

项目 7-红色党建研学平台（赛题 20）

（浙师大软件工程专业内部文档，请勿外传）

目录

文档 1：项目概况 ..... 2

一、项目背景与定位 ..... 2

二、用户群体与需求 ..... 2

三、业务需求细化（参考） ..... 2

四、技术要求 ..... 5

文档 2：基于 Scrum 过程的项目实训方案 ..... 6

一、Scrum 团队组建 ..... 6

二、第一次迭代（1-2 周） ..... 6

三、第二次迭代（2-3 周） ..... 8

四、第三次迭代（3-4 周） ..... 9

五、项目收尾 ..... 10

附件：部分用户故事细化参考（同学们可以根据自己用户故事细化） ..... 10

文档 3：项目技术梳理 ..... 16

一、系统架构 ..... 16

二、前端技术 ..... 16

三、后端技术 ..... 16

四、通用技术点 ..... 17

五、增值功能技术点 ..... 17

文档 4：“虚拟讲解员”功能的技术路径 ..... 19

参考 1：开源项目 SadTalker 和大模型创建虚拟讲解员 ..... 22

参考 2：利用开源大模型 audio2face 创建虚拟讲解员 ..... 23

参考 3：基于开源大模型 Facemoji 创建虚拟讲解员 ..... 24

文档 5：项目技术学习和验证方案 ..... 26

文档 6：项目测试方案 ..... 30

## 文档 1：项目概况

### 一、项目背景与定位

随着信息化时代的到来，各企事业单位党组织对党建教育、培训、考核的需求日益增强，尤其是追求高效、便捷、互动的学习方式。本项目旨在构建一个集红色教育、廉政文化、研学活动管理于一体的全方位红色党建研学平台，服务于县处级以上党政领导干部、公务员、国有企业及事业单位高层管理人员，以及广大党员群众，通过线上线下相结合的方式，提升党建教育的吸引力和实效性。

### 二、用户群体与需求

**核心用户：**各企事业单位党组织负责人（如组织部负责人、二级党委（总支）负责人、党支部负责人）及上述提到的各类党政干部、管理人员。

**用户期望：**打造一个红色党建研学平台，能够全方位、一站式服务红色研学活动。

### 三、业务需求细化（参考）

#### 1. 红色党建研学基地与廉政教育基地设计

- (1) **红色研学基地：**基于现有线下红色教育资源（如嘉兴红船、井冈山等），通过高清图片、视频、VR 全景等形式在平台上展示，同时提供详细的研学内容、历史背景、特色活动介绍，引导用户深入了解并参与。

##### 内容展示：

- 高清图片与视频：**为每个红色研学基地准备高清图片和短视频，展示其独特的历史遗迹、风景和纪念物。
- VR 全景：**利用 VR 技术创建全景漫游体验，让用户仿佛身临其境地参观红色景点。
- 研学内容：**编写详细的研学资料，包括历史背景、重要事件、人物介绍等，以图文、视频或音频形式呈现。
- 特色活动：**介绍基地定期举办的特色活动，如纪念仪式、讲座、展览等，并提供报名链接或联系方式。

##### 互动功能：

- 在线导览：**提供语音导览或文字导览功能，用户可根据自己的兴趣选择路线。
- 虚拟讲解员：**利用 AI 技术创建虚拟讲解员，为用户提供个性化讲解服务。
- 互动问答：**设置与研学内容相关的问答环节，增强用户参与感。

- (2) **廉政教育基地：**构建虚拟空间，围绕廉政主题设计多样化内容，包括但不限于案例分析、互动问答、情景模拟等，鼓励用户自主学习与反思。

##### 内容设计：

- **案例分析：**选取典型廉政案例，通过图文、视频等形式详细解析，揭示腐败的危害和廉政的重要性。
- **互动问答：**设计关于廉政知识的选择题、判断题等，用户答题后可获得反馈和解析。
- **情景模拟：**利用 VR 或游戏化方式，模拟廉政决策场景，让用户在实际操作中学习廉政知识。

#### **学习体验：**

- **自主学习路径：**根据用户的学习进度和兴趣，推荐相关学习材料和活动。
- **反思与讨论：**设置讨论区或论坛，鼓励用户分享学习心得和反思，促进思想交流。

## **2. 综合服务平台**

- (1) **基地入驻与管理：**提供红色基地入驻流程、资料审核、信息发布等功能，确保基地信息的准确性和时效性。
  - **入驻流程：**明确基地入驻的申请条件、提交材料、审核流程和时间节点。
  - **资料审核：**建立严格的审核机制，对基地提交的资料进行真实性、准确性和完整性的审核。
  - **信息发布：**为入驻基地提供信息发布平台，包括基地介绍、活动预告、联系方式等。
- (2) **研学活动预约与管理：**支持用户在线预约研学活动，包括日期、人数、基地选择等，同时提供交通、住宿、餐饮等一站式预订服务。
  - **预约系统：**开发在线预约系统，支持用户选择研学活动的日期、人数、基地等，并生成预约订单。
  - **一站式预订：**整合交通、住宿、餐饮等资源，提供一站式预订服务，简化用户操作流程。
  - **活动管理：**对预约活动进行统一管理，包括活动确认、通知发送、现场签到等。
- (3) **角色体验与互动活动：**设计多种角色体验场景（如红军战士、革命先驱等），以及互动讲解、知识竞赛等趣味活动，增强学习互动性。
  - **角色设计：**根据历史背景和研学需求，设计多种角色体验场景和角色形象。
  - **互动活动：**开发互动讲解、知识竞赛、角色扮演等趣味活动，增强用户参与感和互动性。
  - **活动记录：**记录用户的活动参与情况，包括得分、排名、奖励等，并提供查看和分享功能。
- (4) **文创产品商城：**展示并销售红色文化相关的文创产品，如纪念品、书籍、服饰等，丰富用户体验。
  - **产品展示：**展示红色文化相关的文创产品，包括纪念品、书籍、服饰等，提供详细的产品信息和图片。
  - **购物车与结算：**开发购物车和结算功能，支持用户在线购买文创产品。
  - **订单管理：**对订单进行统一管理，包括订单状态跟踪、物流信息查询等。

(5) **评价与反馈：**设立评价系统，鼓励用户分享研学体验，收集反馈意见，不断优化服务。

- **评价系统：**设立用户评价系统，鼓励用户对研学活动、基地服务、文创产品等进行评价。
- **反馈收集：**收集用户的反馈意见和建议，为平台优化提供数据支持。
- **数据分析：**对评价数据进行统计分析，识别用户需求和痛点，提出改进措施。

### 3. 收益分配与后台管理

**3.1 收益分配机制：**建立合理的收益分配模型，确保平台和红色基地双方利益最大化。明确平台和红色基地之间的收益分配比例和原则，制定详细的结算流程和时间表，确保收益及时、准确地分配给各方。提供收益分配报告和查询功能，确保分配过程的透明化和公正性。

**3.2 后台管理系统：**包含数据分析（用户行为、研学效果等）、订单处理、资源调度、用户管理、内容管理等模块，支持高效运营与决策。具体功能包括：

(1) 资源调度：

- **研学活动调度：**根据预约情况、基地承载能力和活动安排，合理分配研学活动的时间和地点，确保活动顺利进行。
- **人员调度：**对于需要现场导览、讲解或组织活动的情况，提前调度相关人员，确保人员配置充足且合理。
- **物资管理：**管理研学活动所需的物资，如宣传材料、教学用具、VR设备等，确保物资供应充足且及时。

(2) 用户管理：

- **用户信息管理：**存储用户的基本信息（如姓名、联系方式、地址等），并保护用户隐私。
- **用户行为追踪：**记录用户的登录、浏览、购买等行为，用于分析用户需求和优化用户体验。
- **用户权限管理：**根据用户角色（如普通用户、管理员、基地负责人等）分配不同的权限，确保系统安全。

(3) 内容管理：

- **内容发布与审核：**为基地提供内容发布平台，并对发布的内容进行审核，确保内容的准确性和合规性。
- **内容更新与维护：**定期更新研学内容、活动信息和文创产品等，保持平台内容的时效性和吸引力。
- **内容分类与标签：**对平台上的内容进行分类和标签化，方便用户搜索和浏览。

(4) 订单处理：

- **订单生成与确认：**用户完成预约或购买后，自动生成订单并通知相关方进行确认。
- **订单状态跟踪：**提供订单状态跟踪功能，用户可随时查看订单的当前状态（如待支付、已支付、已发货、已完成等）。

- 退款与售后：对于需要退款或售后的订单，提供便捷的退款流程和售后服务，保障用户权益。

(5) 数据分析：

- 用户行为分析：通过数据分析工具，分析用户的浏览、点击、购买等行为，了解用户需求和偏好。
- 研学效果评估：收集用户对研学活动的反馈和评价，评估活动的参与度和满意度，为活动改进提供依据。
- 业务趋势预测：基于历史数据和当前趋势，预测平台的业务发展趋势，为决策提供支持。

#### 四、技术要求

- **系统架构**：采用前后端分离的设计，前端支持小程序和主流浏览器访问，后端采用稳定的服务器架构，确保数据安全与访问速度。
- **界面设计**：界面友好，符合党建教育氛围，支持多设备适配，提供良好的用户体验。
- **技术选型**：前端可使用 React/Vue 等现代框架，后端可选择 Spring Boot 等 Java 框架或 Node.js 等，数据库可采用 MySQL/MongoDB 等，确保系统可扩展性和维护性。
- **安全性**：加强数据加密、访问控制、日志审计等安全措施，保障用户信息和平台数据的安全。

