



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	
0.5	田丰瑞	修改了 6.5 节 Dashboard 接口：B 改成小写， course_path 改成 course_id； 修改 6.8 节正向接口内容； 将 4 个 css 样式文件上传到群文件； 完成 3.4.2.2 作业详情页设计部分； 3.4.2.2 作业详情页完成至 3.4.2.2.2	3.4.2.2.3 接口未定义
0.6	田丰瑞	编写 6.13, 6.14 接口内容； 3.4.2.2.3 内容编写完毕； 修改了 6.12 反向接口格式。	文件上传如何实现？

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。

1 目录

1	目录.....	3
2	用户角色总览.....	5
3	网页说明——学生用户.....	5
3.1	登录相关界面	5
3.1.1	登录界面.....	5
3.1.2	注册界面.....	5
3.1.3	密码找回界面.....	6
3.2	Dashboard 界面.....	7
3.2.1	侧边栏详细设计.....	7
3.2.2	Upcoming Issue 详细设计	10
3.2.3	Recent Info 详细设计	11
3.3	PrevCourse 界面	12
3.4	课程界面	12
3.4.1	课程侧边栏详细设计.....	13
3.4.2	课程内容详情页设计.....	14
4	网页说明——教师用户.....	23
5	网页说明——管理员用户.....	23
6	接口定义.....	23
6.1	声明	23
6.2	Login 接口	23
6.3	Register 接口.....	24
6.4	PasswordReminder 接口	25
6.5	DashboardMenu 接口	26
6.6	UpComingIssue 接口	28
6.7	RecentInfo 接口	30

6.8	GetCoursePanel 接口	31
6.9	GetCourseAnnouncements 接口	33
6.10	GetAnnouncement 接口	34
6.11	GetCourseHomeworks 接口	35
6.12	GetHomework 接口	37
6.13	作业详情页接口	38
6.13.1	SubmitHomework 接口	38
6.13.2	GetExistedHomework 接口	40
6.14	文件上传下载相关接口	41
6.14.1	FileUpload 接口	41
6.14.2	FileDelete 接口	42

