**男性服装搭配系统**

本项目为江西软件职业技术大学易海编写

1. **灵感产生**

**这是活动开始的前一天，突然想到AIPO活动这个场合我应该穿什么的问题，于是我就思考，我自己既然有这个需求，对同样年龄阶段的男性、不同年龄段的男性来说，应该都偶尔有这种需求，而小红书虽然可以找到穿搭搭配，不过太混乱了，找半天还是找不到合适的……于是，我就找Deepseek为我打工……**

1. **项目介绍**
2. **系统概述**

本系统为15-40岁男性用户提供智能化服装搭配服务，包含四大核心场景（日常/运动/商务/约会），通过可视化交互实现个性化推荐。

1. **功能模块**
2. **用户画像构建**

* 三档年龄区间选择
* 动态按钮交互效果（ initializeAgeSelector ）

1. **场景化推荐**

graph LR

A[场景卡片] --> B{用户点击}

B --> C[更新推荐]

C --> D[价格过滤]

D --> E[渲染列表]

* 四类场景卡片（带毛玻璃特效）
* 实时推荐更新机制

1. **智能筛选系统**

* 价格区间：100-5000元
* 步长调节：100元/档
* 实时反馈当前价格

1. **收藏管理系统 功能 实现方式 数据持久化 添加收藏**

* 点击❤按钮
* localStorage存储 取消收藏
* 再次点击已收藏按钮
* 实时更新状态 收藏夹展示
* 独立网格布局
* 跨场景数据聚合

1. **核心解决痛点**
2. **场景适配困惑**

// 场景数据映射结构

const outfitData = {

'daily': { /\* 日常搭配 \*/ },

'business': { /\* 商务搭配 \*/ }

}

* 痛点：男性用户常混淆工作/约会等场景的着装要求
* 解决：通过 initializeSceneSelector 实现精准场景匹配

1. **价格信息模糊**

<input type="range" min="100" max="5000" step="100" value="1000" id="price-input">

* 痛点：网购时价格筛选不直观
* 解决：实时价格区间控制（ initializePriceRange ）

1. **风格年龄错配**

// 年龄区间数据结构

'15-22': [ /\* 学生风格搭配 \*/ ],

'28-40': [ /\* 成熟风格搭配 \*/ ]

* 痛点：22岁穿成熟款显老气，35岁穿学院风显幼稚
* 解决：年龄分层推荐算法

1. **辅助解决痛点**
2. **搭配灵感缺失**

<div class="outfit-grid"><!-- 动态生成推荐 --></div>

* 痛点：缺乏日常搭配灵感来源
* 解决：瀑布流推荐系统每日更新搭配方案

1. **单品管理混乱**

// 收藏管理系统

function toggleFavorite(outfitId) {

localStorage.setItem('favorites', JSON.stringify(state.favorites));

}

* 痛点：看到心仪搭配后容易忘记保存
* 解决：跨场景收藏夹管理系统

1. **AI互动模块**

<section id="ai-interaction">

<h2>AI 互动</h2>

<div class="interaction-container">

<div class="upload-area">

<h3>上传图片</h3>

<input type="file" accept="image/\*">

</div>

<div class="text-display">

<h3>AI 输出</h3>

<div id="text-output"></div>

</div>

</div>

</section>

由于火山引擎服务商出现充值Bug，调用Coze的API出现了问题，原本是使用AI分析穿着，并且给出评分，帮助男性用户解决穿着问题

1. **数据过滤逻辑**

推荐算法 = 当前场景 ∩ 年龄区间 ∩ (价格 ≤ 选定值)

1. **项目结构说明**

n:\AIPO\男性服装搭配系统

├── public/

│ ├── app.js # 核心交互逻辑 (356行)

│ ├── styles.css # 毛玻璃样式系统 (364行)

│ ├── index.html # 主界面结构 (91行)

│ └── images/ # 服装展示图集

├── server.js # 后端服务入口

├── prd.md # 需求文档（含交互规范）

└── package.json # NPM依赖配置

1. **部署**

# 安装依赖

npm install express express-session multer

# 开发模式

npm run dev

# 生产部署

npm start

1. **工具**

全程使用Deepseek-R1整理需求+国际版Trae AI IDE 辅助编程。

* Trae

https://www.trae.ai/?utm\_source=content&utm\_medium=ACCarticle&utm\_campaign=school

