Final Report on SDSC5003 Database Design

1. Group Members and Undergraduate Majors

Member 1: ZHAO Zihao, Computer Science
Member 2: YANG Jinyan, Finance
Member 3: CHEN Yanxi, Data Science and Big Data Technology
Member 4: WANG Weiyi, Software Engineer
Member 5: SHEN Qiyuan, Statistics

2. Contribution of Each Group Member

ZHAO Zihao (Part 1: Topic Selection, Requirements Analysis and Recording Demonstration Video)

Determined the project theme and objectives, conducted requirement analysis to define the data storage needs and user-required functionalities of the database. Recorded a demonstration video.

YANG Jinyan (Part 2: Database Design, Frontend Page Design)

Selected the type of database and designed the logical structure, including table structures, field types, primary keys, foreign keys, etc. Conducted normalization to reduce data redundancy. Designed application program pages.

CHEN Yanxi (Part 3: Creating Database and Tables, Data Input, Function Implementation)

Created the database using a database management system (in MySQL) and established tables with field constraints. Input data into the database based on the design. Wrote SQL statements or used the interface provided by the database management system to implement CRUD operations and more complex queries and data processing functions.

WANG Weiyi (Part 4: Writing Application Program)

Developed a frontend application to implement data interaction (in Python) and database APIs to realize application logic.

SHEN Qiyuan (Part 5: Final Report)

Prepared the final report summarizing the project, including the design, implementation, and outcomes.

3. Software Architecture

Tech Stack
- Python 3.8+
- MySQL 5.7+
- Flask
- HTML/CSS/JavaScript

```
5003/
│
├── README.md # Project documentation
├── requirements.txt # List of Python dependencies (for reference)
├── run.py # Entry point for the application (Flask app)
│
├── project/ # Main project directory
│   ├── requirements.txt # List of Python dependencies (for reference)
│   ├── run.py # Application entry point
│   │
│   ├──── app/ # Core application logic
│   │   ├──── __init__.py # Initializes the app package
│   │   ├──── config.py # Configuration settings for the app
│   │   ├──── db.py # Database connection and queries
│   │   ├──── routes.py # Application routes and view functions
│   │   │
│   │   ├──── static/ # Static files (CSS, JS, Images)
│   │   │   ├──── css/ # CSS files
│   │   │   │   └──── style.css# Main stylesheet
│   │   │   │
│   │   │   ├──── js/ # JavaScript files for dynamic pages
│   │   │   │   ├──── admin_dashboard.js
│   │   │   │   ├──── login.js
│   │   │   │   ├──── register.js
│   │   │   │   ├──── student_dashboard.js
│   │   │   │   └──── teacher_dashboard.js
│   │   │
│   │   ├──── templates/ # HTML templates (Flask rendering)
│   │   │   ├──── admin_dashboard.html
│   │   │   ├──── login.html
│   │   │   ├──── register.html
│   │   │   ├──── student_dashboard.html
│   │   │   └──── teacher_dashboard.html
│   │
│   ├──── database/ # SQL files for database setup and queries
│   │   ├──── Inquiry.sql # SQL script for querying data
│   │   ├──── Insert_data.sql # SQL script for inserting data
│   │   └──── Create_tables.sql# SQL script for creating tables
│
└──── .gitignore # Files and directories to ignore in Git
```

3.1 Backend

The backend is built using the Flask framework, a lightweight and flexible Python web framework that facilitates rapid development and deployment. Key components include:

- **Routing**: Maps URLs to specific functions handling requests, ensuring clear endpoint definitions and easy management.
- **Controllers**: Process requests, interact with the database, and return responses. Organized to handle different types of requests (e.g., GET, POST, PUT, DELETE) for various resources.
- **Services**: Encapsulate business logic, separating concerns and maintaining clean, maintainable code.
- **Models**: Represent the database schema using MySQL, an open source RDBMS that uses SQL to create and manage databases.

3.2 Database

MySQL serves as the relational database management system (RDMBS), chosen for its robustness, reliability, and extensive community support. Key aspects of the database design include:

- **Schema Design**: Carefully structured tables, columns, data types, primary keys, foreign keys, and constraints to ensure data integrity and efficient querying.
- **Normalization**: Reduces data redundancy and improves integrity by organizing data into multiple related tables with established relationships.

- **Indexes**: Created on frequently queried columns to enhance query performance.
- **Query Optimization**: Ensures efficient data retrieval through appropriate SQL queries, index utilization, and optimized joins and subqueries.

3.3 Frontend

The frontend is developed using HTML, CSS, and JavaScript, providing a dynamic and user-friendly interface. Key components include:

- **HTML**: Structures web page content, dynamically populated with data from the backend.
- **CSS**: Styles web pages for a visually appealing and responsive design, with frameworks like Bootstrap streamlining the process.
- **JavaScript**: Adds interactivity and dynamic behavior, utilizing frameworks like jQuery, React, or Vue.js for client-side logic and AJAX requests.

3.4 Integration

The backend and frontend communicate through RESTful APIs, facilitated by Flask's jsonify function for JSON response generation.