2 用户角色总览

用户角色种类：学生、教师、管理员

3 网页说明——学生用户

3.1 登录相关界面

3.1.1 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。

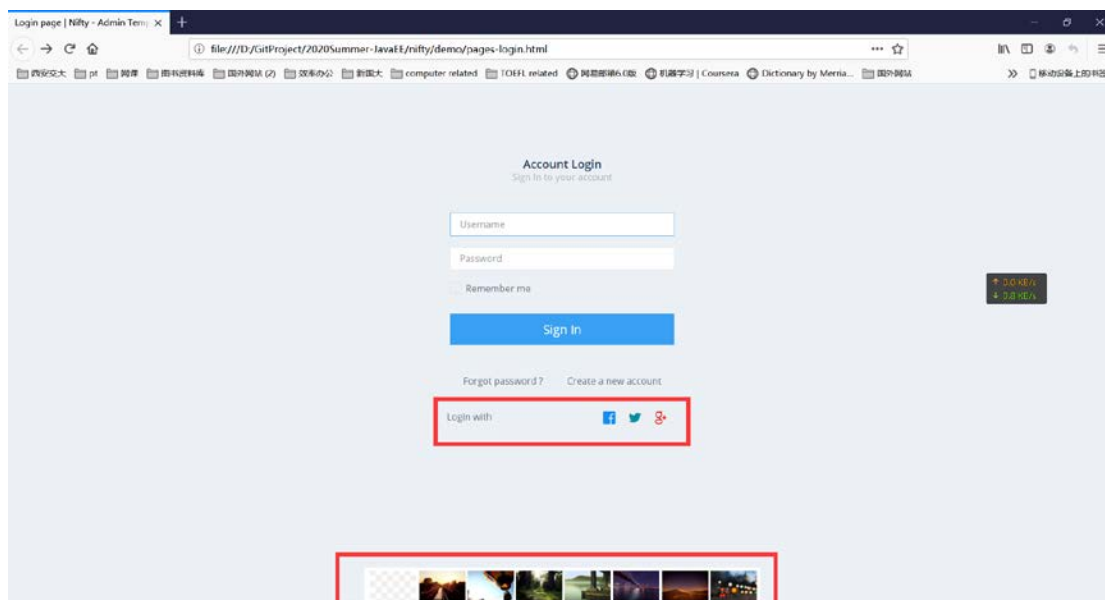


图 2.1.1-1 登录界面

注：红框标识的内容先不实现。

接口路径见本文档中 [Login](#) 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

3.1.2 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

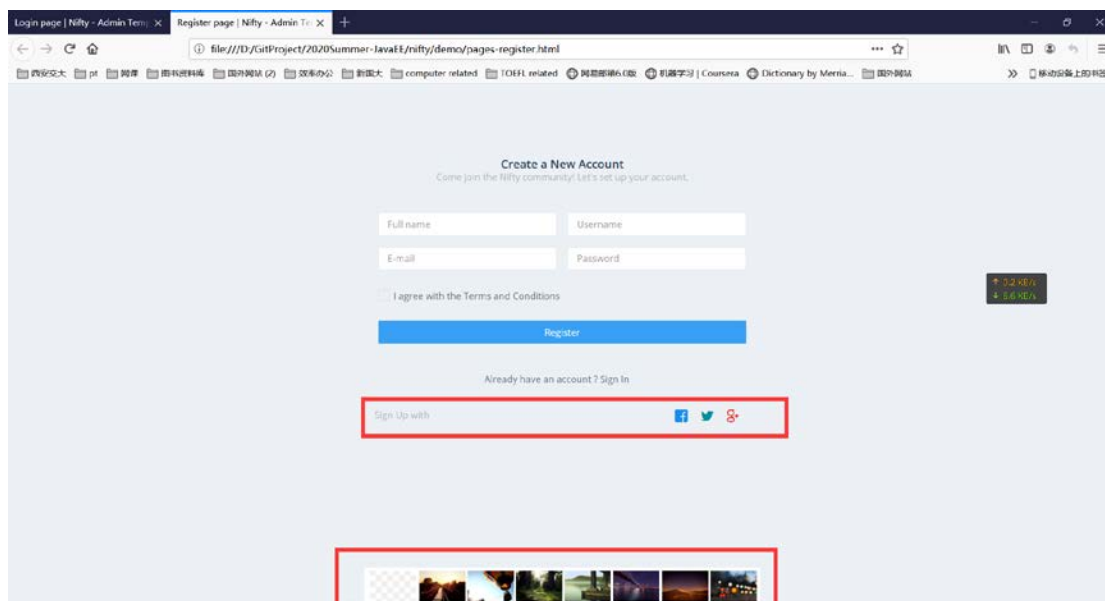


图 2.1.2-1 用户注册界面

注：红框标识的内容先不实现。

注册接口见接口定义中 [Register](#) 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。

3.1.3 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

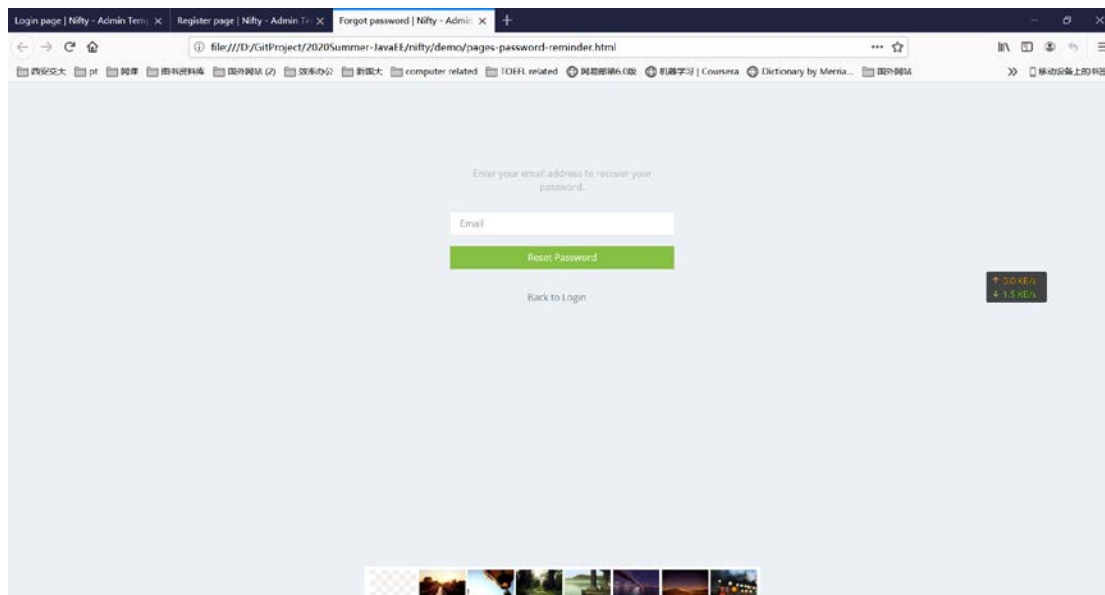


图 2.1.3-1 密码找回界面

接口见接口定义中给 [PassworReminder](#) 接口

3.2 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](#) 接口。

DashBoard 详细设计如下。

3.2.1 侧边栏详细设计

3.2.1.1 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

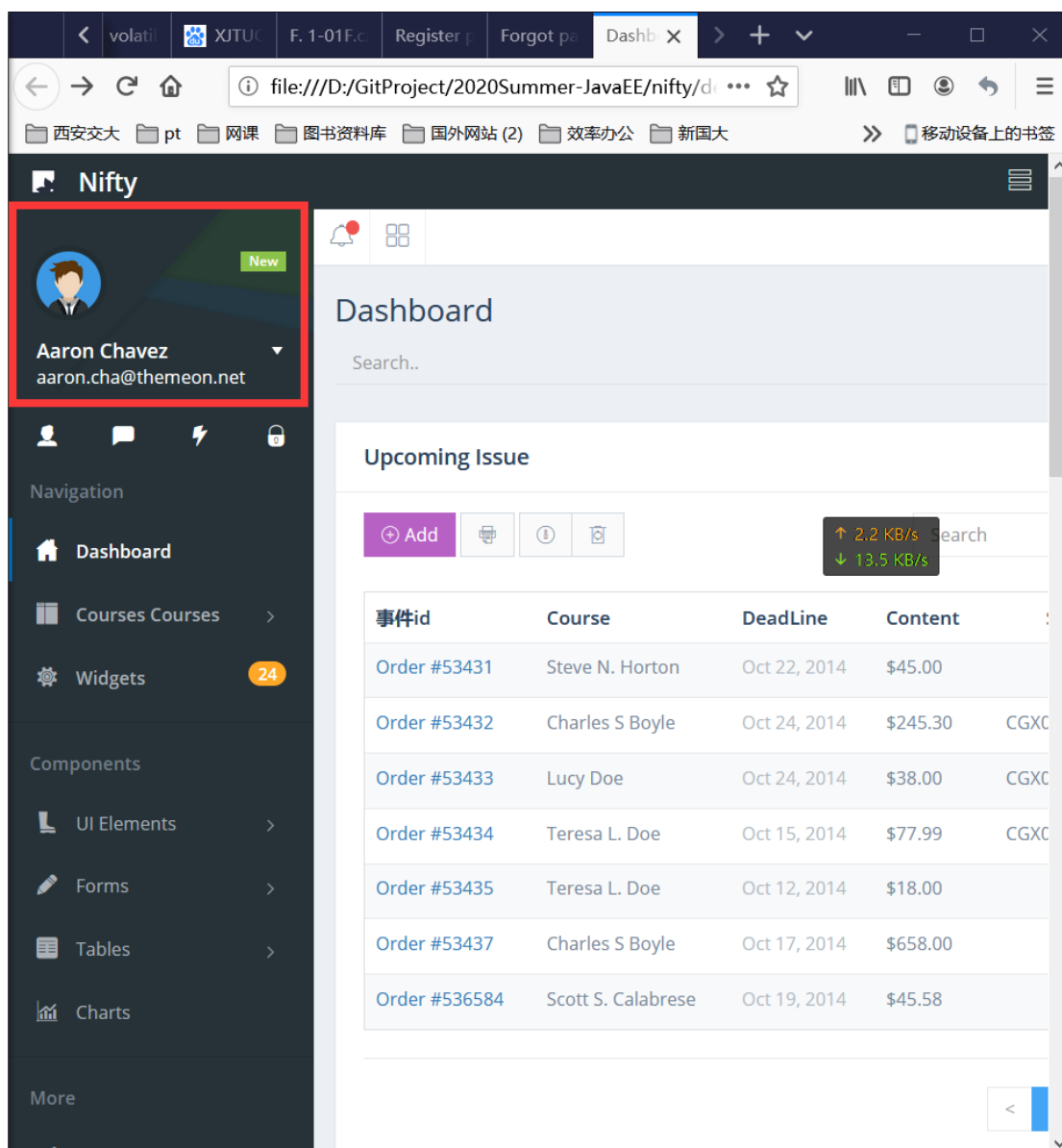


图 2.2.1.1-1 侧边栏用户头像、昵称、邮箱位置

3.2.1.2 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

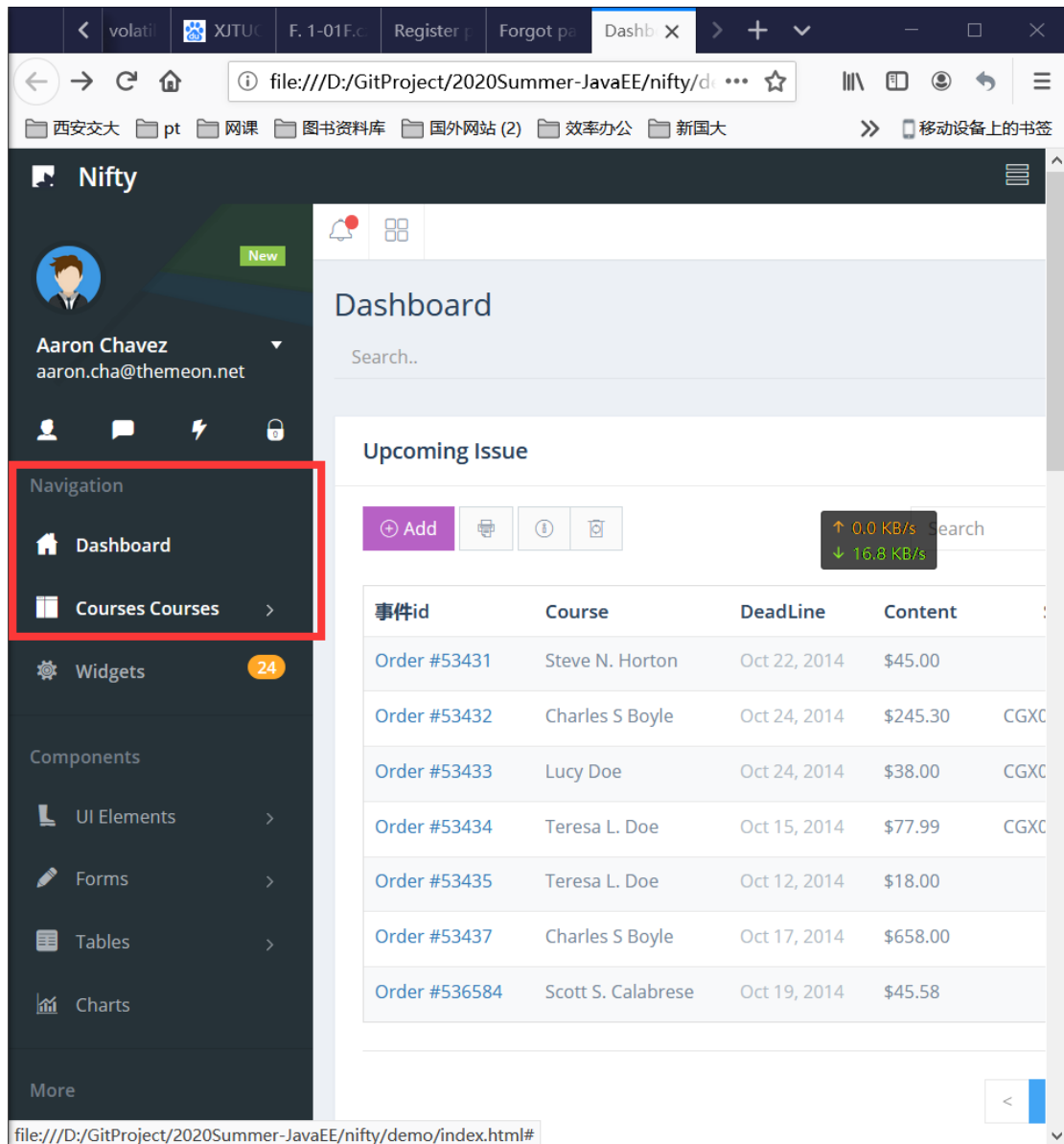


