# **Platforma do organizacji podróży i wyjazdów**

**Imię i nazwisko:** Beytullah Sami Yorulmaz  
 **Nr albumu:** 58509  
 **Przedmiot:** Wzorzec MVC w tworzeniu aplikacji internetowych (L1)

## **1. Wstęp**

Celem projektu było stworzenie aplikacji internetowej służącej do planowania podróży i wyjazdów. Użytkownicy mogą w wygodny sposób zarządzać swoimi planami podróży – dodawać miejsca docelowe, środki transportu, hotele oraz codzienne aktywności. Projekt został zrealizowany zgodnie z założeniami zadania nr 11, a także opiera się na architekturze MVC, która zapewnia przejrzystość kodu i łatwość jego rozbudowy.

## **2. Funkcjonalności aplikacji**

* Tworzenie nowych planów podróży – z możliwością podania miejsca, zakwaterowania, środka transportu, dat, notatek.
* Możliwość przypisania do planu codziennych aktywności i przystanków.
* Edycja oraz usuwanie istniejących planów.
* Przegląd zaplanowanych wyjazdów w sposób czytelny i uporządkowany.
* System logowania i rejestracji – każdy użytkownik widzi tylko swoje plany.

## **3. Technologie użyte w projekcie**

### **Backend:**

* **Node.js + Express** – logika aplikacji i serwer
* **MongoDB + Mongoose** – baza danych oraz modelowanie danych
* **Passport.js** – autoryzacja i sesje użytkownika

### **Frontend:**

* **EJS** – szablony HTML generowane po stronie serwera
* **CSS** – własnoręcznie przygotowana warstwa wizualna (zoptymalizowana pod względem czytelności i wygody)
* **Express Flash & Method Override** – pomocnicze narzędzia do komunikatów i obsługi metod PUT/DELETE

## **4. Model danych**

Dane są przechowywane w MongoDB. Każdy plan zawiera:

* Referencję do użytkownika
* Tytuł, miejsce docelowe, zakwaterowanie, transport
* Zakres dat (od – do)
* Notatki i lista przystanków (z lokalizacją i aktywnościami)

Dzięki wykorzystaniu Mongoose, dane są walidowane zgodnie z określonym schematem, co zwiększa spójność i bezpieczeństwo danych.

## **5. Zastosowanie wzorca MVC**

Projekt został zorganizowany zgodnie z podejściem **Model-View-Controller (MVC)**:

* **Model:** Pliki w folderze /models, np. Plan.js – struktura danych i logika bazy
* **View:** Pliki EJS w folderze /views, np. plans/new.ejs, plans/edit.ejs
* **Controller (Route):** Folder /routes, np. plans.js – logika aplikacji i przetwarzanie danych

Takie podejście pozwala na łatwe rozwijanie aplikacji i utrzymanie przejrzystości kodu.

## **6. Użytkowanie aplikacji**

Po zalogowaniu, użytkownik ma dostęp do swoich planów. Może:

* Tworzyć nowy plan podróży
* Edytować istniejące plany
* Przeglądać szczegóły każdego planu
* Dodać szczegóły transportu i hotelu
* Zapisać plan dzienny z listą miejsc i aktywności

## **7. Wykorzystanie MongoDB Compass**

Do przeglądania i zarządzania danymi w bazie MongoDB, wykorzystano aplikację **MongoDB Compass**, zainstalowaną lokalnie. Dzięki niej mogłem kontrolować dane, sprawdzać struktury dokumentów i w razie potrzeby je modyfikować bezpośrednio.

## **8. Wnioski i podsumowanie**

Zrealizowany projekt spełnia wszystkie wymagania funkcjonalne. Aplikacja działa stabilnie, jest estetyczna, intuicyjna i gotowa do dalszej rozbudowy – np. o udostępnianie planów innym użytkownikom lub integrację z zewnętrznymi API (np. Google Maps, systemy rezerwacyjne).

Projekt stanowi również praktyczne zastosowanie wzorca MVC oraz umożliwia rozwijanie umiejętności związanych z pracą z bazami danych, autoryzacją, szablonami i RESTowym podejściem w tworzeniu aplikacji internetowych.

📁 **Do raportu dołączono folder z zrzutami ekranu przedstawiającymi działanie aplikacji.**