**#Advisor Documentation**

**---**

**## advisor/views**

**---**

**There are four main webpages that are used to navigate the the advisor site.**

**#### index.php**

**Note that `index.php` is the starting point for a new advisor since it contains the forms for registering an advisor.**

**Somethings to note here are that it redirects you to the meeting dashboard if you're already logged in the top.**

**The are error messages that are stored in the user's session.**

**The logic for processing the forms is in `advisor/utils/forms/registerAdvisor.php`**

**#### login.php**

**On the client side it contains a single form that lets the user input their data.**

**It also processes the POST data on the top and verifies if the user's email exists and is valid.**

**#### homepage.php**

**The main web dashboard for the advisor.**

**Contains forms for creating new meetings, and deleting meetings.**

**Also has various SQL queries to search for information on students.**

**#### logout.php**

**There is very little logic here, the main thing is that it clears the session variables and redirects the user to the login `login.php`**

**## advisor/utils**

**---**

**#### dbconfig.php**

**Holds a list of static variables that allows you to access the database**

**#### forms/createMeeting.php**

**Contains logic for creating meetings. Also stores errors into session variables should the inputs be incorrect. Uses raw SQL queries to process inputted values into database.**

**#### forms/deleteMeeting.php**

**Where logic for checking if a meeting exists and deleting such meeting is placed. Also stores errors into session variables should there be any errors. Uses raw SQL queries to process inputted values into database.**

**#### forms/registerAdvisor.php**

**Has logic for creating an advisor . Also stores errors into session variables should the inputs be incorrect. Uses raw SQL queries to process inputted values into database.**

**STUDENT DOCUMENTATION**

**Student directory depends on CommonMethods.php:**

CommonMethods.php must be altered on these lines to be used with your database:

Line 11: $rs = $this 🡪connect(“DATABASE PASSWORD”);

Line 19: $conn = @mysql\_connect( <DATABASE URL>, <USERNAME>, <PASSWORD>)

Commented lines 22 and 29 can be used for debugging

**Student directory includes**:

index.php

login.php

homePage.php

viewMeeting.php

logout.php

cancelMeeting.php

chooseGroupMeeting.php

chooseIndividualMeeting.php

chooseMeeting.php

deleteMeeting.php

meetingChosen.php

**index.php**

This is the registration page for students. It uses a form to accept basic data to be stored and used in your database.

Regex is used to validate email address input which reports an error when incorrect.

Empty fields are handled with error messages.

Email is used to validate if a student record exists in the database table, so if preexisting emails are entered an error is thrown and the information is not entered in to table.

After sql query is executed on line 122, the user is redirected to login.php

**Login.php**

This is the login in portal for students. It uses a form to accept an email that is compared to preexisting data in Student table.

If the text field is left unfilled or an unregistered email is entered an error is thrown and handled.

After a legitimate email is entered four session variables are defined using data from the Student table; they include a boolean to signify that the user is logged in, and three strings for email, studentID and major which are used later.

The user is redirected to homePage.php.

**homePage.php**

This portal is used to handle appointment processes.

Using provided links the users can navigate to:

viewMeeting.php(“View Scheduled Appointment”)

cancelMeeting.php(“Cancel Advising Appointment”)

chooseMeeting.php(“Schedule Advising Appointment”)

The user can also logout and is redirected to login.php after passing through logout.php

**viewMeeting.php**

This portal uses predefined session variables to access the StudentMeeting and Meeting tables to print the students currently scheduled appointment.

If the StudentMeeting query on line 15 when executed on line 18 reports now rows with the StudentID of the student currently logged in message is printed letting them know they have no scheduled appointments

**logout.php**

This small file destroys the current session and redirects the user to the login.php page where they can then login or redirect to index.php

**cancelMeeting.php**

cancelMeeting.php is responsible for allowing a user to cancel their meetings. If the student are already registered for a meeting, then it will display a confirmation to check whether they want to go through with the cancelation. If they select yes then they will be redirected to deleteMeeting.php and the meeting will be deleted. If they select “NO” then they will simply be redirected to the homepage, homepage.php.