图 2.2.1.2-1 Navigation 栏位置

3.2.1.3 Courses 栏下内容

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

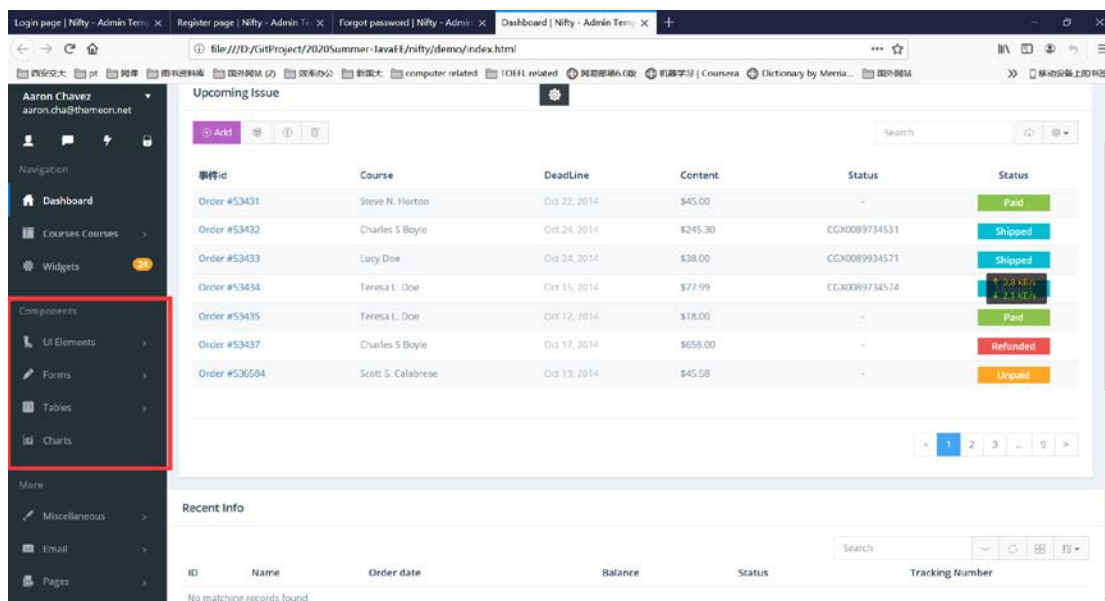


图 2.2.1.3-1 Courses 栏位置

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。详见 DashboardMenu

剩余侧边栏元素暂不考虑。

3.2.2 Upcoming Issue 详细设计

Upcoming Issue 位置见图 2.2.2-1。

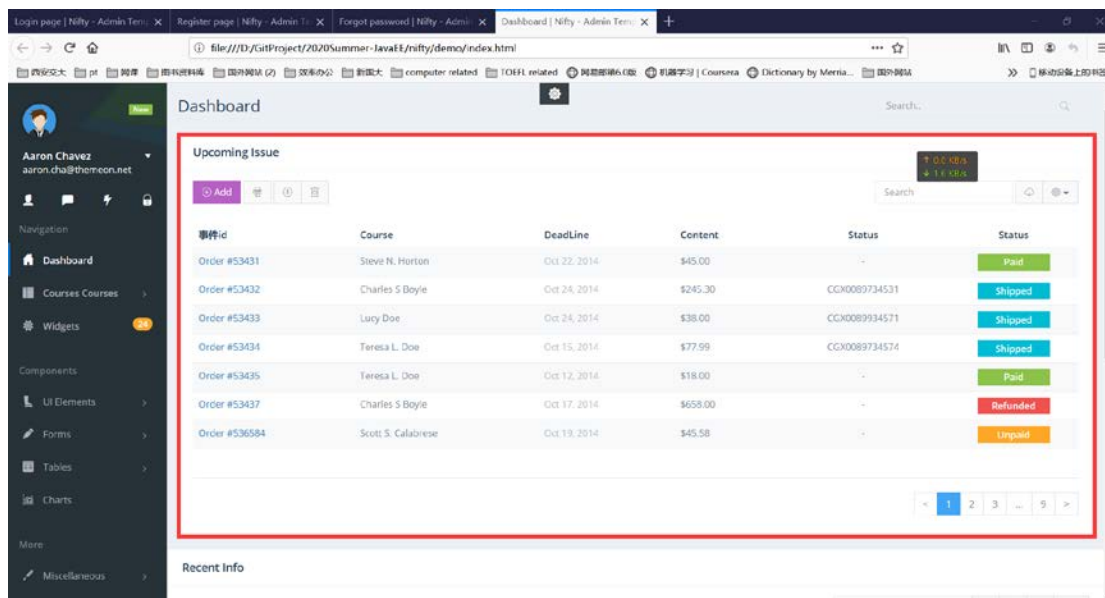


图 2.2.2-1 Upcoming Issue 表格位置

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](#) 接口。

Upcoming Issue

Add	Edit	Delete	Refresh	Search	Reset	Filter
事件id	Course	DeadLine	Content	Status	Status	
Order #53431	Steve N. Horton	Oct 22, 2014	\$45.00	-	Paid	
Order #53432	Charles S Boyle	Oct 24, 2014	\$245.30	CGX0089734531	Shipped	
Order #53433	Lucy Doe	Oct 24, 2014	\$38.00	CGX0089934571	Shipped	
Order #53434	Teresa L. Doe	Oct 15, 2014	\$77.99	CGX0089734574	Shipped	
Order #53435	Teresa L. Doe	Oct 12, 2014	\$18.00	-	Paid	
Order #53437	Charles S Boyle	Oct 17, 2014	\$658.00	-	Refunded	
Order #536584	Scott S. Calabrese	Oct 19, 2014	\$45.58	-	Unpaid	

