

# WEB APLIKACIJA ZA VREDNOVANJE ŠKOLSKIH ZADAĆA

---

Čavka, Ivana

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split / Sveučilište u Splitu**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:228:166793>

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

Download date / Datum preuzimanja: **2024-07-17**



Repository / Repozitorij:

[Repository of University Department of Professional Studies](#)



UNIVERSITY OF SPLIT



**SVEUČILIŠTE U SPLITU**

**SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE**

Preddiplomski stručni studij Informacijska tehnologija

**IVANA ČAVKA**

**ZAVRŠNI RAD**

**WEB APLIKACIJA ZA  
VREDNOVANJE ŠKOLSKIH ZADAĆA**

Split, rujan 2023.

# SVEUČILIŠTE U SPLITU

## SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Preddiplomski stručni studij Informacijska tehnologija

**Predmet:** Programiranje na internetu

## ZAVRŠNI RAD

**Kandidat:** Ivana Čavka

**Naslov rada:** Web aplikacija za vrednovanje školskih zadaća

**Mentorica:** Marina Rodić, viši predavač

Split, rujan 2023.

# Sadržaj

Sažetak.....	2
Summary.....	3
Uvod .....	4
1. Korištene tehnologije .....	5
1.1. PHP, Laravel.....	5
1.2. HTML, CSS, JavaScript, Bootstrap.....	7
1.3. MYSQL, XAMPP.....	8
2. Izrada aplikacije .....	9
2.1. Ideja i postavljanje projekta.....	9
2.2. Postavljanje baze podataka .....	11
2.3. Korisnički računi.....	14
2.3.1. Uloga administrator .....	16
2.3.2. Uloga učitelj .....	22
2.3.3. Uloga učenik.....	30
Zaključak .....	32
Literatura .....	33

## Sažetak

Rad opisuje postupak izrade internet aplikacije pod nazivom „TaskTastic“ u razvojnom okruženju Laravel. Uz navedene tehnologije, korišteni su i jezici HTML (engl. *HyperText Markup Language*), JavaScript i CSS (engl. *Cascading Style Sheets*) u svrhu postavljanja izgleda internetskih stranica aplikacije. Aplikacija je namijenjena administratorima škole, učenicima i učiteljima kao platforma za objavljivanje, predaju i rješavanje domaće zadaće. Svaki korisnik se može registrirati i prijaviti u aplikaciju, a na radnoj ploči uz informacije o svom korisničkom računu, ima mogućnost pregleda zadaća i rezultata. Administratori upravljaju članovima škole i pregledavaju njihov rad. Učitelji mogu dodavati nove domaće zadaće ispunjavanjem predložaka pitanja, postaviti najveći broj bodova zadaće, pregledavati učeničke zadaće i evaluirati iste. Učenici mogu rješavati zadaće i pregledavati njihove statuse, kao i svoje rezultate. Liste podataka mogu se ispisati u *pdf* obliku, a sve važne obavijesti (potvrda *email* adrese, prijava novog člana škole ili razreda, dodijeljena zadaća, riješena ili pregledana zadaća) stižu korisniku u elektronskom poštanskom pretincu.

**Ključne riječi:** internet aplikacija, Laravel, PHP, zadaća

## **Summary**

### **Web application for school assignments review**

The paper describes the process of creating an internet application named "TaskTastic" in the Laravel development environment using the PHP language. In addition to the mentioned technologies, HTML, JavaScript, and CSS languages were utilized to establish the layout of the application's web pages. The application is designed for school administrators, students, and teachers, serving as a platform for assignment posting, submission, and review. Each user can register and log in to the application, and on the dashboard, along with their account information, they have the ability to view assignments and results. Administrators manage school members and review their work. Teachers can add new assignments by completing question, set the maximum points for assignments, review student assignments, and evaluate them. Students can solve assignments, review their statuses, as well as their results. Data lists can be exported in PDF format, and all important notifications (email address confirmation, new school or class member registration, assigned assignment, solved or reviewed assignment) are delivered to the user's email inbox.

**Keywords:** assignment, Laravel, PHP, web application

## Uvod

Ideja aplikacije „TaskTastic“ rođena je nakon razdoblja nastave na daljinu uzrokovanog pandemijom. U najviše škola diljem Hrvatske, bilo je potrebno svakodnevno koristiti nekoliko servisa i računalnih aplikacija kako bi se svaka zadaća izradila, preuzela, riješila, predala na pregled i naposljetku ocijenila. Cilj ove aplikacije je obuhvaćanje cijelog procesa na jednoj internet adresi u svrhu olakšanja rada svakome od korisnika – učitelju, učeniku i administratoru škole. U aplikaciji, administrator predstavlja jednu školu i istom upravlja: dodaje njene razrede usput postavljajući razrednika te šalje pozivnice učenicima i učiteljima za pristup grupi škole. Učitelj može stvarati nove zadaće, dodjeljivati ih učenicima na rješavanje (po razredima ili pojedinačno), pozivati nove učenike u aplikaciju te pregledavati i ocjenjivati zadaće. Učenik može rješavati svoje zadaće i pratiti svoje rezultate u odnosu na prosjek riješenosti zadaće. Svaka zadaća može imati proizvoljan broj pitanja koji se dodaju dinamički, a pitanja su kategorizirana vrstom odgovora. Ponuđeno je pet vrsta pitanja: pitanja s tekstualnim odgovorom, pitanja s brojčanim odgovorom, pitanja s odgovorom u privitku, pitanja s jednostrukim odabirom i pitanja s višestrukim odabirom. Svakom pitanju se može priložiti dokument kao privitak za preuzimanje ili omogućiti pomoć pri rješavanju. Ako se doda opcija pomoći pri rješavanju, na tom pitanju se mogu odrediti i kazneni bodovi za korištenje pomoći. Korisnici imaju tablični uvid u podatke koji se mogu poredati, pretraživati ili grupirati u stranice, a za uvid u rezultate zadaća po mjesecima, učenici i učitelji imaju grafički pregled prosječnih vrijednosti.

Struktura rada podijeljena je na dva glavna poglavlja: opis korištenih tehnologije i izrada same aplikacije. Poglavlje o tehnologijama opisuje glavne alate i jezike korištene u procesu, a to su: programski jezik PHP i razvojno okruženje Laravel; jezici HTML, CSS, JavaScript i okvir Bootstrap; baza podataka MySQL te softverski paket XAMPP (engl. *Cross-Platform, Apache, MySQL, PHP, and Perl*). Nadalje, pri izradi aplikacije izdvojile su se tri etape rada – razvijanje ideje i postavljanje poslužiteljske aplikacije, postavljanje baze podataka te konfiguracija korisničkog dijela aplikacije. Razvijanje korisničkog dijela aplikacije odrađeno je kroz tri područja usko vezana uz tri moguće uloge korisnika: uloga administratora škole, uloga učitelja i uloga učenika. Zaključak rada iznosi osvrt na korištene tehnologije i izvedbu same aplikacije.

# 1. Korištene tehnologije

## 1.1. PHP, Laravel

PHP (rekurzivni akronim od PHP: *Hypertext Preprocessor*) [1] [2] je skriptni jezik osmišljen za olakšanu izradu internetskih stranica i internetskih aplikacija. Stvoren sredinom devedesetih godina prošlog stoljeća kao proceduralan jezik otvorenog kôda, kroz godine se ubrzano razvijao prilagođavajući se potrebama korisnika te je preuzeo načela objektno-orijentiranog programskog jezika. Trenutno je jedan od vodećih jezika po broju internetskih stranica koje ga koriste, uključujući Wordpress platformu. Jednostavna sintaksa bazirana na C, Java i Perl programskim jezicima, iako s nekoliko specifičnosti (npr. \$ znak označava varijablu i piše se uz njeno ime, bez razmaka), omogućuje intuitivno pisanje kôda. Neke od glavnih prednosti PHP jezika nad ostalim skriptnim jezicima (JavaScript, ASP i drugi) su pojednostavljena mogućnost uvođenja različitih biblioteka u radno okruženje, pouzdanost koju nudi nakon niza nadogradnji, velik izbor podržanih biblioteka i dodataka te hvaljen sustav za upravljanje sadržajem (engl. *CMS- Content Management System*). Izvorna internet stranica <http://www.php.net> sadrži svu potrebnu dokumentaciju za rad u jeziku te sve potrebne pakete za instalaciju u sustav koje je moguće preuzeti besplatno. Dostupni su i paketi s uključenim drugim proizvodima za razvoj aplikacija kao što su Apache Web Server poslužitelj i MySQL baza podataka, što pridonosi olakšanoj pripremi radnog okružja. Jedan od takvih paketa, XAMPP (engl. *Cross-Platform, Apache, MySQL, PHP, and Perl*), korišten je i pri izradi „TaskTastic“ aplikacije u verziji za operacijski sustav Windows, uz bok PHP inačici 7.4.28.

Laravel [3] [4] je jedan od trenutno vodećih razvojnih okruženja otvorenog kôda dizajniranih za rad s PHP jezikom. Njegova priča započela je prije samo 12 godina, no unatoč tome što je relativno mlada tehnologija, Laravel je svojom jednostavnom strukturom i temeljito napisanom dokumentacijom pridobio mnoge korisnike. Okruženje je organizirano MVC arhitekturom (engl. *Model-View-Controller*, hrv. Model-Pogled-Upravljač [5]) koja omogućuje prikaz podataka od baze do same stranice u samo tri koraka (Slika 1.). Prvi korak podrazumijeva interakciju s bazom podataka, dohvaćanje, spremanje, osvježavanje i brisanje podataka. Svaka tablica u bazi (osim međutablica, engl. *pivot tables*) predstavlja jedan model u okruženju (engl. *model*). Drugi korak predstavlja potrebnu obradu i usmjeravanje podataka koji se dohvaćaju iz baze i šalju na stranicu. Obavlja se u



takozvanim upravljačima (engl. *controller*), što su zapravo skupine funkcija kategorizirane po modelima s kojima su povezane.

