

DeepSeek 医疗正在从效率提升、模式创新、资源优化、科研突破等多维度深度重构医疗产业，推动其向精准化、普惠化、智能化方向加速转型。在医疗领域，DeepSeek 医疗可以实现 AI 问诊、AI 开药、AI 病历管理和 AI 科研论文检索等功能。值得一提的是 AI 问诊功能，此功能不但已经初步进入各大三甲医院，集成在 HIS 智慧医疗系统中，作为医师的辅助功能，而且还走入寻常百姓家，患者可以通过 AI 问诊功能对病情做出初步判断，从而节省医疗费用，降低医疗负担。

2.1 AI 问诊

AI 可以全天候为患者问诊，而且患者不需要排队，对患者而言更方便，还可以减少患者往返医院的交通、挂号等费用，缓解线下医疗资源紧张问题。AI 问诊适合轻症咨询、慢性病管理或初步症状分析，轻症患者在 AI 问诊的帮助下即可治疗感冒、发烧等疾病，降低医疗成本。

如今，AI 问诊技术已经初步进入某些医院，作为医师的 AI 助手。在医院中，AI 问诊可以连接 HIS 智慧医疗系统，采用数据驱动诊断的宗旨，基于 HIS 智慧医疗系统的医学数据库，提供标准化建议。AI 问诊还可以连接 PubMed 等医学论文数据库，结合全球的医学文献为医师提供前沿建议。AI 问诊可以结合患者历史健康数据进行个性化分析，可以使用智能设备监测的血氧、心电等实时数据，还可以使用体格检查等历史数据。

AI 问诊还适合咨询敏感疾病，如心理疾病或性病，不但避免患者尴尬，而且可以保护患者的隐私。患者如果感觉自己有敏感疾病的症状，则不妨先尝试 AI 问诊，如果自身的症状符合敏感疾病的症状，则可以先使用 AI 问诊，再寻求医师的系统治疗，这是一种稳妥的办法。

2.1.1 通用 AI 问诊

通用 AI 问诊的提示词完全由医师或患者决定。医师或患者必须先在提示词中体现主诉内容，如发热和咳嗽等，然后由 DeepSeek 生成诊断结果。

主诉的规范表述应包含以下几个关键组成部分。

1. 症状或体征

症状或体征的核心内容是明确患者最困扰的健康问题，例如头痛、发热和胸痛等词汇。此外，症状或体征要求使用医学术语，而且需要避免诊断性表述。

2. 时间特征

时间特征需要包含持续时间，即症状出现到就诊的时间，例如持续 2h。此外，时间特征还需要包含起病形式，例如急性、亚急性或慢性。

3. 症状性质

在描述疼痛时，可以包括性质，例如钝痛、绞痛和刺痛；可以包括程度，例如轻度和重度；可以包括放射部位，例如向右肩放射。

在描述其他症状时，可以包括伴随特征，例如咳嗽伴黄痰。

4. 加重或缓解因素

可以注明影响症状的因素，例如活动后胸痛加重或者进食后腹痛缓解。

首先医师或患者在输入框中输入主诉，然后单击“确定”按钮，即可生成诊断结果。通用 AI 问诊页面如图 2-1 所示。

通用AI问诊

通用AI问诊

我
发热，咳嗽，咳痰，咽痛，鼻塞，流涕，乏力，头痛，咽干

DeepSeek AI

诊断结果：
是否有病症：是

症状：发热、咳嗽、咳痰、咽痛、鼻塞、流涕、乏力、头痛、咽干

可能的疾病名称：上呼吸道感染（可能是普通感冒或流感）

治疗方案：
对症治疗：

退热：可以使用退热药物如对乙酰氨基酚（扑热息痛）或布洛芬。

止咳：如果咳嗽严重，可以使用止咳药物如右美沙芬。

化痰：如果痰液粘稠，可以使用化痰药物如氨溴索。

缓解咽痛：可以使用含片或喷雾剂如含有苯佐卡因的喉片。

缓解鼻塞：可以使用鼻用减充血剂如伪麻黄碱。

休息与饮食：

输入主诉，生成诊断

确定 ✓

图 2-1 通用 AI 问诊页面

实现通用 AI 问诊页面的代码如下：

```
<!-- #第2章/通用AI问诊.html -->
<div id="q-app">
  <q-layout view="hHh LpR fFf">
    <q-header elevated class="bg-primary text-white">
      <q-toolbar class="bg-primary text-white q-ny-md shadow-2">
```

```

        <q-btn stretch flat label="通用 AI 问诊" @click="goTo
('http://cnoctave.top')"></q-btn>
        <q-separator dark vertical></q-separator>
        </q-toolbar>
    </q-header>
    <q-page-container>
        <div class="q-ma-md">
            <div class="text-h6">通用 AI 问诊</div>
            <q-chat-message v-show="result_list[0] !== ''" name="我" :
text=ask_list sent></q-chat-message>
            <q-chat-message v-show="result_list[0] !== ''" style=
"white-space: pre-wrap;" name="DeepSeek AI"
: text=result_list></q-chat-message>
            <q-chip v-show="result_list[0] !== ''" color="orange-12"
text-color="white" icon="warning">
                本回答由 AI 生成，内容仅供参考，请仔细甄别。
            </q-chip>
        </div>
    </q-page-container>
</q-layout>
<div class="q-ma-md bg-white" style="position: sticky; bottom: 0;
z-index: 1000;">
    <q-input class="bg-white" style="position: sticky; bottom: 5vh;
z-index: 1001;" placeholder="输入主诉，生成诊断"
outlined v-model="md.op_name" autogrow>
        <template v-slot:append>
            <q-btn color="green" icon-right="check" label="确定" no-caps
@click="ai()"></q-btn>
        </template>
    </q-input>
</div>
</div>

```

编写通用 AI 问诊的系统提示词，代码如下：

```

#第2章/通用 AI 问诊 Prompt.py

system_prompt = """你是一名 AI 医师，
回答问题时使用中文，
通过用户提供的主诉内容生成诊断结果，
生成的内容必须包括是否有病症，
如果有病症，则生成的内容必须包括症状、疾病名称、治疗方案等"""

```

调用通用 AI 问诊 API 的代码如下：

```

#第2章/通用 AI 问诊.py
import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \

```

```

    Depends, HTTPException, status, Annotated
from openai import OpenAI
from 通用AI 问诊 Prompt import system_prompt
import datetime
from starlette.responses import JSONResponse

api_key = '替换为 DeepSeek 的 API Key'

app = FastAPI()

@app.post("/ai/wenzhen")
async def 通用AI 问诊(prompt: Annotated[str, Body(description="通用AI 问诊")]):
    try:
        if not prompt:
            return Result(retcode=1, text="必须填写提示词")
        client = OpenAI(api_key=api_key, base_url="https://api.deepseek.
com")

        response = client.chat.completions.create(
            model="deepseek-chat",
            messages=[
                {"role": "system", "content": system_prompt},
                {"role": "user", "content": prompt},
            ],
            stream=False
        )

        print(response.choices[0].message.content)
        return Result(retcode=0, text="成功",
            result={
                "result_list": [response.choices[0].message.
content],
            })
    except APIException as e:
        return JSONResponse(
            status_code=e.status,
            content={"retcode": e.retcode, "text": e.text, "result":
e.result},
        )

if __name__ == "__main__":
    uvicorn.run(app, host="127.0.0.1", port=8000)

