1.Team Introduction:

* Logan Dudley
  + [logdud1@umbc.edu](mailto:logdud1@umbc.edu)
  + PHP, Authentication, Documentation
* James Gordon
  + [gordon7@umbc.edu](mailto:gordon7@umbc.edu)
  + Error Handling, Input Validation, Documentation
* Phillip Hilliard
  + [phi3@umbc.edu](mailto:phi3@umbc.edu)
  + Database Management, MySQL
* Sara Nabulsi
  + [snabuls@umbc.edu](mailto:snabuls@umbc.edu)
  + Feature Implementation, Alert Handling, Human Relations
* Connor Shaffer
  + [pb10459@umbc.edu](mailto:pb10459@umbc.edu)
  + CSS, Design, Documentation, Feature Implementation

2. Location of Project

The entire project is contained within a folder called Advising-Site-Master, When the user enters this folder in order to gain access to the websites they then need to enter the src folder. The student page can be immediately accessed by entering the student folder. In order to enter the advisor side, the user will have to access the advisor folder and then enter the view folder.

3. Project Description

This website was intended as a means to streamline the advising signup process for students and advisors alike in the College of Natural and Mathematical Sciences (CNMS). The website features a simple appointment creation and editing process, as well as a calendar for easy viewing and managing of appointments on the advisor side. Just as well, features have been provided in the case that unexpected circumstances arise for the advisors, such as editing room assignments, appointment cancellation, and a master password for advisors to be able to edit their colleagues appointments. On the student side, Students can view available appointments from a list of available days and can register for either an individual or group advising session. Students can also cancel and reschedule appointments as they see fit.

4. What was added if given old code?

The entirety of the CSS styling, including color scheme

CSS + Design, Season Over, Calendar, Advisor Password, Pre-advising Wksht, Alerts,

Editing appointments, advisor passwords

Once a student is logged in they have been given the option to fill out the pre-advising form. The form saves their answers in between sessions and is auto-populated once they reenter the page.

Added the ability to open and close the advising season. This makes it so that once the season is closed students are no longer able to log in and make appointments.

Alert features were added to both student and advisor side. Whenever a student logins into their page they are alerted to the fact if an advisor has deleted the appointment they were signed up for. When advisors login, they are alerted to the fact if the master password has been used to login into their account.

5. What was improved upon if given old code?

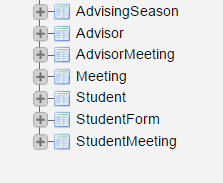
Authentication when users log in was improved. Before, if a user attempted to log in with incorrect credentials, all text fields were erased and an inaccurate error message popped up. Now, contents of text fields are saved between attempts, and appropriate error messages appear. Passwords were added to the advisor side. Students need to now enter both a valid student ID and email in order to login.

Viewing advisors appointments is much easier. Before, all appointments were in a vertical list, with all students in each appointment also listed vertically. Now, advisors can view their appointments through a calendar, and can optionally expand a list of students in each appointment.

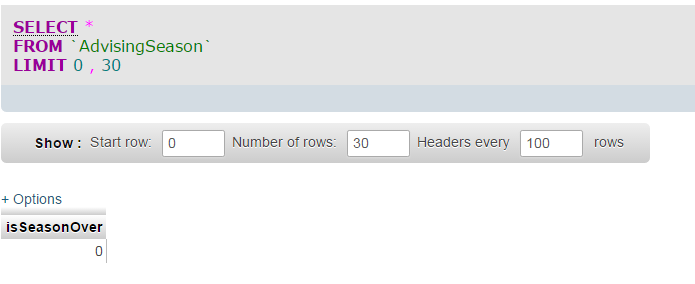
Advisors have now been given the option to set the maximum number of students for a group of students. Originally selecting group advising set the maximum number of students to ten, now the limit can be set between two and forty students.