#### 参考资料：

<https://hpxy.yapt.edu.cn/hsya/mzsd.htm>

[https://hpxy.yapt.edu.cn/?bd\\_vid=15657468687399424044](https://hpxy.yapt.edu.cn/?bd_vid=15657468687399424044)

<http://www.royalcent.com/a/szzt/lzzt/>

<https://jiwei.yantai.gov.cn/col/col6721/>

<https://zhuanlan.zhihu.com/p/698834258>

## 文档 2：基于 Scrum 过程的项目实训方案

### 一、Scrum 团队组建

#### 1. 团队构成

(1) 产品负责人：负责定义红色党建研学平台的愿景，收集并分析党组织、党政干部及管理人员对党建教育、培训、考核的需求，设定产品功能的优先级，确保产品符合业务目标，并与各利益相关者（如企事业单位党组织负责人）保持密切沟通。（技能：市场分析能力、产品规划能力、沟通协调能力）

(2) Scrum 主管（Scrum Master）：作为团队的教练和推动者，确保 Scrum 流程（如每日站会、Sprint 评审会议、Sprint 回顾会议）的有效执行，组织会议，移除团队在开发过程中遇到的障碍，促进团队自组织和持续改进。（技能：敏捷管理知识、团队领导力、问题解决能力）

(3) 开发团队：

- UI/UX 设计师：设计用户界面和用户体验，确保平台界面美观、符合党建教育氛围，并具备良好的用户体验。（技能：视觉设计、用户体验设计、原型工具，如 Sketch、Figma）
- 前端开发工程师：负责实现前端页面和功能，包括 HTML、CSS、JavaScript 及前端框架（如 React、Vue）的应用，确保前端页面在各浏览器和移动设备上的兼容性和性能。（技能：前端开发技术、前端框架应用、响应式设计）
- 后端开发工程师：构建系统架构，设计数据库（如 MySQL、MongoDB），实现后端逻辑和 API 接口，确保系统的高效运行和数据安全。（技能：后端开发技术（如 Java、Node.js）、数据库设计、API 开发）
- 测试工程师：编写测试用例，执行自动化和手动测试，确保软件质量，包括功能测试、性能测试、安全测试等。（技能：测试框架应用（如 JUnit、Jest）、自动化测试工具（如 Selenium）、性能测试工具）
- 数据分析师：负责初步的数据收集与分析，如用户行为分析、研学效果评估，为产品优化提供数据支持。（技能：数据分析工具（如 Excel、Python 数据分析库）、统计分析方法）

#### 2. 团队沟通与文化

- (1) 强调团队协作、透明度和持续改进。
- (2) 鼓励开放交流，建立快速反馈机制。

### 二、第一次迭代（1-2 周）

#### (一) 目标

完成红色党建研学基地与廉政教育基地的初步设计和部分功能开发。

#### (二) 迭代计划

##### 1. 计划会议

- 召集项目团队全体成员，明确本次迭代的目标和范围。
- 基于优先级排序的用户故事，讨论并确定具体的迭代任务清单。

- 设定迭代的时间表和里程碑，确保任务按时完成。

## 2. 用户故事参考：

### (1) 红色研学基地

- 作为访问者，我想要能在红色研学基地平台上看到高清图片和短视频，以便直观地感受红色历史遗迹、风景和纪念物的魅力。
- 作为访问者，我希望能通过 VR 全景功能，身临其境地参观红色研学基地的各个景点，以便更深入地了解历史背景和细节。
- 作为访问者，我想要获取关于每个红色研学基地的详细研学资料，包括历史背景、重要事件和人物介绍，以图文、视频或音频形式呈现，以便系统地学习相关知识。
- 作为访问者，我想要了解并参加基地定期举办的特色活动，如纪念仪式、讲座、展览等，并希望能直接在平台上进行报名或获取联系方式。
- 作为自主游览的用户，我想要使用在线导览功能（语音或文字），以便根据自己的兴趣选择路线并了解每个景点的信息。
- 作为参与者，我希望通过互动问答环节，回答与研学内容相关的问题，以增强我的参与感和知识掌握度。

### (2) 廉政教育基地

- 作为学习者，我想要通过图文、视频等形式详细解析典型廉政案例，以便深刻理解腐败的危害和廉政的重要性。
- 作为学习者，我希望参与关于廉政知识的选择题、判断题等互动问答，并在答题后获得即时的反馈和解析，以提升我的学习效果。
- 作为学习者，我希望通过 VR 或游戏化方式模拟廉政决策场景，以便在实际操作中学习和掌握廉政知识。
- 作为学习者，我希望系统能根据我的学习进度和兴趣，智能推荐相关的廉政学习材料和活动，以提高学习效率。
- 作为学习者，我希望有一个讨论区或论坛，能让我分享学习心得、反思学习内容，并与其他用户进行思想交流，共同进步。

### (三) 任务分配与执行：

1. 需求分析：细化需求，确保所有利益相关者的期望被准确捕捉。
2. 设计：完成初步设计稿，包括页面流程、交互设计和视觉设计。
3. 开发：
  - 前端：实现图片与视频展示页面，VR 全景技术框架搭建。
  - 后端：搭建基础架构，设计数据库模型，实现初步的数据交互 API。
4. 测试：编写并执行单元测试，初步评估功能实现。

### (四) 每日站立会议

每天早上进行 15 分钟左右的站立会议，团队成员依次回答三个问题：昨天做了什么？今天计划做什么？遇到了什么障碍？额外建议：鼓励团队成员在站立会议中分享学习心得和技术难题，促进团队内部的知识共享和问题解决。



#### （五）开发与测试

开发人员按照计划进行开发工作，包括图片和视频的处理、内容编写、页面设计等。

#### （六）迭代评审会议

在迭代结束时，召开评审会议。团队向产品负责人展示完成的工作成果，包括初步的红色研学基地内容展示、廉政教育基地案例分析模块的页面，以及简单的互动问答功能。

#### （七）迭代回顾会议

总结经验教训，讨论哪些方面做得好，哪些需要改进，为下一次迭代提供参考。

### 三、第二次迭代（2-3 周）

#### （一）目标

完善基地的设计与功能，初步搭建综合服务平台的部分模块。

#### （二）迭代计划

##### 1. 计划会议

- 明确本次迭代的重点是完善基地功能和搭建部分综合服务平台模块。
- 基于优先级排序的用户故事，制定详细的任务清单，包括在线导览、虚拟讲解员、预约系统等功能。
- 设定时间表和里程碑，确保任务按时完成。

##### 2. 用户故事参考：

##### 基地入驻与管理：

- 作为红色基地管理员，我想要一个明确的入驻流程，包括申请条件、提交材料、审核流程和时间节点，以便顺利将我的基地信息添加到平台上。
- 作为平台管理员，我需要一个严格的资料审核机制，以确保入驻基地提交的资料真实、准确、完整，从而维护平台的权威性和用户信任。
- 作为红色基地管理员，我想要一个信息发布平台，能够方便地发布基地介绍、活动预告、联系方式等信息，以便吸引用户关注和参与。

##### 研学活动预约：

- 作为用户，我想要一个在线预约系统，能够方便地选择研学活动的日期、人数、基地等，并生成预约订单，以便提前规划我的研学行程。
- 作为用户，我希望在预约研学活动时，能够同时预订交通、住宿、餐饮等一站式服务，以简化操作流程，提升用户体验。
- 作为平台管理员，我需要一个活动管理系统，能够对预约活动进行统一管理，包括活动确认、通知发送、现场签到等，以确保活动的顺利进行。

##### 角色体验与互动：

- 作为用户，我希望能够参与多种角色体验场景，如扮演红军战士、革命先驱等，以便更深入地了解历史背景和人物故事。
- 为平台开发者，我们需要开发互动讲解、知识竞赛、角色扮演等趣味活动，以增强用户的参与感和互动性，提高学习效果。
- 作为用户，我想要记录我的活动参与情况，包括得分、排名、奖励等，并希望有一个分享功能，能够将我的成果展示给朋友和家人。



文创产品商城：

- 作为用户，我想要浏览红色文化相关的文创产品，如纪念品、书籍、服饰等，并希望平台提供详细的产品信息和图片，以便我做出购买决策。
- 作为用户，我希望在选购文创产品时，能够使用购物车功能，并支持在线结算，以便方便快捷地完成购买流程。
- 作为用户，我希望能够方便地查看我的订单状态，跟踪物流信息，并在需要时联系客服解决问题。

（三）任务分配与执行：

1. 设计：进一步优化页面设计，确保用户体验流畅。
2. 开发：
  - 前端：实现在线导览、虚拟讲解员、预约系统等页面。
  - 后端：开发基地入驻流程、资料审核逻辑、预约系统后端接口。
3. 测试：扩展测试用例，覆盖新增功能，确保系统稳定性。
4. 数据分析：初步建立用户行为追踪机制，为后期优化提供依据。

（四）每日站立会议

每天交流工作进展、计划和遇到的问题。

（五）开发与测试

开发人员进行功能开发，如 VR 全景展示的技术实现、情景模拟场景的构建等。

（六）迭代评审会议

展示完善后的基地设计和初步搭建的综合服务平台模块，如入驻流程页面。

（七）迭代回顾会议

总结经验，讨论改进措施。

## 四、第三次迭代（3-4 周）

（一）目标

完成整个项目的开发，进行全面测试和优化。

（二）迭代计划

1. 计划会议

- 确定本次迭代的任务为完成所有剩余功能开发，并进行全面整合和优化。
- 制定详细的任务清单，包括文创产品商城、评价与反馈系统、后台管理系统等功能。
- 设定时间表和里程碑，确保项目按时完成并达到上线标准。

