# CourseFlow: Course Registration Site

Bigger Data

# Introduction

Members: Desiree Caceres, Amanda Farghli, Suhana Lama, Tak Kit Yeung

What is CourseFlow?

CourseFlow is a course registration site, which provides a seamless and efficient solution for university students navigating the complex - and often frustrating - course registration process. With high demand and limited seats, students often face challenges securing their preferred classes. CourseFlow simplifies this process by offering availability updates through scheduling conflict detection and an easy-to-navigate scheduling interface, ensuring students can enroll in the classes they need without the usual stress.

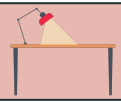
# The Stack

| | |
|---|---|
| Front End  | Repository  |
| Server  | Deployment  |
| Database  | |

# Profit Model

CourseFlow uses yearly sliding scale subscription cost. Where educational institutions such as colleges and universities are charged an initial flat fee, and additional fees are determined by the number of students enrolled, and classes they display through our services.

- $9,000 - initial fee
- $0.25 - per student enrolled
- $4.25 - per course fee

$9,000 + 0.25 per  + $4.25 per 

A sample college with 5,000 students enrolled with 2000 classes would be charged a yearly fee of $18,725

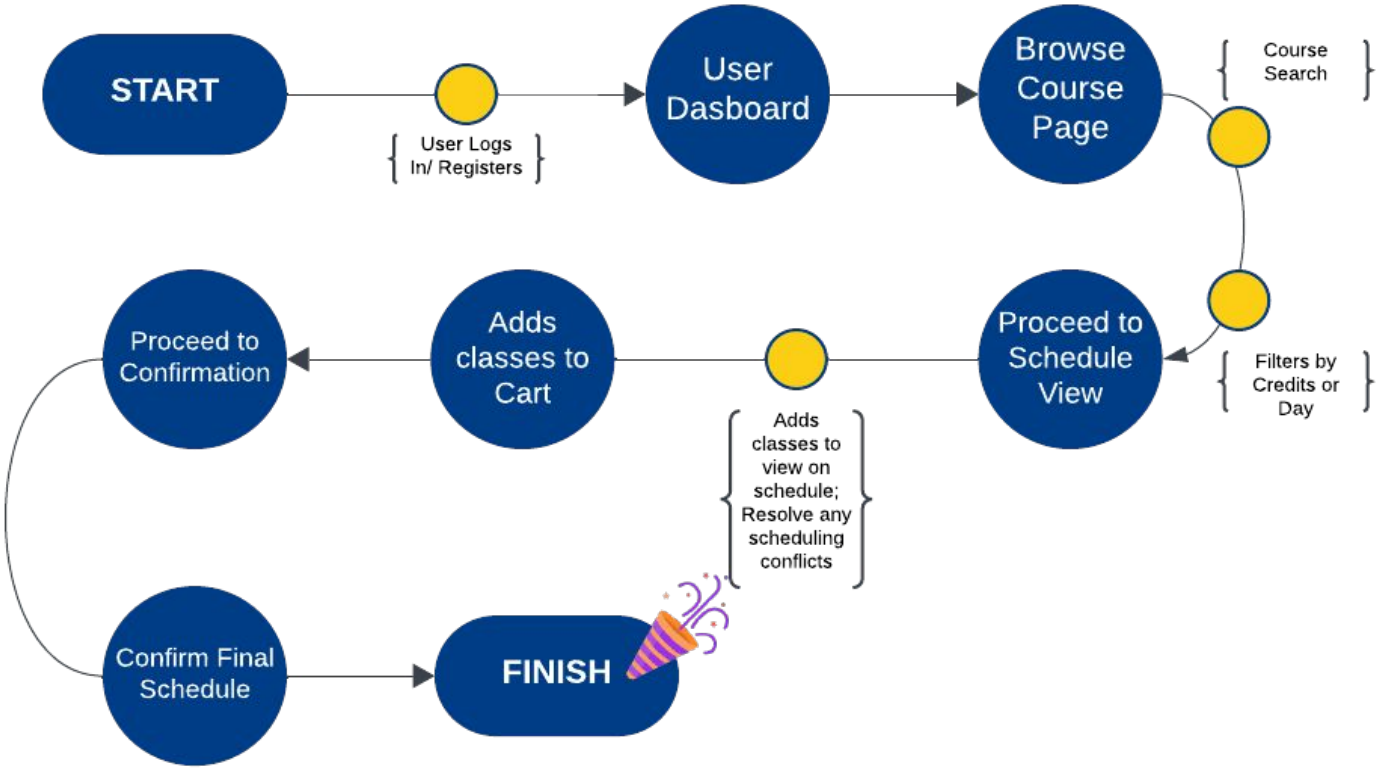
# Profit Model Represented Through Table & SQL Statement