< 1 2 3 ... 9 >

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。

3.2.3 Recent Info 详细设计

Recent Info 位置 见图 3.2.3-1。

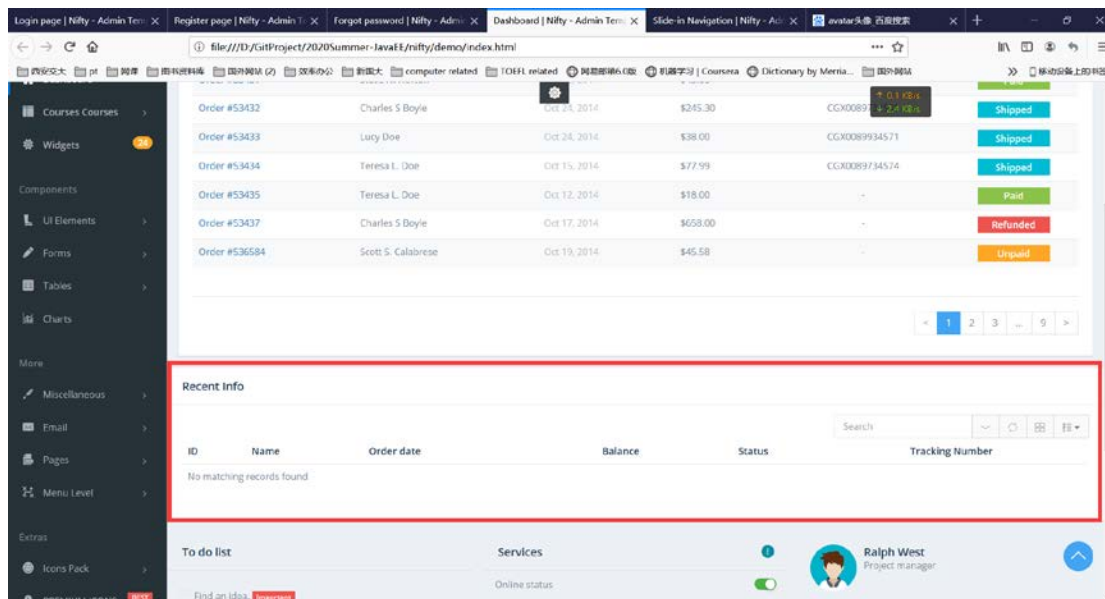


图 3.2.3-1 Recent Info 表位置

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](#) 项。

3.3 PREVCOURSE 界面

待定。

3.4 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。

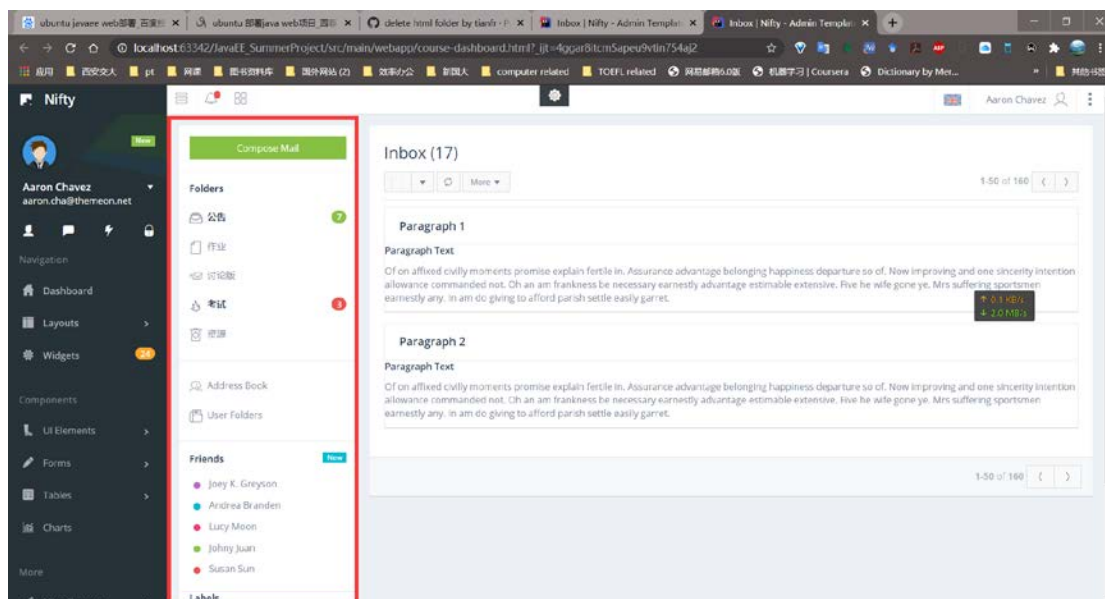


图 3.4-1 课程侧边栏展示图

- 课程内容详情部分：位置如图 3.4-2 所示。

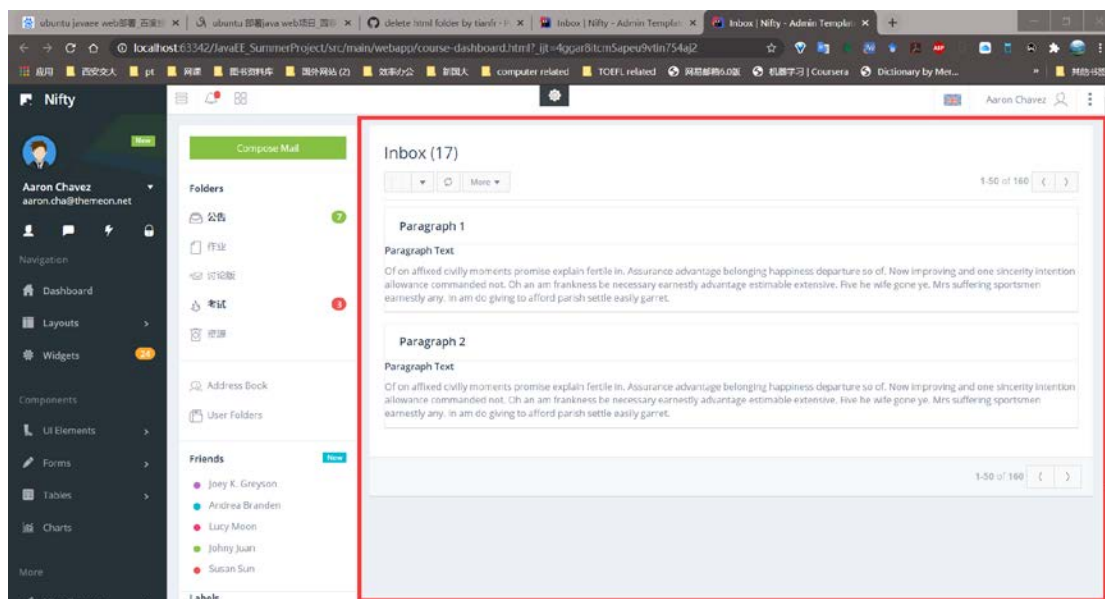


图 3.4-2 课程内容详情部分位置

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.

