

# 博物馆线上服务系统的设计与实现

## 项目介绍

随着经济的快速发展和网络信息化的进步，互联网在不知不觉中改变了人们的生活模式。博物馆网站作为实体博物馆的一个重要补充，有着不可比拟的优越性。同时，由于我国各地经济、文化发展并不平衡，各地博物馆网站质量参差不齐。针对以上现状与问题，以长兴博物馆为例，对其进行线上网站的设计与实现。

本文对博物馆线上服务系统这一内容进行研究，在对系统进行可行性分析与需求分析后，将系统划分为展示，公共服务与网站内容管理三个模块。在系统设计与搭建上，力求具有良好的视觉体验、丰富的呈现效果，在增强用户体验的同时进一步满足公众多层次需求。

3D 藏品与虚拟展览是本系统的一个亮点之一，公众可在本系统进行 3D 藏品与虚拟展览的浏览，用户可通过点击、拖动等方式对藏品与展览进行零距离体验。在本文的第六章中就其实现与设计进行了具体的介绍。

本系统采用 B/S(浏览器/服务器)架构，基于 Python 语言，MySQL 数据库及 Django 框架开发完成。在前端中，采用 Bootstrap 作为前端框架，同时以 HTML、CSS、JavaScript、Ajax 等技术作支撑。在三维呈现中，采用 WebGL、Three.js 等技术。

最后，系统通过功能性与非功能性测试，确保了系统的正常运行。并对现阶段工作与进一步工作进行总结与展望。

## 目 录

1	绪论 .....	1
1.1	系统开发背景.....	1
1.2	系统开发意义.....	1
1.3	系统内容研究.....	3
1.3.1	系统开发的研究内容.....	3
1.3.2	系统开发的设计原则.....	3
1.4	论文的组织结构.....	4
2	相关技术介绍.....	6
2.1	Python 语言 .....	6
2.2	Django 框架 .....	6
2.3	MySQL 数据库 .....	8
2.4	WeBGL .....	8
2.5	Three.js.....	9
3	系统分析.....	10
3.1	系统可行性分析.....	10
3.1.1	技术可行性.....	10
3.1.2	经济可行性.....	10
3.1.3	操作可行性.....	10
3.2	系统需求概述.....	10
3.3	系统功能性需求分析.....	11
3.3.1	游客角色功能需求.....	11
3.3.2	用户角色功能需求.....	12
3.3.3	管理员角色功能需求.....	13
3.4	系统非功能性需求分析.....	14
4	系统设计.....	16
4.1	系统架构设计.....	16

---

4.2 系统模块设计.....	17
4.2.1 展示模块.....	17
4.2.2 公共服务模块.....	18
4.2.3 网站内容管理模块.....	18
4.3 数据库设计.....	19
5 系统实现.....	26
5.1 平台架构.....	26
5.1.1 开发环境.....	26
5.1.2 创建流程.....	26
5.1.3 项目结构.....	28
5.2 展示模块.....	28
5.2.1 博物馆展览模块.....	28
5.2.2 博物馆馆藏模块.....	30
5.2.3 博物馆首页模块.....	32
5.3 公共服务模块.....	36
5.3.1 注册、登录模块.....	36
5.3.2 博物馆资讯模块.....	38
5.3.3 博物馆活动模块.....	39
5.3.4 博物馆服务模块.....	41
5.3.5 文创商城模块.....	41
5.3.6 预约模块.....	44
5.3.7 用户中心模块.....	45
5.4 网站内容管理模块.....	48
5.4.1 预约管理.....	49
5.4.2 展览管理.....	50
5.4.3 藏品管理.....	51
5.4.4 活动管理.....	52
5.4.5 文章管理.....	53

---

5.4.6 商品管理.....	53
5.4.7 订单管理.....	54
5.4.8 用户管理.....	55
5.4.9 消息反馈管理.....	55
6 3D 藏品与虚拟展览的设计与实现.....	56
6.1 3D 藏品.....	56
6.1.1 创建 HTML 页面框架 .....	56
6.1.2 基本的 Three.js 程序的搭建.....	56
6.1.3 外部模型的导入.....	58
6.1.4 添加灯光与阴影.....	59
6.1.5 添加交互事件.....	60
6.1.6 3D 展示后台管理.....	61
6.2 虚拟展览.....	62
6.2.1 创建 HTML 页面框架 .....	62
6.2.2 基本的 Three.js 程序的搭建.....	62
6.2.3 展厅环境的构建.....	62
6.2.4 添加展览作品.....	64
6.2.5 添加交互事件.....	64
7 系统测试.....	66
7.1 系统测试环境.....	66
7.2 功能性测试.....	66
7.3.非功能性测试.....	69
8 总结与展望.....	71
8.1 现阶段工作总结.....	71
8.2 进一步工作展望.....	71

---

# 1 绪论

## 1.1 系统开发背景

随着互联网与大数据时代的到来,博物馆网站的建设已成为越来越多的博物馆推进信息平台建设的一种选择。特别是在 2016 年之后,国家文物局、国家发展和改革委员会、科学技术部、工业和信息化部、财政部发布《“互联网+中华文明”三年行动计划》之后,博物馆的业态也正在发生着变化,博物馆“互联网+”已成为博物馆领域探讨的重要课题。

目前,我国各大中型主要博物馆均已建立了自己的网站门户,甚至运用了大量新兴的科技手段来呈现藏品和场馆的信息。但现阶段由于我国各地经济、文化发展并不平衡,导致各地的博物馆网站质量参差不齐。特别是对于处在非一线城市且数量巨大的中小型博物馆来说,网站存在流于形式、毫无创意、设计简陋等问题。

本课题针对以上现状与问题,以长兴博物馆(浙江省湖州市长兴县博物馆)为例,对其进行线上网站的设计与实现。力求网站具有良好的视觉体验、丰富的呈现效果,在增强用户体验的同时进一步满足公众多层次需求,使长兴博物馆能够跨地域、跨时空发挥更大作用。

## 1.2 系统开发意义

博物馆网站的建立是博物馆信息化的表现之一。其内容一般包括展览信息,博物馆资讯,博物馆活动、馆藏展示、门票预定等功能,以博物馆本身内容为信息载体,在互联网中发布与管理信息,让博物馆发挥着更大作用,进一步更新服务模式、提升服务质量。具体意义如下:

### (1) 丰富博物馆资源,满足公众需求

网上博物馆的建立,将展览信息,博物馆信息,最新动态,展品赏析等等通过数字化的方式传递给公众,在帮助实体博物馆更全面地展示馆藏资源的同时,将一些因为场地、时间等因素未能在实体博物馆展示的文物,通过网站进行

展示，弥补实体博物馆的不足。同时，在互联网上能够借助图像、声音、动画、甚至 3D 技术等多种手段来展示展品或者展览，为公众提供了更为丰富的资源与服务，满足公众的多重需求。

### **(2) 不受时间、地域的限制，进一步扩大传播范围**

与实体博物馆的不同，线上博物馆将不受时间与空间的限制，为公众带来便捷。根据各博物馆的公开信息可以看到，多数博物馆的开放时间均为白天，虽周末开放，但却不能满足公众随时随地参观的需求，网上博物馆的建立便可解决这一问题。此外，针对一些行动不便、其他地域希望了解博物馆的人群，也可通过网上博物馆获取相关的信息与服务，提高博物馆的传播范围。

### **(3) 促进实体博物馆人数的增加**

根据国外研究调查，在线博物馆的建立实际上推动了实体博物馆的运营效果。在这项研究中，Kravchyna 和 Hastings(2002)发现，57%的博物馆网站用户在参观实体博物馆之前和之后都访问了博物馆网站。同样，托马斯和凯里（2005）发现，70%的博物馆游客在参观博物馆之前专门寻找在线信息，57% 的人说他们在网上发现的信息增加了他们参观博物馆的欲望。为此，建立网上博物馆，对于增加客流量有着一定的促进作用。

### **(4) 收集数据资源，改进服务措施**

在公众对博物馆网站进行访问之后，可进一步实现访客信息数据的可视化。通过对访问的数据进行统计分析，可以获取到网站访问人数、访问时间、停留时间等数据信息，进一步了解公众对于什么类型的展览更感兴趣，对什么活动更想参与等信息。同时，公众通过网站可以反馈自己的一些建议与想法，为博物馆提升自身服务提供可靠依据。

### **(5) 推动博物馆文化产业的发展，带来经济效益**

博物馆本身为公益组织，但可通过线上文创商城的构建，为其带来经济效益，进而回馈大众进行博物馆建设。近些年来，故宫博物院在文创领域取得了巨大成功，其开发出的多款产品，备受追捧。截至 2016 年年底，故宫博物院的文创产品共有 9170 种，在 2017 年，线上淘宝网店收入近 5000 万元。博物馆可借鉴故宫博物院在文创商城建设上的有用经验，搭建文创平台，提高自身的自主造血

能力。

## 1.3 系统内容研究

### 1.3.1 系统开发的研究内容

本系统以长兴博物馆为例，搭建博物馆线上服务平台。根据功能的不同，将系统功能划分为展示、公共服务、网站内容管理三大板块。具体研究的内容如下：

#### (1) 展示

作为实体博物馆的有效补充，博物馆线上服务平台应对实体博物馆的相关展览、藏品等内容进行相关的展示与呈现。包括一些未能在实体博物馆展示的展览藏品等。此外，还应利用虚拟技术搭建三维展览与藏品，丰富展示形式，提升用户体验。

#### (2) 公众服务

公众服务是线上服务平台的另一核心部分。系统应将博物馆开放信息、简介、相关新闻资讯等进行详细的呈现。提供线上预约，活动报名、文创商品购买、意见反馈等功能。同时，设置用户个人中心，方便用户查看与管理相关信息、更好提供用户服务。

#### (3) 网站内容管理

管理员可登录网站后台对系统所有信息进行管理与维护。包括预约、展览、藏品、活动、资讯、商品、订单、用户、消息反馈等信息，可实现查看、添加、删除、修改等功能。

### 1.3.2 系统开发的设计原则

#### (1) 服务性

博物馆网站系统其根本目的是为了给公众提供良好的服务。在设计系统时，坚持服务公众是首要的。服务性应以用户为中心，把用户所需放在首位，尽可能的考虑到用户的需求，而不是从博物馆本身的角度向公众展示博物馆规模、研究成果、地位等信息。积极将博物馆的展览信息、活动信息、开放信息等公众希望了解的信息传递给公众，让公众可以以最快速度获取所需信息。

#### (2) 互动性

网站不仅仅是简单的信息展示,设计系统时,应充分利用不断发展的网络技术,提高互动性。比如公众可对活动进行报名、对藏品进行收藏等操作,并可在平台进行用户注册、登录,通过留言板块及时反馈自己的意见等等,通过加强与公众的交流的形式,拉近公众与博物馆的距离。

### **(3) 动态性**

随着网络技术的发展以及宽带的逐步普及,常规的静态网页已经不能满足各方面的需求,为此,还需在动态性下功夫,做到信息内容不断更新。网页显示内容可以随着用户、环境或者不同操作的结果而发生改变的。

### **(4) 创新性**

人们生活水平的提高,使公众对于网站内容也提出了更高的要求。博物馆网站不应仅仅只停留在简单的平面展示,更应与时俱进,向国内外优秀博物馆网站学习与借鉴,通过 3D、虚拟动画等技术呈现藏品和展览的信息,使观众与藏品之间有更为近距离的接触,带来不一样的非凡体验。

### **(5) 丰富性**

博物馆网站上的内容要做到内容充实,能够全方位、多维度地展示博物馆的资源。基于博物馆系统服务的功能,努力丰富网站栏目,拓展各栏目内容。展览藏品的呈现形式尽可能丰富多彩,充分发挥多媒体的表现力,尽可能增加动画、影视和声音等媒体。

### **(6) 美观性**

页面设计是公众浏览网站的第一直观印象,良好的视觉体验才能激发公众继续浏览网页的欲望。统一网站风格,做到色彩搭配合理,符合大众审美需求,做到简约大方,设计出符合用户的使用习惯,兼顾良好的用户体验的博物馆网站。

## **1.4 论文的组织结构**

本论文共分为七个章节,各章节具内容与安排如下:

第一章绪论,此章节主要介绍本系统基于怎样的背景下进行开发的,开发此系统又有何意义。同时,针对本系统需要研究的内容进行了概括与介绍,进一步明确了系统设计的原则。



第二章相关技术介绍，技术是系统搭建的基石。在这一章节中就系统涉及的相关基础技术如 Python 语言、Django 框架、MySQL 数据库进行表述，同时对三维虚拟画面涉及的 WebGL、Three.js 的技术进行详细的介绍。帮助读者进行更好的了解。

第三章系统分析，首先从系统的技术可行性、经济可行性与操作可行性三个方面对系统进行可行性分析，并对系统需求进行简要的概述。对系统的功能性与非功能性两部分需求进行论述，确定系统所需功能与模块。

第四章系统设计，根据系统分析的结果，对系统进行概要设计。对系统采取的架构进行简要的介绍与涉及，并对系统各部分功能模块进行了划分与阐述，同时对数据库进行了设计。

第五章系统实现，在对模块划分完成之后，对系统各模块进行详细的设计的描述。

第六章 3D 藏品与虚拟展览的设计与实现，分别就 3D 藏品的设计与实现、虚拟展览的设计与实现进行了详细的介绍。包括如何搭建基本的 three.js 程序、添加灯光、交互事件等内容。

第七章系统测试，介绍系统测试的相关环境，从系统各模块入手，记录测试的数据，通过测试来检验系统各模块功能是否满足要求。并从安全性、兼容性、可拓展性、性能这几方面对系统进行非功能性测试。

第八章总结与展望，此章节就此系统现阶段设计与成果进行总结与归纳，就系统存在的不足进行下一步工作的展望。

## 2 相关技术介绍

### 2.1 Python 语言

Python 是一种动态的、面向对象的计算机程序设计语言。它于 1989 年底被吉多·范罗苏姆发明，1991 年首次发布。作为一门免费开源的脚本语言，Python 在人工智能、数据处理、网络编程、web 开发等诸多领域都有着十分广泛的应用，也因其语法简单、功能强大而备受大家的欢迎，成为使用最多的语言之一。

具体具有以下几个特点：

- (1) 简单易学：Python 相比于其他的语言而言，语法更为简单轻便，对于初学者而言，几乎没有什么难度。在阅读 Python 程序时，可以更专注于解决问题本身而不是理解程序自身的语法语义。
- (2) 交互性强：虽然 Python 语法简单，但却自带着庞大的库类，具有很强的交互性。这些库类能帮助编程人员在编写程序时处理各种工作。如 Scrapy 库，可用来处理爬虫相关工作，Requests 库用来处理与 http 相关的工作等。
- (3) 面向对象：Python 支持面向对象，相对于面向过程在功能统一的同时也保证了面向对象设计的可扩展性，维护也更为简单，更为利于代码的重构。
- (4) 可移植性：由于 Python 免费开源且因 Python 解释器的存在，经过编写后的代码不需要经过特别的修改便可跨平台运行。目前大部分的主流系统对 Python 语言都提供了支持，其中在一些新版本的 Linux 系统中特别内置了 Python 虚拟机用来执行 Python 代码。

除了以上提到的几个比较重要的特点外，Python 语言还具有可拓展性、可嵌入性、代码规范性等特点。为此，选择该语言作为系统的开发语言。

### 2.2 Django 框架

Django 框架是由 Python 语言编写而成，采用模型、模板、视图的 MTV 模式的开源 Web 应用框架。起源于 2005 年的一个在线新闻的 web 站点网站，如今

已越来越被诸多系统所应用。

下面，对 Django 框架的核心组件进行介绍。

**模型（Model）：**模型是一个定义数据源的数据，与数据库联系，每一个模型对应一个数据表。模型通常在 `models.py` 文件中实现，一般包含数据的属性与行为。

**视图（View）：**是一个 Python 函数或者类。一般是完成逻辑相关的操作，用来接受前端发出请求并且返回相关的响应。

**模板（Template）：**所谓模板其实就是一个 html 格式的文本，将文档的表现形式和内容进行分离，实现数据动态化并可被 Django 编译。即对于相同页面的不同内容，只需一个 html 便可完成，在很大程度实现了减少了代码量，实现代码的精简化。

