

Phase 2: M03 + M04

2025/7/18:

My goal is combining the student system with the tuition system, thus teachers can track and modify specific student's situations. Therefore, I finished files in folders of model and form. For the form files, I created models about Student and tuition because

```
student.py
models > student.py > ...
1  from extensions import db
2  from datetime import datetime
3
4  class Student(db.Model):
5      __tablename__ = 'student'
6
7      id = db.Column(db.Integer, primary_key=True)
8      first_name = db.Column(db.String(64), nullable=False)
9      last_name = db.Column(db.String(64), nullable=False)
10     age = db.Column(db.Integer)
11     grade = db.Column(db.Integer)
12     email = db.Column(db.String(120), unique=True, nullable=True)
13     phone = db.Column(db.String(25), nullable=True)
14     course = db.Column(db.String(64), nullable=True)
15     teacher_id = db.Column(db.Integer, db.ForeignKey('users.id'))
16     tuition_records = db.relationship('Tuition', backref='student', lazy='dynamic')
17
18     def __repr__(self):
19         return f'{self.first_name}, {self.last_name}'
20
21
22 class Tuition(db.Model):
23     __tablename__ = 'tuition'
24     id = db.Column(db.Integer, primary_key=True)
25     student_id = db.Column(db.Integer, db.ForeignKey('student.id'), nullable=False)
26     amount = db.Column(db.Float, nullable=False)
27     change_type = db.Column(db.String(10), nullable=False)
28     class_time = db.Column(db.DateTime, nullable=False)
29     timestamp = db.Column(db.DateTime, default=datetime.utcnow)
30     note = db.Column(db.String(100))
31
32     def __repr__(self):
33         return f'Tuition: {self.amount} for student {self.student_id}'
```

However, I faced two challenges about how to combine students' id with current user id and how to combine student form with Tuition Form. Thus, I search SQLAlchemy document to find the se two code: `teacher_id = db.Column(db.Integer, db.ForeignKey('user.id'))`
`tuition_records = db. relationship('Tuition', backref='student', lazy='dynamic')`. Then, I figured out these two questions. I established a one-to-many relationship using code.

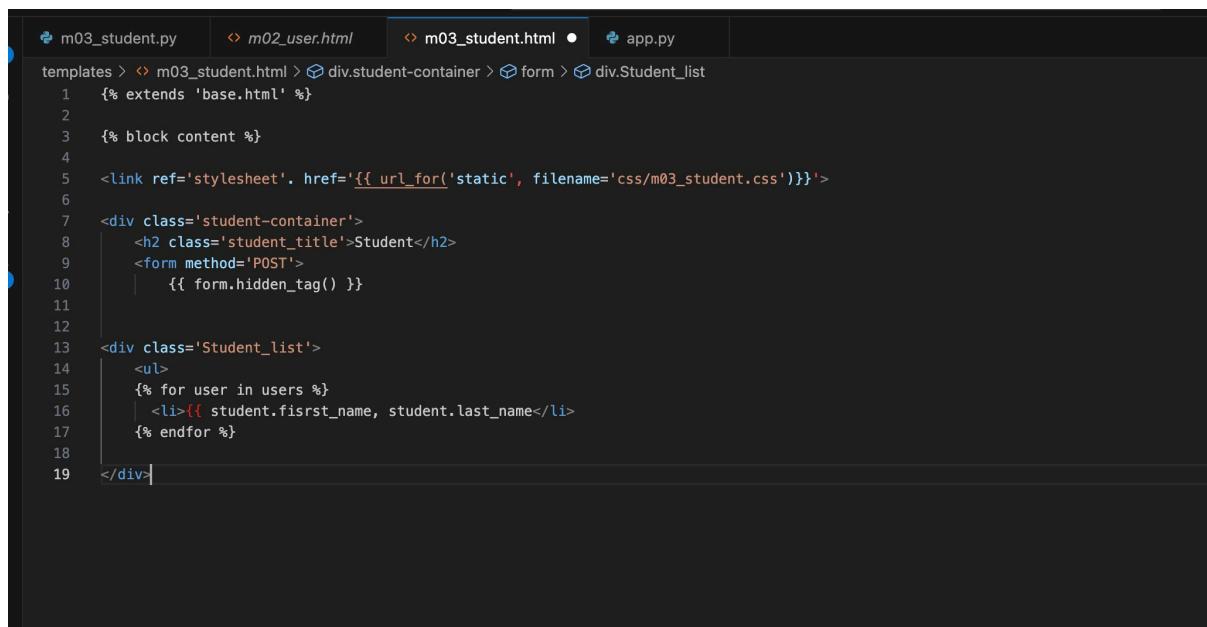
2025/7/23-2025/7/25

Task:

- m03_student.html
- m03_student.css

source:

<https://jinja.palletsprojects.com/en/stable/templates/#list-of-control-structures>



```

m03_student.py      m02_user.html      m03_student.html ● app.py
templates > m03_student.html > div.student-container > form > div.Student_list
1  {% extends 'base.html' %} 
2
3  {% block content %} 
4
5  <link ref='stylesheet'. href='{{ url_for('static', filename='css/m03_student.css')}}'>
6
7  <div class='student-container'>
8    <h2 class='student_title'>Student</h2>
9    <form method='POST'>
10      {{ form.hidden_tag() }} 
11
12
13  <div class='Student_list'>
14    <ul>
15      {% for user in users %}>
16        <li>{{ student.first_name, student.last_name }}</li>
17      {% endfor %}>
18
19 </div>

```

2025/7/26 - 2025/8/3 Tuesday

Bootstrap Modal links:

<https://getbootstrap.com/docs/4.0/components/modal/>

Task:

Part 1 -front - end

m03_student

m03_student.html/css upgrade

m03_student_detail.html

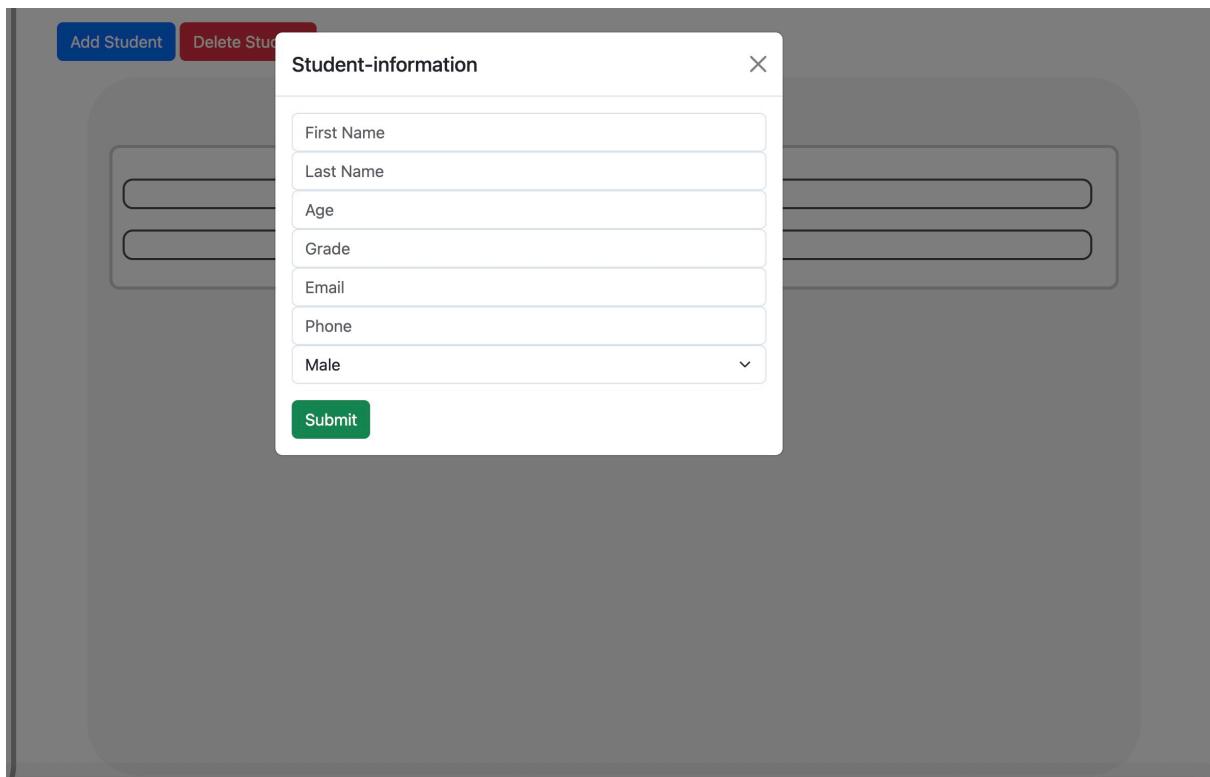
back - end

m03_student.py (form)

m03_student.py(logic)

m03_student.py(model) upgrade

I have added a pop-up feature to my website in the student and tuition module. I used the Bootstrap Modal to set this setting. Meanwhile, I learnt and used JavaScript in the file to enabled smooth animation effects for modal display and transitions using Bootstrap utilities and custom CSS. Therefore, users can add and delete student trough pop-up feature in this website.



