Software Engineering Program

Sino-British Collaborative Education

CDUT

Learning Management System Web Site

- Preliminary Report -

|  |
| --- |
| **Student ID:** |
| **Date:** 28th April 2024 |
| **Group:** |

Table of Contents

[1. Introduction: 3](#_Toc160390204)

[2. Database design: 4](#_Toc160390205)

[3. Wireframes: 5](#_Toc160390206)

[4. Functionality of client and server 6](#_Toc160390207)

[5. References: 7](#_Toc160390208)

# Introduction:

\* In the introduction of an initial report for a web application development module, you should provide an overview of the topic, including its objectives, scope, and significance. This section sets the stage for the reader, outlining what the web application aims to achieve and why it is important. Additionally, you can introduce the technologies and methodologies used in the development process and provide a brief overview of the structure of the report.

The advent of digital technology has changed the educational landscape, introducing innovative solutions to enhance learning experiences and administrative efficiency. This report introduces a web-based Learning Management System (LMS), a comprehensive platform tailored to facilitate the administration, delivering, and monitoring of educational courses.

The scope of this project encompasses the development and testing of a full-stack application that enables: instructors can efficiently manage course content and assessments, students can engage with the material and track their progress, and administrators can oversee the system's operation and user management.

The significance of the LMS lies in its potential to facilitate a seamless educational experience, bridging the gap between traditional and digital learning. It aims to enhance the accessibility of educational resources, improve communication between educators and learners, and provide a centralized platform for academic management.

The LMS is developed using HTML5 and ECMAScript 6 for the client-side, ensuring a responsive and interactive user experience. The server-side is powered by Flask, MySQL, and Python 3, forming a robust backend capable of handling complex data transactions and user interactions. The choice of these technologies is driven by their proven reliability and flexibility in web application development.

The report is structured to provide an overview of the Learning Management System (LMS) details. It starts with an exploration of the database structure, illustrating table relationships and key data fields. Following this, it presents wireframes depicting the potential visual layout of the website. Lastly, it outlines the client-server interactions and technologies required to implement the LMS functionality.

In conclusion, this introduction serves as a precursor to the detailed exploration of the LMS development journey. It emphasizes the system's objectives, scope, and the thoughtful application of technologies to achieve a transformative educational platform.

Preliminary report

● Repeating what is stated in the specification as if it was your decision rather than what you have been told to do.

● Telling the story of how a function will be implemented rather than how the implementation will work.

● Failing to break down tasks into client and server components or attributing all of a multi-stage task to one of these; for example “the student logs into the site” which actually requires multiple steps and cannot be purely done on the client.

● Describing what is stored on the client or server, but not what functionality they have.

● Over-emphasising the choice of technologies, rather than how they are used.

● Writing many-to-one relationships with the foreign key on the wrong side.

● Trying to store foreign keys to two different tables in the same field without distinguishing them.

● Not showing data table descriptions, only sample data that does not formally indicate structure.

● Omitting field types from specifications.

● Stating that the “user will be logged in” without indicating what method is used to signal this.

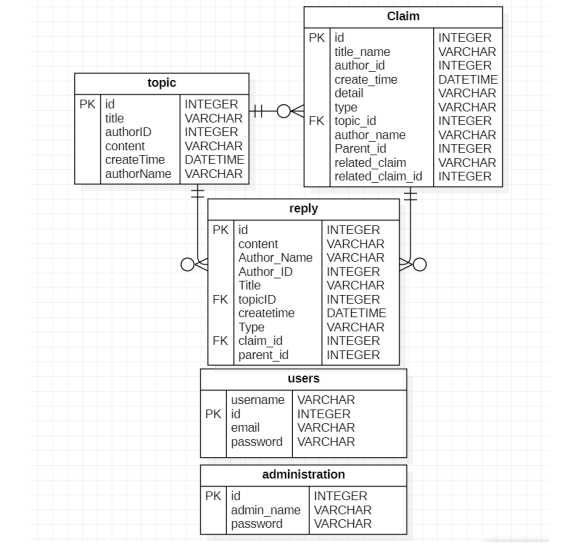
● Large amount of dead space on UI designs.

● Showing UI prototypes instead of wireframes.

● Where search is implemented, not providing a wireframe for the search results page.

● Where AJAX is used, not providing a structure for the JSON messages that will be used.

# Database design:



\* Cover all tables intended for use and provide the correct fields and relationships between them.

学生的课程列表，教师的上传列表。。-学生应该看到他们注册的课程、成绩和即将完成的作业。

教师和学生可以通过消息传递相互沟通

●通知:接收新公告、截止日期的提醒。

●搜索功能:搜索系统内的课程、讲座和其他用户

管理:

学生:

←Course Enrollment: Look over the offered courses and request to be enrolled in those they would like to take (only administrators can decide whether to accept or enroll a student).

课程注册:查看所提供的课程，并申请参加他们想参加的课程(只有管理员可以决定是否接受或注册学生)。

lContent Access: Access enrolled courses, lectures, assignments.

内容访问:访问已注册的课程、讲座、作业。

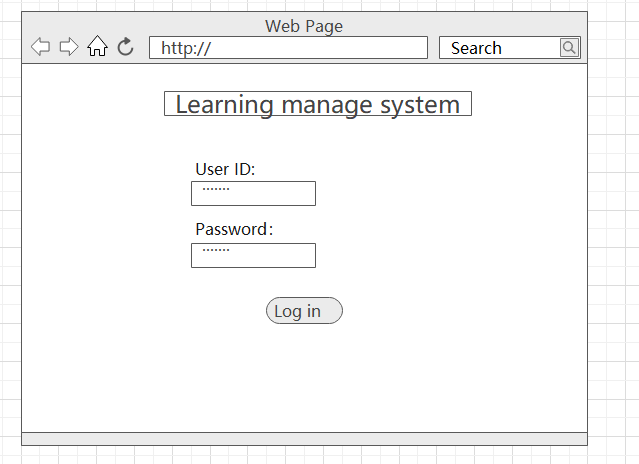
Submission: Submit assignments, within specified deadlines.

