

Web aplikacija za ispite

Ćurić, Domagoj

Undergraduate thesis / Završni rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:364029>

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

Download date / Datum preuzimanja: **2025-03-09**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA**

Stručni studij

Izrada WEB aplikacije: Online ispiti

Završni rad

Domagoj Ćurić

Osijek, 2017.

SADRŽAJ

1. UVOD	1
2. TEHNOLOGIJE IZRADE.....	2
2.1. Tehnologije korisničkog sučelja.....	2
2.2. Tehnologije serverskog sučelja.....	3
3. IZRADA ADMINISTRACIJSKOG DIJELA WEB-APLIKACIJE	4
3.1. Prijava administratora	4
3.2. Kreiranje jednostrukih odgovora u ispitu	7
3.3. Dodavanje pitanja u jednostrukim odgovorima.....	9
3.4. Dodavanje ispita i pitanja s višestrukim odgovorima.....	12
4. IZRADA KORISNIČKOG DIJELA WEB-APLIKACIJE	15
4.1. Registracija korisnika	15
4.2. Prijava korisnika.....	18
4.3. Rješavanje jednostrukih i višestrukih odgovora u ispitu	20
4.3.1. Pokretanje ispita jednostrukih odgovora	23
4.3.2. Rezultati ispita s jednostrukim odgovorima.....	25
4.3.3. Pregled rješenja ispita s jednostrukim odgovorima	28
4.3.4. Pokretanje ispita s višestrukim odabirom odgovora.....	29
4.3.5. Rezultati ispita s višestrukim odgovorima.....	32
4.3.6. Pregled rješenja ispita s višestrukim odgovorima	34
4.3.7. Brojač jednostrukih i višestrukih odgovora	36
5. ZALJUČAK	38
SAŽETAK.....	39
ABSTRACT	40
LITERATURA.....	41
ŽIVOTOPIS.....	42
PRILOZI	42

1. UVOD

Cilj je ovog završnog rada izrada web-aplikacije koja je zamišljena da omogući pristup rješavanju ispita putem Interneta. Ova aplikacija može se podijeliti na dva dijela. Prvi dio pripada učeniku/studentu koji ima mogućnost kreirati račun i prijaviti se svojim računom, nakon čega mu je omogućeno rješavanje ispita. Drugi dio aplikacije vezan je za nastavnika/profesora, on ima pristup administracijskom sučelju uz dobiveno korisničko ime i lozinku od kreatora aplikacije. U administracijskom sučelju omogućeno mu je kreiranje novih ispita i dodavanje pitanja. Administrator također može uređivati i brisati ispite, a samim time i ispitna pitanja. U ovome će se radu detaljnije opisati kôd same aplikacije i tehnologije koje su korištene. Za izradu aplikacije bitna je tehnologija kojom će se aplikacija kreirati, stoga je u drugom poglavlju ukratko opisana tehnologija korištena za njezinu izradu. Poglavlja su zasnovana na samom kôdu i njegovu izgledu u aplikaciji. Zadnje je poglavlje zaključak cjelokupnog rada povezanim s uvodom kao odgovorom na postavljeni cilj.

2. TEHNOLOGIJE IZRADE

Web-aplikacije su otvoreni ili zatvoreni sustavi upravljanja sadržajem važnim za korisnika ili klijenta. To mogu biti jednostavne web-shop (Internet trgovina) aplikacije, sustavi za upravljanje sadržajem (engl. Content Management System – CMS), knjigovodstveni sustavi ili specijalizirani sustavi upravljanja i kontrole kompletnog procesa organizacije proizvodnje u najrazličitijim poslovnim granama. Naziv web-aplikacije ne znači da su nužno smještene na internetu, već se lako, uz relativno mala uložena sredstva, mogu smjestiti u nekakvo interno okruženje korisnika ili klijenta. Web-aplikacija samo je naziv za bilo koji oblik softwera (izrađenim u nekoj web-tehnologiji – ASP, PHP, NET, Coldfusion, Adobe Air...) koji se izvršava na jednom centralnom mjestu (serveru), a na njega se može istovremeno spojiti neograničen broj korisnika koristeći se samo web-preglednikom (Explorer, Firefox, Chrome i sl.). Web-aplikacije izrađuju se u različitim tehnologijama (PHP, ASP, ASP.NET, Flash, Flex, Ajax, JavaScrip...) te koriste sve vrste baza podataka (MySQL, MSSQL, Access...). [1]

2.1. Tehnologije korisničkog sučelja

Tehnologije kojim se kreira ova aplikacija podijeljene su na tehnologije korisničkog sučelja (engl. frontend) i tehnologije serverskog sučelja (engl. backend). U ovom potpoglavlju govori se o tehnologijama korisničkog sučelja namijenjenim aplikaciji. Tu se svrstavaju HTML, Bootstrap, JavaScript i CSS. [2] HTML (engl. HyperText markup language) je osnovni jezik za izradu web-stranica. On omogućuje web-pregledniku da posloži strukturu izgleda stranice, u prevedenom značenju, to je jezik kojim stranice razgovaraju s našim web-preglednicima. Sljedeći u nizu navedenih tehnologija korisničkog sučelja spomenut je Bootstrap. [3] Bootstrap je tehnologija korisničkog sučelja (engl. front-end framework) koja se sastoji od nekoliko CSS i JavaScript dokumenata koji omogućuju bržu kreaciju web-stranica. Sadržajem JavaScript i CSS dokumenata unutar Bootstrapa omogućena je potpuna prilagodba i može ga se koristiti kao dobru polaznu točku u kreiranju web-aplikacija. [6] Jedna je od većih prednosti ove tehnologije mogućnost responzivnog dizajna (engl. RWD – responsive web design) koji omogućuje jednaku prilagodbu izgleda web-stranice na svakom uređaju, tj. veličini ekrana uređaja (tablet, mobitel, računalo...). [4] JavaScript je skriptni jezik

kojim možemo kreirati web-stranice. On se izvodi na klijentskoj strani računala i može se izvoditi u svim modernim web-preglednicima. Kao prednost ovog jezika može se navesti manja potreba komunikacije sa serverom. Može se provjeriti ispravnost podataka pri slanju stranice na server što omogućuje smanjeni promet prema-web serveru. Kao prednost navodi se mogućnost kreiranja sučelja koje reagira na korisnikove akcije tipkovnicom i mišem. Uz sve to može se koristiti za animiranije i bogatije korisničko sučelje.

2.2. Tehnologije serverskog sučelja

Za izradu ove web-aplikacije od tehnologija serverskog sučelja (engl. backend) korišten je programski jezik PHP i MySQL sustav za upravljanje relacijskih baza i podataka. [5] PHP (engl. Hypertext Preprocessor) je skriptni jezik koji se izvršava na poslužitelju (engl. server), a glavna mu je namjena dinamičko stvaranje web-stranica. PHP naredbe najčešće se nalaze unutar HTML koda zajedno s HTML oznakama. Ovaj jezik pripada objektno orijentiranim programskim (engl. OOP – Object-oriented programming) jezicima. MySQL je sustav za upravljanje relacijskim bazama i podacima koji zajedno s programskim jezikom PHP predstavlja jedno od najpopularnijih rješenja za izradu dinamičkih web-stranica i web-aplikacija temeljenih na bazama podataka. MySQL je najrasprostranjeniji otvoreni sustav za upravljanje bazama podataka na webu. Od ostalih koji se mogu usporediti sa MySQL-om mogu se navesti PostgreSQL sustav i Microsoft Access. Prednost svih ovih tehnologija je ta što su svima dostupne i besplatne.

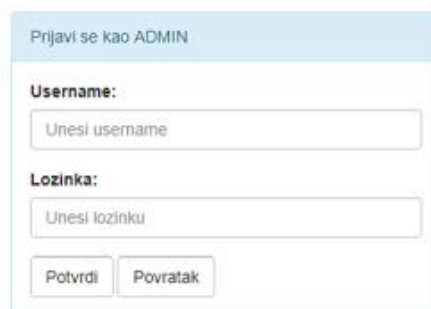
3. IZRADA ADMINISTRACIJSKOG DIJELA WEB-APLIKACIJE

U ovom će poglavlju aplikacija biti prikazana kroz kôd i detaljnije će se opisati funkcionalnost njezinih pojedinih dijelova. Prvi je dio administracijski i podijeljen je na nekoliko cjelina:

1. Prijava
2. Kreiranje ispita
3. Dodavanje pitanja
4. Pregled, uređivanje i brisanje pitanja

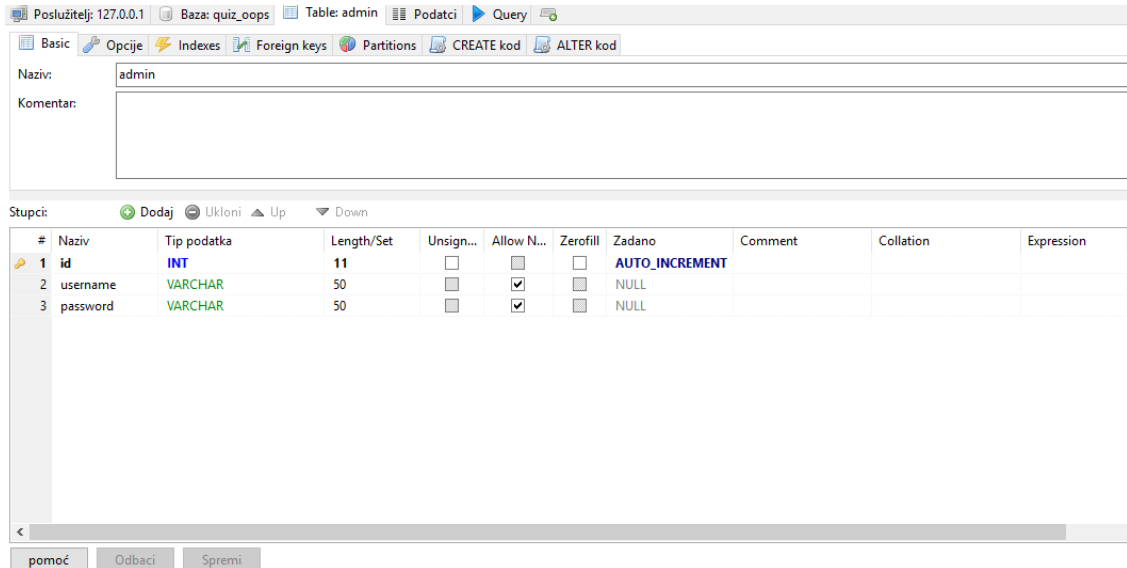
3.1. Prijava administratora

Da bi se napravio funkcionalan kôd, potrebno je razraditi ideju i pretočiti je u njega tako da svaki dio ima smisla. Dio vezan uz prijavu mora imati obrazac za prijavu (login), a u njemu je predviđeno da administrator ima svoje korisničko ime (username) i lozinku (password) kako bi samo on mogao pristupiti kreiranju i uređivanju ispita.



