**怡心阅读**

**[部署文档]**

|  |  |
| --- | --- |
| 成员 | 何宇楷 |
| 成员 | 王剑飞 |
| 成员 | 李扬 |
| 成员 | 夏厚敬 |
| 成员 | 陈子恒 |

目录

[1. 引言 1](#_Toc105663865)

[1.1. 项目背景 1](#_Toc105663866)

[1.2. 目的 1](#_Toc105663867)

[1.3. 涉及名词 1](#_Toc105663868)

[1.4. 项目使用者 2](#_Toc105663869)

[1.5. 参考资料 2](#_Toc105663870)

[1.6. 相关文档 2](#_Toc105663871)

[2. 系统设置 2](#_Toc105663872)

[2.1. 硬件运行环境 2](#_Toc105663873)

[2.2. 软件运行环境 3](#_Toc105663874)

[3. 环境搭建和项目的运行 3](#_Toc105663875)

[3.1. 前端和后端开发环境搭建 3](#_Toc105663876)

[3.1.1. 下载Visual Studio Code 3](#_Toc105663877)

[3.1.2. 依赖的环境 4](#_Toc105663880)

[3.1.3. Vue的安装 5](#_Toc105663881)

[3.1.4. 项目的创建与运行 5](#_Toc105663882)

[3.2. 后端django环境搭建 6](#_Toc105663883)

[3.2.1. 依赖的环境 6](#_Toc105663884)

[3.2.3. django的安装 7](#_Toc105663885)

[3.2.4. 项目的创建与运行 8](#_Toc105663886)

[3.3. 实现前后端交互 8](#_Toc105663887)

[3.3.1. 前端连接后端 8](#_Toc105663888)

[3.3.2. 本地运行实现前后端交互过程 9](#_Toc105663889)

# 引言

## 项目背景

提高图书馆管理效率：传统的图书馆管理方式可能过于依赖手工记录和处理，导致工作效率低下和错误的发生。通过引入图书馆系统，可以自动化图书馆的各项管理任务，如图书采购、编目、借阅和归还等，从而提高管理效率，减少人力成本。

方便读者的图书借阅：图书馆系统可以提供一个在线公共访问目录（OPAC），使读者能够在家中或办公室通过网络搜索和浏览图书馆的藏书目录，并进行图书预约和借阅。这样，读者可以更方便地获取所需的图书资源，避免不必要的等待和排队。

数字化资源管理：随着数字化时代的到来，图书馆不仅有纸质图书，还拥有大量的电子资源，如电子书籍、学术期刊、数据库等。通过建立数字化资源管理系统，可以有效地管理和提供这些电子资源，使读者可以在线阅读和下载，拓展了图书馆的资源范围和服务内容。

统计和分析功能：图书馆系统可以提供统计和分析功能，帮助图书馆管理员了解图书馆的使用情况、读者的需求和借阅趋势等。这些数据可以用于优化图书采购计划、改进服务质量，以及制定更合理的图书馆政策和规划。

提供多样化的服务：图书馆系统还可以提供其他多样化的服务，如预约座位、图书续借、图书评论和推荐等。这些额外的功能可以增加读者的参与度和满意度，提升图书馆的服务质量和形象。

## 目的

本文档为技术分享类图书馆网站的部署文档，本文档的主要目的旨在说明本项目如何在本地运行。

## 涉及名词

[1] 管理员：在实际操作中，拥有本站管理者账号的用户。管理员可以通过管理员账号登录，可以对图书馆图书进行操作

[2] 游客：没有注册网站用户的使用者，可以浏览阅读书籍，注册登录后可成为用户。

## 项目使用者

本平台的实际使用者为有意愿浏览和使用本网站进行书籍阅读的用户。

## 参考资料

[1] 吕云翔. 软件工程——理论与实践[M]. 人民邮电出版社, 2020.

[2] Pressman R S. 软件工程: 实践者的研究方法 (英文精编版)[M]. 机械工业出版社, 2019.