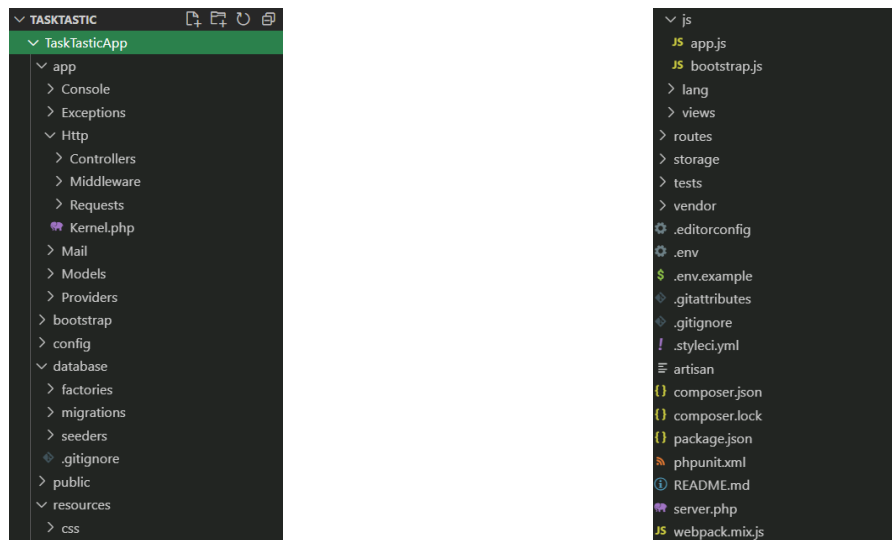
Posljednji korak do stranice je ujedno i dizajn pojedine stranice, odnosno korisničko sučelje (engl. *view*) te sadrži HTML kôd, potrebne CSS i JavaScript dodatke kôda ili čak umetnuti PHP kôd u službi građe same stranice. Za dizajn stranice može se koristiti i Blade, Laravelov alat za predloške (engl. *templating engine*) koji nudi mnoge mogućnosti za laku izradu željenih pogleda, upravljanje proslijeđenim podacima i stvaranje podsekcija za umetanje u druge sekcije. Pri izradi „TaskTastic“ aplikacije korištena je Laravelova verzija 8.83.27.



Slika 1. MVC model

Novi Laravel projekt stvara se pomoću Composera, alata za upravljanje paketima i njihovim zavisnostima u PHP jeziku, dostupnog preko naredbenog retka. Instalaciji naredbenog sučelja Composer prethodi instalacija XAMPP paketa koji sadrži PHP. Nakon postavljanja Composer-a, slijedi preuzimanje i postavljanje Laravel okruženja direktno pri stvaranju projekta, primjerice `composer create-project laravel/laravel TaskTasticApp`. Pri stvaranju projekta, Laravel izrađuje cjelokupan datotečni sustav s gotovom osnovnom podstavom aplikacije, uključujući početnu internet stranicu (engl. *home*, dostupna na internet lokaciji `http://localhost:8000` pri pokretanju servera), osnovne migracijske skripte (izrađuju tablice `User`, `Password resets`, `Failed Jobs` te `Personal access tokens`) i korisnički model (engl. *user*). Raspored datotečnog sustava Laravela (Slika 2.) prati MVC model te odvaja korisnički dio i vezu s bazom podataka od logičkog dijela aplikacije. Također, olakšava kategorizaciju novih komponenti predodređenim dodatnim mapama, kao što su, naprimjer, spremanje datoteka i slika (*Storage* mapa) ili izrada elektronske pošte (*Mail* mapa). Pri stvaranju novih (npr. nove komponente: upravljači, migracijske skripte, modeli i sl.) i obradi ili pokretanju postojećih dijelova projekta, s Laravelom se komunicira preko Artisan sučelja naredbenog retka (engl. *CLI*, *Command-line interface*). Za olakšano spremanje i dohvaćanje podataka iz baze,

Laravel ima rješenje u obliku alata po imenu Eloquent ORM (alat za objektno relacijsko mapiranje, engl. *Object-relational mapping*).



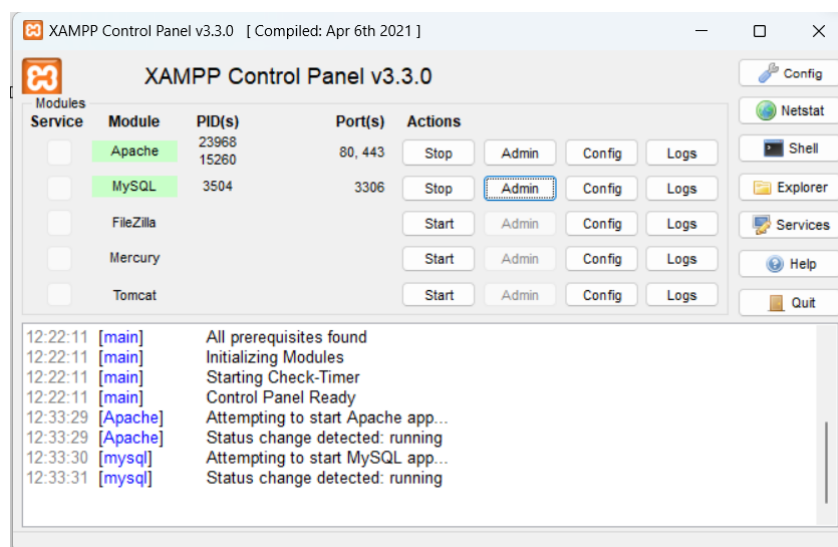
Slika 1. Datotečni sustav Laravel projekta, primjer „TaskTastic“ aplikacija

## 1.2. HTML, CSS, JavaScript, Bootstrap

HTML (skraćeno od engl. *HyperText Markup Language*) je skriptni jezik koji definira tipove sadržaja i podataka koje dokument prikazuje. CSS (skraćeno od engl. *Cascading Style Sheets*) je jezik koji opisuje stilove dijelova dokumenta vezane primjerice uz boju, tipografiju, položaje, okvire i slično. Iako su dva različita jezika, često se uparuju zbog svoje zavisnosti jedan o drugome. Ukratko, HTML opisuje strukturu i dijelove stranice dok CSS opisuje kako će sama stranica i njeni pojedini dijelovi izgledati. JavaScript je objektno-orientirani programski jezik koji je postao jedan od temeljnih alata za izradu internet stranica. Uz njegovu pomoć, moguće je, primjerice, definirati responzivne dijelove stranice i pritom obrađivati podatke na samoj stranici stranici. Zbog svoje široke i česte primjene, JavaScript je kroz godine dobio mnogo radnih okruženja i biblioteka, kao što su Lodash, TypeScript, jQuery (biblioteka čak i uvrštena u novije verzije JavaScripta), React, itd. Bootstrap značajno ubrzava proces izgradnje korisničkog sučelja zahvaljujući svojim cjelokupnim i temeljito testiranim kombinacijama HTML označavanja, CSS stilova i JavaScript ponašanja. Bootstrap je jedan od najkorištenijih radnih okruženja za dizajn izgleda internet stranica - široko je dostupan i besplatan, a nudi velik raspon korisniku prilagođenih, univerzalnih i testiranih dijelova kôda za često korištene komponente korisničkog sučelja. Sve četiri navedene tehnologije korištene su pri izradi „TaskTastic“ aplikacije.

### 1.3. MYSQL, XAMPP

MySQL poslužitelj (engl. *MySQL server*) je sustav za upravljanje bazama podataka koji koristi SQL jezik (skraćeno od engl. *Structured Query Language*). SQL je nedvojbeno najčešće korišteni standardizirani jezik za manipuliranje bazama podataka. SQL naredbe mogu se izravno unijeti u poslužitelj ili u kôd napisan u nekom drugom jeziku ili se pak može koristiti posrednički alat koji u potpunosti skriva SQL sintaksu, kao što je to i slučaj u „TaskTastic“ projektu, gdje je korišten Laravelov Eloquent ORM. Sam MySQL dio je Apache Web XAMPP poslužiteljskog paketa (Slika 3.), besplatnog za preuzimanje na stranici <https://www.apachefriends.org/download.html>. XAMPP ima prilagođene verzije za različite operacijske sustave te puno dodatka među kojima su PHP, Tomcat, Mercury, FileZilla, ali i WordPress, Drupal i Joomla. Za pregled MySQL baze podataka, pri izradi „TaskTastic“ aplikacije korišteno je phpMyAdmin administratorsko sučelje. Laravel projekt stvara se unutar htdocs mape na disku nakon instalacije Composer naredbenog sučelja u sustav računala i samog Laravel okružja.



Slika 3. XAMPP v3.3.0 u primjeru „TaskTastic“ aplikacije

## 2. Izrada aplikacije

### 2.1. Ideja i postavljanje projekta

Ideja projekta je nastala u interakciji s učenicima, studentima i učiteljima kao jedno od pomagala za online nastavu. Aplikacija je namijenjena za izradu zadataka, slanje zadataka učenicima na rješavanje, rješavanje iste te njenu provjeru, sve izravnim putem u svrhu izbjegavanja gubljenja informacija ili stvaranja dezinformacija pri objavljivanju poruka u alatima forumskog tipa ili društvenih mreža. Administrator škole dodaje učitelje, razrede i učenike te prati njihovu aktivnost, učitelj stvara zadatke, dodaje učenike, ocjenjuje zadatke i prati rad učenika, a učenici rješavaju zadatke i prate svoje rezultate. Učenicima su podaci zaštićeni u odnosu na druge učenike kako bi se održao fokus svakog pojedinca na razvoj vlastitih sposobnosti, ali zato može pratiti rezultat u odnosu na prosjek razreda. Svaki korisnik može dostupne podatke pratiti i putem statistike prikazane u grafovima. Pojedine važne obavijesti (primjerice, o novim zadacima ili novo priključenim korisnicima) se automatski šalju na internetsku adresu korisnika, kako ne bi bilo potrebe za učestalim provjeravanjem podataka u aplikaciji. Dizajn projekta osmišljen je kao vizualno poticajan i razigran, koristeći glavnih pet boja: plava, žuta, crvena, bijela i crna. Ime aplikacije je spoj engleskih riječi *task* (hrv. zadatak) i *fantastic* (hrv. odlično) = „TaskTastic“, što također implicira na rad, izvrsnost i zaigranost. Vizualni identitet aplikacije, u navedenim bojama, izrađen je u alatu za dizajn dokumenata Canva (besplatan uređivač dostupan na <https://www.canva.com> internet stranici). Logo (Slika 4.) je sačinjen od čitkog naziva aplikacije ukomponiranog u dva lista papira koja predstavljaju zadatke i dvije kvačice koje predstavljaju izvrsnost. Svaka stranica u aplikaciji sadrži navigacijsko okno s vizualnim identitetom. Na početnoj stranici aplikacije (pogled `home.blade.php`) korištena je generirana slika besplatnog alata [www.wepik.com](http://www.wepik.com) koji prema opisanom zahtjevu, uz pomoć umjetne inteligencije, pripremi nekoliko slikovnih prikaza zadane teme (primjer zahtjeva: „slika koja sadrži radni stol školarca te sadrži žutu, plavu i crvenu boju“) (Slika 6.).