In the event that the user does not currently have a meeting selected, cancelMeeting.php displays a notice and allows the user to go back to the home page.

**chooseGroupMeeting.php**

chooseGroupMeeting.php is like chooseIndividualMeeting.php, it lists all the available group meetings and allows the user to select one using a radio button. It then proceeds to add the appointment to the student Meeting table in the database and increments the number of students registered for the meeting in the Meeting table.

In the event there are no group meetings available or the user is already registered for a meeting the page will display a notice and will have a link for the user to go to the homepage.

**chooseIndividualMeeting.php**

chooseIndividualMeeting.php is similar to chooseGroupMeeting.php, it lists all the available individual meetings and allows the user to select one using a radio button. It then proceeds to add the appointment to the student Meeting table in the database and increments the number of students registered for the meeting in the Meeting table.

In the event, there are no individual meetings available or the user is already registered for a meeting the page will display a notice and will have a link for the user to go to the homepage.

**chooseMeeting.php**

This page simply consists of two buttons that allows the user to choose between registering for a group and an individual meeting.

**deleteMeeting.php**

This page runs the php and sql the allows a student to cancel their registration.

For both types of meetings the appointment is deleted from the studentMeeting table. Then, in the Meetings table the quantity of students registered for that meeting is decremented.

After the meeting is fully deleted the user is redirected to the home page.

**meetingChosen.php**

This page simply consists of the warning message shown when a user attempts to register for a advising meeting while already registered for a meeting.

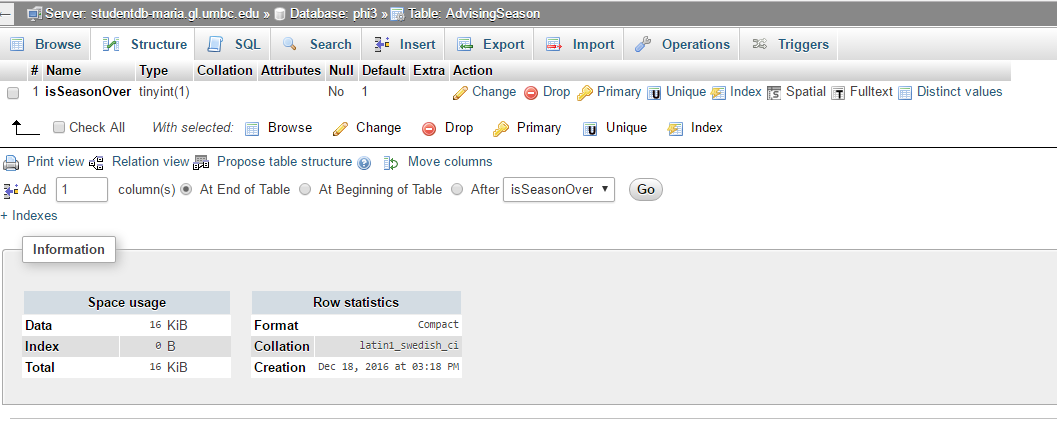
There is a link that allows the user to go back to the home page.