4. Database Design

4.1 Overview

This project is a student course management system designed to help students register for courses, view course information, and provide teachers with the ability to manage courses and view course enrollment. Administrators can manage student and teacher accounts, as well as course information.

4.2 Requirement Analysis

Student Features

1. Register and Log In:

- Students can create an account by providing necessary details such as name, email, and password.
- Once registered, they can log in to access their personalized dashboard.

2. Search for Available Courses and Enroll:

- Students can browse a catalog of available courses.
- They can search for specific courses using keywords, course codes, or categories.
- After finding a course of interest, students can enroll in it, which updates their course list and schedule.

3. View Enrolled Courses:

- Students can view all the courses they have enrolled in, including details such as course name, instructor information, schedule, and course materials.
- They can also track their progress and grades within these courses.

Teacher Features

1. Manage Course Information:

- Teachers can create, update, and delete course information, including descriptions, syllabi, prerequisites, and grading criteria.

2. View Student Course Enrollments:

- Teachers can view the list of students enrolled in their courses.
- This feature enables them to manage class rosters, track attendance, and communicate directly with students.
- Teachers can also view student performance metrics and grades.

Administrator Features

1. Manage Student and Teacher Accounts:

- Administrators oversee the user base, including students and teachers.
- They can create, modify, and delete user accounts, manage roles and permissions, and handle password resets and account recovery.

2. Manage Course Information:

- Administrators have overarching control over all course information across the platform.
- They can approve new courses, manage schedules, and ensure offerings align with the institution’s objectives.

3. View Course Enrollment Statistics:

- Administrators can access detailed analytics and statistics related to course enrollments.
- This includes data on student enrollment trends, course popularity, and performance metrics, aiding in informed decision-making regarding course offerings and strategies.

4.3 Logical Design

Based on the requirements above, we developed the ERD of our database as figure 1 .

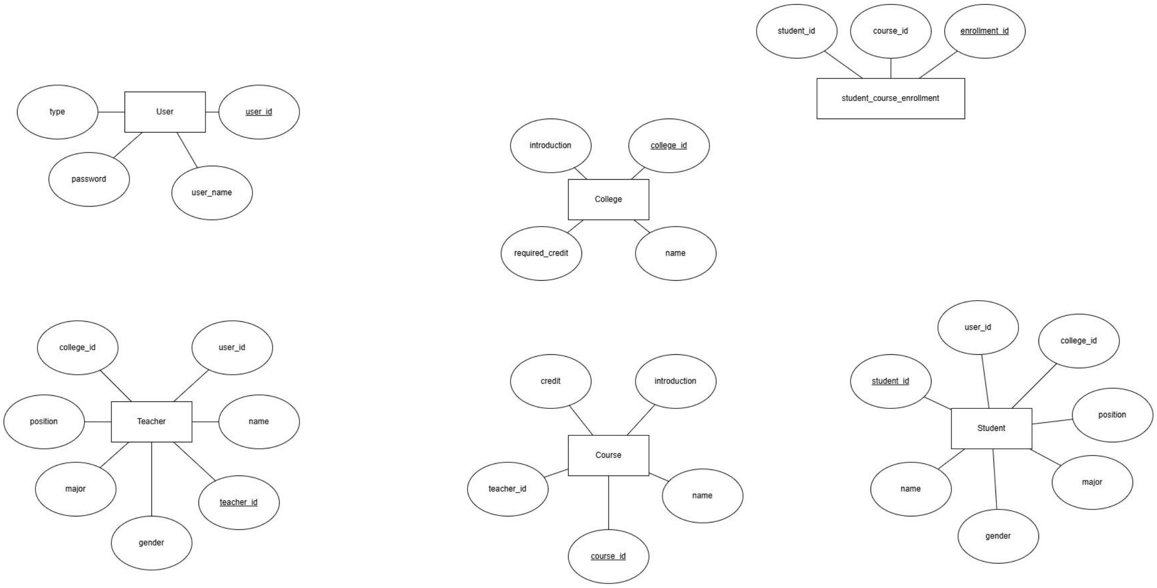


Figure 1 ERD of Database

As a result, detailed description of fields and constraints in user, student, teacher, etc. can be summarized as table 1.

Tbale	Field	Type	Comment	PK	FK
user	user_id	bigint	账号 id	√	√
	user_name	string	账号名		
	password	string	密码		
	type	string	账号类型		
student	student_id	bigint	学号	√	√
	name	string	姓名		
	gender	string	性别		
	major	string	专业		
	credit	double	学分		
	college_id	bigint	学院 id		√
	user_id	bigint	账号 id		√
teacher	teacher_id	bigint	教师工号	√	√
	name	string	姓名		
	gender	string	性别		
	major	string	专业		
	position	string	职称		
	college_id	bigint	学院 id		√
	user_id	bigint	账号 id		√
college	college_id	bigint	学院 id	√	√

	name	string	学院名		
	introdutcion	string	学院介绍		
	required_credit	double	要求学分		
course	course_id	bigint	课程 id	√	√
	teacher_id	bigint	教师工号		√
	name	string	课程名		
	introduction	string	介绍		
	capacity	bigint	课程容量		
	credit	double	学分		
student_course_enrollment	enrollment_id	bigint	选课 id	√	
	student_id	bigint	学号		√
	course_id	bigint	课程 id		√

Table 1 Table Structure of our database

4.4 SQL Inquiry

Our database enables administrators, teachers, and students to perform CRUD (Create, Read, Update, Delete) operations on colleges, majors, courses, enrollments, and user data. As example, we provide scenarios as below.