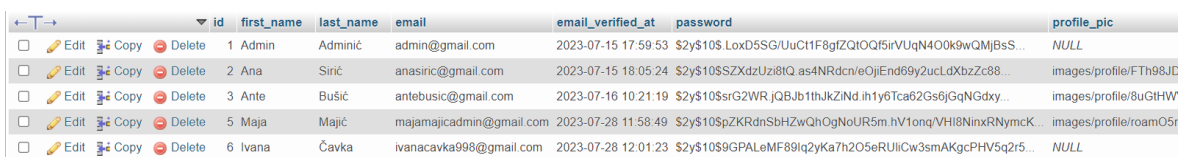
Slika 4. Vizualni identitet aplikacije „TaskTastic“

Svaka stranica, odnosno pogled aplikacije, izrađena je kao unutrašnji dio predloška `app.blade.php`. U svakom drugom pogledu koji se ugrađuje u predložak, na sami početak dodaje se `@extends('layouts.app')` Blade naredba. Predložak sadrži glavni okvir HTML stranice, s vezama na sve korištene CSS i JavaScript skripte, odnosno verziju Bootstrapa te postavkama jezika u `<head>` elementu. `<body>` element sadrži predložak navigacijskog okna s vizualnim identitetom s lijeve strane koji obavlja funkciju preusmjeravanja na početnu stranicu (engl. *home*) te dugmad za korisničke mogućnosti s desne strane. Dok korisnik nije prijavljen, vidljiva je dugmad za padajući izbornik registracije i prijavu (engl. *register and login*), a kada je korisnik prijavljen, vidljiva je dugmad za preusmjeravanje na radnu ploču korisnika (engl. *dashboard*) i odjavu (engl. *logout*). Navedena razlika omogućena je korištenjem `Auth::check()` funkcije koja pripada biblioteci `Illuminate\Support\Facades\Auth` te ju je moguće direktno pozvati u pogledu (engl. *view*). Središnji dio svake stranice aplikacije dohvaća se Blade opcijom `@yield('content')` te se u ostalim pogledima koji se žele umetnuti imenuje kao `@section('content')`. Podnožje stranice izrađeno je također u predlošku i osigurano na poziciji dna stranice neovisno o sadržaju. Na taj način postignut je uniforman izgled svih stranica aplikacije.

Svaki pogled aplikacije izvodi se na naredbu `view()` u nekom od upravljača. Pogledu se može i pridružiti i skup varijabli s dodatkom `with('ime', $varijabla)` gdje prvi argument označava ime varijable koja će biti dostupna u pogledu, a drugi argument varijablu upravljača čija se vrijednost prosljeđuje. U pogledu se varijable dohvaćaju navedenim imenom u okviru dvostrukih vitičastih zagrada, npr. `{{ $ime->atribut }}`. Funkcije upravljača koje generiraju poglede povezane su s internet adresom (engl. *route*) na kojoj su dostupne u `web.php` datoteci. Uz povezivanje željene funkcije s adresom, određuje se i ime adrese za lakše pozivanje unutar aplikacije, moguće posredničke funkcije i metoda izvođenja stranice *GET* (hrv. dohvati) ili *POST* (hrv. pošalji). Navedene dvije metode se koriste ovisno o željenom učinku - GET metodu koristimo pri ispisivanju podatka na stranicu, a POST metodu pri unošenju podataka u sustav sa stranice (npr. pri unosu korisničkih podataka).

## 2.2. Postavljanje baze podataka

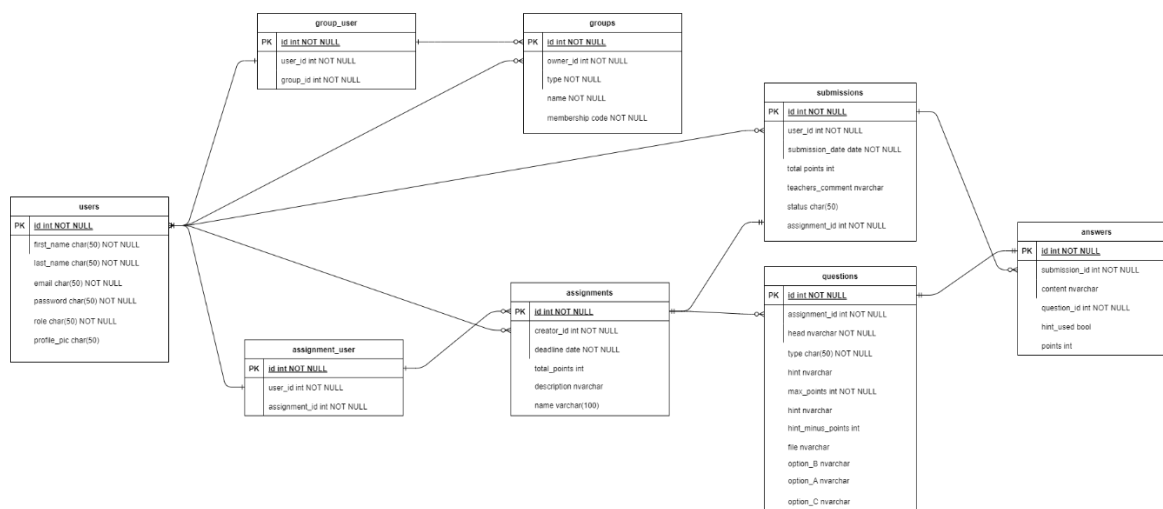
Baza podataka projekta postavljena je u MySQL poslužitelju XAMPP platforme te se postavlja na priključak 3306 pri adresi računala 127.0.0.1. U aplikaciji se u datoteci `.env` postavlja konfiguracija za povezivanje s bazom na spomenuti priključak zajedno s korisničkim podacima administratora baze (korisničko ime, ime baze, lozinka za pristup). Podaci uneseni u bazu i njene postavke mogu se provjeriti u pregledniku na adresi <http://localhost/phpmyadmin/> (Slika 5.). Navedenoj adresi može se direktno pristupiti i pritiskom dugmeta Admin unutar MySQL retka u XAMPP prozoru.



	id	first_name	last_name	email	email_verified_at	password	profile_pic
<input type="checkbox"/>	1	Admin	Adminić	admin@gmail.com	2023-07-15 17:59:53	\$2y\$10\$LoxD5SG/UuC1F8gIZQIQI5irVUqN4O0k9wQMjBsS...	NULL
<input type="checkbox"/>	2	Ana	Sirić	anasiric@gmail.com	2023-07-15 18:05:24	\$2y\$10\$SZXdzUzi8tQ.as4NRdcn/eOjEnd89y2ucl.dXbzZc88...	images/profile/FTh98JD
<input type="checkbox"/>	3	Ante	Bušić	antebusic@gmail.com	2023-07-16 10:21:19	\$2y\$10\$SrG2WR.jQB.Jb1th.JkZiNd.iH1y6Tca82Gs6jGqNGdx...	images/profile/8uGHW
<input type="checkbox"/>	5	Maja	Majić	majamajcadmin@gmail.com	2023-07-28 11:58:49	\$2y\$10\$PzKRdnSbHZwQhOgNoUR5m.hV1onq/VHl8NinxRNymcK...	images/profile/roamO5i
<input type="checkbox"/>	6	Ivana	Čavka	ivanacavka998@gmail.com	2023-07-28 12:01:23	\$2y\$10\$9GPALeMF89lq2yKa7h2O5eRUIiCw3smAKgcPHV5q2r5...	NULL

Slika 5. Ulomak prikaza tablice `users` na `phpmyadmin` sučelju

Baza podataka „TaskTastic“ aplikacije osmišljena je kao sustav osam povezanih tablica (Slika 6.). Glavne tablice su korisnici (engl. *users*), grupe (engl. *groups*), zadaće (engl. *assignments*) i predaje zadaća (engl. *submissions*). Veza tablica korisnici i grupe te korisnici i zadaće povezane su međutablicama (engl. *pivot tables*: *group\_user*, *assignment\_user*) kako bi se mogla ostvariti tablična veza više-na-više (engl. *many to many*). Na taj način omogućena je veza vlasnika grupe ili kreatora zadaće također s tablicom korisnika, ne ometajući povezanost članova određene grupe s grupom ili učenika kojima je dodijeljena pojedina zadaća sa zadaćom. Zadaće su povezane vezom jedan-na-više (engl. *one to many*) s pitanjima koja ih sačinjavaju. Predaje zadaća povezane su također s korisnikom koji ih je kreirao, ali i zadaćom kojoj pripadaju. Naposljetku, tablica predaje zadaće povezana je također vezom jedan-na-više s tablicom odgovora, koji su opet pojedinačno spojeni vezom jedan-na-jedan (engl. *one to one*) po s pitanjem kojem pripada odgovor.



