

纯小白如何用AI创建个人博客?

How a complete beginner can use AI to
create a personal blog

AI模型

演讲人：张浩宸 | 博客创作：冯朝晨 王迟迟 | 制作：徐思涵 万程冰

成员分工



组长 | 冯朝晨：负责博客网站制作、PPT与Word优化

成员 | 王迟迟：负责博客网站的制作、Word后期修改
徐思涵：负责Word与PPT的整体制作
万程冰：负责Word与PPT的整体制作
张浩宸：负责PPT后期修改与汇报

CONTENTS

目录



01 | 博客网站介绍
Introduction to a Blogging Website

02 | 制作过程
Production Process

03 | 感悟与展望
Reflection and Outlook

博客网站介绍

Introduction to a Blogging
Website





博客网站是一个用于发布个人或团队内容的在线平台，通常用于记录、展示和分享信息。博客内容可以包括文字、图片、视频、音频以及代码片段等，并根据创作者的目标和风格不同，可以涵盖从生活日志、教程、技术分享到观点表达等各种类型。

博客网站的基本功能：

- ◆ 内容发布：允许用户发布各种类型的文章，内容可以是生活记录、学习心得、技术教程等。
- ◆ 分类管理：一般支持对文章进行分类或打标签，使得内容管理和查找更加方便。
- ◆ 评论和互动：支持读者在每篇文章下方留言，形成互动，有助于用户获得反馈。
- ◆ 社交分享：通常提供一键分享功能，让用户可以将文章分享到社交平台，扩大阅读量。
- ◆ 订阅和推送：通过RSS或邮件订阅功能，让关注者及时获取最新更新。



本次汇报的主要内容为
作为一个纯小白
怎么使用AI
来制作一个简单的
个人的博客网站



制 作 过 程

Production Process



我现在是一个代码小白，我想制作一个个人博客，用于展示我的PPT和Word文档，我应该采用什么编程语言和什么开发框架去完成这个工作？



记忆已更新

作为代码新手，推荐你选择相对简单、支持丰富的开发语言和框架。以下是几种适合创建个人博客的网站技术组合：

1. 使用Python和Flask/Django框架

- **编程语言：**Python
- **推荐框架：**Flask（轻量级）或Django（功能更丰富）
- **特点：**这两种框架都适合新手，拥有清晰的结构，并且社区支持很好。Django自带的管理后台可以让你轻松上传和管理文档。

实现步骤：

- 创建用户界面以展示PPT和Word文档的链接或预览。
- 利用Flask或Django的模板系统渲染页面，管理内容的展示。
- Django拥有强大的文件处理能力，可以轻松上传和管理不同格式的文档。

2. 使用JavaScript和Node.js + Express框架

- **编程语言：**JavaScript
- **推荐框架：**Express.js（轻量级框架）
- **特点：**Node.js和Express搭配使用适合那些希望全栈开发都使用JavaScript的开发者。它有丰富的库，可以处理文件上传和管理静态资源。

实现步骤：

- 使用Express.js创建后端API来管理文档的上传和下载。
↓
- 前端可以使用简单的HTML+CSS渲染页面，也可以搭配一些前端框架（如Vue、React）增加页



给“ChatGPT”发送消息



既然是作为一个纯小白
那首先需要询问ChatGPT
作为一个代码小白
怎么来制作一个个人的博客网站



作为代码新手，推荐你选择相对简单、支持丰富的开发语言和框架。以下是几种适合创建个人博客的网站技术组合：

1. 使用Python和Flask/Django框架

◆ 特点：这两种框架都适合新手，拥有清晰的结构，并且社区支持很好。

2. 使用JavaScript和Node.js + Express框架

◆ 特点：Node.js和Express搭配使用适合那些希望全栈开发都使用JavaScript的开发者。

3. 使用前端框架如WordPress（非编程方案）

◆ 特点：如果你暂时不打算深入编程，可以考虑使用WordPress。它有很多现成的主题和插件，适合快速搭建展示型博客。

推荐方案总结：如果你对编程有浓厚兴趣，建议使用Python + Flask或Django的组合



结合ChatGPT给出的建议，我们选择了Python + Django的组合来进行个人博客网站的开发。

如何通过Python + Django的组合来制作个人博客，我们继续对AI进行提问：“我打算使用Python和Django框架来完成个人博客网站的制作，那我应该怎么去做？我想把页面做的精美一点。”



使用Python和Django制作个人博客网站是个很好的选择，特别是对于展示文档类内容。以下是逐步实现过程，以及一些提升页面美观的建议：

1. 搭建Django开发环境

◆ 安装Django：首先安装Python环境，然后在终端输入命令安装Django，使用Django命令创建一个新项目，在项目中创建一个应用，用于管理博客相关功能，运行开发服务器

