CHAPTER 3



项目3　法律助手

本项目基于 HTML 结构内容，使用 CSS 进行样式设计，引用 JavaScript 文件，选取 DeepSeek-R1 大模型，并调用开放的 API 实现法律咨询系统。



项目 3 资源

3.1　总体设计

本节包括整体框架和系统流程。

3.1.1　整体框架

整体框架如图 3-1 所示。

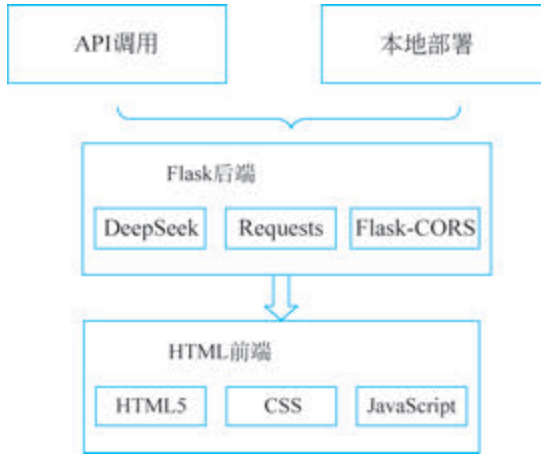


图 3-1　整体框架

3.1.2　系统流程

系统流程如图 3-2 所示。

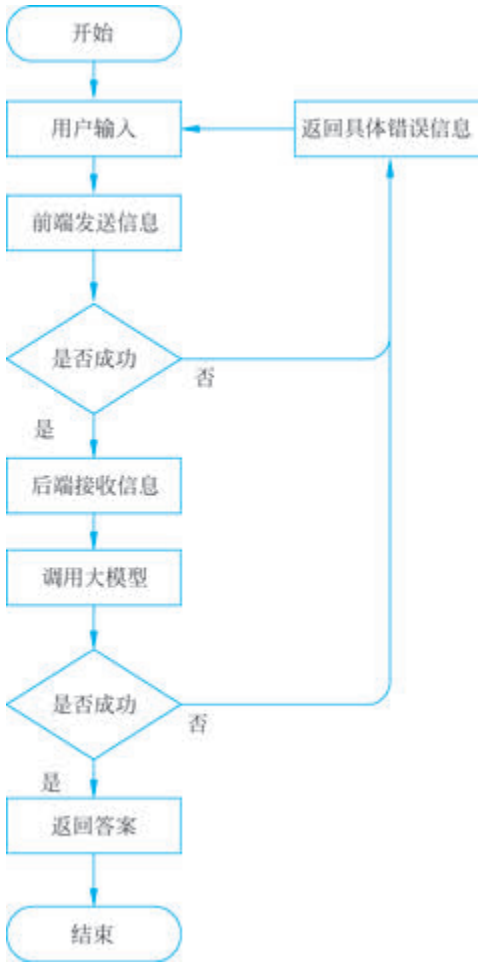


图 3-2　系统流程

3.2　开发环境

本节介绍 Python、PyCharm 和 Flask 的安装步骤，给出创建项目与前端界面的搭建方法，描述大模型 API 的申请流程及 Ollama 本地部署的具体操作。

