

2020暑假小学期JAVAEE

系统详细设计说明书

|  |  |
| --- | --- |
| **项目经理** | 田丰瑞 |
| **项目成员** | 曹大华、巩怡霖、李朝龙、向苡霄、张一卓 |

**版本修改说明**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **修改人** | **修改内容** | **存在问题或争议** |
| 0.1 | 田丰瑞 | 填写登录，注册和Dashboard界面部分内容及接口 | 接口命名规范不统一  3.2节至6章内容不完整 |
| 0.2 | 田丰瑞 | **更正统一了接口命名规范**  **编写6.1声明内容**  补充3.2内容 | **不同用户角色是否使用统一接口**  3.2节至6章内容不完整 |
| 0.3 | 田丰瑞 | **修改了6.4节PasswordReminder接口**  **修改了6.6节UpcomingIssue接口变量名格式**  **在正文和接口处增加超链接，方便从正文直接跳到接口，或从接口处直接跳回正文** | **不同用户角色是否使用统一接口**  **如何向后端告知访问本系统用户身份？**  3.2节至6章内容不完整 |
| 0.4 | 田丰瑞 | **更正了CoursePanel接口的错误** |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

注：此文档从1.0版本开始保存版本历史记录，在正式版本1.0发布之前保留文件历史记录没有太大价值，故此在GitHub上不保留原版本。

目录

[1 目录 3](#_Toc44688095)

[2 用户角色总览 5](#_Toc44688096)

[3 网页说明——学生用户 5](#_Toc44688097)

[3.1 登录相关界面 5](#_Toc44688098)

[3.1.1 登录界面 5](#_Toc44688099)

[3.1.2 注册界面 5](#_Toc44688100)

[3.1.3 密码找回界面 6](#_Toc44688101)

[3.2 Dashboard界面 7](#_Toc44688102)

[3.2.1 侧边栏详细设计 7](#_Toc44688103)

[3.2.2 Upcoming Issue详细设计 10](#_Toc44688104)

[3.2.3 Recent Info详细设计 11](#_Toc44688105)

[3.3 PrevCourse界面 12](#_Toc44688106)

[3.4 课程界面 12](#_Toc44688107)

[3.4.1 课程侧边栏详细设计 13](#_Toc44688108)

[3.4.2 课程内容详情页设计 14](#_Toc44688109)

[4 网页说明——教师用户 16](#_Toc44688110)

[5 网页说明——管理员用户 17](#_Toc44688111)

[6 接口定义 17](#_Toc44688112)

[6.1 声明 17](#_Toc44688113)

[6.2 Login接口 17](#_Toc44688114)

[6.3 Register接口 18](#_Toc44688115)

[6.4 PasswordReminder接口 19](#_Toc44688116)

[6.5 DashBoardMenu接口 19](#_Toc44688117)

[6.6 UpComingIssue接口 21](#_Toc44688118)

[6.7 RecentInfo接口 22](#_Toc44688119)

[6.8 GetCoursePannel接口 23](#_Toc44688120)

[6.9 GetCourseAnnouncement接口 24](#_Toc44688121)

[6.10 GetAnnouncement接口 25](#_Toc44688122)

