Four-Year Course Planning Assistant for Pepperdine University

Frankie Mundo, Eric Njuku
Mentor: Dr. Scalzo
Pepperdine University, 24255 Pacific Coast Highway, Malibu, CA 90263

Introduction

Pepperdine students need an efficient tool to help them organize a course plan to ensure that they graduate within four years.

Our Objective: Develop a web-based application to help students quickly and easily generate a four-year course plan based on their major or minor.

Methods

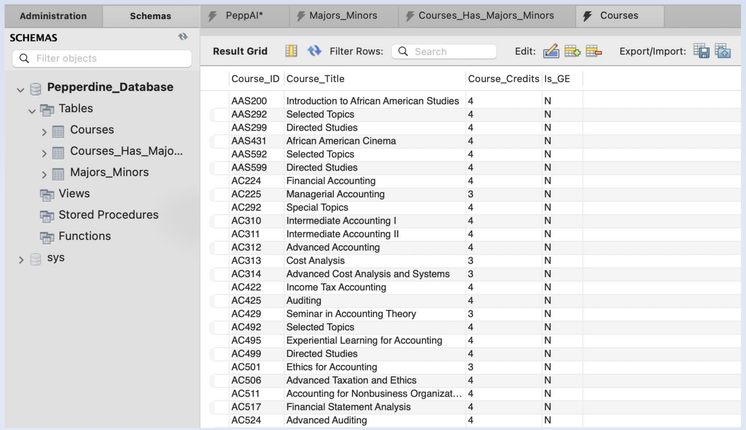
Database Design: Designed relational database using MySQL Workbench:

- Two tables: Courses (Course ID, course title, number of units, and if GE requirement) and Majors_Minors (program name and if major/minor)
- Created a new table called Courses_Has_Majors_Minors with an n:m relationship, where the two elements are primary keys from other tables
- Utilized scripts to fill tables with information from Pepperdine course catalog.

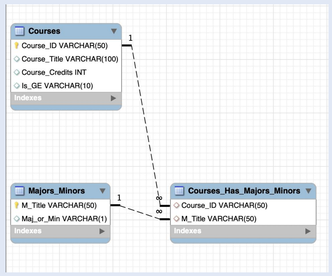
Backend:

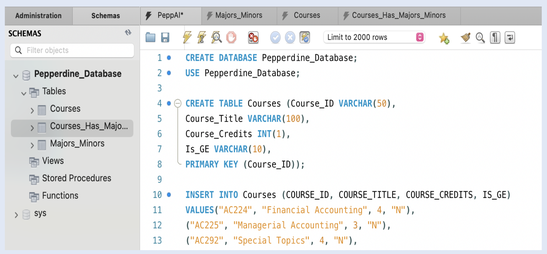
- Used the Node.js runtime environment to start server and run on local port.
- Wrote the code for the backend in a file called PeppAI.js.
    - connects to our MySQL database and uses Express.js framework to handle HTTP requests, outputting plan based on four-year 128 credit plan
- Created a script.js file to handle the fetch request to the backend endpoint:
    - request loops through the course plan data to display each course, evenly divided among four years

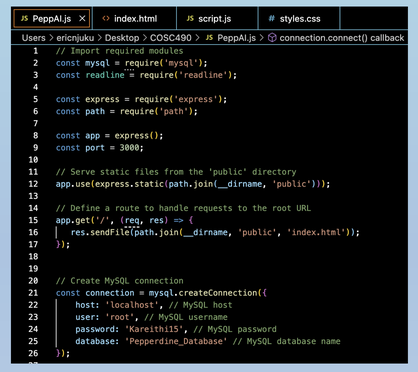
Frontend: Created a file called index.html to manage raw text on the local webpage and file called styles.css to handle the visual styling of the webpage.

