

**《软件工程》项目报告**

**题目： MuiOJ在线评测系统**

**课程名称： 软件工程**

**专业班级： CS1904**

**组 名： 反卷督察组**

**同组成员： 学号： U201915065**

**姓名： 谭滨瀚**

**学号： U201915068**

**姓名： 王艳**

**学号： U201915060**

**姓名： 施远栋**

**学号： U201915042**

**姓名： 许楚辉**

**指导教师： 袁凌**

**报告日期： 2021/12/30**

**计算机科学与技术学院**

**任 务 书**

**一 总体要求**

1. 综合运用软件工程的思想，协同完成一个软件项目的开发，掌软件工程相关的技术和方法；

2. 组成小组进行选题，通过调研完成项目的需求分析，并详细说明小组成员的分工、项目的时间管理等方面。

3. 根据需求分析进行总体设计、详细设计、编码与测试等。

**二 基本内容**

根据给出的题目任选一题，自行组队，设计与开发中软件过程必须包括：

**1. 问题概述、需求分析：**正确使用相关工具和方法说明所开发软件的问题定义和需求分析，比如NABCD模型，Microsoft Visio，StarUML等工具 (20%)；

**2. 原型系统设计、概要设计、详细设计**：主要说明所开发软件的架构、数据结构及主要算法设计，比如墨刀等工具（35%）；

**3. 编码与测试**：编码规范，运用码云等平台进行版本管理，设计测试计划和测试用例（30%）；

**4．功能创新**：与众不同、特别吸引用户的创新（10%）；

**5. 用户反馈**：包括用户的使用记录，照片，视频等（5%）。

目录

[1 问题定义 4](#_Toc91851554)

[1.1 项目背景与意义 4](#_Toc91851555)

[1.1.1 用户需求 4](#_Toc91851556)

[1.1.2解决方案 4](#_Toc91851557)

[1.1.3产品优势 4](#_Toc91851558)

[1.1.4市场竞争 4](#_Toc91851559)

[1.2 项目基本目标 5](#_Toc91851560)

[1.3 可行性分析 5](#_Toc91851561)

[1.4人员管理和项目进度管理 5](#_Toc91851562)

[2 需求分析 7](#_Toc91851563)

[2.1 需求分析概述 7](#_Toc91851564)

[2.1.1 E-R图 7](#_Toc91851565)

[2.1.2用例图（黑体4号加粗,字母、阿拉伯数字为Time New Roman4号加粗） 8](#_Toc91851566)

[2.2 原型系统设计 8](#_Toc91851567)

[2.2.2原型设计工具 8](#_Toc91851568)

[2.2.3 原型设计思路 8](#_Toc91851569)

[原型设计成果 8](#_Toc91851570)

[3.1 系统结构 13](#_Toc91851571)

[3.1.1 FrontEnd 13](#_Toc91851572)

[3.1.2 BackEnd 13](#_Toc91851573)

[3.2关键数据结构定义 14](#_Toc91851574)

[3.2.1 前端数据结构 14](#_Toc91851575)

[3.2.2 后端数据结构 16](#_Toc91851576)

[3.2.3数据库表项设计 17](#_Toc91851577)

[3.3 关键算法设计 18](#_Toc91851578)

[3.3.1 BackEnd 18](#_Toc91851579)

[3.3.1 FrontEnd 20](#_Toc91851580)

[3.4 数据管理说明 21](#_Toc91851581)

[4 实现与测试 21](#_Toc91851582)

[4.1实现环境与代码管理 21](#_Toc91851583)

[4.2 关键函数说明 22](#_Toc91851584)

[4.2.1 backend 22](#_Toc91851585)

[6.3 测试计划和测试用例 24](#_Toc91851586)

[4.3.1 backend 24](#_Toc91851587)

[4.3.1 frontend 24](#_Toc91851588)

[4.4 结果分析 24](#_Toc91851589)

[5 总结 32](#_Toc91851590)

[5.1 用户反馈 32](#_Toc91851591)

[5.1.1用户反馈样例 32](#_Toc91851592)

[5.1.2用户反馈总结 32](#_Toc91851593)

[全文总结 33](#_Toc91851594)

[5.2 全文总结 33](#_Toc91851595)

[5.2.1 谭滨瀚工作总结 33](#_Toc91851596)

[5.2.2施远栋工作总结 33](#_Toc91851597)

[5.2.3许楚辉工作总结 34](#_Toc91851598)

[5.2.4王艳工作总结 34](#_Toc91851599)

[6 体会 34](#_Toc91851600)

[附录 37](#_Toc91851601)

# 1 问题定义

## 1.1 项目背景与意义

现行OJ平台UI不够现代化，用户追求更好的UI界面；功能过于臃肿不利于用户选择；大量用户集中使用几个OJ平台导致高并发运行时运行速度过慢。设计一个观赏性和性能更加的在线评测系统，对于提升用户体验来说具有重要意义。

### 1.1.1 用户需求

* 多数OJ的UI界面比较成熟，但不够现代化。
* 大多数OJ平台功能更加丰富，但对于大多数受众来说功能过于臃肿。
* 使用高峰期评测系统效率低下。

### 1.1.2解决方案

* 创建一个实现主流编程语言编译运行功能的web应用。
* 重设页面分区。
* 优化后端并行调度。

### 1.1.3产品优势

* 具有更加美观的UI界面。
* 覆盖语言不多但是足以满足大多数人的需求。
* 针对受众不广，OJ性能得到进一步保证。

### 1.1.4市场竞争

* 现行OJ用户交互功能更加完善，诸如评论区等功能，后期看进度再进行完善。
* 题库覆盖更加全面，有更好的阶段性做题索引。
* 产品推广
* 在班级，系内做广。
* 和学院合作。
* 给予用户良好的印象，让用户口口相传，让更多的非用户自然地成为用户。

## 1.2 项目基本目标

* 实现用户管理系统。包括用户注册，信息增删查改，用户注销功能。
* 实现代码在线编译。包括代码高亮实现，编译实现，在线编辑和题目评测功能实现。
* 实现提交结果展示功能，OJ优化更新公告界面。
* 在实现上述功能基础下，后续开放竞赛模块，逐步实现评论区、排行榜等功能。

## 1.3 可行性分析

通过以上的NABCD模型分析可以看出，我们的产品是基于用户的需求出发设计，因此更能满足用户的需求并为用户带来便利，同时也改善了现有OJ的一些不足之处。在对目前大多数同类产品分析来看，我们的产品具有独特的创新点，所以在市场方面我们面向用户的出发点具有一定的可行性。同时在实现方面，基于项目要实现的基本目标和思维导图，我们可以很快的用react框架构建出前端网页。基于VS Code的语法编辑和代码高亮可以很方便的为我们解决编译器实现问题，而用户数据管理以及相关测评数据管理等后端操作事项有PHP,Python（flask框架），java(ssm框架）等方向来实现。因此总体来看我们的产品具有较高的可行性。

## 1.4人员管理和项目进度管理

小组成员有: 谭滨瀚, 施远栋, 王艳, 许楚辉, 分工如下:

* 谭滨瀚: 负责所有后端功能的实现和前端页面的debug维护，负责项目的全自动ci/cd部署以及HTTPS加密等运维操作
* 王艳：负责前端框架搭建，实现题库展示，代码编辑提交功能，用户信息页面；修改登录注册功能等；以及数据与后端的联调
* 施远栋: 负责部分题库数据集的清洗及导出，负责网站主页及提交记录页面的实现
* 许楚辉：负责部分题库数据集的清洗及导出，协助网页前端开发

# 2 需求分析

## 2.1 需求分析概述

### 2.1.1 E-R图

图示

描述已自动生成

图2-1在数据库中存放的关系表项



### 2.1.2用例图（黑体4号加粗,字母、阿拉伯数字为Time New Roman4号加粗）

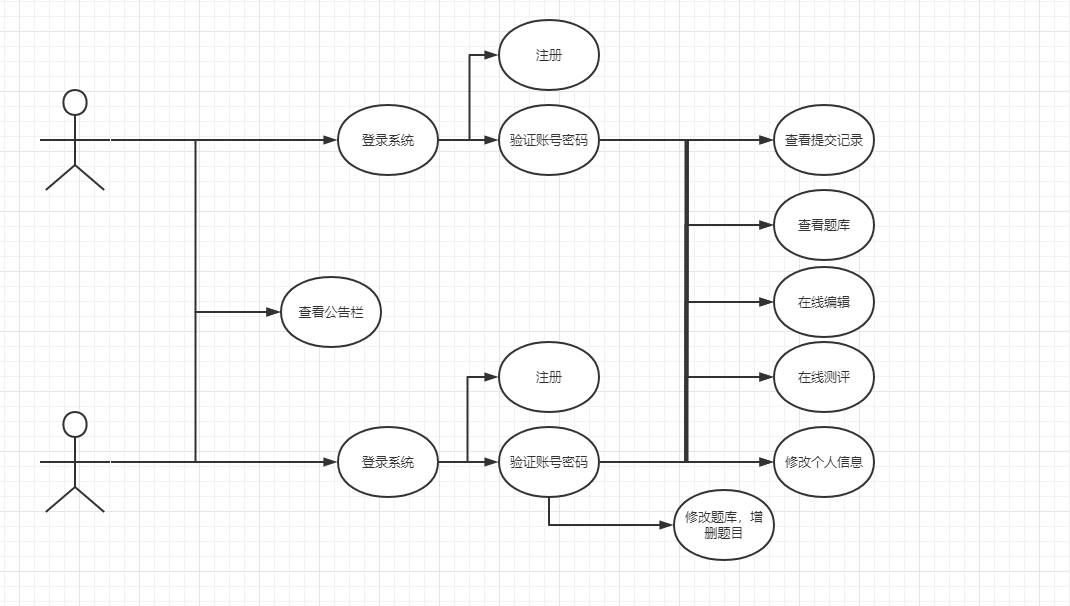


图2-2用户用例示范图

## 2.2 原型系统设计

运用工具设计原型系统，从而更准确说明主要功能和用户交互界面。

### 2.2.2原型设计工具

我们小组的原型系统设计采用了在线平台ProcessOn和墨刀免费版。

### 2.2.3 原型设计思路

我们小组参考了各大OJ平台的界面设计，结合同学和自己的理解，我们认为简洁明了的环境最适合我的项目。所以，在设计中我们采用尽量精简的元素数量，只是对单个元素进行美化，使得我们的界面简洁而又不失美感。

### 原型设计成果

首先是注册界面的原型图：



图2-3注册原型图

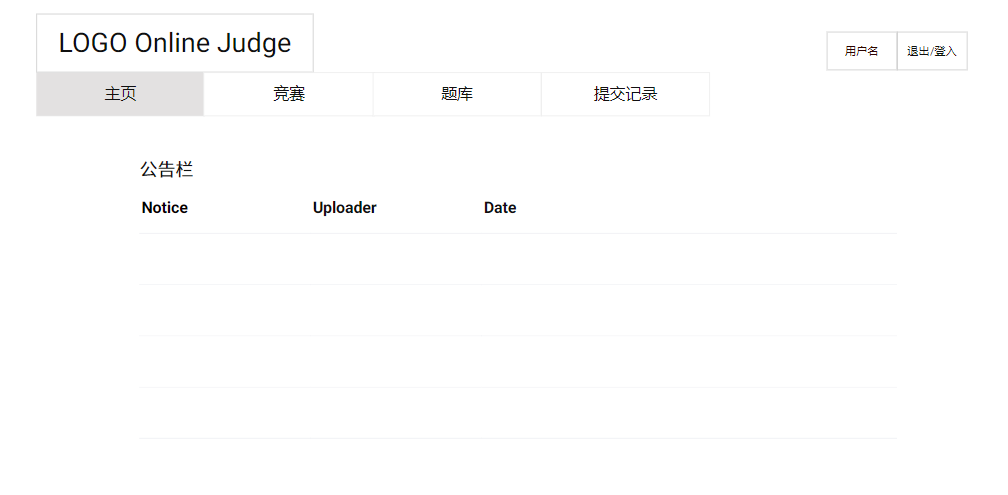
其次是登录界面：



图2-4登录原型图

登录的时候可以选择以用户名的方式登录或者以邮箱的方式登录。

接下来是主要页面，分别是首页原型图：

  
图2-5首页原型图

题库原型图：

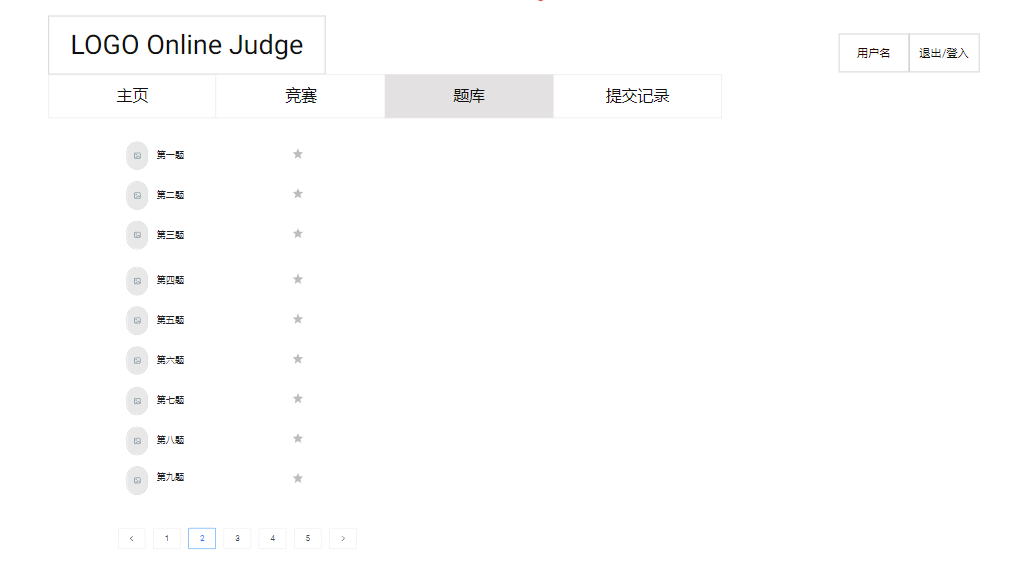


图2-6 题库原型图

题目原型图：

  
图2-7 题目原型图

提交界面原型图：



图2-8 提交界面原型图

**3 概要设计和详细设计**

## 3.1 系统结构

### 3.1.1 FrontEnd

前端系统分为9个模块：

1. 注册账号模块。注册功能分为六个步骤，填入用户名、填入密码、点击注册按钮、传递用户信息的后端管理判断、后台返回响应。
2. 账号登录模块。登录支持用户名密码登录和邮箱密码登录。主要的登录功能分为五个步骤，填入用户名、填入密码点击登录按钮、传递用户信息的后端管理判断，后台返回响应：若为已注册用户则登录成功，若为未注册账号则自动跳转到注册页面。
3. 用户退出登录模块：清楚token记录并跳转到登录页面。
4. 登录拦截模块。该模块主要实现未登录拦截，若用户处于未登录状态将无法跳转到题库等页面。
5. OnlineJudge主界面模块。主要功能：公告栏展示近期的更新的信息或者链接。
6. OnlineJudge题库模块：从后端获取所有题目数据，并展示在页面上，考虑到题目数量过多，有分页功能；并且点击题目可以跳转到题目详情页。
7. OnlineJudge题目编辑模块：根据题目id跳转到相应页面。题目编辑页面分为：题目信息展示，输入例子展示，输出例子展示，以及monaco编辑区，提交题目按按钮，点击按钮将用户编辑的数据传回后端处理并返回。
8. OnlineJudge提交记录展示：提交记录展示主要步骤：前端展示后端处理数据的结果，展示用户提交的题目ID、运行的结果为AC或者CE、运行时间、运行所占内存、使用的语言、以及提交的时间；同样实现了分页功能；
9. OnlineJudge个人信息模块。主要功能：展示用户的用户名、用户ID、用户头像、用户邮箱，以及是否为管理员，算法题目总的提交次数，以及题目的总通过次数。提供给所有用户跳转到其他页面的按钮。