Sl. 3.1. Sučelje za prijavu administratora

Slika 3.1. prikazuje administracijsko sučelje za prijavu u koje se unosi korisničko ime (username) i lozinka (password). Ti podatci povezani su s aplikacijom i nalaze se u bazi podataka izrađenoj za nju. U bazi podataka nalazi se tablica s administracijskim podacima.



Sl. 3.2. Tablica admin unutar baze podataka

Slika 3.2. prikazuje tablicu admin napravljenu u bazi podataka u kojoj su podatci vezani za prijavu administratora. Jednostavna je i sastoji se od triju stupaca (id, username, password). Tablicu popunjava kreator aplikacije i dodjeljuje korisničke račune ovisno o broju administratora.

```

<div class="panel-heading">Prijavi se kao ADMIN</div>
<div class="panel-body">
  <?php
  if(isset($_GET['run']) && $_GET['run']=="failed")
  {
    echo "Your username or password is not correct";
  }
  ?>
  <form action="singinAdmin_sub.php" method="post">
  <div class="form-group">
    <label for="name">Username:</label>
    <input type="text" class="form-control" name="n" id="username" placeholder="Unesi username">
  </div>
  <div class="form-group">
    <label for="pwd">Lozinka:</label>
    <input type="password" class="form-control" name="p" id="password" placeholder="Unesi lozinku">
  </div>
  <button type="submit" class="btn btn-default">Potvrđi</button>
  <a href='index.php'><button class="btn btn-default" type='button'>Povratak</button></a>
</form>
</div>
</div>

```

Sl. 3.3. Kôd sučelja za prijavu administratora (adminlog.php)

Slika 3.3. prikazuje kôd koji se odnosi na prijavu administratora. Dio koji je napravljen u php-u ispisuje poruku u slučaju da je upisano pogrešno korisničko ime ili lozinka. Izrađen je pomoću if naredbe koja provjerava je li uvjet istinit ili pogrešan. Uvjet je postavljen uz superglobalnu varijablu Get (\$_GET). Postavljen je logički operator &&(AND) koji provjerava jesu li korisničko ime i lozinka jednaki „failed“. Ako jesu, ispisuje se poruka: „Your username or password is not correct“. Ovaj dio kôda povezan je s formom koja se nalazi ispod. Prilikom potvrde (engl. submit) forma proslijedi upisane podatke te ih referencira na sljedeću php-datoteku definiranu formom (action="singinAdmin_sub.php"), a metoda je postavljena na post (method="post").

```
<?php
include 'class/users.php';

$signinAdmin=new users;
extract($_POST);
if($signinAdmin->signinAdmin($n,$p)
{
    $signinAdmin->url("admin/index.php");
}
else
{
    $signinAdmin->url("adminlog.php?run=failed");
}

?>
```

Sl. 3.4. Kôd za prijavu administratora (singinAdmin_sub.php)

Slika 3.4. prikazuje datoteku singinAdmin_sub.php koja je vezana uz prethodnu formu. Unutar ove datoteke omogućen je pristup datoteci users.php na način da se upiše ključna riječ (include) i lokacija datoteke koja se želi koristiti, u ovom slučaju je to datoteka users.php koja se nalazi u direktoriju classes. U datoteci users.php definirane su metode koje će se koristiti za lakše i smislenije programiranje – u ovom slučaju riječ je o metodi singninAdmin koja će pobliže biti prikazana u nastavku. Otpakirana (engl. extract) je superglobalna varijabla POST(\$_POST) koja je prikupila sve podatke vezane uz formu za administracijsku prijavu i manipulira njima. Dodan je if uvjet koji u slučaju uvjeta true otvara stranicu index.php, a u suprotnom, odrađuje ispis poruke koja je bila definirana u prethodnoj datoteci superglobalnom varijablom GET i ostaje na stranici za prijavu administratora.

```

47 public function signinAdmin($username,$password)
48 {
49     $query=$this->conn->query("SELECT username,password FROM admin WHERE username='$username' AND password='$password'");
50     $query->fetch_array(MYSQLI_ASSOC);
51     if ($query->num_rows>0)
52     {
53         $_SESSION['username']=$username;
54         return TRUE;
55     }
56     else
57     {
58         return FALSE;
59     }
60 }
61 }

```

Sl. 3.5. Kôd za prijavu administratora unutar klase users (users.php)

Slika 3.5. prikazuje metodu `signinAdmin` koja prima parametre korisničko ime i lozinku. Upostavlja se veza s bazom podataka te se unutar tablice `admin` odabiru stupci korisničko ime i lozinka. Ispituje se jesu li oni jednaki parametrima u metodi nakon čega se odrađuje query i ispituje njegov uvjet. Ako je on ispunjen, korisničko ime bit će spremljeno unutar superglobalne varijable `$_SESSION`, a u suprotnom će se vratiti rezultat `FALSE` (lažan).

3.2. Kreiranje jednostrukih odgovora u ispitu

Kreiranje ispita zamišljeno je tako da mu se dodijeli naziv. Za tu svrhu potrebna je tablica u bazi podataka koja će sadržavati sve nazive ispita.

The screenshot shows a database management interface with the following details:

- Database: `quiz_oops`
- Table: `category`
- Columns:

#	Naziv	Tip podatka	Length/Set	Unsign...	Allow N...	Zerofill	Zadano	Comment	Collation
1	id	INT	4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT		
2	cat_name	VARCHAR	40	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		

Sl. 3.6. Tablica `category` unutra baze podataka

Slika 3.6. prikazuje tablicu (category) kreiranu u bazi podataka, a sastoji se od dvaju stupaca (id, cat_name). Unutar administracijskog panela omogućen je unos naziva ispita koji se sprema u navedenu tablicu.

Dodaj Ispit

Naziv novog ispita :

Pregled ispita

Šifra ispita	Ime ispita	Action
1	WEB programiranje	<input type="button" value="Edit"/> <input type="button" value="Remove"/>
4	OOP	<input type="button" value="Edit"/> <input type="button" value="Remove"/>
11	Baze podataka	<input type="button" value="Edit"/> <input type="button" value="Remove"/>
13	Digitalne komunikacije	<input type="button" value="Edit"/> <input type="button" value="Remove"/>

Sl. 3.7. Sučelje za dodavanje, uređivanje i brisanje ispita (kategorije.php)

Slika 3.7. prikazuje izgled sučelja unutar administracijskog panela koje omogućuje dodavanje ispita, odnosno njihova naziva. Nakon unosa naziva potrebno je kliknuti na gumb Dodaj da bi se ispit upisao u bazu podataka. Ispod je pregled ispita koji su stvoreni u bazi podataka. Omogućeno je uređivanje i brisanje ispita.

```

<h2>Dodaj Ispit</h2>
<div>
<form action="php_action/create.php" method="post">
  <table>
    <tr>
      <th>Naziv novog ispita :</th>
      <td>&nbsp;&nbsp;&nbsp;<input type="text" name="cat_name" placeholder="Naziv ispita" /></td>
    </tr>
    <tr>
      <td><br> <button class="btn btn-success" type="submit">Dodaj</button></td>
    </tr>
  </table>
</form>
</div>

```

Sl. 3.8. Kôd za dodavanje naziva ispita (kategorije.php)

Slika 3.8. prikazuje dio kôda koji se odnosi na unos naziva ispita, a vrši se unutar forme koja se veže na sljedeću php datoteku pod nazivom create.php. Korištena je POST metoda.

```

1  <?php
2
3  require_once 'dbconn.php';
4
5  if($_POST) {
6      $cat_name = $_POST['cat_name'];
7
8      $sql = "INSERT INTO category (cat_name) VALUES ('$cat_name')";
9      if($connect->query($sql) === TRUE) {
10         header('Location: http://localhost:8000/admin/kategorije.php');
11     } else {
12         echo "Error " . $sql . " . " . $connect->connect_error;
13     }
14
15     $connect->close();
16 }
17
18 ?>
19

```

Sl. 3.9. Kôd za unos ispita u bazu podataka (create.php)

Slika 3.9. prikazuje kôd koji je zadužen za unos naziva ispita unutar baze podataka, odnosno tablice category koja sadrži sve nazive ispita. Datoteka koja je prikazana sadrži if uvjet u kojem je smještena superglobalna varijabla POST (\$_POST). Sprema se prosljeđena varijabla u novu varijablu deklariranu kao \$cat_name. Na osmoj liniji upisana je SQL naredba kojom je definirano unošenje varijable \$cat_name u tablicu category. Na devetoj liniji dodan je još jedan if uvjet koji provjerava spajanje s bazom podataka te odrađuje SQL naredbu. Ako je uvjet zadovoljen, podatci će se upisati u bazu podataka i otvorit će se stranica definirana u headeru (funkcija zaglavlja koja šalje http oznaku prema klijentu). U suprotnom, ispisat će SQL grešku. Na kraju je zadana naredba za prekidanje spajanja.

3.3. Dodavanje pitanja u jednostrukim odgovorima

Aplikacija ima mogućnost unošenja dviju vrsta pitanja: jednostrukih i višestrukih odgovora (single i multiple questions). Ovo potpoglavlje odnosi se na pitanja s jednostrukim izborom.

Admin panel

- Početna
- Dodaj ispit(single)
- Dodaj pitanja(single)**
- Pregled pitanja(single)
- Dodaj ispit(multi)
- Dodaj pitanja(multi)
- Pregled pitanja(multi)

Pitanje:

Odgovor 0:

Odgovor 1:

Odgovor 2:

Odgovor 3:

Točan odgovor(riječima):

Activate Windows

Sl. 3.10. Sučelje za dodavanje pitanja (add_ques.php)

Slika 3.10. prikazuje sučelje unutar aplikacije koje omogućuje unos pitanja. Za unos pitanja i odgovora u bazi podataka potrebna je nova tablica.

