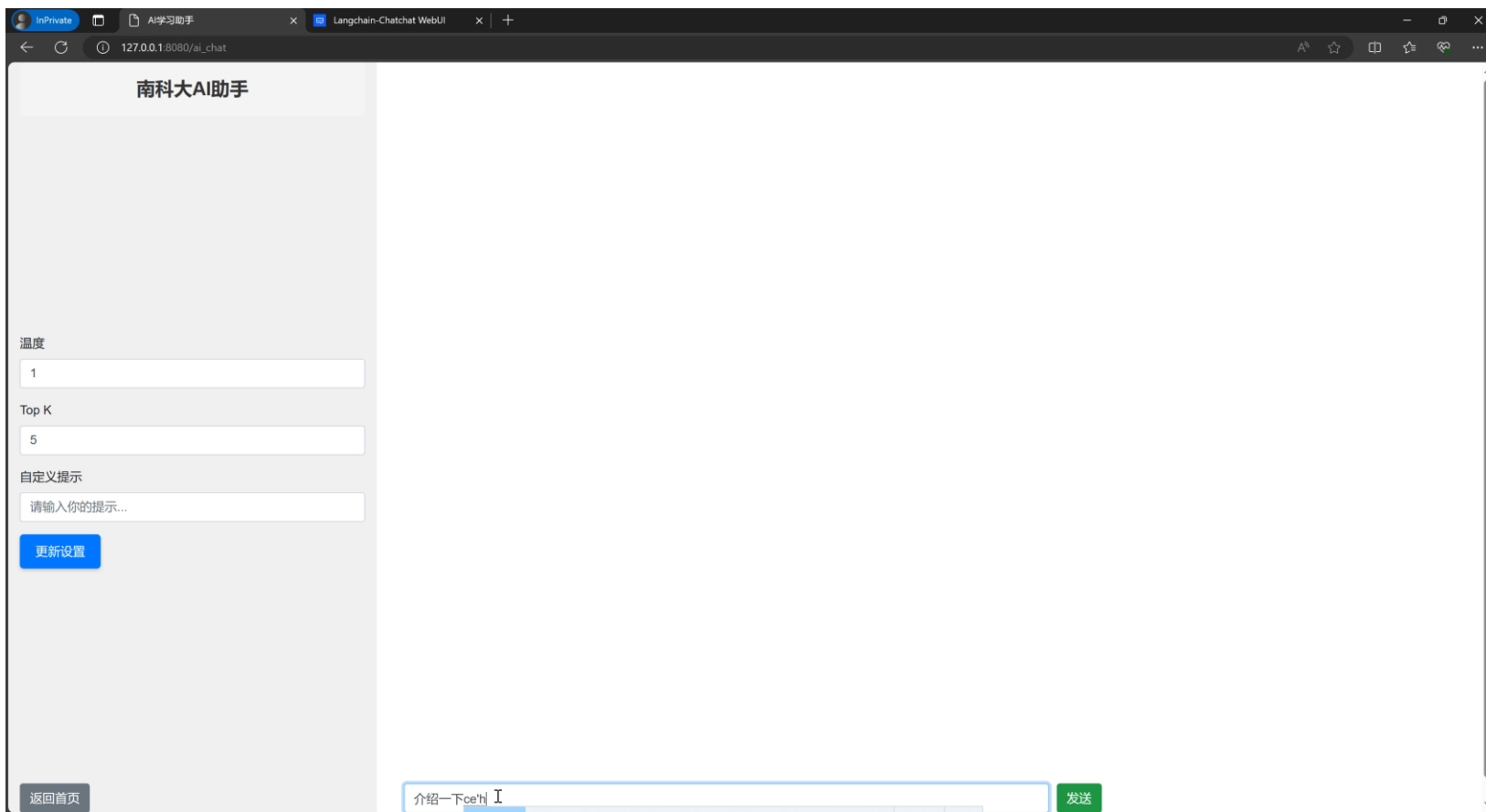


阶段性成果展示

大模型



大模型

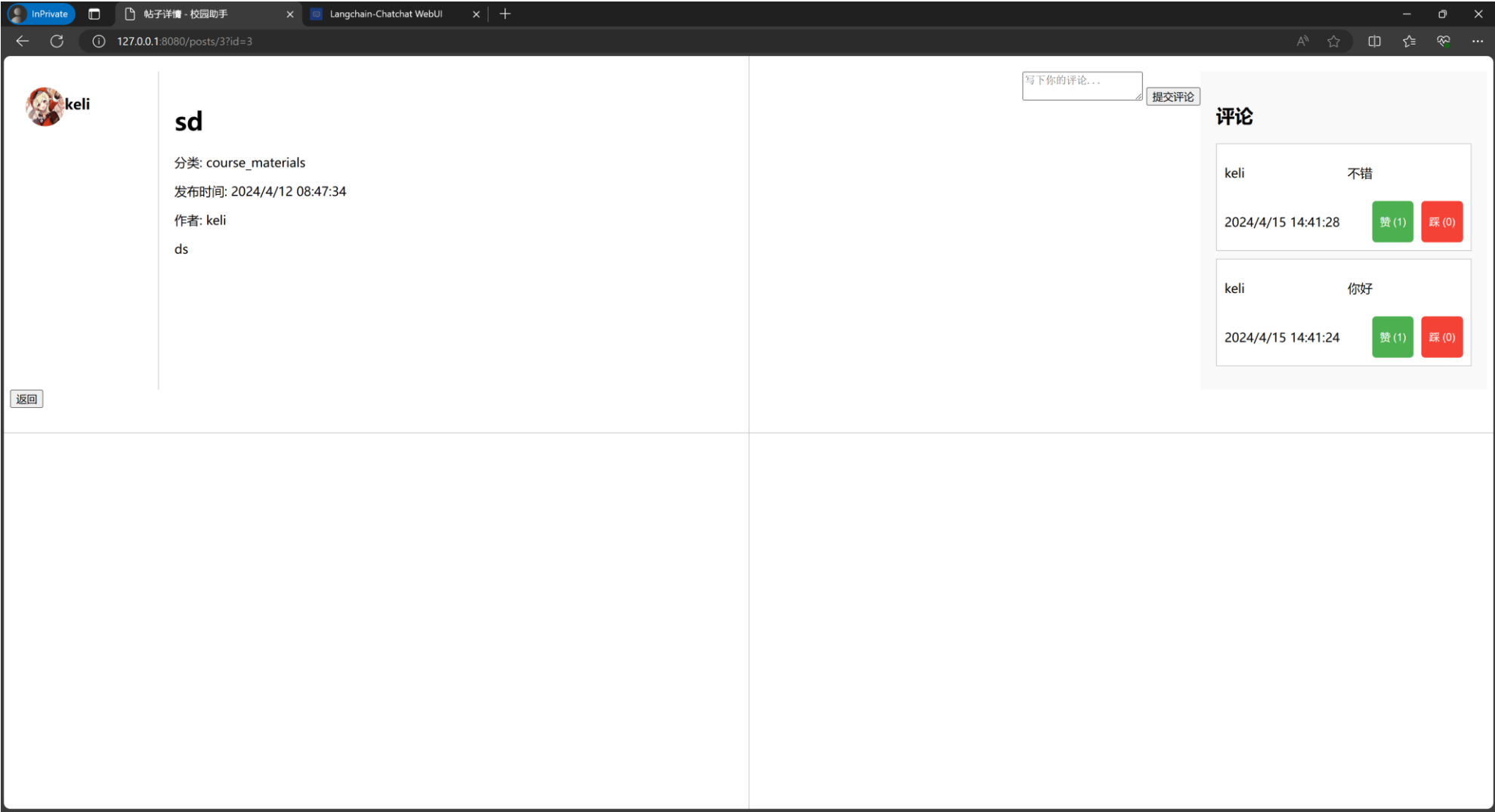
```
async function ask(query, top_k, temperature, stream) : Promise<void> {  
  const data : {...} = {  
    query: query,  
    knowledge_base_name: "samples",  
    top_k: top_k,  
    history: conversationHistory, // 将 history 数组添加到请求数据中  
    score_threshold: 2.0,  
    stream: stream,  
    model_name: "chatglm3-6b",  
    temperature: temperature,  
    max_tokens: 0,  
    prompt_name: "default"  
  };  
};
```

