The Sancus Code

AIO Project - Final Report

Second Term of Work
Summer 2018

By:
Sarah Colborne (Team Lead)
Chris Connors (Developer Team Lead)
Stuart Laurie
Scott Martell
Seth Myers
Nnadozie Ogbudibe
Ryan Stevens (Hacking Team Lead)
Bjorn Huntemann
Noah Attwood

August 1st, 2018

Table of Contents

Acknowledgements

This final report contains content which was modified from the previous group's final report, supplied to us at the beginning of this project.

List of Deliverables

This project includes several deliverables. These deliverables will be presented to the client and delivered to the next team that takes on this project. The following is a list of these deliverables:

1. **The Code**
   The code we are delivering is a mix of HTML, CSS, Bootstrap, PHP, JS, MySQL to create a web portal accessible by the computer science faculty to submit and process academic integrity violation cases.

2. **The Server**
   The server used for this project is located within the CS building. It hosts our web portal, and our database. The server is running Apache (version 2.4.29), MySQL, and LaTeX and is maintained/controlled by the CS help desk.

3. **The Database**
   The Database is used to store and organize all data related to the students, professors, and AIO's involved in an academic integrity violation case.The database features support for a large amount of active cases at any given time, and the ability to store temporary data in the event of a form only being partially completed.

4. **User Manual**
   This document, included in this report, details how a given user would interact with the system, depending on the role they are playing in academic integrity process. It will provide information of the features

5. **Presentation and Demo**
   Our presentation will contain the progress we have made and detail what we have learned this term which can help future groups working on the AIO portal. In the demo, we will demonstrate this progress, and show the state of the project.

6. **Final Report**
   This document contains a detailed report of what has been accomplished this term, what remains outstanding, and what remains left to be done by future teams. It also contains other helpful resources for the next team working on this project, including lessons learned and recommendations.

Project Background Summary

Unfortunately, one of the realities at our university is that students sometimes commit plagiarism on their papers, assignments, and projects. To handle these instances, there is a process which must be followed for each case. This process begins when the professor submits an allegation against one or more students, which an Administrator then assigns to an Academic Integrity Officer (AIO) to be evaluated. The AIO first must check with the Senate to determine if the student has had a previous offense. If so, the case is forwarded to the Senate. If not, the AIO can handle the remaining process themselves, which involves meetings with the student and professor that help the AIO conclude the case with a verdict. The student themselves can accept or deny this verdict, with a denial resulting in the Senate's involvement in a retry of the case.

At the moment, this entire process is performed through numerous emails between all involved parties. While this works, it is inefficient and difficult to manage. Emails are not fail-safe and could lead to the loss of important documents for a case.

The solution being developed is a web portal which drastically reduces the amount of emails and assists with management of each case. This is particularly helpful to AIOs who may need to deal with many cases simultaneously. It also helps standardize the way that cases are submitted and evaluated. This is an ongoing project, worked on by one group in the past, and it is expected that at least two more development iterations will be required.

Project Description

This project involves developing a web portal to be used by professors, academic integrity officers and administrators to more efficiently submit, process, and resolve academic integrity cases.

The role of the professor is to simply submit an allegation. This is done by logging into the portal and filling out Form A with the information about the case and any evidence that may prove wrongdoing. If the professor wants to modify their submission (to add more evidence, for example) they can do this.

The role of the AIO is to process a case once it has been submitted by the professor. Upon logging in the AIO can view a case from a list of their active cases, or assign themselves to a case from a list of unassigned cases. After selecting a case to view, they are able to fill out Forms B and D, which are used for forwarding the case to the senate and making a final decision on the case, respectively. The AIO also has the ability to stop working on a case, close a case once a verdict has been reached, and remove a case if insufficient evidence is provided.

The role of an Administrator is to quite varied. Once logged in, they are presented with a list of all of the active cases. They have the ability to submit a new case on behalf of a professor, and can perform some of the actions an AIO can. They also have the ability to assign or reassign the AIO on a case, and delete a case. They are also responsible for filling out Form C, which is used to schedule a hearing with a student involved in a case.

Comparison of Planned Versus Actual

This table summarizes the work that was completed over the term and compares it to what was planned at the beginning of this iteration of the project.

Completed = feature was fully implemented and merged into the web portal

Partially Complete = feature was partially implemented and either was merged with partial functionality or remains unmerged with progress pushed to a branch. In both cases, progress made and remaining work to be done is documented.

Outstanding = feature implementation was not started

Removed from Scope = feature implementation was removed from the project scope

Planned Features	Actual Progress
Close Case Button	Complete
Remove cases with insufficient evidence provided	Complete
Change AIO	Complete
AIO can view and self-assign unassigned cases	Complete
Logout feature	Complete
Save form progress	Complete
AIO can accept or deny a case	Complete
Login to available account roles using CS ID	Complete
Professor can add evidence to a case	Complete
Download case documents	Complete
Admin FAQ page	Complete
Submit new cases as an admin	Complete

Professor can submit an allegation	Complete
Download PDF versions of forms	Complete
Informative login failures	Complete
Auto-fill account based information in forms	Complete
Admin can notify student and professor of meeting	Partially Complete (email does not work yet, but full UI is present)
Forward Case to Senate	Partially Complete (email to Senate does not yet include attachments)
Student can accept or reschedule meeting	Outstanding
Sort AIO’s tasks by which ones require attention	Outstanding
Load saved progress in forms	Removed from scope
Admin portal to add/remove/modify user accounts and permissions	Removed from scope
Digital signatures	Removed from scope

Outstanding Issues

The following are the issues that are outstanding at the end of the project:

Change AIO Page Case Information - Allows an admin to change the AIO for a given case. However the information on this screen contains placeholder information.

Student Case Information Page - Page does not display any relevant data; all values are place holders. This page may be eliminated in the future, in favor of a main Case Information page that supports multiple students.

Confirm Form Resubmission Screen - When returning back to the case information page with the press of the back button on your browser, the uses observes a form resubmission error. Some preliminary research suggests that this issue can be resolved but we did not have time to pursue the issue further.

Case Status Field - The case status field on the case information screen does not have any logic behind it.

View Cases that Need Attention - The AIO should be able to sort and view cases based on whether they require action by the AIO or not. This task is closely related to the one above.

Adding/Modifying Users - The admin should be able to add new users and remove current users from the portal. Adding a new user will permit them to login to the system. Removing a user will mean they can no longer login to the portal. The admin should also be able to modify user permissions by assigning account roles, such as making an AIO also be a professor.

Resume Saved Work - Progress on form A can be saved to the database, but there is currently no implementation for resuming work on the form.

Forward Case to Senate - Currently the user is able to open a form which allows them to construct and send an email to the senate. However, it does not attach the case evidence nor PDF forms to the email. This is something that a future group will need to look into adding.

Saving form progress - Currently the save form progress works, but if a user attempts to save a second time, an error occurs. This could likely be solved by checking the database for an already existing entry and updating it.

Form Processing on Forms B and D - Currently only Forms A and C have any kind of form processing implemented. Future groups should make it a priority to implement form processing on the rest of the forms.

Bootstrap Dropdowns - Near the end of development we encountered an issue where Bootstrap dropdowns were not displaying as they should. We were unable to fix this issue before the end of the term, and because of this, had to revert back to standard dropdowns. This is an issue future groups will have to look into fixing.

Test Cases - Near the end of development, when a surplus of new features were added, the existing test cases became obsolete, will now all fail, and must be rewritten to accommodate new code organization and new features.

We have also added issues to our GitLab repository pointing out some known bugs which should be resolved in the next term of work.

Original Scope and History of Approved Changes

The original scope for this project was mainly inherited from the outstanding features and issues left from the group working on this project last term. Since much of the front-end user interface was designed and implemented last term, our focus was to develop the back-end functionality required for the portal to function. Another major focus was to implement the desired login system and ensure security.

As with many development projects, some changes were required during the execution phase to our scope. The first change made was to add a feature which would allow administrators to submit a case on behalf of a professor. This was requested by the client and increased the time which would be required to complete the project.

Unfortunately, given the time restraints of the school term, our team was unable to make progress on all of our originally planned features. In order to decrease the time required to fulfill our scope, three features were removed from our scope late in the term with approval from our client. The features were:

1. Digital signatures for submitted forms
2. Admin portal used to add/remove/modify user permissions and roles
3. Reloading previously saved form progress

Our team was fortunate to have a client who is knowledgeable and understanding of the development process and how delays in projects may occur. We were able to negotiate scope changes without worrying that we would be severely disappointing our client.

