**CMSC331 Project2**

1. Our Team

**Member (email)**: *Role(s)*

**Aaron Frishman (**[**aaronf1@umbc.edu**](mailto:aaronf1@umbc.edu)**)**: *Presentation, Slick Sheet*

**Andrew Levin (**[**landrew2@umbc.edu**](mailto:landrew2@umbc.edu)**):** *Database, Student-side*

**Jaime Oreillana (**[**jaime5@umbc.edu**](mailto:jaime5@umbc.edu)**)**: *Advisor-side*

**Kwame Owusu-Boaitey (**[**kwame3@umbc.edu**](mailto:kwame3@umbc.edu)**)**: *Advisor-side*

**Max Ricketts-Uy (**[**ricke1@umbc.edu**](mailto:ricke1@umbc.edu)**)**: *Student-side*

1. <https://swe.umbc.edu/~landrew2/CS331_Project2/src/student/>
2. **Project Description:** Our team has developed a website for scheduling advising appointments for the College of Natural and Mathematical Sciences. This includes individual portals for advisors to create appointments, students to select appointments, and both to manage their schedules and profiles.
3. **Components Added:**

**Student Side**

editStudentInformation.php

freeResponse.php

login.php

otherMajor.html,

registration\_error.html.

editStudentInformation.php

This page allows the student to change their name, major and free responses which are updated in the database and correctly displayed whenever the student reopens this page again. If other is selected they are also sent to the otherMajor page

freeResponse.php

This page prompts the user to answer two free response question which can help the advisor better understand the students future plans and concerns going into the advising appointment.

login.php

This page allows a pre-existing user to login directly which was not an option in the given code. This validates the users email and password using regex and also checks to see if the student has a scheduled appointment.

otherMajor.html

This page is only entered if the user enter “Other” for major when registering or editing their information. This page includes a link to the main departmental advising page for UMBC so they will be able to contact the appropriate person for their desired major. Also includes a button to login / return to the menu page if they accidently entered “Other”.

Registration\_error.html:

This page is entered if the student attempts to register with a pre-existing campus Id, this capability did not exist in the given code. From there the student can login with the same id or register with another campus id.

**Database:**

advisor\_table, advising\_table, student\_table, were renamed to Meetings, Advisor,and Student respectively for clarity.

Added Shut\_off\_Control table to track whether the student side is on or offline.

In the former advisor\_table got rid of campusID1,…,campusID10. The Student table now has the foreign key meetingID to track the meeting the student is registered for.

The Student table no longer has columns for minor, second major , or any of the classes the student is taking, planning on taking, credits for said classes. We also removed the answer fields for many of the questions the previous group asked. We decided that these were not necessary to the functioning of our site.

In Meetings we added an advisorID foriegn key to keep track of which advisor created which meeting.

Also in Meetings we combine the dateTime fields using the DATETIME sql object.

In advisor we added fields for the salt and the password. Since we salted and hashed our passwords we needed to store the salt and the salted and hashed version of the password for login.

**Advisor:**

### Structure:

* Registration Page
* Login Page
* Logout Page
* Advising (dashboard)
* Edit Profile Page
* Controls Page (turn off future advising)
* Utils
* Third Party Libraries

## #### Registration Page

Let's the user register as an advisor. Currently anyone can register as an advisor so it's recommended that a user

##### registration.php

Holds html information for registering a new user, as well as redirecting users if they have already registered ##### process.php Uses queries.php to process a users registration. simple POST processing file for the advisor registration form. Only takes in POST requests.

## #### Login Page

Let's the user login to their account. Perform various errors to the user using URL encoded error messages. Also has links to registration page should user require.

##### login.php

Holds html information for logging users, as well as dynamic error formats to see show users errors ##### process.php Uses queries.php to process a users registration. simple POST processing file for the advisor registration form. Only takes in POST requests.

