

AKADEMIJA TEHNIČKO-UMETNIČKIH STRUKOVNIH STUDIJA BEOGRAD -
ODSEK VISOKA ŠKOLA ZA INFORMACIONE I KOMUNIKACIONE TEHNOLOGIJE

INTERNET TEHNOLOGIJE

WEB APLIKACIJA U LARAVEL FRAMEWORK-U NAMENJENA UČENJU STRANOG JEZIKA

Završni rad

Mentor:

dr Nenad Kojić, dipl. inž.

Student:

Teodora Nedeljković, 5/18

Beograd, 2022.

AKADEMIJA TEHNIČKO-UMETNIČKIH STRUKOVNIH STUDIJA BEOGRAD -
ODSEK VISOKA ŠKOLA ZA INFORMACIONE I KOMUNIKACIONE TEHNOLOGIJE

INTERNET TEHNOLOGIJE

Predmet: Web programiranje PHP 2

Tema: Web aplikacija u Laravel framework-u namenjena učenju stranog jezika

Ocena ____ (_____)

Članovi komisije:

mr Natalija Vugdelija

Milena Čančarević

dr Nenad Kojić

Sadržaj

1. Uvod	5
2. Radno okruženje	6
3. Organizacija	7
3.1. Struktura projekta	7
4. Stranice i funkcionalnosti	11
4.4.1 Index	11
4.4.2 Words	12
4.4.3 Register	13
4.4.4 Login	14
4.4.5 About	15
4.4.6 Practice	15
4.4.7 Task/id	17
4.4.8 Finished Challenge	19
4.4.9 Task info	19
4.4.10 Profile	20
4.4.11 Edit Profile	20
4.4.12 Change Password	21
4.4.13 Change Daily Goal	21
4.4.14 Admin/Home Page	22
4.4.15 Admin/Insert a Word	22
4.4.16 Admin/Words	23
4.4.17 Admin/Edit a Word	23
4.4.18 Admin/Sentences	24
4.4.19 Admin/Insert a Sentence	24
4.4.20 Admin/Edit a Sentence	25
4.4.21 Admin/Task Info	25
4.4.22 Admin/Task Info Edit	26
4.4.23 Admin/Login	26
5. Kodovi	27
5.1 PHP	27
5.1.1 Rute	27
5.1.2 Kontroleri	28
5.1.3 Modeli	62

5.1.4 Middleware	69
5.1.5 Requests.....	70
5.1.6 Seed-ovi.....	76
5.1.7 Views	80
5.2 JS	138
5.2.1 main.js.....	138
5.2.2 practice.js	145
5.2.3 register-page.js	145
5.2.4 select-the-matching-pairs.js	148
5.2.5 select-the-missing-word.js	151
5.2.6 write-this-in-english.js	152
5.3 CSS.....	153
5.3.1 style.css	153
6. Baza podataka	196
6.1 Šema baze podataka	196
7. Zaključak	197
8. Literatura	198

1. Uvod

Svetska globalizacija dovela je čovečanstvo do neophodnosti učenja novih jezika. Ova tendencija izazvala je iskru interesovanja za nove načine učenja. Ispostavilo se da su aplikacije za učenje jezika među najpopularnijim takvim načinima.

Namena web aplikacije „Lapsus Linguae“ je naučiti osnove za sporazumevanje na italijanskom jeziku kroz vežbe, i takođe birajući jezik preko koga će to činiti – srpski ili engleski. Lekcije se sastoje od tekstualnih zadataka tri tipa: izabrati reč koja nedostaje, povezivanje odgovarajućih parova i prevod date rečenice.

Autorizovani korisnik ima pristup vežbama (kvizovi) kroz 3 nivoa koja će se hijerarhijski otključavati od najlakšeg do najtežeg – beginner, intermediate i advanced. Svaka oblast će, pre započinjanja vežbe, korisniku dati opciju čitanja gramatike i vokabulara koja ga očekuje u vežbi. Postoji i opcija da se pogledaju i pretraže reči koje korisnika mogu očekivati u vežbama, a mogu se pogledati i ako korisnik nije autorizovan. Cilj je pružiti svima opciju za učenje tj. proširenje vokabulara na italijanskom jeziku.

Admin ima kompletan pristup i pravo modifikacije nivoa kao i vežbi unutar njih. On može pristupiti statistici vezanoj za rađene vežbe tog dana, registrovanim i ulogovanim korisnicima i ostalim informacijama.

2. Radno okruženje

U toku izrade sajta korišćeno je razvojno okruženje:

- *PhpStorm*;
PhpStorm je uređivač koda u kome je aplikacija napisana.
- *Xampp*;
Xampp je najpopularnije okruženje za razvoj PHP-a. tj. lokalni server koji se koristi za hostovanje PHP veb stranice i prihvaćen je i u Windows-u i Linux-u.
- *PhpMyAdmin*;
PhpMyAdmin je besplatni softverski alat napisan u PHP-u, namenjen za upravljanje MySQL-om preko Veba.

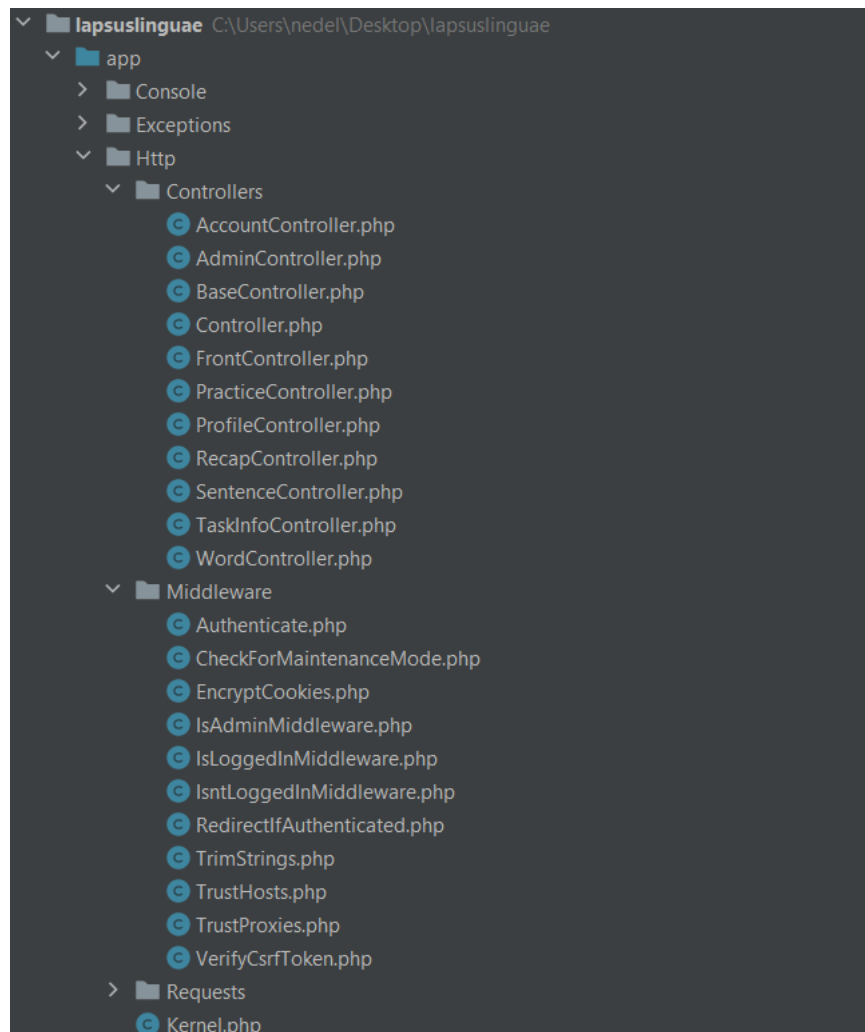
Programski jezici koji su korišćeni za izradu aplikacije:

- *Laravel*;
Laravel je besplatni PHP veb framework otvorenog koda, koji je namenjen razvoju veb aplikacija koje prate arhitektonski obrazac model–pogled–kontroler (MVC) i zasnovan na Symfony-ju.
- *JavaScript*;
JavaScript je programski jezik koji je jedna od osnovnih tehnologija World Wide Web-a, pored HTML-a i CSS-a.
- *jQuery*;
jQuery je JavaScript biblioteka dizajnirana da pojednostavi rukovanje HTML stablom, kao i rukovanje događajima, CSS animaciju i Ajax.
- *HTML5*;
Takozvani jezik za označavanje hiper tekstova, to jest *HTML (Hyper Text Markup Language)*
- *CSS3*;
CSS je jezik formatiranja pomoću kog se definiše izgled elemenata web-stranice.

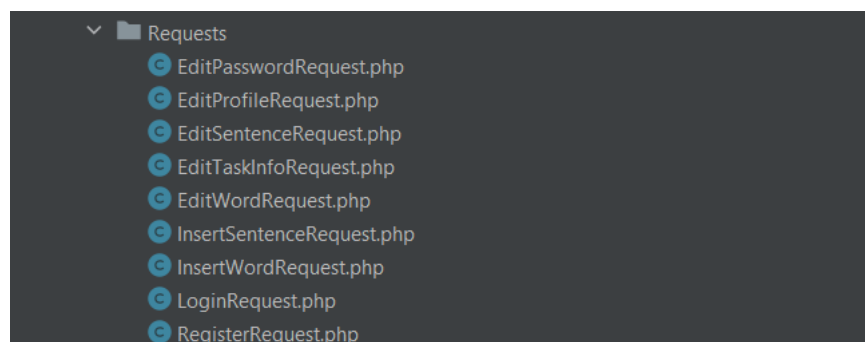
3. Organizacija

3.1. Struktura projekta

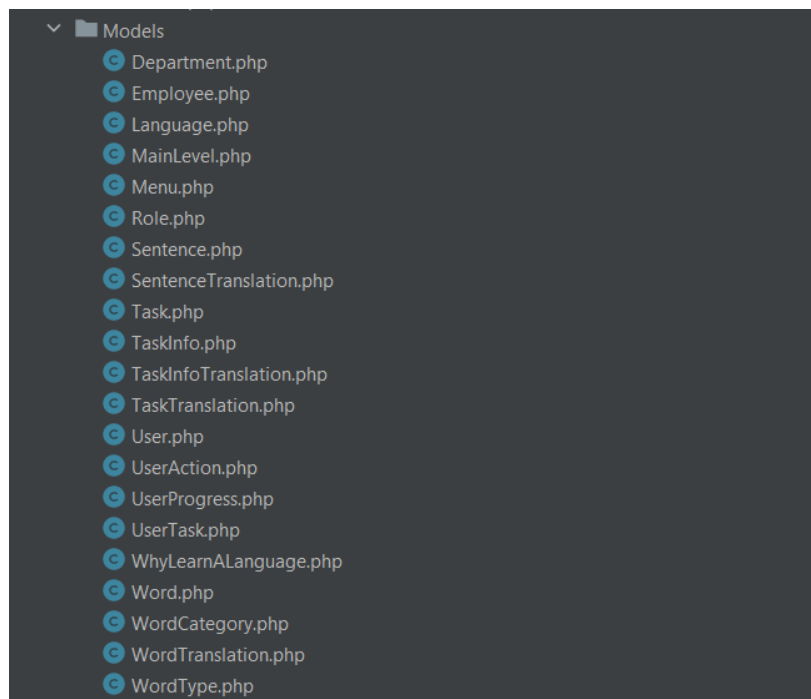
-Struktura projekta, foldera i fajlova unutar njega.



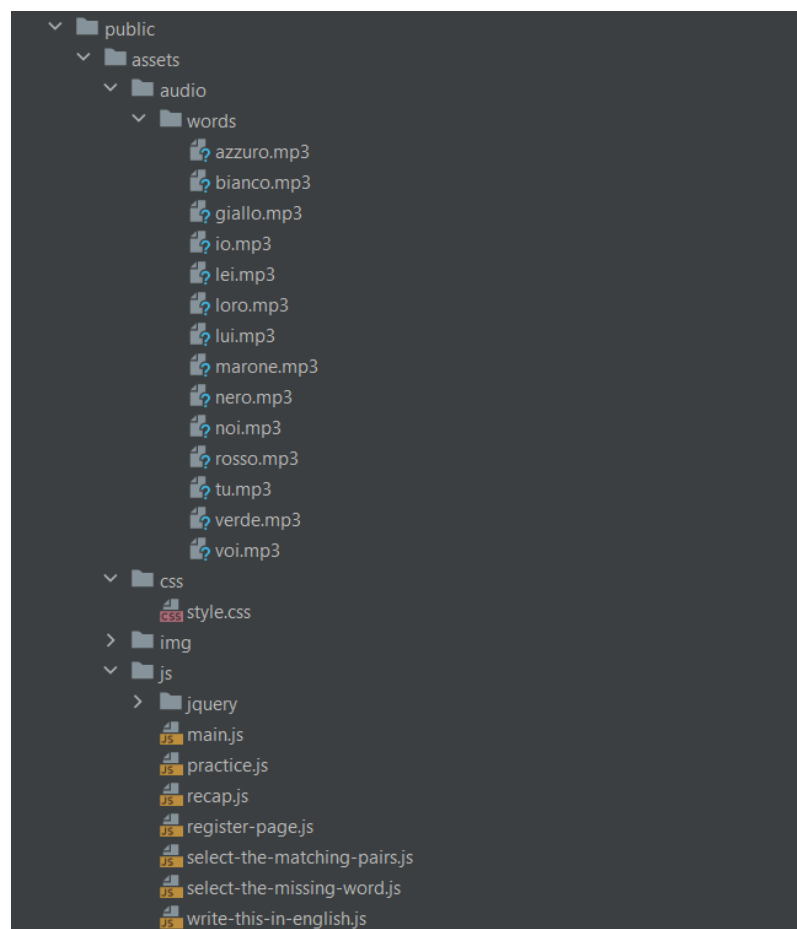
Slika 1 – Prikaz foldera Controllers i Middleware



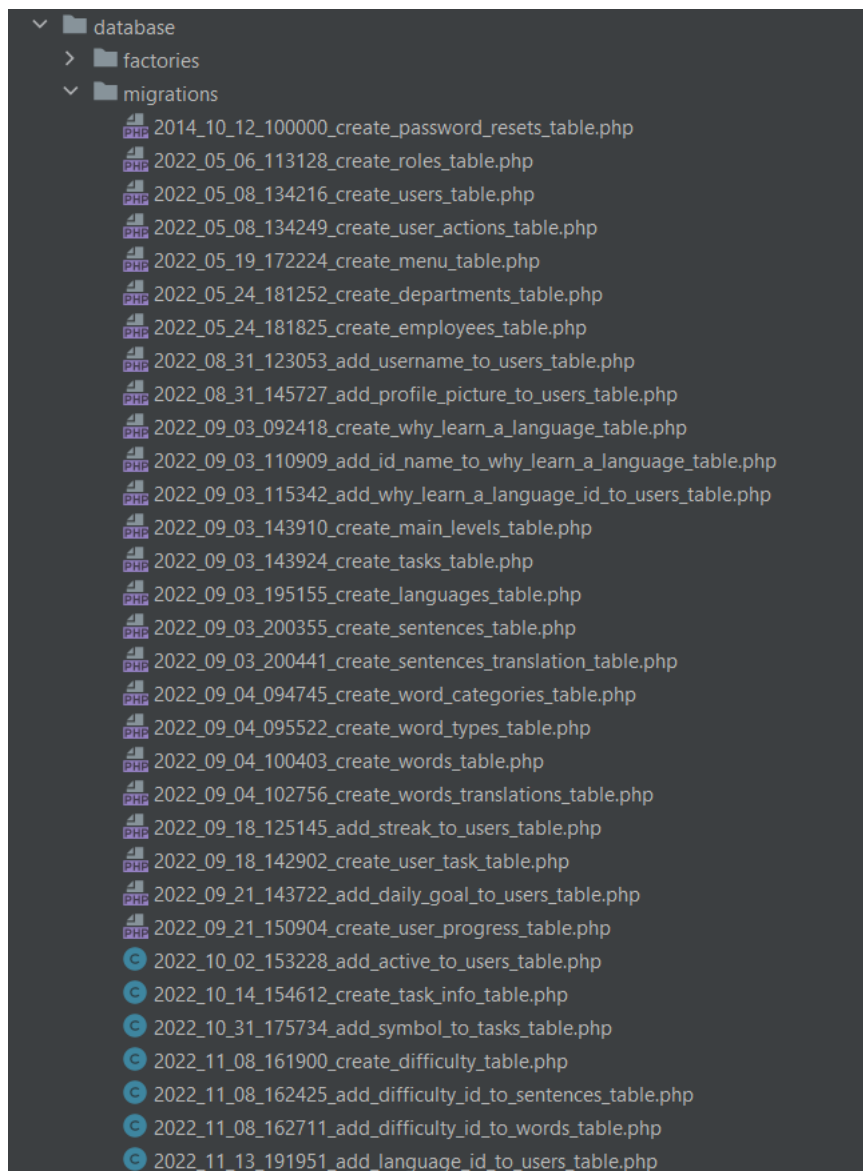
Slika 2 - Prikaz foldera Requests



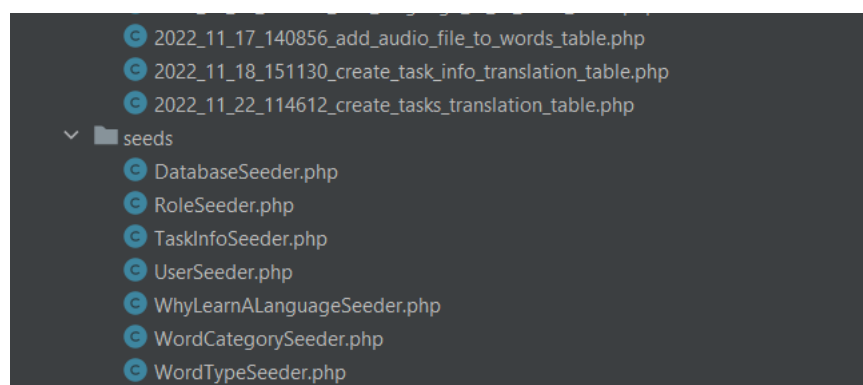
Slika 3 – Prikaz foldera Models



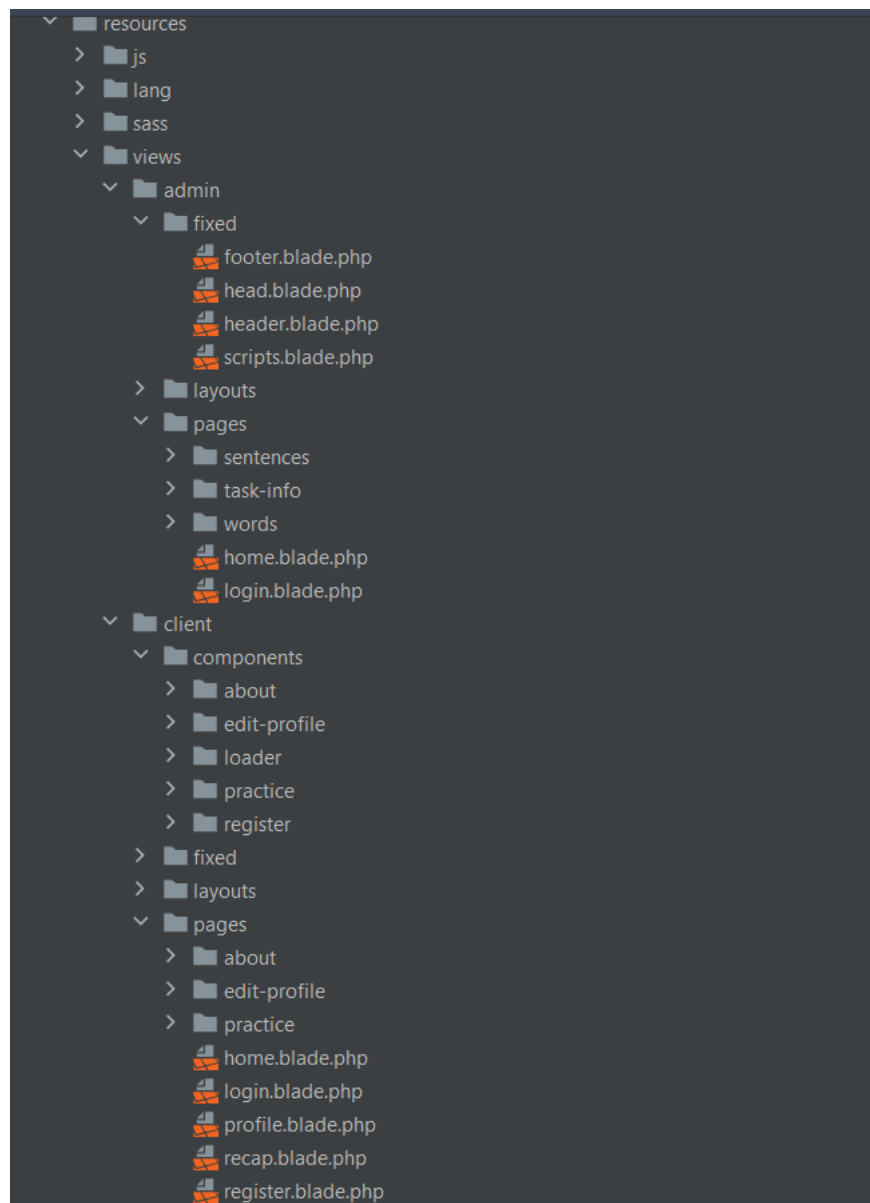
Slika 4 – Prikaz foldera public



Slika 5 – Prikaz foldera migrations unutar foldera database



Slika 6 – Prikaz foldera seeds unutar foldera database

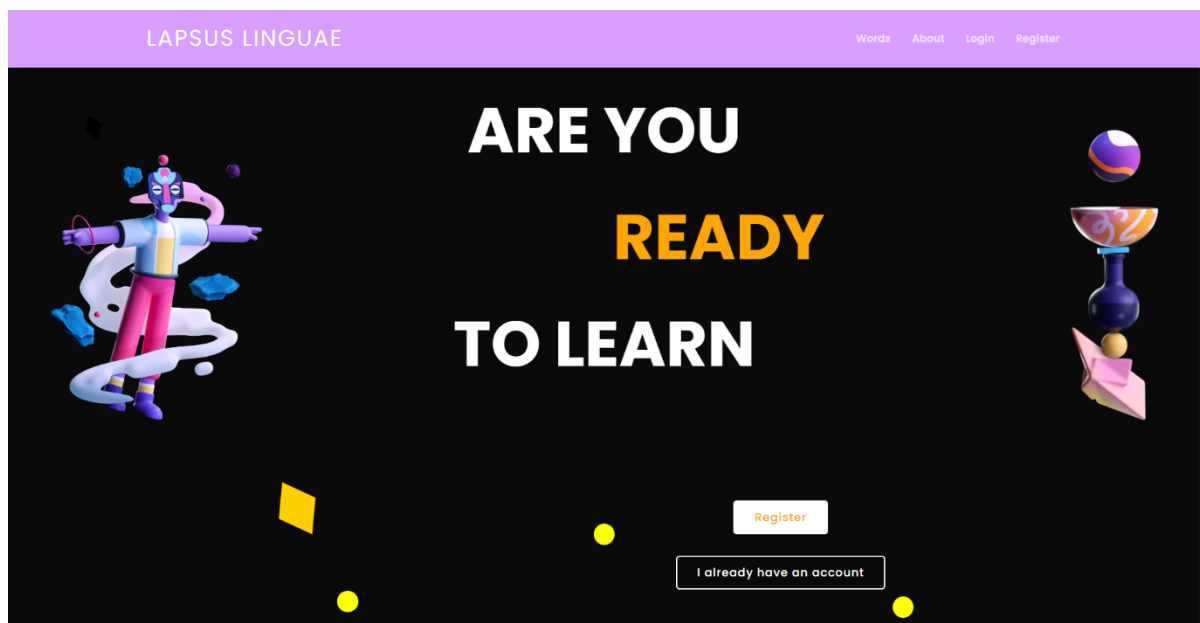


Slika 7 – Prikaz foldera resources

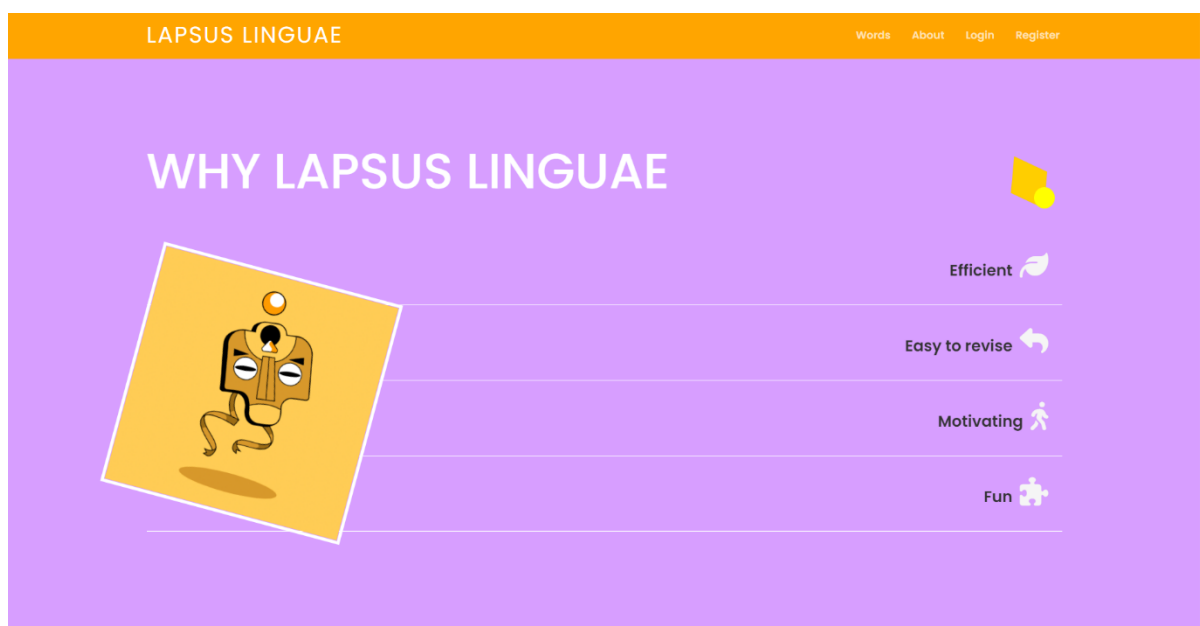
4. Stranice i funkcionalnosti

4.4.1 Index

-Početna stranica koju vide svi korisnici. Dugmad za logovanje tj. registraciju su odmah vidljiva.



Slika 8 - Početna index stranica pr. 1



Slika 9 - Početna index stranica pr. 2

4.4.2 Words

-Strana na kojoj svi korisnici mogu videti tabelu italijanskim reči i njihovih prevoda. Dostupan je audio izgovor svake reči. One se mogu pretražiti preko naziva i kategorije kojoj pripadaju.











Learn and remember italian words

Search words

Select category

▼

Search

Italian	English	Audio
io	I	
tu	you	
lei	she	
lui	he	
noi	we	
voi	you	
loro	they	
rosso	red	
azzurro	blue	
verde	green	

◀

1

2

3

4

5

6

7

8

9

10

...

20

21

▶

Slika 10 - Stranica za reči

4.4.3 Register

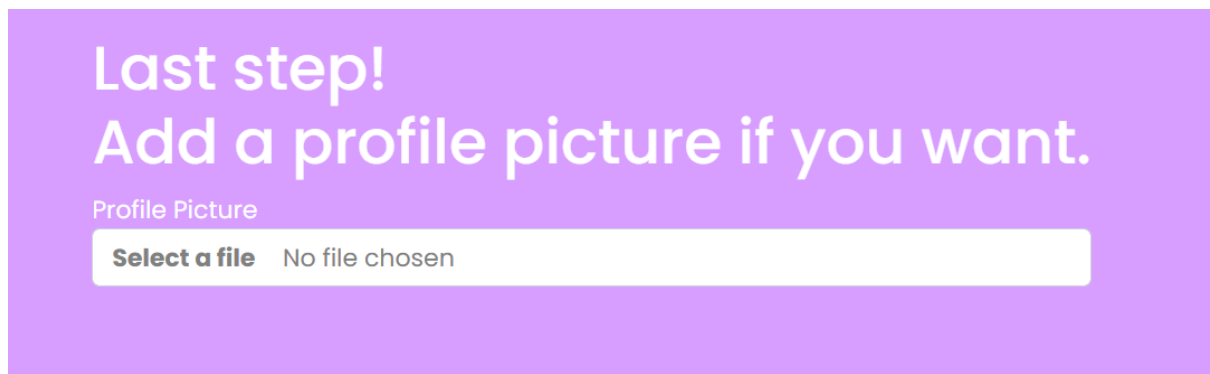
-Realizovano kroz tri koraka. Prvi korak je radi statistike pitanje zašto je korisnik odlučio da uči jezik. Drugi korak je popunjavanje neophodnih informacija za njegov profil. Zatim treći korak nije obavezan i podrazumeva dodavanje profilne slike.



Slika 11 - Register stranica pr. 1

The image shows the second step of the registration process, the 'Register' form. The title 'Register' is at the top. The form contains the following fields: 'First Name' (empty), 'Last Name' (empty), 'Username' (filled with 'admin'), 'Email' (empty), 'Password' (filled with six dots), and 'Your date of birth' (filled with 'dd-----yyyy' and a calendar icon). Below the email field is a text line: 'We'll never share your email with anyone else.'

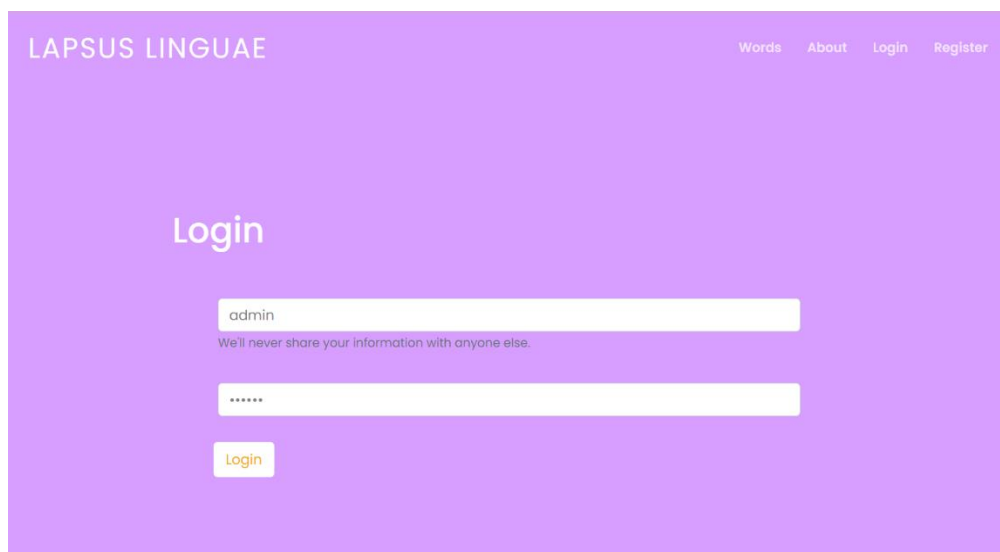
Slika 12 - Register stranica pr. 2



Slika 13 - Register stranica pr. 3

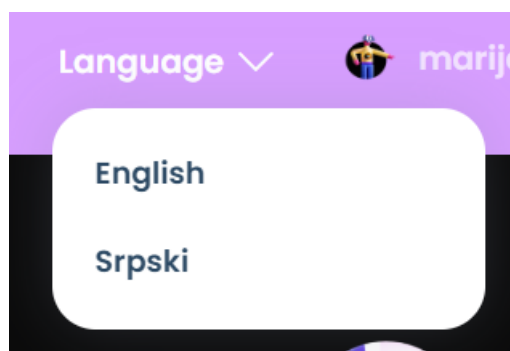
4.4.4 Login

-Logovanje korisnika koje se radi ili pomoću korisničkog imena ili imejla.



Slika 14 - Login stranica

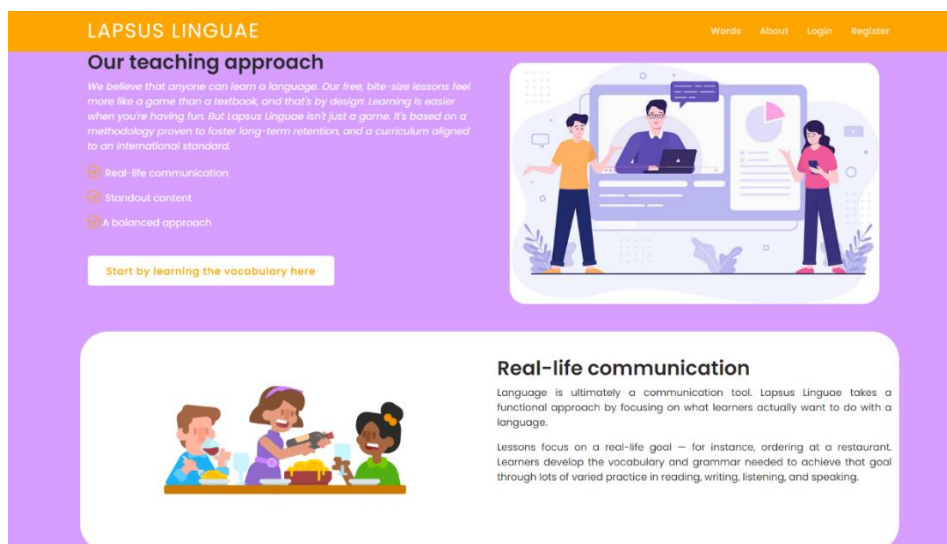
Nakon što je korisnik ulogovan izlazi taster za promenu jezika sajta, i jezika preko koga će se učiti italijanski:



Slika 15 - Taster za promenu jezika sajta

4.4.5 About

-Opis i tekst za upoznavanje sa projektom gde korisnik može saznati njegov pristup i misiju.

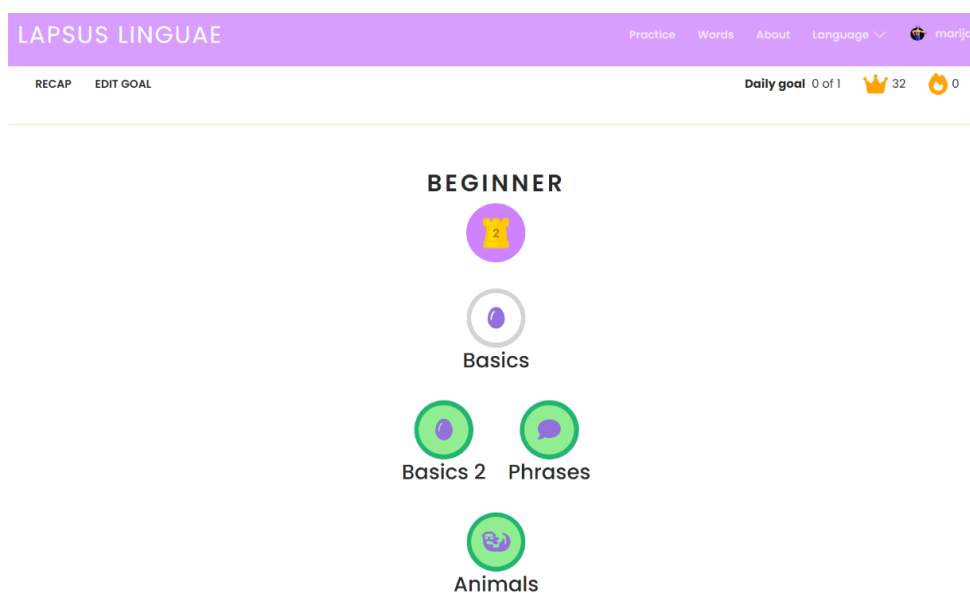


Slika 16 - About stranica

4.4.6 Practice

-Stranica dostupna samo autorizovanim korisnicima. Prikaz svih oblasti koje korisnik može da radi. Navigacioni meni ispod headera sadrži informacije o dnevnom cilju korisnika i da li ga je ispunio. Sadrži informaciju u broju ukupno urađenih vežbi, kao i streaku. Ako korisnik radi vežbu bez propuštanja dana, streak se svaki dan povećava za 1. Ako propusti dan, on se resetuje. Oblasti su zelene ako ih je korisnik u potpunosti završio, providne ako još uvek nije.

Kada korisnik završi sve oblasti u okviru nivoa beginner, otključava se sledeći (intermediate), pa kad se on završi otključava se poslednji nivo (advanced).



Slika 17 - Practice stranica pr. 1

INTERMEDIATE



Places



Body



Clothing



Questions 2



Imperative

Slika 18 - Practice stranica pr. 2

Klikom na oblast otvara se prozor gde moeže da se vidi koliko je vežbi odrađeno, kao i link ka samoj vežbi i pomoći tj. gramatici vezanoj sa izabranu vežbu.



Basics



Slika 19 - Pop up prozor jedne vežbe

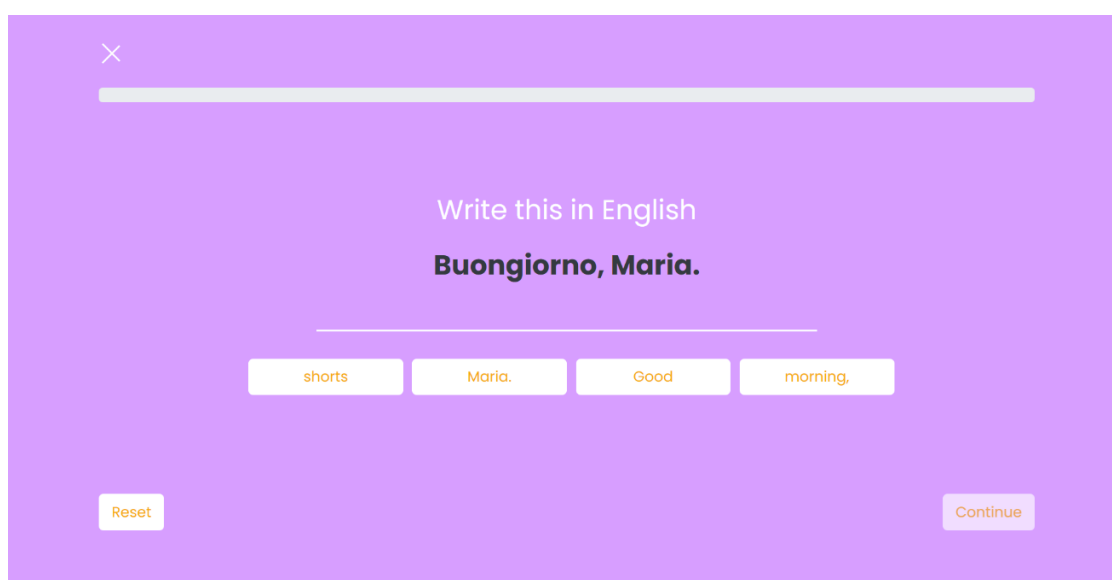
4.4.7 Task/id

-Ulaskom u oblast, korisnik može dobiti jednu od tri tipova vežbi: povezivanje odgovarajućih parova (slika 20), prevod date rečenice (slika 21) i izabrati reč koja nedostaje (slika 22). Ako je korisniku prvi nivo koji radi počinje se sa vežbom povezivanje odgovarajućih parova jer je to najlakši tip vežbe. Zatim reč koja nedostaje kad korisnik drugi put ulazi u oblast, pa tek onda prevod cele rečenice kada treći put radi vežbu. Kada korisnik izabere odgovor, otključava se dugme Continue. Dugme Reset resetuje korisnikov izbor u slučaju da je napravio grešku.