Slika 6. Skica baze podataka „TaskTastic“ aplikacije

Migracijske skripte, tj. klase moraju biti dostupne za svaku tablicu iz baze u mapi `database/migrations`. Postavke su im predodređene u `Illuminate\Database\Migrations` biblioteci. Njihova uloga je izrađivanje tablice u bazi te se izvršavaju naredbom `php artisan migrate`. No, u slučaju nastale greške, migracije se mogu poništiti naredbom `php artisan migrate:rollback` ili izradom dodatne skripte. Velika prednost korištenja migracijskih skripti je to što se mogu koristiti svakim ponovnim postavljanjem projekta na različitim uređajima i bazama podataka. Migracijske skripte za osnovne tablice `User`, `Password resets`, `Failed Jobs` te `Personal access tokens` stvorene su pri izradi samog Laravel projekta, a ostale tablice `Assignments` (Ispis 1.), `Groups`, `Assignment-User`, `Group-User`, `Jobs`, `Questions`, `Submissions`, `Answers` su naknadno dodane. Proces dodavanja nove skripte kreće u naredbenom sučelju s naredbom `php artisan make:migration create_name_table` gdje „*name*“ predstavlja ime tablice koja se želi napraviti. Kako se tablica u bazi povezuje direktno s istoimenim modelom u aplikaciji, nužno je pritom i izraditi sam model (Ispis 2.). Još jedna opcija je i spojiti obje radnje koristeći naredbu `php artisan make:model Ime -m`, gdje „*Ime*“ predstavlja ime modela koji se izrađuje, a „-m“ generiranje migracijske skripte za taj model. Normirano nazivlje za svaki model u aplikaciji je velikim početnim slovom i u jednini, a za svaku tablicu u množini (osim međutablica koje ne moraju imati svoj pripadni model i čije se ime sastoji od imena oba modela povezanih donjom crticom „\_“).

```

<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
class CreateAssignmentsTable extends Migration
{
    /* Run the migrations. @return void*/
    public function up( {
        Schema::create('assignments', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->integer('creator_id');
            $table->datetime('deadline');
            $table->string('description')->nullable();
            $table->integer('total_points')->nullable();
            $table->string('slug')->nullable();
            $table->timestamps();
        });
    }
    /* Reverse the migrations. @return void*/
    public function down(){
        Schema::dropIfExists('assignments');
    }
}

```

*Ispis 1. Migracijska skripta za tablicu assignments u „TaskTastic“ aplikaciji*

Za izradu i obradu podataka vezanih uz modele, Laravel koristi Eloquent ORM alat. Primjerice, za uspostavljanje veza između modela se vrlo jednostavno i intuitivno može osigurati dohvaćanje i spremanje podataka u povezanom modelu. Veze za dohvaćanje nije potrebno definirati s obje strane, već u onom modelu u kojem su potrebni podaci drugog modela (kao što je učinjeno u primjeru iz Ispisa 2.). Ako je pak potrebno spremati podatke, to se može učiniti u pripadnom upravljaču s naredbom `syncWithoutDetaching()`.

```

<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\HasMany;
class Assignment extends Model
{
    use HasFactory;
    /* The attributes that are mass assignable. @var array<int, string>*/
    protected $fillable = [
        'description', 'name', 'deadline', 'total_points', 'creator_id',
    ];
    public function users() {
        return $this->belongsToMany(User::class, 'assignment_user',
            'assignment_id', 'user_id')
            ->withPivot('allowed')->withTimestamps();
    }
    public function submissions(): HasMany{
        return $this->hasMany(Submission::class)
            ->withTimestamps();
    }
    public function questions(): HasMany {
        return $this->hasMany(Question::class);
    }
}

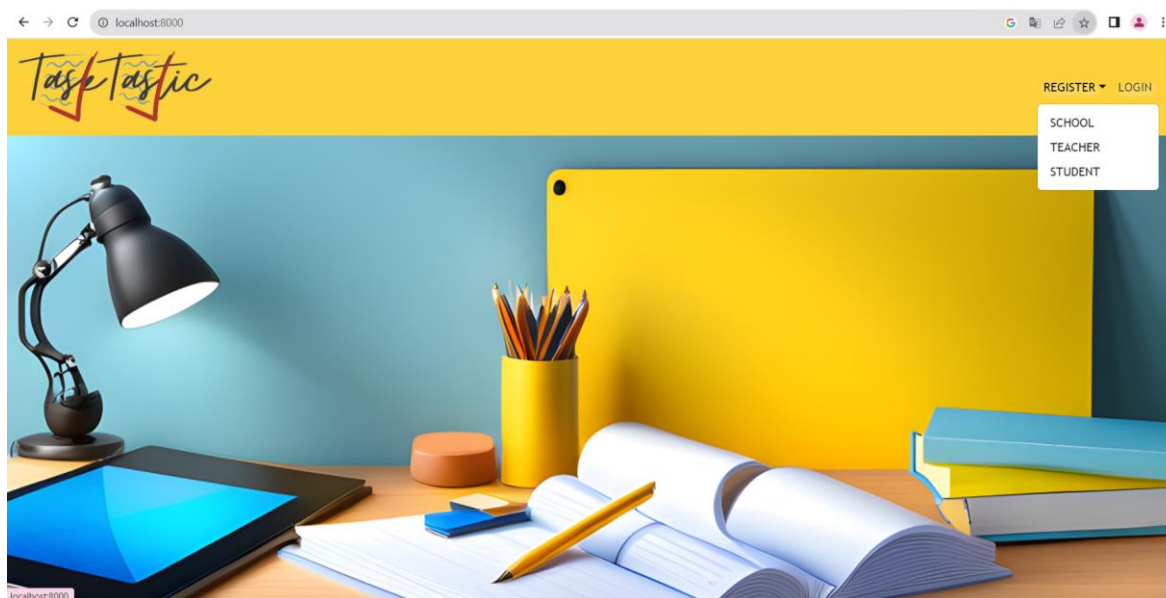
```

*Ispis 2. Assignment model povezan uz migracijsku skriptu iz ispisa 1. i assignments tablicu u „TaskTastic“ aplikaciji*



## 2.3. Korisnički računi

Korisnik (engl. *user*) u aplikaciji ima tri moguće uloge: administrator škole (engl. *admin*), učitelj (engl. *teacher*) i učenik (engl. *student*). Podatak o ulozi (engl. *role*) je jedan od atributa tablice `users`, te njegove vrijednosti mogu biti jedna od navedene tri opcije, spremljene kao konstante u modelu `User` (npr. `const teacherRole = 'teacher';` ). Svaki korisnik mora biti dio neke organizacije, odnosno grupe – administrator škole je upravitelj škole, učitelj je član škole i upravitelj jednog ili više razreda, a učenik je član razreda. Iako svaki korisnik ima iste podatke za unos (ime, prezime, lozinka, potvrdna lozinka te email), registracija korisnika ovisi o njegovoj ulozi: registracija administratora sadrži i registraciju škole, dok se učitelj i učenik mogu priključiti svojoj grupi samo uz članski kôd grupe dobiven putem internetske pošte a poslao ga je upravitelj grupe. Upravitelji grupe povezani su u bazi s atributom `owner_id` u tablici `groups`, a članovi grupe s međutablicom `group_user` zbog relacije više na više (engl. *many to many*). Tako su dostupne tri različite registracijske forme i internetska mjesta kojima se može pristupiti s početne stranice (Slika 7.).



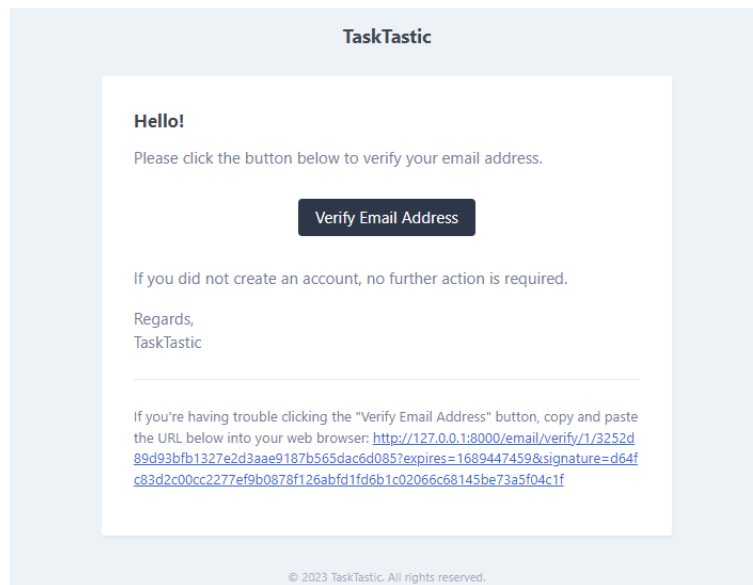
Slika 7. Početna stranica aplikacije „TaskTastic“ s otvorenim izbornikom registracije

No, nijedan od korisnika se neće moći prijaviti u aplikaciju dok ne potvrdi vjerodostojnost svoje internetske adrese (koja se u procesu registracije također provjerava u bazi da je za svakog korisnika jedinstvena) klikom na dugme u pristigloj internetskoj pošti (Slika 8.). Poruka internetske pošte izrađuje se automatski i prosljeđuje korisniku putem naredbi dostupnih u Laravelovoj `Illuminate\Auth\MustVerifyEmail` biblioteci. Za slanje poruke koristi se naredba `$user->sendEmailVerificationNotification();`. Pri izradi „TaskTastic“ aplikacije korišten je besplatni alat <http://www.mailtrap.io> za provjeru odlaznih poruka i njihovo stizanje na željenu internetsku adresu. MailTrap je platforma za isporuku e-pošte za pojedince i tvrtke za testiranje, slanje i kontrolu infrastrukture e-pošte na jednom mjestu. Spajanje na platformu je također olakšano u Laravelovoj konfiguraciji, podešavanjem unaprijed pripremljenih podataka u `.env` datoteci. Moguće je tako osim postavljanja spajanja na servis i odrediti email adresu s koje će se slati pošta te ime pošiljatelja pošte (Ispis 3.).

```
MAIL_MAILER=smtp
MAIL_HOST=sandbox.smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=65125fd133c457
MAIL_PASSWORD=*****
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS="info@tasktastic.com"
MAIL_FROM_NAME="{APP_NAME}"
```

Ispis 3. ulomak `.env` datoteke u kojem je postavljena Mail konfiguracija

Prilikom klika na dugme u tablicu korisnika, u bazi podataka aplikacije se automatski upisuje datum potvrde vjerodostojnosti (atribut `email_verified_at`), dok se korisniku otvara stranica za prijavu u aplikaciju (engl. *login*) gdje upisuje svoje podatke (internetsku adresu i lozinku). Pri obradi podataka koje korisnik upisuje za ulazak korisnika u aplikaciju provjerava se je li njegova adresa potvrđena naredbom `if($user->hasVerifiedEmail());`. Ako je vjerodostojnost potvrđena, korisnik ulazi u aplikaciju, a ako ne, na *login* pogled izlazi poruka korisniku da provjeri svoju poštu i potvrdi vjerodostojnost ili pošalje novu poruku verifikacije na svoju adresu pritiskom na dugme engl. *Send verification email again* (hrv. opet pošalji poruku za potvrdu email adrese). Sami procesi prijave i odjave korisnika (engl. *login* i *logout*), ali i dohvaćanje trenutnog korisnika, podržani su u biblioteci `Illuminate\Support\Facades\Auth`.