Poslužitelj: 127.0.0.1 Baza: quiz_oops Table: questions Podatci Query

Basic Opcije Indexes Foreign keys Partitions CREATE kod ALTER kod

Naziv: questions

Komentar:

Stupci: Dodaj Ukloni Up Down

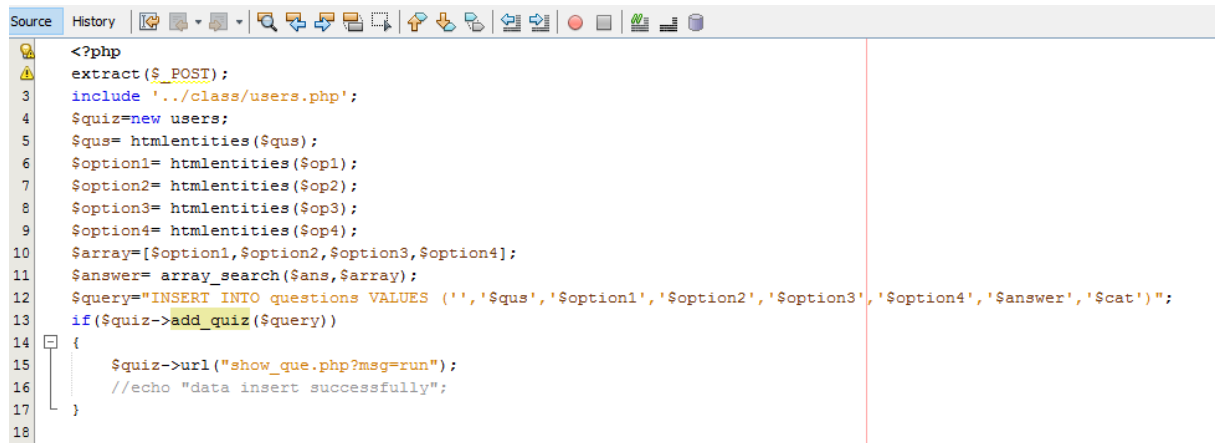
#	Naziv	Tip podatka	Length/Set	Unsign...	Allow N...	Zerofill	Zadano	Comment	Collation
1	id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT		
2	question	VARCHAR	100	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		
3	ans1	VARCHAR	80	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		
4	ans2	VARCHAR	80	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		
5	ans3	VARCHAR	80	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		
6	ans4	VARCHAR	80	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		
7	ans	VARCHAR	4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		
8	cat_id	INT	4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		

pomoć Odbaci Spremi

Sl. 3.11. Tablica questions unutar baze podataka

Slika 3.11. prikazuje tablicu questions koja je kreirana za unošenje pitanja i odgovora. Sastoji se od osam stupaca (id, question, ans1, ans2, ans3, ans4, ans,cat_id). Id označuje oznaku pitanja i postavljen je kao primarni ključ (engl. primary key – jedinstveno određuje jedan redak u tablici), stupac question služi za spremanje pitanja. Stupci ans1doans4 služe za

spremanje ponuđenih odgovora od kojih je jedan točan. Točan odgovor pohranjuje se u stupac ans, a zadnji stupac cat_id vezan je stranim ključem (engl. foreign key – pokazuje na ključ kandidata u drugoj tablici) na tablicu category i njezin stupac id tako da se može pristupiti imenu ispita za određeno pitanje.



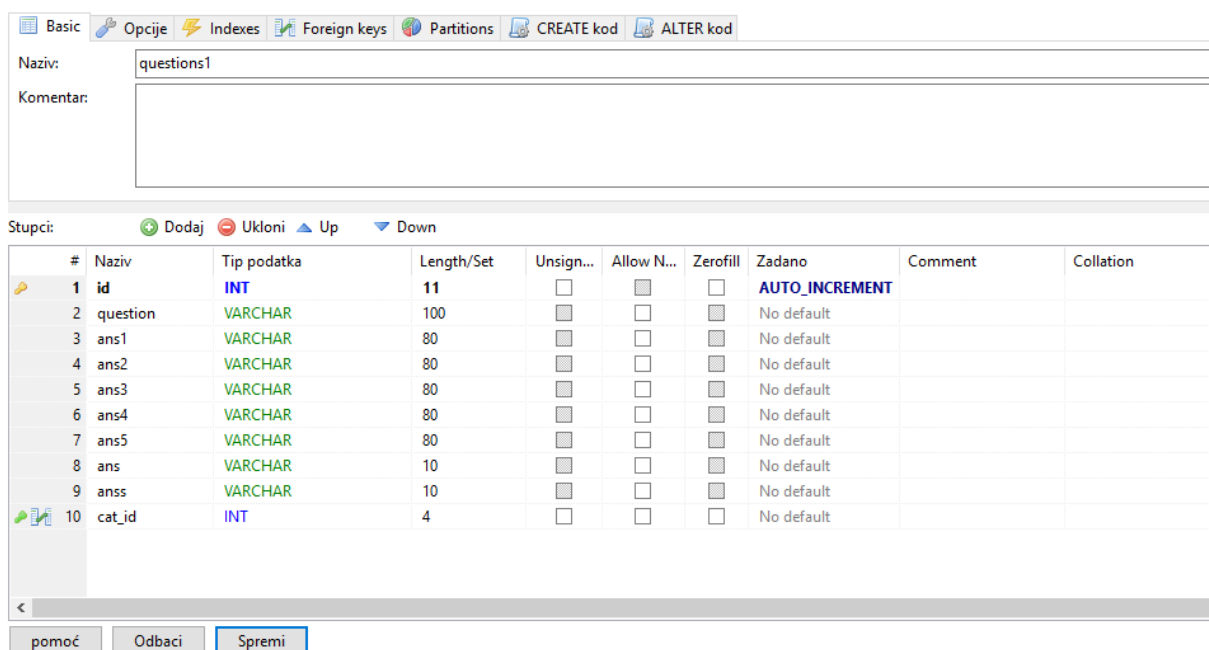
```
<?php
extract($_POST);
3 include '../class/users.php';
4 $quiz=new users;
5 $qus= htmlentities($qus);
6 $option1= htmlentities($op1);
7 $option2= htmlentities($op2);
8 $option3= htmlentities($op3);
9 $option4= htmlentities($op4);
10 $array=[$option1,$option2,$option3,$option4];
11 $answer= array_search($ans,$array);
12 $query="INSERT INTO questions VALUES ('','$qus','$option1','$option2','$option3','$option4','$answer','$cat')";
13 if($quiz->add_quiz($query))
14 {
15     $quiz->url("show_que.php?msg=run");
16     //echo "data insert successfully";
17 }
18
```

SI. 3.12. Kôd za unos pitanja u tablicu questions (add_quiz.php)

Slika 3.12. prikazuje kôd namijenjen za spremanje pitanja i odgovora u tablicu questions. Podatci koji su spremljeni u superglobalnu varijablu POST deklariraju se u ovu datoteku. Omogućuje se pristup klasi user u kojoj je kreirana metoda add_quizi. Njezina je uloga primanje parametra, u ovom je slučaju riječ o varijabli \$query. Pitanje je deklarirano varijablom \$qus, odgovori su dodijeljeni varijabli \$option1, \$option2, \$option3, \$option4. Unutar njihove deklaracije korištena je funkcija htmlentities, a u ovom slučaju aplikacija se oslanja na korisnički unos i koristi se ta funkcija. Princip rada temelji se na tome da kao argument prima string i vraća ga s izmijenjenim znakovima što znači da će se znakovi poput č, ć, đ, š, kao i ostali slični znakovi, mijenjati u njihove reprezentacije u kôdu za posebne znakove. Svi odgovori spremaju se u jednodimenzionalno polje \$array. [8] Točan odgovor pohranjuje se pomoću ugrađene funkcije array_search koja izvodi traženje vrijednosti u polju, i, ako je vrijednost potvrđena, vraća se ključ pronađenog elementa. Dvanaesta linija sadrži varijablu \$query koja definira unos varijabli u bazu podataka, odnosno u tablicu questions. Unutar if uvjeta koristi se metoda add_quiz koja omogućuje spajanje s bazom podataka.

3.4. Dodavanje ispita i pitanja s višestrukim odgovorima

Dodavanje ispita koji imaju višestruke odgovore (multiple choice) ima isti koncept kao i ispiti s jednostrukim odgovorima (single choice). Ovaj sustav sadrži tablicu s nazivima ispita kao i kod jednostrukih odgovora, međutim, ona je odvojena i napravljena zasebno, njezin je naziv `category_multi`. Tablica sadrži stupce `id` i `cat_name` kao i tablica `category`. Razlika koju vrijedi izdvojiti je ta što se ispiti s višestrukim odgovorima u ovom slučaju definiraju tako da svako pitanje ima mogućnost s pet ponuđenih odgovora od kojih je moguće da jedan ili dva odgovora budu točni.



#	Naziv	Tip podatka	Length/Set	Unsign...	Allow N...	Zerofill	Zadano	Comment	Collation
1	id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT		
2	question	VARCHAR	100	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		
3	ans1	VARCHAR	80	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		
4	ans2	VARCHAR	80	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		
5	ans3	VARCHAR	80	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		
6	ans4	VARCHAR	80	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		
7	ans5	VARCHAR	80	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		
8	ans	VARCHAR	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		
9	anss	VARCHAR	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		
10	cat_id	INT	4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		

Sl. 3.13. Tablica `questions1` unutar baze podataka

Slika 3.13. prikazuje tablicu `category_multi` koja se veže uz dodavanje višestrukih odgovora. Struktura tablice slična je prethodnoj tablici `questions` (sl. 3.11.), no vidljivo je da su dodana dva dodatna stupca, a to su `ans5` i `anss`. Stupac `ans5` sadrži ponuđeni odgovor broj 5. Stupac `anss` prema index drugog točnog odgovora u pitanju. U tablici `questions` postojao je samo jedan točan odgovor, no u tablici `questions1` točni odgovori spremaju se u stupce `ans` i `anss`. Primarni ključ (engl. primary key) i strani ključ (engl. foreign key) dodani su na iste stupce kao u tablici `questions`. Sučelje za prikaz ispita i pregled ispita jednak je jednostrukim odgovorima, no dodavanje pitanja jedina je razlika u ovom dijelu.

