Wang Qixin 教授，您好，

我是HONGXincong（24028069d）是您目前这个项目的学生An online reservation system for teaching。 您发的git hub我也看了，我有以下几个意见能提出来。

1. 缺乏登录机制， 代表所有人都可以用这个系统创建Time slot, 假如有爬虫攻击，整个db会被塞满。
2. 创建完了之后，可以频繁发送邮件 骚扰学生。
3. 假如是first in first come, 那创建多个time slot以及mapping没意义，大家抢到了就会是那个时间了。假如有A B C三个时间， 抢B的比例多，那么老师会决定用B的时间。那这样的话不用mapping了， 直接老师在最后选择人多的时间，并且一键结合，另外两个class的学生也会被编排到这个班别
4. **可视化时间轴 UI**：学生直观选择时间，不用下拉框。
5. 显示系统如何保证公平（如“你排在第3位提交，因此优先级较高”）。类似现时学校的学科注册系统，增加waiting list 的可能性
6. 邮件并未显示学科的详细信息，在多个科目须要预定的时候导致学生混乱。并且每个booking 都要新的密码，增加用户的工作量。 系统的可拓展性也不强，比如未来直接在系统发送成绩。学生又要翻找邮件里面的密码。 建议使用登录了一次后，即刻用多种功能，预定多个学科。

除此之外，我自己在暑假的时候，也开始研究您这个系统（虽然目前只有一些简单的功能）。以下是我自己的产品以及您github上作品的一些对比。以及我如何解决。我想在下次meeting前让你先看看我所做的。以及想取得您的意见，如果可以的话想请你多给我些指示。在此次Project中，我能做些什么，要做些什么。

首先，在我的产品里面，我会假设已经获得了学校数据库。因此学生可以用自己的账户登录这个系统，并且登录的时候需要验证法，解决了学生因泄露自己的学生号以及老师给的密码，而导致被人恶意创建booking。也就是 1.所提的问题。并且学生用自己原来的账号 以及每次都会有新验证码，也减少了用户翻找过去邮件的时间。方便性以及安全性都大大提升。

而老师页面也是，目前创建新的time slot,并不用任何验证。假如有人恶意滥用这个系统的不断发邮件给学生，学生将会受到骚扰。

一張含有 文字, 螢幕擷取畫面, 軟體, 電腦圖示 的圖片

AI 產生的內容可能不正確。

一張含有 文字, 軟體, 字型, 網頁 的圖片

AI 產生的內容可能不正確。

Teacher side

在老师页面这里，我同calender的形势展示目前登录老师所创建的time slot, 使UI可视化得到提升，变得比以往更加直观。 并且老师创建time slot时，会有一个更加直观的弹窗。一張含有 文字, 螢幕擷取畫面, 軟體, 數字 的圖片

AI 產生的內容可能不正確。

一張含有 文字, 螢幕擷取畫面, 軟體, 電腦圖示 的圖片

AI 產生的內容可能不正確。

Student Site

学生登录系统后，可以通过搜索教授取得该教授的时间表。 绿色的代表可以booking，点击后会有弹窗出现。 成功预定后则变成黄色的。 灰色的则是人满了，不可预定。此方法是更直观的first in first service 算法。

一張含有 文字, 螢幕擷取畫面, 軟體, 網頁 的圖片

AI 產生的內容可能不正確。

一張含有 文字, 螢幕擷取畫面, 軟體, 電腦圖示 的圖片

AI 產生的內容可能不正確。

一張含有 文字, 螢幕擷取畫面, 軟體, 陳列 的圖片

AI 產生的內容可能不正確。

以上是目前我所做的所有功能，在未来可以根据你的指示开发更多功能。

以下是我的开发方法。 我使用docker+html+jquery+Python+mySql

使用python的原因是我能够用用api引入我所写的功能，提升在未来的扩展性。Python也提升了未来系统引入AI 的可能性。



一張含有 文字, 字型, 螢幕擷取畫面 的圖片

AI 產生的內容可能不正確。

老师 您好，我一直在思考一个问题，就是关于学生选择time slot preference的问题。

我们昨天探讨了，这个Project 排程的方法只需要First in First Service，不会综合考虑其他因素。那学生为何还需要选择自己的时间喜好呢？他们可以直接在Deadline之前预定一个想要的时间就好了（老师发布的其中一个）。这样同时也可以减少了数据库的储存。

我们理工大学的健身房目前也是采用这个方法。很直观的显示蓝色是可以预定，灰色是满人了。而每个学生只能预定其中两个。假如这个time slot满人了，则自动变成灰色，不可预定。我觉得这个做法挺适合我们目前这个项目的。 您觉得呢？

一張含有 文字, 螢幕擷取畫面, 軟體, 電腦圖示 的圖片

AI 產生的內容可能不正確。

Book完之后查看同一个号是否有多余两个的booking。再查看自己是第几个加入的，假如大于capacity则返回

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

AI 產生的內容可能不正確。

樂觀鎖例子

<https://blog.csdn.net/u011397981/article/details/131860683>

<https://medium.com/dean-lin/%E7%9C%9F%E6%AD%A3%E7%90%86%E8%A7%A3%E8%B3%87%E6%96%99%E5%BA%AB%E7%9A%84%E6%82%B2%E8%A7%80%E9%8E%96-vs-%E6%A8%82%E8%A7%80%E9%8E%96-2cabb858726d>

