. Kasneje je Evan You sporočil, da bo zaradi finančnih razlogov premestil svoj sponzorski program na Github Sponsors. (VueJS, 2022)

### 5.3.1 Direktive

Vue uporablja direktive v svojih predlogah, tako kot Angular. Uporabljajo se lahko za upravljanje dogodkov, vezavo podatkov in druge stvari. Na primer, direktiva v-bind se uporablja za reaktivno spreminjanje atributa HTML. (VueJS, 2022)

### 5.3.2 Modifikatorji

Uporaba modifikatorjev je še eno strokovno področje za Vue. Pika služi kot vizualni znak, da gre za posebne pripone. To nakazuje, da bi morala biti direktiva na primer vezana drugače kot privzeto. Poleg tega so lahko številni modifikatorji povezani skupaj. (VueJS, 2022)

### 5.3.3 Vezava podatkov

Vue.js uporablja implementacijo predlog, ki temelji na DOM. V bistvu so vse predloge Vue.js legalen HTML, ki ga je mogoče razčleniti in ima nekaj edinstvenih značilnosti. Zapomnite si to, saj bistveno razlikuje predloge Vue od predlog, ki temeljijo na nizih. (VueJS, 2022)

- Enosmerna vezava podatkov - podatki vedno tečejo v eno smer, kar olajša razumevanje delovanja.
- Dvosmerna vezava podatkov - ne glede na to, kje izvira sprememba, enosmerna vezava sinhronizira pogled in model. (VueJS, 2022)

#### **5.3.4 Rekviziti (Props)**

Ta funkcija Vue je primerljiva z Reactom. Uporaba rekvizitov je eden od načinov za sporočanje podatkov od nadrejenih do podrejenih komponent. Prenašajo se lahko statično ali dinamično, vendar je treba opozoriti, da lahko rekvizit prejme kakršno koli vrednost. (VueJS, 2022)

#### **5.3.5 Vmesnik ukazne vrstice (CLI)**

Vue omogoča dostop do uradnega CLI za izboljšanje razvojnega procesa. Zagotavljanje nabora vnaprej konfiguriranih predlog za različne primere uporabe s strani skupine razvijalcev je ena od posebnih funkcij tega CLI. (VueJS, 2022)

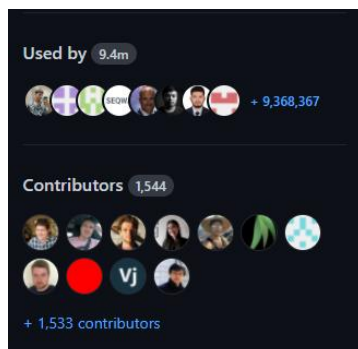
## 6 PRIMERJAVA REACT, ANGULAR, VUEJS

V tem poglavju se bom osredotočil na primerjave ogrodij React, Angular ter VueJS.

### 6.1 Prednosti in slabosti React ogrodja

#### 6.1.1 Prednosti

- **Virtualen DOM** - virtualen DOM (ang. Document Object Model – objektni model dokumenta) je odgovoren za hitro delovanje ogrodja React. DOM je vizualna reprezentacija XML, HTML ali XHTML dokumentov. Predstavlja kopijo pravega dokumenta (spletne strani). Deluje tako, da pri virtualnem dokumentu preveri spremembe in jih posodobi.
- **Komponente ponovne uporabe** - komponente ponovne uporabe (ang. Reusable components) so eden izmed glavnih razlogov, zakaj se ljudje odločajo za ne samo React, ampak na splošno za spletna ogrodja, saj je to glavna stvar, ki programerjem olajša njihovo delo.
- **Skupnost** - ker je React odprtokoden program, ima veliko podporo s strani skupnosti, ki pripomore k izboljšavi in razvoju tehnologije. Po trenutnih podatkih zajema to skupnost 1.544 razvijalcev, uporabljen pa je bil že s strani 9.4 milijona uporabnikov.
- **React Native** - React Native je ogrodje, ki deli svoje podobnosti z ogrodjem React. Omogoča razvoj aplikacij za Android, iOS, Windows in Oculus. Zaradi podobnosti med ogrodjem je prehod bistveno lažji ter omogoča ponovno uporabo komponent med njima (Baranowski, 2022).



Slika 10: Podatki uporabikov in razvijalcev React na GitHub.

Vir: <https://github.com/facebook/react>

### 6.1.2 Slabosti

- **Pomanjkanje dokumentacije** – zaradi hitrega razvoja in posodobitev ogrodja uporabnikom ne pusti dovolj časa, da bi spisali podrobno dokumentacijo o spremembah ali posodobitvah.
- **Hitrost razvoja** – čeprav v mnogih pogledih React pospeši razvoj spletnih strani, lahko določene posodobitve ogrodja upočasnijo razvoj, zaradi potrebe programerjev po učenju novosti.
- **SEO optimizacija** – kot vsa ostala ogrodja, tudi React nima rešitve za optimizacijo spletnih strani pri iskalniku, kar lahko marsikatero podjetje odvrne od uporabe, saj jih iskalniki ne bi prikazali (Baranowski, 2022).

## 6.2 Prednosti in slabosti Angular ogrodja

### 6.2.1 Prednosti

- **MVC arhitektura** – MVC arhitektura postavlja temelje za funkcionalnosti vezave podatkov in obsegov. S to arhitekturo je mogoče izolirati logiko aplikacije od uporabniškega vmesnika. Pri tem primeru krmilnik prejme vse zahteve za aplikacijo in deluje z modelom za pripravo vseh podatkov, ki jih potrebuje za pogled, ki prikaže končni odziv.
- **Moduli** – modul je mehanizem, ki združi direktive, komponente ali storitve, ki so med seboj povezane tako, da skupaj ustvarijo aplikacijo.
- **Direktive po meri** – direktive po meri izboljšajo HTML funkcionalnost in so uporabne pri dinamičnih aplikacijah na strani stranke. Vsi se začnejo z prefiksom *ng*, kar omogoča, da jih lahko HTML prepozna (Sidana, 2021).

### 6.2.2 Slabosti

- **SEO optimizacija** - tako kot React, ima tudi Angular pomanjkljivost pri optimizaciji za iskalnike.

- **Obsežnost in zapletenost** – Angular je zelo obsežno ogrodje in ima svoje specifikke, ki se jih marsikdo težje priuči, kar povzroči večjo krivuljo učenja za nove uporabnike.
- **CLI dokumentacija** – CLI (ang. Command-Line Interface – vmesnik za ukazno vrstico) dokumentacija je zelo osnovna. Čeprav je CLI zelo uporabno orodje za Angular razvijalce, jih je večina mnenja, da trenutno ni podanih dovolj informacij na uradni dokumentaciji za popolno uporabo (Sidana, 2021).