### 3.1.2 BackEnd

后端系统分为11个模块：

1. 数据库连接模块。使用到了Golang中的Xorm库进行与Mysql数据库的连接和交互，Xorm提供了非常多而易用的Api，针对关系型数据库进行了包装，优化了SQL语句的运行时间和效率
2. Web服务模块。使用到了Golang中的Gin框架，为前端系统提供必要的Web远程接口，完成例如用户注册，登录，修改，查看题目，提交结果并通知后端其他模块按需执行的功能
3. 认证模块。使用到了在微服务中非常常见的Json-Web-Token模式，将用户认证的信息通过JWT进行签名，从而确保了用户身份认证的安全性和可靠性
4. 用户信息模块。提供了用户注册，登录，以及获取自己信息的功能
5. 问题模块。提供了问题的创建，修改，删除，查看，以及按列显示摘要的功能
6. 提交模块。提供了提交的创建，查看功能。经过用户提交的记录将会由评测模块进一步处理
7. 工具模块。提供了一系列工具给其他模块使用
8. Docker模块。提供了关于docker/api的一些包装，使其适配本项目的后端使用
9. 密码验证模块。采用了MD5提取密码摘要，将摘要作为盐值与原密码混合，再次MD5提取摘要。提供了密码的加密和验证功能
10. 配置模块。读取各项模块所需要的配置，并应用至各项模块中
11. CORS模块。作为后端中间件提供一些必要的HTTP Header来帮助前端实现跨域
12. 评测模块。当收到提交模块的请求时，调用Docker模块启动虚拟机，编译代码，收集报错信息，并进行代码评测功能

## 3.2关键数据结构定义

### 3.2.1 前端数据结构

1. 登入界面主要数据：username、password分别绑定前端页面的用户名输入框和密码输入框中的内容，并通过基于Promise的fetch方法打包传递给后端服务器。
2. 注册界面主要数据：username、password、repassword分别绑定前端页面的用户名输入框和密码输入框、确认密码输入框中的内容，并并通过基于Promise的fetch方法打包传递给后端服务器。
3. 题库页面主要数据：problemList是一个具体的字典数组，数组中元素是用来存储每门课的基本信息的，数组元素具体格式如下：

problemList[

{

tid:" ",

title:" ",

attempt:" ",

} ]

name表示算法题的ID，title表示算法题的题目，attempt表示当前用户对

该道题的提交次数;

1. 题目编辑页面主要数据：data是一个具体的字典数组，数组中元素是用来存储用户讨论的基本信息的，数组元素具体格式如下：

Data:

{

content: '',

accept: 0,

create\_at: '',

sampleIn: '',

sampleOut: ''

})

content用来表示题目信息，accept用来表示用户通过次数，create\_at用来表示题目提交时间，sampleIn用来表示该提的输入样例，sampleOut用来表示输出样例;

1. 个人信息界面主要数据：rows是用户信息展示的所需的数据，具体如下：

const rows = [

createData('用户名', user?.username),

createData('用户ID', user?.uid?.toString()),

createData('用户邮箱', user.?email),

createData('是否管理员', user?.is\_admin?.toString() === 'true' && '是' || "否") createData('提交次数', data.message.attempt.toString())

createData('通过次数', data.message.accept.toString()) ];

此处引入了Mui Table的组件。 首页主要数据：const rows 为要现实的数据的集合

每个row由下述函数获得

createData(

Notice: string,

Way: string,

Uploader: string,

Date: string,

) {

*return* { Notice, Way, Uploader, Date };

}

1. 提交记录页面主要数据： 数据结构为

const rows=[

createData(

​ item.question\_title,

​ item.status,

​ item.time\_used,

​ item.space\_used,

​ item.language,

​ item.created\_at

)

]

### 3.2.2 后端数据结构

1. JWT数据结构  
   JWT的数据结构中包含三部分  
   a) Header 部分是一个 JSON 对象，描述 JWT 的元数据

{

"alg": "HS256",

"typ": "JWT"

}

alg属性表示签名的算法（algorithm），默认是 HMAC SHA256（写成 HS256）；typ属性表示这个令牌（token）的类型（type），JWT 令牌统一写为JWT。

b) Payload 部分也是一个 JSON 对象，用来存放实际需要传递的数据。

{

"sub": "1234567890",

"name": "John Doe",

"admin": true

}

c) Signature 部分是对前两部分的签名，防止数据篡改。

首先，需要指定一个密钥（secret）。这个密钥只有服务器才知道，不能泄露给用户。然后，使用 Header 里面指定的签名算法（默认是 HMAC SHA256），按照下面的公式产生签名。  
HMACSHA256(

base64UrlEncode(header) + "." +

base64UrlEncode(payload),

secret)  
算出签名以后，把 Header、Payload、Signature 三个部分拼成一个字符串，每个部分之间用"点"（.）分隔，就可以返回给用户。

所以最后的JWT的数据结构会被格式化为三部分

Header.Payload.Signature 其中Header和Payload会被Base64编码

1. HTTP body接受数据结构  
   前端在与后端进行操作的过程中，必然需要与后端有数据的通信，因此这些数据被定义在了一个json文件中，我们在Golang中定义了许多的DataForm，用于Gin框架解析前端传来的信息并将数据一一绑定在数据结构上

由于处理的数据结构众多，这里不一一列举

1. 数据库表项对应的数据结构

用于在Xorm操作的过程中，将数据从关系式数据库中提取出来并存放而定义的一系列数据结构，具体的数据结构与E-R图中所展示的内容大致相同

1. HTTP返回的数据结构

用于发送Token，信息交互的而定义的一些数据结构，所有的数据结构都被用于将数据格式化为Json，Stringify之后发回前端

1. 后端服务之间交互的数据结构

用于在后端服务之前传递信息而定义的一些数据结构

### 3.2.3数据库表项设计

本次数据库表项一共设计了十个表，但是由于最终的工程量的问题只用到了其中的五个

1. User

用于存储用户信息，比如username，password  
用户在注册之后，Xorm会在关系式数据库中使用这样的一个Schema建立一个表项，该表项会用于之后登录的密码验证，用户信息的查找，以及提交记录的查询  
每个用户在注册之后，会获得一个全局唯一的Uid，用于在其他服务中标识自己，我们在设计数据库接口的时候已经保证其全局唯一，因此也可作为用户的唯一认证标识去使用

每一位用户都有着一个isAdmin的字段去标识是否为管理员，以通过身份去验证其管理员的身份，并提供诸如提交题目，修改题目等超级管理员的权限

1. Question

存储着题目的一些基本信息，比如题目，难度，提交记录，创建该题目的用户的Uid

当题目创建的时候，题目的详细信息和评测信息也会随之创建

与用户一样，题目也拥有着全局唯一的Tid，用于在其他服务中标识题目

1. Question\_Content

存储的题目的详细描述信息和测试样例，便于用户查看并根据样例编程bushi

1. Question\_Judge

存储着题目的评测信息，包括数据集等

1. Submission

存储着用户的提交记录，同时保存了当前提交的运行状态

每一个Submission都唯一的对应着一个tid和一个uid，并自身拥有一个全局唯一的sid，用于提交记录的精细查找等操作

## 3.3 关键算法设计

### 3.3.1 BackEnd

BackEnd实现的最主要的两大功能，就是Web前端与后端服务之前的交互，以及后端服务与评测端的交互

1. 对于整体的Web模块的描述

整个Web模块是以Gin框架作为基础，在Gin框架提供诸如Socket端口打开监听，路由转发和Callback回调的功能基础之上，添加了一些额外的中间件来实现适合本后端的一些功能，比如用户的认证和CORS模块的跨域解决方案，都是中间件的一部分，只不过他们处理的一些是Request，一些是Response，当数据流过中间件之后，他们就如同在流水线上进行了一些处理一样，提供了一些额外的信息给后端的服务，比如用户是否进行了认证，认证是否有效，是否是管理员身份等等对于后端服务来说完全无感的操作，真正的做到的模块化的实现。

而对于HTTP的请求来说，无外乎都是首先通过Gin的Router，将请求转发到对应的callback函数进行处理，在函数处理的过程中，会调用后端提供的各种服务，而服务之间也会相互调用，比如在User服务中，我们会调用到数据库的服务和工具模块的服务，最终，在一切正常的情况下，我们返回状态码200，并生成对应的一些正确处理的response

图示

描述已自动生成

图3-1 Web模块的流程图

1. 评测端与后端服务之前的交互模块

这里的后端服务指的是由Golang Main Thread引导和执行的一些服务，后端服务通过调用评测服务的一些函数来完成评测的整个过程。  
之所以我们把评测端与主线程分离出来，主要是想要说明我们整个的一个后端运行接口。作为评测端本身，其实是可以脱离主线程运行的，甚至可以在不同的主机上运行，通过gRPC的方式去调用远程云函数，还可以加上一些MessageQueue的方式去达到负载均衡的一些效果，因为本次工程的时间限制，上述的内容其实都是画大饼，最终的实现是将评测端与主线程放在了一起

未来改进的方向是将Main端与Mysql进行docker compose的融合，然后将Judge端与Main端分离，形成1Main NWorkers的结构，实现分布式的评测，加快速度

评测端采用的主要就是使用docker虚拟环境的方式，从严格的意义上来讲docker并不能被看作是虚拟机，而是利用了linux不同命名空间而实现的，docker与host使用的core和kernel并无区别，但将代码的编译和执行放在docker中，将评测环境与生产环境相隔离，提供了更加安全的操作，避免了恶意代码对于评测系统的破坏

图示

描述已自动生成

图3-2 Main-Worker结构的评测系统

### 3.3.1 FrontEnd

前端的算法设计主要体现在页面渲染和数据传输方面。

1. 页面渲染

在使用react进行前端页面渲染的时候，需要使用到父路由、子路由、父组

子组件等。首先整体导航栏部分是为一个父组件，每次点击跳转到其他页面的时候，导航栏都保持不变；其余组件大体作为一个子组件。跳转到一个页面下的另一个模块的时候需要导入子路由，保证了父页面在跳转之后仍然不变。

1. 数据传递

父子组件传递的时候，父组件在调用子组件的时候可以直接在后面加上需要传递的数据。

本次项目更多的用到兄弟组件之间的传值，如下图所示：

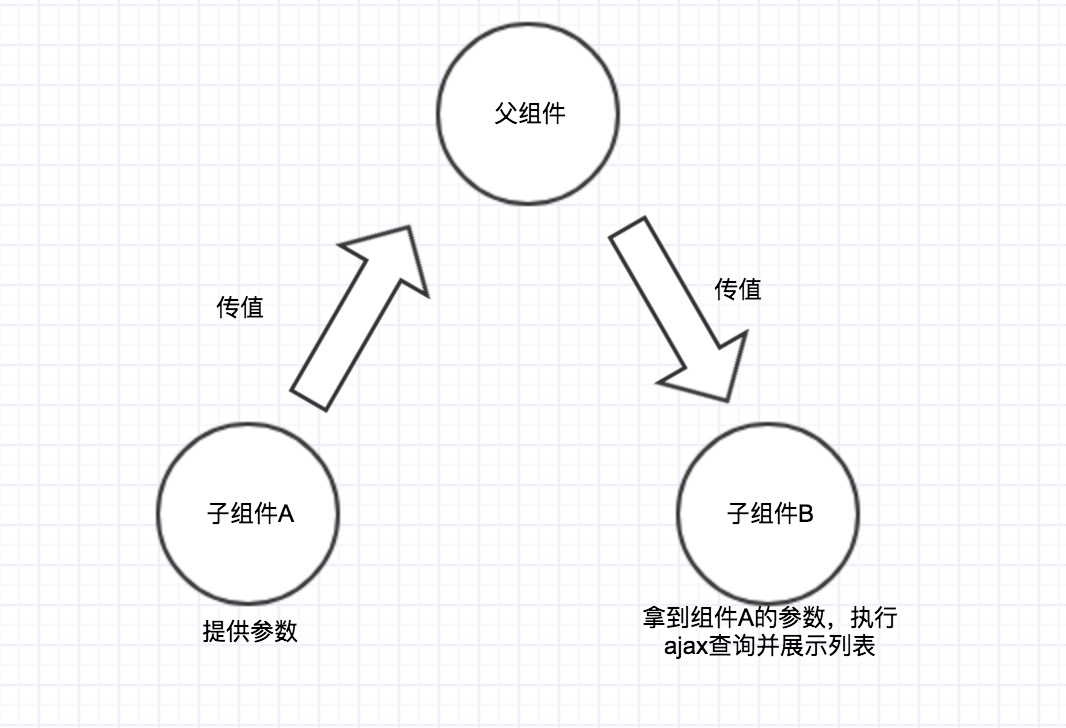


图3-3 兄弟组件之间的传值

## 3.4 数据管理说明

整个系统的数据由关系式数据库mysql进行存放和处理，前端并不存在直接操作数据库的问题，所有的处理都是由后端经过中间件的验证之后提交到Xorm进行处理。相对而言比较的安全和规范

4 实现与测试

## 4.1实现环境与代码管理

##### 整体代码管理使用Gitee进行管理，整体通过vscode终端进行git push、pull、add、commit、fetch、remote、branch等一系列操作进行管理。gitee页面如下图：



图4-1 码云平台代码签入示例

## 4.2 关键函数说明

### 4.2.1 backend

1. 数据库连接模块。

xorm.NewEngine("mysql", mysqlConnect)

xorm提供的函数，用于建立一个连接数据库的通道

db.DB.Ping()

用于测试数据库是否联通

1. Web服务模块。

server := gin.Default()

用于Gin框架的启动

server.Use(middlewares.Cors())

用于Gin框架在处理Response时调用中间件流水线处理

user.Router(server)  
question.Router(server)  
submission.Router(server)

用于将Gin框架接收到的请求转发到各个分支Router上

1. 认证模块。使用到了在微服务中非常常见的Json-Web-Token模式，将用户认证的信息通过JWT进行签名，从而确保了用户身份认证的安全性和可靠性

func AuthJWT() gin.HandlerFunc

返回一个gin框架的处理函数，用于对用户传回的Token进行验证和身份认证

func SignJWT(user \*models.User) (string, error)

提供了将JWT进行签名验证的功能

1. Docker模块。

func InitDocker()

使用参数模块中的提供的参数初始化这个模块

func InitDockerImages()

初始化docker镜像，如果镜像缺失则拉取缺失镜像

1. 密码验证模块。

func SaltPasswdWithSecret(passwdMd5 string) string

使用MD5提取密码加盐的摘要，将密码加密保存到数据库中

1. 评测模块。当收到提交模块的请求时，调用Docker模块启动虚拟机，编译代码，收集报错信息，并进行代码评测功能

func Scheduler(request \*judger.StarterType) (string, []\*judger.JudgeCaseResult, error)