**ORM：**即对象关系映射，在 Django 中通过类的方式去来操作数据库，从而避免使用 SQL 语句时产生的 SQL 语句重复利用率不高、SQL 注入等问题。使用 ORM 模型，可更直观、清晰地对数据库进行操作。此外 Django 支持 MySQL、PostgreSQL 和 SQLite 等多个流行关系数据库引擎，可轻松的实现可数据库的切换。

**URLconf：**即路由系统，是 URL 路径表达式与 view（视图）函数之间的映射，这一路由系统相当于整个网站的目录。URLconf 使用正则表达式的方法来匹配 URL，并告诉 Django 这个 URL 应调用的视图函数是哪一个。

**表单处理：**与前端的 html 的表单不同，Django 中的表单除了是用来提交数据以外，还起到了渲染表单模板与验证数据是否合法的作用。

其框架基本架构如图 2-1 所示。

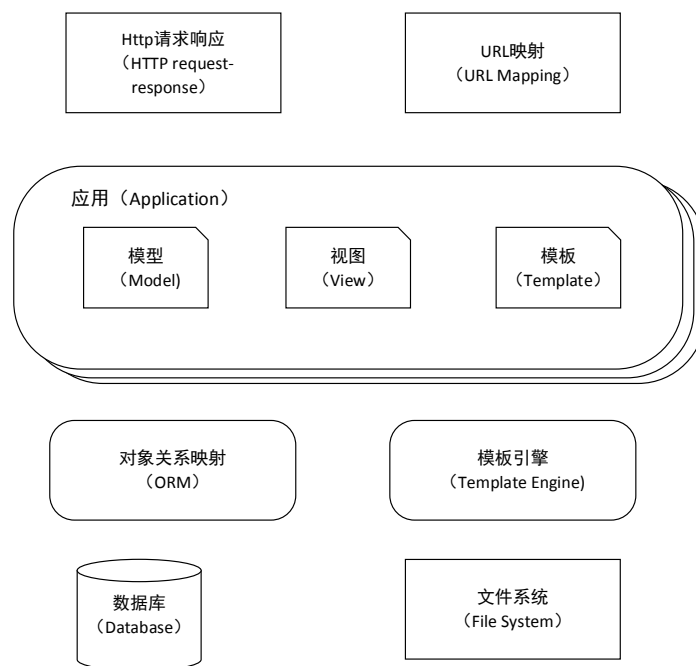


图 2-1 Django 框架基本架构

## 2.3 MySQL 数据库

MySQL 是由瑞典 MySQL AB 公司开发是一个关系型数据库管理系统，为时下最流行的关系型数据库管理系统之一。其具有性能稳定、容易操作、免费开源、可移植、支持多线程等诸多优点，已成为 Web 开发的首选。此外，MySQL 在管理上十分灵活，可通过命令行工具、图像管理工具、GUI 管理工具进行数据库的管理与操作。

## 2.4 WebGL

WeBGL 是由 JavaScript 与 OpenGL ES 2.0 相结合的 3D 绘图标准，能够帮助 HTML5 的 Canvas 标签渲染出 3D 效果，满足 web 端展示 3D 场景的需求。其具有以下特点：

- (1) 无需安装插件：与传统的动态网页相比，WeBGL 网页除了包括 HTML 与 JavaScript 两种语言，还包括 OpenGL ES 2.0 语言。但其实 OpenGL ES 2.0 语言也是由 JavaScript 所编写，所以 WeBGL 网页只是相对于更为复杂，但却保持这和传统的动态网页相同的结构。为此，无需安装任何插件，只需浏览器便可运行。
- (2) 跨平台运行：正是因为 WeBGL 网页在浏览器无需插件便可运行，所

以对与平台没有任何限制，可在台式电脑、笔记本、平板、手机等设备跨平台运行，具有开放性。

- (3) 更为丰富的展示：相对于早期的 3D 呈现技术，能更好、更顺畅的展示三维场景，对构建更为复杂的模型或场景有很大帮助。

## 2.5 Three.js

Three.js 是目前流行的 WebGL 框架之一，属于 JavaScript 库的一种。其通过对 WebGL 底层的封装，很多原本需要很多代码的实现功能，在 Three.js 中只需要几行代码便可完成。同时，提供了很多 3D 显示的功能，帮助更好实现三维场景的渲染。具体特点如下：

- (1) 节省开发时间：WebGL 虽然提供了大量且丰富的接口用来创建三维场景与物体，但其开发过程与 C 语言类似，这使得直接使用原生接口会变得复杂且枯燥。Three.js 对 WebGL 的接口进行了封装，抽象出场景、照相机、渲染器等对象，能大大减少工作开发量。
- (2) 功能丰富：通过 Three.js 能帮助我们构建更为复杂的三维场景，可用于构建三维动画、生成全景场景等。
- (3) 易于学习：Three.js 是网上 GitHub 的一开源项目，提供了完备的官方文档、案例、源代码等信息，代码简单易读，非常适合初学者学习。

## 3 系统分析

### 3.1 系统可行性分析

为了确定本系统是否能按照预期计划完成，现就系统的技术可行性、经济可行性、操作可行性作分析。

#### 3.1.1 技术可行性

系统采用 Python 语言，Django 框架进行编写。虽然 Python 语言之前并没有接触过，但考虑到 Python 语言可以应用到更多的领域，较为容易上手，且在网站制作上也已经十分成熟，故选用 Python 语言开发。在搭建框架上，Django 具有比较强大与丰富功能，内建模板、表单、路由、认证、基本数据库管理等模块，为此可作为一个大型网站的搭建框架。此外，数据库选用 MySQL 即可满足系统的需求与功能。因此基于以上技术的支撑，完成系统的搭建具有技术可行性。

#### 3.1.2 经济可行性

系统只需一台主机便可进行开发与部署，采用相关软件与技术均为开源，不存在经济方面的开销。

#### 3.1.3 操作可行性

本系统在任何系统下均为可以运行，兼容性良好。且系统页面设计友好，系统模块清晰、操作简单易懂，只需点击鼠标便可完成相关操作。即使未接受过计算机相关训练的，无计算机基础的人员也可进行相关操作。

### 3.2 系统需求概述

博物馆线上服务系统作为实体博物馆的一个重要补充，其建立的根本目的就是能更好地为了公众所服务。无论在功能上的设计与非功能的设计都需考虑公众的真实需求，竭尽所能地为公众提供最优质的服务。

公众可以通过系统获取博物馆的相关信息，如展览信息、活动信息、藏品信息、博物馆开放信息等。同时，也应满足公众更多元化的需求，为公众提供便捷，比如可直接线上预约订票、对博物馆进行意见反馈等功能。

同时，公众可注册成为系统用户，进行线上购物、活动报名、藏品收藏等，

体验更多系统内容与资源。

系统管理员作为系统的管理者，可对平台上的展览、文章、活动、藏品、文创商品、订单、注册用户等进行查看、发布、删除、修改等操作，可及时了解与掌握整个系统的动态信息。

### 3.3 系统功能性需求分析

通过上一小节的描述，我们可以很清晰地把系统用户的角色分为三类，分为是游客、注册用户与管理员。下面将对各角色的功能需求进行需求分析。

#### 3.3.1 游客角色功能需求

未在系统注册的用户统称为游客，主要有以下需求：

- （1）注册：游客可在系统任意界面上方点击注册按钮进行注册。
  - （2）展览模块：获取博物馆目前正在展出的展览信息，包括展厅信息等等，同时，可以进行虚拟展厅的体验。
  - （3）活动模块：了解博物馆即将进行的活动，并可查看详细。
  - （4）资讯模块：对博物馆本身简介进行了解，并可浏览近期的博物馆新闻。
  - （5）文创商城模块：对文创商品进行点击浏览。
  - （6）馆藏模块：对博物馆藏品进行了解，并可进行 3D 藏品的查看与体验。
  - （7）服务模块：就博物馆相关开放信息与提供服务进行了解，可直接反馈问题与意见。
  - （8）预约模块：可直接预约博物馆门票，分团体预约与个人预约。
- 通过以上描述，游客角色的用例图如图 3-1 所示。

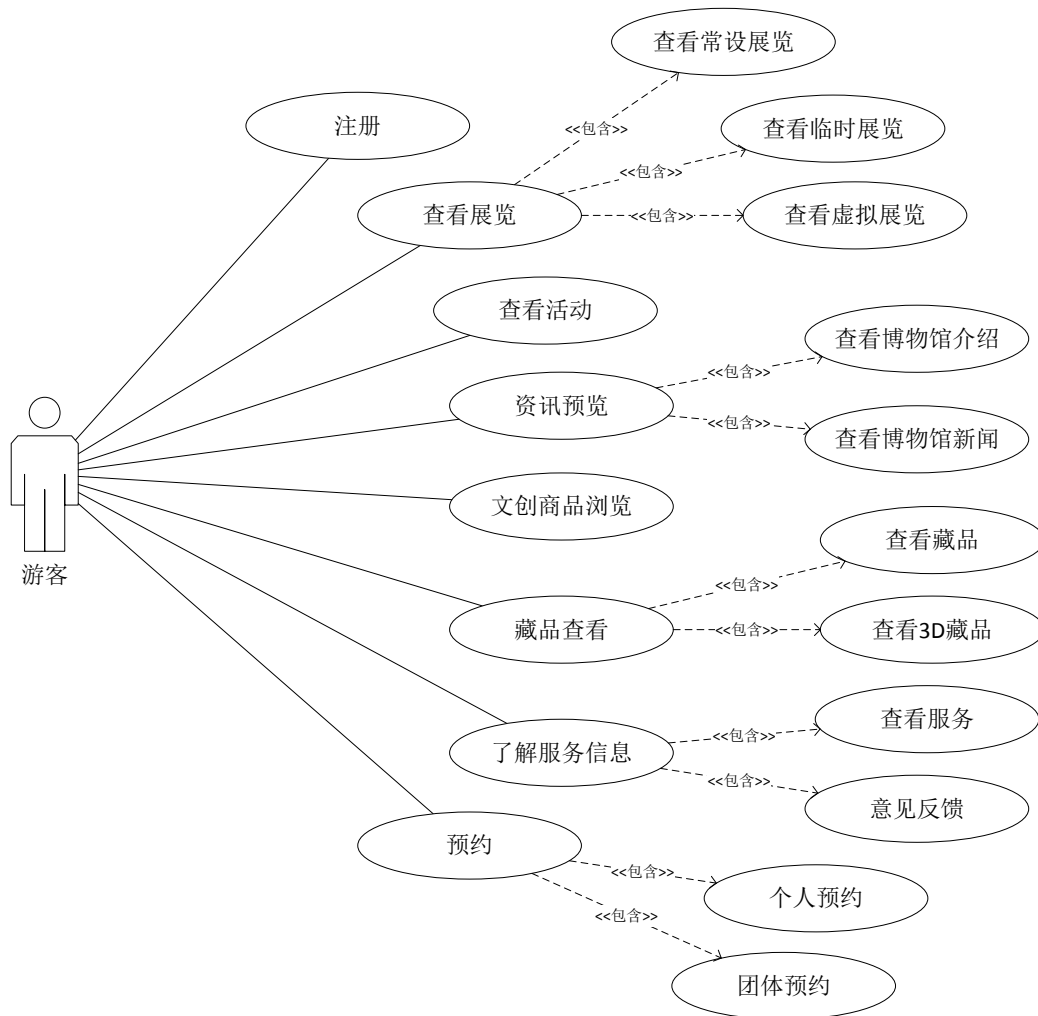


图 3-1 游客角色用例图

### 3.3.2 用户角色功能需求

用户角色的权限在游客基础上有所扩大，游客拥有的功能用户也全部具有，除此以外用户还拥有以下功能。

(1) 登录：用户可进行登录操作。

(2) 用户中心：包括预约管理、报名/收藏管理、订单管理、用户信息管理几个部分，可进行相关管理操作。

(3) 文创商城模块：可对心仪商品加入购物车、形成订单，提交订单并完成购买流程。

(4) 馆藏模块：对博物馆藏品进行收藏。

通过以上描述，用户角色的用例图如图 3-2 所示。



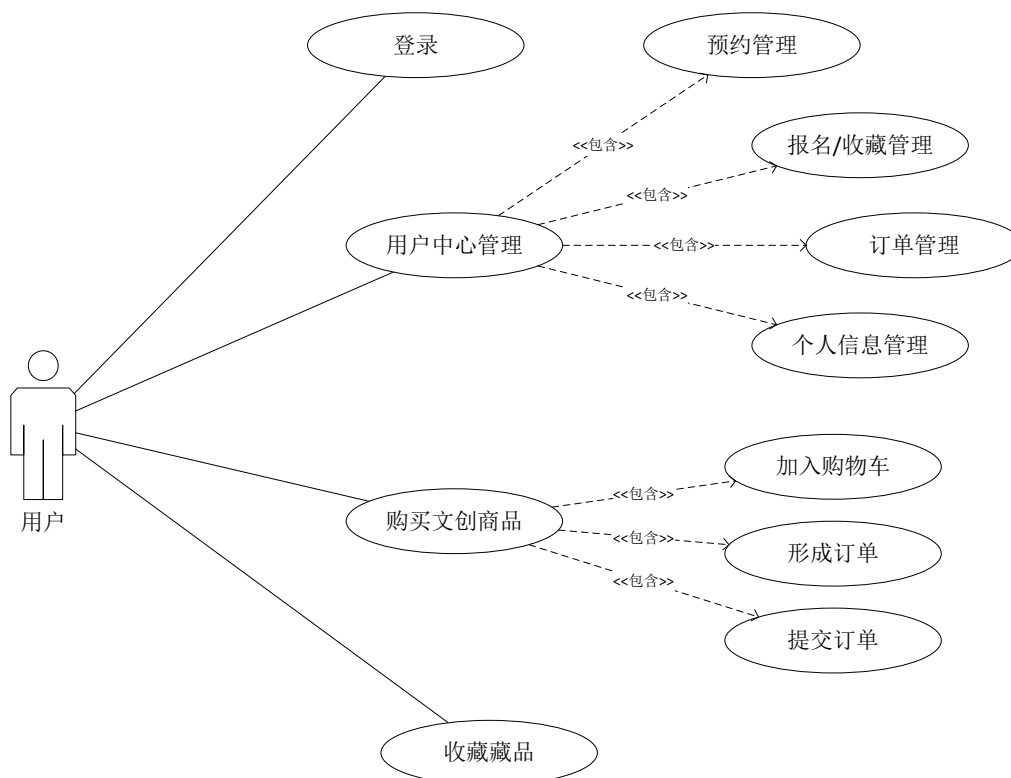


图 3-2 用户角色用例图

### 3.3.3 管理员角色功能需求

管理员是整个系统内容、信息的管理者，拥有着系统的最高权限，具体的功能需求如下：

- (1) 预约管理：对个人与团体的预约记录进行查看、搜索操作。
- (2) 展览管理：对展览、展览展区进行添加、查看、修改、删除、搜索等操作。并可对展览分类进行添加、查看、修改与删除。
- (3) 活动管理：对活动进行添加、查看、修改、删除、搜索等操作。并可对活动分类进行添加、查看、修改与删除。
- (4) 资讯管理：对新闻进行发布、查看、修改、删除、搜索等操作。并可对新闻分类进行添加、查看、修改与删除。
- (5) 商品管理：对商品进行添加、查看、修改、删除、搜索等操作。并可对商品分类进行添加、查看、修改与删除。
- (6) 订单管理：对系统生成订单进行查看、删除、状态变更等操作。并可查看每一订单的详细信息。
- (7) 馆藏管理：对馆藏、3D 馆藏进行添加、查看、修改、删除、搜索等操

