



Универзитет „Св. Кирил и Методиј“ во Скопје
Факултет за информатички науки и компјутерско инженерство

PetMate

Тимски проект

Ментор

Проф. д-р Петре Ламески

Кандидат

Христина Трајческа 213211

Тамара Ончевска 213020

Ања Шулајковска 213158

Септември 2025

Апстракт

Проектот PetMate е мобилна апликација чија цел е да се помагање на сопствениците на миленици да најдат соодветни партнери за парење за нивните миленици од истата раса, како и форум кој што ќе им помогне на одговор на било какви прашања. Преку апликацијата, корисниците имаат можност да креираат и управуваат со профили на нивните миленици, да поставуваат барања за парење, да учествуваат во дискусии на форум и да споделуваат искуства со други сопственици на миленици.

Содржина

Апстракт	ii
1. Вовед	1
2. Технологии што ги користевме	1
3. Функционалности и аспекти на проектот.....	2
4. Backend: Laravel PHP Framework	4
5. Filament Admin Panel	5
6. Frontend: Flutter (Dart).....	5
7. Flutter UI (User Interface)	6
8. Комуникација со Backend(API)	7
9. Заклучок	8

1. Вовед

За реализацијата на проектот, користевме две главни технологии: PHP Laravel за backend развој и Flutter (Dart) за frontend мобилна апликација. Laravel ни овозможи креирање на RESTful API, управување со податоците и обезбедување на сигурна автентикација, додека Flutter ни помогна да изработиме мобилна апликација која е визуелно атрактивна и лесна за користење.

Апликацијата е дизајнирана да ги покрива сите основни аспекти на мобилните апликации и системи, како што се state management, автентикација, персонализирани UI елементи, web services и управување со податоци. Вклучува повеќе од 7 екрани, овозможувајќи корисниците да имаат целосно искуство од моментот на регистрација па сè до управување со нивните миленици.

2. Технологии што ги користевме

Во проектот PetMate користевме неколку различни технологии за да го обезбедиме функционалниот и визуелниот аспект на апликацијата. Овие технологии ни овозможија да изградиме стабилен систем кој ги исполни потребите на нашите корисници:

- Backend: Laravel PHP Framework - Користевме Laravel за креирање на RESTful API, обработка на податоците и обезбедување на сигурност во системот.
- Frontend: Flutter (Dart) - За изградба на мобилната апликација која е достапна за iOS и Android корисниците.
- База на податоци SQLite - Базата на податоци ја користевме за чување и обработка на податоците.
- Автентикација Laravel Sanctum - Овој систем го имплементиравме за безбедно управување со автентикацијата и авторизацијата на корисниците.
- Администраторски панел Filament Admin Panel - Овој панел го користевме за лесно управување со административните задачи преку визуелен интерфејс.

3. Функционалности и аспекти на проектот

Кога го изработувавме проектот PetMate, се трудевме да ги покриеме сите основни аспекти на мобилниот развој, вклучувајќи различни функционалности кои беа оценети според процентуалната важност на проектот. Овие аспекти беа многу важни за да ја постигнеме целта на проектот.

3.1. State Management

State management е клучен аспект на мобилните апликации. За да може да управуваме со состојбата на апликацијата, користевме Provider во Flutter. Овој пакет ни овозможи да управуваме со состојбата на апликацијата централизирано, така што сите екрани можат да го читаат и менуваат истиот податок. Ова го олеснува обновувањето на интерфејсот кога корисниците ќе извршат промени, како што е поставување ново барање за парење или ажурирање на профилот на миленичето.