1. **本实践需要的基础**
2. **会使用DeepSeek**

我想开发一款专注于男性服装搭配的网页，能够帮助从高中到40岁男性推荐衣服的网页，前段使用Html实现，交互使用css和JavaScript实现，风格是毛玻璃特效，请给我生成一份详细的需求文档，prd.md，以及用户使用文档README.md。

1. **PRD.md**

# 男性服装搭配系统需求文档

## 1. 项目概述

### 1.1 产品背景

面向15-40岁男性用户提供智能化服装搭配解决方案，解决日常穿搭选择困难问题，结合毛玻璃视觉特效提升用户体验。

### 1.2 产品目标

- 提供年龄分层的穿搭推荐（高中生/大学生/职场新人/成熟男性）

- 支持不同场景搭配（日常/运动/商务/约会）

- 实现基于用户输入的智能推荐系统

- 打造视觉突出的毛玻璃交互界面

## 2. 用户角色

| 角色类型 | 特征描述 | 核心需求 |

|----------|---------------------------|---------------------------|

| 学生群体 | 15-22岁，预算有限 | 平价潮流/校园穿搭 |

| 职场新人 | 22-28岁，注重形象 | 通勤穿搭/性价比方案 |

| 成熟男性 | 28-40岁，品质要求高 | 商务/轻奢/场合着装 |

## 3. 功能需求

### 3.1 核心功能模块

#### 3.1.1 智能搭配推荐

- 基于用户画像的每日推荐

- 动态过滤系统（价格/风格/场合）

- 3D搭配预览效果

#### 3.1.2 单品推荐系统

- 当季热门单品榜单

- 相似单品推荐

- 尺寸匹配建议

#### 3.1.3 用户个性化

- 风格偏好测试

- 收藏夹管理

- 历史搭配记录

### 3.2 详细功能说明

1. \*\*场景选择器\*\*

- 可视化场景卡片（毛玻璃效果）

- 滑动切换场景

- 实时推荐更新

2. \*\*搭配展示墙\*\*

- 瀑布流布局

- 悬停放大效果

- 一键收藏功能

3. \*\*虚拟试衣间\*\*

- 身体参数输入

- 简易体型建模

- 搭配效果模拟

## 4. 非功能需求

### 4.1 性能要求

- 首屏加载时间 < 1.5s

- 推荐响应时间 < 0.5s

- 兼容Chrome/Firefox/Edge最新版

### 4.2 视觉规范

1. \*\*毛玻璃特效标准\*\*

- 背景模糊度：8px

- 透明度：90%

- 边缘渐变：linear-gradient(45deg, #ffffff22, #ffffff0d)

2. \*\*色彩方案\*\*

- 主色调：#2A2E35（深灰）

- 辅色调：#4B5162（蓝灰）

- 强调色：#C7786B（陶土红）

## 5. 技术方案

### 5.1 前端架构

```mermaid

graph TD

A[HTML结构] --> B[CSS毛玻璃特效]

A --> C[JavaScript交互]

C --> D[推荐算法]

C --> E[本地存储]

B --> F[响应式布局]

**数据存储**

* 用户偏好：LocalStorage
* 搭配数据：JSON静态文件
* 图片资源：CDN加速

1. 项目里程碑

|  |  |  |
| --- | --- | --- |
| **阶段** | **时间线** | **交付物** |
| Alpha | 第1-2周 | 核心推荐功能+基础UI |
| Beta | 第3-4周 | 完整交互+虚拟试衣 |
| RC | 第5周 | 性能优化+跨浏览器测试 |

1. **README.md**

**# 男性服装搭配系统使用指南**

**## 系统要求**

- 现代浏览器（推荐Chrome 90+）

- 屏幕分辨率：≥1280x720

- 网络连接：需要加载图片资源

**## 快速入门**

**### 1. 场景选择**

