



APLIKACIJA ZA POMOČ PROFESORJEM PRI NADZORU DOVOLJENJ DIJAKOV

raziskovalna naloga

Avtorja:
Gašper Gajšek, R4a
Vid Kreča, R4a

Mentor:
Matic Holobar, prof.

Mestna občina Celje, Mladi za Celje

Celje, februar 2019

Zahvala

Zahvaljujema se g. Maticu Holobarju za vso podporo in svetovanje na vseh področjih. Hvala za skrbnost, pomoč in vso podporo.

Prav tako se zahvaljujema vsem profesorjem in ravnateljem šol, ki so odgovorno rešili anketo in so s tem bili pripravljeni sodelovati pri najini raziskovalni nalogi.

Kazalo vsebine

Povzetek	5
Abstract	5
1 Uvod.....	7
2 Predstavitev delovnih okolij in orodij	8
2.1 Github / Github desktop.....	8
2.2 Atom.....	8
2.3 XAMPP.....	8
3 Uporabljene tehnologije	9
3.1 PHP.....	10
3.2 MySQLi ogrodje.....	10
3.3 CSS Grid	10
3.4 MySQL	11
3.5 JavaScript.....	11
3.6 AJAX.....	12
3.7 jQuery.....	12
4 Načrtovanje spletne aplikacije	13
4.1 Podatkovna baza.....	13
4.2 Osnovna HTML struktura	14
4.2.1 Forme brez osveževanja celotne strani	16
5 Težave in rešitve	18
5.1 Posodabljanje strani ob potrjevanju forme.....	18
5.2 Prenos podatkov v URL	18
5.3 Nadzor uporabnikovega statusa	18
6 Analiza ankete	19
7 Zaključek	24
Viri	25

Kazalo slik

Slika 1: Logotip PHP in MySQL.....	10
Slika 2: Primer CSS Grid HTML razporeditve	11
Slika 3: JavaScript logotip	11
Slika 4: jQuery logotip	12
Slika 5: Podatkovna baza med razvijanjem.....	13
Slika 6: Vsebina header.php datoteke.....	14
Slika 7: Del kode iz main.php datoteke	15
Slika 8: Koda za formo za vnos dijaka.....	16
Slika 9: Del JavaScript kode.....	16
Slika 10: PHP koda za vnos v podatkovno bazo	17

Povzetek

V raziskovalni nalogi sva želela raziskati, kako se profesorji in ostalo osebje srednjih šol sooča z novim zakonikom o varovanju podatkov GDPR. Ker so ta zakon sprejeli v začetku šolskega leta 2018/2019, še niso bile znane vse posledice njegove uveljavitve. Raziskovanja sva se lotila tako, da sva si najprej zastavila nekaj tez, nato pa sva naredila anketo, ki smo jo skupaj s pomočjo mentorja razposlali po drugih slovenskih srednjih šolah. Reševali so jo profesorji in prav tako ravnatelji. Med čakanjem na odgovore sva se odpravila reševati problem na lasten način. Ustvarila sva spletno aplikacijo, ki je v pomoč osebju na srednjih šolah. Med ustvarjanjem aplikacije sva naletela na mnoga vprašanja in probleme, ki sva jih na koncu uspešno rešila.

Abstract

In the research paper, we wanted to find out how professors and other high school personnel work with the new GDPR rules. Because these regulations came into effect at the start of 2018/2019 school year, not all of the rules were known yet. We started work with deciding on the thesis and then sending out a survey to professors and headmasters across Slovenia. After that, we started developing our solution, which is a web application that helps professors and personnel working at high schools with the complication. While working on the application we discovered many questions and problems that we solved in the end.

Ključne besede

Spletna aplikacija, soglasja, dijaki

Kratice in okrajšave

HTML – Računalniški jezik za izdelavo spletnih strani

CSS – preprost slogovni jezik za oblikovanje spletnih strani

PHP – odprtokodni programski jezik, uporabljen za razvoj dinamičnih spletnih strani

MySQL– sistem za upravljanje s podatkovnimi bazami

Key words

Web application, consensus, student

Acronyms and abbreviations

HTML – a system for developing web pages

CSS – a language for styling web pages

PHP – open source programming language for developing server-side applications

MySQL – a database system

1 Uvod

V današnjih dneh se posveča vedno več pozornosti na varnost osebnih podatkov. Zato so članice Evropske unije sprejele zakon, ki ga poznamo pod imenom GDPR (ang. General Data Protection Regulation). Ta zakon smo občutili tako dijaki kot profesorji. To se je pokazalo na ta način, da smo morali dijaki 4. letnika (osebe, ki so ali bodo v tekočem letu dopolnile 18 let) podpisati soglasje, da dopuščamo profesorjem deljenje naših osebnih podatkov, kot je uspeh v šoli, z našimi starši.

Soglasja so imela dve možnosti, in sicer dovoljenje za komunikacijo s starši in vpogled ocen dijaka.

Pred začetkom raziskovanja sva si zastavila naslednje hipoteze:

- Profesorji ne vedo za vsakega dijaka posebej, kakšno soglasje je oddal, zato bi lahko prišlo do nenamernega kršenja pravilnika o GDPR.
- Osebje na šoli je bilo dosti obveščeno o novostih novega zakona, a ni vedelo, kako se lotiti reševanja te težave.
- Problem bo nastal, ker se ne sme osvežiti cela spletna stran ob pošiljanju forme, ampak le del.

Ob začetku izdelave aplikacije sva se najprej odločala, na kakšni platformi bova delala. Razmišljala sva o dveh različnih aplikacijah (ena za mobilne naprave, druga pa za računalnik), a sva se na koncu odločila za spletno aplikacijo, saj je potrebna le ena aplikacija, ki pa je prilagojena prav tako za mobilne naprave kot računalnike.

2 Predstavitev delovnih okolij in orodij

2.1 Github / Github desktop

Github je platforma za razvijanje programske opreme, ki je zasnovana na orodju Git. Je oblačna storitev, ki povezuje milijone razvijalcev po celem svetu in je standard med sistemi za nadzor različic. To pomeni, da imamo pregled nad spremembami v kodi, prav tako pa omogoča sodelovanje več razvijalcev. Razlog, da sva uporabila Github, je ta, da sva lahko oba obenem urejala kodo in imela vpogled nad napredkom, hkrati pa sva imela obnovitev na prejšnjo različico v primeru, da se je pri kodiranju kaj zalomilo.

Git vsebuje mape in podmape, uporabljen pa je lahko za mnogo različnih dokumentov (JavaScript, HTML, CSS, Dokumenti, slike ...). Razlog, da sva izbrala Github, je ta, da je storitev zastonj za uporabo in ker je po najinem mnenju eno izmed boljših orodij.

Github desktop je grafični vmesnik za uporabo Githuba. Aplikacija je nameščena na napravo z operacijskim sistemom Windows ali pa Mac OS. Razlog za uporabo Github desktop je ta, da je bolj priročno in pa bolj prijetno za uporabo.

