

**УНИВЕРЗИТЕТ У ПРИШТИНИ**  
**ПРИРОДНО МАТЕМАТИЧКИ ФАУЛТЕТ**

Одсек за информатику



**Е-КВИЗ**

Пројектни задатак: Електронска управа

Ментор:

проф. др. Стефан Панић  
Милан Дејановић, асистент

Кандидат:

Јован Б. Исаиловић, студент

*Косовска Митровица, јануар 2022. године*

## САДРЖАЈ

УВОД.....	3
Добијени задатак.....	3
Додатна реализација.....	3
ОПИС КОРИШЋЕНИХ ТЕХНОЛОГИЈА.....	5
Шта су HTML, CSS и JS?.....	5
Шта је Bootstrap?.....	6
Шта је PHP?.....	6
Шта је MySQL?.....	6
Шта је Apache?.....	6
Шта је MariaDB?.....	7
Шта је phpMyAdmin?.....	7
Шта је XAMPP?.....	7
ОПИС БАЗЕ ПОДАТАКА.....	8
ОПИС КОРИШЋЕНИХ НАРЕДБИ И ФУНКЦИЈА.....	9
ЗАКЉУЧАК.....	13
ЛИТЕРАТУРА.....	14

## УВОД

### Добијени задатак

"Направити е-квиз модерног и респонзивног веб дизајна (на српском језику). Креирати форму за регистрацију корисника. Након регистрације и логовања корисника, корисник има право да решава квиз. Додати почетну страну са информацијама о квизу. Убацити 60 питања у базу података везаних за област Информатике. Одговори мог бити изабрани преко radio button-а, checkbox-ова и текстуалних поља. Случајним одабиром приказати 10 питања и након свих одговора, кликом на дугме, приказати остварени број поена. Остварени број поена се памти у бази података за сваког учесника. Један учесник може неограничени број пута да решава квиз. Креирати и посебну страницу „Leaderboard“ где ће се приказивати најуспешнији такмичари од креирања квиза."

Свака ставка из овог задатка је испоштована, а неке су чак вишеструко унапређене

### Додатна реализација

Као што је претходно поменуто, већина функција и својства овог пројекта јесте самостално додата, па тако следеће тачке и компоненте су додате:

- Додат је систем администратора и суперадминистратора (који се одређује својим корисничким именом, у мом случају "ice"), наиме суперадмин може додавати друге администраторе
- Додат је админ панел, где се могу видети сви корисници, све додате категорије квиза, као и сва додата питања
- Корисник може да гледа своје резултате квиза и да те резултате упоређује са својим просечним вредностима (нпр. просечно време да се заврши квиз, просечан резултат итд.)
- Корисник може додати категорију за коју мисли да би требала да се појави на сајту, коју касније администратор може да одобри
- Корисник, такође, може додавати питања на категорије које већ постоје, али и на оне које су још увек у разматрању

- Додат је систем корисничких профила и систем подешавања истог, на страници подешавања профила може се додати и опис профила, као и променити име, презиме, лозинка итд.
- На профилима додат је просечан резултат сваког корисника - колико су питања додали и колико су квизова одиграли, као и датум регистравања, датум последњег логовања и опис написан од стране власника профила
- Додат је систем достигнућа (уколико корисник одигра, на пример, 5 квизова са по бар 5 тачних одговора добиће одређено достигнуће)
- Додат је систем претраживања корисника по њиховом имену, презимену или корисничком имену (где је најпрецизнија претрага по корисничком имену)
- Додат је респонзивни навигациони елемент, као и заглавље, који се само укључују у датотеку позивањем рђр функције (на овај начин, HTML код изгледа колико-толико читкији)

## **ОПИС КОРИШЋЕНИХ ТЕХНОЛОГИЈА**

Овај пројекат рађен је у "бекенд" програмском језику PHP v8.1.0, за HTTP сервер коришћен је Apache v2.4.54, за веб сервер коришћен је XAMPP v8.1.0, док је за базу података коришћен MySQL, а њен сервер је v10.4.25-MariaDB, као алатка за администрацију MySQL и MariaDB коришћен је phpMyAdmin v5.2.0.