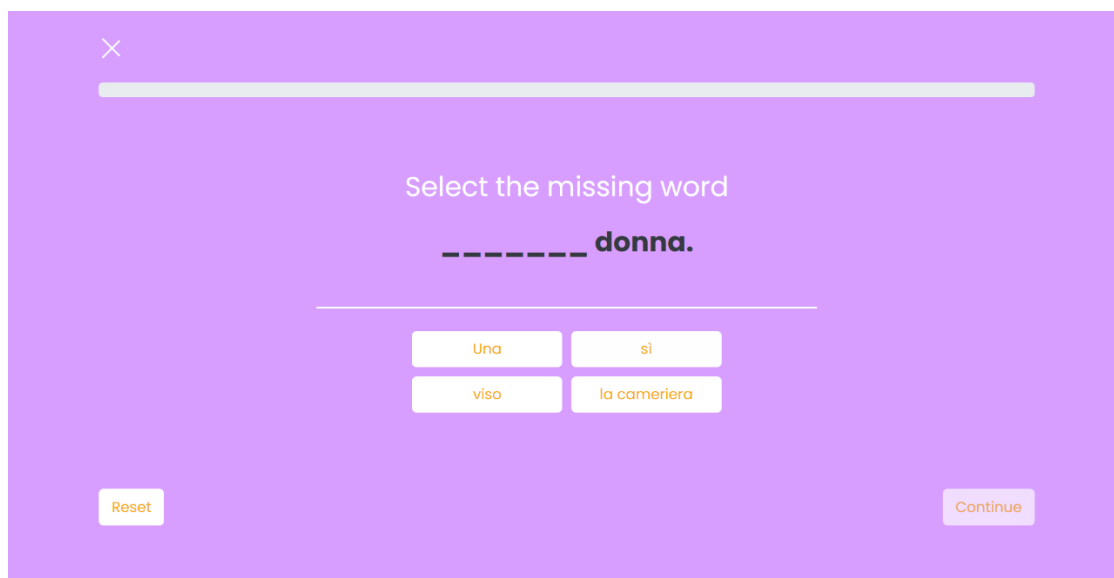
-Kada klikne na dugme koje označava italijansku reč, otključavaju se prevodi. Kada sve izabere dugmad je disabled.



Slika 20 - Vežba tip 1



Slika 21 - Vežba tip 2



Slika 22 - Vežba tip 3

Greške se ispisuju u formatu:



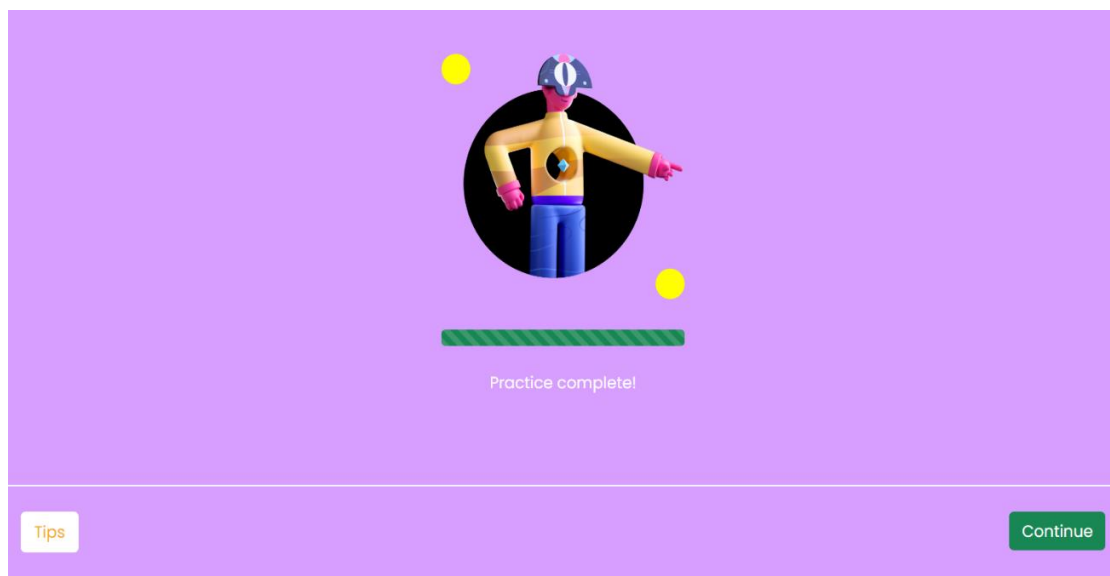
Slika 23 - Ispis greški pr. 1



Slika 24 - Ispis greški pr. 2

4.4.8 Finished Challenge

-Stranica nakon što korisnik završi vežbu, sadrži dugme za gramatiku (Tips) i Continue koje ga vraća na stranicu Practice.



Slika 25 - Stranica koja se prikazuje kada se završi vežba

4.4.9 Task info

-Sadrži gramatiku i savete za odabranu oblast. I dugme za vežbanje te oblasti, koje vodi na stranicu task/id.

LAPSUS LINGUAE

PracticeWordsAboutLanguage▼marija

Phrases

Negations In this section you'll use negations for the first time.

The English no has two main uses:

Particle (e.g. "no!"): this translates directly to the Italian no.

Determiner (e.g. "no one"): you'll learn the translations for this in a later section. The English not almost always translates to the Italian non. However, while not often follows the verb it negates or its auxiliary, the Italian non always precedes it.

Greetings

Ciao is used both ways in Italian: when meeting (also salve) and when parting (also arrivederci or addio).

Buongiorno and buonasera are normally used when meeting, although they can be used when parting as well: the first is used in the first half of the day and the latter in the remaining half.

Buonanotte is always used when parting, as it presumes that the day is over (same as "good night").

Prego is a courtesy form used in many occasions to accompany a kind action, and it's the customary answer to reply to received thanks.

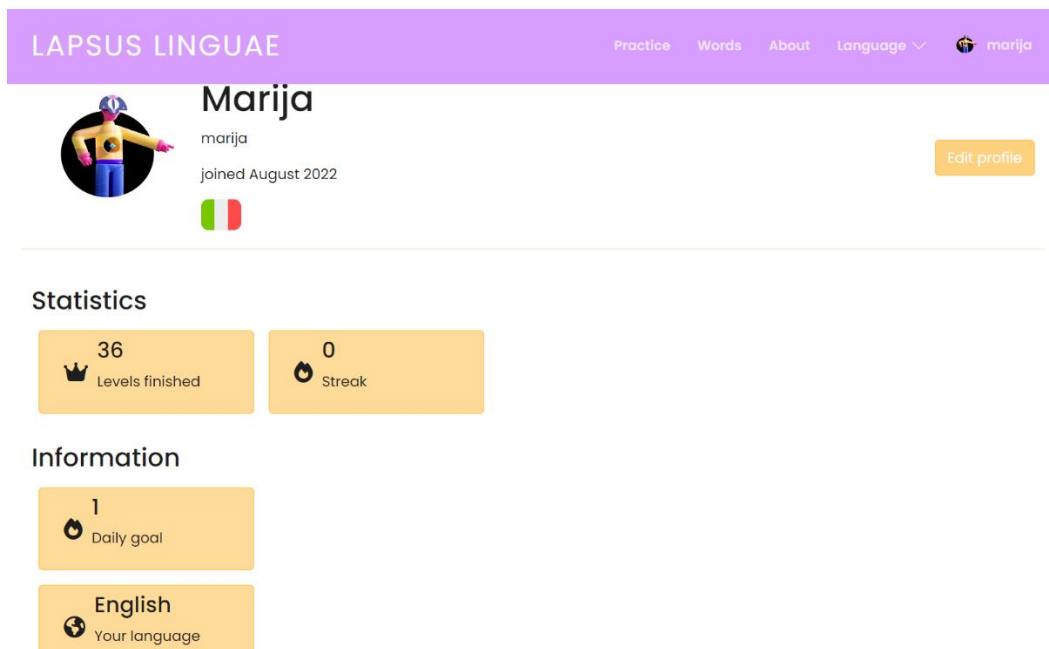
Per favore, per piacere and per cortesia are courtesy forms used when asking for something.

Try practicing this level

Slika 26 - Stranica koja prikazuje informacije o odabranoj oblasti

4.4.10 Profile

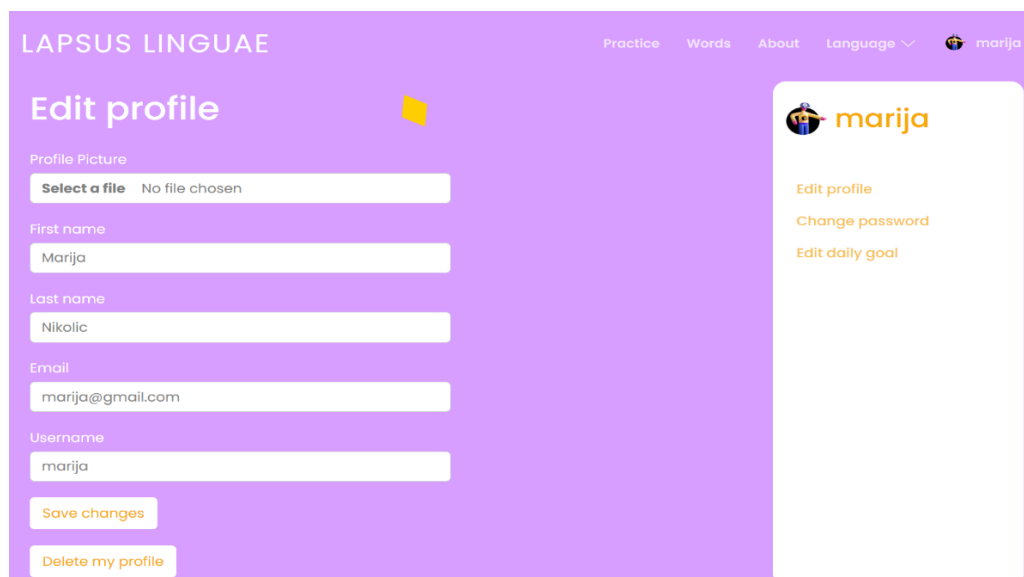
-Dostupno samo autorizovanim korisnicima. Profil korisnika, sadrži njegovu sliku, korisničko ime, mesec i godinu u kom je napravio nalog. Informacije u ukupnom broju završenih vežbi, njegov streak, dnevni cilj i odabrani jezik.



Slika 27 - Profilna stranica

4.4.11 Edit Profile

-Podešavanje profila. Navigacija sa strane sadrži linkove ka stranicama z apromeni dnevnog cilja i lozinke. Automatski se popunjava korisnikovim podacima. Profilna slika nije obavezna. Ostala polja jesu i verifikacijom ako ne budu uneta korisnik će biti obavešten.



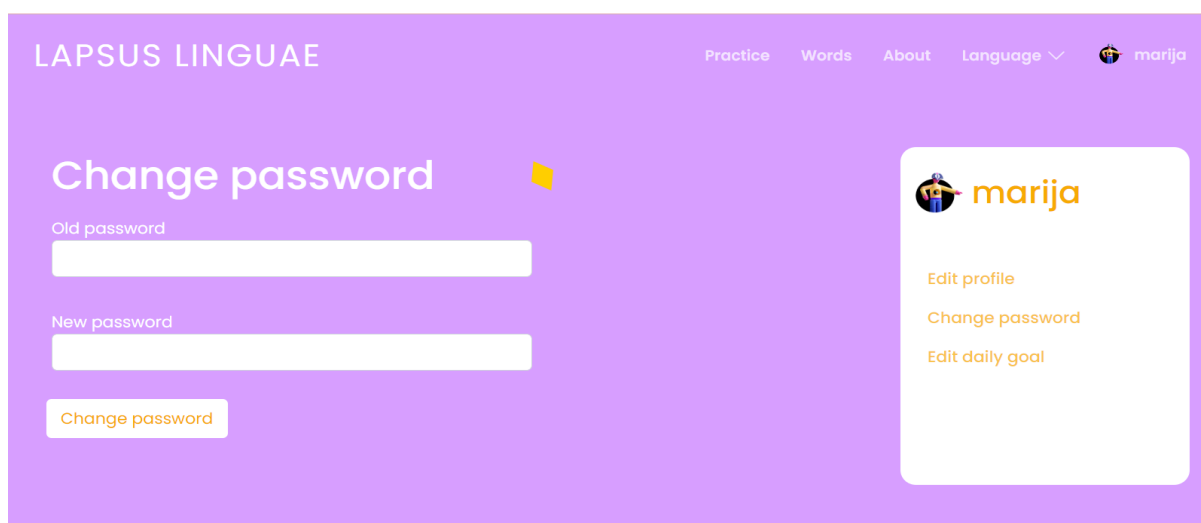
Slika 28 - Stranica za izmenu profila pr. 1



Slika 29 - Ispis greški prilikom izmene profila

4.4.12 Change Password

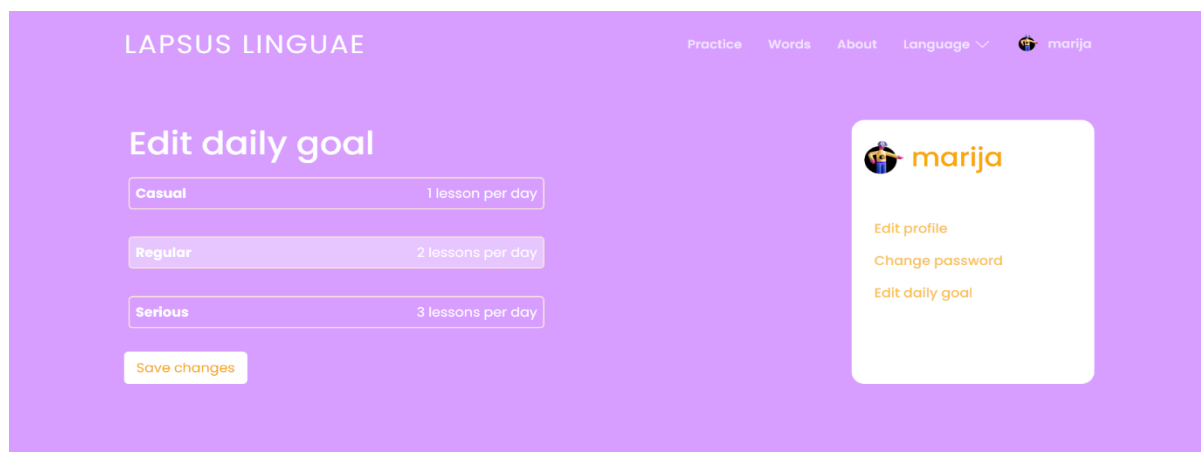
-Promena lozinke, moguće ako je stara (tj. trenutna) lozinka tačna.



Slika 30 - Stranica za izmenu lozinke

4.4.13 Change Daily Goal

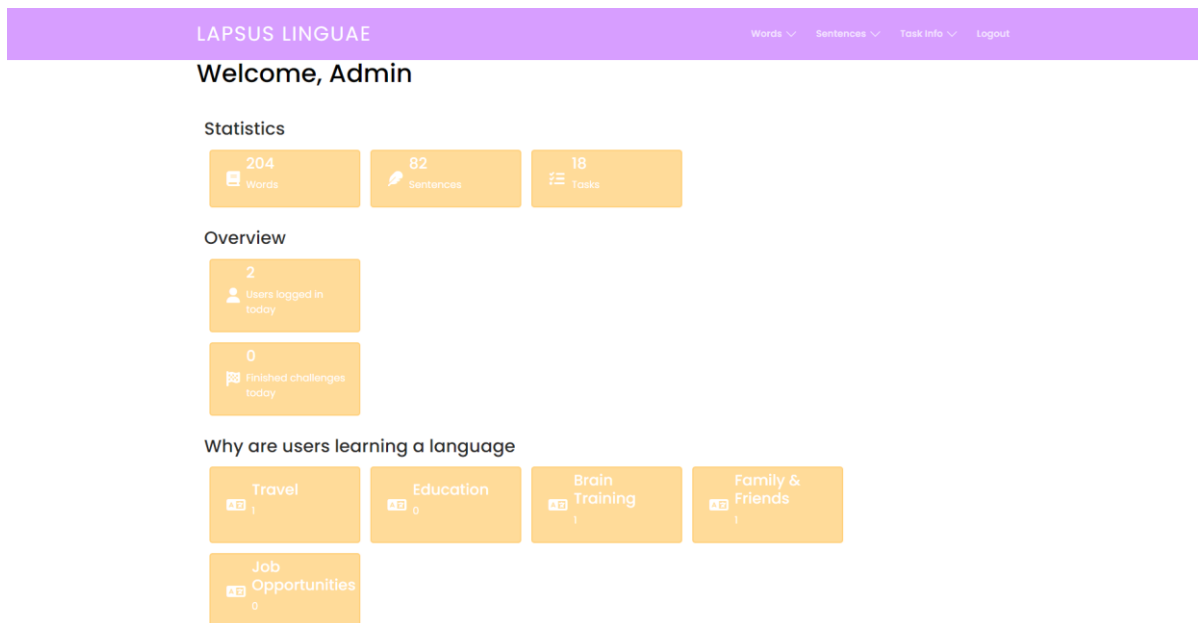
-Korisnik može da promeni svoj dnevni cilj za rad vežbi.



Slika 31 - Stranica za izmenu dnevnog cilja

4.4.14 Admin/Home Page

-Početna stranica za admina, dostupna jedino njemu i zaštićena je kao i ostale middleware-om.



Slika 32 - Početna stranica za admina

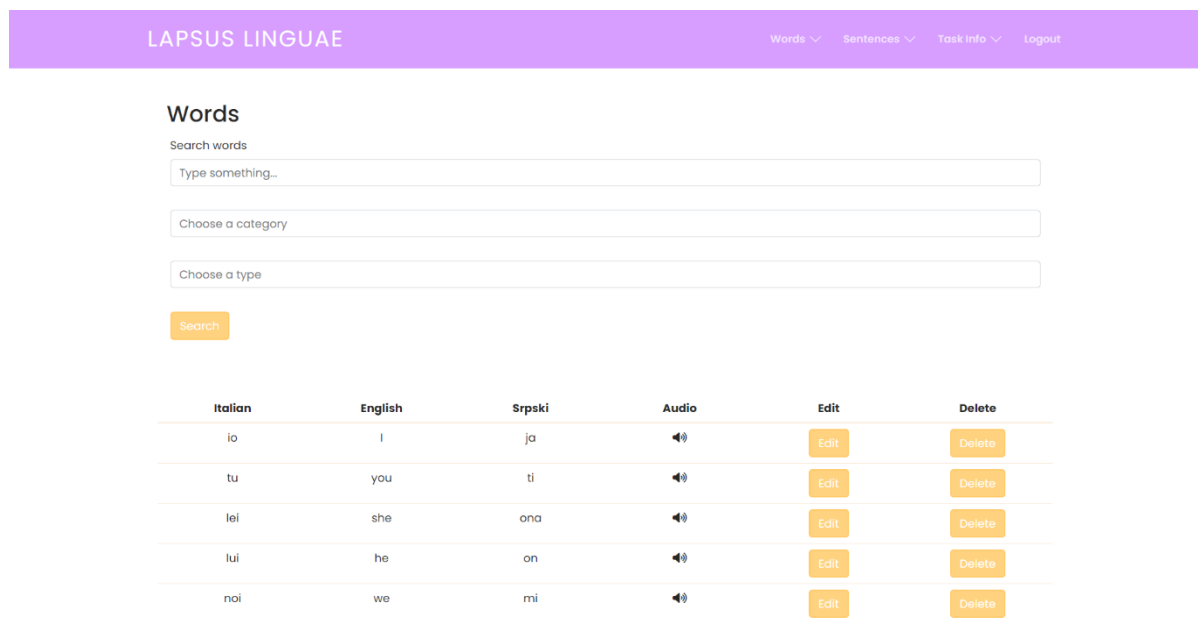
4.4.15 Admin/Insert a Word

-Dodavanje nove reči i njenih prevoda na engleski i sprski jezik.

Slika 33 - Stranica za dodavanje reči

4.4.16 Admin/Words

-Prikaz svih reči (+ audio i njeni prevodi) sa paginacijom. Dugmad za modifikovanje reči i brisanje. Moguće pretraživanje preko imena, kategorije kojoj pripada ili tipa.

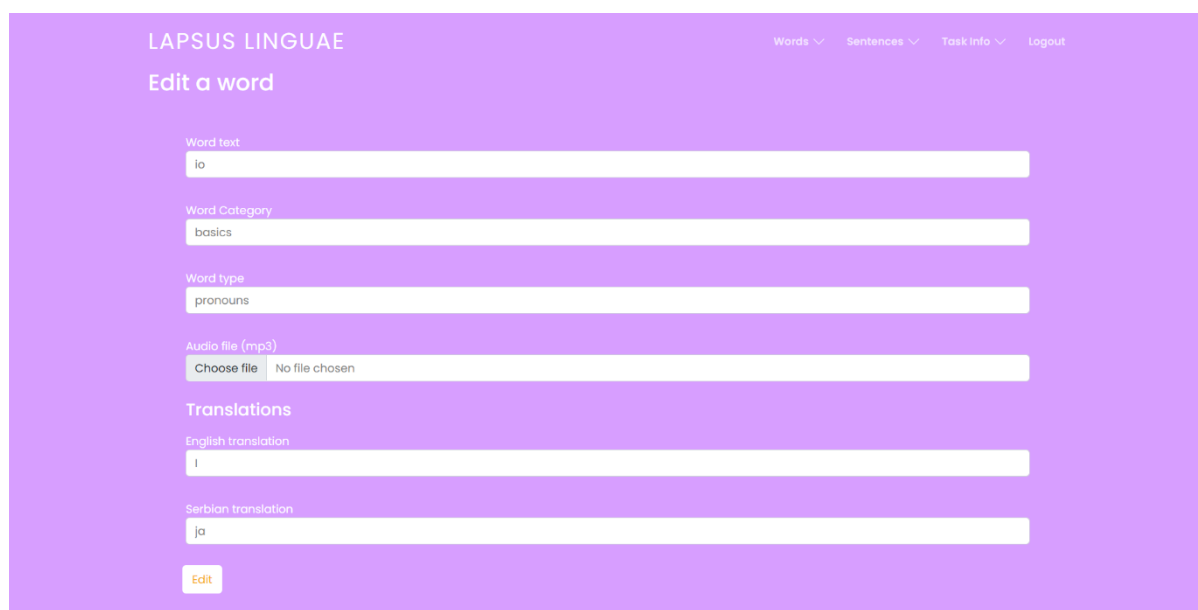


Italian	English	Srpski	Audio	Edit	Delete
io	I	ja		Edit	Delete
tu	you	ti		Edit	Delete
lei	she	ona		Edit	Delete
lui	he	on		Edit	Delete
noi	we	mi		Edit	Delete

Slika 34 - Stranica za prikaz reči kod admina

4.4.17 Admin/Edit a Word

-Modifikovanje reči, automatski se opounjavaju polja i svako je obavezno osim audio fajla.



Word text:

Word Category:

Word type:

Audio file (mp3): No file chosen

Translations:

English translation:

Serbian translation:

[Edit](#)

Slika 35 - Stranica za izmenu reči

4.4.18 Admin/Sentences

-Prikaz rečenice i njenih prevoda. Mogućnost pretrage preko teksta ili oblasti kojoj ona pripada. Takođe, prikaz dugmadi za modifikovanje ili brisanje rečenice.

Italian	English	Srpski	Task	Edit	Delete
Buongiorno, Maria.	Good morning, Maria.	Dobar dan, Maria.	basics	Edit	Delete
Buonasera.	Good evening.	Dobro veče.	basics	Edit	Delete
Buonanotte, Anna.	Good night, Anna.	Laku noć, Anna.	basics	Edit	Delete
Una ragazza.	A girl.	Devojčica.	basics	Edit	Delete
Una donna.	A woman.	Žena.	basics	Edit	Delete

Slika 36 - Prikaz rečenica kod admina

4.4.19 Admin/Insert a Sentence

-Dodavanje nove rečenice i njenih prevoda na engleski i srpski jezik. Sva polja su obavezna.

Sentence text

Text

Task

Choose...

Translations

English translation

Serbian translation

Insert

Slika 37 - Stranica za dodavanje rečenice

4.4.20 Admin/Edit a Sentence

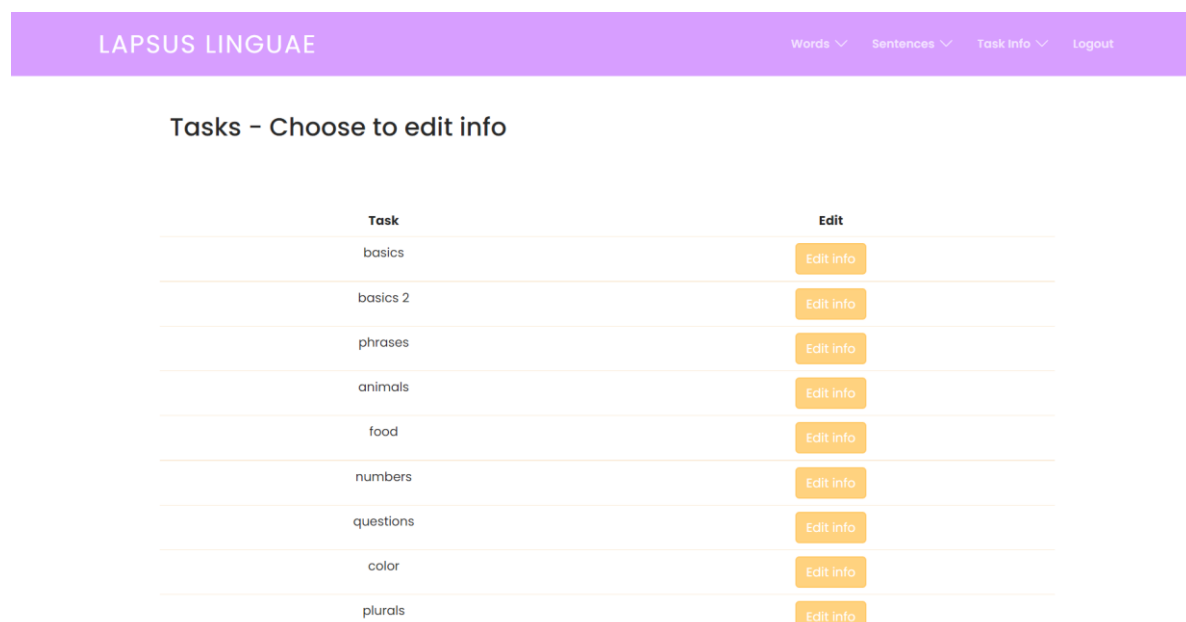
-Modifikovanje rečenice, automatski se popunjavaju polja i svako je obavezno osim audio fajla.



Slika 38 - Stranica za izmenu rečenice

4.4.21 Admin/Task Info

-Prikaz oblasti na koje se može kliknuti i zatim modifikovati gramtike tj. pomoćnog teksta ili informacije o oblasti. S obzirom da oblasti nema puno, i zbog strukture sajta nikad ih neće ni biti puno, nema opcije za pretragu.



Task	Edit
basics	Edit info
basics 2	Edit info
phrases	Edit info
animals	Edit info
food	Edit info
numbers	Edit info
questions	Edit info
color	Edit info
plurals	Edit info

Slika 39 - Stranica za prikaz oblasti gde se mogu izmeniti njihove informacije

4.4.22 Admin/Task Info Edit

-Modifikovanje gramatike tj. pomoćnog teksta ili informacije o oblasti, kao i prevoda na engleski i srpski jezik. Sva polja su obavezna.

LAPSUS LINGUAE

Words Sentences Task Info Logout

Edit the task info

Task: basics

Intro

Welcome to the Italian course! Remember that you can click on the words to see tips of possible translations.

Text

Personal pronouns

The subject pronouns in Italian are:

Io - I

Translations

Serbian Intro

Dobrodošli na kurs italijanskog! Zapamtite da možete kliknuti na reči da biste videli savete za moguće prevode.

Serbian Text

Lične zamenice

Subjektne zamenice u italijanskom jeziku su:

Io - I

Edit

Slika 40 - Stranica za izmenu informacija izabrane oblasti

4.4.23 Admin/Login

-Posebna ruta i stranica za logovanje admina. Jedino se proverava uloga iz baze, da li je admin.

LAPSUS LINGUAE

Admin Login

admin

Login

Slika 41 - Login stranica za admina

5. Kodovi

5.1 PHP

5.1.1 Rute

5.1.1.1 web.php

```
<?php

use App\Http\Controllers\AdminController;
use App\Http\Controllers\FrontController;
use App\Http\Controllers\AccountController;
use App\Http\Controllers\PracticeController;
use App\Http\Controllers\ProfileController;
use App\Http\Controllers\RecapController;
use App\Http\Controllers\SentenceController;
use App\Http\Controllers\TaskInfoController;
use App\Http\Controllers\WordController;
use Illuminate\Support\Facades\Route;

/*
|-----
| Web Routes
|-----
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/

Route::get('/', [FrontController::class, 'home'])->name('home');

Route::get('/about', [FrontController::class, 'about'])->name('about');

Route::middleware('isntLoggedIn')->group(function () {
    Route::get('/login', [AccountController::class, 'getlogin'])->
>name('login');
    Route::post('/login', [AccountController::class, 'login']);

    Route::get('/register', [AccountController::class, 'getregister'])->
>name('register');
    Route::post('/register', [AccountController::class, 'register']);
    Route::get('/check-username-email', [AccountController::class,
'checkExistsEmailOrUsernameDuringRegister']);
});

Route::middleware('isLoggedIn')->group(function () {
    Route::get('/logout', [AccountController::class, 'logout'])->
>name('logout');

    Route::get('/practice', [PracticeController::class, 'index'])->
>name('practice');
    Route::get('/task/exit', [PracticeController::class, 'exitTask'])->
>name('exit-task');
    Route::get('/task/{id}', [PracticeController::class, 'task'])->
>name('task');
    Route::get('/task/{id}/check-translation', [PracticeController::class,
'checkAnswer']);
    Route::get('/practice/task-info/{id}', [PracticeController::class,
```

```

'taskInfoIndex']->name('task-info');

Route::get('/profile',[ProfileController::class, 'index']->name('profile');
Route::get('/edit-profile',[ProfileController::class, 'editProfileIndex']->name('edit-profile');
Route::post('/edit-profile',[ProfileController::class, 'editProfile']);
Route::put('/delete-profile',[ProfileController::class, 'deleteProfile']->name('delete-profile');

Route::get('/change-password',[ProfileController::class, 'changePasswordIndex']->name('change-password');
Route::post('/change-password',[ProfileController::class, 'changePassword']);

Route::get('/edit-goal',[ProfileController::class, 'editGoalIndex']->name('edit-goal');
Route::post('/edit-goal',[ProfileController::class, 'editGoal']);

Route::get('/change-language/{language_id}',[ProfileController::class, 'changeLanguage']->name('change-language');

});

Route::get('/recap',[RecapController::class, 'index']->name('recap');
Route::get('/recap/search-words',[RecapController::class, 'search']);

Route::prefix('admin')->group(function () {
    Route::middleware('isntLoggedIn')->group(function () {
        Route::get('/login', [AdminController::class, 'getlogin']->name('admin-login');
        Route::post('/login', [AdminController::class, 'login']);
    });

    Route::middleware('isAdmin')->group(function () {
        Route::get('/home', [AdminController::class, 'home']->name('admin-home');
        Route::resource('words', WordController::class);
        Route::resource('sentences', SentenceController::class);
        Route::resource('task-info', TaskInfoController::class);
    });
});

```

5.1.2 Kontroleri

5.1.2.1 AccountController

```

<?php

namespace App\Http\Controllers;

use App\Http\Requests\LoginRequest;
use App\Http\Requests\RegisterRequest;
use App\Models\User;
use App\Models\UserAction;
use App\Models\UserProgress;
use App\Models\WhyLearnALanguage;

```

```

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;
use Psy\Util\Json;

class AccountController extends BaseController
{
    public function getregister(){
        $this->data['why_learn_a_language'] = WhyLearnALanguage::all();
        $this->data['svg_container_animation'] = [
            '<div class="icon">
                <svg width="100" height="100"
viewBox="0 0 600 600" xmlns="http://www.w3.org/2000/svg">
                    <path stroke="none" stroke-
width="0" fill="#f5f5f5"
d="M300,582.0697525312426C382.5290701553225,586.8405444964366,449.978979469
0241,525.3245884688669,502.5850820975895,461.55621195738473C556.60642568678
1,396.0723002908107,615.8543463187945,314.28637112970534,586.6730223649479,
234.56875336149918C558.9533121215079,158.8439757836574,454.9685369536778,16
4.00468322053177,381.49747125262974,130.76875717737553C312.15926192815925,9
9.40240125094834,248.97055460311594,18.661163978235184,179.8680185752513,50
.54337015887873C110.5421016452524,82.52863877960104,119.82277516462835,180.
83849132639028,109.12597500060166,256.43424936330496C100.08760227029461,320
.3096726198365,92.17705696193138,384.0621239912766,124.79988738764834,439.7
174275375508C164.83382741302287,508.01625554203684,220.96474134820875,577.5
009287672846,300,582.0697525312426"></path>
                </svg>
                <i class="bx bx-file"></i>
            </div>',
            '<div class="icon">
                <svg width="100" height="100"
viewBox="0 0 600 600" xmlns="http://www.w3.org/2000/svg">
                    <path stroke="none" stroke-
width="0" fill="#f5f5f5"
d="M300,541.5067337569781C382.14930387511276,545.0595476570109,479.87368415
81634,548.3450877840088,526.4010558755058,480.5488172755941C571.52184695816
45,414.80211281144784,517.5187510058486,332.0715597781072,496.5253901046910
4,255.14436215662573C477.37192572678356,184.95920475031193,473.573636565579
14,105.61284051026155,413.0603344069578,65.22779650032875C343.2747038610229
4,18.654635553484475,251.2091493199835,5.337323636656869,175.0934190732945,
40.62881213300186C97.87086631185822,76.43348514350839,51.98124368387456,156
.15599469081315,36.44837278890362,239.84606092416172C21.716077023791087,319
.22268207091537,43.775223500013084,401.1760424656574,96.891909868211,461.97
329694683043C147.22146801428983,519.5804099606455,223.5754009179313,538.201
503339737,300,541.5067337569781"></path>
                </svg>
                <i class="bx bx-tachometer"></i>
            </div>',
            '<div class="icon">
                <svg width="100" height="100"
viewBox="0 0 600 600" xmlns="http://www.w3.org/2000/svg">
                    <path stroke="none" stroke-
width="0" fill="#f5f5f5"
d="M300,503.46388370962813C374.79870501325706,506.71871716319447,464.803455
1963731,527.1746412648533,510.4981551193396,467.86667711651364C555.92873085
11215,408.9015244558933,512.6030010748507,327.5744911775523,490.21105757886
3,256.5855673507754C471.097692560561,195.9906835881958,447.69079081568157,1
38.11976852964426,395.19560036434837,102.3242989838813C329.3053358748298,57
.3949838291264,248.02791733380457,8.279543830951368,175.87071277845988,42.2
42879143198664C103.41431057327972,76.34704239035025,93.79494320519305,170.9
812938413882,81.28167332365135,250.07896920659033C70.17666984294237,320.274

```

```

84674793965,64.84698225790005,396.69656628748305,111.28512138212992,450.495
0937839243C156.20124167950087,502.5303643271138,231.32542653798444,500.4755
392045468,300,503.46388370962813"></path>
</svg>
<i class="bx bx-layer"></i>
</div>',
'<div class="icon">
<svg width="100" height="100"
viewBox="0 0 600 600" xmlns="http://www.w3.org/2000/svg">
<path stroke="none" stroke-
width="0" fill="#f5f5f5"
d="M300,532.3542879108572C369.38199826031484,532.3153073249985,429.10787420
159085,491.63046689027357,474.5244479745417,439.17860296908856C522.88858469
62883,383.3225815378663,569.1668002868075,314.3205725914397,550.74321519292
88,242.7694973846089C532.6665558377875,172.5657663291529,456.2379748765914,
142.6223662098291,390.3689995646985,112.34683881706744C326.66090330228417,8
3.06452184765237,258.84405631176094,53.51806209861945,193.32584062364296,78
.48882559362697C121.61183558270385,105.82097193414197,62.805066853699245,16
7.19869350419734,48.57481801355237,242.6138429142374C34.843463184063346,315
.3850353017275,76.69343916112496,383.4422959591041,125.22947124332185,439.3
748458443577C170.7312796277747,491.8107796887764,230.57421082200815,532.393
2930995766,300,532.3542879108572"></path>
</svg>
<i class="bx bx-slideshow"></i>
</div>',
'<div class="icon">
<svg width="100" height="100" viewBox="0 0 600 600"
xmlns="http://www.w3.org/2000/svg">
<path stroke="none" stroke-width="0" fill="#f5f5f5"
d="M300,566.797414625762C385.7384707136149,576.1784315230908,478.7894351017
131,552.8928747891023,531.9192734346935,484.94944893311C584.6109503024035,4
17.5663521118492,582.489472248146,322.67544863468447,553.9536738515405,242.
03673114598146C529.1557734026468,171.96086150256528,465.24506316201064,127.
66468636344209,395.9583748389544,100.7403814666027C334.2173773831606,76.748
2773500951,269.4350130405921,84.62216499799875,207.1952322260088,107.288914
0133804C132.92018162631612,134.33871894543012,41.79353780512637,160.0025916
5414826,22.644507872594943,236.69541883565114C3.319112789854554,314.0945973
066697,72.72355303640163,379.243833228382,124.04198916343866,440.3218312028
393C172.9286146004772,498.5055451809895,224.45579914871206,558.531796884010
2,300,566.797414625762"></path>
</svg>
<i class="bx bx-arch"></i>
</div>'
];

return view('client.pages.register', $this->data);
}

public function getlogin(){
return view('client.pages.login', $this->data);
}

public function login(LoginRequest $request){
try{
$username = $request->input('username');
$password = md5($request->input('password'));
/*u zavisnosti koja je greska, neka se vrati uvek nazad info
ako za password nije nista ispostovano neka onda pise da je
required, ne treba da se budzi i sa ostalim
greskama. u sledecem requestu neka pise sledeca validaciona
greska ako je ima. i tako sve po redu.*/

