



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、向苒霄、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？
1.0	田丰瑞	喜大普奔！ 3.4.2.5 章节编写完成； 6.14 文件上传下载接口完善完毕； 6.15 接口编写完成 学生用户功能大体实现	3.4.2.3, 3.4.2.4, 3.3 需要编写完教师用户基本功能后实现 文件上传下载进度条？ 文件批量下载？
1.1	田丰瑞	更新 4.1 章节； 6.2 反向接口出现变更； 6.3 正向接口出现变更； 6.16 接口编写完毕 教师功能大体实现	考试，查看成绩，讨论区尚未实现 PrevCourse 界面尚未实现
1.2	田丰瑞	3.3 PrevCourse 章节编写完毕 6.17 接口编写完毕 6.6 反向接口添加 Issue 总数	考试，讨论界面尚未实现
1.3	田丰瑞	Sql 接口写到 6.10	
1.4	田丰瑞	Sql 接口写到 6.13	存储调用如何与 javaee 交互？
1.5	田丰瑞	除文件接口外 sql 接口全部写完	
1.6	田丰瑞	完善 4.4.2.2.2；	

		完善 6.18 接口;	
1.7	田丰瑞	4.4.2.2 基本完成, 6.19 接口完成	

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.....	4
2	用户角色总览.....	7
3	网页说明——学生用户.....	7
3.1	登录相关界面	7
3.1.1	登录界面.....	7
3.1.2	注册界面.....	7
3.1.3	密码找回界面.....	8
3.2	Dashboard 界面.....	9
3.2.1	侧边栏详细设计.....	9
3.2.2	Upcoming Issue 详细设计	12
3.2.3	Recent Info 详细设计	13
3.3	PrevCourses 界面.....	14
3.4	课程界面	16
3.4.1	课程侧边栏详细设计.....	17
3.4.2	课程内容详情页设计.....	18
4	网页说明——教师用户.....	28
4.1	登录相关界面	28
4.1.1	登录界面.....	28
4.1.2	注册界面.....	29
4.1.3	找回密码界面.....	30
4.2	Dashboard 界面.....	30
4.2.1	侧边栏设计.....	30
4.2.2	Upcoming Issue 详细设计	30
4.2.3	Recent Info 详细设计	30
4.3	PrevCourse 界面	30

4.4	课程界面	30
4.4.1	课程侧边栏详细设计	30
4.4.2	课程内容详情页设计	30
5	网页说明——管理员用户	36
6	接口定义	36
6.1	声明	36
6.2	Login 接口	36
6.3	Register 接口	38
6.4	PasswordReminder 接口	41
6.5	DashboardMenu 接口	43
6.6	UpComingIssue 接口	47
6.7	RecentInfo 接口	49
6.8	GetCoursePanel 接口	52
6.9	GetCourseAnnouncements 接口	53
6.10	GetAnnouncement 接口	55
6.11	GetCourseHomeworks 接口	58
6.12	GetHomework 接口	60
6.13	作业详情页接口	63
6.13.1	SubmitHomework 接口	63
6.13.2	GetExistedHomework 接口	67
6.14	文件上传下载相关接口	70
6.14.1	FileUpload 接口	70
6.14.2	FileDelete 接口	71
6.14.3	FileDownload 接口	72
6.14.4	FilesDownload 接口	73
6.15	GetCourseResources 接口	75
6.16	NewCourseIssue 接口	77