Login Submission Processing

```
SELECT * FROM user WHERE user_name = %s
```

Get All Colleges Information

```
SELECT college_id, name FROM college
```

Register User Logic

```
SELECT MAX(user_id) AS max_user_id FROM userINSERT INTO user (user_id, user_name, password, type)SELECT
MAX(student_id) AS max_student_id FROM studentINSERT INTO student (student_id, name, gender, major, credit,
college_id, user_id)SELECT MAX(teacher_id) AS max_teacher_id FROM teacherINSERT INTO teacher (teacher_id, name,
gender, major, position, college_id, user_id)
```

Get Teacher Information (for Welcome Message)

```
SELECT t.teacher_id, t.name AS teacher_name, t.gender, t.major, t.position, c.name AS college_name, u.user_name
AS user_nameFROM teacher tJOIN user u ON t.user_id = u.user_idJOIN college c ON t.college_id = c.college_idWHERE
t.user_id = %s
```

Get All Students Information

```
SELECT s.student_id, s.name AS student_name, s.gender, s.major, s.credit, c.name AS college_name,
u.user_nameFROM student sJOIN user u ON s.user_id = u.user_idJOIN college c ON s.college_id = c.college_id
```

Get All Courses Information

```
SELECT c.course_id, c.name AS course_name, c.introduction, c.credit, t.name AS teacher_nameFROM course cJOIN
teacher t ON c.teacher_id = t.teacher_id
```

Get Teacher's Taught Courses

```
SELECT c.course_id, c.name AS course_name, c.introduction, c.credit, c.capacityFROM course cJOIN teacher t ON
c.teacher_id = t.teacher_idWHERE t.user_id = %s
```

Add Course

```
SELECT MAX(course_id) AS max_course_id FROM courseINSERT INTO course (course_id, name, introduction, credit,
capacity, teacher_id)
```

Get All Teachers Information

```
SELECT t.teacher_id, t.user_id, t.name, t.gender, t.major, c.name AS college_name, t.positionFROM teacher tJOIN college c ON t.college_id = c.college_id
```

Delete Student

```
DELETE FROM student WHERE student_id = %s
```

Delete College

```
DELETE FROM college WHERE college_id = %s
```

Delete User

```
SELECT type FROM user WHERE user_id = %sDELETE FROM student WHERE user_id = %sDELETE FROM teacher WHERE user_id = %sDELETE FROM user WHERE user_id = %s
```

Delete Teacher

```
DELETE FROM course WHERE teacher_id = %sDELETE FROM teacher WHERE teacher_id = %s
```

Delete Course

```
DELETE FROM course WHERE course_id = %s
```

Get All Colleges (Admin)

```
SELECT * FROM college
```

Add College (Admin)

```
SELECT MAX(college_id) AS max_college_id FROM collegeINSERT INTO college (college_id, name, introduction, required_credit)
```

Get All Users (Admin)

```
SELECT * FROM user
```

Get All Enrollments (Admin)

```
SELECT sce.student_id, s.user_id, s.name AS student_name, sce.course_id, c.name AS course_nameFROM student_course_enrollment sceJOIN student s ON sce.student_id = s.student_idJOIN course c ON sce.course_id = c.course_id
```

Get All Courses (Admin)

```
SELECT c.course_id, c.name, c.introduction, t.name AS teacher_name, c.capacity, c.creditFROM course cJOIN teacher t ON c.teacher_id = t.teacher_id
```

Get All Students (Admin)

```
SELECT s.student_id, s.user_id, s.name, s.gender, s.major, c.name AS college_name, s.creditFROM student sJOIN college c ON s.college_id = c.college_id
```

5. Hands-On Database

In this section, I will lead you to experience our database from setting up the environment to realizing CRUD through our H5 page.