| Column Name | Data Type | Description |
|---|---|---|
| University_id | int | Primary key, identifies each institution |
| University_name | varchar(255) | Name of the institution |
| student_count | Int | Number of enrolled students |
| num_classes | Int | Number of classes displayed through CourseFlow |
| base_fee | Decimal (10,2) | Fixed yearly fee (the $9,000 initial fee) |
| student_fee | Decimal (10,2) | Fee calculated as $1 per student enrolled |
| class_fee | Decimal (10,2) | Fee calculated as $35 per class displayed |
| total_fee | Decimal (10,2) | Total yearly fee (sum of base_fee, student_fee, and class_fee) |

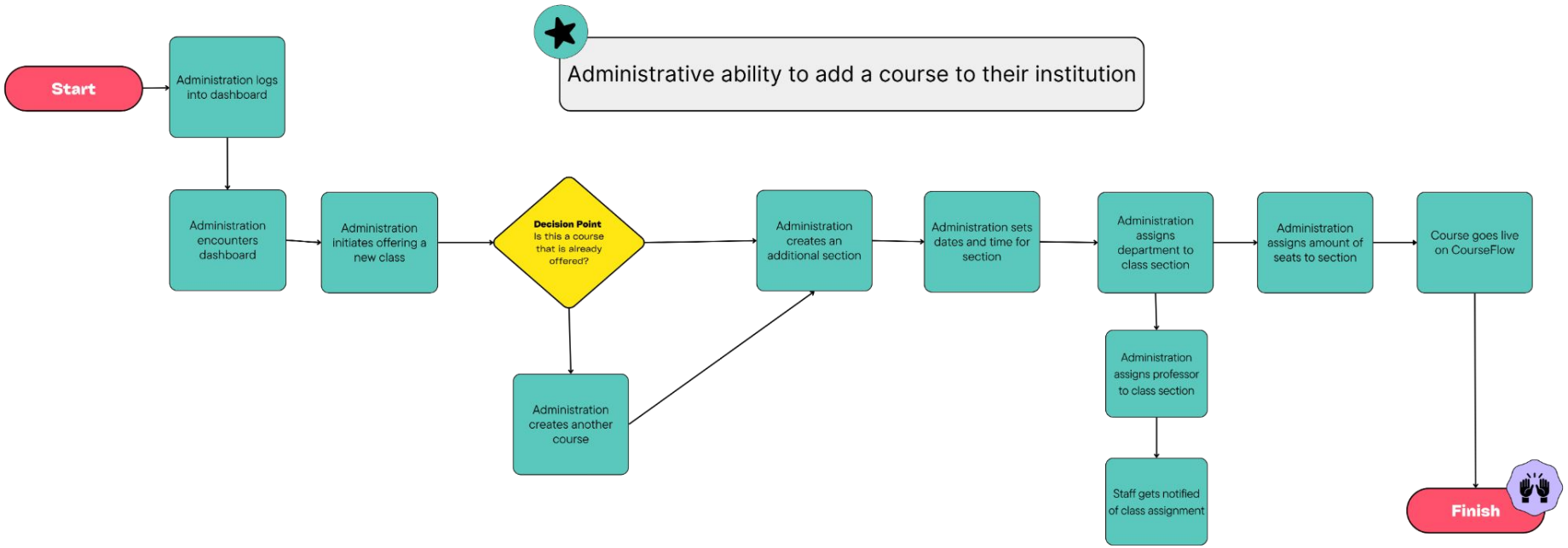
```
CREATE TABLE operating_costs (
    institution_id INT PRIMARY KEY,
    institution_name VARCHAR(255) NOT NULL,
    student_count INT NOT NULL,
    num_classes INT NOT NULL,
    base_fee DECIMAL(10, 2) DEFAULT 9000.00 NOT
NULL,

    student_fee DECIMAL(10, 2) GENERATED ALWAYS
AS (student_count * 1.00) STORED,

    class_fee DECIMAL(10, 2) GENERATED ALWAYS
AS (num_classes * 35.00) STORED,

    total_fee DECIMAL(10, 2) GENERATED ALWAYS
AS (base_fee + student_fee + class_fee) STORED

);
```