```

2.1.2 专科 AI 问诊

相同的主诉在不同的科室中可能会得出不同的诊断结果。专科 AI 问诊为了解决这种问题，首先在 AI 问诊的基础上允许医师选择科室，然后由 DeepSeek 生成诊断结果，确保

DeepSeek 的诊断结果能够贴近所选科室的诊断结果。

首先医师在输入框中输入主诉，并在“科室”下拉菜单中选择合适的科室，然后单击“确定”按钮，即可生成诊断结果。专科 AI 问诊页面如图 2-2 所示。



图 2-2 专科 AI 问诊页面

实现专科 AI 问诊页面的代码如下：

```

<!-- #第 2 章/专科 AI 问诊.html -->
<div id="q-app">
  <q-layout view="hHh LpR fFf">
    <q-header elevated class="bg-primary text-white">
      <q-toolbar class="bg-primary text-white q-ny-md shadow-2">
        <q-btn stretch flat label="专科 AI 问诊" @click="goTo
('http://cnoctave.top')"></q-btn>
      <q-separator dark vertical></q-separator>
    </q-toolbar>
  </q-header>
  <q-page-container>
    <div class="q-ma-md">
      <div class="text-h6">专科 AI 问诊</div>
      <div style="display: flex; justify-content: flex-start;
align-items: center;">
        <div class="text-h6">科室: </div>
        <q-select outlined:model-value="department" use-input
hide-selected fill-input
input-debounce="0" :options="department_list">
          <template v-slot:no-option>
            <q-item>
              <q-item-section class="text-grey">
                无数据匹配，请输入正确的科室
              </q-item-section>
            </q-item>
          </template>
        </q-select>
      </div>
    </div>
  </q-page-container>
</q-layout>
</div>
```

```

        </q-item-section>
    </q-item>
</template>
</q-select>
</div>
<q-chat-message v-show="result_list[0] !== ''" name="我" :
text=ask_list sent></q-chat-message>
<q-chat-message v-show="result_list[0] !== ''" style=
"white-space: pre-wrap;" name="DeepSeek AI"
: text=result_list></q-chat-message>
<q-chip v-show="result_list[0] !== ''" color="orange-12"
text-color="white" icon="warning">
    本回答由 AI 生成，内容仅供参考，请仔细甄别。
</q-chip>
</div>
</q-page-container>
</q-layout>
<div class="q-ma-md bg-white" style="position: sticky; bottom: 0;
z-index: 1000;">
    <q-input class="bg-white" style="position: sticky; bottom: 5vh;
z-index: 1001;" placeholder="输入主诉，生成诊断"
outlined v-model="md.op_name" autogrow>
        <template v-slot:append>
            <q-btn color="green" icon-right="check" label="确定" no-caps
@click="ai()"></q-btn>
        </template>
    </q-input>
</div>
</div>

```

编写专科 AI 问诊的系统提示词，代码如下：

```
#第2章/专科 AI 问诊 Prompt.py
```

```

system_prompt = """你是一名 AI 医师，
回答问题时使用中文，
通过用户提供的科室和主诉内容生成对应科室的诊断结果，
生成的内容必须包括是否有病症，
如果有病症，则生成的内容必须包括症状、疾病名称、治疗方案等。"""

```

调用专科 AI 问诊 API 的代码如下：

```

#第2章/专科 AI 问诊.py
import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from openai import OpenAI
from 专科 AI 问诊 Prompt import system_prompt

```

```

import datetime
from starlette.responses import JSONResponse

api_key = '替换为 DeepSeek 的 API Key'

app = FastAPI()

@app.post("/ai/zkaiwz")
async def 专科AI问诊(prompt: Annotated[str, Body(description="专科 AI 问诊")],
                     department: Annotated[str, Body(description="科室")]):
    try:
        if not prompt:
            return Result(retcode=1, text="必须填写提示词")
        client = OpenAI(api_key=api_key, base_url="https://api.
deepseek.com")

        response = client.chat.completions.create(
            model="deepseek-chat",
            messages=[
                {"role": "system", "content": system_prompt},
                {"role": "user", "content": "主诉内容如下: " + prompt + ", "
                                         "科室如下: " + department},
            ],
            stream=False
        )

        print(response.choices[0].message.content)
        return Result(retcode=0, text="成功",
                      result={
                          "result_list": [response.choices[0].message.
content],
                      })
    except APIException as e:
        return JSONResponse(
            status_code=e.status,
            content={"retcode": e.retcode, "text": e.text, "result":
e.result},
        )

if __name__ == "__main__":
    uvicorn.run(app, host="127.0.0.1", port=8000)

```

2.1.3 标准化 AI 问诊

标准化 AI 问诊的输入模拟传统问诊的流程。针对临床实际情况，首先医师可以详细地询问患者的基础信息、主诉、现病史、既往史、个人史、月经史、性生活史或婚姻史、生

育史、家族史和体格检查等多个方面的相关信息，然后由 DeepSeek 生成诊断结果。标准化 AI 问诊有助于获得更准确的诊断结果。

首先医师在输入框中输入患者的基础信息、主诉、现病史、既往史、个人史、月经史、性生活史或婚姻史、生育史、家族史和体格检查等信息，这些输入框允许留空，然后单击“确定”按钮，即可生成诊断。标准化 AI 问诊页面如图 2-3 所示。

标准化AI问诊

标准化AI问诊

患者基础信息	输入患者基础信息	主诉	输入主诉
现病史	输入现病史	既往史	输入既往史
个人史	输入个人史	月经史	输入月经史
性生活史或婚姻史	输入性生活史或婚姻史	生育史	输入生育史
家族史	输入家族史	体格检查	输入体格检查

确定 ✓

DeepSeek AI

诊断结果:
是否有病症: 是
症状: 发热、咳嗽、咳痰、咽痛、鼻塞、流涕、乏力、头痛、咽干
可能的疾病名称: 上呼吸道感染 (可能是普通感冒或流感)
治疗方案:
对症治疗:
退热: 可以使用退热药物如对乙酰氨基酚 (扑热息痛) 或布洛芬。

图 2-3 标准化 AI 问诊页面

标准化 AI 问诊页面的代码如下：

```

<!-- #第 2 章/标准化 AI 问诊.html -->
<div id="q-app">
  <q-layout view="hHh LpR fFf">
    <q-header elevated class="bg-primary text-white">
      <q-toolbar class="bg-primary text-white q-ny-md shadow-2">
        <q-btn stretch flat label="标准化 AI 问诊" @click="goTo
('http://cnoctave.top')"></q-btn>
        <q-separator dark vertical></q-separator>
      </q-toolbar>
    </q-header>
    <q-page-container>
      <div class="q-ma-md">
        <div class="text-h6">标准化 AI 问诊</div>
        <div class="row" style="display: flex; align-items: center;">
          <div class="col">
            患者基础信息</div>
          <div class="col">
            <q-input class="bg-white" placeholder="输入患者基