作。

(8) 用户管理：可查看注册用户的具体信息，并可完成修改其相关信息，注销用户、添加用户等操作。

(9) 消息反馈管理：就用户所提意见进行查看与搜索。

通过以上描述，管理员角色的用例图如图 3-3 所示。

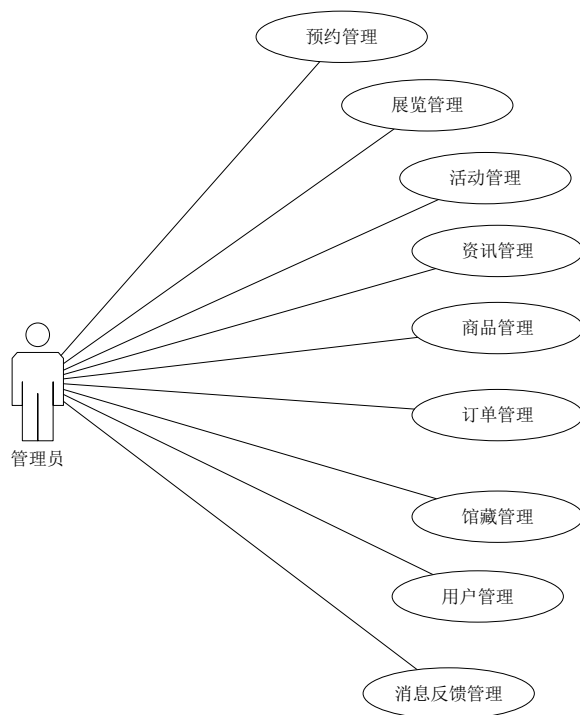


图 3-3 管理员角色用例图

### 3.4 系统非功能性需求分析

为了给用户带来更好的用户体验，谋求更高的系统质量，除了在系统功能性进行需求分析以外，还需关注非功能性的需求。主要包括安全性、兼容性、可扩展性等方面。

#### (1) 安全性需求

系统安全性主要包括自身安全与用户信息安全两方面。一方面可以通过设置管理员权限，规定只允许管理员可以修改系统信息来保证，另一方面对用户的注册信息进行加密，防止用户信息泄露。

#### (2) 兼容性需求

本系统应兼容市面上的任一主流系统，包括 win10、win8、win7、苹果系统

等等，在任一浏览器均可浏览。

### （3）可拓展性需求

系统在面对新的需求功能时，可在原先基础系统上进行拓展。

### （4）性能需求

响应时间的长短是性能评价的标准之一，在用户在系统做出相应操作时，系统应尽快缩短响应时间，保证数据传递顺畅，提高运行效率，尽可能控制在 3 s 以内，并做到 7×24 小时不间断服务，提升用户的体验感。

## 4 系统设计

### 4.1 系统架构设计

系统基于 B/S（浏览器/服务器）模式进行架构搭建，只需安装有浏览器 (Browser)，用户就可以通过浏览器提出服务请求。整个系统遵循 B/S 的三层体系结构，分别为界面层、业务逻辑层、数据访问层，如图 4-1 所示：

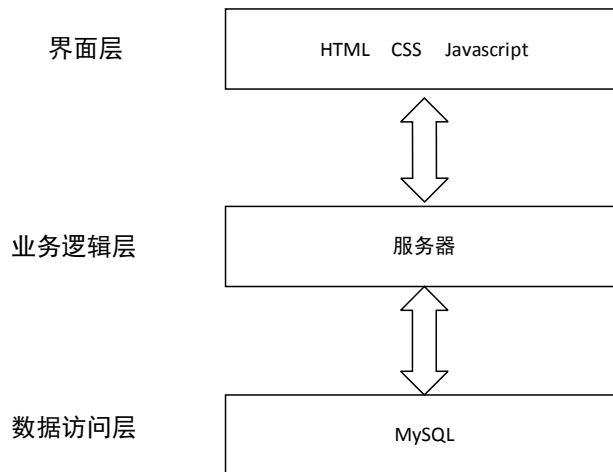


图 4-1 B/S 的三层体系结构

**界面层：**界面层即整个系统的界面，界面层使用 Bootstrap 作为前端框架，同时用 HTML 进行前端页面的编写，CSS 进行页面排版，用 JavaScript 为 HTML 网页增加动态效果。在次基础上，使用 Ajax 技术，完成部分表单信息的提交。

**业务逻辑层：**服务器根据浏览器的响应请求，执行相应的程序，并产生对应的 HTML 代码，在经过适当的处理后返回到浏览器中。

**数据访问层：**即为对数据库的处理，系统采用 Django 框架进行开发，可通过 ORM（对象数据映射）的方式对底层的 MySQL 数据库进行相关的操作。

系统基于 Django 框架进行开发，基于 MTV 架构，这是时下流行的架构模式之一。整个架构分为三层，分别为模型层、模板层、视图层。具体结构示意图如图 4-2 所示：

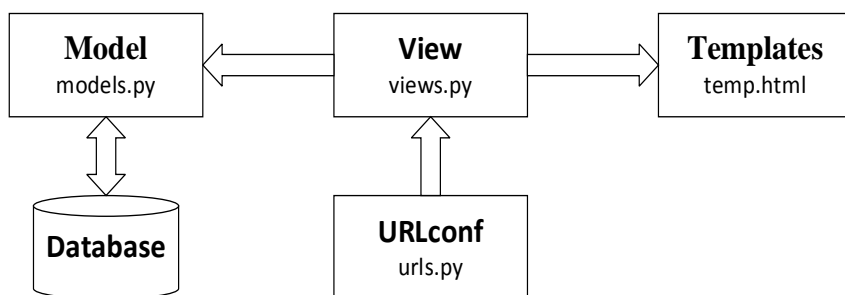


图 4-2 MTV 架构示意图

模型层（Model）：负责处理业务对象与数据库一一对应的关系，在系统中，以 model.py 文件内容来实现。

模板层（Template）：即负责前端页面的呈现，在系统中，以 template.py 文件内容来实现。

视图层（View）：负责业务逻辑部分，决定着哪些数据需要呈现在 Template（模板）中，连接着模型层与模板层，在系统中，以 view.py 文件内容来实现。

## 4.2 系统模块设计

在上一章的功能需求分析，在考虑到公众需求与系统管理的特点，使系统具有安全性、兼容性、可拓展性、性能高效性后，我们按照系统的需求，可把系统模块分为展示模块，公共服务模块与网站内容管理模块三个模块。

### 4.2.1 展示模块

在展示模块中，主要对博物馆的展览与馆藏内容进行展示，并在首页中，将博物馆的大致信息进行初步的呈现，帮助公众及时了解与获取信息。展示模块的内容主要包括博物馆展览、博物馆馆藏与博物馆首页展示这三个内容，具体模块图如图 4-3 所示。

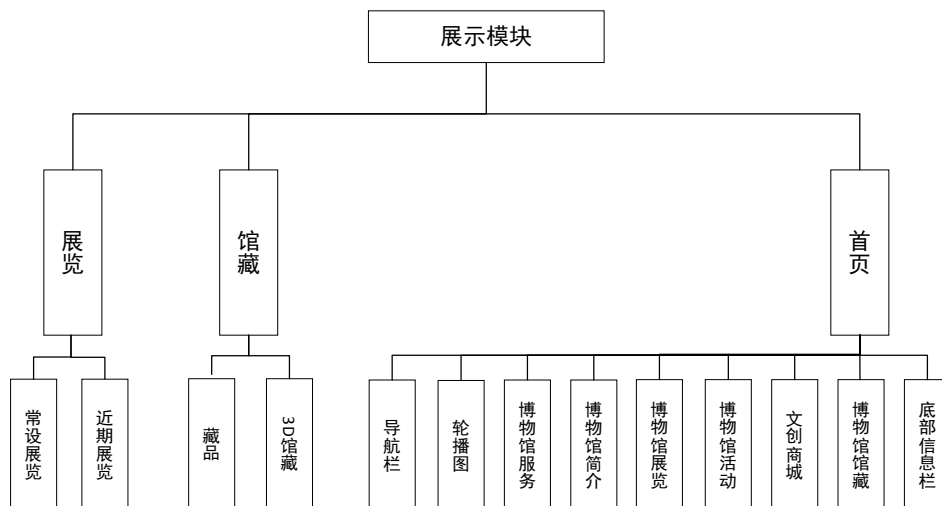


图 4-3 展示功能模块图

#### 4.2.2 公共服务模块

在公共服务模块中,围绕服务这一关键词对该模块进行构建。包括注册登录、博物馆资讯、博物馆活动、博物馆服务、文创商城、用户中心等内容。公众在此模块中,可便捷的获取服务信息,并进行相关的服务体验。具体内容如图 4-4 所示。

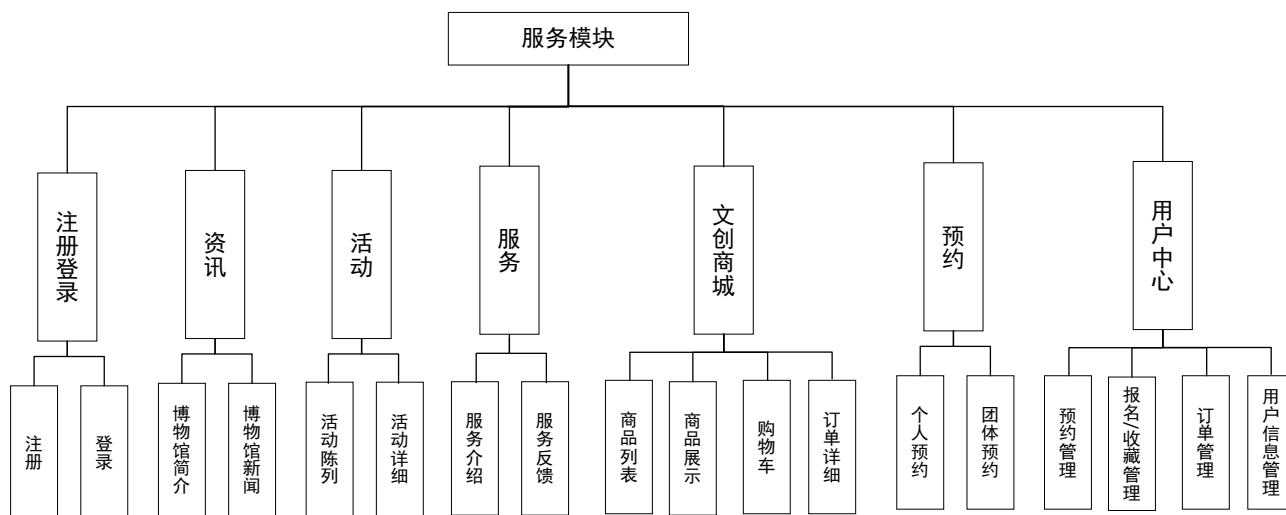


图 4-4 服务功能模块图

#### 4.2.3 网站内容管理模块

在网站内容管理模块中,管理员可对网站的所有信息内容进行管理,并完成相关增加、删除、修改、查询等操作。主要包括预约管理、展览管理、活动管理、

藏品管理、文章管理、商品管理、订单管理、用户管理、消息反馈管理等内容。具体内容如图 4-5 所示。

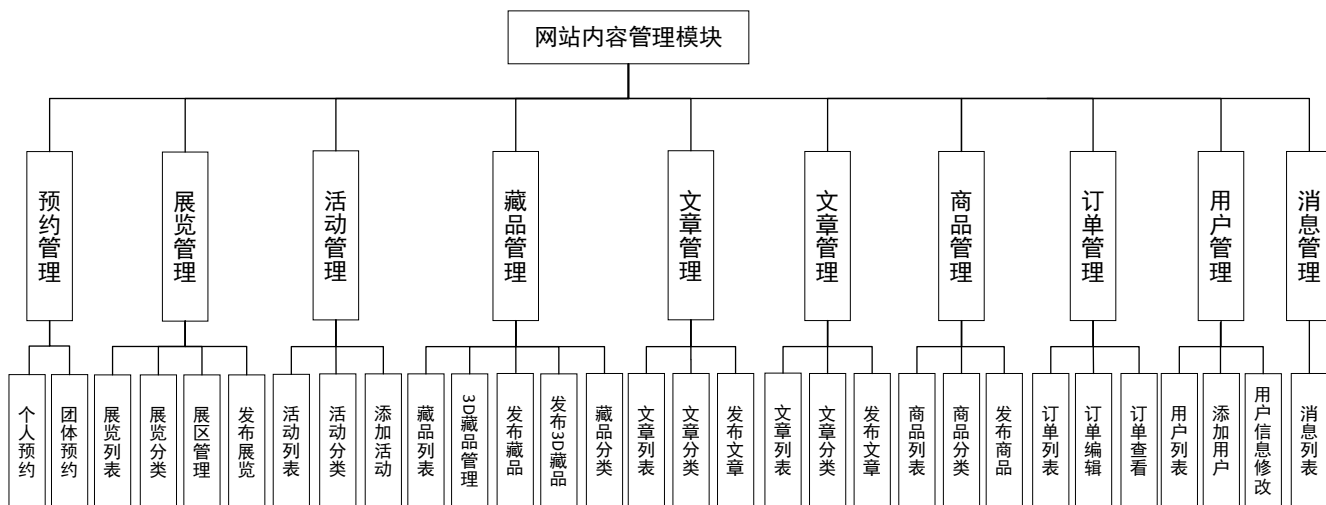


图 4-5 网站内容管理功能模块图

### 4.3 数据库设计

Django 采用 ORM（对象关系映射）的机制，在设计数据库时，只需创建数据模型，即编写对应的 python 类，再进行相关的数据迁移便可，与数据库的表一一对应。以下以创建 activity（活动）数据库对象作为创建数据模型的样例。

```

class Activity(models.Model):
    name = models.CharField(max_length=200)
    desc = models.CharField(max_length=200)
    place = models.CharField(max_length=200)
    activity_time = models.CharField(max_length=200)
    deadline_time=models.CharField(max_length=200)
    thumbnail = models.URLField()
    content = models.TextField()
    pub_time = models.DateTimeField(auto_now_add=True)
    # 引用外键
    category = models.ForeignKey('ActivityCategory', on_delete=models.
SET_NULL, null=True)
    user = models.ManyToManyField(User, related_name = 'activity',
null=True, blank=True)
  
```

根据创建数据模型，可以得到每一模型对应的数据库表。由于系统数据比较庞大，为此在此小节中，选取比较重要且使用比较频繁的若干表进行说明，各表的具体说明（按字母顺序）如下：

(1) activity 表：用于存储博物馆活动信息。包括活动名称、活动简介、活动地点等信息。活动简介（desc）为 varchar 型，最大限制为 200 字符。活动图片（thumbnail）为 varchar 型，以 url 的形式进行存储。活动内容（content）主要包含商品的文字介绍与图片信息，为 longtext 型。发布时间（pub\_time）为活动发布提交的系统时间，在发布活动时，数据库自动添加生成。具体信息如表 4-1 活动信息表所示。

表 4-1 活动信息表

字段名	类型	长度	允许空	含义
id (PK)	int	11	N	活动编号
name	varchar	200	N	活动名称
desc	varchar	200	N	活动简介
place	varchar	200	N	活动地点
activity_time	varchar	200	N	活动时间
deadline_time	varchar	200	N	报名截止时间
thumbnail	varchar	200	N	活动图片
content	longtext	/	N	活动内容
pub_time	datetime	6	N	发布时间
category_id	int	11	Y	活动类别
user_id	varchar	22	Y	报名用户 id

(2) group\_booking 表：记录团体预约的情况。包括领队姓名、身份证号、联系方式、参观人数等信息。其中团体预约编号（id）为主键，为数据库自动生成。预约日期（date）为 datetime 型，根据用户在预约界面选择的日期生成。具体信息如表 4-2 团体预约信息表所示。

表 4-2 团体预约信息表

字段名	类型	长度	允许空	含义
id (PK)	int	11	N	团体预约编号
name	varchar	50	N	领队姓名



identity	varchar	18	N	领队身份证号
telephone	varchar	11	N	领队联系方式
number	varchar	10	N	团体参观人数
date	datetime	6	N	预约日期
created	datetime	200	N	预约创建日期

(3) three\_collection 表：用于存储 3D 藏品信息。包括藏品名称、图片、Obj 文件、Mtl 文件、详细内容等信息。Obj (objproject) 文件为 3D 模型文件格式，Mtl 文件 (mtlproject) 是材质库文件，表述 3D 模型的材质信息。两者均为 varchar 型，以 url 的形式进行存储。具体信息如表 4-3 3D 藏品信息表所示。

表 4-3 3D 藏品信息表

字段名	类型	长度	允许空	含义
id (PK)	int	11	N	3D 藏品编号
name	varchar	200	N	3D 藏品名称
thumbnail	varchar	200	N	3D 藏品图片
objproject	varchar	200	N	Obj 文件
mtlproject	varchar	200	N	Mtl 文件
content	longtext	/	N	3D 藏品详细
pub_time	datetime	6	N	发布时间
user_id	varchar	22	Y	收藏用户 id

(4) exhibition 表：用于存储博物馆展览信息。包括展览名称、简介、图片、展览类别等信息。其中展览简介 (desc) 为 varchar 型，最大长度为 200 字符。展览类别 (category\_id) 为外键，依附于展览分类表。具体信息如表 4-4 展览信息表所示。