Slika 8. Generirana elektronska pošta za provjeru vjerodostojnosti internetske adrese korisnika u MailTrap alatu

### 2.3.1. Uloga administrator

Administratorska uloga omogućava pregled događanja unutar škole. Jedan administrator može upravljati jednom školom. Administrator može pozvati učenike ili učitelje u aplikaciju, može stvarati nove razrede i dodavati im učitelje koji ih vode (jedan učitelj može voditi više razreda). Administrator također može vidjeti popis svih dodanih korisnika te pratiti njihov rad, kao i generirati pdf ispis sadržaja tablica.

Pri prvom dodiru s aplikacijom, administrator škole (u registracijskom izborniku škola, engl. *school*) upisuje svoje podatke i ime škole koju će predstavljati u HTML formu. Klikom na potvrdni dugme `Register`, podaci iz forme šalju se `POST request` metodom na obradu na internet lokaciju povezanu s funkcijom `saveSchool` unutar `UserController` upravljača. Poslani podaci označeni su atributom `ime` (engl. *name*) u HTML kôdu, te se zajedno s upisanom vrijednošću šalju kao `$request` kolekcija parova ključa i vrijednosti (engl. *key-value*) `POST` metodom. Unutar navedene funkcije (Ispis 4.), uneseni podaci se dohvaćaju po ključu i prvo provjeravaju kako bi se donekle osigurao ispravan unos. Primjerice, ime i prezime ne smiju imati manje od 2 znaka ni više od 100, moraju biti niz znakova (engl. *string*) i obavezni su (`'firstName' => ['required', 'string', 'max:100', 'min:2']`).

Potvrдна lozinka mora odgovarati odabranoj lozinki i uvijek moraju biti obje upisane ('confirmPassword' => 'required', 'required\_with:password', 'same:password']), a email adresa mora biti jedinstvena za svakog korisnika ('email' => 'required', 'string', 'max:150', 'email', 'unique:users']). Ove provjere mogu se upisati direktno u funkciju upravljača, no u ovom slučaju zapisane su u SchoolRegistrationRequest.php datoteci čija je klasa prosljeđena kao argument. Pri spremanju podataka o školi ili razreda, generira se niz od 10 znakova i sprema kao članski kod u tablicu grupe.

```
public function saveSchool(Request $request,
                           SchoolRegistrationRequest $registrationRequest) {
    $user = User::create([
        'first_name' => $request->get('firstName'),
        'last_name' => $request->get('lastName'),
        'email' => $request->get('email'),
        'password' => bcrypt($request->get('password')),
        'role' => User::adminRole, ]);
    $user->sendEmailVerificationNotification();
    GroupController::saveGroup($request->get('school'), $user->id,
                               Group::schoolType);
    return redirect()->intended('login')->with('successMessage',
        'Registered successfully. But before logging in, please make sure you
        verify your account with the link received in the mail.')
        ->with('verified', true);
}
```

Ispis 4. saveSchool funkcija u UserController upravljaču

Ako podaci nisu ispravni, korisnik se preusmjerava natrag na formu, gdje ga uz neispravan unos čeka predodređena poruka o vrsti greške. Također, lozinka se kriptira zbog sigurnosti korisnika bcrypt() metodom pri unosu u bazu.

Ulaskom u aplikaciju, korisnik pristupa svojoj radnoj ploči (engl. *dashboard*). U navigacijskom oknu ne prikazuje se više opcija za registraciju ili prijavu, već odjavu (engl. *logout*). Radnoj ploči, kao i drugim korisničkim stranicama, neprijavljeni korisnici ne mogu pristupiti, što je omogućeno korištenjem posredničke komponente (engl. *middleware*) za provjeru vjerodostojnosti. Navedeni priključak vidi se na primjeru za radnu ploču korisnika `Route::get('/dashboard', [DashboardController::class, 'index'])->name('dashboard')->middleware('auth');`. Ovisno o ulozi korisnika radna ploča ima ponešto drugačiji izgled, a pregledna podjela je omogućena korištenjem Blade sekcija – dijelova kôda koji se mogu po potrebi uključiti u druge dijelove naredbom `@include()`.

Administrator škole tako može vidjeti svoje korisničke podatke: sliku profila, ime i prezime, email adresu, ime škole te prije koliko dana su ažurirani njegovi podaci (Slika 9.). U slučaju da korisnik nema sliku profila prikazuje se zadana slika s ikonom osobe.

Svaki korisnik ima mogućnosti uređivanja svojih podataka (ime i prezime, email adresa, slika profila) te mu je prikazano prošlo vrijeme u danima od zadnjeg uređivanja. Korisnik može urediti svoje podatke klikom na dugme „uredi detalje“ (engl. *edit details*), a dugme „promijeni lozinku“ (engl. *change password*) vodi korisnika na stranicu s promjenom lozinke. Pri promjeni lozinke, korisnik mora dvaput unijeti istu novu lozinku koja prolazi istu provjeru i kriptiranje kao i pri registraciji.

TaskTastic

DASHBOARD LOGOUT

Classroom created successfully.

School administrator  
Last updated 3 days ago  
Karla Karlič  
karlakartic@gmail.com  
OŠ Bijaci  
[Edit details](#) [Change password](#)

CREATE NEW CLASSROOM

SEND AN INVITE TO TEACHERS

SEND AN INVITE TO STUDENTS

Classrooms Teachers Students

List of all teachers

10 entries per page

Search...

Last name	First name	Email	Classrooms	Students	Assignments	Delete
Ivić	Ivo	ivoivcbijaci@gmail.com	3	1	0	
Teachery	Teacher	teacherteacher@gmail.com	2	0	0	

Showing 1 to 2 of 2 entries

© 2023 Split, Croatia  
creator Ivana Čavka, an IT student at University Department of Professional Studies  
github.com/ivachavka

Slika 9. Radna ploča administratora škole u „TaskTastic“ aplikaciji (proširena zbog preglednosti)

Lako je uočiti tri dugmeta s aktivnostima koje administrator može napraviti, a to su izrada nove grupe razreda (engl. *create new classroom*), slanje pozivnice učiteljima (engl. *send an invite to teachers*) te slanje pozivnice učenicima (engl. *send invite to students*). Klikom na bilo koje dugme, korisnika se vodi na zasebnu stranicu s formom za unos podataka potrebnih za radnju. Pri izradi novog razreda, administrator treba unijeti ime razreda i odabrati jednog od učitelja koji su dio škole u padajućem izborniku. Da bi dodao nove učitelje škole, administrator treba upisati email adresu učitelja koji će zatim dobiti pozivnicu na istu adresu. Pozivnica je izrađena kao pogled (Ispis 5.), s datotečnim nastavkom `blade.php` i proslijeđenim podacima o grupi u koju se korisnika poziva te je za nju izrađena pripadna `Mail` upravljačka komponenta. Biblioteka koja sadrži potrebnu implementaciju za izradu poruka je `Illuminate\Mail\Mailable`.

Kako proces slanja poruka ne bi zadržavao i usporavao rad aplikacije, korišten je red čekanja, engl. *queue* (potrebne biblioteke `Illuminate\Bus\Queueable` i `Illuminate\Contracts\Queue\ShouldQueue`). Pri pokretanju poslužitelja aplikacije, tj. omogućavanju korištenja same aplikacije, potrebno je u dodatnom naredbenom retku pokrenuti i naredbu `php artisan queue:work` koja osigurava pravovremeno izvođenje procesa u redu čekanja u pozadini.

```
@component('mail::message')
Hello! You are invited to the {{$groupType}} {{$groupName}} <br>
Please input this code to access the {{$groupType}}: {{$code}}
We are happy to have you with us,<br>
{{config('app.name')}}
@endcomponent
```

Ispis 5. `invite.blade.php` pogled pozivnice

Nakon primitka pozivnice, novi korisnik se može registrirati koristeći članski kôd za pristup dobiven u poruci. Kad se taj novi korisnik registrira, administrator će dobiti obavijest na svoju email adresu da se njegovoj školi priključio novi učitelj. Za pozivanje učenika, proces je isti, no uz email adresu novog učenika, za slanje pozivnice potreban je i odabir jednog od razreda škole u padajućem izborniku (Ispis 6.). Sve potvrde izvršenih akcija neovisno o stranici, ispisuju se pomoću `message.blade.php` komponente ispod navigacijskog okna.

U navedenoj komponenti, dizajnirane su poruke uspješno izvedene radnje (engl. *successMessage*) i neuspješno izvedene radnje (engl. *errorMessage*) koje se prikazuju na stranici s tekstom poruke ako je sesiji prosljeđena poruka pomoću `with()` funkcije, kao npr. `return redirect()->intended('dashboard')->with('successMessage','Assignment deleted successfully.');` Ikone korištene na stranicama besplatno su dostupne na <http://www.w3schools.com> platformi i korištene u aplikaciji kao dio HTML kôda kao `<i>` element.

```
public function sendInviteToStudent(Request $request) {
    $request->validate([ 'email' => 'required','string','max:150','email', ]);
    $classroom = Group::where('id',$request->classroom)->firstOrFail();
    $code = $classroom->membership_code; $email = $request->get('email');
    try { Mail::to($email)->queue(new InvitationEmail($code, $classroom->type,
                                                    $classroom->name));
    } catch (\Exception $e) {
        return response()->json($e); }
    return redirect()->intended('dashboard')
        ->with('successMessage','Invitation sent successfully.');
```

Ispis 6. `sendInviteToStudent` funkcija za slanje pozivnice učeniku u `GroupUsersController` upravljaču

Prikaz podataka razreda, učitelja i učenika obuhvaća ostatak pogleda te je izveden kao navigacijska lista sa zasebnim karticama. Navedeni prikaz, uz bok drugim korištenim komponentama pogleda, dostupan je za besplatno korištenje na platformi <https://getbootstrap.com/> uvođenjem odgovarajuće skripte u HTML kôd (Ispis 7.). Dizajn tablice podataka dostupan je također besplatno na <https://startbootstrap.com/previews/sb-admin>. Lista podataka unutar tablice može se sortirati zasebno po svakom stupcu uzlazno i silazno, mogu se pretraživati podaci po znakovima i riječima pomoću pretraživača (engl. *search*) te se mogu dijeliti na stranice po skupinama od 5, 10, 15, 20 ili 25 redaka (engl. *pagination*).