```

```

基础信息" outlined v-model="md.hzjcx" autogrow>
    </div>
    <div class="col">
        <div class="text-h6" style="text-align: center;">
主诉</div>
        </div>
        <div class="col">
            <q-input class="bg-white" placeholder="输入主诉"
outlined v-model="md.zs" autogrow>
            </div>
        </div>
        <div class="row" style="display: flex; align-items: center;">
            <div class="col">
                <div class="text-h6" style="text-align: center;">
现病史</div>
            </div>
            <div class="col">
                <q-input class="bg-white" placeholder="输入现病史"
outlined v-model="md.xbs" autogrow>
            </div>
            <div class="col">
                <div class="text-h6" style="text-align: center;">
既往史</div>
            </div>
            <div class="col">
                <q-input class="bg-white" placeholder="输入既往史"
outlined v-model="md.jws" autogrow>
            </div>
        </div>
        <div class="row" style="display: flex; align-items: center;">
            <div class="col">
                <div class="text-h6" style="text-align: center;">
个人史</div>
            </div>
            <div class="col">
                <q-input class="bg-white" placeholder="输入个人史"
outlined v-model="md.grs" autogrow>
            </div>
            <div class="col">
                <div class="text-h6" style="text-align: center;">
月经史</div>
            </div>
            <div class="col">
                <q-input class="bg-white" placeholder="输入月经史"
outlined v-model="md.yjs" autogrow>
            </div>
        </div>
        <div class="row" style="display: flex; align-items: center;">

```

```

        <div class="col">
            <div class="text-h6" style="text-align: center;">
性生活史或婚姻史</div>
            </div>
            <div class="col">
                <q-input class="bg-white" placeholder="输入性生活
史或婚姻史" outlined v-model="md.xs" autogrow>
            </div>
            <div class="col">
                <div class="text-h6" style="text-align: center;">
生育史</div>
            </div>
            <div class="col">
                <q-input class="bg-white" placeholder="输入生育史"
outlined v-model="md.sys" autogrow>
            </div>
        </div>
        <div class="row" style="display: flex; align-items: center;">
            <div class="col">
                <div class="text-h6" style="text-align: center;">
家族史</div>
            </div>
            <div class="col">
                <q-input class="bg-white" placeholder="输入家族史"
outlined v-model="md.jzs" autogrow>
            </div>
            <div class="col">
                <div class="text-h6" style="text-align: center;">
体格检查</div>
            </div>
            <div class="col">
                <q-input class="bg-white" placeholder="输入体格检查"
outlined v-model="md.tgjc" autogrow>
            </div>
        </div>
        <q-btn style="width: 100%" color="green" icon-right="check"
label="确定" no-caps @click="ai()"></q-btn>
        <!-- <q-chat-message v-show="result_list[0] !== ''" name=
"我" :text=ask_list sent></q-chat-message> -->
        <q-chat-message v-show="result_list[0] !== ''" style=
"white-space: pre-wrap;" name="DeepSeek AI"
:text=result_list></q-chat-message>
        <q-chip v-show="result_list[0] !== ''" color="orange-12"
text-color="white" icon="warning">
            本回答由 AI 生成，内容仅供参考，请仔细甄别。
        </q-chip>

```

```

        </div>
    </q-page-container>
</q-layout>
</div>

```

编写标准化 AI 问诊的系统提示词，代码如下：

```
#第 2 章/标准化 AI 问诊 Prompt.py
```

```

system_prompt = """你是一名 AI 医师，
回答问题时使用中文，
通过用户提供的患者基础信息、主诉、
现病史、既往史、个人史、月经史、
性生活史或婚姻史、生育史、家族史和体格检查生成诊断结果，
生成的内容必须包括是否有病症，
如果有病症，则生成的内容必须包括症状、疾病名称、治疗方案等"""

```

调用标准化 AI 问诊 API 的代码如下：

```
#第 2 章/标准化 AI 问诊.py
```

```

import uvicorn
from typing import Union, Any
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from openai import OpenAI
from 标准化 AI 问诊 Prompt import system_prompt
import datetime
from starlette.responses import JSONResponse

```

```
api_key = '替换为 DeepSeek 的 API Key'
```

```
app = FastAPI()
```

```
@app.post("/ai/bzhaiwz")
```

```

async def 标准化 AI 问诊(prompt: Annotated[Any, Body(description="标准化
AI 问诊")]):

```

```
    try:
```

```
        if not prompt:
```

```
            return Result(retcode=1, text="必须填写提示词")
```

```

            client = OpenAI(api_key=api_key, base_url="https://api.
deepseek.com")

```

```
            response = client.chat.completions.create(
```

```
                model="deepseek-chat",
```

```
                messages=[
```

```
                    {"role": "system", "content": system_prompt},
```

```

        {"role": "user", "content": "患者基础信息: " + prompt["hzjcx"] +
        ", 主诉: " + prompt["zs"] +
        ", 现病史: " + prompt["xbs"] +
        ", 既往史: " + prompt["jws"] +
        ", 个人史: " + prompt["grs"] +
        ", 月经史: " + prompt["yjs"] +
        ", 性生活史或婚姻史: " + prompt["xs"] +
        ", 生育史: " + prompt["sys"] +
        ", 家族史: " + prompt["jzs"] +
        ", 体格检查: " + prompt["tgjc"]}},
    ],
    stream=False
)

print(response.choices[0].message.content)
return Result(retcode=0, text="成功",
              result={
                  "result_list": [response.choices[0].message.
content],
              })
except APIException as e:
    return JSONResponse(
        status_code=e.status,
        content={"retcode": e.retcode, "text": e.text, "result":
e.result},
    )

if __name__ == "__main__":
    uvicorn.run(app, host="127.0.0.1", port=8000)

```

2.2 AI 开药

2.2.1 通用 AI 开药

通用 AI 开药的提示词完全由医师或患者决定。首先医师或患者必须在提示词中体现诊断内容，如上呼吸道感染，然后由 DeepSeek 生成药品处方。

诊断可分为确定性诊断、症状性诊断和排除性诊断。在医师或患者输入诊断内容时，需要根据患者的自身情况输入不同类型的诊断结果。如果患者的病因明确且病理确诊，则可以输入确定性诊断结果，例如“2 型糖尿病，血糖控制不佳”；如果患者的病因不明但症状可描述，则可以输入症状性诊断结果，例如不明原因头痛；如果患者在排除严重疾病后归为良性问题，则可以输入排除性诊断结果，例如排除心梗后的胸壁疼痛。