# User Cases Sequence Charts

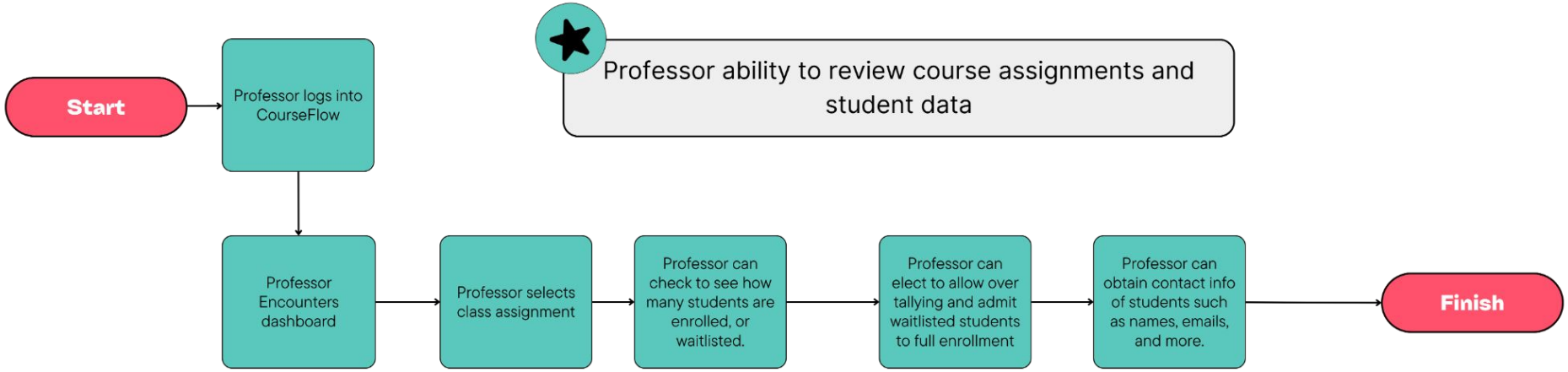
# CourseFlow User Journey



# Administrative Functionality Sequence Diagram



# Professor Functionality Sequence Diagram



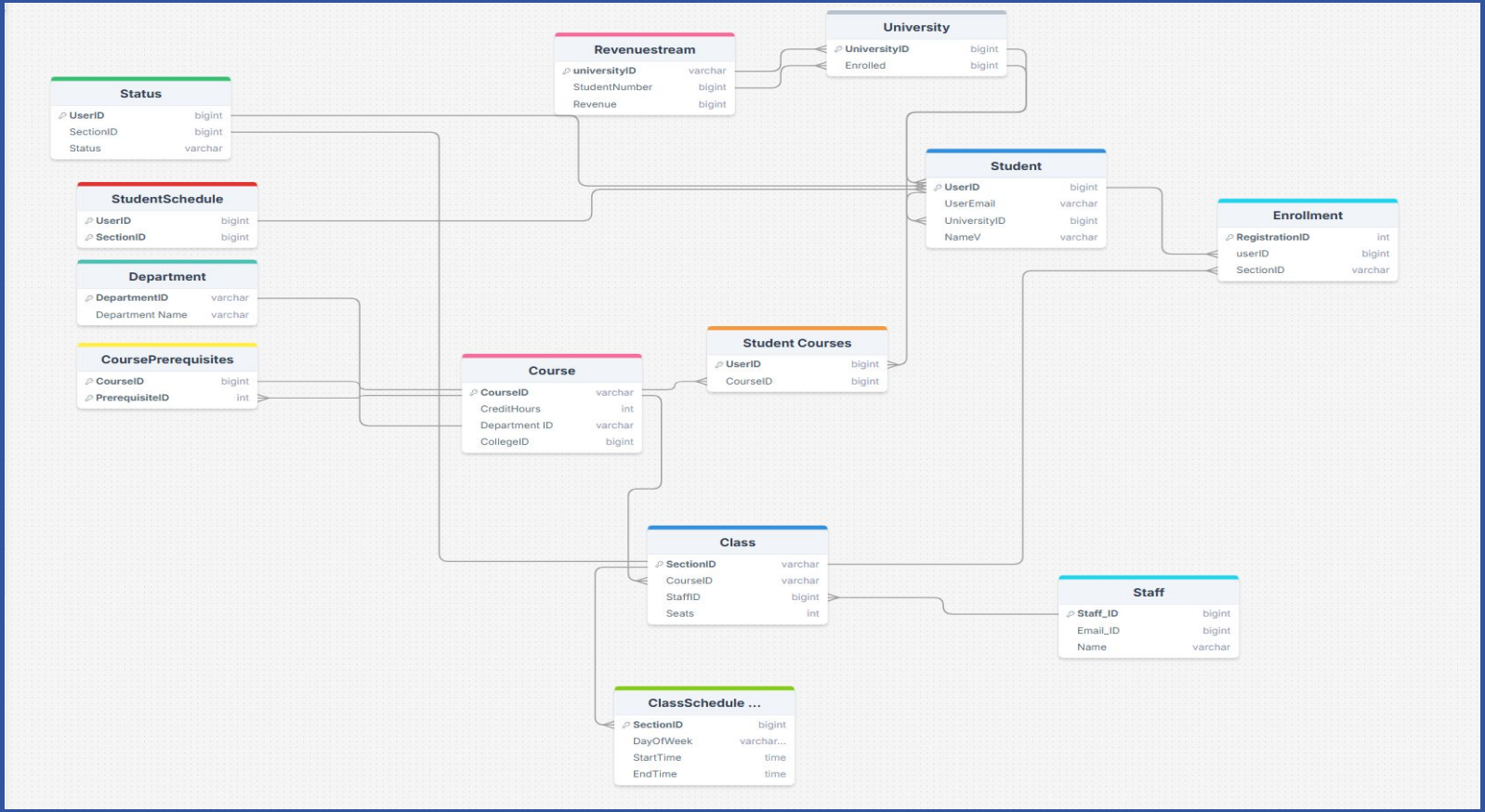
# CourseFlow Demo

Landing Page + User Action

# Stored Procedures

*Demo*

```
DROP PROCEDURE IF EXISTS RegisterUser;
CREATE PROCEDURE RegisterUser(
    IN firstNameIn VARCHAR(50),
    IN lastNameIn VARCHAR(50),
    IN emailIn VARCHAR(255),
    IN passwordIn VARCHAR(255))