表 4-4 展览信息表

字段名	类型	长度	允许空	含义
id (PK)	int	11	N	展览编号
name	varchar	200	N	展览名称

desc	varchar	200	N	展览简介
thumbnail	varchar	200	N	展览图片
pub_time	datetime	6	N	发布时间
category_id	int	11	Y	展览类别
exhibition_time	varchar	200	Y	展览时间

(5) exhibition\_area 表：用于存储展区活动信息。展区构成展览，所以所属展览（exhibition\_id）作为外键。在添加展区时，系统会自动获取当前展览名称与id，无须填写。展区信息除了包括所属展览，还包括展区名称、简介、图片、详细等内容。其中，展区详细（content）包含展区的文字与图片介绍，为 longtext 型。具体信息如表 4-5 展区信息表所示。

表 4-5 展区信息表

字段名	类型	长度	允许空	含义
id (PK)	int	11	N	展区编号
name	varchar	200	N	展区名称
desc	varchar	200	N	展区简介
thumbnail	varchar	200	N	展区图片
content	longtext	/	N	展区详细
pub_time	datetime	6	N	发布时间
exhibition_id	int	11	N	所属展览

(6) message\_advice 表：用于存储意见反馈信息。主要包括姓名、电子邮件、意见内容等信息，且均不能为空。意见内容（content）为 varchar 型，最大不能超过 500 字符。具体信息如表 4-6 意见反馈信息表所示。

表 4-6 意见反馈信息表

字段名	类型	长度	允许空	含义
id (PK)	int	11	N	意见编号
name	varchar	50	N	姓名
email	varchar	254	N	电子邮件

content	varchar	500	N	意见内容
---------	---------	-----	---	------

(7) usermessageread 表：用于存储用户订单的已读信息。由信息内容、信息状态、是否已读等字段构成。其中信息状态（status）为 int 型，不同数字表示不同的订单状态，是否已读（is\_read）为 tinyint 型，默认为 False（未读），订单 id（order\_id）与下单用户 id（user\_id）为外键。具体信息如表 4-7 已读信息表所示。

表 4-7 已读信息表

字段名	类型	长度	允许空	含义
id (PK)	int	11	N	信息编号
content	varchar	500	N	信息内容
status	int	11	N	信息状态
created	datetime	6	N	发布时间
is_read	tinyint	1	N	是否已读
order_id	int	11	N	订单 id
user_id	varchar	22	N	下单用户 id

(8) user 表：用于存储用户信息。用户的信息由用户 id、手机号、姓名、身份证号、密码等构成。用户 id（uid）为主键，采用 ShortUUIDField 保证主键的唯一性。手机号（telephone）要求不能大于 11 位，身份证号（identity）要求必须是 18 位。是否活跃（is\_active）默认 True（活跃），是否工作人员（is\_staff）与是否超级管理员（is\_superuser）默认为 False（否）。注册时间（data\_joined）与上次登录时间（last\_login）均为 datetime 型，为数据库自动添加记录。具体信息如表 4-8 用户信息表所示。

表 4-8 用户信息表

字段名	类型	长度	允许空	含义
uid (PK)	varchar	22	N	用户 id
telephone	varchar	11	N	手机号
username	varchar	100	N	姓名

identity	varchar	18	N	身份证号
password	varchar	128	N	密码
is_active	tinyint	1	N	是否活跃
is_staff	tinyint	1	N	是否工作人员
is_superuser	tinyint	1	N	是否管理员
data_joined	datetime	6	N	注册时间
last_login	datetime	6	Y	上次登录时间

(9) order 表：用于存储订单信息。包括下单姓名、电子邮件、联系方式、联系地址、订单总价等信息。创建时间（created）为订单生成时，数据库自动创建。是否发货（delivery）默认为 False（未发货），订单总价（allprice）为 decimal 型，精确到小数点后两位，最大长度为 10。下单用户 id（user\_id）为外键，与用户表对接。具体信息如表 4-9 订单信息表所示。

表 4-9 订单信息表

字段名	类型	长度	允许空	含义
id (PK)	int	11	N	订单编号
name	varchar	50	N	下单姓名
email	varchar	254	N	电子邮件
telephone	varchar	11	N	联系方式
address	varchar	250	N	联系地址
created	datetime	6	N	创建时间
updated	datetime	6	N	更新时间
paid	tinyint	1	N	是否支付
delivery	tinyint	1	N	是否发货
allprice	decimal	10	N	订单总价
user_id	varchar	22	N	下单用户 id

(10) orderitem 表：用于存储订单明细信息，即订单包含每一具体商品的信息。主要包括商品单价、数量、总价等信息。商品单价（price）与商品总价（totalprice）为 decimal 型，精确到小数点后两位，商品数量（quantity）为整型。订单 id（order\_id）

与商品 id (product\_id) 为表的外键，记录相关联的订单与商品。具体信息如表 4-10 订单明细信息表所示。

表 4-10 订单明细信息表

字段名	类型	长度	允许空	含义
id (PK)	int	11	N	编号
price	decimal	10	N	商品单价
quantity	int	10	N	商品数量
totalprice	decimal	10	N	商品总价
order_id	int	11	N	订单 id
product_id	int	11	N	商品 id

## 5 系统实现

### 5.1 平台架构

#### 5.1.1 开发环境

系统开发使用 Window7 操作系统，以 Python、Django2.1、MySQL 8.0.14、Bootstrap3.3.7 等作为开发环境，用 JetBrains PyCharm 2018.3.3 x64 作为开发工具进行开发。

#### 5.1.2 创建流程

基于以上的开发环境，本系统开发的创建流程如下：

##### （1）新建 project 工程项目

打开终端，使用命令：`django-admin startproject museum` 创建，创建完成后，项目工程下会生成 `museum_website` 文件夹。项目结构如图 5-1 所示：