3.4.1 课程侧边栏详细设计

3.4.1.1 Folders 部分

Folders 侧边栏位置见图 3.4.1.1-1。

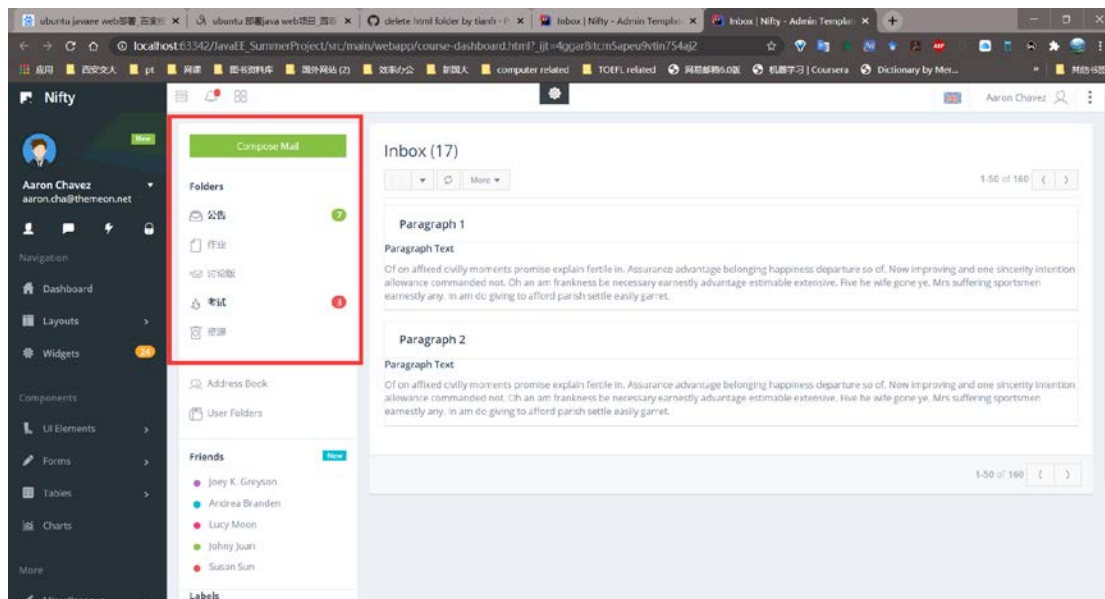


图 3.4.1.1-1 侧边栏 Folders 部分

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。

3.4.1.2 Fouldeer 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 [GetCoursePanel1](#) 接口。

3.4.2 课程内容详情页设计

3.4.2.1 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。

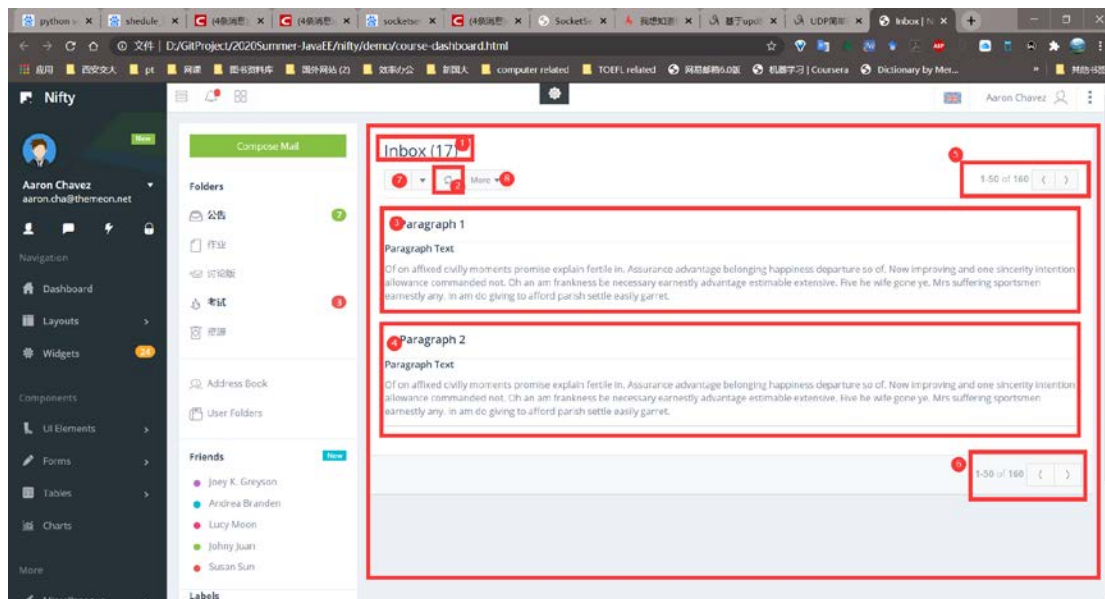


图 3.4.2.1-1 公告详情页

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](#) 接口。

用户可通过点击标号 3, 4 部分内容进入公告具体展示界面

3.4.2.1.1 公告展示具体设计

公告展示具体设计见图 3.4.2.1.1-1

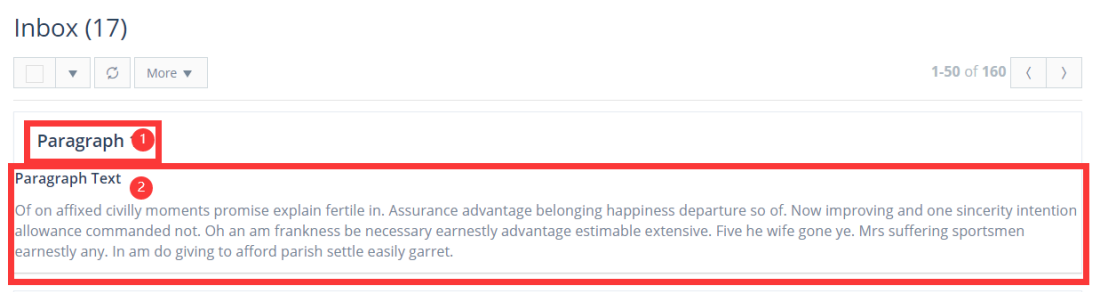


图 3.4.2.1.1-1 具体公告内容示意图

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。

3.4.2.1.2 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。

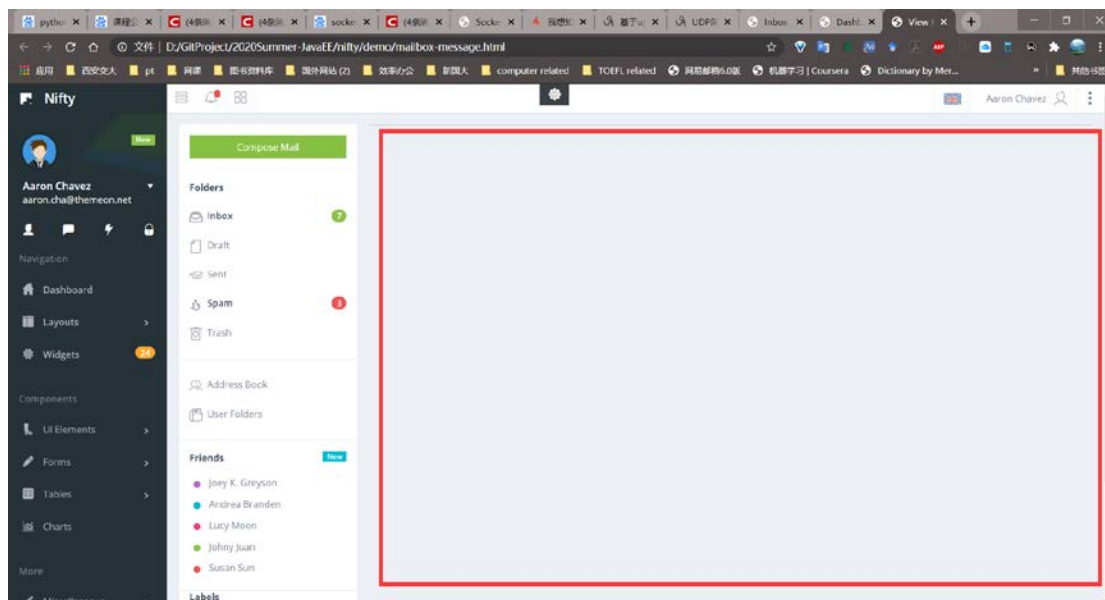


图 3.4.2.1.2-1 具体课程公告页位置

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。

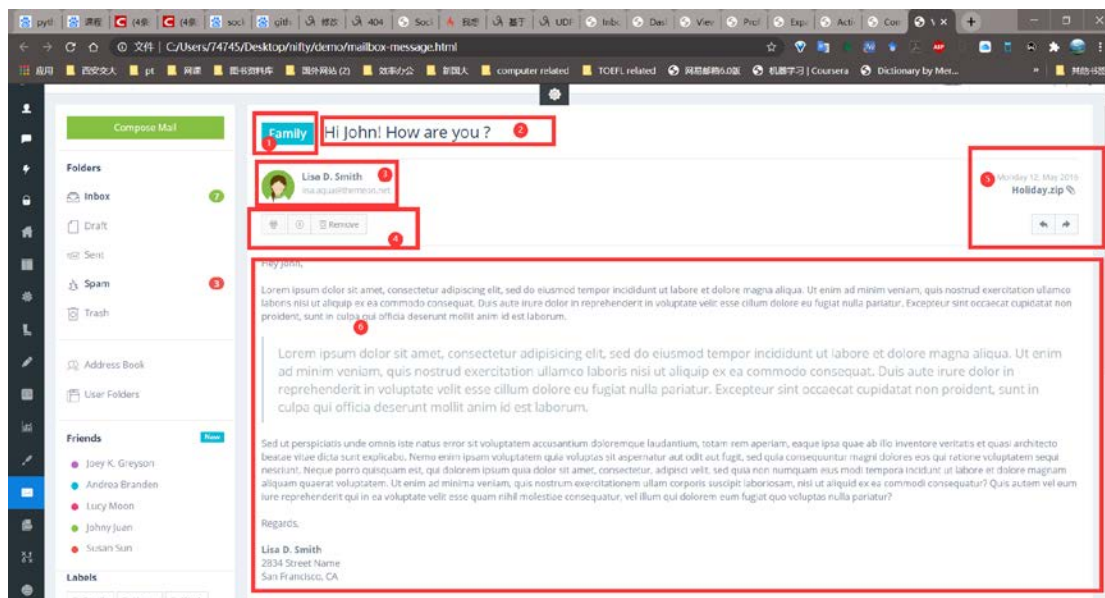


图 3.4.2.1.2-2 公告展示内容具体设计

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。

3.4.2.2 作业详情页设计

3.4.2.2.1 作业展示具体设计

用户通过点击课程界面上的作业按钮，进入作业详情界面，作业详情界面的位置如图 3.4.2.2.1-1 所示。

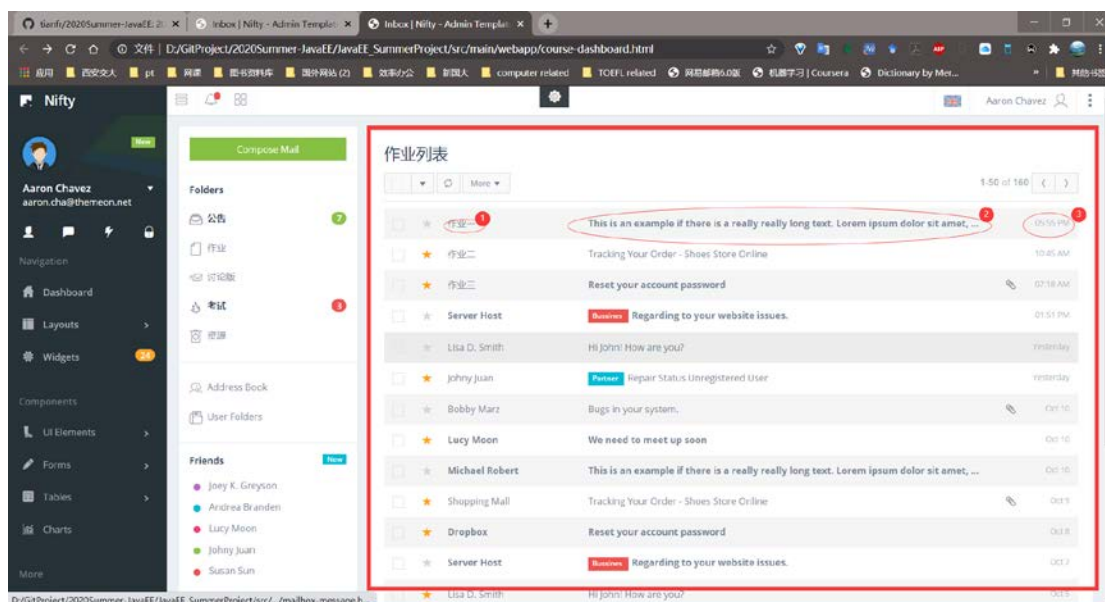


图 3.4.2.2.1-1 作业详情页设计

图中标号 1 位置显示作业的 title，字体加粗显示，标号 2 位置显示作业的实际内容，标号 3 的位置 显示作业的截止日期，格式为：“2020-7-10 23:59 截止”或“DDL:2020-7-10 23:59”

标号 1，2，3 位置的信息通过向后台查询实现，其接口详细设计参加 [GetCourseHomeworks](#) 接口。

用户可以通过点击作业的 title 或作业实际（即标号 1，2 处内容）内容进入课程作业界面。

3.4.2.2.2 课程作业界面具体设计

用户点击某个作业的 title 或实际作业内容进入课程作业界面，课程作业界面位置如图 3.4.2.2.2-1 所示。其显示的内容需要向后端查询，查询格式见 [GetHomework 接口](#)。

同样的，该界面来自 mailbox-message.html，我已放在群文件-更新的文件文件夹中。

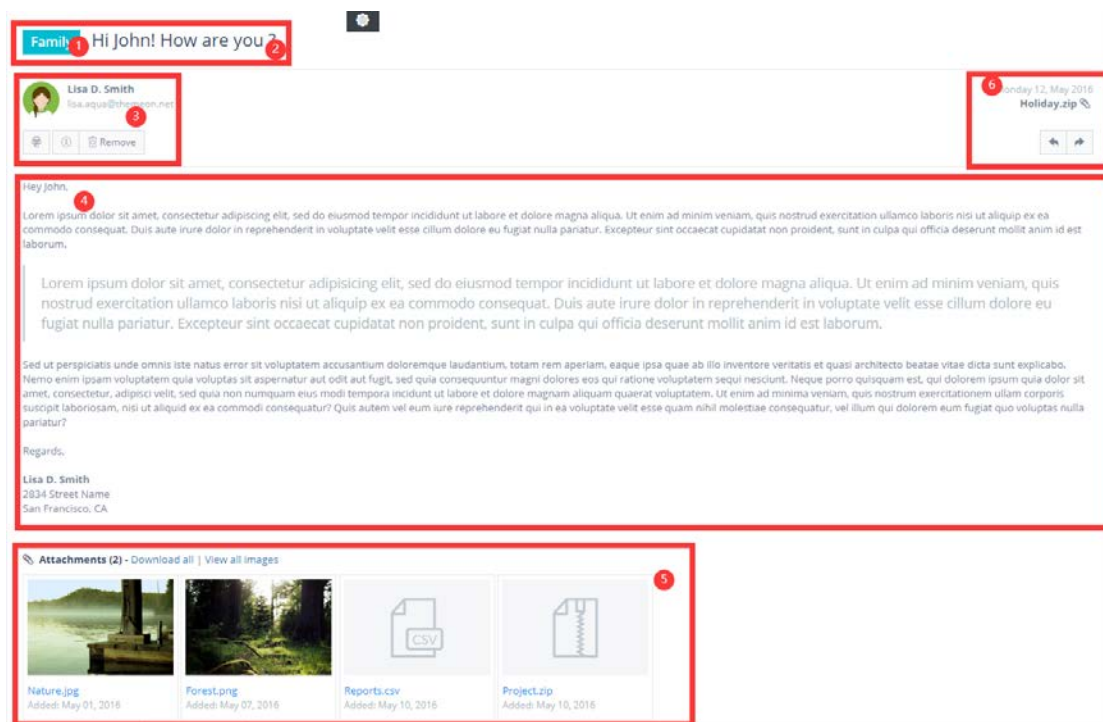


图 3.4.2.2.2-1 课程作业界面具体设计图

- 标号 1 处文字改为公告；
- 标号 2 处文字改为作业具体标题；
- 标号 3 处文字改为发布作业教师名字，教师邮箱，教师头像，下面的三个按钮改为作业提交按钮，作业提交按钮来自 ui-buttons.html 文件，使用图 3.4.2.2.2-2 所示按钮，即图中紫色按钮，用户点击作业提交按钮进入作业提交界面；