2. 用户故事参考

评价与反馈：

- 作为自主游览的用户，我希望有一个 AI 虚拟讲解员，能为我提供个性化的讲解服务，增加游览的趣味性和互动性。
- 作为用户，我想要一个评价系统，能够对我的研学体验、基地服务、文创产品等进行评价，以便帮助其他用户做出选择，并促进平台服务的持续改进。
- 作为平台管理员，我需要收集用户的反馈意见 and 建议，并进行数据分析，以识别用户需求和问题点，从而提出改进措施，优化平台服务。

- 后台管理系统具备数据分析、订单处理、资源调度等能力，支持平台的日常运营和管理。

### （三）任务分配与执行

#### 1. 开发：

- 前端：完成文创产品商城、评价与反馈系统页面开发。
- 后端：实现文创产品商城的购物车、结算功能，完善评价与反馈系统。
- 后台管理系统：开发数据分析、订单处理、资源调度等功能模块。

2. 测试：进行全面集成测试、性能测试和用户验收测试，确保系统质量。

3. 部署：将系统部署到生产环境，进行压力测试和最终调试。

4. 培训与支持：为基地管理人员提供系统使用培训，建立技术支持渠道。

### （三）每日站立会议

持续沟通工作情况。

### （四）开发与测试

开发人员全力完成剩余开发工作。

### （五）迭代评审会议

总结整个项目开发经验，识别成功因素与待改进之处，为团队和个人成长提供反馈。

### （六）迭代回顾会议

总结项目开发过程，为后续类似项目提供经验。

## 五、项目收尾

### （一）产品发布

1. 发布准备：在迭代评审会议后，根据收集到的反馈进行必要的调整和优化。确保所有已知问题得到修复，文档和用户手册已更新至最新版本。
2. 部署策略：制定详细的部署计划，包括回滚策略，以防在生产环境中出现意外情况。确保所有依赖的服务和系统都已准备就绪。
3. 正式发布：按照部署计划，将系统平滑地迁移到生产环境。监控系统的性能和稳定性，确保用户能够顺利访问和使用新功能。

### （三）项目总结与文档归档

1. 项目总结：编写项目总结报告，详细记录项目的目标、过程、成果、经验教训等。
2. 文档归档：将所有项目文档进行整理、归档和保存。包括需求文档、设计文档、测试报告、用户手册等。

### （六）项目考核和团队经验分享

组织团队经验分享会议。让团队成员分享自己的经验和教训，促进知识的传播和团队的学习成长。

**附件：部分用户故事细化参考（同学们可以根据自己用户故事细化）**

**用户故事 1：**作为访问者，我想要能在红色研学基地平台上看到高清图片和短视频，以便直

### 观地感受红色历史遗迹、风景和纪念物的魅力

故事背景：用户作为党政干部、管理人员或党员群众，对红色党建研学活动充满兴趣，希望通过线上平台初步了解各红色研学基地的风貌和历史背景。

用户行为：

- 用户登录红色党建研学平台，进入红色研学基地展示板块。
- 在该板块下，用户可以浏览到多个红色研学基地的高清图片和短视频。
- 通过点击图片或视频，用户可以放大查看细节，或播放短视频以更直观地感受红色研学基地的独特之处。
- 每张图片或短视频下方，都附有简短的文字说明，介绍该地点的历史背景、重要事件或特色亮点。

系统响应：

- 系统展示红色研学基地的高清图片和短视频，确保画面清晰、内容真实。
- 图片和视频按类别或地区有序排列，便于用户查找。
- 提供流畅的播放和查看体验，支持用户缩放、暂停、播放等操作。
- 文字说明简洁明了，帮助用户快速了解红色研学基地的历史背景。

### 用户故事 2：作为访问者，我希望能通过 VR 全景功能，身临其境地参观红色研学基地的各个景点，以便更深入地了解历史背景和细节

故事背景：用户在初步了解红色研学基地后，希望通过更先进的技术手段，如 VR 全景技术，进一步感受红色景点的真实氛围。

用户行为：

- 在红色研学基地展示板块中，用户发现并支持 VR 全景功能的基地选项。
- 用户点击 VR 全景入口，进入 VR 全景体验模式。
- 佩戴 VR 设备（或使用手机/电脑浏览器中的 VR 功能）后，用户仿佛置身于红色景点之中，可以自由移动视角，观察周围环境。
- 系统提供语音导览或文字提示，帮助用户了解当前场景的历史意义和相关信息。

系统响应：

- 系统加载 VR 全景内容，确保画面流畅、无卡顿。
- 支持多种 VR 设备接入，同时提供适配不同屏幕尺寸和分辨率的浏览体验。
- 提供灵活的视角控制功能，允许用户自由转动头部、移动身体以探索周围环境。
- 语音导览或文字提示准确、及时，增强用户的沉浸感和体验感。

### 用户故事 3：作为访问者，我想要获取关于每个红色研学基地的详细研学资料，包括历史背景、重要事件和人物介绍，以图文、视频或音频形式呈现，以便系统地学习相关知识

故事背景：用户对红色研学基地产生了浓厚兴趣，希望深入了解其研学内容。

用户行为：

- 在红色研学基地展示板块中，用户点击“详细研学内容”链接。
- 系统展示该基地的详细研学内容，包括历史背景、重要事件、人物介绍等，以图文、视频或音频形式呈现。

系统响应：

- 系统展示详细的研学内容，确保内容准确、丰富。
- 研学内容以多种形式呈现，满足不同用户的学习需求。

**用户故事 4：作为访问者，我想要了解并参加基地定期举办的特色活动，如纪念仪式、讲座、展览等，并希望能直接在平台上进行报名或获取联系方式**

故事背景：用户对红色研学基地产生了浓厚兴趣，希望深入了解其特色活动安排。

用户行为：

- 在红色研学基地展示板块中，用户点击“特色活动介绍”链接。
- 系统列出该基地定期举办的特色活动信息，如纪念仪式、讲座、展览等，并提供报名链接或联系方式。

系统响应：

- 系统展示详细特色活动介绍，确保内容准确、丰富。
- 特色活动信息及时更新，确保用户获取到最新的活动资讯。
- 系统提供清晰的报名流程和联系方式，方便用户参与特色活动。

**用户故事 5：作为自主游览的用户，我想要使用在线导览功能（语音或文字），以便根据自己的兴趣选择路线并了解每个景点的信息**

故事背景：随着红色教育和廉政文化宣传的深入，越来越多的用户希望通过便捷的方式参与研学活动。为了满足这一需求，项目团队开发了在线导览功能，旨在为用户提供丰富多样的研学路线选择，提升用户的参与度和体验感。

用户行为：

- 用户登录综合服务平台，进入研学活动专区。
- 浏览平台上展示的各类研学路线，通过图片、视频和文字介绍初步了解每条路线的特色和亮点。
- 根据个人兴趣、时间安排和研学目标，选择一条或多条感兴趣的研学路线。
- 确认选择后，将选定的研学路线添加到个人行程计划中。

系统响应：

- 系统展示清晰、直观的研学路线列表，每条路线均配有详细的介绍和预览图。
- 根据用户的浏览历史和偏好，智能推荐相关研学路线，提升用户体验。
- 实时响应用户的选择操作，将选定的研学路线保存至用户个人中心，并生成个性化的行程计划。
- 提供路线规划工具，允许用户根据需求调整或自定义研学路线。

**用户故事 6：作为自主游览的用户，我希望有一个 AI 虚拟讲解员，能为我提供个性化的讲解服务，增加游览的趣味性和互动性**

故事背景：为了增强研学活动的互动性和教育效果，项目团队引入了虚拟讲解员技术。虚拟讲解员能够根据用户选择的研学路线和兴趣点，提供个性化的讲解服务，使用户在参观过程中获得更加深入和丰富的知识体验。

用户行为：

- 在研学活动过程中，用户通过平台或移动设备启动虚拟讲解员服务。
- 根据当前位置或已选研学路线，选择需要听取讲解的景点或区域。
- 通过语音或文字指令与虚拟讲解员进行互动，提出问题和获取更多信息。

系统响应：

- 系统识别用户的选择和位置信息，自动启动虚拟讲解员服务。
- 虚拟讲解员根据用户的选择和兴趣点，提供定制化的讲解内容，包括历史背景、人物故事、文化内涵等。
- 支持语音交互和智能问答功能，实时响应用户的提问和需求。
- 提供多种语言选择和语速调节功能，满足不同用户的听讲需求。

### 用户故事 7：基地入驻流程、资料审核和信息发布功能

故事背景：为了扩大研学活动的覆盖范围和提升服务质量，项目团队决定开放综合服务平台，允许符合条件的红色教育基地和廉政文化基地入驻。通过入驻流程、资料审核和信息发布功能的支持，确保入驻基地的质量和信息的准确性。

用户行为（针对基地管理员）：

- 访问综合服务平台，选择入驻申请入口。
- 填写入驻申请表，提交基地的基本信息、资质证明和相关资料。
- 等待平台审核结果，并根据审核反馈进行必要的修改和补充。
- 审核通过后，登录平台后台管理系统，发布基地介绍、活动预告和联系方式等信息。

系统响应：

- 提供详细的入驻指南和申请表模板，帮助基地管理员顺利完成入驻申请。
- 对提交的入驻申请进行严格的资料审核，确保基地的合法性和资质的有效性。
- 及时反馈审核结果，对未通过审核的申请给出具体原因和建议。
- 为入驻基地提供便捷的信息发布功能，支持图文、视频等多种形式的内容展示。

---

### 用户故事 8：研学活动预约系统

故事背景：为了方便用户参与研学活动并提前规划行程，项目团队上线了研学活动预约系统。该系统允许用户在线浏览活动详情、选择活动时间和地点，并进行预约操作。

用户行为：

- 登录综合服务平台，进入研学活动预约专区。
- 浏览平台上展示的各类研学活动，通过活动介绍、时间安排和地点信息筛选感兴趣的活
- 点击活动详情页面，查看更详细的活动内容、参与要求和注意事项等信息。
- 选择合适的活动时间和人数，进行在线预约操作，并填写必要的个人信息和联系方式。