# 用户角色总览

用户角色种类：学生、教师、管理员

# 网页说明——学生用户

## 登录相关界面

### 登录界面

文件名称：pages-login.html

功能：用户通过输入用户名和密码实现登录，用户点击记住密码可以实现密码保存。

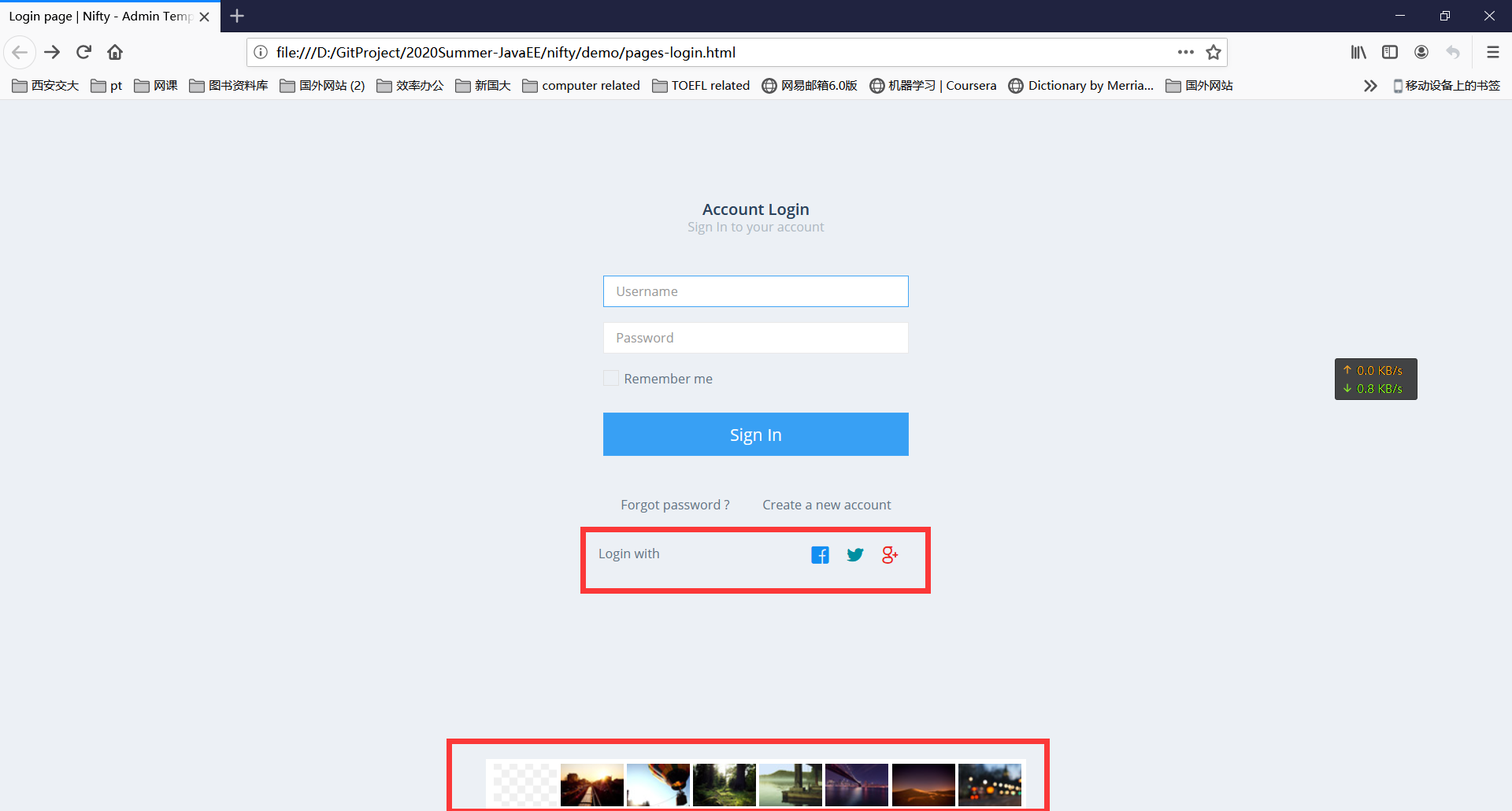


图2.1.1-1登录界面

**注：红框标识的内容先不实现。**

**接口路径见本文档中**[**Login**](#_Login接口)**接口。**

**用户通过输入正确用户名密码跳转到Dashboard界面，如密码输入错误，网页提示密码错误**

### 注册界面

文件名称：pages-register.html

功能：用户通过输入用户名，密码，邮箱地址，昵称实现注册

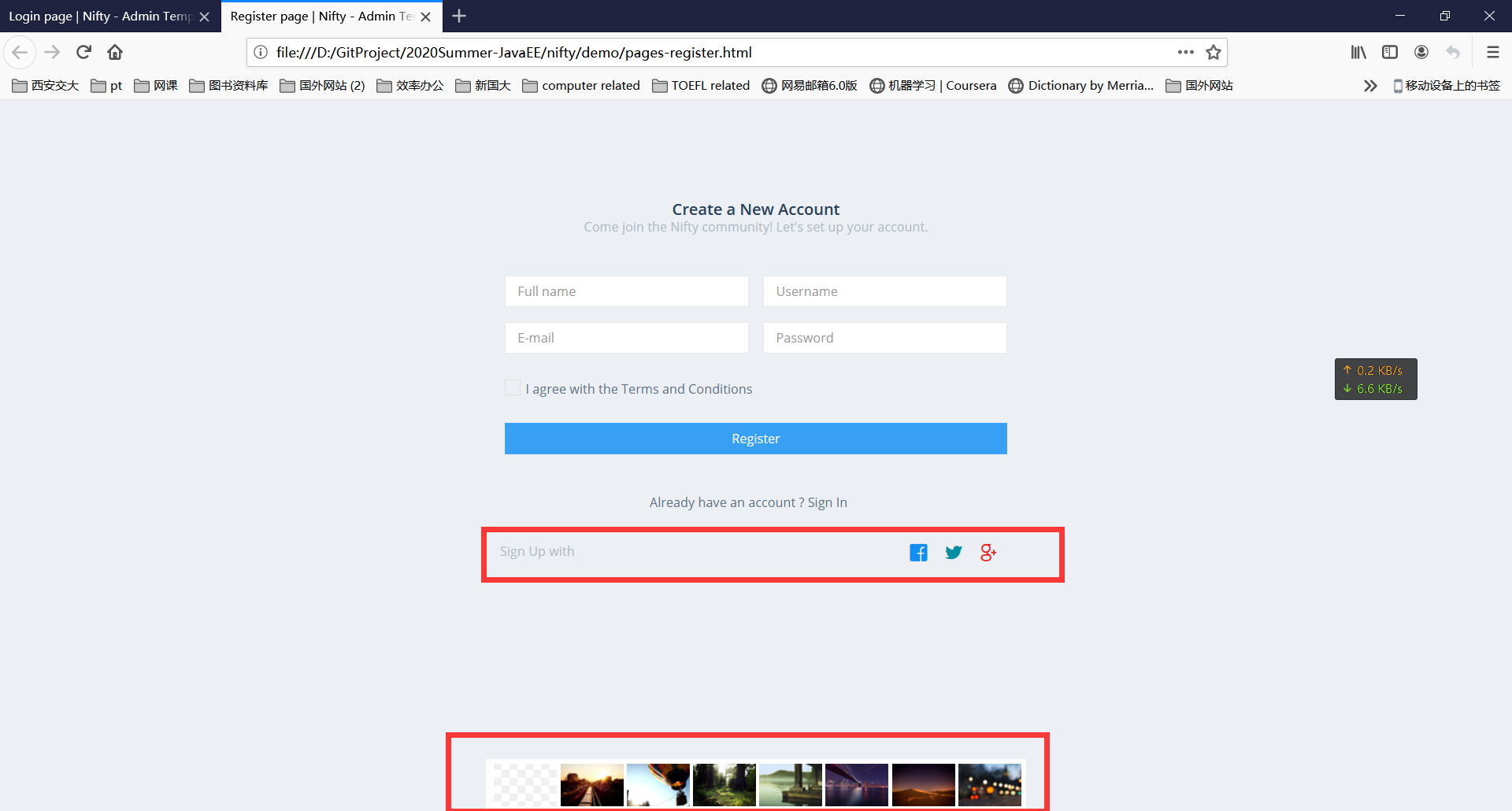


图2.1.2-1用户注册界面

**注：红框标识的内容先不实现。**

**注册接口见接口定义中**[Register](#_Register接口)**接口。**

**用户注册后若无问题提示注册成功后并跳转到登录界面**

**若Username已存在则提示用户名已存在,**

**若邮箱已注册则提示注册邮箱已存在。**

### 密码找回界面

文件名称：pages-password-reminder.html

功能：用户通过输入注册的邮箱地址实现密码找回

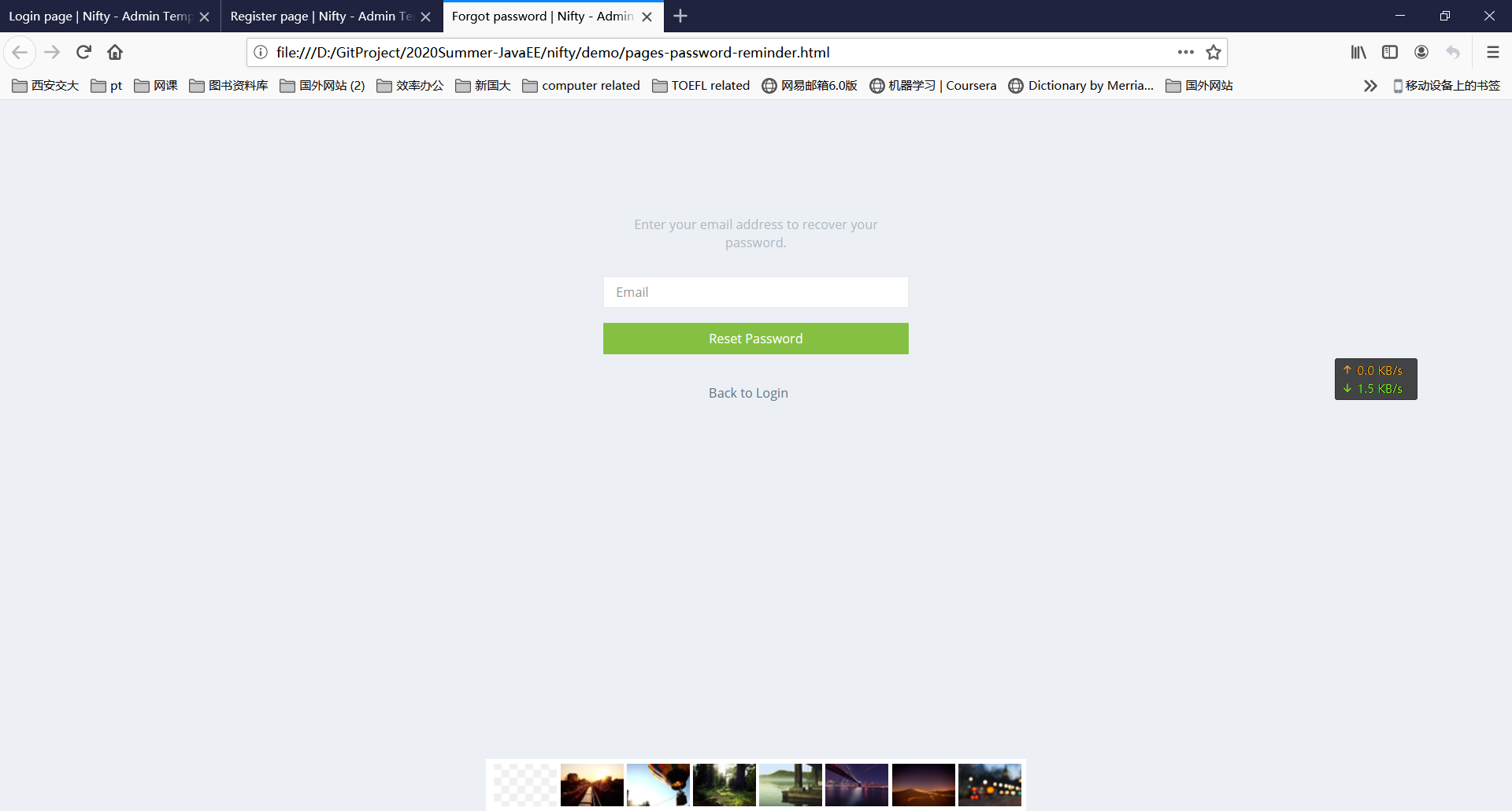


图2.1.3-1密码找回界面

接口见接口定义中给[PassworReminder](#_PasswordReminder接口)接口

## Dashboard界面

文件名称：index.html

界面内容：

* 侧边栏：用于显示用户个人信息和课程信息
* Upcomming Issue：用于显示用户所选课程的发布的近期作业信息
* Recent Info：用于显示用户所选课程发布的近期公告、资源、讨论区新内容信息
* Footer：待定
* 顶部banner：（待定）从左到右依次为：侧边栏拓展按钮，提醒按钮，网页信息按钮，语言按钮，个人信息按钮，聊天按钮。