6.17	GetPrevCourses 接口.....	79
6.18	GetStudentHomeworks 接口	81
6.19	GetStudentHomeworkDetail 接口	83

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。

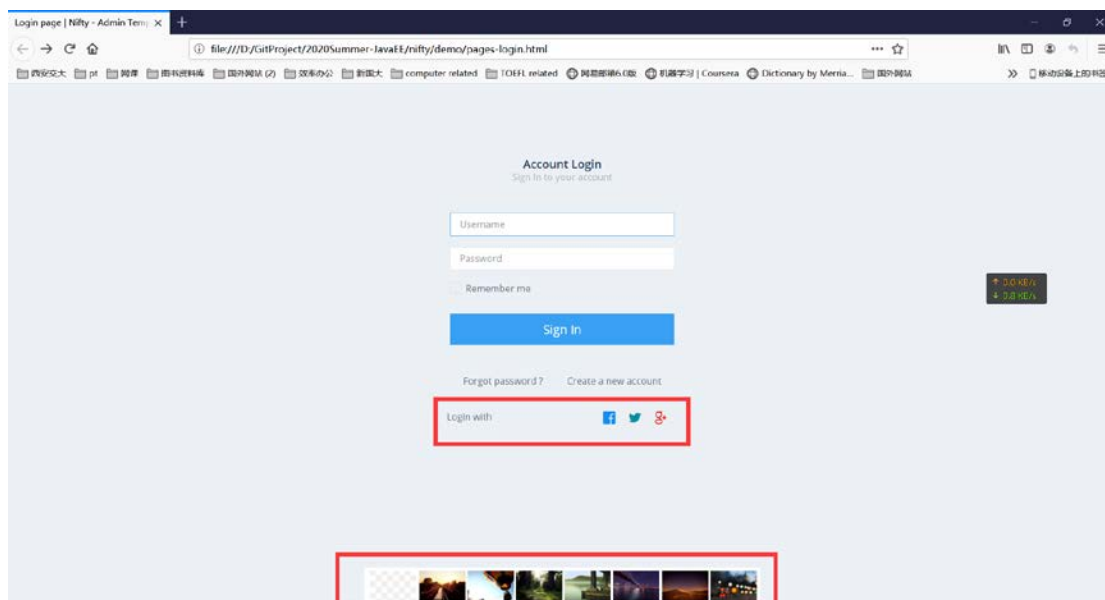


图 2.1.1-1 登录界面

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

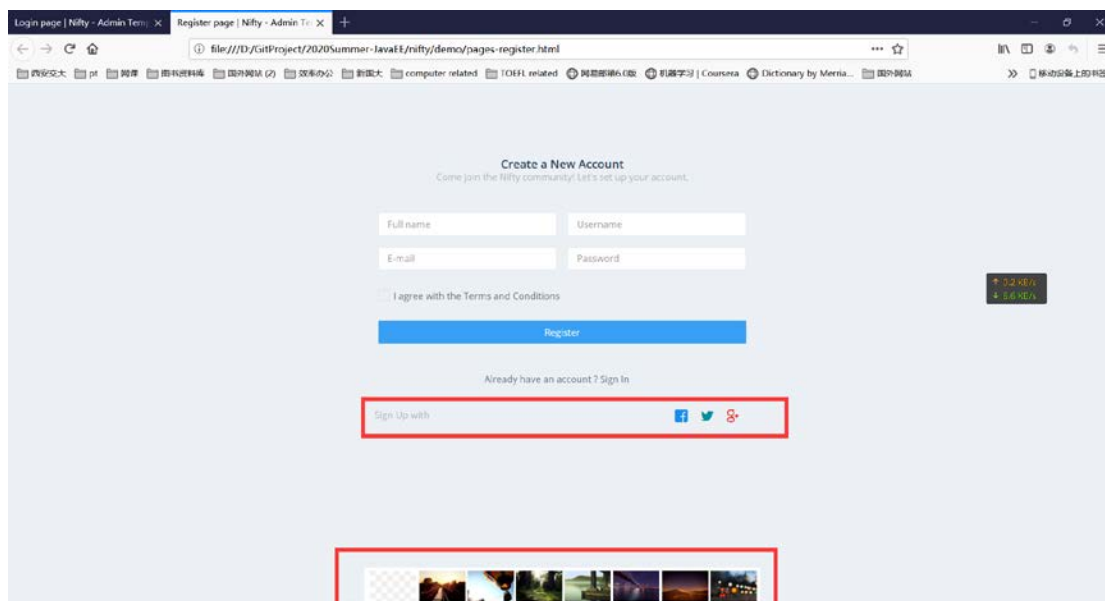


图 2.1.2-1 用户注册界面

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

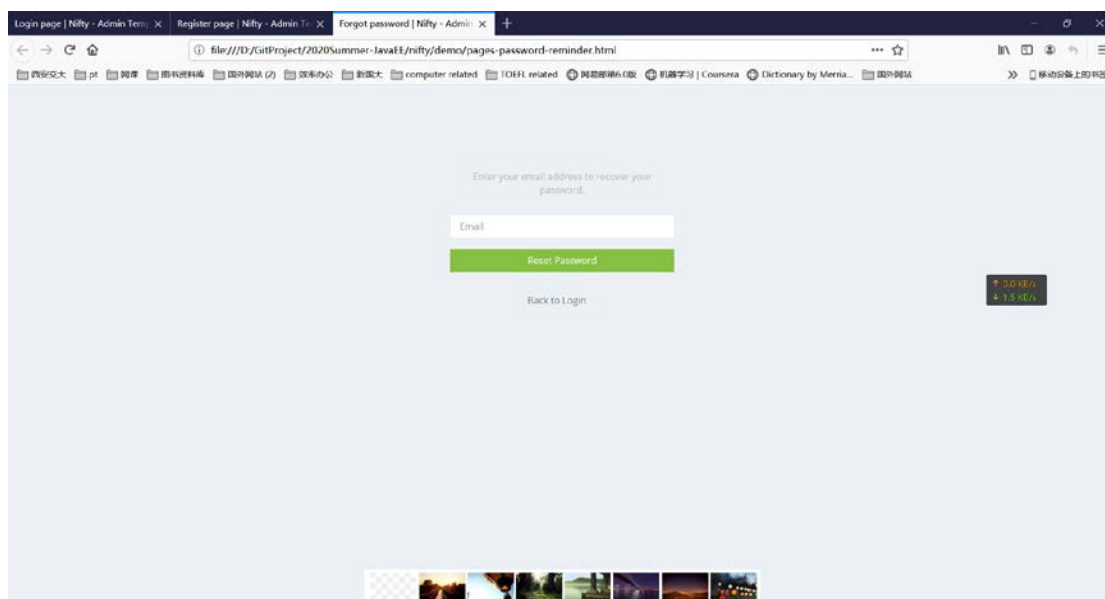


图 2.1.3-1 密码找回界面

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

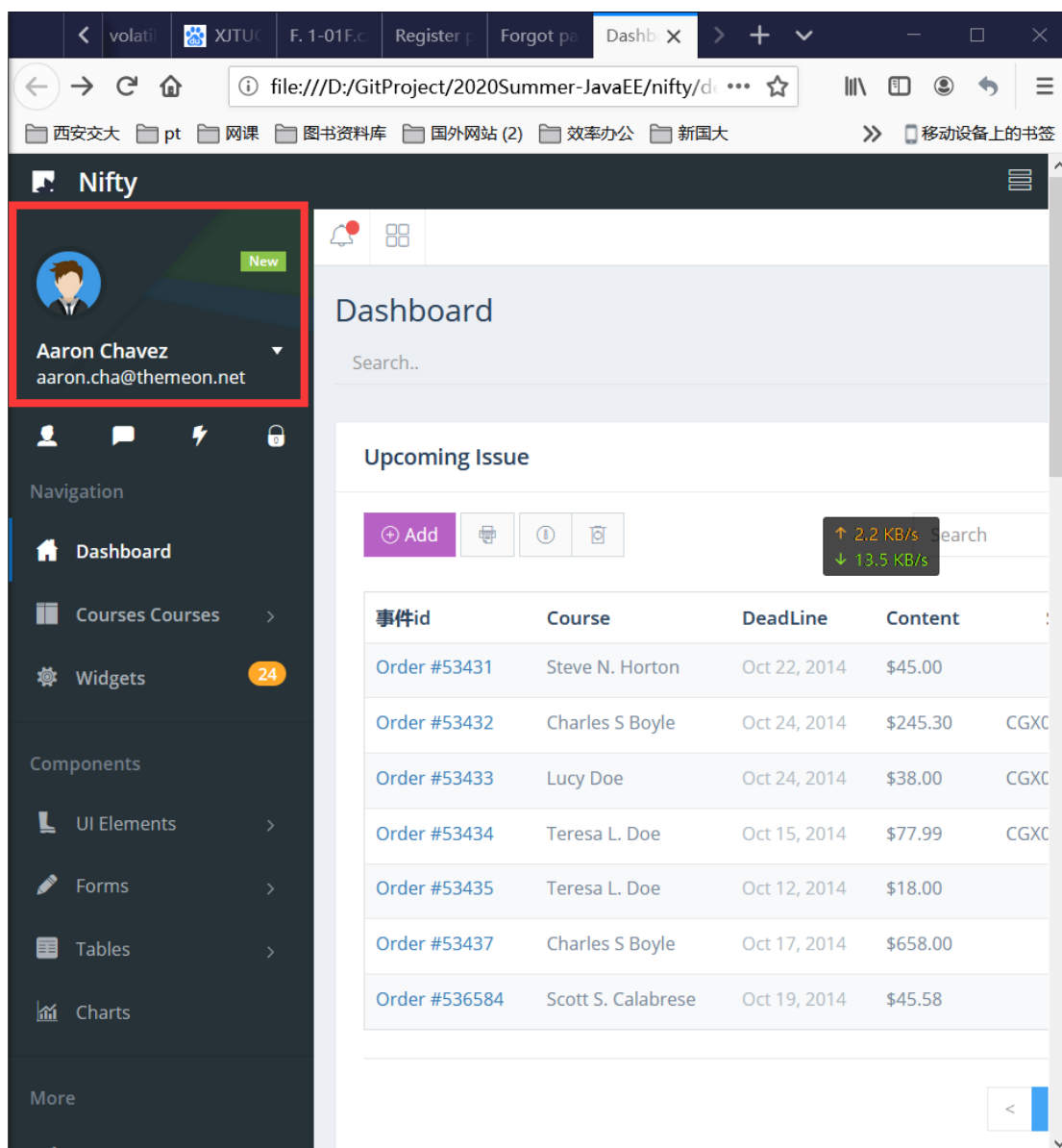


图 2.2.1.1-1 侧边栏用户头像、昵称、邮箱位置

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

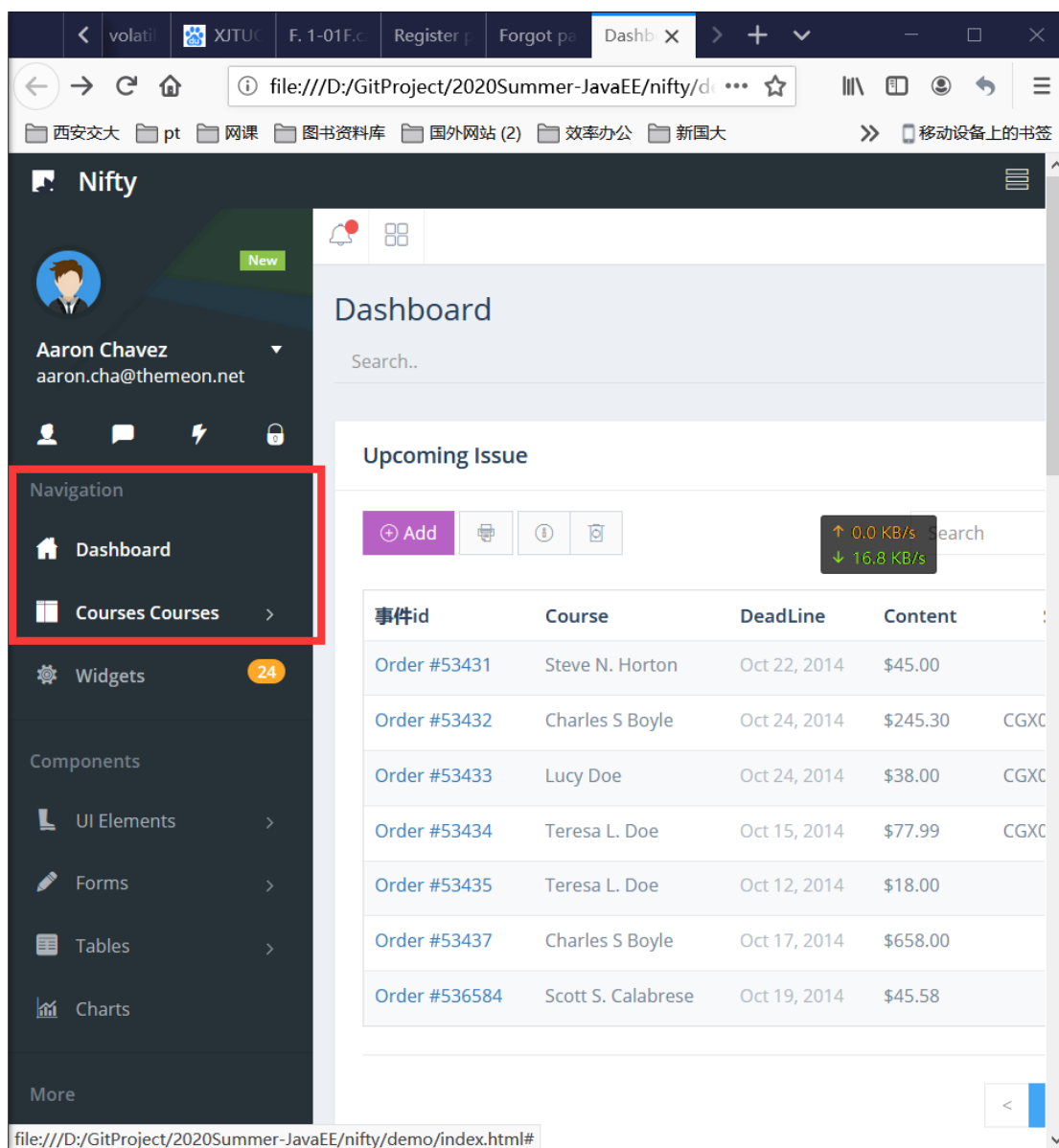


图 2.2.1.2-1 Navigation 栏位置

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。

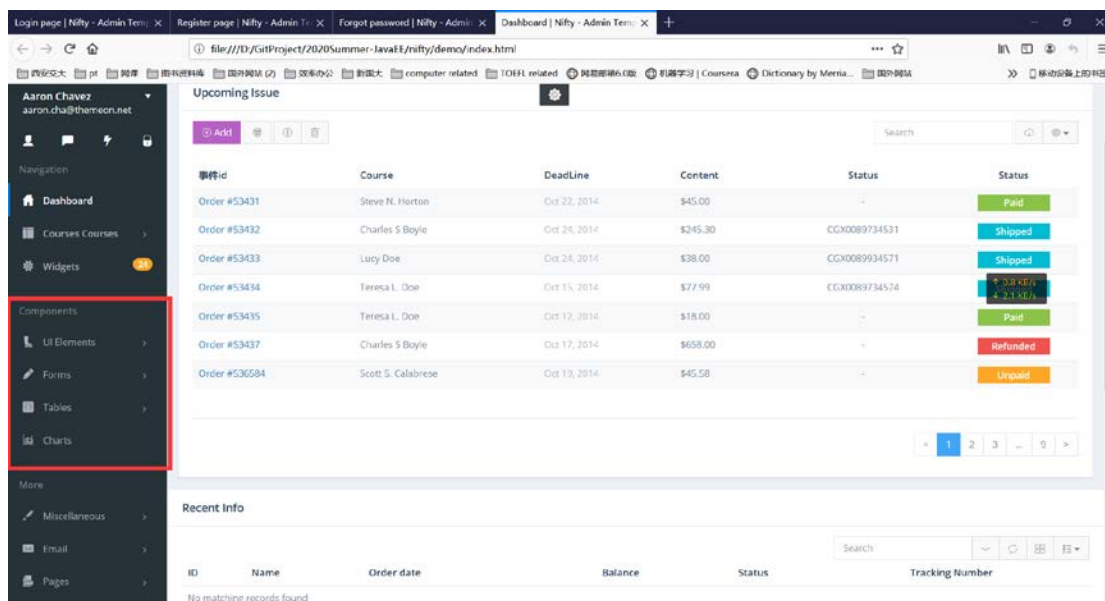


图 2.2.1.3-1 Courses 栏位置

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。

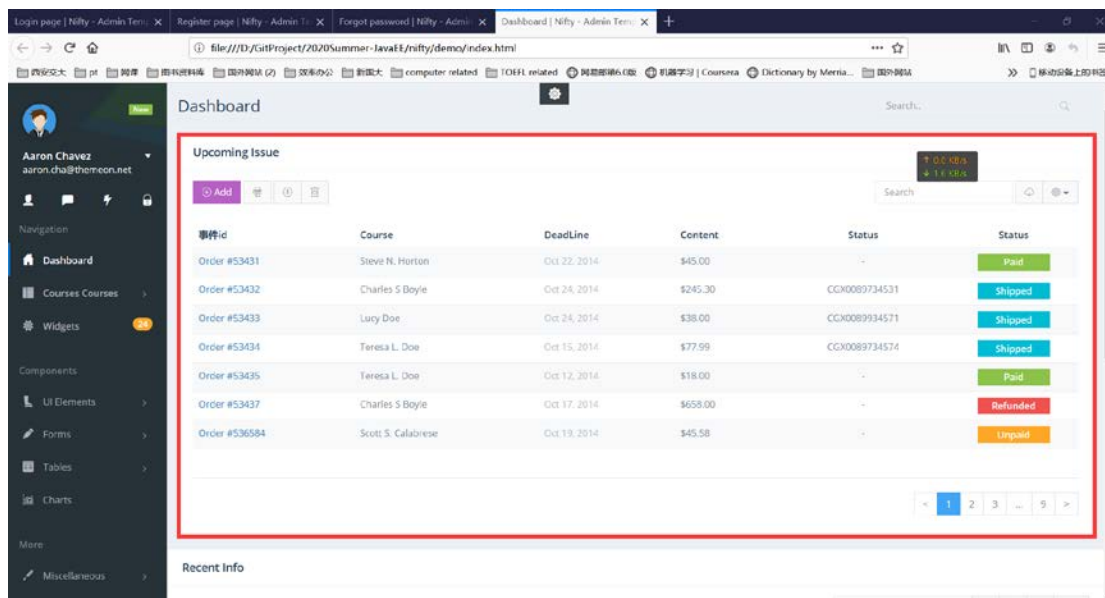


图 2.2.2-1 Upcoming Issue 表格位置

其一共有五个表项，其显示内容见表 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

Search

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。

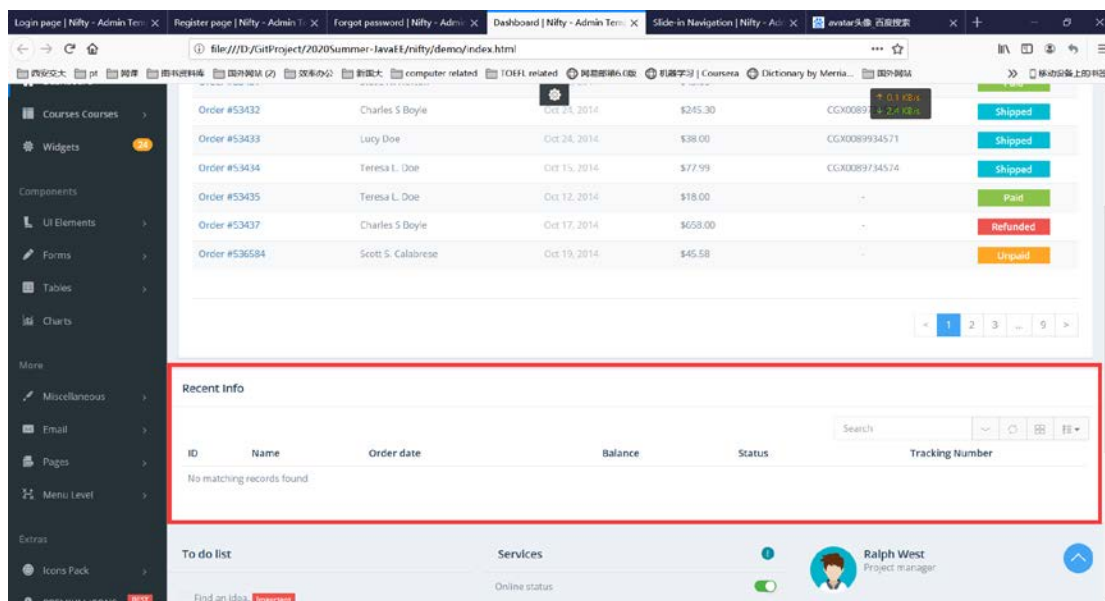


图 3.2.3-1 Recent Info 表位置

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 PREVCOURSES 界面

界面文件名称：prev-courses.html

用户点击侧边栏 PrevCourses 按钮，进入 PrevCourses 界面。

界面样式如 ui-panels.html 所示，具体设计如下图 3.3-1。

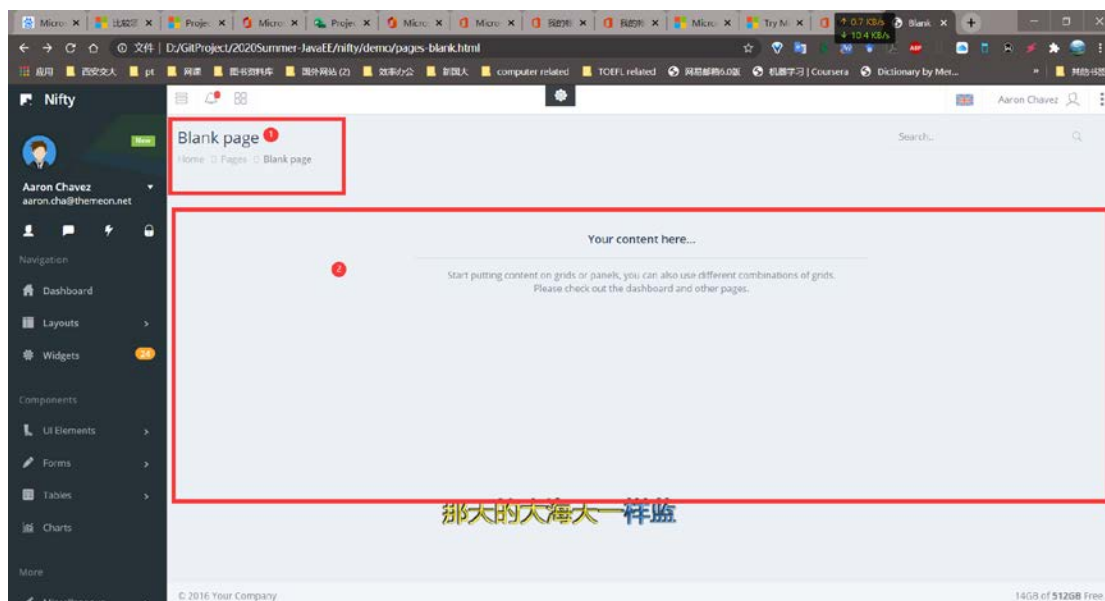


图 3.3-2PrevCourses 界面

标号 1 处 Blank Page 文字改为 Previous Courses;

