

**Документација за проект по предметот Имплементација на
софтверски системи со слободен и отворен код**



Изработено од:

Момчило Илиев, 183248

Сани Раданлиева, 203120

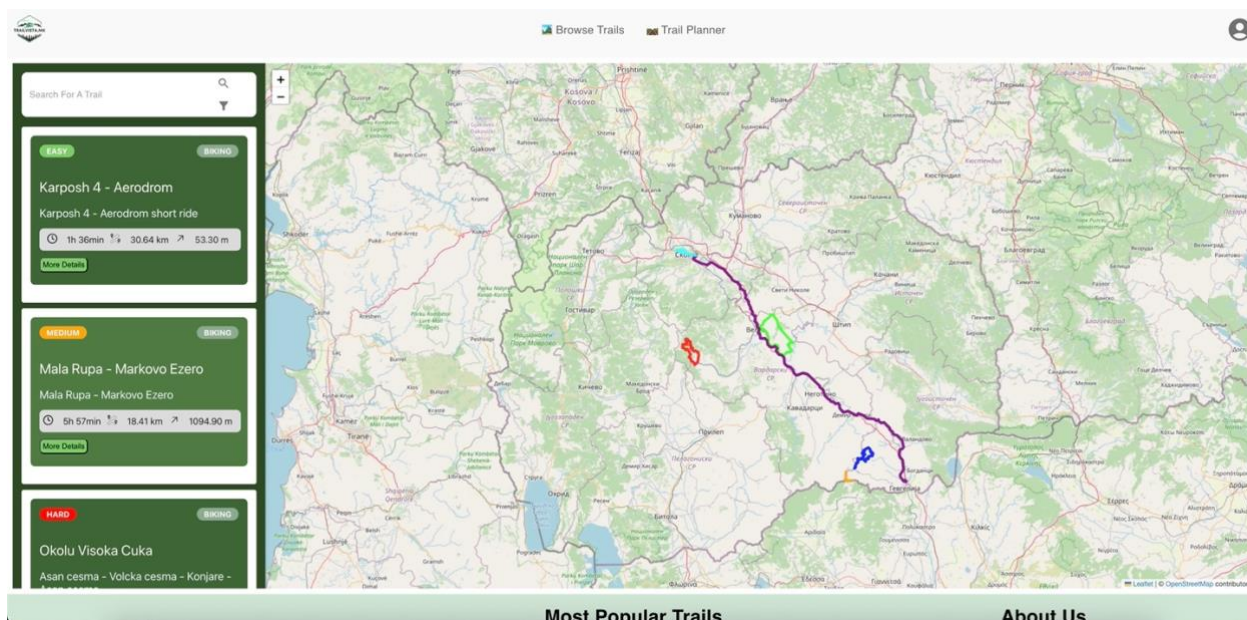
1. Вовед

TrailVista.mk е веб апликација за додавање и преглед на нови и веќе постоечки патеки за велосипедизам, планинарење и трчање. За креирање на оваа веб апликација користено е **Laravel** (за backend делот), **React** (за frontend делот), а за база на податоци користено е **PostgreSQL**. За функционалностите поврзани со прикажување и користење на мапи, користено е **OpenStreetMap**.

2. Основни функционалности

Основни функционалности на TrailVista.mk се:

- **Преглед и пребарување на рути** - корисниците можат да пребаруваат низ веќе постоечки рути според различни филтри, како тежина на рута, должина на рута, вид на рута, време на движење и слично, и истата патека да ја прегледаат детално