## 6.3 Prednosti in slabosti VueJS ogrodja

### 6.3.1 Prednosti

- **Majhna velikost** – velikost ogrodja VueJS zavzema 18KB, kar omogoči preprosto tranzicijo ali izdelavo projekta s tem ogrodjem.
- **Virtualen DOM** – tako kot React, VueJS deluje na princip virtualnega objektnega modela dokumenta, ki omogoča uporabnikom hitro delovanje in boljšo uporabniško izkušnjo.
- **Možnosti integracije** – ker je VueJS odvisen samo od JavaScripta, omogoča preprosto integracijo z obstoječimi aplikacijami, saj ne potrebuje nobenih dodatnih orodij za delovanje.
- **Preprost za učenje** – VueJS ne potrebuje poglobljenega znanja knjižic, JSX ali TypeScript jezika kot ostala ogrodja. Dovolj so osnove HTML, CSS ter JavaScripta.

### 6.3.2 Slabosti

- ***Jezikovna ovira*** – zaradi popularizacije jezika s strani podjetji kot sta Xiaomi in Alibaba je postal VueJS najpopularnejše ogrodje na Kitajskem, ker so določeni viri za učenje ostalih ogrodji zaradi njihove varnostne politike nedostopni. To je povzročilo, da je veliko dokumentacije, virov ter opisov v kitajskem jeziku.
- ***Pomanjkanje podpore za velike projekte*** – VueJS je najmlajši izmed treh največjih ogrodij, ker sta njegova skupnost in razvojna ekipa manjši v primerjavi z ostalimi ogrodji.
- ***Pomanjkanje virov*** – ker VueJS ne prejema večje finančne podpore večjih podjetji, kot na primer React s strani Facebooka ali Angular s strani Google, je razlika med orodji, ki jih ogrodja uporabljajo v stotih, obstoječi vtičniki pa pogosto niso kompatibilni ali podprti.

## 7 ALI SE BO DELO OD DOMA ZAKLJUČILO?

Delo na daljavo je praksa, pri kateri zaposleni svoje delo opravljajo z lokacije, ki ni centralna pisarna, ki jo upravlja delodajalec. Delo na daljavo je postalo vse bolj priljubljeno, saj prinaša prednosti tako delodajalcem kot zaposlenim. Dolgoročna priljubljenost dela na daljavo je povezana s prednostmi, ki jih lahko zagotovi vključno s skrajšanim ali odpravljenim časom vožnje na delo. Delo na daljavo zahteva mešanico prave kulture, procesov in tehnologije, ki je posebej usmerjena v omogočanje virtualnim ekipam, da uspešno delajo od koder koli (vmWare, 2022).

V desetletju pred pandemijo je le nekaj zaposlenih uživalo ureditev dela na daljavo. Večina delodajalcev je želela, da njihovi zaposleni delajo na kraju samem, saj so verjeli, da bodo tako lahko bolje nadzorovali dejavnosti svojih delavcev (Relocate Magazine, 2022).

V začetnih fazah pandemije je bilo nešteto organizacij po vsem svetu prisiljenih hitro pripraviti možnost dela na daljavo, kar je nujen premik od tradicionalnih pisarniških nastavitev zaradi ukazov o ostajanju doma in zaklepanja. Potreba po teh ukrepih je nekatere privedla do tega, da imajo zaposleni v številnih sektorjih prednost, da imajo možnost bodisi popolnoma oddaljenih ali hibridnih delovnih tednov, v katerih nekaj dni preživijo doma, nekaj pa v pisarni. To je ustvarilo tudi dilemo za podjetja: če predpisujejo zahteve samo za pisarne, lahko izgubijo dragocene člane osebja zaradi konkurentov z oddaljenimi in/ali hibridnimi delovnim časom. Tako se zdi, da so hibridne nastavitve zadovoljiv kompromis za številna podjetja – vsaj za zdaj (Osborne, 2022).

Novo pridobljena prožnost, ki so jo številni delavci izkusili med pandemijo, je pustila neizbrisen pečat. Sposobnost boljšega ravnovesja med delom in življenjem ter opustitev vožnje na delo je bila izjemno pozitiven stranski učinek kaotičnega časa. Zdaj pa se milijoni zaposlenih nočejo vrniti nazaj v pisarne. Povečalo se je tudi povpraševanje po delovnih mestih, ki ponujajo vsaj nekaj elementa dela na daljavo (Tatum, 2022).

Možnost za delo od doma se najbolj obeta ljudem, ki imajo v »večji delež koncentracije kot sodelovanja«, torej delavci, ki lahko enako dobro opravljajo dela v izolaciji brez vrveža ekipe okoli. V obzir pa je potrebno vzeti, da novi delavci potrebujejo veliko več podpore in se organizacijam priporoča, da na začetku novi delavci ne delajo od doma in imajo več podpore »iz oči v oči« (Tatum, 2022)

Niso pa na boljšem le zaposleni, z delom od doma lahko podjetje zmanjša stroške, saj delavcem za delo od doma ne plačuje stroškov prevoza. Prav tako zmanjšuje odsotnost z dela in povečuje zavzetost zaposlenih (Verasai, 2022).

Izkušnje so podjetjem omogočile, da so se ogrela za zamisel o delu na daljavo, vendar se zdi, da pravi zagon za spremembe prihaja od zaposlenih, večinoma mlajših, digitalno bolj podkovanih generacij, ki vstopajo v delovno silo. Trenutni hibridni model bi torej lahko razumeli kot kompromis med starejšimi generacijami, ki so navajene delati s polnim delovnim časom, in mlajšim delavci, ki želijo bolj prilagodljive možnosti (Relocate Magazine, 2022).

Delo od doma ali na daljavo bo ostalo. Raziskave so pokazale, da ima delo na daljavo pozitiven vpliv na splošno izkušnjo zaposlenih, v nedavni raziskavi FlexJobs pa je 95 % anketirancev poročalo o višji stopnji produktivnosti pri delu na daljavo (Kambouris, 2021).

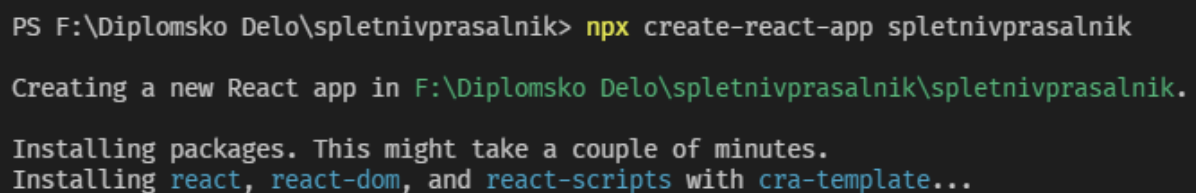
## 8 IZDELAVA PROJEKTNE NALOGE

### 8.1 Opis projektne naloge

Za izdelavo projektne naloge sem si zadal izdelavo COVID-19 vprašalnika, pri katerem uporabnik lahko odgovori ali sestavi svojo anketo z da ali ne odgovori, ker sem želel osvojiti cel "paket" izdelave spletne strani (od zalednega dela razvoja, vse do prikaza podatkov na spletni strani). Potrebne znanja za izdelavo nisem imel, zato mi je največ časa vzelo raziskovanje in učenje, ki pa mi prevelikih težav k sreči ni povzročalo, saj sem neko znanje že pridobil v času študentskega dela in tekom šolanja. Izdelava mi je vzela približno 5 tednov, da sem osvojil potrebna znanja za izdelavo projektne naloge.

