CNMS Academic Advising Website

1. Introduction of Team
   1. Christopher Dinh, [cdinh2@umbc.edu](mailto:cdinh2@umbc.edu)
      1. Startup Script Creator
      2. Additional Advising Documents
   2. Sarah Heiner, [she3@umbc.edu](mailto:she3@umbc.edu)
      1. Documentation and Sitemap
      2. Password Capability
      3. Appointment Scheduling
   3. Jake Horvers, [horvers1@umbc.edu](mailto:horvers1@umbc.edu)
      1. Customer Relations
      2. SQL Queries
      3. Error Message Display
   4. Kristin McLaughlin, [mckris2@umbc.edu](mailto:mckris2@umbc.edu)
      1. CSS and Website Appearance
      2. Student and Advisor View
   5. Robert Rose, [robrose2@umbc.edu](mailto:robrose2@umbc.edu)
      1. End of Season Switch
      2. SQL Queries
2. Location of Project
   1. Git Hub Repository: <https://github.com/RobRoseKnows/cs331-project-2>
   2. Site Link: <https://swe.umbc.edu/~mckris2/CMSC331/proj2/Project2.9.1/cs331-project-2/html/forms/first_page>
3. Project Description

This advising site will finally eliminate the pen-and-paper system currently used by the CNMS department. Students are now able to login to their accounts and sign up for appointments, along with canceling and changing them as needed, view helpful information regarding their major, and fill out pre-advising information to help the process along. Advisors are given the option to create and remove appointments, view their advising schedule, see a list of students who have not yet registered for advising, and remove access to the site at the end of the season.

1. Modifications to Given Code
   1. Elimination of Error Forms

The original project utilized error forms to validate user information, adding in unnecessary pages and making the website more complicated for the user. Our group eliminated these forms. In this site, the user is re-directed back to the page that the error occurs on so that they may correct it. An error message is displayed so that the user understands what they did wrong.

* 1. Added password capability

This project used only a username when the user logged in. As requested by the client, we have implemented the use of passwords so that both students and advisors can log into their respective accounts securely.

* 1. Improved formatting to make website both appealing and intuitive

When given this project, our group discovered that it wasn’t very user friendly, even with the documentation provided. We sought to make the site more intuitive, and added links on all pages to help the user return to the screen they really wanted if they accidentally misclicked.

The advising site has also received a design overhaul, modeled after myUMBC as requested by the client. The tab icon is the same paw print logo used for myUMBC, the background of each page includes a UMBC 50 watermark, and all font and background colors are those in the UMBC color scheme. The re-designed website helps the user to quickly find what they are looking for and organizes the advising information in a way that allows optimized access to existing resources.

A button display has been added at the top of both the advisor and student home screen, making it much easier for each party to locate the function of the website that they need. Students can search for, schedule, and cancel an appointment all from their homepage, while advisors can search for, add/edit appointments, and even print their schedule for the day (formatted with limited CSS to save ink).

* 1. Search appointments option

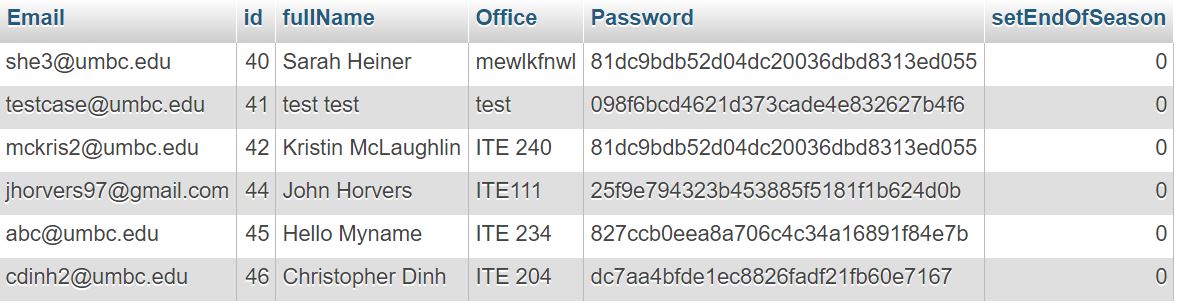
To improve the advisor’s use of the website, we have added in search functionality for all advising appointments. Advisors will now be able to filter their searches by appointment type (group/individual), date, and advisor leading the session.

* 1. Editing appointments

Advisors now have the capability to edit pre-existing appointments. If the number of students allowed in one appointment has to change for some reason, simply update the appointment to reflect the change and students will be notified. Have to change the location of the meeting last minute? Update on the advisor side and students will be able to see that their appointment has changed.

