TripGenie：畅游济南旅行规划助手项目代码说明文档

## 一、项目简介

### 1.1 项目背景

在当今快节奏的生活中，人们渴望轻松规划个性化旅行。随着济南旅游业的蓬勃发展，现有的推荐系统已难以满足游客对个性化、精细化旅游体验的需求，亟待进一步优化升级。同时我们看到agent智能个性化与精细化城市旅行推荐系统的可行性。本系统旨在为用户提供高效、精准的济南旅行线路，助力用户快速规划济南市理想旅程，享受无忧旅行体验。

### 1.2 项目功能概述

1.用户需求获取：初次对话中获取用户需求，3-5轮对话内完成关键信息收集（如是否携带孩子等），为后续推荐提供数据支持。

2.个性化一站式行程规划：整合酒店、门票、交通信息，简化用户查询流程。

3.个性化行程生成：根据用户条件（是否带小孩、预算、时间等）自动生成2~3个最优行程方案。并且进行优化路径重叠与边际时间，提升行程效率。

4.信息获取与存储：采集携程、大众点评、百度地图等平台的景点、住宿、美食、交通等数据。景点存储详细信息和图片链接，如营业时间和门票等。对于酒店和美食数据则收集了位置、人均价格、星级等信息。

5.济南文化特色融合：推介舜耕文化、李清照故居等文化地标与活动。串联趵突泉、大明湖等经典文化地标，结合节庆活动设计专属文化行程。

6.天气感知提醒：提前提醒天气变化，智能建议备雨具、调整行程等贴心提示。

7.智能交互与地图展示：以列表形式清晰展示景点、酒店、美食信息，支持用户。地图可视化：提供景点地图标注、位置关系展示，附带参考链接。

8.行程导出与用户反馈：PDF导出功能能一键生成并导出济南行程规划文档。支持用户反馈，对推荐质量进行评价，用于优化智能体服务质量。

### 1.3 技术栈

本项目采用了以下技术栈：

后端技术：Python，Flask

前端技术：HTML5，CSS3，JavaScript (JS)

数据库：MySQL

## 二、项目结构说明

├── model/ #后端代码

│ ├── data/ #使用数据

│ ├── output/ #输出地图路径

│ ├── utils/ #后端方法，如prompts等

│ ├── itinera.py #模型实现文件

│ ├── itinera\_en.py #英文模型实现文件

│ ├── search.py #检索文件

│ └── spatial.py #空间优化文件

├── spider/ #爬虫文件

├── static/ #静态路径

│ ├── css/ # CSS代码

│ ├── js/ #JS代码

│ ├── history/ #实现历史文件

│ ├── bus\_stations/ #交通文件

│ ├── bg.jpg #背景图片文件

│ ├── logo.jpg #图片文件

├── templates/ #前端代码

│ ├── login.html # 登录页面

│ ├── register.html #注册页面

│ ├── page.html #主页面

│ └── page\_en.html #英文代码页面

├── upload\_images/ #pdf导出时截图文件

├── main.py #执行文件

├── requirements.txt #配置文件

└── README.md # 项目说明文档

## 模块说明

### 3.1 数据预处理文件说明

Spider/spider.py：

自动化爬取携程网站上济南市景点的信息，包括景点名称、票价、特色、分类、评分、热度、所属区域、地址位置及详情页链接，并将这些信息保存到本地。

Spider/climate.py:

自动化爬取中国天气网站上济南市天气信息，包括最高温、最低温、风向、天气图标等，并将这些信息保存到本地。

Spider/hotel\_spider.py：

自动化爬取携程网站上济南市酒店信息，包括名称、位置、星级等，并将这些信息保存到本地。

Spider/food\_spider.py：

自动化爬取大众点评网站上济南市美食信息，包括名称、位置、人均价格等，并将这些信息保存到本地。

Spider/traffic\_spider.py：

自动化爬取济南公交网站上济南市公交信息，包括公交线路，线路站点、票价等，并将这些信息保存到本地。

Model/data/pre.py：

进行坐标转换，将标准经纬度坐标转换为墨卡托坐标便于显示在地图上。

Model/data/reproduce.py：

用于生成词嵌入向量。它读取包含景点数据，调用OpenAI的嵌入模型将每条描述转换为向量，并将这些向量保存为.npy文件。这些嵌入向量可用于后续的相似性计算和推荐系统。