### 8.2 Postopek izdelave projektne naloge

Pri namestitvi programov se nisem srečeval s kompleksnejšimi težavami, saj so bila navodila dovolj jasna. Tako sem si naložil vse potrebne programe, ustvaril privzeto mapo za izdelavo ter ustvaril projekt z ukazom `npx create-react-app spletnivprasalnik`.

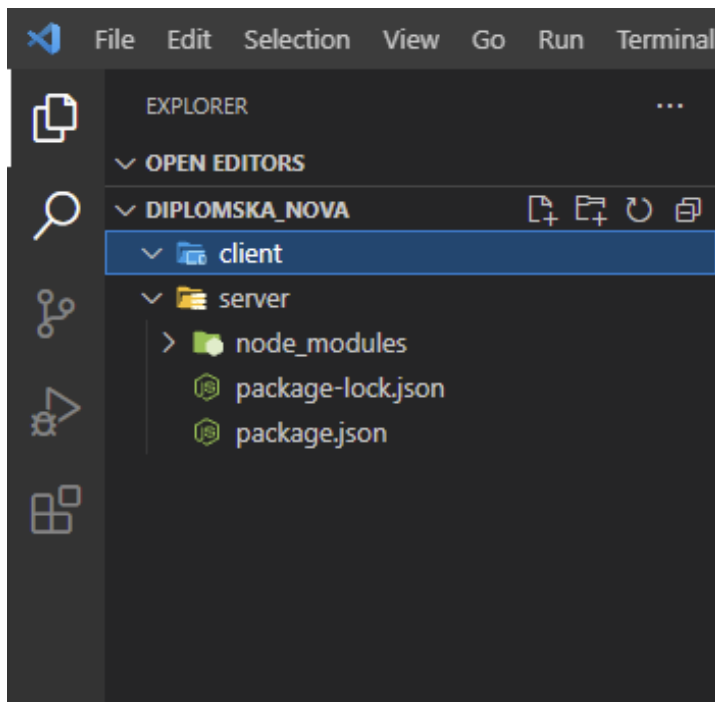


```
PS F:\Diplomsko Delo\spletnivprasalnik> npx create-react-app spletnivprasalnik
Creating a new React app in F:\Diplomsko Delo\spletnivprasalnik\spletnivprasalnik.
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
```

Slika 11: Namestitev potrebnih paketov za izdelavo projektne naloge.

Vir: (Lastni vir)

Ob uspešni ustanovitvi projekta je sledila je kreacija map, s katerimi sem razdelil projekt na zaledni in začetni del. Načeloma predstavlja dobro prakso, da se ta dela ločita po svoji strukturi, saj nam olajša nadaljnje delo in ustvari večjo preglednost.



Slika 12: Priprava map za deljenje backend in front-end naloge.

Vir:(Lastni vir)

Za tem je sledila namestitev potrebnih paketov za delovanje zalednega dela aplikacije. To je predstavljalo vmesne knjižnice, ki sem jih potreboval za postavitev strežnika. Za začetek je to predstavljalo express, cors ter mysql2. Seveda je tekom razvoja projekt postajal kompleksnejši, z raziskovanjem je bila potrebna nadgradnja, tako da ti paketi niso predstavljali končne verzije programa.

```
Is this OK? (yes)
PS F:\Academia\Diplomska_Nova\server> npm install express cors mysql2

added 73 packages, and audited 74 packages in 5s

7 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS F:\Academia\Diplomska_Nova\server> 
```

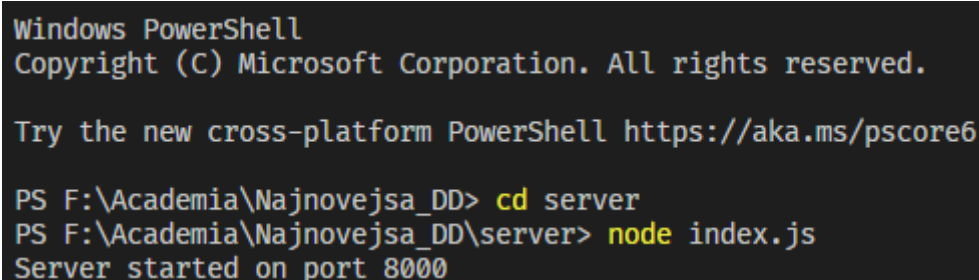
Slika 13: Namestitev paketkov za zaledni del aplikacije.

Vir: (Lastni vir)

Pred razvojem zalednega dela sem na list papirja skiciral osnovni izgled projektnega dokumenta ter pripravil razvojno okolje. Ko sem pridobil okvirno sliko, kako naj bi program deloval, sem to skico prenesel v obliko diagramov ERD, preko brezplačnega spletnega orodja dbdiagram.io.

Nato sem namestil potrebne pakete (knjižnice) za razvoj projektne naloge z ogrodjem ReactJS. Namen teh knjižnic je implementacija

Ko sem imel mape pripravljene in paketke nameščene, je bila prva faza izdelave končana. Tako sem se lotil izdelave preprostega zalednega dela aplikacije, s katero sem testiral, če vse deluje kot zahtevano, da sem lahko predčasno odpravil potrebne napake (ki jih ni bilo malo).



