Find a Tutor Online

Кирил Герасимовски 211185

Имплементација на софтверски системи со слободен и отворен код

08.09.2024

1. Вовед

Платформата **Find a Tutor Online** (FTO) има за цел да ги поврзе студентите со професори кои сакаат да држат приватни часови за различни предмети од основно ниво се до факултетско ниво, олеснувајќи ги сесиите за учење преку Интернет или во живо. Нудејќи флексибилно закажување и повеќе начини за комуникација, FTO служи како мост за персонализирани искуства за учење. Оваа платформа беше дизајнирана да им овозможи на студентите да најдат професори врз основа на нивните уникатни потреби и преференции, додека професорите можат да ја покажат својата експертиза и достапност.

Овој извештај ги прикажува техничките детали за процесот на развој, вклучувајќи и архитектурата во back-end со користење на **Laravel**, **React** за front-end делот, и **PostgreSQL** за базата на податоци и други алатки вклучени во создавање скалабилна, функционална платформа. FTO се фокусира на обезбедување на ефикасен, лесен интерфејс со робусни карактеристики и за студентите и за професори.

2. Цели

Главните цели на проектот Find a Tutor Online вклучува:

Интеракција учител-ученик: Едноставен и непречен начин за учениците да најдат и да комуницираат со професори врз основа на јазикот, достапноста, локацијата и предметите.
Распоред на сесии: Овозможете им на студентите да резервираат сесии според достапноста на туторите.
Управување со профилот: Студентите и професорите можат да управуваат со нивните профили, вклучувајќи прикачување на профилни слики, лични информации и професорите имаат додатни информации што можат да ги прикачат како што е нивниот начин на предавање, јазиците кои ги зборат и километрите кои би ги патувале за приватен час.
Приспособливост и безбедност: уверете се дека системот може да се справи со многу корисници и да одржува безбедно најавување, управување со сесии и приватност на податоците. Дополнително, постојат мерки на претпазливост за манипулирање на податоци, како што е фалсификување на информации од корисниците. Системот има механизми за валидација на податоци и континуирана проверка на точноста на внесените информации, за да се спречат злоупотреби.
Систем за reviews: Имплементиран механизам за оценување и преглед за студентите да ги видат оцените на професорите врз основа на нивните сесии.

3. Технологии

Front-end: React.js
Front-end делот беше развиен користејќи **React.js** за да обезбеди динамичен и брз кориснички интерфејс. Архитектурата на React која е базирана на компоненти ми овозможи да создадам компоненти за повеќекратна употреба, правејќи ја платформата модуларна и лесна за одржување. **useState,useCallback, useNavigate, useParams, useEffect** hooks на React беа многу користени, додека *Axios* беше користен за справување со HTTP барањата до API-то во back-end.

CSS Modules беа употребени за да ги опфати стиловите на поединечни компоненти, осигурувајќи дека интерфејсот останува чист и одржуван. Frontend-от содржи неколку клучни карактеристики, вклучувајќи:

-Кориснички профили: Студентите и професорите можат да ги ажурираат своите профили, да додаваат профилни , додека само професорите можат да изберат и приспособат достапноста, километрите кои би ги патувале за приватните часови и јазиците на кои предаваат.
-Прикажување на сесиите: Студентите можат да ги отворат сите достапни сесии и низ истите да филтрираат со локација, ниво на едукација и предмет за да најдат соодветни професори за нивните потреби.
-Оценување на професорите: Студентите исто така можат да остават review на професорите за кои им биле на час.
-Резервација на сесии: Студентите можат да ја видат достапноста на закажените часови во реално време и соодветно да резервираат сесии ако има слободни места.
Backend: Ларавел
Backend-от беше изграден со користење на **Laravel**, PHP framework позната по својата едноставност и ефикасност во градењето на робусни апликации. Системот за автентикација на Laravel базиран на API беше употребен за да се справи со безбедни најавувања на корисници, додека JWT (JSON Web Tokens) беа користени за управување со кориснички сесии. Ова осигурува дека корисниците можат безбедно да останат автентицирани низ повеќе сесии.

