

OSTBAYERISCHE TECHNISCHE HOCHSCHULE
AMBERG-WEIDEN

Fachkonzept
TravelPlanner

Teamname: **TeamRot**

Teammitglieder:

- Thomas Antony – `t.antony@oth-aw.de`
- Himanshu Mohan Nikam – `h.nikam@oth-aw.de`
- Djatsa Kantsa Valdo Rabelais – `v.djatsa-kantsa@oth-aw.de`

Abgabe: 01. Mai 2025

Semester: Sommersemester 2025

Modul: BigData, Cloud and NoSQL (BCN)

Betreuer: Christopher Neumann

I. Einleitung

TravelPlanner ist eine Webanwendung, die es registrierten Nutzern ermöglicht, Sehenswürdigkeiten in ausgewählten Großstädten zu entdecken und ihre Reise zu planen. Die Plattform bietet eine einfache Benutzeroberfläche mit Funktionen wie Benutzerregistrierung, Login, Städtenavigation und einer Suchfunktion.

A. Motivation

Reisen ist für viele ein zentrales Hobby. Oft fehlt es aber an kompakten Tools zur strukturierten Planung von Städtereisen. TravelPlanner bietet eine schlanke Lösung für Nutzer, die gezielt attraktive Orte in den beliebtesten Städten finden möchten.

B. Alleinstellungsmerkmal

TravelPlanner konzentriert sich bewusst auf 5 große, internationale Städte, statt eine überladene Plattform zu sein. Hauptmerkmale:

- Minimalistisches, schnelles Interface
- Fokus auf **Inspiration und Planung**, nicht auf Buchung
- Klare, einfache Nutzerführung

II. Verwandte Arbeiten

Im Vergleich zu Plattformen wie *TripAdvisor* oder *Google Trips*, die komplex und oft werbelastig sind, bietet TravelPlanner eine fokussierte Alternative für gezielte Reisevorbereitung.

III. Anforderungen

A. MUSS-Anforderungen (Minimum Viable Product – MVP)

1. Registrieren

Als neuer Nutzer möchte ich mich registrieren, um später persönliche Funktionen nutzen zu können.

Akzeptanzkriterien: Formular mit Username und Passwort, Username muss eindeutig sein, Passwort wird sicher gespeichert (Hashing).

2. Anmelden

Als registrierter Nutzer möchte ich mich einloggen können.

Akzeptanzkriterien: Formular für Username und Passwort, Prüfung und Rückmeldung bei falschen Daten.

3. Städte durchsuchen

Als Nutzer möchte ich eine Liste der 5 Städte sehen können.

4. Sehenswürdigkeiten anzeigen

Als Nutzer möchte ich in jeder Stadt die wichtigsten Attraktionen angezeigt bekommen.

B. SOLL-Anforderungen

1. Attraktionen speichern

Als eingeloggter Nutzer möchte ich Attraktionen zu meinen Favoriten hinzufügen können.

Akzeptanzkriterien: Button zum Speichern von Attraktionen, Gespeicherte Attraktionen werden mit Nutzer verknüpft.

2. Gespeicherte Orte anzeigen

Als Nutzer möchte ich eine Übersicht meiner gespeicherten Attraktionen abrufen können.

3. Stadt-Suchfunktion

Als Nutzer möchte ich über eine Suchleiste nach einer Stadt suchen können.

4. Abmelden

Als Nutzer möchte ich mich sicher abmelden können.

C. OPTIONAL-Anforderungen

1. Attraktion aus Favoriten entfernen

Als Nutzer möchte ich gespeicherte Attraktionen wieder löschen können.

2. Profilseite

Als Nutzer möchte ich meine Aktivitäten in einer Profilübersicht sehen können.

3. Responsives Design

Die Web-App soll auf Mobilgeräten gut nutzbar sein.

IV. Technisches Grobkonzept

A. Datenbank – MongoDB

Speicherung von:

- Benutzerdaten (Username, Passwort-Hash)
- (Optional) Gespeicherten Attraktionen der Nutzer

B. Backend – Node.js + Express

Bereitstellung einer **REST API** für:

- Registrierung
- Login
- Abrufen von Städten und Attraktionen
- (Optional) Speichern und Abrufen von Favoriten

C. Frontend – HTML, CSS, JavaScript

Grundlegendes Web-Interface mit späterer Option zur Nutzung von **React.js**.

D. Hosting – AWS

- **EC2 Instanz** für Serverbetrieb
- (Optional) **AWS S3** für statisches Hosting der Webseite

V. Akzeptanzkriterien Übersicht

Funktion	Akzeptanzkriterium
Registrierung	Einzigartiger Username, Passwort sicher gespeichert
Login	Erfolgreiche Anmeldung oder Fehlermeldung
Stadtübersicht	5 Städte werden angezeigt
Attraktionen anzeigen	Pro Stadt 5–10 Sehenswürdigkeiten sichtbar
(SOLL) Attraktion speichern	Button vorhanden, Favoritenliste speichert Auswahl
(SOLL) Gespeicherte Orte anzeigen	Persönliche Favoriten werden angezeigt
(SOLL) Stadt suchen	Eingabe filtert Städte dynamisch

VI. Zielsetzung

- **Benutzerfreundlichkeit** durch einfache Navigation
- **Sicherheit** durch sichere Passwortspeicherung
- **Skalierbarkeit** für spätere Funktionen
- **Mobile Optimierung** als optionale Erweiterung