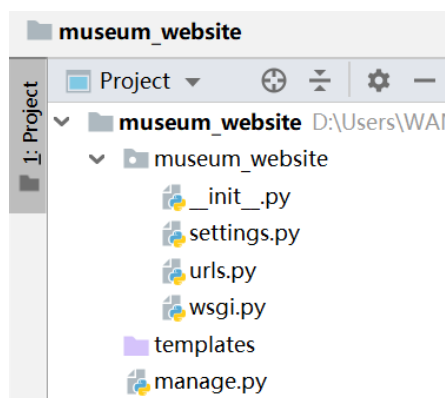


图 5.1 项目结构

其中：

`__init__.py`：初始化文件，文件内容为空，表示当前目录是一个 Python 的模块。

`settings.py`：负责项目的配置，与项目相关的配置信息均在此文件中。

`urls.py`：负责配置 URL 路由，网站网址的映射文件。

`wsgi.py`：网关接口配置文件，部署的时需要用到，一般情况下无需修改。

`manage.py`：负责与项目交互的命令行工具，如创建应用，迁移数据库等。

##### （2）配置数据库

在 `setting.py` 中进行数据库的配置，使 `django` 项目能与数据库进行连接，具体设置如下：

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': "museum",  
        'HOST': '127.0.0.1',  
        'PORT': '3306',  
        'USER': 'root',  
        'PASSWORD': 'root'  
    }  
}
```

### （3） 创建应用

在终端进入到项目所在的路径，通过 `python manage.py startapp [app 名称]` 的命令，创建 APP（独立的小型项目）。并将其添加到 `setting.py` 的 `INSTALLED_APPS` 中。

### （4） 创建模型，同步模型到数据库

在 `app` 中的 `model.py` 文件中，通过类的形式创建 ORM 模型，可对数据库所在的路径进行操作，执行命令 `python manage.py makemigrations` 与 `python manage.py migrate` 两个命令分别生成迁移脚本文件与映射到数据库中。

### （5） 创建 view.py 函数

在 `view.py` 中编写函数，对来自前端页面的请求，进行逻辑相关操作，并把执行的结果返回给浏览器。

### （6） 添加 URL 映射

在主项目与 APP 文件中添加 URL 映射，定义不同 URL 地址请求不同的视图函数。

### （7） 创建模板

将前端页面的 HTML 文件放到 `template.py` 中，并根据 DTL 模板的语法特点，将原 HTML 文件进行修改，实现数据动态化以适应模板的语法规则，可被 Django 编译。

### （8） 项目启动

在项目所在的路径，通过执行命令 `python manage.py runserver` 进行项目的启动，在网址输入 `127.0.0.1:8000`（Django 默认端口）即可对项目进行访问。

### 5.1.3 项目结构

为了使项目结构清晰明了，将项目划分为四个模块，分别为 `apps`、`front`、`museum_website`、`utils`。

`apps` 模块：用来存放每一个 `app` 项目。将每一个独立的小型项目放到每一个 `app` 里，共计 12 个 `app`。具体项目模块图如图 5-2 所示：

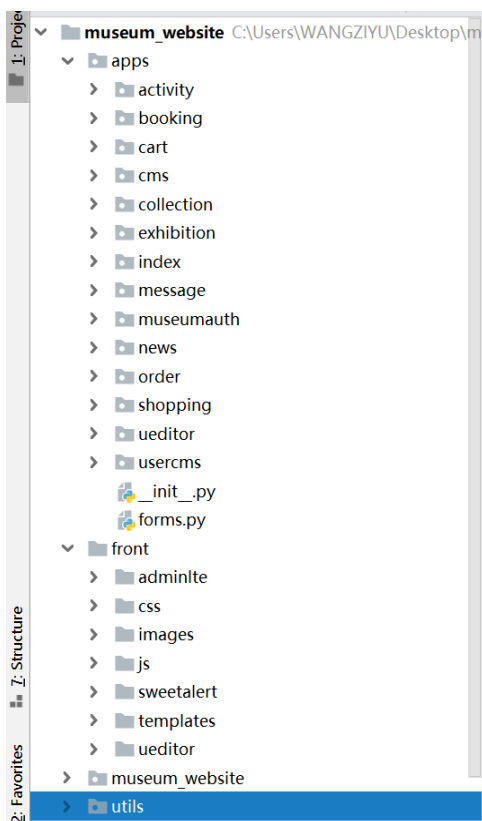


图 5-2 项目模块图

`front` 模块：用来存放 `html`、`css`、`js`、模板、图片等与前端相关的所有信息。

`museum_website` 模块：为项目的主模块，主要包括 `_init_.py`、`settings.py`、`urls.py`、`wsgi.py` 等信息。

`utils` 模块：工具模块，存放相关工具文件。

## 5.2 展示模块

### 5.2.1 博物馆展览模块

#### （1）常设展览



对博物馆常设展览进行展示，用户可在此页面看到常设展览的名称与大致信息，如图 5-3 所示。



图 5-3 常设展览界面

展览详细部分对展览的概括与展厅进行了介绍，如图 5-4 所示。



图 5-4 展览详细界面

常设展览同时分为不同展区，除了了解该展览的详细信息以外，还可对每一展区进行详细的了解。如图 5-5 所示。



图 5-5 展区详细介绍界面

### (2) 临时展览

对近期博物馆临时展出的展览进行展示，用户可看到展览的名称与大致信息，点击后可对其进行详细的了解。临时展览陈列界面如图 5-6 所示。

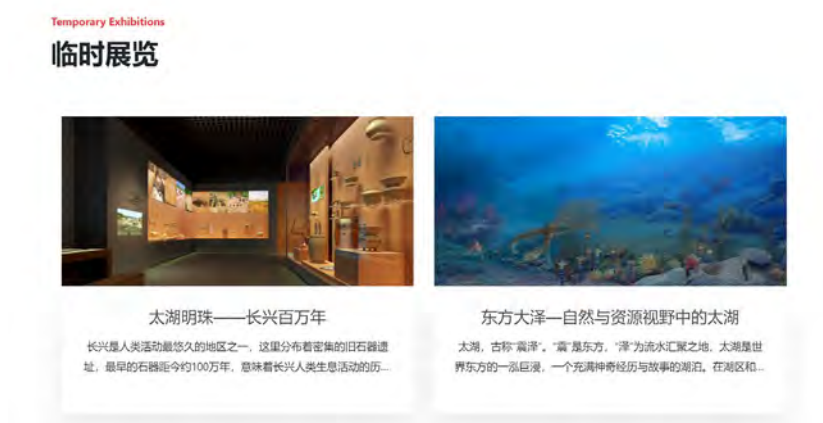


图 5-6 临时展览陈列界面

### (3) 虚拟展览

即虚拟展览，利用网页构造三维的虚拟展厅，用户可通过点击、转动等操作对虚拟展览进行体验。进入虚拟展厅后，系统会对展厅进行初步的文字介绍与语音介绍。用户点击展览的文物，页面视角会对文物进行呈现。点击页面右上方声音按钮，可对声音的音量进行相关调节。具体设计可详见第 6 章。

## 5.2.2 博物馆馆藏模块

### (1) 藏品

藏品陈列页：对馆内藏品进行陈列展示，点击藏品可了解藏品详细。鼠标悬停藏品时，藏品出现放大动效，以此增强用户的互动体验。如图 5-7 所示。

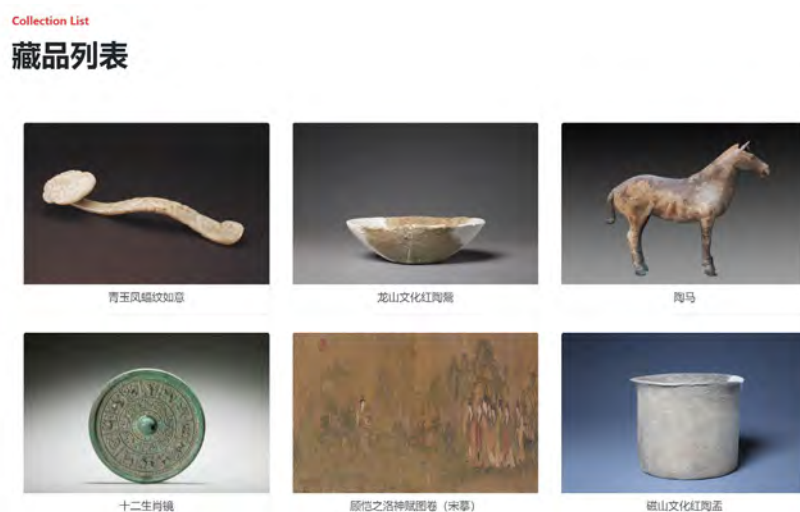


图 5-7 藏品陈列界面

藏品详细页：对藏品进行详细的展示，如下图 5-8 所示。



图 5-8 藏品详细界面

点击藏品图片可放大进行查看；如下图 5-9 所示。



图 5-9 藏品放大界面

用户对藏品感兴趣可以点击收藏按钮，可进行收藏。收藏数据可在用户中心查看。未登录用户不能使用收藏功能，需要登录后才能进行操作。如收藏成功将跳转至收藏成功界面。收藏功能的业务逻辑流程图如下图 5-10 所示。

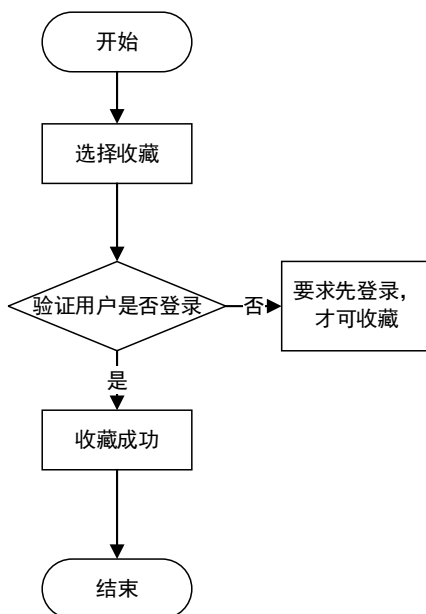


图 5-10 收藏功能的业务逻辑流程图

收藏成功模块界面如图 5-11 所示。



图 5-11 藏品收藏成功界面

## (2) 3D 藏品

3D 藏品的陈列页与详细页在设计上实现与藏品大致相同, 区别在于 3D 藏品多了 3D 展示的部分。该部分全方位多维度的展示了文物各部分特征, 有效还原了藏品的真实效果, 进一步丰富用户的感官体验, 强化对文物的感知与理解。用户可通过点击、拖动等方式, 对 3D 藏品进行旋转, 放大等体验。具体设计可详见第 6 章。

### 5.2.3 博物馆首页模块

首页的设计是系统设计最为关键的模块, 所呈现的效果直接影响到用户对系统的第一印象, 作为博物馆门户网站的首页, 在设计时应考虑架构合理, 页面美

观，内容丰富等因素。具体设计如下：

结构设计：首页结构由导航栏、轮播图、底部信息栏以及博物馆相关信息组成，其中相关信息是对博物馆服务、简介、展览、活动、文创商城、馆藏的大致呈现用户可点击内容获取相关信息。

（1）导航栏：导航栏对系统整个模块清晰的呈现，如未登录会在右侧显示登录/注册，登录系统后，显示用户姓名。首页的导航栏背景设为透明，与轮播图合为一体，凸显简约大气。



图 5-12 首页导航栏、轮播图实现效果

（2）轮播图：轮播图显示博物馆最新的资讯、展览、活动等信息。鼠标如不放在图片上方，轮播图会自动切换播放。轮播图片采用了缓慢放大的动态效果，以增加用户的视觉体验。实现效果见上图 5-12。

（3）博物馆服务：公众访问博物馆网站最想了解的一定是预约订票、开放时间、交通信息等相关服务资讯，为此将服务板块放在博物馆相关信息的首位。对服务时间、交通信息、票务信息进行呈现，并可通过下方“立即预约”按钮，跳转至预约界面。





图 5-13 首页服务、简介模块实现效果

(4) 博物馆简介：对博物馆基本信息进行一个初步介绍，如有想更加了解，点击“查看更多”跳转博物馆简介页面。实现效果见上图 5-13。

(5) 博物馆展览：对博物馆近期展览进行呈现，帮助用户快速获取近期展览信息。如需了解详细，可点击展览进行详细了解。实现效果见图 5-14。



图 5-14 首页最新展览模块实现效果

(6) 博物馆活动：对博物馆最新组织的活动进行呈现，包括活动名称、地点、时间等信息，用户可直接点击进行了解。实现效果见图 5-15。



图 5-15 首页博物馆活动模块实现效果

(7) 博物馆文创商城：文创商城作为近段时间博物馆行业备受关注的领域，这里对文创商品进行简要的介绍与展示，并提供按钮跳转到文创商城。实现效果见图 5-16。



图 5-16 首页博物馆文创商城模块实现效果

(8) 博物馆馆藏：以相册的形式对馆藏进行了展示，点击图片跳转馆藏页面。鼠标悬停图片，图片变红，以增强与用户的互动。实现效果见图 5-17。



图 5-17 首页博物馆馆藏鉴赏模块实现效果

(9) 底部信息栏：提供相关的服务信息、联系方式。实现效果见图 5-18。



图 5-18 首页博物馆底部信息栏模块实现效果

## 5.3 公共服务模块

### 5.3.1 注册、登录模块

#### (1) 注册模块

未注册用户可注册会员，注册信息应包括姓名、手机、身份证号、密码、再次确认密码等信息。如注册填写的手机号已被注册，身份证格式输入错误，两次前后密码输入不一致，或者信息填写不全等情况时，页面会有相应提示并要求重新输入。如注册成功则跳转登录界面。

注册信息：注册信息如表 5-1 所示。

表 5.1 注册信息

填写内容	要求	备注
姓名	需为真实姓名	必填项，允许同名
联系方式	真实联系方式，不超过 11 字符。	必填项，同一手机号不能重复注册。
身份证号	要求 18 字符	必填项
密码	6-20 字符，可是数字、字母、其他符号等	必填项
确认密码	与密码输入一致	必填项

信息校验：通过 from 表单对注册信息进行校验，查看是否符合要求。

信息提示：对不符合规范的填写内容，在页面的上方进行信息提示。

实现效果：如图 5-19 所示：



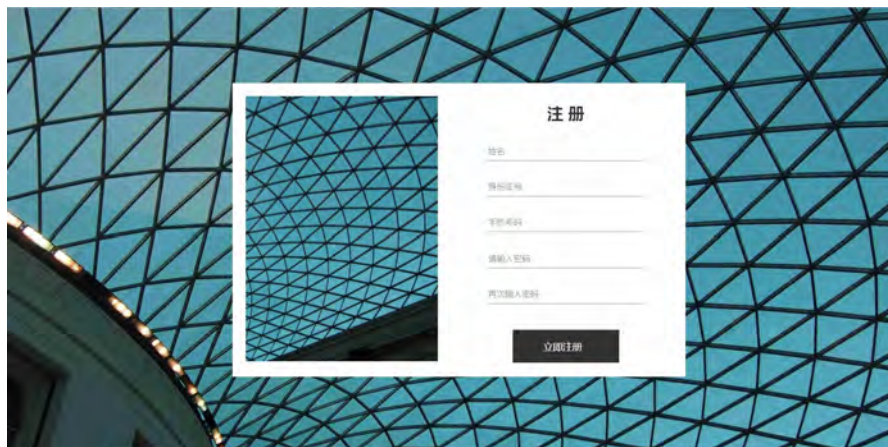


图 5-19 注册页面实现效果

## (2) 登录模块

用户进行登录只需填写联系方式、密码两项信息，系统会将用户填写的信息与数据库信息进行匹配，如信息输入错误，或出现联系方式未注册、密码输入错误等问题时，页面会有相应提示。登录成功后，跳转博物馆首页，导航栏上方显示用户姓名。

信息提示：对不符合规范的填写内容，在页面的上方进行信息提示。

状态留存：用户在登录后，登录信息应一直保留在浏览器中，直到用户关闭浏览器。

权限限制：登录用户可点击用户姓名进入用户中心，查看相关信息；未登录用户不可进入用户中心。

实现效果：如图 5-20 所示：

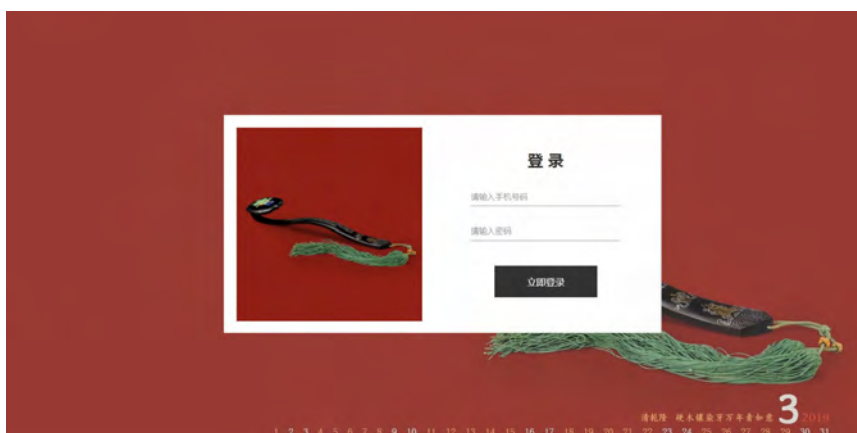


图 5-20 登录页面实现效果

### 5.3.2 博物馆资讯模块

#### (1) 博物馆简介

此页面对博物馆进行详细的介绍,包括简介、历史沿革、博物馆图片等部分。其中点击博物馆图片,可对图片进行放大查看,并可以上下页进行切换。具体页面如图 5.21 所示:



图 5-21 博物馆介绍页面实现效果

#### (2) 博物馆新闻

对博物馆的媒体报道、国内外行内资讯等内容进行陈列展示,显示新闻图片、标题、时间与简介等信息。如图 5-22 所示。



图 5-22 新闻列表页面实现效果

新闻详细页则显示新闻的具体内容。如图 5-23 所示。



图 5-23 新闻详细页面实现效果

### 5.3.3 博物馆活动模块

#### (1) 活动陈列页

对博物馆近期的活动进行陈列展示，显示活动名称、时间、地点等信息。如图 5-24 所示。



图 5-24 活动列表页面实现效果

#### (2) 活动详细页

对活动的详细内容进行展示，用户可进行活动报名操作，报名历史可在用户中心查看。未登录用户不能报名活动，如报名成功将跳转至报名成功界面。活动报名的业务逻辑流程图如下图 5-25 所示。

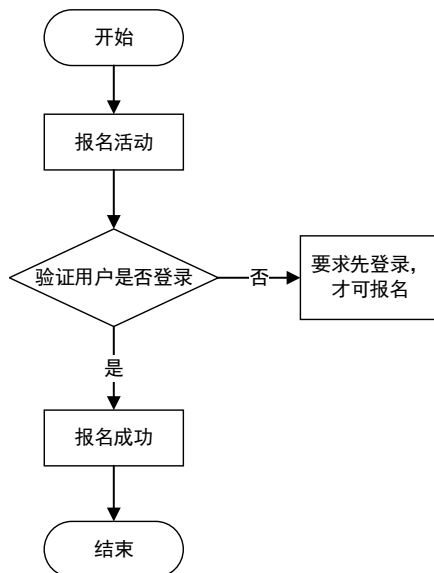


图 5-25 活动报名的业务逻辑流程图

页面实现如图 5-26 所示。



图 5-26 活动详细页面实现效果

5.3.4 博物馆服务模块

(1) 服务介绍

服务的详细介绍页，对交通指南、服务时间、票务信息、租赁轮椅、导览服务、其他服务等博物馆的服务信息进行了全面的介绍。如图 5-27 所示。



图 5-27 服务介绍页面实现效果

(2) 服务反馈

通过填写表单的形式，征求用户对于博物馆的意见与建议。如图 5-28 所示。



图 5-28 服务反馈实现效果

5.3.5 文创商城模块

文创商城板块主要用于博物馆文创商品的线上出售，用户访问网页可对文创商品进行浏览与查看，注册用户可以加入购物车形成订单完成商品的购买服务。

(1) 商品列表



展示博物馆内在售的商品信息，实现图片见图 5-29：

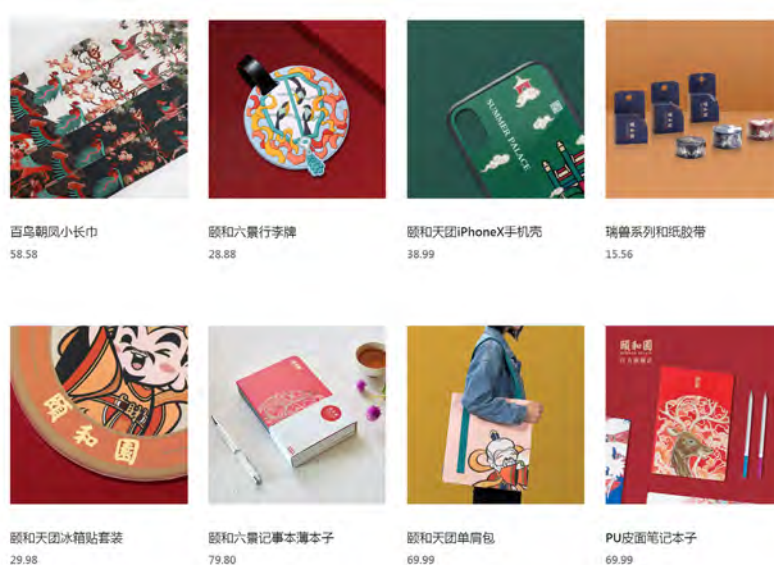


图 5-29 商品列表页面实现效果

## (2) 商品展示

对商品的具体信息进行详细的展示，包括图片、名称、简介、价格、是否有库存、状态、上架日期与具体介绍等信息，向用户全面展示商品。

页面实现如图 5-30 所示：



图 5-30 商品详情页面实现效果

同时，用户可以将心仪商品及数量加入购物车。添加购物车功能的业务逻辑流程图如 5-31 所示：

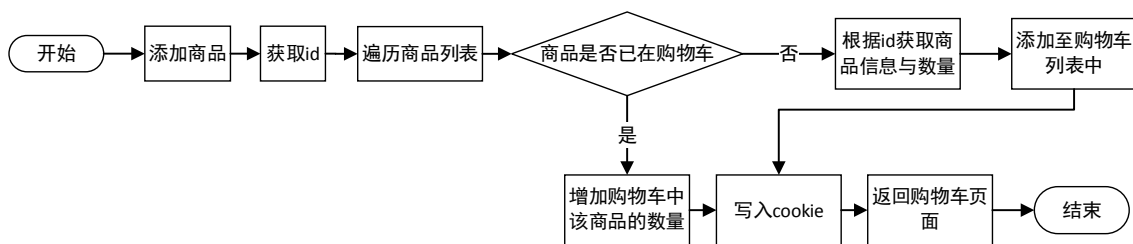


图 5-31 添加购物车业务逻辑流程图

### (3) 购物车

购物车是整个商城模块最为复杂的部分。主要实现结算、清空购物车、继续购物等功能。且在浏览网站与未下订单之前，购物车都会维持其中的信息。购物车栏内，显示每一商品主要信息，包括数量与总价，可对商品进行移除与数量的更新。最下端显示购物车总价格。

根据第 3 章的需求分析可知，购物车功能需要在登录系统后才能进行使用。

实现效果如图 5-32 所示：

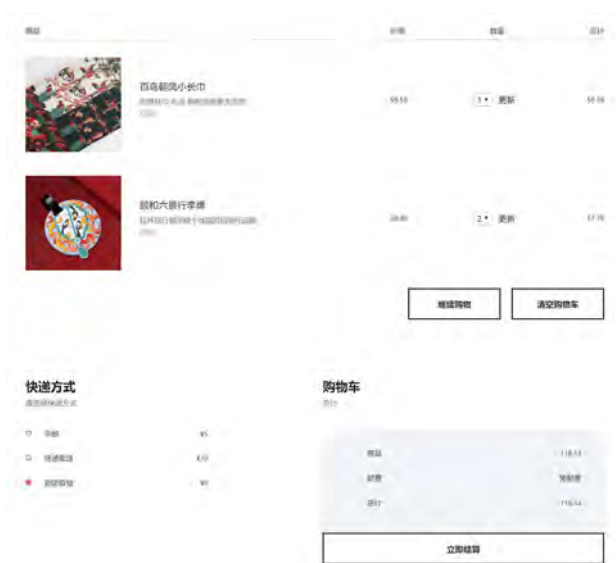


图 5-32 购物车模块实现效果图

购物车模块流程图如图 5-33 所示：

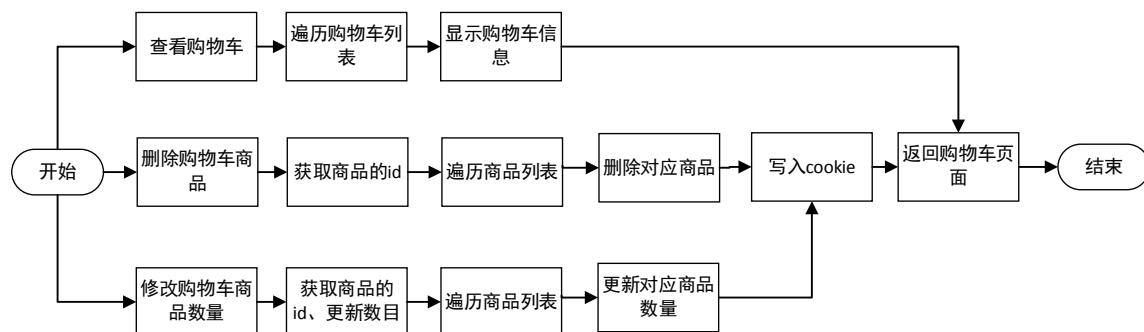


图 5-33 购物车模块业务逻辑流程图

#### (4) 订单详细

用户结算购物车时，生成订单；订单显示订单商品名称、数量、总价等信息。用户需对收货地址（姓名、电子邮件、联系方式、详细地址）进行填写，从而提交订单。实现效果如图 5-34 所示。

图 5-34 订单模块实现效果图

订单提交成功页面如图 5-35 所示。

图 5-35 订单支付成功实现效果图

#### 5.3.6 预约模块

预约模块分为个人预约与团体预约两种类型，用户在未注册的情况下便可进行预约。在预约界面填写相关的信息，并选择入馆日期，便可完成操作。如图 5-36 所示。



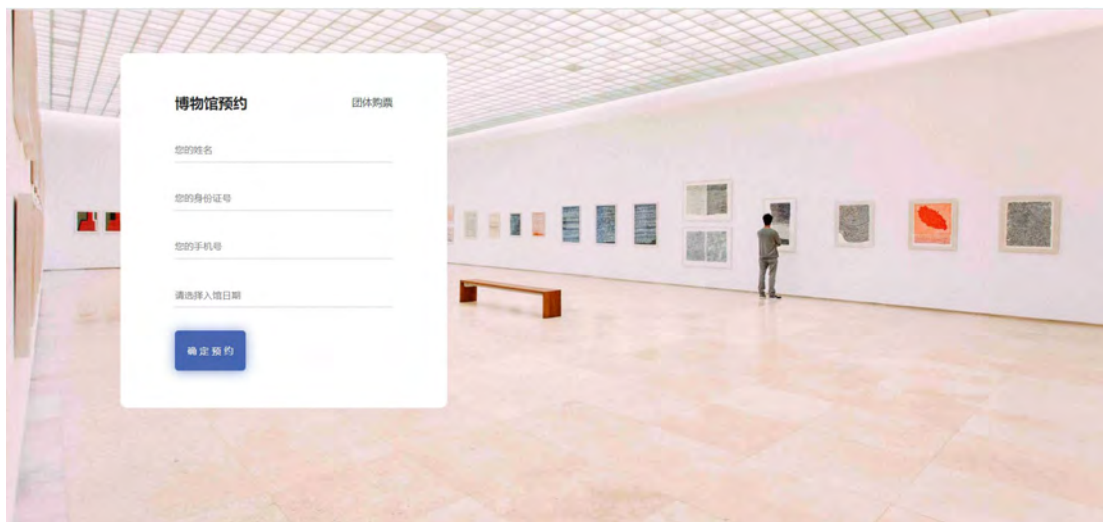


图 5-36 个人预约实现效果图

### 5.3.7 用户中心模块

用户中心应包括预约管理、报名/收藏管理、订单管理、用户信息管理这几个模块。

(1) 预约管理：用户可以查看自己的预约情况，包括个人预约与团体预约。预约页面显示预约姓名、身份证号、联系方式、参观日期等信息。团体预约如图 5-37 所示。

团体预约列表					
时间： <input type="text" value="起始时间"/> — <input type="text" value="结束时间"/>		<input type="button" value="查询"/>	<input type="button" value="清除查询"/>		
领队姓名	领队身份证号	参观人数	参观时间	创建时间	联系方式
w	123456789012345678	12	2019/04/12	2019/04/04 23:05:37	2131
测试	123456789012345678	4	2019/04/18	2019/04/04 23:06:47	1000001006
第1/总共1页					
		<input type="button" value="上一页"/>		<input type="button" value="1"/>	<input type="button" value="下一页"/>

图 5-37 团体预约模块实现效果图

(2) 报名/收藏管理：包括我的活动与我的收藏两个模块，用户可以查看自己已报名活动与已收藏的文物。

我的活动：显示用户已报名的活动信息，包括活动名称、分类、地点、时间等，用户可以就活动名称与分类进行相关搜索操作，亦可取消报名。点击活动名称可查看活动具体内容。

实现效果如图 5-38 所示：



图 5-38 我的活动模块实现效果图

我的收藏：显示用户收藏的藏品与 3D 藏品的信息，包括藏品名称、类别等，用户可就藏品名称与分类进行相关搜索操作，亦可取消收藏。点击藏品名称可查看藏品的具体内容。

实现效果如图 5-39 所示：



图 5-39 我的收藏模块实现效果图

（3）订单管理：包括订单查看与消息中心两个板块。

我的订单：可查看用户订单记录，包括订单编号、金额、创建时间、订单状态、发货状态等信息。可根据时间、订单号进行订单的搜索。实现效果如图 5-40 所示：



图 5-40 订单列表模块实现效果图

点击查看，可了解每一订单的具体内容，如图 5-41 所示。

订单详细 (订单编号: 6)				
基本信息				
下单用户	下单时间	订单状态	发货信息	订单总金额
SldEgo45UZ7oM23LrWuJaS	2019/05/09 08:28:49	已支付	已发货	118.54
收件信息				
收件人姓名	收件人邮箱	收件人联系方式	收件地址	
test2	wzy.22@qq.com	57427830820	江南路1689号	
购买信息				
序号	商品名称	商品数量	商品价格	商品总价
1	朕和天团冰霜贴套装	2	29.98	59.96
2	百鸟朝凤小长巾	1	58.58	58.58

图 5-41 订单详细实现效果图

消息中心：对订单状态与发货状态的消息变更进行提示，左侧未读信息栏显示未读消息的数目，用户可在未读信息中进行查看。如用户已对消息熟悉，可标记已读，未读消息会自动转换成已读消息。

实现效果如图 5-42 所示：



图 5-42 消息中心模块实现效果图

(4) 个人信息管理：包括信心查看与修改信息两个板块，

信心查看：用户可查看自己的个人信息，包括姓名、身份证、联系方式、密码等，其中密码以加密形式展示

实现效果如图 5-43 所示：

个人信息

姓名

test2

手机

100000

身份证号

123456789012345678

密码

.....

图 5-43 个人信息模块实现效果图

修改信息：用户可对姓名、手机、身份证号信息进行修改，修改原理同用户注册。

实现效果如图 5-44 所示：

修改信息

姓名

test2

手机

100000

身份证号

123456789012345678

密码

.....

确认密码

.....

确认修改

图 5-44 修改信息模块实现效果图

## 5.4 网站内容管理模块

管理员通过登录账户可来到后台管理界面。在后台，管理员可对系统信息进行管理，包括预约管理、展览管理、藏品管理、活动管理、新闻管理、商品管理、订单管理、用户管理、消息反馈管理等操作。

后台框架由左部菜单栏与右部的管理内容组成，实现效果如图 5-45 所示。

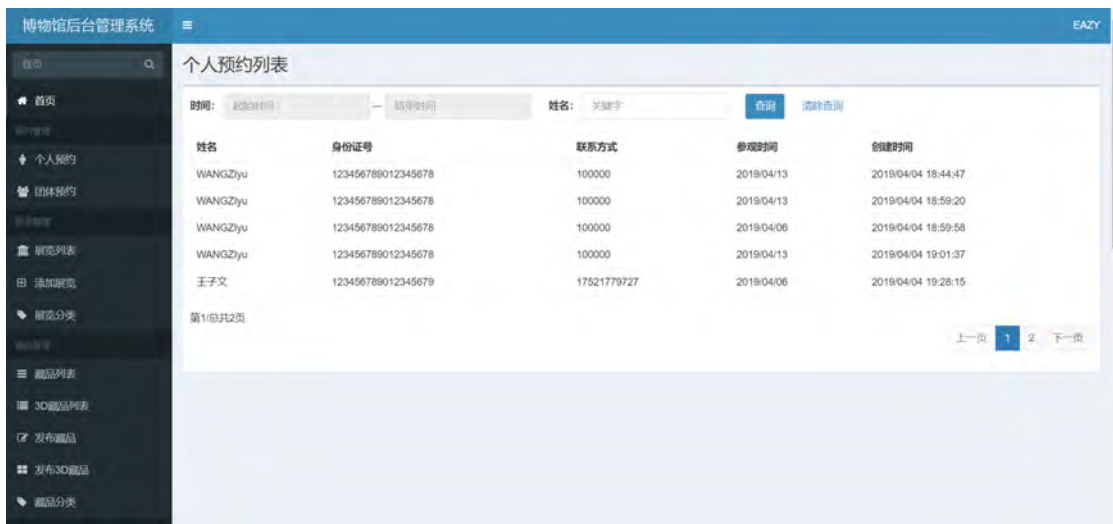


图 5-45 后台模块实现效果图

### 5.4.1 预约管理

预约管理分为个人与团体预约两个模块，可分别查看博物馆的预约情况，并可以根据日期、姓名对预约情况进行查询，可有效了解特定日期与特定人物的预约情况。预约的业务逻辑流程图如图 5-46 所示。

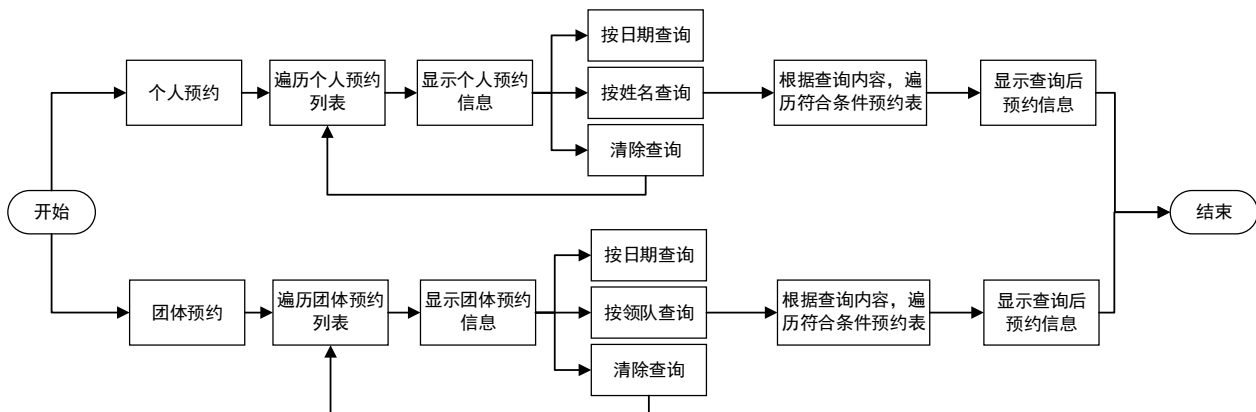


图 5-46 预约业务逻辑流程图

团体预约列表如图 5-47 所示。



图 5-47 团体预约列表实现效果图

### 5.4.2 展览管理

展览管理由展览列表、添加展览、展览分类这三部分构成，对博物馆展出的展览、展区进行管理操作，具体设计如下：

（1）展览列表：展览列表的内容由展览名称、展览分类、展区与操作构成，可就展览名称、分类进行特定查找。并可对展览进行编辑与删除，对展区进行管理等操作。如图 5-48 所示：



图 5-48 展览列表实现效果图

其中展区管理部分可对展区进行编辑、添加、修改、删除等操作。如图 5-49 所示。

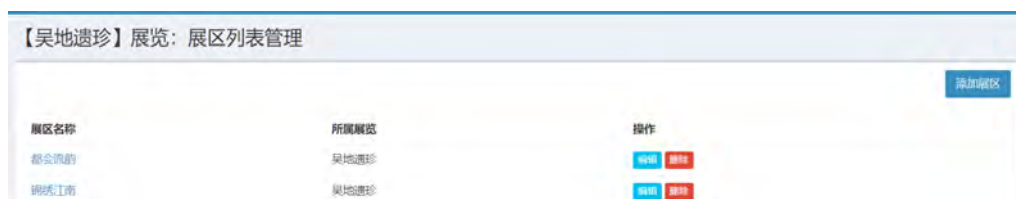


图 5-49 展区列表实现效果图

添加展区时，需填写展区名称、所属展览、展区介绍、展区图片与主要内容，所属展览为页面自动生成，不能修改。如图 5-50 所示：





图 5-50 编辑展区实现效果图

(2) 添加展览：对展览进行发布，需填写展览名称、展览分类、展览时间、展览简介、展览图片等信息，其中展览分类由后端从添加的分类中获取，选择某一分类即可。如图 5-51 所示：

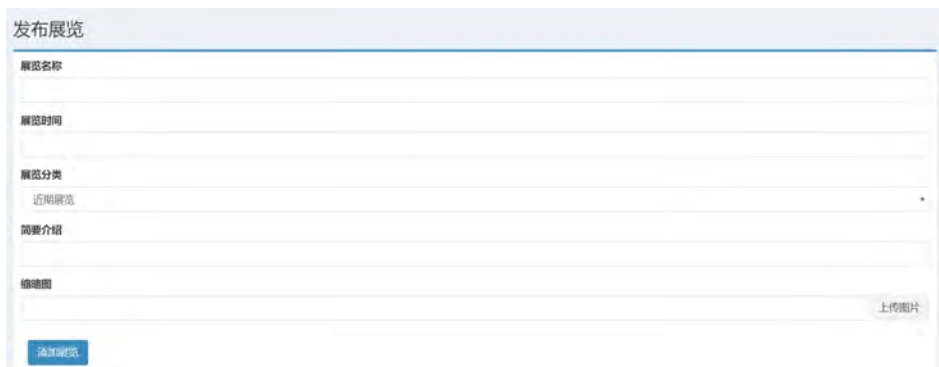


图 5-51 发布展览实现效果图

(3) 展览分类：可对展览分类进行添加、删除、修改等操作。如图 5-52 所示：

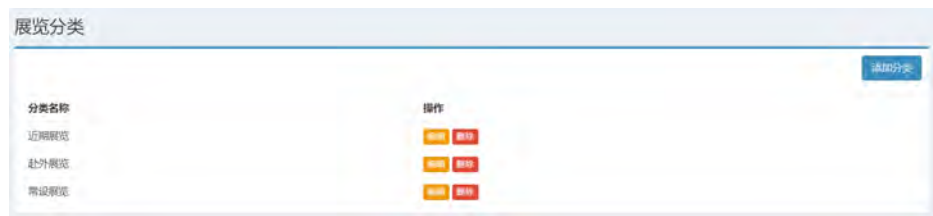


图 5-52 展览分类实现效果图

### 5.4.3 藏品管理

由于藏品与 3D 藏品上传的材料、文件格式并不相同，为了方便管理与划分将藏品与 3D 藏品划分为两个不同模块。藏品模块主要包括藏品列表、发布藏品、

藏品分类，3D 藏品模块包括 3D 藏品列表、发布 3D 藏品。其中 3D 藏品部分放在第六章进行详细说明。

(1) 藏品列表：显示藏品名称、分类、发布时间等信息，可就藏品名称、分类进行特定查找。如图 5-53 所示：



图 5-53 藏品特定查找实现效果图

并可对藏品进行编辑与删除。如图 5-54 所示：



图 5-54 藏品删除实现效果图

(2) 发布藏品：管理员可填写藏品名称、选择藏品分类、上传藏品图片、编辑藏品内容，完成对藏品的发布。

(3) 藏品分类：对藏品分类进行添加、删除、修改等操作。

#### 5.4.4 活动管理

活动管理由活动列表、添加活动、活动分类这三部分构成，对博物馆举办的活动进行管理操作，活动流程图如图 5-55 所示：



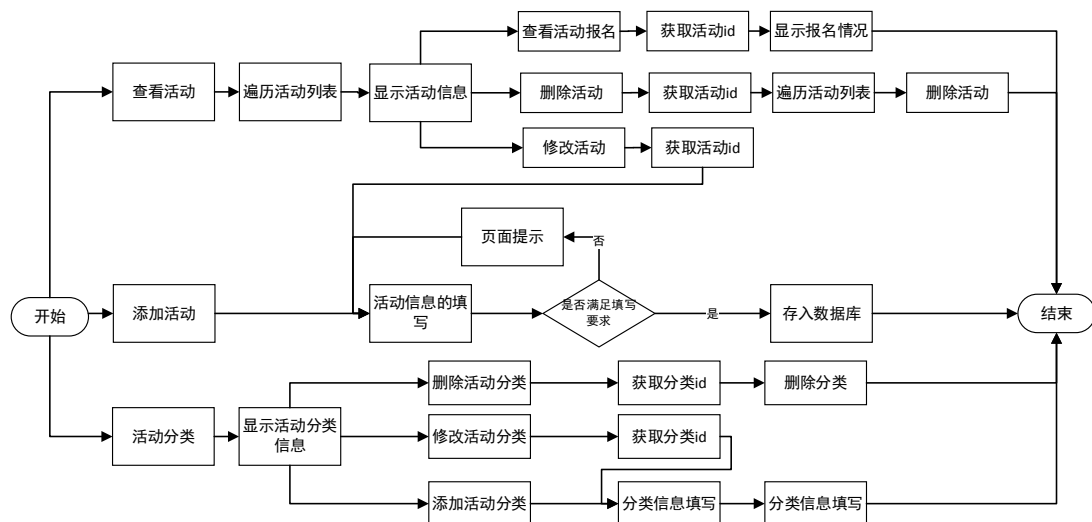


图 5-55 活动业务逻辑流程图

具体设计如下：

(1) 活动列表：活动列表对活动名称、活动分类、活动时间、活动地点等信息进行了显示，并可对活动进行编辑、删除、查看活动报名情况。就活动名称与分类进行特定查找。

(2) 发布活动：对活动进行发布，需填写活动名称、活动地点、活动时间、报名时间、简要介绍、活动图片、活动内容等信息，并选择活动分类，完成活动的发布。

(3) 活动分类：可对活动分类进行添加、删除、修改等操作。

#### 5.4.5 文章管理

文章管理由文章列表、添加文章、文章分类这三部分构成，对博物馆文章进行管理操作，具体设计如下：

(1) 文章列表：文章列表中显示已发布文章的名称、分类、发布时间等信息，可就文章发布的时间、标题与分类进行特定查找，对文章进行编辑与删除操作。

(2) 发布文章：对文章进行发布，需填写文章名称、简要介绍、文章图片、文章内容等信息，并选择文章分类，完成文章的发布。

(3) 文章分类：可对文章分类进行添加、删除、修改等操作。

#### 5.4.6 商品管理

商品管理由商品列表、添加商品、商品分类这三部分构成，对博物馆售卖的

商品进行管理操作，具体设计如下：

(1) 商品列表：显示已发布商品的名称、分类、发布时间、价格等信息，可就商品发布的时间、标题与分类进行特定查找，对商品进行编辑与删除操作。

如图 5-56 所示：



图 5-56 商品列表实现效果图

(2) 发布商品：填写商品名称、商品价格、简要介绍、商品图片、商品内容等信息，并选择商品分类，完成商品的发布。

(3) 商品分类：可对商品分类进行添加、删除、修改等操作。

#### 5.4.7 订单管理

主要由订单列表与订单详细页面组成。

(1) 订单列表：在订单管理页面显示订单编号、下单用户、创建时间、订单状态、发货状态等信息，对订单生成时间、订单号、下单用户进行特定查找。此外，可对订单状态与发货状态进行变更与修改。如图 5-57 所示：



图 5-57 订单列表实现效果图

(2) 订单详细：显示订单的具体信息，分基本信息、收件人信息、购买信息三个模块。基本信息里显示订单的下单用户、下单时间、订单状态、发货状态

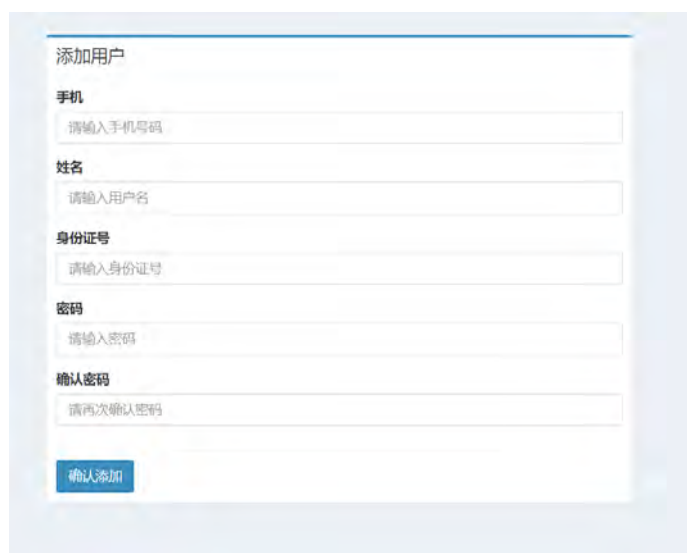
与订单总金额。收件人信息显示收件人的姓名、电子邮件、联系方式与详细地址。购买信息则是对购买商品的名称、具体数量、单价、总价进行呈现。

#### 5.4.8 用户管理

由用户列表、添加用户两个部分组成，可对注册用户进行查看、添加、修改等操作管理。

(1) 用户列表：用户列表中显示注册用户手机号、姓名、身份证号等信息，可就用户姓名、手机号进行精确查找，对用户进行编辑与删除操作。

(2) 添加用户：可在后台直接添加用户，添加原理同注册用户。如图 5-58 所示：



The image shows a web form titled "添加用户" (Add User). It contains five input fields, each with a label and a placeholder text: "手机" (Mobile) with "请输入手机号码" (Please enter mobile number), "姓名" (Name) with "请输入用户名" (Please enter username), "身份证号" (ID Card Number) with "请输入身份证号" (Please enter ID card number), "密码" (Password) with "请输入密码" (Please enter password), and "确认密码" (Confirm Password) with "请再次确认密码" (Please confirm password again). A blue button labeled "确认添加" (Confirm Add) is located at the bottom left of the form.

图 5-58 添加用户实现效果图

#### 5.4.9 消息反馈管理

显示公众对于博物馆建设的相关建议或意见。

## 6 3D 藏品与虚拟展览的设计与实现

博物馆虚拟藏品与展览展示是本系统的特色模块。此模块基于 WEBGL 技术与 Three.js 的使用，利用网页构建了逼真的虚拟场景，全方位的向用户展示了相关藏品与展览，给用户以视觉、听觉上的感知。本章节将对 3D 藏品与虚拟展览的实现作介绍。

### 6.1 3D 藏品

#### 6.1.1 创建 HTML 页面框架

3D 藏品的展示基于 WEB 端，和其他的二维页面实现一样，三维页面也需用到 html、css、JavaScript 等技术进行前端框架的搭建。