Kartica s tablicom razreda sadrži stupce imena razreda (engl. *name*), imena i prezimena učitelja (engl. *teacher*), broja učenika koji pripadaju razredu (engl. *students*) i prosječnog rezultata zadataka svakog razreda (engl. *average score*) u postotcima. Tablica učitelja u pripadnoj kartici sadržava stupce prezimena (engl. *last name*), imena (engl. *first name*), email adrese (engl. *email*), brojem razreda (engl. *classrooms*), brojem učenika (engl. *students*) i broja zadataka (engl. *assignments*).

Kartica tablice učenika sadrži popis svih učenika škole s atributima prezime (engl. *last name*) i ime učenika (engl. *first name*), email adresa (engl. *email*), razred kojem učenik pripada (engl. *classroom*), koliko učenik ima zadanih zadaća (engl. *assigned to*), koliko ih je predano (engl. *solved*) te koliko ih je ostalo za riješiti (engl. *pending*). Administrator ima opciju brisanja svakog od razreda, učenika ili učitelja klikom na ikonicu kance za smeće na posljednjem mjestu retka tablice te se također može generirati pdf ispis za svaku od navedene tri tablice uz pomoć DOMpdf biblioteke [6] (nešto više o biblioteci u poglavlju uloge učitelja uz Ispis 12.).

```

<div id="users" class="card mb-2 mt-2 shadow-lg">
  <ul class="nav nav-tabs" id="users-tab" role="tablist">
    <li class="nav-item" role="presentation">
      <button class="nav-link active btn btn-outline-info" id="clasroom-tab"
        data-bs-toggle="tab" data-bs-target="#clasroom-tab-pane" type="button"
        role="tab" aria-controls="clasroom-tab-pane" aria-selected="true">
        Classrooms
      </button>
    </li>
    <li class="nav-item" role="presentation">
      <button class="nav-link btn btn-outline-info" id="teacher-tab"
        data-bs-toggle="tab" data-bs-target="#teacher-tab-pane" type="button"
        role="tab" aria-controls="teacher-tab-pane" aria-selected="false">
        Teachers
      </button>
    </li>
    <li class="nav-item" role="presentation">
      <button class="nav-link btn btn-outline-info" id="student-tab"
        data-bs-toggle="tab" data-bs-target="#student-tab-pane" type="button"
        role="tab" aria-controls="student-tab-pane" aria-selected="false">
        Students
      </button>
    </li>
  </ul>
  <div class="tab-content" id="myTabContent">
    <div class="tab-pane fade show active" id="clasroom-tab-pane"
      role="tabpanel" aria-labelledby="clasroom-tab" tabindex="0">
      @include('layouts.data.classroom-data')
    </div>
    <div class="tab-pane fade" id="teacher-tab-pane" role="tabpanel"
      aria-labelledby="teacher-tab" tabindex="0">
      @include('layouts.data.teacher-data')
    </div>
    <div class="tab-pane fade" id="student-tab-pane" role="tabpanel"
      aria-labelledby="student-tab" tabindex="0">
      @include('layouts.data.student-data')
    </div>
  </div>
</div>

```

Ispis 7. admin.main.blade.php sekcija pogleda za navigacijsku listu



### 2.3.2. Uloga učitelj

Svaki učitelj je član škole. Učitelj može dodavati nove zadaće s neograničenim brojem pitanja, dodjeljivati ih učenicima na rješavanje, pregledavati riješene zadaće učenika i pratiti njihove rezultate. Također, učitelj ima mogućnost pozivanja novih učenika u razrede kojima upravlja.

Registracija učitelja zahtijeva uz podatke o korisniku i pristupni kôd dobiven od administratora. Pri obradi zahtjeva za registraciju, provjerava se ispravnost podataka u Request komponenti (Ispis 8.). Ista komponenta koristi se i za registraciju učenika.

```
public function rules() {
    $groups = Group::all();
    $codes = $groups->pluck('membership_code')->toArray();
    return [
        'firstName' => ['required', 'string', 'max:100', 'min:2'],
        'lastName' => ['required', 'string', 'max:100', 'min:2'],
        'email' => ['required', 'string', 'max:150', 'email', 'unique:users'],
        'password' => ['required', 'max:100', 'min:10'],
        'confirmPassword' => ['required', 'required_with:password', 'same:password'],
        'membershipCode' => ['required', 'string', 'max:10', Rule::in($codes)],
    ];
}
```

Ispis 8. funkcija `rules()` iz `StudentTeacherRequest` datoteke koja služi za provjeru unesenih podataka u registracijsku formu učitelja i učenika

Autentikacija učitelja ista je kao i za administratora te su stranice učitelja također zaštićene. Radna ploča učitelja (Slika 10.) je ponešto drugačija te uz navigacijsko okno sadrži korisničke podatke s dva dugmeta s poveznicama za forme za ažuriranje profila i promjenu lozinke, dugmad za dodavanje nove zadaće (engl. *create new assignment*) i pozivanje novih učenika (engl. *send an invite to students*) te navigacijske liste. Lista sadrži kartice: tablicu zadaća (engl. *assignments*), grafički prikaz rezultata zadaća po mjesecima (engl. *assignments by months*), tablicu rješenja zadaća (engl. *submissions*), tablicu učenika (engl. *students*) i tablicu razreda (engl. *classrooms*).

TaskTastic

DASHBOARD LOGOUT

Teacher

Last updated 11 days ago

Ana Sirić

anasiric@gmail.com

Edit details Change password

CREATE NEW ASSIGNMENT

SEND AN INVITE TO STUDENTS

Assignments Assignments by months Submissions Classrooms Students

List of all assignments

10 entries per page Search...

Title	Deadline	Assigned to	Solved	Pending	Average score	Assign to	Delete
test	2023-07-28 12:00:00	3	0	3	0		
TEST 1	2023-08-30 12:00:00	1	0	1	0		
TEST 3	2023-08-29 15:00:00	1	0	1	0		
test 4	2023-08-23 12:00:00	0	0	0	0		
test 5	2023-08-25 17:00:00	1	1	0	2		

Showing 1 to 5 of 5 entries

Deadline in more than 3 days

Deadline in less than 3 days

Deadline passed

© 2023 Split, Croatia  
creator ivana Čavka, an IT student at University Department of Professional Studies  
github repository

Slika 10. Radna ploča učitelja škole u „TaskTastic“ aplikaciji (proširen pregled)

Kako bi zadaće koristili i njima upravljali, naravno, treba ih i stvoriti. Za to služi `assignment-form.blade.php` pogled povezan s funkcijom `saveAssignment` u upravljaču za model zadaće, a učitelj je na njega preusmjeren klikom na dugme stvori novu zadaću (engl. `create new assignment`). U novu zadaću trebaju se prvo unijeti naslov (engl. `title`) i glavne upute (engl. `description`) te odabrati krajnji rok predaje (engl. `deadline`). Unos datuma omogućen je korištenjem padajućeg izbornika za izbor datuma i vremena u kalendarskom prikazu. Pri provjeri unesenih podataka u upravljaču se provjerava i je li uneseni datum već prošao prije trenutka izrade zadaće, kako bi se osigurano pravilan izbor samo nadolazećih rokova. Nakon odlomka s podacima za zadaću, slijedi odlomak s dinamičkim dodavanjem pitanja. Za početak dostupno je samo jedno pitanje, a nakon njega nova pitanja dodaju se po želji klikom na dugme s oznakom „+“.

Na klik dugmeta, cijeli skup podataka za pitanje se stvara i dodaje ispred prethodnog. Dodavanje novog pitanja omogućeno je JavaScript kôdom (Ispis 9.) koji također prilagođava atribut imena HTML komponente prateći broj nastavnih pitanja, kako ne bi došlo do miješanja podataka. Broj pitanja nije ograničen. Za svako pitanje potrebno je ispuniti polja s tipom pitanja, s tekstom pitanja, najvećim mogućim brojem bodova, pomoćnim tekstom (engl. *hint*), brojem bodova koji se oduzimaju na korištenje pomoći i prilaganjem datoteke ako je potrebna (slika ili dokument). Tipovi pitanja definirani su u modelu pitanja, te mogu biti pitanje s tekstualnim odgovorom, brojčanim odgovorom, prilaganjem datoteke na pregled, odabira jedne opcije od tri ponuđene ili odabira više opcija od tri ponuđene. U slučaju da učitelj odabere tip s opcijama odabira, prikazuju se inače skriveni unosi svake od tri opcije (A, B ili C).

```
const plus = document.getElementById('plus');
const parent = document.getElementById('parent');
let numOfQuestions = document.getElementById('numOfQuestions');
let number = 1;
numOfQuestions.value = number;
document.getElementById("questionNo").innerHTML = 'No. ' + number;
plus.addEventListener('click', function() {
    number += 1;
    const questions = document.getElementsByName('questions');
    let temp = questions[0].cloneNode(true); //klonira prvi
    const inputs = temp.getElementsByClassName('form-control');
    for (let i = 0; i < inputs.length; i++) {
        inputs[i].value = null;
    }
    numOfQuestions.value = number;
    parent.insertBefore(temp, questions[0]);
    document.getElementById("questionNo").innerHTML = 'No. ' + number;
    for (let j = 0; j < questions.length; j++) {
        const tabs = questions[j].getElementsByClassName('form-control');
        for (let i = 0; i < tabs.length; i++) {
            let name = tabs[i].getAttribute('name');
            let newName = name.replace(/\\d+/g, j + 1);
            tabs[i].setAttribute('name', newName);
        }
    }
})
```

Ispis 9. Ulomak JavaScript kôda u `assignment-form.blade.php` pogledu koji ostvaruje dinamičko dodavanje pitanja u formu zadaće

Klikom na dugme za spremanje zadaće, svi navedeni podaci šalju se upravljaču koji ih provjerava, zbraja ukupan broj bodova te sve unosi u bazu. U bazi se stvara nova zadaća, svaki od unesenih pitanja koji se s bazom pritom i vežu, a priloženi dokument sprema se u `public/storage` mapu. Za spremanje dokumenta, ali i slika općenito (slike profila), koristi se Laravelova naredba `store()` kojoj su potrebne samo tekstualni zapis adrese s koje se dobavlja dokument te pozicija u datotečnom sustavu u koju će se dokument spremiti.