```

```

        $user = User::where([
            ['password', $password],
            ['email', $username]
        ])->orWhere([
            ['password', $password],
            ['username', $username]
        ])->where('active', 1) -> first();

        if($user){
            if($user->role_id==1){ //admin
                return redirect()->route('admin-login');
            }
            else{
                $request->session()->put('user', $user);

                $action = new UserAction;
                $action->user_id = $user->id;
                $action->action = 'login';
                $action->save();
                //streak
                if($user->streak != 0){
                    $date_today = date('Y-m-d');
                    $date_yesterday = date('Y-m-
d', strtotime('yesterday'));
                    $userProgress =
UserProgress::where('user_id', $user->id) -> get();
                    $userProgressYesterday = $userProgress-
->where('date', $date_yesterday) -> first();
                    $userProgressToday = $userProgress-
->where('date', $date_today) -> first();

                    //dd($userProgressYesterday);
                    if(!$userProgressYesterday){
                        if($userProgressToday && $user->streak == 1){
                            //danas je prva vezba koja je uradio i
dobio streak od 1 dana
                            $request->session()->get('user')->streak =
$user->streak;
                        }
                        else{
                            //resetovanje streaka jer nije uradjena
vezba juce
                            if($user->streak != 0){
                                $user->streak = 0;
                                $user->save();
                            }
                            $request->session()->get('user')->streak =
0;
                        }
                    }
                    else{
                        //ima streak duzi od 1,
                        $request->session()->get('user')->streak =
$user->streak;
                    }
                }
                else{
                    //streak mu je nula, stoji ovde da bi se apdejtovao
session, nzm da li je neophodno
                    $request->session()->get('user')->streak = 0;
                }
            }
        }
    }
}

```

```

    }
    //streak

    return redirect()->route('home')->with('loginSuccess',
'Welcome back.');
```

```

    }
}

return back()->with('loginFail', 'Username or password
incorrect.');
```

```

}
catch (\Exception $ex) {
    Log::error($ex->getMessage());
    return back()->with('loginFail', $ex);
}
}

public function logout(Request $request){

    try{
        if(session()->has('user')){

            $action = new UserAction;

            $action->user_id = session()->get('user')->id;
            $action->action = 'logout';

            $action->save();

            $request->session()->remove('user');
        }
    }
    catch (\Exception $ex) {
        Log::error($ex->getMessage());
        return back()->with('logoutFail', $ex);
    }

    return redirect()->route('home');
}

public function register(RegisterRequest $request){
    $first_name = $request->input('first_name');
    $last_name = $request->input('last_name');
    $username = $request->input('username');
    $email = $request->input('email');
    $password = md5($request->input('password'));
    $birth_date = $request->input('birth_date');
    $profile_picture = $request->file('profile_picture');
    $why_learn_a_language_id = $request->input('why-learn-a-language');

    try{
        $user = new User;

        $emailExists = User::where('email',$email)->get()->count();
        $usernameExists = User::where('username',$username)->get()-
>count();

        if($emailExists && $usernameExists){
            return back()->with('regFail','Email and username have been
already taken.');
```

```

        }
    }
}

```



```

        if($emailExists || $usernameExists){
            if($emailExists){
                return back()->with('regFail','Email has already been
taken.');
```

```
            }
            if($usernameExists){
                return back()->with('regFail','Username has already
been taken.');
```

```
            }
        }

        if($profile_picture != null){

            $profile_picture_name =
uniqid()."_.time().".".$profile_picture->extension();
            $profile_picture->storeAs("/public/img/profile_pictures",
$profile_picture_name);

            $user->profile_picture = $profile_picture_name;
        }
        else{
            $user->profile_picture = null;
        }

        $user->first_name = $first_name;
        $user->last_name = $last_name;
        $user->username = $username;
        $user->email = $email;
        $user->password = $password;
        $user->birth_date = $birth_date;
        $user->why_learn_a_language_id = $why_learn_a_language_id;
        $user->role_id = 2;
        $user->save();

        $action = new UserAction;
        $action->user_id = $user->id;
        $action->action = 'register';
        $action->save();

        $user->daily_goal = 1;
        $user->language_id = 1;
        $request->session()->put('user',$user);
        return redirect()->route('home')->with('loginSuccess',
'Welcome.');
```

```
    }
    catch (\Exception $ex) {
        Log::error($ex->getMessage());
        return back()->with('regFail', $ex);
    }
}

public function checkExistsEmailOrUsernameDuringRegister(Request
$request){

    $email = $request->email;
    $username = $request->username;

    $emailExists = User::where('email',$email)->get()->count();
    $usernameExists = User::where('username',$username)->get()-
>count();

```

```

        if($emailExists && $usernameExists){
            return Json::encode("UsernameAndEmailExist");
        }
        else if($emailExists || $usernameExists){
            if($emailExists){
                return Json::encode('EmailExists');
            }
            if($usernameExists){
                return Json::encode('UsernameExists');
            }
        }
        else{
            return Json::encode('NotExists');
        }
    }
}

```

5.1.2.2 AdminController

```

<?php

namespace App\Http\Controllers;

use App\Http\Requests\LoginRequest;
use App\Models\Sentence;
use App\Models\Task;
use App\Models\User;
use App\Models\UserAction;
use App\Models\UserProgress;
use App\Models\WhyLearnALanguage;
use App\Models\Word;
use Illuminate\Support\Facades\Log;

class AdminController extends BaseController
{
    public function getlogin(){
        return view('admin.pages.login');
    }

    public function login(LoginRequest $request){
        try{
            $username = $request->input('username');
            $password = md5($request->input('password'));
            $user = User::where([
                ['password', $password],
                ['email', $username]
            ]->orWhere([
                ['password', $password],
                ['username', $username]
            ]->first();

            if($user){
                if($user->role_id == 1){ //admin
                    $request->session()->put('user', $user);

                    $action = new UserAction;
                    $action->user_id = $user->id;
                    $action->action = 'login';
                }
            }
        }
    }
}

```

```

        $action->save();

        return redirect()->route('admin-home')-
>with('loginSuccess', 'Welcome back.');
```

```

    }
    else{
        return back()->with('loginFail', 'Admin with these
credentials does not exist.');
```

```

    }
}

return back()->with('loginFail', 'Username or password
incorrect.');
```

```

}
catch (\Exception $ex) {
    Log::error($ex->getMessage());
    return back()->with('loginFail', $ex);
}
return view('admin.pages.home');
```

```

}

public function home(){
    $date_today = date('Y-m-d');
    $words = count(Word::all());
    $sentences = count(Sentence::all());
    $tasks = count(Task::all());
    $usersLoggedIn =
count(UserAction::where('created_at','like','%'.$date_today.'%')-
>where('action','login')->get());
    $userProgress = count(UserProgress::where('date',$date_today)-
>get());
    $userWhyLearnLanguage = User::select('why_learn_a_language_id')-
>where('active',1)->get();
    $whyLearnLanguage = WhyLearnALanguage::all();

    foreach($whyLearnLanguage as $el){
        $el->totalUsers = 0;
    }

    foreach($userWhyLearnLanguage as $userWhy){
        foreach ($whyLearnLanguage as $item) {
            if($userWhy->why_learn_a_language_id == $item->id){
                $item->totalUsers = $item->totalUsers + 1;
            }
        }
    }

    $this->data['words'] = $words;
    $this->data['sentences'] = $sentences;
    $this->data['tasks'] = $tasks;
    $this->data['usersLoggedIn'] = $usersLoggedIn;
    $this->data['userProgress'] = $userProgress;
    $this->data['whyLearnLanguage'] = $whyLearnLanguage;

    return view('admin.pages.home', $this->data);
}
}

```

5.1.2.3 BaseController

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class BaseController extends Controller
{
    public $data = [];
}
```

5.1.2.4 FrontController

```
<?php

namespace App\Http\Controllers;

use App\Models\Employee;
use App\Models\User;
use Illuminate\Http\Request;

class FrontController extends BaseController
{
    public function home(Request $request){
        $this->data['traits'] = array(
            array(
                'title' => 'efficient',
                'srb_title' => 'efikasno',
                'text' => 'tekst',
                'icon' => "<i class='fa-solid fa-leaf'></i>"
            ),
            array(
                'title' => 'easy to revise',
                'srb_title' => 'lako za ponavljanje',
                'text' => 'tekst',
                'icon' => "<i class='fa-solid fa-reply'></i>"
            ),
            array(
                'title' => 'motivating',
                'srb_title' => 'motivaciono',
                'text' => 'tekst',
                'icon' => "<i class='fa-solid fa-person-walking'></i>"
            ),
            array(
                'title' => 'fun',
                'srb_title' => 'zabavno',
                'text' => 'tekst',
                'icon' => "<i class='fa-solid fa-puzzle-piece'></i>"
            )
        );

        $userTasks = null;
        if($request->session()->has('user')){
            $user_id = $request->session()->get('user')->id;
            $user = User::where('id', $user_id)->first();
            $userTasks = $user->userTasks;
        }
        $this->data['userTasks'] = $userTasks;
```

```

        return view('client.pages.home', $this->data/*, ['menu' => $this->data['menu']]*/);
    }

    public function about(){
        $team = Employee::where('department_id',1)->take(4)->get();

        $this->data['team'] = $team;

        return view('client.pages.about.index',$this->data);
    }
}

```

5.1.2.5 PracticeController

```

<?php

namespace App\Http\Controllers;

use App\Models\MainLevel;
use App\Models\Sentence;
use App\Models\Task;
use App\Models\TaskInfo;
use App\Models\User;
use App\Models\UserProgress;
use App\Models\UserTask;
use App\Models\Word;
use App\Models\WordCategory;
use Illuminate\Http\Request;
use Nette\Utils\Paginator;
use Psy\Util\Json;

class PracticeController extends BaseController
{
    public function index(Request $request){
        $user_id = $request->session()->get('user')->id;
        $user = User::where('id', $user_id)->first();
        $userTasks = $user->userTasks;
        $tasks = Task::all();

        $date_today = date('Y-m-d');
        $userProgress = UserProgress::where('user_id', $user_id)->where('date', $date_today)->first();

        if($userProgress == null){
            $currentUserProgress = 0;
        }
        else{
            $currentUserProgress = $userProgress->quantity_done;
        }

        $total = 0;
        foreach ($userTasks as $userTask){
            $total += $userTask->levels_finished;
        }
        foreach($tasks as $task){
            $task->finished = false;
            $task->levels_finished = 0;
        }
    }
}

```

```

        /** Task Translation */
        foreach ($task->translations as $translation){
            if($translation->language->id == 2){
                $task->srb_translation = $translation->text;
            }
        }
    }
    if($userTasks){
        foreach($userTasks as $userTask){
            foreach($tasks as $task){
                if($userTask->task_id == $task->id && $userTask-
>finished == true){
                    $task->finished = true;
                }
                if($userTask->task_id == $task->id){
                    $task->levels_finished = $userTask-
>levels_finished;
                }
            }
        }
    }

    /** put in main levels finished*/
    $user_id = $request->session()->get('user')->id;
    $finishedUsersTasks = UserTask::where('user_id',$user_id)-
>where('finished',1)->get();
    $beginner_level_finished = false;
    $intermediate_level_finished = false;

    if($finishedUsersTasks != null){
        $main_levels = MainLevel::all();
        $beginner_level_id = $main_levels->where('title','beginner')-
>first()->id;
        $intermediate_level_id = $main_levels-
>where('title','intermediate')->first()->id;

        $total_beginner_tasks =
count(Task::where('main_level_id',$beginner_level_id)->get());
        $total_intermediate_tasks =
count(Task::where('main_level_id',$intermediate_level_id)->get());

        if($finishedUsersTasks){
            $beginnerTasks = 0;
            $intermediateTasks = 0;
            foreach($finishedUsersTasks as $userTask){
                if($userTask->task->main_level_id ==
$beginner_level_id){
                    $beginnerTasks++;
                }
                if($userTask->task->main_level_id ==
$intermediate_level_id){
                    $intermediateTasks++;
                }
            }
            if($beginnerTasks == $total_beginner_tasks){
                $beginner_level_finished = true;
            }
            else if($intermediateTasks == $total_intermediate_tasks){

```

```

        $intermediate_level_finished = true;
    }
}

$this->data['beginner_level_finished'] = $beginner_level_finished;
$this->data['intermediate_level_finished'] =
$intermediate_level_finished;
$this->data['tasks'] = $tasks;
$this->data['levelsFinished'] = $total;
$this->data['currentUserProgress'] = $currentUserProgress;

return view ('client.pages.practice.index', $this->data);
}

public function exitTask(Request $request){
    if($request->session()->has('sentences')){
        $request->session()->remove('sentences');
    }
    if($request->session()->has('current_challenge')){
        $request->session()->remove('current_challenge');
    }
    if($request->session()->has('challenges_quantity')){
        $request->session()->remove('challenges_quantity');
    }
    if($request->session()->has('progress')){
        $request->session()->remove('progress');
    }
}

return redirect()->route('practice');
}

public function taskInfoIndex($task_id){
    $task = Task::where('id',$task_id)->first();
    $task_info = TaskInfo::where('task_id',$task_id)->first();

    $translations = $task_info->translations;

    foreach ($translations as $translation){
        if($translation->language_id == 1){
            //eng
            $task_info->eng_translation = $translation;
        }
        else if($translation->language_id == 2){
            //srb
            $task_info->srb_translation = $translation;
        }
    }

    /** Task Translation */
    foreach ($task->translations as $translation){
        if($translation->language->id == 2){
            $task->srb_translation = $translation->text;
        }
    }

    $this->data['task'] = $task;
    $this->data['task_info'] = $task_info;
    return view ('client.pages.practice.task-info', $this->data);
}

```

```

    public function streak($request,$user_id,$date_today){
        $user = User::where('id',$user_id)->first();
        $all_user_progress = UserProgress::where('user_id',$user_id);
        $progress_today = $all_user_progress->where('date',$date_today)-
>first();
        if($progress_today->quantity_done == 1){
            //new user_progress row was just inserted; challenge was just
finished
            $user->streak = $user->streak + 1;
            $user->save();
            $request->session()->get('user')->streak = $user->streak;
        }
    }
    public function insertUserProgress($user_id,$date_today){
        $userProgress = UserProgress::where('user_id',$user_id)-
>where('date',$date_today)->first();
        if($userProgress != null){
            $userProgress->quantity_done = $userProgress->quantity_done +
1;
        }
        else{
            $userProgress = new UserProgress;
            $userProgress->user_id = $user_id;
            $userProgress->quantity_done = 1;
            $userProgress->date = $date_today;
        }
        $userProgress->save();
    }
    public function insertUserTask($request,$user_id,$task_id){
        $userTask = UserTask::where('user_id', $user_id)-
>where('task_id',$task_id)->first();
        $task = Task::where('id',$task_id)->first();
        $level_quantity = $task->level_quantity;

        if($userTask != null){
            if($userTask->levels_finished < $level_quantity){
                $userTask->levels_finished = $userTask->levels_finished +
1;
                if($userTask->levels_finished === $level_quantity){
                    $userTask->finished = '1';
                }
            }
        }
        else{
            $userTask = new UserTask;
            $userTask->user_id = $request->session()->get('user')->id;
            $userTask->task_id = $task_id;
            $userTask->finished = 0;
            $userTask->levels_finished = 1;
        }
        $userTask->save();
    }

    //redirecting functions
    public function task(Request $request, $task_id){
        $user_id = $request->session()->get('user')->id;
        $user_language_id = $request->session()->get('user')->language_id;
        $userTask = UserTask::where('user_id',$user_id)-
>where('task_id',$task_id)->first();
        $difficulty_id = 1;
        $task_title = Task::where('id',$task_id)->first()->title;

```



```

        if($task_title == 'basics 2' || $task_title == 'questions 2' ||
$task_title == 'food 2'){
            $difficulty_id = 2;
        }
        if($task_title == 'basics 2'){

        }
        if($task_title == 'questions 2'){

        }
        if($task_title == 'food 2'){

        }

        if($userTask){
            if($userTask->finished){
                return $this->task_matching_pairs($request,
$difficulty_id,$task_title, $user_language_id);
            }
            else{
                if($userTask->levels_finished == 1){
                    return $this->task_write_in($request, $task_id,
$difficulty_id);
                }
                if($userTask->levels_finished == 2){
                    return $this->task_missing_word($request, $task_id,
$difficulty_id, $user_language_id);
                }
                if($userTask->levels_finished == 3 || $userTask-
>levels_finished == 4){
                    return $this->task_matching_pairs($request,
$difficulty_id,$task_title, $user_language_id);
                }
            }
        }
        else{
            return $this->task_matching_pairs($request,
$difficulty_id,$task_title,$user_language_id);
        }
    }

    public function checkAnswer(Request $request, $task_id){
        $type = $request->session()->get('type');
        $user_language_id = $request->session()->get('user')->language_id;

        if($type == 'select_the_matching_pairs'){
            return $this->checkAnswer_matching_pairs($request, $task_id,
$user_language_id);
        }
        else if($type == 'write_this_in_english'){
            return $this->checkAnswer_write_in($request, $task_id,
$user_language_id);
        }
        else if($type == 'select_the_missing_word'){
            return $this->checkAnswer_mising_word($request, $task_id);
        }
    }

    //write this in english/serbian
    public function task_write_in(Request $request, $task_id,
$difficulty_id){
        $current_challenge = 1;

```

```

        $challenges_quantity = 3;
        $words = Word::inRandomOrder()->take(rand(1,3))->get();
        foreach ($words as $word){
            $word->translations;
        }
        $sentences = Sentence::where('task_id',$task_id)-
>where('difficulty_id',$difficulty_id)->take(5)->get();

        $request->session()->put('type','write_this_in_english');
        $request->session()->put('current_challenge', $current_challenge);
        $request->session()-
>put('challenges_quantity',$challenges_quantity);
        $request->session()->put('sentences', $sentences);
        $request->session()->put('progress', 0);

        $this->data['sentence'] = $sentences[$current_challenge - 1];
        $this->data['current_challenge'] = $current_challenge;
        $this->data['challenges_quantity'] = $challenges_quantity;
        $this->data['type'] = 'write_this_in_english';
        $this->data['words'] = $words;

        return view ('client.pages.practice.task', $this->data);
    }
    public function checkAnswer_write_in(Request $request, $task_id,
    $user_language_id){
        $sentences = $request->session()->get('sentences');
        $user_answer = $request->user_answer;
        $sentence_id = $request->sentence_id;
        $stylized_user_answer = strtolower(trim($user_answer));

        //get sentence translation in serbian or english
        $sentence = Sentence::where('id',$sentence_id)->first();
        $translations = $sentence->translations;
        foreach ($translations as $translation){
            if($translation->language->id == $user_language_id){
                $lang_translation = $translation->text;
                $stylized_translation = strtolower(trim($translation-
>text));
                break;
            }
        }

        //if answer is correct
        if($stylized_user_answer == $stylized_translation){
            $current_challenge = $request->session()-
>get('current_challenge');
            $challenges_quantity = $request->session()-
>get('challenges_quantity');
            $this->data['challenges_quantity'] = $challenges_quantity;

            //finished the level
            if($current_challenge == $challenges_quantity){
                $user_id = $request->session()->get('user')->id;
                $date_today = date('Y-m-d');

                //insert in userProgress table-daily progress;quantity and
date of what's been done
                $this->insertUserProgress($user_id,$date_today);

                //streak
                $this->streak($request,$user_id,$date_today);
            }
        }
    }
}

```

```

        //add info about finished level of a task, and if all levels
in the task are finished
        $this->insertUserTask($request,$user_id,$task_id);

        $this->data['task_id'] = $task_id;

        $returnHTML =
view('client.components.practice.finishedchallenge',$this->data)-
>render();// or method that you prefer to return data + RENDER is the key
here
        return response()->json( array('finished' => true,
'html'=>$returnHTML) );
    }
    else{
        $current_challenge = $current_challenge + 1;
        $request->session()->put('current_challenge',
$current_challenge);
        $nextSentence = $sentences[$current_challenge - 1];

        $this->data['sentence'] = $nextSentence;
        $this->data['words'] = Word::inRandomOrder()-
>take(rand(1,3))->get();
        $this->data['current_challenge'] = $current_challenge;

        $returnHTML = view('client.components.practice.write-this-
in-english',$this->data)->render();// or method that you prefer to return
data + RENDER is the key here
        return response()->json( array('success' => true,
'html'=>$returnHTML) );
    }
    }
    else{
        $result = array('correct' => false, 'translation' =>
$lang_translation);
        return Json::encode($result);
    }
}

//select the missing word
public function task_missing_word(Request $request,
$task_id,$difficulty_id, $user_language_id){
    $current_challenge = 1;
    $challenges_quantity = 3;
    $this->data['words'] = Word::inRandomOrder()->take(3)->get();
    $sentences = Sentence::where('task_id',$task_id)
        ->where('text','like','% %')
        ->where('difficulty_id',$difficulty_id)
        ->take($challenges_quantity)
        ->offset(3)
        ->get();

    $current_sentence = $sentences[$current_challenge-1]->text;
    $words_from_the_current_sentence = explode(" ", $current_sentence);
    $random_id = array_rand($words_from_the_current_sentence);
    $answer = $words_from_the_current_sentence[$random_id];

    $words_from_the_current_sentence[$random_id] = "_____";
    $sentence= implode(" ", $words_from_the_current_sentence);

    $request->session()->put('type','select_the_missing_word');

```

```

        $request->session()->put('current_challenge', $current_challenge);
        $request->session()-
>put('challenges_quantity',$challenges_quantity);
        $request->session()->put('sentences', $sentences);
        $request->session()->put('progress', 0);
        //za select the missing word
        $request->session()->put('answer', $answer);

        $this->data['challenges_quantity'] = $challenges_quantity;
        $this->data['current_challenge'] = $current_challenge;
        $this->data['sentence'] = $sentence;
        $this->data['type'] = 'select_the_missing_word';
        //za select the missing word
        $this->data['answer'] = $answer;

        return view ('client.pages.practice.task', $this->data);
    }
    public function checkAnswer_mising_word(Request $request, $task_id){
        $current_challenge = $request->session()->get('current_challenge');
        $challenges_quantity = $request->session()-
>get('challenges_quantity');
        $sentences = $request->session()->get('sentences');
        $answer = $request->session()->get('answer');
        $stylized_answer = strtolower(trim($answer));

        $user_answer = $request->user_answer;
        $stylized_user_answer = strtolower(trim($user_answer));

        //checking if answer is correct
        if($stylized_user_answer == $stylized_answer){
            //finished the level
            if($current_challenge == $challenges_quantity){
                $user_id = $request->session()->get('user')->id;
                $date_today = date('Y-m-d');

                //insert in userProgress table-daily progress;quantity and
date of what's been done
                $this->insertUserProgress($user_id,$date_today);

                //streak
                $this->streak($request,$user_id,$date_today);

                //add info about finished level of a task,and if all levels
in the task are finished
                $this->insertUserTask($request,$user_id,$task_id);

                $this->data['task_id'] = $task_id;
                //for select the missing word
                $request->session()->forget('answer');

                $returnHTML =
view('client.components.practice.finishedchallenge',$this->data)-
>render();// or method that you prefere to return data + RENDER is the key
here
                return response()->json( array('finished' => true,
'html'=>$returnHTML) );
            }
            else{
                $current_challenge = $current_challenge + 1;
                $request->session()->put('current_challenge',
$current_challenge);

```

```

        $nextSentence = $sentences[$current_challenge - 1]->text;

        $words_from_the_next_sentence = explode(" ", $nextSentence);
        $random_id = array_rand($words_from_the_next_sentence);
        $answer = $words_from_the_next_sentence[$random_id];
        $words_from_the_next_sentence[$random_id] = "_____";
        $sentence = implode(" ", $words_from_the_next_sentence);

        $request->session()->put('answer', $answer);

        $this->data['sentence'] = $sentence;
        $this->data['words'] = Word::inRandomOrder()->take(3)-
>get();

        $this->data['answer'] = $answer;
        $this->data['current_challenge'] = $current_challenge;
        $this->data['challenges_quantity'] = $challenges_quantity;

        $returnHTML = view('client.components.practice.select-the-
missing-word', $this->data)->render();// or method that you prefer to return
data + RENDER is the key here
        return response()->json( array('success' => true, 'html' =>
$returnHTML) );
    }

}
else{
    $result = array('correct' => false, 'translation' => $answer);
    return Json::encode($result);
}
}

//select the matching pairs
public function task_matching_pairs(Request
$request, $difficulty_id, $task_title, $user_language_id){
    $current_challenge = 1;
    $challenges_quantity = 3;

    if($task_title == 'basics 2'){
        $task_title = 'basics';
    }
    else if($task_title == 'questions 2'){
        $task_title = 'questions';
    }
    else if($task_title == 'food 2'){
        $task_title = 'food';
    }
    $word_cat_id = WordCategory::where('name', $task_title)->first()-
>id;

    $words = Word::where('word_category_id', $word_cat_id)
        ->where('difficulty_id', $difficulty_id)
        ->inRandomOrder()
        ->take(4 * $challenges_quantity)
        ->get();

    foreach ($words as $word){
        $translations = $word->translations;
        foreach ($translations as $translation){
            if($translation->language->id == $user_language_id){
                $english_translation = $translation->text;
                $stylized_translation =

```

```

strtolower(trim($english_translation));

        $word->translation_id = $translation->id;
        $word->translation_text = $stylized_translation;
        break;
    }
}

$request->session()->put('words', $words);
$slice_degree = 4;
$first_four_words = $words->slice(0, $slice_degree);
$words_left = $words->slice($slice_degree);

$this->data['words'] = $first_four_words;
$this->data['type'] = 'select_the_matching_pairs';
$this->data['challenges_quantity'] = $challenges_quantity;
$this->data['current_challenge'] = $current_challenge;

$request->session()->put('type', 'select_the_matching_pairs');
$request->session()->put('words', $words);
$request->session()->put('words_left', $words_left);
$request->session()->put('current_challenge', $current_challenge);
$request->session()->put('challenges_quantity',
$challenges_quantity);
$request->session()->put('slice_degree', $slice_degree);
$request->session()->put('progress', 0);

return view ('client.pages.practice.task', $this->data);
}

public function checkAnswer_matching_pairs(Request $request,
$task_id,$user_language_id){
    $answer = $request->answer;
    $all_words = $request->session()->get('words');
    $slice_degree = $request->session()->get('slice_degree');
    $words = $request->session()->get('words_left');

    $correct = true;
    $answer_it_word_ids = array();
    foreach($answer as $el){
        array_push($answer_it_word_ids, $el['it_word_id']);
        foreach ($all_words as $word){
            if($el['it_word_id'] == $word->id){
                if($el['translation_id'] != $word->translation_id){
                    $correct = false;
                }
            }
        }
    }
}

try{
    if($correct){
        $current_challenge = $request->session()-
>get('current_challenge');
        $challenges_quantity = $request->session()-
>get('challenges_quantity');

        //finished the level
        if($current_challenge == $challenges_quantity){
            $this->data['task_id'] = $task_id;
            $user_id = $request->session()->get('user')->id;
            $date_today = date('Y-m-d');

```

```

        //insert in userProgress table-daily progress;quantity
and date of what's been done
        $this->insertUserProgress($user_id,$date_today);

        //streak
        $this->streak($request,$user_id,$date_today);

        //add info about finished level of a task,and if all
levels in the task are finished
        $this->insertUserTask($request,$user_id,$task_id);

        $this->data['task_id'] = $task_id;
        //for select the missing word
        $request->session()->forget('answer');

        $returnHTML =
view('client.components.practice.finishedchallenge',$this->data)-
>render();// or method that you prefer to return data + RENDER is the key
here
        return response()->json( array('finished' => true,
'html'=>$returnHTML) );
    }
    else{
        $current_challenge = $current_challenge + 1;
        $request->session()->put('current_challenge',
$current_challenge);
        $this->data['current_challenge'] = $current_challenge;
        $this->data['challenges_quantity'] =
$challenges_quantity;

        $first_four_words = $words->slice(0, $slice_degree);
        $this->data['words'] = $first_four_words;

        $words_left = $words->slice($slice_degree);
        $request->session()->put('words_left',$words_left);

        $returnHTML = view('client.components.practice.select-
the-matching-pairs',$this->data)->render();// or method that you prefer to
return data + RENDER is the key here
        return response()->json(array('success' => true, 'html'
=> $returnHTML));
    }
    else{
        $words = Word::whereIn('id',$answer_it_word_ids)->get();
        foreach ($words as $word){
            $translations = $word->translations;
            foreach ($translations as $translation){
                if($translation->language->id ==
$user_language_id){
                    $english_translation = $translation->text;
                    $stylized_translation =
strtolower(trim($english_translation));

                    $word->translation_id = $translation->id;
                    $word->translation_text =
$stylized_translation;
                    break;
                }
            }
        }
    }
}

```

```

    }

    $result = array('success' => false, 'words' => $words);
    return Json::encode($result);
}
}
catch (\Exception $ex) {
    $result = array('failure' => true, 'text' => $ex-
>getMessage());
    return Json::encode($result);
}
}
}
}

```

5.1.2.6 ProfileController

```

<?php

namespace App\Http\Controllers;

use App\Http\Requests\EditPasswordRequest;
use App\Http\Requests\EditProfileRequest;
use App\Models\Language;
use App\Models\User;
use App\Models\UserAction;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Log;

class ProfileController extends BaseController
{
    public function index(Request $request){
        $user_id = $request->session()->get('user')->id;
        $user = User::where('id', $user_id)->first();
        $userTasks = $user->userTasks;

        $total = 0;
        foreach ($userTasks as $userTask){
            $total += $userTask->levels_finished;
        }

        $this->data['levelsFinished'] = $total;

        return view('client.pages.profile', $this->data);
    }

    public function editProfileIndex(){
        return view('client.pages.edit-profile.edit-profile');
    }

    public function editProfile(EditProfileRequest $request){
        $user_id = $request->session()->get('user')->id;
        $user = User::where('id', $user_id)->first();

        $first_name = $request->input('first_name');
        $last_name = $request->input('last_name');
        $username = $request->input('username');
        $email = $request->input('email');
        $profile_picture = $request->file('profile_picture');
    }
}

```



```

        try{
            $emailExists = DB::table('users')->where('email',$email)
                ->whereNot(function ($query) use ($user_id) {
                    $query->where('id', $user_id);
                })
                ->get()->count();

            $usernameExists = DB::table('users')-
>where('username',$username)
                ->whereNot(function ($query) use ($user_id) {
                    $query->where('id', $user_id);
                })
                ->get()->count();

            if($emailExists > 0 && $usernameExists > 0){
                return back()->with('editProfileFail','Email and username
have been already taken.');
```

```

            }
            if($emailExists > 0 || $usernameExists > 0){
                if($emailExists){
                    return back()->with('editProfileFail','Email has
already been taken.');
```

```

                }
                if($usernameExists > 0){
                    return back()->with('editProfileFail','Username has
already been taken.');
```

```

                }
            }

            if($profile_picture != null){
                $profile_picture_name =
                uniqid()."_.time().".". $profile_picture->extension();
                $profile_picture->storeAs("/public/img/profile_pictures",
                $profile_picture_name);

                $user->profile_picture = $profile_picture_name;
            }

            $user->first_name = $first_name;
            $user->last_name = $last_name;
            $user->username = $username;
            $user->email = $email;
            $user->save();

            $action = new UserAction;
            $action->user_id = $user->id;
            $action->action = 'edit-profile';
            $action->save();

            $request->session()->put('user', $user);

            if($request->session()->get('user')->language_id == 2){
                return redirect()->route('profile')-
>with('editProfileSuccess',"Profil izmenjen uspešno.");;
            }
            else{
                return redirect()->route('profile')-
>with('editProfileSuccess',"Profile edited successfully.");;
            }
        }
    }
}

```

```

    }
    catch (\Exception $ex) {
        Log::error($ex->getMessage());
        return back()->with('editProfileFail', $ex);
    }
}

public function changePasswordIndex(){
    return view('client.pages.edit-profile.change-password');
}

public function changePassword(EditPasswordRequest $request){
    $user_id = $request->session()->get('user')->id;
    $user = User::where('id', $user_id)->first();

    $old_password = md5($request->input('old_password'));
    $new_password = md5($request->input('new_password'));

    try{
        $passwordCorrect = User::where('id',$user_id)-
>where('password',$old_password)->get();

        if($passwordCorrect->count() < 1){
            if($request->session()->get('user')->language_id == 2){
                return back()->with("changePasswordFail","Stara lozinka
nije tačna.");
            }
            else{
                return back()->with("changePasswordFail","Old password
isn't correct.");
            }
        }

        $user->password = $new_password;
        $user->save();

        $action = new UserAction;
        $action->user_id = $user->id;
        $action->action = 'change-password';
        $action->save();

        $request->session()->put('user', $user);

        if($request->session()->get('user')->language_id == 2){
            return redirect()->route('profile')-
>with('changePasswordSuccess',"Lozinka promenjena.");
        }
        else{
            return redirect()->route('profile')-
>with('changePasswordSuccess',"Password changed successfully.");
        }
    }
    catch (\Exception $ex) {
        Log::error($ex->getMessage());
        return back()->with('changePasswordFail', $ex);
    }
}

```

```

public function editGoalIndex(){
    return view('client.pages.edit-profile.edit-goal');
}

public function editGoal(Request $request){
    $user_id = $request->session()->get('user')->id;
    $value = $request->input('daily-goal');

    try{
        $user = User::where('id',$user_id)->first();

        if($value == null){
            return back()->with('editGoalFail', 'Please choose your
daily goal.');
```

```

        }
        $user->daily_goal = $value;
        $user->save();

        $request->session()->put('user', $user);

        if($request->session()->get('user')->language_id == 2){
            return redirect()->route('profile')-
>with('editGoalSuccess',"Dnevni cilj promenjen uspešno.");
        }
        else{
            return redirect()->route('profile')-
>with('editGoalSuccess',"Daily goal changed successfully.");
        }
    }
    catch (\Exception $ex) {
        Log::error($ex->getMessage());
        return back()->with('editGoalFail', $ex);
    }
}

public function deleteProfile(Request $request){
    $user_id = $request->session()->get('user')->id;

    try{
        $user = User::where('id', $user_id)->first();
        $user->active = 0;
        $user->save();

        $request->session()->remove('user');
        return redirect()->route('home');
    }
    catch (\Exception $ex) {
        Log::error($ex->getMessage());
        return back()->with('editGoalFail', $ex);
    }
}

public function changeLanguage(Request $request, $language_id){
    $exists = true;

    if($language_id != 1 && $language_id != 2){
        //1 english
        //2 srpski
    }
}

```

```

        $exists = false;
    }

    if($exists){
        $user_id = $request->session()->get('user')->id;
        $user = User::where('id', $user_id)->first();

        if($language_id != $user->language_id){
            $user->language_id = $language_id;
            $user->save();
        }

        $request->session()->get('user')->language_id =
        $language_id;
    }
}

return back();
}
}

```

5.1.2.7 RecapController

```

<?php

namespace App\Http\Controllers;

use App\Models\Word;
use App\Models\WordCategory;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;
use Psy\Util\Json;

class RecapController extends BaseController
{
    private $words;

    public function __construct(){
        $this->words = Word::all();
    }

    public function index(Request $request, Word $words){

        $words = $words->newQuery();
        $word_categories = WordCategory::all();

        if($request->has('search_word')){
            $text = $request->get('search_word');
            $words->where('text', 'like', '%'.$text.'%');
        }

        if($request->has('word_categories')) {
            $word_cat_id = $request->get('word_categories');
            if($word_cat_id !== '0'){
                $words->where('word_category_id', $word_cat_id);
            }
        }

        $words = $words->paginate(10)->withQueryString();
        //dd($words);
        $data['words'] = $words;
    }
}

```

```

        $data['word_categories'] = $word_categories;
        return view("client.pages.recap", $data);
    }

    public function search(Request $request){
        $search_text = $request->search_word;
        $category_id = $request->word_category_id;

        try{
            if(empty($search_text)){
                if($category_id === '0'){
                    $words = Word::all();
                }
                else{
                    $words = Word::where('word_category_id', $category_id)->get();
                }
            }
            else{
                if($category_id == '0'){
                    $words = Word::where('text','like', '%'.$search_text.'%')->get();
                }
                else{
                    $words = Word::where('text','like', '%'.$search_text.'%')->where('word_category_id', $category_id)->get();
                }
            }

            foreach ($words as $word) {
                foreach ($word->translations as $translation){
                    if($translation->language_id == 1){
                        $word->translation = $translation->text;
                    }
                }
            }

            $result = array('success' => 'success', 'words' => $words);
            return response()->json($result);
        }
        catch (\Exception $ex) {
            $result = array('success' => 'fail', 'text' => $ex->getMessage());
            return Json::encode($result);
        }
    }
}

```

5.1.2.8 SentenceController

```

<?php

namespace App\Http\Controllers;

use App\Http\Requests\EditSentenceRequest;
use App\Http\Requests\InsertSentenceRequest;

```

```

use App\Models\Sentence;
use App\Models\SentenceTranslation;
use App\Models\Task;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;

class SentenceController extends BaseController
{
    private $sentences;
    private $tasks;

    public function __construct(){
        $this->sentences = Sentence::all();
        $this->tasks = Task::all();
    }

    public function index(Request $request, Sentence $sentences, Task
    $tasks)
    {
        $sentences = $sentences->newQuery();

        if($request->has('search-word')){
            $text = $request->get('search-word');
            $sentences->where('text','like','%'.$text.'%');
        }
        if($request->has('tasks')) {
            $task_id = $request->get('tasks');
            if($task_id !== '0'){
                $sentences->where('task_id',$task_id);
            }
        }

        $sentences = $sentences->paginate(10)->withQueryString();