评测模块中的核心调度函数，执行了绝大部分评测模块中的代码

func Runner(  
 sid uint32, codePath string,  
 compileInfo \*judger.CompileInfo,  
 testCases []\*models.Test,  
 timeLimit, spaceLimit, outputPath string,  
 code []byte, buildProduction []byte,  
) (map[string][]byte, error)

评测模块中的运行时函数，接受大量参数，调用docker环境运行程序，比较输入输出

func Compiler(sid uint32, codePath string, code []byte, compileInfo \*JudgerConfig.CompileInfo) ([]byte, error)

将需要编译的代码编译为可执行文件，复制到运行docker中执行

## 测试计划和测试用例

### 4.3.1 backend

后端针对评测模块编写了一系列的单元测试，具体测试内容可以在scheduler\_test.go中查看  
单元测试是标准的项目测试模型，通过不同的返回值来判断函数的运作是否正确

### 4.3.1 frontend

前端针对每一个页面模块进行数据调用测试，数据展示测试，数据切换测试；根据后端提供的接口进行不同样例的测试。并根据请求体和响应体来判断测试正确性。

## 4.4 结果分析

1. 注册数据结构的使用结果分析

* 用户注册测试一: 在注册界面，注册一个新用户，此时用户名为wadwings，密码为wadwings，邮箱为 [3313696160@qq.com](mailto:2313686160@qq.com)注册成功后，会自动跳转到登录页面。查看数据库，可以发现数据库中能够正常存储用户的信息。



图4-2

* 用户注册测试二: 当重复注册时，调用添加用户的接口函数，会显示用户名已注册。

1. 登录数据结构的使用结果分析

* 用户登录测试一： 在登录界面，用户点击用户名登录，输入已注册的用户名和密码。登录成功后，弹框显示登录成功 样式，前端获取token并进行存储，并跳转到主页。查看数据库，可以发现数据库中能够正常存储用户的信息。如下图： 

图4-3

* 用户登录测试二： 在登录界面，用户点击用邮箱登录，输入已注册的邮箱账号和密码。登录成功后，弹框显示登录成功样式，前端获取token并进行存储，并跳转到主页。查看数据库，可以发现数据库中能够正常存储用户的信息。如下图： 

图4-4

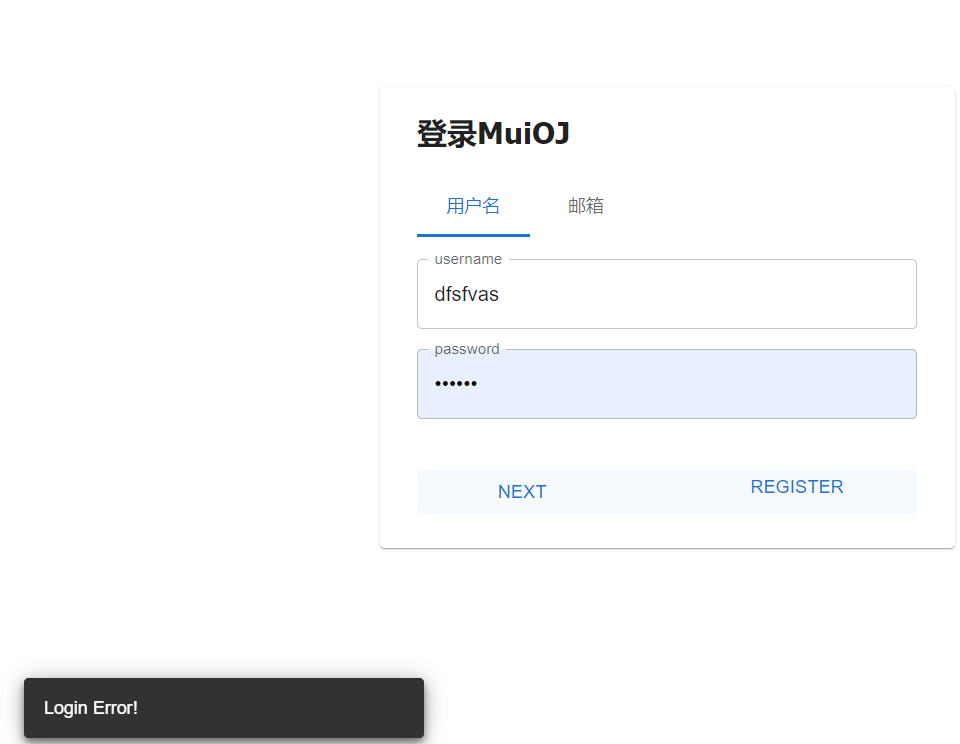
* 用户登录测试三： 在登录界面，用户输入未注册的账号或者邮箱，系统会提示登录失败。如下图： 

图4-5

* 返回的数据如下： 

图4-6

1. 题库数据结构的使用结果分析

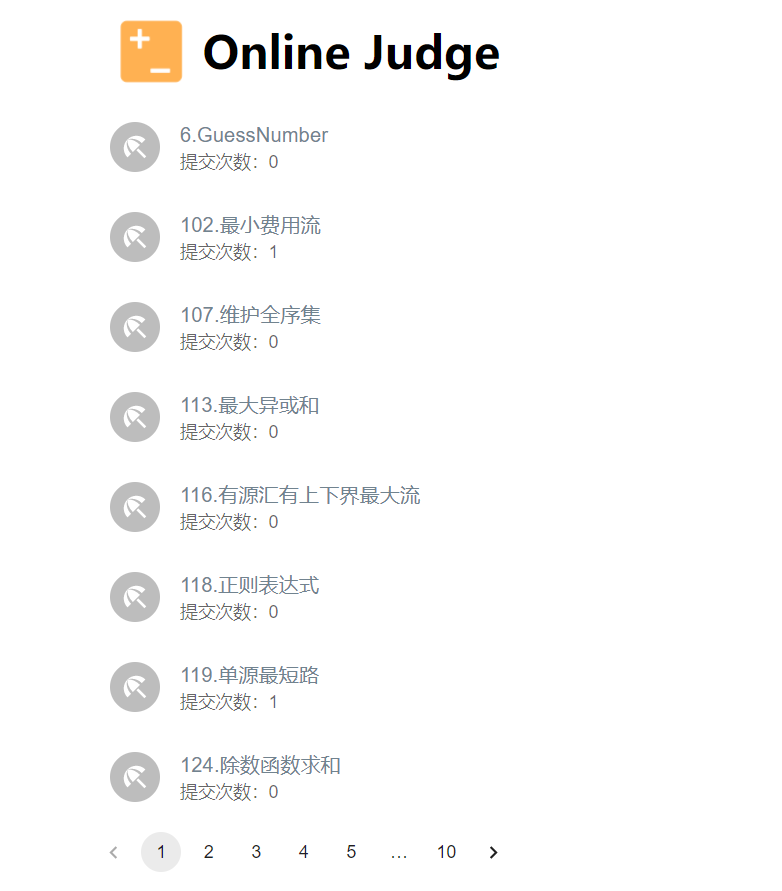
* 点击题库，跳转题库页面，同时向后端发送携带tid的url的get请求，获取到所有题目数据并渲染到页面上。如下图： 

图4-7

* 获取到的数据如图：



图4-8

1. 代码编辑页数据结构的使用结果分析

* 用户随意点击一题，此时便会根据题目ID向后端发送携带tid的url的get请求，获取题目的详细信息，并跳转到代码编辑页面展示题目与代码编辑框，如下图： 

图4-9

* 点击提交后，便将代码数据打包向后端发送post请求，后端收到数据后进行处理。如下图：
* 获取到的数据如图： 

图4-10

1. 用户信息展示数据结构的使用结果分析

* 点击顶部用户名即可跳转到用户界面，同时向后端发送携带uid的url的get请求，获取到数据后及时渲染到页面上。如下图： 

图4-11

* 获取到的数据如图：

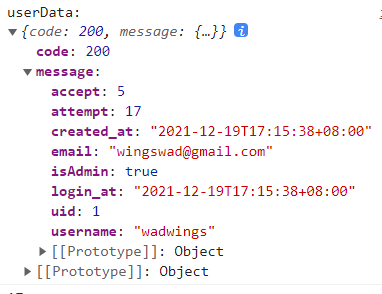


图4-12

1. 提交记录数据结构的使用和结果分析：

* 点击提交记录查看数据:

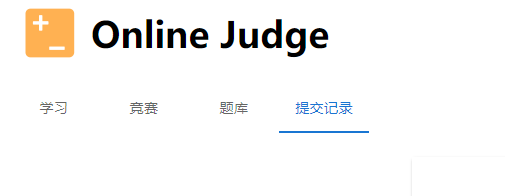


图4-13

* 获取到的数据如图：

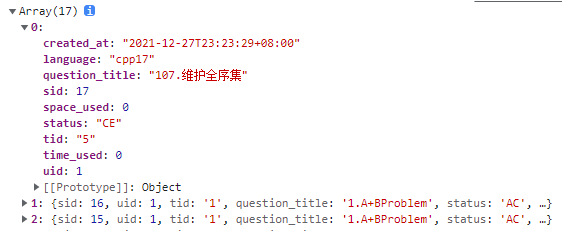
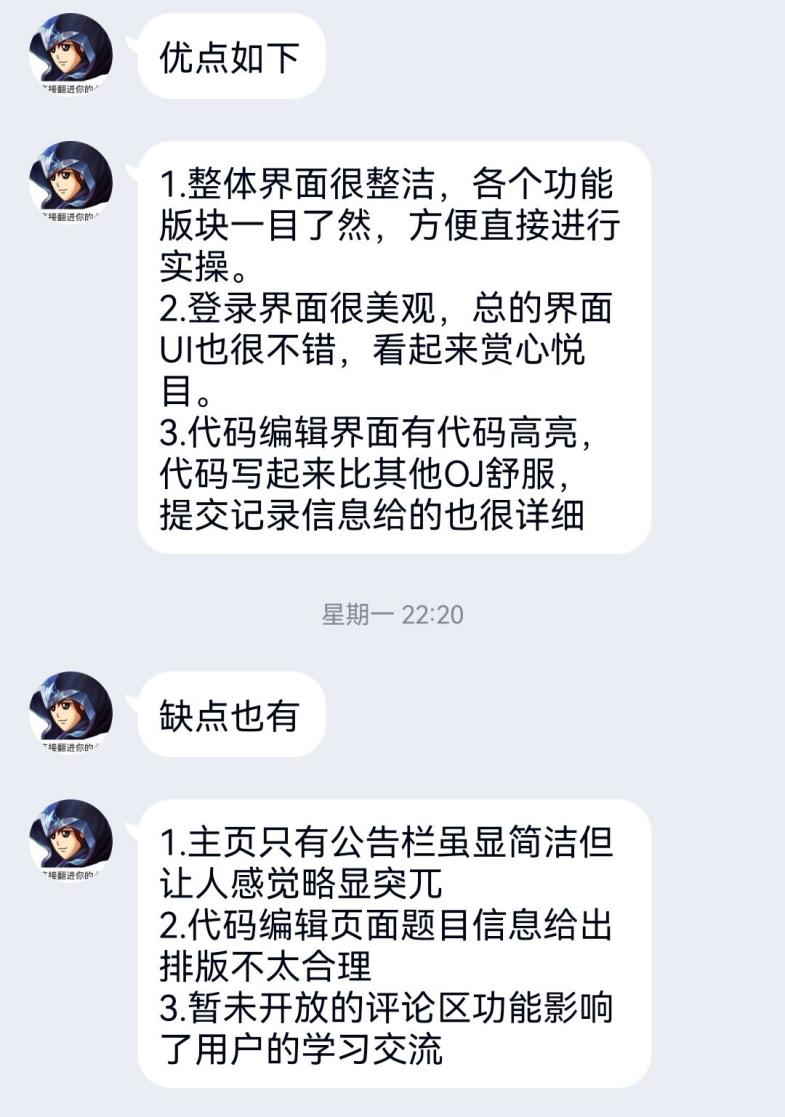


图4-14

# 5 总结

## 5.1 用户反馈

### 5.1.1用户反馈样例



### 5.1.2用户反馈总结

（一）优点

1.OJ系统简洁明了

2.登录注册功能做了响应式设计，可以适应不同的屏幕大小，用户体验感较好

题目数据丰富，可供用户有多重选择

3.monaco代码编辑区有颜色高亮和提错显示，相较于其他编辑器使用起来更人性化

4.提交记录展示较为详细，可以看到自己的提交结果，了解自己对哪种题比较熟悉或者不熟悉，方便随时练手

(二）用户建议

1.OJ主页面内容可以更多样化一些

2.编辑页面的布局可以稍微注意下各个组件的间隔问题；

3.顶部导航栏居中可能会比靠左对齐更好一点

### 全文总结

## 5.2 全文总结

### 5.2.1 谭滨瀚工作总结

总体来说，我对于自己在这个项目中所做的努力是比较满意的。

因为前期的筹备周期比较长，导致最后编写代码的时间比较短，因为编写Online Judge后端的样例并不多，所以相当于是一边在尝试拓展一边又要赶工期，最后能够得到一个bugFree的后端也算是不错的结果。

本次项目中我所做的主要工作：

1. 完成了整个OnlineJudge后端的代码架构设计和实现  
   包括11个不同规模的模块，以及未来的一些发展期望
2. 搭建了前端的整套脚手架环境，解决一些依赖冲突和bug，提供一些建议
3. 完成了整个项目的线上部署<https://muioj.vercel.app/>
4. 完成了对于后端的docker镜像化的操作，手动编写了dockerfile和dockercompose文件，提供了容器化的解决方案，便于重新部署时的依赖解决问题，在后端的docker容器化时使用到了多部构建，大幅减少了最后得到的产物容器的大小
5. 完成了整个前端系统的自动化ci/cd的部署，代码推送到main分支即可自动构建并重新部署

### 5.2.2施远栋工作总结

在这次的工作中，虽然个人基础知识比较薄弱，团队合作经验少但我为已经为团队尽可能的贡献了我所有的力量，所以总的来说我勉强认可自己在本次工作中的努力。

本次项目中我所作的主要工作为：

1.是用python解析清洗xml文件为后端提供题库数据。

2.设计了主页和提交记录两个前端页面。

3.尝试学习了后端知识，为协助组员完成前后端工作尽了一点绵薄之力。

### 5.2.3许楚辉工作总结

本项目中我主要完成的工作：  
1.提供网页界面设计模板  
2.对题库xml文件进行数据清洗  
3.协助组员使用react编写前端

### 5.2.4王艳工作总结

总体来说，我在这个项目中学到了许多。由于使用的是当前较新较快的脚手架vite，并且使用react作为前端搭建的一个语言，但是在react方面自己仍旧是一个初学者，所以在写项目的过程中有过非常痛苦的感觉。但是边查询文档边写边解决bug这种经历，却是让我以很快的速度收获了很多东西。

本次项目中我所做的主要工作：

1. 完成了OnlineJudge前端的总体代码框架搭建
2. 对前端任务进行一些简单的分工
3. 实现了总体顶部跳转导航栏、题库展示页面、代码编辑页面、用户个人信息展示页面以及复用修改登录注册页面；
4. 与后端进行数据的联调，获取数据，测试数据，并渲染数据。