**其中：个人信息按钮点击后出现logout选项。**

DashBoard接口见接口定义的[DashBoardMenu](#_DashBoardMenu接口)接口。

**DashBoard详细设计如下。**

### 侧边栏详细设计

#### 顶部内容

侧边栏顶部为用户头像，接下来是用户昵称和邮箱地址，位置见图2.2.1.1-1

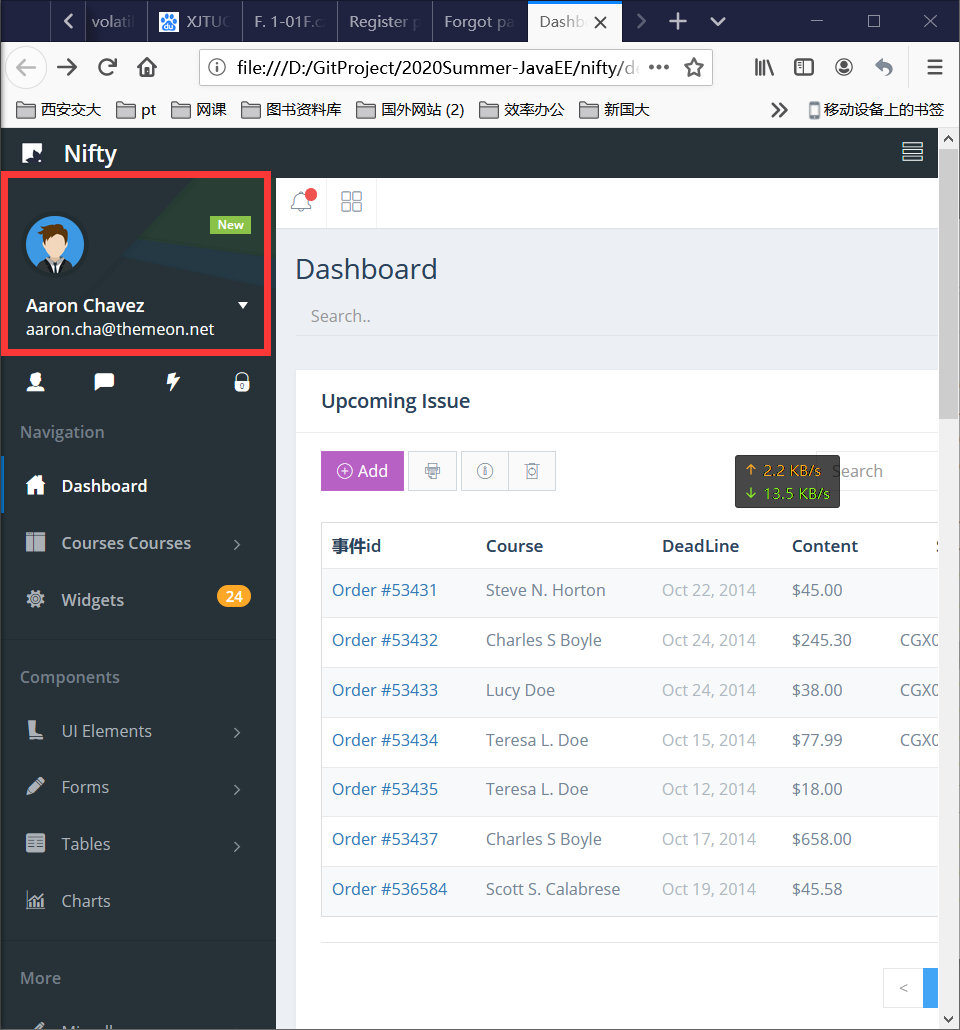


图2.2.1.1-1侧边栏用户头像、昵称、邮箱位置

#### Navigation栏下内容

Navigation栏位置见图2.2.1.2-1，其下面元素分别是DashBoard，PrevCourses两项

DashBoard项点击后跳转到DashBoard界面，PrevCourse项点击后跳转到PrevCourse界面

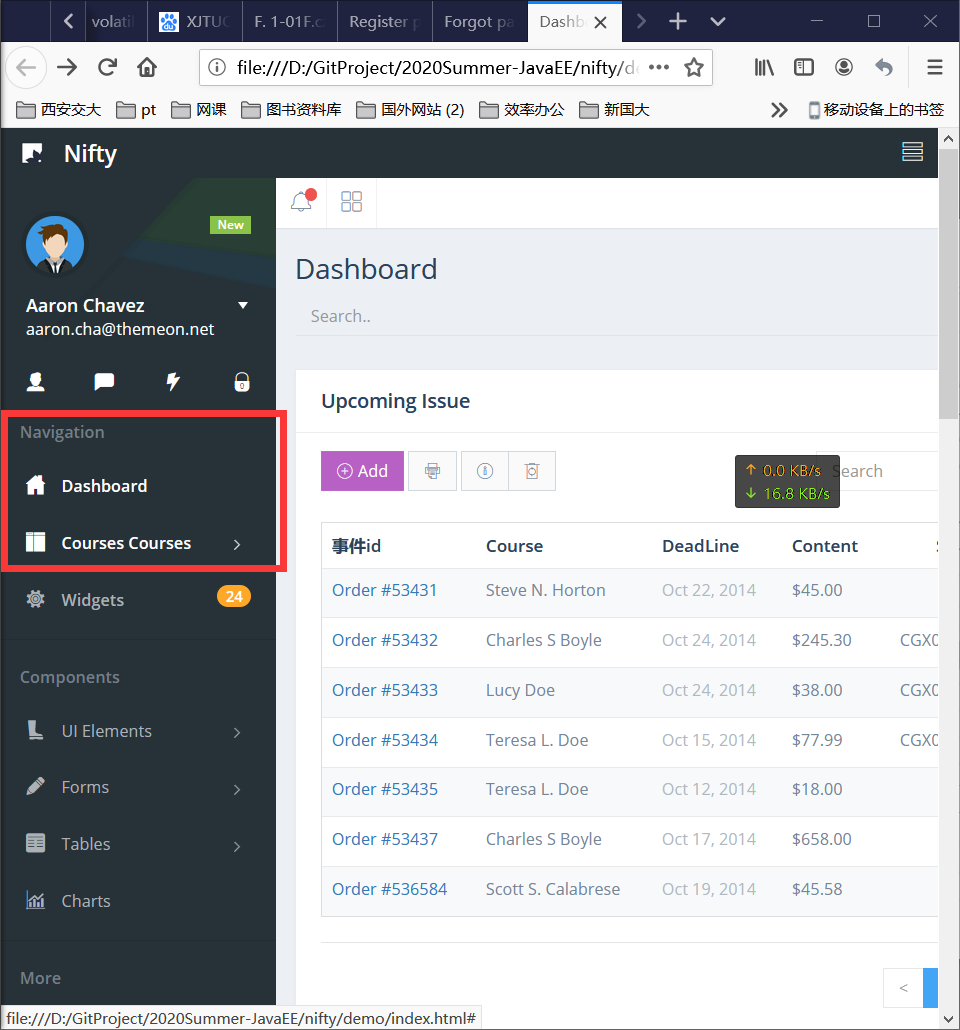


图2.2.1.2-1 Navigation栏位置

#### Courses栏下内容

Courses栏位置见图2.2.1.3-1，即图中Components栏位置。

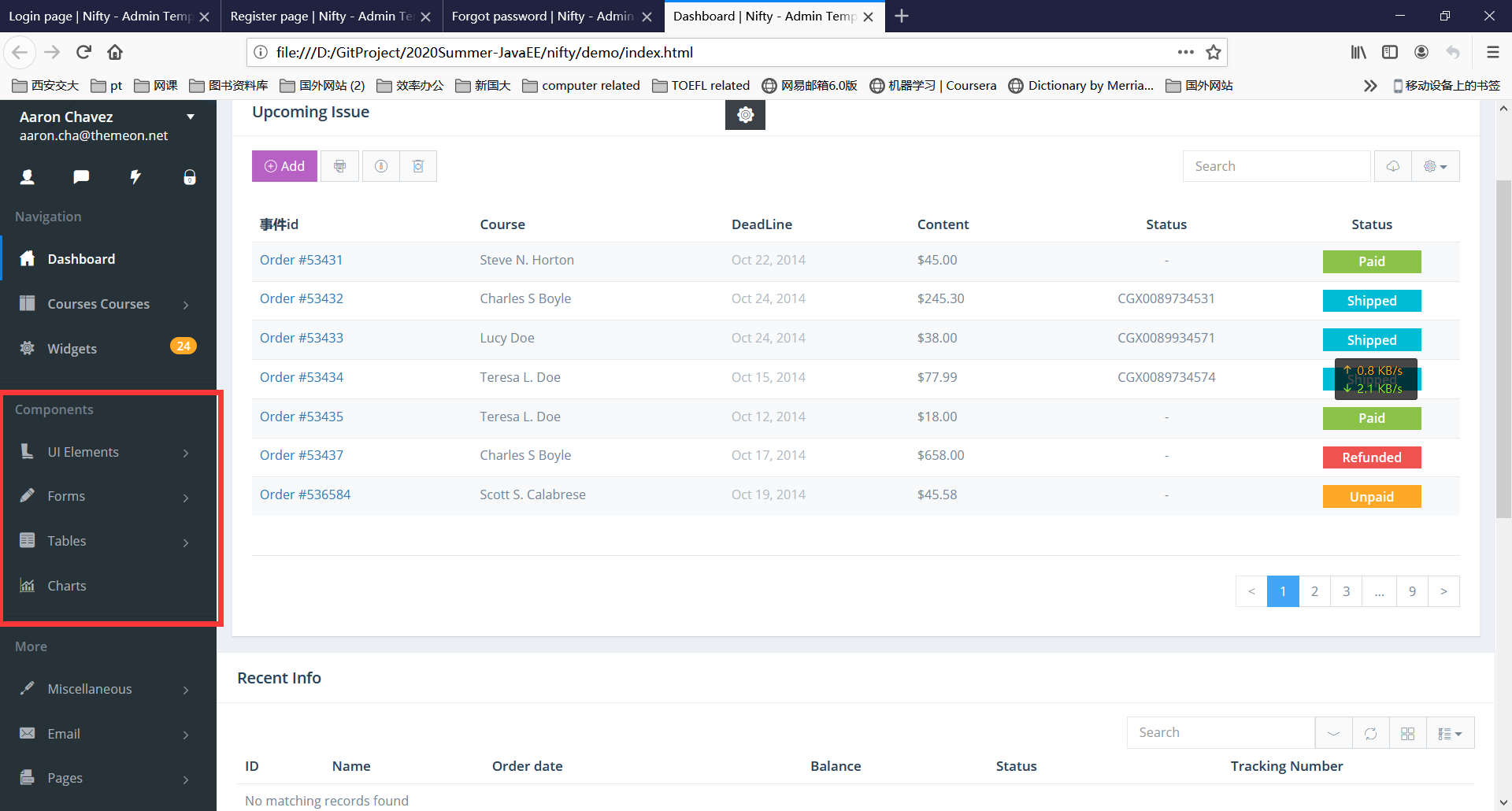


图2.2.1.3-1 Courses栏位置

需将Components改为Courses，Courses栏下内容为学生本学期课程链接，其课程目录通过向网页后端发送请求查询。

**剩余侧边栏元素暂不考虑。**

### Upcoming Issue详细设计

Upcoming Issue位置见图2.2.2-1。

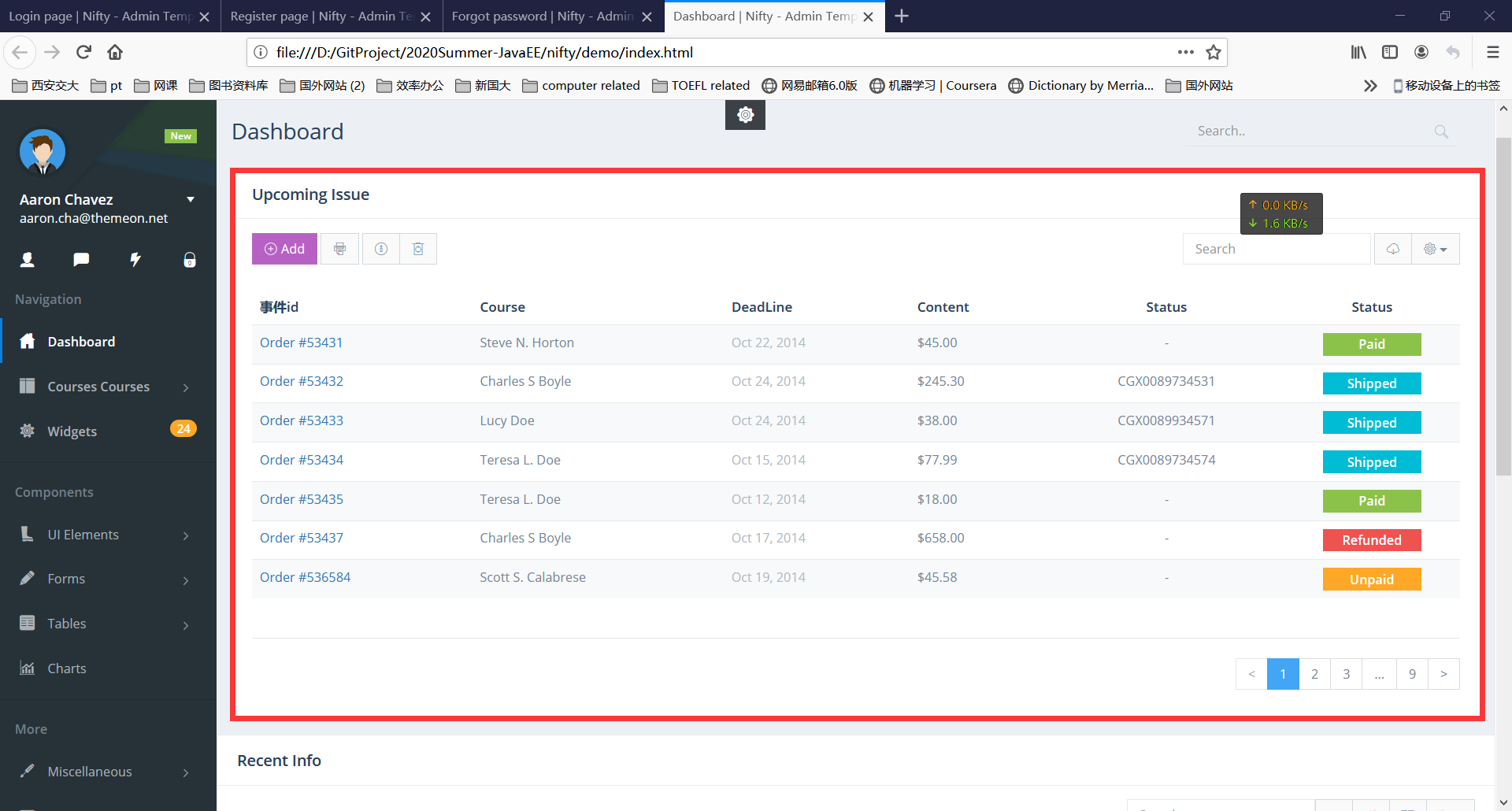


图2.2.2-1 Upcoming Issue表格位置

其一共有五个表项，其显示内容见表格2.2.2-1，表项位置见图2.2.2-2，其中倒数第二个Status栏删除。

表2.2.2-1 Upcoming Issue 表项定义

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 事件ID | Course | Deadline | Content | Status |
| 作业或考试或问卷对应事件ID | 事件所属课程 | 本事件截至日期 | 本事件标题 | 本事件状态（On Schedule 蓝色， Expired 红色， Submitted 绿色）  On Schedule在DDL前一天颜色变黄 |