标号 2 处使用 tables-footable.html 文件如图 3.3-2 所示的 panel。

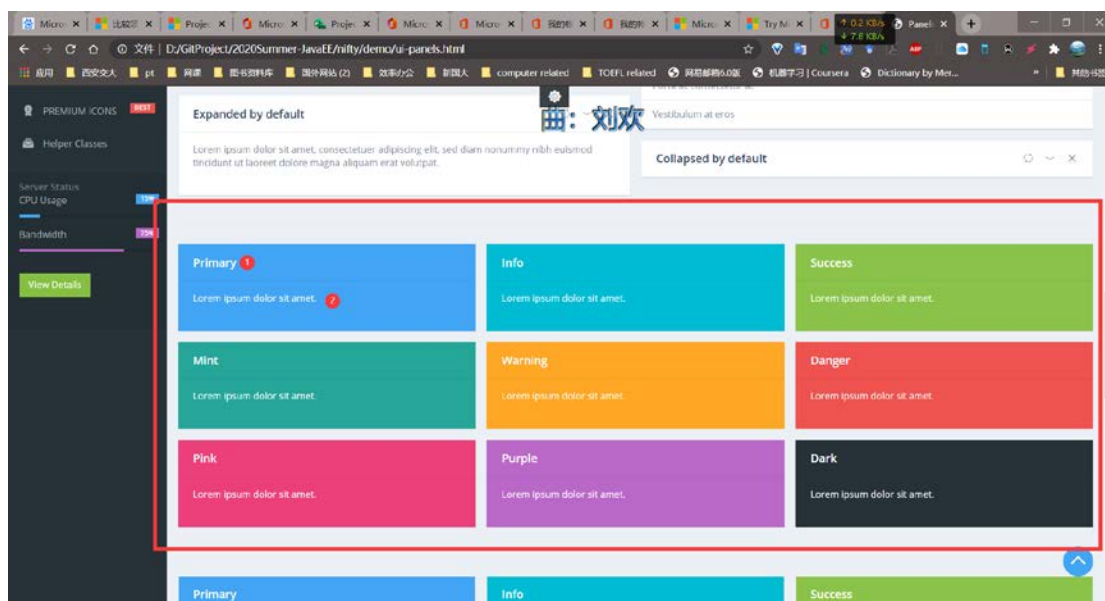


图 3.3-2 PrevCourse 下拉栏

在图 3.3-2 中，1 处文字改为课程的具体名称；

2 处文字分两行，第一行显示课程的开课学院，第二行显示课程的任课教师，第三行显示开课年级。

Panel 示例如表 3.3-1 所示：

表 3.3-1 panel 示例

高等数学
数学与统计学院，曹大华
2017 级

标号 1, 2 处内容详细信息需要查询后台数据库，具体见 [GetPrevCourses](#) 接口。

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。

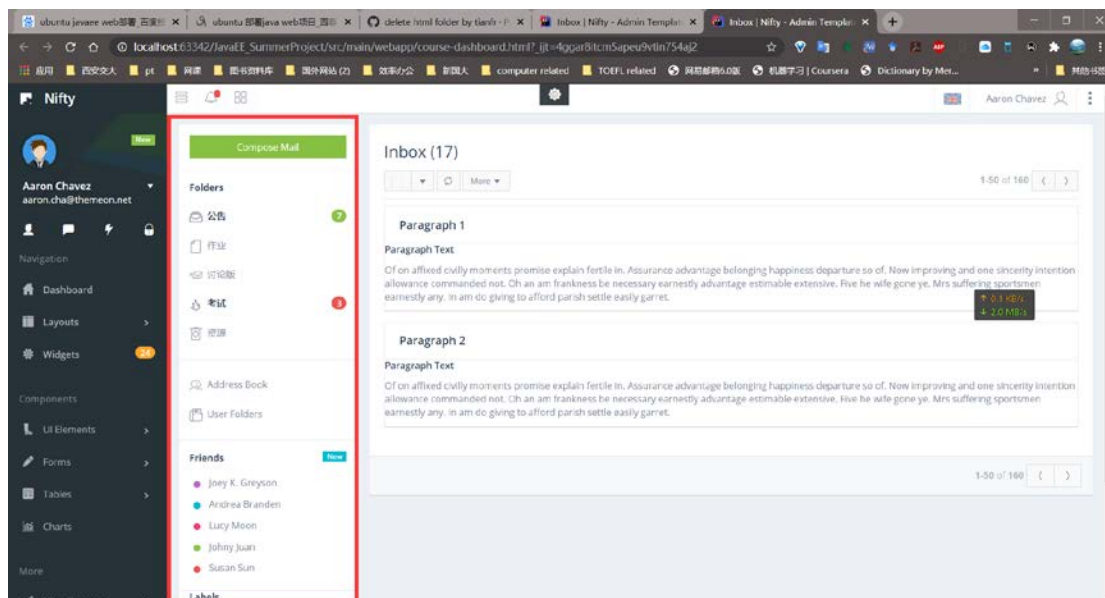


图 3.4-1 课程侧边栏展示图

- 课程内容详情部分：位置如图 3.4-2 所示。

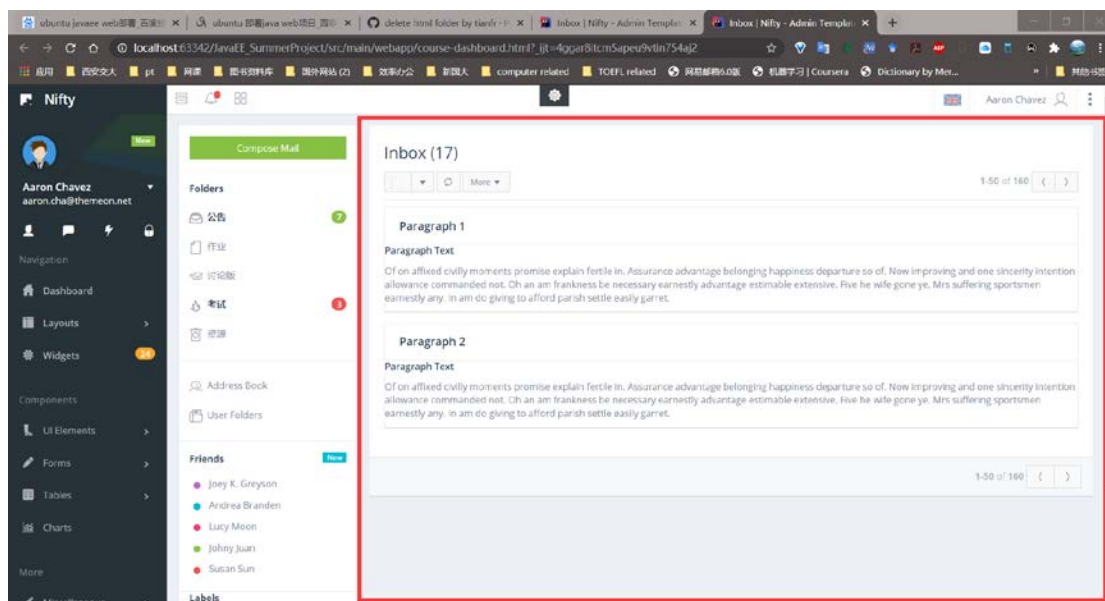


图 3.4-2 课程内容详情部分位置

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。

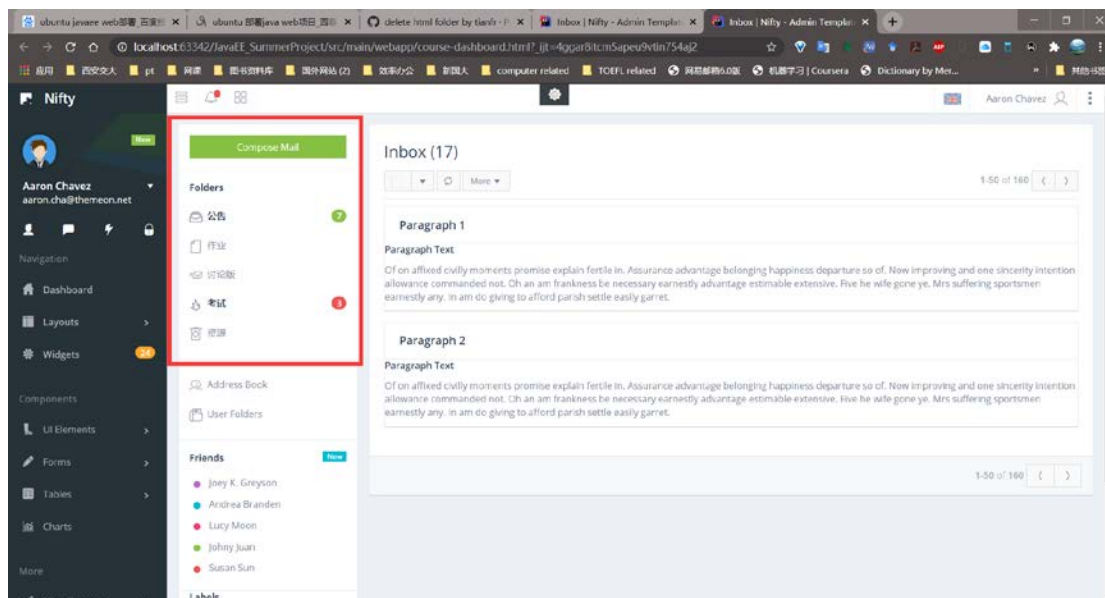


图 3.4.1.1-1 侧边栏 Folders 部分

该部分共有 5 个 button：公告，作业，讨论版，考试，资源。

Folders 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 Foulde 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。