# 6 体会

谭滨瀚：

分工：见工作总结

体会：感觉最大的困难就在于人员之间的协调吧，相对于自己写项目的自由，要求跟其他同学一起合作的话就涉及到很多协同之间的事情。虽然我个人身在技术团队之中，对于这种事情已经是家常便饭了，但是大部分同学仍然是还没有接触过这方面的一个情况。我们作为同学也应该要好好的去引导他们，让他们以正确的方式学习，而不是通过看一些老旧的博客或者效率极低的视频来学习。我原来一直钻研的方向其实是前端，这次因为分工问题我也必须要接触后端这些工程。对于我来说也是自己的一次尝试，因为之前也只写过一些后端的玩具，对于这种需要使用在实战中的一些项目来说我也是第一次。这次能够按时按质完工也算是一种成功

施远栋：

分工：见工作总结

体会：工作总结中提到，个人知识贮备比较薄弱，因此完成这项任务对我来说是一项非常具有挑战性的工作。本次项目中我最主要的矛盾便是我强烈想协助组员不想摆烂的主观意愿和极其薄弱的知识基础和差劲的代码能力间的矛盾。因此本此项目中“学”便成了我个人的主题。进行数据清洗时因为搞错文件解析方式导致苦苦修改无果，最后在组员的帮助下解决。前端页面的设计上，在组员的建议下引入mui组件通过不断学习艰难的完成了任务。总的来说本次任务的完成除了学以外，更离不开小组成员的互相帮助，互相鼓励。大家虽然能力有差别，但任然能工作中做到每个人都尽自己最大的努力，相处之间只有欢声笑语不存在相互埋怨。整个过程中，我不止学到了新知识，提高了代码能力，团队合作能力。更重要的是收获了一群可爱的组员，相信这段共事的经历将会成为我们弥足珍贵的一段回忆。

许楚辉：  
分工：见工作总结  
体会：作为团队里面业务能力最差的成员，没能给团队带来太多的帮助，本次项目多亏了各位组员的倾力投入与对我的无限支持和帮助！前期我主要做了一些题库数据清洗和格式转化的工作，虽然技术上并没有太大的难度，但是由于我不熟悉xml格式和即学即用的beautifulsoup的熟练度，过程也是有些一波三折，还好最后在同学的帮助下顺利完成了工作。后期由于我在前后端方面都是小白，所以无法直接参与具体的细节工作。通过同学分享的文档，我开始自学react，也是第一次看英文文档学习，虽然开头很困难，进度很慢，但是总归是有收获的，我认为这是一个好的开端。最后还是要感谢我的组员们，大家一直在欢声笑语中共同进退，虽然路途坎坷但是仍有属于每一个人的收获！

王艳：

分工：见工作总结

体会：相对于个人项目来说，组队项目其实反而能够让人收获更多的东西。从一开始的分工到开始进入工作，从开始的没有头绪到能够写出页面，这些收获都是实实在在的。由于第一次使用vite脚手架，从一窍不通到感受到了它的好处，轻便快速，打包速度飞快，让人使用感极好。之前自己对于react页只是算是初学者，而这是第一次使用react开始写真正的项目，最开始的过程真的非常痛苦，就连最基本的数据传输过程也需要查文档，上网搜，用起来不流畅而且磕磕碰碰，但是渐渐的在不停的边查边写边调bug的过程中，真的以很快的速度收获了很多，也可以很成功得写出页面，这也会带来一定的成就感。在使用manaco编辑器的时候，由于网上的版本非常多，第一次使加载的时候一直不能高亮显示代码并自动补全和提错，后来找了很多种方法，在组员的帮助下成功解决这个问题。其实小组合作，组内分工和组内协调工作都是很重要的，所以遇到一群好的组员非常重要，大家遇到问题一起商量，一起解决的过程，也是大家一起学习一起进步的过程。最后，这次项目即将开启我对react的更深的探索之路，我会在未来的学习生活里在不断的使用过程中不断地启发自己，以获得更多知识！

# 附录

仓库地址------------------------https://gitee.com/wadwings/muioj.git

1.authRequest.tsx：

*import* registerType *from* "@src/constants/register/registerType";

*import* conf *from* "@config/conf";

*import* React *from* 'react'

*import* Snackbar *from* '@mui/material/Snackbar';

*import* {navigate} *from* '@layouts/main'

const RequestWithAuth = async (

  url: string,

  opts: RequestInit = {},

*//   jsonData: Record<string, unknown>*

): Promise<string> => {

*//initialize fetch options*

  opts.mode = "cors"

  opts.headers= Object.assign (opts.headers ? opts.headers : {}, {

    "Authorization": window.sessionStorage.getItem("token") || ''

  })

*if* (window.sessionStorage.getItem("token") == undefined) {

    console.log("未登录需重新登录")

    navigate('/login');

  }

*try* {

    const data = *await* fetch(`${conf.baseUrl}${url}`, opts).then((result) =>

      result.json()

    )

    console.log("request body:", opts)

    console.log("success response:",data)

*return* data

  } *catch* ({ response }) {

    console.log("request body(error):", opts)

    console.log("request url(error):", `${conf.baseUrl}${url}`)

*throw* Error("send http request failed")

  }

};

*export* *default* RequestWithAuth

2.rest.ts：

*import* loginType *from* '@constants/login/loginType'

*import* conf *from* "@config/conf";

*import* login *from* '@styles/login/login';

interface RestForm {

  [key: string]: string

}

const Endpoint = {

  emailLogin: "/user/login?type=email",

  usernameLogin: "/user/login?type=username",

};

let loginData = {

  code: 0,

  token:'',

  message: '',

  username: '',

  isAdmin: false,

  uid:0

}

const wrapLoginRequest = async (

  url: string,

  jsonData: Record<string, unknown>

): Promise<string> => {

*//initialize fetch options*

  const opts: RequestInit = {

    method: "POST",

    mode: "cors",

    headers: {

      "Content-Type": "application/json",

    },

    body: JSON.stringify(jsonData),

  };

*try* {

      const data = *await* fetch(`${conf.baseUrl}${url}`, opts).then((result) =>

      result.json()

    )

    console.log("resposeData：",data)

    console.log("request body:", opts.body)

    loginData.code = data.code

    loginData.username = data.message.username

    loginData.token = data.message.token

    loginData.uid = data.message.uid

    loginData.isAdmin = data.message.is\_admin

*return* data

  } *catch* ({ response }) {

    console.log(response);

*throw* Error("send http request failed");

  }

};

*export* *default* {

  [loginType.Email]: async (form: RestForm): Promise<string> => {

    console.log("Body form", form)

*return* wrapLoginRequest(Endpoint.emailLogin, {

      email: form.email,

      password: form.password,

    });

  },

  [loginType.Username]: async (form: RestForm): Promise<string> => {

    console.log("Body form", form)

*return* wrapLoginRequest(Endpoint.usernameLogin, {

      username: form.username,

      password: form.password,

    });

  },

};

*export* const loginMessage = loginData

3. register.tsx：

*import* registerType *from* "@src/constants/register/registerType";

*import* conf *from* "@config/conf";

*import* React *from* 'react'

*import* Snackbar *from* '@mui/material/Snackbar';

interface RestForm {

  [key: string]: string

}

const Endpoint = {

  register: "/user/register",

};

let registerData = {

    code:0,

    message:''

}

const wrapLoginRequest = async (

  url: string,

  jsonData: Record<string, unknown>

): Promise<string> => {

*//initialize fetch options*

  const opts: RequestInit = {

    method: "POST",

    mode: "cors",

    headers: {

      "Content-Type": "application/json",

    },

    body: JSON.stringify(jsonData),

  };

*try* {

    const data = *await* fetch(`${conf.baseUrl}${url}`, opts).then((result) =>

      result.json()

    )

    registerData.code = data.code;

    registerData.message = data.message

    console.log("resposeData：",registerData)

    console.log("request body:", opts.body)

    window.sessionStorage.setItem("registerData:",JSON.stringify(data) )

*return* data

  } *catch* ({ response }) {

    console.log(response);

*throw* Error("send http request failed");

  }

};

*export* *default* {

  [registerType.Register]: async (form: RestForm): Promise<string> => {

*//console.log("Body form", form)*

*return* wrapLoginRequest(Endpoint.register, {

        username: form.username,

        email: form.email,

        password: form.password,

    });

  },

};

*export* const registerMessage = registerData

4. allProblems.tsx：

*import* \* *as* React *from* 'react';

*import* List *from* '@mui/material/List';

*import* ListItem *from* '@mui/material/ListItem';

*import* ListItemText *from* '@mui/material/ListItemText';

*import* ListItemAvatar *from* '@mui/material/ListItemAvatar';

*import* Avatar *from* '@mui/material/Avatar';

*import* ImageIcon *from* '@mui/icons-material/Image';

*import* WorkIcon *from* '@mui/icons-material/Work';

*import* BeachAccessIcon *from* '@mui/icons-material/BeachAccess';

*export* *default* function FolderList() {

*return* (

    <List sx={{ width: '100%', maxWidth: 360, bgcolor: 'background.paper' }}>

      <ListItem>

        <ListItemAvatar>

          <Avatar>

            <ImageIcon />

          </Avatar>

        </ListItemAvatar>

        <ListItemText primary="Photos" secondary="Jan 9, 2014" />

      </ListItem>

      <ListItem>

        <ListItemAvatar>

          <Avatar>

            <WorkIcon />

          </Avatar>

        </ListItemAvatar>

        <ListItemText primary="Work" secondary="Jan 7, 2014" />

      </ListItem>

      <ListItem>

        <ListItemAvatar>

          <Avatar>

            <BeachAccessIcon />

          </Avatar>

        </ListItemAvatar>

        <ListItemText primary="Vacation" secondary="July 20, 2014" />

      </ListItem>

    </List>

  );

}

5. board.tsx

*import* \* *as* React *from* 'react';

*import* { styled } *from* '@mui/material/styles';

*import* Table *from* '@mui/material/Table';

*import* TableBody *from* '@mui/material/TableBody';

*import* TableCell, { tableCellClasses } *from* '@mui/material/TableCell';

*import* TableContainer *from* '@mui/material/TableContainer';

*import* TableHead *from* '@mui/material/TableHead';

*import* TableRow *from* '@mui/material/TableRow';

*import* Paper *from* '@mui/material/Paper';

*import* './board.css'

const StyledTableCell = styled(TableCell)(({ theme }) => ({

  [`&.${tableCellClasses.head}`]: {

    backgroundColor: '#a6c1ee',

    color: theme.palette.common.white,

  },

  [`&.${tableCellClasses.body}`]: {

    fontSize: 14,

  },

}));

const StyledTableRow = styled(TableRow)(({ theme }) => ({

  '&:nth-of-type(odd)': {

    backgroundColor: theme.palette.action.hover,

  },

*// hide last border*

  '&:last-child td, &:last-child th': {

    border: 0,

  },

}));

function createData(

  Notice: string,

  Way: string,

  Uploader: string,

  Date: string,

) {

*return* { Notice, Way, Uploader, Date };

}

const rows = [

  createData('hust-oj coming', 'by', 'hust-group', '2021年12月18日'),

  createData('', '', '', '' ),

  createData('', '', '', ''),

  createData('', '', '', ''),

  createData('', '', '', ''),

];

*export* *default* function CustomizedTables() {

*return* (

    <Flex>

    <TableContainer component={Paper} className="table-container">

      <Table aria-label="customized table">

        <TableHead className="tableHead">

        <H3>公告栏</H3>

          <TableRow>

            <StyledTableCell>Notice</StyledTableCell>

            <StyledTableCell align="right"></StyledTableCell>

            <StyledTableCell align="right">Uploader</StyledTableCell>

            <StyledTableCell align="right">Date</StyledTableCell>

          </TableRow>

        </TableHead>

        <TableBody>

          {rows.map((row) => (

            <StyledTableRow key={row.Notice}>

              <StyledTableCell component="th" scope="row"><a href="http://www.w3school.com.cn">

                {row.Notice}

              </a>

              </StyledTableCell>

              <StyledTableCell align="right">{row.Way}</StyledTableCell>

              <StyledTableCell align="right">{row.Uploader}</StyledTableCell>

              <StyledTableCell align="right">{row.Date}</StyledTableCell>

            </StyledTableRow>

          ))}

        </TableBody>

      </Table>

    </TableContainer>

    </Flex>

  );

}

const H3 = styled('h3')`

  padding-left: 3rem;

`

const Flex = styled('div')`

  width: 100%;

  display: flex;

  align-items: center;

  justify-content: center;

`

6. board.css：

a{

  text-decoration: none;

}

a:link{color:#a6c1ee;text-decoration:none}

a:hover{color:#8fd3f4;text-decoration:underline}

.table-container {

  max-width: 70vw;

  margin-top: 20px;

}

7.container.css：

.container {

    display: flex;

    -ms-flex-align: center;

    width: 80%;

    height: 100%;

}

8.container.tsx：

*import* React, { Component } *from* 'react';

*import* './container.css'

*export* *default* function Container() {

*return* (

        <>

        <div className='container'>

        </div>

        </>

    )

}

9. config.tsx：

*import* { editor } *from* 'monaco-editor'

*import* *type* {Monaco} *from* '@monaco-editor/react'

const theme : editor.IStandaloneThemeData = {

  "base": 'vs-dark',

  "inherit": true,

  "rules": [

    {

      "background": "011627",

      "token": ""

    },

    {

      "foreground": "637777",

      "token": "comment"

    },

    {

      "foreground": "addb67",

      "token": "string"

    },

    {

      "foreground": "ecc48d",

      "token": "vstring.quoted"

    },

    {

      "foreground": "ecc48d",

      "token": "variable.other.readwrite.js"

    },

    {

      "foreground": "5ca7e4",

      "token": "string.regexp"

    },

    {

      "foreground": "5ca7e4",

      "token": "string.regexp keyword.other"

    },

    {

      "foreground": "5f7e97",

      "token": "meta.function punctuation.separator.comma"

    },

    {

      "foreground": "f78c6c",

      "token": "constant.numeric"

    },

    {

      "foreground": "f78c6c",

      "token": "constant.character.numeric"

    },

    {

      "foreground": "addb67",

      "token": "variable"

    },

    {

      "foreground": "c792ea",

      "token": "keyword"

    },

    {

      "foreground": "c792ea",

      "token": "punctuation.accessor"

    },

    {

      "foreground": "c792ea",

      "token": "storage"

    },

    {

      "foreground": "c792ea",

      "token": "meta.var.expr"

    },

    {

      "foreground": "c792ea",

      "token": "meta.class meta.method.declaration meta.var.expr storage.type.jsm"

    },

    {

      "foreground": "c792ea",

      "token": "storage.type.property.js"

    },

    {

      "foreground": "c792ea",

      "token": "storage.type.property.ts"

    },

    {

      "foreground": "c792ea",

      "token": "storage.type.property.tsx"

    },

    {

      "foreground": "82aaff",

      "token": "storage.type"

    },

    {

      "foreground": "ffcb8b",

      "token": "entity.name.class"

    },

    {

      "foreground": "ffcb8b",

      "token": "meta.class entity.name.type.class"

    },

    {

      "foreground": "addb67",

      "token": "entity.other.inherited-class"

    },

    {

      "foreground": "82aaff",

      "token": "entity.name.function"

    },

    {

      "foreground": "addb67",

      "token": "punctuation.definition.variable"

    },

    {

      "foreground": "d3423e",

      "token": "punctuation.section.embedded"

    },

    {

      "foreground": "d6deeb",

      "token": "punctuation.terminator.expression"