接口见接口定义中的[UpcomingIssue](#_UpComingIssue接口)接口。

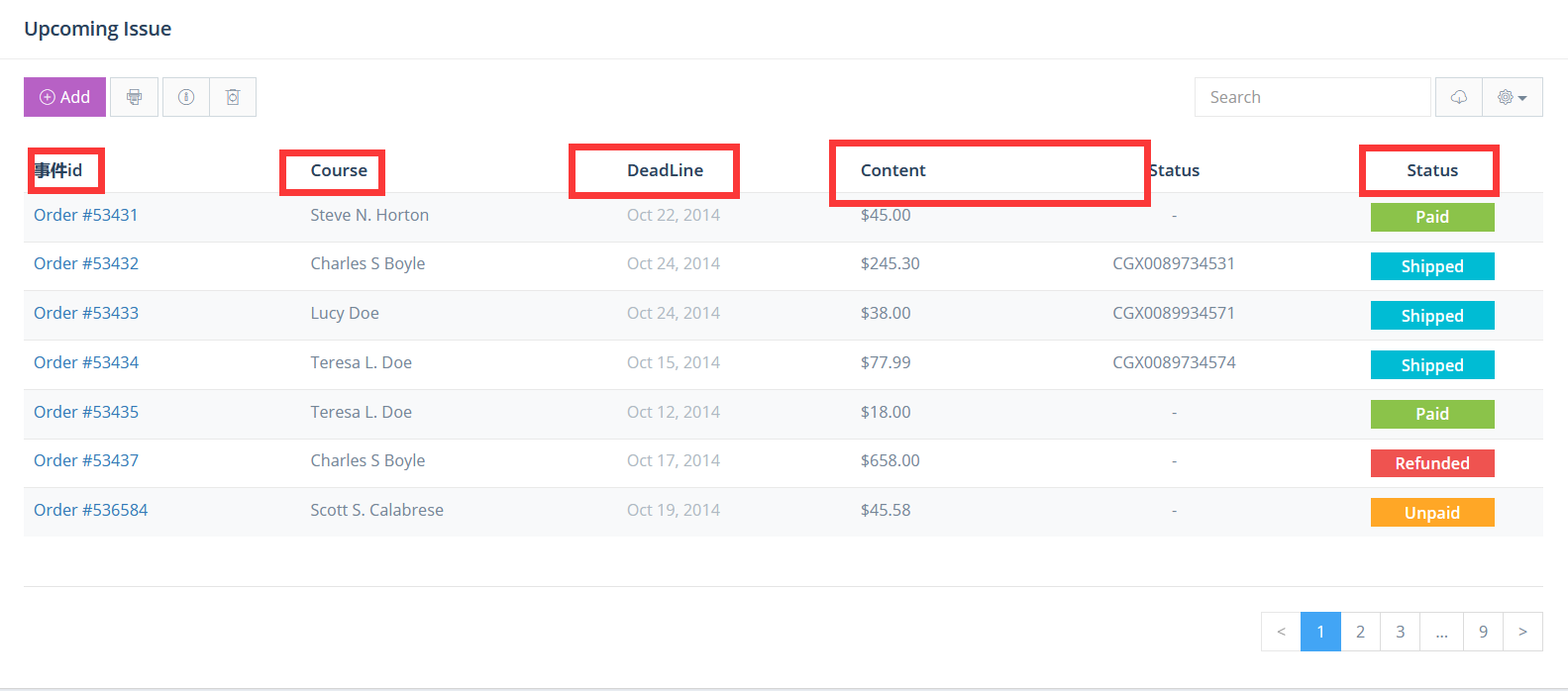


图2.2.2-2 UpComing Issue表项位置

用户点击事件ID跳转到对应ID详情页，即对应课程作业\考试\问卷详情页。

### Recent Info详细设计

Recent Info位置 见图3.2.3-1。

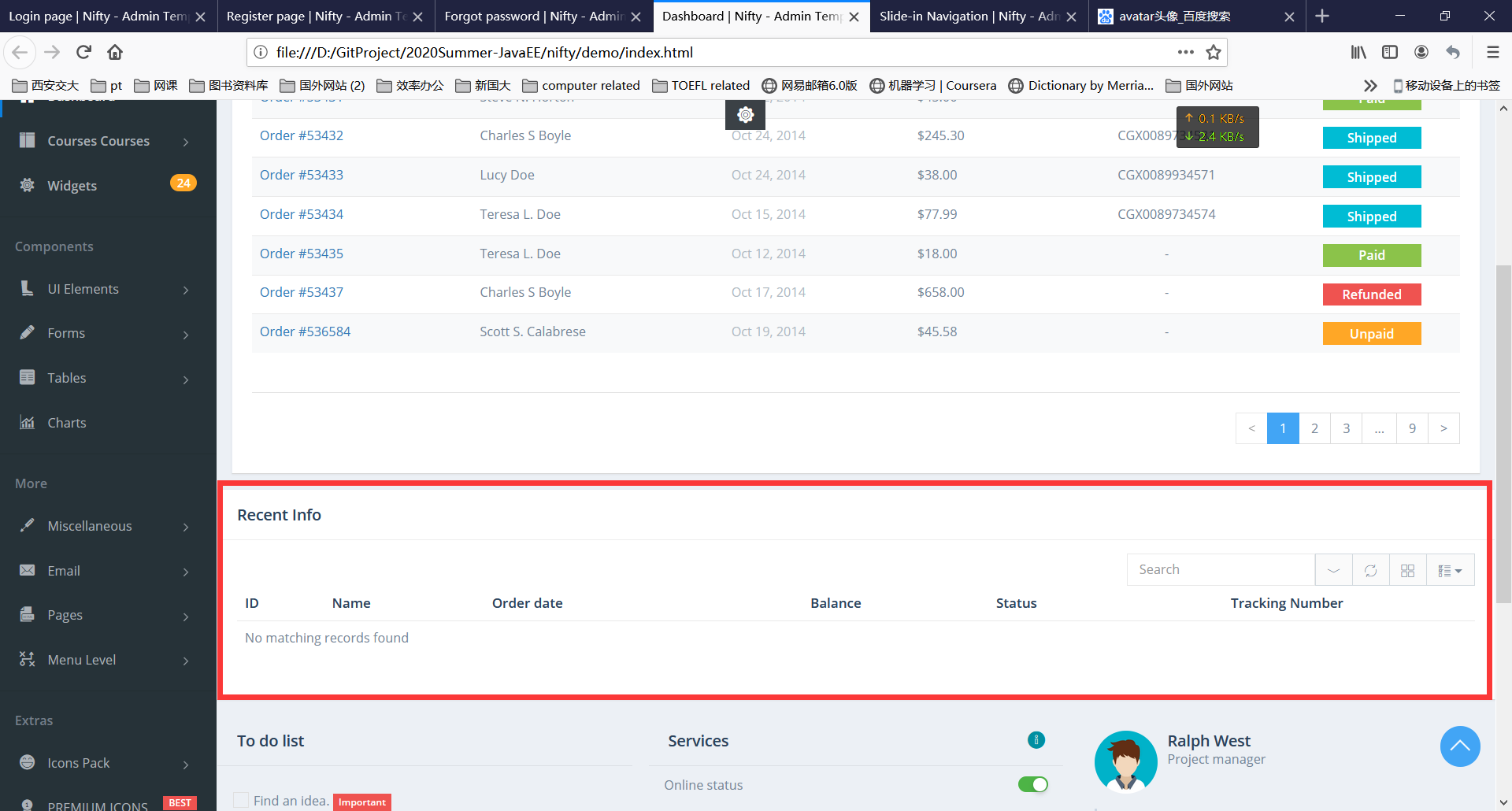


图3.2.3-1 Recent Info表位置

Recent Info表共有四项，每项详细定义见表3.2.3-1。

表3.2.3-1 Recent Info详细定义表

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Course | Content | Publish Datetime |