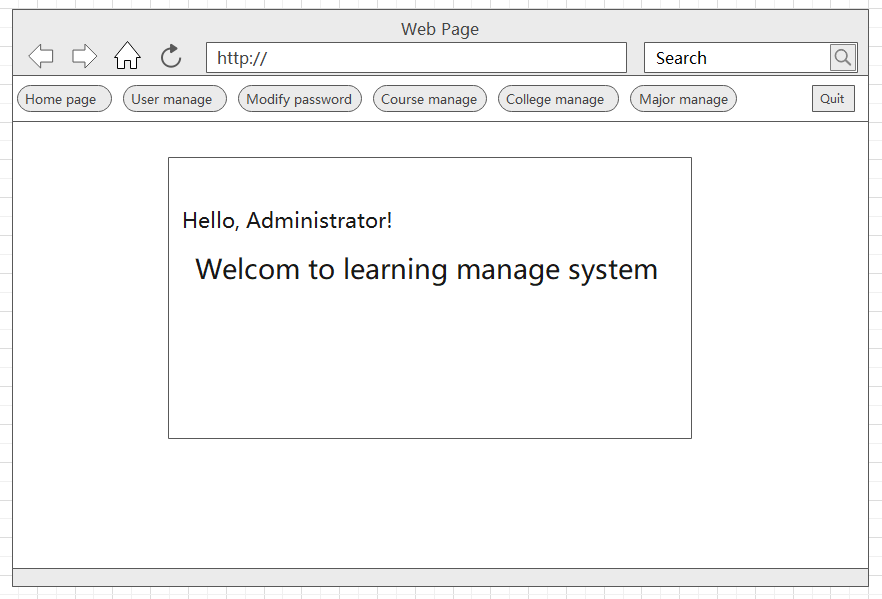
提交:在规定的期限内提交作业。

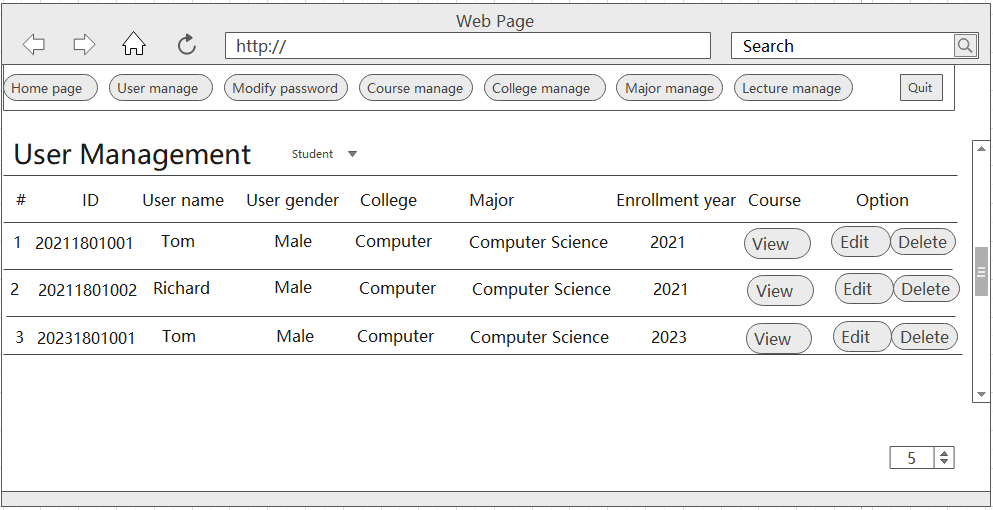
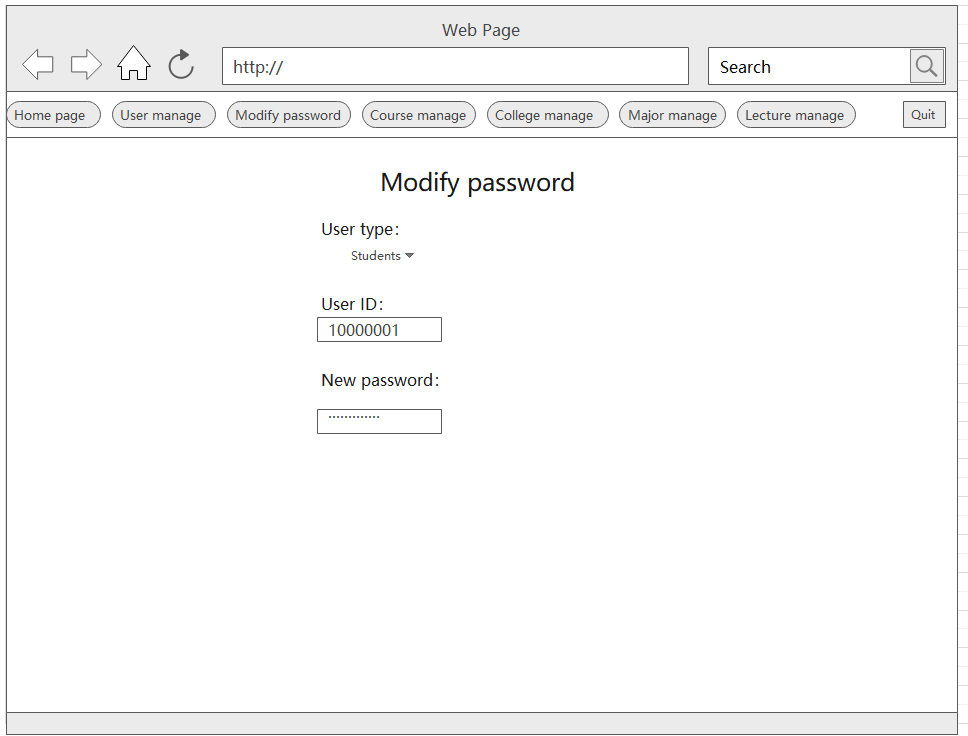
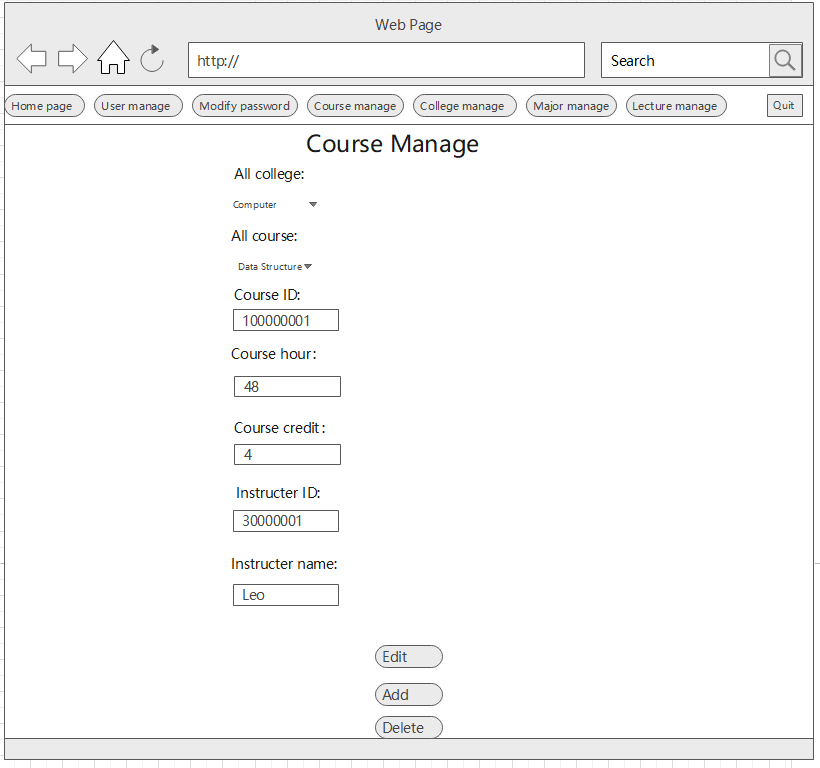
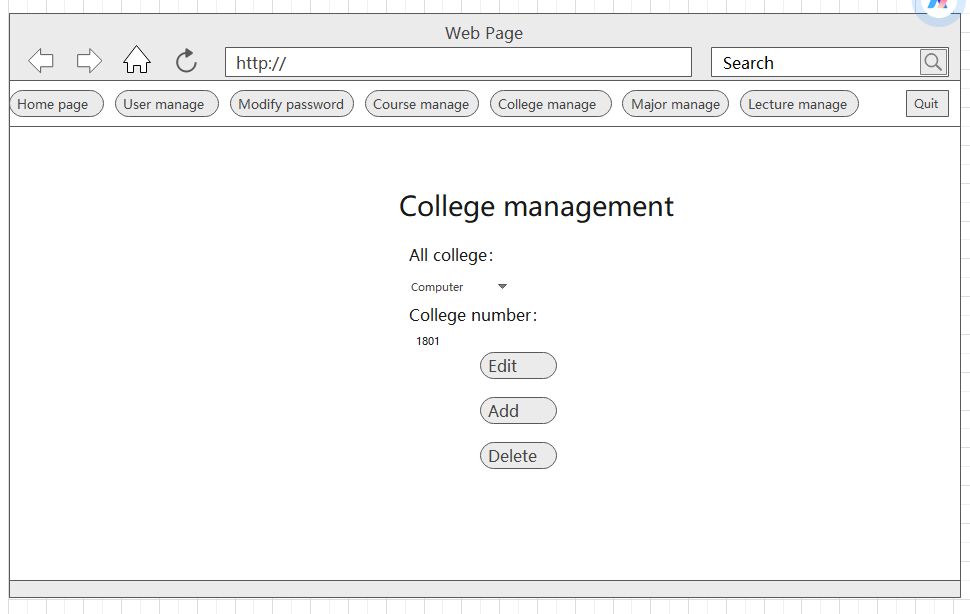
Feedback and Grades: Receive feedback and grades on assignments and assignments.