Početna

Dodaj ispit(single)

Dodaj pitanja(single)

Pregled pitanja(single)

Dodaj ispit(multi)

Dodaj pitanja(multi)

Pregled pitanja(multi)

Pranje:

Unesi pitanje

Odgovor 1:

Unesi odgovor 1

Odgovor 2:

Unesi odgovor 2

Odgovor 3:

Unesi odgovor 3

Odgovor 4:

Unesi odgovor 4

Odgovor 5:

Unesi odgovor 5

Točan odgovor 1(riječima):

Unesi odgovor

Točan odgovor 2(riječima):

Unesi odgovor

Klase

Submit

Sl. 3.14. Sučelje za dodavanje pitanja (add_ques_multi.php)

Slika 3.14. prikazuje sučelje koje je namijenjeno za unos pitanja i ponuđenih odgovora vezanih uz višestruke odgovore (multiple choice). Struktura sučelja slična je kao i u jednostrukim odgovorima. Kao što su u tablici dodana dva stupca, u ovom sučelju dodana su dva inputa. Njihova svrha objašnjena je u prethodnom tekstu vezanom za kreiranje tablice questions1 (stupci ans5 i anss). Pitanje ima mogućnost jednog ili dvaju točnih odgovora. Pri unosu jednog točnog odgovora potrebno je isti odgovor unijeti u rubriku „Točan odgovor 1 (riječima)“ i „Točan odgovor 2 (riječima)“. Za unos dvaju točnih odgovora oni se unose u pojedinačnu rubriku. Na kraju se odabire ispit u koji će se spremi pitanje (padajući izbornik) i potvrditi gumbom „Submit“.

```

1 <?php
2 extract($_POST);
3 include '../class/users.php';
4 $quiz_m=new users;
5 $qus= htmlentities($qus);
6 $option1= htmlentities($op1);
7 $option2= htmlentities($op2);
8 $option3= htmlentities($op3);
9 $option4= htmlentities($op4);
10 $option5= htmlentities($op5);
11 $array=$option1,$option2,$option3,$option4,$option5);
12 $answer= array_search($ans,$array);
13 $answer=$answer+1;
14 $answer1= array_search($ans,$array);
15 $answer1=$answer1+1;
16 $query="INSERT INTO questions1 VALUES ('','$qus','$option1','$option2','$option3','$option4','$option5','$answer','$answer1','$cat_m')";
17 if($quiz_m->add_quiz($query))
18 {
19     $quiz_m->url("show_que.php?msg=run");
20     //echo "data insert successfully";
21 }
22
23
24
25
26 ?>

```

Sl. 3.15. Kôd za unos pitanja u tablicu questions1 (add_quiz_multi.php)

Slika 3.15. prikazuje php kôd namijenjen unosu pitanja u tablicu questions1. Deklaracija varijabli slična je kao i u php kôdu koji je unosio sadržaj pitanja vezan uz jednostruke odgovore (sl. 3.12.), stoga će se navesti razlika između dvaju kôdova jer imaju jednaku funkcionalnost. Razlika je u tome što su dodane dvije varijable više, a to su varijabla \$option5 koja sprema odgovor pod brojem 5 i varijabla \$answer1 koja sprema drugi točan odgovor. Na linijama 13 i 15 (sl. 3.15.) varijable \$answer i \$answer1 uvećane su da se pomakne njihovo indeksno mjesto za 1. Taj dio bit će kasnije objašnjen i trenutno nema neku svrhu. Važno je zapamtiti da indeksi unutar jednostrukih odgovora (točni odgovori) počinju od 0, a kod višestrukih odgovora njihov index uvećava se za 1 te kreće od broja 1 ($0+1=1$). Provjerava se veza i unose podatci u tablicu question1. Ovime je završeno poglavlje vezano za administracijski dio.

4. IZRADA KORISNIČKOG DIJELA WEB-APLIKACIJE

Ovaj dio kôda aplikacije odnosi se na izradu korisničkog sučelja. Kôd se može podijeliti na nekoliko cjelina, a svaka od njih dijeli se na:

1. Registraciju
2. Prijavu
3. Rješavanje ispita (jednostruki i višestruki odgovori)
4. Pregled rješenja (jednostruki i višestruki odgovori)

4.1. Registracija korisnika

Da bi korisnik imao mogućnost korištenja ove web-aplikacije, treba imati otvoren svoj korisnički račun u aplikaciji. Svaki korisnik treba napraviti svoj korisnički račun, a za prijavu se koristi se e-mail adresom koja je jedinstvena.

Registriraj se

Izradi račun

Ime:

Prezime:

Email:

Lozinka:

Dodaj sliku:
 Nije odabran...na datoteka.

Ako si registriran [PRIJAVI SE](#)

Sl. 4.1. Sučelje za izradu korisničkog računa

Slika 4.1. prikazuje izgled sučelja u aplikaciji. Da bi se korisnik uspješno registrirao, potrebno je popuniti sve podatke unutar sučelja. Podatci su koji se popunjavaju ime, prezime, e-mail, lozinka korisničkog računa i fotografija profila. Na kraju je potrebno kliknuti na gumb „Potvrdi“. Na samom dnu nalazi se poveznica (engl. link – poveznica na sljedeći sadržaj u web-aplikaciji ili web-stranici) „Prijavi se“ koja vodi do sučelja za prijavu ako je sve prošlo u redu prilikom registracije. Kad se korisnik uspješno registrira, prikazat će mu se poruka „Your registration successfully done“.

Poslužitelj: 127.0.0.1 Baza: quiz_oops Table: signup Podatci Query

Basic Opcije Indexes Foreign keys Partitions CREATE kod ALTER kod

Naziv: signup

Komentar:

Stupci: Dodaj Ukloni Up Down

#	Naziv	Tip podatka	Length/Set	Unsign...	Allow N...	Zerofill	Zadano	Comment	Collation
1	id	INT	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT		
2	name	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		
3	surname	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		
4	email	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		
5	pass	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		
6	img	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		

Sl. 4.2 Tablica signup u bazi podataka

Slika 4.2. prikazuje tablicu koja je kreirana u svrhu registracije korisnika. Tablica ima naziv signup te sadrži šest stupaca: id, name, surname, email, pass, img. Na stupac id postavljen je primarni ključ koji se automatski uvećava pri dodavanju novog korisnika. Stupac name sadrži ime korisnika, a stupac surname njegovo prezime. Stupac email sadrži njegovu e-mail adresu, dok se unutar stupca pass sprema lozinka koja je upisana unutar sučelja. Zadnji stupac img zadužen je za pohranu slike korisnika. Kada je kreirana tablica signup, omogućen je prostor za unos podataka registracije.

```

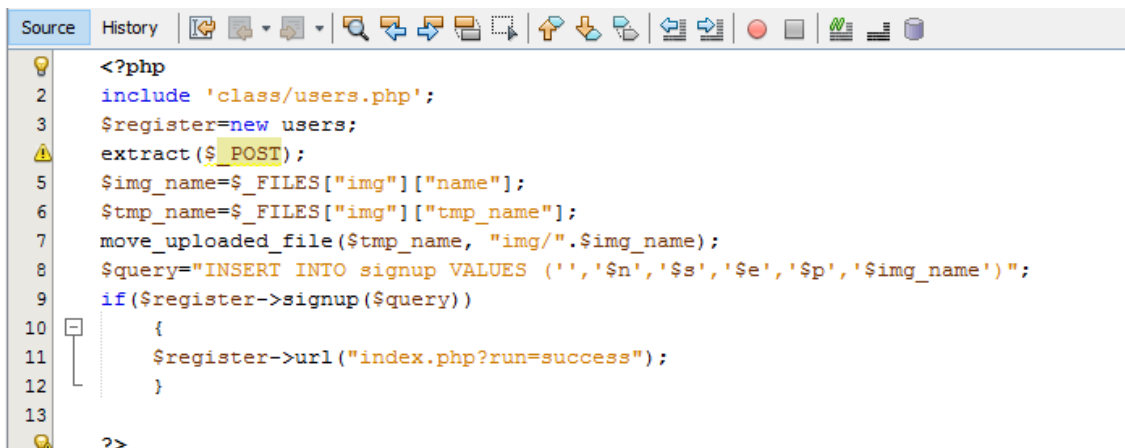
if(isset($_GET['run']) && $_GET['run']=="success")
{
    echo "<mark>Your registration successfully done</mark>";
}

?>
<form method="post" action="signup_sub.php" enctype="multipart/form-data">
<div class="form-group">
    <label for="name">Ime:</label>
    <input type="text" class="form-control" name="n" id="name" placeholder="Unesi ime">
</div>
<div class="form-group">
    <label for="surname">Prezime:</label>
    <input type="text" class="form-control" name="s" id="surname" placeholder="Unesi prezime">
</div>
<div class="form-group">
    <label for="email">Email:</label>
    <input type="text" class="form-control" name="e" id="email" placeholder="Unesi mail">
</div>
<div class="form-group">
    <label for="pwd">Lozinka:</label>
    <input type="password" class="form-control" name="p" id="pwd" placeholder="Unesi lozinku">
</div>
<div class="form-group">
    <label for="image">Dodaj sliku:</label>
    <input type="file" class="form-control" name="img" id="file">
</div>
<button type="submit" class="btn btn-default">Potvrdi</button>
<p><br>Ako si registriran <a href="index.php"> PRIJAVI SE</a></p>
</form>

```

Sl. 4.3. Kôd sučelja za izradu korisničkog računa (index_reg.php)