| 新消息ID | 消息所属内容 | 消息Title | 消息发布时间 |

Recent Info展示教师发布的新的公告，视频、音频、文档资源。

**Recent Info接口见接口定义**[RecentInfo](#_RecentInfo接口)**项。**

## PrevCourse界面

待定。

## 课程界面

界面名称：course-dashboard.html

界面内容：

* 用户侧边栏：见3.2.1设计
* 课程侧边栏：位置如图3.4-1所示。

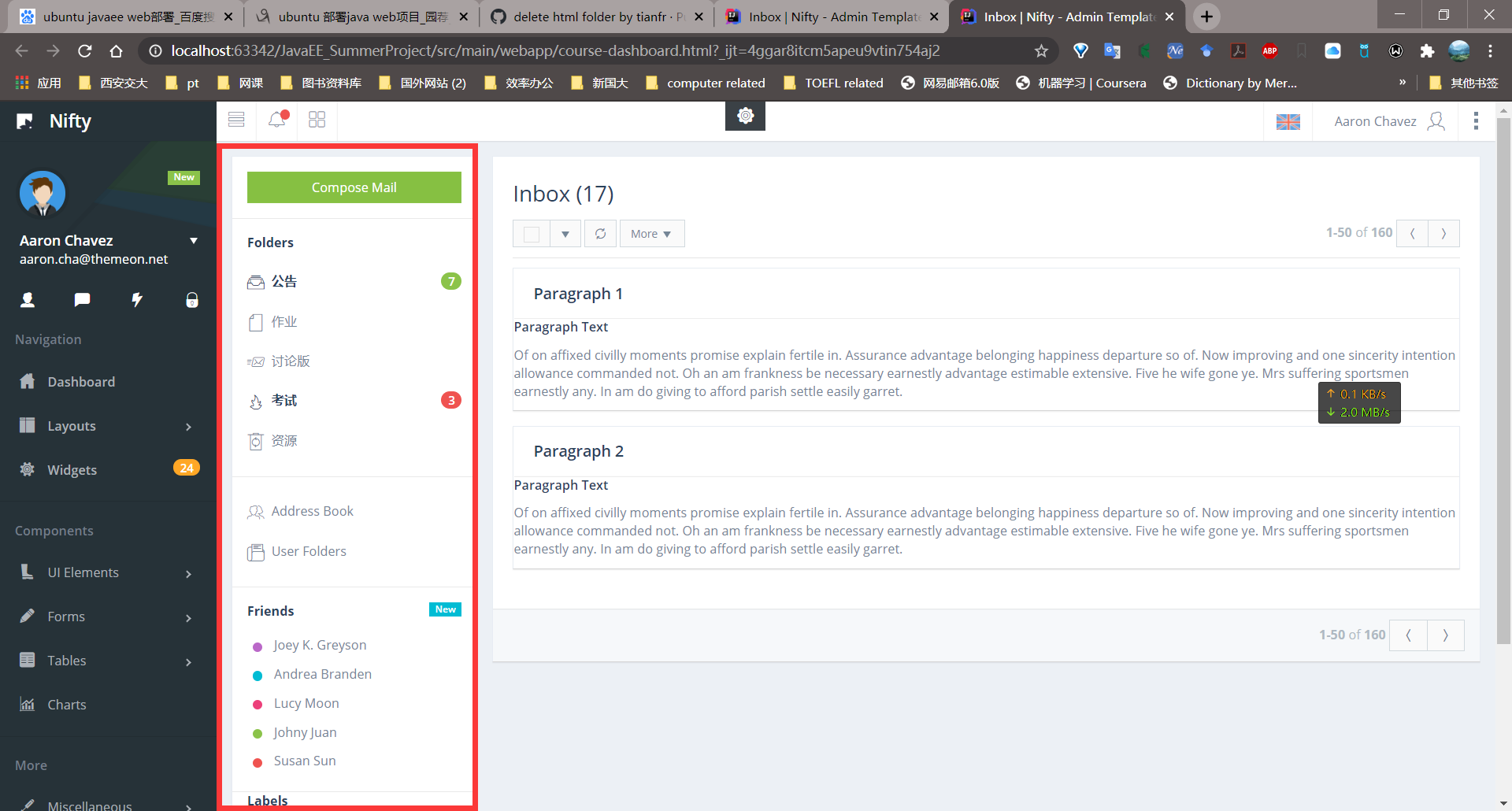


图3.4-1 课程侧边栏展示图

* 课程内容详情部分：位置如图3.4-2所示。

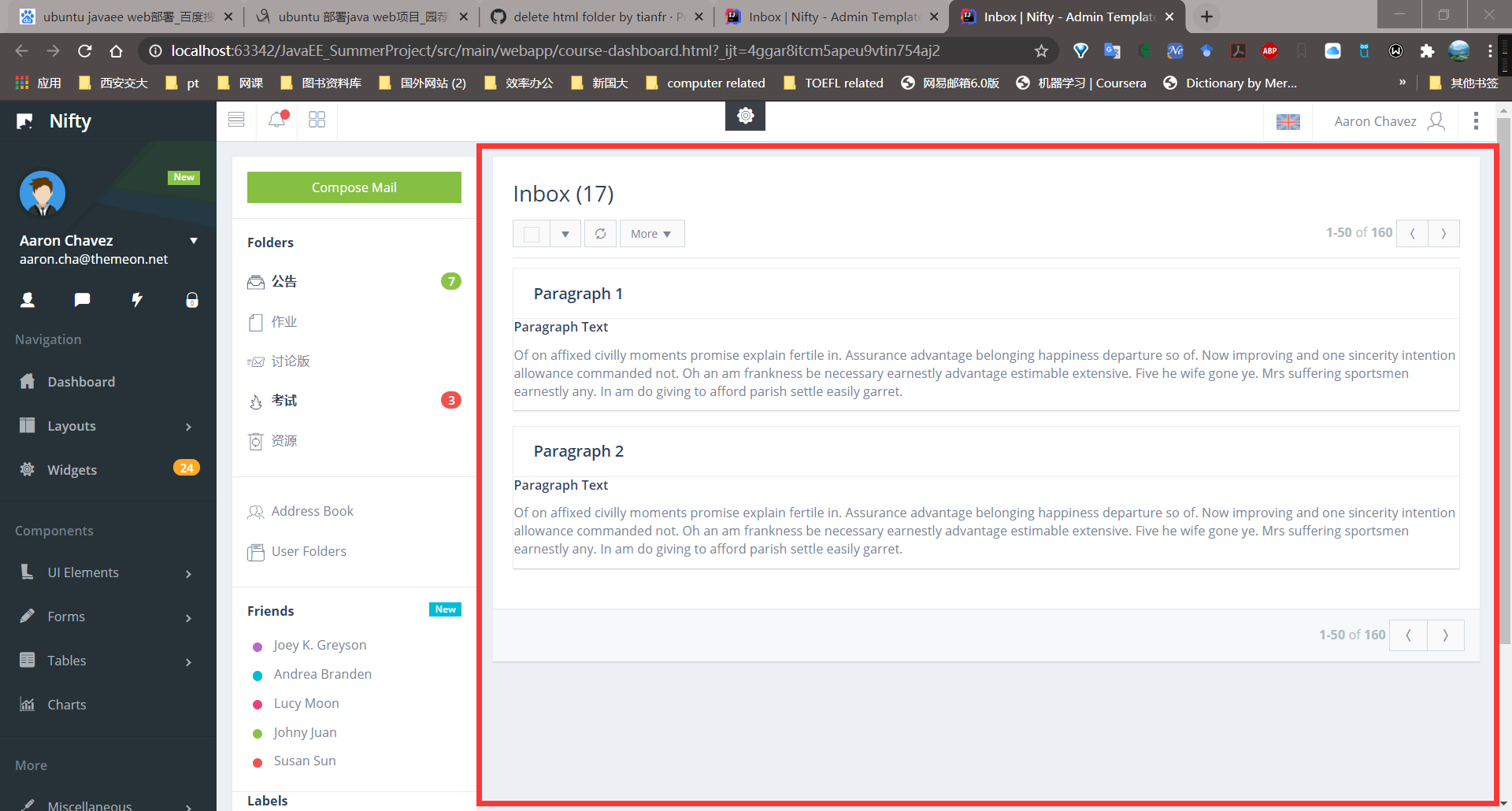


图3.4-2 课程内容详情部分位置

课程侧边栏，课程内容详情部分详细设计见3.4.1与3.4.2.