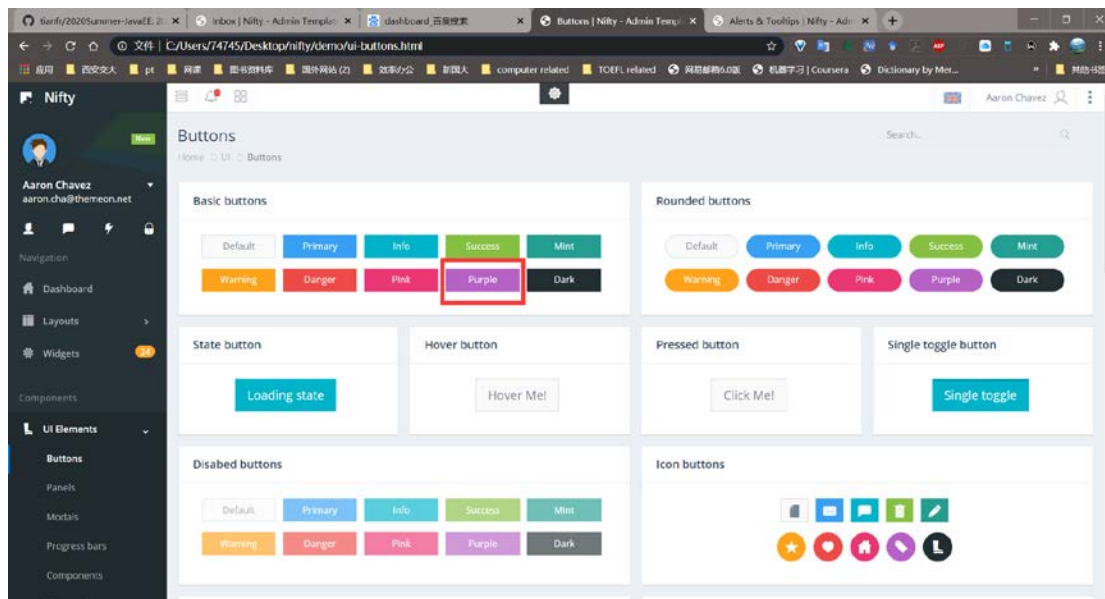


图 3.4.2.2.2-2 作业提交按钮

- 标号 4 处显示作业正文；
- 标号 5 处显示作业附件，附件显示附件名称，删除显示附件添加事件的信息，附件具体信息从后端接口处查询，在 [GetHomework 接口](#) 中包含了作业附件信息，用户通过点击附件图标跳转到附件路径位置实现附件下载，标号 5 上端 View all images, Download all 按钮先不实现；
- 标号 6 处显示作业发布日期。

在标号 3, 4 之间加一行文字，文字格式参照 ui-alerts-tooltips.html 文件中 Basic Example 格式，具体要求见图 3.4.2.2.2-3.

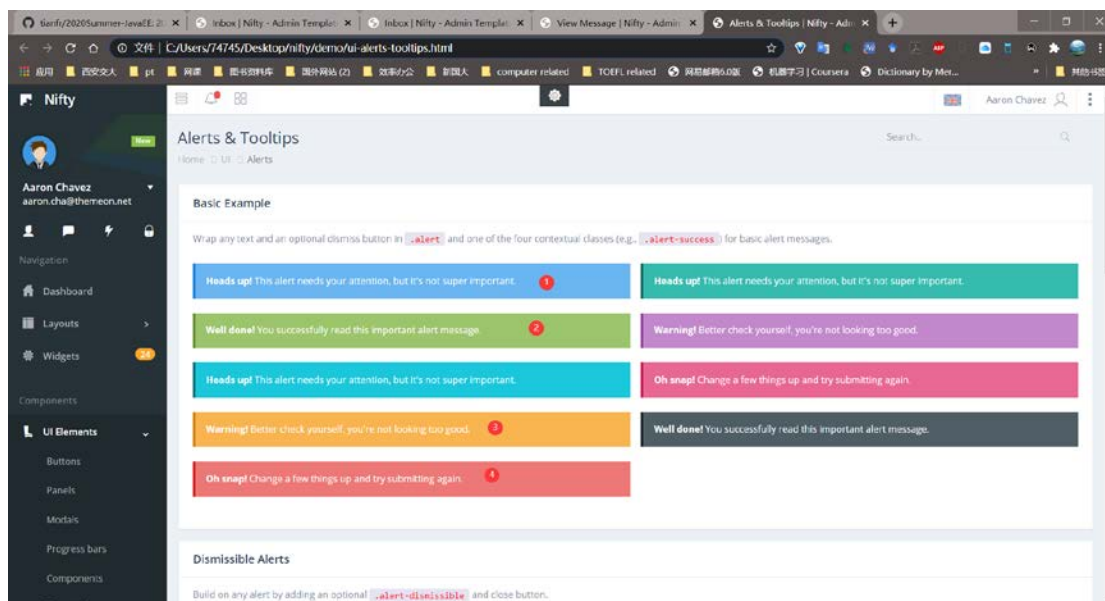


图 3.4.2.2.2-3 作业截止日期行实例

该行显示作业的具体截止日期，参照上图，其依据不同的状态显示不同颜色：

- 当作业在截止日期前一天及以上，且未提交，显示蓝色的背景色，如标号 1 所示；
- 当作业在截止日期前一天内，且未提交，显示黄色背景色，如标号 3 所示；
- 当作业在截止日期前成功提交，显示绿色背景色，如标号 2 所示；
- 当作业超出截止日期，显示红色背景色，如标号 4 所示。

3.4.2.2.3 课程作业提交界面具体设计

用户在课程作业界面点击提交按钮向后端发送数据查询作业提交情况。用户进入课程作业提交界面时，需向后端查找是否有作业草稿保存以及课程作业是否已经提交，查找接口格式见 [GetExistedHomework](#) 接口。

作业提交界面具体设计参考 mailbox-compose.html，如图 3.4.2.2.3-1 所示。

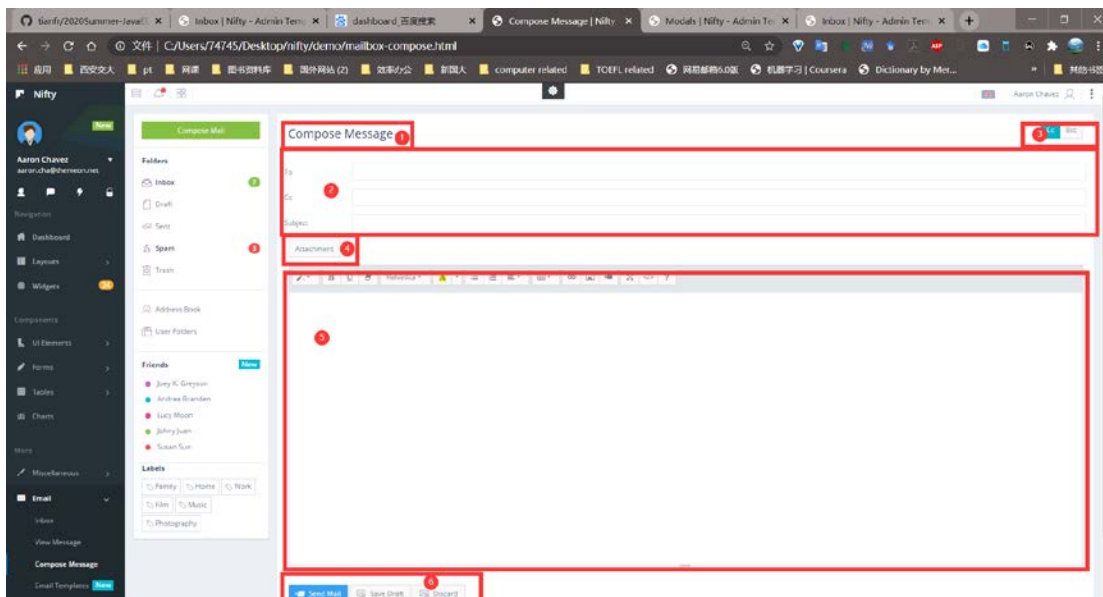


图 3.4.2.2.3-1 作业文本提交示意图

图中标号 2, 3 处删除，图中标号 1 处文字改为作业提交，标号 5 处保留，标号 6 处三个按钮文字改为 Submint, Save Draft, Discard,并实现以下功能：

- Commit 按钮：提交作业向后端发送数据内容，格式见 [SubmitHomework](#) 接口，如果作业提交成功则给出作业提交成功提示（详细设计见下文），如果作业提交失败，则在该界面不动并提示作业提交失败和失败原因；
- SaveDraft 按钮：向后端发送数据内容，格式见 [SubmitHomework](#) 接口；
- Discard 按钮：提示是否放弃草稿，如果放弃则返回前一个网页。