对后端服务器进行查询

```
try {  
  while (true) {  
    const { value : Uint8Array | undefined , done : false | true } = await reader.read();  
    if (done) break;  
  
    let chunk : string = decoder.decode(value, { options: { stream: true } });  
    chunk = previousChunk + chunk;  
    let pattern : RegExp = /\{"answer": "([^"]+)"\}/g;  
    let result;  
    let lastIndex : number = 0;  
  
    while ((result = pattern.exec(chunk)) !== null) {  
      lastIndex = pattern.lastIndex; // Update last index of the pattern  
      displayBotMessage(result[1]);  
    }  
  
    previousChunk = chunk.slice(lastIndex); // Save the remaining chunk  
  
    // Check for the end of the response indicated by the "docs" key  
    if (chunk.includes('"docs":')) {  
      // 将用户的问题添加到 history 中  
      conversationHistory.push({  
        role: "user",  
        content: query  
      });  
    }  
  }  
}
```

对流式传输的内容
进行分块处理
并实时显示

学习建议论坛



学习建议论坛

```
20 usages
@Entity
@Table(name = "posts")
public class Post {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    4 usages
    private String title;
    4 usages
    private String category;
    4 usages
    private String content;
    3 usages
    private String author; // 添加发帖人字段
    4 usages
    private LocalDateTime publishTime; // 添加 publishTime 字段

    4 usages
    @OneToMany(mappedBy = "post", cascade = CascadeType.ALL, orphanRemoval = true)
    private List<Comment> comments = new ArrayList<>();

    public Post() {
    }

    no usages
    public Post(String title, String category, String content, String author, LocalDateTime publishTime) {
        this.title = title;
        this.category = category;
        this.content = content;
        this.author = author;
        this.publishTime = publishTime;
    }
}
```

自动创建数据库的对应表

```
1 usage
public interface PostRepository extends JpaRepository<Post, Long> {

    // 添加新的查询方法以支持多参数搜索和排序
    1 usage
    @Query("SELECT p FROM Post p WHERE " +
        "(:title IS NULL OR p.title LIKE %:title%) AND " +
        "(:category = 'all' OR p.category = :category) " +
        "ORDER BY CASE WHEN :sortBy = 'newest' THEN p.publishTime END DESC, " +
        "CASE WHEN :sortBy = 'oldest' THEN p.publishTime END ASC")
    List<Post> findByTitleContainingAndCategoryOrderByPublishTime(
        @Param("title") String title,
        @Param("category") String category,
        @Param("sortBy") String sortBy);

    // 使用JPQL查询特定类别的帖子，并按发布时间降序排序
    no usages
    @Query("SELECT p FROM Post p WHERE p.category = :category ORDER BY p.publishTime DESC")
    List<Post> findByCategoryOrderByPublishTimeDesc(@Param("category") String category);

    // 使用原生SQL查询特定作者的帖子数量
    no usages
    @Query(value = "SELECT COUNT(*) FROM posts WHERE author = :author", nativeQuery = true)
    int countPostsByAuthor(@Param("author") String author);
}
```

对数据库进行查询

二手交易平台

The screenshot shows an IDE with a project named 'se-project'. The left sidebar displays the project structure, including packages like 'model', 'repository', 'service', and 'task', as well as 'resources' and 'templates'. The main editor window shows the 'Mvcdemo2Application.java' file, which is a Spring Boot application. The code includes package declarations, imports, and a main method that runs the application with custom logging.

```
1 package com.example.mvcdemo2;
2
3
4 > import ...
5
6
7
8
9
10
11
12
13
14 @SpringBootApplication(scanBasePackages = "com.example.mvcdemo2.controller")
15 public class Mvcdemo2Application {
16     4 usages
17     private static final Logger LOGGER = LoggerFactory.getLogger(Mvcdemo2Application.class);
18
19     4 usages
20     public static void main(String[] args) {
21         SpringApplication.run(Mvcdemo2Application.class, args);
22         LOGGER.info("Customized INFO log {}", 1);
23         LOGGER.debug("Customized DEBUG log {}", 2);
24         LOGGER.error("Customized ERROR log {}", 3);
25         LOGGER.trace("Customized TRACE log {}", 4);
26     }
27 }
```

The status bar at the bottom indicates the file path: 'se-project > 12 > SEproject > src > main > java > com > example > mvcdemo2 > Mvcdemo2Application'. It also shows the current time as 15:14, the file encoding as CRLF, the character set as UTF-8, and the number of spaces as 4. The file size is 714 of 2048M.