BEGIN
    INSERT INTO account (firstName, lastName, email, password)
    VALUES (firstNameIn, lastNameIn, emailIn, passwordIn);
END;

DROP PROCEDURE IF EXISTS CheckEmail;
CREATE PROCEDURE CheckEmail(IN emailIn VARCHAR(255))
BEGIN
    SELECT * FROM account WHERE email = emailIn;
END;
```




# 2NF Relational Model Diagram

# Job Delegation

Desiree Caceres : Front-end development

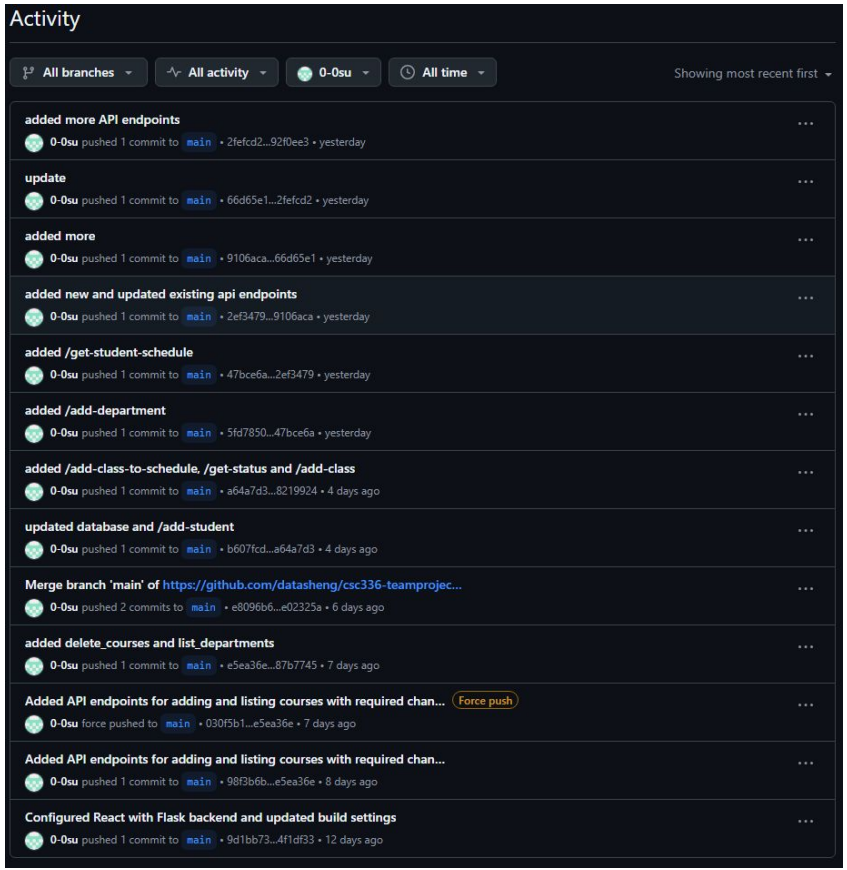
Amanda Farghli : Database setup and implementation, normalization of database,