[3] Ian Sommerille. 软件工程(原书第10版)[M]. 机械工业出版社, 2018.

## 相关文档

[1] 《需求规格说明书》

[2] 《软件设计说明书》

[3] 《部署文档》

[4] 《测试报告》

[5] 《用户使用说明书》

# 系统设置

## 硬件运行环境

处理器：Intel® Core™ i7-11800H CPU @2.30GHz

内存：16GB

硬盘容量：1TB

输入输出设备：鼠标、键盘、显示屏

网络设备：宽带

## 软件运行环境

操作系统：Windows 10

连接的后端数据库：MySQL 8.0.28

前端框架：Vue3

后端框架：django

项目运行环境：Python 3.8

浏览器：Firefox

# 环境搭建和项目的运行

本文档主要讲述如何搭建前后端环境以及如何运行项目。

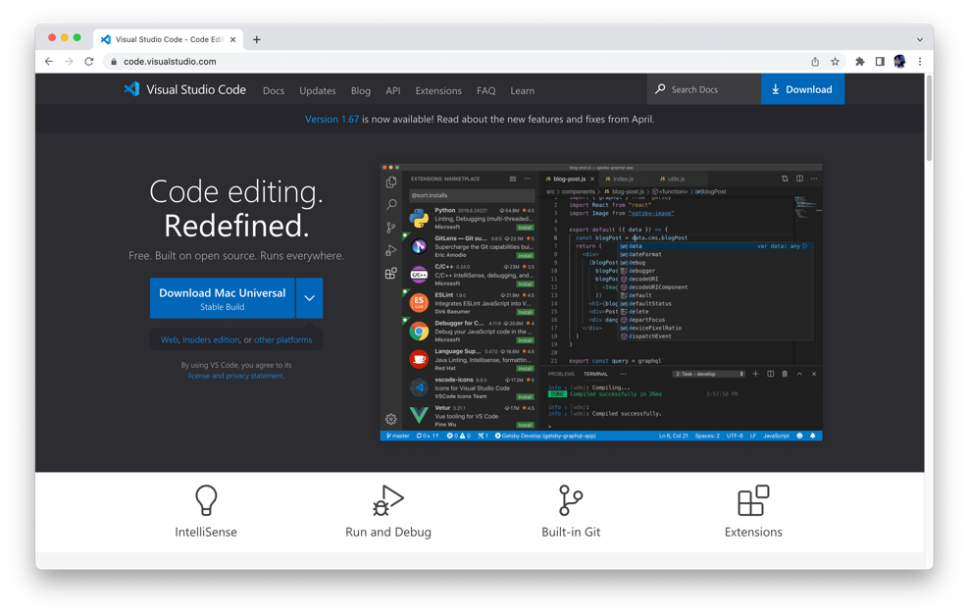
前端使用的是@vue-cli自带的静态服务器，然后在网络相同的环境下，向后端的计算机发送接口的请求，可以使得前后端进行交互。

## 前端和后端开发环境搭建

本项目可以使用Visual Studio Code

### 下载Visual Studio Code

进入Visual Studio Code官网：<https://code.visualstudio.com/>，根据自己的系统进行相应的下载。官网页面如图1所示。