3.2.1　安装 Python

安装 Python 参见 1.2.1 节。

3.2.2　安装 PyCharm

安装 PyCharm 参见 1.2.2 节。

3.2.3　安装 Flask

安装 Flask 参见 1.2.3 节。

3.2.4 创建项目

创建项目参见 1.2.4 节。

3.2.5 创建前端界面

创建前端界面参见 1.2.5 节。

3.2.6 大模型 API 申请

本项目采用阿里云百炼作为 DeepSeek API 调用的平台，参见 1.2.6 节。

3.2.7 实现 Ollama 本地部署

实现 Ollama 本地部署参见 1.2.7 节。

3.3 系统实现

本系统完成前端界面优化、动态交互功能及后端模型通信的代码实现，确保系统能稳定处理用户请求并返回大模型生成的答案。

3.3.1 样式< style >

定义网页 CSS 样式的相关代码见“代码文件 3-1”。

3.3.2 主体< body >

实现如何与大模型进行通信的相关代码见“代码文件 3-2”。

3.3.3 交互< script >

JavaScript 主要完成前后端交互的任务，如前端需要向后端发送一个请求，后端在处理这个请求后将答案返回给前端。相关代码见“代码文件 3-3”。

3.3.4 Flask 后端

后端主要实现路由的编写、前端发送问题的传递及大模型返回问题答案的输出功能。相关代码见“代码文件 3-4”。

3.4 功能测试

本节包括启动项目、发送问题及响应。

3.4.1 启动项目

运行 Flask 后端程序，输出调试信息及相应的运行端口，如图 3-3 所示。

```
 * Serving Flask app 'lab-code/deepseek-local/app.py'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

图 3-3　后端运行结果

打开对应的端口，可以看到前端界面，如图 3-4 所示。



图 3-4　前端界面

3.4.2　发送问题及响应

示例中向 DeepSeek 智能问答系统提问，如果后端控制台界面出现“HTTP 200”状态码，则代表用户在前端输入的问题已经成功发送到后端，后端再向阿里云百炼平台发送申请，调用 API key，将答案返回给前端界面，如图 3-5 所示。