### 课程侧边栏详细设计

#### Folders部分

Folders侧边栏位置见图3.4.1.1-1。

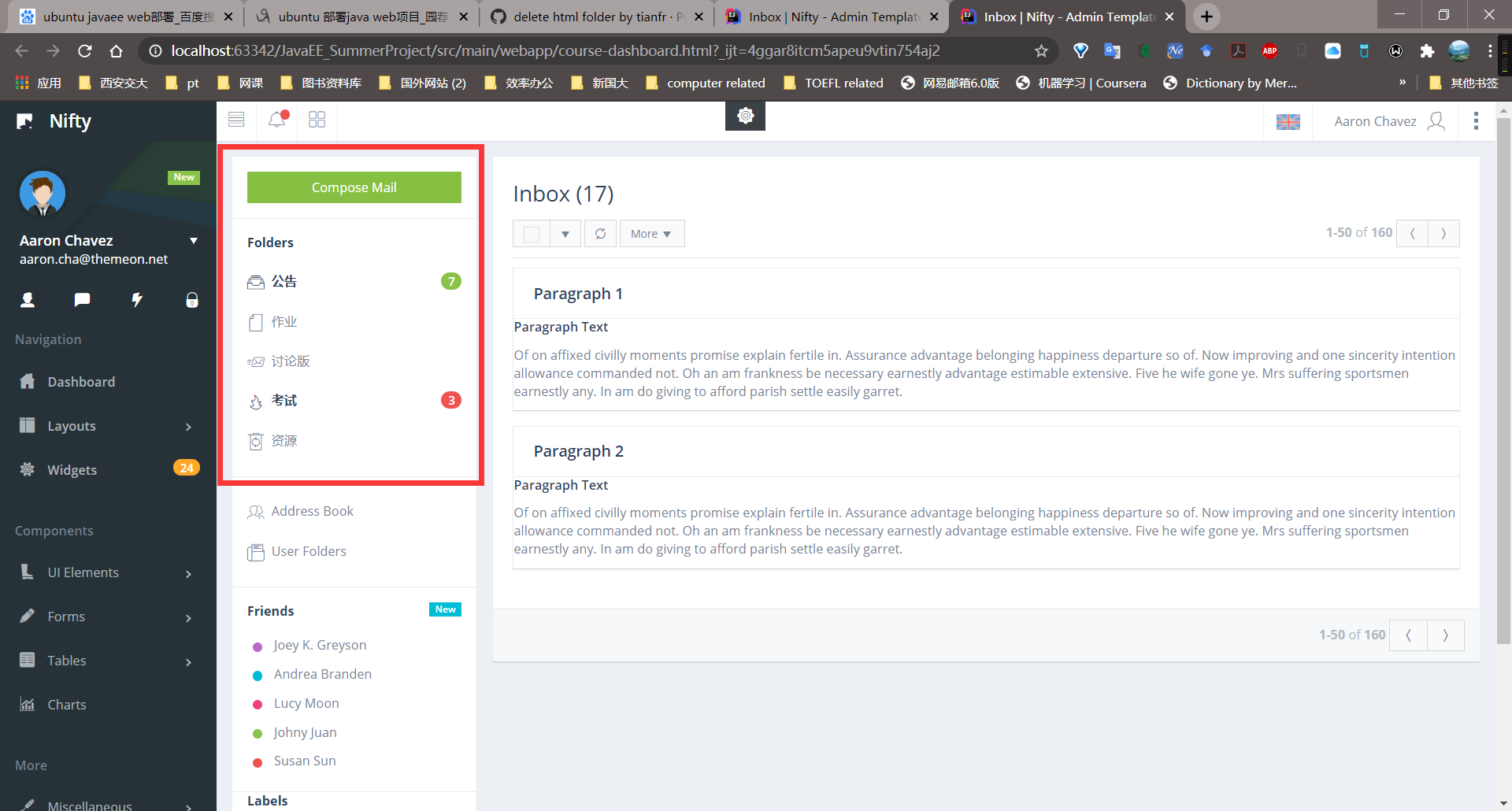


图3.4.1.1-1 侧边栏Folders部分

该部分共有5个button：公告，作业，讨论版，考试，资源。

Foulders上面绿色按钮“compose mail”删除。

用户分别点击每个button进入每个button所对应的课程详情页的部分内容。

#### Foulder栏下部分

Folder栏下面内容为教师个性化定义部分，其显示的数据通过后台接口查询实现。

接口详情见接口定义中的[GetCoursePanel](#_GetCoursePannel接口)接口。

### 课程内容详情页设计

#### 公告详情页设计

用户在课程界面的课程侧边栏上点击公告按钮，进入如图3.4.2.1-1的公告界面。

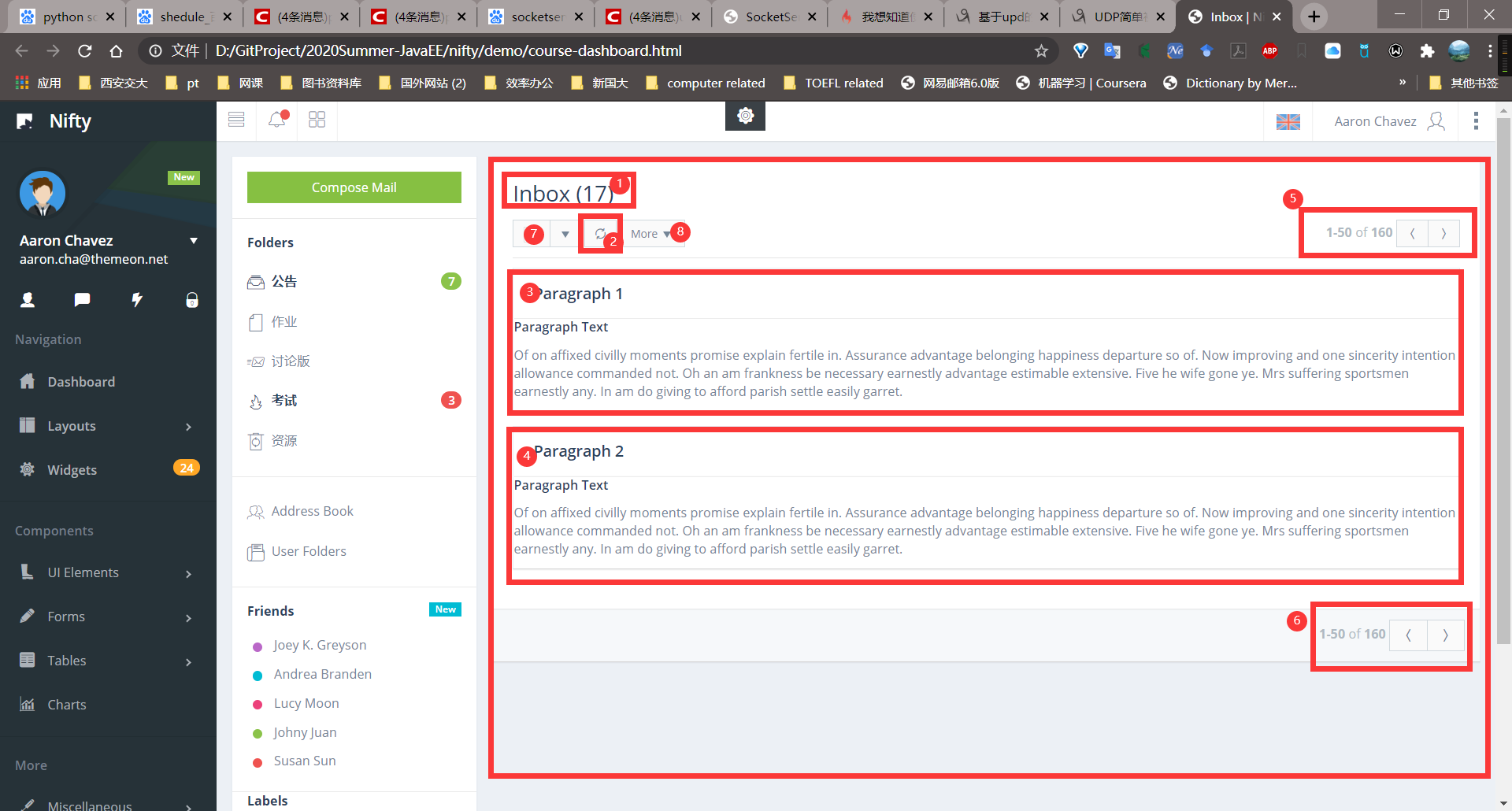


图3.4.2.1-1公告详情页

图中序号1处文字修改为课程公告（Course Pins）.