```html

<!-- 示例代码片段 -->

<div class="scene-selector">

<div class="scene-card" data-scene="casual">

<h3>日常穿搭</h3>

</div>

<div class="scene-card" data-scene="business">

<h3>商务着装</h3>

</div>

</div>

2. 搭配操作流程

选择年龄区间 → 2. 选择使用场景 → 3. 设置价格范围 → 4. 查看推荐结果

3. 核心交互说明

收藏功能

javascript

复制

// 示例收藏逻辑

function toggleFavorite(outfitId) {

const favorites = JSON.parse(localStorage.getItem('favorites')) || [];

const index = favorites.indexOf(outfitId);

if (index === -1) {

favorites.push(outfitId);

} else {

favorites.splice(index, 1);

}

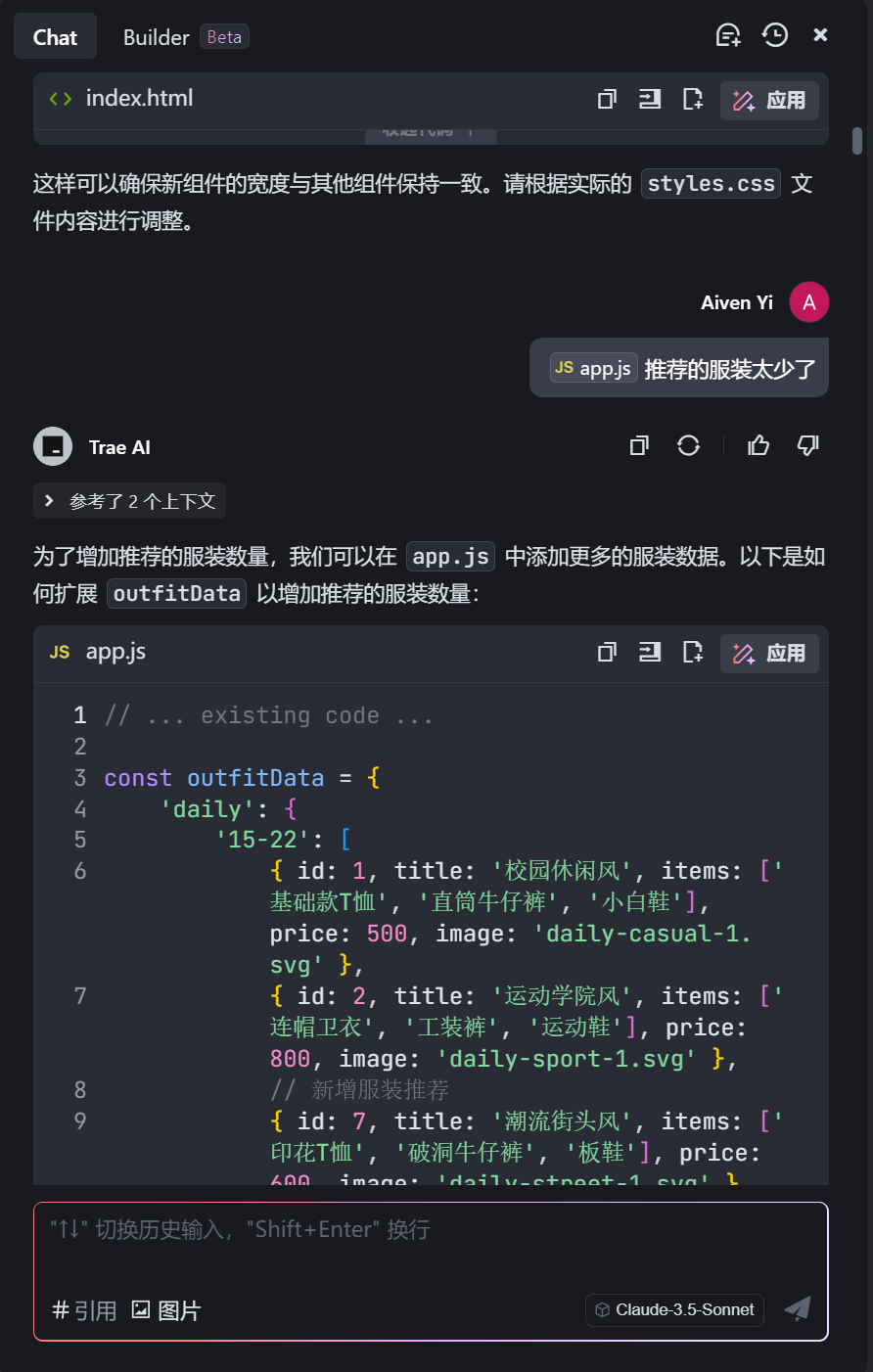
localStorage.setItem('favorites', JSON.stringify(favorites));