    },

    {

      "foreground": "d6deeb",

      "token": "punctuation.definition.arguments"

    },

    {

      "foreground": "d6deeb",

      "token": "punctuation.definition.array"

    },

    {

      "foreground": "d6deeb",

      "token": "punctuation.section.array"

    },

    {

      "foreground": "d6deeb",

      "token": "meta.array"

    },

    {

      "foreground": "d9f5dd",

      "token": "punctuation.definition.list.begin"

    },

    {

      "foreground": "d9f5dd",

      "token": "punctuation.definition.list.end"

    },

    {

      "foreground": "d9f5dd",

      "token": "punctuation.separator.arguments"

    },

    {

      "foreground": "d9f5dd",

      "token": "punctuation.definition.list"

    },

    {

      "foreground": "d3423e",

      "token": "string.template meta.template.expression"

    },

    {

      "foreground": "d6deeb",

      "token": "string.template punctuation.definition.string"

    },

    {

      "foreground": "c792ea",

      "fontStyle": "italic",

      "token": "italic"

    },

    {

      "foreground": "addb67",

      "fontStyle": "bold",

      "token": "bold"

    },

    {

      "foreground": "82aaff",

      "token": "constant.language"

    },

    {

      "foreground": "82aaff",

      "token": "punctuation.definition.constant"

    },

    {

      "foreground": "82aaff",

      "token": "variable.other.constant"

    },

    {

      "foreground": "7fdbca",

      "token": "support.function.construct"

    },

    {

      "foreground": "7fdbca",

      "token": "keyword.other.new"

    },

    {

      "foreground": "82aaff",

      "token": "constant.character"

    },

    {

      "foreground": "82aaff",

      "token": "constant.other"

    },

    {

      "foreground": "f78c6c",

      "token": "constant.character.escape"

    },

    {

      "foreground": "addb67",

      "token": "entity.other.inherited-class"

    },

    {

      "foreground": "d7dbe0",

      "token": "variable.parameter"

    },

    {

      "foreground": "7fdbca",

      "token": "entity.name.tag"

    },

    {

      "foreground": "cc2996",

      "token": "punctuation.definition.tag.html"

    },

    {

      "foreground": "cc2996",

      "token": "punctuation.definition.tag.begin"

    },

    {

      "foreground": "cc2996",

      "token": "punctuation.definition.tag.end"

    },

    {

      "foreground": "addb67",

      "token": "entity.other.attribute-name"

    },

    {

      "foreground": "addb67",

      "token": "entity.name.tag.custom"

    },

    {

      "foreground": "82aaff",

      "token": "support.function"

    },

    {

      "foreground": "82aaff",

      "token": "support.constant"

    },

    {

      "foreground": "7fdbca",

      "token": "upport.constant.meta.property-value"

    },

    {

      "foreground": "addb67",

      "token": "support.type"

    },

    {

      "foreground": "addb67",

      "token": "support.class"

    },

    {

      "foreground": "addb67",

      "token": "support.variable.dom"

    },

    {

      "foreground": "7fdbca",

      "token": "support.constant"

    },

    {

      "foreground": "7fdbca",

      "token": "keyword.other.special-method"

    },

    {

      "foreground": "7fdbca",

      "token": "keyword.other.new"

    },

    {

      "foreground": "7fdbca",

      "token": "keyword.other.debugger"

    },

    {

      "foreground": "7fdbca",

      "token": "keyword.control"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.operator.comparison"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.control.flow.js"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.control.flow.ts"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.control.flow.tsx"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.control.ruby"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.control.module.ruby"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.control.class.ruby"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.control.def.ruby"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.control.loop.js"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.control.loop.ts"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.control.import.js"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.control.import.ts"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.control.import.tsx"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.control.from.js"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.control.from.ts"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.control.from.tsx"

    },

    {

      "foreground": "ffffff",

      "background": "ff2c83",

      "token": "invalid"

    },

    {

      "foreground": "ffffff",

      "background": "d3423e",

      "token": "invalid.deprecated"

    },

    {

      "foreground": "7fdbca",

      "token": "keyword.operator"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.operator.relational"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.operator.assignement"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.operator.arithmetic"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.operator.bitwise"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.operator.increment"

    },

    {

      "foreground": "c792ea",

      "token": "keyword.operator.ternary"

    },

    {

      "foreground": "637777",

      "token": "comment.line.double-slash"

    },

    {

      "foreground": "cdebf7",

      "token": "object"

    },

    {

      "foreground": "ff5874",

      "token": "constant.language.null"

    },

    {

      "foreground": "d6deeb",

      "token": "meta.brace"

    },

    {

      "foreground": "c792ea",

      "token": "meta.delimiter.period"

    },

    {

      "foreground": "d9f5dd",

      "token": "punctuation.definition.string"

    },

    {

      "foreground": "ff5874",

      "token": "constant.language.boolean"

    },

    {

      "foreground": "ffffff",

      "token": "object.comma"

    },

    {

      "foreground": "7fdbca",

      "token": "variable.parameter.function"

    },

    {

      "foreground": "80cbc4",

      "token": "support.type.vendor.property-name"

    },

    {

      "foreground": "80cbc4",

      "token": "support.constant.vendor.property-value"

    },

    {

      "foreground": "80cbc4",

      "token": "support.type.property-name"

    },

    {

      "foreground": "80cbc4",

      "token": "meta.property-list entity.name.tag"

    },

    {

      "foreground": "57eaf1",

      "token": "meta.property-list entity.name.tag.reference"

    },

    {

      "foreground": "f78c6c",

      "token": "constant.other.color.rgb-value punctuation.definition.constant"

    },

    {

      "foreground": "ffeb95",

      "token": "constant.other.color"

    },

    {

      "foreground": "ffeb95",

      "token": "keyword.other.unit"

    },

    {

      "foreground": "c792ea",

      "token": "meta.selector"

    },

    {

      "foreground": "fad430",

      "token": "entity.other.attribute-name.id"

    },

    {

      "foreground": "80cbc4",

      "token": "meta.property-name"

    },

    {

      "foreground": "c792ea",

      "token": "entity.name.tag.doctype"

    },

    {

      "foreground": "c792ea",

      "token": "meta.tag.sgml.doctype"

    },

    {

      "foreground": "d9f5dd",

      "token": "punctuation.definition.parameters"

    },

    {

      "foreground": "ecc48d",

      "token": "string.quoted"

    },

    {

      "foreground": "ecc48d",

      "token": "string.quoted.double"

    },

    {

      "foreground": "ecc48d",

      "token": "string.quoted.single"

    },

    {

      "foreground": "addb67",

      "token": "support.constant.math"

    },

    {

      "foreground": "addb67",

      "token": "support.type.property-name.json"

    },

    {

      "foreground": "addb67",

      "token": "support.constant.json"

    },

    {

      "foreground": "c789d6",

      "token": "meta.structure.dictionary.value.json string.quoted.double"

    },

    {

      "foreground": "80cbc4",

      "token": "string.quoted.double.json punctuation.definition.string.json"

    },

    {

      "foreground": "ff5874",

      "token": "meta.structure.dictionary.json meta.structure.dictionary.value constant.language"

    },

    {

      "foreground": "d6deeb",

      "token": "variable.other.ruby"

    },

    {

      "foreground": "ecc48d",

      "token": "entity.name.type.class.ruby"

    },

    {

      "foreground": "ecc48d",

      "token": "keyword.control.class.ruby"

    },

    {

      "foreground": "ecc48d",

      "token": "meta.class.ruby"

    },

    {

      "foreground": "7fdbca",

      "token": "constant.language.symbol.hashkey.ruby"

    },

    {

      "foreground": "e0eddd",

      "background": "a57706",

      "fontStyle": "italic",

      "token": "meta.diff"

    },

    {

      "foreground": "e0eddd",

      "background": "a57706",

      "fontStyle": "italic",

      "token": "meta.diff.header"

    },

    {

      "foreground": "ef535090",

      "fontStyle": "italic",

      "token": "markup.deleted"

    },

    {

      "foreground": "a2bffc",

      "fontStyle": "italic",

      "token": "markup.changed"

    },

    {

      "foreground": "a2bffc",

      "fontStyle": "italic",

      "token": "meta.diff.header.git"

    },

    {

      "foreground": "a2bffc",

      "fontStyle": "italic",

      "token": "meta.diff.header.from-file"

    },

    {

      "foreground": "a2bffc",

      "fontStyle": "italic",

      "token": "meta.diff.header.to-file"

    },

    {

      "foreground": "219186",

      "background": "eae3ca",

      "token": "markup.inserted"

    },

    {

      "foreground": "d3201f",

      "token": "other.package.exclude"

    },

    {

      "foreground": "d3201f",

      "token": "other.remove"

    },

    {

      "foreground": "269186",

      "token": "other.add"

    },

    {

      "foreground": "ff5874",

      "token": "constant.language.python"

    },

    {

      "foreground": "82aaff",

      "token": "variable.parameter.function.python"

    },

    {

      "foreground": "82aaff",

      "token": "meta.function-call.arguments.python"

    },

    {

      "foreground": "b2ccd6",

      "token": "meta.function-call.python"

    },

    {

      "foreground": "b2ccd6",

      "token": "meta.function-call.generic.python"

    },

    {

      "foreground": "d6deeb",

      "token": "punctuation.python"

    },

    {

      "foreground": "addb67",

      "token": "entity.name.function.decorator.python"

    },

    {

      "foreground": "8eace3",

      "token": "source.python variable.language.special"

    },

    {

      "foreground": "82b1ff",

      "token": "markup.heading.markdown"

    },

    {

      "foreground": "c792ea",

      "fontStyle": "italic",

      "token": "markup.italic.markdown"

    },

    {

      "foreground": "addb67",

      "fontStyle": "bold",

      "token": "markup.bold.markdown"

    },

    {

      "foreground": "697098",

      "token": "markup.quote.markdown"

    },

    {

      "foreground": "80cbc4",

      "token": "markup.inline.raw.markdown"

    },

    {

      "foreground": "ff869a",

      "token": "markup.underline.link.markdown"

    },

    {

      "foreground": "ff869a",

      "token": "markup.underline.link.image.markdown"

    },

    {

      "foreground": "d6deeb",

      "token": "string.other.link.title.markdown"

    },

    {

      "foreground": "d6deeb",

      "token": "string.other.link.description.markdown"

    },

    {

      "foreground": "82b1ff",

      "token": "punctuation.definition.string.markdown"

    },

    {

      "foreground": "82b1ff",

      "token": "punctuation.definition.string.begin.markdown"

    },

    {

      "foreground": "82b1ff",

      "token": "punctuation.definition.string.end.markdown"

    },

    {

      "foreground": "82b1ff",

      "token": "meta.link.inline.markdown punctuation.definition.string"

    },

    {

      "foreground": "7fdbca",

      "token": "punctuation.definition.metadata.markdown"

    },

    {

      "foreground": "82b1ff",

      "token": "beginning.punctuation.definition.list.markdown"

    }

  ],

  "colors": {

    "editor.foreground": "#d6deeb",

    "editor.background": "#011627",

    "editor.selectionBackground": "#5f7e9779",

    "editor.lineHighlightBackground": "#010E17",

    "editorCursor.foreground": "#80a4c2",

    "editorWhitespace.foreground": "#2e2040",

    "editorIndentGuide.background": "#5e81ce52",

    "editor.selectionHighlightBorder": "#122d42"

  }

}

const useTheme = (monaco: Monaco) => {

  monaco.editor.defineTheme("NightOwl", theme)

  console.log("test")

  monaco.editor.setTheme("NightOwl")

}

*export* *default* useTheme

10. editor.d.ts：

*import* { FunctionComponent } *from* 'react'

declare const Header: FunctionComponent

*export* *default* Header

11. editor.module.css：

*/\* .wrapper {*

*display: flex;*

*flex-direction: column;*

*align-items: center;*

*width: 80%;*

*margin-top: 20rem;*

*margin-left: 10rem;*

*} \*/*

.editor-container {

    width: 80%;

    height: 100%;

    margin-left: 20%;

    background-color: rgb(190, 147, 89);

}

.monacoEditor {

    margin-left: 10rem;

}

.breakpoints {

    background: red;

    width: 10px !important;

    height: 10px !important;

    left: 0px !important;

    top: 3px;

    border-radius: 5px;

}

.breakpoints-fake {

    background: red;

    width: 10px !important;

    height: 10px !important;

    left: 0px !important;

    top: 3px;

    border-radius: 5px;

}

12. editor.tsx：

*import* React, { Component, Ref, useRef } *from* 'react';

*//import logo from './logo.svg';*

*//import './App.css';*

*import* { useEffect } *from* 'react'

*import* Editor, { loader } *from* "@monaco-editor/react";

*import* *type* { Monaco } *from* '@monaco-editor/react';

*import* useTheme *from* './config'

*import* {FC} *from* 'react'

*import* styles *from* './editor.module.css'

*import* styled *from* '@emotion/styled'

const defaultCode =

  `export default {

  name: 'name',

  code: 'code'

}`;

interface EditorProps {

  forwardRef: any

}

const MonacoEditor : FC<EditorProps> = ({forwardRef}) => {

  const handleEditorDidMount = (editor: any, monaco : Monaco) => {

    useTheme(monaco)

    forwardRef.current = editor

*// code = value*

    console.log("editor:", editor.getValue())

  }

*return* (

    <EditorLayout>

    <h3>

      因为最近jsDeliver的证书问题,国内访问Editor的Js可能会存在一些问题,因此如果下面的Editor打不开,请尝试使用代理打开

    </h3>

    <Editor

            height="50vh"

            defaultLanguage="cpp"

            defaultValue=""

            onMount={handleEditorDidMount}>

    </Editor>

    </EditorLayout>

  )

}

const EditorLayout = styled.div`

  width: 100%;

`

*export* *default* MonacoEditor

*// render(*

*//   <Editor />,*

*//   document.getElementById('root')*

*// );*

13. login.tsx：

*import* React, { FunctionComponent, useState } *from* "react";

*import* color *from* "@styles/color/color";

*import* { Global, css } *from* "@emotion/react";

*import* { Button, Card, Tab, Tabs, TextField } *from* "@mui/material";

*import* styled *from* "@emotion/styled";

*import* LoginType *from* "@constants/login/loginType";

*import* useStyles *from* "@styles/login/login"

*import* loginText *from* "@constants/login/loginText";

*import* loginMethod *from* "@apis/login/rest";

*import* { loginMessage } *from* "@apis/login/rest"

*import* Snackbar *from* '@mui/material/Snackbar';

*import* { createBrowserHistory } *from* "history";

*import* { Link, useNavigate } *from* 'react-router-dom'

*//路由有问题，不能在点击时就跳转，需要刷新一次在跳转*

const isWidthLimited = document.body.offsetWidth > 400;

const Login: FunctionComponent<unknown> = () => {

*return* (

    <Main id="Container">

      <Global

        styles={css`

          body {

            margin: 0;

            font-family: Gilroy-Medium, Tahoma, PingFangSC-Regular,

              Microsoft Yahei, Myriad Pro, Hiragino Sans GB, sans-serif;

          }

        `}

      />

      <LoginForm />

    </Main>

  );

};

const Main = styled.div`

  height: ${window.innerHeight}px;

  background-image: ${color.backgroundColor};

  position: relative;

  display: flex;

  align-items: center;

  justify-content: center;

`;

const LoginForm: FunctionComponent<unknown> = () => {

  const [loginType, setLoginType] = useState(LoginType.Email);

  const navigate = useNavigate();

*//弹框*

  const [open,setOpen] = React.useState(false)

  const [snackbarMessage,setSnackbarMessage] = React.useState('')

  const tabIndex = {

    [LoginType.Email]: 1,

    [LoginType.Username]: 0,

  };

  const classes = useStyles(isWidthLimited)

  const loginFieldText = loginText[loginType]

  let history = createBrowserHistory();

  const handleClose = (event: React.SyntheticEvent | Event, reason?: string) => {

*if* (reason === 'clickaway') {

*return*;

    }

    setOpen(false);

};

  const loginForm = loginFieldText.map((item) => {

    console.log(item);

*return* (

      <LoginInputBlock key={item}>

        <LoginInputArea

          id={item}

          label={item}

          type={item === 'password' ? 'password' : ''}

          variant={"outlined"}

        ></LoginInputArea>

        {*/\* 弹框组件 \*/*}

        <Snackbar

          open={open}

          autoHideDuration={1500}

          onClose={handleClose}

          message={snackbarMessage}

        />

      </LoginInputBlock>

    );

  });

  const login = () => {

    const opts: { [key: string]: string } = {};

*for* (const id of loginFieldText) {

      const ele = document.querySelector(`#${id}`) *as* HTMLInputElement;

      const value = ele?.value;

      opts[id] = value;

    }

    loginMethod[loginType](opts).then((loginMessage:any) => {

      console.log("loginMessage:", loginMessage)

*//处理登录token及弹框数据*

*if* (loginMessage.code === 200) {

        window.sessionStorage.setItem('token', loginMessage.message.token.toString())

*//有问题，不能及时渲染*

        setSnackbarMessage("Login Succesfully!")

        setOpen(true)

        window.sessionStorage.setItem("loginData",JSON.stringify(Object.assign(loginMessage.message, {"code": loginMessage.code})))

        navigate('/')

      }

*else* {

        setOpen(true)

        setSnackbarMessage("Login Error!")