Slika 4.3. prikazuje napisani kôd koji se veže uz sučelje za registraciju korisnika. Na samom vrhu kôda nalazi se ispis poruke u slučaju izvršene registracije: „Your registration successfully done“. Unutar forme definirana je metoda POST koja se veže uz datoteku signup_sub.php gdje se obavlja manipulacija prikupljenih podataka. Također je dodan i atribut enctype čija je uloga definiranje načina o kodiranju podataka pri slanju na server. Taj se atribut može definirati jedino uz metodu POST. Unutar atributa definirana je vrijednost multipart/form-data i obavezna je kada se upotrebljavaju obrasci za kontroliranje prijenosa podataka, u ovom slučaju za slike. Prikupljanje i slanje podataka odvijaju se aktivacijom gumba „Potvrdite“.



```
<?php
2 include 'class/users.php';
3 $register=new users;
4 extract($_POST);
5 $img_name=$_FILES["img"]["name"];
6 $tmp_name=$_FILES["img"]["tmp_name"];
7 move_uploaded_file($tmp_name, "img/".$img_name);
8 $query="INSERT INTO signup VALUES ('','$n','$s','$e','$p','$img_name)";
9 if($register->signup($query))
10 {
11     $register->url("index.php?run=success");
12 }
13
?>
```

Sl. 4.5. Kôd namijenjen za unos podataka u tablicu signup (signup_sub.php)

Slika 4.5. prikazuje datoteku signup_sub.php koja je zadužena za unos podataka u bazu, tj. u tablicu signup. Navedena datoteka koristi pristup datoteci users.php, odnosno klasi users u kojoj je smještena metoda url i signup, a koristi se na kraju ovog kôda. Na trećoj liniji instanciran je objekt \$register. Ovakavo programiranje pripada objektno orijentiranom programiranju (OOP). Instanciranjem objekta omogućen je pristup klasi users i moguće je dohvatiti sve metode u njoj. U ovom je slučaju potrebna metoda url i signup koja vodi na stranicu index.php te u slučaju uspješne registracije ispisuje poruku: „Your registration successfully done“, a ona je prije definirana u datoteci index_reg.php. U datoteci je korištena funkcija move_upload_file zadužena za premještanje slike korisnika na novu lokaciju. Na osmoj liniji definirana je varijabla \$query koja sadrži SQL naredbu za unos podataka u tablicu signup. U zadnjem dijelu kôda koristi se if naredba u kojoj je smještena metoda signup. Ona prima argument, sada je to varijabla \$query, te se izvršava unos u tablicu. Ako je uvjet zadovoljen, otvara se stranica index.php i ispisuje se poruka za uspješan unos. Taj dio je u metodi url.

4.2. Prijava korisnika

Nakon uspješne registracije korisnika omogućena mu je prijava čiji je postupak sličan kao i prijava administratora, jedina je razlika što se korisnik prijavljuje adresom svoje elektronske pošte, dok se admin prijavljuje korisničkim imenom.



Prijavi se

Email:

Lozinka:

Novi korisnik? [Registriraj se](#) Besplatno



PRIJAVI ISPIT

Sl. 4.6. Sučelje za prijavu korisnika

Slika 4.6. prikazuje sučelje za prijavu korisnika što je ujedno i početna stranica otvaranja aplikacije. S lijeve strane nalazi se link namijenjen adminu. Klikom na „Prijavi se kao ADMIN“, otvara se stranica sa sučeljem za prijavu admina. S desne strane nalazi se link studomata koji podsjeća studenta da prijavi ispit ako ga nije prijavio. Ispod gumba kojim se potvrđuju podatci ulaska u aplikaciju („Potvrdi“) također je link („Registriraj se“) koji navodi korisnike bez otvorenog računa na registraciju.

```

<div class="panel-heading">Prijavi se</div>
<div class="panel-body">
  <?php
  if(isset($_GET['run']) && $_GET['run']=="failed")
  {
    echo "Your email or password is not corrcet";
  }
  ?>
  <form action="signin_sub.php" method="post">
  <div class="form-group">
    <label for="email">Email:</label>
    <input type="email" class="form-control" name="e" id="email" placeholder="Unesi mail">
  </div>
  <div class="form-group">
    <label for="pwd">Lozinka:</label>
    <input type="password" class="form-control" name="p" id="pwd" placeholder="Unesi lozinku">
  </div>
  <button type="submit" class="btn btn-default">Potvrdi</button>
  <p><br>Novi korisnik? <a href="index_reg.php"> Registriraj se</a> Besplatno</p>
  </form>
</div>
</div>

```

Sl. 4.7. Kôd sučelja za prijavu korisnika (index.php)

Slika 4.7. prikazuje kôd namijenjen za prijavu korisnika. Dio kôda koji je napisan unutar if naredbe odnosi se na ispis poruke ako email ili password (lozinka) nisu dobro

napisani. Kao i kod prijave admina, ovaj dio kôda ima istu sintaksu i funkcionalnost. Unutar forme deklarirana je metoda POST koja prikuplja podatke i prosljeđuje ih sljedećoj datoteci (signin_sub.php) definiranoj unutar forme. E-mail ima naziv e, a password (lozinka) ima naziv p. Taj naziv definiran je unutar inputa emaila i passworda, pod atributom name.

```
1 <?php
2 include 'class/users.php';
3
4 $signin=new users;
5 extract($_POST);
6 if($signin->signin($e,$p))
7 {
8     $signin->url("home.php");
9 }
10 else
11 {
12     $signin->url("index.php?run=failed");
13 }
14
15 ?>
```

Sl. 4.8. Kôd za prijavu korisnika (signin_sub.php)

Slika 4.8. prikazuje kôd koji se izvršava nakon potvrde forme koja je prethodno prikazana. U ovom dijelu kôda instanciran je objekt te omogućen pristup metodama unutar klase. If naredba sadrži metodu signin koja prima parametre \$e i \$p i provjerava postoje li upisani e-mail i password (lozinka) unutar baze podataka (tablice signup). U slučaju istinite tvrdnje, tj. pronalaska e-maila i passworda (lozinke) unutar tablice signup, otvorit će se stranica home.php, u suprotnom će se ispisati poruka na stranici index.php (stranica za prijavu korisnika) „Your email or password is not correct“.

4.3. Rješavanje jednostrukih i višestrukih odgovora u ispitu

Nakon prijave korisnika otvara se stranica home.php koja ima funkciju odabira ispita te njihovog pokretanja.

Online ispit

Početna

Profil

Odjava

Dobrodošli

Upute za pokretanje ispita:

1. Pritisni button "Pokreni ispit"
2. U padajućem izborniku odaberi ispit koji želiš riješavati
3. Zatim kada ste odabrali kliknite na button "Pokreni"

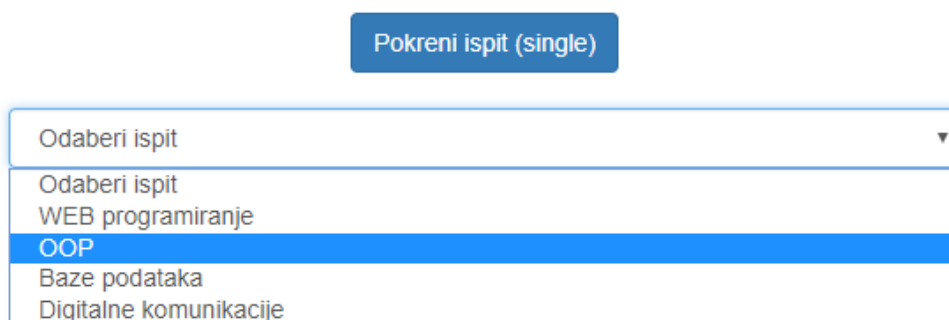
Napomena!!! Imate određeno vrijeme u kojem morate riješiti ispit. Timer se nalazi desno gore u kutu kada pokrenete ispit. SRETN0

Pokreni ispit (single)

Pokreni ispit (multiple)

Sl. 4.9. Sučelje nakon prijave (home.php)

Slika 4.9. prikazuje sučelje stranice home.php. Stranica ima dvije kartice, prvu pod nazivom „Početna“, drugu pod nazivom „Profil“, dok se u gornjem desnom dijelu stranice nalazi link za odjavu. Na kartici „Početna“ nalaze se upute za pokretanje ispita. Postoji mogućnost pokretanja jednostrukih i višestrukih ispita i oni su razdvojeni na dva plava gumba. Klikom na njih otvara se mogućnost odabira ispita. Kartica „Profil“ sadrži podatke vezane za korisnički račun i nije omogućeno uređivanje podataka, već samo pregled. Prikazuje ime, prezime, e-mail adresu i fotografiju korisnika. Link „Odjava“ služi za odjavu korisnika iz aplikacije.



Sl. 4.10. Gumb „Pokreni ispit (single)“ s mogućnošću odabira ispita

Slika 4.10. prikazuje padajući izbornik. Klikom na gumb „Pokreni ispit (single)“. U izborniku se nalaze ispiti (jednostruki odgovori) koje je moguće rješavati. Nakon odabira ispita pokrenut će se odabrani ispit.


```

<div class="col-sm-6">
  <center>
    <button type="button" class="btn btn-primary" data-toggle="tab" href="#select">Pokreni ispit (single)</button>
  </center>
</div>

<div class="col-sm-6"><br>
<div id="select" class="tab-pane fade">
  <form method="post" action="qus_show.php">
<select class="form-control" id="" name="cat">
  <option>Odaberi ispit</option>
<?php
foreach($profile->cat as $category)
{?>
  <option value="<?php echo $category['id']; ?>"><?php echo $category['cat_name']; ?> </option>
<?php }?>
</select>
  <br>
<center><input type="submit" value="Potvrđi" class="btn btn-primary" /></center>
</form>
</div>
</div>

```

Activate Windows

Sl. 4.11. Kôd za ispis ponuđenih ispita (home.php)

Slika 4.11. prikazuje kôd koji se odnosi na prikaz ponuđenih ispita u padajućem izborniku. Unutar oznaka <button> (HTML oznaka kojom se definira korištenje gumbova, eng. Button, unutar stranice) definirano je korištenje gumba i dodana mu je referenca #select, ona se odnosi na div koji je ispod definiran i pomoću njega ispisat će se ispiti. Izbornik je smješten u html oznaci <div> (pripada u HTML oznake, a predstavlja blok strukturalni element među kojim mogu biti ugniježđeni gotovo svi HTML elementi. Tri su osnovna svojstva: style, class i id) gdje je definiran njegov id=select kako bi se povezao s gumbom na koji će se referencirati (#select). Unutar div oznaka nalazi se i forma kojom je definirana metoda POST i referenca na sljedeću datoteku qus_show.php, a služi za prikaz pitanja odabranog testa. Padajući izbornik definira se unutar oznake <select>. Svi su ispiti ispisani pomoću foreach petlje u kojoj se koristi ispis pomoću metode definirane u klasi users.php.

```

public function cat_shows ()
{
  $query=$this->conn->query("SELECT * FROM category");
  while ($row=$query->fetch_array(MYSQLI_ASSOC) )
  {
    $this->cat []=$row;
  }
  return $this->cat;
}

```

Sl. 4.12. Metoda cat_show() unutar klase users (users.php)

Slika 4.12. prikazuje navedenu metodu koja dohvaća ispise iz baze podataka, odnosno tablice category te ih ispisuje u padajući izbornik pomoću foreach petlje.

