**TripManager**

[**https://github.com/borofx/TripManager-final**](https://github.com/borofx/TripManager-final)

ASP.NET Core MVC уеб приложение за планиране за туристически обиколки, което позволява на потребителите да създават персонализирани маршрути от предварително дефинирани забележителности. Приложението включва интерактивна карта, форми за вход и регистрация, и различни нива на достъп за администратори и регистрирани потребители.

**АВТОР** - Борис Захариев Илиев

**Компоненти**

TripManager се състои от два основни компонента:

* **База от данни**, която съхранява информация за потребителите, забележителностите и туровете.
* **Уеб приложение**, което комуникира с (LocalDB инстанцията на) базата от данни и предоставя графичен интерфейс за потребителите.

**Функционалност**

**Потребители**

Приложението работи със задължителна оторизация за всички потребители.

* **Администратор** – може да създава, редактира и премахва забележителности и турове.
* **Регистриран потребител** – може да създава и премахва турове от вече добавените забележителности.

Всеки потребител има:

* Имейл
* Парола
* Роля (потребител или администратор)

**Структура на данните**

**Landmark (Забележителност)**

* Id - уникален идентификатор
* Name - име на забележителността (задължително)
* Latitude - географска ширина (задължително)
* Longitude - географска дължина (задължително)

**Tour (Обиколка)**

* Id - уникален идентификатор
* Name - име на обиколката (задължително)
* Description - описание до 500 символа (задължително)
* UserId - идентификатор на създателя
* User - връзка към потребителя създател
* TourLandmarks - колекция от забележителности в обиколката

**TourLandmark (Връзка между обиколки и забележителности)**

* TourId - идентификатор на обиколката
* LandmarkId - идентификатор на забележителността
* Tour - навигационно свойство към обиколката
* Landmark - навигационно свойство към забележителността

**Функционалност на картата**

Картата е реализирана с leaflet.js и предоставя следните възможности:

**Инициализация**

* Центрирана по подразбиране върху София, България (42.6977, 23.3242)
* Мащаб 6
* Използва OpenStreetMap като основен слой

**Визуализация на забележителности**

* Автоматично зареждане на всички забележителности при стартиране
* Маркер за всяка забележителност на съответните координати
* Попъп с името на забележителността при кликване върху маркера

**Визуализация на обиколки**

* Автоматично зареждане на всички обиколки
* Визуализация на маршрута чрез линии между забележителностите
* Червена линия свързва забележителностите в последователността им
* Попъп с името и описанието на обиколката при кликване върху линията
* Минимум 2 забележителности са нужни за визуализация на обиколка

**Динамично обновяване**

* Автоматично опресняване на данните на всеки 10 секунди
* Премахване на старите линии преди визуализация на нови
* Отделно управление на маркерите и линиите на обиколките

**Валидация**

* Задължително име на забележителност
* Задължителни координати за забележителност
* Задължително име на обиколка
* Ограничение на описанието на обиколката до 500 символа

**Сигурност:**

* TourController изисква оторизация за всички действия
* LandmarkController изисква Admin роля (с изключение на GET /api/landmarks)
* Използва се UserManager за идентификация на потребителите
* Имплементирана е проверка за собственост на обиколките

**REST API Интеграция**

GET /api/landmarks

* Публичен достъп (AllowAnonymous)
* Връща всички забележителности

GET /api/tours

* Изисква оторизация
* Връща списък с обиколките на текущия потребител
* Включва детайли за landmarks в обиколката

TourController MVC Endpoints:

* GET /Tour/ManageLandmarks - Admin
* POST /Tour/DeleteLandmark - Admin
* GET/POST /Tour/EditLandmark - Admin
* GET /Tour/MyTours - За вписани потребители
* POST /Tour/DeleteTour - За вписани потребители
* GET/POST /Tour/CreateTour - За вписани потребители
* GET /Tour/Map - За визуализация на картата

LandmarkController MVC Endpoints:

* GET /Landmark/Index - Admin
* GET/POST /Landmark/Create - Admin
* GET /Landmark/Map – Admin

**Използвани технологии:**

* ASP.NET Core MVC - за server-side логика
* Entity Framework Core - за ORM и достъп до базата данни
* SQL Server - за съхранение на данните
* Leaflet.js - за интерактивна карта
* JavaScript/jQuery - за клиентска функционалност
* Bootstrap - за дизайн

**Основни предизвикателства при реализацията:**

* Интегриране на картовата функционалност с Leaflet.js
* Осъществяване на комуникацията между клиента и сървъра
* Визуализация на маршрути между множество точки на картата

**Практическо приложение:**

* Туристически агенции
* Образователни институции за планиране на екскурзии
* Туристически информационни центрове

**Възможности за развитие:**

* Добавяне на оптимизация на маршрути
* Интегриране на времеви графици
* Добавяне на социални функции (споделяне, коментари)
* Интеграция с външни API за туристическа информация
* Добавяне на статистики и анализи на посещаемостта

Публичен достъп до приложението в github хранилището:

<https://github.com/borofx/TripManager-final>