**河北传媒学院**

**课程报告**

学 院 信息技术与文化管理学院

专业班级

学生学号

学生姓名

基于Spark的河北省旅游景区推荐系统的设计与实现

# 课程背景

随着国家不断发展和进步，人们物质水平也在不断提高。与此同时，人们也更加趋于追逐精神生活的富足。满足自身精神层面富足的方式有很多，而大多数人会选择旅游的方式来进行精神层面的放松。 旅游可以适当满足自身的情感需求，放松身心，使精神层面解除紧绷的状态。但是，在如今快节奏的时代里，人们的假期大都是有限并短暂的，并不足以让人们充分了解景区信息，制定自身旅游计划，充分满足自身需求。

同时，河北省作为一个拥有丰富自然和人文景观资源的地区，也吸引了大量的游客。然而，由于游客数量庞大和景区众多，游客如何选择合适的景区成为一个难题。因此，设计并实现一套基于Spark的河北省旅游景区推荐系统，将能够提供游客个性化的推荐服务，帮助游客更好地规划旅游行程，提升旅游的体验。

Spark技术具有快速处理大规模数据、支持多种语言、具有灵活性和拓展性、并且能应用于实时数据流场景的特点。因此通过Spark技术，可以实现对旅游数据的实时监测。基于此，本设计旨在通过数据爬取、数据存储和Spark技术实现一个基于Spark的便捷的、高效的河北热门旅游景区推荐系统。

# 核心技术

# 2.1 Hadoop

Hadoop是一个开源的大数据处理平台，由Apache基金会开发和维护。其核心技术包括Hadoop Distributed File System (HDFS) 和 Yet Another Resource Negotiator (YARN)。HDFS是一种分布式文件系统，设计用于在商用硬件上运行，能够处理海量数据并实现高容错性和高可用性。而YARN是Hadoop的资源管理和作业调度系统，负责管理集群中的计算资源，并根据应用程序的需求进行分配。

# 2.2 Spark

Apache Spark是一个快速、通用、可扩展的分布式计算框架，特别适合大规模数据处理和实时分析。Spark的核心技术包括Resilient Distributed Datasets (RDDs)、In-Memory Computing以及Spark Core和一系列库。RDDs是Spark的基本数据结构，提供容错机制；In-Memory Computing利用内存进行数据处理，提高速度；Spark Core是基础，包含任务调度、内存管理等功能；而Spark SQL、MLlib、GraphX和Spark Streaming等库为各种数据处理和分析任务提供支持。

# 2.3 Hive

Apache Hive是一个基于Hadoop的数据仓库工具，通过SQL-like查询语言（HiveQL）处理存储在Hadoop上的大规模数据。Hive的核心技术包括HiveQL、Metastore以及MapReduce Integration。HiveQL允许使用熟悉的查询语法访问和分析大数据；Metastore存储表和列的结构信息以及数据位置信息；而MapReduce Integration将HiveQL查询转换为一系列MapReduce任务，在Hadoop集群上执行。

# 2.4 Vue

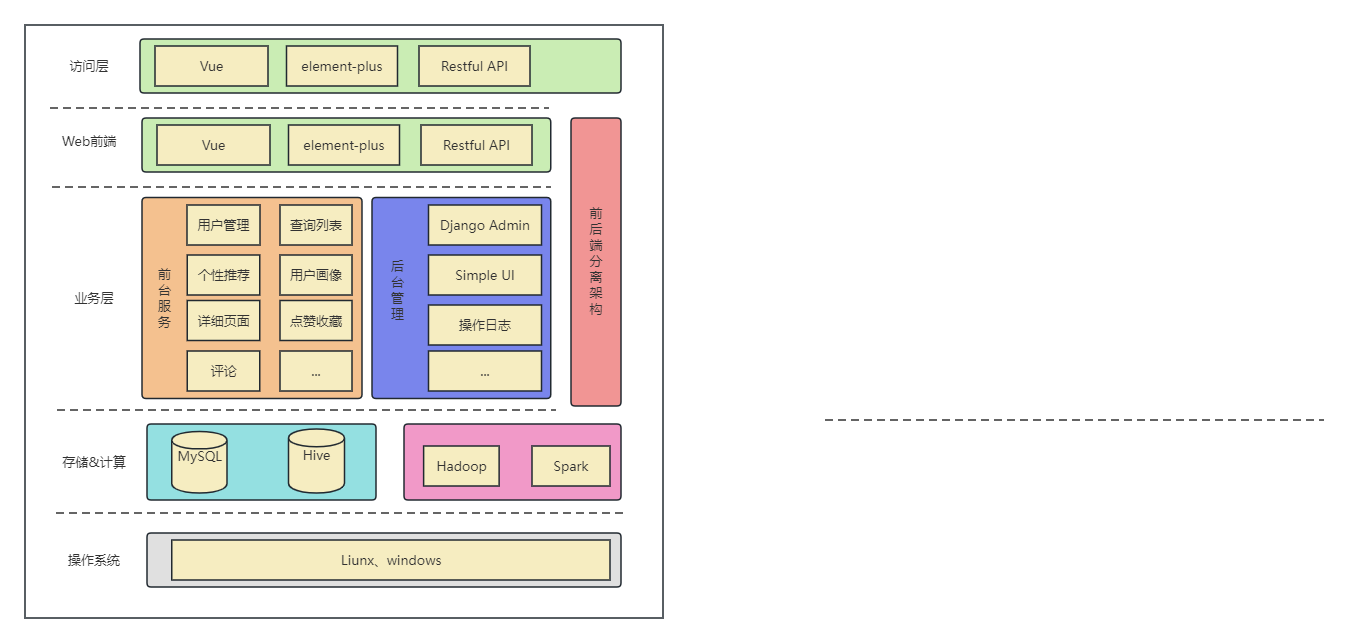
Vue.js是一个流行的前端JavaScript框架，用于构建用户界面和单页应用。Vue的核心技术特点包括Reactive Components、Virtual DOM以及Templating and Directives。Vue通过组件化的方式组织UI，每个组件都是独立且反应式的；使用虚拟DOM优化渲染性能，仅更新实际发生变化的部分DOM；提供易于理解的模板语法和一系列指令，方便声明式地定义组件的结构和行为。

# 2.5 Django

Django是一个高级的、Python编写的Web框架，遵循MVC（Model-View-Controller）架构模式。Django的核心技术特性包括MTV Architecture、ORM (Object-Relational Mapping) 以及“Batteries Included”理念。Django采用Model-Template-View（MTV）架构，其中Model处理数据，Template负责呈现，View处理业务逻辑；内置强大的ORM系统，允许以Python对象的方式操作数据库；秉承“包含电池”的理念，提供了丰富的内置功能，如用户认证、权限管理、站点地图、RSS/Atom feeds生成等，简化Web应用的开发过程。

# 系统设计

# 3.1 系统架构



系统总架构图

本系统由四个主要层次组成：访问层、Web前端、业务服务以及存储与计算。

在最上一层是访问层，它包括Vue（一个流行的JavaScript框架），element-plus（一个UI组件库）和Restful API（一种网络应用接口）。这些工具提供了用户界面，并通过API与其他部分交互。