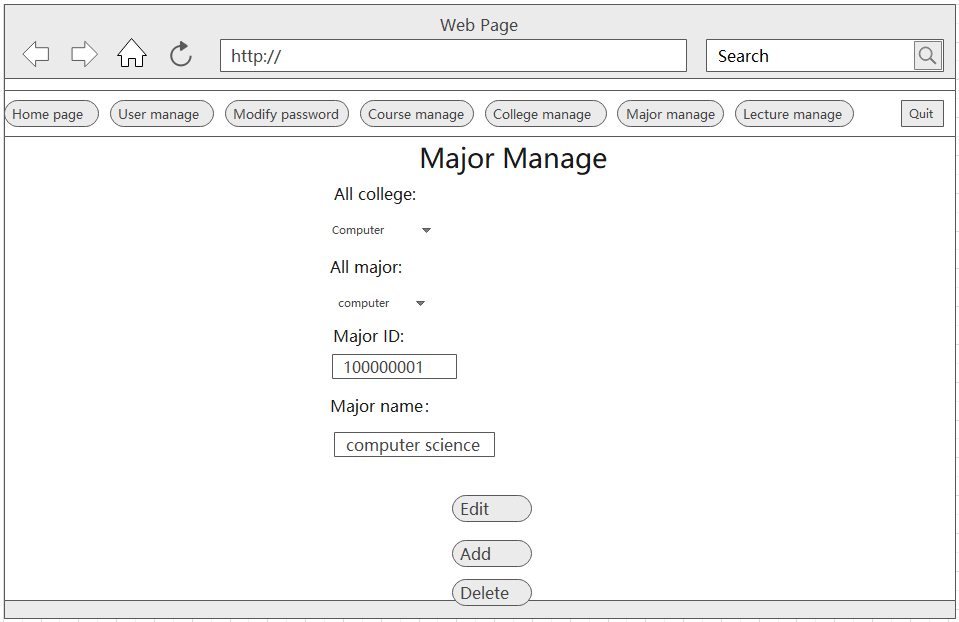
反馈和评分:接收作业和作业的反馈和评分。

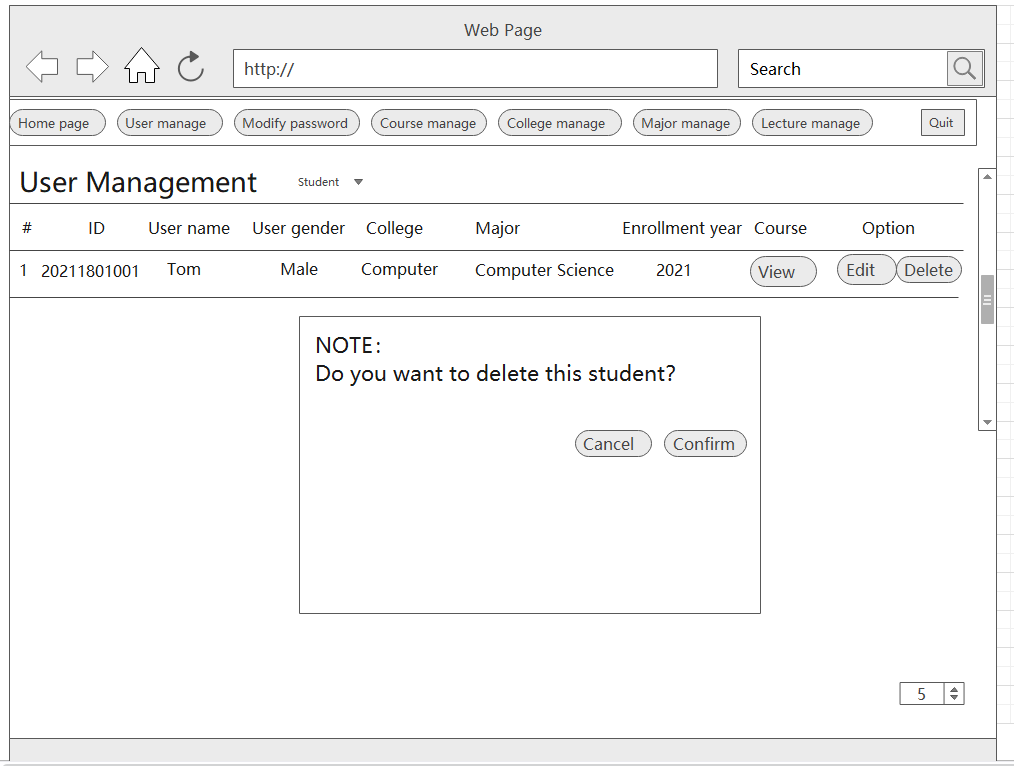
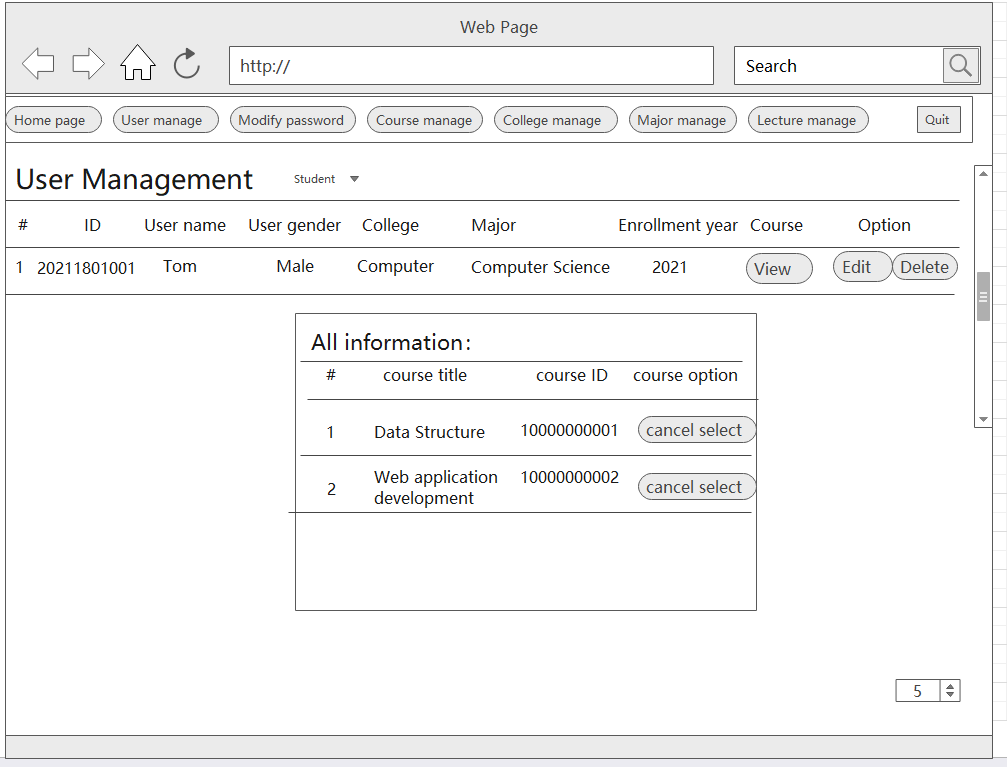
# Wireframes:

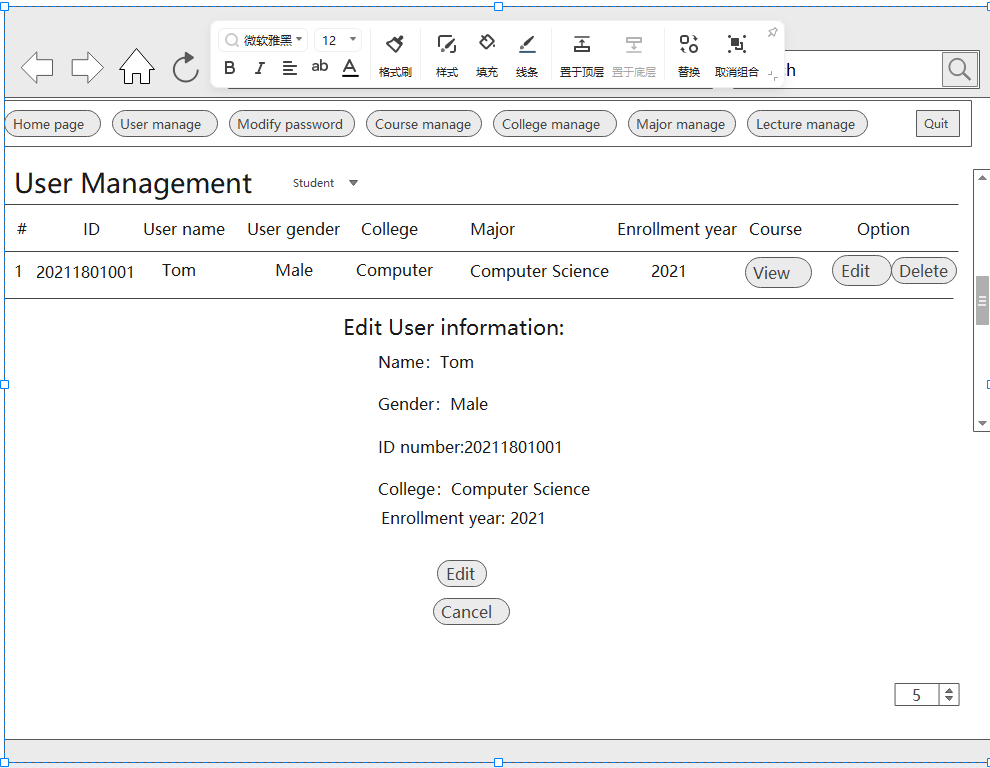
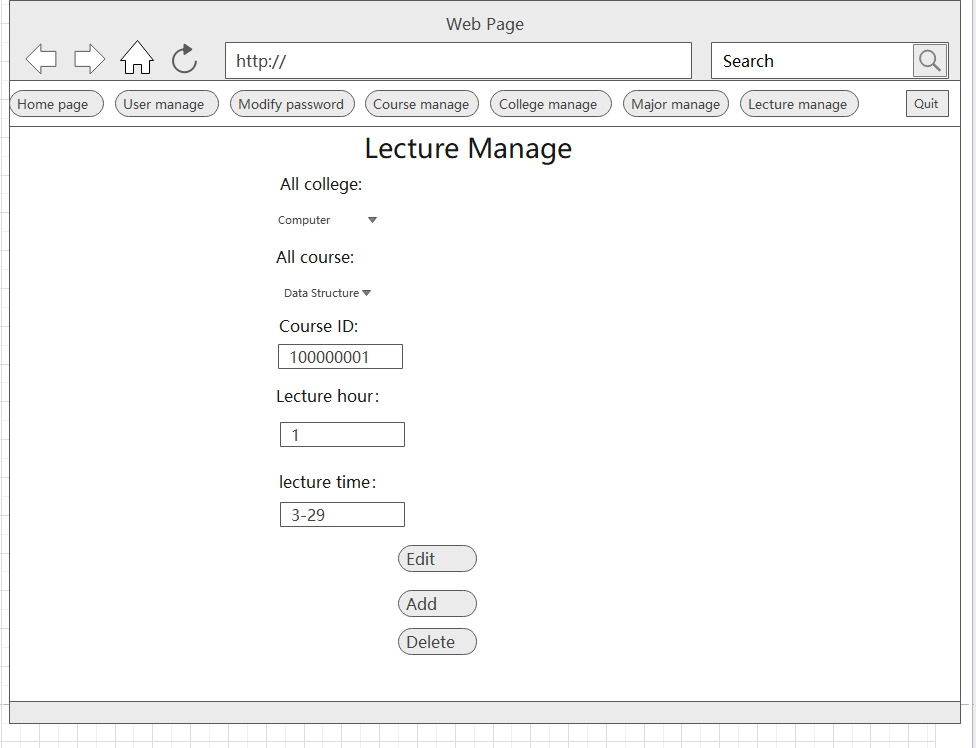
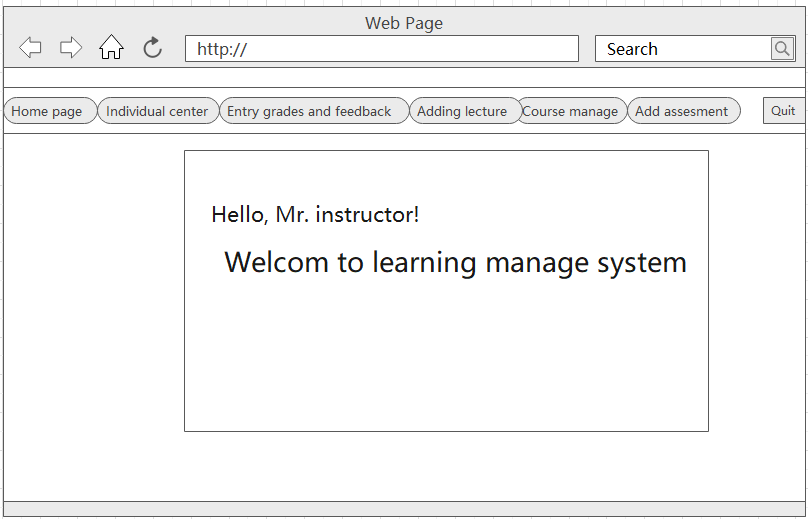
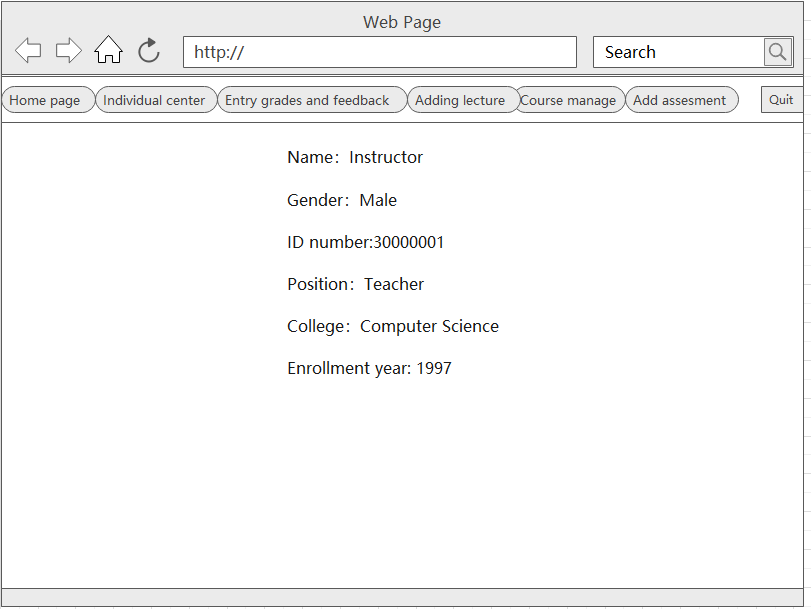
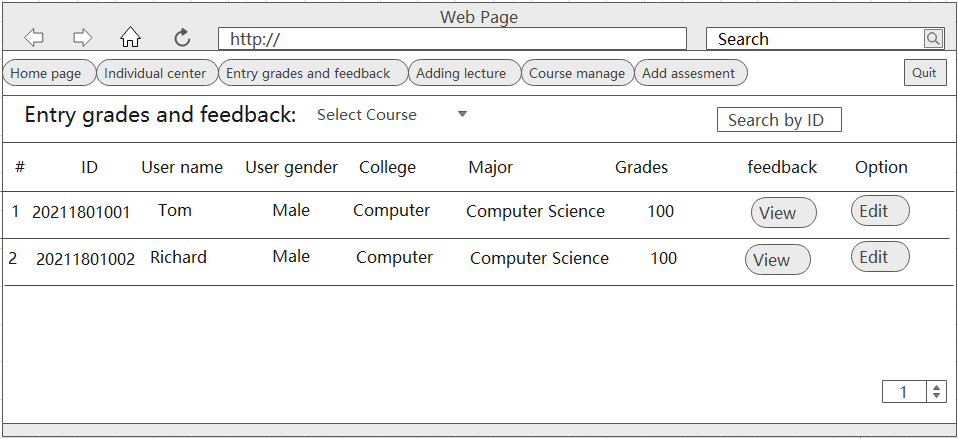