```
#percent {  
  position: absolute;  
  width: 200px;  
  height: 20px;  
  color: #c2c2c2;  
  text-align: center;  
  margin-top: 20px;  
  border: 1px;  
}
```

由于展示的主体为 3D 藏品，因此前端框架的搭建较为简单，只是对进度条部分（#percent）进行相关的设计。此外，在<head>元素里会加载需要用到的相关 Three.js 的外部 JavaScript 库。

#### 6.1.2 基本的 Three.js 程序的搭建

一个基本的 Three.js 程序搭建必须包括三个要素，场景、照相机与渲染器。满足三个要素，网页才能渲染出我们想要的 3D 效果。

场景：场景是整个对象（相机、光源、物体）的容器，本身并没有太多的可选项与函数，通过 `scene = new THREE.Scene()` 便可进行创建。

照相机：用来决定哪些场景可被我们看见。更形象的可以理解为人的眼睛。通过移动摄像机的角度，控制摄像机的位置可观察到虚拟空间的每一处角落。可以通过 `var camera = PerspectiveCamera(fov, aspect, near, far)` 进行透视投影相机的

创建。下面，通过图 6-1 来了解相关参数的具体含义：

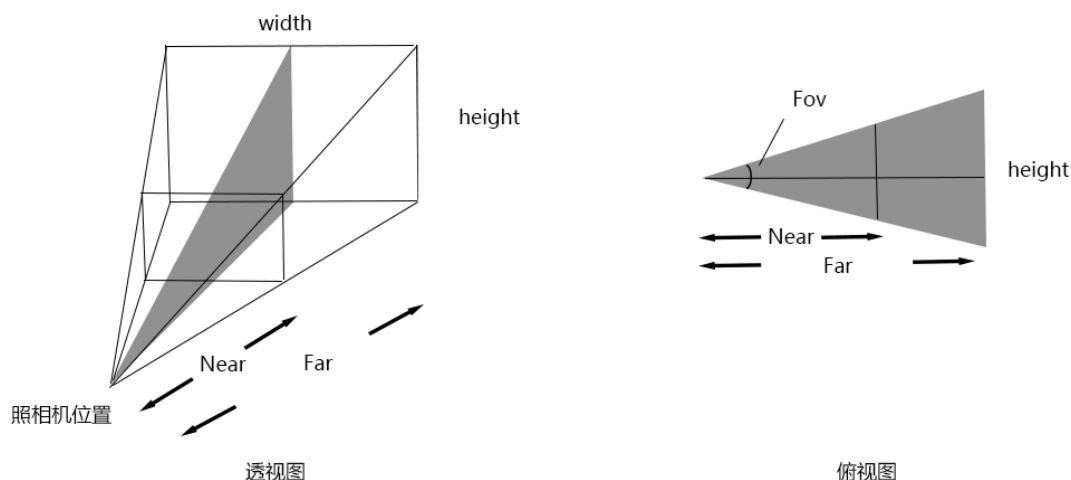


图 6-1 照相机原理图

透视图图中，两个矩形形成的立体梯形部分，为成像景物所在空间的集合，也就是画面所渲染的部分。

- (1) Fov (视锥角度)：相当于眼球张开的角度。视锥角值越大，场景中的物体越小。一般推荐默认角度为 45。
- (2) Aspect (长宽比)：可视区域的横纵比，通常设为 Canvas (画布) 的横纵比例 (`window.innerWidth / window.innerHeight`)。
- (3) Near (近面)：相机能渲染到的最近垂直距离。
- (4) Far (远面)：相机能渲染到的最远垂直距离。

此外，通过 `camera.position.set` 或者 `camera.position.x`、`camera.position.y`、`camera.position.z` 的方式可以对相机初始位置进行设置，并通过 `camera.lookAt` 设置相机的朝向，一般使相机对准场景的中心点即 `scene.position`。

具体设置如下：