系统响应：

- 实时展示最新的研学活动列表，支持按时间、地点和类型等多种方式进行筛选。
- 提供详细的活动详情页面，包括活动介绍、时间安排、地点信息、参与要求等。
- 支持在线预约功能，用户可选择合适的时间和人数进行预约，并实时更新预约状态。

- 发送预约成功通知和提醒信息给用户，确保用户能及时获取活动相关信息并做好准备。

### 用户故事 9：文创产品商城

故事背景：在构建全方位红色党建研学平台的过程中，为了丰富用户体验并增加平台的吸引力，项目决定在综合服务平台中集成文创产品商城。该商城将展示并销售与红色文化相关的文创产品，如纪念品、书籍、服饰等，以满足用户对红色文化纪念品和学习资料的需求。

用户行为：

- 用户进入文创产品商城页面。
- 用户浏览各类文创产品，查看产品详情、图片及价格信息。
- 用户选择心仪的产品加入购物车。
- 用户完成购物车中所有产品的选择和数量的确认。
- 用户选择支付方式进行结算，并填写必要的收货地址等信息。
- 用户提交订单并完成支付流程。

系统响应：

- 系统展示文创产品商城页面，分类清晰，便于用户浏览。
- 系统提供详细的产品信息展示，包括高清图片、详细描述、价格等。
- 系统支持购物车功能，允许用户将多个产品加入购物车，并支持购物车中产品的修改和删除。
- 系统提供结算功能，支持多种支付方式（如支付宝、微信支付等），并自动计算订单总价。
- 系统在用户提交订单后，生成唯一的订单号，并提示用户支付成功或支付失败的信息。
- 系统将用户订单信息保存至数据库，并自动发送订单确认邮件或短信给用户。

### 用户故事 10：评价与反馈系统

故事背景：为了不断提升红色党建研学平台的服务质量和用户体验，项目决定引入评价与反馈系统。该系统将鼓励用户在参与研学活动或体验基地服务后，对活动内容和基地服务进行评价和反馈，从而为平台优化提供数据支持和改进方向。

用户行为：

- 用户完成研学活动或基地服务体验后，进入评价与反馈页面。
- 用户对研学活动的内容、组织、效果等方面进行评价，可以选择星级评分和填写文字评价。
- 用户对基地服务的质量、环境、服务态度等方面进行评价，同样可以选择星级评分和填写文字评价。
- 用户提交评价与反馈信息，并可选择是否匿名提交。

系统响应：

- 系统提供评价与反馈页面，页面布局合理，便于用户操作。
- 系统支持星级评分和文字评价功能，允许用户全面表达自己的评价意见。
- 系统在用户提交评价后，即时将评价信息保存至数据库，并展示在相应活动或基地的评价区域。

- 系统支持匿名评价功能，保护用户的隐私和安全。
- 系统对评价数据进行统计分析，形成评价报告或图表，为平台优化提供数据支持。
- 系统通过评价区域或邮件/短信方式，向用户展示其他用户的评价信息，增加平台的透明度和互动性。

**用户故事 10：**后台管理系统具备数据分析、订单处理、资源调度等能力，支持平台的日常运营和管理。

详细功能请参加文档 1 的需求部分。

新师大软件工程专业



## 文档 3：项目技术梳理

### 一、系统架构

#### 1. 前后端分离

前端：负责用户界面的展示和交互逻辑，支持跨平台访问（如微信小程序、Web 浏览器）。

后端：处理业务逻辑、数据存储和访问控制，确保数据的处理效率和安全性。

#### 2. 服务器架构

采用云服务器部署，如阿里云、腾讯云等，以确保高可用性和可扩展性。

使用负载均衡技术，分散用户请求，提高系统并发处理能力。

部署多节点或集群，实现故障转移和数据备份。

### 二、前端技术

#### 1. 框架选择

React/Vue：选择其中之一作为前端框架。

React Native/Uni-app（移动端应用）：用于开发跨平台的小程序或 APP。

#### 2. 界面设计

使用 UI 框架（如 Ant Design、Vuetify 等）来加速界面开发，确保界面的一致性和美观性。

响应式设计，确保在不同设备（手机、平板、PC）上都能良好展示。

融入党建元素和红色文化风格，提升用户体验。

#### 3. 交互设计

使用 Ajax 或 Fetch 等技术实现前后端数据交互。

集成第三方库（如 Swiper、Video.js）来处理图片轮播、视频播放等功能。

### 三、后端技术

#### 1. 框架选择

- Spring Boot (Java)：适用于大型复杂项目，提供丰富的生态系统和强大的性能。
- Node.js：如果团队对 JavaScript 更熟悉，Node.js 也是一个不错的选择，适合处理高并发 I/O 密集型任务。

#### 2. 数据库选择

MySQL：关系型数据库，适合存储结构化数据，如用户信息、订单数据等。

MongoDB（可选）：非关系型数据库，适合存储大量非结构化或半结构化数据，如日志、文件等。

#### 3. 缓存技术

使用 Redis 等缓存技术，提高数据访问速度，减轻数据库压力。

#### 4. 安全性

实现 HTTPS 协议，确保数据传输安全。

使用 JWT (JSON Web Tokens) 进行用户身份验证和授权。

数据加密存储，防止敏感信息泄露。

定期进行安全审计和漏洞扫描。

### 四、通用技术点

#### 1. 多媒体处理

使用 FFmpeg 等工具处理视频、音频等多媒体文件，支持 VR 全景视频的播放。

#### 2. 地图与定位

集成百度地图或高德地图 API，提供研学基地的地理位置展示和导航功能。

#### 3. 支付与结算

集成第三方支付平台（如微信支付、支付宝），实现在线支付和结算功能。

#### 4. 消息推送

使用 WebSocket 或第三方推送服务（如极光推送、OneSignal），实现实时消息推送功能，如活动通知、订单状态更新等。

### 五、增值功能技术点

#### 1. 利用 VR 技术创建全景漫游体验

技术需求：

- 全景内容制作：拍摄并制作高质量的全景视频或图片，确保画面清晰、拼接无缝。
- VR 播放与渲染：开发或集成 VR 视频播放器和 3D 渲染引擎，支持多种 VR 设备。
- 交互设计：设计用户与 VR 环境的交互方式，如头部追踪、手势识别等。
- 跨平台兼容性：确保 VR 体验能够在不同操作系统和 VR 设备上运行。

具体技术方案：

- 使用专业全景相机（如 Insta360、GoPro Fusion 等）进行拍摄。
- 采用 Unity 或 Unreal Engine 等游戏引擎进行全景内容的制作和渲染。
- 集成 VR SDK（如 Oculus SDK、SteamVR 等）以支持不同 VR 设备。
- 使用 VR 交互库（如 VR Input Module、VR Interaction Framework 等）设计交互方式。

#### 2. 利用 AI 技术创建虚拟讲解员

技术需求：

- 自然语言处理 (NLP)：实现虚拟讲解员与用户之间的自然语言交互。
- 语音合成与识别：将文本转换为语音输出，并识别用户的语音输入。
- 个性化推荐：根据用户行为和历史数据，提供个性化的讲解内容。

- 知识图谱：构建关于红色景点的知识图谱，作为讲解素材的来源。

具体技术方案：

- 使用 NLP 框架（如 TensorFlow、PyTorch 等）训练语言模型。
- 集成语音合成服务（如 Google Text-to-Speech、Amazon Polly）和语音识别服务（如 Google Speech-to-Text、IBM Watson Speech to Text）。
- 开发个性化推荐算法，结合用户画像和协同过滤等技术。
- 利用图数据库（如 Neo4j）构建知识图谱，并开发查询接口供虚拟讲解员使用。

### 3. 通过数据分析工具分析用户行为

技术需求：

- 数据收集：集成 Web 分析工具、API 接口等收集用户行为数据。
- 数据存储：选择适合的数据存储方案，确保数据安全性和可扩展性。
- 数据分析：使用数据分析工具和算法进行数据处理、分析和挖掘。
- 可视化：开发或集成数据可视化工具，将分析结果以图表形式展示。

具体技术方案：

- 使用 Google Analytics、Mixpanel 等 Web 分析工具收集数据。
- 选择云数据库（如 Amazon RDS、Google Cloud SQL）或大数据平台（如 Hadoop、Spark）进行数据存储。
- 使用 Python 数据分析库（如 Pandas、NumPy）和机器学习库（如 scikit-learn、TensorFlow）进行数据分析。
- 集成 Tableau、Power BI 等数据可视化工具或开发自定义可视化界面。

### 4. 基于历史数据和当前趋势预测业务发展趋势

技术需求：

- 时间序列分析：对历史数据进行时间序列分析，识别趋势和季节性变化。
- 预测模型构建：根据历史数据构建预测模型，预测未来业务趋势。
- 情景分析：结合外部环境和内部因素进行情景分析，评估不同情景下的业务影响。
- 决策支持系统：开发决策支持系统，将预测结果和情景分析整合为决策建议。

具体技术方案：

- 使用时间序列分析库（如 statsmodels、Prophet）进行时间序列分析。
- 构建机器学习模型（如 ARIMA、LSTM 等）进行预测。
- 开发情景分析工具或集成第三方工具进行情景分析。
- 开发决策支持系统，结合预测结果和情景分析提供决策支持。可以使用 Python Flask 或 Django 等框架构建 Web 应用，展示预测结果和决策建议。