2.2 Atom

Atom je odprtokodni program za urejanje besedil. To orodje sva uporabila, ker je po najinem mnenju eden izmed boljših urejevalnikov. Atom deluje v povezavi z Github desktop, kar je vse skupaj olajšalo delo.

2.3 XAMPP

Xampp je odprtokodna platforma za vzpostavitev spletnega strežnika z uporabo Apache spletnega strežnika, PHP, in MySQL podatkovne baze. Je zelo preprost za uporabo, prav tako je zastonj, zato sva ga tudi izbrala. XAMPP sva uporabila za testiranje najine aplikacije lokalno, kasneje pa sva vse prestavila na internetni strežnik.

3 Uporabljene tehnologije

Ko sva se odločala, katere tehnologije bova uporabljala pri projektu, sva imela več možnosti. Glavna stvar, ki sva jo morala izbrati, je programski jezik ali tehnologija za razvoj dinamičnih spletnih vsebin. Dobra in preprosta podpora interakcij s podatkovno bazo je prav tako zelo pomembna, saj na tem temelji celoten projekt.

Na voljo sva imela tri glavne izbire. Prva je bil PHP, druga Node.js, tretja pa ASP.NET. Ker sva PHP izmed teh treh možnosti najbolj poznala in ker se nama je zdel najbolj primerna izbira za to aplikacijo, sva se odločila, da ga uporabiva. ASP.NET in Node.js sta zelo uporabna, v veliko primerih tudi hitrejša in bolj primerna, ampak PHP je po najinem mnenju bolj preprost in v tem primeru hitrost delovanja spletne aplikacije ni bila tako pomembna, da bi jo upoštevala pri izbiri programskega jezika.

Odločiti sva se morala tudi, kako bova stilsko oblikovala HTML dokument ali spletno stran. Logična izbira je CSS, ampak za lažje in bolj prilagodljive možnosti oblikovanja sva hotela uporabiti tudi ogrodje za CSS, ki nama bo pri tem pomagalo. Na voljo sva spet imela tri izbire. Prva je bila Grids, druga Bootstrap, tretja pa Flexbox. Izmed teh treh je Grids najbolj primeren, saj je trodimenzionalen, kar je pomembno za pravilno razporejanje elementov spletne aplikacije na različnih velikostih brskalnikov. Flexbox je dvodimenzionalno orientiran in zato ponavadi ni najboljša izbira za projekte, kjer je potrebna podpora tudi na mobilnih napravah. Bootstrap pa je za preprosto spletno aplikacijo malo preveč kompleksen. Prav tako ga lahko Grids skoraj popolnoma nadomesti, hkrati pa je bolj preprost za uporabo in tudi hitrejši.

Pri izbiri tehnologije ali jezika za podatkovno bazo ni bilo veliko odločanja, saj je MySQL najbolj primeren za delo s PHP-jem. Druga možnost je bila SQL in Microsoftov SQL Server, ampak ker je delo s SQL v PHP-ju oteženo zaradi potrebnih dodatnih vtičnikov, sva se odločila da bo MySQL boljša izbira. PHP za delo z MySQL podatkovno bazo potrebuje vtičnik, imenovan MySQLi, ki pa je v večini strežniških paketov že na začetku vključen, zato dodatno delo ni potrebno.

Za olajšano delo in dodatne funkcije sva potrebovala še nekaj dodatnih tehnologij. Potrebovala sva JavaScript za urejanje HTML elementov s kodo po zagonu aplikacije. Odločila sva se tudi za uporabo AJAXa, ker sva potrebovala način izmenjevanja podatkov s MySQL podatkovno bazo brez osveževanja spletne strani. Prav tako sva se odločila za uporabo jQuery, ki je knjižnica za JavaScript, ki omogoča lažje urejanje in izbiranje HTML elementov.

Zelo pomemben del pri razvoju aplikacije je bil nadzor različnih verzij in sprememb kode pri njih. Za ta namen sva uporabila GitHub.com. To je spletna stran, ki omogoča gostovanje skladišč sistema Git. Z uporabo le-te lahko vsako novo verzijo naloživa na internet in lahko posledično starejše verzije in spremembe pregledava kasneje, če je to

potrebno. Za nalaganje na GitHub in spremljanje sprememb v kodi sva uporabila grafični vmesnik, imenovan Github Desktop.

3.1 PHP

PHP je odprtokodni programski jezik, ki se izvaja na strežniku in omogoča razvoj dinamičnih spletnih strani. Je zelo podoben običajno strukturiranim programskim jezikom, kot so C in Perl. Ker se izvaja na strežniku in ne na strani odjemalcev, uporabniki ne morejo dostopati kode tako kot pri HTML in JavaScript. PHP skripte so shranjene v datotekah s končnico .php.

PHP je napisal dansko-kanadski programer Rasmus Lerdorf leta 1994, da bi izboljšal nekaj skript, napisanih v Perl programskem jeziku, ki jih je takrat uporabljal na svoji spletni strani.

Podobne tehnologije PHPju so Microsoftov ASP.NET in odprtokodni Node.js.



Slika 1: Logotip PHP in MySQL

3.2 MySQLi ogrodje

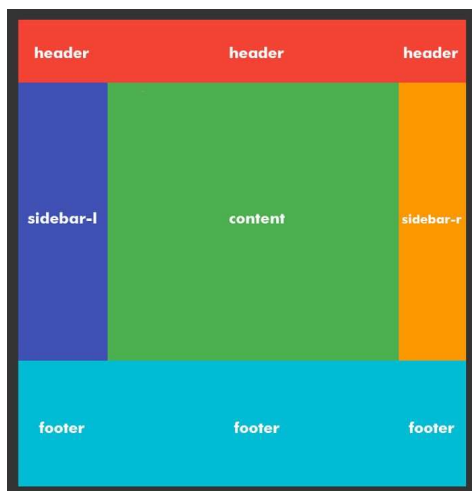
MySQLi je ogrodje, ki omogoča PHP jeziku komunikacijo s MySQL strežnikom. Je nadgradnja MySQL ogrodja za PHP.

Ima objektno orientiran vmesnik, podporo za pripravljene SQL stavke in še veliko drugih naprednih funkcij, ki programerju pomagajo pri pisanju »back-end« kode.

3.3 CSS Grid

Grid je tehnika v CSS-u, ki omogoča uporabnikom, da ustvarijo bolj kompleksne in odzivne razporeditve HTML elementov. Omogoča tudi lažje prilagajanje velikosti in razporeditve čez različne velikosti brskalnika in celo naprave.

Grid omogoča boljše možnosti prilagajanja kot CSS float. Omogoča nam tudi, da ena verzija CSS kode deluje na več napravah in da ni potrebno za različne velikosti brskalnika pisanje različnih velikosti, saj Grid to sam prilagaja.



Slika 2: Primer CSS Grid HTML razporeditve

Večina modernih brskalnikov Grid podpira, zato dodatno delo za podporo ni potrebno. Enako velja za mobilne brskalnike.