Burndown Chart

The following burndown chart (figure 1) indicates our actual velocity throughout the term compared to the ideal estimated velocity. Our team experienced two periods where our velocity was especially slow. This occurred when we initially began the execution phase, due to the learning curve of this project, and also after the midterm presentation with the client. The second instance occurred due to some major changes in the code which required some refactoring as well as our strict merge request reviewing process ensuring that everything that we merge into our master branch is up to our standards.

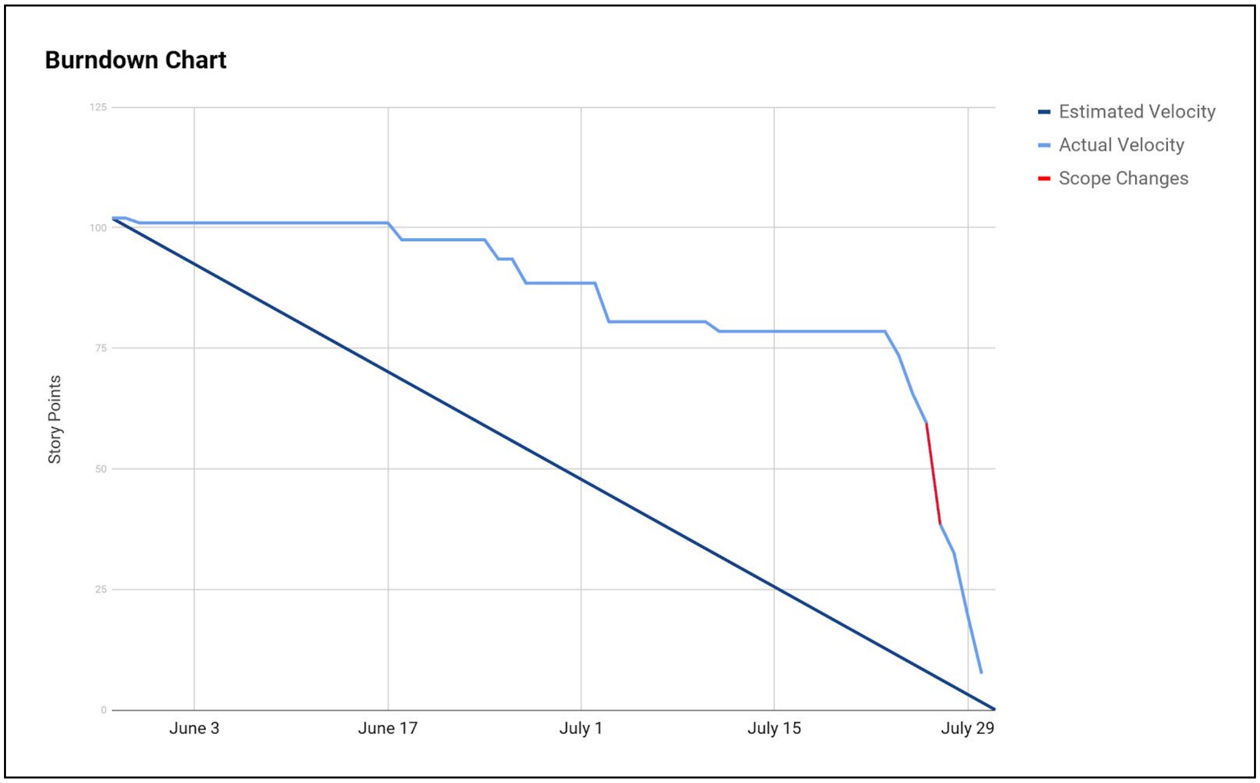


Figure 1: Burndown chart.

Lessons Learned

Being the second group taking on this project was advantageous for our team, as we had the opportunity to read and reflect on the previous group's report to see the lessons *they* had learned. Even with this help, our group still made mistakes and learned lessons along the way which we hope may help the next group ahead of us.

One lesson that we learned is that paired programming and collaboration can help in various ways. First, it helps share knowledge between group members. If one member has knowledge and experience in PHP, for example, it may be valuable to have this member pair up with someone who does not have as much experience. Second, it allows for more rapid progress where someone may be struggling with an issue that another member knows the solution to. Third, collaborating with team members helps build team morale and helps build positive relationships throughout the group.

Another lesson that we have learned is to keep testing in mind throughout the execution phase. Despite including in our plan that we would continuously run the pre-existing tests for the web portal, for most of the term we forgot to stick to this promise in favour of more development. We believe that we made up for this by using a thorough merge request review process, although we would feel even more confident had we implemented standardized testing. The issues with encountered with testing revolved around constant, sweeping changes to the code which actually leads into another lesson, communication on organizational changes. Occasionally, a new merge would result in the tasks of others breaking or becoming buggy because the changes were not discussed with everyone until they were implemented. Constant communication between developers with overlapping tasks ensures that everyone understands how the code works and how it may or may not need to change. (see recommendation section for more on testing).

While a group of nine can be difficult to manage and plan meeting times for, we believe that the times that we did meet outside of scheduled class time were critical to our success. We regret that we did not find time for more of these meetings throughout the term. Given the chance to redo the term, we would find more time to meet even if that means only meeting portions of the entire group at a time. This would facilitate more interaction which likely would have resulted in more progress.

Recommendations for Future Groups

From our experience working on this project this term, we have the following recommendations for the future groups who take on this project:

- Look into adding database logging in order to track which user makes what changes. This would be particularly important when an AIO closes a case.
- Investigate more sophisticated testing tools (need to test that UI interactions have expected results in our database)
- Continue to use merge requests on GitLab to do manual testing on every code change before merging it into master
- Depending on the experience that group members have, it may be wise to begin development using paired programming if the group members have little experience with the technologies this project uses
- There is a large ramp up to learning how the process should work, and should be accounted for in the project timeframe
- Every developer should follow the instructions which outline how to set up a local development environment. There are some details within this report, but an entire instruction document has been written and will be available in the files which are passed on to the next group.

Several modifications were made to the default php.ini file that runs on the peso.cs.dal.ca server. These modifications were made in response to the client’s specifications, which were that users should be able to upload a larger number of evidence files at once, and that users should be allowed to upload evidence files with bigger file sizes. The CS help desk was contacted to make the necessary changes to the php.ini file for the AIO web portal. The following variables were updated:

- post_max_size = 100M
- memory_limit = 128M
- upload_max_filesize = 100M
- max_file_uploads = 50

To ensure the changes had been made, a current member of the team briefly modified the AIO web portal home page to print out the values of the variables listed above. This was done by adding the php statement **echo ini_get('var_name')**, where **var_name** is replaced with the desired variable name from the php.ini file.

Before development on the AIO project resumes, it is strongly recommended that the developers update their local php.ini file in MAMP accordingly. Not doing so may result in developers seeing unexpected error messages while adding new features to the web portal.

Ongoing Support Required and Duration

The ongoing support provided by this team is minimal. In order to continue serving the system as-is at the end of the project, the CS Help Desk needs to continue maintenance on the server and database. Since the previous group informed the CS Help Desk that our database and server need to remain online until a request for termination is received, no action is currently required.

The contents of the server at the end of this project will exactly match that of the contents in the master branch of the GitLab repository. Thus, if anything happens to the server in the meantime before the next group begins work on the project, the code will still be accessible to be reloaded onto a server.

System Documentation

Front End

The front end web portal can be viewed by going to **projects.cs.dal.ca/aio**. This will show the most up to date version of the website that has been moved onto the server. The server being used for this project is **peso.cs.dal.ca** and can be entered via SSH using an application like PuTTY or similar, with the **username: aio** and the **password: ge7ochooCae7**. Upon logging in there will be two folders, **html** and **website**. The **html** folder is a soft link to the project directory **/local/data/websites/projects.cs.dal.ca/public/aio**, which contains all of the files and directories for the project.

FileZilla or a similar program will have to be used to add the most up to date files to the server. It is recommended that developers keep the previous version of all the files in a separate directory in case something goes wrong.