****

**图1 Visual Studio Code官网**

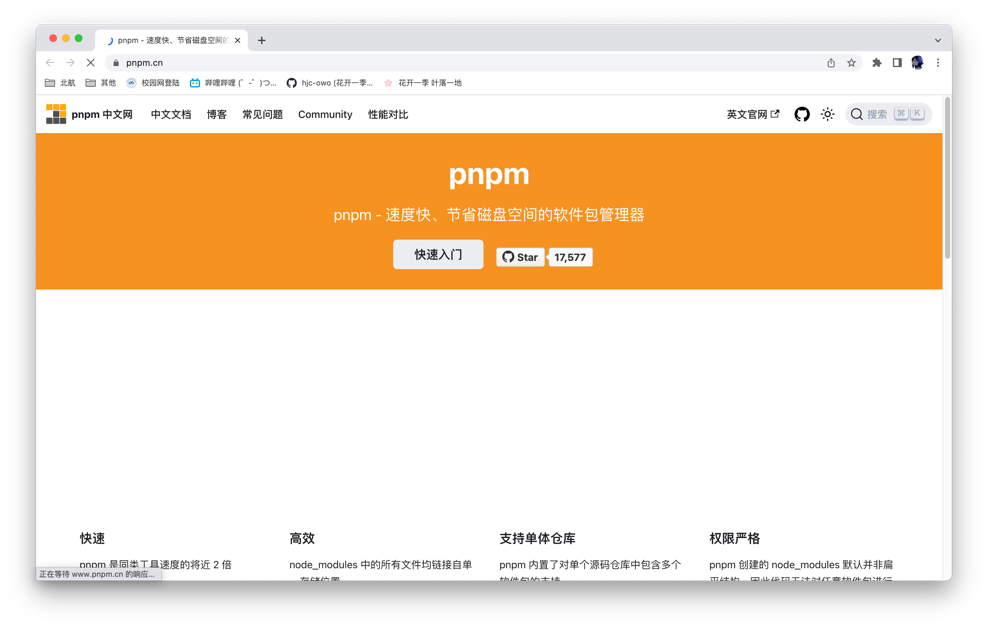
Visual Studio Code有许多非常好用的插件，可根据自己的需求进行插件的选择。

### 依赖的环境

本项目使用的前端框架为Vue3，脚手架版本为@vue/cli 4.5.15，node版本为v18.2.0。项目采用pnpm对包进行管理，版本为7.1.7，pnpm会对包代码进行复用，高效利用磁盘空间的同时，还提高了包的安装速度，通过依赖管理保证了安全性。

我们需要先对Node.js进行安装，Node.js是一个基于 Chrome V8 引擎的 JavaScript 运行环境，官网为：<https://nodejs.org/en/>。

安装成功后我们通过Node.js所带的包管理工具npm安装pnpm，安装指令为npm install -g pnpm。pnpm中文官网首页如图4所示。

****

**图4 pnpm官网首页**

### Vue的安装

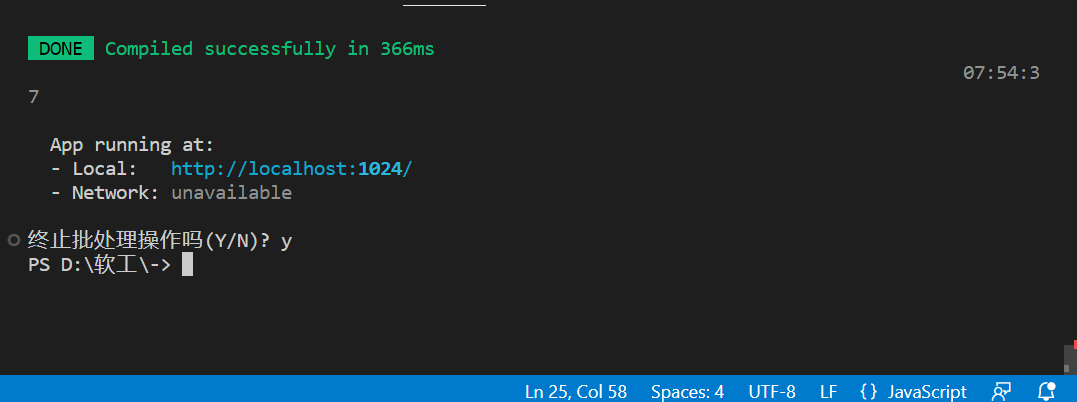
在npm安装成功后，我们使用npm全局安装vue/cli，在终端中输入npm install -g @vue/cli.