1. **会使用Trae，Chat和Builder交替使用**
2. **工具使用**

我想开发一款专注于男性服装搭配的网页，能够帮助从高中到40岁男性推荐衣服的网页，前段使用Html实现，交互使用css和JavaScript实现，风格是毛玻璃特效， prd.md README.md 请根据我的这两个文档，为我开发这款网页，

我需要将这些div设计成扁平化科技化的设计，并且整个网页的背景，主题色是粉色到蓝色到紫色的渐变色

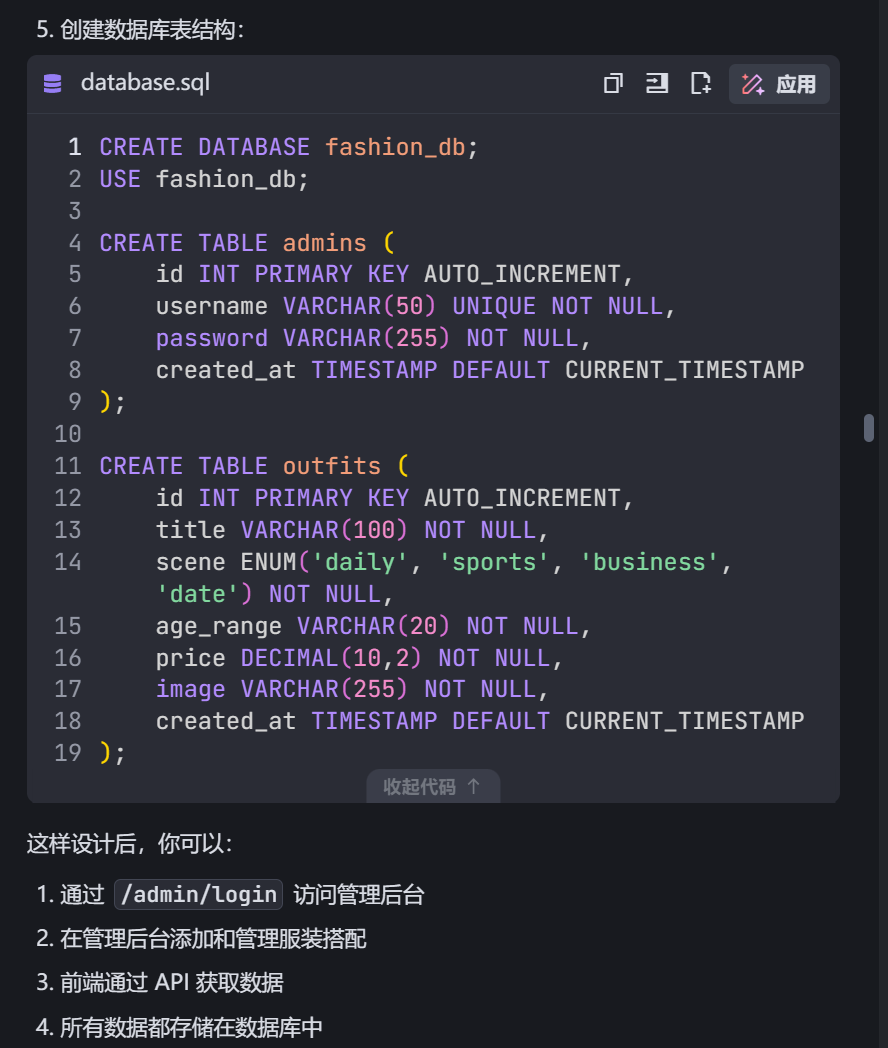
Chat模式



Tip：Chat模式下可以任意文件中Ctrl+Z撤销，但是Builder不能直接撤销



数据库



1. **代码片段**

Html代码

<!DOCTYPE html>

<html lang="zh-CN">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>男性服装搭配系统</title>