4.3.1. Pokretanje ispita jednostrukih odgovora

Nakon odabira ispita omogućeno je rješavanje ispita. Svaki je ispit vremenski ograničen pa se nakon isteka vremena automatski prikazuje konačan rezultat.

Online ispit 09:58

1 All variables in PHP start with which symbol?

1 @

2 %

3 \$

4 &

2 What does PHP stand for?

1 PHP: Hypertext Preprocessor

2 Private Home Page

3 Personal Hypertext Processor

4 None of it

Sl. 4.13. Slika sučelja pri pokretanju ispita

Slika 4.13. prikazuje sučelje nakon pokretanja ispita. U gornjem desnom dijelu stranice nalazi se brojač koji prikazuje preostalo vrijeme za rješavanje ispita. Pitanja su označena crvenom bojom, dok su odgovori predstavljeni zelenom. Moguće je označiti samo jedan odgovor jer je uz njega korišten opcijski gumb (engl. Radio button – tip HTML elementa koji omogućava odabir samo jedne moguće opcije. Nakon oznake točnih odgovora potrebno je potvrditi navedeno klikom na gumb „Potvrdi“. Ako vrijeme istekne, automatski će se potvrditi označeni odgovori.

```

<form method="post" id="form1" action="answer.php">
<table class="table table-bordered">
<thead>
  <tr class="danger">
    <th><?php echo $i;?> <?php echo $qust ['question'];?> </th>
  </tr>
</thead>
<tbody>
  <?php if(isset ($qust ['ans1']))?>
  <tr class="info">
    <td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="radio" value="0" name="<?php echo $qust['id'];?>"/> &nbsp;&nbsp;<?php echo $qust ['ans1'];?></td>
  </tr>
  <?php } ?>
  <?php if(isset ($qust ['ans2']))?>
  <tr class="info">
    <td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="radio" value="1" name="<?php echo $qust['id'];?>"/>&nbsp;&nbsp;<?php echo $qust ['ans2'];?></td>
  </tr>
  <?php } ?>
  <?php if(isset ($qust ['ans3']))?>
  <tr class="info">
    <td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="radio" value="2" name="<?php echo $qust['id'];?>"/>&nbsp;&nbsp;<?php echo $qust ['ans3'];?></td>
  </tr>
  <?php } ?>
  <?php if(isset ($qust ['ans4']))?>
  <tr class="info">
    <td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="radio" value="3" name="<?php echo $qust['id'];?>"/>&nbsp;&nbsp;<?php echo $qust ['ans4'];?></td>
  </tr>
  <?php } ?>

```

Sl. 4.14. Kôd prikazivanja pitanja (ques_show.php)

Slika 4.14. prikazuje kôd kojim je kreirano sučelje nakon pokretanja ispita (sl. 4.13). Prikazivanje pitanja kreirano je formom definiranom metodom POST, a ona se referencira na sljedeću datoteku answer.php zaduženu za prikazivanje rezultata ispita. U formi je definiran atribut id=form1 jer pomoću njega definiramo brojaču (timeru) koji će se dio potvrditi nakon isteka vremena. U kôdu je dodana tablica s pitanjima i odgovorima. Pitanja su u zaglavlju tablice (<thead>) i označena su klasom „danger“ (crvenom bojom), dok su odgovori smješteni u tijelo tablice (<tbody>). Pitanja su ispisana pomoću foreach petlje koja se ne vidi na slici, no ima jednaku sintaksu kao i ispisivanje ispita u padajućem izborniku. Svaki Radio button nalazi se unutar if uvjeta čiji je zadatak funkcijom isset provjeriti je li označen taj odgovor. Svaki odgovor ima svoju vrijednost, od 0 do 3. Tako su im dodijeljene vrijednosti jer se pri postavljanju točnog odgovora pregledava polje u kojem su smješteni odgovori. Podsjetnik je da je unutar polja prvo indeksirano mjesto 0. Kod višestrukih odgovora prikazat će se druga opcija. U atributu name smješten je id svakog odgovora jer je on jedinstven i na njega je postavljen primarni ključ.

```

  <?php } ?>
  <tr class="info">
    <td><input type="radio" checked="checked" style="display: none" value="no_attempt" name="<?php echo $qust['id'];?>"/></td>
  </tr>
</tbody>
</table>
<?php $i++;?>
<center><input type="submit" value="Potvrđi" class="btn btn-success"/></center>
</form>
</div>

```

Sl. 4.15. Kôd prikazivanja pitanja (ques_show.php)

Slika 4.15. prikazuje dio kôda koji se ne vidi na prethodnoj slici (sl. 4.14.), a odnosi se na postavljanje odgovora na pitanje koje nije vidljivo. Služi da se u slučaju neodgovorenog pitanja mogu izračunati rezultati. Atribut value postavljen je na vrijednost „no_attemp“.

4.3.2. Rezultati ispita s jednostrukim odgovorima

Nakon rješavanja ispita u ovom dijelu pojasnit će se prikaz i logika računanja točnih odgovora. Potrebno je naglasiti da se radi o ispitima s jednostrukim odgovorima pa se računanje točnih odgovora vrši drugačije od ispita s višestrukim odgovorima.

Tvoj rezultat

Ukupni broj pitanja	2
Odgovorena pitanja	1
Točni odgovori	1
Pogrešni odgovori	0
Bez odgovora	1
Tvoj rezultat	50%

Početna

Pogledaj rješenja

Sl. 4.16. Sučelje koje prikazuje rezultat ispita (asnwer.php)

Slika 4.16. prikazuje sučelje namijenjeno prikazu rezultata ispita. Rezultati su predočeni u tablici koja se sastoji od dvaju stupaca i šest redaka. Svaka je vrijednost detaljno definirana i nije ju potrebno detaljnije objašnjavati. Ispod tablice nalaze se dva gumba (button) koja vode do određenih stranica. Gumb „Početna“ vraća korisnika na početnu stranicu (home.php), dok gumb „Pogledaj rješenja“ vodi do stranice (view_ans.php) namijenjene prikazu rješenja ispita, što će biti objašnjeno u daljnjem tekstu.

```

<tr>
  <th>Ukupni broj pitanja</th>
  <th><?php echo $total_qus; ?></th>
</tr>
</thead>
<tbody>
<tr>
  <td>Odgovorena pitanja </td>
  <td><?php echo $atempte_qus; ?></td>
</tr>
<tr>
  <td>Točni odgovori</td>
  <td><?php echo $answer['right']; ?></td>
</tr>
<tr>
  <td>Pogrešni odgovori</td>
  <td><?php echo $answer['wrong']; ?></td>
</tr>
<tr>
  <td>Bez odgovora</td>
  <td><?php echo $answer['no_answer']; ?></td>
</tr>
<tr>
  <td>Ivoj rezultat</td>
  <td><?php
    $per=($answer['right']*100)/ ($total_qus);
    echo $per."%";
  ?></td>
</tr>
</tbody>

```

Sl. 4.17. Kôd prikazivanja rezultata (answer.php)

Slika 4.17. prikazuje kôd koji prikazuje tablicu na prethodnoj slici (sl. 4.16.). Ukupan broj pitanja definiran je varijablom \$total_qus, odgovorena pitanja definirana su varijablom \$atempte_qus, točni odgovori varijablom \$answer['right'], pogrešni odgovori \$answer['wrong'], dok su neodgovorena pitanja smještena u varijablu \$answer ['no_answer']. Rezultat se računao u vrijednosti postotka i vidljiv je na dnu u varijabli \$per. Logika računanja napisana klasom users (users.php).

```

public function answer($data)
{
    $ans= implode("", $data);
    $right=0;
    $wrong=0;
    $no_answer=0;
    $query=$this->conn->query("SELECT id,ans FROM questions WHERE cat_id='".$$_SESSION['cat']."'");
    while($qust=$query->fetch_array(MYSQLI_ASSOC))
    {
        if($qust['ans']==$_POST[$qust['id']])
        {
            $right++;
        }
        elseif($_POST[$qust['id']]=="no_attempt")
        {
            $no_answer++;
        }
        else
        {
            $wrong++;
        }
    }
    $array=array();
    $array['right']=$right;
    $array['wrong']=$wrong;
    $array['no_answer']=$no_answer;
    return $array;
}

```

Activ
Cate

Sl. 4.18. Metoda answer (users.php)

Slika 4.18. prikazuje metodu answer zaduženu za brojanje točnih, pogrešnih odgovora te neodgovorenih pitanja. Metoda prima jedan parametar, u ovom slučaju to je superglobalna varijabla `$_POST`. Korištena je funkcija `implode` čiji je zadatak skupljanja dijelova polja u string (skupina znakova). Definirane su varijable koje predstavljaju točne i pogrešne odgovore te one koji nisu označeni. Dodijeljena im je vrijednost nule da se mogu zbrajati. Unutar varijable `$query` definiran je upit nad tablicom `questions` u kojoj se nalaze pitanja i odgovori, tj. u ovom slučaju potrebni su nam stupci `id` i `ans`. Važno je napomenuti da se upitima dodjeljuje namjena kojom se biraju vrijednosti stupaca `id` i `ans` ondje gdje je `cat_id = $_SESSION['cat']`. Tom namjenom definirano je da se uzmu samo vrijednosti koje se nalaze u odabranom ispitu, odnosno u ispitu koji je rješavan. Vidljiva je prisutnost superglobalne varijable `$_SESSION` jer je njome omogućeno da se pamti koji je ispit odabran u prethodnim datotekama. Svaka od varijabli odgovora ispituje se pomoću naredbe `if` sa smještenim uvjetom koji se izjednačuje s prikupljenim podacima te, u slučaju istine, uveća vrijednost `$right` za jedan. Ako uvjet nije zadovoljen, prelazi se na sljedeći u nizu, tj. `elseif` uvjet kojim se ispituje je li primljena vrijednost jednaka neoznačenom odgovoru koji je u prethodnoj datoteci (`qus_show.php`) definiran. U slučaju istinite tvrdnje varijabla `$no_answer` uvećat će se za vrijednost jedan. Treći uvjet izvršava se automatski u slučaju neistinita prethodna dva te

dodjeljuje vrijednost jedan varijabli \$wrong. Na kraju se definira polje u koje se prebacuje svaka od tih vrijednosti.