Следниве главни функционалности беа имплементирани на back-end делот:

-Управување со корисници: Корисничкиот модул им овозможува на професорите и студентите да се регистрираат, да ги ажурираат своите профили и да управуваат со приватните часови.
-Управување со приватните часови: професорите можат да ја дефинираат нивната достапнос, и нивните вештини, додека студентите можат да резервираат сесии врз основа на овие распореди. Сите интеракции се управуваат преку повици на API до back-end делот.

-Ракување со податоци: Моќниот Eloquent ORM на Laravel ги поедноставува интеракциите со базата на податоци, што го олеснува пребарувањето и манипулирањето со податоците од PostgreSQL.
-Безбедност: Податоците се шифрирани, а чувствителните информации беа заштитени со вградените методи за шифрирање на Ларавел.

База на податоци: PostgreSQL

Структурата на базата на податоци во **PostgreSQL** беше дизајнирана да ги складира податоците на корисниците, деталите за сесијата и објави. PostgreSQL беше избран поради неговите моќни можности за барање и способност да се справи со сложени односи, кои беа од витално значење за функции како интеракциите помеѓу професорите и учениците, интеракциите помеѓу user и teacher-profile, следење сесии и ознаки, crud операциите на објавите и reviews.

Клучните табели во базата на податоци вклучуваат:

- **Корисници**: Кориснички информации (професори и студенти) како што се имиња, е-пошта, профилни слики и улоги.
- **Објави**: Управување со достапноста на професор студентски резервации.
- **Reviews**: Табела за справување со прегледите и оценките дадени од студентите за професорите.
- **Teacher-profile**: релациона табела што ги поврзува корисниците со додатните информации на професорот ако избрале улога на професор.
- **Ознаки**: информации за нивото на образование и предметот кој ќе се предава.
- **Post-student**: релациона табела што ги поврзува студентите со објавата во која резервирале место.
- **Tag-post**: релациона табела што ги поврзува ознаките со објавите.

Употребата на карактеристиките на релационата база на податоци на PostgreSQL го олесни одржувањето на интегритетот на податоците додека ги поддржува растечките потреби на платформата.

4. Карактеристики и функционалност

4.1 Управување со кориснички профили

Системот на кориснички профили е еден од темелите на платформата FTO. И туторите и студентите имаат приспособливи профили кои вклучуваат лични информации, како што се име, е-пошта и слика на профилот, како и конкретни детали врз основа на нивната улога ако е професор. Ова ја вклучува нивната достапност, начинот на кој предаваат дали ќе биде онлајн, во живо или двете, јазиците што ги зборуваат и колку би патувале за сесиите во живо.

Клучни аспекти на управувањето со профилот:

- Полиња за уредување: Корисниците можат да ги ажурираат деталите за нивниот профил во секое време.
- Поставување слики: Корисниците можат да прикачуваат или менуваат нивните профилни слики користејќи интуитивен систем за испраќање датотеки.
- Управување со достапност: професорите можат да постават одредени денови и времиња кога се достапни, помагајќи им на студентите да резервираат сесии по нивна погодност.

4.2 Достапност на професори и резервација на објави

Системот за резервација на објави игра централна улога во платформата, овозможувајќи им на студентите да ја видат достапноста на професорите во реално време и соодветно да резервираат место за првиот приватен час во објавата.
Достапноста на професорите е зачувана во структуриран формат, што го олеснува барањето и прикажувањето во frontend-ot.
Студентите можат да резервираат сесија, а штом резервацијата ќе биде потврдена, двете страни се известени со mail.
Професорите можат да ја ажурираат нивната објава и исто така можат да ги прифатат или одбијат барањата за резервации.