。

在标号 6 处的三个按钮右侧显示当前作业状态，状态显示方式使用 ui-alerts-tooltips.html 中的 Basic Example 样式，参考图 3.4.2.2.3-2：

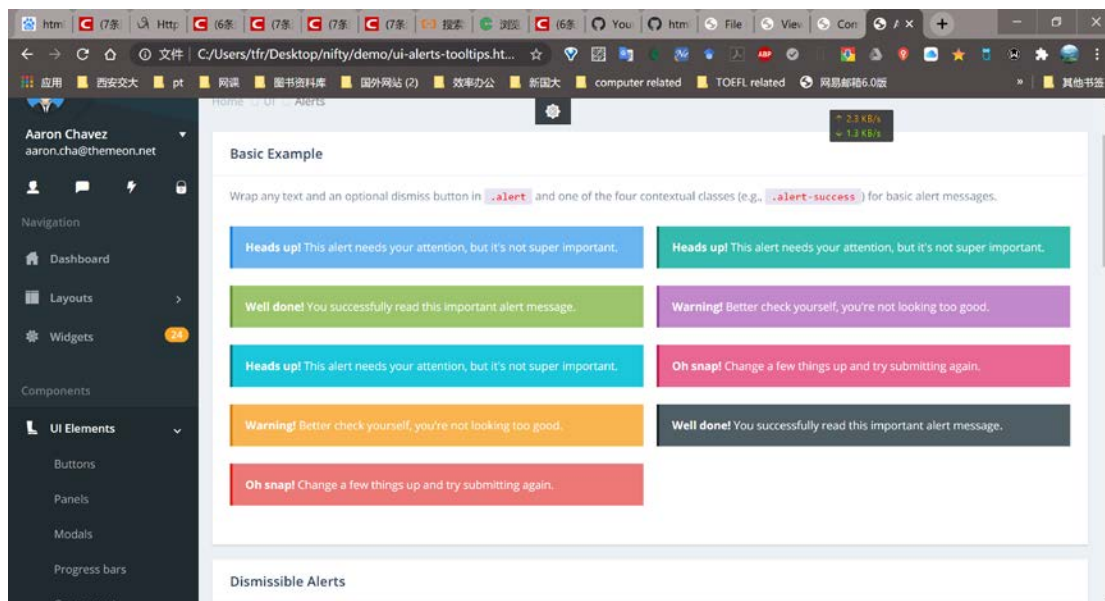


图 3.4.2.2.3-2 作业提交状态警示栏

- 若作业已提交或用户点击作业提交按钮提交作业，服务器收到作业并保存成功后，显示作业已提交，提交时间为 2020-05-12 12:00:23 字样（Well done! Your homework has been submitted, SUBMITTED TIME: 2020-05-12 12:00:23.），使用深绿色背景色；
- 若作业未提交并且未超出截止时间，显示您需要在 2020-05-23 23:59 前提交作业字样（Deadline of this homework is 2020-05-23 23:59.），使用黄色背景色；
- 若作业未提交但超出了截至时间，显示该项作业已过期字样（This homework has been expired.），使用浅红色背景色；
- 若用户点击作业草稿保存按钮提交作业草稿，服务器收到作业并保存成功后，显示作业成功保存，保存时间为 2020-05-12 12:00:23 字样（Your homework draft has been saved. SAVED TIME: 2020-05-12 12:00:23），使用蓝色背景色。

标号 4 处文件上传按钮删除。

在标号 5 和标号 6 之间加入一个文件上传的 panel，panel 具体详见 forms-file-upload.html 文件中图 3.4.2.2.3-3 所示样式。

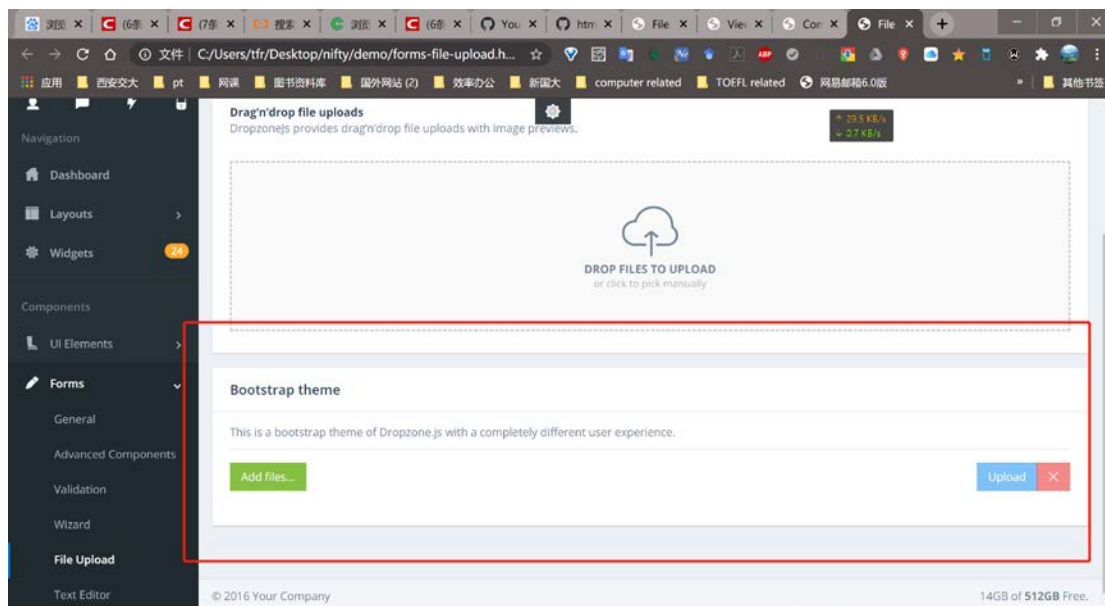


图 3.4.2.2.3-3 文件上传面板样式

其中，点击 Add Files 按钮选择上传的文件，选中文件后判断文件是否合法，合法后立即上传，右侧 Upload 按钮删除，在文件下面显示上传进度，如图 3.4.2.2.3-4 所示。文件上传需要调用 [FileUpload](#) 接口，上传成功后返回文件在浏览器保存路径。

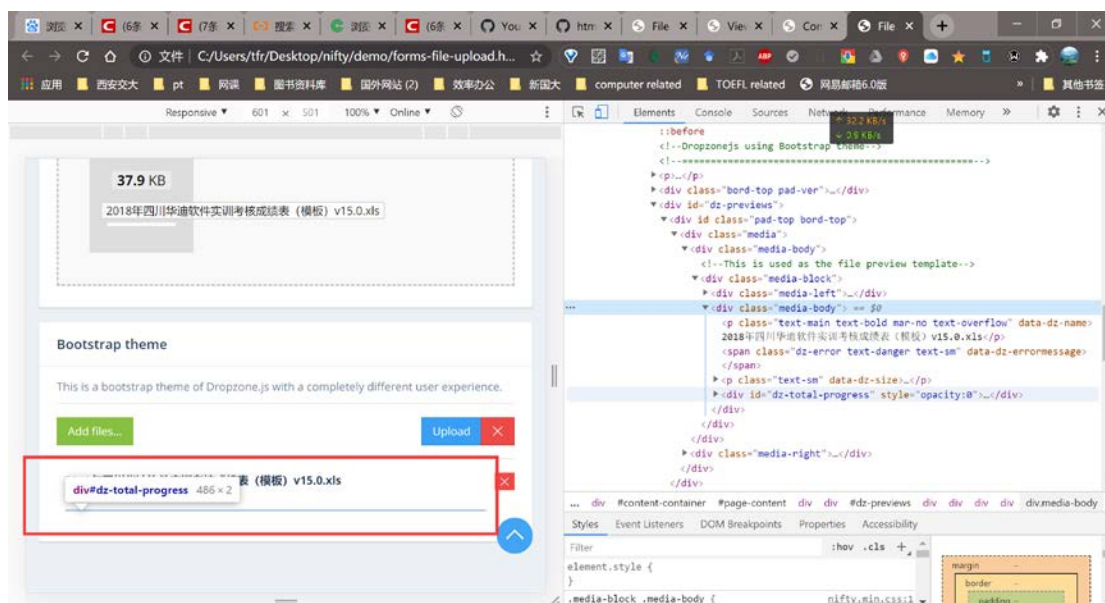


图 3.4.2.2.3-4 文件上传进度

用户可以通过点击右面每个文件红色删除按钮实现文件删除，或通过点击右上方大红色按钮实现文件删除，点击删除按钮后，浏览器向后端发送文件删除请求，即调用 [FileDelete](#) 接口，服务器删除文件后返回删除成功或失败，如果删除成功，则该表项栏目移除，如果删除失败，则提示用户删除失败。

3.4.2.3 讨论吧详情页设计

3.4.2.4 考试详情页设计

3.4.2.5 资源详情页设计

4 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

5 网页说明——管理员用户

6 接口定义

6.1 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。

6.2 LOGIN 接口

该接口出现在 [3.1.1 节登录界面](#)。

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
}
```

6.3 REGISTER 接口

该接口出现在 [3.1.2 节注册界面](#)。

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  
}
```