5.1 Environment Deployment

1. unzip fields to your local machine

2. Install the project dependencies

```bash
```

```
cd projectname

pip install -r requirements.txt
```

3. Configure the database

   - Set the database connection in `config.py` and create database if not exists

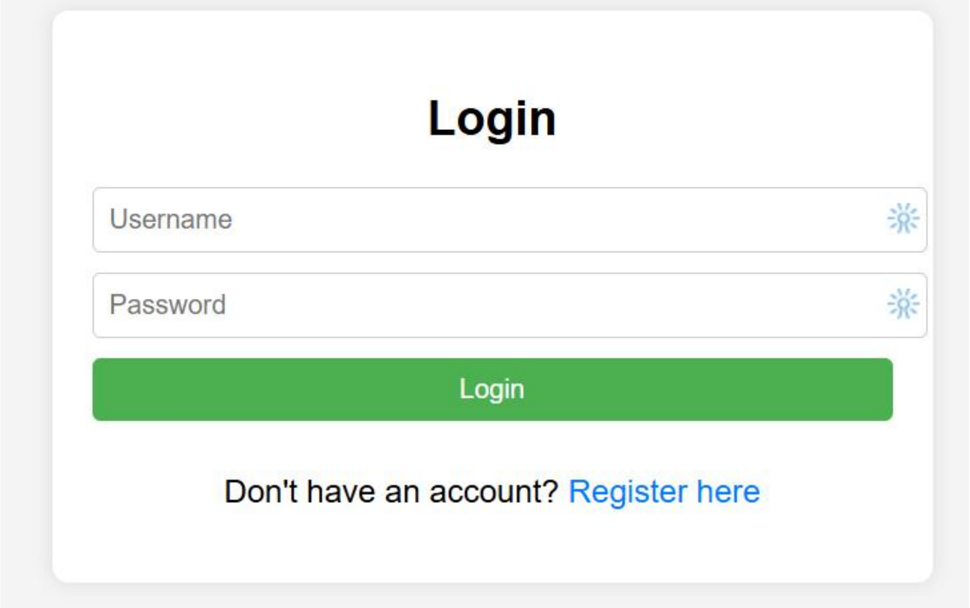
4. Run the project:

```bash
python run.py
```

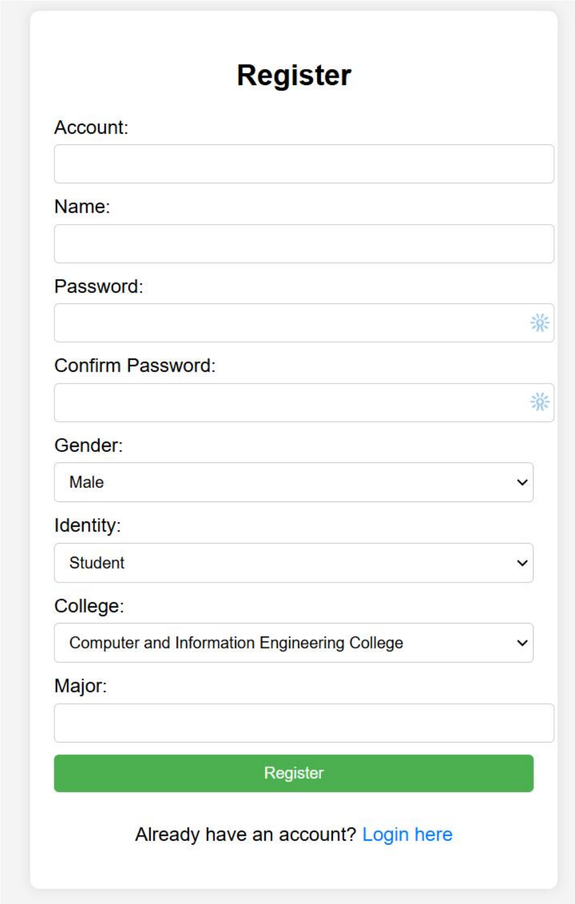
5. Open your browser and visit `http://localhost:5000/` to view the project.

5.2 Webpage Interaction

Visit `http://localhost:5000/`.



Login a reamining account or register a new one. Let’s say I am a student and I haven’t registered yet. Click “Register here” and choose the identity to be “Student”.



Register

Account:

Name:

Password:

Confirm Password:

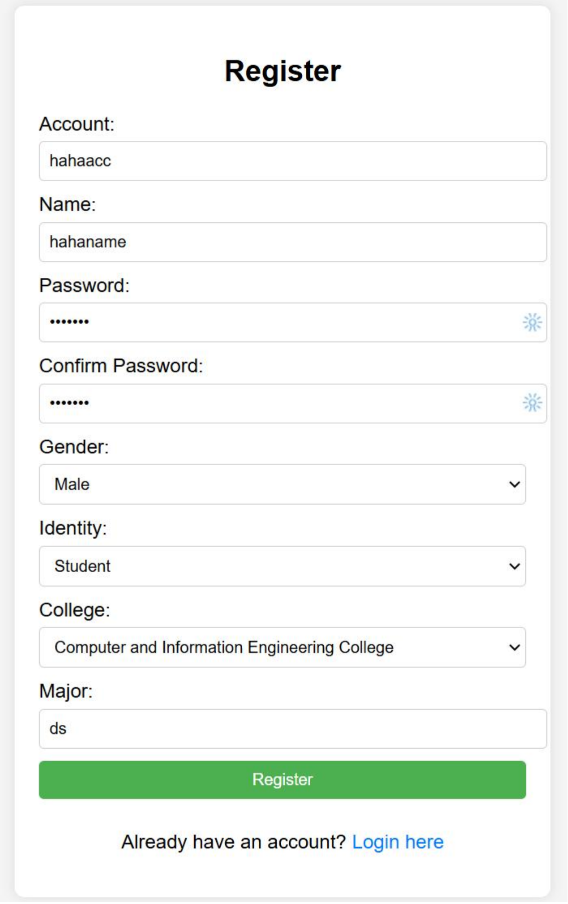
Gender: Male

Identity: Student

College: Computer and Information Engineering College

Major:

Already have an account? Login here



Register

Account: hahaacc

Name: hahaname

Password: •••••••

Confirm Password: •••••••

Gender: Male

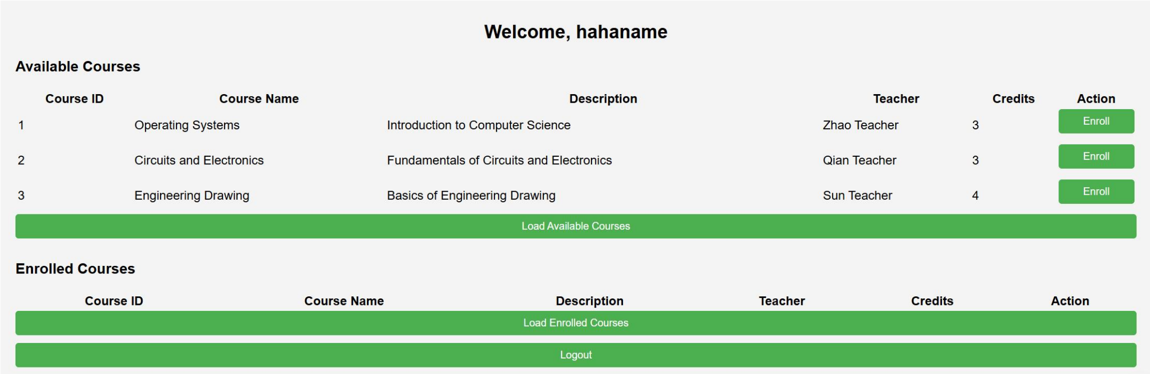
Identity: Student

College: Computer and Information Engineering College

Major: ds

Already have an account? Login here

Type in your information to register, and login at the following page.



Welcome, hahaname

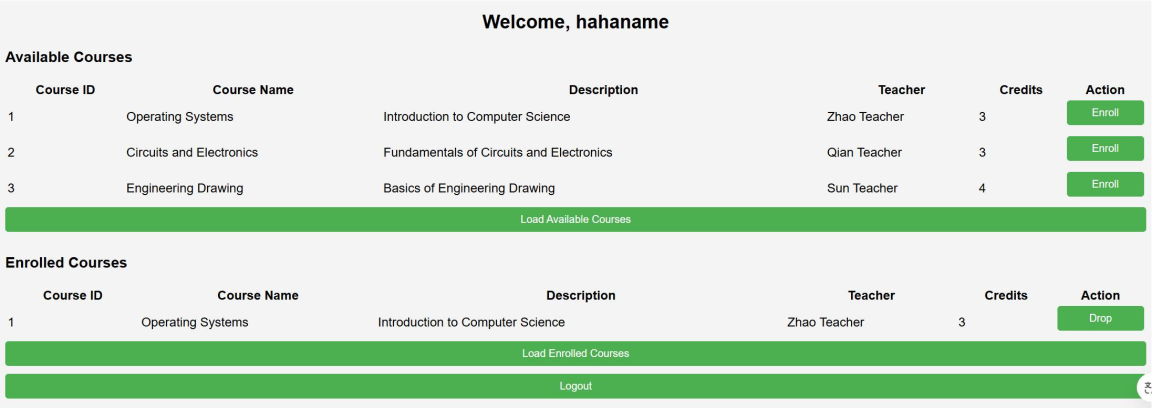
Available Courses

Course ID	Course Name	Description	Teacher	Credits	Action
1	Operating Systems	Introduction to Computer Science	Zhao Teacher	3	Enroll
2	Circuits and Electronics	Fundamentals of Circuits and Electronics	Qian Teacher	3	Enroll
3	Engineering Drawing	Basics of Engineering Drawing	Sun Teacher	4	Enroll

Enrolled Courses

Course ID	Course Name	Description	Teacher	Credits	Action

Click “Enroll” to enroll a class.