Razporeditev, kot je prikazana na sliki 2, je preprosto določena z uporabo CSS Grid v le nekaj vrsticah kode. Z uporabo navadnih CSS ukazov pa bi bilo doseganje takšne razporeditve bolj zahtevno.

3.4 MySQL

MySQL je sistem za upravljanje s podatkovnimi bazami. Je odprtokoden in implementira relacijske podatkovne baze. Uporablja jezik SQL. Deluje na principu odjemalec – strežnik.

Ker je odprtokoden, je za njega na voljo veliko orodij in pripomočkov. Za razliko od Microsoftovega SQL strežnika je MySQL strežnik lažje namestiti na različnih napravah in sistemih.

3.5 JavaScript

JavaScript je skriptni programski jezik, ki omogoča interaktivne spletne strani. Nima nobene povezave z jezikom Java, a je vseeno podoben tipičnim objektnim programskim jezikom. Sodelovanje med HTML in JavaScript je zelo preprosto in dobro implementirano.



Slika 3: JavaScript logotip

3.6 AJAX

AJAX je razvojna tehnika, ki pomaga pri razvoju interaktivnih spletnih aplikacij. Spletna aplikacija si lahko z uporabo AJAXa s strežnikom podatke izmenjuje asinhrono v ozadju, kar pomeni, da ponovno nalaganje spletne strani ni potrebno. S tem je mogoče hitrejša in uporabniku bolj prijazna spreminjanje in pošiljanje podatkov na strežnik in z njega.

Deluje s pošiljanjem zahtev HTTP, bolj podrobno – objektov XMLHttpRequest.

3.7 jQuery

jQuery je ogrodje za JavaScript, ki uporabniku omogoča lažje izbiranje in delo z elementi HTML. Programerju olajša delo, ni pa opazno hitrejša kot uporaba privzetih JavaScript funkcij in objektov.



Slika 4: jQuery logotip

4 Načrtovanje spletne aplikacije

Prvi korak pri načrtovanju spletne aplikacije je bilo zapisovanje in načrtovanje funkcij, ki jih mora aplikacija omogočati. Tu sva prišla do ugotovitve, da mora aplikacija shranjevati veliko različnih podatkov in jih na uporabniku prijazen način tudi prikazati. Omogočati mora tudi hiter in efektiven način za vnos in urejanje teh podatkov.

Vse te informacije morajo biti shranjene v centralni bazi podatkov. Za to sva uporabila MySQL podatkovno bazo. V več tabel sva razdelila vse potrebne podatke za pravilno delovanje spletne aplikacije.

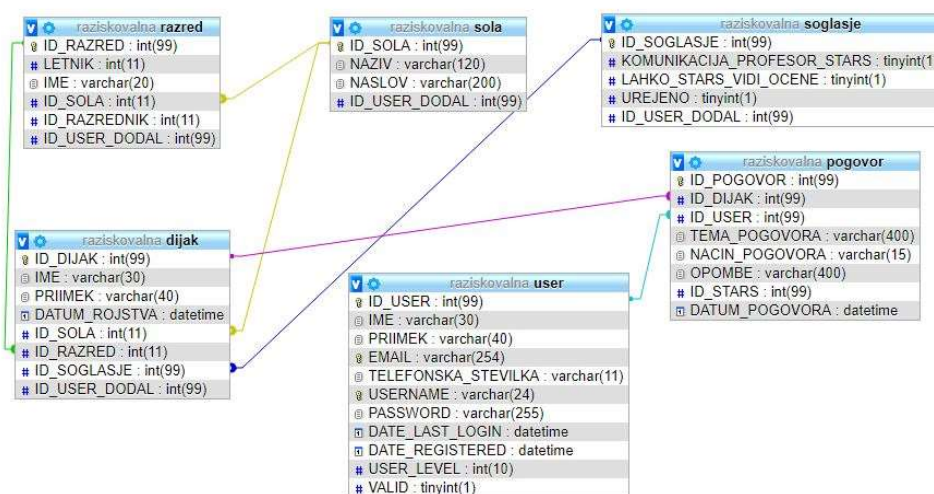
Po tem ko sva končala z načrtovanjem funkcij, ki morajo obstajati, in podatkov, ki morajo biti shranjeni v bazi, je nastopil drugi korak. Odločila sva se, kakšen bo osnovni izgled spletne aplikacije, kako se bo odzivala, ko bo uporabnik uporabljal različne elemente na njej, in kako bova to strukturo naredila s HTML jezikom.

Tretji korak je načrtovanje kode, ki je vmesnik med uporabnikovimi vnosi in podatkovno bazo. Ta koda je napisana večinoma v PHP jeziku.

4.1 Podatkovna baza

Prvi korak pri načrtovanju podatkovne baze za spletno aplikacijo je določanje podatkov, ki morajo biti v njej shranjeni. Prišla sva do ugotovitve, da je potrebnih več tabel, ki jih lahko razdelimo na dve vrsti. Prva vrsta so tabele, v katerih so shranjeni podatki, ki so potrebni za delovanje spletne aplikacije. Primer take tabele je tabela »user«. V njej so shranjeni uporabniki spletne aplikacije.

Druga vrsta tabel so tabele, v katerih se nahajajo podatki o dijakih, njihovih soglasjih, pogovorih, šolah, razredih itd. V te tabele uporabniki vnašajo podatke neposredno.



Slika 5: Podatkovna baza med razvijanjem

Ko sva pridobila vse potrebne informacije, sva lahko začela s pisanjem SQL kode, ki bo ustvarila MySQL podatkovno bazo, vse potrebne tabele in relacije. Prva verzija ni bila popolna, a sva jo med pisanjem PHP kode nenehno izboljševala in dopolnjevala z novimi podatki in spremembami. Ko sva ugotovila, da bi bilo nek problem lažje rešiti, če bi v podatkovni bazi shranjevala določen podatek, sva ta podatek dodala v podatkovno bazo. Na koncu sva prišla do delujoče in učinkovite podatkovne baze, ki ustreza vsem potrebam, ki sva jih imela.

4.2 Osnovna HTML struktura

Ko je bilo določeno, kako bo spletna aplikacija izgledala, sva lahko začela načrtovati HTML strukturo spletne strani. Odločila sva se, da bova HTML kodo razdelila v več datotek. Deljenje je na header, main in footer datoteko.