4.3.3. Pregled rješenja ispita s jednostrukim odgovorima

Nakon pregleda rezultata poveznica (gumb „Pogledaj rješenja“), koja se nalazila ispod tablice s rezultatima, nije pojašnjena. Ovaj dio potpoglavlja govori o prikazu rezultata za odabrani ispit.

Točni odgovori testa WEB programiranje

1 All variables in PHP start with which symbol?	
1	<input type="radio"/> @
2	<input type="radio"/> %
3	<input checked="" type="radio"/> \$
4	<input type="radio"/> &

2 What does PHP stand for?	
1	<input type="radio"/> PHP: Hypertext Preprocessor
2	<input checked="" type="radio"/> Private Home Page
3	<input type="radio"/> Personal Hypertext Processor
4	<input type="radio"/> None of it

Početna

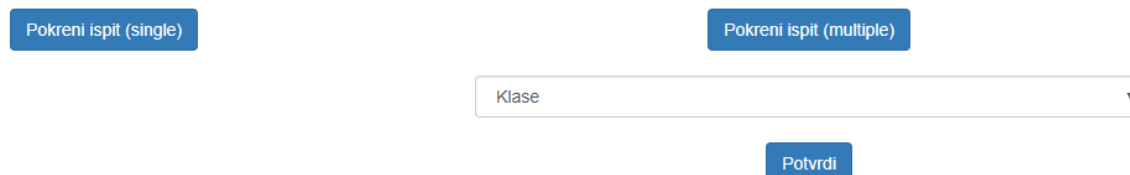
Act
Go t

Sl. 4.19. Sučelje prikaza rezultata (view_ans.php)

Slika 4.19. prikazuje sučelje s prikazom rješenja ispita, tj. točnim odgovorima. Struktura stranice jednaka onoj kod rješavanja ispita. Razlika je u tome što su promijenjene boje pojedinih dijelova u tablici. Točni odgovori obojani su zelenom bojom, dok su ostali netočni odgovori obojani crveno. Na dnu je ostavljena poveznica (gumb „Početna“) koja vodi do početne stranice aplikacije.

it"
aberi ispit koji želiš riješavati
iknite na button "Pokreni"

leno vrijeme u kojem morate riješiti ispit. Timer se nalazi desno gore u kutu kada pokrenete ispit. SRETNO



Pokreni ispit (single) Pokreni ispit (multiple)

Klase

Potvrdi

SI. 4.21. Sučelje početne stranice i pokretanje ispita s višestrukim odgovorima (home.php)

Slika 4.21. prikazuje sučelje početne stranice nakon prijave u aplikaciju. U ovom dijelu pokrenut će se ispit vezan uz višestruke odgovore. Trenutno je odabran ispit „Klase“ koji se pokreće pritiskom na gumb „Potvrdi“. Kôd kojim je napisan ovaj dio sličan je onom s jednostrukim odgovorima. Jedina je razlika uzimanje podataka iz druge tablice. Za ispis ponuđenih ispita korištena je tablica `category_multi` u kojoj su smješteni svi ispiti s višestrukim odgovorima.

1. WEB programiranje se veće uz koje ina?ice?

- JS
- php
- IOS
- Android
- Symbian

2. Broj 6 dijeljiv je sa? (ostatak mora biti cijelobrojan)

- 5
- 4
- 2
- 1
- 0

Potvrdi

Sl. 4.22. Sučelje pri pokretanju ispita (multi_ans.php)

Slika 4.22. prikazuje sučelje namijenjeno rješavanju ispita. Izgled sučelja sličan je onome s jednostrukim odgovorima, no kod višestrukih odgovora dodan je jedan odgovor više i nema više opsijskih gumbova (radio button), nego su dodani potvrdni okviri (engl. checkbox) koji omogućuje korisniku višestruki izbor za navedeni sadržaj. O tome će se više govoriti u dijelu pojašnjenja kôda. Prisutan je brojač koji nakon isteka zatvara ispit i vodi na sljedeću stranicu. Klikom na gumb „Potvrdite“ automatski se zaključuje rješavanje ispita i prebacuje na sljedeću stranicu, odnosno pregled rješenja.

```

<form action="multi_ans_post.php" id="form2" method="post">
<br>
<table class="table table-bordered">
<thead>
<tr class="danger">
<th><?php echo $i;?>.&nbsp;<?php echo $qستا['question'];?></th>
</tr>
</thead>
<tbody>

<tr class="info">
<td><input type="checkbox" name="<?php echo $qستا['id'];?>[]" value="1"><?php echo $qستا ['ans1'];?></td>
</tr>
<tr class="info">
<td><input type="checkbox" name="<?php echo $qستا['id'];?>[]" value="2"><?php echo $qستا ['ans2'];?></td>
</tr>
<tr class="info">
<td><input type="checkbox" name="<?php echo $qستا['id'];?>[]" value="3"><?php echo $qستا ['ans3'];?></td>
</tr>
<tr class="info">
<td><input type="checkbox" name="<?php echo $qستا['id'];?>[]" value="4"><?php echo $qستا ['ans4'];?></td>
</tr>
<tr class="info">
<td><input type="checkbox" name="<?php echo $qستا['id'];?>[]" value="5"><?php echo $qستا ['ans5'];?></td>
</tr>
</tbody>
</table>

```

Sl. 4.23. Kôd prikazivanja pitanja (multi_ans.php)

Slika 4.23. prikazuje kôd koji služi prikazivanju pitanja za višestruke odgovore. Koncept ispisa sličan je kao i u slučaju s jednostrukim odgovorima, razlika je ta što su dohvaćeni drugi podatci, odnosno referencira se na drugu tablicu u bazi podataka (tablica questions1). Dodani su potvrdni okviri (engl. checkbox) i izmijenjeni atributi value u ispisu odgovora. Za razliku od jednostrukih odgovora, u ovom slučaju prvi element ima oznaku 1 do 5. Sve se obavlja formom koja je definirana metodom POST i nakon zaključivanja prosljeđuje podatke sljedećoj datoteci multi_ans_post.php namijenjenoj za prikazivanje rezultata.

4.3.5. Rezultati ispita s višestrukim odgovorima

Nakon rješavanja ispita spremljeni podatci prosljeđuju se datoteci zaduženoj za prikazivanje i računanje rezultata. Datoteka o kojoj se govori ima naziv multi_ans_post.php.

```

$sql = "SELECT * FROM questions1 WHERE cat_id='".$$_SESSION['cat_m']."'";
$result = $connect->query($sql);
$answers=array();
if($result->num_rows > 0)
{
    while($row = $result->fetch_assoc())
    {
        $answers[$row['id']] = array($row['ans'], $row['anss']);
    }
}

```

Sl. 4.24. Kôd za spremanje točnih odgovora (multi_ans_post.php)

Slika 4.24. prikazuje kôd koji ima zadatak spremi sve točne odgovore u polje \$answers. Podatci su dohvaćeni iz tablice question1. U polje se spremaju tako da se prvo upiše id pitanja, a onda se u id dodaju točni odgovori (stupci ans i anss).

```

if (isset($_POST)){
    $totalCorrect = 0;
    $count = 0;
    foreach ($answers as $ans) {$count+= count($ans);}
    foreach ($answers as $row['id'] => $answer){
        $question = $row['id'];
        if(is_array($answer) && isset($_POST[$question])){
            //traži odgovore
            $matches = array_intersect($answer, $_POST[$question]);
            $good_answers = count($matches);
            //Brojanje promaĽaja
            $bad_answers = 0;
            foreach($_POST[$question] as $post_answer){
                if(!in_array($post_answer, $answer)){
                    $bad_answers++;
                }
            }
            //Oduzimanje dobrih i promaĽaja(da se ne bi stvorio minus)
            if($good_answers > $bad_answers){
                $result = $good_answers - $bad_answers;
            }else{
                $result = 0;
            }
            $totalCorrect = $totalCorrect + $result;
        }elseif(isset($_POST[$question]) && strtolower($_POST[$question]) === strtolower($answer)){
            $totalCorrect++;
        }
    }
    $pct = round( (($totalCorrect/$count) * 100), 0);
}

```

Sl. 4.25. Kôd za računanje rezultata (multi_ans_post.php)

Slika 4.25. prikazuje kôd zadužen za računanje rezultata. Na vrhu se provjerava radi li se s prosljeđenim podacima iz prethodne forme. Ukupan broj točnih odgovora (\$totalCorrect) postavlja se na 0, a zajedno s tim i brojač (count) u svrhu prebrojavanja točnih odgovora u polju \$answers. Potrebno je prebrojati točne odgovore da bi se mogao izračunati konačan rezultat ispita budući da pitanje može imati jedan ili dva točna odgovora. Nakon prebrojavanja odgovora foreach petljom sljedećom foreach petljom pristupa se svim id elementima unutar polja \$answers da bi se znalo na kojem se pitanju izvršava provjera odgovora. Svaki element id-a zamjenjuje se varijablom \$questions, a zatim se logičkim operatorom AND (&&) ispituje uvjet u zagradama, kôd se izvršava ako je zadovoljen uvjet i traže se točni odgovori koji se pribrajaju u varijablu \$good_answers. Funkcija koja pomaže pri traženju točnih odgovora je array_intersect. Ona uspoređuje dva ili više polja te vraća njihove jednake elemente, u ovom slučaju točne odgovore ako se preklapaju.[7] Nakon

računanja točnih odgovora potrebno je izračunati pogrešne odgovore. Varijabla `$bad_answers` predstavlja pogrešne odgovore i postavlja se na nultu vrijednost. Za svaki odabrani odgovor koji je različit od točnog odgovora varijabla `$bad_answers` uvećat će se za 1. Da se ne bi stvorio minus u rezultatu, potrebno je ispraviti logiku računanja rezultata. Zato se ispituje uvjet: ako je broj točnih odgovora veći od broja netočnih, izvršava se računanje varijablom `$result` koja uzima razliku između dobrih i loših odgovora (`$good_answers - $bad_answers`). U suprotnom, odnosno u slučaju da je broj pogrešnih odgovora veći od broja točnih odgovora, rezultat će se postaviti na 0 čime će se izbjeći negativan rezultat ispita. Na samom kraju izvršava se računanje ukupnog rezultata koji se računa tako da se početna varijabla `$totalCorrect` zbroji sa samom sobom i varijablom `$result`. Zadnji dio kôda izvršava ispis postotka definiranom funkcijom `round` koja zaokružuje rezultat na bližu decimalu.