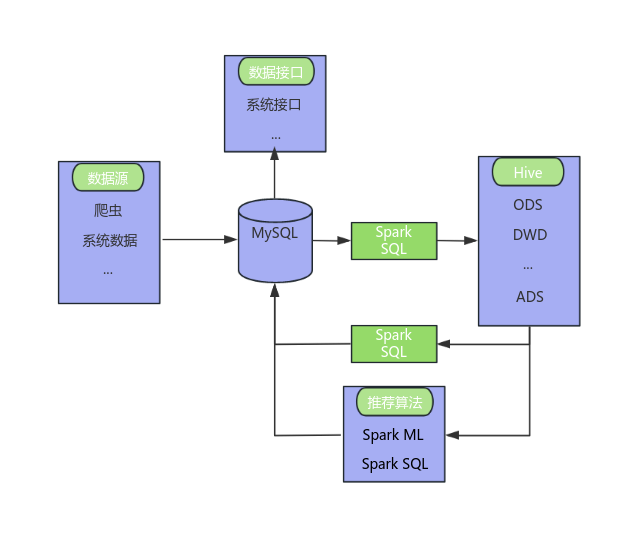
接下来的是Web前端，同样使用了Vue和element-plus作为开发工具。此外还引入了一个名为Django Admin的后台管理平台，这是Python Web框架Django的一部分，可以用来管理和维护网站内容。

再往下就是业务服务层，这部分包含了前后端分离的架构设计。在这个层级中，有一个个人化的推荐模块，允许对用户的兴趣爱好做出预测并提供个性化的建议；还有一个详细的页面展示功能，可能包含个人信息或者工作信息等详情；同时还有评论和点赞收藏等功能来增强用户体验。

最后，在底部是存储与计算层，这里采用了MySQL数据库管理系统和Hive数据仓库技术来进行数据处理和分析。而更深入的数据挖掘则依赖于Apache Spark的大规模分布式运算能力。

整个系统运行在一个Linux或Windows操作系统之上，为所有其他组成部分提供基础支持和服务。这种多层次的设计使得该系统能够有效地满足各种需求，从简单的查询到复杂的个性化推荐都能得到良好的实现。

# 3.2 数仓架构



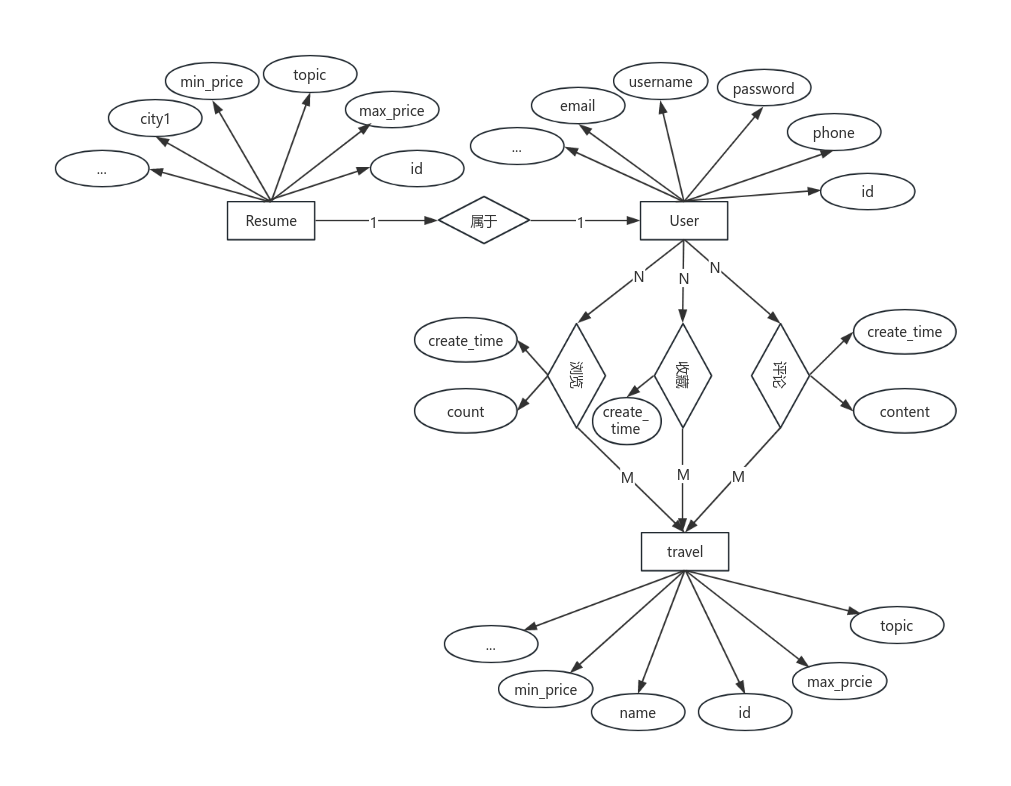
数仓架构图

在系统中，数据从多个来源收集，并通过一系列的数据处理步骤进行清洗、转换和整合。

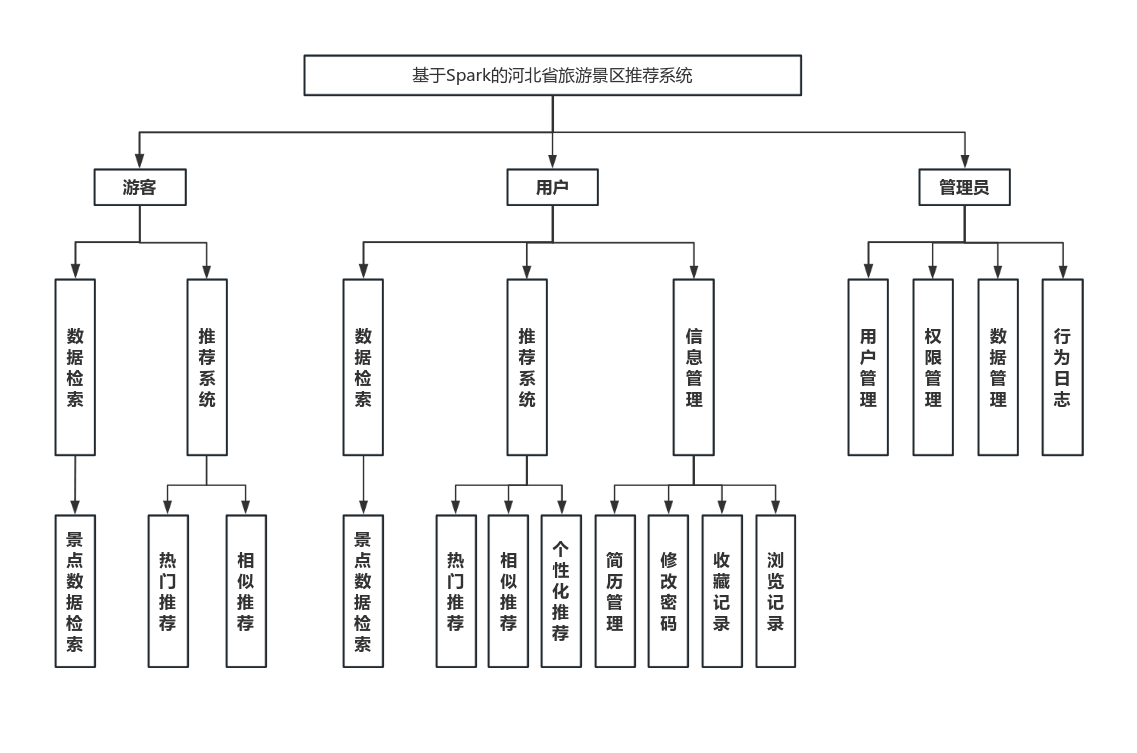
在最底层是MySQL数据库，它是整个系统的数据存储中心。这个数据库包含了爬虫抓取到的各种旅游信息以及一些基础的系统数据。这些原始数据被导入到Hive表中进一步处理。在Hive中，我们可以对数据进行复杂的查询操作，例如计算每个景点的人流量或者游客满意度等指标。我们使用Spark SQL来执行更复杂的数据分析任务。比如，可以利用Spark ML库中的机器学习算法来进行用户行为预测或兴趣偏好挖掘；也可以用SQL语言直接对大数据集进行实时查询和更新。