## 文档 4：“虚拟讲解员”功能的技术路径

### 一、设计思路

虚拟形象设计：设计虚拟讲解员的三维模型和外观，包括服装、发型、面部表情等。

交互流程设计：设计用户与虚拟讲解员之间的交互流程。

(1) 确定各个模块的运行环境和依赖关系。

例如，语音识别和自然语言处理模块可能需要在服务器端进行计算，而语音合成和虚拟形象展示可以在客户端实现。

(2) 设计数据流向和处理流程。

用户的语音输入首先经过语音识别模块转换为文本，然后传递给自然语言处理模块进行理解和分析，生成的回答文本再由语音合成模块转换为语音，最后与虚拟形象的动作和表情同步展示。

例如，在一个博物馆的虚拟讲解员应用中，当游客向虚拟讲解员询问某件展品的信息时，**语音识别模块**快速将游客的语音转换为文字，**自然语言处理模块**理解游客的问题，从数据库中检索出相关的展品介绍文本，经过处理后生成回答文本，再由**语音合成模块**合成为语音并通过扬声器播放，同时虚拟形象做出相应的引导手势和表情，增强讲解的吸引力和生动性。

### 二、开发路径

前端开发：

- 使用 HTML5、CSS3 和 JavaScript 等前端技术，结合 Vue 等现代前端框架，实现虚拟讲解员的界面展示和交互逻辑。
- 集成 WebRTC 或 WebSocket 等实时通信技术，实现前端与后端的实时交互。

后端开发：

- 使用 Spring Boot 等 Java 框架，结合数据库（如 MySQL）和缓存（如 Redis）技术，实现后端服务的开发和部署。
- 开发语音识别（STT）、自然语言处理（NLP）、语音合成（TTS）等 AI 服务接口，并集成到后端服务中。

虚拟形象与动画开发：

- 使用三维建模软件（如 Maya、Blender 等）创建虚拟讲解员的三维模型。
- 利用动画技术为虚拟讲解员添加丰富的表情和动作，并进行实时渲染优化。

知识库与推荐算法开发：

- 构建基于图数据库（如 Neo4j）的知识图谱，存储展品信息、历史文化知识等。
- 开发协同过滤、内容推荐等算法，实现个性化的讲解内容和导览路线推荐。

### 三、所需关键技术和工具

前端技术：HTML5、CSS3、JavaScript、Vue 等前端框架。

后端技术：Java、Spring Boot、MySQL、Redis 等。

三维建模与动画技术：Maya、Blender、Unity、Unreal Engine 等。

AI 技术：语音识别（STT）、自然语言处理（NLP）、语音合成（TTS）等。

实时通信技术：WebRTC、WebSocket 等。

知识图谱与推荐算法：Neo4j、协同过滤、内容推荐等。

#### 四、基于开源项目的技术路径

1. Facemoji：是一款基于 Unity 开发的 **Android 应用**，它包含了两个核心模块：一是脸部追踪技术，二是智能语音对话系统。通过先进的 OpenCV 和 Dlib 库，Facemoji 能捕捉并解析你的面部表情，然后将这些表情栩栩如生地呈现在 Live2D 角色上。这个项目遵循 GPLv3.0 许可协议，任何开发者都可以自由地探索、修改和分发。

前提准备：

- 安装 Unity。
- 了解基本的 Unity 操作和 Android 开发环境配置。

使用步骤：

- 从其开源地址获取项目代码。
- 在 Unity 中打开项目。
- 配置相关的摄像头权限等（如果应用在 Android 设备运行）。
- 确保已经接入了 opencv 和 dlib 库（可能需要手动配置库的引用路径等）。
- 对语音交互模块，如果使用如 turingrobot、iflytek 的 iat（语音识别）和服务 tts（文本转语音）等，需要去相应平台申请账号和配置相关密钥等参数。
- 完成基本设置后进行编译运行在 Android 设备或模拟器上查看效果，调试虚拟形象的面部表情追踪和语音互动。

2. audio2face（FACEGOOD 开源）：是 FACEGOOD 关于数字人的开源项目，其主要实现功能为输入一小段音频，框架的目标是相应地返回一系列连续的嘴唇/面部运动。将注意力机制引入双向 LSTM 网络，将输入音频特征映射到动画参数。

前提准备：

- 基本的机器学习和深度学习知识。
- 有适合运行模型训练和推理的计算机（最好有 GPU）。
- 安装必要的软件依赖如 python 等常见机器学习库等。

使用步骤（如果只是用预训练模型进行推理）：

- 从项目地址：<https://github.com/facegood/audio2face> 克隆代码。
- 按照文档配置好运行环境（包括安装对应的音频处理库等）。
- 准备好音频文件（格式等符合要求）。
- 运行推理脚本，传入音频文件路径等参数，查看生成的表情 blendshape 动画数据（如果要可视化可能还需要导入到支持的三维软件中查看）。

如果要重新训练：

第一阶段：

录制声音数据（如特定的包含各种发音的中文字母表发音、爆破音等文本发音）。

在 maya 中导入录制的声音数据后，根据绑定制作符合模型面部特征的对应发音动画，形成动画数据。

第二阶段：

通过代码实现如文中提到的 lpc 对声音数据处理，将声音数据分割成与动画对应的帧数据以及 maya 动画帧数据的导出。

第三阶段：

将处理后的数据作为神经网络的输入，配置好训练参数等进行训练直到模型的 loss 函数收敛等。

### 3. Awesome Talking Face:

前提准备：

- 有一定编程基础和对深度学习概念的理解。
- 安装好 git 用于克隆代码。
- 准备好合适的运行环境如支持常见深度学习库的 python 环境等，如果要生成 3D 或特定复杂 2D 效果可能还需要相关软件。

使用步骤：

- 从项目地址：<https://gitcode.com/josephpai/awesome-talking-face> 克隆代码。
- 安装所需依赖，如通过 pip 安装项目提到的语音特征提取、面部关键点检测等相关库（声谱图处理库、预训练模型库、openface 或 mediapipe 库等）。
- 准备音频数据和可选的有面部的基础模型（如果要在已有模型上应用）。
- 根据文档和代码示例编写脚本进行音频到面部表情动画的生成流程调用。
- 可以在本地运行脚本查看生成的虚拟形象表情动画是否和语音同步等情况，根据需要调整参数等进行优化。

### 4. Wav2Lip:

简介：Wav2Lip 是一个基于唇形同步的模型，它可以将任意语音转换为特定人物说这段语音的唇形动画。虽然它本身不直接生成面部动画，但唇形同步是虚拟讲解员中非常关键的一部分。

应用场景：适用于需要将语音与特定虚拟形象唇形同步的场景。

### 5. eMotion:

简介：eMotion 是一个开源的面部表情生成模型，它可以根据输入的音频信号生成对应的面部表情动画。该模型结合了音频分析和深度学习技术，能够生成自然流畅的面部表情。

应用场景：适用于需要基于音频生成面部表情动画的虚拟讲解员系统。

### 6. Hugging Face 开源大模型:

简介：Hugging Face 是一个领先的开源自然语言处理（NLP）和机器学习社区，提供了多个预训练的大模型，包括一些可以间接用于虚拟讲解员开发的模型（如文本到语音的 TTS 模型，结合其他面部动画技术使用）。

特定模型：虽然 Hugging Face 直接提供的可能不是完整的虚拟讲解员模型，但其开源生态中的 TTS 模型、语言模型等可以作为开发虚拟讲解员系统的组成部分。

## 7. SadTalker:

一款结合了情感分析与对话生成技术的先进系统，同时也是一款 AI 视频生成工具。

## 8. 其他通用开源大模型:

如 GPT 系列、BERT 等，虽然这些模型主要用于 NLP 任务，但它们可以作为虚拟讲解员系统中处理文本输入（如用户问题）的组件，结合其他面部动画或语音合成技术共同工作。

# 参考 1: 开源项目 SadTalker 和大模型创建虚拟讲解员

## 一、了解 SadTalker 项目

SadTalker 是一种基于隐式 3D 系数调制的 AI 数字人制作技术，它能够通过分析输入的音频信号，驱动 AI 数字人的面部动作，使其表情与语音内容保持同步。相比传统技术，SadTalker 在面部处理方面更加自然、细腻，能够生成更加逼真的虚拟角色表情。

## 二、准备大模型

1. 选择合适的大模型：大模型如 GPT 系列（如 GPT-4）、文心一言大模型或类似的自然语言处理模型，能够提供强大的文本生成和理解能力。这些模型可以作为虚拟讲解人的“大脑”，处理用户输入并生成相应的讲解内容。
2. 模型部署：将选定的大模型部署到合适的计算平台上，确保能够实时处理用户请求并生成高质量的文本输出。

## 三、数据准备

1. 音频数据：收集高质量的音频数据，包括不同语速、语调、情感的讲解语音。这些数据将用于训练 SadTalker 模型，使其能够准确捕捉并模拟讲解人的语音特征。
2. 图像数据：准备虚拟讲解人的图像数据，包括不同角度、表情和动作的面部图像。这些数据将用于构建虚拟讲解人的外观模型。
3. 文本数据：准备用于训练大模型的文本数据，包括讲解稿、问题回答等。这些数据应涵盖广泛的主题和场景，以提高大模型的泛化能力。

## 四、整合 SadTalker 和大模型

1. 接口对接：将 SadTalker 的音频处理接口与大模型的文本生成接口进行对接。当大模型生成讲解文本时，将生成的文本转换为语音信号，并输入到 SadTalker 中生成相应的面部表情和动作。SadTalker 的音频处理接口与大模型的文本生成接口的对接方案如下：

（1）文本生成：使用大模型（如 GPT 系列）的文本生成接口，根据用户输入或预设的主题生成讲解稿或其他文本内容。将生成的文本内容保存为文本文件或直接在内存中处理。