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS F:\Academia\Najnovejsa_DD> cd server
PS F:\Academia\Najnovejsa_DD\server> node index.js
Server started on port 8000
```

Slika 14: Zagon strežnika

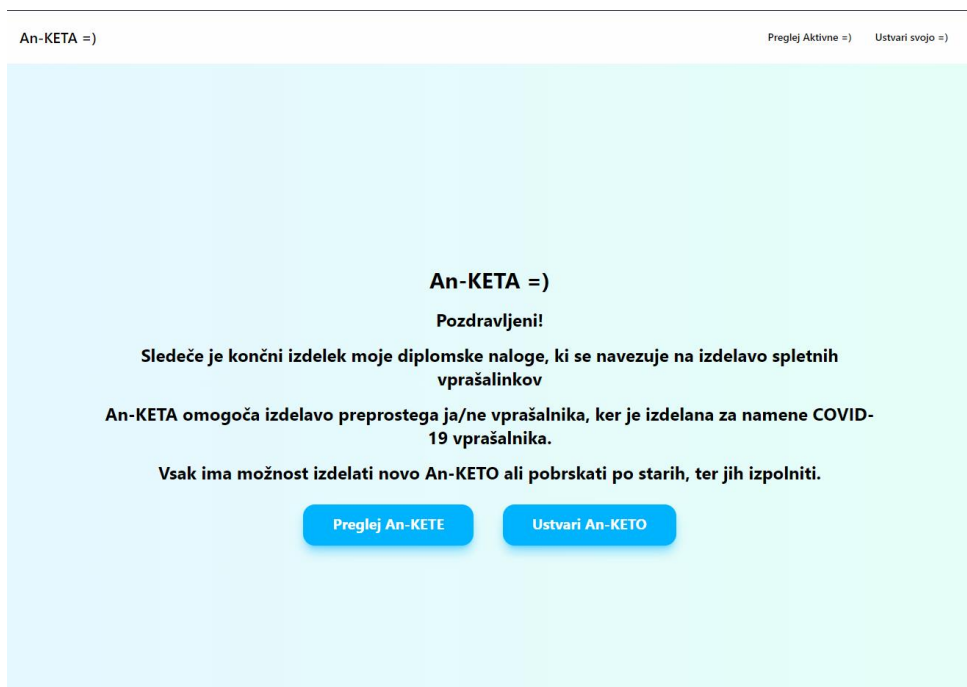
Vir: (Lasten vir)

Pri zalednem delu sem si najprej postavil preprost strežnik s pomočjo navodil na spletu ter tako dalje gradil na tem. Ob znaku sporočila "Server started on port 8000" mi program sporoči, da strežnik uspešno deluje in čaka na zahtevo.

Ko sem vedel, da strežnik deluje, sem se bil najprej primoran naučiti sintakso tega ogrodja, tako da sem prebral dokumentacijo ter pogledal nekaj video posnetkov, da sem pridobil osnovno znanje dela z ReactJS. Ker sem imel izkušnje z HTML, CSS ter JavaScript in sem v tistem obdobju ravno delal kot "Full-stack" razvijalec, mi ni bilo težko osvojiti osnovnega znanja, vendar so mi določeni "logični" procesi delovanja ReactJS povzročali težave.

Pri izdelavi spletne strani sem se najprej lotil izgleda. Izgled strani sem večkrat popravljaj, ampak na koncu pristal na odločitvi preproste SPA, ki vsebuje navigacijski meni zgoraj ter 2 gumba za navigacijo ob vstopu na spletno stran – enega za pregled anket ter drugega za izdelavo ankete in navigacijski meni.

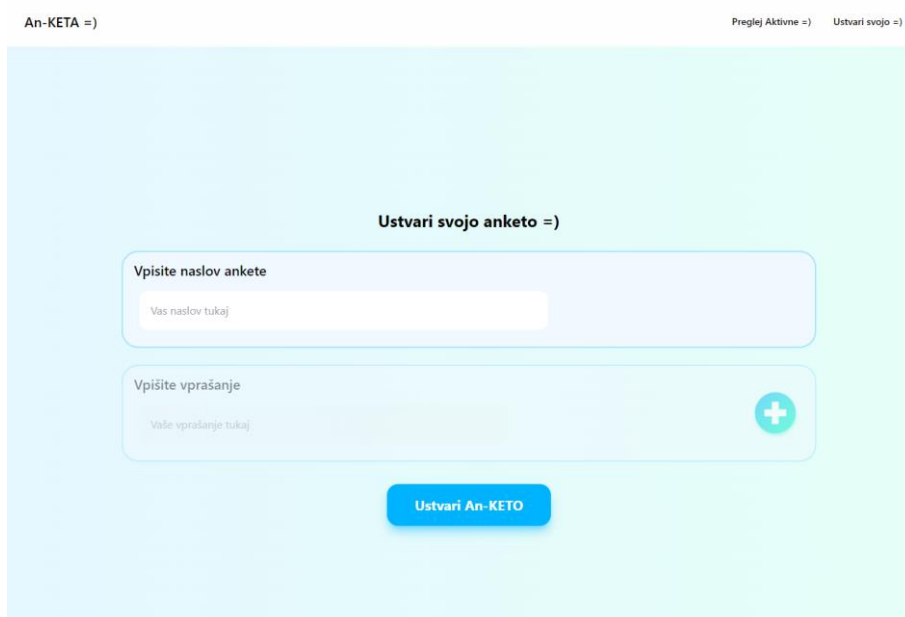
Končni izgled spletne strani je sledeč:



Slika 15: Izgled vstopne spletne strani.

Vir: (Lastni vir)

Tukaj lahko prosto izbiramo, kam želimo spletno stran navigirati. Klik na gumb "**Ustvari An-KETO**" odpre sledeče okno, v katerega vnesemo naslov ankete ter lahko pišemo vprašanja.



Slika 16: Ustvari An-KETO

Vir: (Lastni vir)

Dodatna okna za vprašanja se pojavijo avtomatično za vsako, ko kliknemo na vrhnjega, tako da ni predpostavljene omejitve, koliko vprašanj lahko vsebuje.

Spodaj je primer izdelave ankete za COVID-19 vprašalnik:

The screenshot shows a web interface for creating a survey. At the top, it says "Ustvari svojo anketo =)". Below this, there are five question blocks, each with a title "Vpisite naslov ankete" or "Vpisite vprašanje" and a text input field. The questions are:

- COVID-19 Vprašalnik
- Ali ste bili v zadnjih 10 dneh v stiku z osebo, okuženo s COVID-19?
- Ali se počutite slabo?
- Ste bili v zadnjih treh dneh prehlajeni?
- Ali ste v zadnjih treh dneh imeli vročino?

Slika 17: Primer sestave ankete za COVID-19 vprašalnik 1/2.

Vir: (Lastni vir)

The screenshot shows the continuation of the survey creation interface. It displays five more question blocks, each with a title "Vpisite vprašanje" and a text input field. The questions are:

- Imate bolečine v mišicah?
- Imate prebavne težave (driska ali bruhanje) ?
- Ste imeli morda pozitiven bris na COVID-19?
- Imate povišano telesno temperaturo (nad 37.5°)?
- Vase vprašanje tukaj

At the bottom right of the question blocks, there is a green circular button with a white plus sign. Below the question blocks, there is a green button labeled "Ustvari anketo".

Slika 18: Primer sestave ankete za COVID-19 vprašalnik 2/2.

Vir: (Lastni vir)



Če na začetni strani kliknemo na gumb "**Preglej Ankete**" prikaže drug zavihek, na katerem se nahajajo obstoječe ankete, ki jih imamo možnost izpolniti.



Slika 19: Prikaz strani z obstoječimi anketami.

Vir: (Lastni vir)

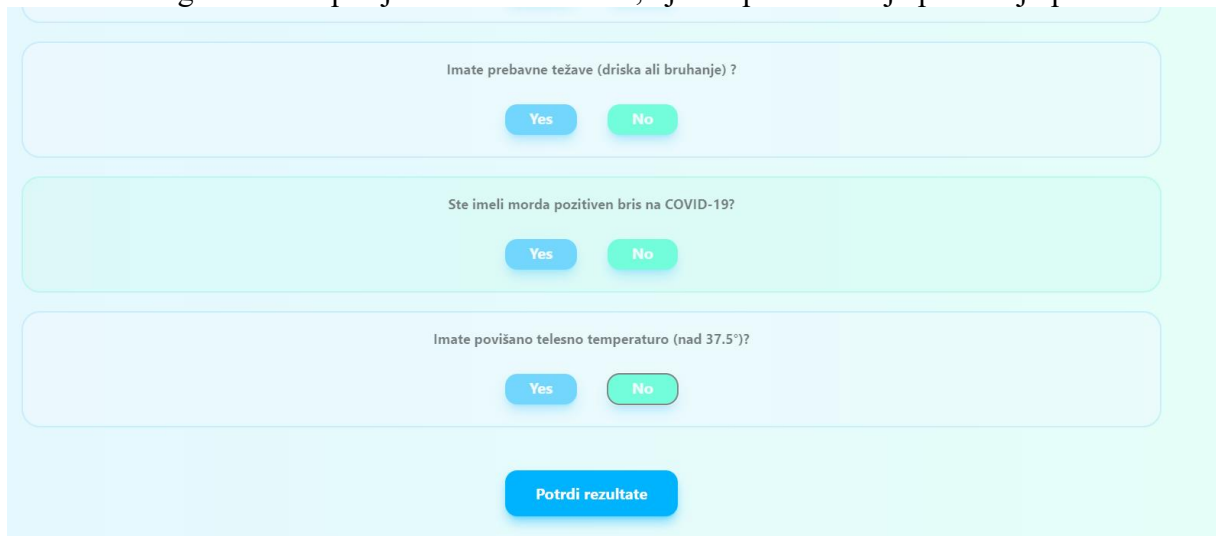
Ko kliknemo na eno izmed možnih anket, se pojavi sledeče okno za reševanje:

The image shows a web interface for a survey titled "Trenutno resujete anketo=) !" with the subtitle "COVID-19 Vprašalnik". The survey consists of five questions, each with a "Yes" and "No" button. The questions are: 1. "Ali ste bili v zadnjih 10 dneh v stiku z osebo, okuženo s COVID-19?" 2. "Ali se počutite slabo?" 3. "Ste bili v zadnjih treh dneh prehlajeni?" 4. "Ali ste v zadnjih treh dneh imeli vročino?" 5. "Se vam zdi da težko dihate?"

Slika 20: Stran, na kateri se prikaže anketa za reševanje.