Што се "фронтенд"-а тиче, коришћени су, наравно HTML 5, CSS 3, Javascript, као и библиотека (library) ових језика - Bootstrap v5.1.3

### **Шта су HTML, CSS и JS?**

HyperText Markup Language или, скраћено, HTML, јесте стандардни маркап језик за документе који су предвиђени да буду приказани у веб претраживачу. Он дефинише структуру веб странице, при томе, написани HTML се састоји из низа елемената, може се рећи да HTML говори претраживачу како да прикаже садржај [W3schools, 2022]. Први пут се појавио 1993, а најновија верзија, HTML5, се појавила 2014, која је значајно унапредила веб технологије и од тад се све више интегришу Javascript и HTML [Pedamkar (2022)].

Cascading Style Sheets или, како га већина људи познаје, CSS, јесте језик који се користи за описивање презентације веб странице, то јест, како ће се једна страница презентовати (приказати) ономе ко је гледа. Дизајниран је тако да омогућује поделу и одвајање садржаја, и у то се убрајају боје, позиција, фонтови и тако даље[W3 (2010)]. CSS је врло лак за разумевање, јер је његова комплетна синтакса на људском (енглеском) језику. Такође, респонзивност се дефинише управо у CSS-у, тј. могуће је исту веб страницу приказати на уређајима различитих димензија и резолуција коришћењем одговарајућег кода.

Javascript, јесте језик који је у последњих 5 година доживео највећу експанзију, у оригиналу Javascript је био језик који се користио за скриптовање и давање "живота" веб страницама, данас он има примену у готово свим сферама технологије, могуће је правити и апликације на телефону коришћењем Javascript-а [Reactnative (2022)], многе средине ван претраживача користе управо Javascript, то су Node.js, Apache CouchDB и многи други [MDN (2022)]

Bootstrap јесте бесплатна, "open-source" библиотека која се користи за креирање веб сајтова и веб апликација. Ова библиотека је дизајнирана да стави мобилну респонзивност на прво место [W3Schools (2022)]. Bootstrap даје велику колекцију синтакских оквира за дизајнирање. Најновија верзија Bootstrap-а је 5.

## **Шта је PHP?**

Значење скраћенице PHP је првобитно значило "Personal Home Page", али, касније је промењено у рекурзивну скраћеницу "PHP: Hypertext Preprocessor". PHP је настао 1994, а прве доступне верзије појавиле су се 1995 [ifj.edu.pl (1999)]. Његова примена је највише заступљена у развијању веб сајтова и апликација, због његове могућности да се потпуно интегрише у HTML. Javascript се извршава на рачунару корисника, док оно што је специфично за PHP јесте да се он извршава на серверу [php.net (2022)].

## **Шта је MySQL?**

MySQL је један од најпознатијих технологија у менаџменту података. MySQL је релациони систем менаџмента база података који је развио Oracle и који је базиран на SQL (Structured Query Language) . MySQL је важан део скоро сваког популарног софтвера, јер се користи за креирање апликација које могу да користе купци, или неки системи где је неопходно сачувати неке податке (свака регистрација и пријава користи неку врсту базе података) [talend (2023)]. Компаније које користе MySQL су NASA, CERN, многобројне банке, Youtube, Wikipedia, Twitter итд. [MySQL (2023)]

## **Шта је Apache?**

Apache јесте најкоришћенији софтвер за веб сервер и он покреће око 67% веб сајтова на свету. Развијен од стране Apache Software Foundation и он је "open-source" софтвер који је, наравно, бесплатан. Apache је брз, сигуран и поуздан, такође, он је врло прилагодљив, па је због тога његова примена врло широка (тј. може се прилагодити према нечијим потребама) [wpbeginner (2022)].