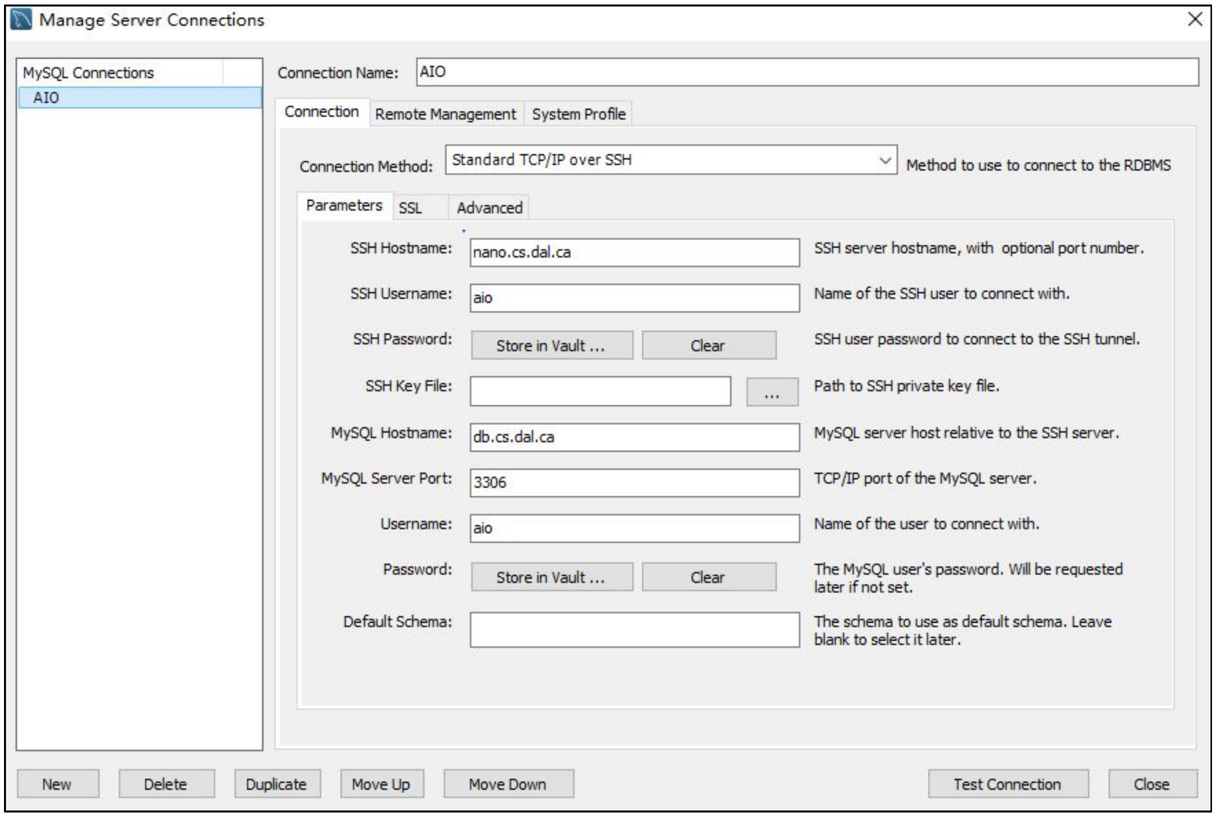
If a new file is being added directly to the server make sure to remove the group write permission, which can be done using the command **chmod g-w filename.php**. If you are in the **html** directory, you can perform the same action on all the files in the project using **chmod -R g-w .** (the period is part of the command).

The server will have to be used to test some things. Sending email, in particular, was something that needed to be tested on the server since most local hosting applications like MAMP do not support SMTP by default.

Back End

The database for this project is hosted on **db.cs.dal.ca**. It can be accessed by visiting **https://myadmin.cs.dal.ca/** and using the **username: aio** and the **password: ge7ochooCae7**.

The database can also be accessed through MySQL workbench using the information shown in figure 2 below. The SSH hostname should be entered as **peso.cs.dal.ca**, not **nano.cs.dal.ca**.



Connection Name: AIO

Connection Method: Standard TCP/IP over SSH — Method to use to connect to the RDBMS

SSH Hostname:	nano.cs.dal.ca	SSH server hostname, with optional port number.
SSH Username:	aio	Name of the SSH user to connect with.
SSH Password:	Store in Vault ... / Clear	SSH user password to connect to the SSH tunnel.
SSH Key File:		Path to SSH private key file.
MySQL Hostname:	db.cs.dal.ca	MySQL server host relative to the SSH server.
MySQL Server Port:	3306	TCP/IP port of the MySQL server.
Username:	aio	Name of the user to connect with.
Password:	Store in Vault ... / Clear	The MySQL user's password. Will be requested later if not set.
Default Schema:		The schema to use as default schema. Leave blank to select it later.

Figure 2: Access database from MySQL workbench.

Local Development

For local development the database will have to be exported from the server and imported on your own machine. To do this you must navigate to **https://myadmin.cs.dal.ca/** and log in. Once this is done click **Export**, leaving everything as is, and clicking **Go**. This will save a local copy of the database on you computer in a file which should be called **db_cs_dal_ca.sql**.

For local development MAMP will also have to be installed for local hosting. Once this is done go to **http://localhost/MAMP/** and click on the **MySQL** header. Then select **PhpMyAdmin**. Once PhpMyAdmin is open locally click **Import** and select the **db_cs_dal_ca.sql** file downloaded in the last step, leave everything as is, and click **Go**. The database should now run on the MAMP local MySQL server.

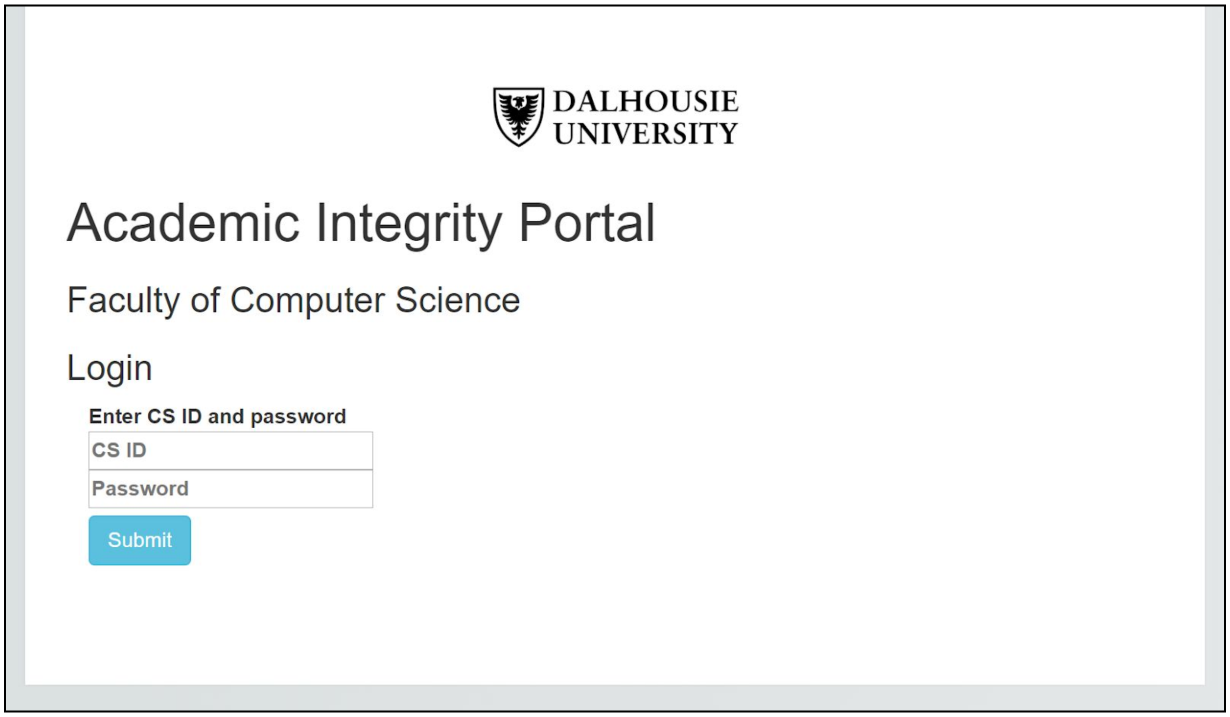
Now to locally host the web portal itself, clone the remote repository on GitLab into the **../MAMP/htdocs** folder. The website will now be running on the MAMP Apache server. This location is recommended for the local git repository so that files do not have to be moved around every time a change is made. This immensely speeds up the development and testing process.

Testing

Test cases can be run from the **run_functional_tests.sh** file within the **..MAMP/tests** folder. The tests do not currently autofill the username/password sections, and the way in which roles are handled have changed completely. Therefore, the only real functionality these test cases have is opening the main page on localhost and setting values to be used for different roles. They will need to be almost completely rewritten to allow for functionality with the new roles, buttons, and dropdowns. It is recommended that test case authors understand what will be necessary for the test cases before reorganizing code, as the tests will move further away from functional as more features are added and overhauled. The tests also largely test for HTML text on each page, which is not recommended. Database testing should be written as a priority.