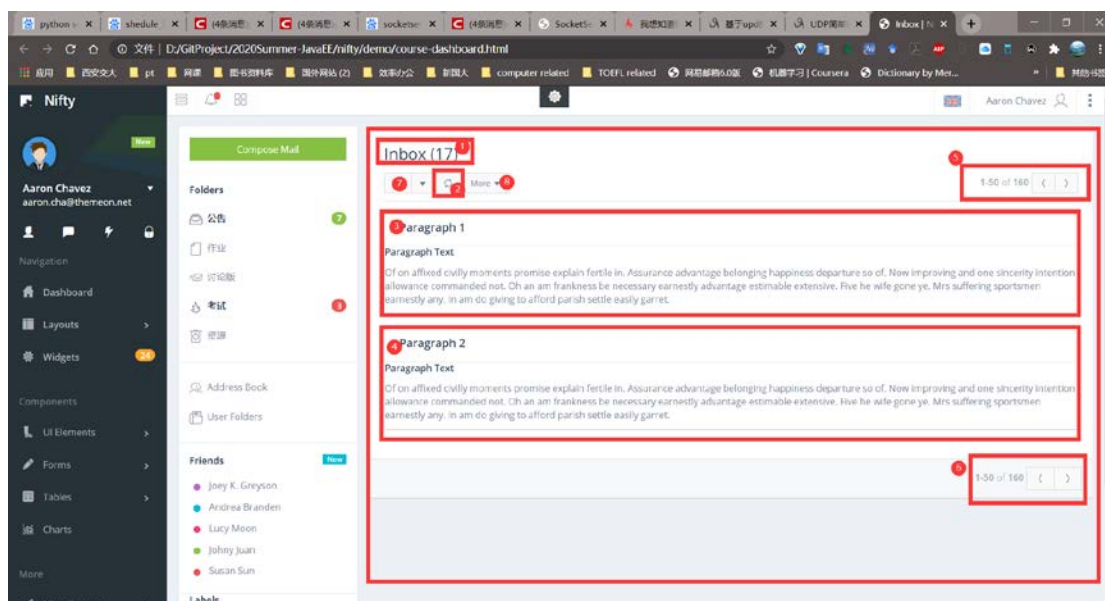


图 3.4.2.1-1 公告详情页

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

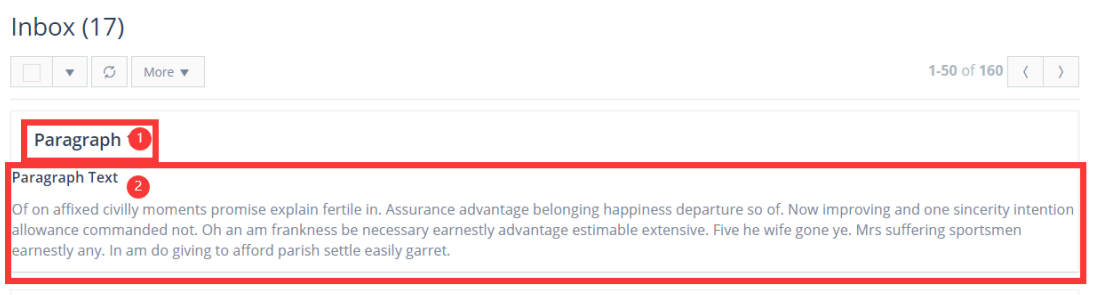


图 3.4.2.1.1-1 具体公告内容示意图

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。

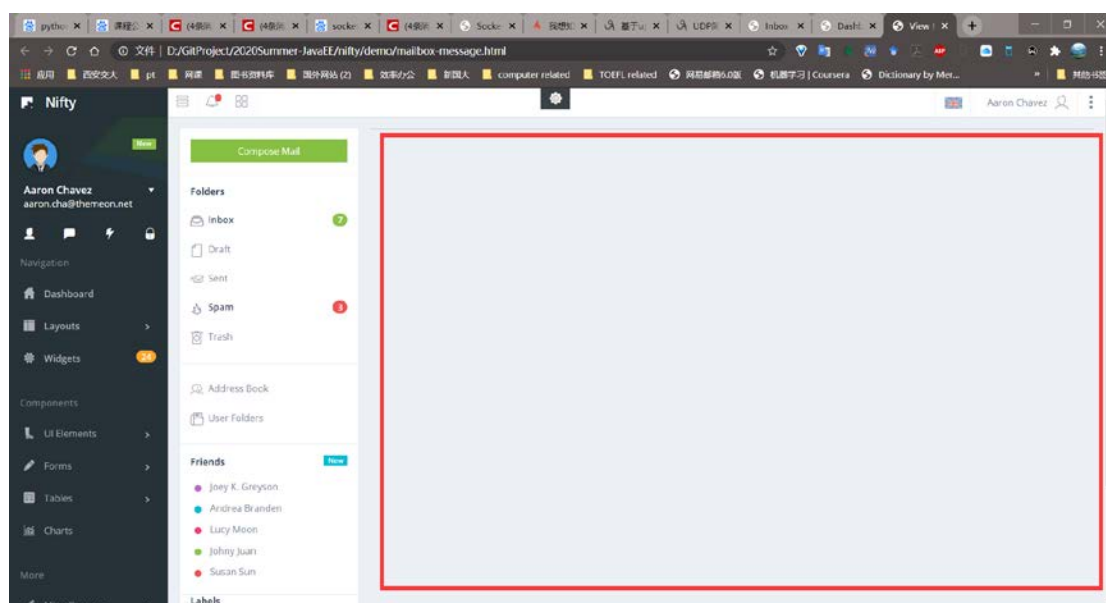


图 3.4.2.1.2-1 具体课程公告页位置

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。

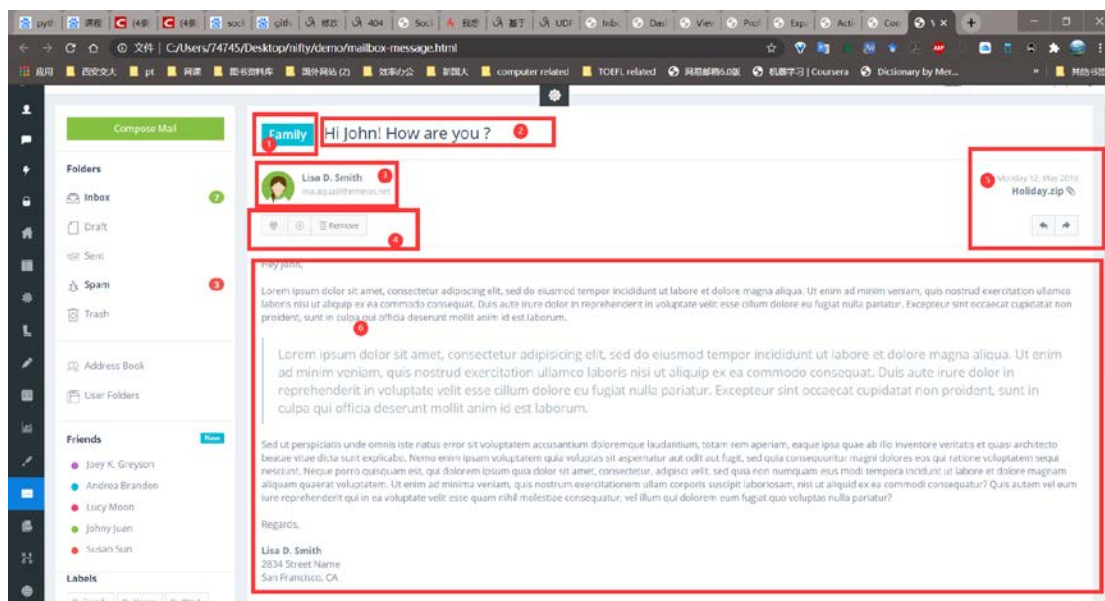


图 3.4.2.1.2-2 公告展示内容具体设计

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。

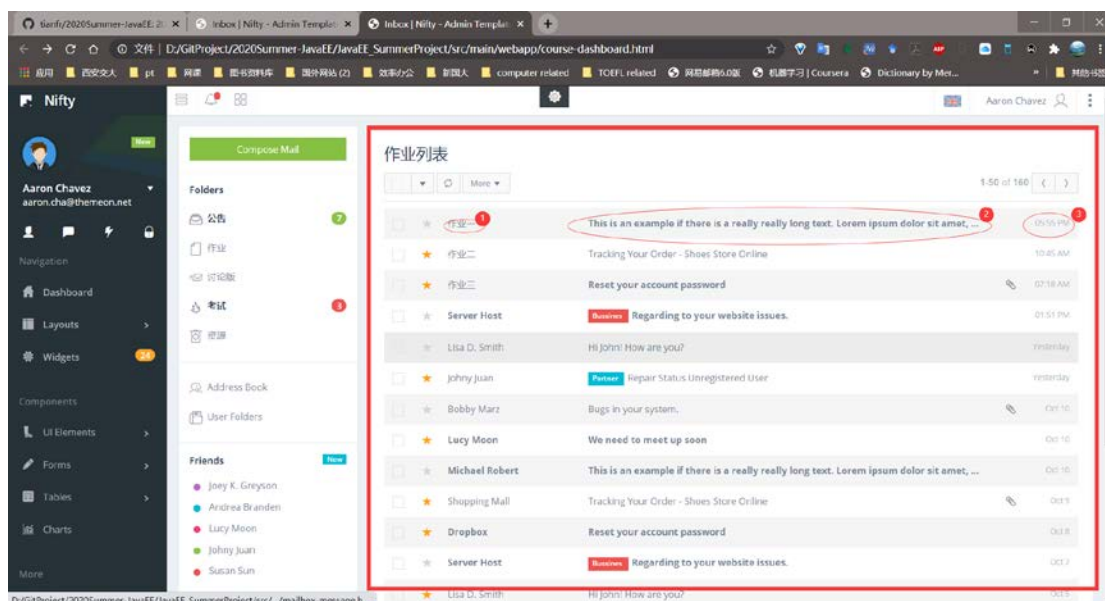


图 3.4.2.2.1-1 作业详情页设计

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。

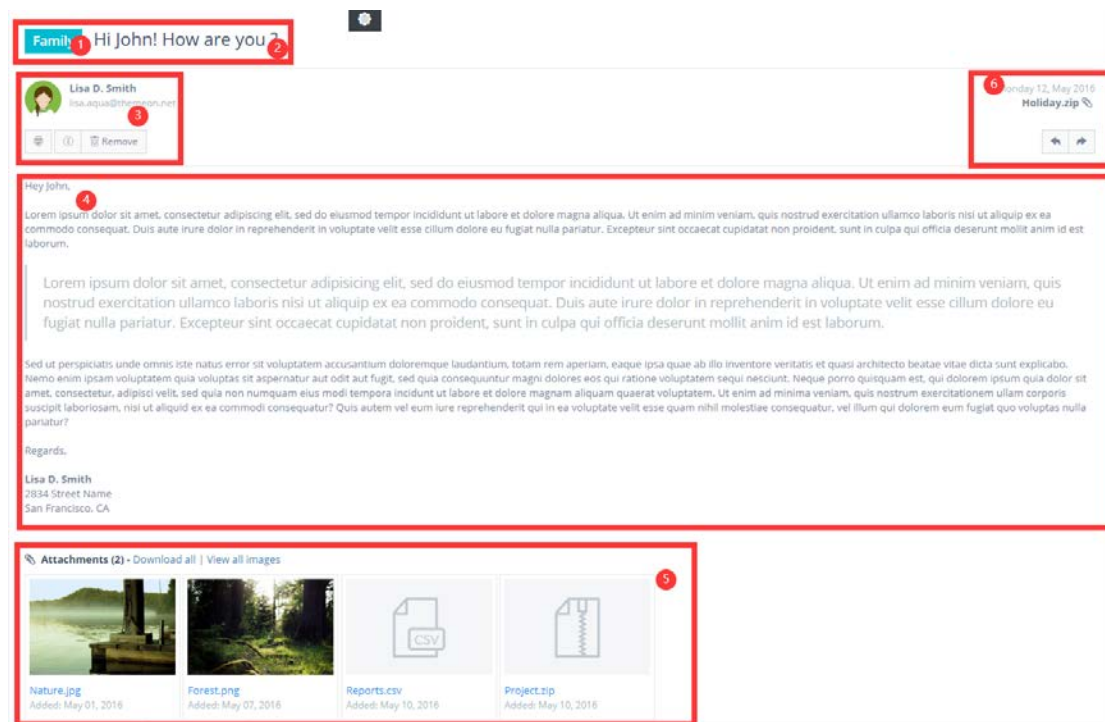


图 3.4.2.2.2-1 课程作业界面具体设计图

- 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；

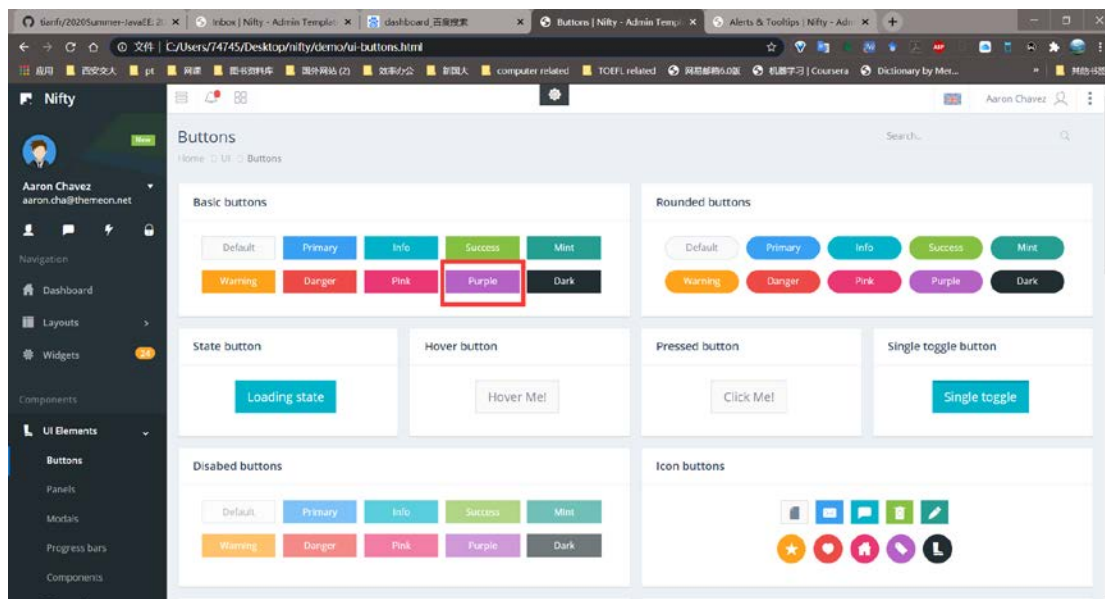


图 3.4.2.2.2-2 作业提交按钮

- 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。

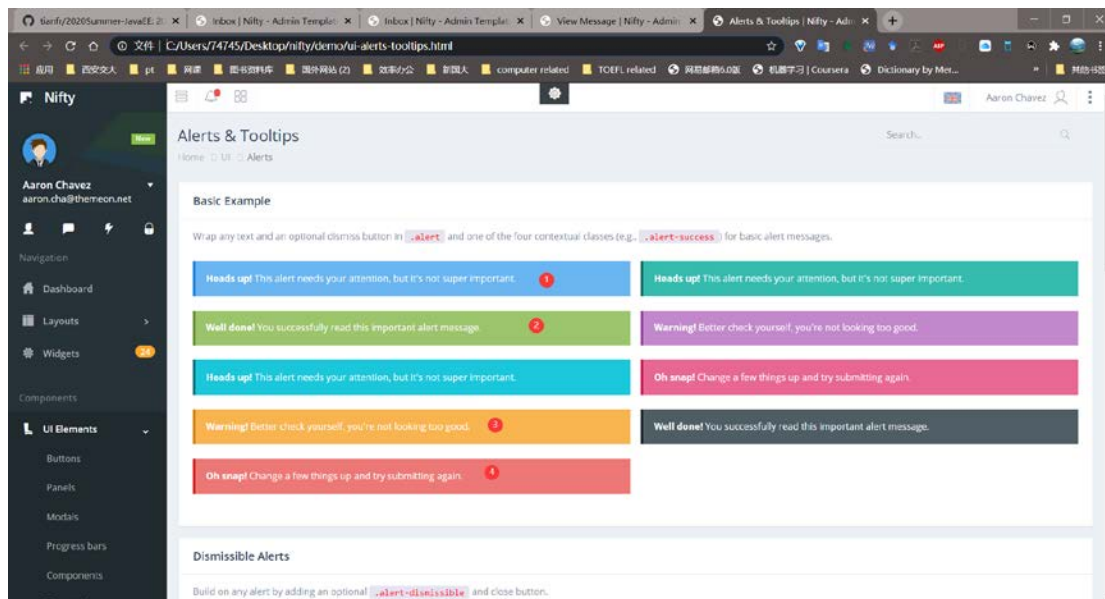


图 3.4.2.2.2-3 作业截止日期行实例

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。

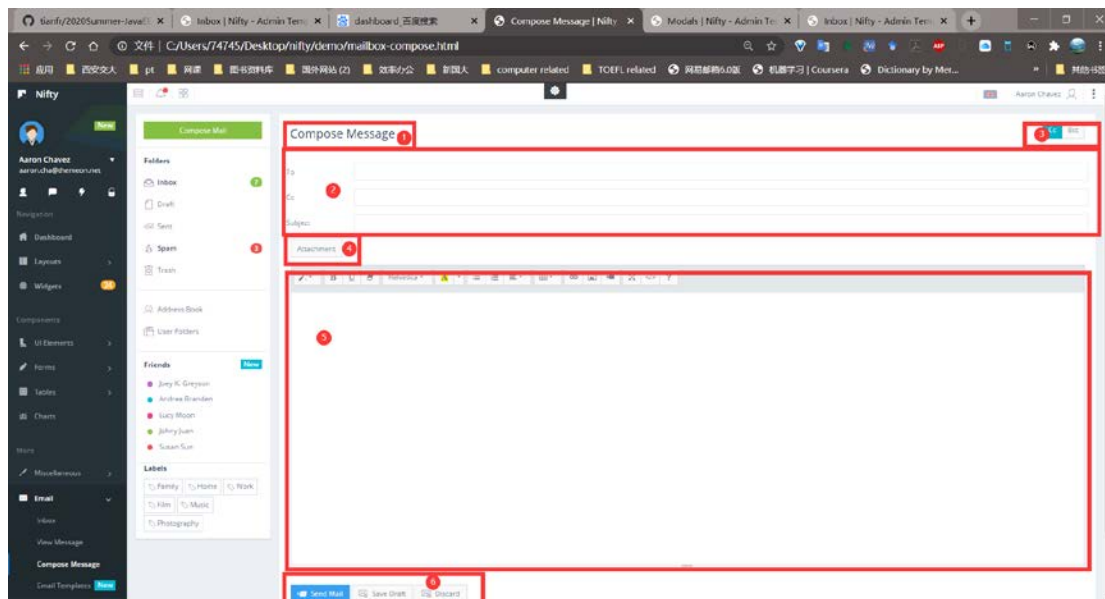


图 3.4.2.2.3-1 作业文本提交示意图

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:

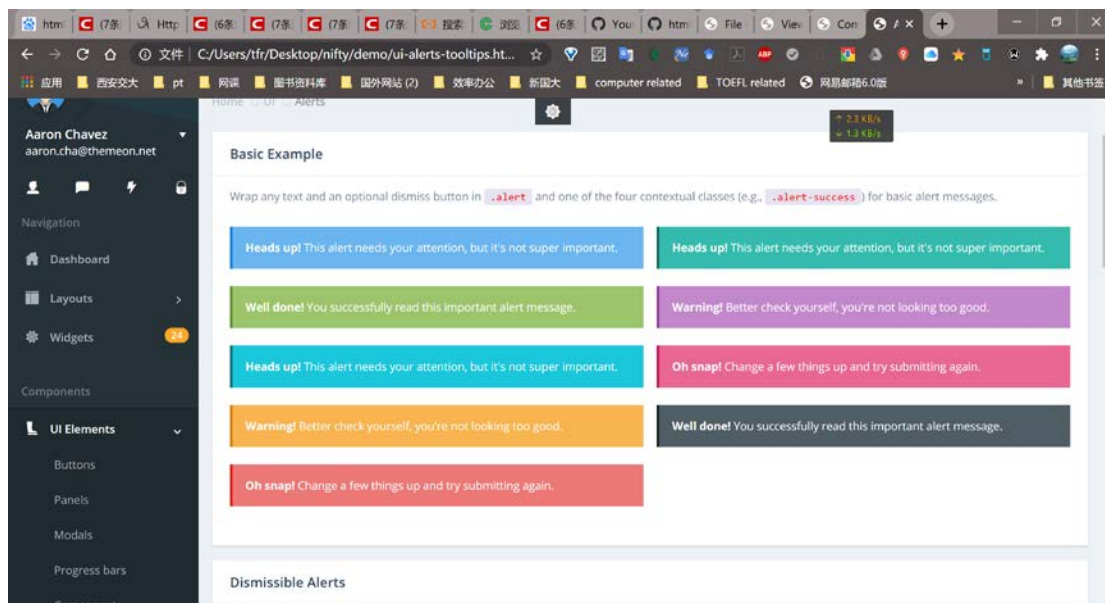


图 3.4.2.2.3-2 作业提交状态警示栏

- 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。

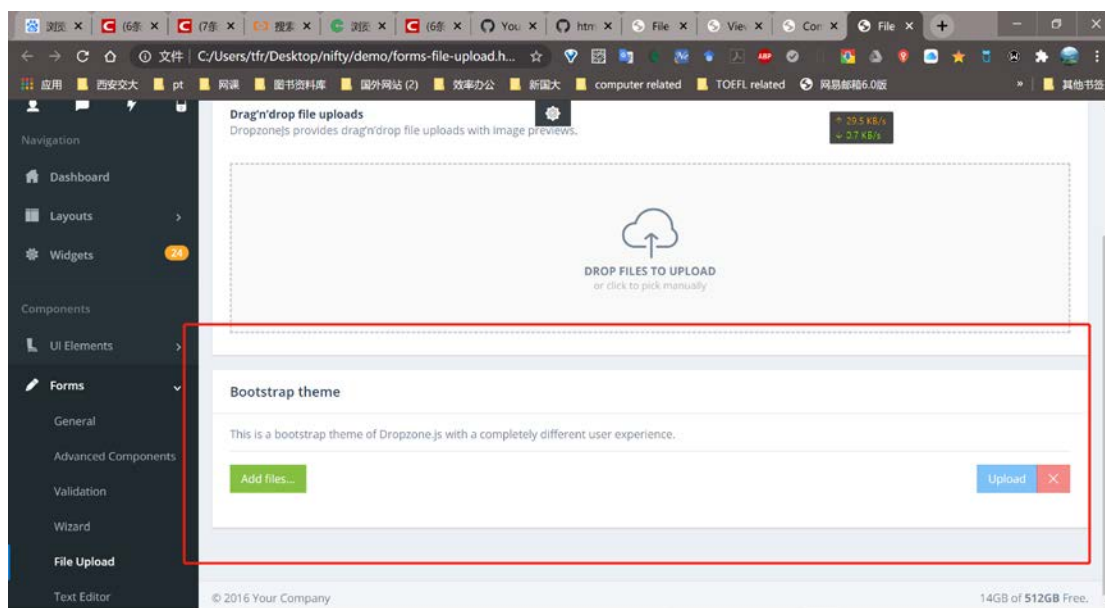


图 3.4.2.2.3-3 文件上传面板样式

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。

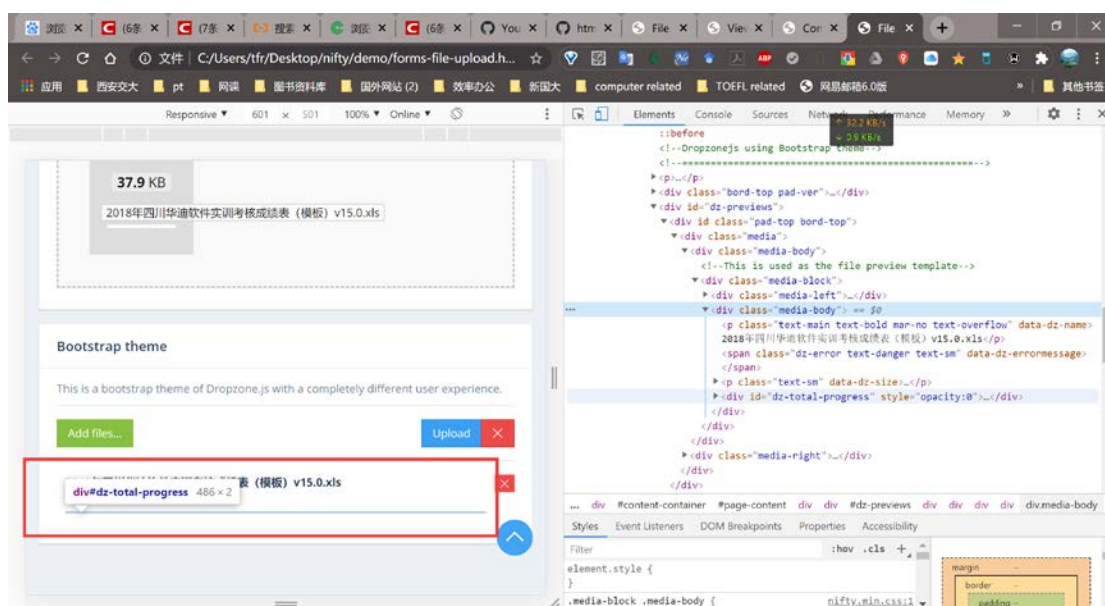


图 3.4.2.2.3-4 文件上传进度

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 讨论详情页设计

待定。

3.4.2.4 考试详情页设计

待定。

3.4.2.5 资源详情页设计

用户点击课程侧边栏上的资源栏，进入资源详情页。

资源详情页参考 mailbox.html 文件，具体设计见图 3.4.2.5-1。

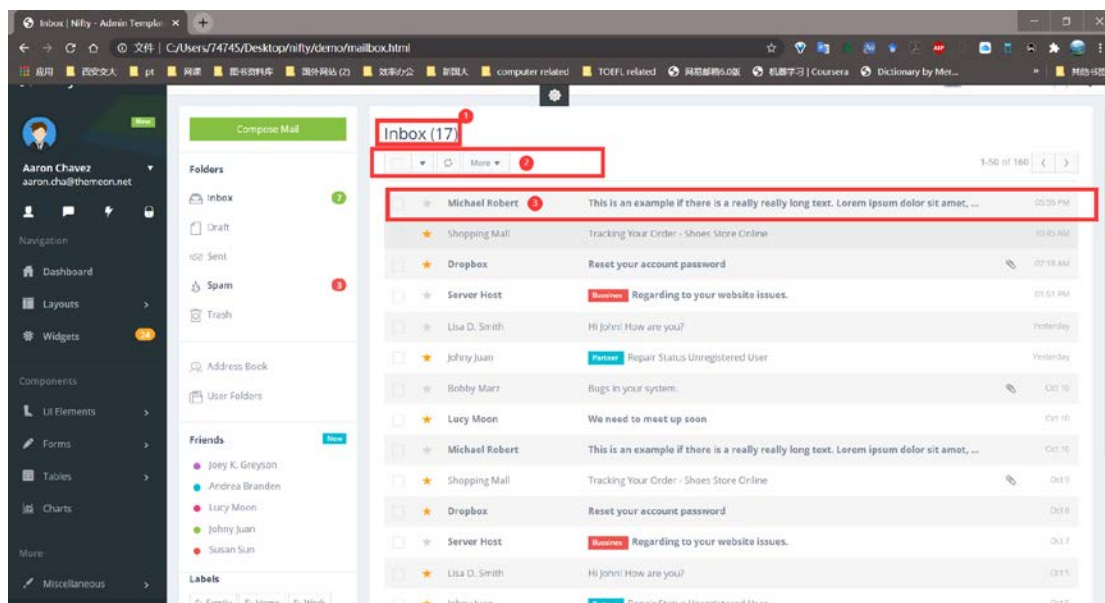


图 3.4.2.5-1 资源详情页设计

如图 3.4.2.5-1 所示，标号 1 处文字改为课程资源；标号 3 开始方框用来选中该文件，左边文字“Michael Robert”改为资源文件名称，右边“This is an example if there is a really really long text...”改为资源的具体介绍，右边事件改成资源文件大小，用户点击“Michael Robert”或“This is an example if there is a really really long text...”处文件实现单个文件资源的下载；标号 2 处按钮详细设计如图 3.4.2.5-2 所示：

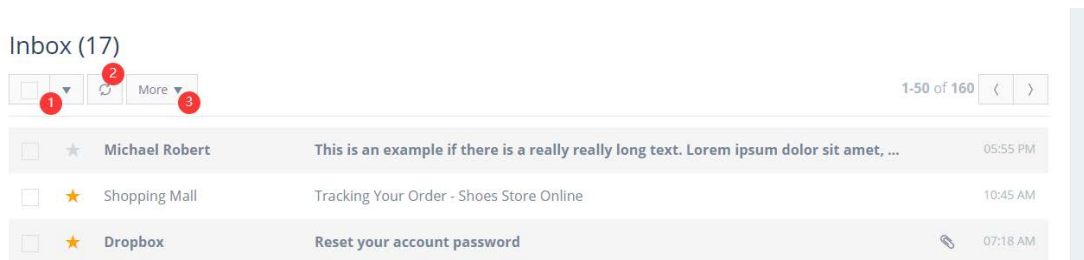


图 3.4.2.5-2 资源详情页按钮设计

在图 3.4.2.5-2 图中，标号 1 按钮设计如下：点击左边小方格则全选本页所有资源，再次点击取消全选；右边倒三角符号保留，功能目前先不实现；标号 2 和标号 3 按钮删除位置替换为一个资源下载 button，button 样式为 ui-buttons.html 界面中的 basic button，如图 3.4.2.5-3 所示：

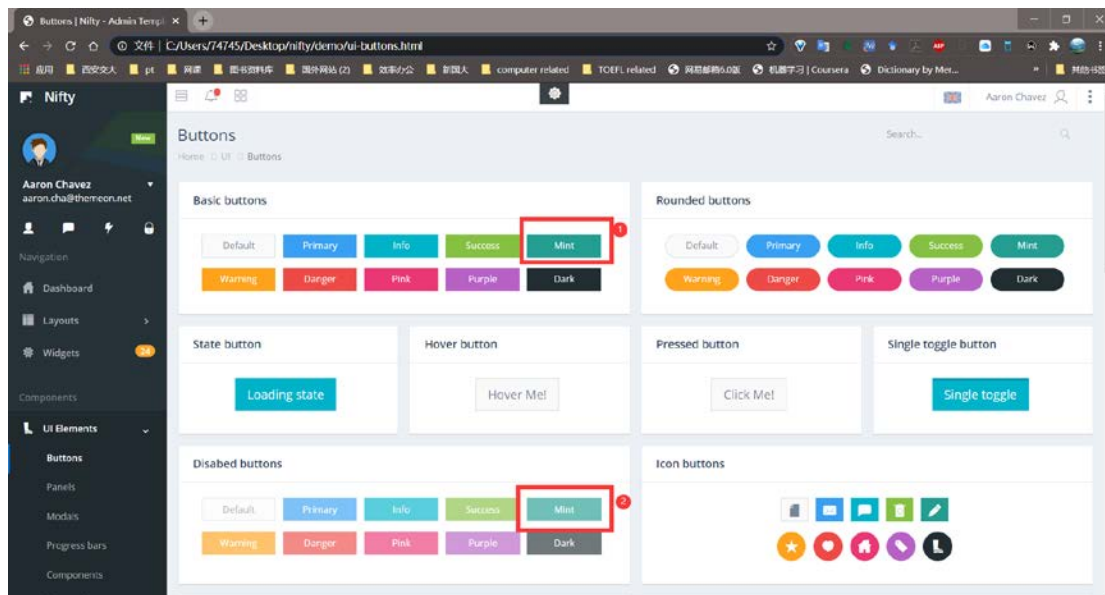


图 3.4.2.5-3 资源下载所用 button

即用图 3.4.2.5-3 标号 1 位置的 button，文字改为全部下载。当用户点击选中了某些文件左边方框，并单击该按钮，实现文件的批量下载。当用户并未选中任何文件时，该 button 不可点击，即用图 3.4.2.5-3 的标号 2 位置的 button。

图 3.4.2.5-1 中资源文件列表显示需要向后端查询，查询接口细节见 GetCourseResources 接口；

图 3.4.2.5-2 和 3.4.2.5-3 中资源文件的下载和资源文件的批量下载需要向后端发起请求，接口细节见 [FileDownload](#) 和 [FilesDownload](#) 接口。