（2）文本转语音（TTS）：调用文本转语音（Text-To-Speech, TTS）服务或库，将上一步生成的文本内容转换为音频信号。这一步是实现 SadTalker 与大模型对接的关键。市面上有许多成熟的 TTS 服务可供选择，如百度 AI 开放平台、阿里云语音合成等。也可以使用



开源的 TTS 库，如 Mozilla 的 DeepSpeech、Google 的 Text-to-Speech 等。

(3) 音频处理：将 TTS 生成的音频信号作为 SadTalker 的输入。SadTalker 的音频处理接口会分析音频信号中的语音内容、发音方式等信息，并生成相应的 3D 面部动作系数。这些系数将驱动虚拟角色的面部肌肉运动，实现表情与音频的同步。

(4) 面部动作生成：SadTalker 根据生成的 3D 面部动作系数，合成虚拟角色的面部表情和动作。生成的面部表情和动作将与音频信号保持同步，形成完整的虚拟讲解人视频。

(5) 注意点：

- 在对接过程中，需要确保 TTS 服务或库生成的音频格式与 SadTalker 支持的音频格式兼容。如果不兼容，可能需要进行音频格式的转换。
- 在对接过程中，可能会遇到各种错误和异常情况，如文本生成失败、TTS 转换错误等。需要设计合理的错误处理机制来应对这些问题。

2. 同步处理：确保音频信号的生成、面部表情和动作的生成以及文本的输出能够保持同步，以提供流畅的虚拟讲解体验。

## 五、训练与调优

1. 模型训练：使用准备好的数据对 SadTalker 和大模型进行训练。在训练过程中，需要不断优化模型的参数以提高其性能。
2. 性能评估：通过测试集评估模型的性能，包括语音合成的自然度、面部表情的逼真度以及文本生成的准确性等。
3. 调优迭代：根据评估结果对模型进行调优迭代，以提高其整体性能。

## 六、应用部署

1. 部署平台：选择合适的平台部署虚拟讲解人应用，如 Web 应用、移动应用或嵌入式设备等。
2. 用户交互：设计用户交互界面，使用户能够方便地输入请求并获取虚拟讲解人的反馈。
3. 实时监控：部署实时监控系统以跟踪应用的运行状态和用户反馈，以便及时发现并修复问题。

## 参考 2：利用开源大模型 audio2face 创建虚拟讲解员

参考视频：Audio2face 驱动定制虚拟人，接入文心一言/大模型，实时语音驱动交互  
[https://www.bilibili.com/video/BV1wa4y1977s/?spm\\_id\\_from=333.337.search-card.all.click](https://www.bilibili.com/video/BV1wa4y1977s/?spm_id_from=333.337.search-card.all.click)

实验目标：

本实验旨在利用开源的 audio2face 大模型，结合其他相关技术，创建一个虚拟讲解员系统。该系统将能够根据输入的音频（用户的语音指令或问题），生成对应的面部动画或表情，并通过屏幕展示给用户，实现更加生动和自然的交互体验。

#### 实验准备：

- 安装 Python 及必要的库，如用于处理音频和视频的库（如 librosa、opencv-python 等）。
- 下载并安装 audio2face 开源大模型，确保能够正常运行。
- 准备用于测试虚拟讲解员系统的音频文件或实时音频输入设备（如麦克风）。
- （可选）准备用于展示虚拟讲解员面部动画的屏幕或投影仪。

#### 实验步骤：

##### 步骤 1：熟悉 audio2face 大模型

- 研究 audio2face 大模型的原理、架构和训练方法。
- 学习如何加载和使用 audio2face 大模型，包括输入音频的格式要求、输出面部动画的数据格式等。

##### 步骤 2：音频处理

- 编写代码，实现音频的录制功能，能够从麦克风实时获取音频数据。
- 实现音频的读取和预处理功能，包括采样率转换、噪声去除等，确保音频数据符合 audio2face 大模型的输入要求。

##### 步骤 3：生成面部动画

- 使用 audio2face 大模型，将预处理后的音频数据转换为面部动画或表情。
- 调整面部动画的参数，如面部表情的夸张程度、动画的平滑度等，以获得更加自然和生动的效果。

##### 步骤 4：展示虚拟讲解员

- 设计或选择一个虚拟讲解员的形象，将其与生成的面部动画相结合。
- 使用屏幕或投影仪展示虚拟讲解员的面部动画，实现与用户的交互。确保面部动画与虚拟讲解员的形象同步且流畅。
- （可选）添加文本到语音转换功能，使虚拟讲解员能够用语音回答用户的问题或提供讲解内容，进一步增强交互体验。

##### 步骤 5：测试和优化

- 测试虚拟讲解员系统的功能，包括音频处理、面部动画生成和展示效果。
- 根据测试结果，对系统进行优化和改进，提高交互的自然性和流畅性。

## 参考 3：基于开源大模型 Facemoji 创建虚拟讲解员

### 一、前期准备

#### 明确需求

- 确定虚拟数字人的应用场景，例如直播、客服、教学等。
- 规划数字人的形象、性格、语言风格等特征。

#### 技术环境搭建

- 安装所需的开发工具，如 Unity 等。
- 配置好相关的开发环境和依赖项。

## 二、模型创建与设计

### 人物建模

- 使用 3D 建模软件（如 Blender、Maya 等）创建数字人模型，包括身体、头部、四肢等。
- 设计数字人的外观细节，如发型、服装、配饰等。

### 面部表情设计

- 在建模软件中为数字人的面部创建骨骼和肌肉系统，以便实现丰富的表情变化。
- 定义不同表情的关键帧和过渡效果。

## 三、导入到 Facemoji

- 将创建好的数字人模型导入到 Unity 中的 Facemoji 项目中。
- 配置 Facemoji 插件
- 设定面部追踪的参数，如敏感度、准确性等。

## 四、语音与表情同步设置

语音输入集成：接入语音识别模块，使其能够接收用户的语音输入。

表情与动作映射：根据语音的内容和情感，设置对应的面部表情和身体动作的映射规则。

调试与优化：进行大量的测试，观察语音与表情、动作的同步效果。调整映射规则和参数，以优化同步的准确性和自然度。

## 五、交互功能开发

对话逻辑设计：编写数字人与用户的对话逻辑和响应规则。

情感识别与反馈：集成情感识别模块，使数字人能够根据用户的情感做出相应的反应。

## 六、性能优化与测试

优化模型和动画：对数字人的模型进行简化和优化，以提高渲染性能；优化动画的播放和过渡效果，减少卡顿。

全面测试：在不同的设备和网络环境下进行测试，确保稳定性和流畅性；收集用户反馈，进一步改进和优化数字人的表现。

## 七、部署与发布

选择部署平台：根据应用场景，选择合适的平台进行部署，如网站、移动应用等。

发布上线：将虚拟数字人发布到目标平台，供用户使用。

## 文档 5：项目技术学习和验证方案

### 一、系统架构

#### 实验 1：前后端分离架构部署

目标：搭建一个简单的前后端分离应用，前端使用 Vue，后端使用 Spring Boot。

步骤：

##### (1) 创建 Spring Boot 项目，设置 REST API、

- 安装 Java 开发环境 (JDK) 和相关的开发工具 (如 IntelliJ IDEA 或 Eclipse)。
- 使用 Spring Initializr (<https://start.spring.io/>) 创建 Spring Boot 项目，选择所需的依赖，如 Spring Web、Spring Data JPA 等。
- 在项目的配置文件 (application.properties 或 application.yml) 中设置数据库连接信息 (如果需要)。
- 创建数据实体类 (Entity)，代表数据库中的表结构。
- 创建数据访问接口 (Repository)，用于与数据库进行交互。
- 创建服务类 (Service)，实现业务逻辑。
- 创建控制器类 (Controller)，定义 RESTful API 的端点，处理 HTTP 请求，并返回相应的数据。

##### (2) 创建 Vue 项目，使用 Axios 进行 API 调用

- 安装 Node.js 和 npm (Node 包管理器)。
- 使用 Vue CLI (<https://cli.vuejs.org/>) 创建 Vue 项目。
- 在项目中安装 Axios 库：npm install axios。
- 创建页面组件 (Component)，用于展示数据和处理用户交互。
- 在组件的方法中，使用 Axios 发送 HTTP 请求获取后端的数据，并进行相应的处理和展示。

##### (3) 部署应用到云服务器，配置跨域访问

- 选择云服务提供商 (如阿里云、腾讯云等)，创建云服务器实例。
- 安装所需的运行环境，如 JDK (后端)、Node.js (前端)。
- 将 Spring Boot 项目打包成可执行的 JAR 文件，并上传到云服务器。
- 在云服务器上运行后端应用：java -jar your-backend-app.jar。
- 将 Vue 项目进行构建 (npm run build)，生成静态文件。
- 配置 Web 服务器 (如 Nginx)，将前端的静态文件部署到云服务器，并设置反向代理指向后端应用的端口。
- 在后端应用的配置中，添加跨域访问的配置，允许前端的请求访问。

##### (4) 测试在不同平台 (微信小程序、Web 浏览器) 的访问

- 在 Web 浏览器 (如 Chrome、Firefox 等) 中输入前端应用的 URL，检查页面是否正常加载，数据是否正确获取和展示，用户交互是否正常工作。
- 如果要在微信小程序中测试，需要先注册微信开发者账号，创建小程序项目。