Vir: (Lastni vir)

Ko so vsi odgovori izpolnjeni, se pojavi gumb "**Potrdi rezultate**", preko katerega oddamo anketo. Ti odgovori se zapišejo v JSON datoteko, kjer se potem berejo pri branju podatkov.

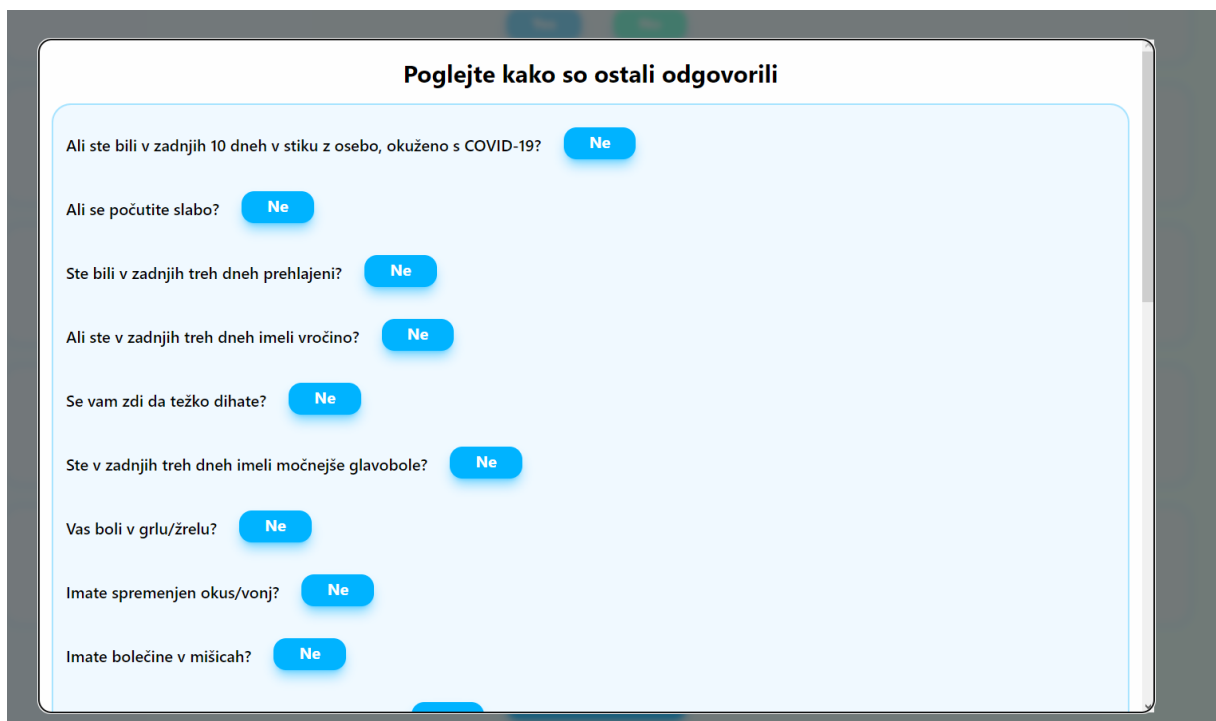


The screenshot shows a survey form with three questions, each with 'Yes' and 'No' buttons. The questions are: 'Imate prebavne težave (driska ali bruhanje) ?', 'Ste imeli morda pozitiven bris na COVID-19?', and 'Imate povišano telesno temperaturo (nad 37.5°)?'. At the bottom is a blue button labeled 'Potrdi rezultate'.

Slika 21: Gumb za oddajo ankete.

Vir: (Lastni vir)

Ko sprožimo to akcijo (potrdi rezultate), se odpre pojavno okno, na katerem so zapisana vprašanja ter shranjeni odgovori preteklih anketirancev. To nam omogoča pregled preteklih anket in statistiko udeležbe.



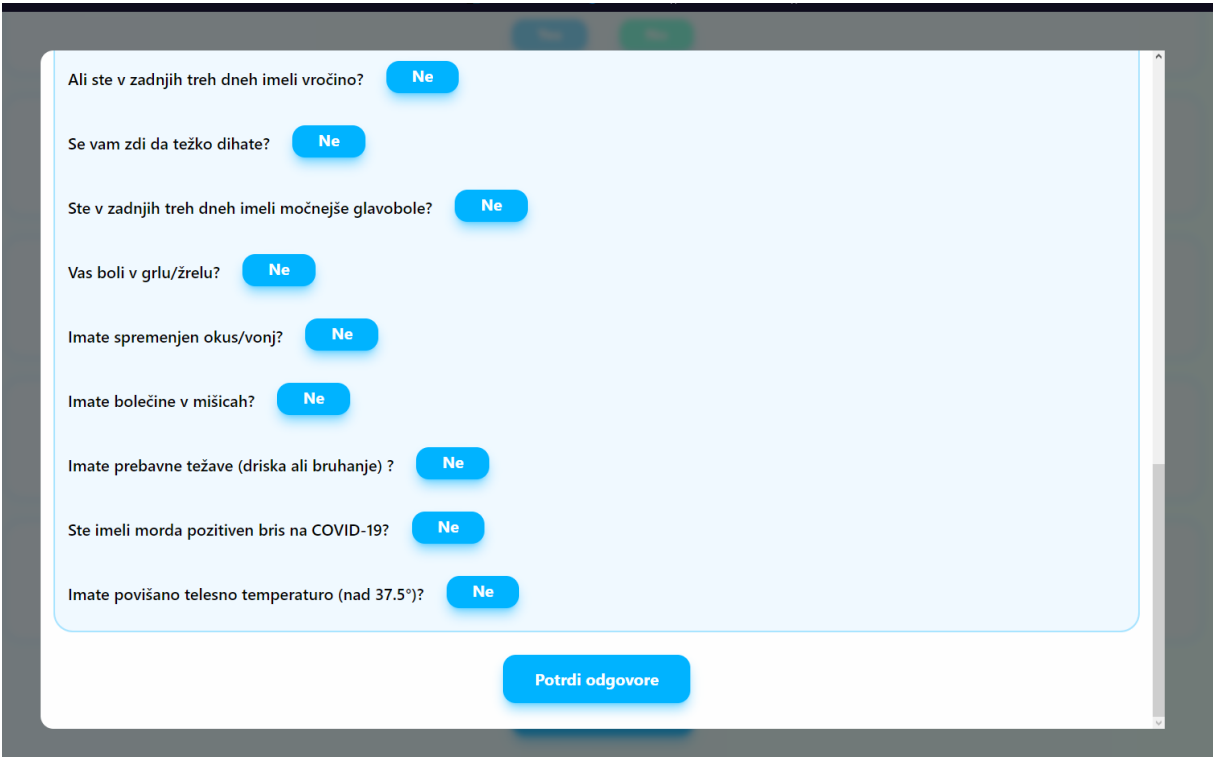
The screenshot shows a window titled 'Poglejte kako so ostali odgovorili'. It lists nine questions with 'Ne' (No) buttons next to them, indicating previous responses. The questions are: 'Ali ste bili v zadnjih 10 dneh v stiku z osebo, okuženo s COVID-19?', 'Ali se počutite slabo?', 'Ste bili v zadnjih treh dneh prehlajeni?', 'Ali ste v zadnjih treh dneh imeli vročino?', 'Se vam zdi da težko dihate?', 'Ste v zadnjih treh dneh imeli močnejše glavobole?', 'Vas boli v grlu/žrelu?', 'Imate spremenjen okus/vonj?', and 'Imate bolečine v mišicah?'.

Slika 22: Pojavno okno po oddaji ankete 1/2.

Vir: (Lastni vir)



Na dnu pojavnega okna se nahaja gumb "**Potrdi odgovore**", ki shrani zapise v strežnik.



The image shows a confirmation window for a survey. It contains nine questions, each with a blue button labeled "Ne" (No) selected. At the bottom right, there is a blue button labeled "Potrdi odgovore" (Confirm answers).

| Question                                          | Answer |
|---------------------------------------------------|--------|
| Ali ste v zadnjih treh dneh imeli vročino?        | Ne     |
| Se vam zdi da težko dihate?                       | Ne     |
| Ste v zadnjih treh dneh imeli močnejše glavobole? | Ne     |
| Vas boli v grlu/žrelu?                            | Ne     |
| Imate spremenjen okus/vonj?                       | Ne     |
| Imate bolečine v mišicah?                         | Ne     |
| Imate prebavne težave (driska ali bruhanje) ?     | Ne     |
| Ste imeli morda pozitiven bris na COVID-19?       | Ne     |
| Imate povišano telesno temperaturo (nad 37.5°)?   | Ne     |

Potrdi odgovore

Slika 23: Pojavno okno po oddaji ankete 2/2.

Vir: (Lastni vir)

## 9 SKLEP

S tem poglavjem zaključujem s pisnim delom diplomskega dela. V naslednjem poglavju bom pa na kratko predstavil izdelano nalogo.

Pri raziskovalnem delu sem se naučil mnogo novih stvari. Od podrobnega vsebinskega delovanja ReactJS, VueJS ter AngularJS, do delovanja SPA in MPA aplikacij, njihovih omejitev in prednosti, ter na podlagi intervjuja s strani strokovnjaka dobil vpogled glede omenjenih tehnologij na trgu.

Čeprav imam izkušnje z izdelavo spletnih strani, mi je tehnični del vseeno predstavljal malo težav, saj mi je to predstavljalo mešanico programiranja in oblikovanja spletnih strani ter programiranja z objektno orientiranim programiranjem, kar mi je bilo malo kontradiktorno.

***Hipoteza 1: Ogradje Angular je najpopularnejše ogrodje leta 2021.***

Hipoteza je ovržena. Po že narejenih raziskavah na raznih forumih, kot je StackOverflow, so ankete in analitični podatki prenosov paketov pokazali, da je ReactJS ogrodje, ki je najbolj priljubljeno med razvijalci (GitHub, 2022).