Suhana Lama : Back-end development, database connecting, coordinator, troubleshooting

Tak Kit Yeung : Front-end & Back-end Development, functionality testing, troubleshooting

For communication, we spoke often through Discord, sharing what work we've started/gotten done. We would also meet on call to further discuss our progress and any issues we were facing. All updates to the code were added to the main branch of our repository. If we had our own separate branches, the code would be added there, tested, and then added to the main branch.

# Project Activity Suhana Lama




## Explanation of Tasks

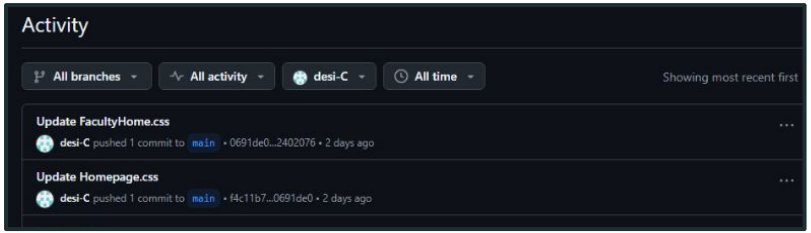
1. Setting up and creating various endpoints to allow the front-end team to pull data to display to the user.

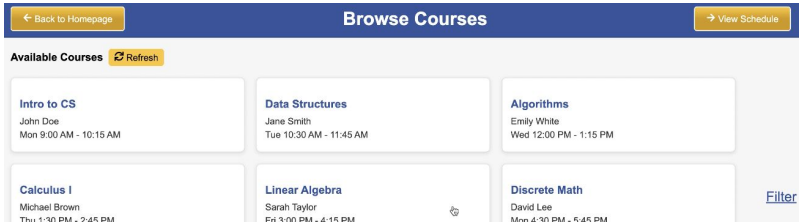
2. Acted as the the team's coordinator where they made sure to know what front-end needed and would communicate those needs to the back-end team.

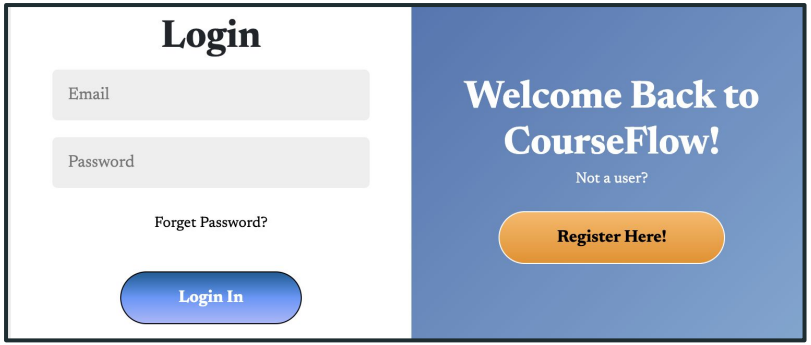
3. Assigned goals to the team to and would enforce deadlines for project to finish in a timely manner.