**Database:**

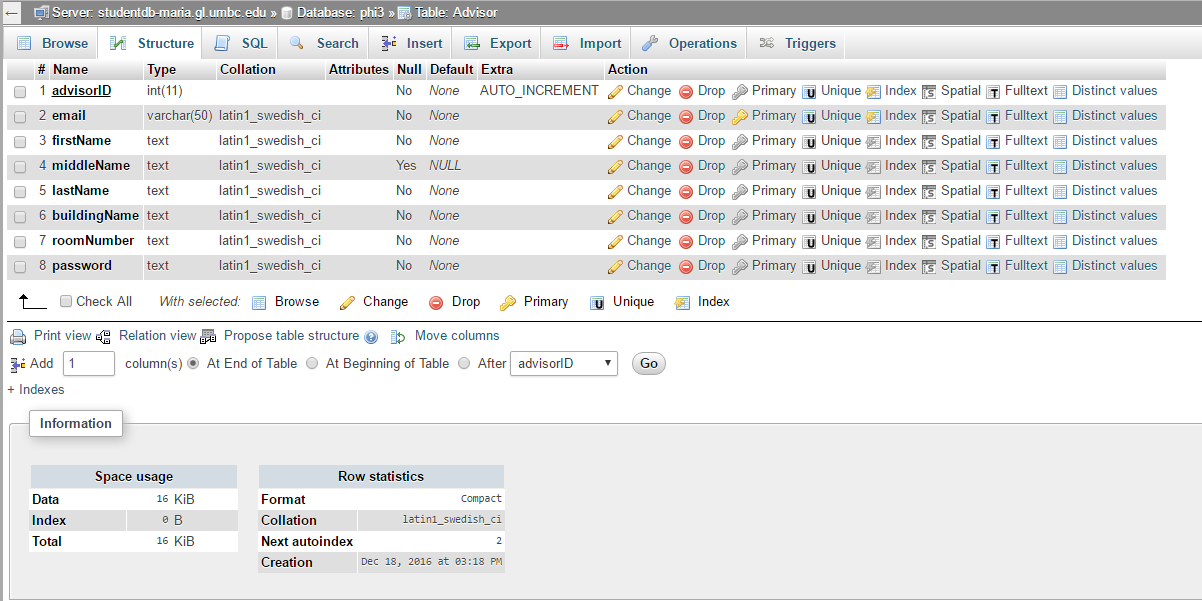
The database can be set up by running dropDB.php followed by setUpDB.php. PHP code for creating the tables in the database can be found in setUpDB.php.