      }

    });

  };

*return* (

    <FormLayout>

      <H2>登录MuiOJ</H2>

      {*/\* <Logo src={logo} /> \*/*}

      <TabList>

        <Tabs className={classes.tabs} value={tabIndex[loginType]}>

          <Tab

            label="用户名"

            onClick={() => setLoginType(LoginType.Username)}

            className={classes.tabs}

          />

          <Tab

            label="邮箱"

            onClick={() => setLoginType(LoginType.Email)}

            className={classes.tabs}

          />

          {*/\* <Tab*

*label="Lark认证"*

*onClick={() => setLoginType(LoginType.LarkOauth)}*

*className={classes.default}*

*/> \*/*}

        </Tabs>

      </TabList>

      {loginForm}

      <LoginSubmitButton

        onClick={() => login()}

        className={classes.root}

        variant="text"

        disableFocusRipple

      >

        Next

      </LoginSubmitButton>

      <RegisterButton><Link to={'/register'} style={{textDecoration: "none", color:"#1976d2"}}>Register</Link></RegisterButton>

      <LoginSwitch></LoginSwitch>

    </FormLayout>

  );

};

*export* const loginData = loginMessage

const TabList = styled.div`

  margin-bottom: 10px;

`

const H2 = styled.h2`

  font-weight: 600;

  font-size: ${isWidthLimited ? "auto" : "4vmin"};

`;

const FormLayout = styled(Card)`

  padding: 0 30px 30px 30px;

  width: 400px;

  display: flex;

  flex-direction: column;

  justify-content: center;

  height: auto;

  transition: height 0.5s;

`;

const LoginInputArea = styled(TextField)`

  margin: 0.5rem 0;

  max-width: 400px;

  position: relative;

`;

const LoginSubmitButton = styled(Button)`

  margin: 2rem 0 1rem;

  padding-right: 15rem;

  background-color: "#1e68f9"

`;

const RegisterButton = styled(Button)`

 margin-left: 13rem;

 margin-top: -3.5rem;

`

const LoginSwitch = styled.div`

  display: flex;

  align-items: center;

  justify-content: center;

  width: 100%;

  position: relative;

  max-width: 300px;

`;

const LoginInputBlock = styled.div`

  position: relative;

  display: flex;

  flex-direction: column;

  justify-content: center;

`

export default Login;

14. problemDisplay.css

.problemContainer {

    width: 80%;

    margin-top: 2rem;

    margin-left: -2rem;

}

15. problemDisplay.jsx：

*import* './problemDisplay.css'

*import* React, { Component } *from* 'react'

*import* Tab *from* '@mui/material/Tab';

*import* styled *from* "@emotion/styled";

*import* List *from* '@mui/material/List';

*import* ListItem *from* '@mui/material/ListItem';

*import* ListItemText *from* '@mui/material/ListItemText';

*import* ListItemAvatar *from* '@mui/material/ListItemAvatar';

*import* Avatar *from* '@mui/material/Avatar';

*import* ImageIcon *from* '@mui/icons-material/Image';

*import* WorkIcon *from* '@mui/icons-material/Work';

*import* BeachAccessIcon *from* '@mui/icons-material/BeachAccess';

*import* { Link } *from* 'react-router-dom';

*import* Typography *from* '@mui/material/Typography';

*import* Pagination *from* '@mui/material/Pagination';

*import* Stack *from* '@mui/material/Stack';

*import* GetProblem *from* '@apis/common/authRequest'

*export* *default* class ProblemDisplay extends Component {

  constructor(props) {

    super(props);

    this.state = {

      page: 1,

      problemList: [],

      problemUrl: "/question/list/",

    }

  }

  styles = { textDecoration: "none", color: "#71838f" }

  handleChange = (event, value) => {

    const res = GetProblem(this.state.problemUrl + value, {})

    res.then(res => {

      this.setState({ problemList: res.message, page: value })

      console.log("this.problemList", this.state.problemList)

    })

  };

  componentDidMount() {

    const res = GetProblem(this.state.problemUrl + this.state.page, {})

    res.then(res => {

      this.setState({ problemList: res.message })

      console.log("this.problemList", this.state.problemList)

    })

  }

  render() {

*return* (

      <div className='problemContainer'>

        <List sx={{ width: '100%', maxWidth: 360, bgcolor: 'background.paper' }}>

          {

*//maps循环*

            this.state.problemList.map(item => {

*return* (

                <ListItem key={item.tid}>

                  <ListItemAvatar>

                    <Avatar>

                      <BeachAccessIcon />

                    </Avatar>

                  </ListItemAvatar>

                  <Link to={'/edit/' + item.tid} style={this.styles}>

                    <ListItemText primary={item.title} secondary={"提交次数：" + item.attempt} />

                  </Link>

                </ListItem>

              )

            })

          }

        </List>

        <Stack spacing={2}>

          {*/\* <Typography>Page: {this.state.page}</Typography> \*/*}

          <Pagination count={10} page={this.state.page} onChange={this.handleChange} />

        </Stack>

      </div>

    );

  }

}

16. problemItem.jsx：

*import* { FC, useEffect, useState, FunctionComponent, useRef } *from* 'react'

*import* request *from* '@apis/common/authRequest'

*import* styles *from* './problemItem.module.css'

*import* Editor *from* '@components/editor/editor'

*import* Button *from* '@mui/material/Button';

*import* Snackbar *from* '@mui/material/Snackbar';

*import* AppBar *from* '@mui/material/AppBar';

*import* Toolbar *from* '@mui/material/Toolbar';

*import* Typography *from* '@mui/material/Typography';

*import* IconButton *from* '@mui/material/IconButton';

*import* MenuIcon *from* '@mui/icons-material/Menu';

*import* {useNavigate} *from* 'react-router-dom'

*import* { ColorLens } *from* '@mui/icons-material';

const ProblemItem = (props) => {

  const [open, setOpen] = useState(false)

  const [snackbarMessage, setSnackbarMessage] = useState('')

  const navigate = useNavigate();

  const [data, setData] = useState(

    {

      content: '',

      accept: 0,

      create\_at: '',

      sampleIn: '',

      sampleOut: ''

    })

  const editorRef = useRef(null);

  const urlParams = new URL(window.location.href);

  const tid = urlParams?.pathname.substring(6);

  console.log("当前页面路由：", tid)

*//const tid = props.match.params.id*

  useEffect(() => {

    request('/question/item/' + tid, {}).then(res => {

*if* (res.code === 200) {

        console.log("getQuestionItem:", res)

        let getData = {

          content: res.message.content,

          accept: res.message.accept,

          create\_at: res.message.create\_at,

          sampleIn: res.message.sample[0].in,

          sampleOut: res.message.sample[0].out

        }

        setData(getData)

      }

    })

  }, [])

*//提交代码*

  const handleClick = () => {

    const code = editorRef.current.getValue()

    console.log("code", code)

    const urlParams = new URL(window.location.href);

    const tid = urlParams?.pathname.substring(6);

    request('/question/item/' + tid, {

      method: "POST",

      headers: {

        "Content-Type": "application/json",

      },

      body: JSON.stringify({

        language: "cpp17",

        code: code

      }),

    }).then(res => {

*if* (res.code === 200) {

        setOpen(true);

        setSnackbarMessage("提交成功！")

        console.log("submit success!")

        setTimeout(() => {

          navigate("/")

        }, 1000)

      }

*else* {

        console.log("submit error!")

      }

    })

  }

  const handleClose = (event, reason) => {

*if* (reason === 'clickaway') {

*return*;

    }

    setOpen(false);

  }

  console.log("ProblemItemData", data)

*// const goBack = () => {*

*//   console.log("goback");*

*//   goback();*

*// }*

*return* (<div className={styles.container}>

    <AppBar position="static" color="transparent">

      <Toolbar variant="dense">

        <IconButton onClick={()=>{navigate('/')}}edge="start" color="default" aria-label="menu" sx={{ mr: 2 }}>

          <MenuIcon />

        </IconButton>

        <Typography  variant="h6" color="default" component="div">

          Back

        </Typography>

      </Toolbar>

    </AppBar>

    <Snackbar

      open={open}

      autoHideDuration={1500}

      onClose={handleClose}

      message={snackbarMessage}

    />

    <div className={styles.title}>

      &ensp;{data.content}

    </div>

    <div className={styles.content}>

      <Editor forwardRef={editorRef}></Editor>

    </div>

    <Button onClick={handleClick} className={styles.Button} variant="outlined" color="success">

      Submit

    </Button>

  </div>

  )

}

*export* *default* ProblemItem

17. problemItem.module.css

.container {

    position: absolute;

    display: flex;

    flex-direction: column;

    margin-left: 10%;

    width: 80%;

    height: 100%;

}

.title {

    display: flex;

    position: relative;

    width: 100%;

    right: 0;

    border-radius: 5px;

    font-size: 1.3rem;

    color: #71838f;

    font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva, Verdana, sans-serif;

}

.content {

    display: flex;

    position: relative;

    width: 100%;

    min-height: 55%;

    align-items: center;

}

.Button {

}

18. register.tsx：

*import* React, { FunctionComponent, useState } *from* "react";

*import* color *from* "@styles/color/color";

*import* { Global, css } *from* "@emotion/react";

*import* { Button, Card, TextField, IconButton } *from* "@mui/material";

*import* styled *from* "@emotion/styled";

*import* RegisterType *from* "@constants/register/registerType";

*import* useStyles *from* "@styles/login/login"

*import* registerText *from* "@constants/register/registerText";

*import* RegisterMethod *from* "@apis/register/register"

*import* { registerMessage } *from* "@apis/register/register"

*import* Snackbar *from* '@mui/material/Snackbar';

*import* { createBrowserHistory } *from* "history";

*import* {useNavigate} *from* 'react-router-dom'

*//import CloseIcon from '@mui/icons-material/Close';*

const isWidthLimited = document.body.offsetWidth > 400;

const Register: FunctionComponent<unknown> = () => {

*return* (

        <Main id="Container">

            <Global

                styles={css`

          body {

            margin: 0;

            font-family: Gilroy-Medium, Tahoma, PingFangSC-Regular,

              Microsoft Yahei, Myriad Pro, Hiragino Sans GB, sans-serif;

          }

        `}

            />

            <RegisterForm />

        </Main>

    );

};

const Main = styled.div`

  height: ${window.innerHeight}px;

  background-image: ${color.backgroundColor};

  position: relative;

  display: flex;

  align-items: center;

  justify-content: center;

`;

const RegisterForm: FunctionComponent<unknown> = () => {

    const registerType = RegisterType.Register;

*//弹框*

    const [open, setOpen] = React.useState(false);

    const [snackbarMessage,setSnackbarMessage] = React.useState("")

    const classes = useStyles(isWidthLimited);

    const registerFieldText = registerText[registerType]

    let history = createBrowserHistory();

    const navigate = useNavigate()

    const handleClose = (event: React.SyntheticEvent | Event, reason?: string) => {

*if* (reason === 'clickaway') {

*return*;

        }

        setOpen(false);

        console.log("openfalse:", open)

    };

    const RegisterForm = registerFieldText.map((item) => {

*return* (

            <RegisterInputBlock key={item}>

                <RegisterInputArea

                    id={item}

                    label={item}

                    variant={"outlined"}

                    type={item === 'password' ? "password" : ""}

                ></RegisterInputArea>

                <Snackbar

                    open={open}

                    autoHideDuration={1500}

                    onClose={handleClose}

                    message={snackbarMessage}

                />

            </RegisterInputBlock>

        );

    });

    const register = () => {

        const opts: { [key: string]: string } = {};

*for* (const id of registerFieldText) {

            const ele = document.querySelector(`#${id}`) *as* HTMLInputElement;

            const value = ele?.value;

            opts[id] = value;

        }

*//console.log("registerMessage",registerMessage);*

*//console.log("registerMethod[registerType](opts):",opts)*

        RegisterMethod[registerType](opts).then((res: any) => {

*if* (res.code === 200) {

                setSnackbarMessage("Register Success!")

                setOpen(true)

                navigate("/login");

            }*else*{

                setSnackbarMessage("Register Success!")

                setOpen(true)

                navigate("/login");

            }

        })

    };

*return* (

        <FormLayout>

            <H2>注册MuiOJ</H2>

            {*/\* <Logo src={logo} /> \*/*}

            {RegisterForm}

            <RegisterSubmitButton

                onClick={() => register()}

                className={classes.root}

                variant="text"

                disableFocusRipple

            >

                Next

            </RegisterSubmitButton>

            <RegisterSwitch></RegisterSwitch>

        </FormLayout>

    );

};

const H2 = styled.h2`

  font-weight: 600;

  font-size: ${isWidthLimited ? "auto" : "4vmin"};

`;

const FormLayout = styled(Card)`

  padding: 30px 30px 10px 30px;

  width: 400px;

  display: flex;

  flex-direction: column;

  justify-content: center;

  height: auto;

  transition: height 0.5s;

`;

const RegisterInputArea = styled(TextField)`

  margin: 0.5rem 0;

  max-width: 400px;

  position: relative;

`;

const RegisterSubmitButton = styled(Button)`

  margin: 1rem 0 1rem;

`;

const RegisterSwitch = styled.div`

  display: flex;

  align-items: center;

  justify-content: center;

  width: 100%;

  position: relative;

  max-width: 300px;

`;

const RegisterInputBlock = styled.div`

  position: relative;

  display: flex;

  flex-direction: column;

  justify-content: center;