Tvoj rezultat (100%)

Broj ukupnih točnih odgovora u ispitu	4
Rezultat ispita (ako se označi krivi odgovor oduzimaju se bodovi)	4
Tvoji odgovori	1. Pitanje:Odabrao si 2 odgovor/a: 1 2 2. Pitanje:Odabrao si 2 odgovor/a: 3 4

Početna

Pogledaj rješenja

SI. 4.26. Sučelje namijenjeno prikazu rezultata višestrukih odgovora (`multu_ans_post.php`)

Slika 4.26. prikazuje tablicu s rezultatima vezanim za riješeni ispit. Prikazuje broj točnih odgovora u cjelokupnom ispitu i rezultat u slučaju označavanja pogrešnih odgovora. Uz sve to, u zadnjoj rubrici prikazani su odgovori koje je korisnik označio. Gumbovi ispod tablice vode do početne stranice („Početna“) i pregleda rješenja ispita („Pogledaj rješenja“).

4.3.6. Pregled rješenja ispita s višestrukim odgovorima

Koncept pregleda rješenja zasnovan je na istom principu kao i ispiti s jednostrukim odgovorima. Budući da ispiti s višestrukim odgovorima imaju stupac više (`anss`), u tablici (`questions`) potrebno je dodati još jedan uvjet pri ispisu točnih odgovora.

Točni odgovori testa Klase

1 WEB programiranje se veće uz koje ina?ice?	
1	<input type="checkbox"/> JS
2	<input type="checkbox"/> php
3	<input type="checkbox"/> IOS
4	<input type="checkbox"/> Android
5	<input type="checkbox"/> Symbian

2 Broj 6 dijeljiv je sa? (ostatak mora biti cijelobrojan)	
1	<input type="checkbox"/> 5
2	<input type="checkbox"/> 4
3	<input type="checkbox"/> 2
4	<input type="checkbox"/> 1
5	<input type="checkbox"/> 0

Početna

SI. 4.28. Sučelje za ispis rješenja ispita (view_ans_multi.php)

Slika 4.28. prikazuje sučelje koje predstavlja prethodno objašnjen kôd. Pogrešni odgovori označeni su crvenom bojom, dok su točni odgovori označeni zelenom. Na dnu se nalazi gumb („Početna“) koji vodi do početne stranice aplikacije prilikom prijave (home.php).

4.3.7. Brojač jednostrukih i višestrukih odgovora

U ovom potpoglavlju bit će kratko objašnjena kreacija brojača (timera). Ovaj dio napravljen je pomoću JavaScript kôda i vezan je za ispite s jednostrukim i višestrukim odgovorima.

```

<script type="text/javascript">
function timeout()
{
    var minute=Math.floor(timeLeft/60);
    var second=timeLeft%60;

    var mint=checktime(minute);
    var sec=checktime(second);
    if(timeLeft<=0)
    {
        clearTimeout(tm);
        document.getElementById("form1").submit();
    }
    else
    {
        document.getElementById("time").innerHTML=mint+":"+sec;
    }
    timeLeft--;
    var tm= setTimeout(function(){timeout()},1000);
}
function checktime(msg)
{
    if(msg<10)
    {
        msg="0"+msg;
    }
    return msg;
}
</script>

```

Sl. 4.28. Kôd brojača (multi_ans.php , qus_show.php)

Slika 4.28. prikazuje kôd kojim se definira kreacija brojača. Da bi bilo moguće koristiti JavaScript kôd, potrebno ga je ograničiti, odnosno otvoriti oznake u kojima će se pisati. Oznaka koja služi za to je <script>. U oznakama su definirane varijable vezane za minute i sekunde. Korištena je funkcija Math.floor koja se koristi kod deklaracije minuta. Ta funkcija zaokružuje broj na niže do najbližeg cijelog broja. Nakon deklaracije sekunda i minuta koristi se if petlja za provjeru isteka vremena. U slučaju isteka zaključuje formu kojom je definiran brojač. Ako prvi uvjet nije istinit, izvršava se sljedeći, u ovom slučaju else uvjet koji vraća sadržaj vremena. Zatim slijedi umanjivanje varijable timeLeft za 1, odnosno umanjivanje za jednu sekundu. U zadnjem dijelu funkcije timeout() definirana je varijabla tm koja koristi metodu setTimeout koja poziva funkciju timeout() i definira ju na 1000 ms, odnosno na jednu sekundu. Druga funkcija checktime korištena je u prvoj funkciji, a služi za izmjenu sekunda i minuta, bez nje ne bi bio moguć prikaz dinamične promjene vremena.

5. ZALJUČAK

Ovaj rad omogućuje detaljniji prikaz razvijanja web-aplikacije Online ispiti. Rad se zasniva na pregledu kôda podijeljenom na nekoliko cjelina. Zadatak je rada i izrade web-aplikacije omogućiti jednostavnije i pristupačnije rješavanje ispita, a izrada aplikacije razdvojena je na dvije cjeline: administratorsku koja se odnosi na učitelje/profesore i korisničku koja se odnosi na učenike/studente. Kao prednosti navedene aplikacije ističu se jednostavnije i brže kreiranje ispita te automatsko ispravljanje nakon rješavanja što pridonosi uštedi vremena koje ono oduzima. Ispite nije potrebno printati i rješivi su na bilo kojem mjestu, naravno uz mogućnost povezivanja Internetom. Zadnja navedena prednost može se razmotriti kao i mana jer ako korisnik nema omogućen pristup Internetu, nije u mogućnosti pristupiti ispitu. Kao mana aplikacije može se istaknuti i ograničenje u pristupu. Primjerice, ako se ispit rješava u vlastitom domu, nije moguće kontrolirati korisnike koji mogu koristiti dodatna sredstva za pomoć pri njihovu rješavanju. Najbolja implementacija ove aplikacije bila bi u određenom prostoru gdje profesor ima mogućnost nadgledanja. Aplikacija ima funkcionalnost i svrhu, no uvijek se može nadograditi i poboljšati pa ju treba promotriti i na taj način.

SAŽETAK

Izrada web-aplikacije Online ispiti podijeljena je na pet poglavlja. U uvodu je predstavljen zadatak koji se treba odraditi, njezina funkcionalnost te predodređeni koncept i struktura. Sljedeće poglavlje definira značenje web-aplikacije i tehnologije kojima je moguće izraditi. Navedeni su alati i tehnologije korištene u radu. Treće poglavlje predstavljeno je kao administracijski dio, odnosno dio namijenjen profesorima/učiteljima, detaljno je prikazan kôd i logika navedene cjeline. Nakon administracijskog dijela najveći dio kreiranja aplikacije posvećen je korisničkom dijelu. U tom poglavlju prikazan je i opisan kôd, ujedno i popraćen slikama. U zaključku su objedinjene prednosti i mane ove aplikacije i naglašena mogućnost njezine nadogradnje.

Ključne riječi: MySQL, online ispiti, php, prikaz rezultata, web aplikacija

ABSTRACT

WEB APPLICATION : Online Exam

Building the web application Online Exams is divided into five chapters. In the introduction, the task which requires performing is presented, along with its functionality, concept and structure. The next chapter defines the meaning of the web application and technologies which may be utilized to build it. Furthermore, there is a list of tools and technologies used during the work. The third part represents an administrative part, i.e. the part intended for professors/students, with a detailed representation of the code and logics of the relevant unit. After the administrative part, the largest part of application creation is reserved for the users. In that chapter, the code is represented and described, accompanied with explained figures. The conclusion encompasses the advantages and disadvantages of the subject application and emphasizes the possibility of its upgrade.

Key words: MySQL, online exam, php, view result, web-application

LITERATURA

- [1] Web aplikacije - https://www.it-b.hr/str/izrada_web_aplikacija/ [5.12.2017.]
- [2] Uvod u HTML - <https://tesla.carnet.hr/mod/book/view.php?id=5430&chapterid=885> [5.12.2017.]
- [3] Ivezic, Jovan, BOOTSTRAP TUTORIJAL – GRID SISTEM - https://www.popwebdesign.net/popart_blog/2014/01/bootstrap-tutorijal-grid-sistem/# [5. 12. 2017.]
- [4] Uvod u JavaScript: JavaScript Introduction - <https://www.mojwebdizajn.net/web-programiranje/vodic/javascript/uvod-u-javascript.aspx> [5. 12. 2017.]
- [5] Mujadžević, Edin, Uvod u PHP i MySQL - http://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/d350_polaznik.pdf [5. 12. 2017.]
- [6] Što to znači responzivan dizajn? - <http://www.digitalnitureizam.hr/hr/sto-to-znaci-responzivan-dizajn/d19/> [5. 12. 2017.]
- [7] array_intersect - <http://php.net/manual/en/function.array-intersect.php> [5.12.2017]
- [8] Rasmus Leardorf, Kevin Tatroe i Peter MacIntyre – Programiranje PHP

ŽIVOTOPIS

Domagoj Ćurić rođen je 20. ožujka. 1993. u Đakovu gdje je pohađao i osnovnu školu OŠ „Ivan Goran Kovačić“. Nakon osnovne škole upisuje se u Elektrotehničku i prometnu školu u Osijek, smjer Tehničar za računalstvo nakon čega pohađa Elektrotehnički fakultet u Osijeku, smjer Informatika (stručni studij). Od stranih jezika služi se engleskim jezikom.

PRILOZI

Na optičkom disku priloženom uz završni rad nalazi se .doc i .pdf verzija završnog rada te kod web aplikacije. Kod aplikacije je raspoređen po mapama, ovisno o njegovoj funkciji, te su sve datoteke nazvane originalno kao i u dokumentaciji.