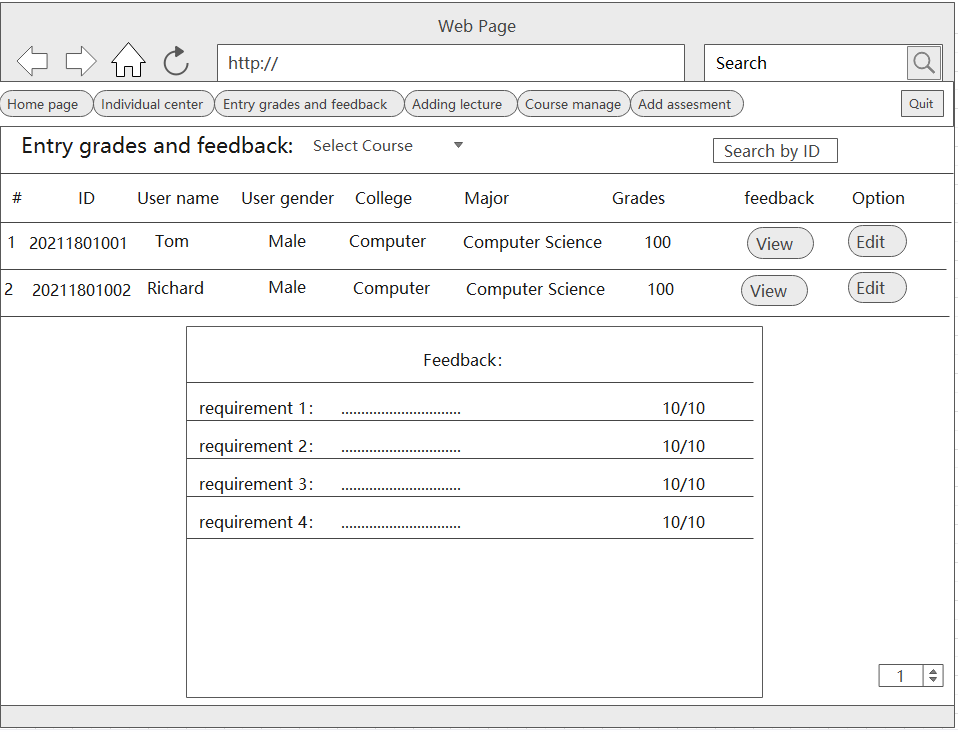
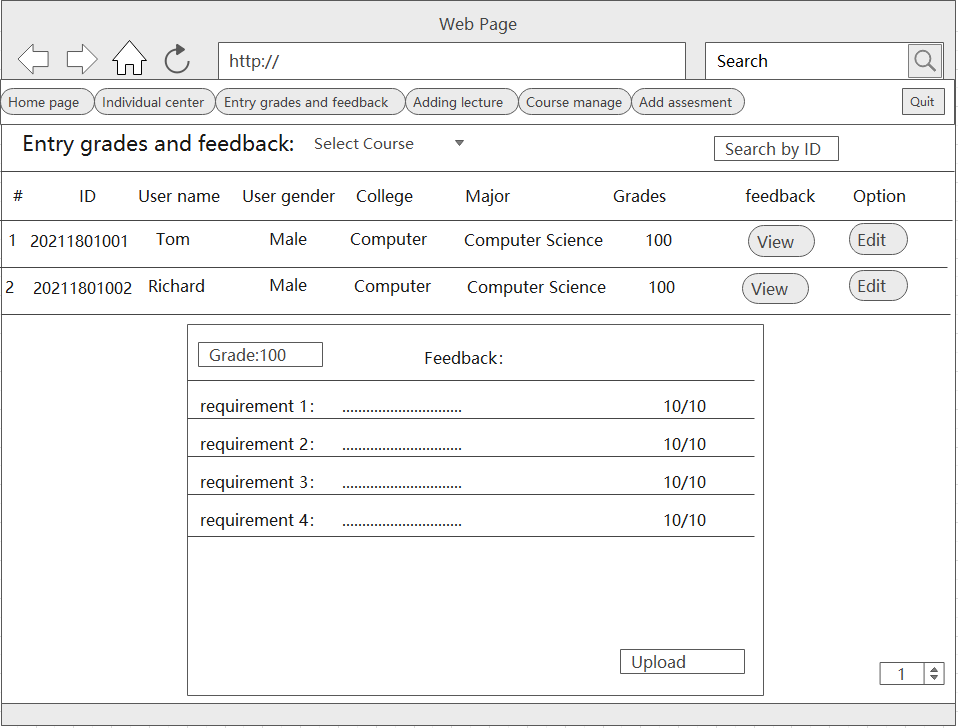
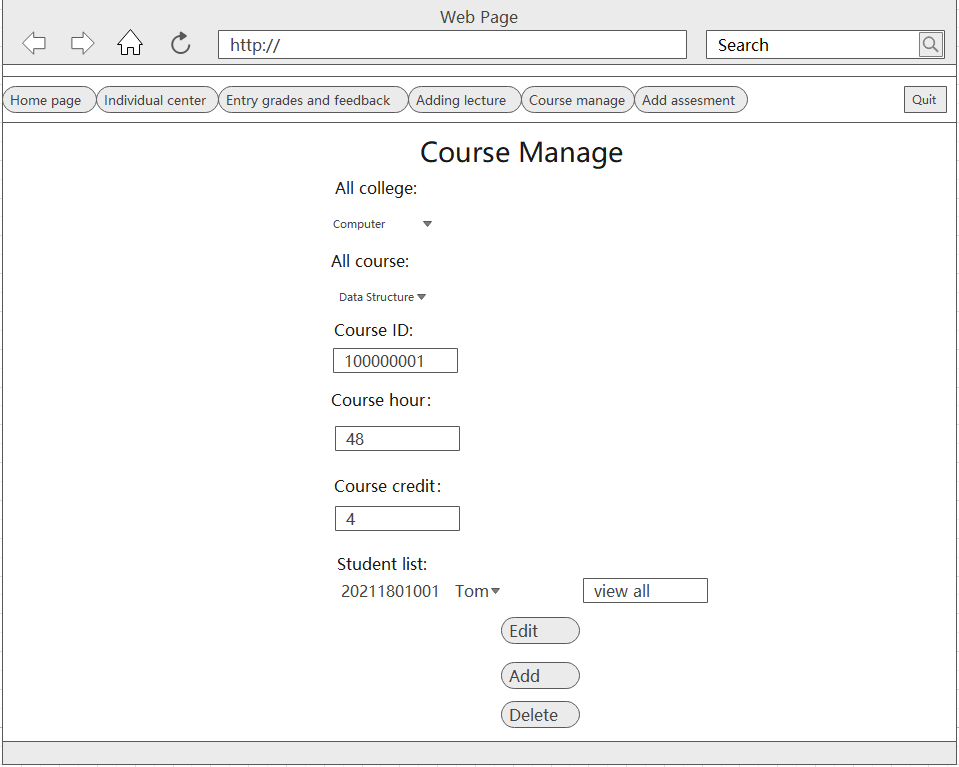
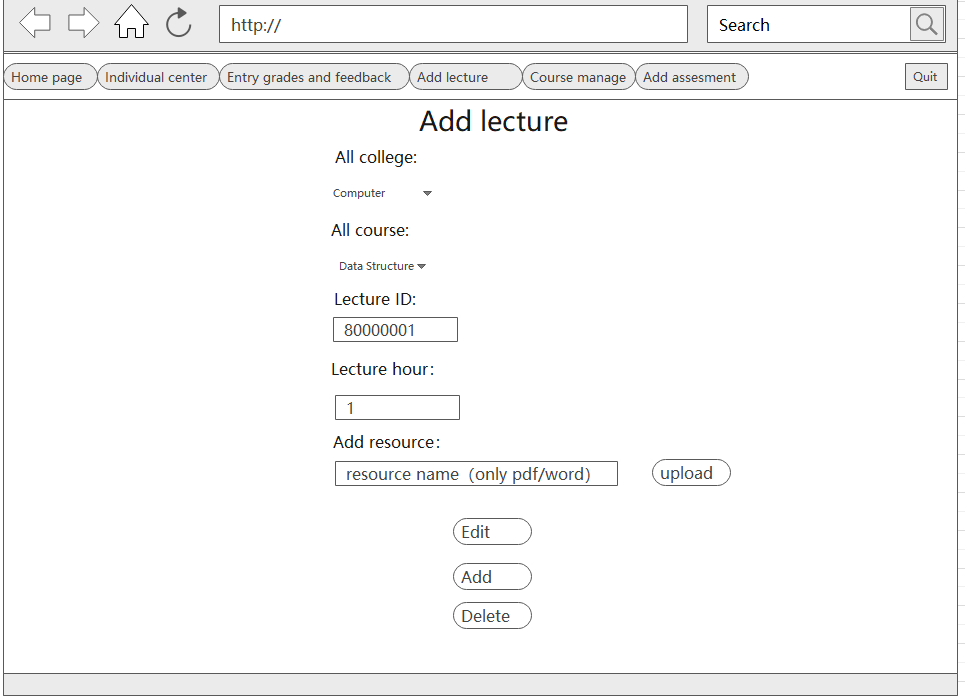