***Hipoteza 2: Ogradja kot so Angular, Vue in React bodo nadomestila splošne spletne jezike.***

Hipoteza je na ta dan zavrnjena. Ob pogovoru z g. Robertom Španom sem zastavil to vprašanje, na katerega je delno odgovoril z da in ne. Meni, da lahko v prihodnosti, ko bodo te tehnologije izpopolnjene, vidi, da bi postale nova norma pri izdelavi spletnih strani. Vendar dokler bo to predstavljalo prevelik strošek ali omejitve delovanja trenutnih obstoječih spletnih strani, ki temeljijo na kakšni starejši tehnologiji, ki ni kompatibilna z ogrodji, se osebe za to predelavo ne bodo odločile. Podobno vprašanje sem postavil programerjem, zaposlenim pri Novi KBM, ki so mi bolj ali manj podali soglasen odgovor, da je izbira tehnologije odvisna od posameznikovih potreb. Opažajo pa spremembo trendov, da vse več novih spletnih strani stremi k novejšim tehnologijam. Prav tako so mnjenja, da vse spletne tehnologije izhajajo iz treh osnovnih jezikov, ki bi jih vsak spletni razvijalec moral poznati: HTML, CSS ter JavaScript.

***Hipoteza 3: Sintaksa JavaScript ogrodij je prenosljiva.***

Hipoteza je potrjena. Vsa primerjana ogrodka dopuščajo pisanje čiste JavaScript kode. Edino v primerih, kjer bi si poimenovanja privzetih funkcij nasprotovala, so narejene manjše dodelave pri poimenovanju.

***Hipoteza 4: Uporaba orodij za oddaljeno komunikacijo lahko nadomesti delo v službi.***

Hipoteza je potrjena. V času epidemije COVID-19 so podjetja bila na nek način prisiljena spremeniti način svojega poslovanja ter komuniciranja s partnerji in strankami. COVID-19 je samo pospešil nekaj, kar mnogi menijo, da bi se zgodilo v prihodnosti. Vsa komunikacija dela, sestanki je potekalo preko e-pošte ali aplikacij instantnih sporočil (npr. Skype, MS Teams), za pomoč uporabnikom so se uporabili programi za povezovanje na oddaljene seje (npr. IBM BigFix, Microsoft Hita Pomoč, ISL, TeamViewer). Zaradi potrebe nadomestitev komunikacije za učinkovito vodenje/upravljanje projektov so podjetja začela uvajati raznorazne programe za sledenje napakam in agilno upravljanje projektov (Jira, Asana, Trello, ...). S tem so skupine ter timi pri delu od doma lažje opravljali svoje delo, saj so si naloge lahko prerazporejali brez dodatne komunikacije ali nepotrebnih sestankov. Je pa potrebno upoštevati, da vseeno človeški stik pusti več vtisov in ustvari več mnenj ter pripomore dobaviteljem ali partnerjem lažje izbrati odločitev (Ozimek, 2020).

## 10 VIRI IN LITERATURA

Baranowski, W. (5. junij 2022). *React JS: Advantages and Disadvantages*. Pridobljeno iz MassivePixel: <https://massivepixel.io/blog/react-advantages-disadvantages/>

Codecraft. (30. Avgust 2022). *Codecraft - ES6 JavaScript & TypeScript*. Pridobljeno iz Codecraft: <https://codecraft.tv/courses/angular/es6-typescript/for-of/>

Dizowki, M. (27. junij 2022). *What's the difference between AngularJS and Angular?* Pridobljeno iz Gorrior: <https://gorrior.io/blog/angularjs-vs-angular/>

E. Moore, G. (19. april 1965). *Cramming more components*. Pridobljeno iz NewsRoom: <https://newsroom.intel.com/wp-content/uploads/sites/11/2018/05/moores-law-electronics.pdf>

Flanagan, D. (10. maj 2011). *JavaScript: The Definitive Guide: Activate Your Web Pages*. O'Reilly Media, Inc.: Newton. Pridobljeno iz Wikipedia: [https://en.wikipedia.org/wiki/Single-page\\_application](https://en.wikipedia.org/wiki/Single-page_application)

GitHub. (5. julij 2022). *Front-end frameworks popularity (React, Vue, Angular and Svelte)*. Pridobljeno iz GitHub: <https://gist.github.com/tkrotoff/b1caa4c3a185629299ec234d2314e190>

Google. (31. Avgust 2022). *Angular - Binding Overview*. Pridobljeno iz Angular: <https://angular.io/guide/binding-overview>

Google. (1. September 2022). *Angular - CLI Overview*. Pridobljeno iz Angular: <https://angular.io/cli>

Google. (1. September 2022). *Angular - Routing*. Pridobljeno iz Angular: <https://angular.io/guide/routing-overview>

Grybniak, S. (6. januar 2021). *An Overviews of Vue.js Front-end Framework*. Pridobljeno iz C# Corner: <https://www.c-sharpcorner.com/article/an-overview-of-vue-js-frontend-framework/>

Guo, D. (13. avgust 2019). *The History and Legacy of jQuery*. Pridobljeno iz LogRocket: <https://blog.logrocket.com/the-history-and-legacy-of-jquery/>