User Manual

Login Page
The login page, seen in figure 3, is where the user lands when they navigate to the web portal. Here a user can log in with their CS ID, which will be validated in the back-end to detect their authorized roles in the portal.


DALHOUSIE UNIVERSITY

Academic Integrity Portal

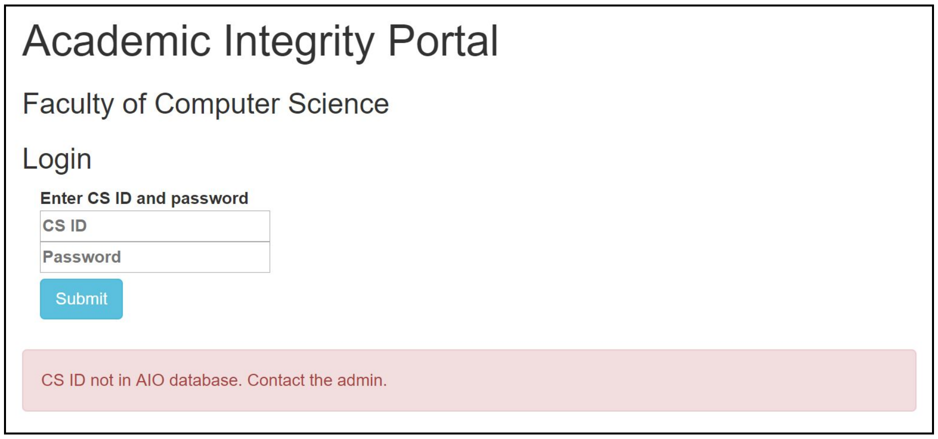
Faculty of Computer Science

Login

Enter CS ID and password

Figure 3: The login page

If the user's CS ID has not been added to the database of users for this web portal, they will not be able to login and will see the error message displayed in figure 4.


Academic Integrity Portal

Faculty of Computer Science

Login

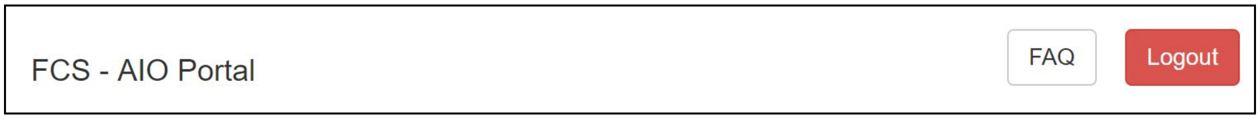
Enter CS ID and password

CS ID not in AIO database. Contact the admin.

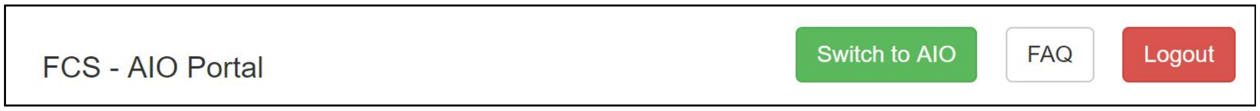
Figure 4: The login page, with an error

Upon successful login, the user will be logged in as their role with the most authority by default. They will also be navigated to the Active Cases page. Depending on the number of roles the user has, the navigation bar at the top of the page will include different options for switching roles next to the default FAQ and Logout buttons.

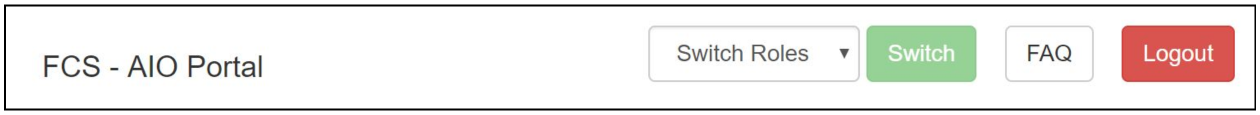
If the user has one role, no role switching options will appear:

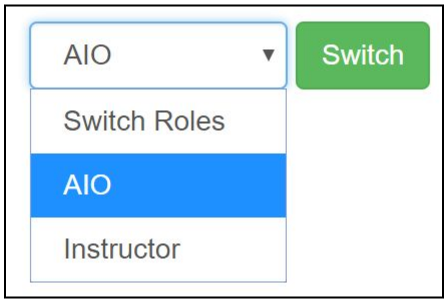


If the user has two roles, a button to change role will appear:



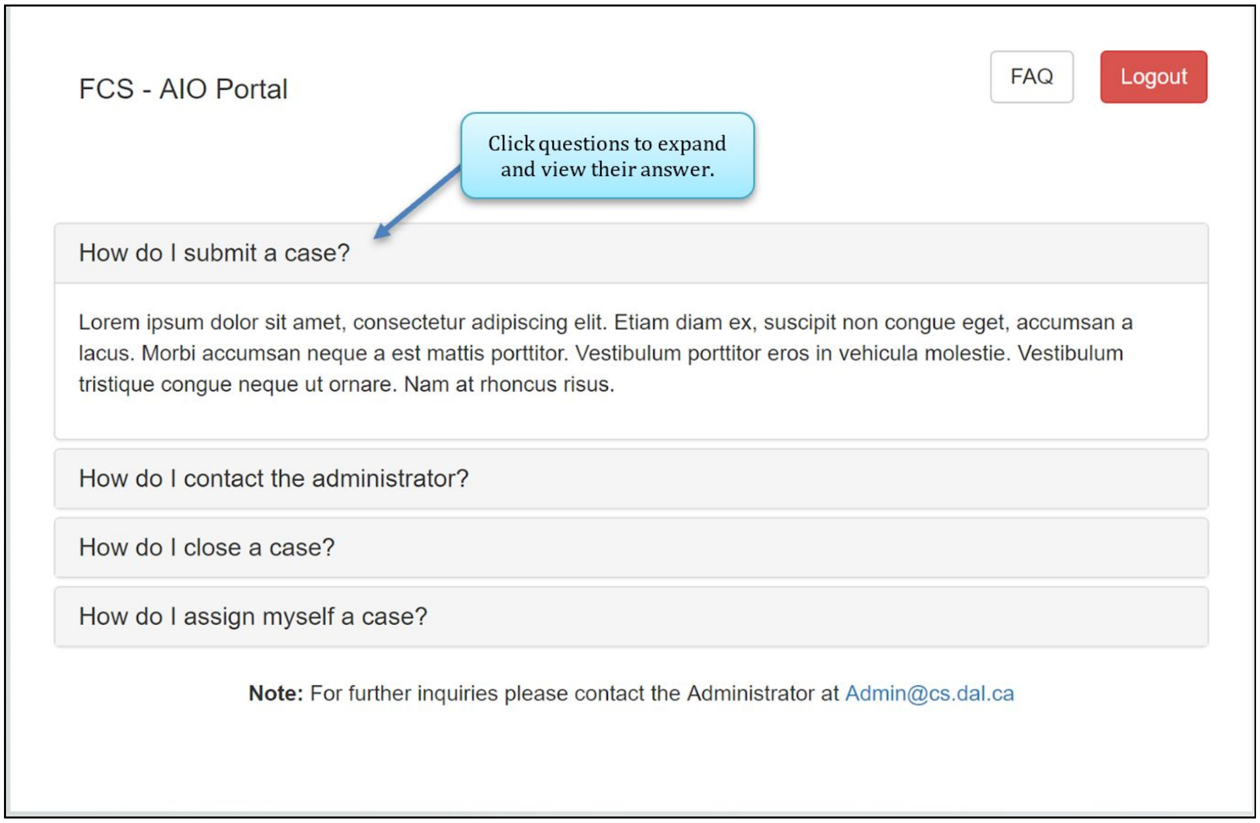
If the user has three roles, a drop-down menu will appear:





FAQ Page

The FAQ page can be accessed from any user role. The Administrator has answers to common questions to help AIOs and Professors in the AIO Web Portal process.



FCS - AIO Portal

How do I submit a case?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam diam ex, suscipit non congue eget, accumsan a lacus. Morbi accumsan neque a est mattis porttitor. Vestibulum porttitor eros in vehicula molestie. Vestibulum tristique congue neque ut ornare. Nam at rhoncus risus.

How do I contact the administrator?

How do I close a case?