6. Database Setup

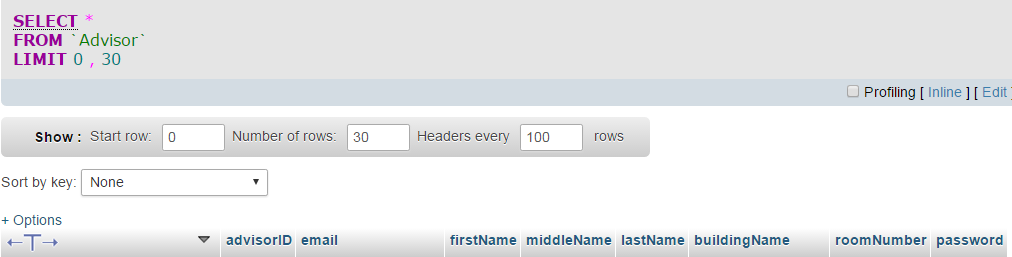
The current database uses seven tables to store all relevant information and perform the necessary functions of the website. The first table is called AdvisingSeason, this stores only one entry and that is an integer that makes the advising season either open or closed. The next table is called Advisor, this table stores all of the basic information about each advisor including their names, email, office. The table also stores an advisor ID number for them that will be referenced by other tables, as well as a password that allows them to log back onto their account. The third table is called AdvisorMeeting, this tables stores three integer values; the first is a unique ID for the data entered, the second ID references to which advisor the meeting is for, the third field references to an entry in the Meeting table. The next table is called Meeting, this stores all relevant information for the advising meetings including; meeting start and end times, the room in which the meeting is held, the type of meeting, how many students have signed up for the meeting and the maximum number of students that can sign up for the meeting. The next table is called Student; this stores all of the basic information for the students including name, email, school ID, major, and a flag that says their current meeting status. The next table is called StudentForm, this table stores all information included in the pre-advising form, this table has a total of sixty-three data fields in order to store all of the relevant information. The final table is called StudentMeeting, this table stores which student ID’s correspond to the meeting ID they have signed up for.



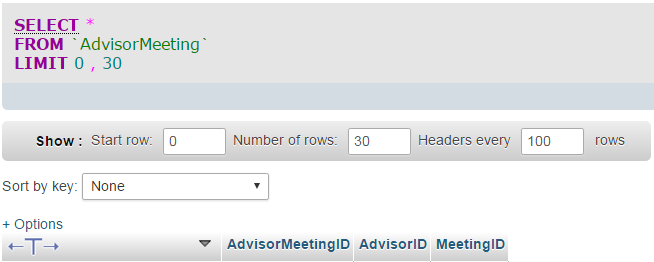
-list of tables in the database



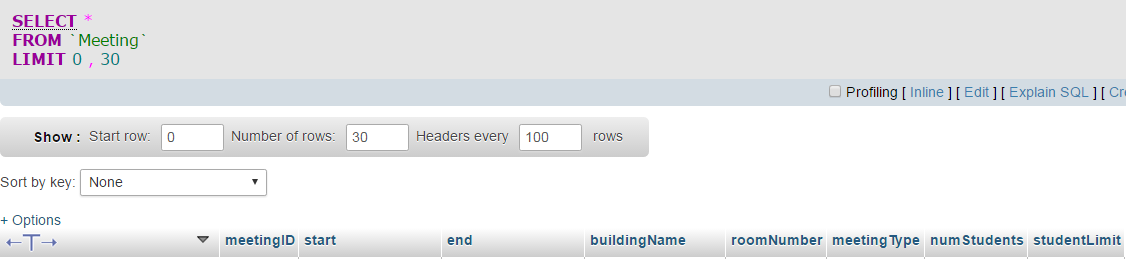
-AdvisingSeason table screenshot. Shows that advising season is not over



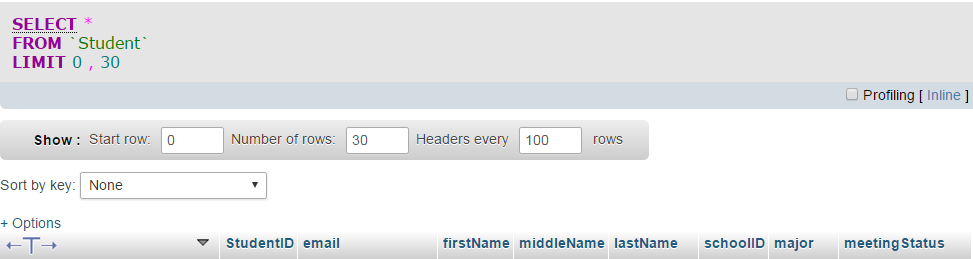
-Advisor table, showing individual data fields