## #### Logout Page

##### logout.html

A confirmation page that lets the user logout of his current active session by deleting all of the session variables in php. If the user says no to the logout then the user is redirected back to advisor dashboard

##### logout.php

Processes user logout deleting his current active session and redirecting that user to the login page.

## #### Advisor Management Page (Big Kahuna)

Let's the advisor see the appointments they currently have in a calendar UI using bootstrap-calendar, an open-source calendar.

##### advising.php

Holding the calendar, modals as well as creating an appointment. Various components are brought through the website.

##### createAppointment.php

POST Request Only processes logic to generate a new appointment based on information given in advising.php #####deleteAppointment.php POST Request Only deletes an appointment given #####feedEvents.php used for AJAX call inside of calendar.js bootstrap-calendar.js requires this as a part of their schematics.

##### getStudentInfo.php

POST Request Only a GET request that obtains student information based on student ID.

## #### Edit Profile Information Page

Let's the advisor edit all of his profile information, including his password. #####editProfile.php Has form that lets user edit their information and update it with whatever information they wish to place.

##### updateAdvisorProfile.php

Processes information based on submission of user to DB

## #### Controls Page

Let's the advisor select whether or not to turn off advising appointments so that users can't login and use the advising site.

##### controls.php

Holds HTML for confirming/declining the appointment for registering on a website.

##### switch.php

Processes logic to turn off the student advisor login page. Edits the table such that it turns off the website.

## #### Utils

##### dbconfig.php

Has recallable function that manages connection to remote DB. Easily editable

##### queries.php

Holds various functions that all run queries onto the DB. Goal was to orientate all queries within one folder/file to separate complexity.

1. **Improvements from Old Code:**

**Student Side:**

appointmentPage.php, cancelPage.php

CommonMethods.php

group.php, group1.php, index.html

individual.php, indiv.php, logout.php

menu.php, processMenu.php

register.php, viewAppt.php.

appointmentPage.php:

Minor changes to include custom built CSS selectors

cancelPage.php:

Added SQL and PHP to actually handle request to cancel an appointment which was nonexistent in the given code. SQL updates both meeting id status of the student as well as the available value for meeting in the Meeting table

group1.php:

group1 now provides the student with the options to search for appointments based day and reset their options once they are displayed. A lot of styling was added as well to simplify and clean up output.

group.php:

group.php now completes the updating sql command in the database. Before this page used multiple if’s update the meeting that was selected but our code only uses only uses 2 lines to do the same thing.

Index.html:

This is the registration page and replaces login.html which would was meant to register and login the user but did not work. Now index.html POSTs to register.php

Individual.php

Our new updated individual page uses far fewer line and allows the user to choose from all available appointments or choose based on advisor.

Indiv.php

Indiv updates the meetings and student table once an individual appointment selected. The indiv provided attempted to do this in over 150 lines but did not work while our successfully updates the database and assigns a meeting to a student in ~40.

Logout.php

Mostly remained the same, redirected to index.html

menu.php

Our menu page has added cancel appointment and edit student information functionality.

processMenu.php

Is mostly unchanged, being that it only acts as a form.

Register.php

Registers the student, adding them to the database, and send them to freeResponse or the error page if their campus id has already been registered. The former team’s register did none of these things. There was no input validation and a student was never added to the database.

ViewAppt:

This page selects the meeting information from our database and displays it to the student. This was not possible with the given code before, since no information was ever recorded in the database.

**Database:**

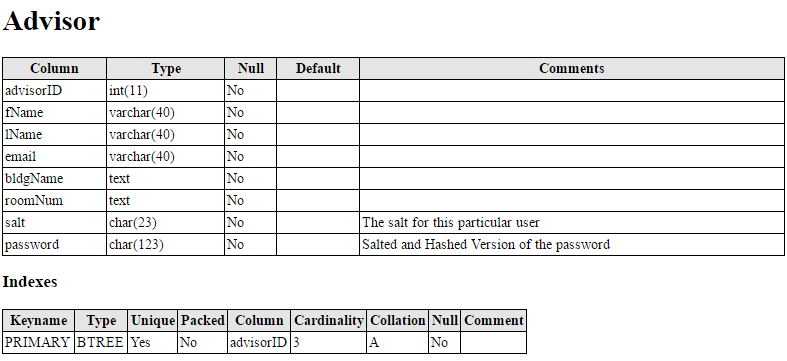
Table names were changed for clarity.