In addition, I add student_detail page, thus, user(teacher) can modify student basic information to help them record and communicate with students.

Furthermore, I embedded the tuition module into this. page so that users can manage tuition right after editing a student's basic information. This design improves efficiency for independent users who need to handle large amount of information and data. Users can click tuition button to into the tuition page. In

order to make whole page more beautiful. I used <https://fontawesome.com/search?q=money&o=r> to be icon for my button.

Profile

Student

Calendar

Course

Student Information

First Name
YIXIN

Last Name
ZHOU

Email
yixinzhou728817@gmail.com

Gender
Male

Phone
+1-778-251-0959

Age
19

Grade
12

Course

Tuition

submit

Part 2

M03_student_tuition_html

m03_student_tuition_css

m03_student.py(logic / form / model) upgrade for tuition

I upgraded the data structure in the student model by integrating tuition-related fields and defining two billing modes. This allows users to choose the appropriate mode based on their individual needs. To support a clean and intuitive user interface, I also implemented the tuition calculation logic within the module.

```

logic > m03_student.py > tuition > tuition_amount
 43
 44     @student_bp.route('/student/<int:student_id>/tuition', methods=['GET', 'POST'])
 45     @login_required
 46     def tuition(student_id):
 47         student = Student.query.get_or_404(student_id)
 48         t_form = TuitionForm()
 49         tuition_records = Tuition.query.filter_by(student_id=student.id).all()
 50
 51
 52         tuition_amount = calculate_tuition_amount(
 53             t_form.mode.data,
 54             t_form.lesson_quantity.data,
 55             t_form.duration_hours.data,
 56             t_form.rate_per_lesson.data,
 57             t_form.rate_per_hour.data
 58         )
 59
 60         total_tuition_amount = sum(t.tuition_amount or 0.0 for t in tuition_records)
 61         total_lessons = sum(t.lesson_quantity or 0.0 for t in tuition_records)
 62         total_paid_amount = sum(t.paid_amount or 0.0 for t in tuition_records)
 63         balance = round(total_tuition_amount - total_paid_amount, 2)
 64
 65     return render_template(
 66         'm03_student_tuition.html',
 67         student=student,
 68         tuition_records=tuition_records,
 69         t_form=t_form,
 70         total_lessons=total_lessons,
 71         total_paid_amount=total_paid_amount,
 72         balance=balance,
 73         total_tuition_amount=total_tuition_amount
 74     )
 75
 76