```
var camera = new THREE.PerspectiveCamera(45, window.innerWidth /
    window.innerHeight, 0.1, 1000);
camera.position.x = -30;
camera.position.y = 20;
camera.position.z = 30;
camera.lookAt(scene.position)
```

渲染器：负责通过照相机我们看到场景的样子，系统需要呈现的是 3D 图像，所以这里选用的是 WebGLR 渲染器，通过 `var renderer = new THREE.WebGLR`

enderer()进行渲染器创建。通过 `renderer.setClearColor`、`renderer.setSize` 进行渲染器颜色与大小的设置，同时通过 `renderer.shadowMapEnabled = true`，激活阴影效果。

为了使渲染呈现最好的效果，同时采用了实时渲染的方式。在 `function renderScene()`函数中，通过 `requestAnimationFrame(renderScene)`的方式进行递归调用，让动画持续进行。

```
function renderScene() {  
    var clock = new THREE.Clock();  
    var delta = clock.getDelta();  
    trackballControls.update(delta);  
    requestAnimationFrame(renderScene);  
    renderer.render(scene, camera);  
}
```

### 6.1.3 外部模型的导入

创建好场景、相机、渲染器后，需要对 3D 模型进行导入。Three.js 支持各类 3D 模型类型，因 OBJ 格式在日常中使用最多，为此，在本系统使用 OBJ 文件进行导入。

OBJ 文件只是单纯的 3D 模型，如想添加模型纹理则需同时导入 MTL 文件。首先，需要在<head>元素里，导入 `OBJLoader.js`、`OBJMTLLoader.js` 两个库。通过 `var mtlLoader = new THREE.MTLLoader()` 创建 `mtlLoader` 对象，再通过 `mtlLoader.load(filename,onSuccess(materials))`对 MTL 文件进行加载，在 `onSuccess(materials)`中完成对 `objLoader` 对象的创建与加载。其中，通过 `objLoader.load(filename,onSuccess(object ),onProgress(xhr),onError(error))`加载 `objLoader` 对象。具体参数含义如下：

- (1) `filename`：为 obj 的文件名，默认文件位置为当前文件夹。
- (2) `onSuccess(materials)`：加载成功后所执行的函数。
- (3) `onProgress(xhr)`：加载过程中的处理，`xhr` 对象属性用于计算加载进程。
- (4) `onError(error)`：加载失败后的处理。

此外，通过 `object.position`、`object.rotation`、`object.scale.set` 的方式可以对物体初始位置、初始旋转位置、初始大小进行设置。效果如图 6-2 所示。

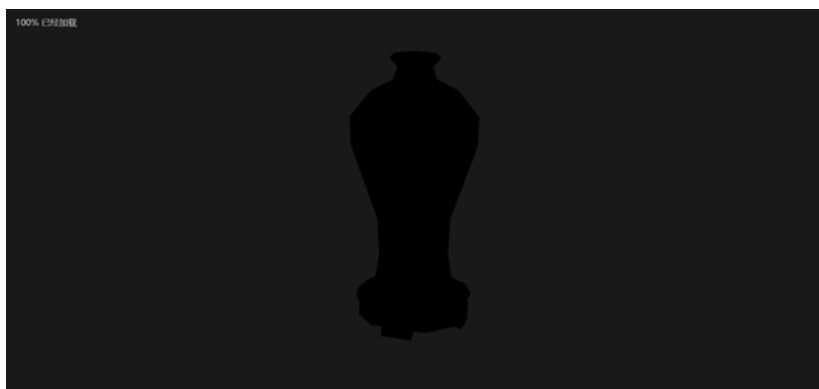


图 6-2 外部模型搭导入实现效果图

#### 6.1.4 添加灯光与阴影

做到这一步，3D 场景已经算是完成了。但此时的场景因为没有添加光源是一片漆黑，什么都看不到的，为此，我们需为场景添加光源。

Three.js 提供环境光、点光源、聚光灯、方向光、半球光等多种类型的光源，为了是展品能够更好的展示，更具有观赏性，系统采用了方向光、半球光、环境光三种光源相结合的方式。

创建光源需要创建对象，设置光的位置，添加到场景这三步，下面以半球光为例进行光源的创建。效果如图 6-3 所示。

- (1) 创建对象：var hemiLight = new THREE.HemisphereLight(groundColor, color, intensity);其中 groundColor 为天空的反光颜色，color 为地面的反光颜色，intensity 为光的强度。
- (2) 设置光的位置：通过 hemiLight.position.set(x,y,z)设置光线 x、y、z 的位置。
- (3) 添加到场景：scene.add(hemisphereLight);即可。