4.3 Филтри

Еден од најважните аспекти на Find a Tutor Online е способноста на студентите да ги филтрираат туторите врз основа на предметот за кој што бараат приватни часови, ниво на образованието и локацијата каде ќе се одвиваат предавањата. Ова е особено вредно за студентите кои бараат професори во нивниот град.

Дополнително, професорите можат да одредат дали се достапни за онлајн туторство, лично туторство или и двете. Студентите потоа можат да бараат тутори врз основа на нивниот стил на учење.

4.4 Оценки и критики

По завршувањето на сесијата, студентите можат да ги оценат и прегледаат своите професори. Овој систем обезбедува транспарентност и им помага на другите студенти да донесуваат информирани одлуки при изборот на професор. Професорите исто така имаат корист од позитивните критики, бидејќи го зголемува нивниот кредибилитет на платформата.

4.5 Е-пошта

Известувањата по е-пошта се користат за да се информираат и учениците и наставниците за важни настани. На пример, кога ученикот се запишува на објава, и наставникот и ученикот добиваат известување преку е-пошта. Facade на пошта на Laravel се користи за испраќање на овие известувања, обезбедувајќи комуникација во реално време помеѓу системот и неговите корисници.Исто така корисниците можат да праќаат emails до администраторот на системот со помош на контакт формата, додека е-пошта

нотификациите се активираат при уписот на студентите или креирањето на објавата. Системот, исто така, користи прилагодени шаблони за е-пошта, обезбедувајќи персонализирана содржина и содржина специфична за контекстот.

```php
Mail::to($teacher->email)->send(new TeacherNotification($teacher, $post, $user));
Mail::to($user->email)->send(new StudentNotification($teacher, $post, $user));
```

```php
1 reference | 0 overrides
public function submit(Request $request)
{
    $data = $request->validate([
        'name' => 'required|string|max:255',
        'email' => 'required|string|email|max:255',
        'message' => 'required|string',
    ]);

    Mail::send('emails.contact', ['data1'=>$data], function ($message)use ($data)  {
        $message->from($data['email'], $data['name'])
        ->to('kiril.gerasimovski@gmail.com')
                ->subject('New Contact Form Submission');

    });

    return response()->json(['message' => 'Email sent successfully'], 200);
}
```

4.6 Password Reset

Функционалноста за ресетирање лозинка е направена со користење на приспособено известување за ресетирање лозинка. Оваа класа го проширува основното известување Laravel, испраќајќи приспособена е-пошта со врска за ресетирање преку одреден приказ. УРЛ-адресата за ресетирање го вклучува потребниот токен и корисничка е-пошта, обезбедувајќи безбедно и непречено менување на лозинката за корисниците.

```php
public function toMail($notifiable)
{

    $url = url('/password/reset-form?token=' . $this->token . '&email=' . urlencode($notifiable->email));

    return (new MailMessage)
        ->view('emails.reset-password', ['user' => $notifiable, 'resetUrl' => $url])
        ->subject('Password Reset Request');
}
```

4.7 Factories and Seeders

Factories и Seeders ги користам за генерирање на податоци за тестирање и за пополнување на базата на податоци за време на развојот. На пример, **UserFactory** создава корисници со различни улоги (ученик, наставник), додека **PostFactory** создава тест објави поврзани со случајни корисници и ознаки. **ReviewFactory** е одговорен за генерирање reviews помеѓу наставниците и учениците, обезбедувајќи да не се создаваат дупликат критики.

```php
User::factory(50)->create(['role' => 'student']);

User::factory(10)->create(['role' => 'teacher'])->each(function ($user) {
    TeacherProfile::factory()->create(['user_id' => $user->id]);
});

Post::factory(20)->create();

Review::factory(20)->create();
```

DatabaseSeeder ги користи овие фабрики за да создаде реална база на податоци од корисници, објави и рецензии. Дополнително, ReviewSeeder ги пополнува прегледите засновани на односите на наставниците и учениците, обезбедувајќи сеопфатна покриеност на тестот за системот.