4 网页说明——教师用户

4.1 登录相关界面

4.1.1 登录界面

与 3.1.1 类似，教师通过输入教师用户名密码实现登录，登录接口详见接口 [Login](#) 接口，**但注意，此接口需要修改。**

4.1.2 注册界面

与 3.1.2 注册界面类似，但需要在 3.1.2 功能实现的基础上修改，具体修改见图 4.1.2-1。

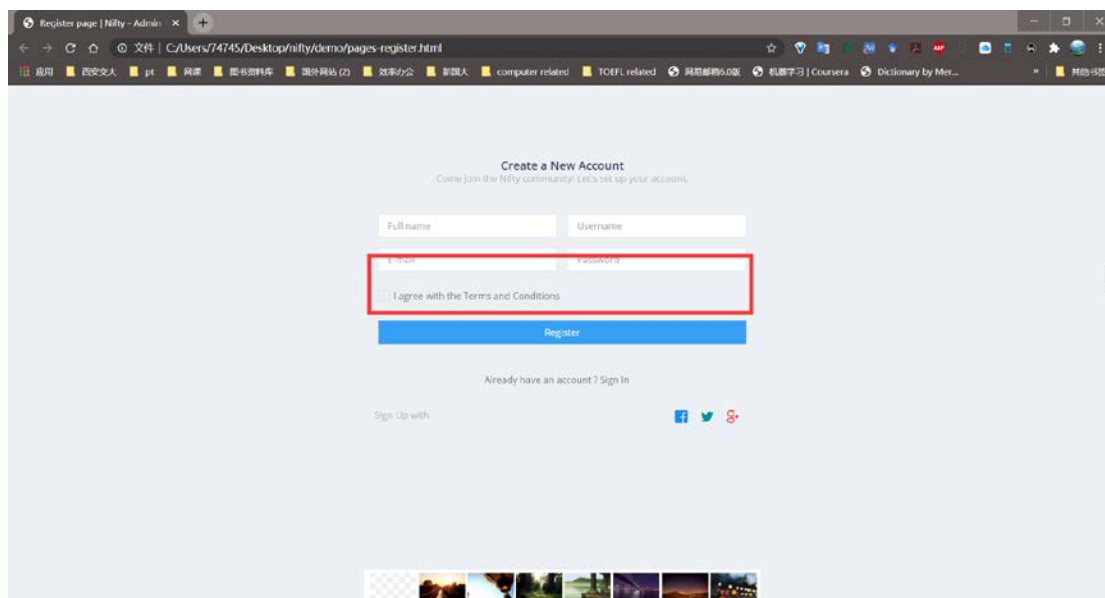


图 4.1.2-1 用户注册界面修改

在图红框标注位置处，即在 Email 和 password 填写栏下方，“I agree with the Terms and Conditions”上方插入一个选项栏，选项栏参考 ui-list-group.html 格式，具体如图 4.1.2-2 所示：

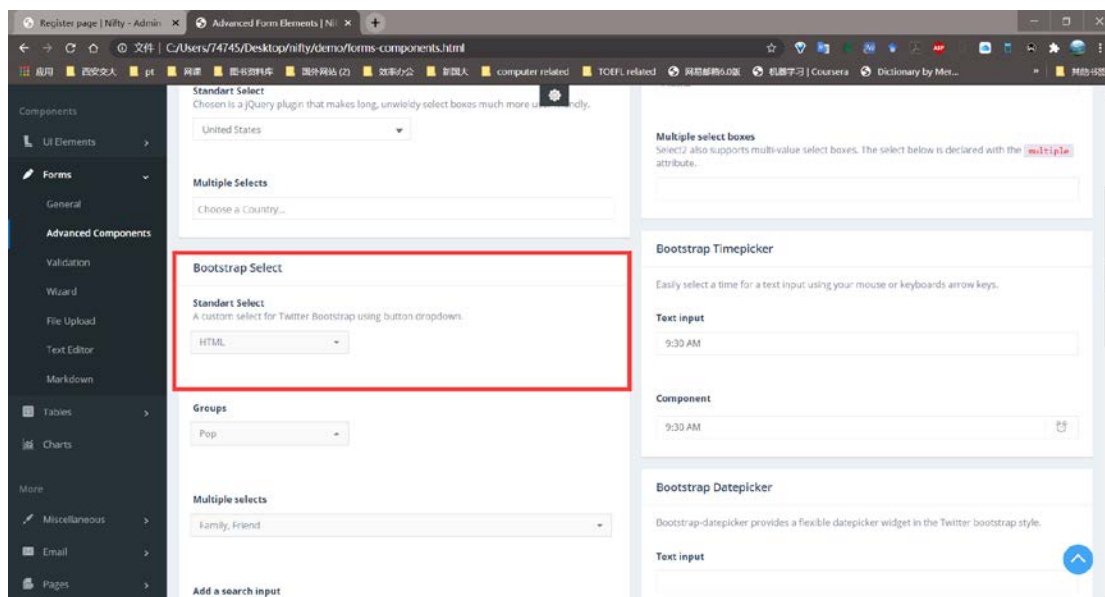


图 4.1.2-2 角色选择栏

使用上图红框所示的角色选择栏，且栏中有三项，分别是教师，学生，管理员。用户通过点击某个选择栏，选择角色。

注意，同 3.1.2，需要向后端发送数据，同样需要调用 [Register 接口](#)，但正向接口格式需要更改。

4.1.3 找回密码界面

同 3.1.3。

4.2 DASHBOARD 界面

4.2.1 侧边栏设计

同 3.2.1，侧边栏显示教师本学期上的课

4.2.2 Upcoming Issue 详细设计

暂同 3.2.2。

4.2.3 Recent Info 详细设计

暂同 3.2.3

4.3 PREVCOURSE 界面

同 3.3，显示老师本学期前教过的课程

4.4 课程界面

4.4.1 课程侧边栏详细设计

大体设计同 3.4.1。

4.4.2 课程内容详情页设计

4.4.2.1 公告详情页设计

基本与 3.4.2.1 相同，但进入公告页后，需在界面做上角添加一个 button，如图 4.4.2.1-1 所示。

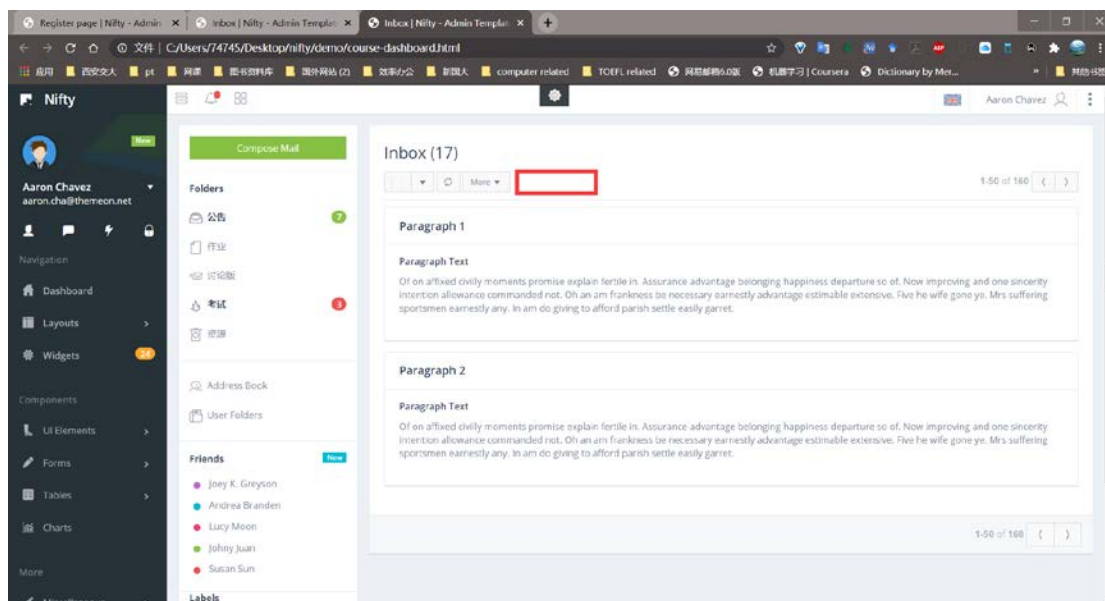


图 4.4.2.1-1 公告详情页设计

Button 样式参照 ui-buttons.html 中基本 button 样式，如图 4.4.2.1-2 所示

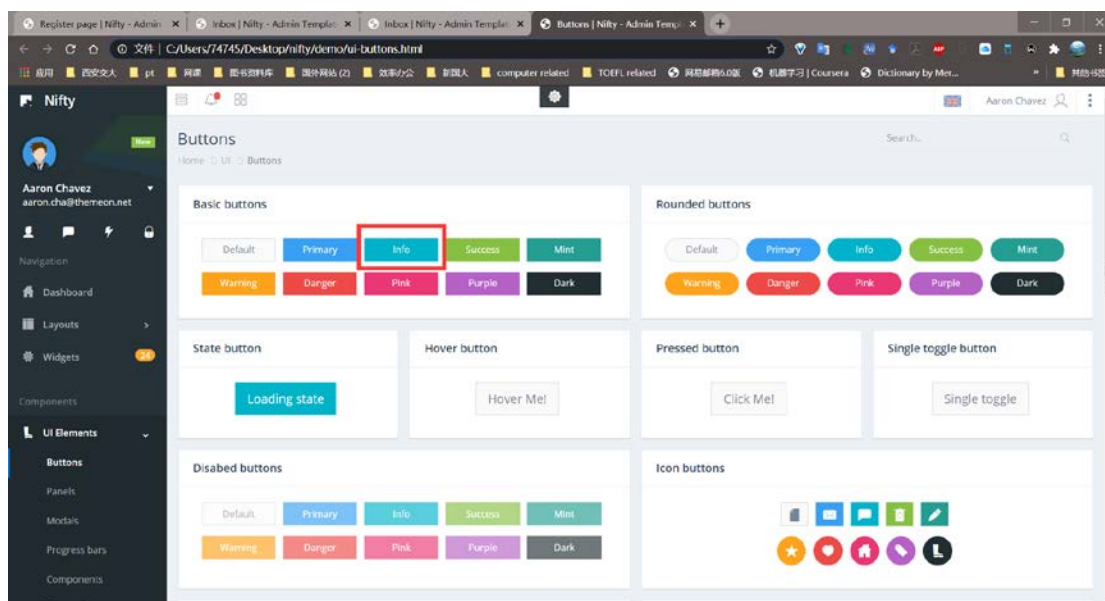


图 4.4.2.1-1button 选择

该 button 的文字改为 new，作用为：当教师点击该 button 时，进入新建公告界面。

该 button 在学生用户登录进入该界面时隐藏。

浏览器如何知道登录用户是学生用户还是教师用户？用户登录后 Login 反向接口返回该用户是学生还是教师或管理员，用户点击界面时通过角色决定是否显示该 button。

4.4.2.1.1 新建公告界面

用户通过

界面名称: new-announcement.html

教师在公告界面点击“new” button, 进入新建公告界面。

新建公告界面具体设计参考 mailbox-compose.html, 如图 3.4.2.2.3-1 所示。

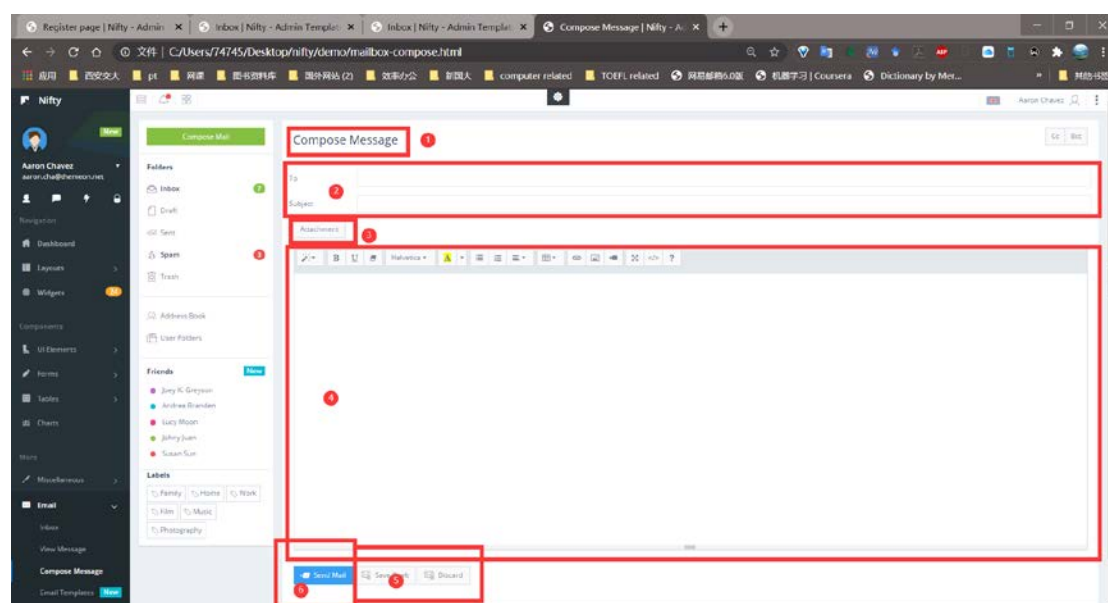


图 3.4.2.2.3-1 作业文本提交示意图

图中标号 3, 5 处删除, 图中标号 1 处文字改为新公告, 标号 5 处保留, 标号 6 处按钮文字发布新公告, 编号 2 地方两行写入栏变为一行, 左边文字改为: 公告标题。

标号 4 处教师用来写新公告内容。

教师点击标号 6 处 button 后, 浏览器向后端发送数据, 数据格式见[接口 NewCourseIssue](#)。

4.4.2.2 作业详情页设计

与学生用户进入时查看到的界面大致相同, 但如 4.4.2.1 节同理所属, 在相同位置新增一个“new” button, 教师点击后, 进入新建作业界面。

4.4.2.2.1 新建作业界面

文件名称: new-homework.html

与 [4.4.2.1.1 新建公告界面](#) 基本一致, 但在图 3.4.2.2.3-1 基础上, 在标号 4 下面加入文件上传框架, 框架模板参照 3.4.2.2.3 课程作业提交界面使用模板, 标号 1 处文字改为新作业, 教师填写完基本内容后点击标号 6 处 button 提交新作业, 与后端数据接口格式见 [NewCourseIssue 接口](#)。