`

*export* *default* Register;

19. submit.tsx：

*import* React, { Component } *from* 'react'

*import* request *from* '@apis/common/authRequest'

*import* { FC, useEffect, useState } *from* 'react'

*import* Paper *from* '@mui/material/Paper'

*import* Table *from* '@mui/material/Table'

*import* TableBody *from* '@mui/material/TableBody'

*import* TableCell *from* '@mui/material/TableCell'

*import* TableContainer *from* '@mui/material/TableContainer'

*import* TableHead *from* '@mui/material/TableHead'

*import* TablePagination *from* '@mui/material/TablePagination'

*import* TableRow *from* '@mui/material/TableRow'

*import* Submitinfo *from* '@apis/common/authRequest'

*import* styled *from* '@emotion/styled'

interface Column {

  id: 'name' | 'status' | 'mark' | 'time' | 'memory' | 'language' | 'date'

  label: string

  minWidth?: number

  align?: 'center'

  format?: (value: number) => string

}

const columns: Column[] = [

  { id: 'name', label: 'ID', minWidth: 100, align: 'center' },

  { id: 'status', label: 'Status', minWidth: 100, align: 'center' },

  {

    id: 'time',

    label: 'UsingTime\u00a0(ms)',

    minWidth: 100,

    align: 'center',

    format: (value: number) => value.toLocaleString('en-US'),

  },

  {

    id: 'memory',

    label: 'Memory\u00a0(kb)',

    minWidth: 130,

    align: 'center',

    format: (value: number) => value.toLocaleString('en-US'),

  },

  { id: 'language', label: 'Language', minWidth: 130, align: 'center' },

  {

    id: 'date',

    label: 'Submit date',

    minWidth: 130,

    align: 'center',

    format: (value: number) => value.toFixed(2),

  },

]

interface Data {

  name: string

  status: string

  time: number

  memory: number

  language: string

  date: Date

}

function createData(

  name: string,

  status: string,

  time: number,

  memory: number,

  language: string,

  date: Date

): Data {

*return* { name, status, time, memory, language, date }

}

*export* *default* function ColumnGroupingTable() {

  const [page, setPage] = React.useState(0)

  const [rowsPerPage, setRowsPerPage] = React.useState(10)

  const [pageData, setPageData] = React.useState<any>({})

  useEffect(() =>{

    let data: any = window.sessionStorage.getItem('loginData')

*if* (data) {

      data = JSON.parse(data)

      const uid = data?.uid

*if* (uid) {

        request('/submission/' + uid).then((data: any) => {

          console.log(data)

          console.log(data.message)

          setPageData(data?.message || [])

        })

      }

    }

  }, [])

  console.log(pageData)

  let rows: any

*if* (pageData.map) {

    rows = pageData.map((item: any) =>

      createData(

        item.question\_title,

        item.status,

        item.time\_used,

        item.space\_used,

        item.language,

        item.created\_at

      )

    )

  } *else* {

    rows = []

  }

  const handleChangePage = (event: unknown, newPage: number) => {

    setPage(newPage)

  }

  const handleChangeRowsPerPage = (

    event: React.ChangeEvent<HTMLInputElement>

  ) => {

    setRowsPerPage(+event.target.value)

    setPage(0)

  }

*return* (

  <Layout>

    <Paper sx={{ maxWidth: '80%' }}>

      <TableContainer>

        <Table stickyHeader aria-label="sticky table">

          <TableHead>

            <TableRow>

              <TableCell align="center" colSpan={3}>

                Result

              </TableCell>

              <TableCell align="center" colSpan={3}>

                Details

              </TableCell>

            </TableRow>

            <TableRow>

              {columns.map((column) => (

                <TableCell

                  key={column.id}

                  align={column.align}

                  style={{ top: 57, minWidth: column.minWidth }}

                >

                  {column.label}

                </TableCell>

              ))}

            </TableRow>

          </TableHead>

          <TableBody>

            {rows

              .slice(page \* rowsPerPage, page \* rowsPerPage + rowsPerPage)

              .map((row: any, index: number) => {

*return* (

                  <TableRow

                    hover

                    role="checkbox"

                    tabIndex={-1}

                    key={pageData[index].sid}

                  >

                    {columns.map((column) => {

                      const value = row[column.id]

*return* (

                        <TableCell key={column.id} align={column.align}>

                          {column.format && typeof value === 'number'

                            ? column.format(value)

                            : value}

                        </TableCell>

                      )

                    })}

                  </TableRow>

                )

              })}

          </TableBody>

        </Table>

      </TableContainer>

      <TablePagination

        rowsPerPageOptions={[10, 25, 100]}

        component="div"

        count={rows.length}

        rowsPerPage={rowsPerPage}

        page={page}

        onPageChange={handleChangePage}

        onRowsPerPageChange={handleChangeRowsPerPage}

      />

    </Paper>

    </Layout>

  )

}

const Layout = styled.div`

  width: 100%;

  display: flex;

  align-items: center;

  justify-content: center;

`

20. user.css

.container {

    position: absolute;

    display: flex;

    flex-direction: row;

    margin-left: 5%;

    width: 90%;

}

.main {

    display: flex;

    flex-direction: column;

    position: absolute;

    text-align: center;

    width: 60%;

    margin-left: 30%;

    margin-top: 10rem;

    margin-bottom: 5rem;

    border-radius: 5px;

    background-color: rgba(0,0,0,0.1);

}

.side {

    display: flex;

    flex-direction: column;

    position: absolute;

    width: 30%;

    margin-top: 10%;

    margin-right: 2rem;

    padding-top: 1rem;

    align-items: center;

*/\* background-color: rgba(0,0,0,0.3); \*/*

}

21. user.tsx：

*import* React, { FunctionComponent, useEffect, useState } *from* "react";

*import* styled *from* "@emotion/styled";

*import* './user.css'

*import* Box *from* '@mui/material/Box';

*import* Card *from* '@mui/material/Card';

*import* CardActions *from* '@mui/material/CardActions';

*import* CardContent *from* '@mui/material/CardContent';

*import* Button *from* '@mui/material/Button';

*import* Typography *from* '@mui/material/Typography';

*import* { Link , useNavigate, Navigate} *from* 'react-router-dom'

*import* Table *from* '@mui/material/Table';

*import* TableBody *from* '@mui/material/TableBody';

*import* TableCell *from* '@mui/material/TableCell';

*import* TableContainer *from* '@mui/material/TableContainer';

*import* TableHead *from* '@mui/material/TableHead';

*import* TableRow *from* '@mui/material/TableRow';

*import* Paper *from* '@mui/material/Paper';

*import* Avatar *from* '@mui/material/Avatar';

*import* Stack *from* '@mui/material/Stack';

*import* request *from* '@apis/common/authRequest'

*import* avatorImage *from* '@images/avatar.jpg'

const bull = (

    <Box

        component="span"

        sx={{ display: 'inline-block', mx: '2px', transform: 'scale(0.8)' }}

    >

        •

    </Box>

);

function createData(

    name: string,

    data: string,

) {

*return* { name, data };

}

interface userForm {

    [key: string]: string

  }

const user : userForm = window.sessionStorage.getItem('token') == undefined ? "" : JSON.parse(window.sessionStorage.getItem('loginData') || '');

let  submitNum = 0

const rows = [

    createData('用户名', user?.username),

    createData('用户ID', user?.uid?.toString()),

    createData('用户邮箱', '3313696160@qq.com'),

    createData('是否管理员', user?.is\_admin?.toString() === 'true' && '是' || "否")

];

const Main: FunctionComponent<unknown> = () => {

    const [noUse, setNoUse] = useState(false)

    useEffect(() =>{

    let data: any = window.sessionStorage.getItem('loginData')

*if* (data) {

      data = JSON.parse(data)

      const uid = data?.uid

*if* (uid) {

        request('/user/my').then((data: any) => {

          console.log("userData:",data)

          console.log(data.message.attempt)

          rows[4] = createData('提交次数', data.message.attempt.toString())

          rows[5] = createData('通过次数', data.message.accept.toString())

          setNoUse(!noUse)

        })

      }

    }

  }, [])

*return* (

        <div>

            <TableContainer component={Paper}>

                <Table sx={{ minWidth: 500 }} aria-label="simple table">

                    <TableHead>

                        <TableRow>

                            <TableCell></TableCell>

                        </TableRow>

                    </TableHead>

                    <TableBody>

                        {rows.map((row) => (

                            <TableRow key = {row.name}>

                                <TableCell component="th" scope="row">

                                    {row.name}

                                </TableCell>

                                <TableCell align="right">{row.data}</TableCell>

                            </TableRow>

                        ))}

                    </TableBody>

                </Table>

            </TableContainer>

        </div>

    );

};

const styles = { textDecoration: "none", color: "#71838f" }

const Side: FunctionComponent<unknown> = () => {

    const navigate = useNavigate()

*return* (

        <>

            <Card sx={{ minWidth: 275, marginBottom: "1rem" }}>

                <CardActions>

                    <Button onClick={()=>{navigate('/')}}size="large" >

                        首页

                    </Button>

                </CardActions>

            </Card>

            <Card sx={{ minWidth: 275, marginBottom: "1rem", alignItems: "center" }}>

                <CardContent>

                    <Typography variant="h5" component="div">

                        be{bull}a{bull}good{bull}day

                    </Typography>

                </CardContent>

                <CardActions>

                    <Button size="large">

                        <Link to='./' style={styles}>题库</Link>

                    </Button>

                </CardActions>

            </Card>

            <Card sx={{ minWidth: 275, marginBottom: "1rem" }}>

                <CardContent>

                    <Typography variant="h5" component="div">

                        be{bull}a{bull}good{bull}day

                    </Typography>

                </CardContent>

                <CardActions>

                    <Button size="large">

                        <Link to='./' style={styles}>讨论</Link>

                    </Button>

                </CardActions>

            </Card>

            <Card sx={{ minWidth: 275, marginBottom: "1rem" }}>

                <CardContent>

                    <Typography variant="h5" component="div">

                        be{bull}a{bull}good{bull}day

                    </Typography>

                </CardContent>

                <CardActions>

                    <Button size="large">

                        <Link to='./' style={styles}>提交记录</Link>

                    </Button>

                </CardActions>

            </Card>

        </>

    );

};

const User: FunctionComponent<unknown> = () => {

    const problemList = []

*return* (

        <div className="container">

            <div className="side">

                <Side></Side>

            </div>

            <div className="main">

                <Stack direction="row" spacing={10}>

                    <Avatar sx={{ width: 100,

                                  height: 100,

                                  marginLeft: "45%",

                                  marginBottom: 2,

                                  marginTop: 2 }}

                    alt="Cindy Baker" src={avatorImage} />

                </Stack>

                <Main></Main>

            </div>

        </div>

    )

}

*export* *default* User

22. InfoType.js：

const InfoType = {

  info: Symbol("info"),

  success: Symbol("success"),

  warning: Symbol("warning"),

  error: Symbol("error"),

}

*export* *default* InfoType

23. loginText.ts：

*import* loginType  *from* "./loginType";

*export* *default* {

    [loginType.Email]:['email','password'],

    [loginType.Username]:['username','password']

}

24. loginType.ts：

*export* *default* {

    Email: Symbol("email"),

    Username: Symbol("username")

}

25. registerText.ts：

*import* RegisterType  *from* "./registerType";

*export* *default* {

    [RegisterType.Register]:['username','email','password']

}

26. registerType.ts：

*export* *default* {

    Register: Symbol("register"),

}

27. content.tsx

*import* PropTypes *from* 'prop-types';

*import* Tabs *from* '@mui/material/Tabs';

*import* Tab *from* '@mui/material/Tab';

*import* Typography *from* '@mui/material/Typography';

*import* Box *from* '@mui/material/Box';

*import* ProblemDisplay *from* '@src/components/problemDisplay/problemDisplay';

*import* Board *from* '@components/board/board'

*import* Submit *from* '@src/components/submission/submit'

*import* {useState} *from* 'react'

*import* styled *from* '@emotion/styled'

function TabPanel(props: any) {

  const { children, value, index, ...other } = props;

*return* (

    <div

      role="tabpanel"

      hidden={value !== index}

      id={`simple-tabpanel-${index}`}

      aria-labelledby={`simple-tab-${index}`}

      {...other}

    >

      {value === index && (

        <Box sx={{ p: 3 }}>

          <Typography>{children}</Typography>

        </Box>

      )}

    </div>

  );

}

TabPanel.propTypes = {

  children: PropTypes.node,

  index: PropTypes.number.isRequired,

  value: PropTypes.number.isRequired,

};

function a11yProps(index:number) {

*return* {

    id: `simple-tab-${index}`,

    'aria-controls': `simple-tabpanel-${index}`,

  };

}

*export* *default* function BasicTabs() {

  const [value, setValue] = useState(0);

  const handleChange = (event: any, newValue: any) => {

    setValue(newValue)

  };

*return* (

    <Box sx={{ width: '100%' }}>

      <Box sx={{ borderBottom: 0, borderColor: 'divider' }}>

        <Tabs value={value} onChange={handleChange} aria-label="basic tabs example">

          <Tab label="学习" {...a11yProps(0)} />

          <Tab label="竞赛" {...a11yProps(1)} />

          <Tab label="题库" {...a11yProps(2)} />

          <Tab label="提交记录" {...a11yProps(3)} />

        </Tabs>

      </Box>

      <TabPanel value={value} index={0}>

        <Board></Board>

      </TabPanel>

      <TabPanel value={value} index={1}>

        竞赛

      </TabPanel>

      <TabPanel value={value} index={2}>

          <ProblemDisplay ></ProblemDisplay>

      </TabPanel>

      <TabPanel value={value} index={3}>

        <Submit></Submit>

      </TabPanel>

    </Box>

  );

}

const StyledTabs = styled(Tabs)`

  display: flex;

  align-items: center;

  justify-content: center;

`

28. main.tsx：

*import* {FC} *from* 'react'

*import* styled *from* '@emotion/styled'

*import* Header *from* '@views/header'

*import* Content *from* '@layouts/content'

*import* {useNavigate} *from* 'react-router-dom'

let navigate : any

const Layout: FC= () => {

  navigate = useNavigate()

*return* <>

    <HeaderLayout>

      <Header></Header>

    </HeaderLayout>

    <HeaderPlaceHolder />

    <Container>

      <Content/>

    </Container>

  </>

}

*export* {navigate}

const HeaderPlaceHolder = styled.div`

  height: 10vh;

  min-height: 80px;

`

const HeaderLayout = styled.div`

  height: 10vh;

  min-height: 80px;

  position: fixed;

  margin: 0 5rem;

  width: calc(100% - 10rem);

  background: white;

  overflow: hidden;

  z-index: 1;

`

const Container = styled.div`

  display: flex;

  margin: 0 5rem;

  flex-direction: column;

  alignItems: center;

`

*export* *default* Layout;

29. router.tsx

*import* React *from* 'react'

*import* { BrowserRouter, Route, Routes } *from* 'react-router-dom'

*//import { createHashHistory } from 'history'*

*import* ProblemDisplay *from* '@components/problemDisplay/problemDisplay';

*import* Login *from* '../components/login/login'

*import* Main *from* '@layouts/main'

*import* Register *from* '@components/register/register'

*import* User *from* '@components/user/user'