        $this->data['sentences'] = $sentences;
        $this->data['tasks'] = $tasks->get();

        return view('admin.pages.sentences.index', $this->data);
    }

    public function create(Task $tasks)
    {
        $this->data['tasks'] = $tasks->get();
        return view('admin.pages.sentences.create', $this->data);
    }

    public function store(InsertSentenceRequest $request)
    {
        try {
            $text = $request->get('text');
            $task_id = $request->get('task_id');
            $eng_translation = $request->get('eng_translation');
            $srbs_translation = $request->get('srbs_translation');

            $sentence = new Sentence;
            $sentence->text = $text;
            $sentence->task_id = $task_id;

            $sentence->save();

            $translation = new SentenceTranslation();

```

```

        $translation->sentence_id = $sentence->id;
        $translation->language_id = 1; //english
        $translation->text = $eng_translation;
        $translation->save();

        $translation = new SentenceTranslation();
        $translation->sentence_id = $sentence->id;
        $translation->language_id = 2; //serbian
        $translation->text = $srb_translation;
        $translation->save();

        return redirect()->route('sentences.index')->with('success',
"Successfully added.");
    }
    catch(\Exception $ex) {
        Log::error($ex->getMessage());
        return back()->with('fail', 'Error inserting the sentence');
    }
}

public function show($id)
{
    //
}

public function edit($id, Task $tasks)
{
    try{
        $sentence = Sentence::find($id);
        $translations = $sentence->translations;

        foreach ($translations as $translation){
            if($translation->language_id == 1){
                //eng

                $eng_translation = $translation;
            }
            else if($translation->language_id == 2){
                //srb
                $srb_translation = $translation;
            }
        }

        $this->data['sentence'] = $sentence;
        $this->data['tasks'] = $tasks->get();
        $this->data['eng_translation'] = $eng_translation;
        $this->data['srb_translation'] = $srb_translation;

        if($sentence){
            return view('admin.pages.sentences.edit', $this->data);
        }
        else{
            return back();
        }
    }
    catch(\Exception $ex) {
        Log::error($ex->getMessage());
        return back()->with('fail', $ex);
    }
}

```

```

public function update(EditSentenceRequest $request, $id)
{
    try {
        $text = $request->get('text');
        $task_id = $request->get('task_id');
        $eng_translation = $request->get('eng_translation');
        $srb_translation = $request->get('srb_translation');

        $sentence = Sentence::find($id);
        $all_sentences = Sentence::all();

        foreach ($all_sentences as $sen) {
            if($sen->id != $id && $sen->text == $text) {
                $this->data['sentence'] = $sentence;
                $this->data['tasks'] = Task::all();
                $this->data['fail'] = 'Sentence already exists';

                return view('admin.pages.sentences.edit', $this->data);
            }
        }
        $sentence->text = $text;
        $sentence->task_id = $task_id;
        $sentence->save();

        $translation =
        SentenceTranslation::where('sentence_id', $sentence->id)
            ->where('language_id', 1)->first(); //english
        $translation->text = $eng_translation;
        $translation->save();

        $translation =
        SentenceTranslation::where('sentence_id', $sentence->id)
            ->where('language_id', 2)->first(); //serbian
        $translation->text = $srb_translation;
        $translation->save();

        return redirect()->route('sentences.index')->with("success",
        "Successfully edited.");
    }
    catch(\Exception $ex) {
        Log::error($ex->getMessage());
        return back()->with('fail', $ex);
    }
}

public function destroy($id)
{
    try{
        $translations = SentenceTranslation::where('sentence_id', $id)-
        >get();
        foreach ($translations as $translation) {
            SentenceTranslation::destroy($translation->id);
        }
        Sentence::destroy($id);
        return redirect()->back()->with("success", "Successfully
        deleted.");
    }
    catch(\Exception $ex) {
        Log::error($ex->getMessage());
        return back()->with('fail', $ex);
    }
}

```



```
}  
}
```

5.1.2.9 TaskInfoController

```
<?php  
  
namespace App\Http\Controllers;  
  
use App\Http\Requests\EditTaskInfoRequest;  
use App\Models\Task;  
use App\Models\TaskInfo;  
use App\Models\TaskInfoTranslation;  
use Illuminate\Http\Request;  
use Illuminate\Support\Facades\Log;  
  
class TaskInfoController extends BaseController  
{  
    private $tasks;  
  
    public function __construct(){  
        $this->tasks = Task::all();  
    }  
  
    public function index(Task $tasks)  
    {  
        $tasks = $tasks->newQuery();  
  
        $tasks = $tasks->paginate(10)->withQueryString();  
  
        $this->data['tasks'] = $tasks;  
  
        return view('admin.pages.task-info.index', $this->data);  
    }  
  
    public function create()  
    {  
        //  
    }  
  
    public function store(Request $request)  
    {  
        //  
    }  
  
    public function show($id)  
    {  
        //  
    }  
  
    public function edit($id, Task $tasks)  
    {  
        $task = Task::find($id);  
        $task_info = TaskInfo::where('task_id',$id)->first();  
  
        $translations = $task_info->translations;  
  
        foreach ($translations as $translation){  
            if($translation->language_id == 2){  
                //srb
```

```

        $srb_translation = $translation;
    }
}

$this->data['task_info'] = $task_info;
$this->data['tasks'] = $tasks->get();
$this->data['task'] = $task;
$this->data['srb_translation'] = $srb_translation;

if($task){
    return view('admin.pages.task-info.edit', $this->data);
}
else{
    return back();
}
}

public function update(EditTaskInfoRequest $request, $id)
{
    try {
        $intro= $request->get('intro');
        $text = $request->get('text');
        $srb_intro = $request->get('srb_intro');
        $srb_text = $request->get('srb_text');

        $task_info = TaskInfo::find($id);

        $task_info->intro = $intro;
        $task_info->text = $text;
        $task_info->save();

        $translation =
TaskInfoTranslation::where('task_info_id',$task_info->id)
->where('language_id',2)->first(); //serbian
        $translation->intro = $srb_intro;
        $translation->text = $srb_text;
        $translation->save();

        return redirect()->route('task-info.index')->with("success",
"Successfully edited.");
    }
    catch(\Exception $ex) {
        Log::error($ex->getMessage());
        return back()->with('fail', $ex);
    }
}

public function destroy($id)
{
    //
}
}

```

5.1.2.10 WordController

```

<?php

namespace App\Http\Controllers;

use App\Http\Requests\EditWordRequest;

```

```

use App\Http\Requests\InsertWordRequest;
use App\Models\Word;
use App\Models\WordCategory;
use App\Models\WordTranslation;
use App\Models\WordType;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Log;
use Illuminate\Http\Request;

class WordController extends BaseController
{
    private $words;
    private $word_categories;

    public function __construct(){
        $this->words = Word::all();
        $this->word_categories = WordCategory::all();
    }

    public function index(Request $request, Word $words, WordCategory
$word_categories, WordType $word_types)
    {
        $words = $words->newQuery();

        if($request->has('search-word')){
            $text = $request->get('search-word');
            $words->where('text','like','%'.$text.'%');
        }
        if($request->has('word_categories')) {
            $word_cat_id = $request->get('word_categories');
            if($word_cat_id !== '0'){
                $words->where('word_category_id',$word_cat_id);
            }
        }
        if($request->has('word_types')) {
            $word_type_id = $request->get('word_types');
            if($word_type_id !== '0'){
                $words->where('word_type_id',$word_type_id);
            }
        }

        $words = $words->paginate(10)->withQueryString();

        $this->data['words'] = $words;
        $this->data['word_categories'] = $word_categories->get();
        $this->data['word_types'] = $word_types->get();

        return view('admin.pages.words.index', $this->data);
    }

    public function create(WordCategory $word_categories, WordType
$word_types)
    {
        $this->data['word_categories'] = $word_categories->get();
        $this->data['word_types'] = $word_types->get();

        return view('admin.pages.words.create', $this->data);
    }

    public function store(InsertWordRequest $request)
    {

```

```

        try {
            $text = $request->get('text');
            $word_category_id = $request->get('word_category_id');
            $word_type_id = $request->get('word_type_id');
            $eng_translation = $request->get('eng_translation');
            $srbs_translation = $request->get('srbs_translation');
            $audio_file = $request->file('audio_file');

            $word = new Word();
            $word->text = $text;
            $word->word_category_id = $word_category_id;
            $word->word_type_id = $word_type_id;

            $destinationPath = 'audio/words'; //goes directly to the public
folder where assets are
            $myaudio = $audio_file->getClientOriginalName();
            $audio_file->move(public_path($destinationPath), $myaudio);

            $word->audio_file = $myaudio;

            $word->save();

            $translation = new WordTranslation();
            $translation->word_id = $word->id;
            $translation->language_id = 1; //english
            $translation->text = $eng_translation;
            $translation->save();

            $translation = new WordTranslation();
            $translation->word_id = $word->id;
            $translation->language_id = 2; //serbian
            $translation->text = $srbs_translation;
            $translation->save();

            return redirect()->route('words.index')->with('success',
'Successfully added.');
```

```

        }
        catch(\Exception $ex) {
            Log::error($ex->getMessage());
            return back()->with('fail', 'Error inserting the sentence.');
```

```

        }
    }

    public function show($id)
    {
        //
    }

    public function edit($id, WordCategory $word_categories, WordType
$word_types)
    {
        try{
            $word = Word::find($id);
            $translations = $word->translations;
            foreach ($translations as $translation){
                if($translation->language_id == 1){
                    //eng
                    $eng_translation = $translation;
                }
                else if($translation->language_id == 2){
                    //srbs

```

```

        $srb_translation = $translation;
    }
}

if($word){
    $this->data['word'] = $word;
    $this->data['word_categories'] = $word_categories->get();
    $this->data['word_types'] = $word_types->get();
    $this->data['eng_translation'] = $eng_translation;
    $this->data['srb_translation'] = $srb_translation;

    return view('admin.pages.words.edit', $this->data);
}
else{
    return back();
}
}
catch(\Exception $ex) {
    Log::error($ex->getMessage());
    return back()->with('fail', "Error connecting to database");
}
}

public function update(EditWordRequest $request, $id)
{
    try {
        $text = $request->get('text');
        $word_category_id = $request->get('word_category_id');
        $word_type_id = $request->get('word_type_id');
        $eng_translation = $request->get('eng_translation');
        $srb_translation = $request->get('srb_translation');
        $audio_file = $request->file('audio_file');

        $word = Word::find($id);
        $all_words = Word::all();

        foreach ($all_words as $w){
            if($w->id != $id && $w->text == $text){
                $this->data['word'] = $word;
                $this->data['word_categories'] = WordCategory::all();
                $this->data['word_types'] = WordType::all();
                $this->data['fail'] = 'Word already exists';

                return view('admin.pages.words.edit', $this->data);
            }
        }

        $word->text = $text;
        $word->word_category_id = $word_category_id;
        $word->word_type_id = $word_type_id;

        if($audio_file != null){
            $destinationPath = 'audio/words'; //goes directly to the
public folder where assets are
            $myaudio = $audio_file->getClientOriginalName();
            $audio_file->move(public_path($destinationPath), $myaudio);

            $word->audio_file = $myaudio;
        }

        $word->save();
    }
}

```

```

        $translation = WordTranslation::where('word_id',$word->id)
            ->where('language_id',1)->first(); //english
        $translation->text = $eng_translation;
        $translation->save();

        $translation = WordTranslation::where('word_id',$word->id)
            ->where('language_id',2)->first(); //serbian
        $translation->text = $srb_translation;
        $translation->save();

        return redirect()->route('words.index')->with("success",
"Successfully edited.");
    }
    catch(\Exception $ex) {
        Log::error($ex->getMessage());
        return back()->with('fail', $ex);
    }
}

public function destroy($id)
{
    try{
        //dd($id);
        $translations = WordTranslation::where('word_id',$id)->get();
        foreach ($translations as $translation){
            WordTranslation::destroy($translation->id);
        }
        Word::destroy($id);
        return redirect()->back()->with("success", "Successfully
deleted.");
    }
    catch(\Exception $ex) {
        Log::error($ex->getMessage());
        return back()->with('fail', $ex . "Error deleting the entry");
    }
}
}

```

5.1.3 Modeli

5.1.3.0 Department.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Department extends Model
{
    public function employees(){
        return $this->hasMany(Employee::class);
    }
}

```

5.1.3.1 Employee.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Employee extends Model
{
    public function department() {
        return $this->belongsTo(Department::class);
    }
}
```

5.1.3.2 Language.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Language extends Model
{
    //
}
```

5.1.3.3 MainLevel.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class MainLevel extends Model
{
    use HasFactory;
    protected $table = 'main_levels';
}
```

5.1.3.4 Menu.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Menu extends Model
{
    //
}
```

5.1.3.5 Role.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Role extends Model
{
    public function users() {
        return $this->hasMany(User::class);
    }
}
```

5.1.3.6 Sentence.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Sentence extends Model
{
    public function translations() {
        return $this->hasMany(SentenceTranslation::class);
    }

    public function task() {
        return $this->belongsTo(Task::class);
    }
}
```

5.1.3.7 SentenceTranslation.php

```
<?php

namespace App\Models;

use App\Models\Language;
use Illuminate\Database\Eloquent\Model;

class SentenceTranslation extends Model
{
    protected $table = 'sentences_translation';

    public function language() {
        return $this->belongsTo(Language::class);
    }
}
```

5.1.3.8 Task.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
```



```

class Task extends Model
{
    public function translations() {
        return $this->hasMany(TaskTranslation::class);
    }
}

```

5.1.3.9 TaskInfo.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class TaskInfo extends Model
{
    use HasFactory;

    protected $table = 'task_info';

    public function translations() {
        return $this->hasMany(TaskInfoTranslation::class);
    }
}

```

5.1.3.10 TaskInfoTranslation.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class TaskInfoTranslation extends Model
{
    use HasFactory;

    protected $table = 'task_info_translation';

    public function language() {
        return $this->belongsTo(Language::class);
    }
}

```

5.1.3.11 TaskTranslation.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class TaskTranslation extends Model

```

```

{
    use HasFactory;

    protected $table = 'tasks_translation';

    public function language() {
        return $this->belongsTo(Language::class);
    }
}

```

5.1.3.12 User.php

```

<?php

namespace App\Models;

use App\Scopes\UserScope;
use Illuminate\Database\Eloquent\Model;

class User extends Model
{
    protected static function boot()
    {
        parent::boot();

        static::addGlobalScope(new UserScope);
    }

    public function role()
    {
        return $this->belongsTo(Role::class);
    }

    public function userActions() {
        return $this->hasMany(UserAction::class);
    }

    public function userTasks() {
        return $this->hasMany(UserTask::class);
    }

    public function userProgress() {
        return $this->hasMany(UserProgress::class);
    }
}

```

5.1.3.13 UserAction.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class UserAction extends Model
{
    public function user() {
        return $this->belongsTo(User::class);
    }
}

```

```
}  
}
```

5.1.3.14 UserProgress.php

```
<?php  
  
namespace App\Models;  
  
use Illuminate\Database\Eloquent\Model;  
  
class UserProgress extends Model  
{  
    protected $table = 'user_progress';  
}
```

5.1.3.15 UserTask.php

```
<?php  
  
namespace App\Models;  
  
use Illuminate\Database\Eloquent\Model;  
  
class UserTask extends Model  
{  
    protected $table = 'user_task';  
  
    public function task(){  
        return $this->belongsTo(Task::class);  
    }  
}
```

5.1.3.16 WhyLearnALanguage.php

```
<?php  
  
namespace App\Models;  
  
use Illuminate\Database\Eloquent\Model;  
  
class WhyLearnALanguage extends Model  
{  
    protected $table = "why_learn_a_language";  
}
```

5.1.3.17 Word.php

```
<?php  
  
namespace App\Models;  
  
use Illuminate\Database\Eloquent\Model;  
  
class Word extends Model  
{  
    public function translations(){  
        return $this->hasMany(WordTranslation::class);  
    }  
}
```

```

    }

    public function wordCategory() {
        return $this->belongsTo(WordCategory::class);
    }

    public function wordType() {
        return $this->belongsTo(WordType::class);
    }
}

```

5.1.3.18 WordCategory.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class WordCategory extends Model
{
}

```

5.1.3.19 WordTranslation.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class WordTranslation extends Model
{
    protected $table = 'words_translations';

    public function language() {
        return $this->belongsTo(Language::class);
    }
}

```

5.1.3.20 WordType.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class WordType extends Model
{
    use HasFactory;
}

```

5.1.4 Middleware

5.1.4.1 IsAdminMiddleware.php

```
<?php

namespace App\Http\Middleware;

use Closure;

class IsAdminMiddleware
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if($request->session()->has('user') && $request->session()->get('user')->role_id == 1){
            return $next($request);
        }

        return redirect()->route('home');
    }
}
```

5.1.4.2 IsLoggedInMiddleware.php

```
<?php

namespace App\Http\Middleware;

use Closure;

class IsLoggedInMiddleware
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if(!$request->session()->has('user')){
            return redirect()->route('home');
        }

        return $next($request);
    }
}
```

5.1.4.3 IsNotLoggedInMiddleware.php

```
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;

class IsNotLoggedInMiddleware
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure(\Illuminate\Http\Request): (\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse)  $next
     * @return \Illuminate\Http\Response|\Illuminate\Http\RedirectResponse
     */
    public function handle(Request $request, Closure $next)
    {
        if($request->session()->has('user')){
            return redirect()->route('home');
        }

        return $next($request);
    }
}
```

5.1.5 Requests

5.1.5.1 EditPasswordRequest.php

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class EditPasswordRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array<string, mixed>
     */
    public function rules()
    {
        return [
            'old_password' => 'required|max:15|min:6',
            'new_password' => 'required|max:15|min:6',
        ];
    }
}
```

```

    ];
}
}

```

5.1.5.2 EditProfileRequest.php

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class EditProfileRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'first_name' => 'required|max:255',
            'last_name' => 'required|max:255',
            'username' => 'required|max:15|min:1',
            'profile_picture' => 'image|mimes:jpeg,png,jpg,gif,svg',
            'email' => 'required|max:255|email|min:6'
        ];
    }
}

```

5.1.5.3 EditSentenceRequest.php

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class EditSentenceRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }
}

```

```

/**
 * Get the validation rules that apply to the request.
 *
 * @return array<string, mixed>
 */
public function rules()
{
    return [
        'text' => 'required|max:255|min:3',
        'task_id' => 'required|gt:0|exists:tasks,id',
        'eng_translation' => 'required|max:255|min:1',
        'srb_translation' => 'required|max:255|min:1'
    ];
}

```

5.1.5.4 EditTaskInfoRequest.php

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class EditTaskInfoRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return false;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array<string, mixed>
     */
    public function rules()
    {
        return [
            'intro' => 'required|max:255|min:1',
            'text' => 'required|min:1',
            'srb_intro' => 'required|max:255|min:1',
            'srb_text' => 'required|min:1'
        ];
    }
}

```

5.1.5.5 EditWordRequest.php

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

```



```

class EditWordRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array<string, mixed>
     */
    public function rules()
    {
        return [
            'text' => 'required|max:255|min:1',
            'word_category_id' =>
                'required|gt:0|exists:word_categories,id',
            'word_type_id' => 'required|gt:0|exists:word_types,id',
            'eng_translation' => 'required|max:255|min:1',
            'srb_translation' => 'required|max:255|min:1',
            'audio_file' => 'nullable|file|mimes:mp3'
        ];
    }
}

```

5.1.5.6 insertSentenceRequest.php

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class InsertSentenceRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array<string, mixed>
     */
    public function rules()
    {
        return [

```

```

        'text' => 'required|max:255|min:3|unique:sentences',
        'task_id' => 'required|gt:0|exists:tasks,id',
        'eng_translation' => 'required|max:255|min:1',
        'srb_translation' => 'required|max:255|min:1'
    ];
}
}

```

5.1.5.7 InsertWordRequest.php

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class InsertWordRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array<string, mixed>
     */
    public function rules()
    {
        return [
            'text' => 'required|max:255|min:1|unique:words',
            'word_category_id' =>
                'required|gt:0|exists:word_categories,id',
            'word_type_id' => 'required|gt:0|exists:word_types,id',
            'eng_translation' => 'required|max:255|min:1',
            'srb_translation' => 'required|max:255|min:1',
            'audio_file' => 'required|file|mimes:mp3'
        ];
    }
}

```

5.1.5.8 LoginRequest.php

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class LoginRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *

```

```

        * @return bool
        */
public function authorize()
{
    return true;
}

/**
 * Get the validation rules that apply to the request.
 *
 * @return array
 */
public function rules()
{
    return [
        'username' => 'required|max:255|min:1',
        'password' => 'required|max:32|min:6'
    ];
}
}

```

5.1.5.9 RegisterRequest.php

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class RegisterRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'first_name' => 'required|max:255',
            'last_name' => 'required|max:255',
            'username' => 'required|max:15|min:1|unique:users',
            'profile_picture' => 'image|mimes:jpeg,png,jpg,gif,svg',
            'email' => 'required|max:255|email|min:6|unique:users',
            'password' => 'required|min:6|max:15',
            'birth_date' => 'required'
        ];
    }

    public function messages()

```

```

    {
        return [
            'password.required' => 'Password 6-15 characters.'
        ];
    }
}

```

5.1.6 Seed-ovi

5.1.6.1 RoleSeeder.php

```

<?php

use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;

class RoleSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    private $roles = ['admin', 'client'];

    public function run()
    {
        for($i = 0; $i < count($this->roles); $i++){
            DB::table('roles')->insert([
                'name' => $this->roles[$i]
            ]);
        }
    }
}

```

5.1.6.2 TaskInfoSeeder.php

```

<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;

class TaskInfoSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        DB::table('task_info')->insert(array([
            'task_id' => '1',
            'intro' => 'Welcome to the Italian course! Remember that you
can click on the words to see tips of possible translations.',
            'text' => "Personal pronouns

The subject pronouns in Italian are:

```

Io - I
Tu - Singular You (informal)
Lui - He
Lei - She / Singular You (formal)
Esso/Essa - It (archaic and literary)
Noi - We
Voi - Plural You / You all
Loro - They (speaking of people)

Articles

Articles have to match gender and number of the noun they refer to.

The singular determinate articles (the) are:

Lo - masculine, used before Z, S+consonant, GN, and some rarer consonant clusters.

Il - masculine, used before consonants except the above.

La - feminine, used before all consonants.

L' - an elision of the above used before vowels.

The indeterminate articles (a/an) are:

Uno - masculine, used before Z, S+consonant, GN, and some rarer consonant clusters.

Un - masculine, used in all other cases.

Una - feminine, used before all consonants.

Un' - feminine, used before vowels.

```
",
    'outro' => ''
]
));
}
```

5.1.6.3 UserSeeder.php

```
<?php

use App\Models\User;
use App\Models\Role;
use Illuminate\Database\Seeder;
use Faker\Factory as Faker;
use Illuminate\Support\Facades\DB;

class UserSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        DB::table('users')->insert(array([
            'first_name' => 'Admin',
            'last_name' => 'Admin',
            'email' => 'admin@gmail.com',
            'password' => md5('sifra1!'),
            'birth_date' => '1999-01-31',
```

```

        'role_id' => 1
    ],
    [
        'first_name' => 'Teodora',
        'last_name' => 'Nedeljkovic',
        'email' => 'teodora@gmail.com',
        'password' => md5('sifra1!'),
        'birth_date' => '1999-09-14',
        'role_id' => 2
    ]));

    $faker = Faker::create();
    $i = 0;
    while($i<1){
        $id = DB::table('users')->insertGetId([
            'first_name' => $faker->unique()->firstName,
            'last_name' => $faker->unique()->lastName,
            'email' => $faker->unique()->email,
            'password' => md5($faker->password),
            'birth_date' => $faker->date('Y-m-d','now'),
            //role_id' => Role::all()->random()->id
            'role_id' => 2
        ]);

        $i++;
    }
}

```

5.1.6.4 WhyLearnALanguageSeeder.php

```

<?php

use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;

class WhyLearnALanguageSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */

    private $reasons = array('Travel','Education','Brain Training','Family
& Friends','Job Opportunities');
    private $id_names = array('travel','education','brain-
training','family-and-friends','job-opportunities');

    public function run()
    {
        /*for($i = 0; $i < count($this->reasons); $i++){
            DB::table('why_learn_a_language')->insert([
                'title' => $this->reasons[$i],
                'id_name' => $this->id_names[$i]
            ]);
        }*/
    }
}

```

5.1.6.5 WordCategorySeeder.php

```
<?php

use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;

class WordCategorySeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */

    private $categories =
['basics','object','food','number','color','question','day of the
week','time of day',

'place','body','clothing','family','job','household','politics','religion',
'medicine','school','history'];

    public function run()
    {
        for($i = 0; $i<count($this->categories); $i++){
            DB::table('word_categories')->insert([
                'name' => $this->categories[$i]
            ]);
        }
    }
}
```

5.1.6.6 WordTypeSeeder.php

```
<?php

use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;

class WordTypeSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */

    private $types =
['nouns','verbs','adjectives','pronouns','adverbs','prepositions','determin
ers','conjunctions'];

    public function run()
    {
        for($i = 0; $i<count($this->types); $i++){
            DB::table('word types')->insert([
                'name' => $this->types[$i]
            ]);
        }
    }
}
```

```
}
```

5.1.7 Views

5.1.7.1 Admin

5.1.7.1.1 admin/pages/home.blade.php

```
@extends('admin.layouts.layout')

@section('title') Lapsus Linguae - Home @endsection
@section('description')
@endsection
@section('keywords') language, learning, lapsus, linguae, lapsus linguae
@endsection

@section('content')
    <section class="bg white-background admin-home">
        <div class="container">
            <h1 class="text-black">Welcome, {{ session()->get('user')->first_name }}</h1>
        </div>

        <div class="container services section-bg mt-5 flex-column">

            <div class="statistics container">
                <div>
                    <h3>Statistics</h3>
                </div>
                <div class="d-flex">
                    <div class="statistic-component">
                        <i class="fa-solid fa-book"></i>
                        <div>
                            <h4>{{ $words }}</h4>
                            <p>Words</p>
                        </div>
                    </div>
                    <div class="statistic-component">
                        <i class="fa-solid fa-feather"></i>
                        <div class="d-flex flex-column">
                            <h4>{{ $sentences }}</h4>
                            <p>Sentences</p>
                        </div>
                    </div>
                    <div class="statistic-component">
                        <i class="fa-solid fa-list-check"></i>
                        <div class="d-flex flex-column">
                            <h4>{{ $tasks }}</h4>
                            <p>Tasks</p>
                        </div>
                    </div>
                </div>
            </div><br>
            <div class="overview container">
                <div>
                    <h3>Overview</h3>
                </div>
                <div class="d-flex">
                    <div class="statistic-component">
                        <i class="fa-solid fa-user"></i>
                        <div>
                            <h4>{{ $usersLoggedIn }}</h4>

```



```

        <p>Users logged in today</p>
    </div>
</div>
</div>
<div class="d-flex">
    <div class="statistic-component">
        <i class="fa-solid fa-flag-checkered"></i>
        <div>
            <h4>{{ $userProgress }}</h4>
            <p>Finished challenges today</p>
        </div>
    </div>
</div>
</div><br>
<div class="overview container">
    <div>
        <h3>Why are users learning a language</h3>
    </div>
    <div class="d-flex flex-wrap">
        @foreach($whyLearnLanguage as $el)
            <div class="statistic-component">
                <i class="fa-solid fa-language"></i>
                <div>
                    <h4>{{ $el->title }}</h4>
                    <p>{{ $el->totalUsers }}</p>
                </div>
            </div>
        @endforeach
    </div>
</div><br>
</div>
</section>
@endsection

```

5.1.7.1.2 admin/pages/login.blade.php

```

@extends('admin.layouts.layout')

@section('title') Lapsus Linguae - Admin Login @endsection
@section('description')
@endsection
@section('keywords') language, learning, lapsus, linguae, lapsus linguae
@endsection

@section('content')
    <section id="main" class="bg login-container lilac-background">
        <div class="container col-sm-12 col-md-7">
            <h1>Admin Login</h1>
            <form class="m-5" action="{{ route('admin-login') }}"
method="POST">
                @csrf
                <div class="form-group">
                    <input type="text" name="username" class="form-control"
aria-describedby="emailHelp" placeholder="Email or username">
                </div><br/>
                <div class="form-group">
                    <input type="password" name="password" class="form-
control" id="exampleInputPassword1" placeholder="Password">

```

```

        </div><br/>
        <button type="submit" class="btn-get-started btn white-bg-
button">Login</button>
    </form>

    @if(session()->has('loginFail'))
        <div class="alert alert-danger">
            {{ session()->get('loginFail')}}
        </div>
    @endif
    @php
        foreach ($errors->all() as $message) {
            echo '<div class="alert alert-danger">'.
                $message .
                '</div>';
        }
    @endphp

</div>
</section>

@endsection

```

5.1.7.1.3 admin/pages/words/create.blade.php

```

@extends('admin.layouts.layout')

@section('title') Lapsus Linguae - Insert a sentence @endsection
@section('description')
@endsection
@section('keywords') language, learning, lapsus, linguae, lapsus linguae
@endsection

@section('content')

    <section id="main" class="bg lilac-background">
        <div class="container col-sm-12 col-md-7">
            <h2>Insert a word</h2>
            <form class="m-5" action="{{ route('words.store') }}"
method="POST" enctype="multipart/form-data">
                @csrf
                <div class="form-group">
                    <label for="text">Word</label>
                    <input type="text" name="text" id="text" class="form-
control">
                </div><br/>
                <div class="form-group">
                    <label for="word_category">Word category</label>
                    <select name="word_category_id" class="form-control"
id="word_category">
                        <option value="0">Choose...</option>
                        @foreach($word_categories as $cat)
                            <option value="{{ $cat->id }}">{{ $cat->name
}}</option>
                        @endforeach
                    </select>
                </div><br/>
                <div class="form-group">
                    <label for="word_type">Word type</label>
                    <select name="word_type_id" class="form-control"

```

```

id="word_type">
        <option value="0">Choose...</option>
        @foreach($word_types as $type)
            <option value="{{ $type->id }}">{{ $type->name
    }}</option>
        @endforeach
    </select>
</div><br>
<div class="form-group">
    <label for="inputAudio">Audio file (mp3)</label>
    <input type="file" name="audio_file" class="form-
control" id="inputAudio">
</div>
<div class="form-group mt-3">
    <h4>Translations</h4>
</div>
<div class="form-group">
    <label for="eng_translation">English
translation</label>
    <input type="text" name="eng_translation"
id="eng_translation" class="form-control">
</div><br/>
<div class="form-group">
    <label for="srb_translation">Serbian
translation</label>
    <input type="text" name="srb_translation"
id="srb_translation" class="form-control">
</div><br/>
    <button type="submit" class="btn-get-started btn white-bg-
button">Insert</button>
</form>

    @php
        foreach ($errors->all() as $message) {
            echo '<div class="alert alert-danger">'.
                $message .
                '</div>';
        }
    @endphp
</div>
</section>

@endsection

```

5.1.7.1.4 admin/pages/words/index.blade.php

```

@extends('admin.layouts.layout')

@section('title') Lapsus Linguae - Words @endsection
@section('description')
@endsection
@section('keywords') language, learning, lapsus, linguae, lapsus linguae
@endsection

@section('content')

    <section id="main" class="bg white-background admin-index ">
        <div class="container col-sm-12 col-md-9 pt-5">
            <h2>Words</h2>

```

```

        <div class="row">
            @if(session()->has('fail'))
                <div class="alert alert-success">
                    {{ session()->get('fail') }}
                </div>
            @endif
            @if(session()->has('success'))
                <div class="alert alert-success">
                    {{ session()->get('success') }}
                </div>
            @endif
        </div>
        <div class="row">
            <form method="GET" action="{{ route('words.index') }}">
                @csrf
                <div class="form-group">
                    <label class="mb-2" for="search-word">Search
words</label>
                    <input type="search" name="search-word"
class="form-control" placeholder="Type something...">
                </div><br>
                <div class="form-group">
                    <select name="word_categories" class="form-
control">
                        <option value="0">Choose a category</option>
                        @foreach($word_categories as $cat)
                            <option value="{{ $cat->id }}">{{ $cat-
>name }}</option>
                        @endforeach
                    </select>
                </div><br>
                <div class="form-group">
                    <select name="word_types" class="form-control">
                        <option value="0">Choose a type</option>
                        @foreach($word_types as $type)
                            <option value="{{ $type->id }}">{{ $type-
>name }}</option>
                        @endforeach
                    </select>
                </div><br>
                <div class="form-group">
                    <button class="btn-get-started btn orange-bg-
button">Search</button>
                </div>
            </form>
        </div>

        <div class="row mt-4 pt-5">
            <table class="table admin-table">
                <thead>
                    <tr>
                        <th>Italian</th>
                        <th>English</th>
                        <th>Srpski</th>
                        <th>Audio</th>
                        <th>Edit</th>
                        <th>Delete</th>
                    </tr>
                </thead>
                <tbody id="words-table-body">
                    @foreach($words as $word)

```

```

        <tr>
            <td>{{ $word->text }}</td>
            @foreach($word->translations as $translation)
                <td>{{ $translation->text }}</td>
            @endforeach
            <td>
                @if($word->audio_file == null)
                    <i class="fa-solid fa-volume-high
audio-word"></i>
                @else
                    <audio id="word-player-{{ $word->id
}}">
                        <source src="{{
asset('assets/audio/words/'.$word->audio_file) }}" type="audio/mp3">
                    </audio>
                    <i
onclick="document.getElementById('word-player-'+{{ $word->id }}).play()"
class="fa-solid fa-volume-high audio-word"></i>
                @endif
            </td>
            <td><a class="btn-get-started btn orange-bg-
button" href="{{ route('words.edit', ['word' => $word->id]) }}">Edit</a>
            </td>
            <td>
                <form method="POST" action="{{
route('words.destroy', ['word' => $word->id]) }}">
                    @method('DELETE')
                    @csrf
                    <button class="btn-get-started btn
orange-bg-button">Delete</button>
                </form>
            </td>
        </tr>
    @endforeach
</tbody>
</table>
<div class="pagination-container pt-5 pb-5">
    {{ $words->links('pagination::bootstrap-4') }}
</div>
</div>

    @php
        foreach ($errors->all() as $message) {
            echo '<div class="alert alert-danger">'.
                $message .
                '</div>';
        }
    @endphp

</div>
</section>

@endsection

```

5.1.7.1.5 admin/pages/words/edit.blade.php

```

@extends('admin.layouts.layout')

@section('title') Lapsus Linguae - Edit a word @endsection

```

```

@section('description')
@endsection
@section('keywords') language, learning, lapsus, linguae, lapsus linguae
@endsection

@section('content')

    <section id="main" class="bg lilac-background">
        <div class="container col-sm-12 col-md-10">
            <h2>Edit a word</h2>
            <form class="m-5" action=" {{ route('words.update',['word' =>
$word]) }}" method="POST" enctype="multipart/form-data">
                @csrf
                @method('PUT')
                <div class="form-group">
                    <label for="text">Word text</label>
                    <input type="text" name="text" class="form-control"
id="text" value="{{ $word->text }}">
                </div><br/>
                <div class="form-group">
                    <label for="word_cat">Word Category</label>
                    <select name="word_category_id" class="form-control"
id="word_cat">
                        @foreach($word_categories as $cat)
                            <option {{ $cat->id == $word->word_category_id
? "selected" : "" }} value="{{ $cat->id }}">{{ $cat->name }}</option>
                        @endforeach
                    </select>
                </div><br/>
                <div class="form-group">
                    <label for="word_type">Word type</label>
                    <select name="word_type_id" class="form-control"
id="word_type">
                        @foreach($word_types as $type)
                            <option {{ $type->id == $word->word_type_id ?
"selected" : "" }} value="{{ $type->id }}">{{ $type->name }}</option>
                        @endforeach
                    </select>
                </div><br/>
                <div class="form-group">
                    <label for="inputAudio">Audio file (mp3)</label>
                    <input type="file" name="audio_file" class="form-
control" id="inputAudio">
                </div>
                <div class="form-group mt-3">
                    <h4>Translations</h4>
                </div>
                <div class="form-group">
                    <label for="eng_translation">English
translation</label>
                    <input type="text" name="eng_translation"
id="eng_translation" class="form-control" value="{{ $eng_translation->text
}}">
                </div><br/>
                <div class="form-group">
                    <label for="srb_translation">Serbian
translation</label>
                    <input type="text" name="srb_translation"
id="srb_translation" class="form-control" value="{{ $srb_translation->text
}}">
                </div><br/>
            </form>
        </div>
    </section>

```

```

        <button type="submit" class="btn-get-started btn white-bg-
button">Edit</button>
    </form>

    @if(isset($fail))
        <div class="alert alert-danger">
            {{ $fail }}
        </div>
    @endif
    @php
        foreach ($errors->all() as $message) {
            echo '<div class="alert alert-danger">'.
                $message .
                '</div>';
        }
    @endphp
</div>
</section>

@endsection

```

5.1.7.1.6 admin/pages/sentences/edit.blade.php

```

@extends('admin.layouts.layout')

@section('title') Lapsus Linguae - Edit a sentence @endsection
@section('description')
@endsection
@section('keywords') language, learning, lapsus, linguae, lapsus linguae
@endsection

@section('content')

    <section id="main" class="bg lilac-background">
        <div class="container col-sm-12 col-md-10">
            <h2>Edit a sentence</h2>
            <form class="m-5" action=" {{
route('sentences.update', ['sentence' => $sentence]) }}" method="POST">
                @csrf
                @method('PUT')
                <div class="form-group">
                    <label for="text">Sentence text</label>
                    <input type="text" name="text" class="form-control"
id="text" value="{{ {{ $sentence->text }}">
                </div><br/>
                <div class="form-group">
                    <label for="task">Task</label>
                    <select name="task_id" class="form-control" id="task">
                        @foreach($tasks as $task)
                            <option {{ {{ $task->id == $sentence->task_id ?
"selected" : "" }} value="{{ {{ $task->id }}">{{ {{ $task->title }}</option>
                        @endforeach
                    </select>
                </div><br/>
                <h4>Translations</h4>
                <div class="form-group">
                    <label for="eng_translation">English
translation</label>
                    <input type="text" name="eng_translation"
id="eng_translation" class="form-control" value="{{ {{ $eng_translation->text

```

```

}}">
        </div><br/>
        <div class="form-group">
            <label for="srb_translation">Serbian
translation</label>
            <input type="text" name="srb_translation"
id="srb_translation" class="form-control" value="{{ $srb_translation->text
}}">
        </div><br/>
        <button type="submit" class="btn-get-started btn white-bg-
button">Edit</button>
    </form>

    @if(isset($fail))
        <div class="alert alert-danger">
            {{ $fail }}
        </div>
    @endif
    @php
        foreach ($errors->all() as $message) {
            echo '<div class="alert alert-danger">'.
                $message .
                '</div>';
        }
    @endphp
</div>
</section>