```
hemisphereLight = new THREE.HemisphereLight(0xffffff, 0xffffff,  
0.25);  
hemiLight.position.set(100, 0, 100);  
scene.add(hemisphereLight);
```



图 6-3 添加光源实现效果图

### 6.1.5 添加交互事件

为了是用户能从任意角度观察到 3D 藏品的各处特征与纹理，实现移动鼠标便可以控制相机的效果，我们需在相机上绑定相关的控件，对相机进行控制，实现用户与场景的交互。先在<head>元素里，添加轨迹球控件 `trackballControls.js`，通过 `trackballControls = new THREE.TrackballControls(camera);` 创建控件并绑定在相机上，相关设置说明如下：

- (1) 通过 `rotateSpeed` 设置按住鼠标左键后，拖动查看 3D 藏品时画面的旋转速度。
- (2) 通过 `zoomSpeed` 设置用滚轮调整画面远近时的速度。
- (3) 通过 `panSpeed` 设置按住鼠标右键后的平移速度。
- (4) `noZoom = false`、`noPan = false` 的参数默认为 `false`，如为 `true`，默认为禁用鼠标滑轮与平移功能。
- (5) 通过 `dynamicDampingFactor` 设置控制的灵敏度。

对 3D 藏品放大查看，实现效果如图 6-4 所示。



图 6-4 放大实现效果图



对 3D 藏品缩小查看，实现效果如图 6-5 所示。

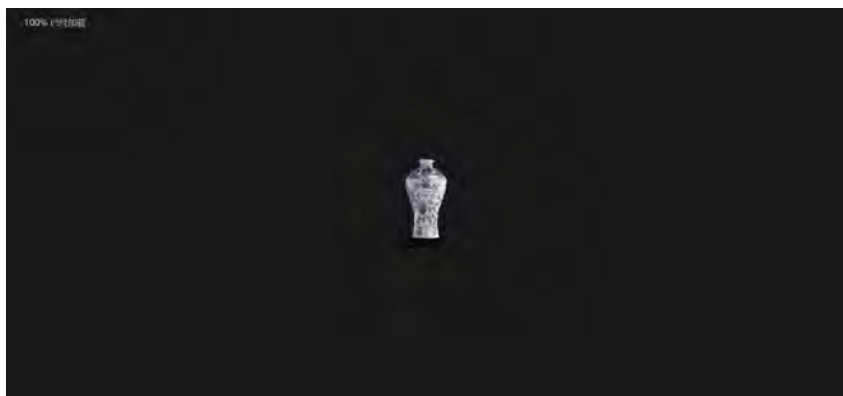


图 6-5 缩小实现效果图

对 3D 藏品进行拖拽，实现效果如图 6-6 所示。



图 6-6 对展品拖拽实现效果图

#### 6.1.6 3D 展示后台管理

通过以上的五个步骤，3D 模型的设计基本已经完成。管理员可在后台藏品管理模块对 3D 藏品进行相关的查看、发布、删除、编辑等操作。主要由 3D 藏品列表与发布 3D 藏品两个模块组成。

**3D 藏品列表：**显示 3D 藏品名称、发布时间等信息，可就藏品名称特定查找。并可对 3D 藏品进行编辑与删除。如图 6-7 所示：



图 6-7 3D 藏品列表实现效果图

**发布 3D 藏品：**除了填写藏品名称、藏品图片、藏品内容等信息外，还需上

传 obj 文件、mtl 文件、纹理图等 3D 藏品特有的信息。如图 6-8 所示：

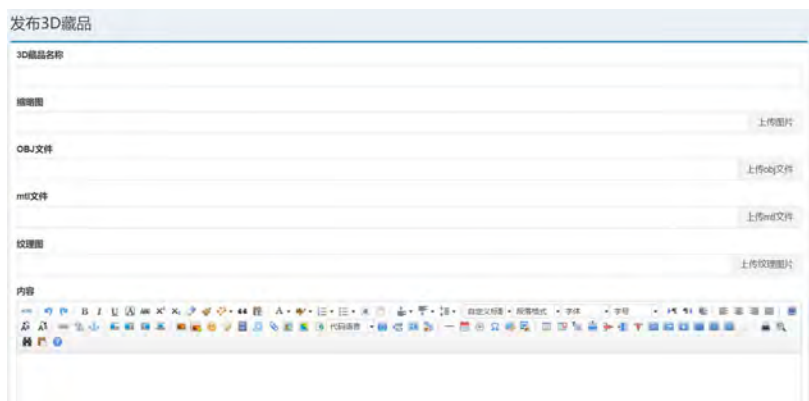


图 6-8 发布 3D 藏品实现效果图

## 6.2 虚拟展览

虚拟展览的实现原理与 3D 藏品的实现大致相同，只不过更为复杂一些，现就实现的过程进行表述。

### 6.2.1 创建 HTML 页面框架

与 3D 藏品的展示一样，虚拟展览的展示同样需要以前端页面为载体进行页面的呈现。

### 6.2.2 基本的 Three.js 程序的搭建

对虚拟展厅进行场景、照相机与渲染器的创建，其中，场景与照相机的创建与 3D 藏品部分创建相似，这里就不详细阐述。渲染器除了创建 WebGLR 渲染器对象，通过 `requestAnimationFrame(animate)` 函数对画面进行实时的渲染外，同时需执行 `window.addEventListener("resize", onWindowResize, false)` 与 `document.addEventListener("mousedown", onDocumentMouseDown, false)` 两条语句对 `onWindow-Resize`（画面大小重置）与 `onDocumentMouseDown`（鼠标点击）行为进行监听。

### 6.2.3 展厅环境的构建

完成基本的 Three.js 程序的搭建后，需要为整个场景添加地面、天花板、墙壁、灯光效果等相关环境因素。其中，加入灯光效果与 3D 藏品部分创建相似，地面、天花板、墙壁则较为复杂，现以如何添加地面进行描述。

- （1）创建纹理 Texture 对象：通过 `var floorTexture = new THREE.ImageUtils.loadTexture(file)`，返回纹理对象，从而对纹理的贴图进行加载，file

参数为纹理贴图路径。

- (2) 对纹理进行属性设置：对 Texture 纹理属性 wrapS、wrapT、repeat 进行设置。使 floorTexture.wrapS = floorTexture.wrapT = THREE.RepeatWrapping，并通过 floorTexture.repeat.set(x,y); 设定在 x 与 y 方向上的重复次数，达到纹理平铺重复的效果。
- (3) 建立材质 Material 对象：将纹理对象 texture 赋值给材质对象的贴图属性 map，并设置属性 side，指定几何体的哪个面应用了材质，从而完成材质对象的创建。
- (4) 创建几何体 Geometry 对象：通过 var floorGeometry = new THREE.PlaneGeometry(width, height, widthSegments, heightSegments); 建立一个可以用于纹理映射的二维平面几何体。width 和 height 分别表示沿着 X 轴与 Y 轴的宽度与高度，默认值为 1；widthSegments 和 heightSegments 表示宽度与高度的分段数，默认值也为 1。
- (5) 建立并添加网格：单独的几何体对象或者材质对象是不能被渲染的，需要只有二者结合成网格（mesh）对象后才能被渲染到屏幕上。之后可对网格的位置、旋转、大小进行初始化设置，通过 scene.add(mesh) 添加到场景中，即完成对地面的添加操作。

实现效果如图 6-9 所示：



图 6-9 环境构建实现效果图

#### 6.2.4 添加展览作品

虚拟展厅由六面构成，作品的展示位于展厅的左、右、前、后四面，为此在添加展览图时，应按每一面进行添加。

以添加作品在左面墙为例，通过函数 `addToLeft(a,b,c,d,e)` 对作品信息进行添加，其中 `a` 表示图片宽度，`b` 表示作品的 `z` 轴位置，`c` 表示作品的图片源，`d` 为作品的背景文件源，`e` 为作品展示文字。

与添加地面步骤类似，需对作品创建纹理 `Texture` 对象、材质 `Material` 对象与几何体 `Geometry` 对象，并添加网格（`mesh`）使作品渲染至屏幕上。并对作品的 `x`、`y`、`z` 轴各方向的位置与旋转角度进行设定。

将背景文件源与作品展示文字放入 `objectDescription[]` 数组中存储本地，通过 `scene.add()`，`objects.push()` 两个方法实现作品与作品文件的添加。

实现效果如图 6-10 所示：



图 6-10 添加作品实现效果图

#### 6.2.5 添加交互事件

目前所实现的场景均为加载完成，用户不能与场景产生任何的联系。为了能使用户更好地体验虚拟场景，我们还需为场景添加一系列的交互事件。

##### （1）添加鼠标触发事件

添加 `onDocumentMouseMove`（鼠标移动）、`onDocumentMouseUp`（鼠标松开）、`onDocumentMouseDown`（鼠标点击）、`onDocumentMouseOut`（鼠标离开某元素）事件，使用户能在虚拟空间进行“移动”，实现视角的变化。

## （2）添加媒体事件

为了更为全面、多视角的展示作品信息，我们对虚拟场景进行了文字与声音的添加。在点击作品以及点击浏览后的初始界面会触发相关的媒体事件，显示相应的文字信息与语音播放。页面的左上方可对声音的音量进行调节操作。

以击浏览后初始界面的文字与语音添加为例，通过 `function welcome(a,b)` 将声音与文字进行添加，其中 `a` 表示背景文件源，`b` 表示展示文字。当点击“点击浏览”按钮之后，加载图标与按钮消失，执行 `welcomeSpeak()` 函数。通过 `new Audio()` 方法创建加载音频对象，并通过 `voiceGuide.play()` 等语句实现语音播放；在文字展示部分，利用 `js` 中 `Document` 的 `getElementById()` 方法，对信息与信息栏进行呈现。在音频播放结束之后，执行 `voiceGuide.onended()` 事件，信息与信息栏也随之消失。实现效果如图 6-11 所示：

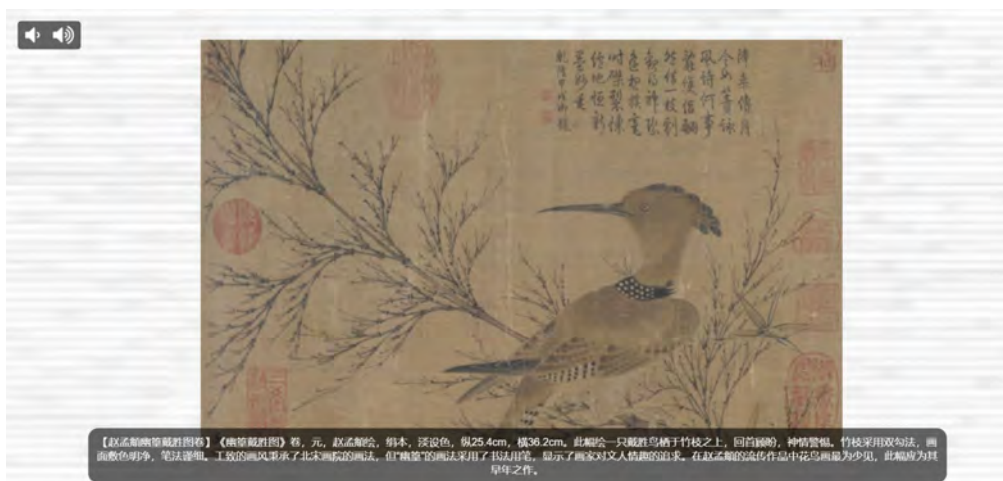


图 6-11 声音、文字实现效果图

以下代码调节音量降低的实现方法。

```
document.getElementById("controls_minus").onclick = function()
{
  if (voiceGuideVolume-0.1>0)
  {
    voiceGuideVolume = voiceGuideVolume - 0.1;
    voiceGuide.volume = voiceGuideVolume;
  }
}
```

## 7 系统测试

测试的根本目的在于检测系统是否存在问题，能否满足用户的使用需求。在此系统的测试过程中，主要采用白盒测试与黑盒测试两者相结合的测试方法，根据测试结果进一步对系统进行修改与完善，使系统能符合设计要求。

### 7.1 系统测试环境

操作系统 Windows 7 旗舰版 64 位 SP1 ( DirectX 11 )

电脑型号 技嘉 B85M-D3V Plus 台式电脑

处理器 英特尔 第四代酷睿 i5-4590 @ 3.30GHz 四核

主板 技嘉 B85M-D3V Plus ( Lynx Point )

内存 16 GB ( GLOWAY DDR3 1600MHz )

主硬盘 西数 WDC WD10EZEX-00BN5A0 ( 1 TB / 7200 转/分 )

数据库管理系统: MySQL

浏览器: 谷歌浏览器

### 7.2 功能性测试

此处选取具有代表性的测试用例与结果进行呈现。

表 7.1 描述用户注册、登录模块，验证用户在注册与登录时输入错误信息或信息不规范时，页面是否会进行提示；验证用户正确填写信息后，能否完成注册、登录的流程。

表 7.1 注册、登录用户测试用例表

编号	操作描述	预期结果	实际结果
1	注册内容未填全	页面提示需填全	与预期相符
2	联系方式输入已被注册	页面显示改手机号已被注册	与预期相符
3	填写身份证号不是 18 位	页面提示需输入镇正确的证件号	与预期相符

4	设置密码少于 6 字符 或多余 20 字符	页面提示密码格式有误	与预期相符
5	前后密码输入不一致	页面提示密码输入不一	与预期相符
6	登录内容未填全	页面提示需填全	与预期相符
7	登录信息错误（联系方式未注册、密码输入错误）	页面提示输入错误	与预期相符

表 7.2 描述用户预约模块，验证用户在信息填写正确的情况下，是否能实现个人与团体预约。

表 7.2 用户预约测试用例表

编号	操作描述	预期结果	实际结果
1	个人预约，相关信息填写合法且正确	数据库中插入此条预约记录	与预期相符
2	团体预约，相关信息填写合法且正确	数据库中插入此条预约记录	与预期相符
3	信息未填全或有遗漏	页面提示信息未填写	与预期相符

表 7.3 描述博物馆虚拟展览模块，验证用户能否正常访问与使用虚拟展览功能，并进行相关操作。

表 7.3 虚拟展览测试用例表

编号	操作描述	预期结果	实际结果
1	点击虚拟展览	可进入展览加载页面	与预期相符
2	点击开始浏览	进入虚拟展览，场景加载完成，且有文字语音介绍	与预期相符
3	点击某一作品	画面视角来到该作品前，并有文字语音介绍	与预期相符
4	鼠标长按页面移动	画面视角随鼠标移动方向发生变化	与预期相符

5	再次点击另一作品	画面视角切换到该作品前，并有文字语音介绍	与预期相符
6	点击左上角音量调节	可对声音音量进行调节	与预期相符

表 7.4 描述博物馆 3D 藏品模块，验证 3D 藏品模块各功能操作能否正常运行。

表 7.4 3D 藏品模块测试用例表

编号	操作描述	预期结果	实际结果
1	加入收藏	收藏成功	与预期相符
2	点击图片进入 3D 展示页面	跳转展示页面，3D 藏品已加载或加载完成	与预期相符
3	展示页面内，鼠标滑轮向前滑动	页面放大	与预期相符
4	展示页面内，鼠标滑轮向后滑动	页面缩小	与预期相符
5	展示页面内，鼠标左键长按+鼠标滑动	场景拖动	与预期相符

表 7.5 描述文创商城模块，验证加入购物车、更新商品数量、移除商品、清空购物车、提交订单等功能能否正常操作。

表 7.5 文创商城模块测试用例表

编号	操作描述	预期结果	实际结果
1	用户加入商品到购物车	如登录用户加入购物车成功，未登录用户要求先登录	与预期相符
2	修改购物车内的商品数量	购物车内的商品数量已更新	与预期相符
3	删除购物车内某商品	该商品已移除购物车	与预期相符
4	将购物车的商品全部移除	购物车清空	与预期相符



5	结算购物车内商品	转入订单支付页面	与预期相符
6	输入收件信息后，提交订单	订单提交成功，个人中心收到下单成功提醒	与预期相符

表 7.6 描述活动管理模块，验证管理员能否查看已发布活动、删除活动、修改活动、发布活动以及能否对活动分类进行添加、删除、修改等操作。

表 7.6 活动管理测试用例表

编号	操作描述	预期结果	实际结果
1	查看已发布活动内容	显示活动名称、时间、分类地点等信息。	与预期相符
2	删除某一活动	可删除该活动	与预期相符
3	修改某一活动信息	可对活动进行修改	与预期相符
4	查看活动报名情况	显示活动报名的人员信息	与预期相符
5	发布活动	可对活动进行发布	与预期相符
6	添加活动分类	添加分类成功	与预期相符
7	修改活动分类	可对分类进行修改	与预期相符
8	删除活动分类	可删除该分类	与预期相符

### 7.3.非功能性测试

根据第三章需求分析的结果，现就安全性、兼容性、可拓展性、性能这几方面进行系统的非功能性测试。测试过程与结果见表 7.7

表 7.7 非功能性测试用例表

内容	操作描述	预期结果	实际结果
安全性	查看系统内容是否只能由管理员进行操作管理	只有管理员能登录后台管理系统	与预期相符
安全性	用户信息是否加密	用户信息只有管理员可见，且密码均进行加密。	与预期相符
兼容性	win10、win8、win7、	系统可兼容以上主流系	与预期相符

	苹果系统运行系统	统，并可顺畅运行	
可扩展性	添加任意一功能，如设置商品评论功能	可以添加	与预期相符
性能	进行预约、购买、3D预览等操作，统计页面切换、数据响应时间	页面切换、数据响应的 时间均小于 3s	与预期相符

## 8 总结与展望

### 8.1 现阶段工作总结

基于博物馆行业在数字化的发展与需求与中小型博物馆网站质量参差不齐的现状，本课题以博物馆线上服务系统的设计与实现为主题，采用 Django 框架，python 语言以及 WEBGL，Three.js 等相关技术作支撑，进行系统的搭建工作。目前，系统的各方面工作已基本完成。

在开发系统之前，对系统进行了充分的可行性分析，确定本系统能如期按计划完成。并基于博物馆的现状与需求对整个系统从游客、用户、管理员三者的角度对系统进行了功能性的需求分析以及从系统本身角度对系统进行了非功能性需求分析。

整个系统的内容可以划分为前端与后端两个模块，前端模块注册与登录、用户中心、博物馆展览、博物馆馆藏、博物馆资讯、博物馆活动、博物馆服务、文创商城等模块，后端部分包括预约管理、展览管理、活动管理、资讯管理、商品管理、订单管理、馆藏管理、用户管理、消息反馈管理等模块。

除了这些常规模块的设置，为了能适应数字化发展的需要，同时给公众提供最真实且优质的服务，系统设计了 3D 藏品与虚拟展览两个模块。用户可足不出户，便可在线上体验与感受博物馆藏品与展览。

通过数据库与各模块的详细设计，系统在完成搭建之后，针对系统功能，对各模块进行了详细且严谨的测试。通过测试表明，此系统符合要求，可正常运行。至此，现阶段的工作暂告一段落。

### 8.2 进一步工作展望

虽然本系统的功能已基本实现，且能基本满足博物馆日常的需求，但系统仍然存在这一些不足与可以进一步完善的地方。希望能在日后能进一步探索、研究与改进。

- (1) 支付接入：在文创商城模块可接入支付宝、微信、银联等接口，实现在线支付。

- (2) 身份验证：在用户实名注册时，与公安部相关认证系统相联合，确保用户输入信息匹配且正确。
- (3) 系统上线：系统目前只处于系统本地运行阶段，后期可搭建服务器，进行相关部署工作，实现线上访问。
- (4) 丰富虚拟展览功能：目前实现的虚拟展览只是最为基本的形式之一，后期可对虚拟技术进行详细的研究，实现可还原博物馆展览实景，进行全景漫游访问。