**Tables:**

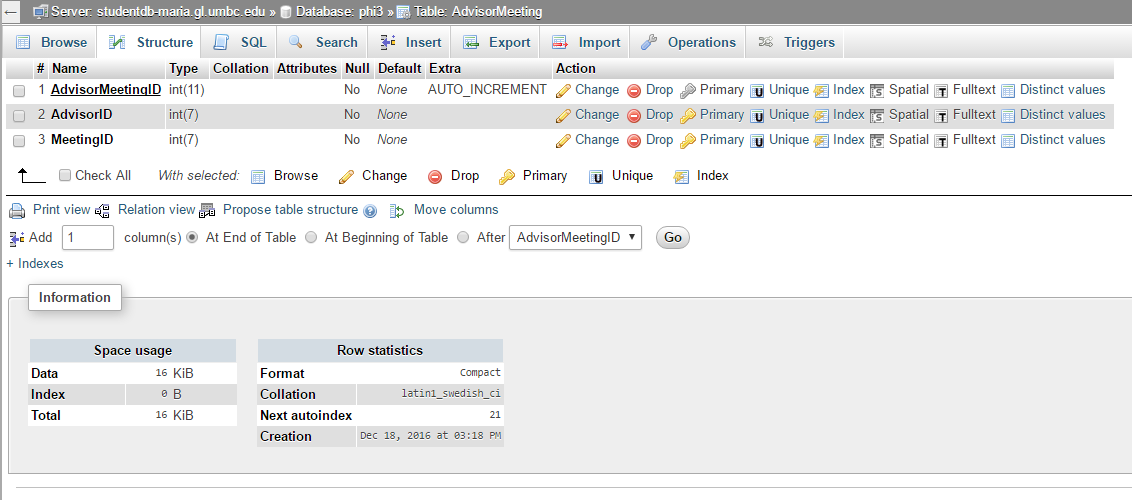
**Advising Season**



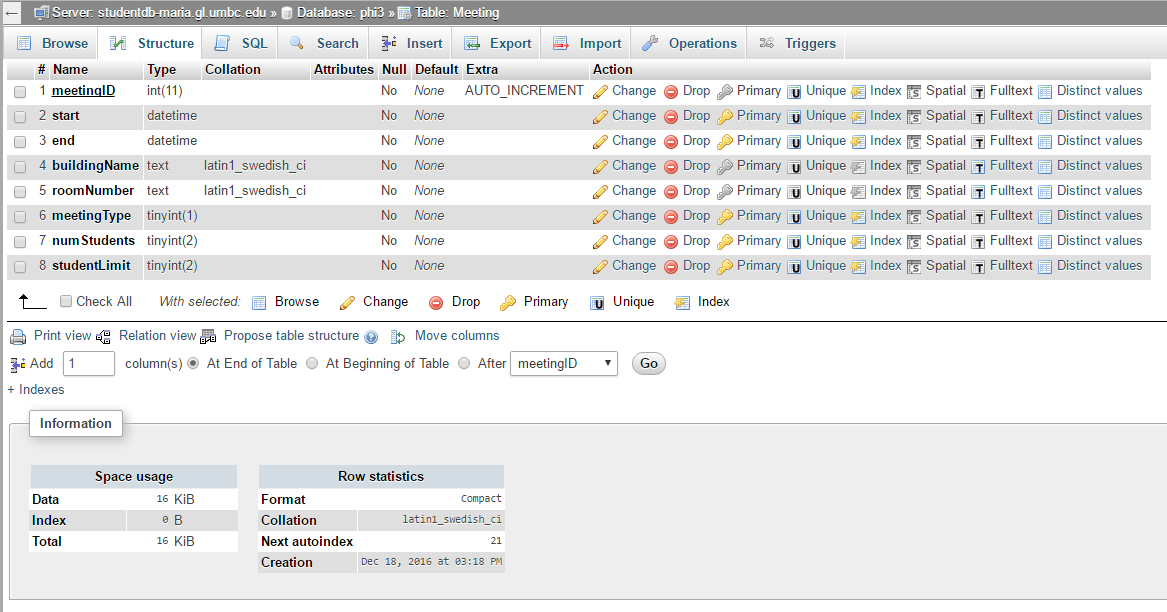
**Advisor**



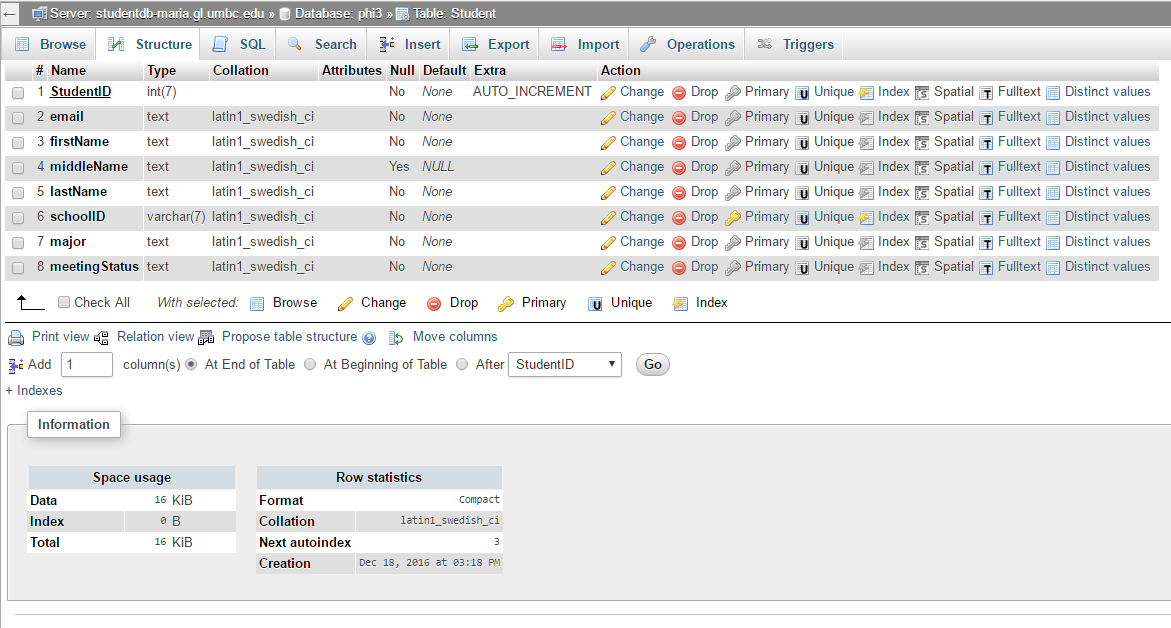
**AdvisorMeeting**



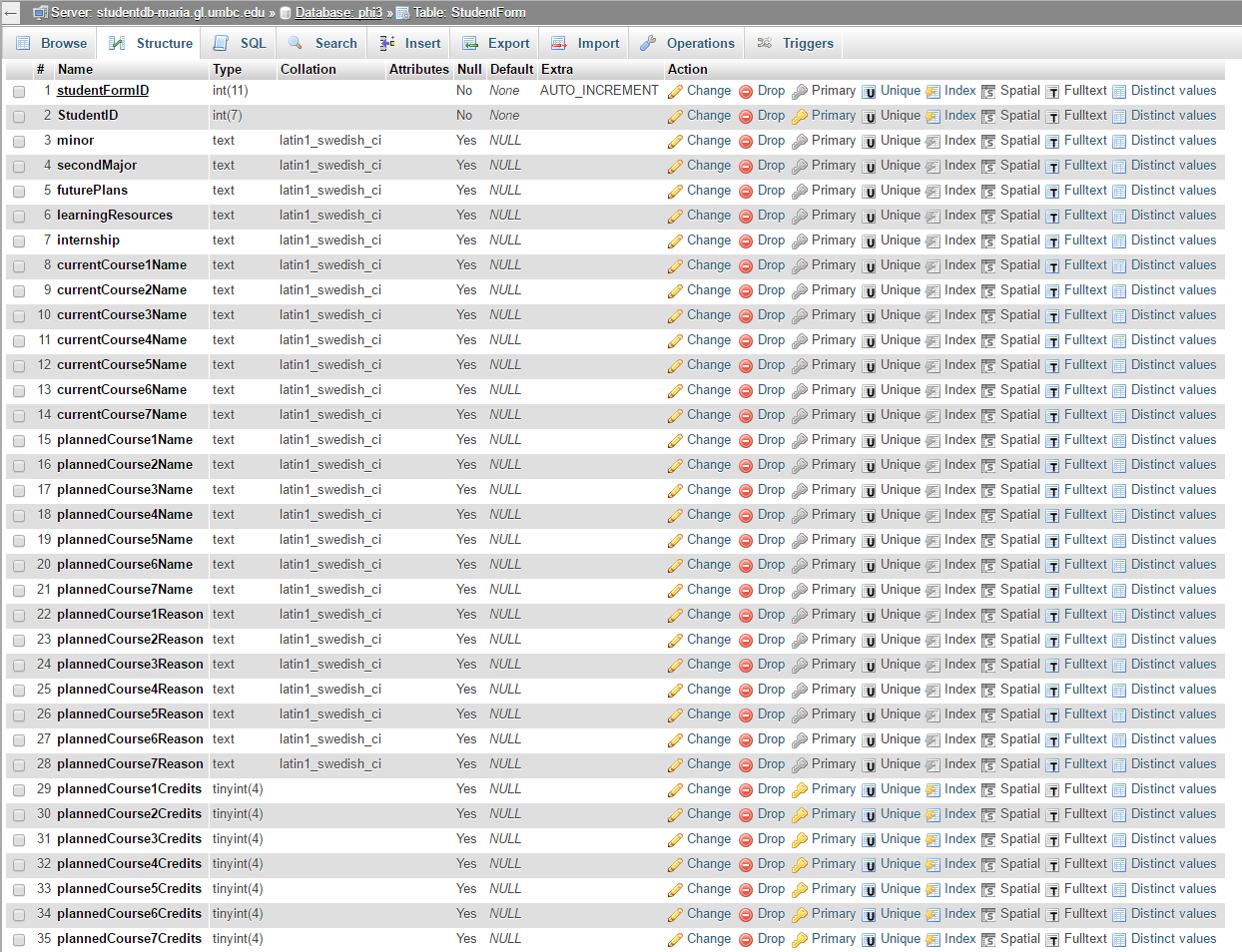
**Meeting**



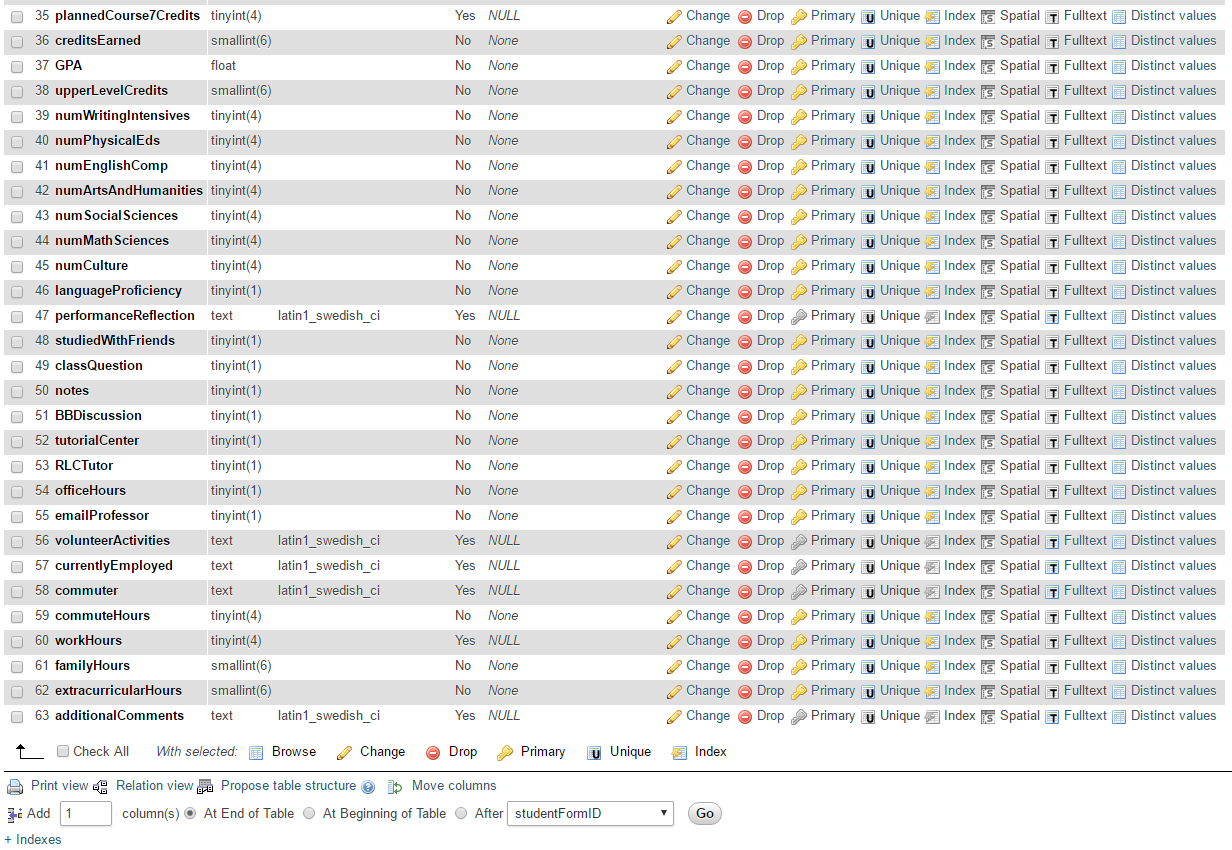
**Student**



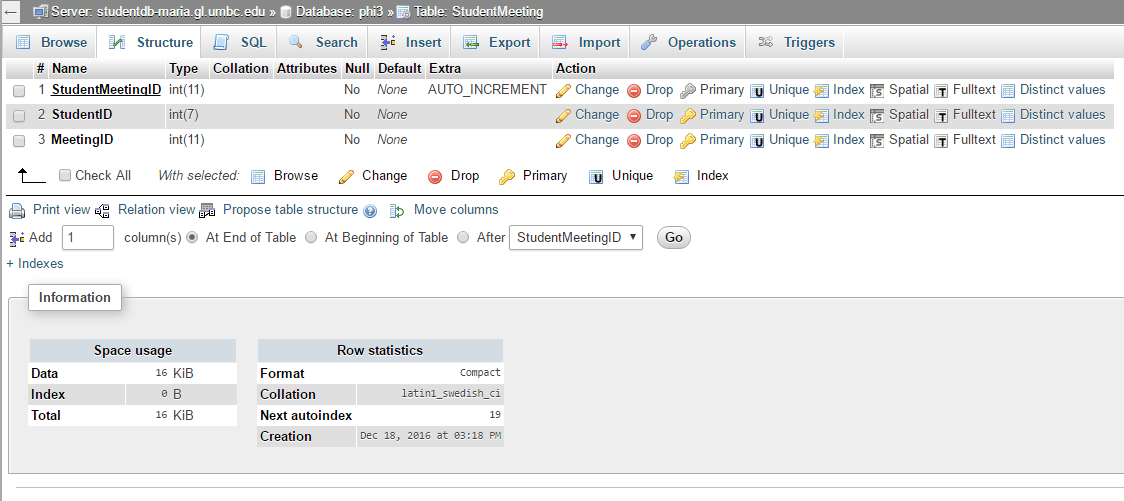
**StudentForm**



**StudentForm (continued)**



**StudentMeeting**



**Comments:**

Videos on how to use the service as provided can be found at:

<https://www.youtube.com/playlist?list=PLCzDYNvPl8px07Wu9PNevwMN3bOPmCtmx>