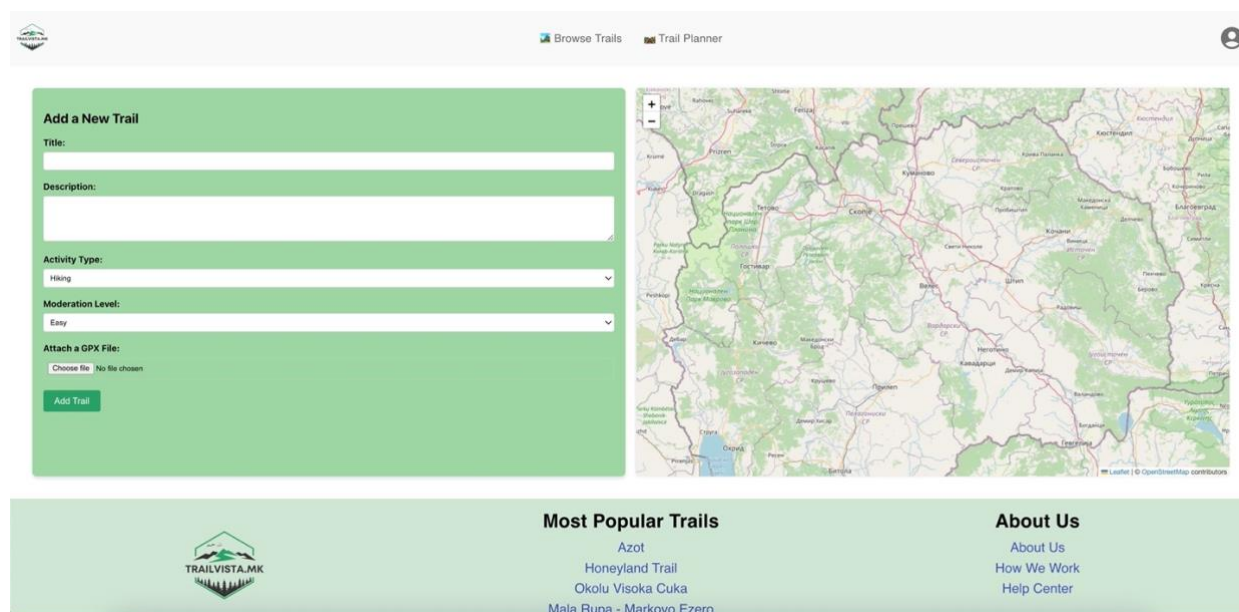


Слика 1. Преглед и пребарување на рути

Во рамки на оваа страна, прикажани се на мапата веќе постоечките рути. За функционалноста поврзана со мапата користена е **OpenStreetMap**, со библиотека **react-leaflet**, преку која се креира **MapContainer** во кој е прикажана мапата. Се користи **Polyline** (исто од библиотеката **react-leaflet**) за исцртување на рутите кои се прикажани на мапата.

На левата страна може да се види една **Sidebar** компонента во која се излистани сите постоечки рути. Овде може да се пребарува по името на рутата, или да се искористи филтерот за понапредно пребарување.

- **Додавање на нова рута** - корисниците можат да додадат нова рута, внесувајќи име на рута, краток опис, вид на активност/спорт и тежина на рутата, и грх фајл