经过上述过程得到的结果会被用于生成个性化的旅行建议。这部分工作通常由一个叫做ADS（Advertising Decision System）的应用完成，它可以根据用户的个人信息和历史记录为他们提供最适合他们的旅行方案。

# 3.3 ER图



# 3.4系统功能设计



系统功能设计架构图

游客模块: 负责数据检索和推荐系统，旨在为游客提供定制化和相关性强的旅游资讯。

用户模块: 含有数据检索、推荐系统和服务管理等功能，面向不同的用户群体，提供多样性的信息服务。

管理员模块: 设计了一系列管理系统级的任务，如信息管理、用户管理、权限管理、数据管理和行为日志，保障系统的稳定运营。

景点模块: 针对具体的景点提供了多种服务选项，如景点数据检索、热门推荐和相似推荐，方便游客规划行程。

# 系统实现

# 4.1 爬虫设计与实现

目标网站：

驴妈妈旅游网（https://www.lvmama.com）

实现步骤：

①筛选出河北省旅游景区



②确定需要的字段

|  |  |
| --- | --- |
| 字段名 | 字段类型 |
| id | int(11) |
| name | text |
| topic | text |
| low\_price | text |
| high\_price | double |
| address | text |
| city | text |
| url | text |
| img | text |
| create\_time | datetime |

③找出分页规则

<http://s.lvmama.com/ticket/H9K310000P{page}PRO3?keyword=%E6%B2%B3%E5%8C%97&tabType=route350#list>

该URL中的page就是分页参数



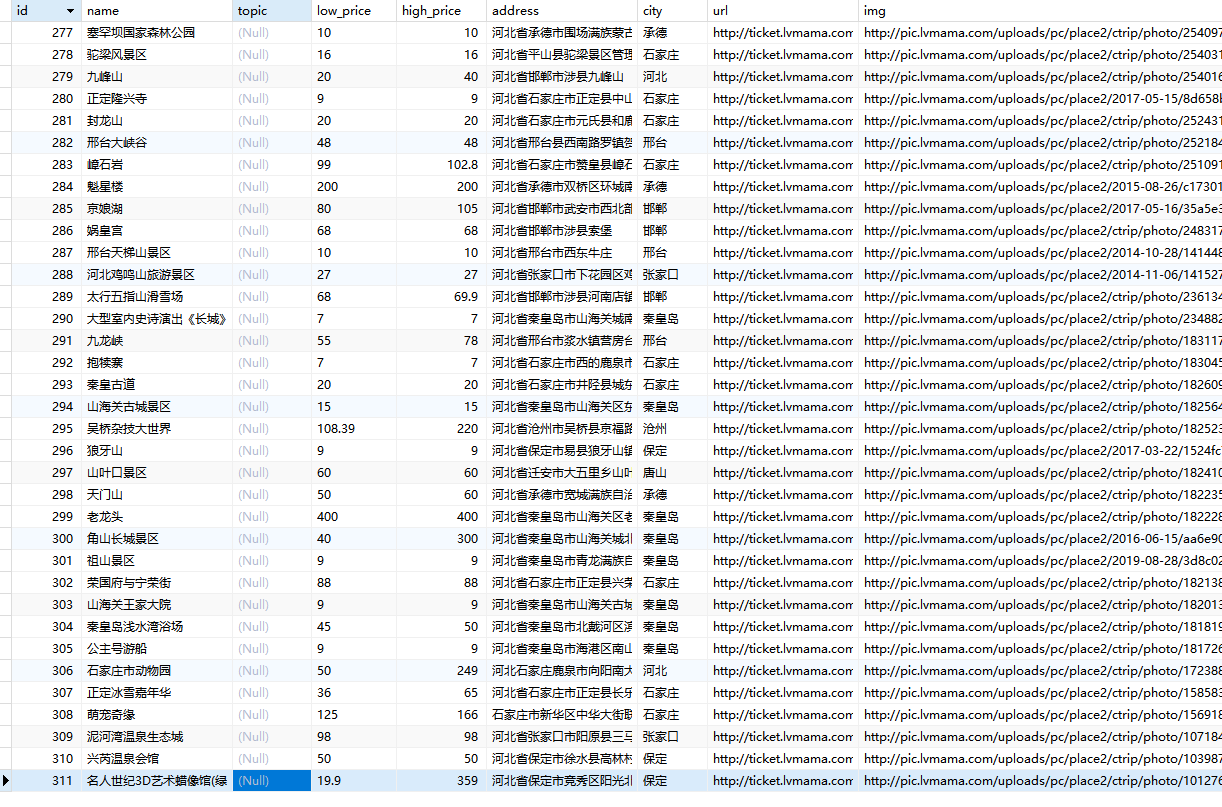
④页面爬取与解析

核心解析代码如下



⑤存入数据库

使用pymysql库将爬取的数据存入MySQL的spider数据库中

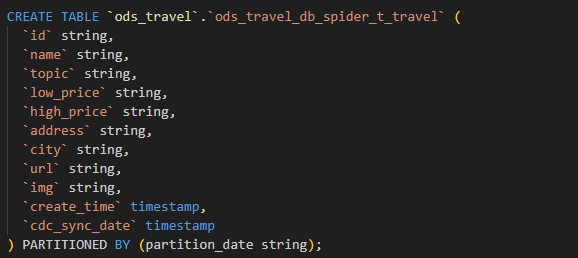


# 数据治理（ETL，数仓建模，特征提取）

①ODS层

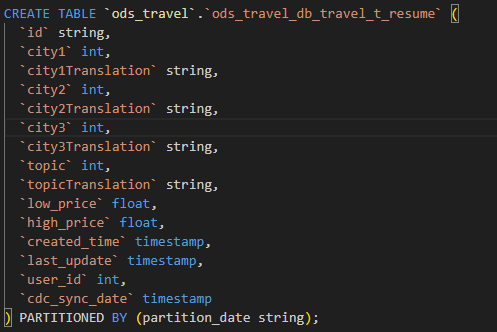
`ods\_travel`.`ods\_travel\_db\_spider\_t\_travel`

该表是为了存储增量爬取的景点数据，所以采用了分区表的形式建表，通过爬取的时间create\_time进行分区，建表语句如下



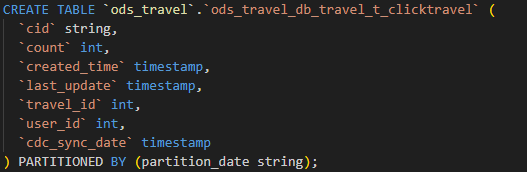
`ods\_travel`.`ods\_travel\_db\_travel\_t\_resume`

该表是为了存储系统增量的用户画像数据



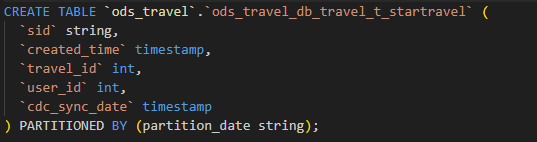
`ods\_travel`.`ods\_travel\_db\_travel\_t\_clicktravel`

该表是为了存储系统增量的用户的景点点击数据



`ods\_travel`.`ods\_travel\_db\_travel\_t\_startravel`

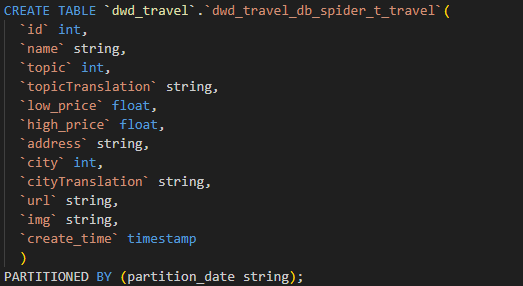
该表是为了存储系统增量的用户的景点收藏数据



②DWD层

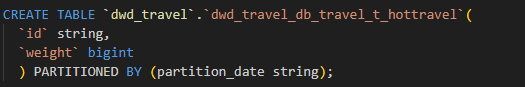
`dwd\_travel`.`dwd\_travel\_db\_spider\_t\_travel`