Navedena funkcija kao rezultat ispisuje novu adresu spremljenog dokumenta te se ona upisuje u bazu u atribut zadaće nazvan datoteka (engl. *file*).

Lista zadaća uz opcije pretraživanja, sortiranja po svakom stupcu tablice i grupiranja zadaće u stranice nudi dugmad za brisanje zadaća (engl. *delete*) i dodjeljivanje zadaća studentima (engl. *assign to*) te podatke o tome koliko je studenata zadaća dodijeljena (engl. *assigned to*), koliko je studenata predalo zadaću (engl. *solved*), koliko ih još mora riješiti (engl. *pending*) i koji je prosječni rezultat zadaće (engl. *average score*). Zadaće kojima je prošao rok predaje (engl. *deadline*), ne mogu se više dijeliti te je njihova dugmad crvena. Zadaće kojima ubrzo prolazi rok predaje (točnije, za tri dana) imaju žutu dugmad, a one za koje ima više vremena za rješavanje imaju plavu dugmad. Navedene boje i njihovo značenje pojašnjene su ispod tablice u legendi.

Dugme u obliku papirnato aviona predstavlja preusmjeravanje na formu dodjeljivanja zadaće studentima. Prilikom preusmjeravanja, na formu se šalje i parametar `§assignment->id` kako bi se određena zadaća mogla povezati sa studentima koji su izabrani. U formi je moguće izabrati više ponuđenih studenata preko unosa višestrukog izbora (engl. *checkbox*), od kojih svi pripadaju razredima učitelja koji je prijavljen te su tako i kategorizirani po razredima. Učitelj može izabrati sve učenike razreda klikom na kućicu željenog razreda ili pojedinačno, a ako odabere sve učenike nekog razreda, razred će se automatski označiti kao odabran. Navedena forma izrađena je dinamički u Blade petlji zbog fleksibilnosti odabira pri različitim brojevima studenata i razreda, a pozadinski kôd za manipulaciju odabira napisan je kao skripta pogleda u JavaScript jeziku (Ispis 10.).

```
<script>
  var allClassrooms = document.getElementsByName('classroom');
  var checkboxes = document.querySelectorAll('input.subOption');
  checkboxes.forEach(checkbox => checkbox.addEventListener('click', function() {
    var id = checkbox.id.match(/\d+/g);
    var classroom = document.getElementById(id[0]);
    var divClassroom = document.getElementById('class'+classroom.id);
    var checkboxGroup = divClassroom.getElementsByClassName('subOption');
    var checkedCount = 0;
```

```

for(let i = 0; i < checkboxGroup.length; i++) {
    if(checkboxGroup[i].checked) {
        checkedCount++;
    }
}
classroom.checked = (checkedCount == checkboxGroup.length);
classroom.indeterminate = checkedCount>0 && checkedCount<checkboxGroup.length;
});
allClassrooms.forEach(classroom => classroom.addEventListener('click', function() {
    var divClassroom = document.getElementById('class'+classroom.id);
    var checkboxesPerClassroom = divClassroom.getElementsByClassName('subOption');
    if (checkboxesPerClassroom != null) {
        for (var i = 0; i < checkboxesPerClassroom.length; i++) {
            checkboxesPerClassroom[i].checked = classroom.checked;
        }
    }
}));
</script>

```

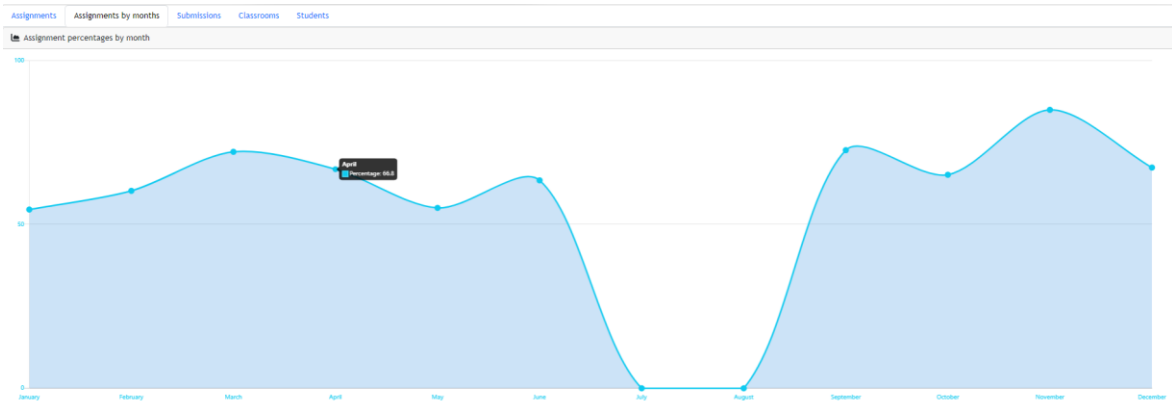
Ispis 10. Ulomak `assign-students.blade.php` pogleda koji sadži JavaScript kôd za manipulaciju odabirom

Pri podnošenju forme, zahtjev se preusmjeri na upravljački dio koji izrađuje nove veze učenika i zadaće u međutablici `assignment_user`. Također, automatski se šalje email pošta na adresu studenata kako bi dobili obavijest o dodijeljenoj zadaći. Mail komponenta izrađena je kao i ona kod administratora te se njen proces odrađuje u redu čekanja (engl. *queue*).

Brisanje pojedine zadaće omogućeno je klikom na dugme prikazan sličicom kantice za smeće. Nakon uspješnog brisanja u bazi, zajedno sa svim pripadnim vezama uz odabranu zadaću, korisnika se vraća na radnu ploču s porukom o uspješnom brisanju.

Statusi zadaća (broj studenata kojima su dodijeljene, broj zadaća koje su riješene, broj zadaća koje su preostale) obrađena je većinski pomoću funkcije `count()` koja se vrši nad kolekcijama podataka dohvaćenih iz baze povezanih s dotičnom zadaćom. Zahvaljujući funkcijama koje u svakom od modela dohvaćaju pripadne podatke iz tabličnih veza (`hasOne()`, `belongsTo()`, `hasMany()`, `belongsToMany()`) vrlo je lako dohvatiti povezane podatke i na samom pogledu u službi atributa objekta kojeg se prosljeđuje na stranicu. U ovom konkretnom slučaju, `$assignment->submissions->count()` je primjer u kojem se kolekcija predaja zadaća, engl. *submissions*, dohvaćena u modelu uz pomoć `hasMany()` funkcije može prebrojati i jednom naredbom dobiti gotov podatak, odnosno broj predanih zadaća.

Kartica s grafičkim prikazom sveukupnih rezultata zadaća (Slika 11.) po mjesecima izrađena je također uz pomoć <http://www.startbootstrap.com> predložka. Kôd koji manipulira grafom ispisan je u JavaScriptu. Podaci su uneseni u graf kao niz izračunatih vrijednosti (u postocima zbog uniformnosti podataka neovisno o broju bodova) te se mapiraju u niz od 12 mjeseci kroz godinu. Na prelazak pokazivača preko točke koja predstavlja podatak za dani mjesec, prikazuje se točan iznos uz naziv mjeseca.



Slika 11. Grački prikaz rezultata po mjesecima

Kartica predanih zadaća (engl. *submissions*) sadrži tablicu s naslovom zadaće, rokom predaje, datumom predaje (engl. *submission date*), statusom zadaće (pregledana ili predana), ukupnim brojem bodova (engl. *total points*), prosječnim rezultatom (engl. *average score*) i dugmetom za pregledavanje predane zadaće (engl. *review*). Ukupni broj bodova i prosječni rezultat izračunati su kao i u prethodnim karticama, a status zadaće mijenja se iz predana (engl. *turned in*) u procesu spremanja ispravka u pregledana (engl. *reviewed*). Dugme za pregled predane zadaće vodi učitelja na pogled s formom za upisivanje dodijeljenog broja bodova i ukupnog komentara na zadaću. Forma sadrži ispis svih pitanja zajedno s odgovorima studenta te informacije o tome je li bila dostupna datoteka za pitanje, je li bila dostupna pomoć pri rješavanju i je li pomoć iskorištena. U slučaju da je pomoć iskorištena, u upravljaču se učeniku automatski oduzima zadani broj bodova neovisno o bodovima koje mu dodjeljuje učitelj, što je i naglašeno na stranici. Predana zadaća može se preuzeti kao pdf dokument klikom na dugme ikonice dokumenta unutar pregleda predaje zadaće, no samo ako je zadaća već pregledana.

Kartica s razredima prikazuje tablicu sa stupcima naziva razreda, brojem učenika, popisom učenika i prosječnim rezultatom svih učenika razreda. Svi se stupci, kao i u drugim tablicama, mogu poredati uzlazno i silazno po svakom stupcu, pretraživati po ključnim riječima i grupirati po stranicama. Kartica s učenicima prikazuje prezime i ime učenika, email adresu, kojem razredu pripada, koliko ima zadanih zadaća, koliko ih je predano te koliko ih je ostalo za riješiti. Navedena tablica može se ispisati u pdf obliku (Slika 12.) klikom na dugme s ikonom pdf dokumenta. Svi navedeni podaci prikupljaju se na isti način metodama `count()` i korištenjem kolekcija kao povratnih vrijednosti funkcija dohvaćanja povezanih podataka iz baze (Ispis 11.).

```
@foreach ($students as $student)
<tr>
    <td>{{ $student->last_name }}</td> //prezime
    <td>{{ $student->first_name }}</td> //ime
    <td>{{ $student->email }}</td> //email
    <td>{{ $student->groups[0]->name }}</td> //razred
    <td>{{ $student->assignments->count() }}</td> //broj zadaća
    <td>{{ $student->submissions->count() }}</td> //broj riješenih zadaća
    <td>{{ $student->assignments->count() - $student->submissions->count() }}</td>
    //broj preostalih zadaća
</tr>
@endforeach
```

Ispis 11. Ulomak `student-data.blade.php` pogleda koji ispisuje redove tablice učenika