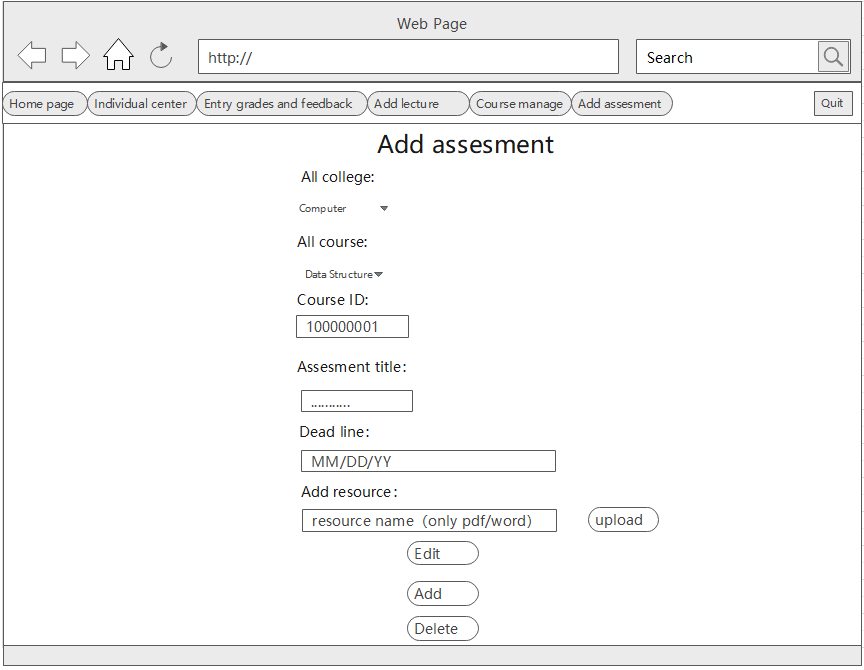
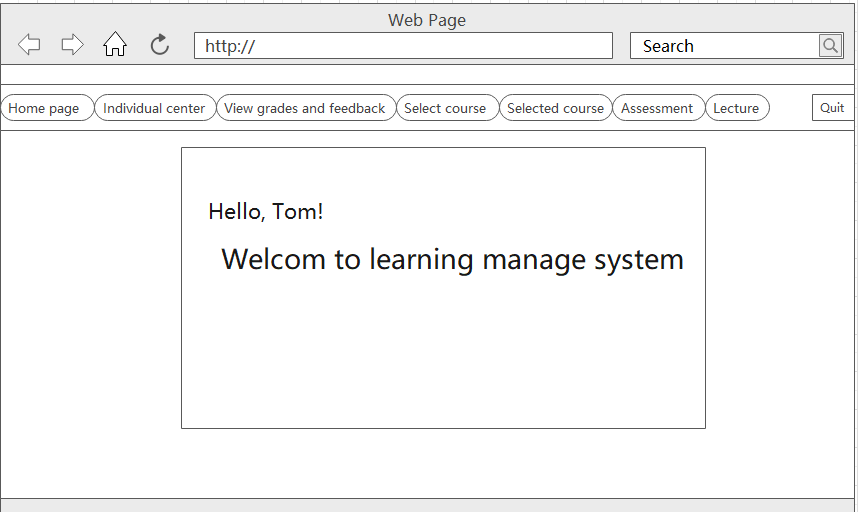
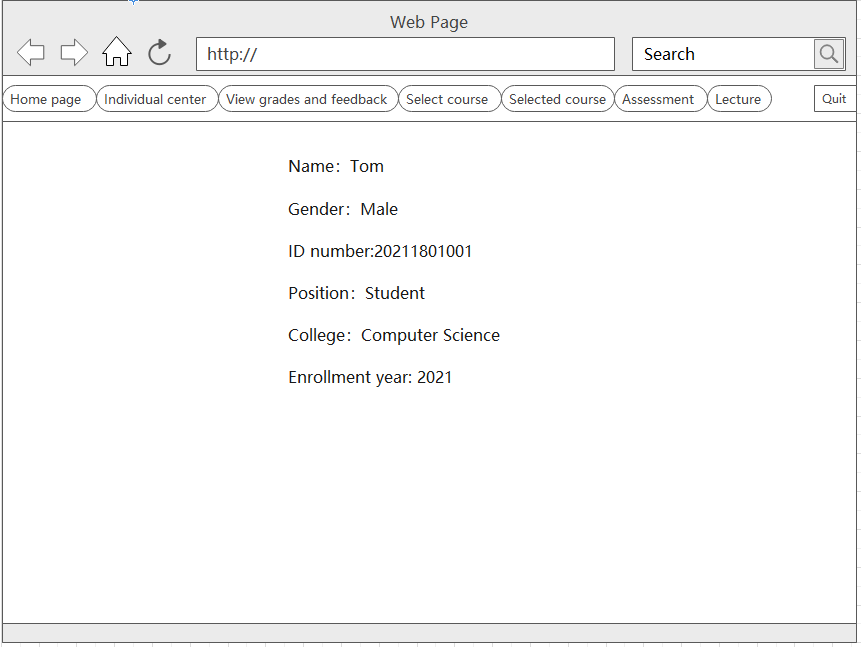
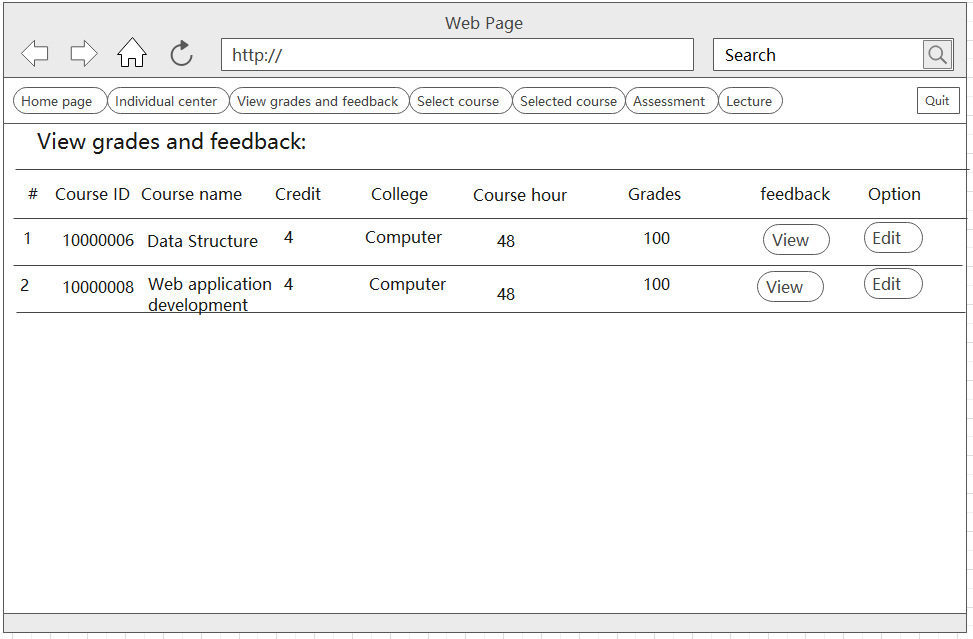
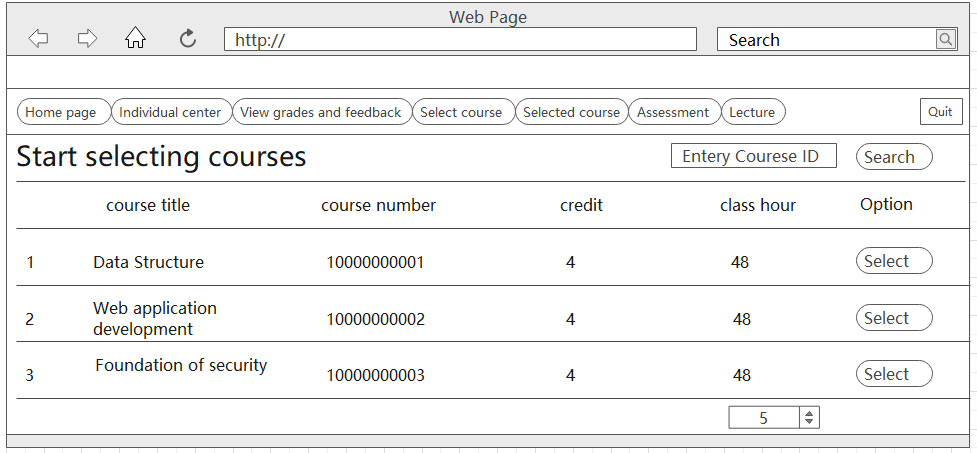