### 3.2 后端主要模块说明

Model/utils/all\_prompts.py：

用于训练智能体的prompt，生成中文版行程规划提示。提供了多种函数，用于根据用户输入生成系统提示、检查行程反转、获取行程时间、确定起始点以及处理用户需求。这些提示被设计为与语言模型交互，以生成符合用户需求的行程规划。

Model/utils/funcs.py：

工具函数库，为行程规划系统提供多种辅助功能。它包括数据加载、嵌入向量处理、模糊匹配、距离计算、列表操作等功能。

Model/utils/proxy\_call.py：

代理调用模块，用于封装对OpenAI API的请求。它通过OpenAI客户端初始化连接，并提供了chat和stream\_chat方法用于发送对话消息并获取响应，还提供了embedding方法用于生成文本嵌入向量。

Model/itinera.py：

模型文件，是行程规划的核心模块，负责根据用户需求生成旅行计划。通过解析用户输入、调用搜索引擎和空间优化模块，生成最优的旅行路线，并结合地图可视化展示结果。此外，还支持动态调整行程顺序，确保行程的高效性和个性化。

Model/itinera\_en.py：

英文模型文件，是行程规划的核心模块。和itinera.py功能相同。

Model/search.py：

检索模块，用于根据用户输入的描述查询相关的景点信息。它通过计算输入描述与预存储的景点嵌入向量之间的余弦相似度，快速检索出最相关的景点。支持正向和负向描述的查询，能够排除不符合用户需求的景点。

Model/spatial.py：

空间优化模块，用于处理旅行点的空间分布和路径规划。通过聚类算法将旅行点分组，确保每个组内的点在地理上相对集中。同时，利用旅行商问题（TSP）算法优化组内和组间的访问顺序，减少旅行距离。

Main.py：

执行文件，基于 Flask 框架搭建后端服务。它处理前端请求，调用行程规划模块生成旅行计划，并提供地图可视化、PDF 导出等功能。此外，还支持用户登录、注册、保存反馈和查询公交信息等操作，是前后端交互的核心文件。

### 3.3 前端主要模块说明

Static/css/headstyle.css：

前端样式文件，定义了全局样式和组件的外观。包括主题颜色、字体、按钮、弹窗、输入框等的样式，以及深色模式的支持。此外，还为左侧边栏、主内容区、右侧搜索栏等页面布局提供了详细的样式定义。

Static/css/headstyle\_weather.css：

前端天气信息展示样式文件，定义了全局样式和组件的外观。实现横向滚动天气卡片展示，包含滚动条美化及按钮样式，提升界面布局与交互美观度。

Static/js/function.js：

前端交互脚本文件，负责实现用户界面的各种功能。它包括主题切换、语言切换、消息发送、行程规划请求、天气信息获取、历史记录显示等功能。此外，还支持用户反馈提交、截图功能以及动态更新界面内容，为用户提供流畅的交互体验。

Static/js/function\_weather.js：

前端天气信息展示交互脚本文件，实现天气页面的切换、CSV数据读取、天气卡片展示及温度折线图绘制功能，展示未来多天的天气信息。

Templates/login.html：

登录页面，提供用户登录功能。它包含一个登录表单，用户可以输入用户名和密码进行登录。页面使用了简洁的样式设计，通过JavaScript调用后端API进行身份验证，并根据返回结果提示用户登录成功或失败。

Templates/register.html：

注册页面，提供用户注册功能。它包含一个注册表单，用户可以输入用户名、密码和确认密码进行注册。页面设计简洁，通过JavaScript调用后端API完成注册逻辑，并根据返回结果提示用户注册成功或失败，支持跳转到登录页面。

Templates/page.html：

项目的主页面，集成了聊天功能、主题切换、语言切换、历史记录查看、天气信息查询和公共交通查询等功能。页面设计简洁，通过JavaScript动态加载数据并实现交互逻辑，为用户提供丰富的功能体验。

Templates/page\_en.html：

项目的英文页面，和page.html功能一致。