2. 设置基础模型和数据库

◆ 定义数据库模型，迁移数据库，创建上传和展示功能

3. 提升页面美观度

◆ 引入CSS框架：使用Bootstrap或Tailwind CSS等CSS框架，让页面更美观。

◆ 自定义样式：在项目的静态文件夹中创建自定义CSS文件。

这套方案可以帮助你快速上手Django开发博客，并通过引入样式和CSS框架，提升页面的美观度！



ChatGPT一次性给出了较多方面的建议
但是并不是很具体
所以我们分步骤来慢慢搭建个人的博客网站

1

搭建Django开发环境



3

博客网站具体开发



2

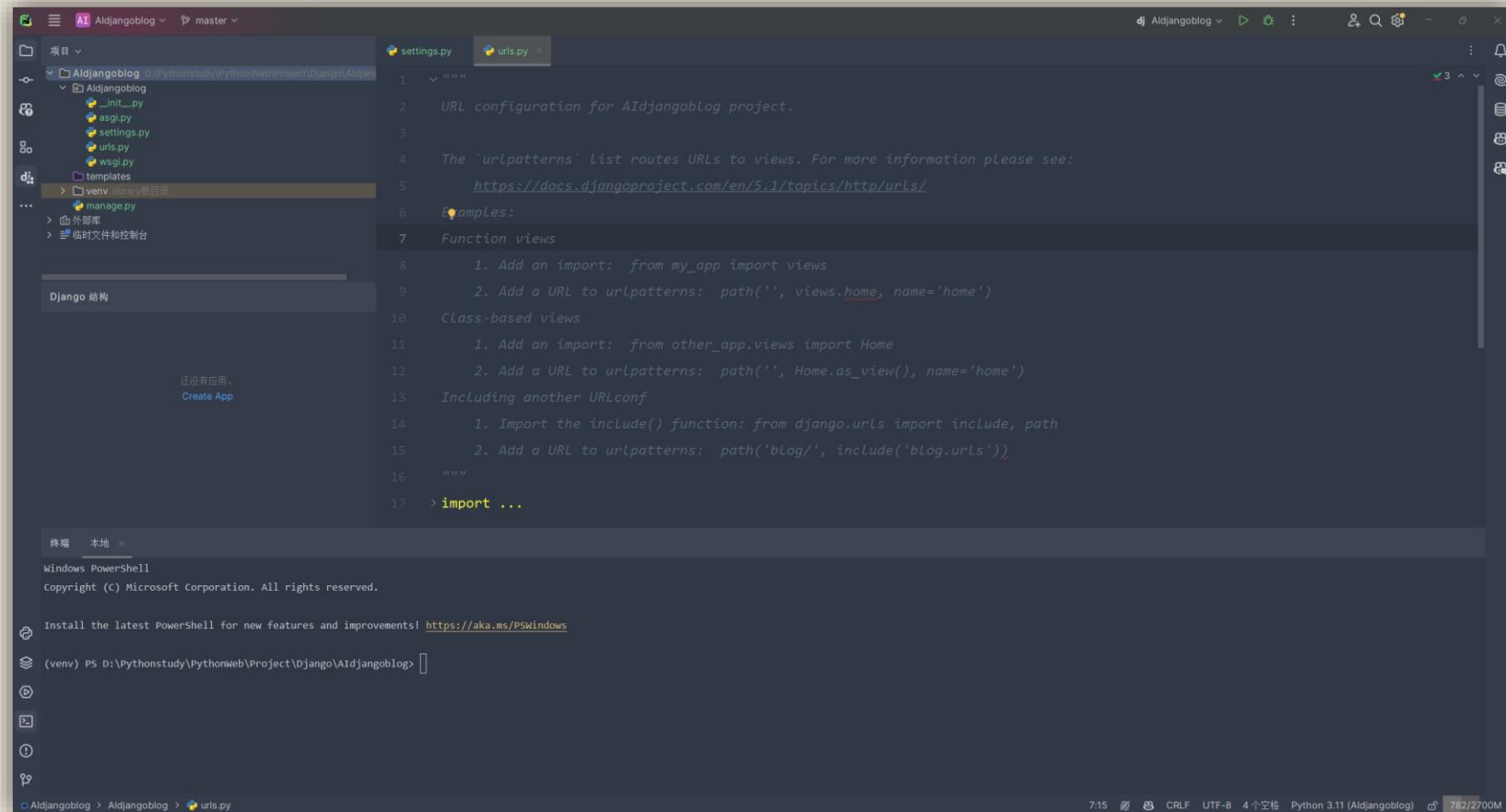
设置基础模型和数据库





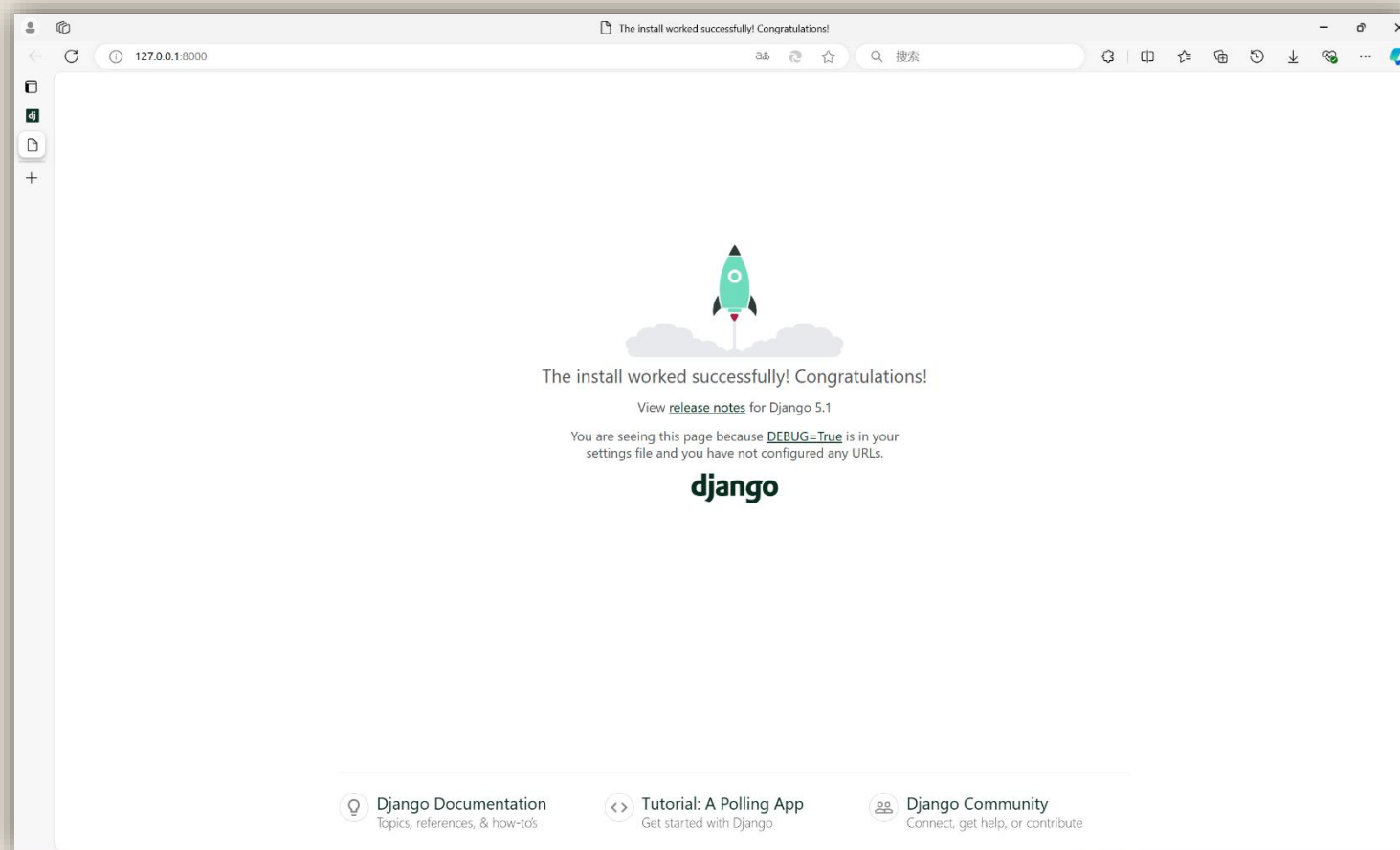
首先是搭建Django开发环境

我们利用Pycharm来快速搭建一个Django项目 “Aldjangoblog”，为了避免python库的冲突，我们在虚拟环境中进行项目的开发。





在创建好项目之后，我们需要在项目中创建一个应用“blog”，用于管理博客相关功能。
在命令行运行创建APP的命令后，项目目录中多了一个“blog”文件夹，表示APP创建成功。
随后我们初步启动了我们的项目，在浏览器打开网址，如图所示：



对于我上面的项目，我怎么修改我的设置去连接我的mysql数据库，同时告诉我应该怎么新建一个数据库来存储相应的内容



要在Django项目中连接MySQL数据库，您需要进行以下步骤：

1. 安装MySQL驱动

首先，确保您安装了MySQL数据库的Python驱动。可以使用 `mysqlclient` 或者 `PyMySQL`。这里以 `mysqlclient` 为例：

在命令行中运行：

```
bash

pip install mysqlclient
```

复制代码

如果安装过程中遇到问题，您也可以使用 `PyMySQL`，安装命令如下：

```
bash

pip install PyMySQL
```

复制代码

2. 修改Django项目的 settings.py

打开您的Django项目的 `settings.py` 文件，找到 `DATABASES` 部分，并将其修改为使用MySQL。示例配置如下：

```
python

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql', # 使用MySQL引擎
        'NAME': 'your_database_name',      ↓ # 数据库名称
        'USER': 'your_username',           # 数据库用户名
```