@endsection

```

5.1.7.1.7 admin/pages/sentences/index.blade.php

```

@extends('admin.layouts.layout')

@section('title') Lapsus Linguae - Sentences @endsection
@section('description')
@endsection
@section('keywords') language, learning, lapsus, linguae, lapsus linguae
@endsection

@section('content')

    <section id="main" class="bg white-background admin-index">
        <div class="container col-sm-12 col-md-7 pt-5">
            <h2>Sentences</h2>
            <div class="row">
                @if(session()->has('fail'))
                    <div class="alert alert-success">
                        {{ session()->get('fail') }}
                    </div>
                @endif
                @if(session()->has('success'))
                    <div class="alert alert-success">
                        {{ session()->get('success') }}
                    </div>
                @endif
            </div>

            <div class="row">
                <form method="GET" action="{{ route('sentences.index') }}">

```



```

        @csrf
        <div class="form-group">
            <label class="mb-2" for="search-word">Search
sentences</label>
            <input type="search" name="search-word"
class="form-control" placeholder="Type something...">
        </div><br>
        <div class="form-group">
            <select name="tasks" class="form-control">
                <option value="0">Choose a task</option>
                @foreach($tasks as $task)
                    <option value="{{ $task->id }}">{{ $task-
>title }}</option>
                @endforeach
            </select>
        </div><br>
        <div class="form-group">
            <button class="btn-get-started btn orange-bg-
button">Search</button>
        </div>
    </form>
</div>

<!-- from words table -->
<div class="row mt-4 pt-5">
    <table class="table admin-table">
        <thead>
            <tr>
                <th>Italian</th>
                <th>English</th>
                <th>Srpski</th>
                <th>Task</th>
                <th>Edit</th>
                <th>Delete</th>
            </tr>
        </thead>
        <tbody id="words-table-body">
            @foreach($sentences as $sentence)
                <tr>
                    <td>{{ $sentence->text }}</td>
                    @foreach($sentence->translations as
$translation)
                        <td>{{ $translation->text }}</td>
                    @endforeach
                    <td>
                        {{ $sentence->task->title }}
                    </td>
                    <td>
                        <a class="btn-get-started btn orange-bg-
button" href="{{ route('sentences.edit', ['sentence' => $sentence->id])
}}">Edit</a>
                    </td>
                    <td>
                        <form method="POST" action="{{
route('sentences.destroy', ['sentence' => $sentence->id]) }}">
                            @method('DELETE')
                            @csrf
                            <button class="btn-get-started btn
orange-bg-button">Delete</button>
                        </form>
                    </td>
                </tr>
            </tbody>
        </table>
    </div>

```

```

        @endforeach
    </tbody>
</table>
<div class="pagination-container pt-5 pb-5 d-flex">
    {{ $sentences->links('pagination::bootstrap-4') }}
</div>
</div>
<!-- -->

    @php
        foreach ($errors->all() as $message) {
            echo '<div class="alert alert-danger">'.
                $message .
                '</div>';
        }
    @endphp

</div>
</section>

@endsection

```

5.1.7.1.8 admin/pages/sentences/create.blade.php

```

@extends('admin.layouts.layout')

@section('title') Lapsus Linguae - Insert a sentence @endsection
@section('description')
@endsection
@section('keywords') language, learning, lapsus, linguae, lapsus linguae
@endsection

@section('content')

    <section id="main" class="bg lilac-background">
        <div class="container col-sm-12 col-md-7">
            <h2>Insert a sentence</h2>
            <form class="m-5" action=" {{ route('sentences.store') }}"
method="POST">
                @csrf
                <div class="form-group">
                    <label for="text">Sentence text</label>
                    <input type="text" name="text" class="form-control"
id="text" placeholder="Text">
                </div><br/>
                <div class="form-group">
                    <label for="task">Task</label>
                    <select name="task_id" class="form-control" id="task">
                        <option value="0">Choose...</option>
                        @foreach($tasks as $task)
                            <option value="{{ $task->id }}">{{ $task->title
}}</option>
                        @endforeach
                    </select>
                </div><br>
                <h4>Translations</h4>
                <div class="form-group">
                    <label for="eng_translation">English
translation</label>

```

```

        <input type="text" name="eng_translation"
id="eng_translation" class="form-control">
        </div><br/>
        <div class="form-group">
            <label for="srb_translation">Serbian
translation</label>
            <input type="text" name="srb_translation"
id="srb_translation" class="form-control">
            </div><br/>
            <button type="submit" class="btn-get-started btn white-bg-
button">Insert</button>
        </form>

        @php
            foreach ($errors->all() as $message) {
                echo '<div class="alert alert-danger">'.
                    $message .
                    '</div>';
            }
        @endphp

    </div>
</section>

@endsection

```

5.1.7.1.9 admin/fixed/scripts.blade.php

```

<!-- Vendor JS Files -->
<script src="{{
asset('assets/vendor/purecounter/purecounter.js')}}"></script>
<script src="{{ asset('assets/vendor/aos/aos.js')}}"></script>
<script src="{{
asset('assets/vendor/bootstrap/js/bootstrap.bundle.min.js')}}"></script>
<script src="{{
asset('assets/vendor/glightbox/js/glightbox.min.js')}}"></script>
<script src="{{ asset('assets/vendor/isotope-
layout/isotope.pkgd.min.js')}}"></script>
<script src="{{ asset('assets/vendor/swiper/
swiper-
bundle.min.js')}}"></script>
<script src="{{ asset('assets/vendor/php-email-
form/validate.js')}}"></script>

<!-- JQuery File -->
<script src="{{ asset('assets/js/jquery/jquery-3.6.1.min.js') }}"></script>

<!-- Template Main JS File -->
<script src="{{ asset('assets/js/main.js')}}"></script>

```

5.1.7.1.10 admin/fixed/footer.blade.php

```

<footer id="footer">

    <div class="footer-top">
        <div class="container">
            <div class="row">
                <div class="col-lg-3 col-md-6 footer-contact">
                    <h3>Lapsus Linguae</h3>
                    <p>
                        Belgrade<br>

```

```

        Serbia <br><br>
        <strong>Phone:</strong> +1 5589 55488 55<br>
        <strong>Email:</strong> info@example.com<br>
    </p>
</div>

    <div class="col-lg-2 col-md-6 footer-links">
        <h4>Useful Links</h4>
        <ul>
            <li><i class="bx bx-chevron-right"></i> <a
href="#">Home</a></li>
            <li><i class="bx bx-chevron-right"></i> <a
href="#">About us</a></li>
            <li><i class="bx bx-chevron-right"></i> <a
href="#">Terms of service</a></li>
            <li><i class="bx bx-chevron-right"></i> <a
href="#">Privacy policy</a></li>
        </ul>
    </div>

    <div class="col-lg-3 col-md-6 footer-links">
        <h4>Our Services</h4>
        <ul>
            <li><i class="bx bx-chevron-right"></i> <a
href="#">Web Design</a></li>
            <li><i class="bx bx-chevron-right"></i> <a
href="#">Web Development</a></li>
            <li><i class="bx bx-chevron-right"></i> <a
href="#">Product Management</a></li>
            <li><i class="bx bx-chevron-right"></i> <a
href="#">Marketing</a></li>
            <li><i class="bx bx-chevron-right"></i> <a
href="#">Graphic Design</a></li>
        </ul>
    </div>

</div>
</div>
</div>

<div class="container">

    <div class="copyright-wrap d-md-flex py-4">
        <div class="me-md-auto text-center text-md-start">
            <div class="copyright">
                &copy; Copyright <strong><span>Lapsus
Linguae</span></strong>. All Rights Reserved
            </div>
            <div class="credits">
                Designed by <a
href="https://bootstrapmade.com/">Teodora Nedeljković</a>
            </div>
        </div>
        <div class="social-links text-center text-md-right pt-3 pt-md-
0">
            <a href="#" class="twitter"><i class="bx bxl-
twitter"></i></a>
            <a href="#" class="facebook"><i class="bx bxl-
facebook"></i></a>
            <a href="#" class="instagram"><i class="bx bxl-
instagram"></i></a>

```

```

        <a href="#" class="google-plus"><i class="bx bxl-
skype"></i></a>
        <a href="#" class="linkedin"><i class="bx bxl-
linkedin"></i></a>
    </div>
</div>

</div>
</footer>

```

5.1.7.1.11 admin/layouts/layout.blade.php

```

<!DOCTYPE html>
<html>

@include("admin.fixed.head")
@include("admin.fixed.header")

@yield("content")

@include("admin.fixed.footer")

@yield("scripts")

</html>

```

5.1.7.1.12 admin/fixed/header.blade.php

```

<header id="header" class="fixed-top ">
    <div class="container d-flex align-items-center justify-content-
between">
        <h1 class="logo"><a href="{{ route('admin-home') }}">LAPSUS
LINGUAE</a></h1>

        <nav id="navbar" class="navbar">
            <ul>
                @if(session()->has('user'))
                    <li class="dropdown "><a href="#"><span>Words</span> <i
class="bi bi-chevron-down"></i></a>
                        <ul>
                            <li><a href="{{ route('words.create') }}"
class="hover-underline">Insert</a></li>
                            <li><a href="{{ route('words.index') }}"
class="hover-underline">Edit or delete</a></li>
                        </ul>
                    </li>
                    <li class="dropdown "><a
href="#"><span>Sentences</span> <i class="bi bi-chevron-down"></i></a>
                        <ul>
                            <li><a href="{{ route('sentences.create') }}"
class="hover-underline">Insert</a></li>
                            <li><a href="{{ route('sentences.index') }}"
class="hover-underline">Edit or delete</a></li>
                        </ul>
                    </li>
                    <li class="dropdown "><a href="#"><span>Task
Info</span> <i class="bi bi-chevron-down"></i></a>
                        <ul>
                            <li><a href="{{ route('task-info.index') }}"
class="hover-underline">Edit</a></li>

```

```

        </ul>
        </li>
        <li><a href="{{ route('logout') }}" class="hover-
underline">Logout</a></li>
        @endif
    </ul>
    <i class="bi bi-list mobile-nav-toggle"></i>
</nav><!-- .navbar -->
</div>
</header>

```

5.1.7.1.13 admin/pages/task-info/edit.blade.php

```

@extends('admin.layouts.layout')

@section('title') Lapsus Linguae - Edit a sentence @endsection
@section('description')
@endsection
@section('keywords') language, learning, lapsus, linguae, lapsus linguae
@endsection

@section('content')

    <section id="main" class="bg lilac-background">
        <div class="container col-sm-12 col-md-10">
            <h2>Edit the task info</h2>
            <h4 style="border-bottom: antiquewhite solid 2px">Task: {{
$task->title }}</h4>
            <form class="m-5" action=" {{ route('task-
info.update',['task_info' => $task_info]) }}" method="POST">
                @csrf
                @method('PUT')
                <div class="form-group">
                    <label for="intro">Intro</label>
                    <input type="text" name="intro" id="intro" class="form-
control" value="{{ $task_info->intro }}">
                </div><br/>
                <div class="form-group">
                    <label for="text">Text</label>
                    <textarea name="text" id="text" class="form-control"
rows="15">
                        {{ $task_info->text }}
                    </textarea>
                </div><br/>
                <h4>Translations</h4>
                <div class="form-group">
                    <label for="srb_intro">Serbian Intro</label>
                    <input type="text" name="srb_intro" id="srb_intro"
class="form-control" value="{{ $srb_translation->intro }}">
                </div><br/>
                <div class="form-group">
                    <label for="srb_text">Serbian Text</label>
                    <textarea name="srb_text" id="srb_text" class="form-
control" rows="15">
                        {{ $srb_translation->text }}
                    </textarea>
                </div><br/>
                <button type="submit" class="btn-get-started btn white-bg-
button">Edit</button>
            </form>

```

```

        @if(isset($fail))
            <div class="alert alert-danger">
                {{ $fail }}
            </div>
        @endif
        @php
            foreach ($errors->all() as $message) {
                echo '<div class="alert alert-danger">'.
                    $message .
                    '</div>';
            }
        @endphp
    </div>
</section>

@endsection

```

5.1.7.1.14 /admin/pages/task-info/index.blade.php

```

@extends('admin.layouts.layout')

@section('title') Lapsus Linguae - Task Info @endsection
@section('description')
@endsection
@section('keywords') language, learning, lapsus, linguae, lapsus linguae
@endsection

@section('content')

    <section id="main" class="bg white-background admin-index ">
        <div class="container col-sm-12 col-md-9 pt-5">
            <h2>Tasks - Choose to edit info</h2>
            <div class="row">
                @if(session()->has('fail'))
                    <div class="alert alert-success">
                        {{ session()->get('fail') }}
                    </div>
                @endif
                @if(session()->has('success'))
                    <div class="alert alert-success">
                        {{ session()->get('success') }}
                    </div>
                @endif
            </div>

            <div class="row mt-4 pt-5">
                <table class="table admin-table">
                    <thead>
                        <tr>
                            <th>Task</th>
                            <th>Edit</th>
                        </tr>
                    </thead>
                    <tbody id="words-table-body">
                        @foreach($tasks as $task)
                            <tr>
                                <td>{{ $task->title }}</td>
                                <td><a class="btn-get-started btn orange-bg-
button" href="{{ route('task-info.edit', ['task_info' => $task->id])

```

```

}}">Edit info</a>
                                </td>
                                </tr>
                                @endforeach
                                </tbody>
                                </table>
                                <div class="pagination-container pt-5 pb-5">
                                    {{ $tasks->links('pagination::bootstrap-4') }}
                                </div>
                                </div>

                                @php
                                    foreach ($errors->all() as $message) {
                                        echo '<div class="alert alert-danger">'.
                                            $message .
                                            '</div>';
                                    }
                                @endphp

                                </div>
                                </section>

                                @endsection

```

5.1.7.2 Client

5.1.7.2.1 client/pages/home.blade.php

```
@extends('client.layouts.layout')

@section('title') Lapsus Linguae - Learn a Language @endsection
@section('description')
@endsection
@section('keywords') language, learning, lapsus, linguae, lapsus linguae,
italian, english, serbian, study, speak
@endsection

@section('content')
    <section id="hero" class="bg">
        <div>
            <div class="p-4 welcome-container d-flex align-items-lg-center
flex-column align-items-md-end">
                @if(session()->has('user'))
                    @if(session()->get('user')->language_id == 1)
                        <p>ARE YOU</p>
                        <p>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
                        READY</p>
                        <p>TO LEARN</p>
                    @elseif(session()->get('user')->language_id == 2)
                        <p>JESI LI</p>
                        <p>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
                        SPREMAN</p>
                        <p>DA UČIŠ</p>
                    @endif
                @else
                    <p>ARE YOU</p>
```


5.1.7.2.2 client/pages/login.blade.php

```
@extends('client.layouts.layout')

@section('title') @endsection
@section('description')
@endsection
@section('keywords') @endsection

@section('content')

    <section id="main" class="bg login-container lilac-background">
        <div class="container col-sm-12 col-md-7">
            <h1>Login</h1>
            <form class="m-5" action="{{ route('login') }}" method="POST">
                @csrf
                <div class="form-group">
                    <input type="text" name="username" class="form-control"
                    aria-describedby="emailHelp" placeholder="Email or username">
                    <small id="emailHelp" class="form-text text-
                    muted">We'll never share your information with anyone else.</small>
                </div><br/>
                <div class="form-group">
                    <input type="password" name="password" class="form-
                    control" id="exampleInputPassword1" placeholder="Password">
                </div><br/>
                <button type="submit" class="btn-get-started btn white-bg-
                    button">Login</button>
            </form>

            @if(session()->has('loginFail'))
                <div class="alert alert-danger">
                    {{ session()->get('loginFail')}}
                </div>
            @endif
            @php
                foreach ($errors->all() as $message) {
                    echo '<div class="alert alert-danger">'.
                        $message .
                        '</div>';
                }
            @endphp

        </div>
    </section>

@endsection
```

5.1.7.2.3 client/pages/profile.blade.php

```
@extends('client.layouts.layout')

@section('title') Lapsus Linguae - Your Profile @endsection
@section('description')
@endsection
@section('keywords') lapsus linguae, lapsus, linguae, profile, my profile
@endsection
```



```

@endif
@if(session()->has('editGoalSuccess'))
    <div class="container">
        <div class="alert alert-success">
            {{ session()->get('editGoalSuccess') }}
        </div>
    </div>
@endif

<div class="statistics container">
    <div>
        <h3>Statistics</h3>
    </div>
    <div class="d-flex">
        <div class="statistic-component">
            <i class="fa-solid fa-crown"></i>
            <div>
                <h4>{{ $levelsFinished }}</h4>
                @if(session('user')->language_id == 1)
                    <p>Levels finished</p>
                @elseif(session('user')->language_id == 2)
                    <p>Završeno nivoa</p>
                @endif
            </div>
        </div>
        <div class="statistic-component">
            <i class="fa-solid fa-fire"></i>
            <div class="d-flex flex-column">
                <h4>
                    @if(session('user')->streak == null)
                        0
                    @else
                        {{ session('user')->streak }}
                    @endif
                </h4>
                @if(session('user')->language_id == 1)
                    <p>Streak</p>
                @elseif(session('user')->language_id == 2)
                    <p>Dana za redom</p>
                @endif
            </div>
        </div>
    </div>
</div><br>
<div class="information container">
    <div>
        <h3>Information</h3>
    </div>
    <div class="d-flex">
        <div class="statistic-component">
            <i class="fa-solid fa-fire"></i>
            <div class="d-flex flex-column">
                <h4>
                    @if(session()->get('user')->daily_goal == null)
                        0
                    @else
                        {{ session()->get('user')->daily_goal }}
                    @endif
                </h4>
                @if(session('user')->language_id == 1)

```

```

                <p>Daily goal</p>
                @elseif(session('user')->language_id == 2)
                <p>Dnevni cilj</p>
                @endif
            </div>
        </div>
    </div>
    <div class="d-flex">
        <div class="statistic-component">
            <i class="fa-solid fa-earth-americas"></i>
            <div class="d-flex flex-column">
                @if(session('user')->language_id == 1)
                <h4>English</h4>
                <p>Your language</p>
                @elseif(session('user')->language_id == 2)
                <h4>Srpski</h4>
                <p>Vaš jezik</p>
                @endif
            </div>
        </div>
    </div>
</div>
</div>
</section>
@endsection

```

5.1.7.2.4 client/pages/recap.blade.php

```

@extends('client.layouts.layout')

@section('title') Lapsus Linguae - Your Profile @endsection
@section('description')
@endsection
@section('keywords') lapsus linguae, lapsus, linguae, profile, words, recap
@endsection

@section('content')
    <section class="bg white-background recap">
        <div class="d-flex flex-column justify-content-center align-items-center pt-4">
            <div class="d-flex flex-column justify-content-center col-lg-6 mb-4">
                <div>
                    @if(session()->has('user'))
                        @if(session()->get('user')->language_id == 1)
                            <h1>Learn and remember italian words</h1>
                        @elseif(session()->get('user')->language_id == 2)
                            <h1>Nauči i zapamti italijanske reči</h1>
                        @endif
                    @else
                        <h1>Learn and remember italian words</h1>
                    @endif
                </div>
                <div>
                    <form action="{{ route('recap') }}" method="GET">
                        @csrf
                        <div class="form-group pb-3">
                            @if(session()->has('user'))
                                @if(session()->get('user')->language_id ==

```

```

1)
                                <label for="search-word">Search
words</label>
                                @elseif(session()->get('user')->language_id
== 2)
                                <label for="search-word">Pretraži
reči</label>
                                @endif
                                @else
                                <label for="search-word">Search
words</label>
                                @endif
                                <input type="text" id="search-word"
name="search_word" class="form-control">
                                </div>
                                <div class="form-group pb-3">
                                <select name="word_categories" id="word-
categories" class="form-select">
                                @if(session()->has('user'))
                                @if(session()->get('user')->language_id
== 1)
                                <option value="0">Select
category</option>
                                @elseif(session()->get('user')->
language_id == 2)
                                <option value="0">Odaberi
kategoriju</option>
                                @endif
                                @else
                                <option value="0">Select
category</option>
                                @endif
                                @foreach($word_categories as $word_category)
                                <option value="{{ $word_category->id }}"
>{{ $word_category->name }}</option>
                                @endforeach
                                </select>
                                </div>
                                <div class="form-group pb-3">
                                @if(session()->has('user'))
                                @if(session()->get('user')->language_id ==
1)
                                <input type="submit" id="search-words-
button" value="Search" class="btn-get-started btn orange-bg-button">
                                @elseif(session()->get('user')->language_id
== 2)
                                <input type="submit" id="search-words-
button" value="Pretraži" class="btn-get-started btn orange-bg-button">
                                @endif
                                @else
                                <input type="submit" id="search-words-
button" value="Search" class="btn-get-started btn orange-bg-button">
                                @endif
                                </div>
                                </form>
                                </div>
                                </div>
                                <div class="col-lg-6" id="words-table-container">
                                <table class="table" id="words-table">
                                <thead>
                                <tr>

```



```

        @if(session()->has('user'))
            @if(session()->get('user')->language_id ==
1)
                <th>Italian</th>
                <th>English</th>
            @else
                <th>Italijanski</th>
                <th>Srpski</th>
            @endif
        @else
            <th>Italian</th>
            <th>English</th>
            <th>Srpski</th>
        @endif
        <th>Audio</th>
    </tr>
</thead>
<tbody id="words-table-body">
    @foreach($words as $word)
        <tr>
            <td>{{ $word->text }}</td>
            @foreach($word->translations as
$translation)
                @if(session()->has('user'))
                    @if($translation->language_id ==
session()->get('user')->language_id)
                        <td>{{ $translation->text
}}</td>
                    @endif
                @else
                    <td>{{ $translation->text
}}</td>
                @endif
            @endforeach
            <td>
                @if($word->audio_file == null)
                    <i class="fa-solid fa-volume-high audio-
word"></i>
                @else
                    <audio id="word-player-{{ $word->id
}}">
                        <source src="{{
asset('assets/audio/words/'.$word->audio_file) }}" type="audio/mp3">
                    </audio>
                    <i
onclick="document.getElementById('word-player-'+{{ $word->id }}).play()"
class="fa-solid fa-volume-high audio-word"></i>
                @endif
            </td>
        </tr>
    @endforeach
</tbody>
</table>
<div class="pagination-container pt-5 pb-5">
    {{ $words->links('pagination::bootstrap-4') }}
</div>
</div>
</div>
</section>

```

```

@endsection

@section('scripts')
    <script src="{{ asset('assets/js/recap.js') }}"></script>
@endsection

```

5.1.7.2.5 client/pages/register.blade.php

```

@extends('client.layouts.layout')

@section('title') @endsection
@section('description')
@endsection
@section('keywords') lapsus linguae, lapsus, linguae, register, profile
@endsection

@section('content')
    <section id="main" class="bg register-container lilac-background">
        <div class="container survey survey-one active">
            <h1>Why are you learning a language?</h1>
            <section class="reg-services d-flex p-0">
                @foreach($why_learn_a_language as $key => $reason)

                    @include('client.components.register.why-learn-a-
language', [
                        'reason' => $reason,
                        'svg_container' =>
$svg_container_animation[$key]
                    ])
                @endforeach
            </section>
        </div>

        <div class="container col-11 col-md-9 survey survey-two">
            <h1>Register</h1>
            <form class="m-5" id="register-form" action="{{
route('register') }}" method="POST" enctype="multipart/form-data">
                @csrf
                <div class="form-group">
                    <label for="first_name">First Name</label>
                    <input type="text" name="first_name" class="form-
control" id="first_name" value="{{ old('first_name') }}">
                </div>
                <div class="form-group">
                    <label for="last_name">Last Name</label>
                    <input type="text" name="last_name" class="form-
control" id="last_name" value="{{ old('last_name') }}">
                </div>
                <div class="form-group">
                    <label for="username">Username</label>
                    <input type="text" name="username" class="form-
control" id="username" value="{{ old('username') }}">
                </div>
                <div class="form-group">
                    <label for="inputEmail">Email</label>
                    <input type="email" name="email" class="form-
control" id="email" aria-describedby="emailHelp" value="{{ old('email')

```

```

}}">
        <small id="emailHelp" class="form-text text-
muted">We'll never share your email with anyone else.</small>
    </div>
    <div class="form-group">
        <label for="password">Password</label>
        <input type="password" name="password" class="form-
control" id="password" value="{{ old('password') }}">
    </div>
    <div class="form-group">
        <label for="birth_date">Your date of birth</label>
        <input type="date" name="birth_date" class="form-
control" id="birth_date" value="{{ old('birth_date') }}">
    </div>
</form>
<div class="alert alert-danger" id="register-form-errors">

</div>
@if ($errors->any())
    <div class="alert alert-danger" >
        <ul>
            @foreach ($errors->all() as $error)
                <li>{{ $error }}</li>
            @endforeach
        </ul>
    </div>
@endif
@if(session()->has('regFail'))
    <div class="alert alert-danger">
        {{ session()->get('regFail') }}
    </div>
@endif
<div class="p-5">
    <button type="button" class="btn-get-started btn white-
bg-button btn-previous">Previous</button>
    <button type="button" id="validate-register-form"
class="btn-get-started btn white-bg-button active">Next</button>
</div>
</div>

<div class="container col-md-8 col-lg-7 survey survey-three">
    <div class="form-group">
        <h1>Last step!<br> Add a profile picture if you
want.</h1>
        <label for="InputProfilePicture">Profile
Picture</label>
        <input type="file" form="register-form"
name="profile_picture" class="form-control custom-file-input"
id="InputProfilePicture">
    <div>

    </div>
</div>

    <div class="p-5">
        <button type="button" class="btn-get-started btn white-
bg-button btn-previous">Previous</button>
        <button type="submit" form="register-form" class="btn-
get-started btn white-bg-button">Register</button>
    </div>

```

```

        </div>
    </section>
@endsection

@section('scripts')
    <script src="{{ asset('assets/js/register-page.js') }}"></script>
@endsection

```

5.1.7.2.6 client/pages/practice/index.blade.php

```

@extends('client.layouts.layout')

@section('content')

    <section class="bg white-background">
        <div class="pb-5">
            @include('client.components.practice.practice-nav')
            <div class="practice-section d-flex justify-content-center
container pb-5">
                <div class="levels-container">
                    <div class="beginner tasks-container">
                        <div class="castle">
                            <h2>BEGINNER</h2>
                            <div class="image">
                                

                                </div>
                            </div>
                        </div>
                    @php
                        $tmp = array();
                        $i = 0;
                    @endphp

                    @foreach($tasks as $key => $task)

                        @if($task->main_level_id == 1)
                            @if($key % 3 == 0 || $key == 0)

                                <div class="task-container">
                                    @include('client.components.practice.task-bubble',[
                                        'task' => $task, 'disabled' =>
false
                                        ])
                                </div>
                            @else
                                @php
                                    $i++;
                                    array_push($tmp, $task);

                                    if($i == 2){

                                        echo '<div class="task-
container">';

                                        foreach($tmp as $task1){

                                            @endphp
                                        @include('client.components.practice.task-bubble',[
                                            'task' => $task1, 'disabled' =>
false

```

```

        ])
        @php
        }
        echo '</div>';

        $tmp = array();
        $i = 0;
    }
    @endphp

    @endif
    @endif

    @endforeach
</div>
<div class="intermediate tasks-container">
    <div class="castle mt-5">
        <h2>INTERMEDIATE</h2>
        <div class="image">
            

        </div>
    </div>
    @php
        $tmp = array();
        $i = 0;
    @endphp

    @foreach($tasks as $key => $task)

        @if($task->main_level_id == 2)
            @if($key % 3 == 0 || $key == 0)

                <div class="task-container">
@include('client.components.practice.task-bubble',[
                    'task' => $task, 'disabled' =>
!$beginner_level_finished
                ])
            </div>
        @else
            @php
                $i++;
                array_push($tmp, $task);

                if($i == 2){

                    echo '<div class="task-
container">';
                    foreach($tmp as $task1){
                        @endphp
@include('client.components.practice.task-bubble',[
                            'task' => $task1, 'disabled' =>
!$beginner_level_finished
                        ])
                    @php
                    }
                    echo '</div>';

                    $tmp = array();

```

```

        $i = 0;
    }
    @endphp

    @endif
    @endif

    @endforeach
</div>
<div class="advanced tasks-container">
    <div class="castle mt-4">
        <h2>ADVANCED</h2>
        <div class="image">
            

        </div>
    </div>
    @php
        $tmp = array();
        $i = 0;
    @endphp

    @foreach($tasks as $key => $task)

        @if($task->main_level_id == 3)
            @if($key % 3 == 0 || $key == 0)

                <div class="task-container">

@include('client.components.practice.task-bubble',[
                    'task' => $task, 'disabled' =>
!$intermediate_level_finished
                ])
            </div>
            @else
                @php
                    $i++;
                    array_push($tmp, $task);

                    if($i == 2){

                        echo '<div class="task-
container">';

                        @endphp
                        foreach($tmp as $task1){

@include('client.components.practice.task-bubble',[
                            'task' => $task1, 'disabled' =>
!$intermediate_level_finished
                        ])
                        @php
                            }
                            echo '</div>';

                            $tmp = array();
                            $i = 0;
                        }
                    @endphp

                @endif
            @endif

```

```

                @endforeach
            </div>
        </div>

    </div>
</div>
</section>
@endsection

@section("scripts")
    <script src="{{ asset('assets/js/practice.js') }}"></script>
@endsection

```

5.1.7.2.7 client/pages practice/task.blade.php

```

@extends('client.layouts.practice-layout')

@section('content')

    <section class="bg task-background lilac-background" id="background">
        @if($type === 'write_this_in_english')
            @include('client.components.practice.write-this-in-english', ['sentence', $sentence])

            @elseif($type === 'select_the_missing_word')
                @include('client.components.practice.select-the-missing-word', ['sentence', $sentence])

            @elseif($type === 'select_the_matching_pairs')
                @include('client.components.practice.select-the-matching-pairs')
            @endif

    </section>

@endsection

@section('scripts')

    @if($type === 'write_this_in_english')
        <script src="{{ asset('assets/js/write-this-in-english.js') }}"></script>

    @elseif($type === 'select_the_missing_word')
        <script src="{{ asset('assets/js/select-the-missing-word.js') }}"></script>

    @elseif($type === 'select_the_matching_pairs')
        <script src="{{ asset('assets/js/select-the-matching-pairs.js') }}"></script>
    @endif

@endsection

```

5.1.7.2.8 client/pages/practice/task-info.blade.php

```
@extends('client.layouts.layout')

@section('title') @endsection
@section('description')
@endsection
@section('keywords') tips, task info, info, task information, help
@endsection

@section('content')

    <section id="main" class="bg practice-info white-background pb-3">
        <div class="container d-flex flex-wrap pt-5 pb-3">
            <div class="col-md-6">

                @if(session()->get('user')->language_id == 1)
                    <h1>{{ ucfirst($task->title) }}</h1>
                    <p>{{ $task_info->intro }}</p>
                    <p>{!! nl2br($task_info->text) !!}</p>
                @elseif(session()->get('user')->language_id == 2)
                    <h1>{{ ucfirst($task->srb_translation) }}</h1>
                    <p>{{ $task_info->srb_translation->intro }}</p>
                    <p>{!! nl2br($task_info->srb_translation->text) !!}</p>
                @endif

            </div>
            <div class="col-md-6 d-flex justify-content-center align-items-center">
                
            </div>
        </div>
        <div class="container pb-5">
            @if(session()->get('user')->language_id == 1)
                <a href="{{ route('task', $task->id) }}" class="btn-get-started btn orange-bg-button">Try practicing this level</a>
            @elseif(session()->get('user')->language_id == 2)
                <a href="{{ route('task', $task->id) }}" class="btn-get-started btn orange-bg-button">Vežbaj nivo ovde</a>
            @endif
        </div>
    </section>

@endsection
```

5.1.7.2.9 client/pages/edit-profile/change-password.blade.php

```
@extends('client.layouts.layout')

@section('content')

    <section class="bg lilac-background">
        <div class="container pt-5">
            <div class="d-flex justify-content-md-around pt-2 pb-5">
                <div class="col-md-9 container pb-5">
                    <div class="col-lg-7 col-md-11">
                        <form action="{{ route('change-password') }}"
method="POST" id="change-password-form">
                            @csrf
                            <div class="form-group position-relative">
```



```

1) @if(session()->get('user')->language_id ==
    <h1>Change password</h1>
    @elseif(session()->get('user')->language_id
== 2)
    <h1>Promeni lozinku</h1>
    @endif
    <svg class="detail-diamond position-
absolute" xmlns="http://www.w3.org/2000/svg" width="73.485" height="73.485"
viewBox="0 0 73.485 73.485">
        <path id="Path_12275" data-name="Path
12275" d="M14.541,14.541,65.657,5.657,56.772,56.772,5.657,65.657Z"
transform="matrix(0.259, -0.966, 0.966, 0.259, -6.928, 61.956)"
fill="#ffce00"/>
    </svg>
</div>
<div class="form-group">
    @if(session()->get('user')->language_id ==
1)
        <label for="old-password"><span>Old
password</span></label>
    @elseif(session()->get('user')->language_id
== 2)
        <label for="old-password"><span>Stara
lozinka</span></label>
    @endif
    <input type="password" name="old_password"
class="form-control" id="old-password">
    </div><br/>
    <div class="form-group">
        @if(session()->get('user')->language_id ==
1)
            <label for="new-password"><span>New
password</span></label>
        @elseif(session()->get('user')->language_id
== 2)
            <label for="new-password"><span>Nova
lozinka</span></label>
        @endif
        <input type="password" name="new_password"
class="form-control" id="new-password">
    </div><br/>
        @if(session()->get('user')->language_id == 1)
            <button type="submit" class="btn-get-
started btn white-bg-button">Change password</button>
        @elseif(session()->get('user')->language_id ==
2)
            <button type="submit" class="btn-get-
started btn white-bg-button">Promeni lozinku</button>
        @endif
    </form>
    @if ($errors->any())
        <div class="alert alert-danger mt-5" >
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif
    @if(session()->has('changePasswordFail'))

```

```


#### 5.1.7.2.10 client/pages/edit-profile/edit-goal.blade.php



```

@extends('client.layouts.layout')

@section('content')

 <section class="bg lilac-background">
 <div class="container pt-5 pb-5">
 <div class="d-flex justify-content-md-around pt-2 pb-5">
 <div class="col-md-9">
 <div class="col-lg-7 col-md-11">
 <form action=" {{ route('edit-goal') }}"
method="POST" id="edit-goal-form">
 @csrf
 <div class="form-group title">
 @if(session()->get('user')->language_id ==
1)
 <h1>Edit daily goal </h1>
 @elseif(session()->get('user')->language_id
== 2)
 <h1>Izmeni dnevni cilj </h1>
 @endif
 </div>
 <div class="form-group">
 @if(session()->get('user')->language_id ==
1)
 <label for="basic-
goal">Casual1 lesson per day</label>
 @elseif(session()->get('user')->language_id
== 2)
 <label for="basic-
goal">Opušteno1 vezba dnevno</label>
 @endif
 <input type="radio" name="daily-goal"
class="form-control" value="1" id="basic-goal">
 </div>

 <div class="form-group">
 @if(session()->get('user')->language_id ==
1)
 <label for="regular-
goal">Regular2 lessons per day</label>
 @elseif(session()->get('user')->language_id
== 2)
 <label for="regular-
goal">Regularno2 vezbe dnevno</label>

```



114


```

```

        @endif
        <input type="radio" name="daily-goal"
class="form-control" value="2" id="regular-goal">
        </div><br/>
        <div class="form-group">
            @if(session()->get('user')->language_id ==
1)
                <label for="serious-
goal"><strong>Serious</strong><span>3 lessons per day</span></label>
            @elseif(session()->get('user')->language_id
== 2)
                <label for="serious-
goal"><strong>Ozbiljno</strong><span>3 vežbe dnevno</span></label>
            @endif
            <input type="radio" name="daily-goal"
class="form-control" value="3" id="serious-goal">
            </div><br/>
            @if(session()->get('user')->language_id == 1)
                <button type="submit" class="btn-get-
started btn white-bg-button">Save changes</button>
            @elseif(session()->get('user')->language_id ==
2)
                <button type="submit" class="btn-get-
started btn white-bg-button">Sačuvaj izmene</button>
            @endif
        </form>
        @if(session()->has('editGoalFail'))
            <div class="alert alert-danger">
                {{ session()->get('editGoalFail') }}
            </div>
        @endif
    </div>
</div>
<div class="col-md-3" id="edit-profile-navbar-container">
    @include('client.components.edit-profile.edit-profile-
navbar')
</div>
</div>
</div>
</section>
@endsection

```

5.1.7.2.11 client/pages/edit-profile/edit-profile.blade.php

```

@extends('client.layouts.layout')

@section('title') Lapsus Linguae - Your Profile @endsection
@section('description')
@endsection
@section('keywords') lapsus linguae, lapsus, linguae, profile, my profile,
edit, @endsection

@section('content')
    <section class="bg lilac-background">
        <div class="container pt-5 ">
            <div class="d-flex justify-content-md-around pt-2 pb-5">
                <div class="col-md-9" id="edit-profile-container">
                    <div class="col-lg-7 col-md-11">
                        <form name="edit-profile" method="POST" action="{{

```

```

route('edit-profile') }}" enctype="multipart/form-data">
    @csrf
    <div class="form-group position-relative">
        @if(session()->get('user')->language_id ==
1)
            <h1>Edit profile</h1>
        @elseif(session()->get('user')->language_id
== 2)
            <h1>Izmeni profil</h1>
        @endif
        <svg class="detail-diamond position-
absolute" xmlns="http://www.w3.org/2000/svg" width="73.485" height="73.485"
viewBox="0 0 73.485 73.485">
            <path id="Path 12275" data-name="Path
12275" d="M14.541,14.541,65.657,5.657,56.772,56.772,5.657,65.657Z"
transform="matrix(0.259, -0.966, 0.966, 0.259, -6.928, 61.956)"
fill="#ffce00"/>
        </svg>
    </div>
    <div class="form-group">
        @if(session()->get('user')->language_id ==
1)
            <label for="profile_picture">Profile
Picture</label>
        @elseif(session()->get('user')->language_id
== 2)
            <label for="profile_picture">Profilna
slika</label>
        @endif
        <input type="file" name="profile_picture"
class="form-control custom-file-input" id="profile-picture">
    </div>
    <div class="form-group">
        @if(session()->get('user')->language_id ==
1)
            <label for="first_name">First
name</label>
        @elseif(session()->get('user')->language_id
== 2)
            <label for="first_name">Ime</label>
        @endif
        <input type="text" name="first_name"
id="first_name" class="form-control" value="{{ session()->get('user')->first_name }}">
    </div>
    <div class="form-group">
        @if(session()->get('user')->language_id ==
1)
            <label for="last_name">Last
name</label>
        @elseif(session()->get('user')->language_id
== 2)
            <label for="last_name">Prezime</label>
        @endif
        <input type="text" name="last_name"
id="last_name" class="form-control" value="{{ session()->get('user')->last_name }}">
    </div>
    <div class="form-group">
        @if(session()->get('user')->language_id ==
1)

```

```

        <label for="email">Email</label>
        @elseif(session()->get('user')->language_id
== 2)
            <label for="email">Imejl</label>
        @endif
        <input type="text" name="email" id="email"
class="form-control" value="{{ session()->get('user')->email }}">
    </div>
    <div class="form-group">
        @if(session()->get('user')->language_id ==
1)
            <label for="email">Username</label>
        @elseif(session()->get('user')->language_id
== 2)
            <label for="email">Korisničko
ime</label>
        @endif
        <input type="text" name="username"
id="username" class="form-control" value="{{ session()->get('user')->
username }}">
    </div>
    <div class="form-group">
        @if(session()->get('user')->language_id ==
1)
            <input type="submit" class="btn-get-
started btn white-bg-button" value="Save changes">
        @elseif(session()->get('user')->language_id
== 2)
            <input type="submit" class="btn-get-
started btn white-bg-button" value="Sačuvaj izmene">
        @endif
    </div>
</form>
<form method="POST" action="{{route('delete-
profile')}}">
    @csrf
    @method('PUT')
    <div class="form-group">
        @if(session()->get('user')->language_id ==
1)
            <button class="btn-get-started btn
white-bg-button">Delete my profile</button>
        @elseif(session()->get('user')->language_id
== 2)
            <button class="btn-get-started btn
white-bg-button">Izbriši profil</button>
        @endif
    </div>
</form>
@if ($errors->any())
    <div class="alert alert-danger" >
        <ul>
            @foreach ($errors->all() as $error)
                <li>{{ $error }}</li>
            @endforeach
        </ul>
    </div>
@endif
@if(session()->has('editProfileFail'))
    <div class="alert alert-danger">
        {{ session()->get('editProfileFail') }}

```

```

        </div>
    @endif
</div>
</div>
<div class="col-md-3" id="edit-profile-navbar-container">
    @include('client.components.edit-profile.edit-profile-
navbar')
</div>
</div>
</div>
</section>
@endsection

```

5.1.7.2.12 client/pages/about/index.blade.php

```

@extends('client.layouts.layout')

@section('content')
    <section id="hero" class="about bg lilac-background">
        <div class="container">
            <div class="row pb-4">
                <div class="col-md-6 order-1 order-md-2" data-aos="zoom-in"
data-aos-delay="150">
                    
                </div>
                <div class="col-md-6 pt-4 pt-lg-0 order-2 order-md-1
content" data-aos="fade-right">

                    @if(session()->has('user'))
                        @if(session()->get('user')->language_id == 1)
                            <h3>Our teaching approach</h3>
                            <p class="fst-italic">
                                We believe that anyone can learn a
language. Our free, bite-size lessons feel more like a game than a
textbook, and that's by design: Learning is easier when you're having fun.