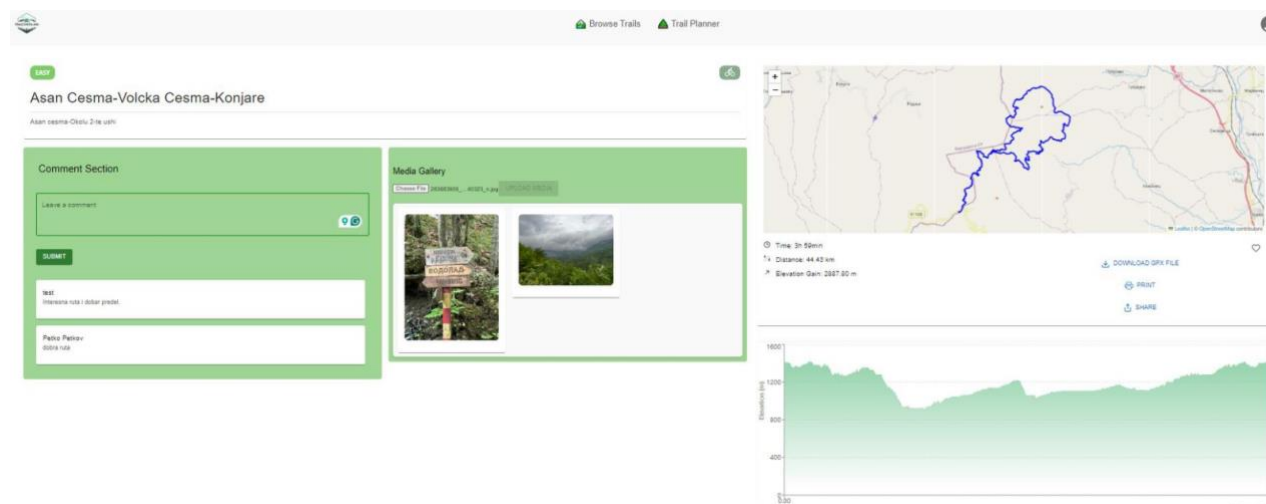
Слика 2. Додавање на нова рута

- **Користење на интерактивна мапа** - корисниците можат да ја искористат интерактивната мапа за да ја видат целосната патека, нејзината должината, тежината и мапа со елевација на целата рута

- **Оставање коментари за рутата** - корисниците можат да ги оценуваат рутите, споделувајќи искуства, фотографии и видеа за патеките

- **Споделување на рути** - корисниците можат да ги споделуваат креираните рути

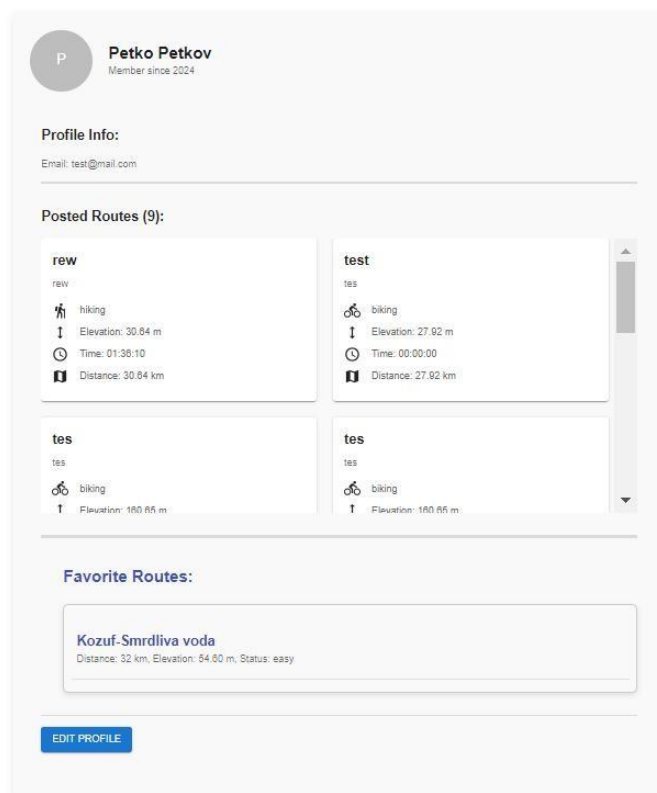
- **Преземање на рути** - корисниците можат да ги преземат рутите и да ги искористат на нивните GPS уреди



Слика 3. Детали за мапа, оставање коментари, споделување и преземање на рута

Исто така, овде може да се забележи мапа со елевација, креирана со помош на библиотеката **recharts**, која ги зема координатите од грх фајлот и ја исцртува елевацијата.

- **Кориснички профили** - корисниците можат да креираат свој профил на којшто може да ги видат основните кориснички информации, да ги прегледаат рутите кои тие ги објавиле и да ги видат зачуваните рути



Слика 4. Кориснички профил

3. Инсталација

За инсталација на проектот TrailVista потребен е **php 8.2** со **Laravel 11**. На следниве линкови може да се најдат инсталациите:

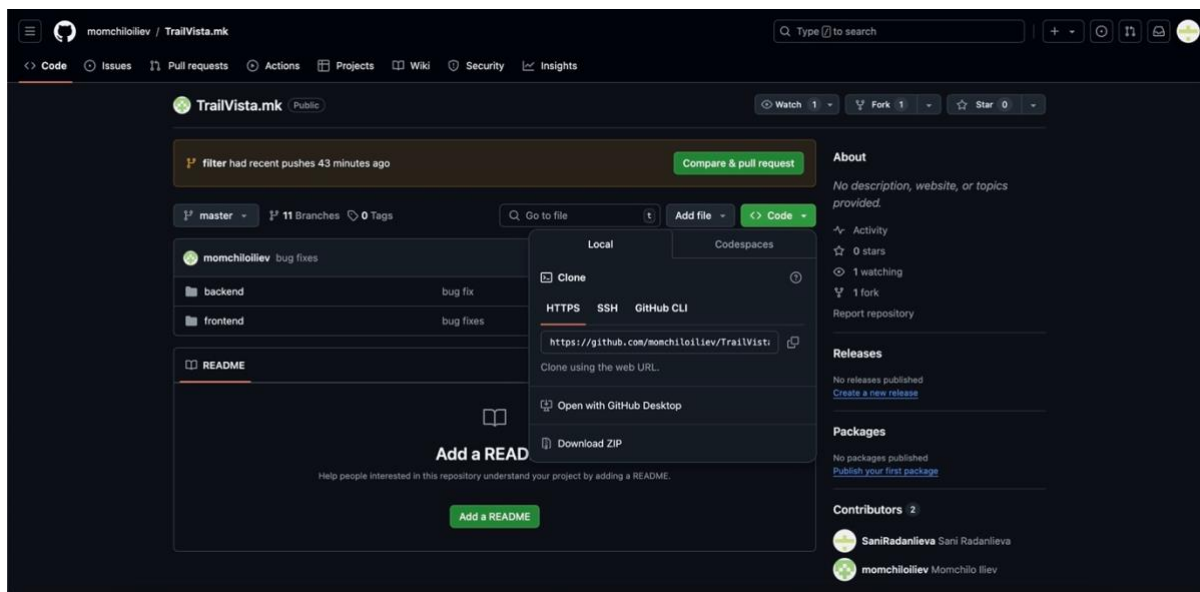
- За инсталација на php: <https://www.php.net/manual/en/install.php>
- За инсталација на composer: <https://getcomposer.org/doc/00-intro.md>
- За инсталација на npm: <https://docs.npmjs.com/downloading-and-installing-node-js-and-npm>
- За инсталација на PostgreSQL: <https://www.postgresql.org/download/>

4. Клонирање на репозиториумот

Клонирање на репозиториумот се врши преку еден од овие два линкови:

<https://github.com/momchiloiliev/TrailVista.mk>

<https://gitlab.finki.ukim.mk/ioss/trail-vista>



Слика 5. Клонирање на репозиториумот преку GitHub

За клонирање на проектот преку GitHub, потребно е да се избере клонирање преку HTTPS, SSH, GitHub CLI. На пример, преку користење на SSH, потребно е да се креира фолдер, да се копира SSH линкот, и локално во терминал во креирано фолдер потребно е да се изврши ***git clone -SSH <url>***.

За инсталација на зависности, потребно е да се влезе во директориумот каде што се наоѓа проектот. За инсталација на зависности потребни за backend, потребно е да се влезе во истоимениот директориум, и да се изврши командата ***composer install***. За инсталација на зависности потребни за frontend, потребно е да се влезе во истоимениот директориум, и да се изврши командата ***npm install***.

Следно, потребно е да се конфигурира ***.env*** фајлот, за користење на базата на податоци во рамки на проектот.

```
DB_CONNECTION=pgsql
DB_HOST=127.0.0.1
DB_PORT=5433
DB_DATABASE=
DB_USERNAME=
DB_PASSWORD=
```

Слика 5. Конфигурирање на ***.env***

По конфигурирање на базата на податоци и по успешно конектирање, потребно е да се креираат миграциите и шемата на базите на податоци, во backend фолдерот на проектот, преку командата ***php artisan migrate***.