3.2. Authentication

За сигурна автентикација и авторизација на корисниците, се одлучивме да користиме Laravel Sanctum. Оваа технологија овозможи лесна и безбедна автентикација преку токени. Со помош на Sanctum, корисниците можат да се регистрираат, да се логираат и да управуваат со своите профили без да имаат потреба од сложени безбедносни механизми.

```
public function login(LoginRequest $request): JsonResponse
{
    $user = User::query()
        ->firstWhere( column: 'email', $request->get( key: 'email'));

    if (! $user || ! Hash::check($request->get( key: 'password'), $user->password)) {
        throw ValidationException::withMessages([
            'email' => ['The provided credentials are incorrect.'],
        ]);
    }

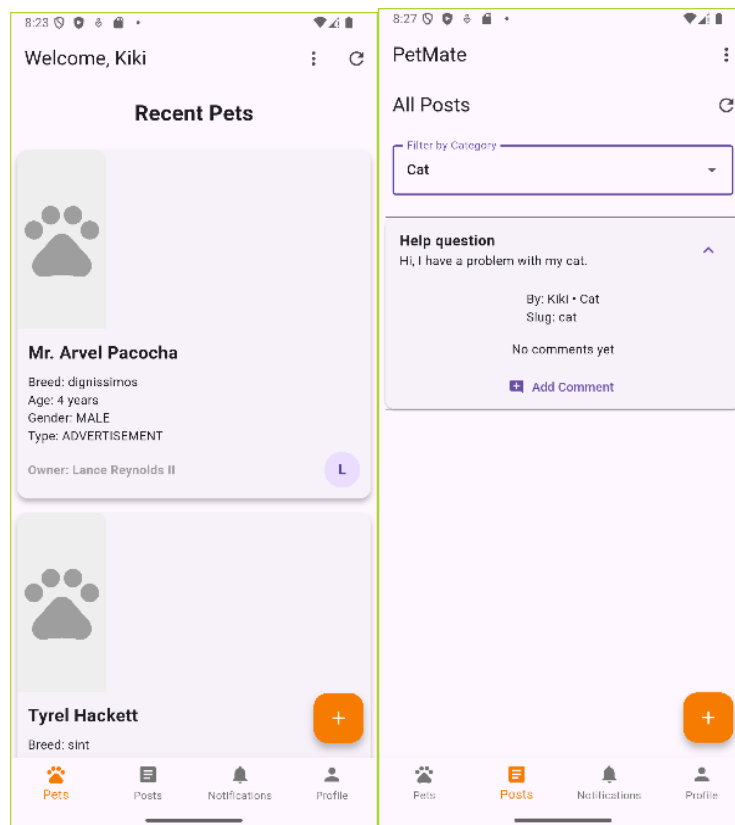
    $token = $user->createToken( name: 'token')->plainTextToken;

    return JsonResponse::make($user)->additional(['token' => $token]);
}
```

Слика1: Пример за API автентикација во Laravel

3.3. Custom UI Elements

Креирање и користење custom widgets во Flutter за да ја подобриме визуелната привлечност на апликацијата. Овие персонализирани UI елементи вклучуваат елементи како профилни слики за миленичиња, картички за барања за парење и персонализирани форми за внесување податоци. Тие елементи се создадени да бидат визуелно атрактивни и лесни за користење, со што го подобруваат корисничкото искуство.



Слика 2: Изглед на некои custom widgets

3.4. Web Services

Во проектот користевме RESTful web services за комуникација помеѓу мобилната апликација и backend серверот. Laravel API точките ни овозможуваат да создадеме функционалности кои можат да се повикаат од Flutter апликацијата, како што се поставување барања за парење и добивање информации за миленичиња.

3.5. Data Handling

Управувањето со податоците беше многу важно за оваа апликација. Користевме Eloquent ORM во Laravel за моделите, што ни овозможи да создадеме релации меѓу различните ентитети, како што се корисници, миленичиња и барања за парење. Базата на податоци SQLite ја користевме за чување на сите овие податоци и за брзо пристапување до нив.

3.6. Navigation

За управување со навигацијата помеѓу екрани во Flutter, користевме Navigator 2.0. Ова ни овозможи динамички да ги менуваме патеките на апликацијата во зависност од акциите на корисниците, како што се преглед на профил на милениче или поставување ново барање за парење.