- 在小程序的代码中，使用小程序提供的网络请求 API 访问后端的接口，检查数据的获取和展示是否正常。

#### 实验 2：服务器架构配置

目标：在云服务器上配置负载均衡和多节点部署。

步骤：

- 在阿里云或腾讯云上部署多个实例。
- 配置负载均衡器，分散用户请求。
- 设置故障转移和数据备份策略。
- 进行压力测试，验证系统并发处理能力。

## 二、前端技术

#### 实验 3：Vue 框架与 UI 框架使用

目标：使用 Vue 和 Vuetify 开发一个具有党建元素和红色文化风格的界面。

步骤：

- 创建 Vue 项目，引入 Vuetify。
- 设计并实现一个响应式的党建主题页面。
- 使用 Axios 技术从后端获取数据并展示。

#### 实验 4：交互设计实现

目标：在 Vue 项目中集成第三方库，实现图片轮播和视频播放功能。

步骤：

- 选择并集成 Swiper 库实现图片轮播。
- 集成 Video.js 库实现视频播放功能。
- 测试在不同设备和浏览器上的兼容性和响应性。

## 三、后端技术

#### 实验 5：后端框架与数据库配置

目标：使用 Spring Boot 和 MySQL 搭建一个简单的用户管理系统。

步骤：

- 创建 Spring Boot 项目，连接 MySQL 数据库。
- 设计并实现用户注册、登录功能。
- 使用 JPA 或 MyBatis 进行 ORM 映射和数据操作。

#### 实验 6：缓存与安全性配置

目标：在 Spring Boot 项目中配置 Redis 缓存和 HTTPS 协议。

步骤：

(1) 集成 Redis，实现用户信息的缓存。

- 在项目的 Maven 或 Gradle 配置文件中添加 Redis 依赖。
- 配置 Redis 服务器的连接信息，包括主机地址、端口、密码等。

- 创建 Redis 缓存管理器 (CacheManager) 和缓存配置 (CacheConfiguration)。
  - 在需要缓存用户信息的服务方法中, 使用注解 (如 @Cacheable、@CachePut、@CacheEvict) 来标识缓存的操作。
  - 测试缓存功能, 观察用户信息的读取和写入是否正确地利用了 Redis 缓存。
- (2) 配置 HTTPS 协议, 确保数据传输安全。
- 生成或获取 SSL 证书, 可以通过自签名证书进行测试, 或者从权威的证书颁发机构购买证书。
  - 在 Spring Boot 的配置文件中, 配置服务器的 SSL 相关属性, 如证书路径、证书密码等。
  - 重启应用服务器, 使 HTTPS 配置生效。
  - 使用工具 (如浏览器或在线检测工具) 验证 HTTPS 连接是否正常, 证书是否有效。
- (3) 实现 JWT 用户身份验证和授权。
- 引入 JWT 相关的依赖库。
  - 创建用户认证服务, 负责生成和验证 JWT 令牌。
  - 在用户登录时, 生成包含用户信息的 JWT 令牌, 并返回给客户端。
  - 在需要授权的接口上, 添加拦截器或过滤器, 验证传入的 JWT 令牌的有效性和权限。
  - 测试用户身份验证和授权流程, 确保只有合法用户能够访问受保护的资源。
- (4) 进行安全审计和漏洞扫描: 安全审计工具 (如 OWASP ZAP、Nessus 等)

#### 四、通用技术点

##### 实验 7: 多媒体处理与地图集成

目标: 在一个 Vue 应用中集成 FFmpeg 和百度地图 API。

步骤:

- 使用 FFmpeg 处理视频文件, 支持 VR 全景视频播放。
- 集成百度地图 API, 展示研学基地的地理位置。
- 实现基于地图的导航功能。

##### 实验 8: 支付与消息推送

目标: 在一个 Vue 应用中集成微信支付和 WebSocket 消息推送。

步骤:

- 集成微信支付 API, 实现在线支付功能。
- 使用 WebSocket 实现实时消息推送功能。
- 测试支付流程和消息推送的实时性。

#### 五、增值功能技术点

##### 实验 9: VR 全景漫游体验

目标: 使用 Unity 和 VR SDK 创建一个全景漫游体验。

步骤:

- 使用专业全景相机拍摄并制作全景视频。

- 在 Unity 中导入全景视频，进行渲染和交互设计。
- 集成 VR SDK，支持不同 VR 设备。
- 测试跨平台兼容性。

实验 10：虚拟讲解员（也可以基于开源大模型来开发）

目标：使用 NLP 和语音合成技术创建一个虚拟讲解员。

步骤：

- 使用 TensorFlow 训练 NLP 模型，实现自然语言交互。
- 集成 Google Text-to-Speech 和 Speech-to-Text 服务。
- 开发个性化推荐算法，提供讲解内容。
- 使用 Neo4j 构建知识图谱，并开发查询接口。

实验 11：用户行为分析与业务趋势预测

目标：使用数据分析工具和机器学习模型分析用户行为并预测业务趋势。

步骤：

- 集成 Google Analytics 收集用户行为数据。
- 选择云数据库进行数据存储。
- 使用 Pandas 和 scikit-learn 进行数据分析和模型构建。
- 开发决策支持系统，提供预测结果和决策建议。



## 文档 6：项目测试方案

### 一、测试目标

确保全方位红色党建研学平台的功能完备性、性能稳定性、用户体验良好性，以及数据安全性和收益分配机制的准确性。

### 二、测试范围

- 功能测试：涵盖所有业务需求细化中的功能模块。
- 性能测试：系统在高并发情况下的响应时间和稳定性。
- 用户体验测试：界面友好性、易用性和互动性。
- 安全性测试：数据保护、用户隐私和权限管理。
- 兼容性测试：不同浏览器、操作系统和设备上的兼容性。
- 收益分配测试：验证收益分配机制的准确性和结算流程的顺畅性。

### 三、测试策略

#### 1. 功能测试

测试用例设计：根据业务需求细化，设计详细的测试用例，包括正常场景和异常场景。

测试执行：

红色党建研学基地与廉政教育基地设计：

- 测试高清图片、视频、VR 全景的加载速度和清晰度。
- 验证研学内容、历史背景、特色活动的准确性和完整性。
- 检查在线导览、虚拟讲解员、互动问答等功能的交互性和准确性。

综合服务平台：

- 验证基地入驻流程的顺畅性和资料审核的准确性。
- 测试研学活动预约、一站式预订、活动管理的功能实现。
- 评估角色体验与互动活动的趣味性和互动性。
- 文创产品商城的展示、购物流程、订单管理功能测试。
- 评价与反馈系统的用户评价、反馈收集、数据分析功能测试。

后台管理系统：

#### (1) 用户管理

- 用户信息管理：测试用户信息的录入、修改、查询等功能，验证用户信息的准确性和完整性。检查用户隐私保护措施是否到位，确保用户信息不被泄露。
- 用户行为追踪：验证用户行为追踪功能的准确性，包括登录、浏览、购买等行为的记录和分析。检查行为追踪数据是否可用于后续的用户需求分析和体验优化。
- 用户权限管理：测试不同用户角色的权限分配情况，确保系统安全。检查权限变更的灵活性和及时性，以适应不同场景下的需求变化。

#### (2) 内容管理

- 内容发布与审核：测试内容发布平台的易用性和审核流程的严格性，确保发布内容的准确性和合规性。

- 内容更新与维护：模拟内容更新和维护的场景，验证系统能否及时、准确地更新研学内容、活动信息和文创产品等。
- 内容分类与标签：检查内容分类和标签的准确性和有效性，确保用户能够方便地搜索和浏览所需内容。

### （3）订单处理

- 订单生成与确认：测试订单生成和确认功能的自动化程度，确保订单信息的准确性和完整性。检查订单确认通知的及时性和准确性。
- 订单状态跟踪：验证订单状态跟踪功能的实时性和准确性，确保用户能够随时查看订单的当前状态。
- 退款与售后：测试退款和售后服务的流程，确保用户权益得到保障。检查退款和售后处理结果的准确性和及时性。

### （4）数据分析

- 用户行为分析：使用数据分析工具模拟用户行为，验证分析结果的准确性和有用性。检查分析结果是否可用于后续的用户需求分析和体验优化。
- 研学效果评估：收集用户对研学活动的反馈和评价数据，验证评估结果的客观性和准确性。检查评估结果是否能为活动改进提供有效依据。
- 业务趋势预测：基于历史数据和当前趋势模拟预测过程，验证预测结果的合理性和准确性。检查预测结果是否能为平台运营和决策提供有力支持。

测试报告：记录测试结果，包括成功项、失败项、问题描述和截图。

## 2. 性能测试

测试场景设计：模拟高并发用户访问、大量数据传输等场景。

测试工具：使用 LoadRunner、JMeter 等工具进行性能测试。

测试指标：响应时间、吞吐量、并发用户数、系统资源占用率等。

测试报告：记录性能测试结果，分析系统瓶颈，提出优化建议。

## 3. 用户体验测试

测试方法：邀请目标用户群体（可以邀请同学或者朋友模拟用户）进行实际操作，记录他们的反馈意见。

测试内容：界面友好性、操作便捷性、内容可读性、互动性等。

测试报告：整理用户反馈，分析用户体验问题，提出改进建议。

## 4. 安全性测试

测试内容：数据加密、用户权限管理、SQL 注入、跨站脚本（XSS）、跨站请求伪造（CSRF）等。

测试工具：使用 OWASP ZAP、Nessus 等工具进行安全扫描和渗透测试。

测试报告：记录安全漏洞和潜在风险，提出修复建议。

## 5. 兼容性测试

测试环境：不同版本的浏览器（如 Chrome、Firefox、Edge、Safari）、操作系统（如 Windows、macOS、iOS、Android）和设备（如手机、平板、电脑）。