V header datoteki, ki je .php datoteka, se nahaja <header> značka (ang. tag). V njej so vse glavne značke, kot so: title, link za stylesheet, link za font, meta charset, link za ikono v navigacijskem meniju in JavaScript skripte. V tej datoteki se tudi zapre header značka in odpre body značka. Prav tako pa je čisto na začetku z uporabo PHP poklicana PHP skripta, ki nadzira, kaj lahko uporabnik dostopa. Imenuje se session.php in uporabnika, v primeru da ni vpisan ali je na strani, kjer ne bi smel biti, preusmeri na vpisno stran.

```
1 <?php
2     require("session.php");
3 >?
4 <!DOCTYPE html>
5 <html>
6 <head>
7     <title>Tehnogled</title> <!-- 7000 PHP dynamic title system-->
8     <link rel="stylesheet" type="text/css" href="bin/main.css">
9     <link href="https://fonts.googleapis.com/css?family=Roboto" rel="stylesheet">
10    <meta charset="UTF-8">
11    <meta name="author" content="Vid Kreča">
12    <link rel="shortcut icon" type="image/x-icon" href="bin/favicon.ico"/>
13
14    <!-- JAVASCRIPT -->
15    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script> <!-- jQuery -->
16    <script src="js/admin.js"></script>
17 </head>
18 <body>
```

Slika 6: Vsebina header.php datoteke

Main del, ki se nahaja v main.php datoteki, je del, ki nadzira glavno vsebino spletne aplikacije. V primeru, da vpisan uporabnik še nima potrjenega računa, mu tu pokaže le sporočilo, da mora kontaktirati administratorja, ker njegov račun še ni potrjen. Če pa je njegov račun potrjen, mu izpiše meni z gumbi do različnih sekcij aplikacije. V primeru, da je uporabnik administrator, mu pokaže tudi gumb do administratorske strani, kjer lahko vnaša in ureja različne podatke. Ko uporabnik pritisne na gumb za sekcijo, se preko URL-a z uporabo metode GET prenese spremenljivka »location«. Ta spremenljivka določa, kaj bo main.php datoteka izpisala. V primeru, da je vrednost spremenljivke »dijaki«, se bo izpisala stran za dijake. S tem sva dosegla, da je v podstraneh potrebno manj kode, saj ni potrebno ponovno izrisovati menija in drugih pomembnih HTML elementov. Na strani se izpiše tudi gumb za izpis uporabnika.

```
20 if ($valid == true) {
21     if (isset($_GET["location"])) {
22         switch ($_GET["location"]) {
23             case "dijaki": {
24                 include("template/dijaki.php");
25                 break;
26             }
27             case "pogovori": {
28                 include("template/pogovori.php");
29                 break;
30             }
31             case "soglasja": {
32                 include("template/soglasja.php");
33                 break;
34             }
35             case "mojRacun": {
36                 include("template/mojRacun.php");
37                 break;
38             }
39             case "admin": {
40                 include("template/admin.php");
41                 break;
42             }
43             case "uredi": {
44                 include("template/uredi.php");
45                 break;
46             }
47         }
48     }
49 }
```

Na sliki (levo) je prikazan del kode iz main.php datoteke. Prikazana koda preveri, če je račun vpisanega uporabnika potrjen in v primeru, da je, pridobi lokacijo iz URL-a. Če lokacija (location parameter) v URL-u ni določena, ne izpiše nobenega menija. Če pa je določena, izpiše primeren meni na sredini strani.

Admin stran je na voljo samo uporabnikom, ki imajo status administratorja. V primeru, da uporabnik, ki nima tega statusa, poskusi dostopati na to stran, ga skripta preusmeri nazaj na začetno stran.

Slika 7: Del kode iz main.php datoteke

V zadnjem delu pa se nahaja footer sekcija, kjer se nahajata samo 2 znački, in sicer `</body>` in `</html>`. Pomembni sta, ker zapirata celotni dokument.

Odločiti sva se tudi morala, kje bodo navigacijski elementi in kakšnega tipa bodo.

Uporabljenih je veliko HTML form in razdelimo jih lahko na dva tipa. Prvi tip so forme, ki imajo za akcijo PHP datoteko. Te forme ob pritisku na gumb »submit« osvežijo stran ali celo preusmerijo uporabnika na drugo stran. Drug tip form pa ob pritisku na gumb »submit« sproži JavaScript dogodek in ta pokliče AJAX PHP datoteko, ki v ozadju uredi, shrani ali pošlje podatke, brez da bi se stran kadarkoli osvežila. Drug način je zelo uporaben, ko hočemo, da uporabnik ostane na isti strani in da se hkrati ohranijo nekateri podatki iz forme, ki jo je poslal.

Obe vrsti form sta uporabni, a pri administratorskem meniju v aplikaciji sva se odločila, da je pomembno, da uporabnik (admin) vidi, kaj je ravno vnesel v bazo podatkov. Zaradi tega sva se odločila, da je najbolj smiselno, da se stran ne osveži ob potrjevanju forme.

Aplikacija za pomoč profesorjem pri nadzoru dovoljenj dijakov

```
78 <div id="dodajDijaka"> <!--
79 <form id="dodajDijakaForm" method="post">
80 <input type="text" placeholder="Ime" name="ime" id="ime">
81 <input type="text" placeholder="Priimek" name="priimek" id="priimek">
82 <input type="date" name="datumRojstva" min="1970-01-01" id="datumRojstva">
83
84 <select name="id_sola" id="id_sola_select" onchange="if (this.selectedIndex) id_sola_selectedIndexChanged(this.options[this.selectedIndex].value);">
85 <option value="-1" selected disabled hidden>Izberi šolo dijaka</option>
86 <?php
87 //Izpiše vse šole v podatkovni bazi kot izbire v drop down meniju
88 $query = "SELECT NAZIV, ID_SOLA FROM SOLA";
89 $sql = $conn->prepare($query);
90 $sql->execute();
91 $sql->store_result();
92 if ($sql->num_rows > 0) {
93     $sql->bind_result($naziv, $id_sola);
94     while ($sql->fetch()) {
95         echo '<option value="'.$id_sola.'">'.$naziv.'</option>';
96     }
97 } else echo '<option value="">V podatkovni bazi ni šol!</option>';
98 ?>
99 </select>
100 <select name="id_razred" id="id_razred_select">
101 <!-- AJAX doda <option> elemente iz razredi iz izbrane šole-->
102 </select>
103
104 <input type="submit" value="Dodaj dijaka"><span id="resultDodajDijaka"></span>
105 </form>
106 </div>
```

Slika 8: Koda za formo za vnos dijaka

4.2.1 Forme brez osveževanja celotne strani

Če želimo uporabiti AJAX pri formi, moramo narediti dogodek, ki bo čakal na pritisek »submit« gumba na specifični formi. To je najlažje narediti z JavaScript jezikom. V aplikaciji se ta koda nahaja v posebni datoteki, ki je naložena v »header« sekciji v HTML. To pomeni, da se ta JavaScript koda naloži na vseh straneh na spletni aplikaciji in je tako ni potrebno na vsaki strani kjer je forma ponovno pisati.

```
1 jQuery(document).ready(function ($) {
2     $("#dodajDijakaForm").submit(function (event) {
3         event.preventDefault();
4         //validation
5
6         var formData = new FormData(this);
7         $.ajax({
8             url: 'ajax/dodaj.php?x=dijak',
9             type: 'POST',
10            data: formData,
11            async: true,
12            cache: false,
13            contentType: false,
14            processData: false,
15            success: function (returndata)
16            {
17                //show return answer
18                //if (returndata != "") alert(returndata); //izpisovanje errorov (odstrani ko ni več potrebno!)
19                $("#dodajDijakaForm")[0].reset(); //resetiranje forme na prvotno stanje
20                $("#id_razred_select").hide();
21                $("#resultDodajDijaka").html("Dijak dodan!");
22            },
23            error: function(){
24                //alert("error in ajax form submission"); //izpisovanje errorov (odstrani ko ni več potrebno!)
25            }
26        });
27        return false;
28    });
29 });
```

Slika 9: Del JavaScript kode

Ko dogodek prestreže klik, pokliče posebno PHP datoteko, ki vsebuje skripto za vnos, urejanje ali brisanje podatkov v podatkovni bazi. Ta skripta dobi podatke iz forme preko POST metode. POST metoda je namesto GET metode uporabljena zato, ker tako uporabnik težje prestreže te podatke in je tako varnost spletne aplikacije izboljšana.