Javatpoint. (30. Avgust 2022). *Javatpoint - React Props*. Pridobljeno iz Javatpoint: <https://www.javatpoint.com/react-props>

- Kambouris, A. (11. november 2021). *Remote Work Is Here to Stay. It's Time to Update the Way You Lead*. Pridobljeno iz Entrepreneur:  
<https://www.entrepreneur.com/article/395311>
- Kaplan, K. (3. februar 2020). *Why Every Business Needs A Website*. Pridobljeno iz Forbes:  
<https://www.forbes.com/sites/theyec/2020/02/03/why-every-business-needs-a-website/?sh=2b4a686c6e75>
- MDN contributors. (6. junij 2022). *JavaScript*. Pridobljeno iz MDN web docs:  
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- MDN contributors. (30. marec 2022). *Learn to style HTML using CSS*. Pridobljeno iz MDN web docs: <https://developer.mozilla.org/en-US/docs/Learn/CSS>
- Meta Platforms, Inc. (30. Avgust 2022). *ReactJS - State and Lifecycle*. Pridobljeno iz ReactJS: <https://reactjs.org/docs/state-and-lifecycle.html>
- Osborne, C. (3. marec 2022). *The end of fully remote work? Google begins shift to the hybrid office*. Pridobljeno iz ZDnet: <https://www.zdnet.com/article/the-end-of-fully-remote-work-google-begins-shift-to-the-hybrid-office/>
- Ozimek, A. (28. april 2020). *The Future of Remote Work*. Pridobljeno iz SSRN:  
<https://deliverypdf.ssrn.com/delivery.php?ID=973027002121126085023077102099114102121064057008063030022095022122071065004095104006055120040032010124035120108115084083064122058020029085067020026070083116008068103073043095002065010102019001122098095064124089>
- Relocate Magazine. (5. april 2022). *Remote work is here to stay — where to next?* Pridobljeno iz Relocate Magazine: <https://www.relocatemagazine.com/articles/remote-work-is-here-to-stay-where-to-next-kpmg-0422>
- Rodriguez Martinez, E. (7. maj 2017). *React Native Blueprints*. Birmingham: Packt Publishing Ltd. Pridobljeno iz Wikipedia:  
[https://en.wikipedia.org/wiki/React\\_\(JavaScript\\_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library))
- Sidana, U. (22. november 2021). *What are the Advantages and Disadvantages of Angular?* Pridobljeno iz Edureka: <https://www.edureka.co/blog/advantages-and-disadvantages-of-angular/>

- Tatum, M. (25. maj 2022). *Is the 'remote work window' about to close?* Pridobljeno iz BBC:  
<https://www.bbc.com/worklife/article/20220519-is-the-remote-work-window-about-to-close>
- Verasai, A. (15. januar 2022). *REMOTE WORK IS HERE TO STAY: ARE YOU READY?*  
Pridobljeno iz The HR Digest: <https://www.thehrdigest.com/remote-work-is-here-to-stay-are-you-ready/>
- vmWare. (5. julij 2022). *What is remote work?* Pridobljeno iz vmWare:  
<https://www.vmware.com/topics/glossary/content/remote-work.html>
- VueJS. (1. September 2022). *VueJS - Data Binding Syntax*. Pridobljeno iz VueJS:  
<https://v1.vuejs.org/guide/syntax.html>
- VueJS. (1. September 2022). *VueJS Documentation*. Pridobljeno iz VueJS:  
<https://vuejs.org/guide/introduction.html>
- W3Schools. (30. Avgust 2022). *W3Schools - React Class Components*. Pridobljeno iz W3Schools: [https://www.w3schools.com/react/react\\_class.asp](https://www.w3schools.com/react/react_class.asp)

## 11 PRILOGE

Pri iskanju intervjuvanca preko raznih forumov in socialnih omrežij, sem dobil odgovor od g. Roberta Špana, ki je zaposlen kot front end programmer v podjetju GenEplanet.

Njegova primarna naloga je programiranje spletne strani, prav tako pa skrbi za delovanje strežnikov in sodeluje pri načrtovanju podatkovnih baz.

Podjetje je bilo ustanovljeno leta 2008 s strani Marka Bitenca, ki je imel “preprosto” idejo o izboljšavi posameznikovega življenja. Primarna dejavnost podjetja je genska analiza človeka, katera posamezniku lahko pripomore pri izboljšanju – in podaljšanju njegovega življenja. Leta 2019 je bilo podjetje s strani Financial Times uvrščeno med top 1000 najhitreje rastočih podjetij v skupini.