Automatska izrada pdf dokumenta omogućena je za sve tablice u aplikaciji pomoću DOMPdf dodatka. DOMPdf je biblioteka koja omogućuje generiranje pdf dokumenata iz HTML i CSS sadržaja, na zahtjev pripadne funkcije u pdf upravljaču (engl. PDFController). Kao i kod Mail komponente, nužno je pripremiti pogled koji će se prikazati zajedno s dodijeljenim parametrima. No, DOMPdf nije dio početnog paketa Laravel projekta, pa ga je potrebno dodati u sustav naredbom `composer require barryvdh/laravel-dompdf` i postavljanjem konfiguracije u `config/app.php` datoteci. U `web.php` datoteci spomenuta funkcija pridružena je adresi na kojoj će biti dostupna, a ista je priložena dugmetu s ikonicom pdf dokumenta. Generirana pdf datoteka se na pritisak dugmeta automatski preuzima u pregledniku. Ime datoteke također se definira u upravljačkoj funkciji (Ispis 12.).

```

public function generateStudentPDF()
{
    $user = User::find(auth()->user()->id);
    $students = array();
    if($user->role == User::teacherRole) {
        $classrooms = Group::where('owner_id', $user->id)->get();
        foreach ($classrooms as $class) {
            foreach ($class->users()->get() as $st)
                array_push($students, $st);
        }
    } elseif ($user->role == User::adminRole) {
        $school = Group::where('owner_id', $user->id)->firstOrFail();
        $teachers = $school->users()->get();
        foreach ($teachers as $teacher) {
            $tempClass = Group::where('owner_id', $teacher->id)->get();
            foreach ($tempClass as $class) {
                foreach ($class->users()->get() as $st)
                    array_push($students, $st);
            }
        }
    }
    $data = [
        'students' => $students,
    ];
    $pdf = PDF::loadView('PDF/studentPDF', $data);
    return $pdf->download('student-data.pdf');
}

```

Ispis 12. Funkcija `PDFController` upravljača zadužena za generiranje pdf dokumenta s tablicom učenika

Last name	First name	Email	Classrooms	Assigned	Solved	Pending
Ani?	Ana	anaanic@gmail.com	PYP1	2	0	2
Mati?	Mate	matematic@gmail.com	PYP1	2	0	2
Ani?	Ante	anteantic@gmail.com	PYP1	1	0	1

Slika 12. Prikaz pdf ispisa liste učenika

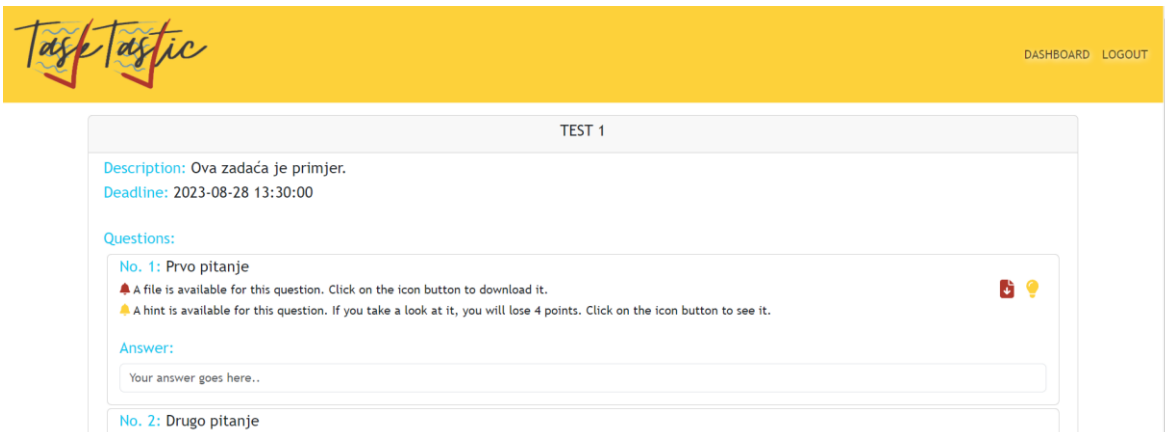


### 2.3.3. Uloga učenik

Svaki učenik može pregledavati zadatke koje su mu dodijeljene, rješavati ih do određenog roka predaje i pratiti svoje rezultate, bilo samo za sebe ili u odnosu na prosječni rezultat.

Registracija učenika ima iste komponente kao i registracija učitelja, osim što uneseni pristupni kôd pripada razrednoj grupi. Nakon uspješne registracije, potvrde email adrese i prijave u aplikaciju, učenik pristupa svojoj radnoj ploči. Učenička radna ploča sadrži karticu profila s dugmadi za promjenu lozinke i uređivanje detalja, kao što su i one kod učitelja i administratora, te također navigacijsku listu s karticama: zadatke (engl. *assignments*), predane zadatke (engl. *submissions*) i kartica s grafičkim prikazom uspjeha kroz mjesece (engl. *assignments by months*).

Kartica zadataka sadrži tablicu sa stupcima naslov zadatke (engl. *title*), rok predaje (engl. *deadline*), status zadatke (engl. *status*), osobni rezultat (engl. *my score*), prosječni rezultat (engl. *average score*) te dugme za otvaranje zadatke (engl. *open*, Slika 13.). Na otvaranje zadatke prikazuje se stranica s formom za rješavanje zadatke, sa svim pripadnim pitanjima i podacima. S desne strane zaglavlja svakog pitanja dostupna je dugmad za pomoć i preuzimanje datoteke ako su dostupni za navedeno pitanje. Oblici odgovora prilagođeni su tipu pitanja tako što su unosi oblika tekst (engl. *input type="text"*), broj (engl. *input type="number"*), jednostruki (engl. *radio*) ili višestruki odabir (engl. *checkbox*). Dugmad za otvaranje zadataka kodirani su bojom – oni koji se mogu predati u roku tri dana su žute boje, oni koji imaju više vremena do roka od tri dana su plave boje i oni kojima je rok prošao su crvene boje i onemogućeni za klik.



Slika 13. Dio pogleda u kojem učenici upisuju odgovore na pitanja

Kartica predanih zadaća (engl. *submissions*) sadrži tablicu s naslovom zadaće, rokom predaje, datumom predaje (engl. *submission date*), statusom zadaće (pregledana ili predana), ukupnim brojem bodova (engl. *total points*), prosječnim rezultatom (engl. *average score*) i dugmetom za pregledavanje predane zadaće (engl. *review*). Obje spomenute tablice mogu se ispisati u pdf obliku, dok se i predana zadaća može preuzeti kao pdf dokument klikom na dugme ikonice dokumenta unutar pregleda predaje zadaće, no samo ako je zadaća već pregledana.

Kartica s grafičkim prikazom uspjeha kroz mjesece (engl. *assignments by months*) organizirana je i sastavljena na isti način kao i kod učitelja, uključujući i liniju za prikaz podataka koji se odnose isključivo na učenika. Na taj način učenik može pratiti odnos svog rezultata i prosječnog rezultata svih onih koji su rješavali zadaću.

## Zaključak

Tehnologije za razvoj aplikacije pokazale su se kao izvrstan odabir. Laravel kao razvojno okruženje je maksimalno prilagođen brzom izradi platforme svojom intuitivnošću sintakse, velikom paletom biblioteka i mogućnostima različitih alata. Predznanje PHP-a nije nužno, kako PHP ima obilježja jako slična drugim jezicima kao što su C, Perl ili JavaScript. Korištenjem samo nekoliko naredbi Composer i/ili Artisan sučelja može se bez problema dohvatiti bilo koji paket za rad. Kada se još uzme u obzir da je sama instalacija okruženja, pogotovo uz XAMPP platformu, rješiva u samo tri koraka, postaje očito da će se Laravel još neko vrijeme zadržati na vrhu ljestvice tehnologija za izradu internet aplikacija. S Laravelovom kompozicijom, a u podlozi PHP-om, korisniku je čak i nametnuta izrada čistog, čitkog, dosljednog i strukturiranog kôda. Tako dobar sustav omogućen je prvenstveno zaslugom zajednice velikog broja programera koji pridonose njegovom rastu i pružaju podršku putem foruma, blogova i društvenih medija. Time se održava neprekidan razvoj u korak svim potrebama, ali i novim mogućnostima koje se pojavljuju iz dana u dan.

„TaskTastic“ aplikacija ispunjava glavnu motivacijsku ideju direktnog pristupa učenika i učitelja zadaćama i njihovim rezultatima, a sve unutar zatvorene zajednice škole. Svakako ima područja koja bi se dodatno mogla razraditi i unaprijediti kako bi platforma bila još više prilagođena potrebama korisnika u stvarnom životu. Primjerice, mogu se unijeti predlošci pitanja koji bi se mogli dodati direktno u zadaću bez ponovnog ispisivanja. Također, mogle bi se dodati kategorije zadaća po, recimo, predmetima ili polugodištu. Učenicima bi zadaće mogle biti i zabavnije za rješavanje ako bi im se omogućilo rješavanje dodatnih zadataka za dodatne bodove. Za poticanje motivacijske atmosfere, učiteljima bi se mogle nuditi gotove poruke pozitivnog tona („Odličan posao!“, „Sve si bolji/bolja!“, „Potrudi se malo više. Ti to možeš!“, „Prouči gradivo još jednom, i zatraži pomoć na nastavi, ako ti treba.“ ...). Što se tiče građe aplikacije, za prilagodbu široj dostupnosti, uz potencijalnu nadogradnju poslužitelja i baze, trebalo bi provesti i određena testiranja usmjerena na održavanje velikog broja korisnika i zahtjeva. Također, opcija je i poboljšati sigurnost korisnika korištenjem sustava većeg stupnja sigurnosti.

## Literatura

- [1] ASTech, „The Future of PHP: A Closer Look at Its Evolution and Relevance“, <https://www.linkedin.com/pulse/future-php-closer-look-its-evolution-relevance> (posjećeno 11.8.2023.)
  
- [2] „PHP Manual“, <https://www.php.net/manual/en/> (posjećeno 20.7.2023.)
  
- [3] „Laravel documentation“, <https://laravel.com/docs/10.x> (posjećeno 12.7.2023.)
  
- [4] Arnel, J., „Web application development with Laravel PHP Framework version 4“: Helsinki Metropolia University of Applied Sciences, 2014.
  
- [5] „Introduction to Laravel and MVC Framework“, <https://www.geeksforgeeks.org/introduction-to-laravel-and-mvc-framework/> (posjećeno 11.8.2023.)
  
- [6] „Laravel 8 Generate PDF File using DomPDF“, <https://www.itsolutionstuff.com/post/laravel-8-pdf-laravel-8-generate-pdf-file-using-dompdfexample.html> (posjećeno 18.8.2023.)