MariaDB је релациони систем менаџмента база података, један од најпопуларнијих система сервера база података, чији су познати корисници Wikipedia, Wordpress и Google. MariaDB је настала 2009. када је MySQL купљен од стране Oracle-а и од тада је добио више ажурирања од самог MySQL-а, тренутно је на верзији 10.9, док је MySQL на 5.6 [MariaDB (2023)].

## **Шта је phpMyAdmin?**

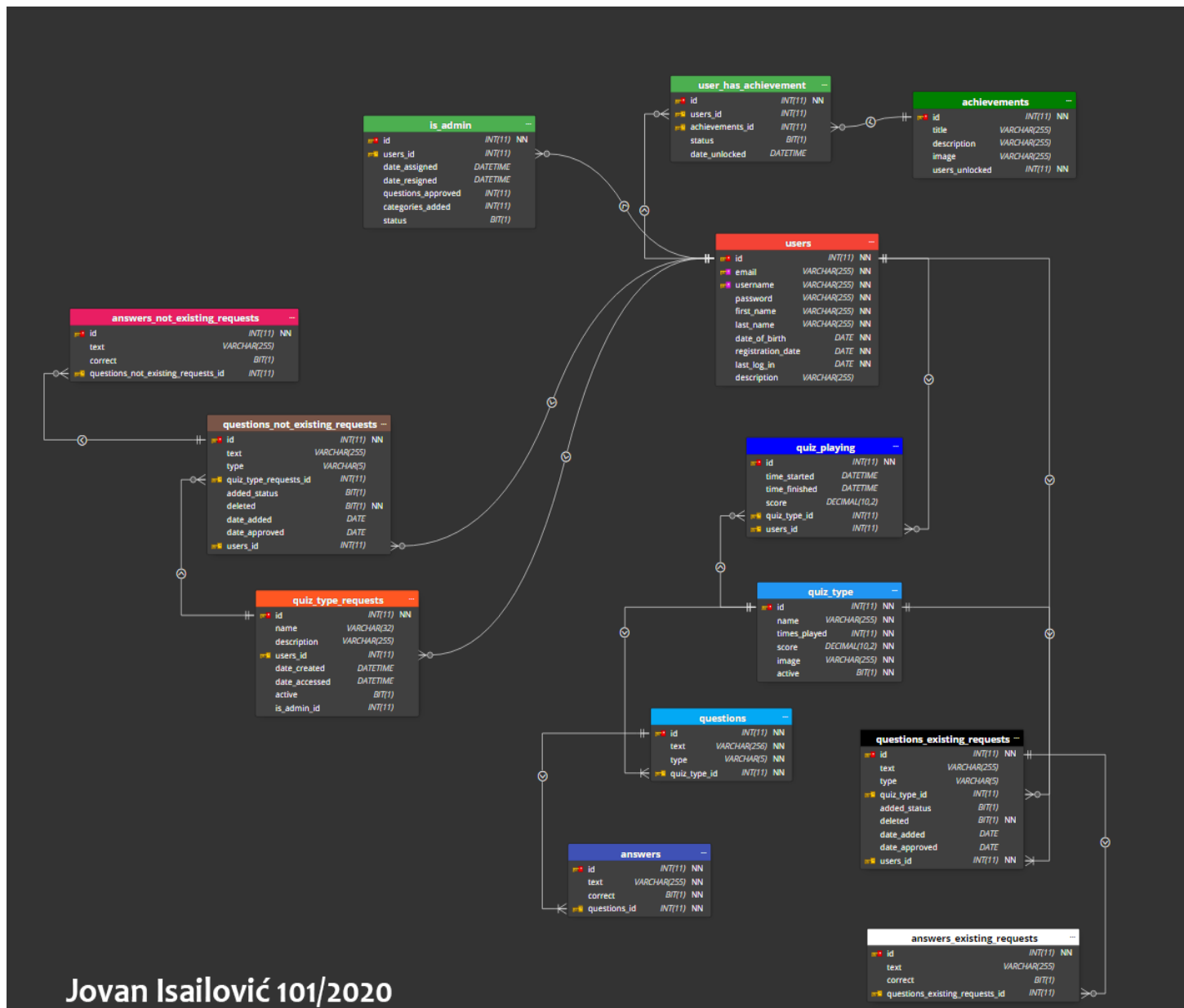
phpMyAdmin јесте бесплатна алатка која је написана у PHP-у и она је направљена са намером да се носи са администрацијом MySQL-а над веб-ом. phpMyAdmin подржава широк спектар операција над MySQL-ом и MariaDB-ом. Простије, као и оне операције које се често користе, могу се извршити преко корисничког интерфејса, међу њима спадају менаџмент базе података, креирање, мењање и брисање табела, колона, веза, индекса, корисника, пермисија итд. [phpMyAdmin (2023)]