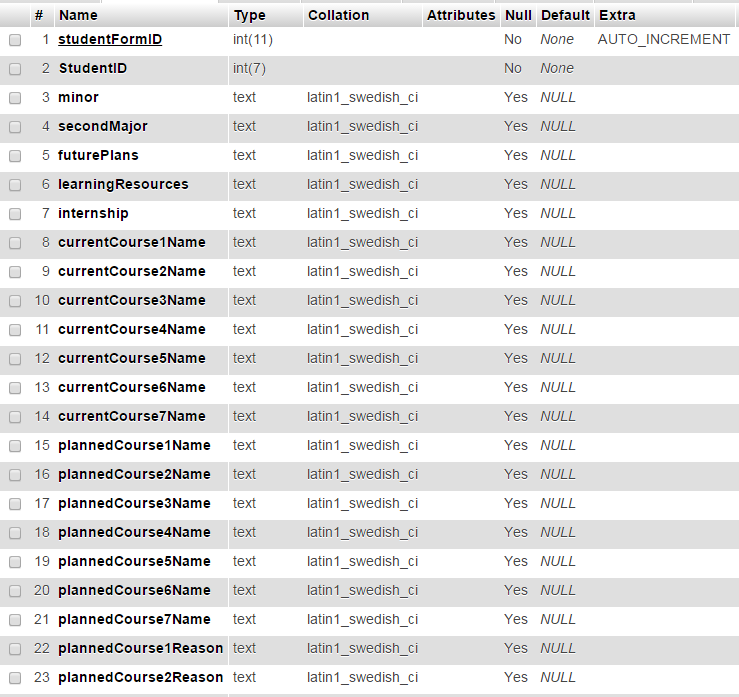
-AdvisorMeeting table, showing individual data fields



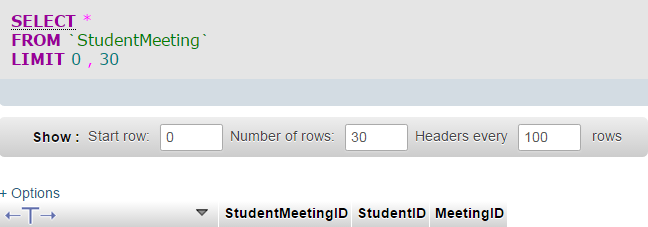
-Meeting table, showing individual data fields



-Student table, showing individual data fields



-StudentForm table, showing the first 23 out of 63 data fields



-StudentMeeting table, showing individual data fields

-- phpMyAdmin SQL Dump

-- version 4.0.10.17

-- https://www.phpmyadmin.net

--

-- Host: studentdb-maria.gl.umbc.edu

-- Generation Time: Dec 19, 2016 at 05:08 AM

-- Server version: 10.1.19-MariaDB

-- PHP Version: 5.4.44

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8 \*/;

--

-- Database: `phi3`

--

-- --------------------------------------------------------

--

-- Table structure for table `AdvisingSeason`

--

CREATE TABLE IF NOT EXISTS `AdvisingSeason` (

`isSeasonOver` tinyint(1) NOT NULL DEFAULT '1'

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--

-- Table structure for table `Advisor`

--

CREATE TABLE IF NOT EXISTS `Advisor` (

`advisorID` int(11) NOT NULL AUTO\_INCREMENT,

`email` varchar(50) NOT NULL,

`firstName` text NOT NULL,

`middleName` text,

`lastName` text NOT NULL,

`buildingName` text NOT NULL,

`roomNumber` text NOT NULL,

`password` text NOT NULL,

PRIMARY KEY (`advisorID`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=2 ;

-- --------------------------------------------------------

--

-- Table structure for table `AdvisorMeeting`

--

CREATE TABLE IF NOT EXISTS `AdvisorMeeting` (

`AdvisorMeetingID` int(11) NOT NULL AUTO\_INCREMENT,

`AdvisorID` int(7) NOT NULL,

`MeetingID` int(7) NOT NULL,

PRIMARY KEY (`AdvisorMeetingID`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=11 ;

--

-- Dumping data for table `AdvisorMeeting`

--

-- --------------------------------------------------------

--

-- Table structure for table `Meeting`

--

CREATE TABLE IF NOT EXISTS `Meeting` (

`meetingID` int(11) NOT NULL AUTO\_INCREMENT,

`start` datetime NOT NULL,

`end` datetime NOT NULL,

`buildingName` text NOT NULL,

`roomNumber` text NOT NULL,

`meetingType` tinyint(1) NOT NULL,

`numStudents` tinyint(2) NOT NULL,

`studentLimit` tinyint(2) NOT NULL,

PRIMARY KEY (`meetingID`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=11 ;