测试内容：功能完整性、界面布局、交互效果等。

测试报告：记录兼容性问题和建议解决方案。

## 四、测试计划

测试阶段划分：分为单元测试、集成测试、系统测试和验收测试四个阶段。

测试时间安排：根据项目开发进度和里程碑，合理安排测试时间。

测试资源分配：包括测试人员、测试工具、测试环境等资源的分配。

风险管理：识别测试过程中可能遇到的风险，制定应对措施。

## 五、测试总结

总结报告：汇总各阶段的测试结果，分析测试中发现的问题和解决方案。

遗留问题：记录未解决或需后续跟踪的问题。

优化建议：根据测试结果，提出平台优化建议，以提升用户体验和系统性能。

## 附件：《红色党建研学平台》功能测试用例设计参考

### 一、红色党建研学基地与廉政教育基地设计

#### 1 高清图片、视频、VR 全景加载测试

测试用例编号：T01

测试目标：验证高清图片、视频、VR 全景的加载速度和清晰度。

前置条件：网络连接正常，服务器运行稳定。

测试步骤：

- 在不同网络环境下（如 WiFi、4G/5G）打开图片、视频、VR 全景。
- 记录加载时间。
- 检查图片、视频、VR 全景的清晰度是否符合要求。

预期结果：加载时间合理，内容清晰无模糊。

异常场景：网络中断或低带宽时，应有加载失败提示，并能在网络恢复后继续加载。

#### 2 研学内容、历史背景、特色活动准确性测试

测试用例编号：T02

测试目标：验证研学内容、历史背景、特色活动的准确性和完整性。

前置条件：系统已录入相关数据。

测试步骤：

- 逐一查看研学内容、历史背景、特色活动的描述。
- 比对官方资料或历史文献，验证准确性。
- 检查是否有遗漏的信息。

预期结果：所有信息准确无误，无遗漏。

异常场景：发现错误或遗漏信息，记录并反馈。

### 3 在线导览、虚拟讲解员、互动问答功能测试

测试用例编号：T03

测试目标：检查在线导览、虚拟讲解员、互动问答等功能的交互性和准确性。

前置条件：系统已配置相关功能。

测试步骤：

- (1) 使用在线导览功能，验证路径规划和导航准确性。
- (2) 与虚拟讲解员互动，检查回答内容的准确性和交互流畅性。
- (3) 参与互动问答，验证问题回答的正确性和及时性。

预期结果：功能交互流畅，回答准确无误。

异常场景：交互过程中出现卡顿、错误回答或无法回答，记录并反馈。

## 二、综合服务平台

### 1. 基地入驻流程测试

测试用例编号：T04

测试目标：验证基地入驻流程的顺畅性和资料审核的准确性。

前置条件：系统已开放入驻申请。

测试步骤：

- (1) 模拟基地管理员提交入驻申请，填写相关资料。
- (2) 提交申请后，等待系统审核。
- (3) 验证审核结果的准确性和通知的及时性。

预期结果：流程顺畅，审核结果准确，通知及时。

异常场景：提交不完整或错误资料时，应有明确的错误提示和拒绝机制。

### 2. 研学活动预约与管理测试

测试用例编号：T05

测试目标：测试研学活动预约、一站式预订、活动管理的功能实现。

前置条件：系统已发布研学活动。

测试步骤：

- (1) 用户登录后，选择研学活动进行预约。
- (2) 检查预约流程、支付、确认等环节是否顺畅。
- (3) 管理员登录后台，验证活动管理和订单管理功能。

预期结果：功能实现完整，流程顺畅，无错误。

异常场景：预约超额、重复预约等情况的处理。

### 3. 文创产品商城测试

测试用例编号：T06

测试目标：文创产品商城的展示、购物流程、订单管理功能测试。

前置条件：商城已上架文创产品。
测试步骤：
(1) 浏览商城商品，验证展示效果。
(2) 加入购物车，进行结算和支付。
(3) 检查订单管理功能，包括订单状态、物流信息等。
预期结果：展示效果良好，购物流程顺畅，订单管理准确。
异常场景：库存不足、支付失败等情况的处理。

#### 4. 评价与反馈系统测试

测试用例编号：T07
测试目标：用户评价、反馈收集、数据分析功能测试。
前置条件：用户已完成某项活动或购物。
测试步骤：
(1) 用户提交评价或反馈。
(2) 检查评价或反馈是否成功保存并展示。
(3) 使用数据分析工具，验证分析结果的准确性和有用性。
预期结果：评价和反馈系统运作正常，分析结果准确。
异常场景：评价和反馈意见提交出错等情况。

### 三、后台管理系统

#### 1. 用户管理

##### 1.1 用户信息管理测试

测试用例编号：T08
测试目标：测试用户信息的录入、修改、查询等功能，验证用户信息的准确性和完整性。同时检查用户隐私保护措施是否到位。
前置条件：后台管理系统已登录，具备用户管理权限。
测试步骤：
● 录入新用户信息，包括用户名、密码、邮箱等，并尝试保存。
● 修改已存在用户的信息，并验证修改是否成功。
● 查询用户信息，验证查询结果的准确性和完整性。
● 检查用户隐私保护措施，如数据加密、访问权限控制等。
预期结果：用户信息录入、修改、查询功能正常，隐私保护措施到位。
异常场景：录入非法字符、重复用户名等情况应有明确提示，并阻止保存。

##### 1.2 用户行为追踪测试

测试用例编号：T09
测试目标：验证用户行为追踪功能的准确性，包括登录、浏览、购买等行为的记录和分析。
前置条件：用户已进行一系列操作，如登录、浏览商品、购买文创产品等。
测试步骤：
(1) 登录后台管理系统，查看用户行为追踪数据。

<p>(2) 分析追踪数据，验证其准确性和完整性。</p> <p>(3) 检查行为追踪数据是否可用于后续的用户需求分析和体验优化。</p> <p>预期结果：用户行为追踪数据准确，可用于后续分析。</p> <p>异常场景：追踪数据丢失、错乱等情况应有错误日志记录，并通知管理员。</p>
--

1.3 用户权限管理测试

<p>测试用例编号：T10</p> <p>测试目标：测试不同用户角色的权限分配情况，确保系统安全。</p> <p>前置条件：系统已设置不同用户角色及其对应权限。</p> <p>测试步骤：</p> <p>(1) 登录不同角色的用户账号，尝试访问和操作其权限范围内的功能。</p> <p>(2) 尝试访问和操作超出其权限范围的功能，验证系统是否阻止。</p> <p>(3) 修改用户权限，验证权限变更的灵活性和及时性。</p> <p>预期结果：用户权限分配准确，系统安全性高。</p> <p>异常场景：权限分配错误、权限冲突等情况应有明确提示，并允许管理员及时调整。</p>
---

2. 内容管理

2.1 内容发布与审核测试

<p>测试用例编号：T11</p> <p>测试目标：测试内容发布平台的易用性和审核流程的严格性。</p> <p>前置条件：后台管理系统已登录，具备内容管理权限。</p> <p>测试步骤：</p> <p>(1) 发布新的研学内容、活动信息或文创产品信息。</p> <p>(2) 提交审核，并等待审核结果。</p> <p>(3) 验证审核流程的严格性，确保发布内容的准确性和合规性。</p> <p>预期结果：内容发布平台易用，审核流程严格，发布内容准确合规。</p> <p>异常场景：发布违规内容时应被拦截并通知管理员。</p>
--

2.2 内容更新与维护测试

<p>测试用例编号：T12</p> <p>测试目标：模拟内容更新和维护的场景，验证系统能否及时、准确地更新研学内容、活动信息和文创产品等。</p> <p>前置条件：系统已发布相关内容。</p> <p>测试步骤：</p> <p>(1) 修改已发布的内容，如更新研学活动的时间、地点等。</p> <p>(2) 验证修改后的内容是否及时、准确地展示给用户。</p> <p>(3) 模拟内容维护场景，如删除不再需要的研学内容或活动信息。</p> <p>预期结果：内容更新和维护功能正常，用户看到的信息始终是最新的。</p> <p>异常场景：更新失败、内容丢失等情况应有错误日志记录，并通知管理员。</p>
--

3. 订单处理

3.1 订单生成与确认测试

测试用例编号：T13
测试目标：测试订单生成和确认功能的自动化程度，确保订单信息的准确性和完整性。
前置条件：用户已完成购物或预约活动。
测试步骤： (1) 验证订单生成过程是否自动化，包括商品信息、用户信息、支付信息等。 (2) 检查订单确认通知的及时性和准确性，确保用户收到正确的订单信息。
预期结果：订单生成和确认功能正常，信息准确无误。
异常场景：订单信息错误、支付失败等情况应有明确提示，并允许用户重新操作。

3.2 订单状态跟踪测试

测试用例编号：T14
测试目标：测试系统是否能够准确跟踪和显示订单的各个状态（如待支付、已支付、待发货、已发货、已完成、已取消等），确保用户和管理员能够实时了解订单状态。
前置条件：系统已生成并保存了多个处于不同状态的订单。
测试步骤： (1) 登录后台管理系统，进入订单管理界面。 (2) 查看所有订单的列表，验证每个订单的状态是否正确显示。 (3) 选择一个或多个特定状态的订单，进行详细查看，验证订单详情中的状态信息是否一致。 (4) 模拟订单状态的变更（如手动更改订单状态或模拟用户支付、发货等操作），验证系统是否能够及时更新订单状态。 (5) 通知用户订单状态变更的功能测试，确保用户能够及时收到订单状态变化的通知（如邮件、短信或 APP 推送）。
预期结果： 订单管理界面能够准确显示所有订单的当前状态。 订单详情页面中的状态信息与订单列表中的状态信息一致。 订单状态变更时，系统能够及时更新并通知相关用户。
异常场景： 订单状态更新失败时，应有错误日志记录，并通知管理员。 通知用户订单状态变更的功能失败时，应有重试机制或手动通知的备选方案。