模擬多線程官方文件

<https://docs.python.org/3/library/threading.html>

悲觀鎖

https://www.cnblogs.com/goloving/p/13590955.html

**🔑 避免/解決死鎖的方法**

**1. 固定鎖的順序**

* 規定所有 transaction 都要 **按同一順序鎖資源**。
* 例如：不管誰訂票，永遠先鎖 seat1 → 再鎖 seat2 → 再鎖 seat3。  
  ✅ 這樣就不會互相等待了。

**2. 縮小鎖範圍**

* 把 SELECT ... FOR UPDATE 包在最小的 transaction 裡。
* 不要鎖了 seat 還去做其他事（比如算價格、發 email），那會增加被卡的時間。

**3. 設定 Transaction Timeout**

MySQL 提供：

SET innodb\_lock\_wait\_timeout = 5;

意思是：如果一個 transaction 等鎖超過 5 秒，就報錯，不會無限卡住。  
應用程式收到錯誤後，可以 **回滾 (rollback) + 重試**。

**4. 使用樂觀鎖代替悲觀鎖（有些情境）**

* 悲觀鎖：一開始就鎖住 → 容易死鎖。
* 樂觀鎖：先查 seat 狀態，最後更新時用 WHERE seat\_id=? AND version=? 保證不被改過。
* 適合「衝突少」的場景，但在高併發搶票，失敗率會高。

**📌 總結**

* **死鎖一定會發生**（在高併發下），所以不能只靠 DB，要在應用層做好 **timeout + 重試**。
* 常見做法：
  1. 設定 innodb\_lock\_wait\_timeout
  2. Transaction 失敗時 rollback + 重試
  3. 固定鎖順序

https://socket.io/zh-CN/docs/v4/emitting-events/

https://blog.p2hp.com/archives/10826

<https://python-socketio.readthedocs.io/en/latest/server.html>

<https://medium.com/@parth116.rejoice/socket-io-with-python-fast-apis-1a1956b3989a>

一張含有 文字, 螢幕擷取畫面, 軟體, 電腦圖示 的圖片

AI 產生的內容可能不正確。

這個是我在老師的界面一張含有 文字, 軟體, 電腦圖示, 作業系統 的圖片

AI 產生的內容可能不正確。 一張含有 文字, 螢幕擷取畫面, 軟體, 網頁 的圖片

AI 產生的內容可能不正確。 一張含有 文字, 螢幕擷取畫面, 軟體, 網頁 的圖片

AI 產生的內容可能不正確。

一張含有 文字, 螢幕擷取畫面, 軟體, 設計 的圖片

AI 產生的內容可能不正確。

一張含有 文字, 螢幕擷取畫面, 字型, 軟體 的圖片

AI 產生的內容可能不正確。 一張含有 文字, 螢幕擷取畫面, 軟體, 網頁 的圖片

AI 產生的內容可能不正確。

**UI/UX 優化（UI/UX Enhancement）**

新版系統在介面設計上進行了全面重構，整體風格更簡潔、模組化，並強化了互動性與可視化呈現，使教師與學生都能更直觀、高效地操作系統。

在\*\*教師端（Teacher Side）\*\*方面，系統新增了可視化的課程時段管理功能。教師可透過「Create Time Slot」頁面使用彈窗式介面快速建立課程時段，直接輸入標題、開始與結束時間、課程容量、截止日期等資訊，並可同時新增多個時段，大幅簡化原本多步驟的操作流程。  
建立完成後，教師可於「View Time Slot」頁面查看所有課程安排，系統會依據不同的時段群組（Group ID）自動分類顯示，並以卡片式介面呈現課程資訊，如課程名稱、時間、內容與可容納人數。教師可在每張卡片下方直接點擊「View Students」查看該時段的學生名單，並能於列表中進行簽到（Check-in）或新增學生操作，實現更高效率的課程管理。

在\*\*學生端（Student Side）\*\*方面，整體介面設計更貼近實際使用情境。學生登入後，可在「Available Time Slots」頁面中直接瀏覽所有可預約課程時段，系統以表格形式清楚列出課程標題、教師姓名、上課時間、剩餘名額與截止日期。  
學生只需點擊「Book」按鈕即可完成預約，預約成功後可於「My Coming Booking」頁面即時查看已登記課程。  
同時，系統提供「Waiting Booking」模組，協助學生追蹤尚未確認的預約申請，使整體預約流程更透明、明確。

整體 UI 採用紅白對比配色與統一字體風格，頁面佈局乾淨一致；彈窗式操作與卡片式呈現結合提升了使用的流暢度與視覺層次。  
無論在教師端或學生端，系統皆能即時回饋操作結果，並支援動態更新（如剩餘名額自動刷新），提升了互動性與系統響應速度。

此版本的 UI/UX 優化不僅提升了操作體驗與資訊清晰度，也為後續功能擴展（例如郵件通知整合、即時更新推播與課程整合管理）提供了穩定的設計基礎。