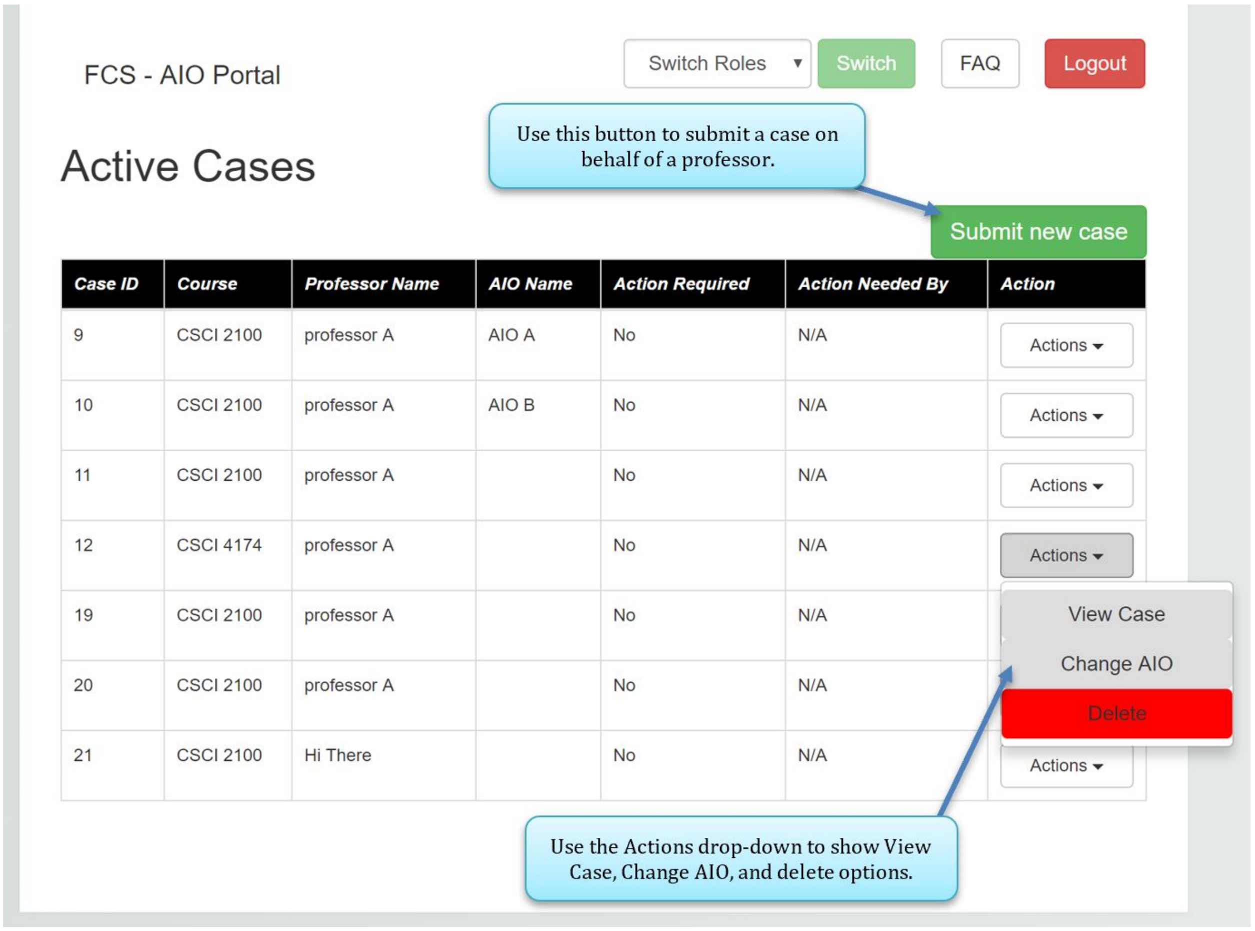
How do I assign myself a case?

Note: For further inquiries please contact the Administrator at Admin@cs.dal.ca

The remainder of this manual is split into sections by each role: Administrator, AIO, and Professor.

Administrator Manual

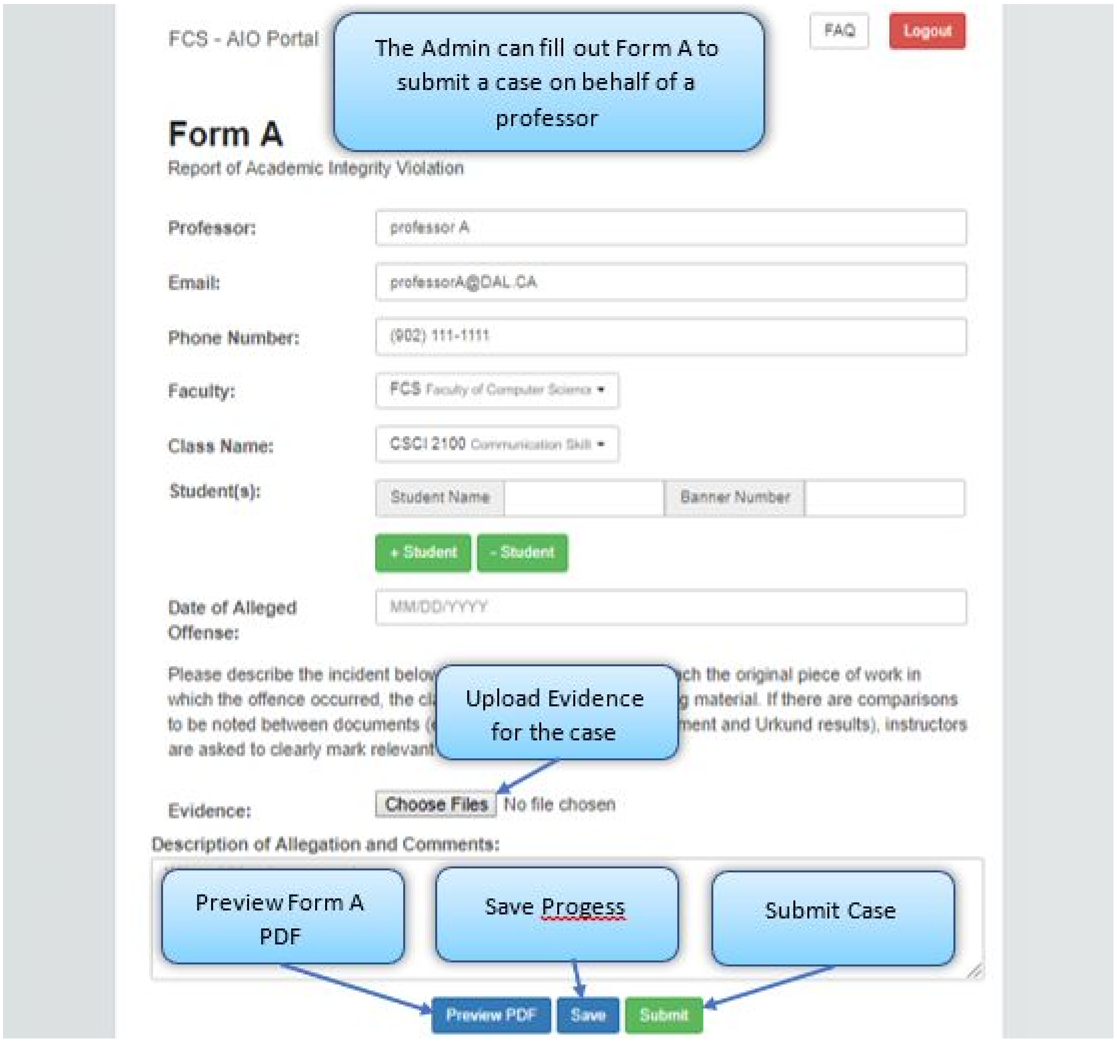
Active Cases Page


FCS - AIO Portal
Switch Roles
Switch
FAQ
Logout
Use this button to submit a case on behalf of a professor.
Submit new case
View Case
Change AIO
Delete
Use the Actions drop-down to show View Case, Change AIO, and delete options.