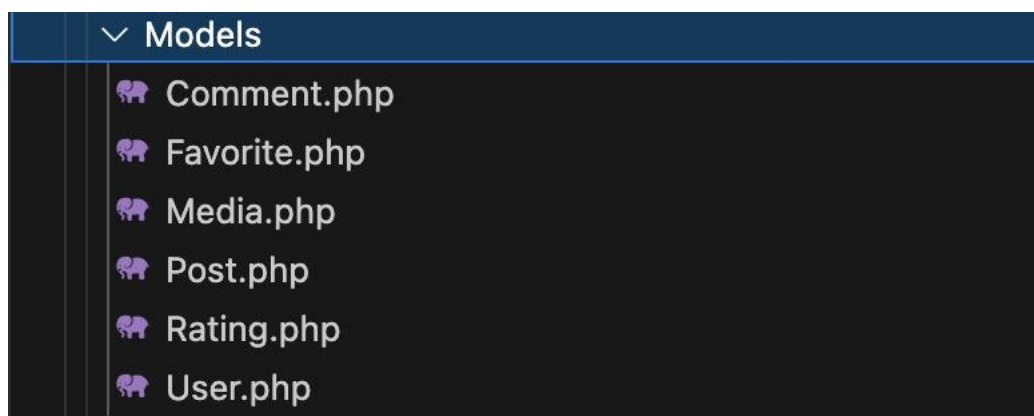
5. Архитектура на проектот

Овој проект има јасно дефинирана структура на директориуми и фајлови која ја олеснува организацијата на кодот и ресурсите. Подолу е преглед на основните директориуми и нивната намена:

- **app/** е главниот директориум на апликацијата, кој содржи модели, контролери, middleware и други класи.
- **bootstrap/** содржи фајлови за иницијализација на апликацијата. Фајлот `app.php` ја иницијализира Laravel рамката.
- **config/** содржи конфигурациски фајлови за апликацијата.
- **database/** вклучува миграции, фабрики и полначи за базата на податоци.
- **public/** е директориум од кој се сервираат јавни ресурси, како CSS, JavaScript и слики.
- **resources/** содржи ресурси како Blade темплејти, JavaScript фајлови, локализационски фајлови и assets.
- **routes/** е директориум кој ги содржи фајловите за дефинирање на апликациските рути, како `web.php`, `api.php`, и други.
- **storage/** се користи за складирање на компајлирани Blade темплејти, сесија, кеширање и фајлови генерирани од апликацијата.
- **tests/** е директориум за unit и feature тестови.
- **vendor/** содржи Composer пакети кои се инсталирани како зависности за проектот.

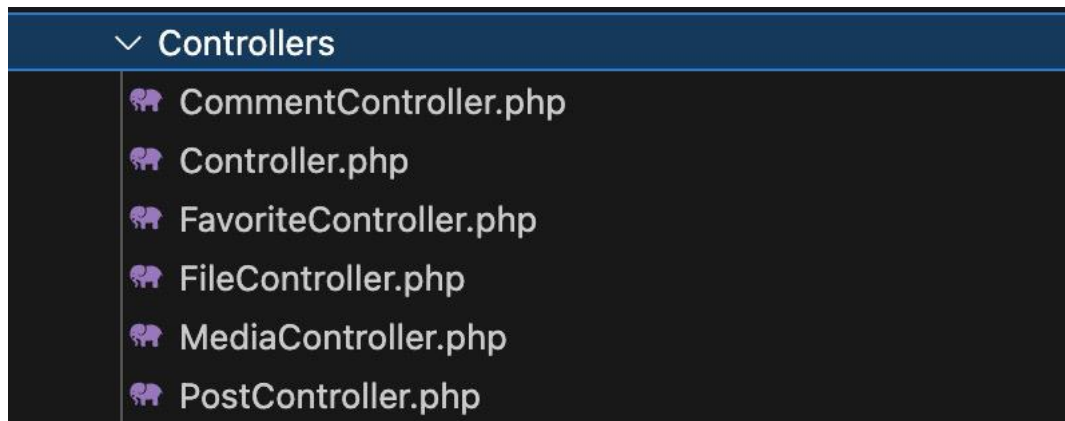
6. Структура на проектот

TrailVista.mk проектот содржи 6 модели, меѓу кои, Comment, Favorite, Media, Post, Rating, User.