                                But Lapsus Linguae isn't just a game. It's
based on a methodology proven to foster long-term retention, and a
curriculum aligned to an international standard.
                            </p>
                            <ul>
                                <li><i class="bi bi-check-
circle"></i>&nbsp;<Real-life communication</li>
                                <li><i class="bi bi-check-
circle"></i>&nbsp;<Standout content</li>
                                <li><i class="bi bi-check-circle"></i><A
balanced approach</li>
                            </ul>
                            <div>
                                <a href="{{ route('recap') }}" class="btn-
get-started scrollto white-bg-button">
                                    Start by learning the vocabulary here
                                </a>
                            </div>
                        @elseif(session()->get('user')->language_id == 2)
                            <h3>Naš pristup predavanju</h3>
                            <p class="fst-italic">
                                Verujemo da svako može da nauči jezik. Naše

```



```

        @if(session()->get('user')->language_id == 1)
            <h3>Real-life communication</h3>
            <p>
                Language is ultimately a communication
tool. Lapsus Linguae takes a functional approach by focusing on what
learners actually want to do with a language.
            </p>
            <p>
                Lessons focus on a real-life goal – for
instance, ordering at a restaurant. Learners develop the vocabulary and
grammar needed to achieve that goal through lots of varied practice in
reading, writing, listening, and speaking.
            </p>
            @elseif(session()->get('user')->language_id == 2)
                <h3>Komunikacija u stvarnom životu</h3>
                <p>
                    Jezik je konačno sredstvo komunikacije.
Lapsus Linguae ima funkcionalan pristup fokusirajući se na ono što učenici
zapravo žele da rade sa jezikom.
                </p>
                <p>
                    Lekcije se fokusiraju na stvarni cilj - na
primer, naručivanje u restoranu. Učenici razvijaju rečnik i gramatiku
potrebnu za postizanje tog cilja kroz mnogo različitih vežbi u čitanju,
pisanju, slušanju i govoru.
                </p>
            @endif
        @else
            <h3>Real-life communication</h3>
            <p>
                Language is ultimately a communication tool.
Lapsus Linguae takes a functional approach by focusing on what learners
actually want to do with a language.
            </p>
            <p>
                Lessons focus on a real-life goal – for
instance, ordering at a restaurant. Learners develop the vocabulary and
grammar needed to achieve that goal through lots of varied practice in
reading, writing, listening, and speaking.
            </p>
        @endif
    </div>
</div>

<div class="row detail d-flex">
    <div>
        @if(session()->has('user'))
            @if(session()->get('user')->language_id == 1)
                <h3>Standout content
                </h3>
                <p>Along with commonly used phrases like "Dov'è
il bagno?"
                    (Where is the bathroom?), Lapsus Linguae
learners also encounter sentences
                    like "Il tuo orso beve birra." (Your bear
drinks beer).
                </p>
                <p>
                    Why the quirky sentences?
                    They're memorable and more fun to learn.
                    Our unexpected content also pushes learners

```



```

to think
                                carefully about the language they're
learning.
                                </p>
                                @elseif(session()->get('user')->language_id == 2)
                                <h3>Istaknuti sadržaj
                                </h3>
                                <p>Zajedno sa često korišćenim frazama poput
„Dov'e il bagno?“
                                (Gde je kupatilo?), učenici Lapsus Linguae
se takođe susreću sa rečenicama
                                poput „Il tuo orso beve birra“. (Tvoj
medved pije pivo).
                                </p>
                                <p>
                                Zašto čudne rečenice?
                                Oni su nezaboravni i zabavniji su za
učenje.
                                Naš neočekivani sadržaj takođe podstiče
učenike na razmišljanje
                                pažljivo o jeziku koji uče.
                                </p>
                                @endif
                                @else
                                <h3>Standout content
                                </h3>
                                <p>Along with commonly used phrases like "Dov'è il
bagno?"
                                (Where is the bathroom?), Lapsus Linguae
learners also encounter sentences
                                like "Il tuo orso beve birra." (Your bear
drinks beer).
                                </p>
                                <p>
                                Why the quirky sentences?
                                They're memorable and more fun to learn.
                                Our unexpected content also pushes learners to
think
                                carefully about the language they're learning.
                                </p>
                                @endif
                                </div>
                                <div class="d-flex align-items-center justify-content-
center">
                                
                                </div>
                                </div>

                                <div class="row detail d-flex">
                                <div class="d-flex align-items-center justify-content-
center">
                                
                                </div>
                                <div>
                                @if(session()->has('user'))
                                @if(session()->get('user')->language_id == 1)
                                <h3>A balanced approach</h3>
                                <p>
                                Lapsus Linguae allows learners to discover

```

```

patterns on their own without needing
                                to focus on language rules – the same way
you learned your first language as a
                                child. This approach, called "implicit
learning," is ideal for developing a
                                strong foundational knowledge of a language
and its rules.
                                </p>
                                <p>
                                But explicit instruction is useful for some
concepts. So Lapsus Linguae offers
                                both! In addition to lessons, learners can
access Tips for explanations on
                                grammar, pronunciation, and helpful
phrases.
                                </p>
                                @elseif(session()->get('user')->language_id == 2)
                                <h3>Uravnotežen pristup</h3>
                                <p>
                                Lapsus Linguae omogućava učenicima da sami
otkriju obrasce bez potrebe
                                da se fokusirate na jezička pravila – na
isti način na koji ste naučili svoj prvi jezik kao a
                                dete. Ovaj pristup, nazvan „implicitno
učenje“, idealan je za razvoj a
                                snažno osnovno poznavanje jezika i njegovih
pravila.
                                </p>
                                <p>
                                Ali eksplicitno uputstvo je korisno za neke
koncepte. Dakle, Lapsus Linguae nudi
                                oboje! Pored lekcija, učenici mogu
pristupiti i Savetima za objašnjenja na
                                gramatiku, izgovor i korisne fraze.
                                </p>
                                @endif
                                @else
                                <h3>A balanced approach</h3>
                                <p>
                                Lapsus Linguae allows learners to discover
patterns on their own without needing
                                to focus on language rules – the same way you
learned your first language as a
                                child. This approach, called "implicit
learning," is ideal for developing a
                                strong foundational knowledge of a language and
its rules.
                                </p>
                                <p>
                                But explicit instruction is useful for some
concepts. So Lapsus Linguae offers
                                both! In addition to lessons, learners can
access Tips for explanations on
                                grammar, pronunciation, and helpful phrases.
                                </p>
                                @endif
                                </div>
                                </div>

                                @include('client.pages.about.team',$team)

```

```

        <div class="pt-2 mb-5 row d-flex flex-column align-items-center
            pt-5 pb-5 justify-content-center text-center white-
background detail-circle">
            @if(session()->has('user'))
                @if(session()->get('user')->language_id == 1)
                    <h3>Contact us</h3>
                    <p>Are you having any trouble? Find help here.</p>
                    <p>Journalists and bloggers, please reach out to:
<b style="color:orange">press@lapsuslinguae.com</b></p>
                @elseif(session()->get('user')->language_id == 2)
                    <h3>Kontaktirajte nas</h3>
                    <p>Da li imate poteškoća? Pronađite pomoć ovde.</p>
                    <p>Novinari i blogeri, molimo vas pošaljite nam
poruku ovde: <b style="color:orange">press@lapsuslinguae.com</b></p>
                @endif
            @else
                <h3>Contact us</h3>
                <p>Are you having any trouble? Find help here.</p>
                <p>Journalists and bloggers, please reach out to: <b
style="color:orange">press@lapsuslinguae.com</b></p>
            @endif
        </div>
    </div>
</section>
@endsection

```

5.1.7.2.13 client/pages/about/team.blade.php

```

<div id="team" class="team section-bg pt-5 pb-5">

    <div class="section-title">
        @if(session()->has('user'))
            @if(session()->get('user')->language_id == 1)
                <h3>Our Team</h3>
                <p>Some of the people responsible for our work.</p>
            @elseif(session()->get('user')->language_id == 2)
                <h3>Naš tim</h3>
                <p>Par ljudi zaslužni za naš rad.</p>
            @endif
        @else
            <h3>Our Team</h3>
            <p>Some of the people responsible for our work.</p>
        @endif
    </div>
    <div class="row pb-5">
        @foreach($team as $key => $employee)
            @php $key = $key + 1; @endphp
            @include('client.components.about.teammember',
                ['employee' => $employee,
                 'photo_name' => 'team-'. $key. '.jpg'
                ])
        @endforeach
    </div>
</div>

```

5.1.7.2.14 client/layouts/layout.blade.php

```
<!DOCTYPE html>
<html>
    <!-- include("client.components.loader.loader") -->

    @include("client.fixed.head")
    @include("client.fixed.header")

    @yield("content")

    @include("client.fixed.footer")

    @include("client.fixed.scripts")

    @yield("scripts")

</html>
```

5.1.7.2.15 client/layouts/practice-layout.blade.php

```
<!DOCTYPE html>
<html>
<!-- include("client.components.loader.loader") -->

@include("client.fixed.head")

@yield("content")

@include("client.fixed.scripts")

@yield("scripts")

</html>
```

5.1.7.2.15 client/fixed/header.blade.php

```
<header id="header" class="fixed-top ">
<div class="container d-flex align-items-center justify-content-between">
    <h1 class="logo"><a href="{{ route('home') }}">LAPSUS LINGUA</a></h1>
    <!-- Uncomment below if you prefer to use an image logo -->
    <!-- <a href="index.html" class="logo"></a>-->

    <nav id="navbar" class="navbar">
        <ul>
            @if(session()->has('user') )

                @if(session()->get('user')->language_id == 1)

                    <li><a class="nav-link scrollto hover-underline"
href="{{ route('practice') }}">Practice</a></li>
                    <li><a class="nav-link scrollto hover-underline"
href="{{ route('recap') }}">Words</a></li>
                    <li><a class="nav-link scrollto hover-underline"
href="{{ route('about') }}">About</a></li>
                @elseif(session()->get('user')->language_id == 2)
                    <li><a class="nav-link scrollto hover-underline"
href="{{ route('practice') }}">Vežbe</a></li>
                    <li><a class="nav-link scrollto hover-underline"
```



```

class="hover-underline">Profile</a></li>
        <li><a href='. route("edit-profile") .'
class="hover-underline">Edit profile</a></li>
        <li><a href='. route("logout") .' class="hover-
underline">Logout</a></li>
    </ul>
</li>';
}
else if(session()->get('user')->language_id == 2){
    echo '</span>
</a>
<ul>
        <li><a href='. route("profile") .'
class="hover-underline">Profil</a></li>
        <li><a href='. route("edit-profile") .'
class="hover-underline">Podesi profil</a></li>
        <li><a href='. route("logout") .' class="hover-
underline">Izloguj se</a></li>
    </ul>
</li>';
}
}
else{
    echo '<li><a class="nav-link scrollto hover-underline"
href='. route("login") .'>Login</a></li>';
    echo '<li><a class="nav-link scrollto hover-underline"
href='. route("register") .'>Register</a></li>';
}
@endphp
</ul>
<i class="bi bi-list mobile-nav-toggle"></i>
</nav><!-- .navbar -->
</div>
</header>

```

5.1.7.2.16 client/fixed/footer.blade.php

```

<footer id="footer">
    <div class="footer-top">
        <div class="container">
            <div class="row d-flex justify-content-around justify-content-
sm-between ">
                <div class="col-lg-3 col-md-6 footer-contact">
                    <h3>Lapsus Linguae</h3>
                    @if(session()->has('user'))
                        @if(session()->get('user')->language_id == 1)
                            <p>
                                Belgrade <br>
                                Serbia <br><br>
                                <strong>Phone:</strong> +1 5589 55488
55<br>
                                <strong>Email:</strong>
info@example.com<br>
                            </p>
                        @elseif(session()->get('user')->language_id == 2)
                            <p>
                                Beograd <br>
                                Srbija <br><br>
                                <strong>Telefon:</strong> +1 5589 55488

```

```

55<br>
                                <strong>Imejl:</strong>
info@example.com<br>
                                </p>
                                @endif
                                @else
                                <p>
                                Belgrade <br>
                                Serbia <br><br>
                                <strong>Phone:</strong> +1 5589 55488 55<br>
                                <strong>Email:</strong> info@example.com<br>
                                </p>
                                @endif
                                </div>

                                <div class="col-lg-2 col-md-6 footer-links">

                                @if(session()->has('user'))
                                @if(session()->get('user')->language_id == 1)
                                <h4>Useful Links</h4>
                                <ul>
                                <li><i class="bx bx-chevron-right"></i> <a
href="{{ route('home') }}">Home</a></li>
                                <li><i class="bx bx-chevron-right"></i> <a
href="{{ route('about') }}">About</a></li>
                                <li><i class="bx bx-chevron-right"></i> <a
href="{{ route('profile') }}">Profile</a></li>
                                </ul>
                                @elseif(session()->get('user')->language_id == 2)
                                <h4>Korisni Linkovi</h4>
                                <ul>
                                <li><i class="bx bx-chevron-right"></i> <a
href="{{ route('home') }}">Početna</a></li>
                                <li><i class="bx bx-chevron-right"></i> <a
href="{{ route('about') }}">O nama</a></li>
                                <li><i class="bx bx-chevron-right"></i> <a
href="{{ route('profile') }}">Profil</a></li>
                                </ul>
                                @endif
                                @else
                                <h4>Useful Links</h4>
                                <ul>
                                <li><i class="bx bx-chevron-right"></i> <a
href="{{ route('home') }}">Home</a></li>
                                <li><i class="bx bx-chevron-right"></i> <a
href="{{ route('about') }}">About</a></li>
                                <li><i class="bx bx-chevron-right"></i> <a
href="{{ route('profile') }}">Profile</a></li>
                                </ul>
                                @endif

                                </div>

                                <div class="col-lg-3 col-md-6 footer-links">
                                @if(session()->has('user'))
                                @if(session()->get('user')->language_id == 1)
                                <h4>Our Practice</h4>
                                <ul>
                                <li><i class="bx bx-chevron-right"></i> <a
href="{{ route('practice') }}">Practice</a></li>
                                </ul>

```

```

                @elseif(session()->get('user')->language_id == 2)
                    <h4>Naši zadaci</h4>
                    <ul>
                        <li><i class="bx bx-chevron-right"></i> <a
href="{{ route('practice') }}">Vežbe</a></li>
                    </ul>
                @endif
            @else
                <h4>Our Practice</h4>
                <ul>
                    <li><i class="bx bx-chevron-right"></i> <a
href="{{ route('practice') }}">Practice</a></li>
                </ul>
            @endif
        </div>
    </div>
</div>
</div>

<div class="container">
    <div class="copyright-wrap d-md-flex py-4">
        <div class="me-md-auto text-center text-md-start">
            <div class="copyright">
                &copy; Copyright <strong><span>Lapsus
Linguae</span></strong>.
            </div>
            <div class="credits">
                @if(session()->has('user'))
                    @if(session()->get('user')->language_id == 1)
                        Designed by Teodora Nedeljković
                    @elseif(session()->get('user')->language_id == 2)
                        Dizajnirala Teodora Nedeljković
                    @endif
                @else
                    Designed by Teodora Nedeljković
                @endif
            </div>
        </div>
        <div class="social-links text-center text-md-right pt-3 pt-md-
0">
            <a href="#" class="twitter"><i class="bx bxl-
twitter"></i></a>
            <a href="#" class="facebook"><i class="bx bxl-
facebook"></i></a>
            <a href="#" class="instagram"><i class="bx bxl-
instagram"></i></a>
            <a href="#" class="google-plus"><i class="bx bxl-
skype"></i></a>
            <a href="#" class="linkedin"><i class="bx bxl-
linkedin"></i></a>
        </div>
    </div>
</div>
</footer>

```

5.1.7.2.17 client/fixed/head.blade.php

```

<head>
    <meta charset="utf-8">

```



```

<meta content="width=device-width, initial-scale=1.0" name="viewport">

<title>@yield('title')</title>
<meta content='@yield('description')' name="description">
<meta content='@yield('keywords')' name="keywords">
<meta name="csrf-token" content="{{ csrf_token() }}">

<!-- Favicons -->
<link href="{{ asset('assets/img/favicon.ico') }}" rel="icon">
<link href="{{ asset('assets/img/apple-touch-icon.png') }}" rel="apple-
touch-icon">

<!-- Google Fonts -->
<link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,6
00,600i,700,700i|Roboto:300,300i,400,400i,500,500i,600,600i,700,700i|Poppin
s:300,300i,400,400i,500,500i,600,600i,700,700i" rel="stylesheet">

<!-- Vendor CSS Files -->
<link href="{{ asset('assets/vendor/aos/aos.css') }}" rel="stylesheet">
<link href="{{
asset('assets/vendor/bootstrap/css/bootstrap.min.css') }}" rel="stylesheet">
<link href="{{ asset('assets/vendor/bootstrap-icons/bootstrap-
icons.css') }}" rel="stylesheet">
<link href="{{ asset('assets/vendor/boxicons/css/boxicons.min.css') }}"
rel="stylesheet">
<link href="{{
asset('assets/vendor/glightbox/css/glightbox.min.css') }}" rel="stylesheet">
<link href="{{ asset('assets/vendor/swiper/swiper-bundle.min.css') }}"
rel="stylesheet">

<!-- Template Main CSS File -->
<link href="{{ asset('assets/css/style.css') }}" rel="stylesheet">

<!-- Font Awesome Kit Link -->
<script src="https://kit.fontawesome.com/31a4bd794a.js"
crossorigin="anonymous"></script>

<!-- =====
* Template Name: Techie - v4.7.0
* Template URL: https://bootstrapmade.com/techie-free-skin-bootstrap-3/
* Author: BootstrapMade.com
* License: https://bootstrapmade.com/license/
===== -->
</head>

```

5.1.7.2.18 client/fixed/scripts.blade.php

```

<!-- Vendor JS Files -->
<script src="{{
asset('assets/vendor/purecounter/purecounter.js') }}"></script>
<script src="{{ asset('assets/vendor/aos/aos.js') }}"></script>
<script src="{{
asset('assets/vendor/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
<script src="{{
asset('assets/vendor/glightbox/js/glightbox.min.js') }}"></script>
<script src="{{ asset('assets/vendor/isotope-
layout/isotope.pkgd.min.js') }}"></script>
<script src="{{ asset('assets/vendor/swiper/swiper-
bundle.min.js') }}"></script>

```



```

        <a class="nav-link scrollto" id="show-edit-profile-container"
href="{{ route('edit-profile') }}">Podesi profil</a>
        <a class="nav-link scrollto" id="show-change-password-
container" href="{{ route('change-password') }}">Izmeni šifru</a>
        <a class="nav-link scrollto" href="{{ route('edit-goal')
}}">Izmeni dnevni cilj</a>
    @endif
</div>

```

5.1.7.2.21 client/components/practice/finishedchallenge.blade.php

```

<section class="bg task-background lilac-background" id="background">
    <div class="h-100 d-flex align-items-center justify-content-center
flex-column">
        <div class="container finished-container">
            <div class="w-75 d-flex align-items-center justify-content-
center flex-column">
                <div class="pb-4">
                    <div class="position-relative finished-image">
                        <svg class="detail-circle-bottom"
xmlns="http://www.w3.org/2000/svg" width="30" height="30" viewBox="0 0 30
30">
                            <circle id="Ellipse_279" data-name="Ellipse
279" cx="15" cy="15" r="15"/>
                        </svg>
                        <svg class="detail-circle-bottom"
xmlns="http://www.w3.org/2000/svg" width="30" height="30" viewBox="0 0 30
30">
                            <circle id="Ellipse_279" data-name="Ellipse
279" cx="15" cy="15" r="15"/>
                        </svg>
                        
                    </div>
                </div>
                <div class="progress mb-4">
                    <div class="progress-bar progress-bar-striped bg-
success progress-bar-animated" role="progressbar" aria-valuenow="100" aria-
valuemin="0" aria-valuemax="100" style="width: 100%"></div>
                </div>
            </div>
            <div class="d-flex align-items-center justify-content-
center">
                @if(session()->get('user')->language_id == 1)
                    <p>Practice complete!</p>
                @else
                    <p>Vežba gotova!</p>
                @endif
            </div>
        </div>
    </div>
</div>

    <div class="finished-buttons-container w-100 pt-4">
        <div class="container d-flex justify-content-between">
            <div class="button-container d-flex justify-content-between
align-items-baseline">
                @if(session()->get('user')->language_id == 1)
                    <a href="{{ route('task-info', $task_id) }}"
class="btn-get-started btn white-bg-button">Tips</a>
                @endif
            </div>
        </div>
    </div>

```

```

                @else
                    <a href="{{ route('task-info', $task_id) }}"
class="btn-get-started btn white-bg-button">Pomoć</a>
                @endif

            </div>
            <div class="button-container d-flex justify-content-between
align-items-baseline">
                <a href="{{ route('practice') }}" class="btn-get-
started btn green-bg-button ">
                    @if(session()->get('user')->language_id == 1)
                        Continue
                    @else
                        Nastavi
                    @endif
                </a>
            </div>
        </div>
    </div>
</div>
</section>

```

5.1.7.2.22 client/components/practice/practice-nav.blade.php

[illegible]

5.1.7.2.23 client/components/practice/progress-bar.blade.php

```
<div class="progress-container">
    @php
        //dd($challenges_quantity, $current_challenge);
        $step = 100 / $challenges_quantity;
        $current_percentage = $step * ($current_challenge - 1);
        //dd($current_percentage);
    @endphp
    <div class="x-sign" id="exit-task">
        <a href="{{ route('exit-task') }}">
            <span></span>
            <span></span>
        </a>
    </div>
    <div class="progress">
        <div class="progress-bar bg-warning" aria-valuenow="{{
$current_challenge }}" aria-valuemin="0" aria-valuemax="{{
$challenges_quantity }}" role="progressbar" style="width:
{{ $current_percentage }}%"></div>
    </div>
</div>
```

5.1.7.2.24 client/components/practice/select-the-matching-pairs.blade.php

```
<div class="challenge-container" id="select-the-matching-pairs">
    @include('client.components.practice.progress-
bar',['challenges_quantity' => $challenges_quantity, 'current_challenge' =>
$current_challenge])
    <div class="task-container">
        <div class="question-container">
            <div class="question-container">
                @if(session()->get('user')->language_id == 1)
                    <p>Select the matching pairs</p>
                @else
                    <p>Poveži odgovarajuće parove</p>
                @endif
            </div>
        </div>
        <div class="answer-container">
            <div class="choices-container d-flex ">
                <div class="italian-choices">
                    @foreach($words as $word)
                        <div class="choice">
                            <input type="button" class="btn-get-started btn
white-bg-button choice-input it-word" id="it-word-{{ $word->id }}" data-
id="{{ $word->id }}" value="{{ $word->text }}">
                        </div>
                    @endforeach
                </div>
                <div class="english-choices translations">
                    @php
                        $words = $words->shuffle()
                    @endphp
                    @foreach($words as $word)
                        <div class="choice">
                            <input type="button" class="btn-get-started btn
white-bg-button choice-input translation" id="translation-{{ $word-
>translation_id }}" data-id="{{ $word->translation_id }}" value="{{ $word-
```

```

>translation_text }}">
        </div>
    @endforeach
</div>
</div>
</div>
</div>
<div class="button-container d-flex justify-content-between align-items-baseline">
    <button class="btn-get-started btn white-bg-button" id="reset-choice">
        <span>Reset</span>
    </button>
    <button class="btn-get-started btn white-bg-button check-choice" id="check-choice">
        @if(session()->get('user')->language_id == 1)
            <span>Continue</span>
        @else
            <span>Nastavi</span>
        @endif
    </button>
</div>
<div id="error-container">
    <div class="alert alert-warning in alert-dismissible">
        @if(session()->get('user')->language_id == 1)
            <div>
                <strong>Correct answer:&nbsp;</strong>
            </div>
        @else
            <div>
                <strong>Tačan odgovor:&nbsp;</strong>
            </div>
        @endif
        <div id="error-text">

    </div>

    </div>
</div>
</div>
</div>

```

5.1.7.2.25 client/components/practice/select-the-missing-word.blade.php

```

<div class="challenge-container" id="select-the-missing-word">
    @include('client.components.practice.progress-bar')
    <div class="task-container">
        <div class="question-container">
            @if(session()->get('user')->language_id == 1)
                <p>Select the missing word</p>
            @else
                <p>Odaberi reč koja nedostaje</p>
            @endif
            <strong id="sentence-id">{{ strip_tags($sentence) }}</strong>
        </div>
        <div class="answer-container">
            <div id="answer">
                <p id="answer-text"></p>
            </div>
            <div class="choices-container d-flex ">
                @php

```

```

        $all_words = array();

        foreach ($words as $word){
            array_push($all_words, $word->text);
        }

        array_push($all_words,$answer);

        shuffle($all_words);
        foreach ($all_words as $wordy){
            echo '<div class="choice">
                <input type="button" class="btn-get-
started btn white-bg-button choice-input" value="'. $wordy .' ">
                </div>';
        }
    @endphp
</div>
</div>
</div>

<div class="button-container d-flex justify-content-between align-
items-baseline">
    <button class="btn-get-started btn white-bg-button" id="reset-
choice">
        <span>Reset</span>
    </button>
    <button class="btn-get-started btn white-bg-button check-choice"
id="check-choice">
        @if(session()->get('user')->language_id == 1)
            <span>Continue</span>
        @else
            <span>Nastavi</span>
        @endif
    </button>
</div>
<div id="error-container">
    <div class="alert alert-warning in alert-dismissible">
        @if(session()->get('user')->language_id == 1)
            <strong>Correct answer:&nbsp;</strong><p id="error-
text"></p>
        @else
            <strong>Tačan odgovor:&nbsp;</strong><p id="error-
text"></p>
        @endif
    </div>
</div>
</div>

```

5.1.7.2.26 client/components/practice/task-bubble.blade.php

```

<div style="position: relative" >
    <div @if(!$disabled) class="task task-click" @else class="task task-
disabled" @endif data-id="{{ $task->id }}">
        <div class="task-bubble-ring @if($task->finished === true)
circular-progress @endif">
            {!! $task->symbol !!}
        </div>
        @if(session()->get('user')->language_id == 1)
            <h3>{{ ucfirst($task->title) }}</h3>
        @elseif(session()->get('user')->language_id == 2)

```

```

        <h3>{{ ucfirst($task->srb_translation) }}</h3>
    @endif
</div>
    @include("client.components.practice.task-info-pop-up", ["task" =>
    $task])
</div>

```

5.1.7.2.27 client/components/practice/task-info-pop-up.blade.php

```

<div class="task-info-container" id="task-info-container-{{ $task->id }}"
data-id="{{ $task->id }}">
    <span class="little-arrow">

    </span>
    <div>
        @if($task->finished)
            @if(session()->get('user')->language_id == 1)
                <p>Congratulations! You finished this level!</p>
            @elseif(session()->get('user')->language_id == 2)
                <p>Svaka čast! Završili ste nivo!</p>
            @endif
        @else
            <p>Level {{ $task->levels_finished }}/{{ $task->level_quantity
    }}</p>
        @endif

    </div>
    <div>
        @if(session()->get('user')->language_id == 1)
            <h4>{{ ucfirst($task->title) }}</h4>
        @elseif(session()->get('user')->language_id == 2)
            <h4>{{ ucfirst($task->srb_translation) }}</h4>
        @endif

    </div>
    <div class="link-container">
        @if(session()->get('user')->language_id == 1)
            <a href="{{ route('task', $task->id) }}" class="btn-get-started
btn white-bg-button">Practice</a>
        @elseif(session()->get('user')->language_id == 2)
            <a href="{{ route('task', $task->id) }}" class="btn-get-started
btn white-bg-button">Vežba</a>
        @endif

    </div>
    <div class="link-container">
        @if(session()->get('user')->language_id == 1)
            <a href="{{ route("task-info", $task->id) }}" class="btn-get-
started btn white-bg-button">Tips</a>
        @elseif(session()->get('user')->language_id == 2)
            <a href="{{ route("task-info", $task->id) }}" class="btn-get-
started btn white-bg-button">Pomoć</a>
        @endif

    </div>
</div>

```


5.1.7.2.28 client/components/practice/write-this-in-english.blade.php

```

<div class="challenge-container" id="write-this-in-english">
    @include('client.components.practice.progress-
bar', ['challenges_quantity' => $challenges_quantity, 'current_challenge' =>
$current_challenge])
    <div class="task-container">
        <div class="question-container">
            @if(session()->get('user')->language_id == 1)
                <p>Write this in English</p>
            @else
                <p>Napiši ovo na srpskom</p>
            @endif
            <strong data-id="{{ $sentence->id }}" id="sentence-id">{{
$sentence->text }}</strong>
        </div>
        <div class="answer-container">
            <div id="answer">
                <p id="answer-text"></p>
            </div>
            <div class="choices-container d-flex">

                @php
                    $translations = $sentence->translations;
                    foreach ($translations as $translation){
                        if($translation->language->id == session()-
>get('user')->language_id){
                            $lang_translation = $translation;
                        }
                    }

                    $sentence_words = explode(' ', $lang_translation->text);
                    $all_words = array();

                    foreach ($sentence_words as $key => $sentence_word){
                        array_push($all_words, $sentence_word);
                    }
                    foreach ($words as $word){
                        foreach ($word->translations as $translation){
                            if($translation->language_id == session()-
>get('user')->language_id){
                                array_push($all_words,$translation->text);
                            }
                        }
                    }
                    shuffle($all_words);
                    foreach ($all_words as $key=>$wordy){
                        echo '<div class="choice">
                            <input type="button" class="btn-get-started
btn white-bg-button choice-input" id="'. $key.'" value="'. $wordy .'">
                            </div>';
                    }
                @endphp

            </div>
        </div>
        <div class="button-container d-flex justify-content-between align-
items-baseline">
            <button class="btn-get-started btn white-bg-button" id="reset-
choice">
                <span>Reset</span>

```

```

        </button>
        <button class="btn-get-started btn white-bg-button check-choice"
id="check-choice">
            @if(session()->get('user')->language_id == 1)
                <span>Continue</span>
            @else
                <span>Nastavi</span>
            @endif
        </button>
    </div>
    <div id="error-container">
        <div class="alert alert-warning in alert-dismissible">
            @if(session()->get('user')->language_id == 1)
                <strong>Correct answer:&nbsp;</strong><p id="error-
text"></p>
            @else
                <strong>Tačan odgovor:&nbsp;</strong><p id="error-
text"></p>
            @endif
        </div>
    </div>
</div>

```

5.1.7.2.29 client/components/register/why-learn-a-language.blade.php

```

<div>
    <label for="{{ $reason->id_name }}">
        <div data-aos="zoom-in" class="pointer" data-aos-delay="300">
            <div class="icon-box iconbox-teal">
                {!! $svg_container !!}
                <p>{{ $reason->title }}</p>
            </div>
        </div>
    </label>
</div>
<input type="radio" form="register-form" id="{{ $reason->id_name }}"
name="why-learn-a-language" value="{{ $reason->id }}" class="d-none">

```

5.2 JS

5.2.1 main.js

```

/**
 * Template Name: Techie - v4.7.0
 * Template URL: https://bootstrapmade.com/techie-free-skin-bootstrap-3/
 * Author: BootstrapMade.com
 * License: https://bootstrapmade.com/license/
 */
(function() {
    /*LOADER ANIMATION*/
    window.onload = function () {
        //document.getElementById("loader-wrapper").style.display = "none";
    }
    /*LOADER ANIMATION*/
    "use strict";
    /** Capitalize first letter - function ...*/
    function capitalizeFirstLetter(string) {
        return string.charAt(0).toUpperCase() + string.slice(1);
    }
}
/**

```

```

* Easy selector helper function
*/
const select = (el, all = false) => {
  el = el.trim()
  if (all) {
    return [...document.querySelectorAll(el)]
  } else {
    return document.querySelector(el)
  }
}

/**
* Easy event listener function
*/
const on = (type, el, listener, all = false) => {
  let selectEl = select(el, all)
  if (selectEl) {
    if (all) {
      selectEl.forEach(e => e.addEventListener(type, listener))
    } else {
      selectEl.addEventListener(type, listener)
    }
  }
}

/**
* Easy on scroll event listener
*/
const onscroll = (el, listener) => {
  el.addEventListener('scroll', listener)
}

/**
* Navbar links active state on scroll
*/
let navbarlinks = select('#navbar .scrollto', true)
const navbarlinksActive = () => {
  let position = window.scrollY + 200
  navbarlinks.forEach(navbarlink => {
    if (!navbarlink.hash) return
    let section = select(navbarlink.hash)
    if (!section) return
    if (position >= section.offsetTop && position <= (section.offsetTop +
section.offsetHeight)) {
      navbarlink.classList.add('active')
    } else {
      navbarlink.classList.remove('active')
    }
  })
}
window.addEventListener('load', navbarlinksActive)
onscroll(document, navbarlinksActive)

/**
* Scrolls to an element with header offset
*/
const scrollto = (el) => {
  let header = select('#header')
  let offset = header.offsetHeight

```

```

    if (!header.classList.contains('header-scrolled')) {
      offset -= 16
    }

    let elementPos = select(el).offsetTop
    window.scrollTo({
      top: elementPos - offset,
      behavior: 'smooth'
    })
  }

  /**
   * Toggle .header-scrolled class to #header when page is scrolled
   */
  let selectHeader = select('#header')
  if (selectHeader) {
    const headerScrolled = () => {
      if (window.scrollY > 100) {
        selectHeader.classList.add('header-scrolled')
      } else {
        selectHeader.classList.remove('header-scrolled')
      }
    }
    window.addEventListener('load', headerScrolled)
    onscroll(document, headerScrolled)
  }

  /**
   * Back to top button
   */
  let backtotop = select('.back-to-top')
  if (backtotop) {
    const toggleBacktotop = () => {
      if (window.scrollY > 100) {
        backtotop.classList.add('active')
      } else {
        backtotop.classList.remove('active')
      }
    }
    window.addEventListener('load', toggleBacktotop)
    onscroll(document, toggleBacktotop)
  }

  /**
   * Mobile nav toggle
   */
  on('click', '.mobile-nav-toggle', function(e) {
    select('#navbar').classList.toggle('navbar-mobile')
    this.classList.toggle('bi-list')
    this.classList.toggle('bi-x')
  })

  /**
   * Mobile nav dropdowns activate
   */
  on('click', '.navbar .dropdown > a', function(e) {
    if (select('#navbar').classList.contains('navbar-mobile')) {
      e.preventDefault()
      this.nextElementSibling.classList.toggle('dropdown-active')
    }
  }, true)

```

```

/**
 * Scroll with offset on links with a class name .scrollto
 */
on('click', '.scrollto', function(e) {
  if (select(this.hash)) {
    e.preventDefault()

    let navbar = select('#navbar')
    if (navbar.classList.contains('navbar-mobile')) {
      navbar.classList.remove('navbar-mobile')
      let navbarToggle = select('.mobile-nav-toggle')
      navbarToggle.classList.toggle('bi-list')
      navbarToggle.classList.toggle('bi-x')
    }
    scrollTo(this.hash)
  }
}, true)

/**
 * Scroll with offset on page load with hash links in the url
 */
window.addEventListener('load', () => {
  if (window.location.hash) {
    if (select(window.location.hash)) {
      scrollTo(window.location.hash)
    }
  }
});

/**
 * Preloader
 */
let preloader = select('#preloader');
if (preloader) {
  window.addEventListener('load', () => {
    preloader.remove()
  });
}