4. Initiated all the team meetings and deadlines.

# Project Activity Desiree Caceres



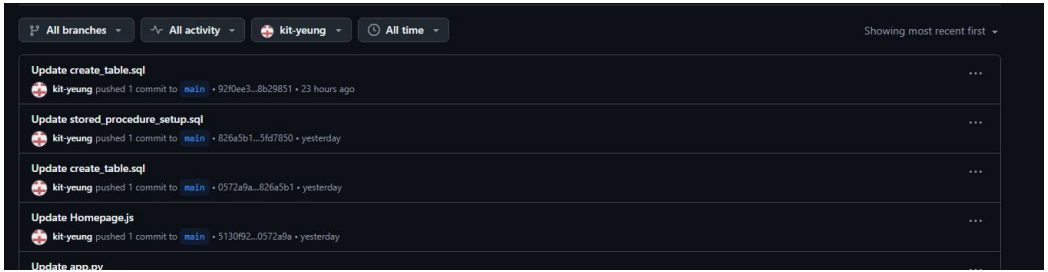




## Explanation of Tasks

1. Frontend development lead.
2. Setup majority of frontend and overall polishing the end-user experience.
3. Established the ability to browse & filter classes.
4. Established login page & registration page.
5. Initiated the ability to register for classes and add to schedule with the front end.

# Project Activity Tak Kit Yeung

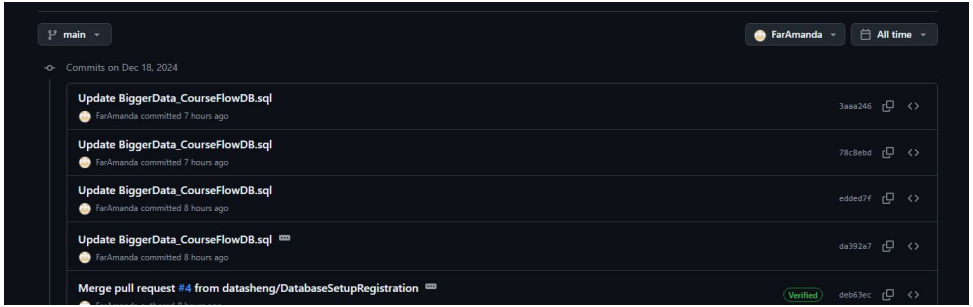


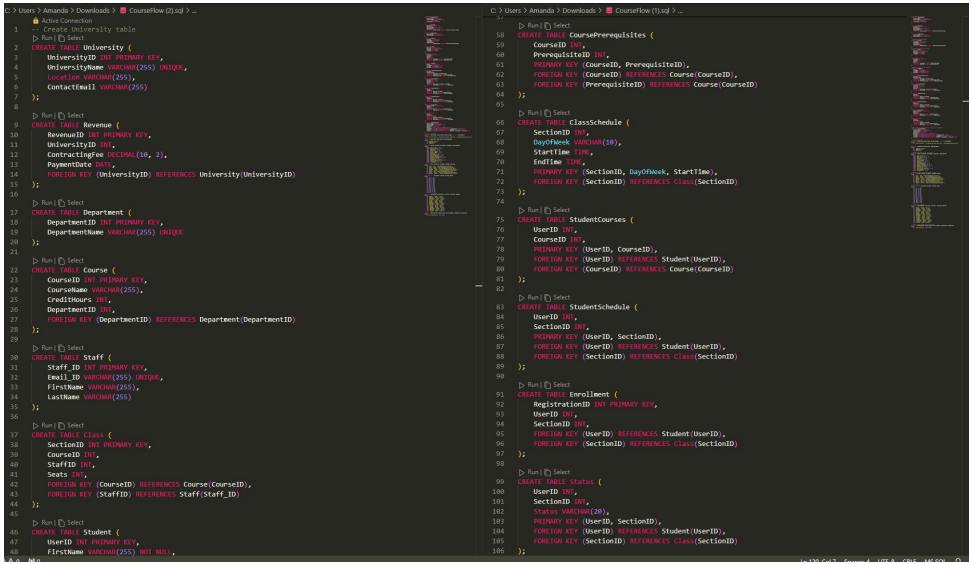


## Explanation of Tasks

1. Would test project thoroughly to ensure everyone had the same user experience, no matter the device
2. Troubleshooted every error that members encountered and found a fix.
3. Provided thorough documentation for the project to ensure team's understanding of current progress at every stage of the development process.
4. Setup endpoints for database
5. Setup profile landing page.

# Project Activity Amanda Farghli



## Explanation of Tasks

1. Set up a database that would handle the backend of the project.
2. Ensured database satisfied 2nf requirement
3. Supplied database with “dummy” data that can be used for testing.
4. Amended & tweaked database for ease of datapulling
5. Establishing pricing model and formatted revenue table for automatic generation of operating costs for an educational institution

# Q&A