Welcome, hahaname

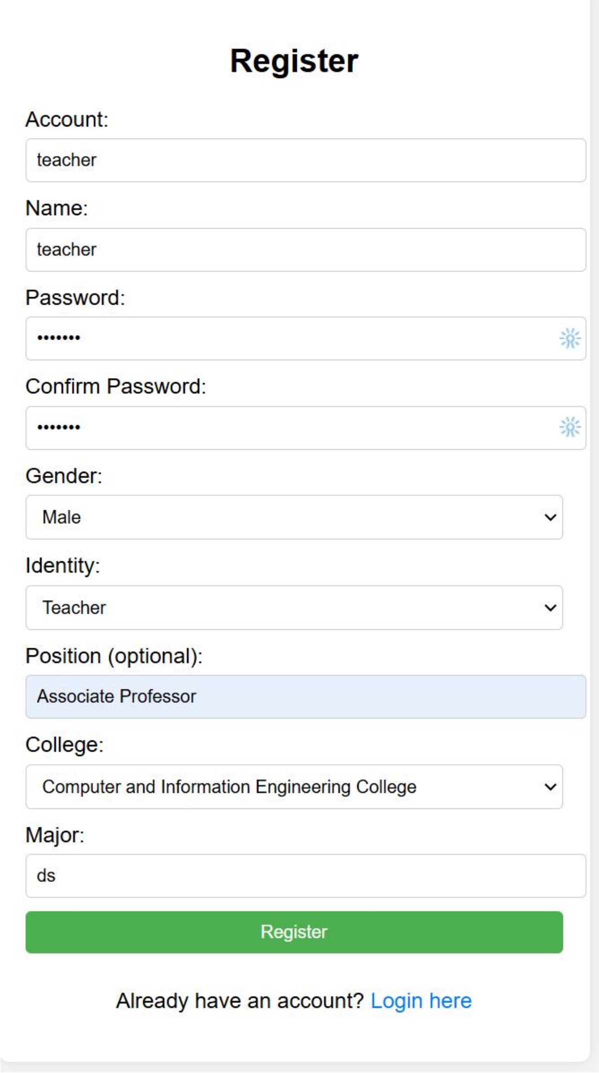
Available Courses

Course ID	Course Name	Description	Teacher	Credits	Action
1	Operating Systems	Introduction to Computer Science	Zhao Teacher	3	Enroll
2	Circuits and Electronics	Fundamentals of Circuits and Electronics	Qian Teacher	3	Enroll
3	Engineering Drawing	Basics of Engineering Drawing	Sun Teacher	4	Enroll

Enrolled Courses

Course ID	Course Name	Description	Teacher	Credits	Action
1	Operating Systems	Introduction to Computer Science	Zhao Teacher	3	Drop

Click “Drop” to drop a class.



Register

Account: teacher

Name: teacher

Password: •••••••

Confirm Password: •••••••

Gender: Male

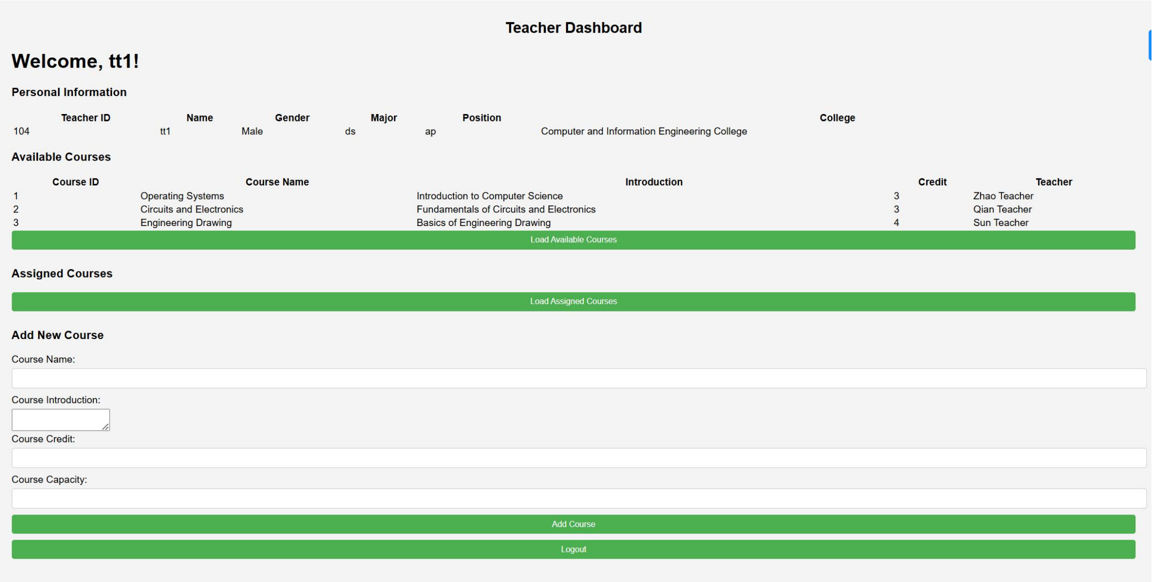
Identity: Teacher

Position (optional): Associate Professor

College: Computer and Information Engineering College

Major: ds

If you are a teacher, choose the identity to be “Teacher” in the register page. And specify your position as well if applicable.



Teacher Dashboard

Welcome, tt1!

Personal Information

Teacher ID	Name	Gender	Major	Position	College
104	tt1	Male	ds	ap	Computer and Information Engineering College

Available Courses

Course ID	Course Name	Introduction	Credit	Teacher
1	Operating Systems	Introduction to Computer Science	3	Zhao Teacher
2	Circuits and Electronics	Fundamentals of Circuits and Electronics	3	Qian Teacher
3	Engineering Drawing	Basics of Engineering Drawing	4	Sun Teacher