复制代码



给“ChatGPT”发送消息



在项目已经可以成功启动的基础上，我们进行下一步的工作，设置基础模型和数据库。为了正确配置数据库，我们向ChatGPT进行了提问：“对于我上面的项目，我怎么修改我的设置去连接我的mysql数据库，同时告诉我应该怎么新建一个数据库来存储相应的内容？”



根据ChatGPT给出的回答，我们首先在命令行中，使用mysql数据库，创建了一个 “aiblog” 数据库用于本项目

```
mysql> CREATE DATABASE aiblog CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
Query OK, 1 row affected (0.02 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| aiblog   |
| cumt_software_cup |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
| xdl_cumt |
| xm01     |
+-----+
8 rows in set (0.00 sec)
```



```
(venv) PS D:\Pythonstudy\PythonWeb\Project\Django\AIdjangoblog> pip install mysqlclient
Collecting mysqlclient
  Downloading mysqlclient-2.2.5-cp311-cp311-win_amd64.whl.metadata (4.8 kB)
Downloading mysqlclient-2.2.5-cp311-cp311-win_amd64.whl (207 kB)
Installing collected packages: mysqlclient
Successfully installed mysqlclient-2.2.5
(venv) PS D:\Pythonstudy\PythonWeb\Project\Django\AIdjangoblog> 
```

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql', # 使用MySQL 引擎
        'NAME': 'aiblog',                    # 数据库名称
        'USER': 'root',                      # 数据库用户名
        'PASSWORD': 'root',                 # 数据库密码
        'HOST': 'localhost',                # 数据库主机（一般为localhost）
        'PORT': '3306',                     # MySQL 默认端口
    }
}
```

然后在项目的虚拟环境中安装MySQL数据库的驱动库mysqlclient

然后修改项目的settings.py配置文件，去连接我们刚刚创建好的数据库。



现在我的项目创建与数据库配置已将完成，为了开发我的博客网站，我的下一步工作应该是什么？要怎么完成？我希望在我的网站上去展示我的PPT与Word文档，结合我的需求，下一步应该怎么开发？



记忆已更新

为了在您的博客网站上展示PPT和Word文档，您可以添加文件上传和展示功能。这包括文件的上传、存储、展示和下载。以下是分步指导：

1. 创建文件模型

在 `models.py` 中定义模型，存储上传的文件以及文件的基本信息。

```
python

from django.db import models

class Document(models.Model):
    title = models.CharField(max_length=200)
    file = models.FileField(upload_to='documents/')
    uploaded_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.title
```

复制代码

2. 迁移模型到数据库

在命令行中运行以下命令，创建数据库表：

```
bash

python manage.py makemigrations
```

复制代码



给“ChatGPT”发送消息



ChatGPT 也可能会犯错。请核查重要信息。

修改完数据库的配置之后，项目的基本准备已经完成，下一步就是博客网站的具体开发了。为了能在网站上去展示我们的PPT与Word文档，我们向ChatGPT询问我的下一步的开发工作与实现方法。



```
1  from django.db import models
2
3
4  # Create your models here.
   2 个用法
5  class Document(models.Model):
6      title = models.CharField(max_length=255)
7      upload_date = models.DateTimeField(auto_now_add=True)
8      file = models.FileField(upload_to='documents/')
9
10  @↑    def __str__(self):
11         return self.title
12
13
14  # 2 个用法
15  class PPTDocument(models.Model):
16      title = models.CharField(max_length=255)
17      upload_date = models.DateTimeField(auto_now_add=True)
18      file = models.FileField(upload_to='ppt_documents/')
19
20  @↑    def __str__(self):
        return self.title
```

首先要创建文件模型，去存储我们上传的Word文档与PPT，并记录一些基本的信息

对于具体的代码实现，我们使用了Copilot去帮助我们完成代码，创建了两个模型用于分别存储Word文档与PPT。

然后在后台管理面板中注册这两个模型便于后台管理，随后又修改了项目配置文件去处理文件的上传，再在路由配置文件里面添加MEDIA_URL和MEDIA_ROOT来处理文件上传。



我想为我的博客网站设计一个主页，希望页面精美一点，然后突出“AI大模型与智慧办公”这几个字，请你帮我设计一个页面，给出代码，同时给出后端的视图函数代码

为了设计一个精美的主页，并突出“AI大模型与智慧办公”主题，我们可以将页面分为几个区域：顶部导航栏、主页主要内容展示区域、底部区域等。可以使用Bootstrap等CSS框架来简化布局和样式，同时提升页面的美观度。

1. HTML模板代码

在 `templates/blog/home.html` 文件中创建主页模板，设置页面的布局、样式和内容。