首先医师或患者在输入框中输入诊断结果，然后单击“确定”按钮，即可生成药品处方。通用 AI 开药页面如图 2-4 所示。



图 2-4 通用 AI 开药页面

实现通用 AI 开药页面的代码如下：

```

<!-- #第 2 章/通用 AI 开药.html -->
<div id="q-app">
  <q-layout view="hHh LpR fFf">
    <q-header elevated class="bg-primary text-white">
      <q-toolbar class="bg-primary text-white q-ny-md shadow-2">
        <q-btn stretch flat label="通用 AI 开药" @click="goTo
('http://cnoctave.top')"></q-btn>
        <q-separator dark vertical></q-separator>
      </q-toolbar>
    </q-header>
    <q-page-container>
      <div class="q-ma-md">
        <div class="text-h6">通用 AI 开药</div>
        <q-chat-message v-show="result_list[0] !== ''" name="我"
:text=ask_list sent></q-chat-message>
        <q-chat-message v-show="result_list[0] !== ''" style=
"white-space: pre-wrap;" name="DeepSeek AI"
:text=result_list></q-chat-message>
        <q-chip v-show="result_list[0] !== ''" color="orange-12"
text-color="white" icon="warning">
          本回答由 AI 生成，内容仅供参考，请仔细甄别。
        </q-chip>
      </div>
    </q-page-container>
  </q-layout>
  <div class="q-ma-md bg-white" style="position: sticky; bottom: 0;
z-index: 1000;">
    <q-input class="bg-white" style="position: sticky; bottom: 5vh;
```

```

z-index: 1001;" placeholder="输入诊断, 生成药品处方"
        outlined v-model="md.op_name" autogrow>
        <template v-slot:append>
            <q-btn color="green" icon-right="check" label="确定" no-caps
@click="ai()"></q-btn>
        </template>
    </q-input>
</div>
</div>

```

编写通用 AI 开药的系统提示词, 代码如下:

#第 2 章/通用 AI 开药 Prompt.py

```

system_prompt = """你是一名 AI 医师,
回答问题时使用中文,
通过用户提供的诊断生成药品处方,
生成的内容必须包括药品名称、药品用量和注意事项"""

```

调用通用 AI 开药 API 的代码如下:

#第 2 章/通用 AI 开药.py

```

import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from openai import OpenAI
from 通用 AI 开药 Prompt import system_prompt
import datetime
from starlette.responses import JSONResponse

api_key = '替换为 DeepSeek 的 API Key'

app = FastAPI()

@app.post("/ai/kaiyao")
async def 通用 AI 开药(prompt: Annotated[str, Body(description="通用 AI 开药")]):
    try:
        if not prompt:
            return Result(retcode=1, text="必须填写提示词")
        client = OpenAI(api_key=api_key, base_url="https://api.
deepseek.com")

        response = client.chat.completions.create(
            model="deepseek-chat",
            messages=[
                {"role": "system", "content": system_prompt},

```

```

        {"role": "user", "content": prompt},
    ],
    stream=False
)

print(response.choices[0].message.content)
return Result(retcode=0, text="成功",
              result={
                  "result_list": [response.choices[0].message.
content],
              })
except APIException as e:
    return JSONResponse(
        status_code=e.status,
        content={"retcode": e.retcode, "text": e.text, "result":
e.result},
    )

if __name__ == "__main__":
    uvicorn.run(app, host="127.0.0.1", port=8000)

```

2.2.2 专科 AI 开药

相同的诊断在不同的科室中可能需要不同的药品处方。专科 AI 问诊为了解决这种问题，在 AI 开药的基础上允许医师先选择科室，然后由 DeepSeek 生成药品处方，确保 DeepSeek 的开药结果能够接近所选科室的常用药品范围。

首先医师在输入框中输入诊断结果，并在“科室”下拉菜单中选择合适的科室，然后单击“确定”按钮，即可生成药品处方。专科 AI 开药页面如图 2-5 所示。

图 2-5 专科 AI 开药页面

实现专科 AI 开药页面的代码如下：

```
<!-- #第2章/专科 AI 开药.html -->
<div id="q-app">
  <q-layout view="hHh LpR fFf">
    <q-header elevated class="bg-primary text-white">
      <q-toolbar class="bg-primary text-white q-ny-md shadow-2">
        <q-btn stretch flat label="专科 AI 开药" @click="goTo
('http://cnoctave.top')"></q-btn>
        <q-separator dark vertical></q-separator>
      </q-toolbar>
    </q-header>
    <q-page-container>
      <div class="q-ma-md">
        <div class="text-h6">专科 AI 开药</div>
        <div style="display: flex; justify-content: flex-start;
align-items: center;">
          <div class="text-h6">科室: </div>
          <q-select outlined :model-value="department" use-input
hide-selected fill-input
input-debounce="0" :options="department_list">
            <template v-slot:no-option>
              <q-item>
                <q-item-section class="text-grey">
                  无数据匹配, 请输入正确的科室
                </q-item-section>
              </q-item>
            </template>
          </q-select>
        </div>
        <q-chat-message v-show="result_list[0] !== ''" name="我"
:text=ask_list sent></q-chat-message>
        <q-chat-message v-show="result_list[0] !== ''" style=
"white-space: pre-wrap;" name="DeepSeek AI"
:text=result_list></q-chat-message>
        <q-chip v-show="result_list[0] !== ''" color="orange-12"
text-color="white" icon="warning">
          本回答由 AI 生成, 内容仅供参考, 请仔细甄别。
        </q-chip>
      </div>
    </q-page-container>
  </q-layout>
  <div class="q-ma-md bg-white" style="position: sticky; bottom: 0;
z-index: 1000;">
    <q-input class="bg-white" style="position: sticky; bottom: 5vh;
z-index: 1001;" placeholder="输入诊断, 生成药品处方"
outlined v-model="md.op_name" autogrow>
      <template v-slot:append>
        <q-btn color="green" icon-right="check" label="确定" no-caps
@click="ai()"></q-btn>
      </template>
    </div>
</div>
```

```

        </q-input>
    </div>
</div>

```

编写专科 AI 开药的系统提示词，代码如下：

#第2章/专科 AI 开药 Prompt.py

```

system_prompt = """你是一名 AI 医师，
回答问题时使用中文，
通过用户提供的诊断和科室生成对应科室的药品处方，
生成的内容必须包括药品名称、药品用量和注意事项"""

```

调用专科 AI 开药 API 的代码如下：

```

#第2章/专科 AI 开药.py
import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from openai import OpenAI
from 专科 AI 开药 Prompt import system_prompt
import datetime
from starlette.responses import JSONResponse

api_key = '替换为 DeepSeek 的 API Key'

app = FastAPI()

@app.post("/ai/zkaiky")
async def 专科 AI 开药(prompt: Annotated[str, Body(description="专科 AI 开药")],
                      department: Annotated[str, Body(description="科室")]):
    try:
        if not prompt:
            return Result(retcode=1, text="必须填写提示词")
        client = OpenAI(api_key=api_key, base_url="https://api.
        deepseek.com")

        response = client.chat.completions.create(
            model="deepseek-chat",
            messages=[
                {"role": "system", "content": system_prompt},
                {"role": "user", "content": "诊断如下: " + prompt + ", "
                "科室如下: " + department},
            ],
            stream=False
        )
    
```

```
print(response.choices[0].message.content)
return Result(retcode=0, text="成功",
              result={
                  "result_list": [response.choices[0].message.content],
              })
except APIException as e:
    return JSONResponse(
        status_code=e.status,
        content={"retcode": e.retcode, "text": e.text, "result":
e.result},
    )

if __name__ == "__main__":
    uvicorn.run(app, host="127.0.0.1", port=8000)
```