首页

- 允许用户搜索以及根据某些标准排序和筛选
- 展示热门商品
- 提供功能栏目允许用户发布商品以及查看历史的交易和发布

商品详情界面

- 展示商品的各種信息
- 允許用戶發起聊天
- 展示商品的浏览量
- 允許用戶關注賣家

商品发布界面

- 允许为商品添加各种属性，如标签、描述等
- 允许添加图片

GPA计算器

GPA计算器

课程名称: 编辑课程名称

成绩: 百分制

课程学分

课程1

课程2

课程3

增加课程:

2

增加

开始计算

清空结果

保存结果

其他常见算法说明

(1)标准4.0

成绩	GPA
100~90	4.0
89~80	3.0
79~70	2.0
69~60	1.0
59~0	0

(2)改进4.0(1)

成绩	学分
100~85	4.0
84~70	3.0
69~60	2.0
59~0	0

(3)改进4.0(2)

成绩	学分
100~85	4.0
84~75	3.0
74~60	2.0
59~0	0

(4)北大4.0

成绩	学分
100~90	4.0
89~85	3.7
84~82	3.3
81~78	3.0
77~75	2.7
74~72	2.3
71~68	2.0
67~64	1.5
63~60	1.0
59~0	0

(5)加拿大4.3

成绩	学分
100~90	4.3
89~85	4.0
84~80	3.7
79~75	3.3
74~70	3.0
69~65	2.7
64~60	2.3
59~0	0

展开查看

空间预约

The screenshot displays an IDE environment for a Spring Boot application named 'Mvcdemo2Application'. The project structure on the left includes controllers, data files, models, repositories, services, and tasks. The central editor shows a SQL query filtering the 'reservation' table by date and location, with results showing reservation details like ID, participants, date, location, and time slot.

	id	participants	date	location	time_slot
1	3		1 2021-03-02	aadwdsa	12:00-13:00
2	4		2 2024-04-24	涵泳	09:00-11:00
3	5		1 2024-04-28	琳恩	09:00-11:00
4	6		2 2024-04-20	琳恩	09:00-11:00
5	7		2 2024-04-18	涵泳	17:00-19:00

The console at the bottom shows the application's startup logs, including information about the Spring DispatcherServlet initialization and the execution of the 'TopicPopularity' task.

```
20:51:27.229 [main] INFO c.e.mvcdemo2.Mvcdemo2Application - Started Mvcdemo2Application in 2.953 seconds (process running for 3.502)
20:51:27.230 [main] INFO c.e.mvcdemo2.Mvcdemo2Application - Customized INFO log 1
20:51:27.230 [main] ERROR c.e.mvcdemo2.Mvcdemo2Application - Customized ERROR log 3
20:52:04.015 [http-nio-8080-exec-1] INFO o.a.c.c.C.[Tomcat].[localhost].[/] - Initializing Spring DispatcherServlet 'dispatcherServlet'
20:52:04.015 [http-nio-8080-exec-1] INFO o.s.web.servlet.DispatcherServlet - Initializing Servlet 'dispatcherServlet'
20:52:04.016 [http-nio-8080-exec-1] INFO o.s.web.servlet.DispatcherServlet - Completed initialization in 1 ms
20:52:07.658 [http-nio-8080-exec-4] INFO c.e.mvcdemo2.task.TopicPopularity - Enter main page
```