```
html

<!DOCTYPE html>
<html lang="zh">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>AI大模型与智慧办公</title>
  <!-- 引入 Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
  <style>
    /* 自定义样式 */
    .hero-section {
      background: url('https://your-image-link.com/background.jpg') no-repeat center;
      background-size: cover;
      height: 80vh;
      color: white;
      display: flex;
      align-items: center;
```

复制代码



给“ChatGPT”发送消息

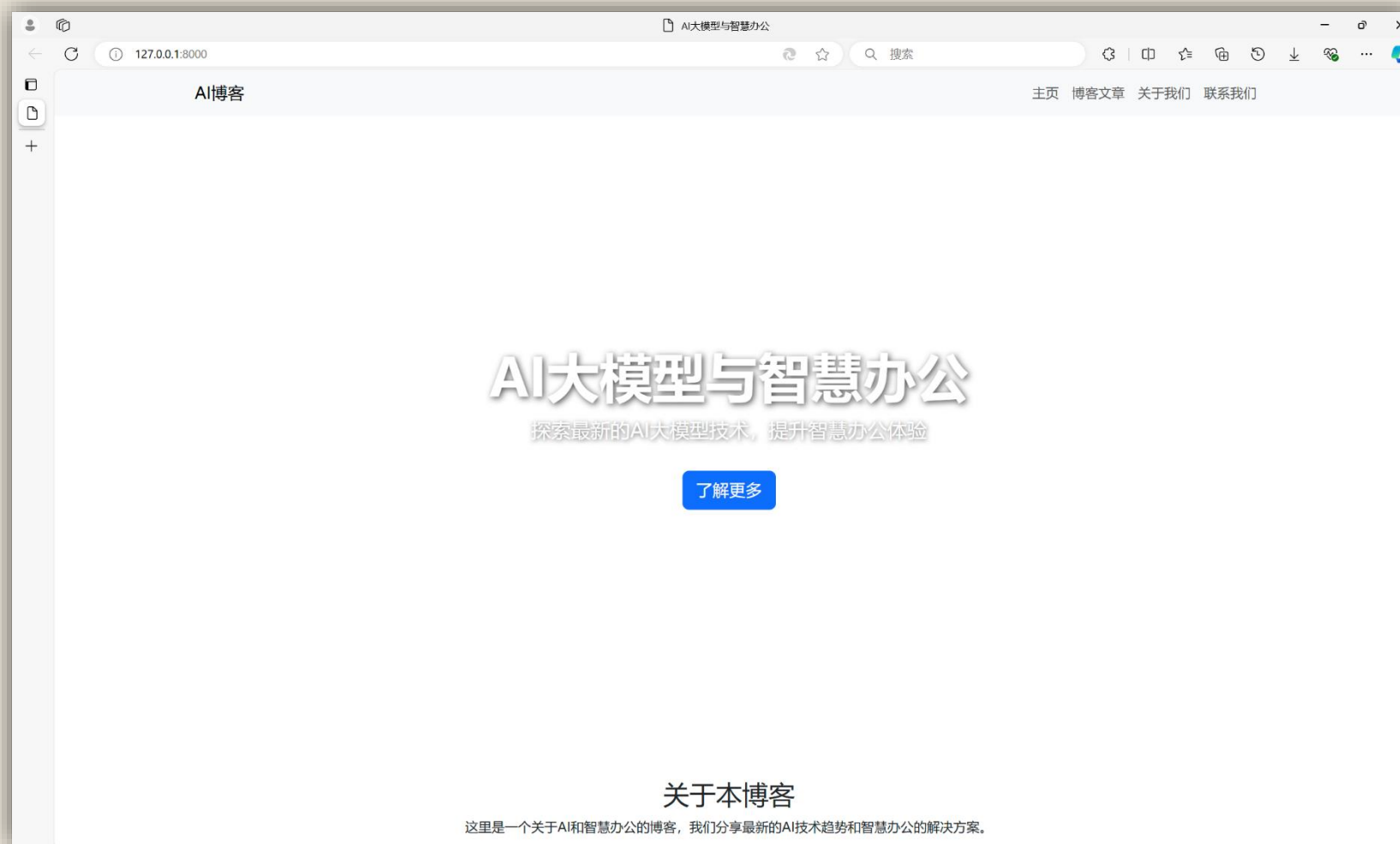


ChatGPT 也可能会犯错。请核查重要信息。

完成了上述工作之后，我们开始来设计我们的网站主页。我们先对ChatGPT提出了我们对网站主页的设计要求。

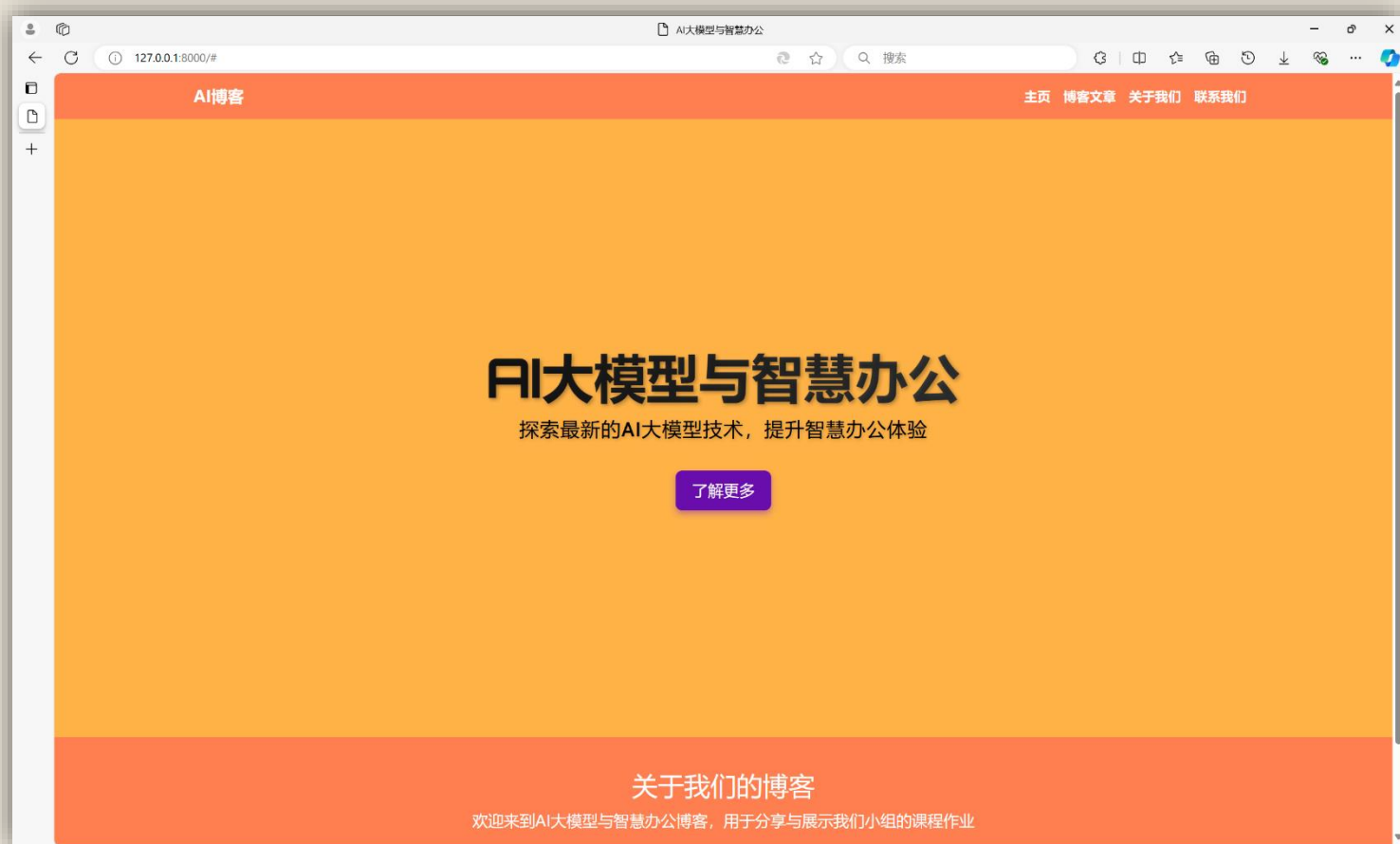