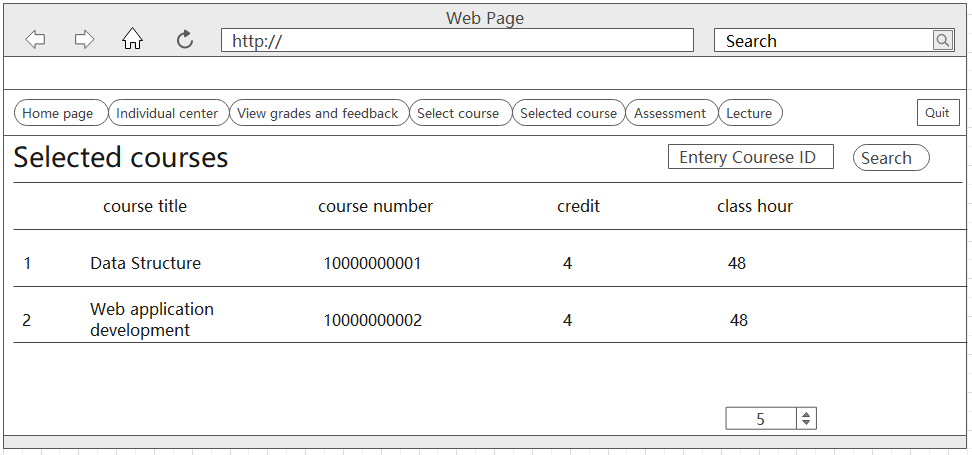
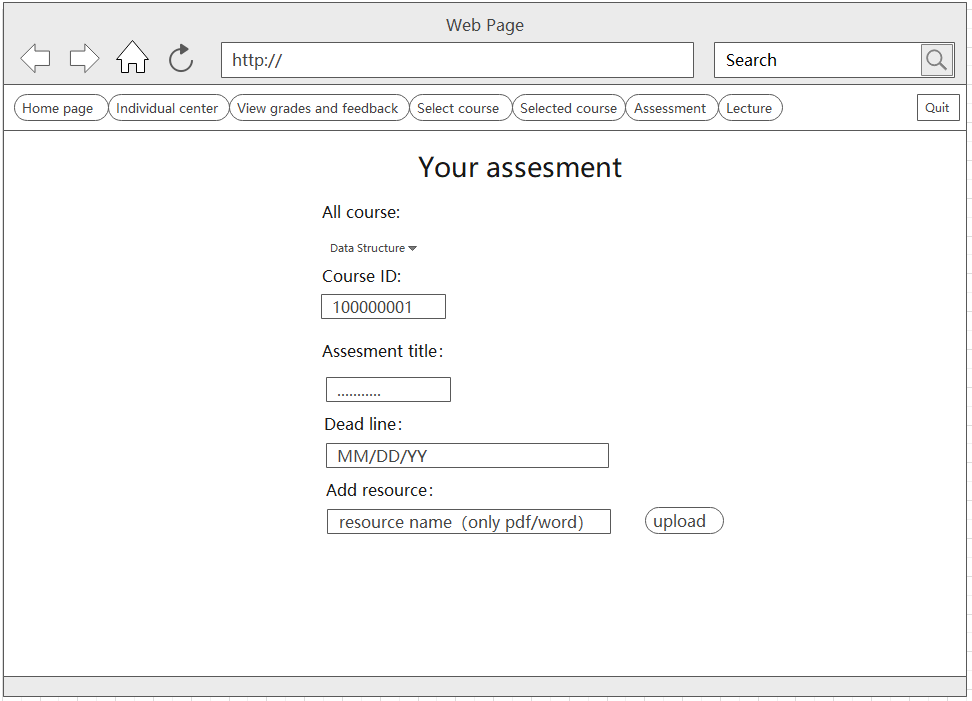
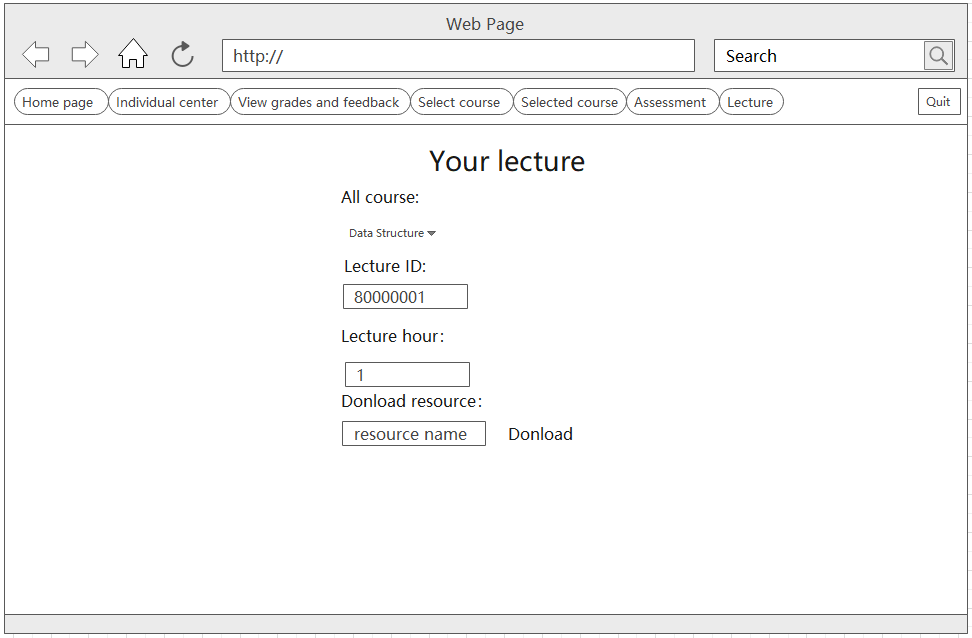
* 1. Log in
  2. Home page （administer）