该表存放清洗后的景点数据



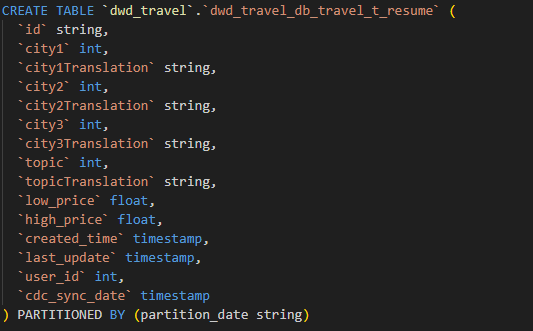
`dwd\_travel`.`dwd\_travel\_db\_travel\_t\_hottravel`

该表存放热门景点数据



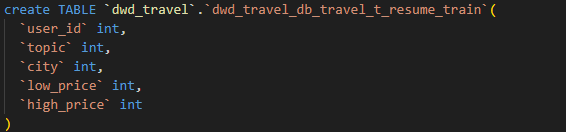
`dwd\_travel`.`dwd\_travel\_db\_travel\_t\_resume`

该表存放用户画像数据



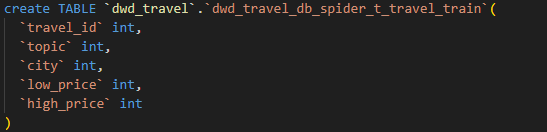
`dwd\_travel`.`dwd\_travel\_db\_travel\_t\_resume\_train`

该表存放用户画像训练集数据



`dwd\_travel`.`dwd\_travel\_db\_spider\_t\_travel\_train`

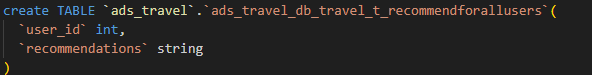
该表存放景点训练集数据



③ADS层

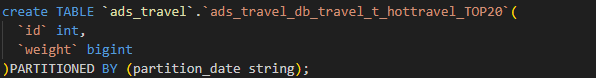
`ads\_travel`.`ads\_travel\_db\_travel\_t\_recommendforallusers`

存放推荐模型预测的推荐数据，推荐结果为json格式



`ads\_travel`.`ads\_travel\_db\_travel\_t\_hottravel\_TOP20`

存放热门景点TOP20数据



# 推荐算法

①离线推荐算法

Spark ML是Apache Spark的机器学习库。它提供了一组丰富的机器学习算法和工具，用于处理大规模数据集和分布式计算环境。使用Spark ML是为了方便在大量数据下进行机器学习，给推荐算法的实现提供技术支撑。

针对本课题，我们可以采用基于用户和景点的协同过滤算法，具体实现流程如下：

数据预处理：将景点数据和用户简历数据提取出的特征进行特征工程，例如对门票价格/期望价格进行离散化、对城市、景点类型进行编码等，得出符合推荐系统需要的数据格式。

构建用户和景点的特征向量：将预处理好的用户和景点数据转换成对应的特征向量，其中用户特征向量表示用户的各项特征，例如门票价格、景点类型等，景点特征向量表示景点的各项特征。

计算用户和景点的相似度：采用余弦相似度计算用户向量和景点向量之间的相似度，得到相似度矩阵。

对相似度矩阵进行加权：根据用户的历史行为和对不同特征的偏好，对相似度矩阵中的权重进行调整，得到加权相似度矩阵。

预测用户对景点的喜好程度：对于每个用户，根据其历史行为和加权相似度矩阵，计算用户对未看过的景点的喜好程度。

景点推荐和模型评估：根据预测得分，对每个用户推荐若干个高分景点。同时，通过对模型的评估，不断优化模型参数，提高推荐的准确性和覆盖率。

通过以上算法流程的实现，我们可以构建一个完整的基于协同过滤的景点推荐系统，为用户提供更加精准和个性化的匹配服务。

②实时推荐

用户的收藏列表 -> 查询出同样收藏该职位(排除了自己)的用户列表 -> 查询出用户列表所以用户的收藏列表(排除自己的收藏列表)

# WEB系统实现

①数据库表结构

|  |  |  |
| --- | --- | --- |
| 表名 | User | |
| 字段名 | 字段类型 | 备注 |
| id | bigint(20) | 用户ID |
| password | varchar(128) | 用户密码 |
| last\_login | datetime(6) | 最后登录时间 |
| is\_superuser | tinyint(1) | 是否超级用户 |
| username | varchar(150) | 用户名 |
| first\_name | varchar(150) | 用户名字 |
| last\_name | varchar(150) | 用户姓氏 |
| email | varchar(254) | 电子邮箱 |
| is\_staff | tinyint(1) | 是否员工 |
| is\_active | tinyint(1) | 是否激活 |
| date\_joined | datetime(6) | 加入日期 |
| name | varchar(10) | 姓名 |
| birth | date | 出生日期 |
| genderCode | int(11) | 性别代码 |
| genderTranslation | varchar(2) | 性别翻译 |
| phone | varchar(11) | 电话号码 |
| photo | varchar(100) | 照片 |
| init | tinyint(1) | 是否初始化 |
| last\_update | datetime(6) | 最后更新时间 |
| resume\_id | int(11) | 简历ID |

|  |  |  |
| --- | --- | --- |
| 表名 | Travel | |
| 字段名 | 字段类型 | 备注 |
| id | int(11) | ID |
| name | longtext | 名称 |
| topic | int(11) | 景点类型id |
| topicTranslation | longtext | 景点类型翻译 |
| low\_price | float | 最低价格 |
| high\_price | float | 最高价格 |
| address | longtext | 地址 |
| city | int(11) | 城市 |
| cityTranslation | longtext | 城市翻译 |
| url | longtext | URL |
| img | longtext | 图片 |
| create\_time | timestamp | 创建时间 |

|  |  |  |
| --- | --- | --- |
| 表名 | Resume | |
| 字段名 | 字段类型 | 备注 |
| id | bigint(20) | 用户偏好ID |
| city1 | int(11) | 期望城市1 |
| city1Translation | varchar(20) | 城市1翻译 |
| city2 | int(11) | 期望城市2 |
| city2Translation | varchar(20) | 城市2翻译 |
| city3 | int(11) | 期望城市3 |
| city3Translation | varchar(20) | 城市3翻译 |
| topic | bigint(20) | 偏好话题 |
| topicTranslation | varchar(20) | 话题翻译 |
| low\_price | double | 最低价格 |
| high\_price | double | 最高价格 |
| created\_time | datetime(6) | 创建时间 |
| last\_update | datetime(6) | 最后更新时间 |
| user\_id | bigint(20) | 用户ID |
| id | bigint(20) | 用户偏好ID |

|  |  |  |
| --- | --- | --- |
| 表名 | Recommendforallusers | |
| 字段名 | 字段类型 | 备注 |
| user\_id | int(11) | 用户ID |
| recommendations | longtext | 推荐列表 |

|  |  |  |
| --- | --- | --- |
| 表名 | hot\_top20 | |
| 字段名 | 字段类型 | 备注 |
| id | int(11) | 景点ID |
| weight | bigint(20) | 权重 |

|  |  |  |
| --- | --- | --- |
| 表名 | Click | |
| 字段名 | 字段类型 | 备注 |
| cid | bigint(20) | 统计ID |
| count | int(11) | 次数 |
| create\_time | datetime(6) | 创建时间 |
| last\_update | datetime(6) | 最后更新时间 |
| travel\_id | bigint(20) | 景点ID |
| user\_id | bigint(20) | 用户ID |

|  |  |  |
| --- | --- | --- |
| 表名 | Star | |
| 字段名 | 字段类型 | 备注 |
| sid | bigint(20) | 统计ID |
| create\_time | datetime(6) | 创建时间 |
| travel\_id | bigint(20) | 景点ID |
| user\_id | bigint(20) | 用户ID |

|  |  |  |
| --- | --- | --- |
| 表名 | Comment | |
| 字段名 | 字段类型 | 备注 |
| cid | bigint(20) | ID |
| content | longtext | 次数 |
| create\_time | datetime(6) | 创建时间 |
| travel\_id | bigint(20) | 景点ID |
| user\_id | bigint(20) | 用户ID |