```

```

logic > m03_student.py > tuition > add_tuition_amount
 75
 76
 77     @student_bp.route('/student/<int:student_id>/add_tuition', methods=['POST'])
 78     @login_required
 79     def add_tuition(student_id):
 80         student = Student.query.get_or_404(student_id)
 81         t_form = TuitionForm()
 82
 83         if t_form.validate_on_submit():
 84             mode = t_form.mode.data
 85             tuition_amount = calculate_tuition_amount(
 86                 mode,
 87                 t_form.lesson_quantity.data,
 88                 t_form.duration_hours.data,
 89                 t_form.rate_per_lesson.data,
 90                 t_form.rate_per_hour.data
 91             )
 92
 93             new_tuition = Tuition(
 94                 student_id=student.id,
 95                 lesson_quantity=t_form.lesson_quantity.data if mode == 'lesson' else None,
 96                 duration_hours=t_form.duration_hours.data if mode == 'duration_hours' else None,
 97                 rate_per_lesson=t_form.rate_per_lesson.data,
 98                 rate_per_hour=t_form.rate_per_hour.data,
 99                 paid_amount=t_form.paid_amount.data,
100                 note=t_form.note.data,
101                 mode=mode,
102                 tuition_amount=tuition_amount,
103                 timestamp=datetime.utcnow()
104             )
105
106             try:
107                 db.session.add(new_tuition)
108                 db.session.commit()
109                 flash("Tuition record added successfully.", "success")
110             except Exception as e:
111                 db.session.rollback()
112                 print(f"[ERROR] Failed to add tuition: {e}")
113                 flash("Failed to add tuition record.", "danger")
114             else:
115                 print("[DEBUG] Form validation failed.")
116                 for field, errors in t_form.errors.items():
117                     print(f"[DEBUG] Validation error in '{field}': {errors}")
118                 flash("Invalid form data. Please check required fields.", "danger")
119

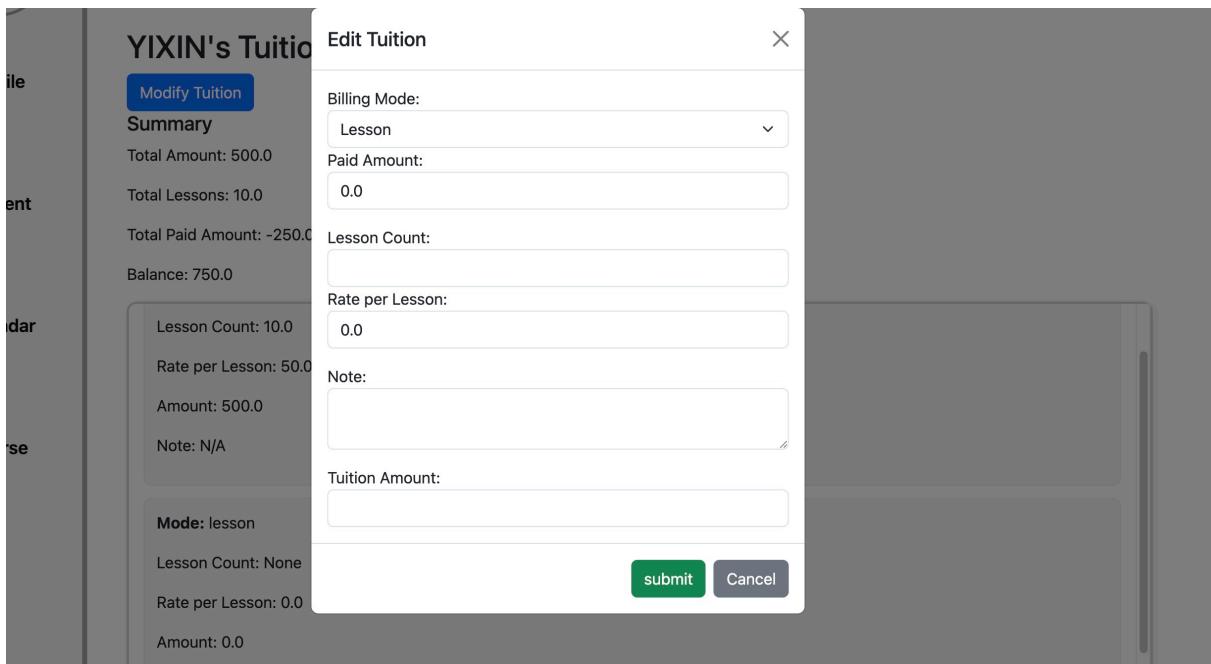
```

```

logic > m03_student.py > tuition > tuition_amount
  79 def add_tuition(student_id):
  80     mode=mode,
  81     tuition_amount=tuition_amount,
  82     timestamp=datetime.utcnow()
  83
  84
  85     try:
  86         db.session.add(new_tuition)
  87         db.session.commit()
  88         flash("Tuition record added successfully.", "success")
  89     except Exception as e:
  90         db.session.rollback()
  91         print(f"[ERROR] Failed to add tuition: {e}")
  92         flash("Failed to add tuition record.", "danger")
  93
  94     else:
  95         print("[DEBUG] Form validation failed.")
  96         for field, errors in t_form.errors.items():
  97             print(f"[DEBUG] Validation error in '{field}': {errors}")
  98         flash("Invalid form data. Please check required fields.", "danger")
  99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122

```

Therefore, users can view the tuition amount and payment status for each student. In the pop-up window of the tuition module, users can input the rate, number of lessons, total tuition amount, and amount paid. The backend logic then processes these inputs, performs the necessary calculations, and renders the results on the main page.



The screenshot shows a mobile application interface with a sidebar on the left containing navigation items: 'Student', 'Calendar', and 'Course'. The main content area displays summary information for each category.