4. Backend: Laravel PHP Framework

Backend делот на проектот е изграден со помош на Laravel PHP Framework. Laravel е еден од најпопуларните PHP фрејмворци и нуди богат сет на алатки и механизми кои го олеснуваат развојот на веб апликации. Во овој проект, Laravel не само што се користи за креирање на API, туку и за управување со сите серверски активности, бази на податоци и комуникација помеѓу frontend и backend.

Проектот користи Model-View-Controller (MVC) архитектура која ја олеснува поделбата на логиката и презентацијата. Основните компоненти на MVC архитектурата се:

- **Модели (Models):** Служат за поврзување со базата на податоци и управување со податоците.
- **Контролери (Controllers):** Служат како посредници помеѓу моделите и видиците (views), обработуваат барања од корисниците и ги враќаат резултатите.
- **Рути (Routes):** Одредуваат како да се обработат барањата на корисниците и кои контролери и методи да се повикаат.

5. Filament Admin Panel

За административно управување со податоците, проектот користи Filament Admin Panel. Filament е моќен и лесен за користење администраторски панел за Laravel апликации. Со него, администраторите на апликацијата можат да управуваат со корисници, миленичиња, барања за парење и други податоци без потреба од пишување код.

```
class PetResource extends Resource
{
    protected static ?string $model = Pet::class;

    no usages
    protected static ?string $navigationIcon = 'heroicon-s-user-group';

    Tamara213020
    public static function form(Form $form): Form
    {
        return $form
            ->schema([
                Forms\Components\TextInput::make('name')
                    ->required(),
                Forms\Components\TextInput::make('breed')
                    ->required(),
                Forms\Components\TextInput::make('age')
                    ->required()
                    ->numeric()
            ])
    }
}
```

Слика 3. PetResource во Filament

Функциите што ги нуди Filament ја зголемуваат продуктивноста при управување со големи количини податоци и ја олеснуваат администраторската работа.

6. Frontend: Flutter (Dart)

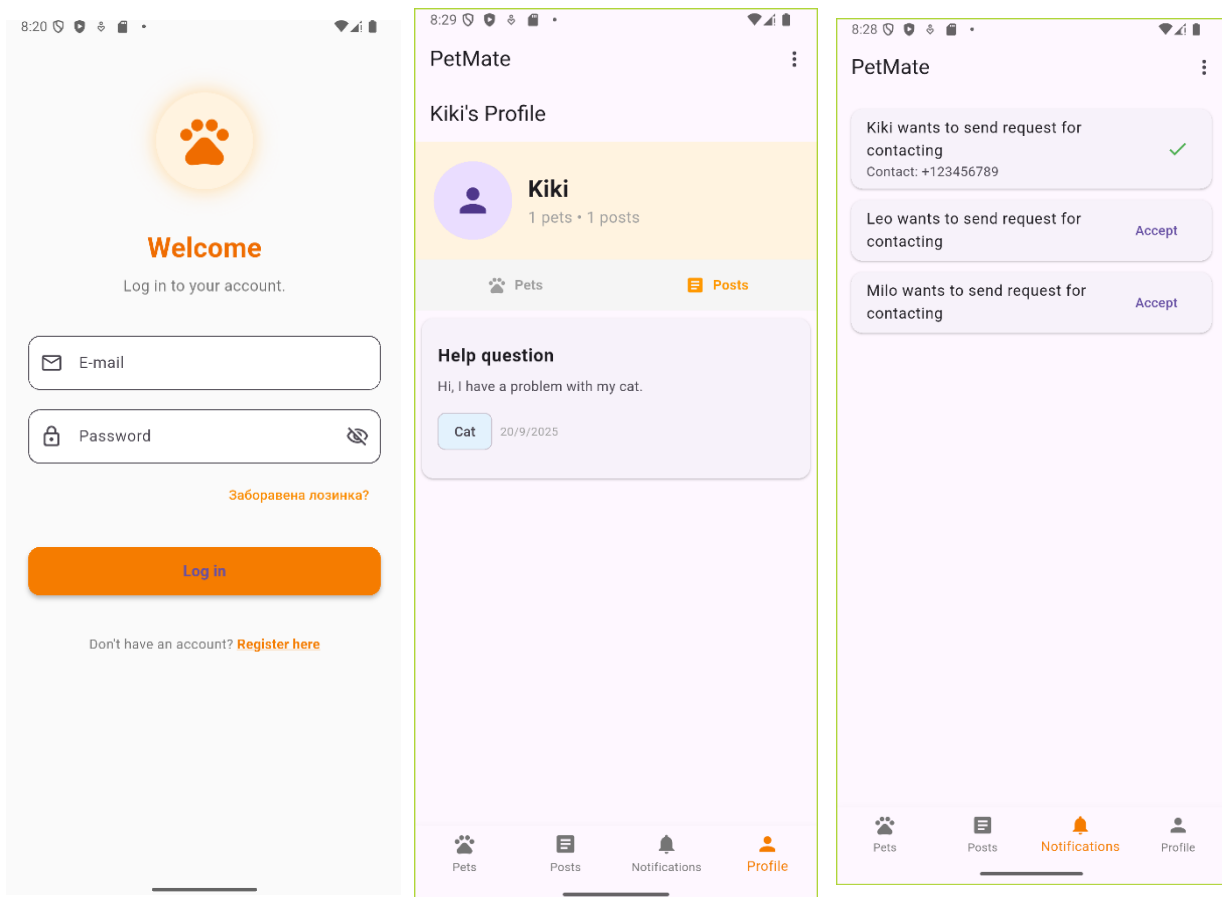
Frontend делот на проектот е изграден со Flutter, што овозможува развој на високо продуктивни и визуелно атрактивни мобилни апликации за iOS и Android. Flutter користи Dart како основен јазик за програмирање и нуди висококвалитетни UI компоненти кои лесно можат да бидат интегрирани со backend серверот преку RESTful API.