### 项目的创建与运行

之后在终端输入vue create [项目名]就可以进行项目创建。

选择完成后就可以开始进行项目代码的正式编写，如果在编写的过程中，想要安装其他的依赖，可以在终端使用：npm install xxx。该命令可以进行相应依赖的安装。

在代码编写完成之后，在终端输入指令:npm run serve。该命令可以运行前端项目的代码。其运行的截图如图7所示。



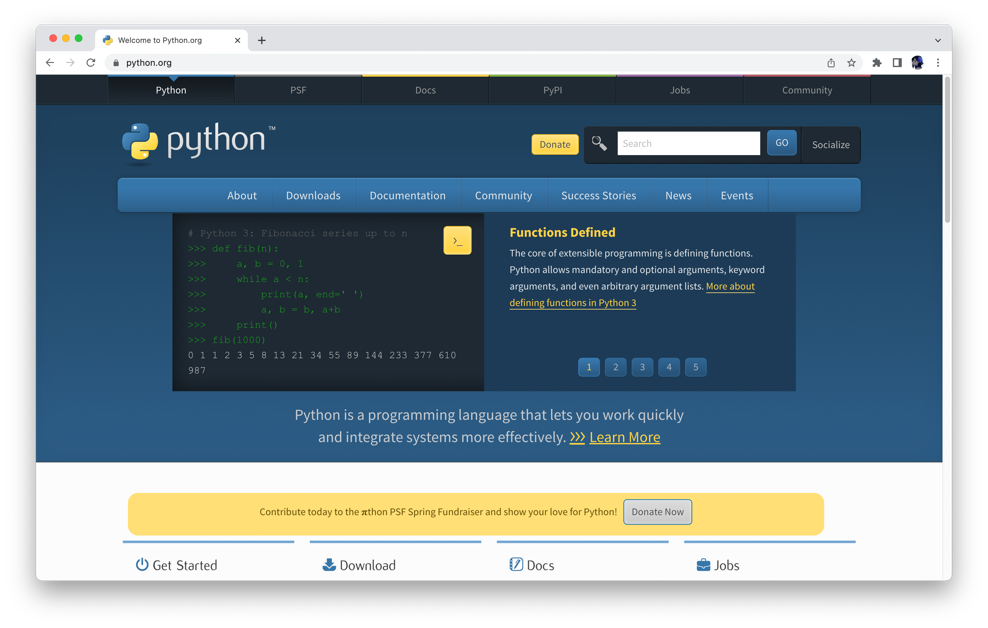
**图7 前端项目运行截图**

我们可以使用npm run build将完成的项目打包。

## 后端django环境搭建

### 依赖的环境

django框架是基于python环境安装的后端框架，所以我们需要先进行python环境的安装，直接到python的官网安装即可。Python官网的地址为：<https://www.python.org/>，其页面如图8所示。