PHP skripta nato te podatke obdela. V primeru, da je forma, ki jo je poklicala, forma za vnos podatkov, jih potem ta skripta vnese v podatkovno bazo. Če je forma za spremembo podatkov, jih skripta spremeni v podatkovni bazi z uporabo SQL UPDATE stavka. Če je forma za brisanje podatkov iz podatkovne baze, je uporabljen SQL DELETE stavek, ki izbriše dano vrstico iz ustrezne tabele v podatkovni bazi.

```
4 switch ($_GET["x"]) {
5     case "dijak": {
6         //-----DIJAK-----
7         $query = "INSERT INTO SOGLASJE(KOMUNIKACIJA_PROFESOR_STARS, LAHKO_STARS_VIDI_OCENE, UREJENO, ID_USER_DODAL)
8             VALUES (0, 0, 0, ?)";
9         $sql = $conn->prepare($query);
10        $sql->bind_param('i', $_SESSION["ID_USER"]);
11        $sql->execute();
12        $last_id = $conn->insert_id;
13
14        $query = "INSERT INTO DIJAK(IME, PRIIMEK, DATUM_ROJSTVA, ID_SOLA, ID_RAZRED, ID_SOGLASJE, ID_USER_DODAL)
15            VALUES (?, ?, ?, ?, ?, ?, ?)";
16        $sql = $conn->prepare($query);
17        $sql->bind_param('sssiiii', $ime, $priimek, $datum_rojstva, $id_sola, $id_razred, $last_id, $_SESSION["ID_USER"]);
18
19        $ime = $_POST["ime"];
20        $priimek = $_POST["priimek"];
21        $datum_rojstva = $_POST["datumRojstva"];
22        $id_sola = $_POST["id_sola"];
23        $id_razred = $_POST["id_razred"];
24        break;
25    }
```

Slika 10: PHP koda za vnos v podatkovno bazo

5 Težave in rešitve

Pri načrtovanju in pisanju spletne aplikacije sva večkrat prišla do težav.

5.1 Posodabljanje strani ob potrjevanju forme

Prva večja težava je bila posodabljanje strani ob potrjevanju form. To se je dogajalo, ker sva v akciji HTML forme neposredno klicala PHP skripto. To je pomenilo, da je uporabnika ob kliku na »submit« gumb preusmerilo na PHP datoteko.

Ker tega dogajanja nisva želela, sva začela iskati rešitve. Po kratkem iskanju sva našla dve rešitvi. Prva je bila bolj začasna in ne prav uporabna – usmerjevanje tarče (target parameter v HTML kodi) v iFrame HTML element. Če bi uporabila to rešitev, se stran ne bi posodabljala, a tudi ne bi imela funkcionalnosti, ki jo omogoča druga rešitev. Rešitev, za katero sva se odločila, da jo uporabiva, je uporaba AJAX tehnike. Z uporabo le-te lahko aplikacija v ozadju opravlja operacije, stran, na kateri je forma, pa se ne osveži.

5.2 Prenos podatkov v URL

Še ena težava, ki sva jo imela, je prenašanje podatkov prek URL naslova. Želela sva, da bi si lahko uporabniki med sabo pošiljali URL naslove in tako dostopali do različnih vsebin v spletni aplikaciji. Ugotovila sva, da je to mogoče z uporabo spremenljivk v naslovu. Te spremenljivke lahko kasneje z uporabo PHP jezika uporabiva za nalaganje želene vsebine.

5.3 Nadzor uporabnikovega statusa

Zelo pomembno je, da v aplikaciji lahko do podatkov dostopajo samo uporabniki z ustreznimi pravicami.

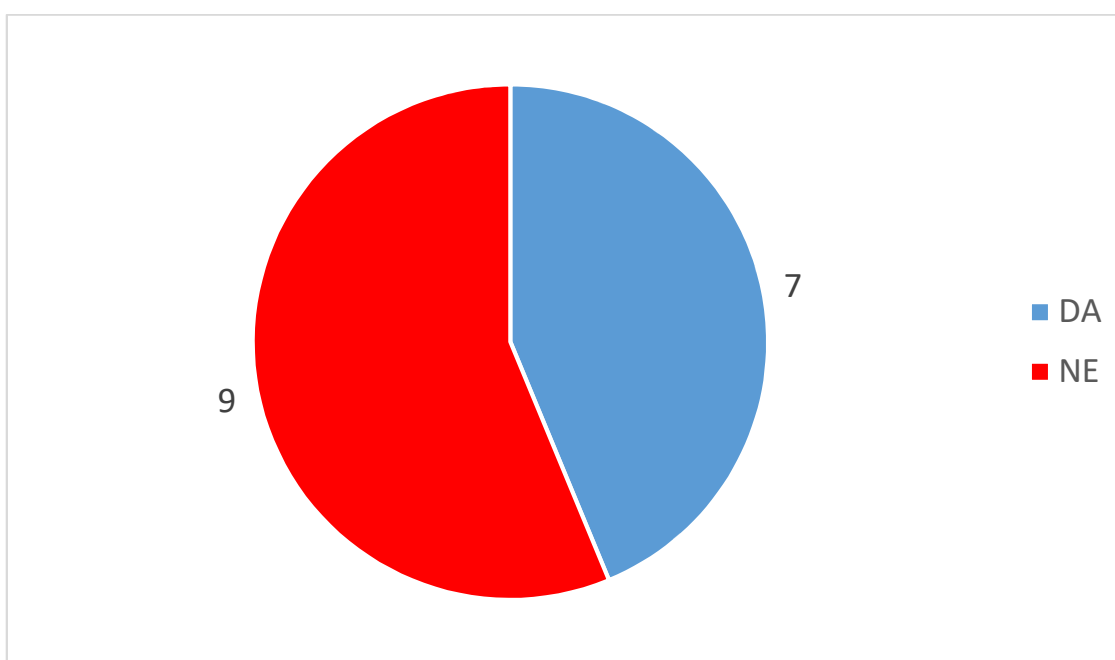
Prva stopnja tega sistema je potrjevanje novih uporabnikov. Vsak nov uporabnik ima v podatkovni bazi vrednost »valid«, nastavljeno na »false«. Dokler mu administrator te vrednosti ne spremeni na »true«, ta uporabnik ne more uporabljati spletne aplikacije.