使用ChatGPT给出的代码与应该进行的操作，我们先对ChatGPT初步给出的代码构建的界面进行了预览



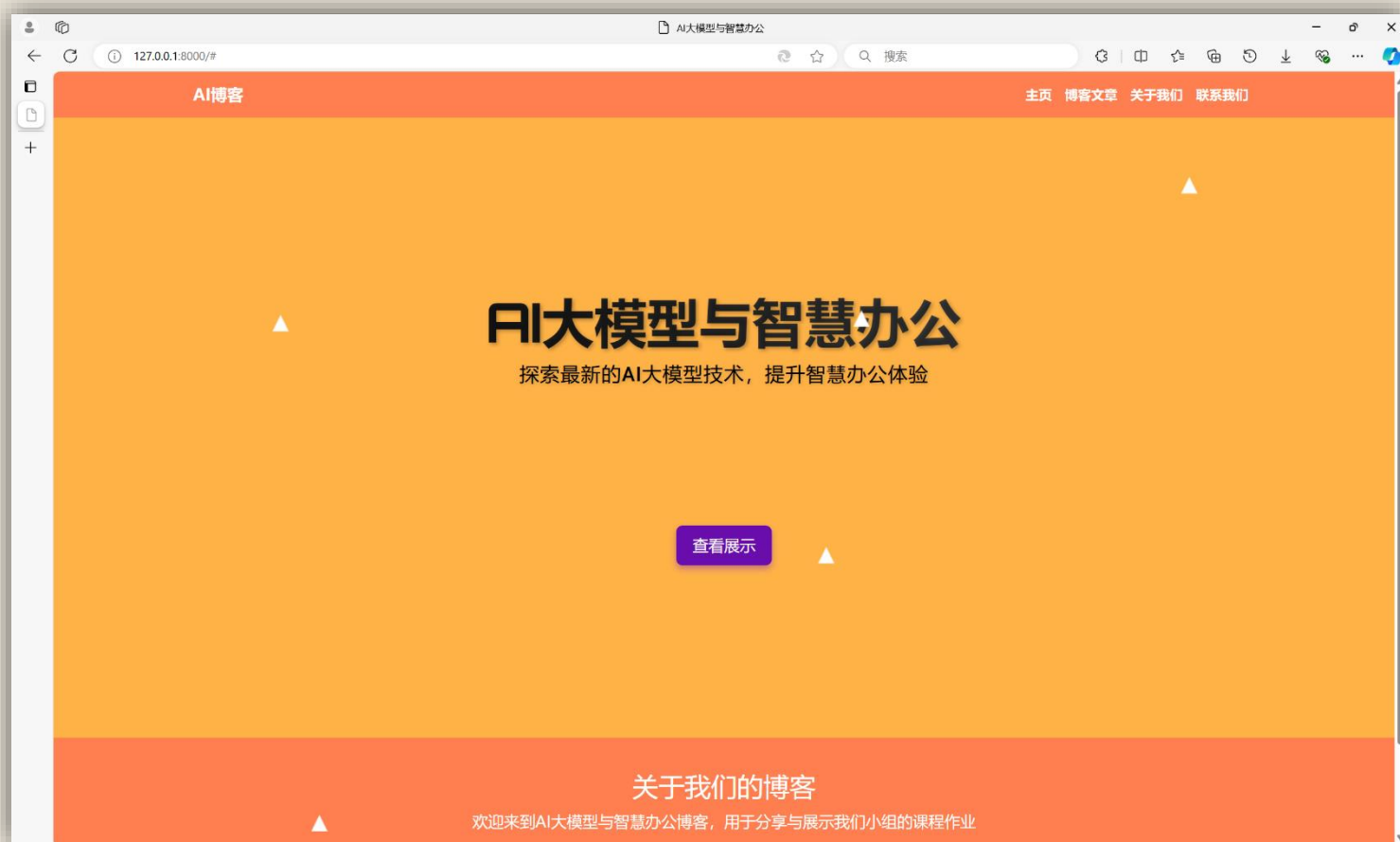


对于现有的界面，我们感觉界面颜色较为单一，所以我们首先改变了一下页面的颜色，使之看起来靓丽一点；同时将字体修改为黑色，加上一点效果，让字体看起来科技一点；并对现有的按钮进行了修改，让鼠标悬停的时候可以有炫酷的效果。多次修改后得到了下图：



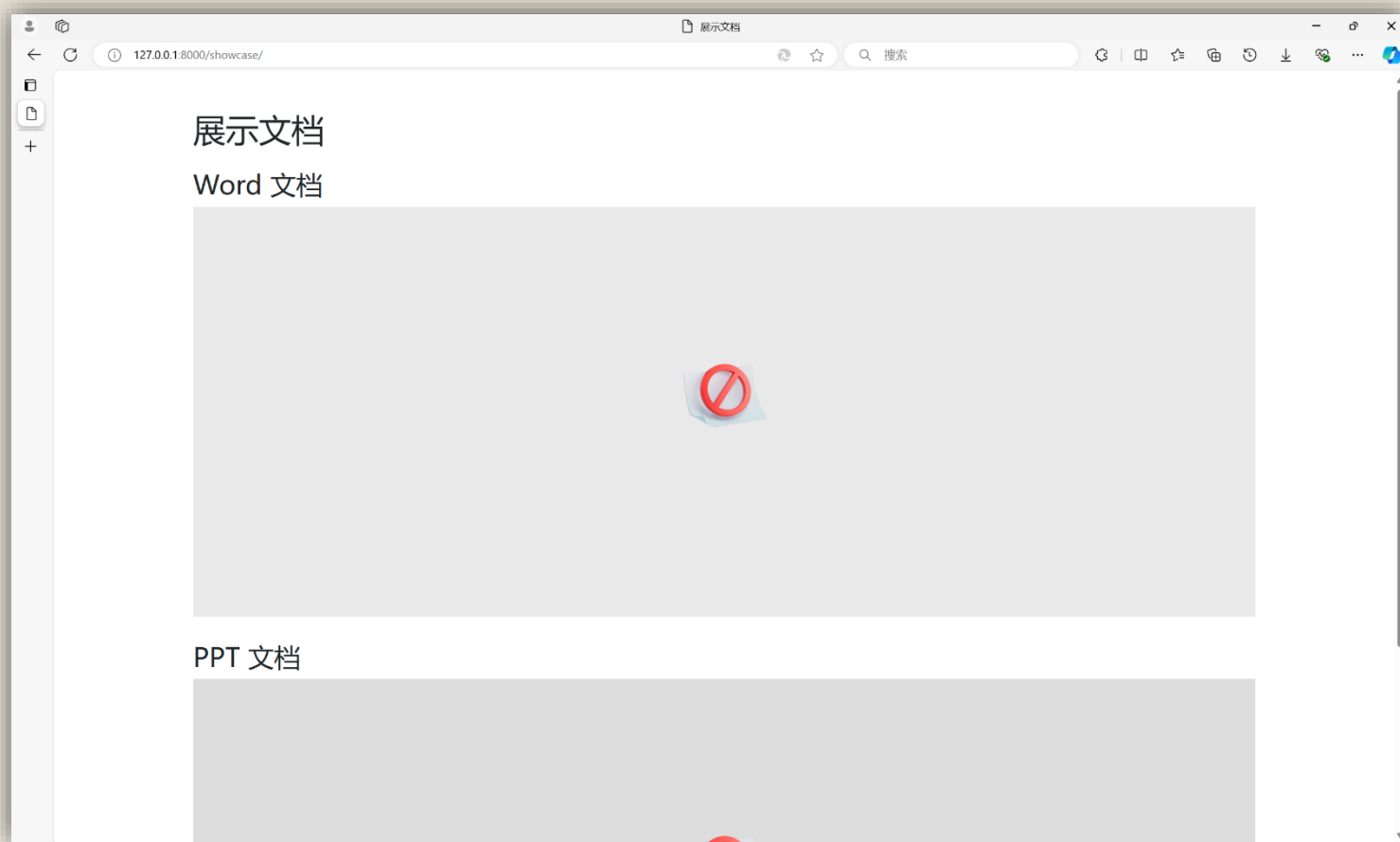


现在的主界面看起来有点单调，所以我们打算让界面有一些可以随机出现的图形元素来丰富一下界面的动态效果，对ChatGPT提出我们的需求之后，使用ChatGPT给出的代码对界面进行了修改，页面会随机从屏幕边缘出现五个白色的三角形，然后当它移动到屏幕边缘时，会再次消失。





在主页完成之后，我们开始设计展示的界面，我们希望在界面上可以查看我们的Word文档与PPT。对ChatGPT提出要求后，我们得到了新界面的代码，然后在后端处理好页面的跳转逻辑之后，运行项目，可以看到新编写的展示页面。





为了让你的 `showcase.html` 页面与主页面的风格统一，我们可以应用相似的样式和布局，确保整体一致性。以下是经过美化后的 `showcase.html` 页面：

html

复制代码