Active Cases

Case ID	Course	Professor Name	AIO Name	Action Required	Action Needed By	Action
9	CSCI 2100	professor A	AIO A	No	N/A	Actions
10	CSCI 2100	professor A	AIO B	No	N/A	Actions
11	CSCI 2100	professor A		No	N/A	Actions
12	CSCI 4174	professor A		No	N/A	Actions
19	CSCI 2100	professor A		No	N/A	
20	CSCI 2100	professor A		No	N/A	
21	CSCI 2100	Hi There		No	N/A	Actions

Submit New Case



FCS - AIO Portal

FAQ Logout

Form A

Report of Academic Integrity Violation

Professor:	professor A
Email:	professorA@DAL.CA
Phone Number:	(902) 111-1111
Faculty:	FCS Faculty of Computer Science
Class Name:	CSCI 2100 Communication Skill
Student(s):	Student Name / Banner Number
Date of Alleged Offense:	MM/DD/YYYY

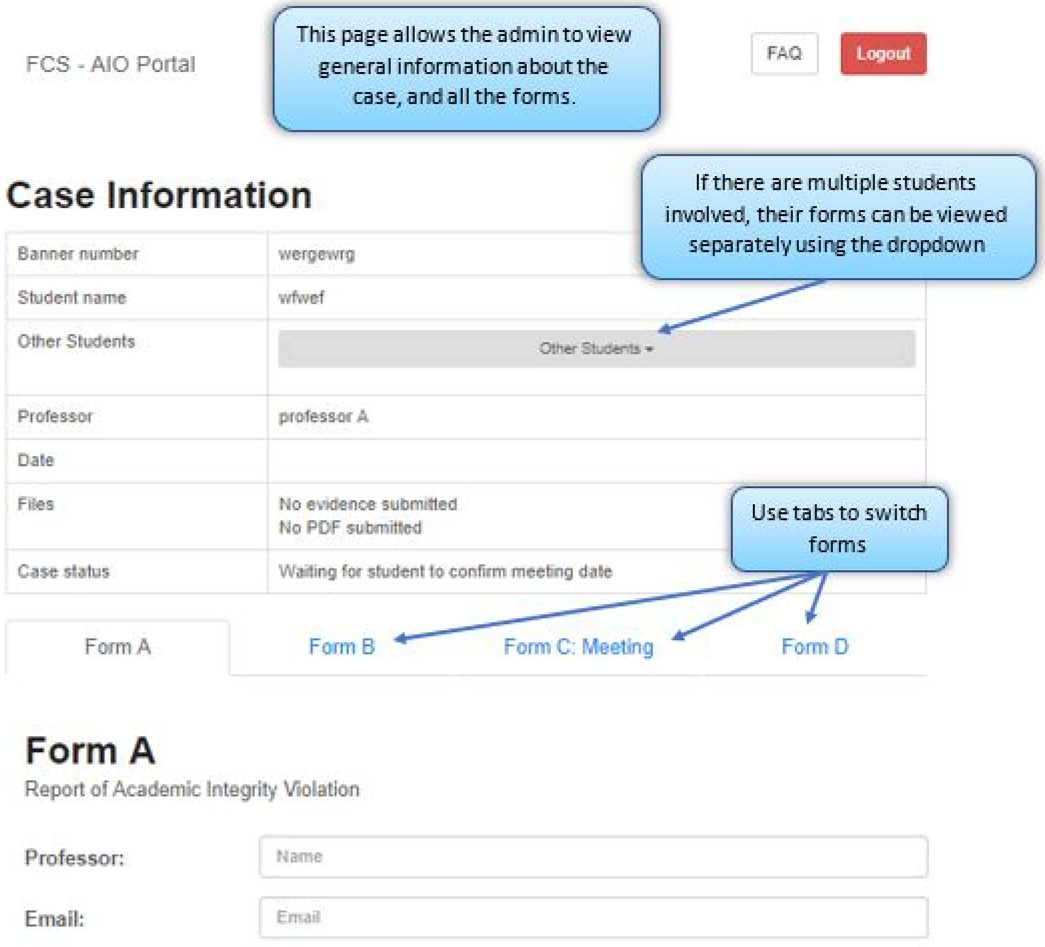
Please describe the incident below ... ch the original piece of work in which the offence occurred, the cl ... g material. If there are comparisons to be noted between documents (... ment and Urkund results), instructors are asked to clearly mark relevant

Evidence: Choose Files No file chosen

Description of Allegation and Comments:

The Admin can fill out Form A to submit a case on behalf of a professor
+ Student
- Student
Upload Evidence for the case
Preview Form A PDF
Save Progess
Submit Case
Preview PDF
Save
Submit

View Case


FCS - AIO Portal
FAQ
Logout
This page allows the admin to view general information about the case, and all the forms.
If there are multiple students involved, their forms can be viewed separately using the dropdown
Use tabs to switch forms
Other Students
Form A
Form B
Form C: Meeting
Form D
Name
Email

Case Information

Banner number	wergewrg
Student name	wfwef
Other Students	Other Students ▾
Professor	professor A
Date	
Files	No evidence submitted No PDF submitted
Case status	Waiting for student to confirm meeting date

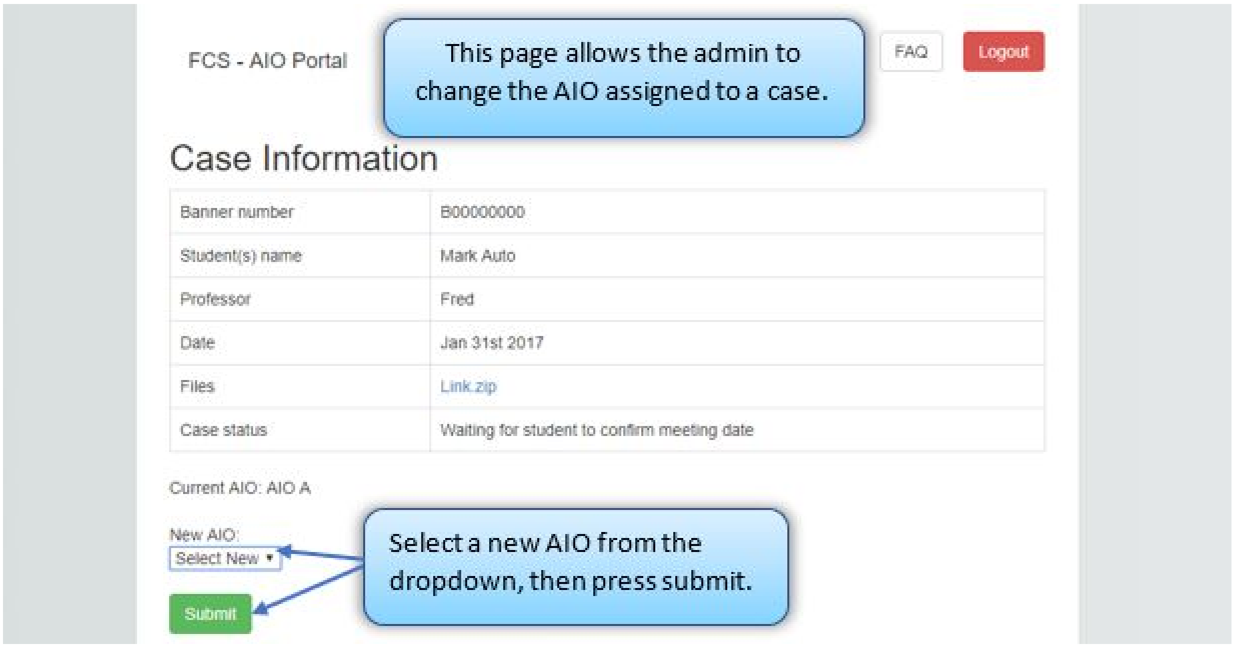
Form A

Report of Academic Integrity Violation

Professor:

Email:

Change AIO


FCS - AIO Portal

FAQ
Logout
This page allows the admin to change the AIO assigned to a case.
Select a new AIO from the dropdown, then press submit.

Case Information

Banner number	B00000000
Student(s) name	Mark Auto
Professor	Fred
Date	Jan 31st 2017
Files	Link.zip
Case status	Waiting for student to confirm meeting date