/**
 * Testimonials slider
 */
new Swiper('.testimonials-slider', {
  speed: 600,
  loop: true,
  autoplay: {
    delay: 5000,
    disableOnInteraction: false
  },
  slidesPerView: 'auto',
  pagination: {
    el: '.swiper-pagination',
    type: 'bullets',
    clickable: true
  },
  breakpoints: {
    320: {
      slidesPerView: 1,
      spaceBetween: 40
    },

```

```

        1200: {
            slidesPerView: 3,
            spaceBetween: 40
        }
    }
});

/**
 * Portfolio isotope and filter
 */
window.addEventListener('load', () => {
    let portfolioContainer = select('.portfolio-container');
    if (portfolioContainer) {
        let portfolioIsotope = new Isotope(portfolioContainer, {
            itemSelector: '.portfolio-item'
        });

        let portfolioFilters = select('#portfolio-filters li', true);

        on('click', '#portfolio-filters li', function(e) {
            e.preventDefault();
            portfolioFilters.forEach(function(el) {
                el.classList.remove('filter-active');
            });
            this.classList.add('filter-active');

            portfolioIsotope.arrange({
                filter: this.getAttribute('data-filter')
            });
            portfolioIsotope.on('arrangeComplete', function() {
                AOS.refresh()
            });
        }, true);
    }
});

/**
 * Initiate portfolio lightbox
 */
const portfolioLightbox = GLightbox({
    selector: '.portfolio-lightbox'
});

/**
 * Portfolio details slider
 */
new Swiper('.portfolio-details-slider', {
    speed: 400,
    loop: true,
    autoplay: {
        delay: 5000,
        disableOnInteraction: false
    },
    pagination: {
        el: '.swiper-pagination',
        type: 'bullets',
        clickable: true
    }
});

```

```

/**
 * Animation on scroll
 */
window.addEventListener('load', () => {
  AOS.init({
    duration: 1000,
    easing: 'ease-in-out',
    once: true,
    mirror: false
  })
});

/*****
***** */
/**
 * Next/Previous logic
 */
let currentSurveyIndex = 0;
let surveys = document.querySelectorAll('.survey');
let surveysAllButActive =
document.querySelectorAll('.survey:not(.active)');

for(let i = 0; i < surveysAllButActive.length; i++){
  surveysAllButActive[i].style.display = "none";
}

function nextPrevious(currentSurveyIndex){
  document.querySelector('.survey.active').style.display = 'none';
  document.querySelector('.survey.active').classList.remove('active');

  surveys[currentSurveyIndex].classList.add('active');
  surveys[currentSurveyIndex].style.display = "flex";

  if(currentSurveyIndex === 0) {
    document.querySelector(".btn-previous").disabled = true;
  }
  else if(currentSurveyIndex === surveys.length - 1){
    document.querySelector(".btn-next").disabled = true;
  }
  else{
    document.querySelector(".btn-previous").disabled = false;
    document.querySelector(".btn-next").disabled = false;
  }
}

let icon_box = document.querySelectorAll(".icon-box");
for(let i=0; i<icon_box.length; i++){
  icon_box[i].addEventListener('click',function (){
    currentSurveyIndex++;
    nextPrevious(currentSurveyIndex)
  })
}

let buttonsPrevious = document.querySelectorAll(".btn-previous");
for(let i=0; i<buttonsPrevious.length; i++){
  buttonsPrevious[i].addEventListener('click',function (){
    currentSurveyIndex--;
    nextPrevious(currentSurveyIndex)
  })
}

```

```

    }
    let buttonsNext = document.querySelectorAll(".btn-next");
    for(let i=0; i<buttonsNext.length; i++){
        buttonsNext[i].addEventListener('click',function (){
            currentSurveyIndex++;
            nextPrevious(currentSurveyIndex)
        })
    }

    /** Progress bar */
    /** circular */

    let circular_progress = document.querySelectorAll('.circular-
progress')
    for(let i = 0; i<circular_progress.length; i++){
        let progressValue = 0;
        let progressEndValue = 100;
        let speed = 8;

        let progress = setInterval(() => {
            progressValue++;
            circular_progress[i].style.background = `conic-gradient(
                #22b571 ${progressValue * 3.6}deg,
                lightgreen ${progressValue * 3.6}deg
            )`;
            if(progressValue === progressEndValue){
                clearInterval(progress);
            }
        },speed)
    }

    /** linear */

    /** Edit Profile & Change profile forms */
    /*document.querySelector("#show-edit-profile-
container").addEventListener('click',function (){
        document.querySelector('#edit-profile-container').style.display =
'block';
        document.querySelector('#change-password-container').style.display
= 'none';
    })
    document.querySelector("#show-change-password-
container").addEventListener('click',function (){
        document.querySelector('#change-password-container').style.display
= 'block';
        document.querySelector('#edit-profile-container').style.display =
'none';
    })*//

    /** Animation for click on edit goal */
    $(document).on('click', '#edit-goal-form label', function(){
        $('#edit-goal-form label').css('background','none');
        this.style.background = 'rgba(255,255,255,0.4)';
    })
})()

```


5.2.2 practice.js

```
$(document).on('click', '.task-click', function(){

    var task_id = $(this).data('id')
    var clicked = $('#task-info-container-'+task_id)
    var all_pop_ups = $('.task-info-container');

    if(clicked.css('display') === 'none'){
        clicked.css('display','block');
    }
    else{
        clicked.css('display','none');
    }

    all_pop_ups.each(function(index){
        if($(this).data('id') !== task_id){
            $(this).css('display','none')
        }
    })
})
```

5.2.3 register-page.js

```
(function() {
    /**
     * Next/Previous logic
     */
    let currentSurveyIndex = 0;
    let surveys = document.querySelectorAll('.survey');
    let surveysAllButActive =
document.querySelectorAll('.survey:not(.active)');

    for(let i = 0; i < surveysAllButActive.length; i++){
        surveysAllButActive[i].style.display = "none";
    }

    function nextPrevious(currentSurveyIndex){
        document.querySelector('.survey.active').style.display = 'none';
document.querySelector('.survey.active').classList.remove('active');

        surveys[currentSurveyIndex].classList.add('active');
        surveys[currentSurveyIndex].style.display = "flex";

        if(currentSurveyIndex === 0) {
            document.querySelector(".btn-previous").disabled = true;
        }
        else if(currentSurveyIndex === surveys.length - 1){
            document.querySelector(".btn-next").disabled = true;
        }
    }
})
```



```

var first_name = document.getElementById('first_name').value
var last_name = document.getElementById('last_name').value
var username= document.getElementById('username').value
var email = document.getElementById('email').value
var password = document.getElementById('password').value
var birth_date = document.getElementById('birth_date').value

var valid = true;
var messages = '';

if(!checked){
    valid = false;
    messages += "<li>Please select why you're learning a
language.</li>"
}
if(!regexName.test(first_name)){
    valid = false
    messages += "<li>First name is required.</li>"
}
if(!regexName.test(last_name)){
    valid = false
    messages += "<li>Last name is required.</li>"
}
if(!regexUsername.test(username)){
    valid = false
    messages += "<li>Username is required.</li>"
}
if(!regexEmail.test(email)){
    valid = false
    messages += "<li>Email is required.</li>"
}
if(!regexPassword.test(password)){
    valid = false
    messages += "<li>Password is required.</li>"
}
if(!birth_date){
    valid = false
    messages += "<li>Birth date is required.</li>"
}

if(!valid){
    errorsBlock.style.display = "block";
    errorsBlock.innerHTML = '<ul>'
    errorsBlock.innerHTML += messages
    errorsBlock.innerHTML += '</ul>'
}

if(valid){
    $.ajax({
        type: "GET",
        url: "http://127.0.0.1:8000/check-username-email",
        headers: {'Access-Control-Allow-Origin':
"http://127.0.0.1:8000/"},
        data : { email: email, username: username},
        dataType: "json",
        success: function(result){
            if( result === 'NotExists'){
                currentSurveyIndex++;
                nextPrevious(currentSurveyIndex)
            }
            else{

```

```

        valid = false;
        if(result === 'EmailExists'){
            messages += "<li>Email has already been
taken.</li>"
        }
        if(result === 'UsernameExists'){
            messages += "<li>Username has already been
taken.</li>"
        }
        if(result === 'UsernameAndEmailExist'){
            messages += "<li>Email and username have been
already taken.</li>"
        }
        errorsBlock.style.display = "block";
        errorsBlock.innerHTML = '<ul>'
        errorsBlock.innerHTML += messages
        errorsBlock.innerHTML += '</ul>'

        return valid
    }

    },
    error: function (error) {
        console.log(error);
    }
    })
}

}) ()

```

5.2.4 select-the-matching-pairs.js

```

/** Disabled Continue button */
function disableContinueButton(){
    //var answer_container = document.querySelector('#answer-text')
    var translations = document.getElementsByClassName('translation')
    for(var i = 0; i < translations.length; i++){
        translations[i].disabled = true
    }

    //if(answer_container.innerHTML === ``){
        document.querySelector('#check-choice').disabled = true
    //}
}
disableContinueButton()

/** Clear Locale Storage Item(answer) */
function clearLocalStorageItem(item){
    if(window.localStorage.getItem(item)){
        localStorage.removeItem(item);
    }
}
clearLocalStorageItem('answer')

/** Choice click */
$(document).on('click', '.it-word', function(){
    document.querySelector('#check-choice').disabled = false

```

```

var translations = document.getElementsByClassName('translation')
for(var i = 0; i < translations.length; i++){
    translations[i].disabled = false
}
var italian_choices = document.getElementsByClassName('it-word')
for(var i = 0; i < italian_choices.length; i++){
    italian_choices[i].disabled = true
}

if(!window.localStorage.getItem('answer')){
    var answer = []
    answer[0] = {it_word_id : $(this).data('id'), translation_id :
null}
    window.localStorage.setItem('answer', JSON.stringify(answer))
}
else{
    answer = JSON.parse(window.localStorage.getItem('answer'))
    answer[answer.length] = {it_word_id : $(this).data('id'),
translation_id : null}
    window.localStorage.setItem('answer', JSON.stringify(answer))

    for(let i = 0; i < answer.length; i++){
        if(answer[i].translation_id != null){
            let translation = document.querySelector('#translation-' +
answer[i].translation_id)

            translation.disabled = true
        }
    }
}
}))

/** Reset choices */
function resetChoices(){
    document.querySelector('#check-choice').disabled = true

    if(window.localStorage.getItem('answer')){
        localStorage.removeItem('answer');
    }

    var choices = document.getElementsByClassName('choice-input')
    for(var i = 0; i < choices.length; i++){
        choices[i].disabled = false
    }
}
$(document).on('click', '#reset-choice', function(){
    resetChoices()
}))

/** Click on the translation choice */
$(document).on('click', '.translation', function(){
    var translations = document.getElementsByClassName('translation')
    for(var i = 0; i < translations.length; i++){
        translations[i].disabled = true
    }
    var italian_choices = document.getElementsByClassName('it-word')
    for(var i = 0; i < italian_choices.length; i++){
        italian_choices[i].disabled = false
    }
}

```

```

        if(window.localStorage.getItem('answer')){
            answer = JSON.parse(window.localStorage.getItem('answer'))
            if(answer.length <= 4){
                answer[answer.length-1].translation_id = $(this).data('id')
            }

            for(let i = 0; i < answer.length; i++){
                let it_word = document.querySelector('#it-word-' +
answer[i].it_word_id)

                it_word.disabled = true
            }

            window.localStorage.setItem('answer', JSON.stringify(answer))
        }
    })

    /** Check the answer */
    $(document).on('click', '#check-choice', function(){
        var answer = JSON.parse(window.localStorage.getItem('answer'))
        resetChoices()

        $_token = "{{ csrf_token() }}";
        $.ajax({
            type: "GET",
            url: window.location.href + "/check-translation",
            headers: {'Access-Control-Allow-Origin': "http://127.0.0.1:8000/",
                'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')},
            data : { answer: answer},
            dataType: "json",
            cache: false,
            success: function(result){
                if(result.success){
                    document.querySelector('#background').innerHTML =
result.html
                }
                else if(result.success === false){
                    document.querySelector('#error-container').style.display =
'flex'

                    for(var i = 0; i < result.words.length; i++){
                        document.querySelector('#error-text').innerHTML +=
'<p>' +
                            result.words[i].text +
                            ' = ' +
                            result.words[i].translation_text +
                            '</p>'
                    }
                }
                else if(result.finished){
                    document.querySelector('#background').innerHTML =
result.html
                }
                else if(result.failure){
                    document.querySelector('#error-container').style.display =
'flex'

                    document.querySelector('#error-text').innerHTML +=
result.text
                }
            },

```

```

        error: function (error) {
            console.log(error);
        }
    })
})

```

5.2.5 select-the-missing-word.js

```

/** Disabled Continue button */
function disableContinueButton(){
    var answer_container = document.querySelector('#answer-text')
    if(answer_container.innerHTML === ``){
        document.querySelector('#check-choice').disabled = true
    }
}
disableContinueButton()
/** Select the missing word click */

$(document).on('click', '.choice-input', function(){
    var answer_container = document.querySelector('#answer-text')

    var choices = document.getElementsByClassName('choice-input')
    for(var i = 0; i<choices.length; i++){
        choices[i].disabled = true
    }

    document.querySelector('#check-choice').disabled = false
    var value = this.value;
    answer_container.innerHTML += value + ' '
})

/** Reset choices */
$(document).on('click', '#reset-choice', function(){
    var answer_container = document.querySelector('#answer-text')
    var choices = document.getElementsByClassName('choice-input')
    for(var i = 0; i<choices.length; i++){
        choices[i].disabled = false
    }
    document.querySelector('#check-choice').disabled = true
    answer_container.innerHTML = ``
})

$(document).on('click', '#check-choice', function(){
    var user_answer = document.querySelector('#answer-text').innerHTML

    $_token = "{{ csrf_token() }}";
    $.ajax({
        type: "GET",
        url: window.location.href + "/check-translation",
        headers: {'Access-Control-Allow-Origin': "http://127.0.0.1:8000/",
            'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')},
        data : { user_answer: user_answer},
        dataType: "json",
        cache: false,
        success: function(result){
            console.log(result)
            if(result.finished){
                document.querySelector('#background').innerHTML =
result.html
            }
        }
    })

```

```

        else if(result.correct === false){
            document.querySelector('#error-container').style.display =
'flex'
            document.querySelector('#error-text').innerHTML =
result.translation
        }
        else{
            document.querySelector('#background').innerHTML =
result.html
        }
    },
    error: function (error) {
        console.log(error);
    }
})
})
})

```

5.2.6 write-this-in-english.js

```

/** Disabled Continue button */
function disableContinueButton(){
    var answer_container = document.querySelector('#answer-text')
    if(answer_container.innerHTML === ``){
        document.querySelector('#check-choice').disabled = true
    }
}
disableContinueButton()
/** Write this in english choice click */
$(document).on('click', '.choice-input', function(){
    var answer_container = document.querySelector('#answer-text')

    document.querySelector('#check-choice').disabled = false
    var value = this.value;
    answer_container.innerHTML += value + ' '
})

/** Reset choices */
$(document).on('click', '#reset-choice', function(){
    var answer_container = document.querySelector('#answer-text')
    document.querySelector('#check-choice').disabled = true
    answer_container.innerHTML = ``
})

$(document).on('click', '#check-choice', function(){
    var user_answer = document.querySelector('#answer-text').innerHTML
    //var stylized_user_answer = user_answer.toLowerCase().trim()

    var sentence_id_container = document.getElementById('sentence-id');
    var sentence_id = sentence_id_container.getAttribute('data-id');
    $_token = "{ { csrf_token() } }";
    $.ajax({
        type: "GET",
        url: window.location.href + "/check-translation",
        headers: { 'Access-Control-Allow-Origin': "http://127.0.0.1:8000/",
            'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content') },
        data : { user_answer: user_answer, sentence_id: sentence_id},
        dataType: "json",
        cache: false,
        success: function(result){

```



```

        if(result.finished){
            document.querySelector('#background').innerHTML =
result.html
            //document.querySelector('#write-this-in-
english').innerHTML = 'All done!'
        }
        else if(result.correct == false){
            document.querySelector('#error-container').style.display =
'flex'
            document.querySelector('#error-text').innerHTML =
result.translation
        }
        else{
            document.querySelector('#background').innerHTML =
result.html
        }
    },
    error: function (error) {
        console.log(error);
    }
})
})

```

5.3 CSS

5.3.1 style.css

```

/**
 * Template Name: Techie - v4.7.0
 * Template URL: https://bootstrapmade.com/techie-free-skin-bootstrap-3/
 * Author: BootstrapMade.com
 * License: https://bootstrapmade.com/license/
 */

/*-----*/
# General
/*-----*/

body {
    font-family: "Poppins", sans-serif;
    color: #444444;
}

a {
    color: #f7aa02;
    text-decoration: none;
}

a:hover {
    color: darkorange;
    text-decoration: none;
}

h1, h2, h3, h4, h5, h6 {
    font-family: "Poppins", sans-serif;
}

ul{
    list-style-type: none;
    margin: 0px;
}

.cursor-pointer{

```

```

        cursor: pointer;
    }
    form div{
        padding: 5px;
    }
    /* overriding bootstrap's select css */
    .form-select:focus {
        border-color: transparent;
        outline: 0;
        box-shadow: none;
    }
    .pagination .page-item.active .page-link{
        z-index: 3;
        color: #fff;
        background-color: orange;
        border-color: orange;
    }
    .pagination .page-link{
        color: orange;
        background-color: rgba(255,165,0,0.1);
        border: rgba(255,165,0,0.9);
    }
    .page-link:focus {
        z-index: 3;
        color: darkorange;
        background-color: navajowhite;
        outline: 0;
        box-shadow: none;
    }
    .page-link:hover {
        z-index: 2;
        color: white;
        background-color: navajowhite;
        border-color: navajowhite;
    }
    .nav-link:focus, .nav-link:hover {
        color: white;
    }
}

audio{
    width: 150px;
}
/*-----
# Back to top button
-----*/
.back-to-top {
    position: fixed;
    visibility: hidden;
    opacity: 0;
    right: 15px;
    bottom: 15px;
    z-index: 996;
    background: #f7aa02;
    width: 40px;
    height: 40px;
    border-radius: 4px;
    transition: all 0.4s;
}
.back-to-top i {
    font-size: 28px;

```

```

    color: #fff;
    line-height: 0;
}
.back-to-top:hover {
    background: #7b27d8;
    color: #fff;
}
.back-to-top.active {
    visibility: visible;
    opacity: 1;
}

/*-----
# Preloader
-----*/
#preloader {
    position: fixed;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    z-index: 9999;
    overflow: hidden;
    background: #fff;
}

#preloader:before {
    content: "";
    position: fixed;
    top: calc(50% - 30px);
    left: calc(50% - 30px);
    border: 6px solid #f7aa02;
    border-top-color: #e7e4fe;
    border-radius: 50%;
    width: 60px;
    height: 60px;
    -webkit-animation: animate-preloader 1s linear infinite;
    animation: animate-preloader 1s linear infinite;
}

@-webkit-keyframes animate-preloader {
    0% {
        transform: rotate(0deg);
    }
    100% {
        transform: rotate(360deg);
    }
}

@keyframes animate-preloader {
    0% {
        transform: rotate(0deg);
    }
    100% {
        transform: rotate(360deg);
    }
}

/*-----
# Disable aos animation delay on mobile devices
-----*/
@media screen and (max-width: 768px) {

```

```

    [data-aos-delay] {
      transition-delay: 0 !important;
    }
  }
}
/*-----
# Header
-----*/
#header {
  transition: all 0.5s;
  z-index: 999;
  padding: 20px 0;
  background-color: #d79eff;
}
#header.header-scrolled, #header.header-inner-pages {
  background-color: orange;
  padding: 12px 0;
}
#header .logo {
  font-size: 32px;
  margin: 0;
  padding: 0;
  line-height: 1;
  font-weight: 400;
  letter-spacing: 2px;
  text-transform: uppercase;
}
#header .logo a {
  color: white;
}
#header .logo img {
  max-height: 40px;
}

/*-----
# Navigation Menu
-----*/
/**
 * Desktop Navigation
 */
.navbar {
  padding: 0;
}
.navbar ul {
  margin: 0;
  padding: 0;
  display: flex;
  list-style: none;
  align-items: center;
}
.navbar li {
  position: relative;
}
.navbar > ul > li {
  padding: 10px 0 10px 24px;
}
.navbar a {
  display: flex;
  align-items: center;
  justify-content: space-between;
  padding: 0 3px;
  font-family: "Poppins", sans-serif;

```

```

    font-size: 15px;
    font-weight: 600;
    color: rgba(255, 255, 255, 0.7);
    white-space: nowrap;
    transition: 0.3s;
    position: relative;
}
.navbar a i, .navbar a:focus i {
    font-size: 19px;
    line-height: 0;
    margin-left: 5px;
}
.navbar > ul > li > .hover-underline:before {
    content: "";
    position: absolute;
    width: 100%;
    height: 2px;
    bottom: -5px;
    left: 0;
    background-color: #fff;
    visibility: hidden;
    width: 0px;
    transition: all 0.3s ease-in-out 0s;
}
.navbar a:hover:before, .navbar li:hover > a:before, .navbar .active:before
{
    visibility: visible;
    width: 100%;
}
.navbar a:hover{
    color: white;
}
/* .navbar a:hover, .navbar .active, .navbar .active:focus, .navbar li:hover
> a {
    color: #fff;
}*/
.navbar .getstarted, .navbar .getstarted:focus {
    padding: 8px 25px;
    margin-left: 30px;
    border-radius: 5px;
    color: #fff;
    transition: 0.3s;
    font-size: 14px;
    font-weight: 600;
    border: 2px solid rgba(255, 255, 255, 0.5);
}
.navbar .getstarted:hover, .navbar .getstarted:focus:hover {
    color: #fff;
    border-color: #fff;
}
.navbar .getstarted:before, .navbar li:hover > .getstarted:before {
    visibility: hidden;
}
.navbar .dropdown ul {
    display: block;
    position: absolute;
    left: 24px;
    top: calc(100% + 30px);
    margin: 0;
    padding: 10px 0;
    z-index: 99;
}

```

```

    opacity: 0;
    visibility: hidden;
    background: #fff;
    box-shadow: 0px 0px 30px rgba(127, 137, 161, 0.25);
    transition: 0.3s;
    border-radius: 20px;
}
.navbar .dropdown ul li {
    min-width: 200px;
}
.navbar .dropdown ul a {
    padding: 10px 20px;
    font-size: 14px;
    color: #2c4964;
}
.navbar .dropdown ul a i {
    font-size: 12px;
}
.navbar .dropdown ul a:hover, .navbar .dropdown ul .active:hover, .navbar
.dropdown ul li:hover > a {
    color: #f7aa02;
}
.navbar .dropdown:hover > ul {
    opacity: 1;
    top: 100%;
    visibility: visible;
}
.navbar .dropdown .dropdown ul {
    top: 0;
    left: calc(100% - 30px);
    visibility: hidden;
}
.navbar .dropdown .dropdown:hover > ul {
    opacity: 1;
    top: 0;
    left: 100%;
    visibility: visible;
}
@media (max-width: 1366px) {
    .navbar .dropdown .dropdown ul {
        left: -90%;
    }
    .navbar .dropdown .dropdown:hover > ul {
        left: -100%;
    }
}

/**
 * Mobile Navigation
 */
.mobile-nav-toggle {
    color: #fff;
    font-size: 28px;
    cursor: pointer;
    display: none;
    line-height: 0;
    transition: 0.5s;
}

@media (max-width: 770px) {
    .mobile-nav-toggle {

```

```

        display: block;
    }

    .navbar ul {
        display: none;
    }
}

.navbar-mobile {
    position: fixed;
    overflow: hidden;
    top: 0;
    right: 0;
    left: 0;
    bottom: 0;
    background: rgba(28, 47, 65, 0.9);
    transition: 0.3s;
    z-index: 999;
}

.navbar-mobile .mobile-nav-toggle {
    position: absolute;
    top: 15px;
    right: 15px;
}

.navbar-mobile ul {
    display: block;
    position: absolute;
    top: 55px;
    right: 15px;
    left: 15px;
    padding: 10px 0;
    background-color: #fff;
    overflow-y: auto;
    transition: 0.3s;
    border-radius: 30px;
}

.navbar-mobile > ul > li {
    padding: 0;
}

.navbar-mobile a, .navbar-mobile a:focus {
    padding: 10px 20px;
    font-size: 15px;
    color: #2c4964;
}

.navbar-mobile a:hover:before, .navbar-mobile li:hover > a:before, .navbar-mobile .active:before {
    visibility: hidden;
}

.navbar-mobile a:hover, .navbar-mobile .active, .navbar-mobile li:hover > a {
    color: #f7aa02;
}

.navbar-mobile .getstarted, .navbar-mobile .getstarted:focus {
    margin: 15px;
}

.navbar-mobile .dropdown ul {
    position: static;
    display: none;
    margin: 10px 20px;
    padding: 10px 0;
    z-index: 99;
    opacity: 1;
}

```

```

visibility: visible;
background: #fff;
box-shadow: 0px 0px 30px rgba(127, 137, 161, 0.25);
}
.navbar-mobile .dropdown ul li {
  min-width: 200px;
}
.navbar-mobile .dropdown ul a {
  padding: 10px 20px;
}
.navbar-mobile .dropdown ul a i {
  font-size: 12px;
}
.navbar-mobile .dropdown ul a:hover, .navbar-mobile .dropdown ul
.active:hover, .navbar-mobile .dropdown ul li:hover > a {
  color: #f7aa02;
}
.navbar-mobile .dropdown > .dropdown-active {
  display: block;
}

/*-----
# Hero Section
-----*/
.bg {
  width: 100%;
  /*background: linear-gradient( rgba(247, 170, 2, 0.9) 0%, rgba(247,
223, 2, 1) 100%), url("../img/hero-bg.jpg") center center no-repeat;*/
  background-color: rgb(10,10,10);
  background-size: cover;
  color: white;
}
#white-background{
  background-color: white;
}
.blue-background{
  background-color: rgb(82, 20, 255);
}
.yellow-background{
  background-color: rgb(255, 206, 0);
}
.orange-background{
  background-color: orange;
}
.lilac-background{
  background-color: #d79eff;
}
.white-background{
  background-color: white;
}
.task-background{
  height: 100vh;
  padding: 0px;
}
/*****/
.welcome-container{
  position: relative;
}
.welcome-container p{
  font-size: 90px;
  font-weight: bold;
}

```



```

}
@media (max-width: 691px) {
  .welcome-container p{
    font-size: 70px;
  }
}
@media (max-width: 550px) {
  .welcome-container{
    padding: 0 !important;
  }
  .welcome-container p{
    font-size: 50px;
  }
}
.welcome-container p:nth-child(2){
  color: orange;
}
#blue-figure-container{
  bottom: 0px;
  left: 60px;
}
#blue-figure-container img{
  width: 300px;
}
#shape-container{
  bottom: 0px;
  right: 60px;
}
#shape-container img{
  height: 410px;
}
.register-login-container{
  position: relative;
}
.register-login-container svg{
  position: absolute;
}
.register-login-container .space .detail-diamond{
  bottom: 0px;
  right: 0px;
}
.register-login-container .space .detail-circle-bottom{
  top: 80px;
  left: 20px;
}
circle{
  fill: yellow;
}
.rg-container .detail-circle-bottom{
  right: -40px;
  bottom: -40px;
}
#hero .container, #hero .container-fluid {
  padding-top: 84px;
}
#hero h1 {
  margin: 0;
  font-size: 52px;
  font-weight: 700;
  line-height: 64px;
  color: #fff;
}

```

```

}
#hero h2 {
  /*color: rgba(255, 255, 255, 0.8);*/
  margin: 10px 0 0 0;
}
#hero .btn-get-started {
  font-family: "Poppins", sans-serif;
  font-weight: 500;
  font-size: 16px;
  letter-spacing: 1px;
  display: inline-block;
  padding: 10px 28px;
  border-radius: 5px;
  transition: 0.5s;
  margin-top: 30px;
  color: #fff;
  border: 2px solid #fff;
}
#hero .btn-get-started:hover {
  background: #fff;
  color: rgb(247,170,2);
}
#hero .animated {
  animation: up-down 2s ease-in-out infinite alternate-reverse both;
}
@media (max-width: 991px) {
  #hero {
    text-align: center;
  }
  #hero .container, #hero .container-fluid {
    padding-top: 68px;
  }
  #hero .animated {
    -webkit-animation: none;
    animation: none;
  }
  #hero .hero-img {
    text-align: center;
  }
}
@media (max-width: 768px) {
  #hero h1 {
    font-size: 26px;
    line-height: 36px;
  }
  #hero h2 {
    font-size: 30px;
    line-height: 24px;
  }
  #hero .hero-img img {
    width: 60%;
  }
}
@media (max-width: 575px) {
  #hero .hero-img img {
    width: 80%;
  }
}
@media (max-width: 575px), (max-height: 600px) {
  #hero {
    height: auto;
  }
}

```

```

    }
}

@-webkit-keyframes up-down {
    0% {
        transform: translateY(10px);
    }
    100% {
        transform: translateY(-10px);
    }
}

@keyframes up-down {
    0% {
        transform: translateY(10px);
    }
    100% {
        transform: translateY(-10px);
    }
}

/*-----
# Sections General
-----*/

section {
    padding-top: 80px;
    overflow: hidden;
}

.section-title {
    text-align: center;
    padding-bottom: 30px;
}

.section-title h2 {
    font-size: 32px;
    font-weight: bold;
    text-transform: uppercase;
    margin-bottom: 20px;
    padding-bottom: 20px;
    position: relative;
}

.section-title h2::after {
    content: "";
    position: absolute;
    display: block;
    width: 50px;
    height: 3px;
    background: #f7aa02;
    bottom: 0;
    left: calc(50% - 25px);
}

.section-title p {
    margin-bottom: 0;
}

/*-----
# About
-----*/

.about img{
    border-radius: 45px;
    padding: 20px;
}

.about .detail-circle{

```

```

    border-radius: 45px;
    color: rgb(40,40,40);
}
.about .detail{
    padding: 40px 0;
    border-bottom: 2px whitesmoke solid;
}
.about .detail h3{
    text-align: left;
}
.about .detail p{
    text-align: justify;
}
.about .detail div{
    width: 50%;
}
.about .detail img{
    width: 450px;
}

.about h3 {
    font-weight: 600;
    font-size: 32px;
    color: rgb(40,40,40);
}
.about .content ul {
    list-style: none;
    padding: 0;
}
.about .content ul li {
    padding-bottom: 10px;
}
.about .content ul i {
    font-size: 20px;
    padding-right: 4px;
    color: #f7aa02;
}
.about .content p:last-child {
    margin-bottom: 0;
}
.about .content .read-more {
    font-family: "Poppins", sans-serif;
    font-weight: 500;
    font-size: 16px;
    letter-spacing: 1px;
    display: inline-block;
    padding: 10px 50px 10px 28px;
    border-radius: 5px;
    transition: 0.5s;
    color: #fff;
    background: linear-gradient(45deg, #f7aa02 0%, #7b27d8 100%);
    position: relative;
}
.about .content .read-more:hover {
    background: linear-gradient(180deg, #f7aa02 0%, #7b27d8 100%);
}
.about .content .read-more i {
    font-size: 22px;
    position: absolute;
    right: 20px;
    top: 12px;
}

```

```

}

/*-----
# Counts
-----*/

.counts {
  background: linear-gradient(90deg, rgba(88, 70, 249, 0.5) 0%, rgba(123,
39, 216, 0.5) 100%), url("../img/counts-bg.png") center center no-repeat;
  padding: 80px 0 60px 0;
}

.counts .counters span {
  font-size: 48px;
  display: block;
  color: #fff;
  font-weight: 600;
  font-family: "Poppins", sans-serif;
}

.counts .counters p {
  padding: 0;
  margin: 0 0 20px 0;
  font-size: 15px;
  color: rgba(255, 255, 255, 0.8);
}

/*-----
# Services
-----*/

.services{
  color: white;
  display: flex;
  flex-direction: column;
  justify-content: flex-end;
  align-items: flex-end;
  width: 100%;
  padding-bottom: 80px;
  position: relative;
}

.services img{
  width: 350px;
  position: absolute;
  left: 170px;
  bottom: 100px;
  z-index: 998;
  border: white 5px solid;
  transform: rotate(15deg);
}

.services h2{
  font-size: 70px;
  text-align: left;
  color: white;
}

.why-header{
  padding: 40px 0;
}

.why-header svg{
  position: absolute;
  right: 10px;
  bottom: 30px;
}

.why-icon-container{
  width: 100%;

```

```

border-bottom: 1px solid white;
padding: 15px;
}
.why-title-container{
display: flex;
justify-content: space-between;
align-items: center;
}
.services .icon-box {
width: 100%;
padding: 5px;
text-align: center;
transition: all ease-in-out 0.3s;
}
.services .icon-box .fa-solid{
font-size: 40px;
}
.services .icon-box i {
font-size: 36px;
transition: 0.5s;
position: relative;
color: whitesmoke;
}
.services .icon-box h4 {
font-weight: 600;
margin: 10px 0 15px 0;
font-size: 22px;
color: rgb(50,50,50);
}
.services .icon-box h4 a {
color: #2c4964;
transition: ease-in-out 0.3s;
}
.services .icon-box p {
line-height: 24px;
font-size: 14px;
margin-bottom: 0;
}
.hover-shadow:hover {
border-color: #fff;
box-shadow: 0px 0 25px 0 rgba(0, 0, 0, 0.1);
}
.services .iconbox-blue i {
color: #47aeff;
}
.services .iconbox-blue:hover .icon i {
color: #fff;
}
.services .iconbox-blue:hover .icon path {
fill: #47aeff;
}
.services .iconbox-orange i {
color: #ffa76e;
}
.services .iconbox-orange:hover .icon i {
color: #fff;
}
.services .iconbox-orange:hover .icon path {
fill: #ffa76e;
}
.services .iconbox-pink i {

```

```

        color: #e80368;
    }
    .services .iconbox-pink:hover .icon i {
        color: #fff;
    }
    .services .iconbox-pink:hover .icon path {
        fill: #e80368;
    }
    .services .iconbox-yellow i {
        color: #ffbb2c;
    }
    .services .iconbox-yellow:hover .icon i {
        color: #fff;
    }
    .services .iconbox-yellow:hover .icon path {
        fill: #ffbb2c;
    }
    .services .iconbox-red i {
        color: #ff5828;
    }
    .services .iconbox-red:hover .icon i {
        color: #fff;
    }
    .services .iconbox-red:hover .icon path {
        fill: #ff5828;
    }
    .services .iconbox-teal i {
        color: #11dbcf;
    }
    .services .iconbox-teal:hover .icon i {
        color: #fff;
    }
    .services .iconbox-teal:hover .icon path {
        fill: #11dbcf;
    }
}

@media only screen and (max-width: 995px){
    .register-container .services .pointer{
        width: 20%;
    }
    section{
        padding-top: 100px;
    }
}

@media only screen and (max-width: 770px){
    .register-container .services .icon-box{
        padding: 5px;
    }
    .register-container .services .pointer{
        width: 100%;
        display: flex;
        justify-content: center;
        align-items: center;
    }
    .detail{
        flex-direction: column;
        align-items: center;
    }
    .detail div{
        width: 80% !important;
    }
}

```

```

    }
}

/*-----
# Features
-----*/

.features .icon-box h4 {
    font-size: 20px;
    font-weight: 700;
    margin: 5px 0 10px 60px;
}
.features .icon-box i {
    font-size: 48px;
    float: left;
    color: #f7aa02;
}
.features .icon-box p {
    font-size: 15px;
    color: #848484;
    margin-left: 60px;
}
.features .image {
    background-position: center center;
    background-repeat: no-repeat;
    background-size: cover;
    min-height: 400px;
}

/*-----
# Testimonials
-----*/

.testimonials .testimonials-carousel, .testimonials .testimonials-slider {
    overflow: hidden;
}
.testimonials .testimonial-item {
    box-sizing: content-box;
    min-height: 320px;
}
.testimonials .testimonial-item .testimonial-img {
    width: 90px;
    border-radius: 50%;
    margin: -40px 0 0 40px;
    position: relative;
    z-index: 2;
    border: 6px solid #fff;
    box-shadow: 0px 2px 15px rgba(0, 0, 0, 0.1);
}
.testimonials .testimonial-item h3 {
    font-size: 18px;
    font-weight: bold;
    margin: 10px 0 5px 45px;
    color: #2c4964;
}
.testimonials .testimonial-item h4 {
    font-size: 14px;
    color: #999;
    margin: 0 0 0 45px;
}
.testimonials .testimonial-item .quote-icon-left, .testimonials
.testimonial-item .quote-icon-right {
    color: #b1a9fc;

```



```

    font-size: 26px;
}
.testimonials .testimonial-item .quote-icon-left {
    display: inline-block;
    left: -5px;
    position: relative;
}
.testimonials .testimonial-item .quote-icon-right {
    display: inline-block;
    right: -5px;
    position: relative;
    top: 10px;
}
.testimonials .testimonial-item p {
    font-style: italic;
    margin: 0 15px 0 15px;
    padding: 20px 20px 60px 20px;
    background: #fff;
    position: relative;
    border-radius: 6px;
    position: relative;
    z-index: 1;
    box-shadow: 0 0px 20px 0 rgba(0, 0, 0, 0.1);
}
.testimonials .swiper-pagination {
    margin-top: 20px;
    position: relative;
}
.testimonials .swiper-pagination .swiper-pagination-bullet {
    width: 12px;
    height: 12px;
    background-color: #fff;
    opacity: 1;
    border: 1px solid #f7aa02;
}
.testimonials .swiper-pagination .swiper-pagination-bullet-active {
    background-color: #f7aa02;
}
@media (max-width: 767px) {
    .testimonials {
        margin: 30px 10px;
    }
}

/*-----
# Portfolio
-----*/
.portfolio .portfolio-item {
    margin-bottom: 30px;
}
.portfolio #portfolio-filters {
    padding: 0;
    margin: 0 auto 25px auto;
    list-style: none;
    text-align: center;
    border-radius: 50px;
    padding: 2px 15px;
}
.portfolio #portfolio-filters li {
    cursor: pointer;
    display: inline-block;

```