②系统功能设计

用户功能

注册登录：用户在使用本系统进行景区预约、订票等操作时需要进行注册以及登录操作，以便更好地制定个人旅游规划；

景区推荐：根据调查问卷收集到的用户喜好，以及登录用户的个人信息、搜索习惯等，通过推荐算法，为登录用户提供个性化的景区推荐。添加推荐算法的主要目标是为了提高用户体验，推荐更符合用户个人需求的旅游景区；

景区浏览：用户可以查看景区的详情介绍、图片、地址、主题、评分 、评论等信息，以便帮助用户更好的了解景区；

景区筛选：用户可以根据景区主题、月份、距离远近等进行筛选，以便用户更快捷的获取符合个人预期的景区；

景区点赞和收藏：用户可以对自己感兴趣的景区进行点赞或收藏操作，以便用户制定个性化旅游计划时，可以以此作为参考；

制定个性化旅游计划：用户可根据个人点赞、浏览或者收藏的景区，制定个性化旅游规划；

个人信息管理：用户对个人基本信息的完善及后续修改。例如:用户名称、密码等；

②管理员功能

1）管理员登录：管理员通过输入用户名、密码等进入本系统后台，登录时验证账号密码，系统随机生成验证码，防止恶意登录；

2）旅游景区管理：此处可以管理通过爬虫进行数据采集的景区数据，另外可以对旅游景区进行新增修改等操作；

3）评论管理：管理员管理用户对景区评论信息；

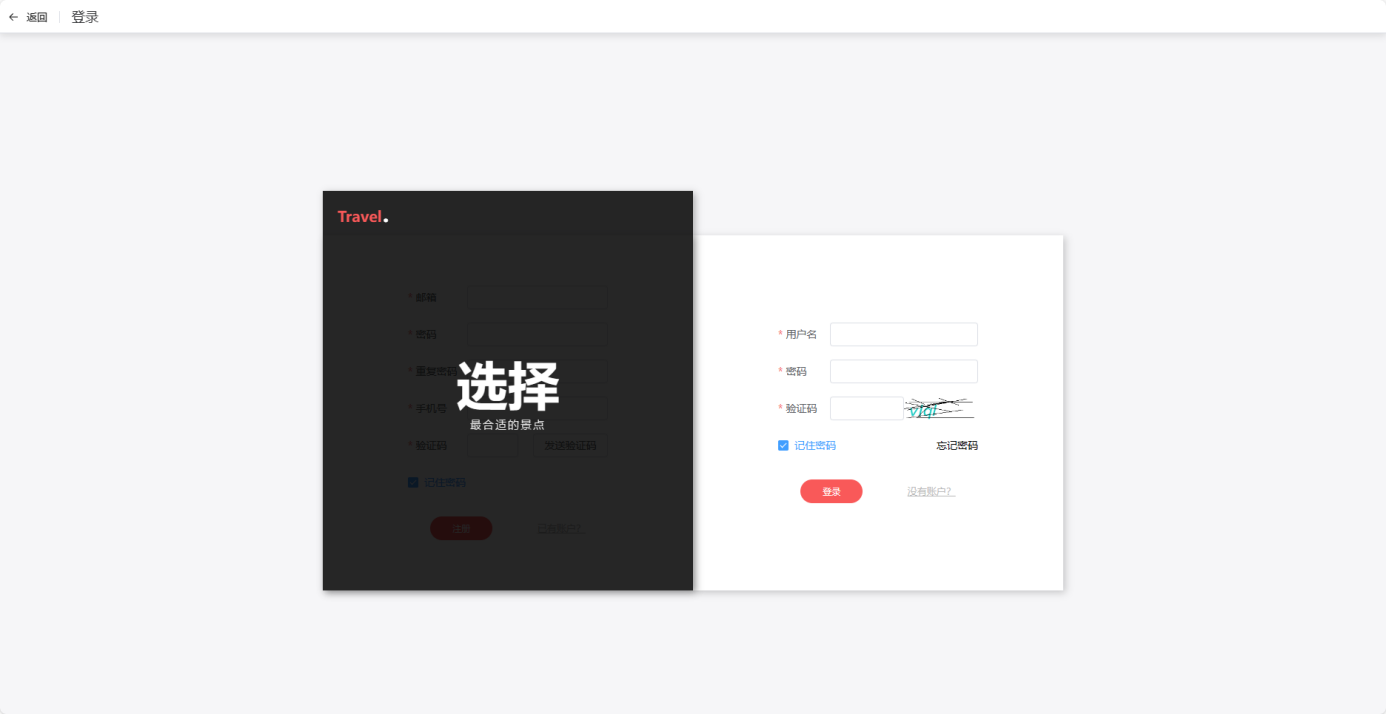