2.2.3 在药品收费项目范围内开药

不同的医院有不同的药品收费项目范围。为了 AI 开药的结果能够尽量在医院能开出药的范围之内，医师可以在 AI 开药的基础上额外传入药品收费项目范围文档，对 DeepSeek 起到限制作用，从而使 DeepSeek 的开药结果更有实际的参考价值。

首先医师在输入框中输入诊断结果，接着单击“上传文档”按钮上传药品收费项目范围文档，然后单击“确定”按钮，即可生成药品处方。在药品收费项目范围内开药页面如图 2-6 所示。

在药品收费项目范围内开药

在药品收费项目范围内开药

药品收费项目:

我
上呼吸道感染

DeepSeek AI

药品处方
退热镇痛药

药品名称: 对乙酰氨基酚 (扑热息痛)

用量: 每次500mg, 每4-6小时一次, 每日不超过4000mg。

注意事项: 避免与其他含有对乙酰氨基酚的药物同时服用, 以免过量; 肝功能不全者慎用。

止咳药

药品名称: 右美沙芬 (如氢溴酸右美沙芬口服液)

用量: 每次10-20mg, 每日3-4次。

注意事项: 痰多时不宜使用, 避免与酒精同服。

化痰药

药品名称: 氨溴索 (如盐酸氨溴索口服液)

输入诊断, 生成药品处方

图 2-6 在药品收费项目范围内开药页面

在药品收费项目范围内开药，实现此页面的代码如下：

```
<!-- #第2章/在药品收费项目范围内开药.html -->
<div id="q-app">
```

```

<q-layout view="hHh LpR fFf">
  <q-header elevated class="bg-primary text-white">
    <q-toolbar class="bg-primary text-white q-ny-md shadow-2">
      <q-btn stretch flat label="在药品收费项目范围内开药" @click=
"goTo('http://cnoctave.top')"></q-btn>
      <q-separator dark vertical></q-separator>
    </q-toolbar>
  </q-header>
  <q-page-container>
    <div class="q-ma-md">
      <div class="text-h6">在药品收费项目范围内开药</div>
      <div style="display: flex; justify-content: flex-start;
align-items: center;">
        <div class="text-h6">药品收费项目: </div>
        <q-btn color="primary" icon-right="upload" label="上
传文档" @click="goTo('http://cnoctave.top')"></q-btn>
      </div>
      <q-chat-message v-show="result_list[0] !== ''" name="我"
:text=ask_list sent></q-chat-message>
      <q-chat-message v-show="result_list[0] !== ''" style=
"white-space: pre-wrap;" name="DeepSeek AI"
:text=result_list></q-chat-message>
      <q-chip v-show="result_list[0] !== ''" color="orange-12"
text-color="white" icon="warning">
        本回答由 AI 生成, 内容仅供参考, 请仔细甄别。
      </q-chip>
    </div>
  </q-page-container>
</q-layout>
<div class="q-ma-md bg-white" style="position: sticky; bottom: 0;
z-index: 1000;">
  <q-input class="bg-white" style="position: sticky; bottom: 5vh;
z-index: 1001;" placeholder="输入诊断, 生成药品处方"
outlined v-model="md.op_name" autogrow>
    <template v-slot:append>
      <q-btn color="green" icon-right="check" label="确定" no-caps
@click="ai()"></q-btn>
    </template>
  </q-input>
</div>
</div>

```

编写在药品收费项目范围内开药的系统提示词, 代码如下:

```
#第2章/在药品收费项目范围内开药 Prompt.py
```

```
system_prompt = "" "你是一名 AI 医师,
```

回答问题时使用中文，
通过用户提供的药品收费项目范围和诊断生成药品处方，
生成的内容必须包括药品名称、药品用量和注意事项"""

调用在药品收费项目范围内开药 API 的代码如下：

```
#第2章/在药品收费项目范围内开药.py
import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from openai import OpenAI
from 在药品收费项目范围内开药 Prompt import system_prompt
import datetime
from starlette.responses import JSONResponse

api_key = '替换为 DeepSeek 的 API Key'

app = FastAPI()

@app.post("/ai/zyzpsfxmfwnky")
async def 在药品收费项目范围内开药(prompt: Annotated[str, Body(
    (description="在药品收费项目范围内开药"))],
    document: Annotated[str, Body(description="药品收费项目
范围文档路径"))]:
    try:
        if not prompt:
            return Result(retcode=1, text="必须填写提示词")
        client = OpenAI(api_key=api_key, base_url="https://api.
deepseek.com")
        document_text = ""
        with open(document, 'r', encoding='utf-8') as file:
            document_text = file.read().strip()
        if not document_text:
            return Result(retcode=1, text="不支持此药品收费项目范围文档格式，" \
                "或药品收费项目范围文档损坏")
        response = client.chat.completions.create(
            model="deepseek-chat",
            messages=[
                {"role": "system", "content": system_prompt},
                {"role": "user", "content": "诊断如下：" + prompt + "，"
                    "药品收费项目范围如下：" + document_text},
            ],
            stream=False
```

```
)

print(response.choices[0].message.content)
return Result(retcode=0, text="成功",
              result={
                  "result_list": [response.choices[0].message.
content],
              })
except APIException as e:
    return JSONResponse(
        status_code=e.status,
        content={"retcode": e.retcode, "text": e.text, "result":
e.result},
    )

if __name__ == "__main__":
    uvicorn.run(app, host="127.0.0.1", port=8000)
```

2.3 AI 病历管理

2.3.1 AI 病历生成

AI 病历生成的提示词完全由医师决定。首先医师必须在提示词中体现主诉、检查、处置和治疗信息，其他信息可酌情补充，然后由 DeepSeek 生成病历。

病历的组成部分如下。

1. 基本信息

基本信息包括患者基本信息、病历编号和就诊时间等。患者基本信息包括姓名、性别、年龄、出生日期、民族、婚姻状况、职业、住址和联系电话等，有助于医生了解患者的基本情况，为后续的诊断和治疗提供参考。病历编号即医院为患者分配的唯一标识符，用于病历的归档和检索。就诊时间即患者首次就诊的日期和时间 and 后续复诊的时间记录。

2. 主诉与现病史

主诉即患者就诊时最主要的痛苦或最明显的症状、体征及其持续时间。主诉是患者就诊的主要原因，也是医生初步判断病情的重要依据。

现病史即患者从发病到就诊时的疾病发生、发展、变化及治疗经过的详细记录，包括症状的出现时间、性质、程度、缓解或加重因素，以及伴随症状等。现病史是病历的核心部分，对于疾病的诊断和治疗具有重要意义。