****

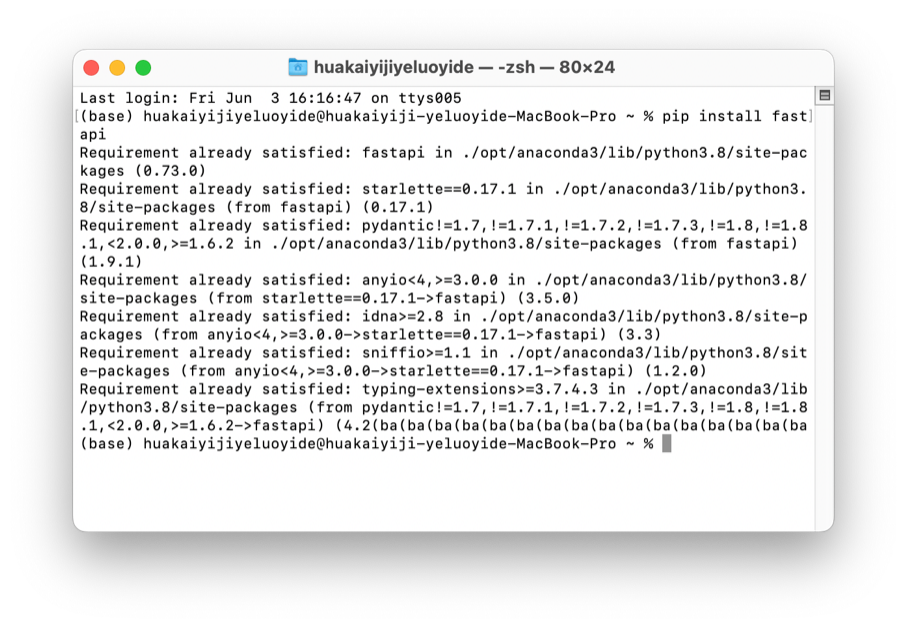
**图8 开发者工具界面**

还可以使用使用anaconda集成环境，anaconda可以一键创建虚拟环境，方便运行不同版本的python代码，使其互相不受干扰。

Anaconda的下载与安装网上的资料非常丰富，在此不再赘述其下载与安装，只阐述其使用方法。Anaconda创建环境的指令是：conda create -n web python==3.9。其中web是虚拟集成环境的名称，该名称可以自己定义；python==3.9表示这次创建的虚拟环境的python版本为3.9，anaconda虚拟环境可以自动配置，使用非常方便。Anaconda进入虚拟环境的指令是：conda activate web，web是你想要激活的虚拟环境的名称。我们开发人员也是使用该集成环境开发后端代码的，下面部署的操作步骤都是在conda虚拟环境当中实现的。

### django 的安装

在安装完python并将它添加到系统变量之后，我们需要安装django i框架。在终端输入：pip install django，即可进行django安装。其安装过程如图9所示。