4）用户管理：管理员在此处查看所注册本系统的用户基本信息；

5）旅游计划管理：管理用户所制定的旅游计划；

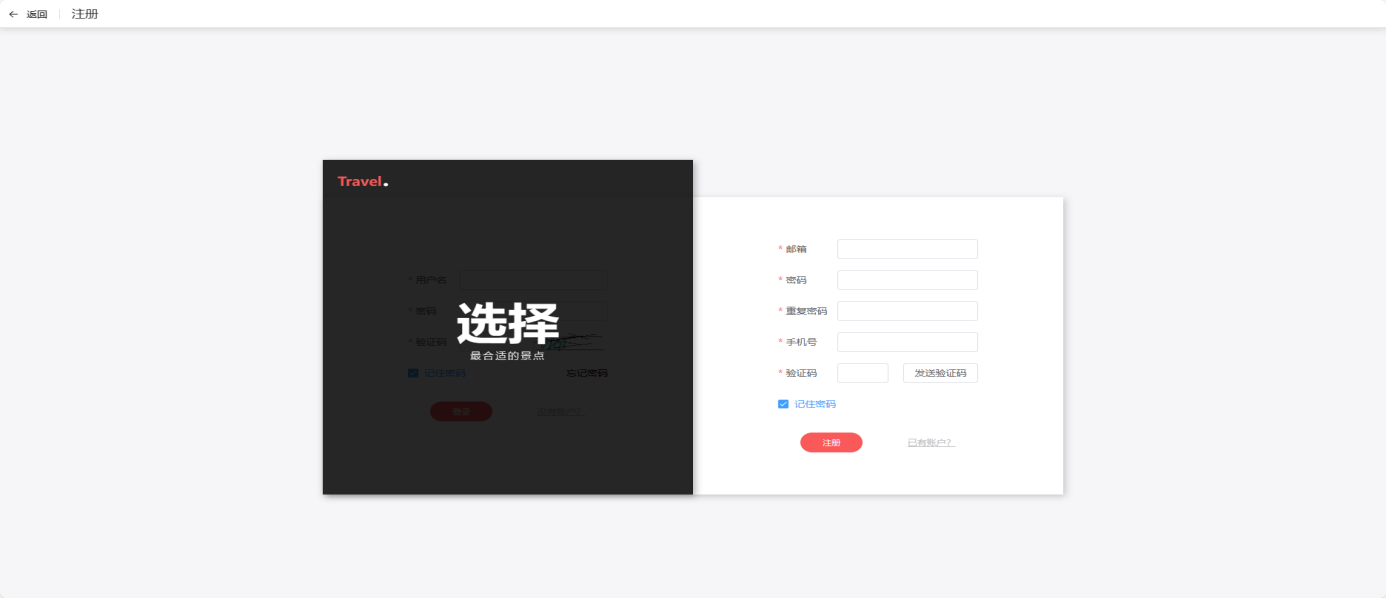
④系统展示

用户界面

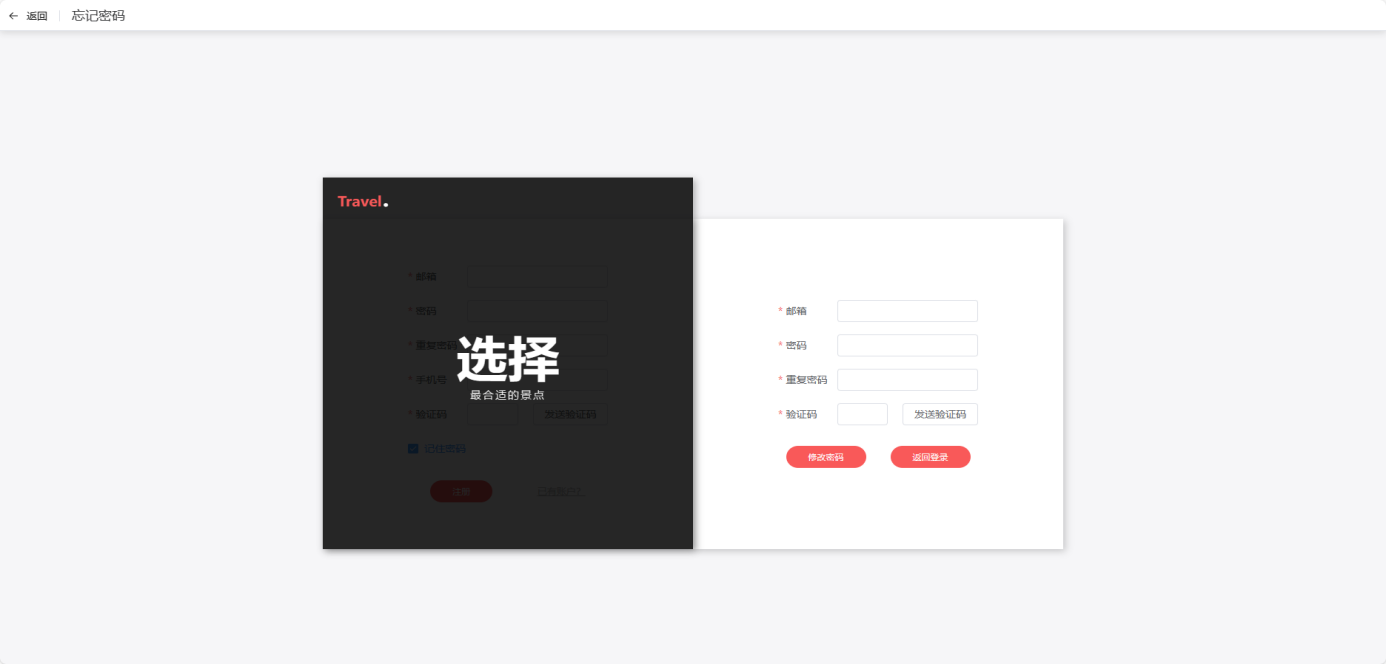
1）登录页面



1. 注册页面



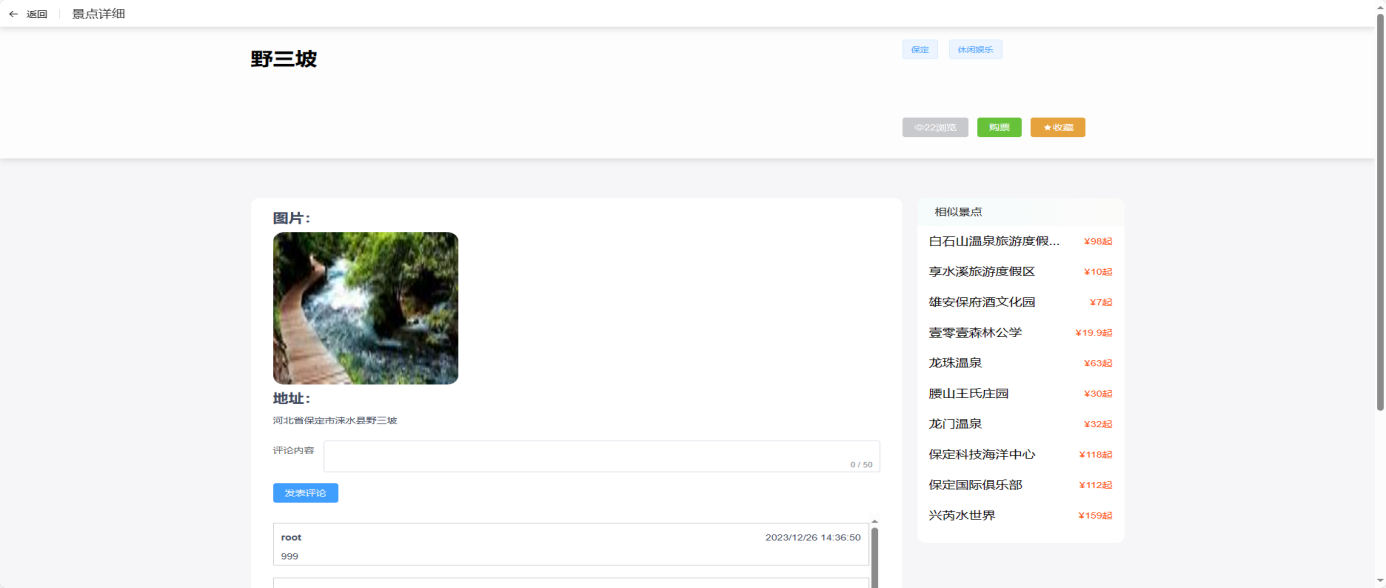
1. 修改密码



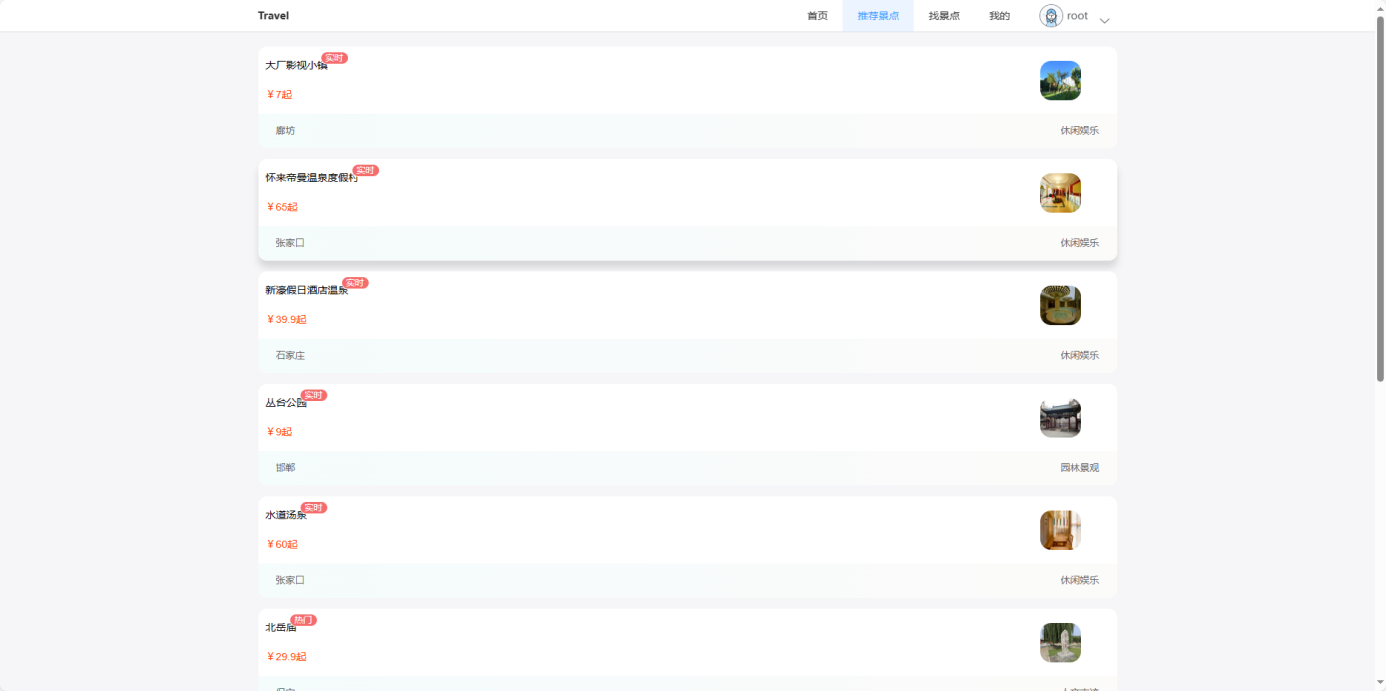
1. 系统主页