## **Шта је XAMPP?**

XAMPP или Cross-Platform, Apache, MySQL, PHP and Perl, јесте пакет који омогућава испоруку Apache-а за различите сервере и покретаче програма са командних линија, такође, омогућава испоруку Apache API, MariaDB, PHP и Perl модула. Cross-Platform значи да XAMPP ради на више уређаја, без обзира на њихов оперативни систем. XAMPP је изузетно користан јер се, пре избацивања апликације на тржиште или у рад, може тестирати локално, на серверу саме машине (localhost). [K. Ashwani (2021)] Оно што је важно напоменути, јесте да XAMPP долази са инсталираним phpMyAdmin модулом. Избачен је у рад 2002. године, али је и даље актуелан и одржаван.

## ОПИС БАЗЕ ПОДАТАКА

У бази података "квиз" постоји 13 табела. Изглед ЕР дијаграма је:



Jovan Isailović 101/2020

Слика 1. ЕР Дијаграм



## **ОПИС КОРИШЋЕНИХ НАРЕДБИ И ФУНКЦИЈА**

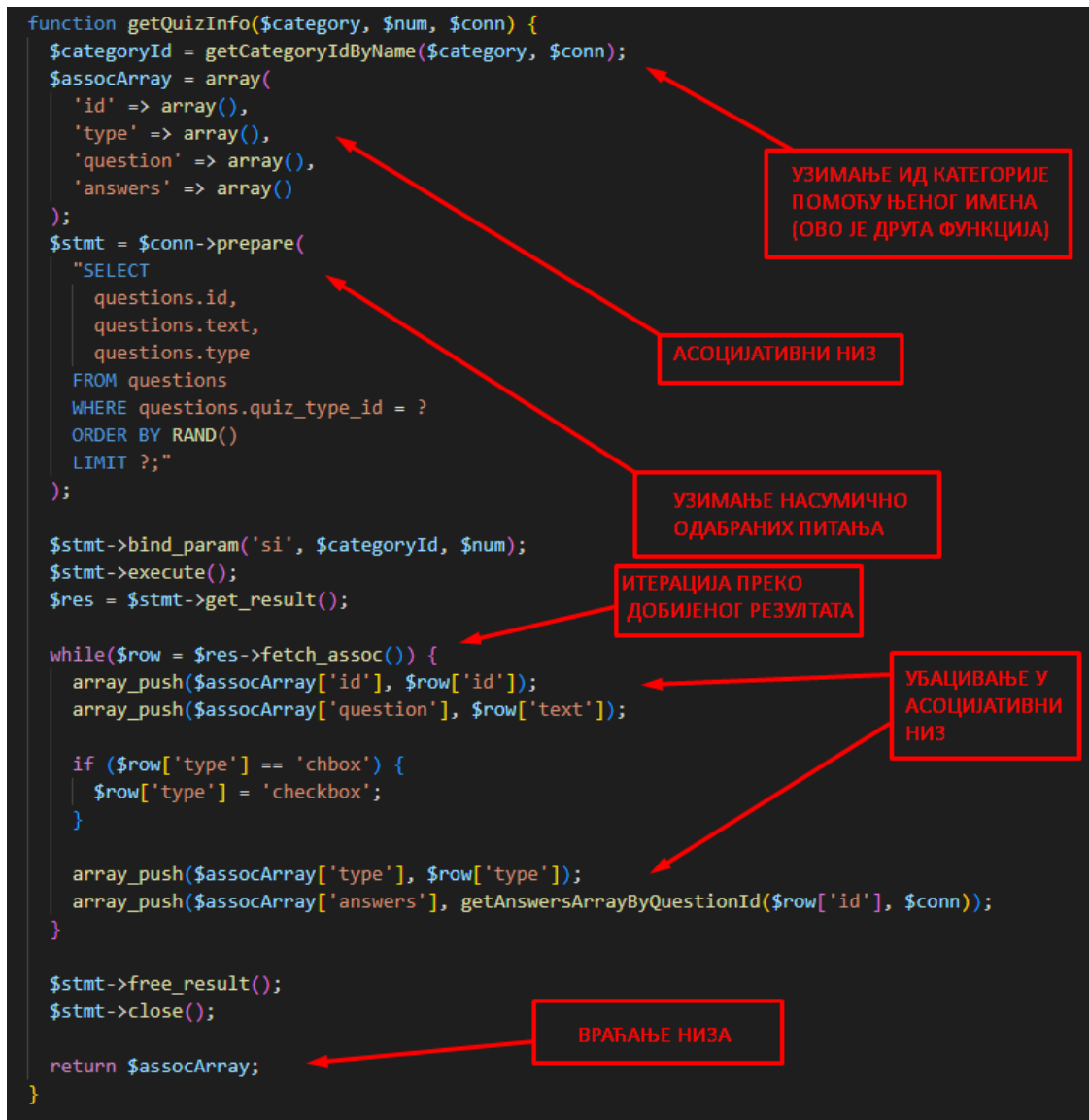
Као и за опис функционалности целокупне апликације и приликом описа коришћених наредби и функција има превише ставки које би морале да се пређу у циљу описивања сваке функције.

Главни разлог за то јесте сама хијерархија пројекта. Наиме, свака страница има своју датотеку са функцијама које она користи. Па тако, на пример, за пријављивање постоји страница `login.php` и `login_functions.php` у којој су исписане све потребне функције за пријављивање корисника.