```

padding: 10px 15px;
font-size: 14px;
font-weight: 600;
line-height: 1;
text-transform: uppercase;
color: #444444;
margin-bottom: 5px;
transition: all 0.3s ease-in-out;
border-radius: 5px;
font-family: "Poppins", sans-serif;
}
.portfolio #portfolio-filters li:hover, .portfolio #portfolio-filters
li.filter-active {
  color: #fff;
  background: #f7aa02;
}
.portfolio #portfolio-filters li:last-child {
  margin-right: 0;
}
.portfolio .portfolio-wrap {
  transition: 0.3s;
  position: relative;
  overflow: hidden;
  z-index: 1;
}
.portfolio .portfolio-wrap::before {
  content: "";
  background: rgba(88, 70, 249, 0.8);
  position: absolute;
  left: 30px;
  right: 30px;
  top: 30px;
  bottom: 30px;
  transition: all ease-in-out 0.3s;
  z-index: 2;
  opacity: 0;
}
.portfolio .portfolio-wrap .portfolio-info {
  opacity: 0;
  position: absolute;
  top: 10%;
  left: 0;
  right: 0;
  text-align: center;
  z-index: 3;
  transition: all ease-in-out 0.3s;
}
.portfolio .portfolio-wrap .portfolio-info h4 {
  font-size: 20px;
  color: #fff;
  font-weight: 600;
}
.portfolio .portfolio-wrap .portfolio-info p {
  color: #ffffff;
  font-size: 14px;
  text-transform: uppercase;
}
.portfolio .portfolio-wrap .portfolio-links {
  opacity: 0;
  left: 0;
  right: 0;

```

```

    bottom: 10%;
    text-align: center;
    z-index: 3;
    position: absolute;
    transition: all ease-in-out 0.3s;
}
.portfolio .portfolio-wrap .portfolio-links a {
    color: rgba(255, 255, 255, 0.6);
    margin: 0 2px;
    font-size: 28px;
    display: inline-block;
    transition: 0.3s;
}
.portfolio .portfolio-wrap .portfolio-links a:hover {
    color: #fff;
}
.portfolio .portfolio-wrap:hover::before {
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    opacity: 1;
}
.portfolio .portfolio-wrap:hover .portfolio-info {
    opacity: 1;
    top: calc(50% - 48px);
}
.portfolio .portfolio-wrap:hover .portfolio-links {
    opacity: 1;
    bottom: calc(50% - 50px);
}

/*-----
# Portfolio Details
-----*/
.portfolio-details {
    padding-top: 40px;
}
.portfolio-details .portfolio-details-slider img {
    width: 100%;
}
.portfolio-details .portfolio-details-slider .swiper-pagination {
    margin-top: 20px;
    position: relative;
}
.portfolio-details .portfolio-details-slider .swiper-pagination .swiper-
pagination-bullet {
    width: 12px;
    height: 12px;
    background-color: #fff;
    opacity: 1;
    border: 1px solid #f7aa02;
}
.portfolio-details .portfolio-details-slider .swiper-pagination .swiper-
pagination-bullet-active {
    background-color: #f7aa02;
}
.portfolio-details .portfolio-info {
    padding: 30px;
    box-shadow: 0px 0 30px rgba(44, 73, 100, 0.08);
}

```

```

.portfolio-details .portfolio-info h3 {
  font-size: 22px;
  font-weight: 700;
  margin-bottom: 20px;
  padding-bottom: 20px;
  border-bottom: 1px solid #eee;
}
.portfolio-details .portfolio-info ul {
  list-style: none;
  padding: 0;
  font-size: 15px;
}
.portfolio-details .portfolio-info ul li + li {
  margin-top: 10px;
}
.portfolio-details .portfolio-description {
  padding-top: 30px;
}
.portfolio-details .portfolio-description h2 {
  font-size: 26px;
  font-weight: 700;
  margin-bottom: 20px;
}
.portfolio-details .portfolio-description p {
  padding: 0;
}

/*-----
# Pricing
-----*/

.pricing .box {
  padding: 20px;
  background: #fff;
  text-align: center;
  border-radius: 5px;
  position: relative;
  overflow: hidden;
  box-shadow: 0px 2px 15px rgba(0, 0, 0, 0.1);
}
.pricing h3 {
  font-weight: 400;
  margin: -20px -20px 20px -20px;
  padding: 20px 15px;
  font-size: 16px;
  font-weight: 600;
  color: #2c4964;
  background: #f8f8f8;
}
.pricing h4 {
  font-size: 36px;
  color: #f7aa02;
  font-weight: 600;
  font-family: "Roboto", sans-serif;
  margin-bottom: 20px;
}
.pricing h4 sup {
  font-size: 20px;
  top: -15px;
  left: -3px;
}
.pricing h4 span {

```

```

    color: #bababa;
    font-size: 16px;
    font-weight: 300;
}
.pricing ul {
    padding: 0;
    list-style: none;
    color: #444444;
    text-align: center;
    line-height: 20px;
    font-size: 14px;
}
.pricing ul li {
    padding-bottom: 16px;
}
.pricing ul i {
    color: #f7aa02;
    font-size: 18px;
    padding-right: 4px;
}
.pricing ul .na {
    color: #ccc;
    text-decoration: line-through;
}
.pricing .btn-wrap {
    margin: 20px -20px -20px -20px;
    padding: 20px 15px;
    background: #f8f8f8;
    text-align: center;
}
.pricing .btn-buy {
    background: linear-gradient(42deg, #f7aa02 0%, #7b27d8 100%);
    display: inline-block;
    padding: 10px 35px;
    border-radius: 4px;
    color: #fff;
    transition: none;
    font-size: 15px;
    font-weight: 400;
    font-family: "Roboto", sans-serif;
    font-weight: 600;
    transition: 0.3s;
}
.pricing .btn-buy:hover {
    background: linear-gradient(180deg, #f7aa02 0%, #7b27d8 100%);
}
.pricing .featured h3 {
    color: #fff;
    background: #f7aa02;
}
.pricing .advanced {
    width: 200px;
    position: absolute;
    top: 18px;
    right: -68px;
    transform: rotate(45deg);
    z-index: 1;
    font-size: 14px;
    padding: 1px 0 3px 0;
    background: #f7aa02;
    color: #fff;
}

```

```

}

/*-----
# Frequently Asked Questions
-----*/

.faq {
  background: linear-gradient(42deg, #f7aa02 0%, #7b27d8 100%);
}
.faq .section-title h2, .faq .section-title p {
  color: #fff;
}
.faq .section-title h2::after {
  background: rgba(255, 255, 255, 0.6);
}
.faq .faq-list {
  padding: 0 100px;
}
.faq .faq-list ul {
  padding: 0;
  list-style: none;
}
.faq .faq-list li + li {
  margin-top: 15px;
}
.faq .faq-list li {
  padding: 30px;
  background: #fff;
  border-radius: 5px;
  position: relative;
}
.faq .faq-list a {
  display: block;
  position: relative;
  font-family: "Roboto", sans-serif;
  font-size: 16px;
  line-height: 24px;
  font-weight: 500;
  padding: 0 30px;
  outline: none;
  cursor: pointer;
}
.faq .faq-list .icon-help {
  font-size: 24px;
  position: absolute;
  right: 0;
  left: 20px;
  color: #8577fb;
}
.faq .faq-list .icon-show, .faq .faq-list .icon-close {
  font-size: 24px;
  position: absolute;
  right: 0;
  top: 0;
}
.faq .faq-list p {
  margin-bottom: 0;
  padding: 10px 0 0 0;
}
.faq .faq-list .icon-show {
  display: none;
}

```

```

.faq .faq-list a.collapsed {
  color: #2c4964;
}
.faq .faq-list a.collapsed:hover {
  color: #f7aa02;
}
.faq .faq-list a.collapsed .icon-show {
  display: inline-block;
}
.faq .faq-list a.collapsed .icon-close {
  display: none;
}
@media (max-width: 1200px) {
  .faq .faq-list {
    padding: 0;
  }
}

/*-----
# Contact
-----*/

.contact .info-box {
  color: #444444;
  text-align: center;
  box-shadow: 0 0 30px rgba(214, 215, 216, 0.6);
  padding: 20px 0 30px 0;
  background: #fff;
}
.contact .info-box i {
  font-size: 32px;
  color: #f7aa02;
  border-radius: 50%;
  padding: 8px;
}
.contact .info-box h3 {
  font-size: 20px;
  color: #2c4964;
  font-weight: 700;
  margin: 10px 0;
}
.contact .info-box p {
  padding: 0;
  line-height: 24px;
  font-size: 14px;
  margin-bottom: 0;
}
.contact .php-email-form {
  box-shadow: 0 0 30px rgba(214, 215, 216, 0.6);
  padding: 30px;
  background: #fff;
}
.contact .php-email-form .error-message {
  display: none;
  color: #fff;
  background: #ed3c0d;
  text-align: left;
  padding: 15px;
  font-weight: 600;
}
.contact .php-email-form .error-message br + br {
  margin-top: 25px;
}

```

```

}
.contact .php-email-form .sent-message {
  display: none;
  color: #fff;
  background: #18d26e;
  text-align: center;
  padding: 15px;
  font-weight: 600;
}
.contact .php-email-form .loading {
  display: none;
  background: #fff;
  text-align: center;
  padding: 15px;
}
.contact .php-email-form .loading:before {
  content: "";
  display: inline-block;
  border-radius: 50%;
  width: 24px;
  height: 24px;
  margin: 0 10px -6px 0;
  border: 3px solid #18d26e;
  border-top-color: #eee;
  -webkit-animation: animate-loading 1s linear infinite;
  animation: animate-loading 1s linear infinite;
}
.contact .php-email-form input, .contact .php-email-form textarea {
  border-radius: 5px;
  box-shadow: none;
  font-size: 14px;
}
.contact .php-email-form input:focus, .contact .php-email-form
textarea:focus {
  border-color: #f7aa02;
}
.contact .php-email-form input {
  padding: 10px 15px;
}
.contact .php-email-form textarea {
  padding: 12px 15px;
}
.contact .php-email-form button[type=submit] {
  background: #f7aa02;
  border: 0;
  padding: 10px 24px;
  color: #fff;
  transition: 0.4s;
  border-radius: 5px;
}
.contact .php-email-form button[type=submit]:hover {
  background: #7b27d8;
}
@-webkit-keyframes animate-loading {
  0% {
    transform: rotate(0deg);
  }
  100% {
    transform: rotate(360deg);
  }
}

```



```

@keyframes animate-loading {
  0% {
    transform: rotate(0deg);
  }
  100% {
    transform: rotate(360deg);
  }
}

/*-----
# Breadcrumbs
-----*/

.breadcrumbs {
  padding: 15px 0;
  background: #f2f6f9;
  min-height: 40px;
  margin-top: 85px;
}
@media (max-width: 992px) {
  .breadcrumbs {
    margin-top: 56px;
  }
}

.breadcrumbs h2 {
  font-size: 28px;
  font-weight: 300;
}

.breadcrumbs ol {
  display: flex;
  flex-wrap: wrap;
  list-style: none;
  padding: 0;
  font-size: 14px;
  margin: 0;
}

.breadcrumbs ol li + li {
  padding-left: 10px;
}

.breadcrumbs ol li + li::before {
  display: inline-block;
  padding-right: 10px;
  color: #3c6387;
  content: "/";
}

@media (max-width: 768px) {
  .breadcrumbs .d-flex {
    display: block !important;
  }
  .breadcrumbs ol {
    display: block;
  }
  .breadcrumbs ol li {
    display: inline-block;
  }
  .challenge-container{
    padding: 0px 30px !important;
  }
  .challenge-container .alert{
    font-size: 12px !important;
  }
  .question-container p{

```

```

        font-size: 20px;
    }
    .task h3{
        font-size: initial !important;
    }
}

/*-----
# Footer
-----*/
#footer {
    color: #fff;
    font-size: 14px;
    background: linear-gradient(45deg, rgba(0, 0, 0, 0.9) 0%, rgba(0, 0, 0, 0.9) 100%), url("../img/hero-bg.jpg") center center no-repeat;
    background-size: cover;
}
#footer .footer-top {
    padding-top: 60px;
    position: relative;
}
#footer .footer-top .footer-contact {
    margin-bottom: 30px;
}
#footer .footer-top .footer-contact h3 {
    font-size: 28px;
    margin: 0 0 30px 0;
    padding: 2px 0 2px 0;
    line-height: 1;
    font-weight: 500;
    text-transform: uppercase;
}
#footer .footer-top .footer-contact p {
    font-size: 14px;
    line-height: 24px;
    margin-bottom: 0;
    font-family: "Poppins", sans-serif;
}
#footer .footer-top h4 {
    font-size: 16px;
    font-weight: bold;
    position: relative;
    padding-bottom: 12px;
}
#footer .footer-top .footer-links {
    margin-bottom: 30px;
}
#footer .footer-top .footer-links ul {
    list-style: none;
    padding: 0;
    margin: 0;
}
#footer .footer-top .footer-links ul i {
    padding-right: 2px;
    color: #fff;
    font-size: 18px;
    line-height: 1;
}
#footer .footer-top .footer-links ul li {
    padding: 10px 0;
    display: flex;

```

```

    align-items: center;
}
#footer .footer-top .footer-links ul li:first-child {
    padding-top: 0;
}
#footer .footer-top .footer-links ul a {
    color: rgba(255, 255, 255, 0.8);
    transition: 0.3s;
    display: inline-block;
    line-height: 1;
}
#footer .footer-top .footer-links ul a:hover {
    text-decoration: underline;
    color: #fff;
}
#footer .footer-newsletter {
    font-size: 15px;
}
#footer .footer-newsletter h4 {
    font-size: 16px;
    font-weight: bold;
    position: relative;
    padding-bottom: 12px;
}
#footer .footer-newsletter form {
    margin-top: 30px;
    background: #fff;
    padding: 6px 10px;
    position: relative;
    border-radius: 5px;
    text-align: left;
    border: 1px solid white;
}
#footer .footer-newsletter form input[type=email] {
    border: 0;
    padding: 4px 8px;
    width: calc(100% - 100px);
}
#footer .footer-newsletter form input[type=submit] {
    position: absolute;
    top: -1px;
    right: -2px;
    bottom: -1px;
    border: 0;
    background: none;
    font-size: 16px;
    padding: 0 20px;
    background: rgba(123, 39, 216, 0.8);
    color: #fff;
    transition: 0.3s;
    border-radius: 0 5px 5px 0;
    box-shadow: 0px 2px 15px rgba(0, 0, 0, 0.1);
}
#footer .footer-newsletter form input[type=submit]:hover {
    background: #f7aa02;
}
#footer .copyright-wrap {
    border-top: 1px solid orange;
}
#footer .credits {
    padding-top: 5px;
}

```

```

    font-size: 13px;
    color: #fff;
}
#footer .credits a {
    color: #fff;
}
#footer .social-links a {
    font-size: 18px;
    display: inline-block;
    background: rgba(255, 255, 255, 0.1);
    color: #fff;
    line-height: 1;
    padding: 8px 0;
    margin-right: 4px;
    border-radius: 50%;
    text-align: center;
    width: 36px;
    height: 36px;
    transition: 0.3s;
}
#footer .social-links a:hover {
    background: rgba(255, 255, 255, 0.2);
    color: #fff;
    text-decoration: none;
}

/****VLAVA****/

/*team*/
.team .member {
    margin-bottom: 20px;
    overflow: hidden;
    text-align: center;
    border-radius: 5px;
    background: #fff;
    box-shadow: 0px 2px 15px rgba(0, 0, 0, 0.1);
}
.team .member .member-img {
    position: relative;
    overflow: hidden;
}
.team .member .social {
    position: absolute;
    left: 0;
    bottom: 0;
    right: 0;
    height: 40px;
    opacity: 0;
    transition: ease-in-out 0.3s;
    background: rgba(255, 255, 255, 0.85);
    display: flex;
    align-items: center;
    justify-content: center;
}
.team .member .social a {
    transition: color 0.3s;
    color: #052c3f;
    margin: 0 10px;
    display: inline-flex;

```

```

        align-items: center;
        justify-content: center;
    }
    .team .member .social a:hover {
        color: #85b0be;
    }
    .team .member .social i {
        font-size: 18px;
        line-height: 0;
    }
    .team .member .member-info {
        padding: 25px 15px;
    }
    .team .member .member-info h4 {
        font-weight: 700;
        margin-bottom: 5px;
        font-size: 18px;
        color: #052c3f;
    }
    .team .member .member-info span {
        display: block;
        font-size: 13px;
        font-weight: 400;
        color: #aaaaaa;
    }
    .team .member .member-info p {
        font-style: italic;
        font-size: 14px;
        line-height: 26px;
        color: #777777;
    }
    .team .member:hover .social {
        opacity: 1;
    }
    /*team*/

/*LOAD ANIMATION*/
#loader-wrapper{
    width: 100%;
    height:100%;
    position: fixed;
    z-index: 1000;
    top:0;
    left:0;
    background-color: #191c1f;
    display: flex;
    justify-content: center;
    align-items: center;
}

.loader {
    background: #000;
    background: radial-gradient(#222, #000);
    bottom: 0;
    left: 0;
    overflow: hidden;
    position: fixed;
    right: 0;
    top: 0;
    z-index: 9999;

```

```

}

.loader-inner {
  bottom: 0;
  height: 60px;
  left: 0;
  margin: auto;
  position: absolute;
  right: 0;
  top: 0;
  width: 100px;
}

.loader-line-wrap {
  animation:
    spin 2000ms cubic-bezier(.175, .885, .32, 1.275) infinite
;
  box-sizing: border-box;
  height: 50px;
  left: 0;
  overflow: hidden;
  position: absolute;
  top: 0;
  transform-origin: 50% 100%;
  width: 100px;
}

.loader-line {
  border: 4px solid transparent;
  border-radius: 100%;
  box-sizing: border-box;
  height: 100px;
  left: 0;
  margin: 0 auto;
  position: absolute;
  right: 0;
  top: 0;
  width: 100px;
}

.loader-line-wrap:nth-child(1) { animation-delay: -50ms; }
.loader-line-wrap:nth-child(2) { animation-delay: -100ms; }
.loader-line-wrap:nth-child(3) { animation-delay: -150ms; }
.loader-line-wrap:nth-child(4) { animation-delay: -200ms; }
.loader-line-wrap:nth-child(5) { animation-delay: -250ms; }

.loader-line-wrap:nth-child(1) .loader-line {
  border-color: hsl(0, 80%, 60%);
  height: 90px;
  width: 90px;
  top: 7px;
}

.loader-line-wrap:nth-child(2) .loader-line {
  border-color: hsl(60, 80%, 60%);
  height: 76px;
  width: 76px;
  top: 14px;
}

.loader-line-wrap:nth-child(3) .loader-line {
  border-color: hsl(120, 80%, 60%);
  height: 62px;
  width: 62px;
  top: 21px;
}

```

```

}
.loader-line-wrap:nth-child(4) .loader-line {
  border-color: hsl(180, 80%, 60%);
  height: 48px;
  width: 48px;
  top: 28px;
}
.loader-line-wrap:nth-child(5) .loader-line {
  border-color: hsl(240, 80%, 60%);
  height: 34px;
  width: 34px;
  top: 35px;
}

@keyframes spin {
  0%, 15% {
    transform: rotate(0);
  }
  100% {
    transform: rotate(360deg);
  }
}

/*****/
/****mine added****/
.btn{
  transition: all 0.3 !important;
}
#hero .white-bg-button, .white-bg-button{
  border: 2px solid white;
  background-color: #fff;
  color: rgb(247,170,2);
}
#hero .white-bg-button:hover, .white-bg-button:hover {
  border: 2px solid white;
  background-color: rgba(255,255,255,0.4);
  color: #fff;
}
.orange-bg-button{
  background-color: rgba(255,165,0,0.5);
  border: 2px solid rgba(255,165,0, 0.2);
  color: white;
}
.orange-bg-button:hover {
  background-color: rgba(255,165,0,0.8);
  color: #fff;
}
.green-bg-button{
  background-color: rgb(25,135,84);
  color:white;
}
.green-bg-button:hover {
  background-color: darkgreen;
  color: #fff;
}
/*****/
/*FORM*/
.form-control, .btn{
  border-radius: 5px;
}

```

```

.form-control{
    color: grey;
}

input:-webkit-autofill,
input:-webkit-autofill:hover,
input:-webkit-autofill:focus,
input:-webkit-autofill:active,
textarea:-webkit-autofill,
textarea:-webkit-autofill:hover,
textarea:-webkit-autofill:focus,
select:-webkit-autofill,
select:-webkit-autofill:hover,
select:-webkit-autofill:focus {
    -webkit-text-fill-color: grey;
    -webkit-box-shadow: 0 0 0 30px white inset !important;
    transition: background-color 5000s ease-in-out 0s;
    font-size: 1rem;
}
/*no bootstrap focus border*/
textarea:active,
input:active,
textarea:focus,
input:focus,
button:focus,
button:active,
label:focus,
.btn:active,
.btn.active
{
    outline:0 !important;
    -webkit-appearance:none;
    box-shadow: none !important;
}

.form-control:focus{
    color:#2c4964;
}

/*****pointer style*****/
.pointer:hover{
    cursor: pointer;
}

/*****register*****/
#register-form-errors{
    display: none;
}
.register-container, .login-container{
    min-height: 100vh;
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
}
.register-container .survey{
    min-height: 80vh;
    display: flex;
    justify-content: center;
    align-items: center;
}

```



```

        flex-direction: column;
    }
    .survey-three{
        display: flex;
        justify-content: space-between !important;
    }
    .survey-three div{
        height: 50%;
    }
    .reg-services{
        color: white;
        display: flex;
        flex-wrap: wrap;
        justify-content: center;
        align-items: center;
    }
    .reg-services .icon-box{
        width: 180px;
        text-align: center;
        padding: 70px 20px 80px 20px;
        transition: all ease-in-out 0.3s;
    }
    .reg-services .icon-box .fa-solid{
        font-size: 40px;
    }
    .reg-services .icon-box .icon {
        margin: 0 auto;
        width: 100px;
        height: 100px;
        display: flex;
        align-items: center;
        justify-content: center;
        transition: ease-in-out 0.3s;
        position: relative;
    }
    .reg-services .icon-box .icon i {
        font-size: 36px;
        transition: 0.5s;
        position: relative;
    }
    .reg-services .icon-box .icon svg {
        position: absolute;
        top: 0;
        left: 0;
    }
    .reg-services .icon-box .icon svg path {
        transition: 0.2s;
        fill: antiquewhite;
    }
    .reg-services .icon-box h4 {
        font-weight: 600;
        margin: 10px 0 15px 0;
        font-size: 22px;
    }
    .reg-services .icon-box h4 a {
        color: #2c4964;
        transition: ease-in-out 0.3s;
    }
    .reg-services .icon-box p {
        line-height: 24px;
        font-size: 14px;
    }

```

```

        margin-bottom: 0;
    }
    .hover-shadow:hover {
        border-color: #fff;
        box-shadow: 0px 0 25px 0 rgba(0, 0, 0, 0.1);
    }
    .reg-services .iconbox-blue i {
        color: #47aeff;
    }
    .reg-services .iconbox-blue:hover .icon i {
        color: #fff;
    }
    .reg-services .icon-box .icon:hover svg path{
        fill: orange;
    }

/*****PROFILE*****/
.profile{
    background-color: white;
    color: rgb(30,30,30);
}
.profile-info{
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding-top: 50px;
    padding-bottom: 10px;
    margin-bottom: 40px;
    border-bottom: antiquewhite 2px solid;
}
.profile .information{
    padding-bottom: 70px;
}
.main-profile-info{
    display: flex;
}
.main-profile-info #profile-picture{
    height: 140px;
    width: 140px;
    border-radius: 50%;
    margin: 20px;
    font-size: 140px;
}

#edit-profile-container form svg{
    right: 30px;
    top: 0;
    width: 40px;
}
#edit-profile-container .form-group{
    padding: 10px;
}
#edit-profile-container .form-group label{
    padding-bottom: 5px;
}
#edit-profile-navbar-container{
    background-color: rgb(255,255,255);
    padding: 10px;
    color: #ffc720;
    border-radius: 15px;
}

```

```

#edit-profile-navbar{
    margin-top: 15%;
}
#edit-profile-navbar a{
    color: rgba(247, 170, 2, 0.7);
    transition: 0.3s;
    font-weight: 500;
}
#edit-profile-navbar a:hover{
    color: rgba(247, 170, 2, 1)
}
#change-password-form svg{
    width: 30px;
    right: -20px;
    bottom: 0;
}
/*****italian flag*****/
#flag-container{
    position: relative;
}
#flag-container p{
    position: absolute;
    top: 390px;
    left: 450px;
    font-size: 75px;
    font-weight: bold;
}

/*****arrow*****/
#arrow-container{
    display: flex;
    justify-content: center;
    margin-top: 180px;
    margin-bottom: 100px;
}
#arrow-container img{
    height: 120px;
}
/*****x sign*****/
.x-sign{
    width: 27px;
    height: 60px;
}
.x-sign span{
    display: block;
    width: 27px;
    height: 2px;
    background-color: white;
}
.x-sign span:first-child{
    transform: rotate(45deg);
}
.x-sign span:nth-child(2){
    transform: rotate(-45deg);
}

/*****PRACTICE*****/
.challenge-container{

```

```

    height: inherit;
    padding: 0px 100px;
}
.progress-container{
    height: 80px;
}
.question-container, .answer-container{
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
}
.question-container{
    font-size: 30px;
}
.question-container strong{
    color: #343a40;
}
.challenge-container div{
    padding-top: 20px;
}
.challenge-container .task-container{
    padding-top: 30px;
}
.challenge-container .progress-container{
    height: 10%;
}
.challenge-container .progress-container .progress{
    padding-top: 0px;
}
.challenge-container .task-container{
    height: 70%;
}
.challenge-container .button-container{
    height: 10%;
}
.choices-container div{
    padding: 5px;
}
.challenge-container #answer{
    width: 550px;
    height: 30px;
    border-bottom: 2px solid white;
    position: relative;
}
#answer .line{
    display: block;
    width: inherit;
    height: 1px;
    border-bottom: 1px solid black;
}
.answer-container .choice{
    width: 170px;
}
.answer-container .choice button{
    width: inherit;
}
#answer #answer-text{
    position: absolute;
    top: 0px;
}

```

```

    left: 0px;
}
.alert-dismissible {
    padding-right: 2rem;
}
#error-container{
    justify-content: center;
    display: none;
    position: relative;
    width: 50%;
    margin: 0px auto;
}
#error-container div{
    display: flex;
}
#error-container div p{
    margin: 0px;
}

/** these below were intially select-the-matching-pairs */
#error-container .alert{
    padding: 15px;
    margin-bottom: 55px;
    width: 100%;
    position: absolute;
    top: -80px;
}
#error-container .alert div{
    width: 50%;
}
#error-container #error-text{
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    padding: 0px;
}
/** select-the-missing-word */
#select-the-missing-word .task-container{
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
}
#select-the-missing-word .choices-container{
    width: 350px;
    flex-wrap: wrap;
}
#select-the-missing-word .choices-container div{
    width: 50%;
}
#select-the-missing-word .choices-container div input, #select-the-
matching-pairs .choices-container div input{
    width: 100%;
}

/**/
#select-the-matching-pairs .italian-choices, #select-the-matching-pairs
.english-choices{
    padding: 40px;
}

```

```

/****/
#write-this-in-english .task-container{
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
}
#write-this-in-english .answer-container .choice{
    min-width: 80px; /*OVDE NASTAJE PROBLEM AKO JE VECE OD 80, ne mogu da
stavim manje jer gore width 170px se onda primenjuje*/
    margin: 5px;
}
#write-this-in-english .answer-container .choice input{
    width: inherit; /*OVDE NASTAJE PROBLEM AKO JE VECE OD 80, ne mogu da
stavim manje jer gore width 170px se onda primenjuje*/
}
#write-this-in-english .choices-container{
    flex-wrap: wrap;
    justify-content: center;
}

/*****PRACTICE*****/
.practice-nav{
    display: flex;
    justify-content: space-between;
    align-items: center;
    border-bottom: antiquewhite 2px solid;
    margin-bottom: 60px;
}
.practice-nav-img{
    border-radius: 8px;
    width: 45px;
}
.practice-nav ul{
    margin: 20px;
}
.practice-nav ul li{
    padding: 13px;
    display: flex;
    align-items: center;
    color: rgb(40,40,40);
}
.practice-nav ul li i{
    font-size: 30px;
    padding: 5px;
    color: orange;
}
.practice-nav ul li a{
    color: rgb(40,40,40);
}
.practice-nav ul li a:hover{
    color: orange;
}
.levels-container{
    width: 65%;
}
.practice-info{
    color: rgb(40,40,40)
}

```

```

}
.practice-information-container{
    width: 35%;
    background-color: white;
    padding: 15px;
    color: black;
    border-radius: 10px;
}
.practice-information-container h4{
    margin:0px;
}
.practice-information-container a{
    color: #0d6efd;
}
.xp-progress{
    width: 100%;
    display: inline-flex;
}
.xp-progress img{
    height: 60px;
    width: auto;
}
.xp-progress .user-progress-bar-container{
    width: 100%;
}
.levels-container .tasks-container{
    display: flex;
    justify-content: center;
    align-items: center;
    flex-wrap: wrap;
    border-bottom: antiquewhite 3px solid;
}
.levels-container .tasks-container .castle{
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
    margin-bottom: 20px;
}
.levels-container .tasks-container .castle h2{
    font-weight: 600;
    letter-spacing: 4px;
    color: rgb(40,40,40);
}
.levels-container .tasks-container .castle .image{
    height: 80px;
}
.levels-container .tasks-container .castle .image img{
    height: 80px;
}
.levels-container .tasks-container .task-container{
    width: 100%;
    display: flex;
    justify-content: center;
    align-items: center;
}
.levels-container .tasks-container .task-container .task{
    margin: 15px;
    display: flex;
    justify-content: center;
    align-items: center;
}

```

```

        flex-direction: column;
        color: white;
    }
    .task:hover{
        cursor: pointer;
    }
    .task:hover .task-bubble-ring:before{
        transition: 0.2s;
        background-color: #b1a9fc;
    }
    /** circular, for tasks and levels **/
    .task-bubble-ring{
        position: relative;
        border-radius: 50%;
        display: grid;
        place-items: center;
        background-color: lightgray;
    }
    .task-bubble-ring i{
        color: mediumpurple;
        z-index: 998;
        font-size: 31px;
    }
    .task-bubble-ring:before{
        content: "";
        position: absolute;
        height: 84%;
        width: 84%;
        border-radius: 50%;
        background: white;
    }
    .task-bubble-ring{
        width: 80px;
        height: 80px;
    }
    .task:hover .circular-progress:before{
        transition: 0.1s;
        background-color: #aaf2aa;
    }
    .task .circular-progress:before{
        background-color: #90EE90;
    }
    .task h3{
        color: rgb(40,40,40);
    }
    .task-disabled:hover{
        cursor: auto;
    }
    .task-disabled:hover .task-bubble-ring:before{
        background-color: lightgray;
    }
    .task-disabled i{
        color: darkgrey;
    }
    /*task info*/
    .task-info-container{
        display: none;
        padding: 10px;
        position: absolute;
        z-index: 1000;
        margin-left: auto;
    }

```



```

margin-right: auto;
left: -40px;
right: -40px;
text-align: center;
border-radius: 25px;
color: white;
background-image: url("../img/striped-task-info-bg.jpg");
box-shadow: inset 0 0 0 1000px rgba(247,170,2,.9);
}
.little-arrow{
width: 0px;
height: 0px;
position: relative;
border-left: 12px solid transparent;
border-right: 12px solid transparent;
border-bottom: 12px solid rgba(247,170,2,1);
top: -44px;
left: 1px;
z-index: 1000;
}
.task-info-container div{
margin: 5px;
}
.task-info-container h4{
font-weight: 800;
font-size: 16px;
}
.task-info-container .link-container a{
/*color: orange;
font-weight: 440;*/
width: 100%;
height: 100%;
}

/*****Edit Goal Form*****/
.custom-file-input::-webkit-file-upload-button {
display: none;
}
.custom-file-input::before {
content: 'Select a file';
margin: 5px 18px 0px 0px;
outline: none;
-webkit-user-select: none;
font-weight: 700;
}
.custom-file-input:hover::before {
border-color: black;
}
.custom-file-input:active::before {
background: -webkit-linear-gradient(top, #e3e3e3, #f9f9f9);
}

/*****Edit Goal Form*****/
#edit-goal-form label{
display: flex;
justify-content: space-between;
padding: 6px;
border: 2px solid antiquewhite;

```

```

    border-radius: 5px;
    transition: 0.3s;
}
#edit-goal-form label:hover{
    cursor: pointer;
    background: rgba(255,255,255,0.4);
}
#edit-goal-form input{
    display: none;
}

/*****profile statistic*****/
.statistic-component{
    display: flex;
    align-items: center;
    width: 240px;
    padding: 6px 16px;
    margin: 8px;
    border: 2px solid rgba(255,165,0, 0.2);
    border-radius: 5px;
    background-color: rgba(255,165,0, 0.4)
}
.statistic-component i{
    font-size: 24px;
    padding: 10px;
}

/*****finished task page*****/
.finished-container{
    display: flex;
    justify-content: center;
    align-items: center;
    padding: 3rem;
}
.finished-container .finished-image svg{
    position: absolute;
}
.finished-container .finished-image .detail-circle-bottom:nth-child(1){
    top: 0px;
    left: 0px;
}
.finished-container .finished-image .detail-circle-bottom:nth-child(2){
    bottom: 30px;
    right: 0px;
}
.finished-buttons-container{
    padding: 50px;
    border-top: 2px solid white;
}

/*Recap*/
.recap{
    color: black;
}

/***** Recap page - table *****/
.table>:not(:first-child) {
    border-top: none;
}
#words-table-container table{

```

```

}
#words-table-container table thead{

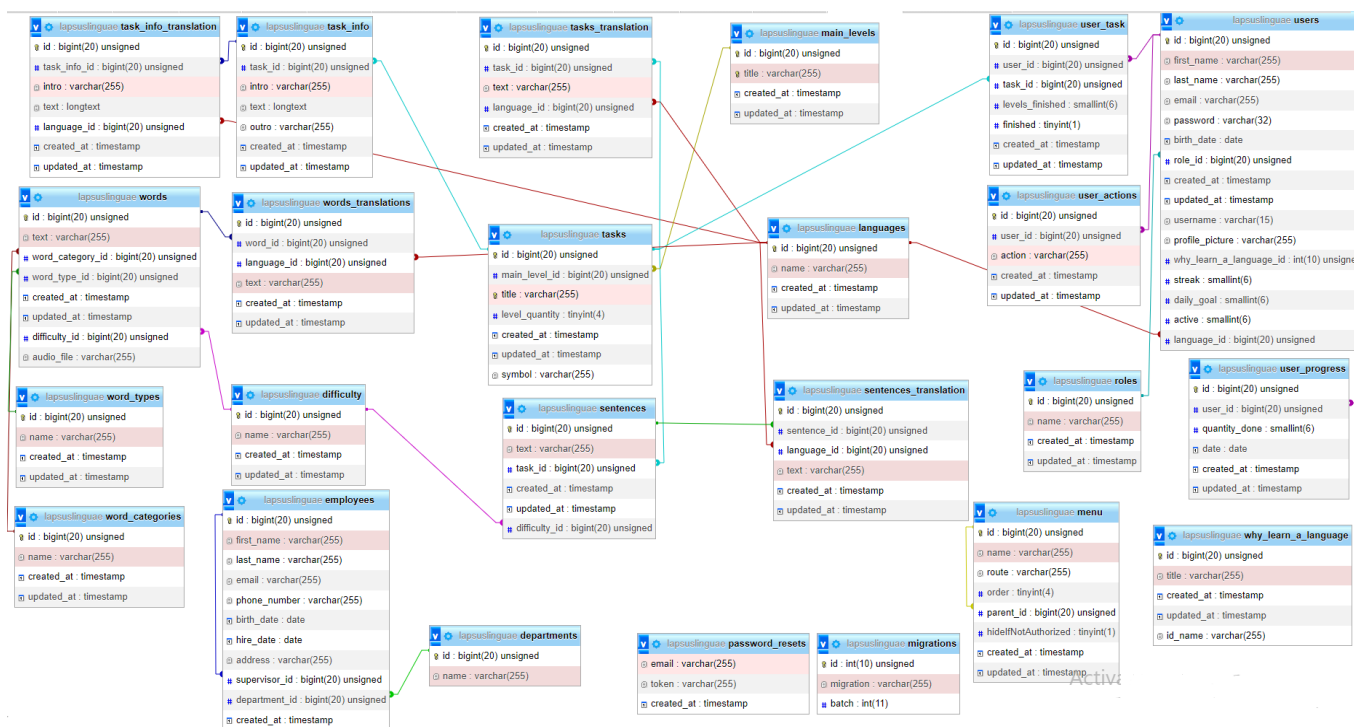
}
tbody, td, tfoot, th, thead, tr {
    border: none;
    color: rgb(40,40,40)
}

/*****      ADMIN      *****/
.admin-index h2, .admin-index label{
    color: rgb(40,40,40);
}
.admin-home h3{
    color: rgb(40,40,40);
}
.admin-table td, .admin-table th, #words-table td, #words-table th{
    text-align: center;
}
table td{
    min-width: 130px;
}
table tr{
    border-bottom: 1px antiquewhite solid;
}
/*****      audio      *****/
.audio-word:hover{
    cursor: pointer;
}

```

6. Baza podataka

6.1 Šema baze podataka



Slika 42 - Prikaz šeme baze podataka

7. Zaključak

Aplikacije za učenje jezika imale su veliki uticaj na lingvistiku jer omogućavaju ljudima da počnu učenje jezika odmah i bez nastavnika. Primarna ideja i cilj aplikacija za onlajn učenje je da obrazovanje učini dostupnim na telefonima i računarima, što podstiče učenje u bilo koje vreme i na mestu na kojem učenici žele.

U tradicionalnom obrazovanju nastavnik koordinira proces učenja, podstiče učenike na rad i pomaže im da reše probleme. Onlajn obrazovanje se, naprotiv, oslanja na volju, samodisciplinu i upornost učenika. Cilj tehnologije je da pomogne studentima da organizuju proces samostalnog učenja i da ostanu na pravom putu.

8. Literatura

1. <https://laravel.com/docs/9.x>
2. <https://www.php.net/>
3. <https://getbootstrap.com/docs/4.1/getting-started/introduction/>
4. <https://developer.mozilla.org/en-US/>
5. <https://fontawesome.com/icons>
6. <https://www.aimprosoft.com/blog/how-to-build-a-language-learning-app/>
7. <https://grammar.collinsdictionary.com/italian-easy-learning/how-do-you-ask-questions-using-question-words-in-italian>
8. <https://www.italianpod101.com/blog/2019/10/24/italian-numbers/>
9. <https://webdizajn.ict.edu.rs/materijali/web-programiranje-php2/predavanja>
10. <https://www.lingohut.com/en/l60/learn-italian>
11. <https://stackoverflow.com/>