<link rel="stylesheet" href="styles.css"> <!-- 修改路径 -->

</head>

<body>

<div class="container">

<header class="glass-effect">

<h1>男性服装搭配助手</h1>

</header>

<main>

<section id="user-profile" class="glass-effect">

<h2>个人信息</h2>

<div class="age-selector">

<h3>选择年龄区间</h3>

<div class="age-options">

<button data-age="15-22">学生群体 (15-22岁)</button>

<button data-age="22-28">职场新人 (22-28岁)</button>

<button data-age="28-40">成熟男性 (28-40岁)</button>

</div>

</div>

</section>

<section id="scene-selector" class="glass-effect">

<h2>场景选择</h2>

<div class="scene-cards">

<div class="scene-card" data-scene="daily">

<h3>日常穿搭</h3>

<p>适合日常通勤、休闲场合</p>

</div>

<div class="scene-card" data-scene="sports">

<h3>运动风格</h3>

<p>适合运动、户外活动</p>

</div>

<div class="scene-card" data-scene="business">

<h3>商务正装</h3>

<p>适合职场、正式场合</p>

</div>

<div class="scene-card" data-scene="date">

<h3>约会装扮</h3>

<p>适合约会、社交场合</p>

</div>

</div>

</section>

<section id="price-range" class="glass-effect">

<h2>价格范围</h2>

<div class="price-slider">

<input type="range" min="100" max="5000" step="100" value="1000" id="price-input">

<span id="price-value">￥1000</span>

</div>

</section>

<section id="outfit-recommendations" class="glass-effect">

<h2>搭配推荐</h2>

<div class="outfit-grid">

<!-- 搭配推荐将通过JavaScript动态生成 -->

</div>

</section>

<section id="favorites" class="glass-effect">

<h2>我的收藏</h2>

<div class="favorites-grid">

<!-- 收藏夹内容将通过JavaScript动态生成 -->

</div>

</section>

</main>

</div>

<section id="ai-interaction" class="glass-effect">

<h2>AI 互动</h2>

<div class="interaction-container glass-effect">

<div class="upload-area">

<h3>上传图片</h3>

<input type="file" accept="image/\*">

</div>

<div class="text-display">

<h3>AI 输出</h3>

<div class="text-output" id="text-output">

<!-- AI 输出的文字将通过JavaScript动态生成 -->

</div>

</div>

</div>

</section>

<script src="./app.js"></script> <!-- 修改路径 -->

</body>

</html>

我想插入一个div组件在“我的收藏”下面，我看了代码中，应该是使用类似代码：<section id="favorites" class="glass-effect">

<h2>我的收藏</h2>

<div class="favorites-grid">

<!-- 收藏夹内容将通过JavaScript动态生成 -->

</div>

</section>

请你为我设计好，保证风格一致性。这个新增加的组件是一个左边上传图片的区域，右边是输出文字的文字展示框，可以随着文字的长度在这个div中滚动查看文字，我打算接入的是coze的API，coze的API如下：curl --location --request POST 'https://api.coze.cn/v3/chat' \

--header 'Authorization: Bearer pat\_OYDacMzM3WyOWV3Dtj2bHRMymzxP\*\*\*\*' \

--header 'Content-Type: application/json' \

--data-raw '{

"bot\_id": "73428668\*\*\*\*\*",

"user\_id": "123123\*\*\*",

"stream": false,

"auto\_save\_history":true,

"additional\_messages":[

{

"role":"user",

"content":"早上好",

"content\_type":"text"