Предност оваквог начина рада јесте смањивање кода у делу где треба да влада доминантност HTML кода у односу на 100, 200 или, чак, 300 линија PHP кода. На овај начин време за дебаговање и проналажење проблема, као и решавање истих се знатно скраћује и једноставно може се посебно радити на фронтенд, а посебно на бекенд функцијама. Преплитања, наравно, има, поготово када је потребно исписати неку информацију која је узета из базе.

У случају овог пројекта, узимање сваког податка се складиштило у (обично) асоцијативном низу (array-и), који је имао различите кључеве, а ти кључеви су често и сами били низови. Разлог оваквог рада јесте што је понекад било потребно исписати исте податке из више редова, на пример, сваки корисник има своје корисничко име, па тако први корисник који је узет из базе иде у "корисничко име" кључ неког асоцијативног низа, друго корисничко име иде на друго место итд., исти принцип јесте за, рецимо, име или било који други податак (сви подаци са индексом  $n$  су од  $(n - 1)$ -тог корисника).

Ово делује јако конфузно кад се напише, али пошто ће се овде објаснити начини функционисања најбитнијих функција, на следећем примеру би требало да се разјасни:



Слика 17. Изглед функције генерисања питања за квиз

Овом функцијом, ми на сигурни начин као резултат добијамо један асоцијативни низ који је \$category категорије и има \$num питања.