* 1. User management（admin）
  2. Modify password（admin）
  3. Course manage（admin）
  4. College management
  5. Major manage（admin）



* 1. User manage（delete）
  2. User manage （view option）

1. User manage （Edit）
2. Lecture manage（admin）
3. Home page(instructor) 
4. Individual center 
5. Entry grades and feedback
6. Entry grades and feedback（feedback view）
7. Entry grades and feedback（edit）
8. Course manage（instructor）
9. Adding Lecture（instructor）
10. Add assessment 
11. Home page（student）
12. Individual center（student）
13. View grades and feedback 
14. Select course 
15. Selected course 
16. Assessment(student) 
17. Lecture 

# Functionality of client and server

* 1. Register

**Client:** HTML+ CSS+ JavaScript ( jQuery+ DOM)

As same as log in ；registration page requires username, password, password2, email

<Using Method1 in 4.1.1 to transfer the data>

**Server:** Flask +MySQL+render\_template+flash+request

"POST" request to the backend "register" route. Write code in the view function to determine the type of request. if request.method=="POST", then capture the user input and determine if the user input is complete by using "all([username,password,password2,email])", followed by the user name is duplicated (by comparing the user's input with the data stored in the database), and the password is the same for both entries. Feedback on the user's input is given via "flash". If this is correct, To ensure the security of the user's password, a hash transcoding ("generate\_password\_hash") is used so that the user's password is not visible in the database，and then add the user information to the database (newUser = users(username = username,password=password,email=email) db.session.add(newUser)

* 1. Login

**Client:** HTML+ CSS+ JavaScript (jQuery+ DOM)

use jQuery to make pop-up boxes, write the login and registration code in html, and use jQuery to manipulate DOM to show and hide. Clicking on the "login" button will bring up the interface, avoiding page jumps. The user enters information on the login/registration page (login page requires username and password; and then clicks the "submit" button, which sends "POST" to the backend. <Using Method1 in 4.1.1 to transfer the data>

**Server:** Flask +MySQL+render\_template+flash+request

* "POST" request to the backend "login" route. Write code in the view function to determine the type of request
* determine if the username and password entered by the user and the data stored in the database match (users = users.query.filter(users.username== username, users.password==password).first()), if they do, the login is successful. If it does not match, the user is given feedback via "flash": ""Wrong password"" etc.

\* Try also to cover all possible functions in your website.

# Conclusion:

\* Summarize the main points covered in the report, discuss whether the project objectives were met and evaluate the success of the implementation. Reflect on any challenges encountered during the development process and discuss lessons learned. Additionally, consider the potential for future improvements or enhancements to the web application.

# References:

\* If there are any references used, include them as well.