7. Flutter UI (User Interface)

Flutter нуди широк спектар на widget-и за изградба на интерфејсите. Секој екран во мобилната апликација се состои од комбинација на различни widget-и.

За интерфејсот на PetMate, развиени се неколку клучни екрани како:

- Login/Registration Screen: Овозможува регистрација и логирање на users.
- Pet Profile Screen: За управување со профилите на миленичињата.
- Breeding Request Screen: За креирање на барања за парење.
- Forum Screen: За учество во форумот и објавување на постови.



Слика 4. Дел од екраните

8. Комуникација со Backend(API)

Frontend компонентите во Flutter се поврзани со backend системот преку HTTP барања. За да се овозможи комуникација, користиме Flutter пакетот http за испраќање GET и POST барања до Laravel серверот.

```
class PetService {  
    final String baseUrl = 'http://10.0.2.2:8000/api/v1';  
  
    Future<List<Pet>> getPets({int? userId, String? type}) async {  
        try {  
            String url = '$baseUrl/pets?';  
  
            final params = [];  
            if (userId != null) params.add('user_id=$userId');  
            if (type != null) params.add('type=$type');  
  
            if (params.isNotEmpty) {  
                url += params.join('&');  
            }  
  
            print('🔄 Fetching pets from: $url');
```

Слика 5. Fetching на податци/Комуникација со Backend

9. Заклучок

Проектот PetMate ги користи силните страни на Laravel за backend развој, со користење на MVC архитектура, политики за безбедност, и Filament за административно управување. Flutter ја носи апликацијата на мобилните платформи, овозможувајќи брза и интуитивна интеракција со корисниците. Заедно, овие технологии создаваат стабилна, безбедна и кориснички ориентирана апликација за сопствениците на миленици.

PetMate е успешна имплементација која ги комбинира сите важни аспекти на мобилниот развој, од автентикација, преку управување со податоци, до персонализирани UI елементи и web services. Сето ова го изработивме со цел да му обезбедиме на корисникот најдоброто искуство при користење на апликацијата. Благодарение на добро дефинираните технологии и тимската работа, овој проект претставува успешен пример за мобилна апликација која ги задоволува сите потреби на современите сопственици на миленици.