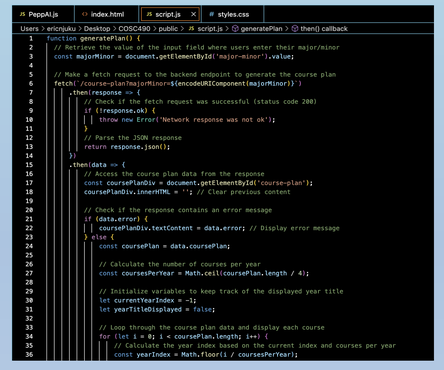


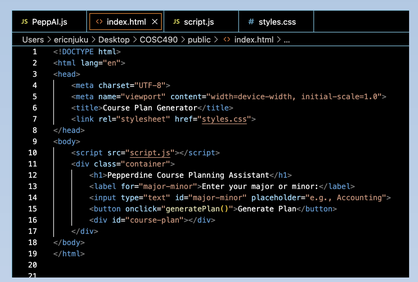
Course_ID	Course_Title	Course_Credits	Is_GE
AAS200	Introduction to African American Studies	4	N
AAS292	Selected Topics	4	N
AAS299	Directed Studies	4	N
AAS431	African American Cinema	4	N
AAS592	Selected Topics	4	N
AAS599	Directed Studies	4	N
AC224	Financial Accounting	4	N
AC225	Managerial Accounting	3	N
AC292	Special Topics	4	N
AC310	Intermediate Accounting I	4	N
AC311	Intermediate Accounting II	4	N
AC312	Advanced Accounting	4	N
AC313	Cost Analysis	3	N
AC314	Advanced Cost Analysis and Systems	3	N
AC422	Income Tax Accounting	4	N
AC425	Auditing	4	N
AC429	Seminar in Accounting Theory	3	N
AC492	Selected Topics	4	N
AC495	Experiential Learning for Accounting	4	N
AC499	Directed Studies	4	N
AC501	Ethics for Accounting	3	N
AC506	Advanced Taxation and Ethics	4	N
AC511	Accounting for Nonbusiness Organizat…	4	N
AC517	Financial Statement Analysis	4	N
AC524	Advanced Auditing	4	N











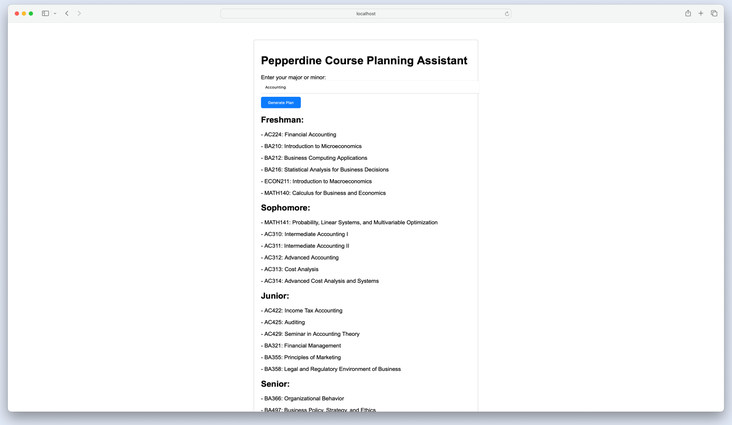
Results

Our result is a local web application with an input field for users to enter their major or minor.

- User inputs their program of choice and clicks the “Generate Plan” button and a list of required classes for their program is displayed.
- The courses are organized by year

The database is fully implemented, with all information present, and the code is fully functioning with no errors, but we were unable to add handling for GE courses on the front end and prerequisites in general.

```
Server running at http://localhost:3000
Connected to MySQL database
Enter your major or minor: Computer Science/Mathematics
Course plan for Computer Science/Mathematics:
Freshman:
- COSC101: Programming Principles I with Python
- COSC105: Programming Principles I with R
- COSC121: Programming Principles II
- MATH150: Calculus I
- MATH151: Calculus II
- MATH220: Formal Methods
- MATH221: Discrete Structures
```



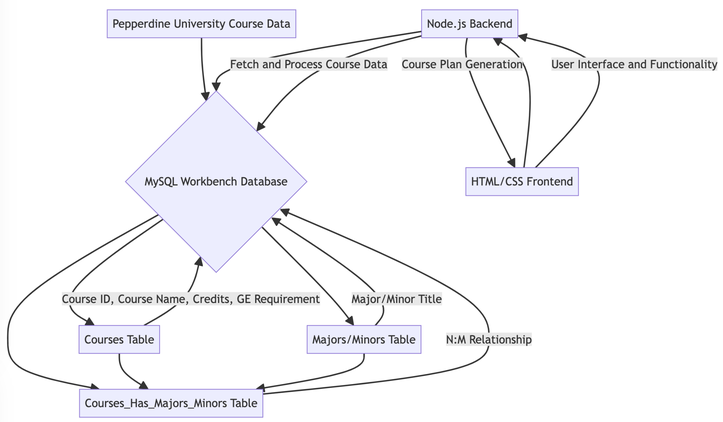
Discussion

Our tool effectively displays the courses necessary for each major or minor at Pepperdine and the code itself displays the list of the classes in addition to potential GE classes that the student can take.

Future improvements:

- Account for "or" classes, where there are multiple options for a class to take
- Add the GE classes to the front-end user interface
- Implement a table in the database to account for prerequisite courses and add the code to handle this
- Improve the user interface to make it more visually appealing

Pepperdine Course Planning Assistant Architecture



The diagram above illustrates the flow of data and the interactions between the different components of the application, showing the integration of a MySQL Workbench database, a Node.js backend, and an HTML/CSS frontend.

Conclusion

We have effectively constructed a Pepperdine course planning assistant on the full stack:

- Constructed a database using MySQL and utilized scripts to populate the tables within
- Created a Node.js server to run a local webpage
- Managed information with JavaScript
- Developed the front end of the user interface with HTML and CSS.

This webpage will allow students to generate a four-year course plan based on their major/minor, with plans to implement more frontend/backend improvements to the application in the future.