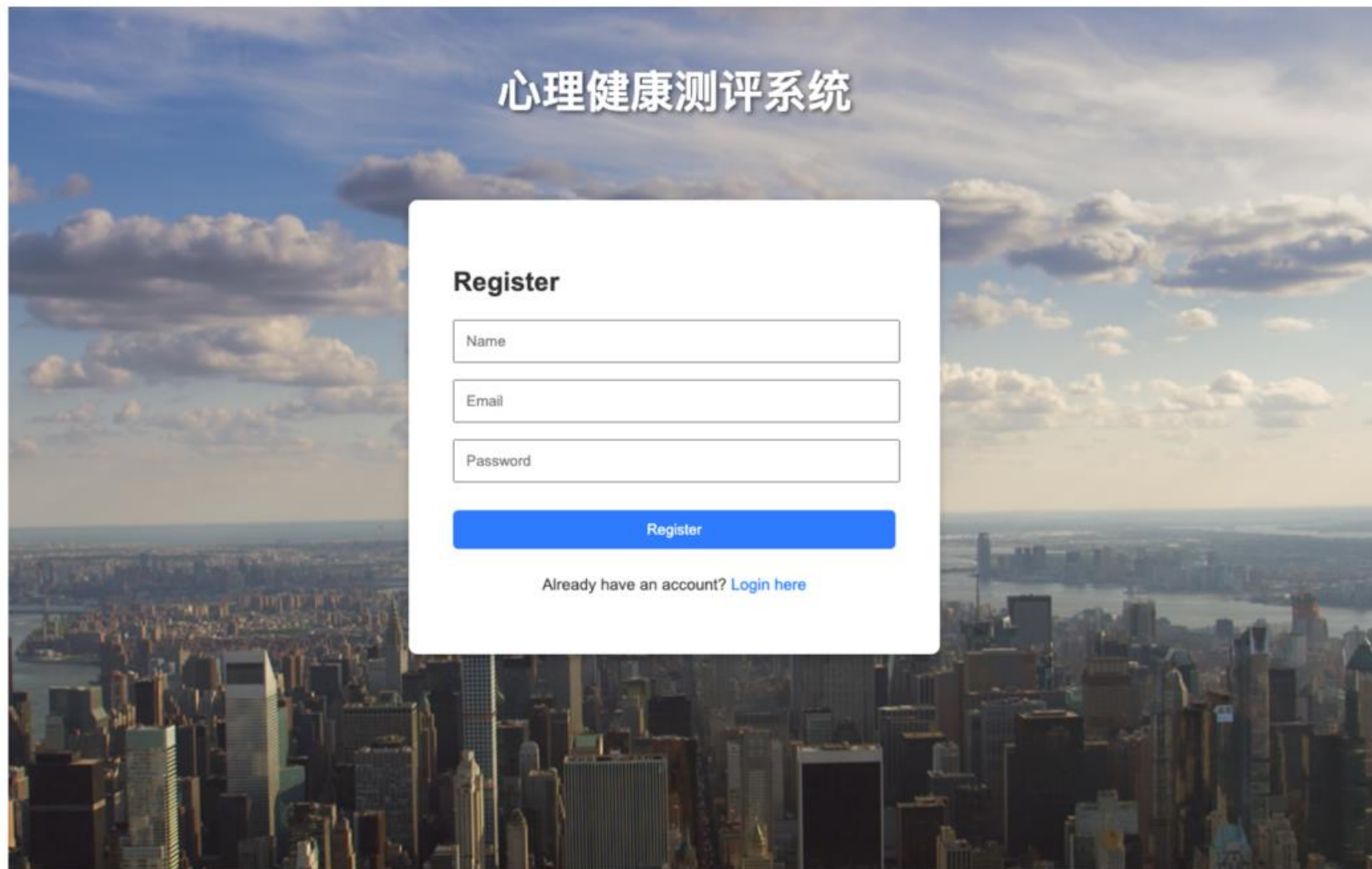
预约空间界面

- 选择预约地点
- 选择预约时间
- 展示预约空间图
- 根据人数预约空间

已预约空间查看界面

- 允许查看已预约的空间列表
- 展示预约目的和预约人

心理健康测评系统



心理健康测评系统

Register

Name

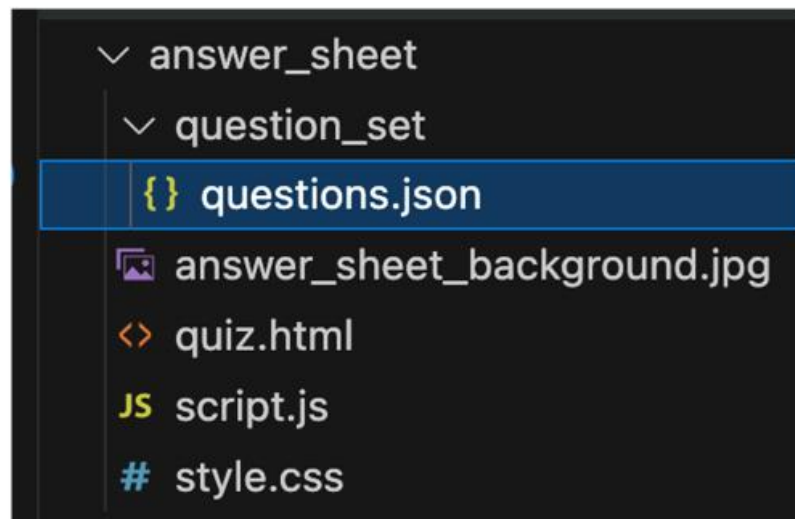
Email

Password

Register

Already have an account? [Login here](#)

题库



通过将题库储存在本地json文件中，通过js将调用json文件的内容返回到前端页面中



题库选择



每一个题库都有开始回答以及查看回答历史两个选项，开始回答新建数据存入数据库中，回答历史是从数据库调取数据展示到前端