**Project Description**

* Assist advisors with class selection for students

**What was added**

* Added Validation
  + List Validation
    - Add Classes Validation
      * The original code allowed any class to be added to the student’s list of already taken classes. However, classes have prerequisites that are required in order to take the higher level classes. Therefore, classes are now checked and whether their prerequisite has been taken.
    - Remove Classes Validation
      * The original code allowed for any class to be removed without consequences. However, when a student removes a fundamental class, it is assumed that any upper level class after has not been taken, and must also be removed. Therefore, classes that are removed warn the student that upper level classes will be removed and remove the upper level classes.
  + Form Validation
    - The original code would accept invalid input for the student’s name, email, and student id and even store them in their database. This is a massive security flaw that is especially vulnerable to SQL injection. With our changes the program will not allow the user to submit the form without a valid name, student id, and email.
  + Output Validation
    - After submitting classes taken and valid user information the program will output classes the student may take next semester. Previously the program would output every class the student hasn’t taken yet, but with our changes only classes that the student has met the requirements for will be displayed. For example, if the student has only taken CMSC201 the program will output CMSC202 and not CMSC341.

**What was improved Upon**

* Revamped look of website
  + Removed sidebar (Moved navigation to header)
  + Changed the user input form styles (Made better use of space)
  + Changed the font styles
  + Removed redundant space from header
  + Added footer at the bottom of page
  + Handled text overflow for the class selection boxes (The original program will not display the text which exceed the text field)
  + Made the links in the navbar functional (the original sidebar is not functional because it links to the nothing or nonexistent file)
  + Added browser tab icon
* Database Setup
  + The given code contained references to the “satisfied” and “prerequisites” columns in the database; however, they were absent from the documentation and the provided database dump. The database was then modified to comply with the code. Additionally, the code contained multiple queries that were selecting the same thing, resulting in a worse performance than necessary. The code was reduced to only necessary database queries which resulted in a more readable code base along with a performance increase.

**Database Set Up**

The database used within the project contained two tables, classes and students. Classes contained information for every class supported within the program. It stored the course number, course name, a binary value for whether it was required or not, a binary value for whether or not the current student has taken it, the number of credits, a binary value for whether or not it was an elective class, a binary value for whether or not the current student has met the requirements for it, its department, and finally its prerequisite classes. The students table just contains the name, campus id, and email for each student successfully processed by the program.

|  |  |
| --- | --- |
| CREATE TABLE IF NOT EXISTS `classes` (  `courseNum` varchar(8) NOT NULL,  `name` text NOT NULL,  `required` tinyint(1) DEFAULT NULL,  `taken` tinyint(1) NOT NULL,  `credits` int(11) NOT NULL,  `elective` tinyint(1) NOT NULL,  `satisfied` tinyint(1) NOT NULL,  `department` varchar(4) NOT NULL,  `prerequisites` text  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  INSERT INTO `classes` (`courseNum`, `name`, `required`, `taken`, `credits`, `elective`, `satisfied`, `department`, `prerequisites`) VALUES  ('CMSC203', 'Discrete Structures', 1, 1, 3, 0, 1, 'CMSC', 'CMSC201 CMSC202'); | CREATE TABLE IF NOT EXISTS `students` (  `name` text NOT NULL,  `student\_id` text NOT NULL,  `email` text NOT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  INSERT INTO `students` (`name`, `student\_id`, `email`) VALUES  ('Ian Kirk', 'IB95854', 'ikirk1@umbc.edu'); |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Example Classes Entry** | | | | |
| **courseNum** | **name** | **required** | **taken** | **credits** |
| CMSC203 | Discrete Structures | 1 | 1 | 3 |
| **elective** | **satisfied** | **department** | **prerequisites** |  |
| 0 | 1 | CMSC | CMSC201 CMSC202 |  |

|  |  |  |
| --- | --- | --- |
| **Example Students Entry** | | |
| **name** | **student\_id** | **email** |
| Ian Kirk | IB95854 | ikirk1@umbc.edu |

**Languages used and where**

* HTML/CSS
  + Index.html
    - Index.html is the landing page that contains a header, nav bar, and an iframe that displays classSelect.html
  + ClassSelect.html
    - Contains the class lists for the user to select which classes they’ve taken and the forms for the user to input their identifying information.
  + Contact.html
    - Accessible via the nav bar, contains contact information for members of the team
  + ClassDescription.html
    - Accessible via the nav bar, contains a picture of the classes available and the requirements between them
* Javascript (ClassSelect.html)
  + All in ClassSelect.html, functions
  + Handles class selection lists and validation
  + Form input validation
    - Onsubmit function on the form to check user input
* PHP/SQL (databaseTest.php)
  + Handles database interactions
  + Output validation

**Slick sheet**

|  |  |
| --- | --- |
| The landing page of the website, here the user can select which classes they’ve taken or they can access other information such as the class description or the contact page. | Step1.png |
| After selecting their classes the user can then scroll down to see which ones they’ve selected. Here they will also input their identifying information. | Step2.png |
| Upon clicking submit a new page is loaded displaying the courses available to take for the student. The user can return to the home page using the nav bar for another session. | Step3.png |
| This page displays the relationships between the computer science classes available. | ClassDesc.png |
| This page displays the contact information for the team. | Contact.png |
| A preview of the database table “classes” in phpmyadmin | Database.png |