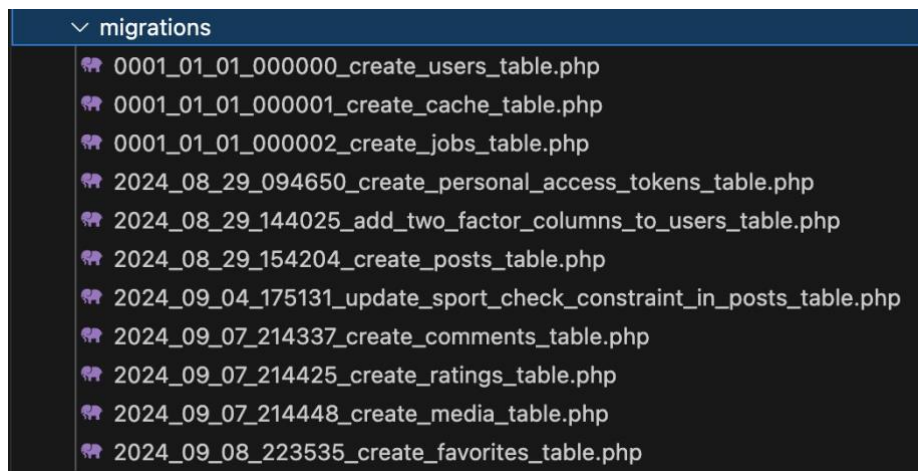
Слика 6. Models

Имаме 6 контролери, како CommentController, Controller, FavoriteController, FileController, MediaController и PostController.



Слика 7. Controllers

Миграциите кои беа креирани се следниве:



Слика 8. Migrations

На следните слики може да се разгледаат рутите кои се користат во рамки на проектот:

```
//POSTS
Route::get('/posts', [PostController::class, 'index']);
Route::get('/posts/{id}', [PostController::class, 'show']);
// Route::post('/posts', [PostController::class, 'store']);

//POSTS COMMENTS RATINGS MEDIA
Route::get('/posts/{postId}/comments', [CommentController::class, 'index']);
Route::get('/posts/{postId}/ratings', [RatingController::class, 'index']);
Route::get('/posts/{postId}/media', [MediaController::class, 'index']);

//MOST FAVORITED ROUTES
Route::get('/most-favorited-routes', [FavoriteController::class, 'mostFavoritedRoutes']);

//GPX FILES
Route::get('/storage/gpx-files/{filename}', [FileController::class, 'getGpxFile']);
Route::get('/download-gpx/{filename}/{title}', [FileController::class, 'getGpxFile']);
```

Слика 9. Јавни руте

```

// Protected routes (Require authentication)
Route::middleware('auth:sanctum')
->group(function () {

    //GET USER DATA
    Route::get('/user', function (Request $request) {
        |   return new UserResource($request->user());
    });

    //POSTS
    Route::post('/posts', [PostController::class, 'store']); // Add a new post
    Route::put('/posts/{id}', [PostController::class, 'update']); // Edit a post
    Route::delete('/posts/{id}', [PostController::class, 'destroy']); // Delete a post

    //POSTS COMMENTS RATINGS MEDIA
    Route::post('/posts/{postId}/comments', [CommentController::class, 'store']);
    Route::post('/posts/{postId}/ratings', [RatingController::class, 'store']);
    Route::post('/posts/{postId}/media', [MediaController::class, 'store']);

    //FAVORITES
    Route::post('/favorites/{postId}', [FavoriteController::class, 'addToFavorites']);
    Route::delete('/favorites/{postId}', [FavoriteController::class, 'removeFromFavorites']);
    Route::get('/favorites', [FavoriteController::class, 'getFavorites']);

});

```

Слика 10. Руте кои бараат автентикација