```
Press CTRL+C to quit
127.0.0.1 - - [07/Jun/2025 20:09:06] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [07/Jun/2025 20:12:16] "POST /api/ask HTTP/1.1" 200 -
127.0.0.1 - - [07/Jun/2025 20:12:54] "POST /api/clause-search HTTP/1.1" 200 -
127.0.0.1 - - [07/Jun/2025 20:15:47] "POST /api/clause-search HTTP/1.1" 200 -
127.0.0.1 - - [07/Jun/2025 20:21:26] "POST /api/clause-search HTTP/1.1" 200 -
```

图 3-5　后端输出信息

此时前端已经返回答案并输出内容，如图 3-6 所示。



(a)　(b)

图 3-6　前端返回答案

CHAPTER 4



项目4　面　试　官

本项目基于微信小程序框架搭建前端界面，通过 WXML（WeiXin Markup Language，微信标记语言）构建页面结构、WXSS（WeiXin Style Sheets，微信样式表）实现界面样式设计，整合 JavaScript 脚本完成前端业务逻辑开发。后端基于 Node.js 与 Express 框架构建 RESTful API 接口，处理前端请求及核心业务逻辑。数据存储层面，依托 MySQL 数据库结合 Sequelize ORM 框架实现结构化数据管理，同时借助微信小程序云数据库存储用户注册等轻量信息，并集成 DeepSeek API 智能生成面试题且同步写入数据库。系统通过开放 API 调用与本地后端服务的协同配合实现数据高效处理，核心功能涵盖用户登录、面试题库管理、AI 智能出题及数据筛选展示等，有效提升面试准备与管理工作的智能化、便捷性水平。

项目4资源

4.1　总体设计

本节包括整体框架和系统流程。

4.1.1　整体框架

整体框架如图 4-1 所示。

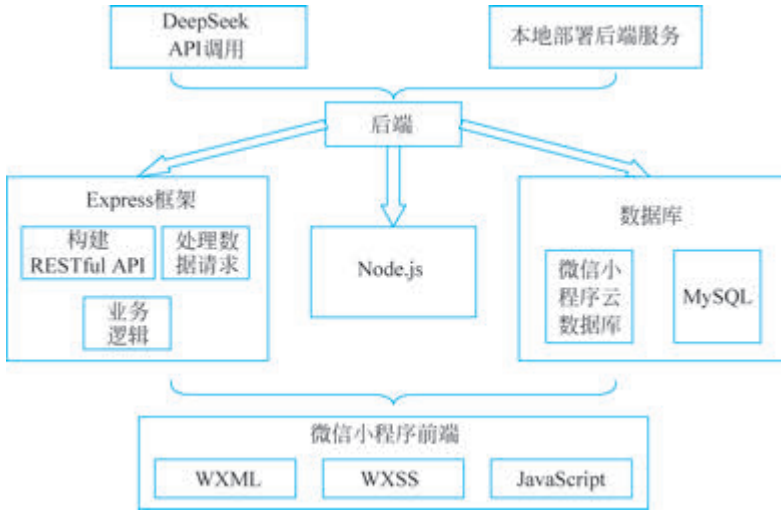


图 4-1　整体框架

4.1.2 系统流程

系统流程如图 4-2 所示。