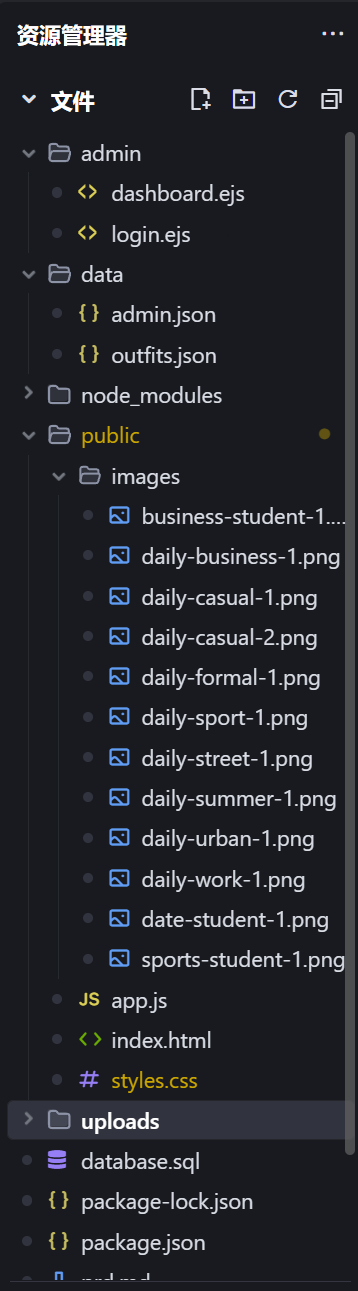
}

]

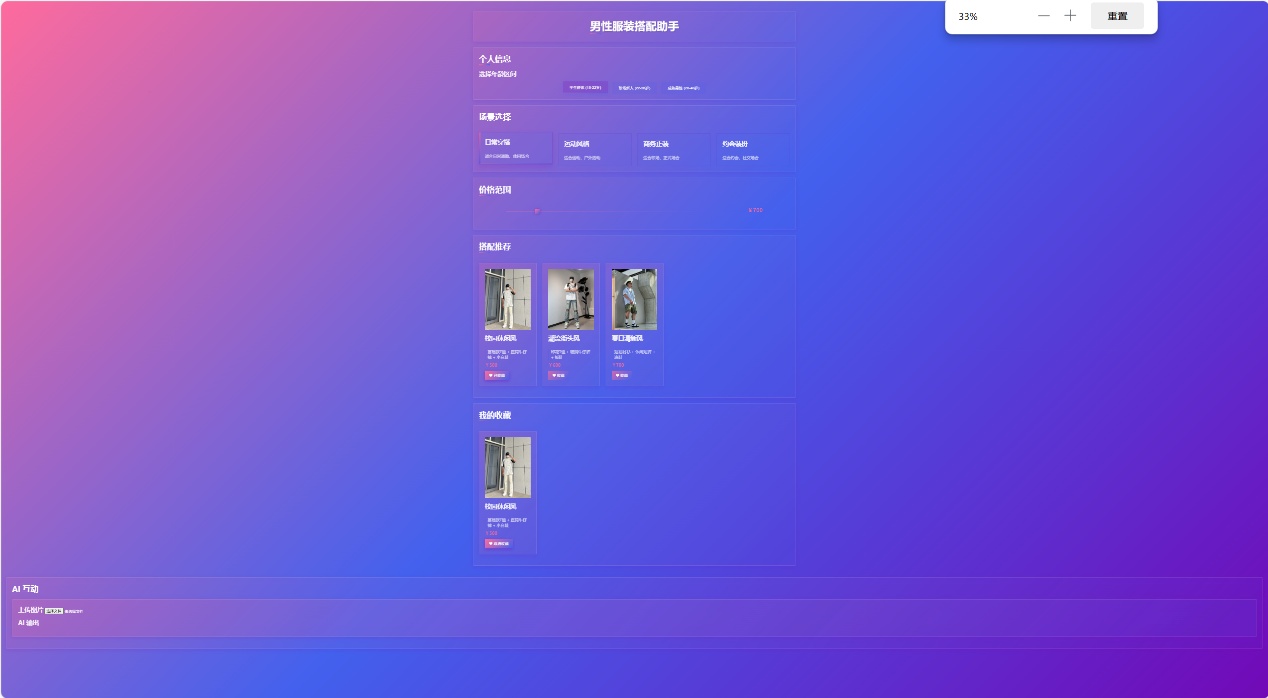
}'

，接入API之后就是有了AI功能了

总体架构



1. **运行演示**



此项目后续可以出更加详细的细节教程