Keeping information about a student’s meeting is substantially less redundant.

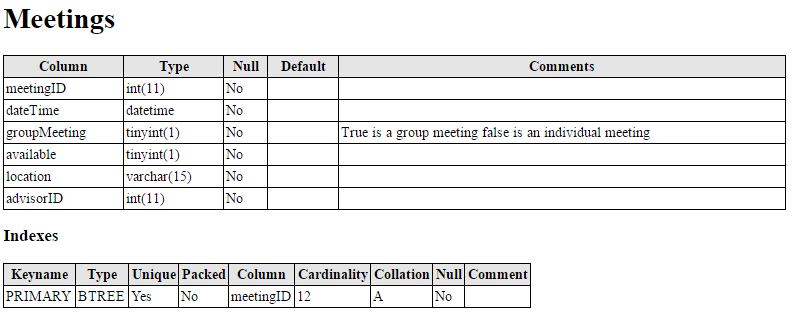
Student no longer has useless information.

Logging in on the advisor side is now secure thanks to salting and hashing.

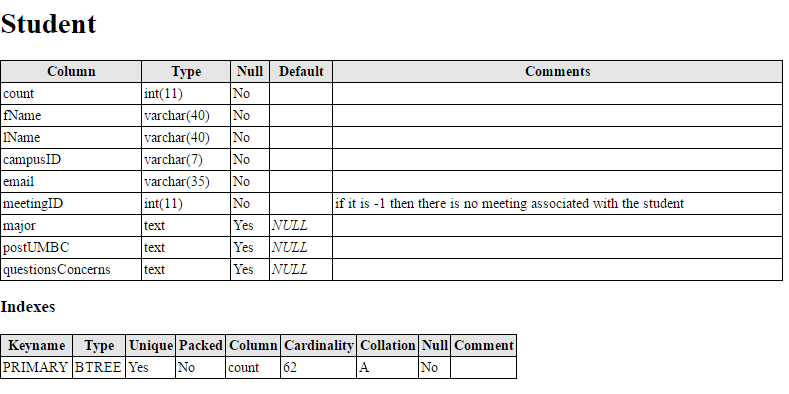
1. **Database Structure**



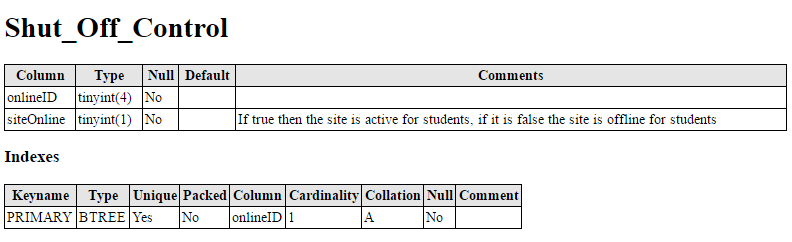
This table stores all the personal information of the advisor.



This table stores all the information pertaining to each meeting and the Id of the advisor that created it



This table stores all the information pertaining to each student and which meeting they are registered for.



This table simply stores if the system is currently online for students.

1. Language Used:

Student:

PHP:

All pages excluding index.html, otherMajor.html, registration\_error.html contain PHP.

**Advisor:**

#### Third Party Libraries:

* Bootstrap - front-end framework for fast designing
* JQuery - animations for frontend as well as Ajax calls to get information
* Bootstrap Calendar - UI for making pristine and elegant calendars
* Moment - for managing time and dates of meetings
* Underscore - templating engine and various handy JS functions

1. Supporting Materials:

YouTube Video: <https://www.youtube.com/watch?v=9FTsBLH52XU>