3. 既往史与个人史

既往史即患者过去的健康状况和患病情况，包括传染病、外伤、手术、过敏史等。既往史有助于医生了解患者的健康背景，为当前疾病的诊断和治疗提供参考。

个人史包括患者的出生地、居住地、职业、生活习惯和心理社会史等。个人史对于评估患者的健康风险和生活方式对疾病的影响具有重要意义。

4. 家族史

家族史记录患者家族成员的健康状况和患病情况，特别是遗传性疾病和传染性疾病的病史。家族史有助于医生了解患者的遗传背景，为疾病的预防和治疗提供参考。

5. 体格检查部分

体格检查即医生对患者进行的全面身体检查记录，包括生命体征、一般状况、头颈部、胸部、腹部、四肢、脊柱等部位的检查结果。体格检查是病历的重要组成部分，对于疾病的诊断和鉴别诊断具有重要意义。

6. 辅助检查部分

辅助检查包括实验室检查、影像学检查、心电图和内镜检查等的结果记录。辅助检查有助于医生更准确地诊断疾病，评估病情严重程度，制订治疗方案。

7. 诊断与鉴别诊断

诊断即医生根据患者的病史、体格检查和辅助检查结果，初步判断患者所患的疾病。

鉴别诊断即列出与初步诊断相似或可能混淆的其他疾病，并说明鉴别依据。

8. 治疗计划与病程记录

治疗计划即医生根据诊断结果制订的治疗方案，包括药物治疗、手术治疗和物理治疗等。治疗计划应明确、具体，便于患者和家属理解并配合执行。

病程记录即记录患者从就诊到出院或转诊过程中的病情变化、治疗效果、不良反应等情况。病程记录是病历的动态部分，反映了患者疾病的治疗过程和效果。

9. 知情同意书与医患沟通记录

知情同意书即记录患者或家属对治疗方案、手术风险、药物副作用等信息的知情同意情况。知情同意书是保护患者权益和明确医生责任的重要文件。

医患沟通记录即记录医生与患者或家属之间的沟通情况，包括病情解释、治疗方案讨论、心理支持等。医患沟通记录有助于建立良好的医患关系，提高治疗效果。

10. 出院记录与随访计划

出院记录即记录患者出院时的病情、治疗效果、出院医嘱等信息。出院记录是患者出院后继续治疗和康复的重要依据。

随访计划，包括随访时间、随访内容、随访方式等。随访计划有助于医生及时了解患者的康复情况，调整治疗方案。

首先医师在输入框中输入主诉、检查、处置和治疗信息等内容，然后单击“确定”按钮，即可生成电子病历。AI 病历生成页面如图 2-7 所示。



图 2-7 AI 病历生成页面

实现 AI 病历生成页面的代码如下：

```

<!-- #第 2 章/AI 病历生成.html -->
<div id="q-app">
  <q-layout view="hHh LpR fFf">
    <q-header elevated class="bg-primary text-white">
      <q-toolbar class="bg-primary text-white q-ny-md shadow-2">
        <q-btn stretch flat label="AI 病历生成" @click="goTo
('http://cnoctave.top')"></q-btn>
        <q-separator dark vertical></q-separator>
      </q-toolbar>
    </q-header>
    <q-page-container>
      <div class="q-ma-md">
        <div class="text-h6">AI 病历生成</div>
        <q-chat-message v-show="result_list[0] !== ''" name="我"
:text=ask_list sent></q-chat-message>
        <q-chat-message v-show="result_list[0] !== ''" style=
"white-space: pre-wrap;" name="DeepSeek AI"
:text=result_list></q-chat-message>
        <q-chip v-show="result_list[0] !== ''" color="orange-12"
text-color="white" icon="warning">
          本回答由 AI 生成，内容仅供参考，请仔细甄别。
        </q-chip>
      </div>
    </q-page-container>
  </q-layout>
  <div class="q-ma-md bg-white" style="position: sticky; bottom: 0;
z-index: 1000;">
    <q-input class="bg-white" style="position: sticky; bottom: 5vh;
```

```

z-index: 1001;"
    placeholder="输入主诉、检查、处置和治疗信息，生成电子病历"
    outlined v-model="md.op_name" autogrow>
    <template v-slot:append>
        <q-btn color="green" icon-right="check" label="确定" no-caps
@click="ai()"></q-btn>
    </template>
</q-input>
</div>
</div>

```

编写 AI 病历生成的系统提示词，代码如下：

```

#第2章/AI 病历生成 Prompt.py

system_prompt = """你是一名 AI 医师，
回答问题时使用中文，
通过用户提供的主诉、检查、处置和治疗等信息生成电子病历，
生成的内容必须按照电子病历的格式进行输出"""

```

调用 AI 病历生成 API 的代码如下：

```

#第2章/AI 病历生成.py
import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from openai import OpenAI
from AI 病历生成 Prompt import system_prompt
import datetime
from starlette.responses import JSONResponse

api_key = '替换为 DeepSeek 的 API Key'

app = FastAPI()

@app.post("/ai/blgl")
async def AI 病历生成(prompt: Annotated[str, Body(description="AI 病历生成")]):
    try:
        if not prompt:
            return Result(retcode=1, text="必须填写提示词")
        client = OpenAI(api_key=api_key, base_url="https://api.
deepseek.com")

        response = client.chat.completions.create(
            model="deepseek-chat",
            messages=[
                {"role": "system", "content": system_prompt},

```

```
        {"role": "user", "content": prompt},
    ],
    stream=False
)

print(response.choices[0].message.content)
return Result(retcode=0, text="成功",
              result={
                  "result_list": [response.choices[0].message.
content],
              })
except APIException as e:
    return JSONResponse(
        status_code=e.status,
        content={"retcode": e.retcode, "text": e.text, "result":
e.result},
    )

if __name__ == "__main__":
    uvicorn.run(app, host="127.0.0.1", port=8000)
```

2.3.2 AI 病历打印

AI 病历需要打印为纸质版，用于医院工作流程中的病案管理环节。要打印 AI 病历，首先需要将 AI 病历中的文字转换为 Word 文档，然后医师可以按需修改 Word 文档中的内容，最后由医师将 Word 文档打印为纸质版。

首先医师在输入框中输入内容，然后单击“确定”按钮，由 DeepSeek 生成 AI 病历。

医师在生成 AI 病历之后可以检查 AI 病历的正确性和有效性，在检查无误后，单击生成的 AI 病历上方的“下载病历”按钮即可下载 Word 文档。AI 病历打印页面如图 2-8 所示。

AI病历打印

AI病历打印

我

主诉：发热，咳嗽，咳痰，咽痛，鼻塞，流涕，乏力，头痛，咽干。检查信息：上呼吸道感染。治疗信息：对乙酰氨基酚，每次500mg，每日3次；右美沙芬，每次10-20mg，每日3次

下载病历

DeepSeek AI

电子病历
患者基本信息
姓名：XXX
性别：XXX
年龄：XXX
就诊日期：XXXX年XX月XX日
病历号：XXXXXX
主诉
患者主诉发热、咳嗽、咳痰、咽痛、鼻塞、流涕、乏力、头痛、咽干，症状持续X天。
现病史
患者于X天前出现发热，体温最高达XX℃，伴有咳嗽、咳痰（痰液颜色为XX），咽痛明显，鼻塞、流涕，伴有乏力和头痛。无明显呼吸困难、胸痛等症状。自行服用XX药物（如有），症状未见明显缓解。
既往史