```
{% load static %}
<!DOCTYPE html>
<html lang="zh">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>展示文档</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@500&family=Orbitron:wght@500" rel="stylesheet">
  <style>
    body {
      font-family: 'Roboto', sans-serif; /* 使用Roboto字体 */
      background-color: #f8f9fa; /* 设置背景颜色 */
    }
    .navbar {
      background-color: #ff7f50; /* 珊瑚橙色 */
    }
    .navbar .navbar-brand, .navbar .nav-link {
      color: white !important; /* 字体颜色为白色 */
      font-weight: bold;
    }
    .navbar .nav-link:hover {
      color: #f0f0f0 !important; /* 悬停时的字体颜色 */
    }
    .hero-section {
      background-color: #ffb347; /* 背景颜色 */
    }
```



给“ChatGPT”发送消息

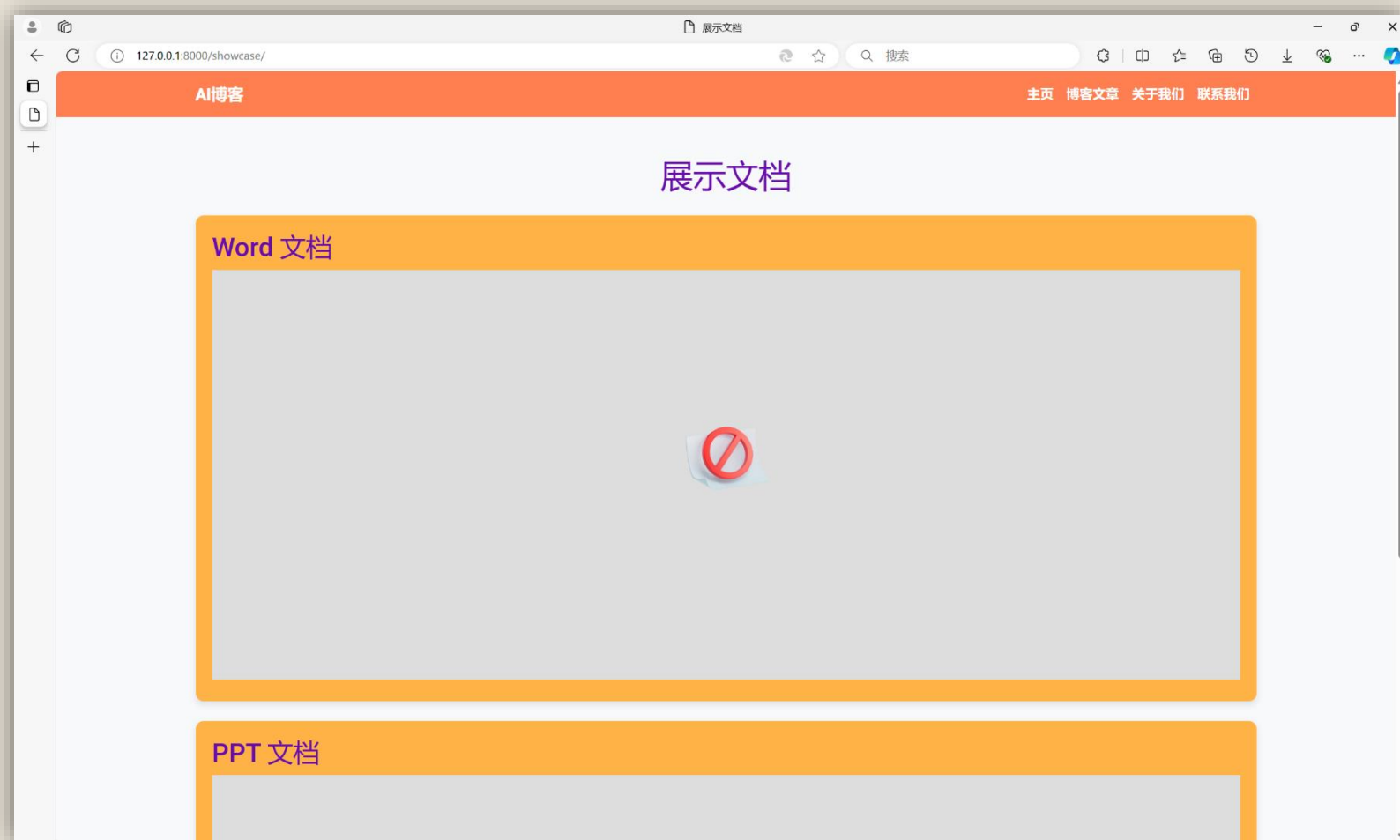


ChatGPT 也可能会犯错。请核查重要信息。

现在页面看起来还是比较单调的，所以我们先对页面进行美化处理，让ChatGPT结合我们所做的主页样式，去美化我们的展示页面。

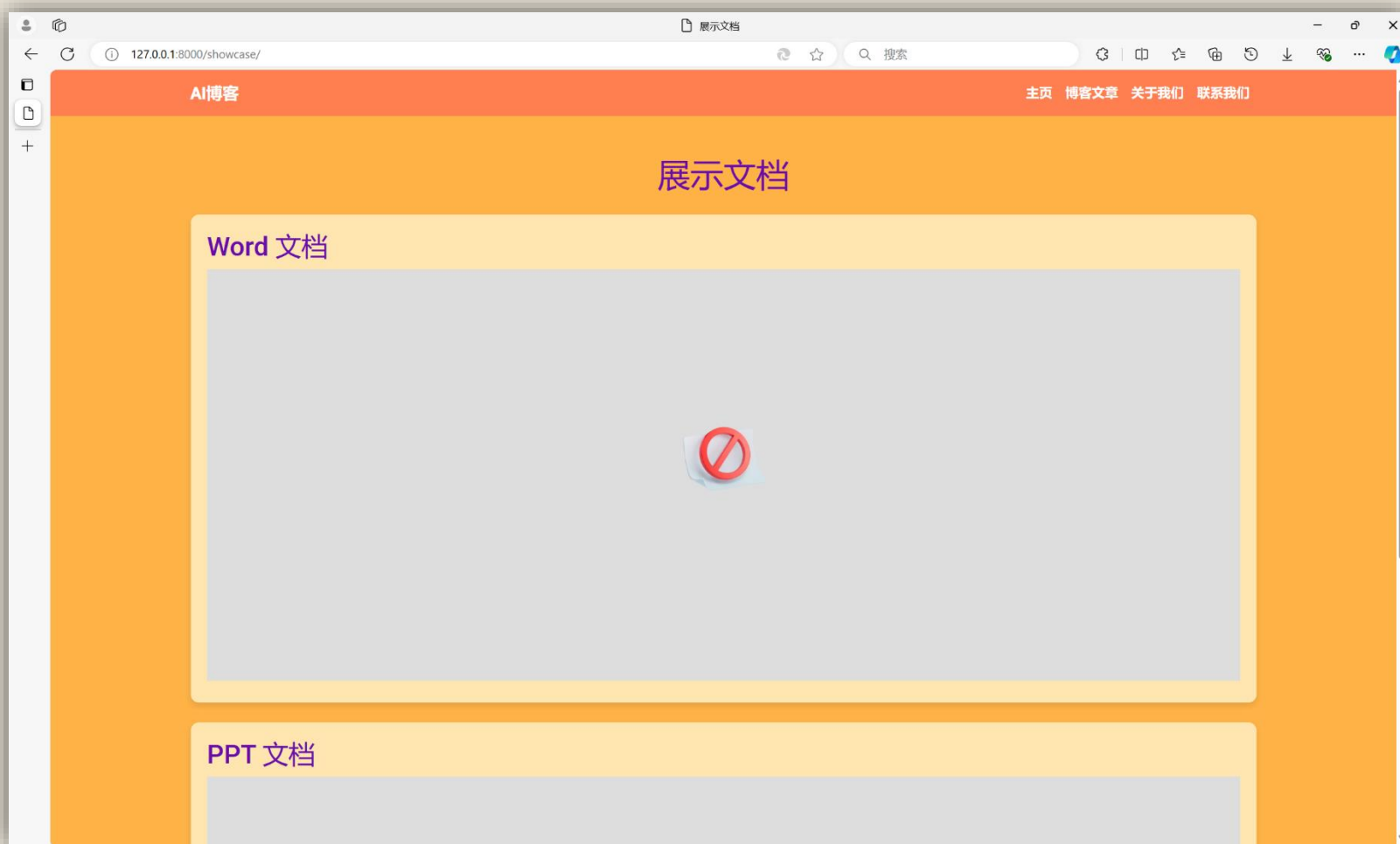


ChatGPT第一次修改之后，得到了新的界面



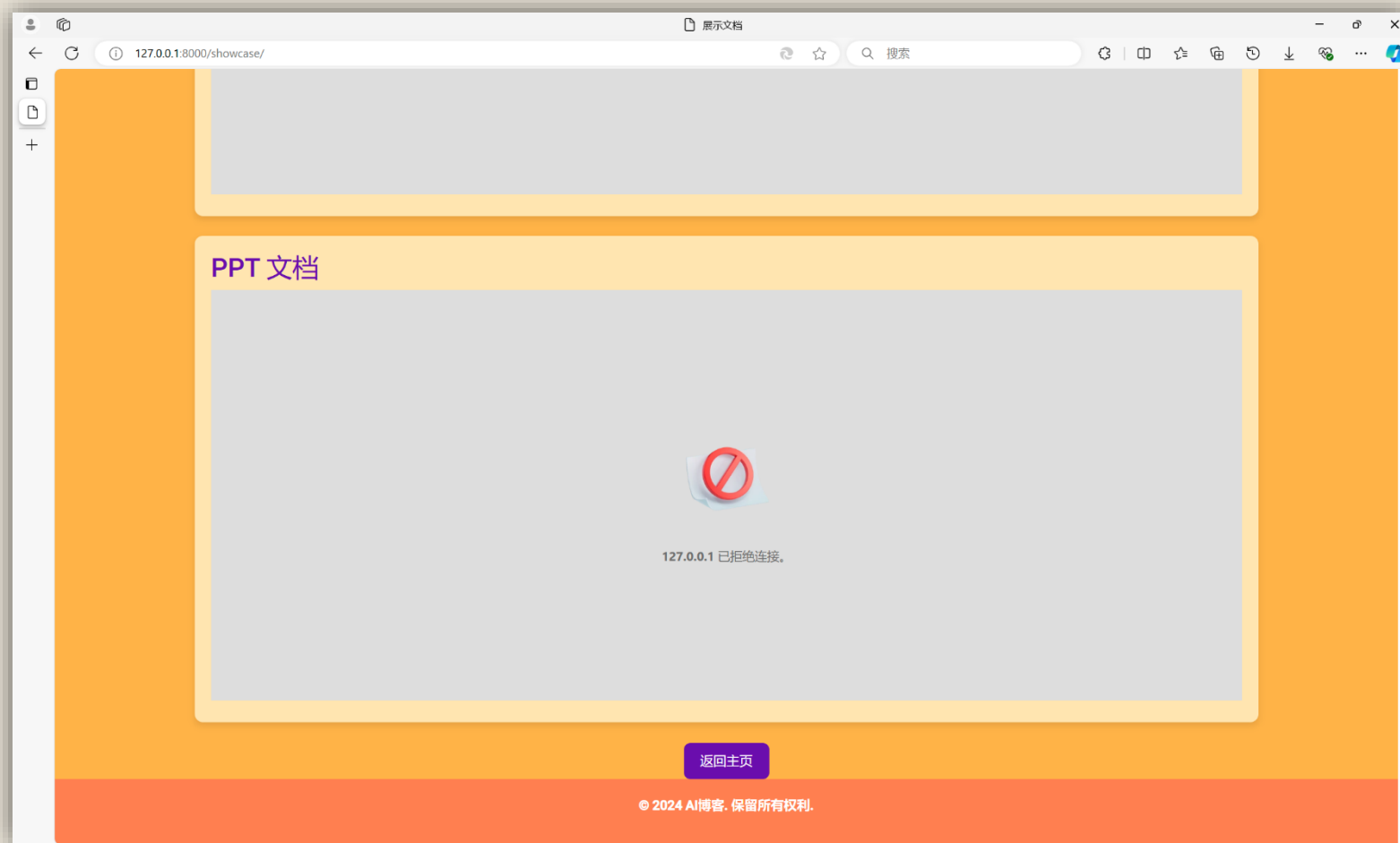


但是界面还是有大量的空白，所以我们继续进行了修改，得到了新的界面



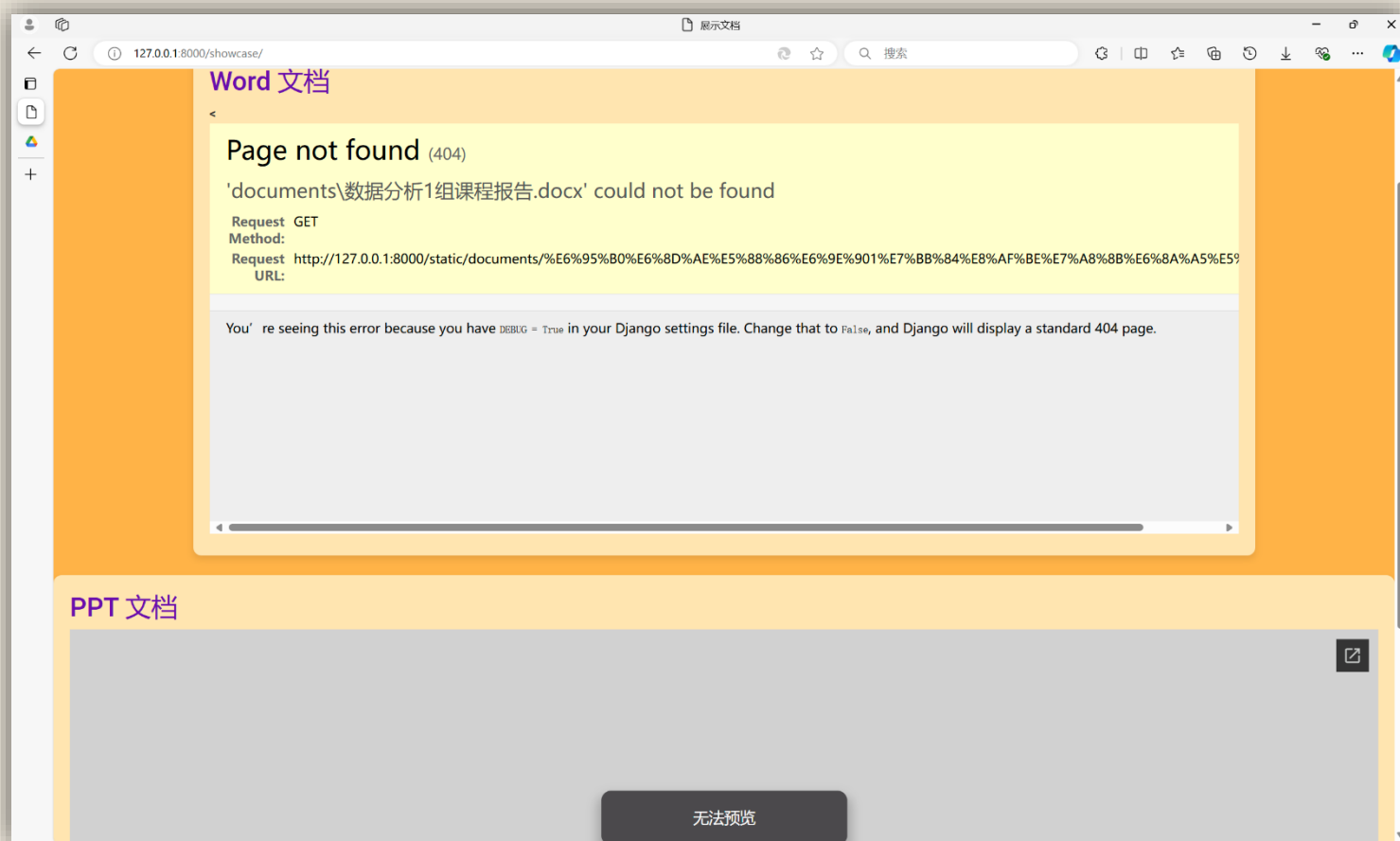


现在页面的主体样式已经修改好了，但是还缺少一个返回主页面的按钮，所以我们继续进行修改，在主内容展示区添加了一个用于返回主界面的按钮



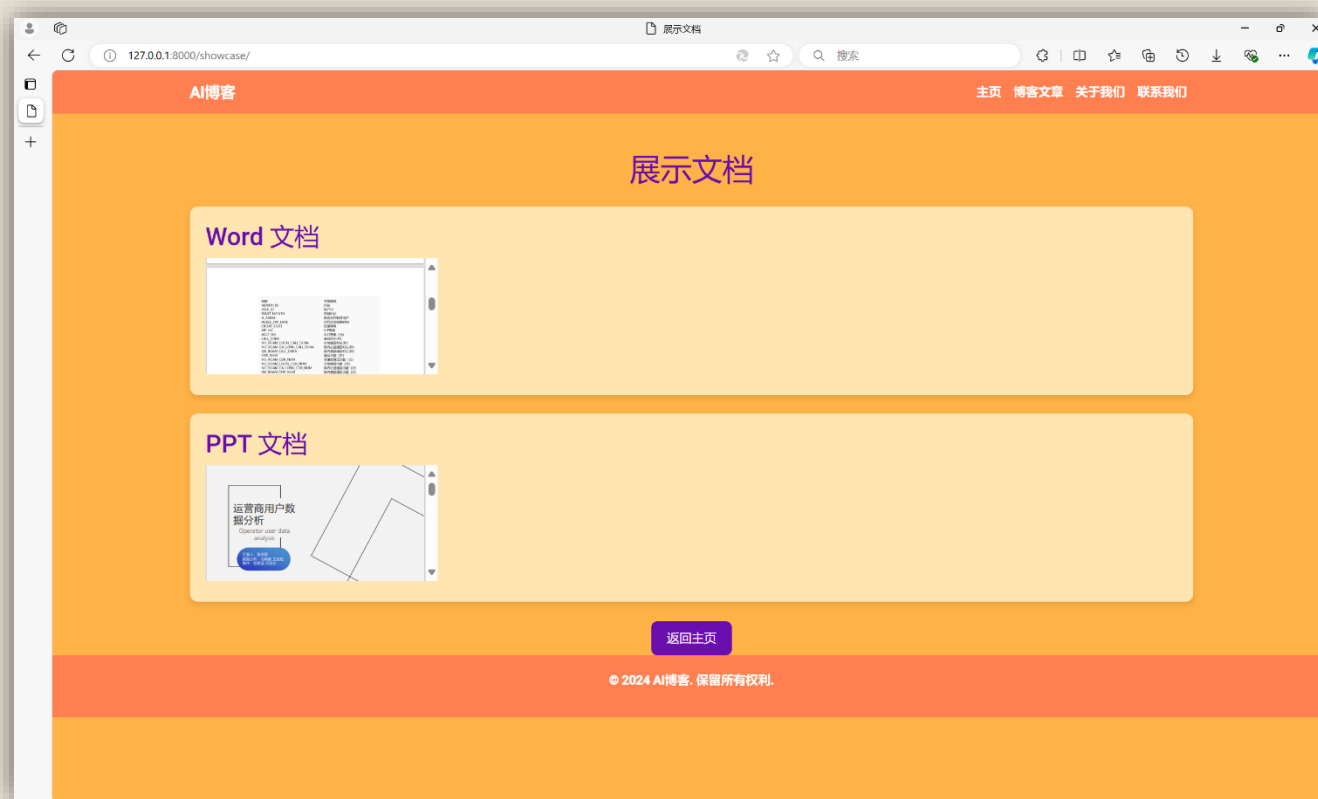


现在展示界面的主体框架也已经做好了，下一步就是把需要展示的Word文档与PPT放到界面上来。所以我们对ChatGPT提出了我们的新的需求，初次修改之后，我们遇到了页面报错与无法查看的问题





在把问题给ChatGPT修改之后，我们发现界面还是没有办法成功预览文件，反而打开界面就会自动下载。我们把遇到的问题再次提供给GPT，发现可能是浏览器不支持的原因，需要一些插件，所以我们采用另一种办法，直接上传PDF文件，因为大多数浏览器都支持PDF的查看。所以我们再次提出需求，利用ChatGPT进行了修改