AdvisorMeetings and StudentMeetings are junction tables between Advisor and Meeting and Student and Meeting respectively. The only things they contain are the foreign keys from the tables they join and an identifier ID.

#Advisor Documentation

---

## advisor/views

---

There are four main webpages that are used to navigate the the advisor site.

#### index.php

Note that `index.php` is the starting point for a new advisor since it contains the forms for registering an advisor.

Somethings to note here are that it redirects you to the meeting dashboard if you're already logged in the top.

The are error messages that are stored in the user's session.

The logic for processing the forms is in `advisor/utils/forms/registerAdvisor.php`

#### login.php

On the client side it contains a single form that lets the user input their data.

It also processes the POST data on the top and verifies if the user's email exists and is valid.

#### homepage.php

The main web dashboard for the advisor.

Contains forms for creating new meetings, and deleting meetings.

Also has various SQL queries to search for information on students.

#### logout.php

There is very little logic here, the main thing is that it clears the session variables and redirects the user to the login `login.php`

## advisor/utils

---

#### dbconfig.php

Holds a list of static variables that allows you to access the database

#### forms/createMeeting.php

Contains logic for creating meetings. Also stores errors into session variables should the inputs be incorrect. Uses raw SQL queries to process inputted values into database.

#### forms/deleteMeeting.php

Where logic for checking if a meeting exists and deleting such meeting is placed. Also stores errors into session variables should there be any errors. Uses raw SQL queries to process inputted values into database.

#### forms/registerAdvisor.php

Has logic for creating an advisor . Also stores errors into session variables should the inputs be incorrect. Uses raw SQL queries to process inputted values into database.