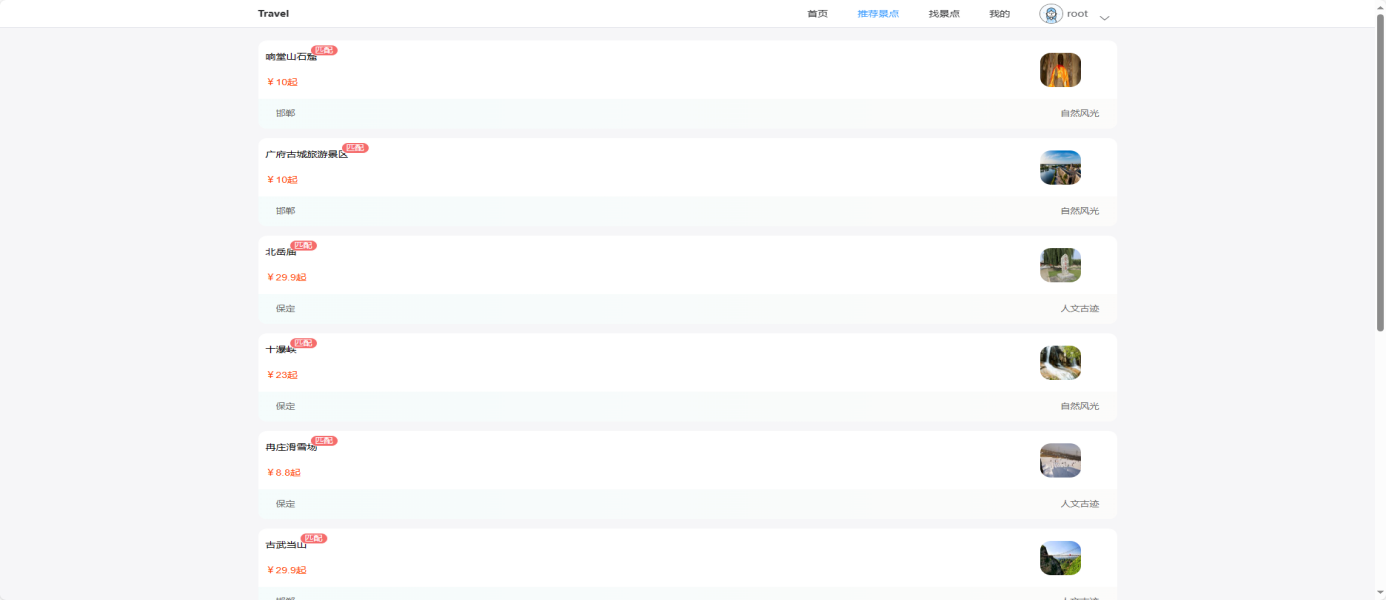


1. 详细页面

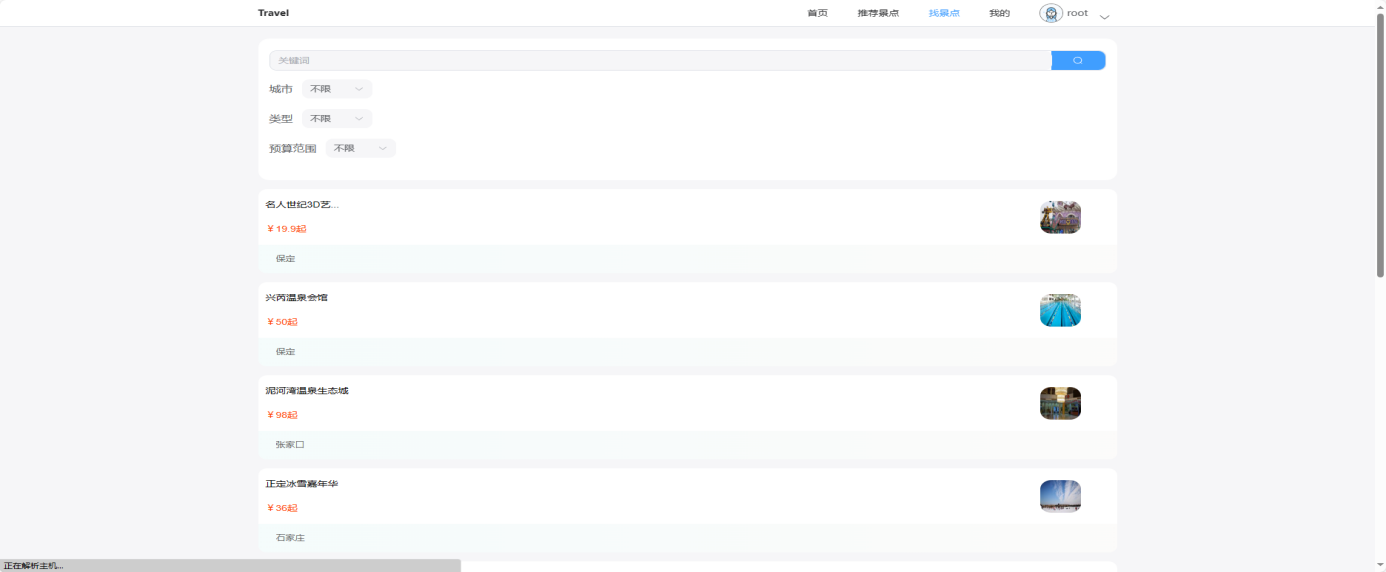


1. 推荐页面

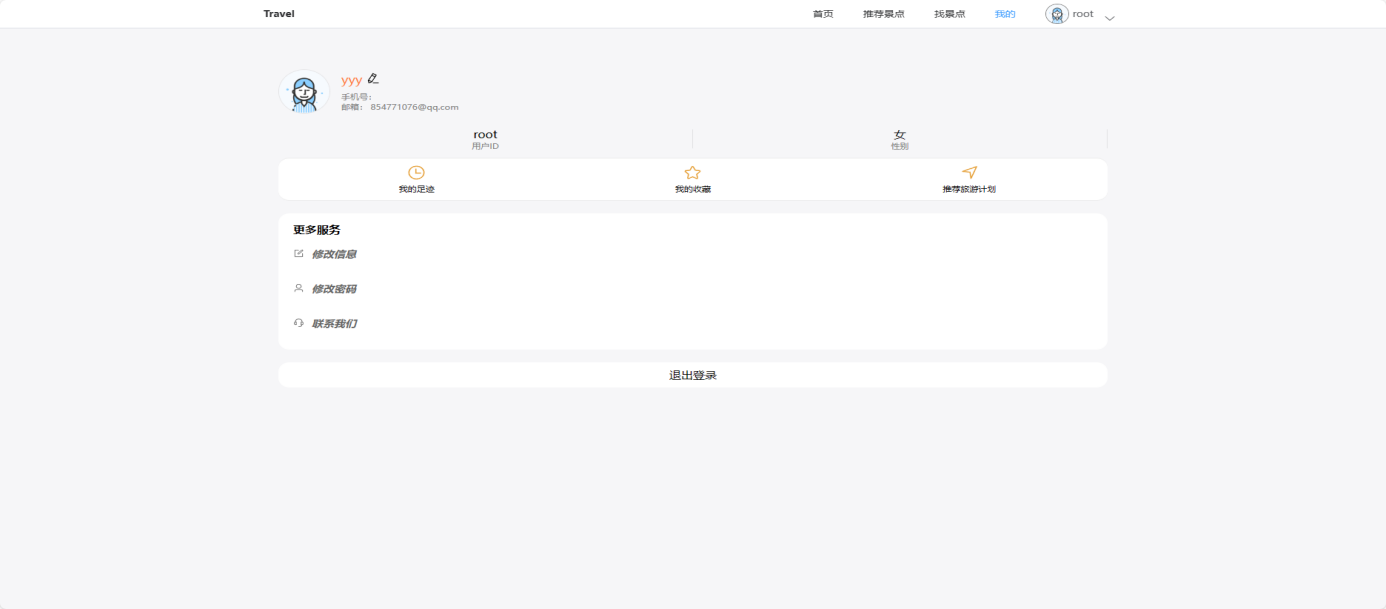




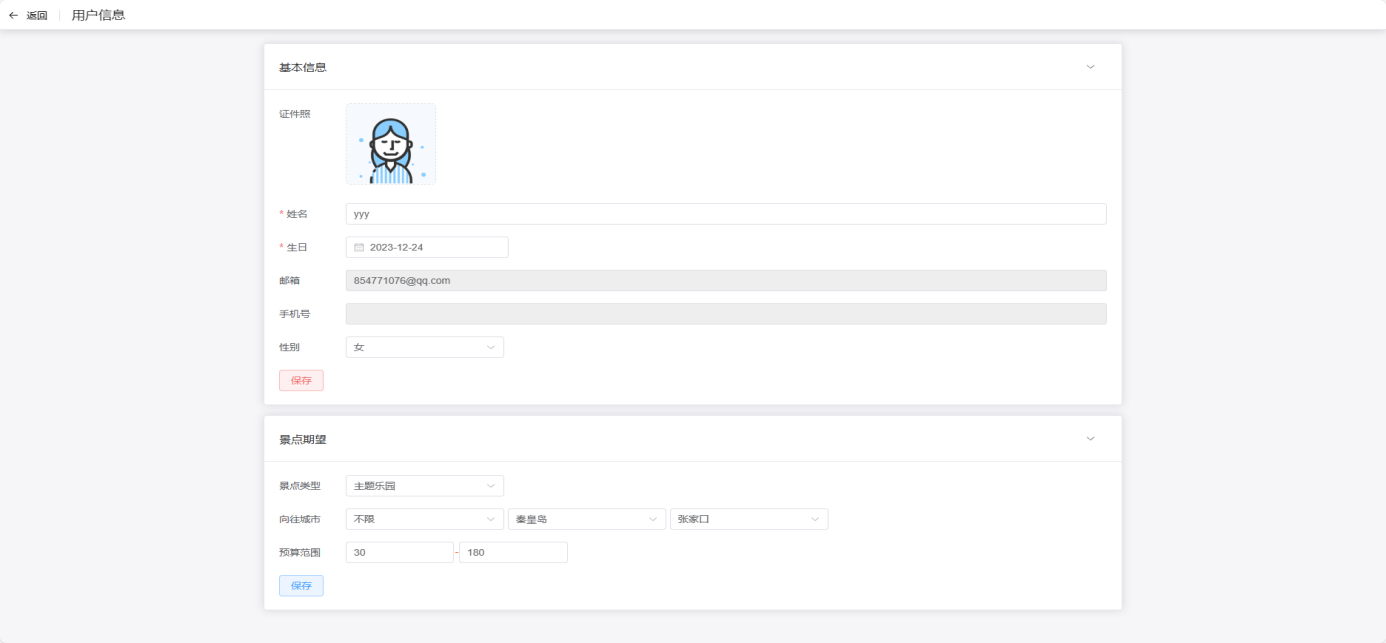
1. 查找页面



1. 我的页面



1. 修改用户信息页面



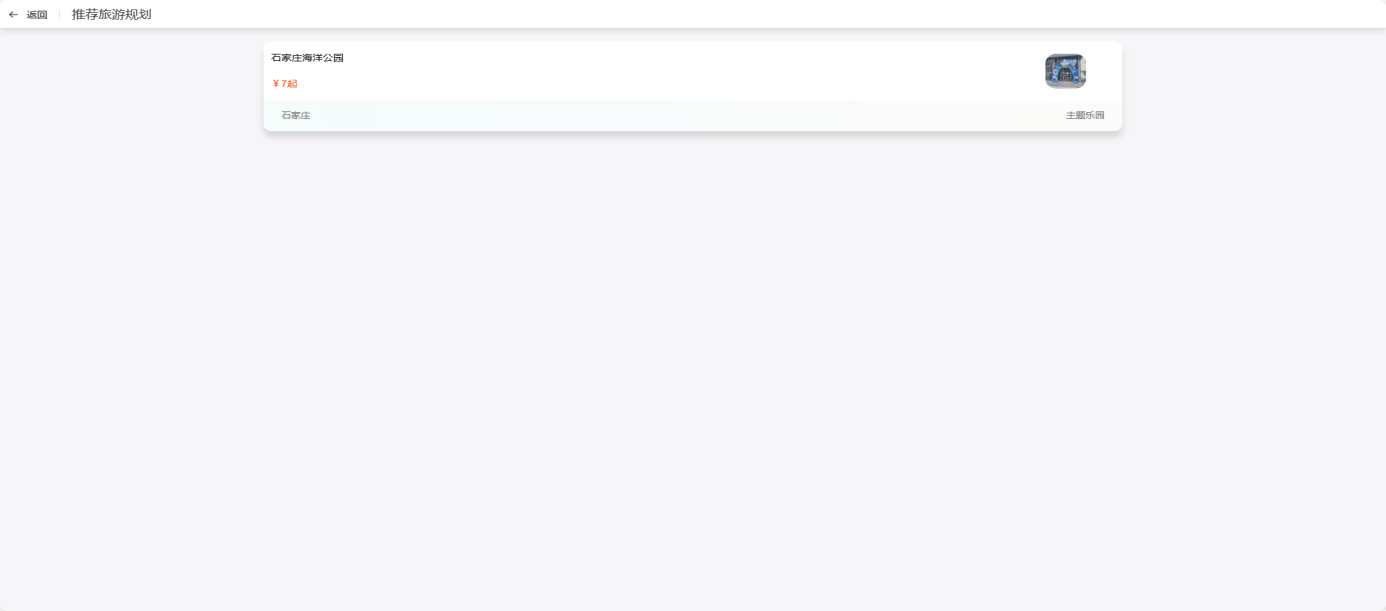
1. 浏览记录页面



1. 收藏页面



1. 推荐旅游计划页面



管理员后台