1. Database Setup
   1. The Advisor Database: advisors

This database stores vital information provided by the advisor (email, full name, office), along with a password so that the advisor may log in and see their homepage as needed. The setEndOfSeason field in the table is used when the advisors declare the advising season to be closed and wish to disable the website. This will occur if all advisors choose to close the website for the season.

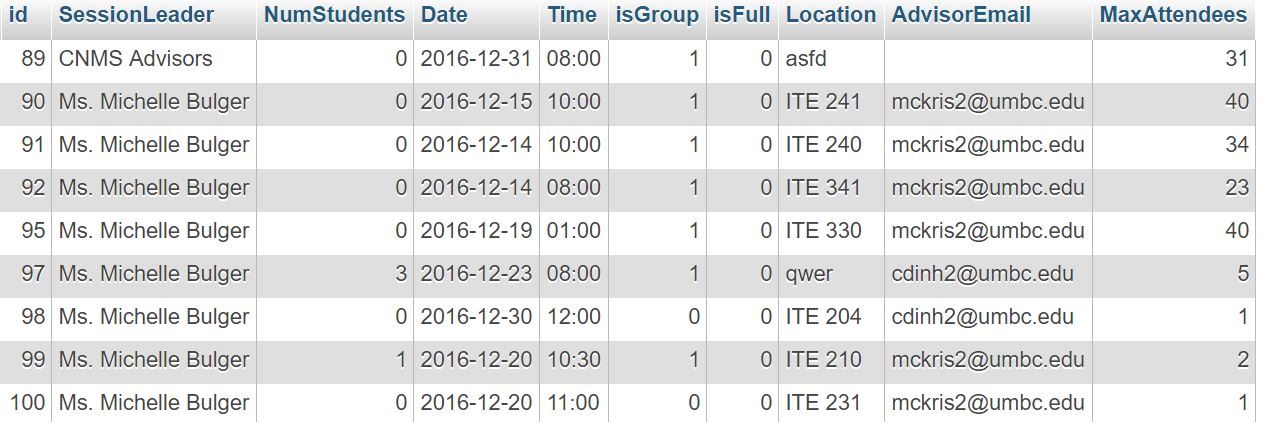


**Table structure for table advisors**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| Email | text | No |  |
| ***id*** | int(11) | No |  |
| fullName | text | No |  |
| Office | text | No |  |
| Password | text | No |  |
| setEndOfSeason | tinyint(1) | Yes | 0 |

* 1. The Appointment Database: appointments

This table keeps track of the many pieces of information that are important to creating a successful appointment. Basic facts, like the advisor who created the session, date, time, location, and session leader are stored. Additionally, there are trackers for the number of students currently signed up for the appointment, along with the maximum number that can attend. There are two flags, isGroup and isFull which help sort out the appointments into those that are for groups and those for individual, and the appointments that are still available for students to register for.

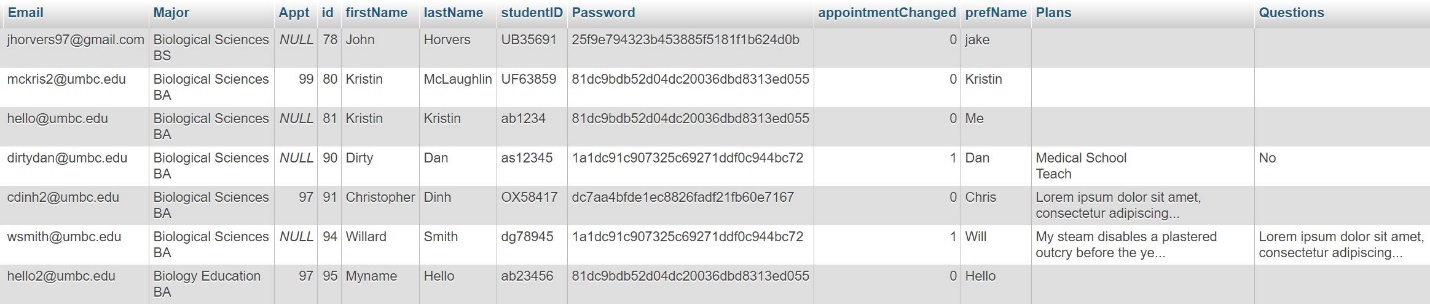


**Table structure for table appointments**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***id*** | int(11) | No |  |
| SessionLeader | text | No |  |
| NumStudents | tinyint(4) | No | 0 |
| Date | varchar(10) | No |  |
| Time | varchar(5) | No |  |
| isGroup | tinyint(1) | No | 0 |
| isFull | tinyint(1) | No | 0 |
| Location | text | No |  |
| AdvisorEmail | text | No |  |
| MaxAttendees | tinyint(40) | No |  |

* 1. The Student Database: students
     1. This database stores vital information provided by the student (email, first/last name, preferred name, student ID and major), along with a password so that they may log in and schedule advising appointments as necessary. There are also fields that store if the user has scheduled an appointment, and if they have changed their appointment