- Student**:
 - Summary
 - Total Amount: 500.0
 - Total Lessons: 10.0
 - Total Paid Amount: -250.0
 - Balance: 750.0
- Calendar**:
 - Lesson Count: 10.0
 - Rate per Lesson: 50.0
 - Amount: 500.0
 - Note: N/A
- Course**:
 - Mode:** lesson
 - Lesson Count: None
 - Rate per Lesson: 0.0
 - Amount: 0.0
 - Note: N/A

2025/8/4 -2025/8/6 Debugging

2025/11/12 Combining with GitHub

```
Last login: Wed Nov 12 15:27:35 on ttys000
yixinzhou@dhcp-172-17-45-210 ~ % cd Desktop/M01
yixinzhou@dhcp-172-17-45-210 M01 % pwd
/Users/yixinzhou/Desktop/M01
yixinzhou@dhcp-172-17-45-210 M01 % ls
Readme.txt           forms           static
__pycache__          logic           templates
app.db               migrations       users.json
app.py               models          venv
config.py            requirements.txt
extensions.py        runtime.txt
yixinzhou@dhcp-172-17-45-210 M01 % git status
fatal: not a git repository (or any of the parent directories): .git
yixinzhou@dhcp-172-17-45-210 M01 % git config --global user,email "yixinzhou728817@gmail.com"
error: key does not contain a section: user,email
yixinzhou@dhcp-172-17-45-210 M01 %
```

A terminal window is shown with the following command and error message:

```
tch)
origin https://github.com/ZacahryZhou/YiXin-Zhou-Zachary-Zhou--Coding-Project-Education-Web.git (push)
[yixinzhou@dhcp-172-17-45-210 M01 % git push -u origin main
remote: Invalid username or token. Password authentication is not supported for Git operations.
fatal: Authentication failed for 'https://github.com/ZacahryZhou/YiXin-Zhou-Zachary-Zhou--Coding-Project-Education-Web.git'
yixinzhou@dhcp-172-17-45-210 M01 %
```

```
new file: forms/auth.py
new file: forms/m02_user.py
new file: forms/m03_student.py
new file: forms/m04_course.py
new file: logic/__init__.py
new file: logic/auth.py
new file: logic/m02_user.py
new file: logic/m03_student.py
new file: logic/m04_course.py
new file: migrations/README
new file: migrations/alembic.ini
new file: migrations/env.py
new file: migrations/script.py.mako
new file: migrations/versions/d9256aa6c0d4_initial_migration_with_updated_all_
new file: models/__init__.py
new file: models/calendar.py
new file: models/course.py
new file: models/feedback.py
new file: models/student.py
new file: models/user.py
new file: requirements.txt
new file: runtime.txt
new file: static/.DS_Store
new file: static/css/auth.css
new file: static/css/base.css
new file: static/css/index.css
new file: static/css/m02_user.css
new file: static/css/m03_student.css
new file: static/css/m03_student_detail.css
new file: static/css/m03_student_tuition.css
new file: static/css/m04_course_section.css
new file: static/css/m04_courses.css
new file: static/images/.DS_Store
new file: static/images/1.png
new file: static/images/2.png
new file: static/images/avatar-female.png
new file: static/images/avatar-male.png
new file: static/images/avatar.png
new file: static/images/homepage.png
new file: static/images/sidebar-logo.png
new file: static/logo.png
new file: templates/.DS_Store
new file: templates/base.html
new file: templates/dashboard.html
new file: templates/index.html
new file: templates/login.html
new file: templates/m02_user.html
new file: templates/m03_student.html
new file: templates/m03_student_detail.html
new file: templates/m03_student_tuition.html
new file: templates/m04_course.html
new file: templates/m04_course_section.html
new file: templates/register.html
new file: users.json
```

```
ject-Education-Web.git/'  
yixinzhou@dhcp-172-17-45-210 M01 % git push -u origin main  
Username for 'https://github.com': ZacharyZhou  
Password for 'https://ZacharyZhou@github.com':  
Enumerating objects: 73, done.  
Counting objects: 100% (73/73), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (68/68), done.  
Writing objects: 100% (73/73), 9.54 MiB | 1007.00 KiB/s, done.  
Total 73 (delta 9), reused 0 (delta 0), pack-reused 0 (from 0)  
To https://github.com/ZacahryZhou/YiXin-Zhou-Zachary-Zhou--Coding-Project-Educat:  
 * [new branch]      main -> main  
Branch 'main' set up to track 'origin/main'.  
yixinzhou@dhcp-172-17-45-210 M01 %
```

After August - Spent a lot of time prepare for IELTS

After September - Continue Updating