У quiz.php делу кода ова функција се позива на овај начин:

```
$quizType = $_GET['type'];
$quizQuestionNumber = 10;
$quizInfo = getQuizInfo($quizType, $quizQuestionNumber, $conn);
```

Да бисмо приказали наших, рецимо, 10 питања категорије "Информатика", ми ћемо урадити једну `for()` петљу преко нашег `$quizInfo` асоцијативног низа, наравно, када одређујемо број итерација, ми ћемо узети неки податак из његовог кључа и онда узети његову дужину, разлог овоме је што ако имамо 4 параметра (ид, текст...) дужина ће нам бити 4, а имамо 10 питања, тако да ми морамо да проверимо колико ид-ева, у ствари, има и преко њих одредити број наших итерација потребних за прелажење преко цилог низа.

На овај начин, урађен је сваки приказ података било које дужине, јер помоћу асоцијативног низа много је прегледније видети о чему се ради, чак и када је у питању асоцијативни низ чији кључеви нису низови, него нека одређена вредност.

Главни део кода јесте свакако оцењивање квиза и да ли је корисник изабрао (или унео) тачан одговор. Квиз страница, у ствари, не садржи никакву логику иза оцењивања, већ само има функције за узимање питања из базе података. Пример како смо узели питања из базе јесте *Слика 17*, али поред тога, ова страница има и задатак да убаци почетно време играња квиза у базу података, име корисника и која категорија квиза је започета. Затим, прави се сесија која садржи сва изабрана питања (без информација који одговор је тачан) и одабране одговоре.

Одговори се бележе помоћу њиховог ид-а, што је погодно, јер корисник не може да убаци још одговора и поремети код. Када се притисне дугме за завршетак квиза, корисник се шаље на `quiz_submit.php` која је, у ствари, празна страница и она служи само да затвори сесију и провери који су одговори тачни управо на основу њиховог ид-а (за питања које корисник уноси бришу се сви размаци и специјални знакови осим заграда ради боље провере). Када се затвори сесија и добије колико је тачних одговора поклопљено у базу се уписује време кликтања дугмета и колико је тачних одговора имао, затим корисник се шаље на `result.php?id=x`, где је `x` ID његовог задњег покушаја.

Сада се поставља питање ко може да види овај резултат, коришћењем сесије можемо утврдити да ли је корисник који посећује резултат са неким ID-ем, у ствари, корисник који је играо квиз са тим ID-ем, уколико није - он ће бити враћен на почетну страницу. Ово не важи за администратора, који приликом пријављивања на сајт добија специјалну сесију која даје тачно или нетачно (`$_SESSION['admin']`). На овај начин (провером сесије и који корисник је улогован) свака одредница пермисије функционише, тако се одређује да ли ће се админ панел приказати у навбару, да ли ће корисник бити враћен на "мој профил" ако покуша да посети профил са својим корисничким именом, да ли корисник уопште може приступити админ панелу, или, на пример, да ли корисник уопште може да приступи игрању квиза (јер корисник мора бити улогован да би то радио).

Овим је објашњено основно функционисање целокупне апликације, јер све ради на принципу асоцијативних низова, проверавање неких услова и упоређивањем тренутне сесије.

Наравно, исто важи и за РНР наредбе, доста их је коришћено осим основних петљи неке од њих су:

- `$stmt = $conn->prepare([QUERY])` (где је `$conn` `mysqli` конекција) *// припрема одређени SQL [QUERY]*
- `$stmt->bind_param('type', $variable)` *// ставља да су све \$variable променљиве одређеног 'type' типа, на пример, ако хоћемо да печатиремо да је наша променљива \$brojGodina цели број написали бисмо \$stmt->bind\_param('i', \$brojGodina)*
- `$stmt->execute()` *// извршава SQL [QUERY] који је печатiran вредностима у bind\_param*
- `$stmt->store_result()` *// складишти резултате SQL [QUERY]-а, који се касније могу пребројати на овај начин: \$stmt->num\_rows()*
- `$row = $stmt->get_result()` *// складишти све редове резултата SQL [QUERY] у променљиву \$row, који се касније, појединачно, могу приказати коришћењем неке петље*
- `sizeof($array)` *// величина нiza \$arraz*
- `array_push($array, $variable)` *// убацивање променљиве \$variable у низ \$arraz*
- `foreach($array as $value)` *// итерација преко нiza \$array где је свака вредност у низу означена као \$value*
- `explode($separator, $string)` *// раздвајање nekog \$string-a у array по nekom karakteru, на пример: explode('\_', 'string\_1') => array('string', '1')*
- `strlen($string)` *// враћа дужину nekog stringa \$string*

## **ЗАКЉУЧАК**

Било је доста сметњи приликом рада на овом пројекту, највећи проблем јесте, дефинитивно био улажење у размишљање неког ко хоће да вара на квизу, покуша да на било који начин избегне лош резултат или да на неки начин покуша да се дочепа позиције администратора или на било који начин наштети сајту.

Наравно, овај пројекат сигурно није 100% отпоран на то, наиме, није уведен систем бановања некога ко спамаје питања и тако уништава прегледност неког ко би их одобравао, такође због лошег написаног SQL-а и мог првог рада у њему, није успела првобитна замисао да се омогући додавање нове категорије јер се мора мигрирати велика количина података из једне табеле у другу, јер морају се сва питања пребацити заједно са категоријом у "постојеће категорије", а са питањима морају се убацити онда и одговори, ово ако се ради док је клијент, тј. админ улогован, трајаће вероватно дуго (када бисмо радили са хиљадама питања), али, у суштини, сматрам да је квиз функционалан.

Поред ових бекенд проблема, велика препрека била је направити колико-толико добар фронтенд изглед. Јер већина модерних сајтова који имају квизове раде на принципу "једно питање на једној страници", што би одузело доста времена, они који имају целе квизове на једној страници су, у главном, застарели, или су квизови више типа бирања слике и сл.

## ЛИТЕРАТУРА

- ifj.edu.pl (1999): ifj.edu.pl, A brief history of PHP, 1999, <https://ifj.edu.pl/private/krawczyk/php/intro-history.html>
- K. Ashwani (2021): K. Ashwani, What is XAMPP? and How to Install XAMPP?, 2021, <https://www.devopsschool.com/blog/what-is-xampp-and-how-to-install-xampp/>
- MariaDB (2023): MariaDB Foundation, MariaDB in brief, 2023, <https://mariadb.org/en/>
- MDN (2022): developer.mozilla.org, Javascript, 2022, <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- MySQL (2023): MySQL, MySQL Customers by Industry, 2023, <https://www.mysql.com/customers/industry/>
- Pedamkar (2022): Priya Pedamkar, Introduction to Versions of Html, 2022, <https://www.educba.com/versions-of-html/>
- php.net (2022): php.net, PHP: What is PHP?, 2022, <https://www.php.net/manual/en/intro-what-is.php>
- phpMyAdmin (2023): phpMyAdmin, phpMyAdmin, 2023,
- Reactnative (2022): reactnative.dev, React Native - Learn once, write anywhere, 2022, <https://reactnative.dev/>
- talend (2023): talend, What is MySQL? Everything you need to know, 2023, <https://www.talend.com/resources/what-is-mysql/>
- W3 (2010): w3.org, HTML & CSS, 2010, <https://www.w3.org/standards/webdesign/htmlcss#whatcss>
- W3Schools (2022): w3schools.com, Bootstrap 3 Tutorial, 2022, <https://www.w3schools.com/bootstrap/>
- W3schools, 2022: W3schools.com, HTML Introduction, 2022, [https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)
- wpbeginner (2022): wpbeginner, What is Apache? What is Webserver?, 2022, <https://www.wpbeginner.com/glossary/apache/>