-- --------------------------------------------------------

--

-- Table structure for table `Student`

--

CREATE TABLE IF NOT EXISTS `Student` (

`StudentID` int(7) NOT NULL AUTO\_INCREMENT,

`email` text NOT NULL,

`firstName` text NOT NULL,

`middleName` text,

`lastName` text NOT NULL,

`schoolID` varchar(7) NOT NULL,

`major` text NOT NULL,

`meetingStatus` text NOT NULL,

PRIMARY KEY (`StudentID`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=3 ;

-- --------------------------------------------------------

--

-- Table structure for table `StudentForm`

--

CREATE TABLE IF NOT EXISTS `StudentForm` (

`studentFormID` int(11) NOT NULL AUTO\_INCREMENT,

`StudentID` int(7) NOT NULL,

`minor` text,

`secondMajor` text,

`futurePlans` text,

`learningResources` text,

`internship` text,

`currentCourse1Name` text,

`currentCourse2Name` text,

`currentCourse3Name` text,

`currentCourse4Name` text,

`currentCourse5Name` text,

`currentCourse6Name` text,

`currentCourse7Name` text,

`plannedCourse1Name` text,

`plannedCourse2Name` text,

`plannedCourse3Name` text,

`plannedCourse4Name` text,

`plannedCourse5Name` text,

`plannedCourse6Name` text,

`plannedCourse7Name` text,

`plannedCourse1Reason` text,

`plannedCourse2Reason` text,

`plannedCourse3Reason` text,

`plannedCourse4Reason` text,

`plannedCourse5Reason` text,

`plannedCourse6Reason` text,

`plannedCourse7Reason` text,

`plannedCourse1Credits` tinyint(4) DEFAULT NULL,

`plannedCourse2Credits` tinyint(4) DEFAULT NULL,

`plannedCourse3Credits` tinyint(4) DEFAULT NULL,

`plannedCourse4Credits` tinyint(4) DEFAULT NULL,

`plannedCourse5Credits` tinyint(4) DEFAULT NULL,

`plannedCourse6Credits` tinyint(4) DEFAULT NULL,

`plannedCourse7Credits` tinyint(4) DEFAULT NULL,

`creditsEarned` smallint(6) NOT NULL,

`GPA` float NOT NULL,

`upperLevelCredits` smallint(6) NOT NULL,

`numWritingIntensives` tinyint(4) NOT NULL,

`numPhysicalEds` tinyint(4) NOT NULL,

`numEnglishComp` tinyint(4) NOT NULL,

`numArtsAndHumanities` tinyint(4) NOT NULL,

`numSocialSciences` tinyint(4) NOT NULL,

`numMathSciences` tinyint(4) NOT NULL,

`numCulture` tinyint(4) NOT NULL,

`languageProficiency` tinyint(1) NOT NULL,

`performanceReflection` text,

`studiedWithFriends` tinyint(1) NOT NULL,

`classQuestion` tinyint(1) NOT NULL,

`notes` tinyint(1) NOT NULL,

`BBDiscussion` tinyint(1) NOT NULL,

`tutorialCenter` tinyint(1) NOT NULL,

`RLCTutor` tinyint(1) NOT NULL,

`officeHours` tinyint(1) NOT NULL,

`emailProfessor` tinyint(1) NOT NULL,

`volunteerActivities` text,

`currentlyEmployed` text,

`commuter` text,

`commuteHours` tinyint(4) NOT NULL,

`workHours` tinyint(4) DEFAULT NULL,

`familyHours` smallint(6) NOT NULL,

`extracurricularHours` smallint(6) NOT NULL,

`additionalComments` text,

PRIMARY KEY (`studentFormID`),

UNIQUE KEY `StudentID` (`StudentID`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=5 ;

-- --------------------------------------------------------

--

-- Table structure for table `StudentMeeting`

--

CREATE TABLE IF NOT EXISTS `StudentMeeting` (

`StudentMeetingID` int(11) NOT NULL AUTO\_INCREMENT,

`StudentID` int(7) NOT NULL,

`MeetingID` int(11) NOT NULL,

PRIMARY KEY (`StudentMeetingID`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=11 ;

7. Languages Used

We used PHP, JavaScript, and MySQL to complete this project. PHP was used on almost every page, JavaScript was used for authentication and validation, and MySQL let us interact with the database.