输入主诉、检查、处置和治疗信息，生成电子病历

确定

图 2-8 AI 病历打印页面

实现 AI 病历打印页面的代码如下：

```
<!-- #第2章/AI 病历打印.html -->
<div id="q-app">
  <q-layout view="hHh LpR fFf">
    <q-header elevated class="bg-primary text-white">
      <q-toolbar class="bg-primary text-white q-ny-md shadow-2">
        <q-btn stretch flat label="AI 病历打印" @click="goTo
('http://cnoctave.top')"></q-btn>
      <q-separator dark vertical></q-separator>
    </q-toolbar>
  </q-header>
  <q-page-container>
    <div class="q-ma-md">
      <div class="text-h6">AI 病历打印</div>
      <q-chat-message v-show="result_list[0] !== ''" name="我"
:text=ask_list sent></q-chat-message>
      <div style="display: flex; justify-content: flex-start;
align-items: center;">
        <q-btn color="primary" icon-right="download" label=
"下载病历" @click="goTo('http://cnoctave.top')"></q-btn>
      </div>
      <q-chat-message v-show="result_list[0] !== ''" style=
"white-space: pre-wrap;" name="DeepSeek AI"
:text=result_list></q-chat-message>
      <q-chip v-show="result_list[0] !== ''" color="orange-12"
text-color="white" icon="warning">
        本回答由 AI 生成，内容仅供参考，请仔细甄别。
      </q-chip>
    </div>
  </q-page-container>
</q-layout>
  <div class="q-ma-md bg-white" style="position: sticky; bottom: 0;
z-index: 1000;">
    <q-input class="bg-white" style="position: sticky; bottom: 5vh;
z-index: 1001;"
placeholder="输入主诉、检查、处置和治疗信息，生成电子病历"
outlined v-model="md.op_name" autogrow>
      <template v-slot:append>
        <q-btn color="green" icon-right="check" label="确定" no-caps
@click="ai()"></q-btn>
      </template>
    </q-input>
  </div>
</div>
```

编写 AI 病历打印的系统提示词，代码如下：

```
#第2章/AI 病历打印 Prompt.py
```

```
system_prompt = """你是一名 AI 医师，
回答问题时使用中文，
通过用户提供的主诉、检查、处置和治疗等信息生成电子病历，
生成的内容必须按照电子病历的格式进行输出，
生成的内容将用于打印病历，
需要确保生成的内容格式正确，便于打印"""
```

调用 AI 病历打印 API 的代码如下：

```
#第2章/AI 病历打印.py
import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from openai import OpenAI
from AI 病历打印 Prompt import system_prompt
import datetime
from starlette.responses import JSONResponse

api_key = '替换为 DeepSeek 的 API Key'

app = FastAPI()

@app.post("/ai/aibldy")
async def AI 病历打印(prompt: Annotated[str, Body(description="AI 病历打印")]):
    try:
        if not prompt:
            return Result(retcode=1, text="必须填写提示词")
        client = OpenAI(api_key=api_key, base_url="https://api.
deepseek.com")

        response = client.chat.completions.create(
            model="deepseek-chat",
            messages=[
                {"role": "system", "content": system_prompt},
                {"role": "user", "content": prompt},
            ],
            stream=False
        )

        print(response.choices[0].message.content)
        return Result(retcode=0, text="成功",
            result={
                "result_list": [response.choices[0].message.
content],
```

```

        })
    except APIException as e:
        return JSONResponse(
            status_code=e.status,
            content={"retcode": e.retcode, "text": e.text, "result":
e.result},
        )

if __name__ == "__main__":
    uvicorn.run(app, host="127.0.0.1", port=8000)

```

在 AI 病历页面中，每个 AI 病历都配备一个“下载病历”按钮。首先医师可以单击“下载病历”按钮将对应的 AI 病历下载为 Word 文档，然后医师对 Word 文档进行适当修改即可打印为纸质病历。下载病历，调用 API 的代码如下：

```

#第2章/下载病历.py
import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from openai import OpenAI

import datetime
from starlette.responses import JSONResponse
import matplotlib.pyplot as plt
import numpy as np
import json
from starlette.responses import JSONResponse, FileResponse
import markdown
from docx import Document
from docx.shared import Inches
from bs4 import BeautifulSoup
import math

app = FastAPI()

@app.post("/ai/xzbl")
async def 下载病历(
    markdown_content: Annotated[str, Body(description="下载病历")]):
    try:
        if not markdown_content:
            return Result(retcode=1, text="Markdown 不能为空")
        id_ = IdGenerator.gen()
        html_content = markdown.markdown(markdown_content)
        soup = BeautifulSoup(html_content, 'html.parser')

```

```

doc = Document()
for element in soup.children:
    if element.name == 'h1':
        doc.add_heading(element.text, 0)
    elif element.name == 'h2':
        doc.add_heading(element.text, 1)
    elif element.name == 'h3':
        doc.add_heading(element.text, 2)
    elif element.name == 'h4':
        doc.add_heading(element.text, 3)
    elif element.name == 'h5':
        doc.add_heading(element.text, 4)
    elif element.name == 'h6':
        doc.add_heading(element.text, 5)
    if element.name == 'p':
        p = doc.add_paragraph(element.text)
    if element.name == 'blockquote':
        p = doc.add_paragraph(element.text)
    if element.name == 'code':
        p = doc.add_paragraph(element.text)
    if element.name == 'ul' or element.name == 'ol':
        for li in element.find_all('li'):
            p = doc.add_paragraph(li.text, style='ListBullet')
    if element.name == 'strong':
        for em in element.find_all('em'):
            p.add_run(em.text).italic = True
            p.add_run(element.text).bold = True
    if element.name == 'em':
        p.add_run(element.text).italic = True
    if element.name == 'a':
        if element.get('title'):
            p.add_hyperlink(
                element.get('title'),
                address=element.get('href'),
                fragment=element.text)
        else:
            p.add_hyperlink(
                element.text,
                address=element.get('href'),
                fragment=element.text)
    if element.name == 'img':
        doc.add_picture(element.get('src'), width=Inches(1.25))
    if element.name == 'table':
        tr_list = element.find_all('tr')
        th_list = element.find('tr').find_all('th')
        td_list = element.find('tr').find_all('td')
        th_num = len(th_list)
        td_num = len(td_list)
        rows = len(tr_list)

```

```

        cols = math.ceil(max(th_num, td_num / (rows - 1)))
        p = doc.add_table(rows=rows, cols=cols)
        for i in range(rows):
            for j in range(cols):
                if i == 0 and j < th_num:
                    p.cell(i, j).text = th_list[j].text
                else:
                    if j < td_num:
                        p.cell(i, j).text = td_list[(i - 1) * cols +
j].text
                    doc.save(f'{id_}.docx')
                    return FileResponse(path=f'{id_}.docx')
            except APIException as e:
                return JSONResponse(
                    status_code=e.status,
                    content={"retcode": e.retcode, "text": e.text, "result":
e.result},
                )

if __name__ == "__main__":
    uvicorn.run(app, host="127.0.0.1", port=8000)