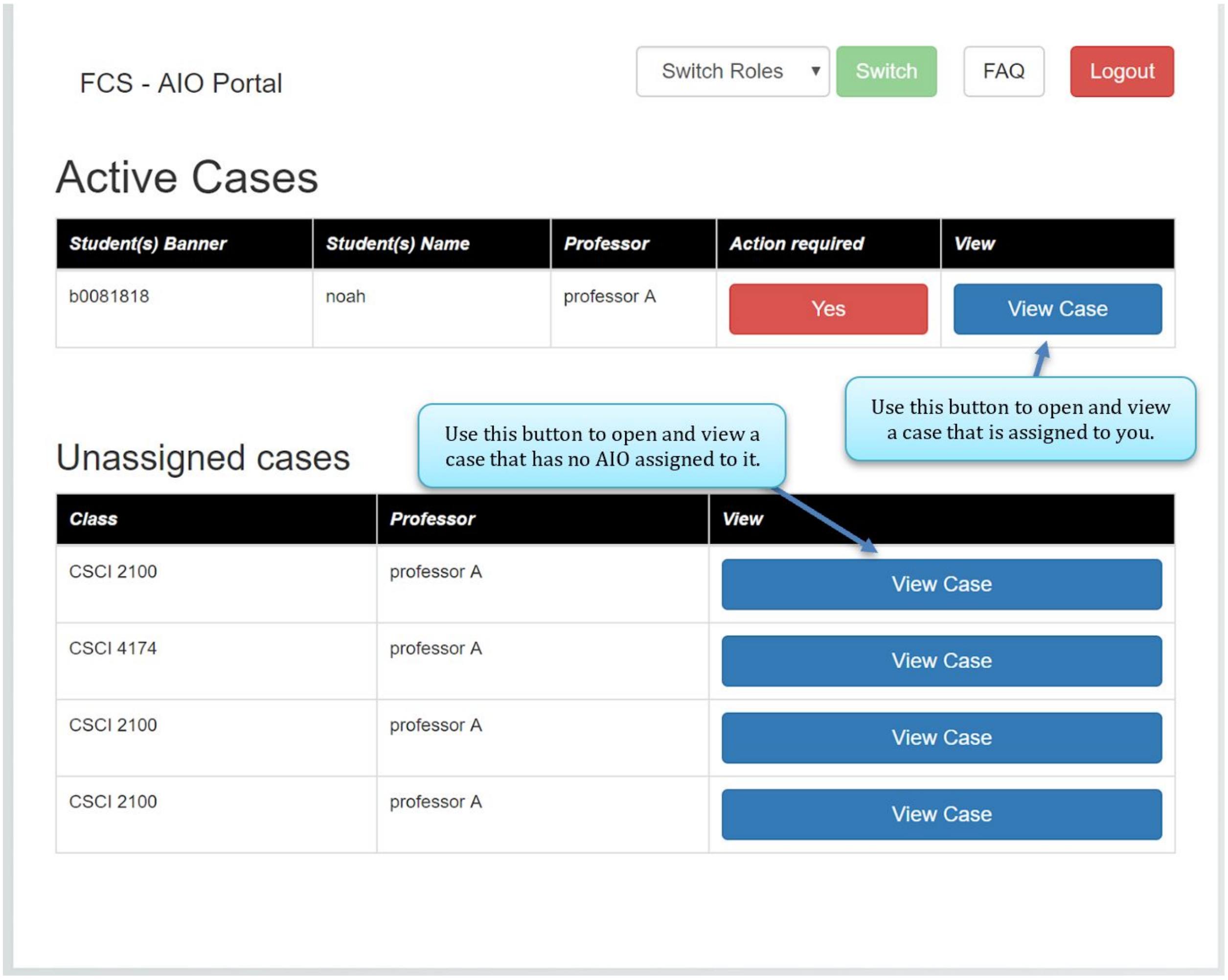
Current AIO: AIO A

New AIO: Select New

AIO Manual

Active Cases Page

Upon login as an AIO, you will be greeted by the AIO Active cases page. On this page you will see the two main tables of importance.



FCS - AIO Portal

Active Cases

Student(s) Banner	Student(s) Name	Professor	Action required	View
b0081818	noah	professor A	Yes	View Case

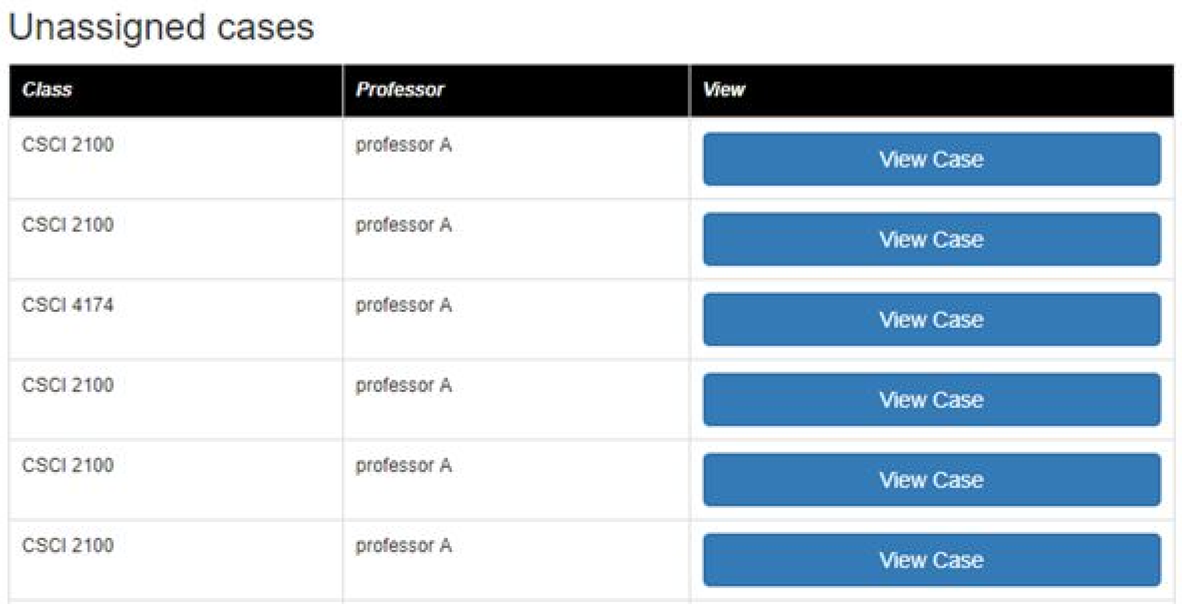
Use this button to open and view a case that is assigned to you.

Unassigned cases

Use this button to open and view a case that has no AIO assigned to it.

Class	Professor	View
CSCI 2100	professor A	View Case
CSCI 4174	professor A	View Case
CSCI 2100	professor A	View Case
CSCI 2100	professor A	View Case

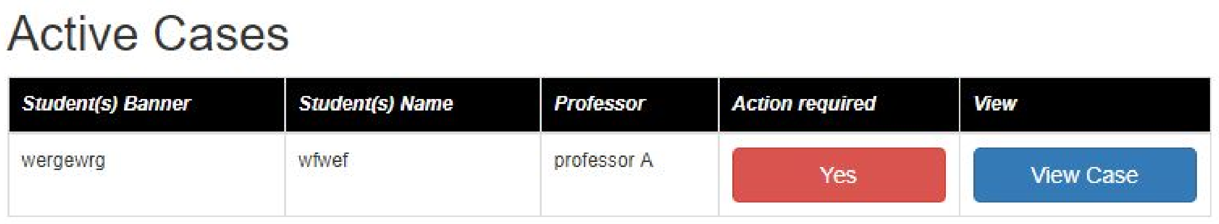
View Case (Unassigned)
The unassigned cases table shows which cases do not yet have an active AIO working on them**.**


Unassigned cases

Class	Professor	View
CSCI 2100	professor A	View Case
CSCI 2100	professor A	View Case
CSCI 4174	professor A	View Case
CSCI 2100	professor A	View Case
CSCI 2100	professor A	View Case
CSCI 2100	professor A	View Case