*import* ProblemItem *from* '@components/problemItem/problemItem'

class RouterConfig extends React.Component {

    render() {

*return* (

            <BrowserRouter>

                <Routes> {*/\*只匹配其中一个 \*/*}

                    <Route path='/' element={<Main/>}></Route>

                    <Route path='/login' element={<Login />} />

                    <Route path='/register' element={<Register />} />

                    <Route path='/user' element={<User />} />

                    <Route path='/edit/:tid' element={<ProblemItem />} />

                </Routes>

            </BrowserRouter>

        )

    }

}

*export* *default* RouterConfig

30. color.ts

*export* *default* {

    backgroundColor: "white",

    buttonColor: "#1e68f9",

    loginFontColor: "#3f8fe3",

    defaultSvgColor: "#7d848d",

  };

31. login.ts

*import* { makeStyles } *from* "@mui/styles";

*import* color *from* "@styles/color/color";

*export* *default* (isWidthLimited: boolean): any => makeStyles(() => ({

    root: {

        background: color.buttonColor,

        "&:hover": {

            background: color.buttonColor,

        },

        color: "white",

        border: 0,

        borderRadius: 3,

        height: 48,

        padding: "0 30px",

    },

    default: {

        color: color.defaultSvgColor,

        backgroundColor: "transparent",

        minHeight: "30px",

        padding: isWidthLimited ? "0 10px" : "0 5px",

        minWidth: isWidthLimited ? "50px" : "auto",

        marginRight: isWidthLimited ? "20px" : 0,

        fontSize: isWidthLimited ? "auto" : "4vmin",

    },

    tabs: {

        minHeight: "30px",

        color: color.defaultSvgColor,

        backgroundColor: "transparent",

        padding: isWidthLimited ? "0 10px" : "0 5px",

        minWidth: isWidthLimited ? "50px" : "auto",

        marginRight: isWidthLimited ? "20px" : 0,

        fontSize: isWidthLimited ? "auto" : "4vmin",

    },

    querySms: {

        "&:hover": {

            background: 'transparent',

        }

    }

}))

32. header.tsx：

*import* React, { Component, FC, useEffect, useState } *from* 'react';

*import* './header.css'

*// import Content fro@layouts/Content*

*import* { Link } *from* 'react-router-dom'

*import* { loginMessage } *from* '@apis/login/rest'

*import* styled *from* '@emotion/styled';

*import* {Global, css} *from* '@emotion/react'

*import* homeAvatar *from* '@images/homeAvatar.png'

const Header: FC<{}> = () => {

    const [isLogin, setIsLogin] = useState(false)

    const [username, setUsername] = useState("")

    useEffect(() => {

        let data : any = window.sessionStorage.getItem('loginData')

*if* (data !== null) {

            data = JSON.parse(data)

            console.log("LoginName:",typeof (data))

            setUsername(data.username)

            setIsLogin(true)

        }

    }, [])

    const logout = (e: never) => {

        window.sessionStorage.clear()

        setIsLogin(false)

    }

*return* (

        <HeaderOutside>

            <HeaderLayout>

                <HeaderBlock>

                    <img style={{ borderRadius: "5px", width: "50px", height: "50px" }} src={homeAvatar} className='titlePicture'></img>

                    <HeaderTitle>Online Judge</HeaderTitle>

                </HeaderBlock>

                <HeaderBlock>

                    <StyledLink to={'/user'} >{username}</StyledLink>

                    {isLogin ?  <StyledLink to={"/login"} onClick={logout} >退出</StyledLink> :

                                <StyledLink to={"/login"} >登录</StyledLink>}

                </HeaderBlock>

            </HeaderLayout>

        </HeaderOutside>

    )

}

const HeaderOutside = styled.div`

    width: 100%;

`

const HeaderLayout = styled.div`

    position: relative;

    display: flex;

    width: 100%;

    justify-content: space-between;

    align-items: center;

`

const HeaderBlock = styled.div`

    padding: 1rem;

    text-align: center;

    display: flex;

    align-items: center;

`

const HeaderTitle = styled.h3`

    font-size: 36px;

    margin: 0 0 0 1rem;

`

const ContentLayout = styled.div`

    position: relative;

    display: flex;

    width: 100%;

    justify-content: center;

    align-items: center;

`

const StyledLink = styled(Link)`

    margin: 0 0.5rem;

    :hover{

        text-decoration: none;

    }

`

*export* *default* Header;

32. main.tsx

*import* React *from* 'react'

*import* ReactDOM *from* 'react-dom'

*import* Router *from* '@router/router'

ReactDOM.render(

  <React.StrictMode>

    <Router />

  </React.StrictMode>,

  document.getElementById('root')

)

33.jwt.go

*package auth*

*import (*

*"MuiOJ-backEnd/models"*

*"MuiOJ-backEnd/services/config"*

*"github.com/dgrijalva/jwt-go"*

*"time"*

*)*

*type StandardClaims = jwt.StandardClaims*

*type Claims struct {*

*Uid uint32 `json:"uid,omitempty"`*

*Username string `json:"username,omitempty"`*

*IsAdmin bool `json:"is\_admin,omitempty"`*

*StandardClaims*

*}*

*func SignJWT(user \*models.User) (string, error) {*

*nextTime := time.Now()*

*nextTime = nextTime.AddDate(0,0,1)*

*jwtObject := &Claims{*

*Uid: user.Uid,*

*Username: user.Username,*

*IsAdmin: user.IsAdmin,*

*StandardClaims: StandardClaims{*

*ExpiresAt: nextTime.Unix(),*

*},*

*}*

*tokenObj := jwt.NewWithClaims(jwt.SigningMethodHS512, jwtObject)*

*return tokenObj.SignedString([]byte(config.Secret))*

*}*

*func VerifyJWT(token string) (\*Claims, error) {*

*tokenClaims, err := jwt.ParseWithClaims(token, &Claims{}, func(token \*jwt.Token) (interface{}, error) {*

*return []byte(config.Secret), nil*

*})*

*if tokenClaims != nil {*

*if claims, ok := tokenClaims.Claims.(\*Claims); ok && tokenClaims.Valid {*

*return claims, nil*

*}*

*}*

*return nil, err*

*}*

34.清洗.py

*import sys  
from bs4 import BeautifulSoup  
import sys  
import json  
from bs4 import BeautifulSoup  
  
#soup = BeautifulSoup(open(&apos;D:\华科网课\大三上\计网实验\实验一\index.html&apos;))  
#定义请求头和url  
# url = "C:\Users\syd\AppData\Roaming\Tencent\QQ\Temp\%W@GJ$ACOF(TYDYECOKVDYB.pnghttps://uoj.ac/problem/1"  
# headers = {  
#     "User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.132 Safari/537.36"  
# }  
#发送请求  
# response = requests.get(url,headers=headers)  
# content = response.content.decode(&apos;utf8&apos;)  
#实例化bs对象  
file=open(&apos;./fps-loj-small-pics.xml&apos;,&apos;r&apos;,encoding=&apos;utf-8&apos;)  
xml\_text=file.read()  
btree = BeautifulSoup(xml\_text,"html.parser")  
#print(btree.prettify())  
  
  
  
result=[]  
  
# item\_list=btree.find\_all(&apos;item&apos;)  
# item =  
# description = item.find(&apos;description&apos;)  
# p1\_list = description.find\_all(&apos;p&apos;)  
  
item\_list=btree.find\_all(&apos;item&apos;)  
count=0  
for item in item\_list:  
    abc = {}  
    #获得标题  
    title = item.find(&apos;title&apos;)  
    temp=title.string  
    temp=&apos;&apos;.join(temp.split(&apos;]&apos;))  
    temp = &apos;&apos;.join(temp.split(&apos;[&apos;))  
    temp = &apos;&apos;.join(temp.split(&apos;<&apos;))  
    temp = &apos;&apos;.join(temp.split(&apos;>&apos;))  
    temp = (temp.split(&apos;#&apos;,1))  
    temp = &apos;&apos;.join(temp[1].split())  
    #print(temp)  
    xx = {"title":temp}  
    abc.update(xx)  
    #获得题目描述及测试集  
    description = item.find(&apos;description&apos;)  
    input\_list = item.find\_all(&apos;test\_input&apos;)  
    output\_list = item.find\_all(&apos;test\_output&apos;)  
  
    #获得描述  
    p1\_list = description.find\_all(&apos;p&apos;)  
    tax=&apos;&apos;  
    j=0  
    while(j<len(p1\_list)):  
        span = p1\_list[j].find\_all("span",attrs={"class":"katex-mathml"})  
        for item2 in span:  
            item2.extract()  
        tax1=p1\_list[j].text  
        tax=tax+tax1  
        #print(p1\_list[j].text)  
        j=j+1  
    #获取测试集  
    xx = {"content": tax}  
    abc.update(xx)  
    code\_list=item.find\_all(&apos;code&apos;,class\_="lang-plain")  
    cnt=0  
    sample\_list = []  
    sample\_dict = {}  
    for code in code\_list:  
        if(code!=None and cnt%2==0):  
            # print("样例输入")  
            # print(code.text)  
            tt = {"in":code.text}  
            sample\_dict.update(tt)  
        else:  
            if(code!=None):  
                # print("样例输出")  
                # print(code.text)  
                tt = {"out":code.text}  
                sample\_dict.update(tt)  
                sample\_list.append(sample\_dict)  
                sample\_dict = {}  
        cnt = cnt + 1  
  
    abc["sample"] = sample\_list  
    i = 0  
    test\_list = []  
    while(i<len(input\_list) and i<len(output\_list)):  
        test\_dict = {}  
        # print("测试输入")  
        # print(input\_list[i].text)  
        # print("测试输出")  
        # print(output\_list[i].text)  
        # abc.update(xx)  
        # abc.update(yy)  
        test\_dict["in"] = input\_list[i].text  
        test\_dict["out"] = output\_list[i].text  
        test\_list.append(test\_dict)  
        i=i+1  
    abc["test"] = test\_list  
    result.append(abc)  
with open("jsonFile.json", "w", encoding=&apos;utf-8&apos;) as jsonFile:  
    jsonFile.write(json.dumps(result, ensure\_ascii=False, indent = 4))  
  
  
# titles = btree.find\_all(&apos;title&apos;)  
# with open("test.txt","w+") as f:  
#     sys.stdout = f  
#     for title in titles:  
#         temp=title.string  
#         temp=&apos;&apos;.join(temp.split(&apos;]&apos;))  
#         temp = &apos;&apos;.join(temp.split(&apos;[&apos;))  
#         temp = &apos;&apos;.join(temp.split(&apos;<&apos;))  
#         temp = &apos;&apos;.join(temp.split(&apos;>&apos;))  
#         temp = (temp.split(&apos;#&apos;,1))  
#         temp = &apos;&apos;.join(temp[1].split())  
#         print(temp)*

35. jwt\_object.go

*package auth*

*import (*

*"errors"*

*"github.com/gin-gonic/gin"*

*)*

*func GetAuthObj(c \*gin.Context) (\*Claims, error) {*

*authObjectRaw, exists := c.Get("AuthObject")*

*if !exists{*

*return nil, errors.New("access denied")*

*}*

*authObject, ok := authObjectRaw.(\*Claims)*

*if !ok {*

*return nil, errors.New("token format invalid")*

*}*

*return authObject, nil*

*}*

*func GetAuthObjWithAdmin(c \*gin.Context) (\*Claims, error) {*

*authObjectRaw, exists := c.Get("AuthObject")*

*if !exists{*

*return nil, errors.New("access denied")*

*}*

*authObject, ok := authObjectRaw.(\*Claims)*

*if !ok {*

*return nil, errors.New("token format invalid")*

*}*

*if authObject.IsAdmin != true{*

*return nil, errors.New("admin required")*

*}*

*return authObject, nil*

*}*

36. submit.go

package common

import (

  "MuiOJ-backEnd/controllers/auth"

  "MuiOJ-backEnd/models"

  "MuiOJ-backEnd/models/forms"

  JudgetModels "MuiOJ-backEnd/models/judger"

  JudgerService "MuiOJ-backEnd/services/judger"

  "MuiOJ-backEnd/services/question"

  SubmissionService "MuiOJ-backEnd/services/submission"

  "MuiOJ-backEnd/services/user"

  "MuiOJ-backEnd/utils/file"

  "errors"

  "fmt"

  "os"

)

func CodeSubmit(tid uint32, submitForm \*forms.SubmitForm, authObject \*auth.Claims, isContest bool) (uint32, error) {

  questionJudge, err := question.JudgeInfo(tid)

*if* err != nil {

*return* 0, err

  }

  questionDetail, err := question.Detail(tid)

*if* err != nil {

*return* 0, err

  }

*if* !isContest && questionDetail.Hide && !authObject.IsAdmin {

*return* 0, errors.New("permission denied")

  }

  fileName, err := files.CodeGenerateFileNameWithMkdir(authObject.Uid)

*if* err != nil {

*return* 0, err

  }

  filePath, err := files.CodePath(fileName)

*if* err != nil {

*return* 0, err

  }

*if* err := os.WriteFile(filePath, []byte(submitForm.Code), 0644); err != nil {

*return* 0, err

  }

  submission, err := SubmissionService.Create(tid, authObject.Uid, submitForm.Language, fileName)

*if* err != nil {

*return* 0, err

  }

*go* func(submission \*models.Submission) {

    starterParameter := &JudgetModels.StarterType{

      Code:       []byte(submitForm.Code),

      Sid:        submission.Sid,

      Tid:        submission.Tid,

      Language:   submission.Language,

      Test:   questionJudge.Dataset,

    }

    question.UpdateAttemptCount(tid)

    user.UpdateAttemptCount(authObject.Uid)

    status, result, err := JudgerService.Scheduler(starterParameter)

*if* status == "CE" || err != nil {

      fmt.Println(err)

      err := SubmissionService.CE(submission.Sid, result)

*if* err != nil {

*return*

      }

    }

    SubmissionService.JudgeResponseCallback(submission.Sid, result)

  }(submission)

*return* submission.Sid, nil

}

37. auth.go

package middlewares

import (

  "MuiOJ-backEnd/controllers/auth"

  "github.com/gin-gonic/gin"

  "strings"

)

func AuthJWT() gin.HandlerFunc{

*return* func(c \*gin.Context) {

    token := c.Request.Header.Get("Authorization")

    token = strings.ReplaceAll(token, "Bearer ", "")

*if* token == "" {

      c.JSON(403, gin.H{

        "code": 403,

        "message": "Access Denied",

      })

      c.Abort()

*return*

    }

    claims, err := auth.VerifyJWT(token)

*if* err != nil {

      c.JSON(403, gin.H{

        "code": 403,

        "message": "Access Denied",

      })

      c.Abort()

*return*

    }

    c.Set("AuthObject", claims)

  }

}

37. cors.go

package middlewares

import (

  "github.com/gin-gonic/gin"

  "net/http"

)

func Cors() gin.HandlerFunc {

*return* func(c \*gin.Context){

    w := c.Writer

    w.Header().Set("X-Powered-By", "Wings/1.0")

    w.Header().Set("Access-Control-Allow-Origin", "\*")

    w.Header().Set("Access-Control-Allow-Methods", "OPTIONS, POST, GET, PUT, DELETE")

    w.Header().Add("Access-Control-Allow-Headers", "Origin, X-Requested-With, Content-Type, Accept, Authorization")

    method := c.Request.Method

*if* method == "OPTIONS" {

      c.JSON(http.StatusOK, "Options Request!")

    }

    c.Next()

  }

}