```

2.4 AI 科研论文检索

科研论文通过严谨的实验设计和数据分析，揭示疾病的发生、发展机制，为医学理论的发展提供新依据。医师可以参考科研论文探索新治疗方法，将新治疗方法应用到临床决策当中，促进诊疗规范的形成，提高医疗质量与安全。

学者在发表科研论文时可以展示自己的研究成果，与同行进行讨论和交流，促进学术思想的碰撞和融合。目前，三甲医院大多通过科研论文搭建学术交流平台，采用多语言国际化的方式发表科研论文，推动国际合作与交流，锻炼医师的科研能力、逻辑思维能力和表达能力，培养医学人才。

由于科研论文检索网站的数量众多，可以检索的内容各有侧重，因此医师需要根据自己的实际需求进行检索。中文医学文献检索网站主要包括中国知网、万方数据知识服务平台和维普期刊资源整合服务平台等，此外还有更专注于医学专业的中国生物医学文献服务系统，即 SinoMed。外文医学文献检索网站主要包括 PubMed、Embase、Web of Science 和 ScienceDirect 等，其中 PubMed 更专注于医学专业。综合型文献检索网站主要包括 Scopus 和 Google Scholar 等，这类检索网站涵盖的学科范围广泛，适合用于交叉学科的研究方向。专业医学数据库主要包括 Cochrane Library 和 ClinicalKey 等，这类网站一般用于检索医学数据，对于实际的医学研究有重要的参考价值。

医师在输入框中输入论文关键词，并在“检索网站”下拉菜单中选择合适的检索网站，然后单击“确定”按钮，即可通过 DeepSeek 检索科研论文。AI 科研论文检索页面如图 2-9 所示。



图 2-9 AI 科研论文检索页面

实现 AI 科研论文检索页面的代码如下:

```
<!-- 第 2 章/AI 科研论文检索.html -->
<div id="q-app">
  <q-layout view="hHh LpR fFf">
    <q-header elevated class="bg-primary text-white">
      <q-toolbar class="bg-primary text-white q-ny-md shadow-2">
        <q-btn stretch flat label="AI 科研论文检索" @click="goTo
('http://cnoctave.top')"></q-btn>
        <q-separator dark vertical></q-separator>
      </q-toolbar>
    </q-header>
    <q-page-container>
      <div class="q-ma-md">
        <div class="text-h6">AI 科研论文检索</div>
        <div style="display: flex; justify-content: flex-start;
align-items: center;">
          <div class="text-h6">检索网站: </div>
          <q-select outlined :model-value="website" use-input
hide-selected fill-input
            input-debounce="0" :options="website_list">
            <template v-slot:no-option>
              <q-item>
                <q-item-section class="text-grey">
                  无数据匹配, 请输入正确的检索网站
                </q-item-section>
              </q-item>
            </template>
          </q-select>
        </div>
      </div>
    </q-page-container>
  </q-layout>
</div>
```

```

        </div>
        <q-chat-message v-show="result_list[0] !== ''" name="我"
:text=ask_list sent></q-chat-message>
        <q-chat-message v-show="result_list[0] !== ''" style=
"white-space: pre-wrap;" name="DeepSeek AI"
        :text=result_list></q-chat-message>
        <q-chip v-show="result_list[0] !== ''" color="orange-12"
text-color="white" icon="warning">
            本回答由 AI 生成，内容仅供参考，请仔细甄别。
        </q-chip>
    </div>
</q-page-container>
</q-layout>
<div class="q-ma-md bg-white" style="position: sticky; bottom: 0;
z-index: 1000;">
    <q-input class="bg-white" style="position: sticky; bottom: 5vh;
z-index: 1001;"
        placeholder="选择检索网站，输入论文关键词，检索论文"
        outlined v-model="md.op_name" autogrow>
        <template v-slot:append>
            <q-btn color="green" icon-right="check" label="确定" no-caps
@click="ai()"></q-btn>
        </template>
    </q-input>
</div>
</div>

```

编写 AI 科研论文检索的系统提示词，代码如下：

```
#第2章/AI 科研论文检索 Prompt.py
```

```

system_prompt = """"你是一名 AI 医师，
回答问题时使用中文，
通过用户提供的论文检索网站名称和论文关键词检索论文，
如果论文检索网站名称为 PubMed，则检索 https://pubmed.ncbi.nlm.nih.gov/网站上的
论文，
如果论文检索网站名称为 Octave 中文网，则检索 http://cnoctave.top/网站上的论文，
如果论文检索网站名称为谷歌学术，则检索 https://scholar.google.com/网站上的论文，
如果论文检索网站名称为万方医学网，则检索 https://med.wanfangdata.com.cn/网站
上的论文，
如果论文检索网站名称为中华医学会，则检索 https://www.cma.org.cn/?c=0 网站上的论文，
如果论文检索网站名称为中国生物医学文献服务系统，则检索 http://www.sinomed.ac.cn/
firstPage.do?method=more 网站上的论文，
如果论文检索网站名称为 NSTL 国家科技图书文献中心，则检索 https://www.nstl.gov.cn/
网站上的论文，
如果论文检索网站名称为中国知网，则检索 https://c61.oversea.cnki.net/网站上的论文，
如果论文检索网站名称为 ClinicalKey，则检索 https://www.clinicalkey.com/网站

```

上的论文，
如果论文检索网站名称为 Embase，则检索 <https://www.embase.com/> 网站上的论文，
生成的内容必须包含论文名称，
必须包含 5 条以上的检索结果"""

调用 AI 科研论文检索 API 的代码如下：

```
#第2章/AI 科研论文检索.py
import uvicorn
from typing import Union
from tools import IdGenerator, Result, APIException
from fastapi import FastAPI, Query, Body, Path, \
    Depends, HTTPException, status, Annotated
from openai import OpenAI
from AI 科研论文检索 Prompt import system_prompt
import datetime
from starlette.responses import JSONResponse

api_key = '替换为 DeepSeek 的 API Key'

app = FastAPI()

@app.post("/ai/kylwjs")
async def AI 科研论文检索(prompt: Annotated[str, Body(description="AI 科
研论文检索")],
                           website_name: Annotated[str, Body(description="AI 科研
论文检索")]):
    try:
        if not prompt:
            return Result(retcode=1, text="必须填写提示词")
        if not website_name:
            return Result(retcode=1, text="必须填写论文检索网站名称")
        client = OpenAI(api_key=api_key, base_url="https://api.
deepseek.com")

        response = client.chat.completions.create(
            model="deepseek-chat",
            messages=[
                {"role": "system", "content": system_prompt},
                {"role": "user",
                 "content": f"论文检索网站:{website_name}。论文关键词:{prompt}"},
            ],
            stream=False
        )

        print(response.choices[0].message.content)
```

```
        return Result(retcode=0, text="成功",
                       result={
                           "result_list": [response.choices[0].message.
content],
                       })
    except APIException as e:
        return JSONResponse(
            status_code=e.status,
            content={"retcode": e.retcode, "text": e.text, "result":
e.result},
        )

if __name__ == "__main__":
    uvicorn.run(app, host="127.0.0.1", port=8000)
```