6.4 PASSWORDREMINDER 接口

该接口出现在 [3.1.3 节密码找回界面](#)。

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  
}
```

6.5 DASHBOARDMENU 接口

该接口出现在 3.2.1.3 节 [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_id": course_id

    },

    {

        "id": "course2",

        "course_name": course_name,

        "course_id": course_id

    },

    {

        "id": "course3",

        "course_name": course_name,

        "course_id": course_id

    },//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

6.6 UPCOMINGISSUE 接口

该接口出现在 [3.2.2 节 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
]
}

```

6.7 RECENTINFO 接口

该接口出现在 [3.2.3 节 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

]

}

```

6.8 GETCOURSEPANEL 接口

该接口出现在 [3.4.1.2 节 Folder 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_id": course_id,  
}),
```

反向接口格式

```
{  
    "id": "course_panel",  
    "value": [  
        {  
            "panel_id": panel_id,  
            "panel_name": panel_name,  
        },  
        {  
            "panel_id": panel_id,  
            "panel_name": panel_name,  
        },  
    ],  
    //项目数量不定，需要遍历访问
```



```
    ]
}
```

6.9 GETCOURSEANNOUNCEMENTS 接口

该接口出现在 [3.4.2.1 节公告详情页设计](#)。

接口路径：

```
/GetCourseAnnouncements
```

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_announcements",
    "course_id": course_id,
}),
```

反向接口格式

```
{
    "id": "course_announcements",
    "total_num": num_of_total_announcements,
    "value": [
        {
            "announcement_title": announcement_title, //公告标题
```

"announcement_id": announcement_id, //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_id": announcement_id,

"announcement_content": announcement_content,

},

//项目数量不定, 需要遍历访问

]

}

6.10 GETANNOUNCEMENT 接口

该接口出现在 [3.4.2.1.2 节课程公告页设计](#)。

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_id": course_id,

"announcement_id": announcement_id //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": announcement_datetime, //教师发布该公告的时间

}
```

6.11 GETCOURSEHOMEWORKS 接口

该接口出现在 [3.4.2.2.1 作业展示具体设计节](#)。

接口路径：

```
/GetCourseHomeworks
```

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_homeworks",

"course_id": course_id,

}),
```

反向接口格式

```
{

  "id": "course_homeworks",

  "total_num": num_of_total_homeworks,

  "value": [

    {

      "homework_title": homework_title, //作业标题

      "homework_id": homework_id, //作业访问具体路径，用户点击作业标题后向后端接口
      发送该路径进入公告具体信息页

      "homework_content": homework_content, //作业内容

      "homework_deadline": homework_deadline, //作业截止日期

    },

    {

      "homework_title": homework_title, //作业标题

      "homework_id": homework_id, //作业访问具体路径，用户点击作业标题后向后端接口
      发送该路径进入公告具体信息页

      "homework_content": homework_content, //作业内容

      "homework_deadline": homework_deadline, //作业截止日期

    },

    //项目数量不定，需要遍历访问

  ]

}
```

6.12 GETHOMEWORK 接口

该接口出现在 [3.4.2.2.2 课程作业界面具体设计](#) 节中。

接口路径:

```
/GetHomework
```

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_homework",
    "course_id": course_id,
    "homework_id": homework_id, //注意这个变量在 GetCourseHomeworks 中出现过
}),
```

反向接口格式

```
{
    "homework_title": homework_title, //作业标题
    "author": teacher_name, //作业发布教师名称
    "author_email": teacher_email, //作业发布教师邮箱
    "author_avatar": authon_avatar, //默认为初始头像
    "homework_deadline": datetime //作业的 ddl
    "homework_content": announcement_details, //这里有点争议, 我设想返回的是 html 格式文档
    "homework_attachment": [
        {
```

```

        "attachment_path": path_to_attachment_in_webserver, //用户通过点击附件图
        标跳转到该路径实现附件下载

        "attachment_name": attachment_name,

    },

    {

        "attachment_path": path_to_attachment_in_webserver,

        "attachment_name": attachment_name,

    },

    {

        "attachment_path": path_to_attachment_in_webserver,

        "attachment_name": attachment_name,

    },

    //项数不定需要遍历，如果没有附件，列表为空。

]

    "datetime": annoucement_datetime, //教师发布该公告的时间

}

```

6.13 作业详情页接口

6.13.1 SubmitHomework 接口

该接口出现在 [3.4.2.2.3 课程作业提交界面具体设计](#) 节中。

接口路径：

```
/SubmitHomework
```

接口 js 函数路径

```
Js/myjs.js?v=1
```

正向接口格式：

```
data: JSON.stringify({
```

```

    "status": "0",

    "message": "submit_homework",

    "course_id": course_id,

    "homework_id": homework_id, //注意这个变量在 GetCourseHomeworks 中出现过

    "is_draft": "yes" //yes 表示草稿, no 表示提交作业

    "homework_content": homework_content, //用户在网页上写的内容保存到这里, html 格式

    "have_attachments": "yes" //yes 表示有上传附件, no 表示没有上传附件

    "homework_attachments": [

        {

            "file_name": file_name,

            "file_path": file_path, //在 FileUpload 接口获取的文件路径

        },

        {

            "file_name": file_name,

            "file_path": file_path, //在 FileUpload 接口获取的文件路径

        },

        {

            "file_name": file_name,

            "file_path": file_path, //在 FileUpload 接口获取的文件路径

        },

        //文件数量不固定

    ]

    },

```

反向接口格式

```

{

    "success_value": 0, //1 表示成功, 0 表示失败

    "datetime": datetime, //作业成功提交的具体时间

```

```
"fail_content": "错误详细信息，由后端返回", //错误详细信息：服务器连接失败，删除失败，
服务器错误.....
}
```

6.13.2 GetExistedHomework 接口

该接口出现在 [3.4.2.2.3 课程作业提交界面具体设计](#)节。

接口路径：

```
/GetHomeworkDraft
```

接口 js 函数路径：

```
Js/myjs.js?v=1
```

正向接口设计

```
data: JSON.stringify({
    "status": "0",
    "message": "get_existed_homework",
    "course_id": course_id,
    "homework_id": homework_id,
}),
```

反向接口设计

```
{
    "homework_status": "drafted, submitted, null, expired"//drafted 表示已保存草稿，s
ubmitted 表示已提交， null 表示未提交无草稿， expired 表示作业超过截至时间
    "homework_content":homework_content, //用户在网页上写的内容保存到这里，html 格式，若
状态为 null 或状态 expired 并且未提交作业此项为空，
    "submitted_time": datetime, //作业草稿或提交作业的具体时间
```



```

    "homework_deadline": datetime, //该作业的 ddl

    "have_attachments": "yes" //yes 表示有上传附件, no 表示没有上传附件

    "homework_attachments": [

        {

            "file_name": file_name,

            "file_path": file_path, //在 FileUpload 接口获取的文件路径

        },

        {

            "file_name": file_name,

            "file_path": file_path, //在 FileUpload 接口获取的文件路径

        },

        {

            "file_name": file_name,

            "file_path": file_path, //在 FileUpload 接口获取的文件路径

        },

        //文件数量不固定, 若状态为 null 或状态 expired 并且未提交作业此项为空

    ]

}

```

6.14 文件上传下载相关接口

本项接口有争议：如何实现浏览器实时显示上传进度？

6.14.1 FileUpload 接口

该接口出现在 [3.4.2.2.3 课程作业提交界面具体设计](#) 节中。

接口路径：

/FileUpload

接口 js 函数路径:

```
Js/myjs.js?v=1
```

限制: 若是作业附件上传, 单个文件大小不超过 1024MB。

正向接口格式:

```
data: JSON.stringify({
    "status": "0",
    "message": "file_upload",
    "role": "student\teacher", //学生在 homework 界面上传文件则为学生角色, 若老师上传文件
    为老师角色
    "course_id": course_id, //学生或教师上传文件时所在的课程的课程名
    "related_id": related_id, //若为学生角色, 则该项存储 homework_id, 若为教师角色, 该项
    存储 XXXX_id
    "file_name": file_name,
    "file_content": file_content, //不确定是否应该这么写
}),
```

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: 服务器连接失败, 上传失败,
    服务器错误, 文件过大, 文件类型不支持.....
    "file_path": file_path, //若上传成功, 该项返回文件在服务器保存的路径, 若失败则为 null
}
```

6.14.2 FileDelete 接口

该接口出现在 [3.4.2.2.3 课程作业提交界面具体设计](#)节中。

接口路径:

```
/FileDelete
```

接口 js 函数路径:

```
Js/myjs.js?v=1
```

正向接口格式:

```
data: JSON.stringify({
    "status": "0",
    "message": "file_delete",
    "role": "student\teacher", //学生在 homework 界面上传文件则为学生角色，若老师上传文件
    为老师角色
    "course_id": course_id, //学生或教师上传文件时所在的课程的课程名
    "related_id": related_id, //若为学生角色，则该项存储 homework_id，若为教师角色，该项
    存储 XXXX_id
    "file_name": file_name,
    "file_path": file_path, //上传文件成功时返回的文件路径
}),
```

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：服务器连接失败，删除失败，
    服务器错误.....
}
```