5. Предизвици со кои се соочуваат

Додека проектот напредуваше непречено во најголем дел, во текот на развојот се појавија неколку предизвици:

5.1 Интеграција на Frontend и Backend

Обезбедувањето непречена комуникација помеѓу frontend делот на React и backend делот на Laravel беше критично. Ова вклучуваше внимателно управување со барањата за API и обезбедување дека **CORS** (Споделување ресурси со вкрстено потекло) е правилно конфигуриран за да овозможи безбеден пренос на податоци помеѓу двата системи.
Исто така дека немав предходно знаење за React и општо со frontend-от доста предизвикувачки ми беше да се справам со hooks алатките на React, ама ги совладив препреките на крајот.

5.2 Ракување со подигнувања на датотеки

Поставувањето профилни слики претставуваше технички предизвик, особено кога се интегрира ракување со датотеки помеѓу предниот дел и задниот дел. Ова бараше изградба на сопствени API рути во Ларавел за безбедно поставување и складирање на слики, како и обезбедување дека сликите може ефикасно да се преземат и прикажуваат во предниот дел.

5.3 Динамичко управување со објавите

Функцијата за динамичка достапност бара внимателно планирање за да се осигура дека професорите можат да ги ажурираат своите распореди во реално време и да спречат двојни резервации или reviews, или да се спречат студентите да ги имаат истите функционалности како и професорите. Ова вклучуваше дизајнирање шеми за бази на податоци што може да се справат со сложени податоци и обезбедување дека промените направени од професорите се рефлектираат веднаш на frontend делот.

5.4 Автентикација

Спроведувањето на автентикација базирана на API користејќи токени JWT вклучуваше управување со безбедно складирање на токени и обезбедување на валидност на токените на различни уреди и сесии. Обезбедувањето истекување и обновување на токен, исто така, бараше внимателно управување за да се спречи неовластен пристап.

6. Идни подобрувања

Додека тековната верзија на Find a Tutor Online ми ги исполни своите првични цели, постојат неколку области за подобрување и дополнителни функции што би можеле да се имплементираат во иднина:

- ➔ Развој на мобилни апликации: Додека платформата реагира, развојот на посветена мобилна апликација ќе го подобри корисничкото искуство.
- ➔ Комуникација во реално време: Интегрирањето на пораки или видео повици во реално време помеѓу учениците и туторите може да го подобри искуството за учење.
- ➔ Интеграција на плаќање: Автоматизирањето на плаќањата за туторските сесии, вклучително и безбедното ракување со трансакциите, би го поедноставило процесот и за студентите и за туторите.
- ➔ Напредни филтри за пребарување: Подобрувањето на функционалноста за пребарување за да вклучи повеќе грануларни филтри, како што се оценки за учители, специфични прозорци за достапност и методи на настава, ќе ја подобри употребливоста на платформата.

7. Заклучок

Платформата Find a Tutor Online успешно ја постигнува својата цел да ги поврзе студентите и туторите на беспрекорен начин и лесен за корисниците. Преку комбинација од робустен backend дел со Laravel, динамичен frontend дел со React.js и добро структурирана база на податоци со PostgreSQL, системот обезбедува високо флексибилно, скалабилно решение за персонализирано учење. Иако имаше предизвици, употребата на модерни технологии, внимателна архитектура и вниманието на корисничкото искуство и овозможија на платформата да успее да ги исполни своите цели.

Понатамошните подобрувања и проширувања на функциите ќе овозможат FTO да остане вредна алатка за учениците и туторите.

8. References

- Laravel Documentation: https://laravel.com/docs/
- React Documentation: https://reactjs.org/docs/
- Laravel Learning: https://laracasts.com/
- https://www.reddit.com/
- https://stackoverflow.com/