Druga stopnja pa nadzira razliko med navadnim uporabnikom in administratorjem. Uporabljena je spremenljivka v podatkovni bazi, imenovana »user_level«. Je številka, ki določa stopnjo dostopa za uporabnika.

6 Analiza ankete

Anketo sva opravila preko spleta in sicer uporabila sva Google forms. Vprašanja sva razposlala več profesorjem, odgovorilo pa jih je 16. Rezultati, ki sva jih prejela, so bili pričakovani.

1. Ali so se pri vas z uvedbo novega zakona pokazale kakšne težave?

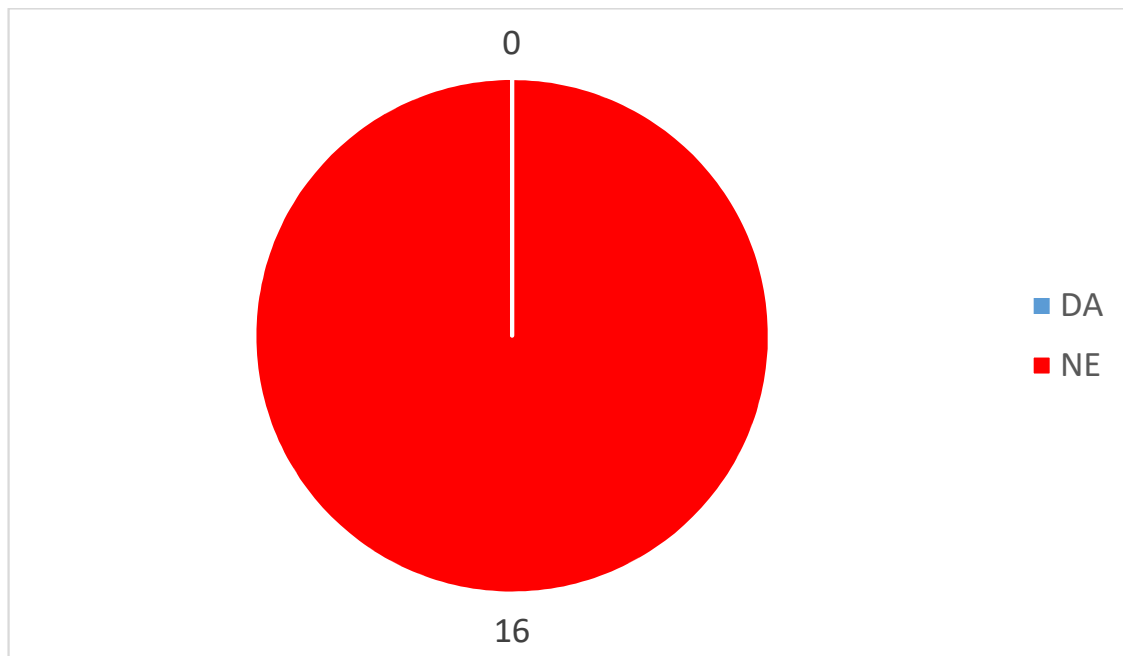


2. Če ste pri prejšnjem vprašanju odgovorili z DA, potem nam prosim naštejite nekaj težav.

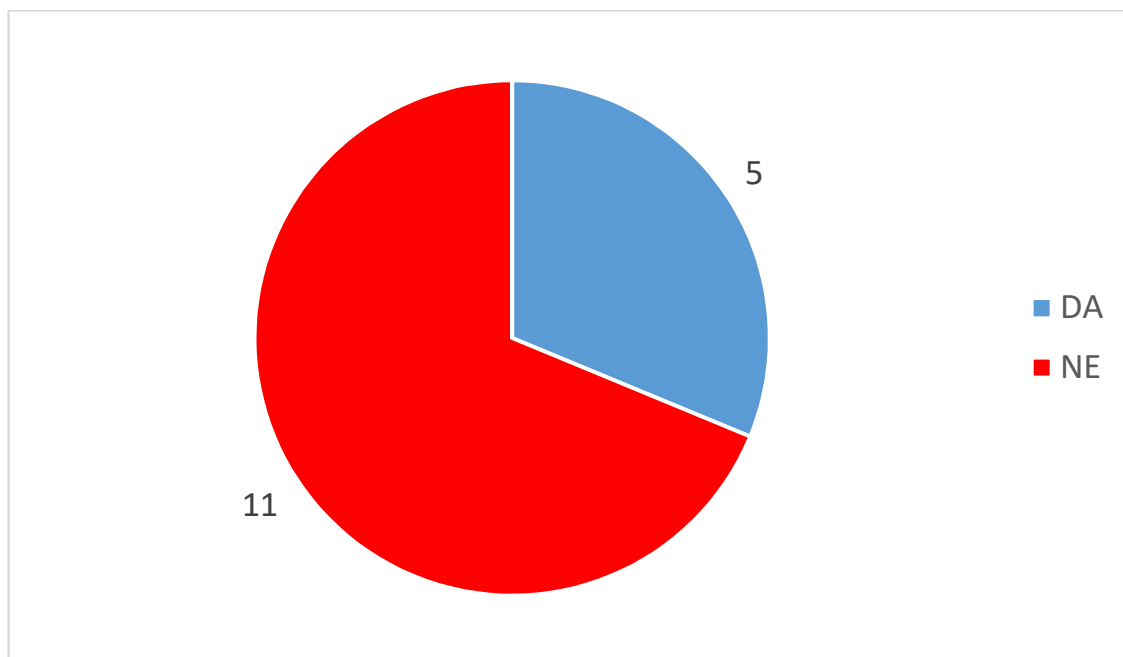
Odgovori so bili:

- ogromno »papirologije«,
- zmeda,
- premalo izobražen kader,
- izogibanje komunikaciji s starši,
- starši problematičnih otrok bodo še manj obveščeni o težavah svojih otrok.