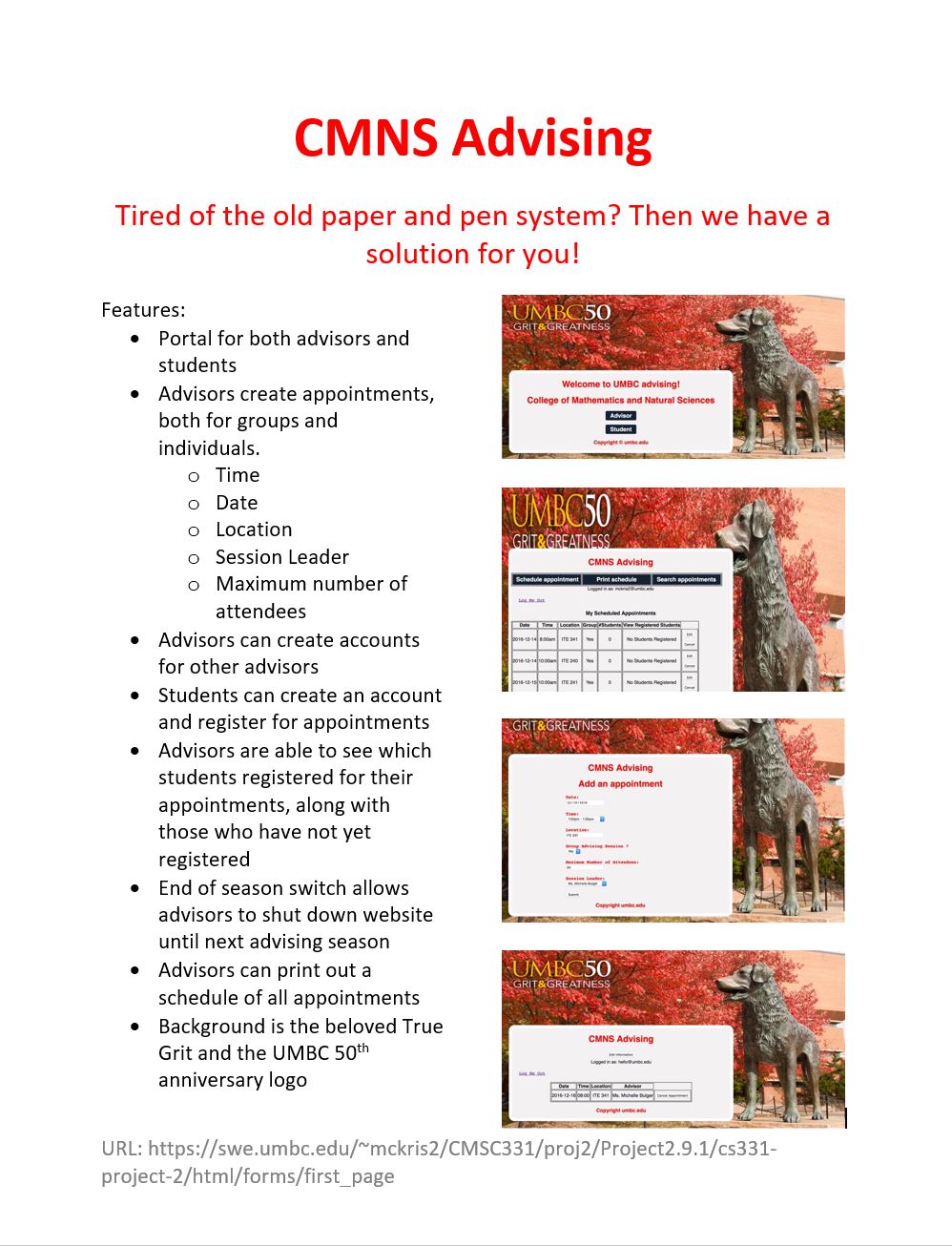
Table structure for table students



**Table structure for table students**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| Email | text | No |  |
| Major | text | No |  |
| Appt | int(11) | Yes | NULL |
| ***id*** | int(11) | No |  |
| firstName | text | No |  |
| lastName | text | No |  |
| studentID | varchar(9) | No |  |
| Password | text | No |  |
| appointmentChanged | tinyint(1) | No | 0 |
| prefName | text | No |  |
| Plans | text | No |  |
| Questions | text | No |  |

1. Languages Used
   1. PHP
      1. Validation of login and registration pages
      2. Advisor and student home pages
      3. Table handling
      4. End of season switch
      5. Appointment search, registration, and cancelation.
   2. MySQL
      1. Table handling
      2. Validation of login and registration pages
2. Slick Sheet

****