4.4.2.2.2 已提交作业列表查看界面设计

教师用户通过点击某项作业进入作业详情页（设计同 3.4.2.2.2），如图 4.4.2.2.2-1 所示。

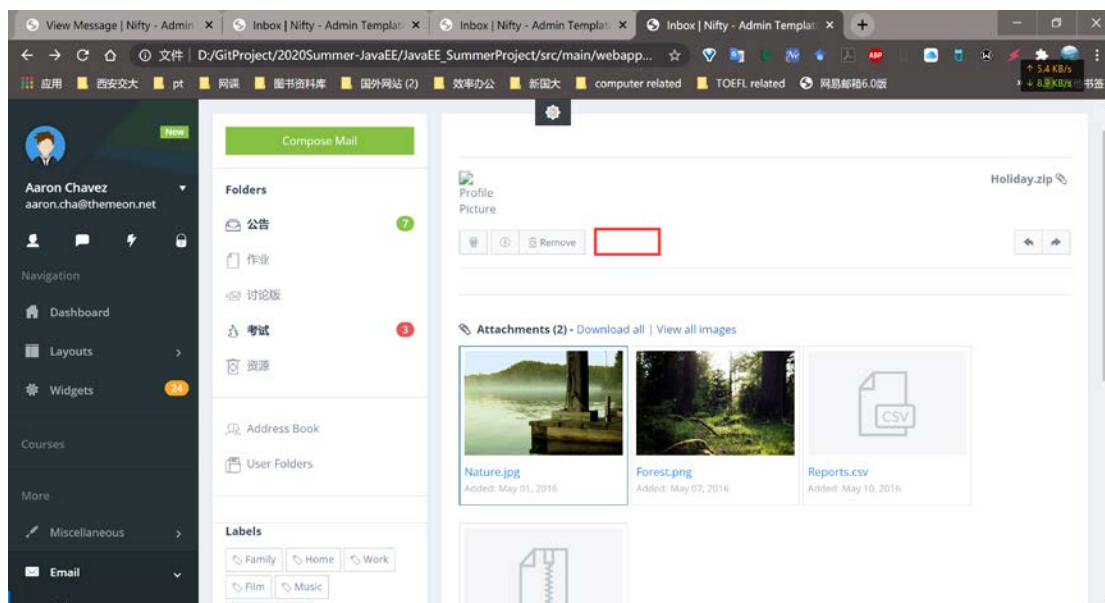


图 4.4.2.2.2-1 已提交作业点击按钮位置

在设计 3.4.2.2.2 的基础上，在图中红框位置新加一个 button，文字填写为查看已提交作业，样式参考 ui-buttons.html 的 button，如图 4.4.2.2.2-1 所示：

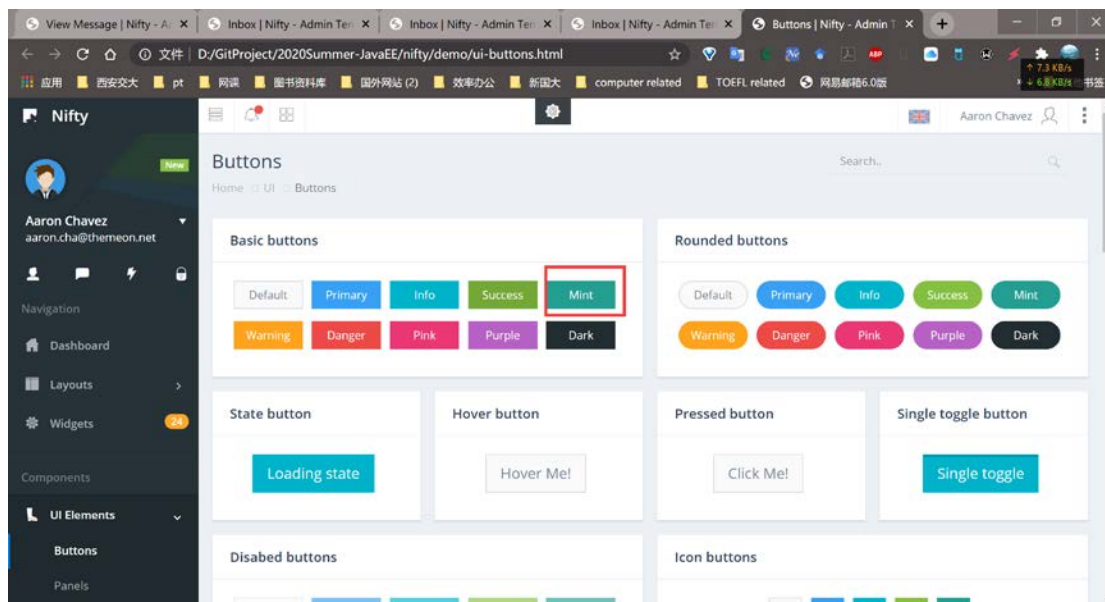


图 4.4.2.2.2-2 查看已提交 button 选择

教师用户点击查看已提交作业按钮，进入已提交作业界面界面，如图 4.4.2.2.2-3 所示：

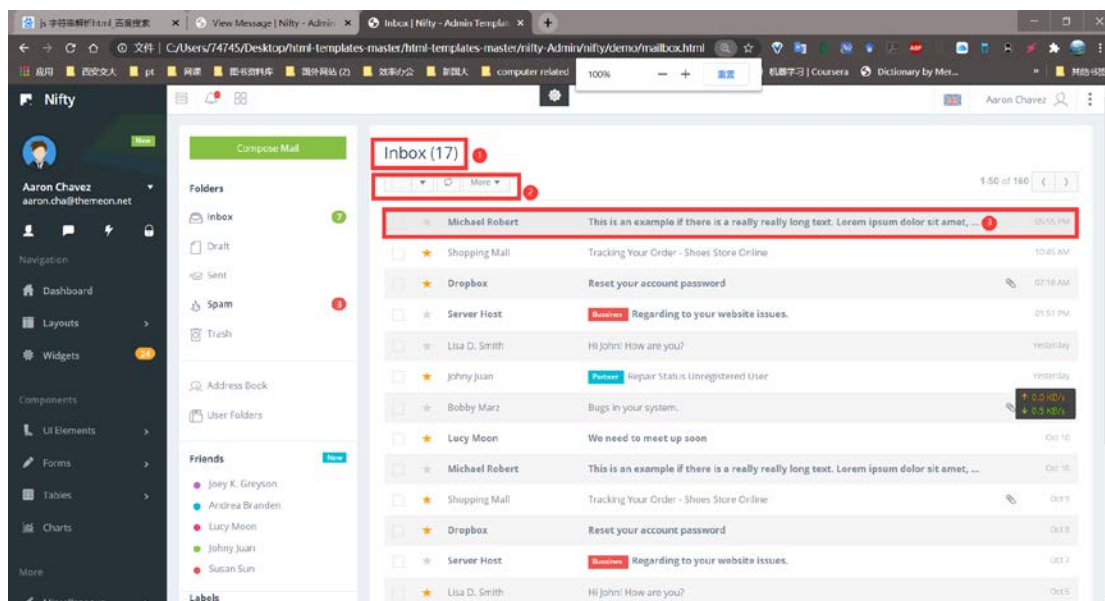


图 4.4.2.2.2-3 已提交作业界面

标号 1 处文字改为已提交作业列表；

标号 2 处按钮先不实现功能；

标号 3 处 MichaelRobert 处显示作业 title 标题，This is an example... 处显示作业的 content；

用户通过点击标号 3 处的 titile 和 content 进入学生提交的作业具体详情界面。

获取已经提交作业列表接口见 [GetStudentHomeworks](#) 接口。

4.4.2.2.3 学生提交的作业具体详情界面设计

教师通过点击学生作业标题或作业内容，进入作业详情界面。作业详情界面显示参考图 4.4.2.2.3 设计。

与 [4.4.2.1.1 新建公告界面](#) 基本一致，但在图 3.4.2.2.3-1 基础上，在标号 4 下面加入文件上传框架，框架模板参照 3.4.2.2.3 课程作业提交界面使用模板，标号 1 处文字改为新课程资源，教师填写完基本内容后点击标号 6 处 button 提交新资源，与后端数据接口格式见 [NewCourseIssue 接口](#)。

注意若教师未上传任何资源，则提示“您未上传任何资源文件！”，提示的窗口参照 ui-modals.html 界面的如图 4.4.2.5.1-1 所示界面

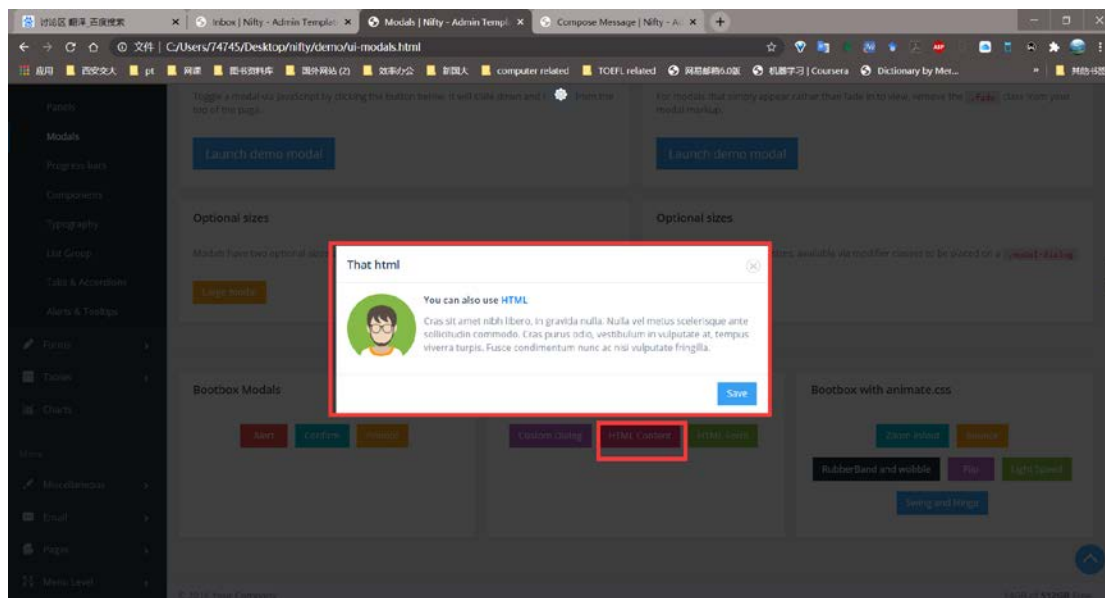


图 4.4.2.5.1-1 新建资源界面

5 网页说明——管理员用户

再说。

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 节学生登录界面](#)，[4.1.1 节教师登录界面](#)。

接口在 1.0 版本后反向接口出现变更

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  
    "role": "teacher/student/manager", //分别表示教师用户, 学生用户, 管理员  
    "role_id": person_id, //教师, 学生或管理员 id  
    "fail_content": "错误详细信息, 由后端返回", //错误详细信息: 邮箱不存在  
  
}
```

数据查询格式：

```
SET @login_username = 'tianfengrui';

SET @login_password = 'tianfengrui';

SELECT

    student_password AS correct_password

FROM

    students

WHERE

    students.student_username = @login_username

UNION

SELECT

    teacher_password

FROM

    teachers

WHERE

    teachers.teacher_username = @login_username;
```

6.3 REGISTER 接口

注意：本接口在 1.0 版本后正向接口格式已更改。

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,
    "role": "teacher/student/manager" //分别指明是教师，学生，管理员角色
}),
```

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
}
```

Sql 语句格式：

第一步：判断用户名是否重复：

```
/* 6.3 接口 */

/* 第一步：判断用户名是否重复 */

USE SummerProject;
```

```
SET @new_username = "tianfengrui";

SELECT

    student_id AS user_id,

    student_username AS username

FROM

    students

WHERE

    students.student_username = @new_username

UNION

SELECT

    teacher_id AS user_id,

    teacher_username AS username

FROM

    teachers

WHERE

    teachers.teacher_username = @new_username;
```

若无重复，插入数据：

教师用户：

```
/* 第二步：教师用户插入 */

INSERT INTO SummerProject.teachers (

    teacher_id,

    teacher_name,

    teacher_username,

    teacher_password,

    teacher_email,

    insert_date)
```


VALUES

```
(
    "tch20000110", "caodahua" , "dddd", "dddd" , "3434@223.com", LOCALTIME()
);
```

学生用户:

/* 第二步: 学生用户插入 */

INSERT INTO SummerProject.students (

students.student_id,

student_name,

student_username,

student_password,

student_email,

insert_date

)

VALUES (

"stu20000110",

"caod3ahua",

"ddd3d",

"ddd3d",

"34334@223.com",

LOCALTIME()

);

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  
}
```

Sql 语句格式:

```
/* 6.4 PasswordReminder */  
  
USE SummerProject;  
  
SELECT  
  
    teachers.teacher_username AS username,  
  
    teachers.teacher_password AS PASSWORD,
```

```
teachers.teacher_email AS email

FROM

teachers

WHERE

teacher_email = "648251476@qq.com"

UNION

SELECT

student_username AS username,

student_password AS PASSWORD,

students.student_email AS email

FROM

students

WHERE

student_email = "648251476@qq.com";
```

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。

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

Sql 语句格式：

```

/* 6.5 DashboardMenu */
-- 教师学生同一个 sql 语句
/* 获取用户个人信息等，教师学生同一个 sql 语句*/

USE SummerProject;

SET @person_id = "tch20200102";

SELECT

    students.student_name AS person_name,

    students.student_avatar AS person_avatar,

    students.student_email AS person_email

FROM

    students

WHERE

    student_id = @person_id

UNION

SELECT

    teachers.teacher_name AS person_name,

```

```
teachers.teacher_avatar AS person_avatar,  
teachers.teacher_email AS person_email
```

```
FROM
```

```
teachers
```

```
WHERE
```

```
teacher_id = @person_id;
```

```
/* 查询所显示的课程 */
```

```
USE SummerProject;
```

```
SET @person_id = "stu20200202";
```

```
SET @person_role = "teacher";
```

```
SET @semester = "spring";
```

```
SET @hierarchy = "2019";
```

```
SELECT
```

```
courses.course_id,  
courses.course_name,  
courses.course_hierarchy,  
courses.course_semester
```

```
FROM
```

```
courses
```

```
WHERE
```

```
courses.course_id IN (
```

```

SELECT

    students_courses.course_id AS course_id

FROM

    students_courses

WHERE

    student_id = @person_id

UNION

SELECT

    teachers_courses.course_id AS course_id

FROM

    teachers_courses

WHERE

    teachers_courses.teacher_id = @person_id

)

AND courses.course_hierarchy = @hierarchy

AND courses.course_semester = @semester;

```

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",

    "total_num": num_of_total_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  
  
    ]  
  
}
```

Sql 语句格式:

```
/*6.6 Upcoming Issue */  
  
/* 找到学生所有所选课程 */  
  
SET @student_id = "stu20200306";  
  
SELECT students_courses.course_id FROM students_courses WHERE student_id = @student_id;  
  
  
  