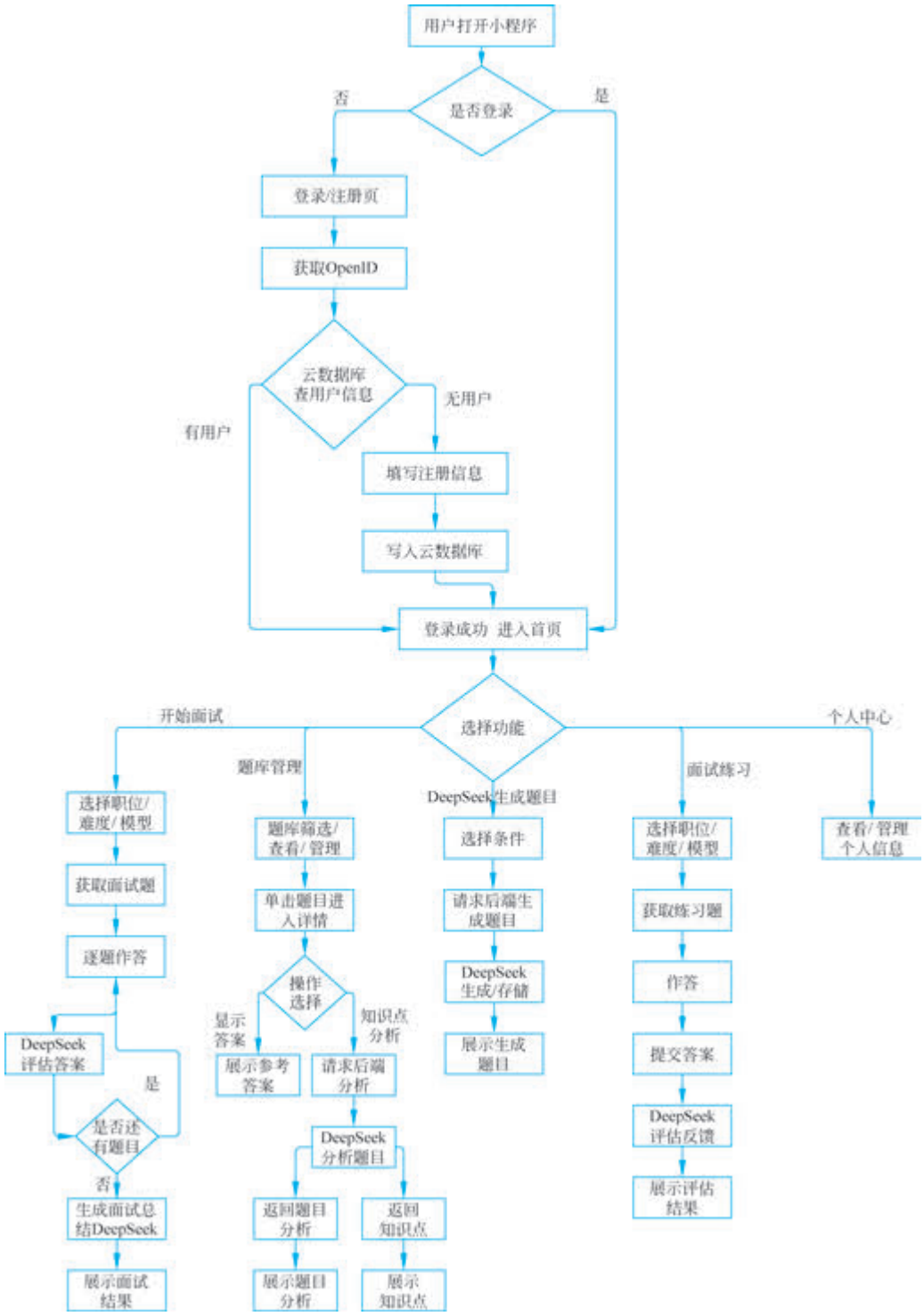


图 4-2 系统流程

4.2 开发环境

本节给出注册微信小程序账号、微信开发者工具、创建项目、MySQL 数据库及大模型 API 的安装过程。

4.2.1 注册微信小程序账号

访问微信公众平台，如图 4-3 所示；填写相关信息如图 4-4 所示；注册成功如图 4-5 所示；小程序界面如图 4-6 所示。



图 4-3 微信公众平台



图 4-4 注册账号



图 4-5 注册成功



图 4-6 小程序界面

4.2.2 安装微信开发者工具

访问微信官网，选择“稳定版”，如图 4-7 所示；下载进度如图 4-8 所示；单击“下一步”按钮，微信开发者工具安装界面如图 4-9 所示；单击“我接受”按钮，微信开发者工具安装许可证协议界面如图 4-10 所示；单击“浏览”选择安装位置(不推荐系统盘)，如图 4-11 所示；微信开发者工具安装过程如图 4-12 所示；安装完成界面如图 4-13 所示。



图 4-7 下载微信开发者工具

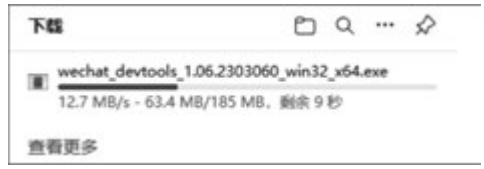


图 4-8 下载进度



图 4-9 微信开发者工具安装界面

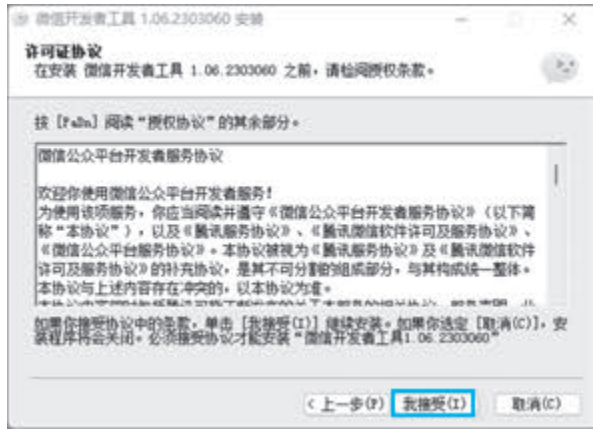


图 4-10 安装许可证协议



图 4-11 微信开发者工具选定安装位置



图 4-12 微信开发者工具安装过程

4.2.3 创建项目

打开微信开发者工具，依次单击“小程序→＋”，如图 4-14 所示。



图 4-13 安装完成



图 4-14 创建项目

在官网中登录小程序，依次单击“开发→开发管理→开发设置→AppID”，复制小程序 ID，如图 4-15 所示。



图 4-15 获取微信小程序 ID

返回微信开发者工具，填写项目名称、目录、AppID，如图 4-16 所示；微信小程序界面如图 4-17 所示；微信小程序样式及配置如图 4-18 所示。



图 4-16　创建微信小程序



图 4-17　微信小程序界面



图 4-18　微信小程序样式及配置

4.2.4　安装 MySQL

安装 MySQL 参见 2.2.6 节。

4.2.5　大模型 API 申请

本项目采用阿里云百炼作为 DeepSeek API 调用的平台，参见 1.2.6 节。