Intervju je bil opravljen 8.10.2021. Zaradi ukrepov epidemije COVID-19 je bil le ta opravljen na daljavo preko Discord-a.

### **1. Koliko časa ste že zaposleni kot programer pri podjetju »Geneplanet«?**

*Pri podjetju Geneplanet sem zaposlen približno dve leti.*

### **2. Kaj vas je v prepričalo za programiranje spletnih strani in ne aplikacij ali zalednega programiranja?**

*Predvsem to, da sem samouk. Lažje se mi je bilo naučiti spletnega (frontend) programiranja z uporabo brezplačnih spletnih gradiv, kot pa zalednega programiranja. Predvsem zaradi vidika razvoja in pomanjkanja podatkov, na katerih bi se lahko učil. Nisem imel ravno volje dan za dnem kreirati baz in tabel, da bi jih lahko manipuliral ter se učil.*

### **3. S katerimi tehnologijami se srečujete pri svojem delu?**

*Pri svojem delu se primarno srečujem z JavaScript, Angular, npm, ExpressJS, SCSS, html5. Če pa kdaj priskočim na pomoč k kateremu drugemu projektu pa tudi z React-om, dotNET, TypeScript in TailwindCSS.*

### **4. Ali podjetje že od začetka uporablja te tehnologije ali so bile spremembe vpeljane tekom kakšnega projekta (na primer nadgradnje spletne strani)?**

*Od začetka.*

### **5. Kaj je bila ključna odločitev za izbor te tehnologije?**

*V preteklosti sem se spoznal z React in Vue tako da sem imel predpostavko za delo. Vendar pri tem podjetju so za front end razvoj uporabljali Angular, tako da nisem sodeloval pri odločitvi.*

### **6. Zdaj ko gledate za nazaj, ali menite da bi lahko z katero drugo tehnologijo boljše oz. lažje izpeljali ta projekt?**

*Mogoče bi bilo hitreje razvijati z Vue + Typescript.*

### **7. Zakaj da/ne?**

*Ker ima Angular precej več "boilerplate" (ponavljajoče se) kode, ki se precej ponavlja in vzame čas, ter strmejšo krivuljo učenja.*

### **8. Sam sem opazil da veliko podjetij uporablja starejše tehnologije za svoje spletne strani. Kaj menite da je vzrok za to?**

*»Legacy koda« (stara koda), ki se je ponavadi podjetje ne odloči posodobiti, ker za to ne vidi kratkoročne koristi, ali ker je spisana na preveč "coupled" (odvisen) način. To pomeni da sprememba enega dela kode bi zahtevala spremembo v drugem delu, ki se zanaša na prvega, etc., s tem se poveča tveganje za vpeljavo novih bugov (, težko napoveš, koliko časa bo trajalo prepisovanje kode etc.).*

### **9. Menite da boste pri prihodnjih projektih uporabili nove tehnologije?**

*Po vsej verjetnosti bom zaenkrat ostal pri Angularju, saj sem pri svojem delu postal precej večšč. Seveda pa sem pa pripravljen tudi za zamenjavo, če se ponudi dobra priložnost.*

### **10. Zakaj da/ne?**

*Ker bi rad podrobneje spoznal Angular, saj sem že dve leti namenil delu in učenju s to tehnologijo, ter se izpopolnil v njegovi uporabi. Ker mi je delo z spletnimi ogradjami dokaj »novo«, bi se rad čim več naučil kako ta tehnologija deluje, ter kakšne izboljšave lahko doprinese v prihodnosti.*

### **11. Ali menite da so/postajajo tehnologije SPA nova norma?**

*To je težko vprašanje za odgovoriti. Definitivno se mi zdi, da se svet giblje v to smer, in postajajo tehnologije iz dneva v dan čedalje popularnejše, vendar ne bi rekel da so norma (še), vidim pa njihov potencial v prihodnosti.*

## 12. Zakaj da/ne?

*Predvsem zaradi vidika učenja. Zelo veliko ljudi, ki se želi učiti programirati, jim osebe predlagajo HTML, CSS, JavaScript. Prej bi si upal trditi da so ti trije jeziki »norma«, saj za dosti ljudi to predstavlja prve programske jezike, in brez dvoma, so HTML, CSS, JS lažje za razumeti in osvojiti, kot pa če bi se uporabnik srečal z Angular, React ali Vue. HTML dopuša programiranje kjerkoli, tudi v beležki, dokler je notacija pravilna in vrsta dokumenta označena (kratica .html). Za ogrodja pa moraš najprej naložiti orodja, IDE, pakete, pravilno kreirati projekt – torej tudi potrebno osnovno znanje CMD-ja (ukazne vrstice), preden lahko pričneš z delom, kar pa marsikoga lahko odvrne.*