Assigned Courses

Add New Course

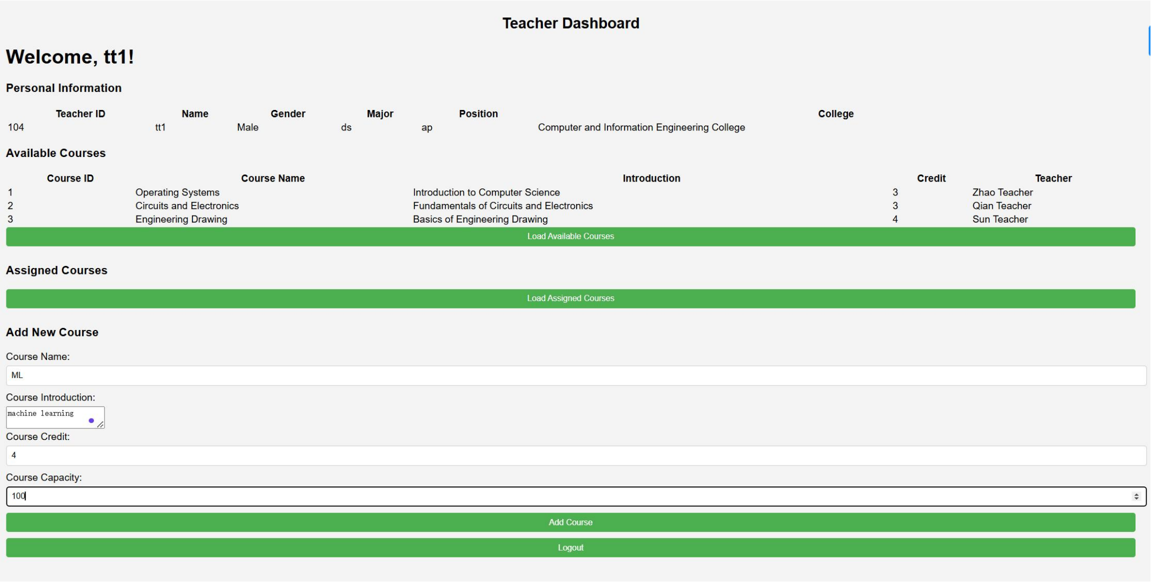
Course Name:

Course Introduction:

Course Credit:

Course Capacity:

The dashboard will show your personal infos and courses you have assigned. You can assign new courses here as well.



Teacher Dashboard

Welcome, tt1!

Personal Information

Teacher ID	Name	Gender	Major	Position	College
104	tt1	Male	ds	ap	Computer and Information Engineering College

Available Courses

Course ID	Course Name	Introduction	Credit	Teacher
1	Operating Systems	Introduction to Computer Science	3	Zhao Teacher
2	Circuits and Electronics	Fundamentals of Circuits and Electronics	3	Qian Teacher
3	Engineering Drawing	Basics of Engineering Drawing	4	Sun Teacher

Assigned Courses

Add New Course

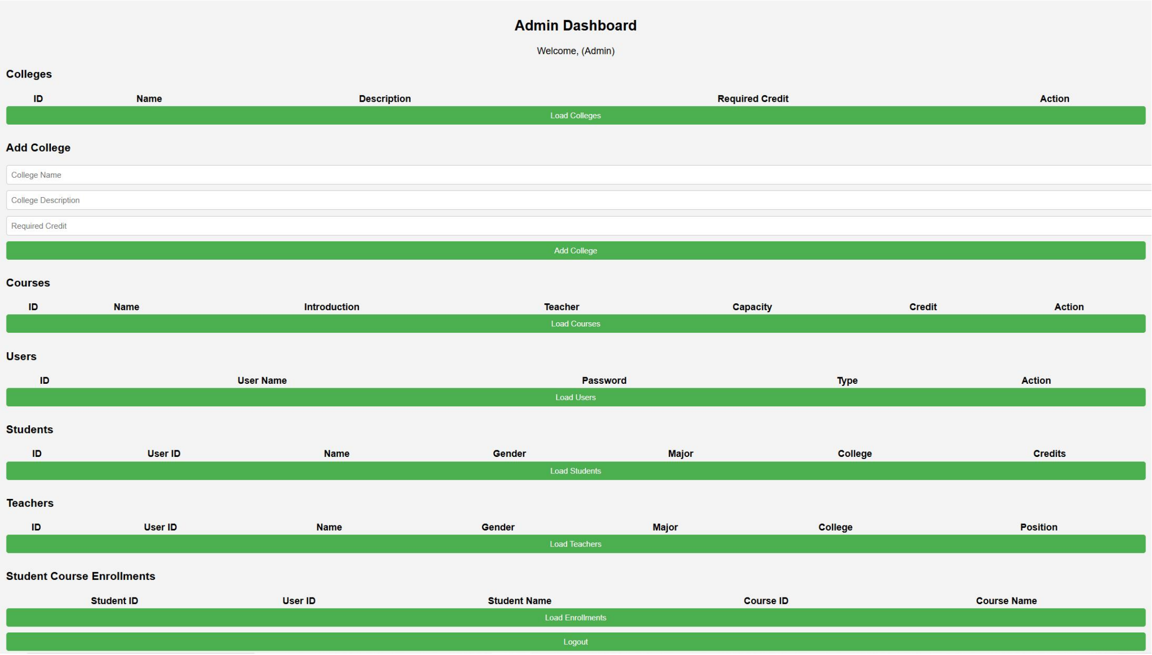
Course Name: ML

Course Introduction: machine learning

Course Credit: 4

Course Capacity: 100

Add a new course by typing in the course infos and clicking “Add Course”.