修改之后的页面可以查看上传的PDF文件，但是样式有的小问题，所以我们继续进行了样式的修改
在调整了样式之后，展示界面已经可以正常查看PDF文档了！个人博客网站搭建完成！



感悟与展望

Reflection and Outlook



AI显著提升了开发效率与解决问题的能力

在项目开发过程中，AI帮助我们在代码编写、错误调试以及功能优化上加速了进程。不仅在细节调整上提供了便捷的解决方案，还让复杂问题的处理变得直观。

例如，在前端设计时，AI帮助我们快速调整CSS样式、完成响应式布局；在后端逻辑开发中，AI更是让我们能够快速解决Django的配置问题，加快了整个开发进度。

AI成为了学习与创新的“虚拟导师”

利用AI的过程中，不仅仅是在项目上取得了成效，还让我们学会了许多新知识和实践技巧。AI的帮助如同一位“导师”，让我们快速掌握了如数据库操作、文档展示、数据可视化等功能实现方法。尤其是当遇到瓶颈时，AI总能给予清晰的思路和思考角度，帮助我们突破技术上的限制，为我们打开了更多创新和实践的可能性。

通过完成个人博客项目，我们深刻体会到AI的强大和便利，同时也反思到自身能力的提升空间。AI在速度和解决方案上具备显著优势，但我们意识到，扎实的基础知识和独立解决问题的能力同样重要。

尽管AI提供了丰富的建议，我们也需要发挥创造性思维，定制符合项目需求的解决方案。总之，将AI视为辅助工具，而非唯一依赖，以更主动的姿态提升自我，强化内在的技术实力。





本次AI制作的个人博客访问网址
<http://nachtohne.s7.tunnelfrp.com>
可能会因为网络原因无法访问



感谢您的观看!

AI模型