/* 对学生 每一个所选课程查询 */  
  
SET @course_id = "coursedemo001";  
  
SET @search_tables = ( SELECT courses.course_content_table FROM courses WHERE courses.course_id = @course_id );  
  
SET @STSQL = CONCAT('SELECT issue_id, issue_deadline, issue_content FROM ', @search_tables, ' WHERE issue_type = \'homework\' OR issue_type = \'examination\');  
  
PREPARE stmt FROM @STSQL;  
  
EXECUTE stmt
```

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,
            "datetime": datetime,
```

```

        "content":info_title,

    },

    //项目数量不定，需要遍历访问

]

}

```

Sql 语句格式

```

/* 6.7 */

/* 找到学生所有所选课程 */

SET @student_id = "stu20200306";

SELECT students_courses.course_id FROM students_courses WHERE student_id = @student_id;

/* 对学生 每一个所选课程查询 */

SET @course_id = "coursedemo001";

SET @search_tables = ( SELECT courses.course_content_table FROM courses WHERE courses.course_id = @course_id );

SET @STSQL = CONCAT('SELECT issue_id, issue_content, insert_date FROM ', @search_tables, ' WHERE issue_type != \'homework\' AND issue_type != \'examination\'' );

PREPARE stmt FROM @STSQL;

EXECUTE stmt;

```

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

```
]
}
```

Sql 语句格式:

```
/* 6.8 */

/* 找到学生所有所选课程 */

SET @student_id = "stu20200306";

SELECT students_courses.course_id FROM students_courses, courses WHERE students_courses.student_id = @student_id AND courses.course_hierarchy > 2016 AND students_courses.course_id = courses.course_id;
```

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

  ]

}
```

Sql 语句格式:


```
USE SummerProject;

SET @course_id = "coursedemo001";

SET @search_tables = ( SELECT courses.course_content_table FROM courses WHERE courses.course_id = @course_id );

SET @STSQL = CONCAT('SELECT issue_title, issue_id, issue_content, insert_date FROM ', @search_tables, ' WHERE issue_type = \'announcement\' AND issue_belong = \'announcement\'');

PREPARE stmt FROM @STSQL;

EXECUTE stmt;
```

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": author_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
}
```

Sql 格式

```
/* 6.10 */
```

```
/* 6.10 */
```

```
/*第一步获取 announcement 信息 */
USE SummerProject;

SET @course_id = "coursedemo001";

SET @announcement_id = "tch20200101_content_announcement001";

SET @search_tables = (
    SELECT
        courses.course_content_table
    FROM
```

```

        courses

WHERE

        courses.course_id = @course_id

);

SET @STSSQL = CONCAT(

        'SELECT issue_title, issue_id, issue_content, insert_date FROM ',

        @search_tables,

        ' WHERE issue_id = \'',

        @announcement_id,

        '\ '

);

PREPARE stmt

FROM

        @STSSQL;

EXECUTE stmt;

```

```

/* 第二部根据获取 announcement 对每一个 announcement_id 获取教师信息
   announcement_id 前 11 位是教师 id */

USE SummerProject;

SET @teacher_id = "tch20200101";

SELECT teachers.teacher_avatar, teachers.teacher_email FROM teachers WHERE teacher_
id = @teacher_id;

```

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
```

```
]
```

```
}
```

Sql 语句:

```
/* 6.11 */
```

```
USE SummerProject;
```

```
SET @course_id = "coursedemo001";
```

```
SET @search_tables = (
```

```
SELECT
```

```
courses.course_content_table
```

```
FROM
```

```
courses
```

```
WHERE
```

```
courses.course_id = @course_id
```

```
);
```

```
SET @STSQL = CONCAT(
    'SELECT issue_id, issue_content, issue_deadline FROM ',
    @search_tables,
    ' WHERE issue_type = \'homework\' AND issue_belong in (\'announcement\', \'examination\', \'discussion\', \'homework\')'
);

PREPARE stmt
FROM
    @STSQL;

EXECUTE stmt;
```

6.12 GETHOMEWORK 接口

该接口出现在 [3.4.2.2 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
```



```
}
```

Sql 语句:

```
/* 6.12 */

/* 获取对应 homework_id 作业 */

USE SummerProject;

SET @course_id = "coursedemo001";

SET @homework_id = "tch20200101_content_homework001";

SET @search_tables = (

    SELECT

        courses.course_content_table

    FROM

        courses

    WHERE

        courses.course_id = @course_id

);

SET @STSQL = CONCAT(

    'SELECT issue_id, issue_content, issue_attachment_path, issue_deadline, insert_

date FROM ',

    @search_tables,

    ' WHERE issue_id = \'' , @homework_id, '\''

);

PREPARE stmt
```

```
FROM

    @STSQL;

EXECUTE stmt;

-- 对于作业 id, 前 11 位是老师编号,

USE SummerProject;

SET @teacher_id = "tch20200101";

SELECT

    teachers.teacher_name,

    teachers.teacher_avatar,

    teachers.teacher_email

FROM

    teachers

WHERE

    teacher_id = @teacher_id;
```

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.....
}
```

Sql 语句

```
/* 6.13.1 */
USE SummerProject;

SET @course_id = "coursedemo001";

SET @homework_id = "tch20200101_content_homework001";

SET @student_id = "stu20200203";

SET @is_draft = "no";

SET @homework_content = "newnewnewnewenwenwenwsjflskdfjslkdfjaslk";

SET @homewrok_attachments_path = "path1;path2;";

SET @homewrok_attachments_name = "name1;name2;";

SET @search_tables = (
```

```

SELECT

    courses.course_homework_table

FROM

    courses

WHERE

    courses.course_id = @course_id

);

SET @STSQL = CONCAT(

    'INSERT INTO ',

    @search_tables,

    '(homework_id, student_id, content, homework_file_path, homework_file_name, is_
draft, insert_date)

VALUES (\'',

    @homework_id,

    '\', \'',

    @student_id,

    '\', \'',

    @homework_content,

    '\', \'',

    @homewrok_attachments_path,

    '\', \'',

    @homewrok_attachments_name,

    '\', \'',

    @is_draft,

    '\', \'',

    LOCALTIME ( ),

    '\')
```

ON DUPLICATE KEY UPDATE

```
content='',
    @homework_content,
    '\', homework_file_path='',
    @homework_attachments_path,
    '\', homework_file_name='',
    @homework_attachments_name,
    '\', is_draft='',
    @is_draft,
    '\', insert_date='',
    LOCALTIME (),
    '\')
);
```

PREPARE stmt

FROM

@STSQL;

EXECUTE stmt;

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:

/GetExistedHomework

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
}

```

Sql 语句

```

/* 6.13.2 */

USE SummerProject;

SET @course_id = "coursedemo001";

SET @homework_id = "tch20200101_content_homework001";

SET @student_id = 'stu20200203';

SET @is_draft = "no";

SET @search_tables = (

    SELECT

        courses.course_homework_table

    FROM

        courses

    WHERE

        courses.course_id = @course_id

);

```

```

SET @STSQL = CONCAT(

    'SELECT homework_id, student_id, content, homework_file_path, homework_file_name, is_draft, insert_date FROM ',

    @search_tables,

    ' WHERE homework_id = \'' ,

    @homework_id,

    '\'' AND student_id = \'' ,

    @student_id,

    '\''

);

PREPARE stmt

FROM

    @STSQL;

EXECUTE stmt;

```

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.....
}
```

6.14.3 FileDownload 接口

本项接口存在争议，如何实现下载进度条？(已解决)

该接口出现在 [3.4.2.5 资源详情页](#) 节中。

接口路径:

/FileDownload

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_download",
    "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

```
{
    "file_content":file_content //文件内容
}
```

6.14.4 FilesDownload 接口

本项接口存在争议，如何实现下载进度条？

该接口出现在 [3.4.2.5 资源详情页](#) 节中。

接口路径:

/FilesDownload

接口 js 函数路径:

Js/myjs.js?v=1

正向接口格式:

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

```

        "status": "0",

        "message": "files_download",

        "files":[

            {

                "file_name":file_name,

                "file_path":file_path,

            },

            {

                "file_name":file_name,

                "file_path":file_path,

            },

            {

                "file_name":file_name,

                "file_path":file_path,

            },

        ]

    })),

```

反向接口格式

```

{

    "files":[

        {

            "file_content":file_content,

        },

        {

            "file_content":file_content,

        },

    ]

}

```

```
    },
    {
        "file_content": file_content,
    },
]
}
```

6.15 GETCOURSERESOURCES 接口

该接口出现在 [3.4.2.5 资源详情页](#) 节中。

接口路径：

/GetCourseResources

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_resources",
    "course_id": course_id,
}),
```

反向接口格式

```
{
    "id": course_resources,
    "total_num": num_of_total_resources,
```

```

    "value": [
        {
            "resource_name": resource_name,
            "resource_path": path_to_resource_on_server,
            "resource_size": size_of_resource,
            "resource_upload_time": upload_time_of_resource,
            "resource_type": type_of_resource, // mp4, mp3, rar, pdf, doc, ppt, et
c....
        },
        {
            "resource_name": resource_name,
            "resource_path": path_to_resource_on_server,
            "resource_size": size_of_resource,
            "resource_upload_time": upload_time_of_resource,
            "resource_type": type_of_resource, // mp4, mp3, rar, pdf, doc, ppt, et
c....
        },
        {
            "resource_name": resource_name,
            "resource_path": path_to_resource_on_server,
            "resource_size": size_of_resource,
            "resource_upload_time": upload_time_of_resource,
            "resource_type": type_of_resource, // mp4, mp3, rar, pdf, doc, ppt, et
c....
        },
    ]
}

```



```
SELECT resource_id, resource_type, resource_name, resource_path, insert_date FROM
course_demo001_resource

WHERE

resource_belong = "resources";
```

6.16 NewCourseIssue 接口

该接口出现在 [4.4.2.1.1 新建公告界面](#) 节中。

接口路径：

```
/NewCourseIssue
```

接口 js 函数路径：

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

正向接口格式：

```
data:JSON.stringify({
    "status": "0",
    "message": "new_course_issue",
    "course": course,
    "teacher_id": teacher_id,
    "panel_id": panel_id,
    "type": "announcement\homework\examination\discussion\resource",//发布的是作业、
    公告、考试、讨论区、资源
    "title": "issue_title",
    "content": "content",
    "attachment":[
        {
```

```

        "file_name": file_name,

        "file_path": file_path,

    },

    {

        "file_name": file_name,

        "file_path": file_path,

    },

    {

        "file_name": file_name,

        "file_path": file_path,

    },

]

}),

```

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.....

}

```

Sql 语句

```

/* 6.16 */

SELECT max(issue_id) FROM course_demo001_content WHERE issue_type = 'homework';

```

```
INSERT INTO course_demo001_content(issue_id, issue_type, issue_title, i
ssue_content, issue_attachment_name, issue_attachment_path, issue_deadl
ine, issue_table, issue_belong, insert_date)
```

```
VALUES
```

```
(
    issue_id,
    issue_type,
    issue_title,
    issue_content,
    issue_attachment_name,
    issue_attachment_path,
    issue_deadline,
    issue_table,
    issue_belong,
    LOCALTIME()
)
```

6.17 GETPREVCOURSES 接口

该接口出现在 [3.3PrevCourses](#) 节中。

接口路径:

```
/GetPrevCourses
```

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_prev_courses",

"role": student/teacher/manager,

"person_id": person_id,

}),
```

反向接口格式:

```
{

  "id": "prev_courses",

  "total_num": num_of_total_courses,

  "value": [

    {

      "course_title": course_title,

      "course_teacher": course_teacher,

      "course_hierarchy": course_hierarchy, //2016,2017 级

      "course_college": course_college, //开课学院

    },

    {

      "course_title": course_title,

      "course_teacher": course_teacher,

      "course_hierarchy": course_hierarchy, //2016,2017 级

      "course_college": course_college, //开课学院

    },

    {

      "course_title": course_title,

      "course_teacher": course_teacher,

      "course_hierarchy": course_hierarchy, //2016,2017 级

      "course_college": course_college, //开课学院

    },

  ],

}
```

//项目数量不定，需要遍历访问

```
]
}
```

Sql 语句

```
/* 6.17 */

SELECT courses.course_id, courses.course_name, courses.course_hierarchy, courses.course_semester FROM courses, students_courses

WHERE students_courses.student_id = "stu20200201"

AND

students_courses.course_id = courses.course_id ;
```

6.18 GETSTUDENTHOMEWORKS 接口

该接口出现在 [4.4.2.2 已提交作业查看列表设计](#) 节中。

接口路径:

```
/GetStudentHomeworks
```

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_student_homework",
    "course_id": course_id,
    "homework_id": homework_id,
```

```
)),
```

反向接口格式:

```
{
  "total_num": num_of_total_homeworks,
  "homeworks":[
    {
      "student_id": student_id,
      "homework_title": homework_title,
      "homework_content": homework_content,
    },
    {
      "student_id": student_id,
      "homework_title": homework_title,
      "homework_content": homework_content,
    },
    {
      "student_id": student_id,
      "homework_title": homework_title,
      "homework_content": homework_content,
    },
    {
      "student_id": student_id,
      "homework_title": homework_title,
      "homework_content": homework_content,
    },
  ]
}
```

```
}
```

6.19 GETSTUDENTHOMEWORKDETAIL 接口

该接口出现在 [4.4.2.2.3 学生提交的作业具体详情界面](#)处。

接口路径:

```
/GetStudentHomeworkDetail
```

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_student_homework_detail",
    "course_id": course_id,
    "homework_id": homework_id,
    "student_id": student_id,
}),
```

反向接口格式:

```
{
    "studnet_name": student_name,
    "studnet_email": student_email,
    "student_avatar": avatar_path,
    "submitted_time": submitted_time,
```

```
"homework_title": homework_title,  
"homework_content": homework_content,  
"attachments_num": num_of_total_attachments,  
"homework_attachments": [  
    {  
        "attachment_name": attachment_name,  
        "attachment_path": attachment_path,  
    },  
    {  
        "attachment_name": attachment_name,  
        "attachment_path": attachment_path,  
    },  
    {  
        "attachment_name": attachment_name,  
        "attachment_path": attachment_path,  
    },  
]  
}
```