Admin Dashboard

Welcome, (Admin)

Colleges

ID	Name	Description	Required Credit	Action

Add College

Courses

ID	Name	Introduction	Teacher	Capacity	Credit	Action

Users

ID	User Name	Password	Type	Action

Students

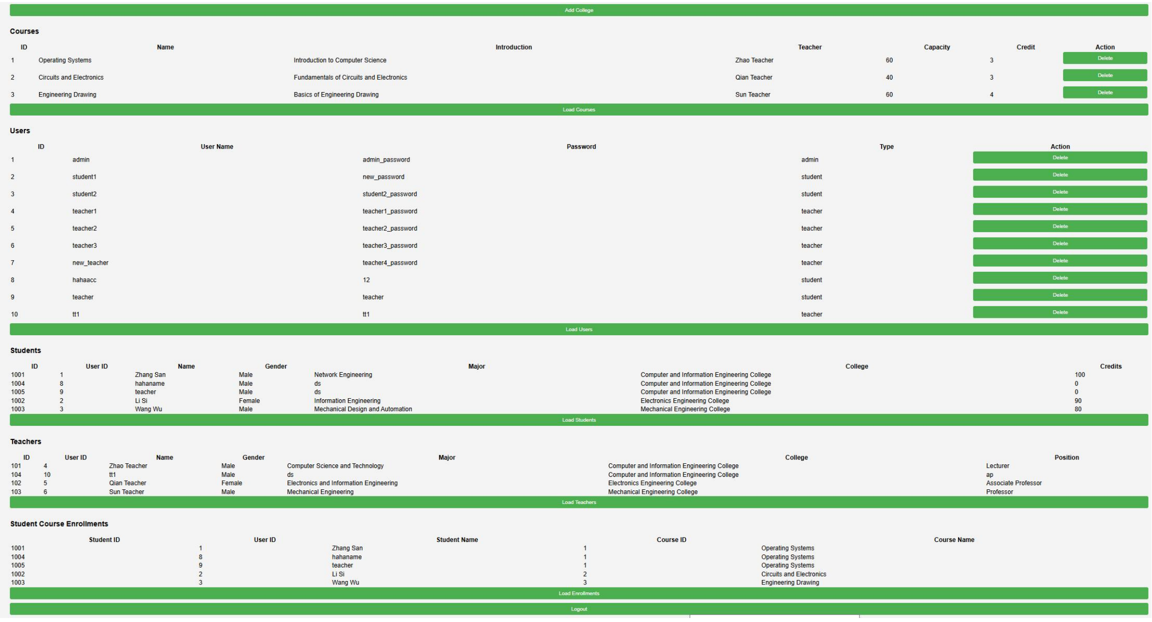
ID	User ID	Name	Gender	Major	College	Credits

Teachers

ID	User ID	Name	Gender	Major	College	Position

Student Course Enrollments

Student ID	User ID	Student Name	Course ID	Course Name



Courses

ID	Name	Introduction	Teacher	Capacity	Credit	Action
1	Operating Systems	Introduction to Computer Science	Zhao Teacher	60	3	Delete
2	Circuits and Electronics	Fundamentals of Circuits and Electronics	Qian Teacher	40	3	Delete
3	Engineering Drawing	Basics of Engineering Drawing	Sun Teacher	60	4	Delete

Users

ID	User Name	Password	Type	Action
1	admin	admin_password	admin	Delete
2	student1	new_password	student	Delete
3	student2	student2_password	student	Delete
4	teacher1	teacher1_password	teacher	Delete
5	teacher2	teacher2_password	teacher	Delete
6	teacher3	teacher3_password	teacher	Delete
7	new_teacher	teacher4_password	teacher	Delete
8	hahaacc	12	student	Delete
9	teacher	teacher	student	Delete
10	tt1	tt1	teacher	Delete

Students

ID	User ID	Name	Gender	Major	College	Credits
1001	1	Zhang San	Male	Network Engineering	Computer and Information Engineering College	100
1004	8	hahaname	Male	ds	Computer and Information Engineering College	0
1005	9	teacher	Male	ds	Computer and Information Engineering College	0
1002	2	Li Si	Female	Information Engineering	Electronics Engineering College	90
1003	3	Wang Wu	Male	Mechanical Design and Automation	Mechanical Engineering College	80

Teachers

ID	User ID	Name	Gender	Major	College	Position
101	4	Zhao Teacher	Male	Computer Science and Technology	Computer and Information Engineering College	Lecturer
104	10	tt1	Male	ds	Computer and Information Engineering College	ap
102	5	Qian Teacher	Female	Electronics and Information Engineering	Electronics Engineering College	Associate Professor
103	6	Sun Teacher	Male	Mechanical Engineering	Mechanical Engineering College	Professor

Student Course Enrollments

Student ID	User ID	Student Name	Course ID	Course Name
1001	1	Zhang San	1	Operating Systems
1004	8	hahaname	1	Operating Systems
1005	9	teacher	1	Operating Systems
1002	2	Li Si	2	Circuits and Electronics
1003	3	Wang Wu	3	Engineering Drawing

What’s more, if you are the administrator, you can login as “admin”, where you can add colleges, or drop colleges, courses and users.

Above, you have experienced the whole function of our database.

6. Links to Source Code, Documents, and Demonstration Video

Source code & Documents: https://github.com/Leo-sqy/5003

Demonstration Video: https://drive.google.com/file/d/18Ev0M7EWfVeJUuRoKXlDxpMIM5y1yve9/view?usp=drive_link