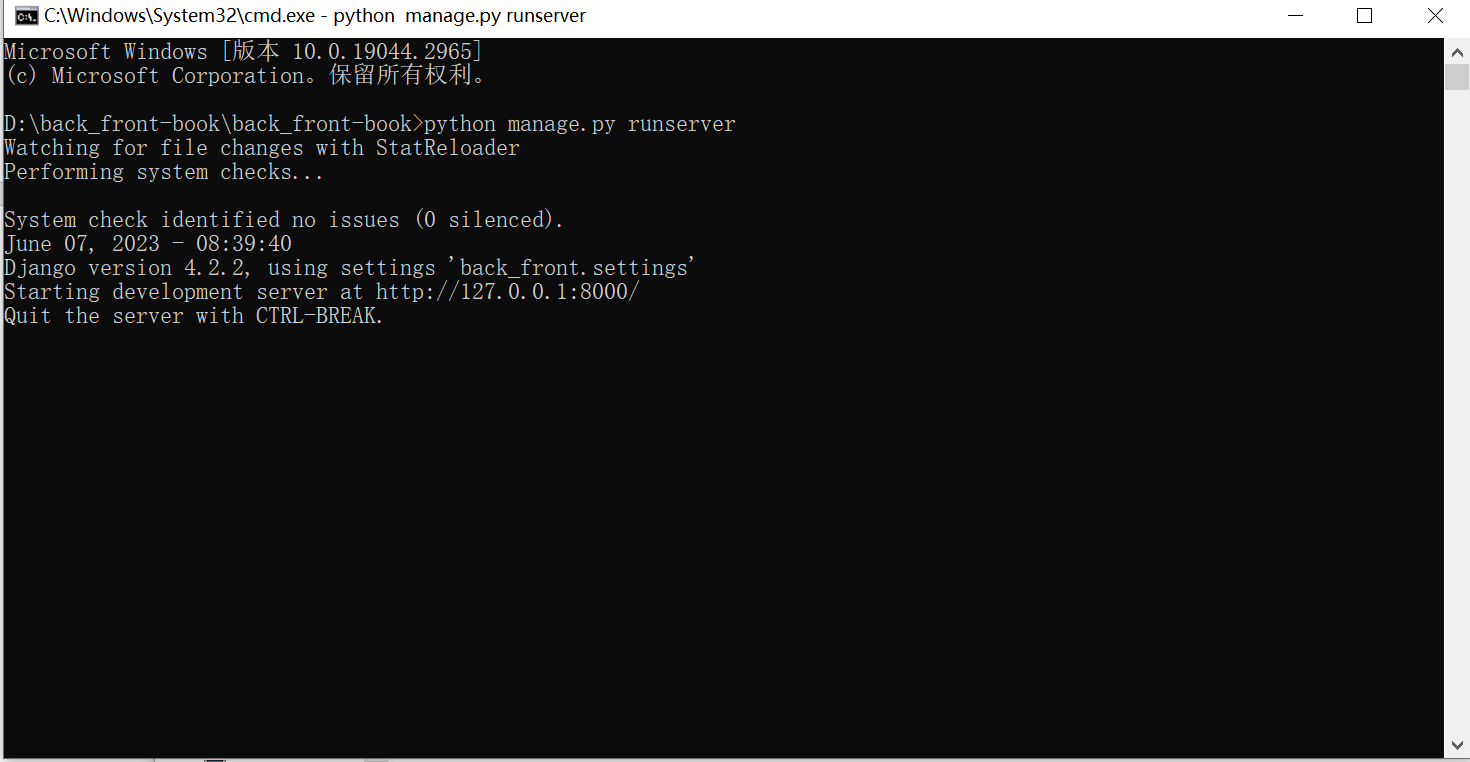
****

**图9 Fastapi的安装过程**

### 项目的创建与运行

在代码编写的过程中，想要安装其他的第三方库，可以在终端使用：pip install xxx，该命令可以进行相应依赖的安装。

在代码编写完成之后，在终端输入指令：python manage.py makemigrations python manage.py migrate python manage.py runserver。该命令可以运行后端项目的代码。其运行的截图如图10所示。

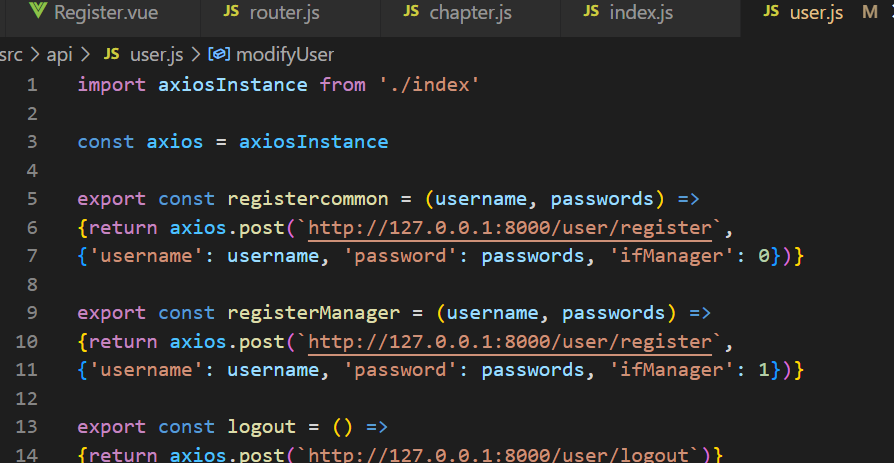


**图10 后端项目运行截图**

## 实现前后端交互

### 前端连接后端

在我们的前端项目源文件/src/api当中，设置了后端的接口地址，相关代码如图11所示。



**图11 前端设置后端地址**

我们的项目在服务器上同样有相同的配置。在真正投入使用时，我们使用服务器的地址；在本地进行前后端测试时，运行后端项目，将服务器地址注释掉，换成本地后端的地址即可。测试后即可将后端代码部署到服务器上。当然如果有其他的后端项目完成了相同的内容，并且所有的接口都相同时，也可以实现前后端项目的交互，体现了我们的项目的可维护性。

### 本地运行实现前后端交互过程

在网站使用过程中，前端会向后端请求数据，如果在本地网络上进行网站的运行，所以只需要运行前端代码的个人电脑和运行后端的电脑连上相同的无线网络，后端给前端提供接口，然后前端向接口传递参数，后端返回数据给前端，前端进行显示，这样就完成了前后端的交互。前后端进行交互的代码示例如图12所示。



**图12 前后端交互代码示例**