图中序号2处按钮点击功能为：刷新公告面板（即向后端接口重新请求接受公告数据）。

图中序号3，4为具体公告内容，其详细设计见3.4.3.1.1节

图中序号5，6，7，8处功能目前先不实现，在系统整个运行后再做细节调整。

序号3，4内容需要通过向后端请求数据实现，接口具体格式见接口定义中的[GetCourseAnnouncement](#_GetCourseAnnouncement接口)接口。

##### 公告展示具体设计

公告展示具体设计见图3.4.2.1.1-1

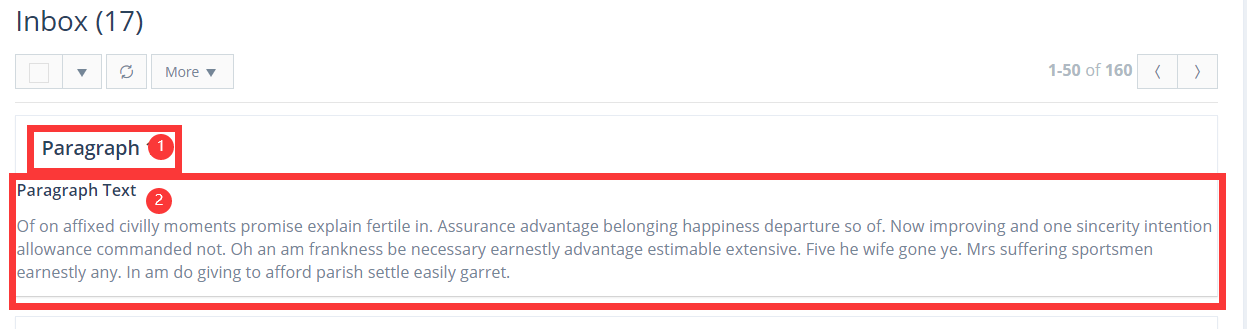


图3.4.2.1.1-1具体公告内容示意图

序号1处显示为公告标题，其可点击，点击后跳转到本公告详细界面，具体设计见章节3.4.2.1.2课程公告页设计。

序号2处显示公告内容，显示公告前3~5行内容。

##### 课程公告页设计

用户通过点击课程界面公告标题进入具体课程公告页，具体公告页位置如图3.4.2.1.2-1所示。

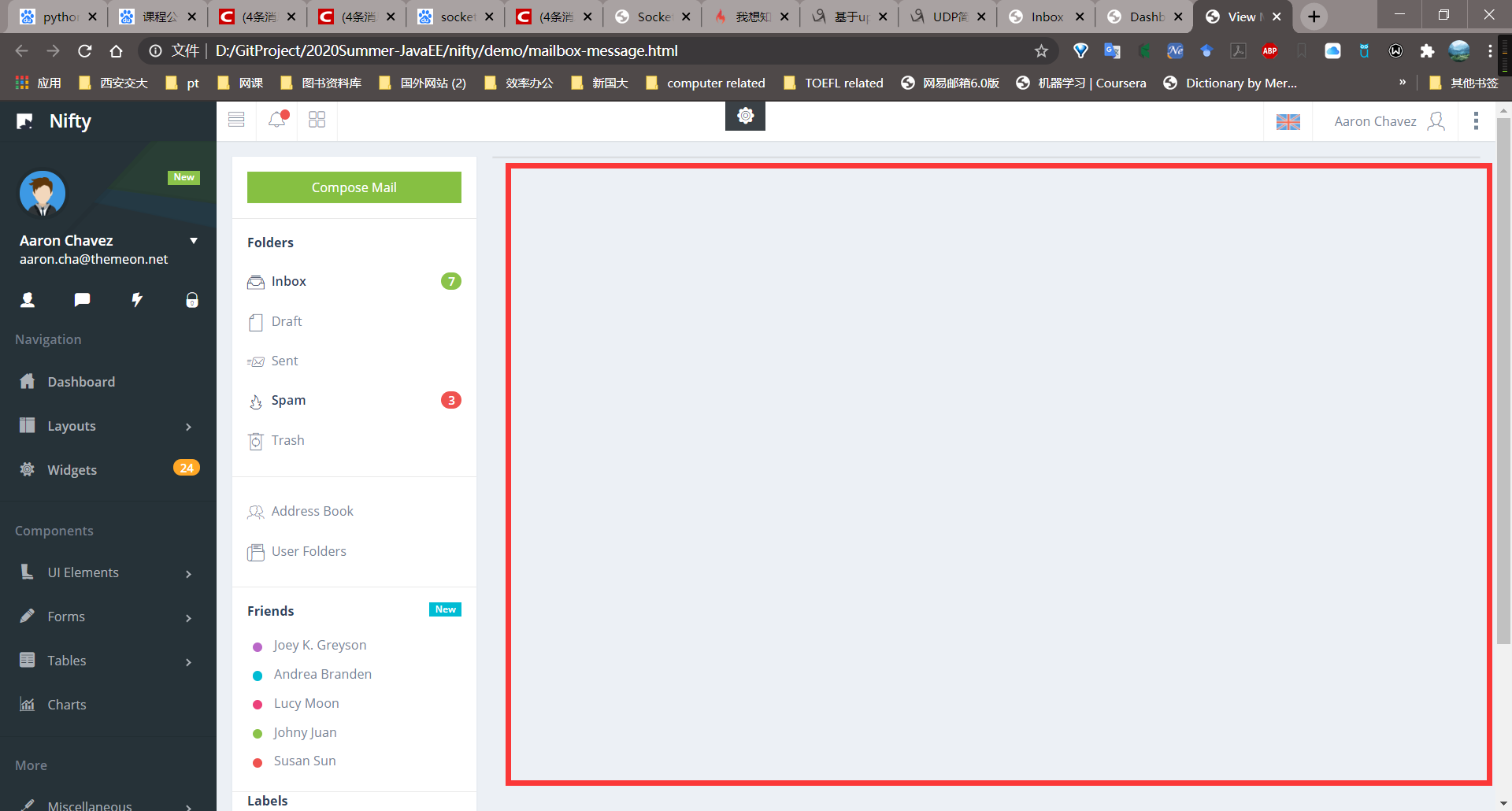


图3.4.2.1.2-1具体课程公告页位置

其左侧课程侧边栏设计与之前3.4.1设计保持一致。

具体公告内容展示参考文件mailbox-message.html文件。

注意：Github版本上的mailbox-message.html文件已损坏，我已将需要的该文件发送到qq群中名为“更新的文件”的文件夹下。

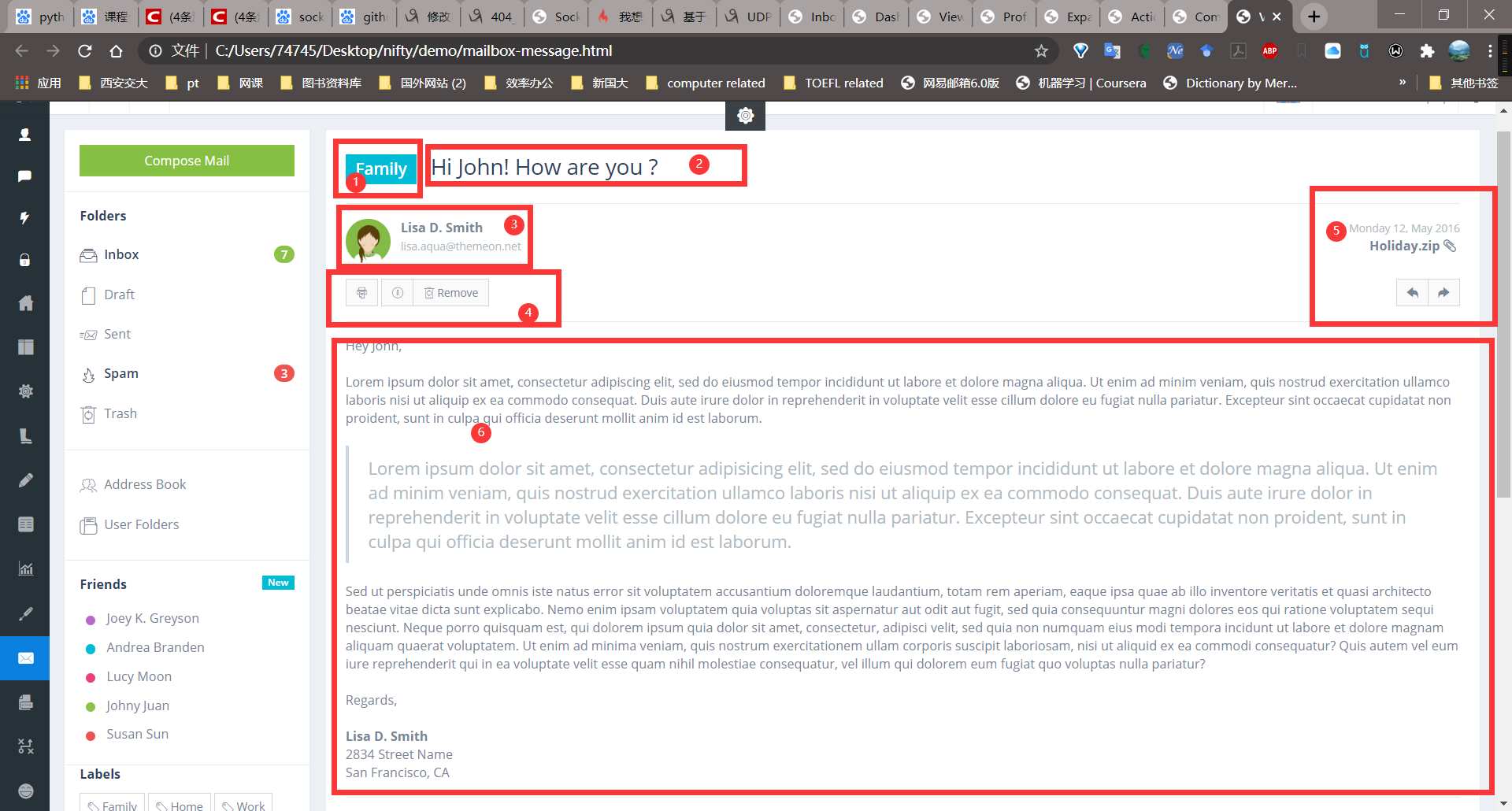


图3.4.2.1.2-2公告展示内容具体设计

在图3.4.2.1.2-2中，标号1部分文字改为公告，标号2部分显示公告标题，标号3部分展示公告发布老师名称，头像（默认头像不动即可），邮箱，标号6部分展示公告具体内容，标号5部分第一行更新公告发布日期，标号4部分功能不用实现，保留按钮即可。

其标号2，标号3，标号5，标号6内容通过后端接口请求后端获取数据，定义见接口定义的GetAnnouncement接口。

#### 作业详情页设计

#### 讨论吧详情页设计

#### 考试详情页设计

#### 资源详情页设计

# 网页说明——教师用户

登录界面

Dashboard界面

课程界面

公告相关界面

作业相关界面

讨论区相关界面

考试、问卷调查相关界面

成绩查看相关界面

# 网页说明——管理员用户

# 接口定义

## 声明

**由于后端数据库变量命名采用以小写字母组成的英文单词和下划线分割构成表项名称，故所有前后端发送的内容建议以小写英文单词加下划线命名消息。**

## Login接口

该接口出现在[3.1.1节登录界面](#_登录界面)。

接口路径：

/Login

Js函数路径：

js/myjs.js?v=1

请求方法：

Post

正向接口格式：

            data:JSON.stringify({

                "status": "0",

                "message": "login",

                "user\_name": user\_name,

                "pass\_word": pass\_word,

            }),

Username, Password,分别表示用户名，密码。

反向接口格式：

{

    "success\_value": 0, // 1表示成功，0表示失败

    "fail\_content":"错误详细信息，由后端返回", //错误详细信息：邮箱不存在

}

## Register接口

该接口出现在[3.1.2节注册界面](#_注册界面)。

接口路径：

/Register

接口js函数路径：

Js/myjs.js?v=1

请求方法：

Post

正向接口格式：

            data:JSON.stringify({

                "status": "0",

                "message": "register",

                "user\_name": user\_name,

                "pass\_word": pass\_word,

                "email": email,

                "full\_name": full\_name,

            }),

反向接口格式：

{

    "success\_value": 0, // 1表示成功，0表示失败

    "fail\_content":"错误详细信息，由后端返回", //错误详细信息：邮箱不存在

}

## PasswordReminder接口

该接口出现在[3.1.3节密码找回界面](#_密码找回界面)。

接口路径：

/PasswordReminder

接口js函数路径：

Js/myjs.js?v=1

请求方法：

Post

正向接口格式：

data: JSON.stringify({

    "status": "0",

    "message": "password\_reminder",

    "email": email,

}),

反向接口格式：

{

    "success\_value": 0, // 1表示成功，0表示失败

    "fail\_content":"错误详细信息，由后端返回", //错误详细信息：邮箱不存在

}

## DashBoardMenu接口

该接口出现在3.2.1.3节[Courses栏下内容](#_Courses栏下内容)

接口路径：

/DashBoardMenu

接口js函数路径

Js/myjs.js?v=1

请求方法：

Post

正向接口格式：

            data: JSON.stringify({

                "status": "0",

                "user\_id": user\_id,

                "message": "dashboard\_menu",

            }),

反向接口格式：

{

    "avatar":avatar\_path,

    "user\_name": user\_name,

    "id": "courses",

    "value": [

        {

            "id": "course1",

            "course\_name": course\_name,

            "course\_path": course\_path

        },

        {

            "id": "course2",

            "course\_name": course\_name,

            "course\_path": course\_path

        },

        {

            "id": "course3",

            "course\_name": course\_name,

            "course\_path": course\_path

        },//当前学期课程数量不定，需要用循环遍历，id自增

    ]

}

注意：id为courses项的value项为列表，其项目数量不确定。

Coursename项存储课程名称，显示到侧边栏上，Coursepath项存储访问该课程的网址

## UpComingIssue接口

该接口出现在[3.2.2节Upcoming Issue详细设计](#_Upcoming_Issue详细设计)。

接口路径：

/UpcomingIssue

接口js函数路径

Js/myjs.js?v=1

正向接口格式：

            data: JSON.stringify({

                "status": "0",

                "message": "upcoming\_issue",

            }),

反向接口格式

{