3. Ali ste omejili odnos oziroma komunikacijo s starši, odkar velja nov zakon?



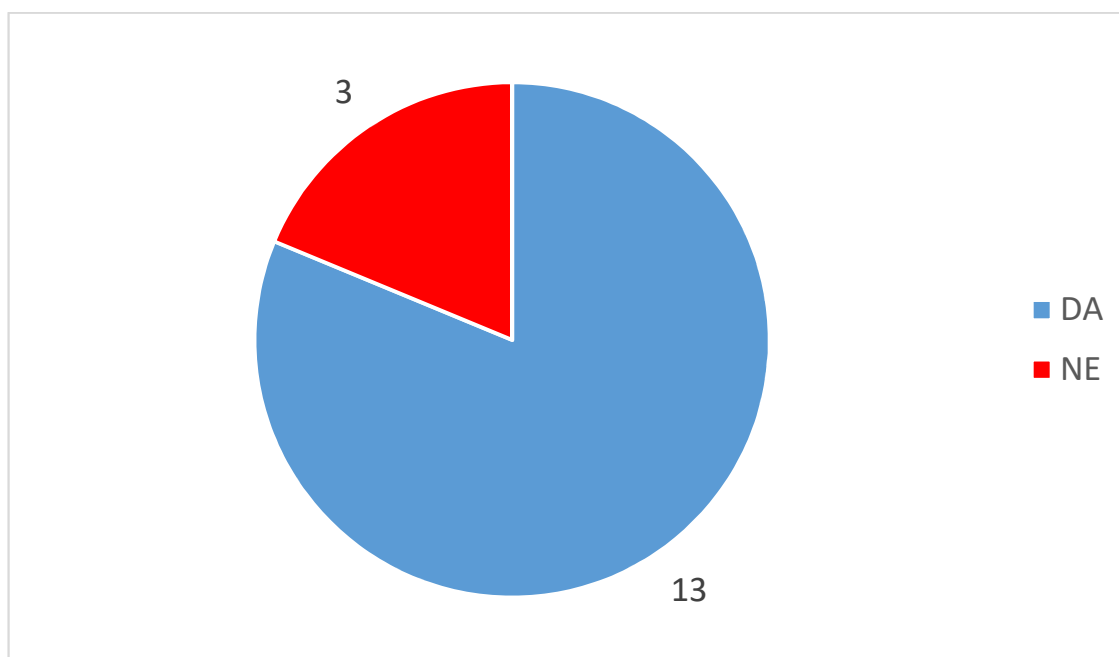
4. Ali se zavedate posledic, ki vas lahko doletijo, v primeru kršitev?



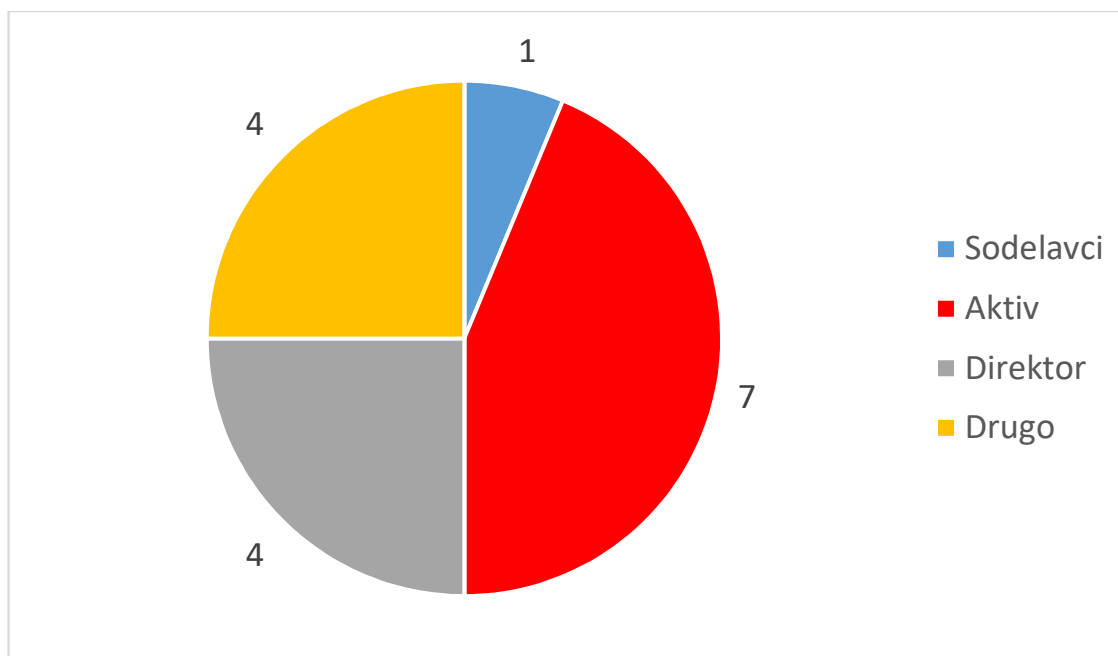
5. Ali menite, da bi vsi zaposleni morali imeti vpogled v to, kdo je podpisal dovoljenje?



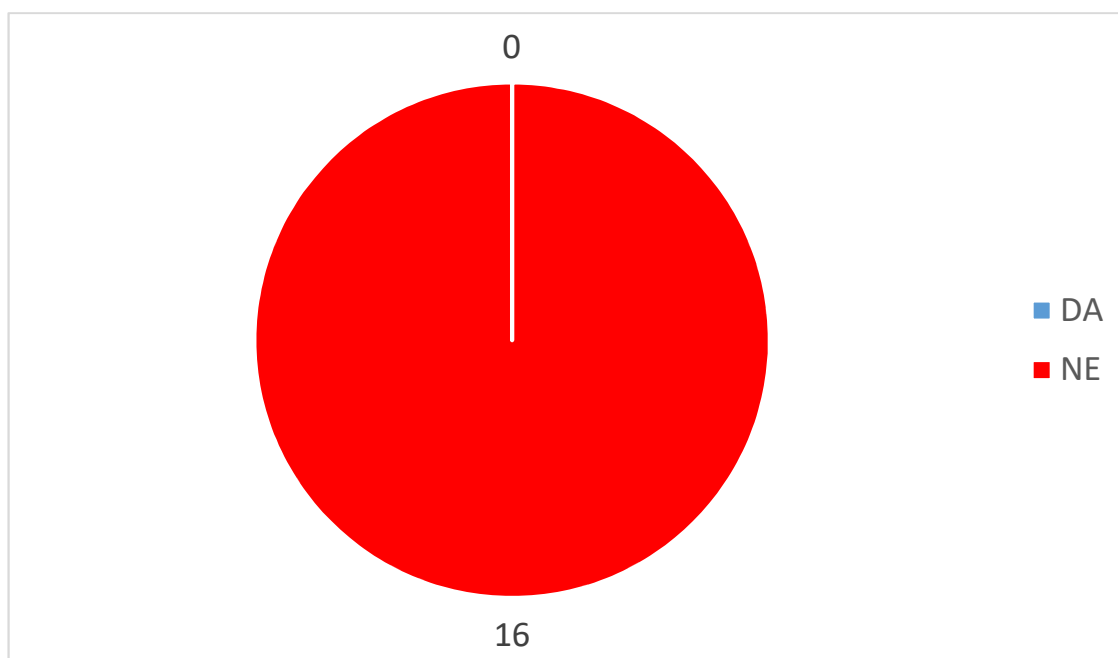
6. Ali ste v zvezi z novim zakonom prejeli dovolj informacij?