    "id": "issues",

    "value": [

        {

            "id": "issue\_id",

            "value":issue\_id,

            "issue\_id\_path": issue\_id\_path,

            "course":course,

            "deadline":deadline,

            "content":issue\_title,

            "status": status //expired, on\_schedule, submitted, On\_schedule\_yellow

        },

        {

            "id": "issue\_id",

            "value":issue\_id,

            "issueidpath": issue\_id\_path,

            "course":course,

            "deadline":deadline,

            "content":issue\_title,

            "status": status //expired, on\_shedule, submitted, on\_shedule\_yellow

        },

        //项目数量不定，需要遍历访问

    ]

}

## RecentInfo接口

该接口出现在[3.2.3节Recent Info 详细设计](#_Recent_Info详细设计)。

接口路径：

/RecentInfo

接口js函数路径

Js/myjs.js?v=1

正向接口格式：

            data: JSON.stringify({

                "status": "0",

                "message": "recent\_info",

            }),

反向接口格式

{

    "id": "recent\_info",

    "value": [

        {

            "id": "info\_id1",

            "value":info\_id,

            "info\_id\_path": info\_id\_path,

            "course":course,

            "datetime":datetime,

            "content":info\_title,

        },

        {

            "id": "info\_id2",

            "value":info\_id,

            "issue\_id\_path": issue\_id\_path,

            "course":course,

            "deadline":deadline,

            "content":info\_title,

        },

            //项目数量不定，需要遍历访问

    ]

}

## GetCoursePanel接口

该接口出现在[3.4.1.2节Folder栏下部分](#_Foulders栏下部分)。

接口路径：

/GetCoursePanel

接口js函数路径

Js/myjs.js?v=1

正向接口格式：

            data:JSON.stringify({

                "status": "0",

                "message": "get\_course\_panel",

                "course": course\_name,

                "teacher\_user\_name": teacher\_user\_name,

            }),

反向接口格式

{

    "id": "course\_panel",

    "value": [

        {

            "panel\_id": "panel\_id1",

            "panel\_name": panel\_name,

        },

        {

            "panel\_id": "panel\_id2",

            "panel\_name": panel\_name,

        },

            //项目数量不定，需要遍历访问

    ]

}

## GetCourseAnnouncement接口

该接口出现在[3.4.2.1节公告详情页设计](#_公告详情页设计)。

接口路径：

/GetCourseAnnouncement

接口js函数路径

Js/myjs.js?v=1

正向接口格式：

data: JSON.stringify({

    "status": "0",

    "message": "get\_course\_announcement",

    "course": course,

}),

反向接口格式

{

    "id": "course\_announcement",

    "total\_num": num\_of\_total\_announcement,

    "value": [

        {

            "announcement\_title": announcement\_title, //公告标题

            "announcement\_path": announcement\_path, //公告访问具体路径，用户点击公告标题后向后端接口发送该路径进入公告具体信息页

            "announcement\_content": announcement\_content,//公告内容

        },

        {

            "announcement\_title": announcement\_title,

            "announcement\_path": announcement\_path,

            "announcement\_content": announcement\_content,

        },

            //项目数量不定，需要遍历访问

    ]

}

## GetAnnouncement接口

该接口出现在[3.4.2.1.2节课程公告页设计](#_课程公告页设计)。

接口路径：

/GetAnnouncement

接口js函数路径

Js/myjs.js?v=1

正向接口格式：

data: JSON.stringify({

    "status": "0",

    "message": "get\_announcement",

    "course": course,

    "announcement\_path": announcement\_path //注意：这个变量在get\_course\_announcement中出现过。

}),

反向接口格式

{

    "announcement\_title": announcement\_title,//公告标题

    "author": teacher\_name,//公告发布教师名称

    "author\_email": teacher\_email, //公告发布教师邮箱

    "author\_avatar": authon\_avatar, //默认为初始头像

    "announcement\_content": announcement\_details,//这里有点争议，我设想返回的是html格式文档

    "datetime": annoucement\_datetime,//教师发布该公告的时间

}