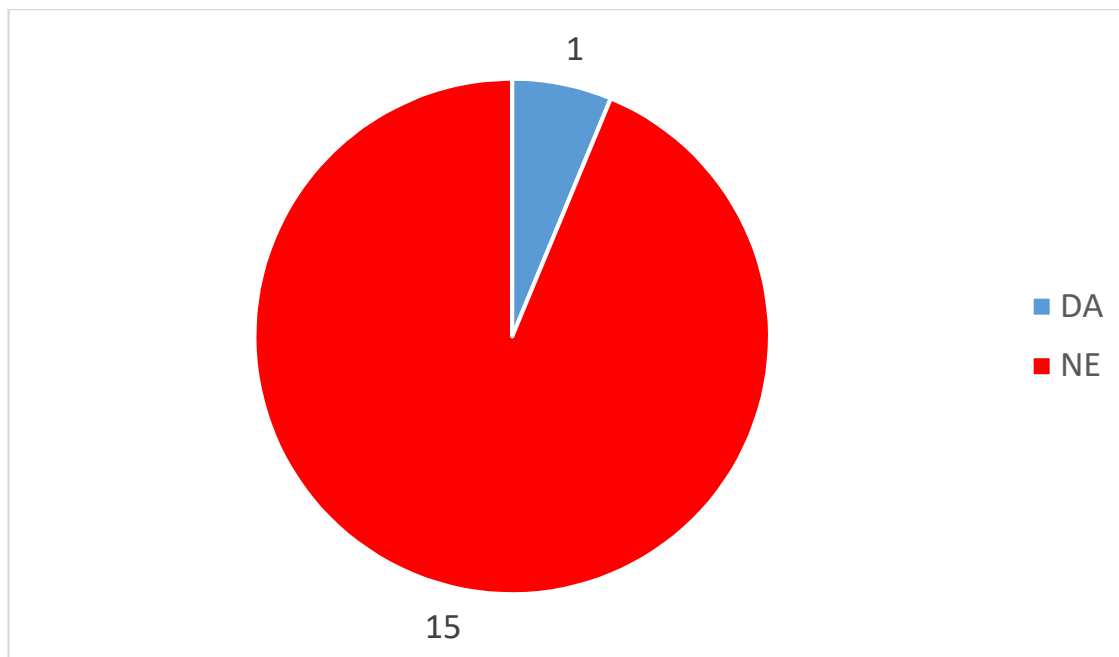
7. Kdo na šoli vam je posredoval informacije?



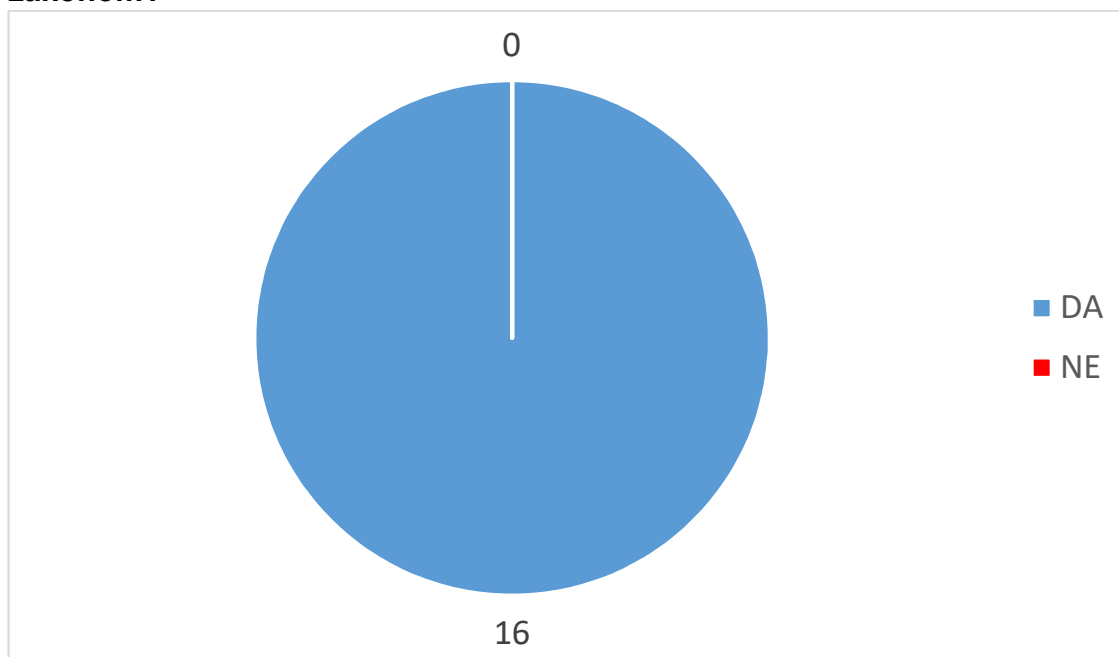
8. Ali ste imeli težave s starši v zvezi z novim zakonom?



9. Ali ste v zvezi z novim zakonom imeli težave z dijaki?



10. Ali so po vašem mnenju dijaki bili dovolj seznanjeni v zvezi z novim zakonom?



7 Zaključek

Namen najine raziskovalne naloge je bil, da raziščeva, kako so se profesorji in ravnatelji odzvali na uveljavitev novega zakona o GDPR in ustvariti aplikacijo, ki bi olajšala nadzor izbranih soglasij dijakov.

Vse najine teze so bile potrjene, ob ustvarjanju aplikacije pa sva naletela na nekaj problemov, ki sva jih uspešno rešila. Največja težava je bila osveževanje spletne strani ob klicanju ukaza, česar pa nisva želela. Prav tako sva odpravila tudi ta problem.

Najin nadaljnji cilj je izboljšati aplikacijo in uporabniški vmesnik prilagoditi, da bo zelo preprost za uporabo.

Viri

Anon., 2017. *Wikipedia*. [Elektronski]

Available at: <https://en.wikipedia.org/wiki/XAMPP>

[Poskus dostopa 4. marec 2019].

Anon., 2018. *Wikipedija*. [Elektronski]

Available at: [https://en.wikipedia.org/wiki/Atom_\(text_editor\)](https://en.wikipedia.org/wiki/Atom_(text_editor))

[Poskus dostopa 4. marec 2019].

Anon., 2019. *Atom*. [Elektronski]

Available at: <https://atom.io/>

[Poskus dostopa 4. marec 2019].

Anon., brez datuma *w3schools*. [Elektronski]

Available at: https://www.w3schools.com/whatis/whatis_github.asp

[Poskus dostopa 4. marec 2019].

BROWN, K., 2017. *How-to-geek*. [Elektronski]

Available at: <https://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/>

[Poskus dostopa 4. marec 2019].

Neznan, 2016. *Botw-pd.s3.amazonaws*. [Elektronski]

Available at: https://botw-pd.s3.amazonaws.com/styles/logo-thumbnail/s3/082014/js1_0.png?itok=9fCD5b30

[Poskus dostopa 6. marec 2019].

Neznan, 2019. *tmimgcdn*. [Elektronski]

Available at: https://s.tmimgcdn.com/blog/wp-content/uploads/2017/04/CSS-Grid_grid-template-areas.jpg?x54449

Royal, 2016. *Pngall*. [Elektronski]

Available at: <http://www.pngall.com/php-logo-png>

[Poskus dostopa 26. februar 2019].