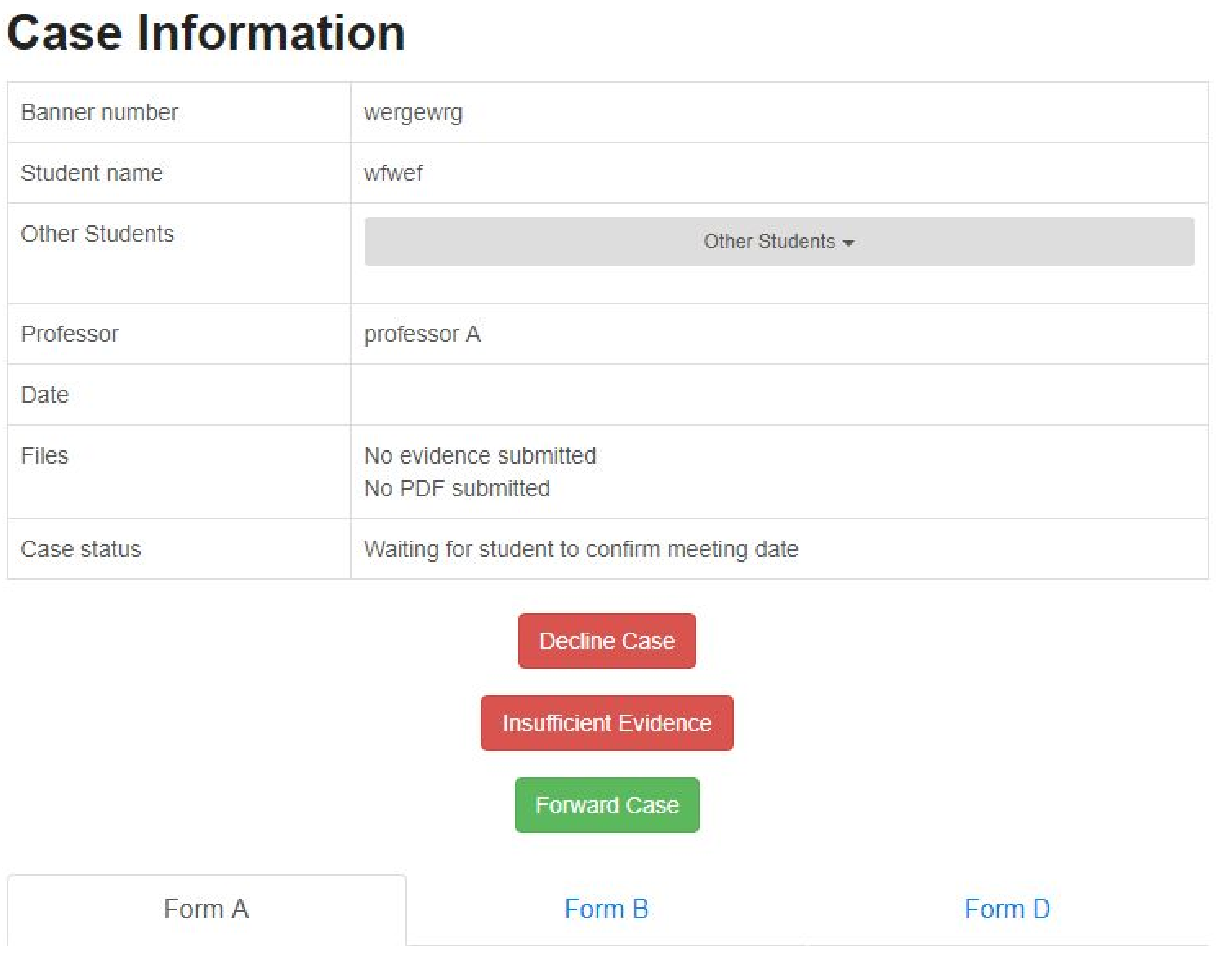
View Case button: will take you to the case information screen where you can view a detailed information page about the given case. More on that on the next page.

View Case (Assigned)
The active cases table will show you all cases that are currently assigned to you, and will indicate if action is required within those cases.


Active Cases

Student(s) Banner	Student(s) Name	Professor	Action required	View
wergewrg	wfwef	professor A	Yes	View Case

Case Information (Assigned case)
When viewing a case that is assigned to you the case information screen will display 3 new buttons.


Case Information

Banner number	wergewrg
Student name	wfwef
Other Students	Other Students
Professor	professor A
Date	
Files	No evidence submitted No PDF submitted
Case status	Waiting for student to confirm meeting date

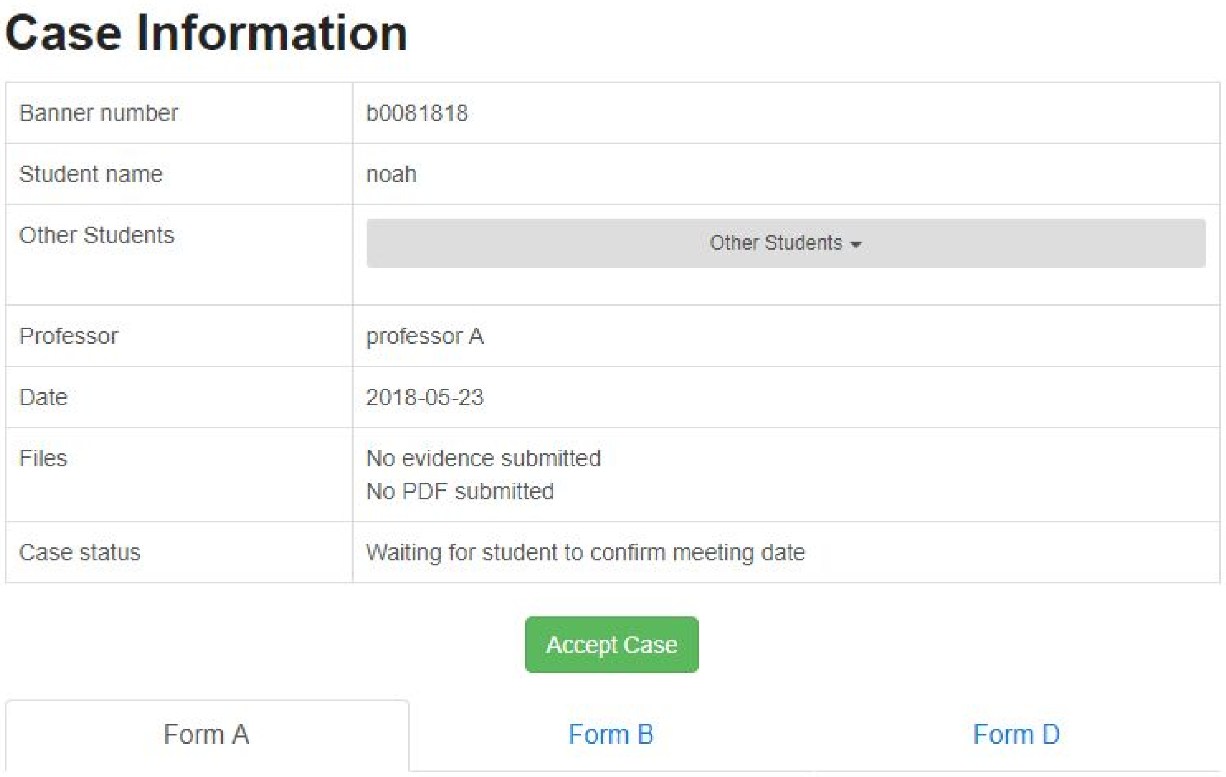
Decline Case button: Gives the AIO the ability to decline the current case, and it will be added back to the list of unassigned cases.

Insufficient Evidence button: If the AIO decides the case doesn't have enough evidence to proceed the insufficient evidence button will remove the case from the active cases table. And send an email to the professor explaining that the case did not contain enough evidence.

Close Case Button *(Not shown, replaces the Insufficient Evidence button)***:** If a verdict has been delivered, the AIO can close the case and either remove from the database or archive it using this button.

Forward Case button: When the accused party has committed multiple offences, the case is forwarded to the senate based on the AIO's discretion. Hitting the Forward Case button, will zip all the evidence and forms for a given case, and send that information to the senate.

Case Information (Unassigned case)
When viewing a case that is not currently assigned to any AIO's the option will become available to claim the case.



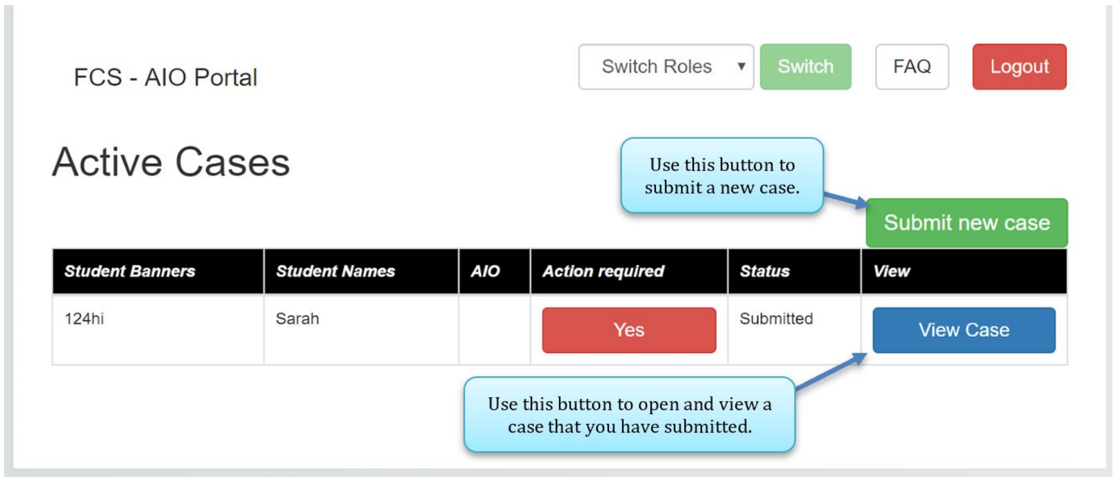
Case Information

Banner number	b0081818
Student name	noah
Other Students	Other Students ▾
Professor	professor A
Date	2018-05-23
Files	No evidence submitted No PDF submitted
Case status	Waiting for student to confirm meeting date

Accept case button: When Hitting the Accept case button, the currently logged in AIO will be assigned to the case, and it will be moved to their active cases table.

Professor Manual

Active Cases Page



FCS - AIO Portal

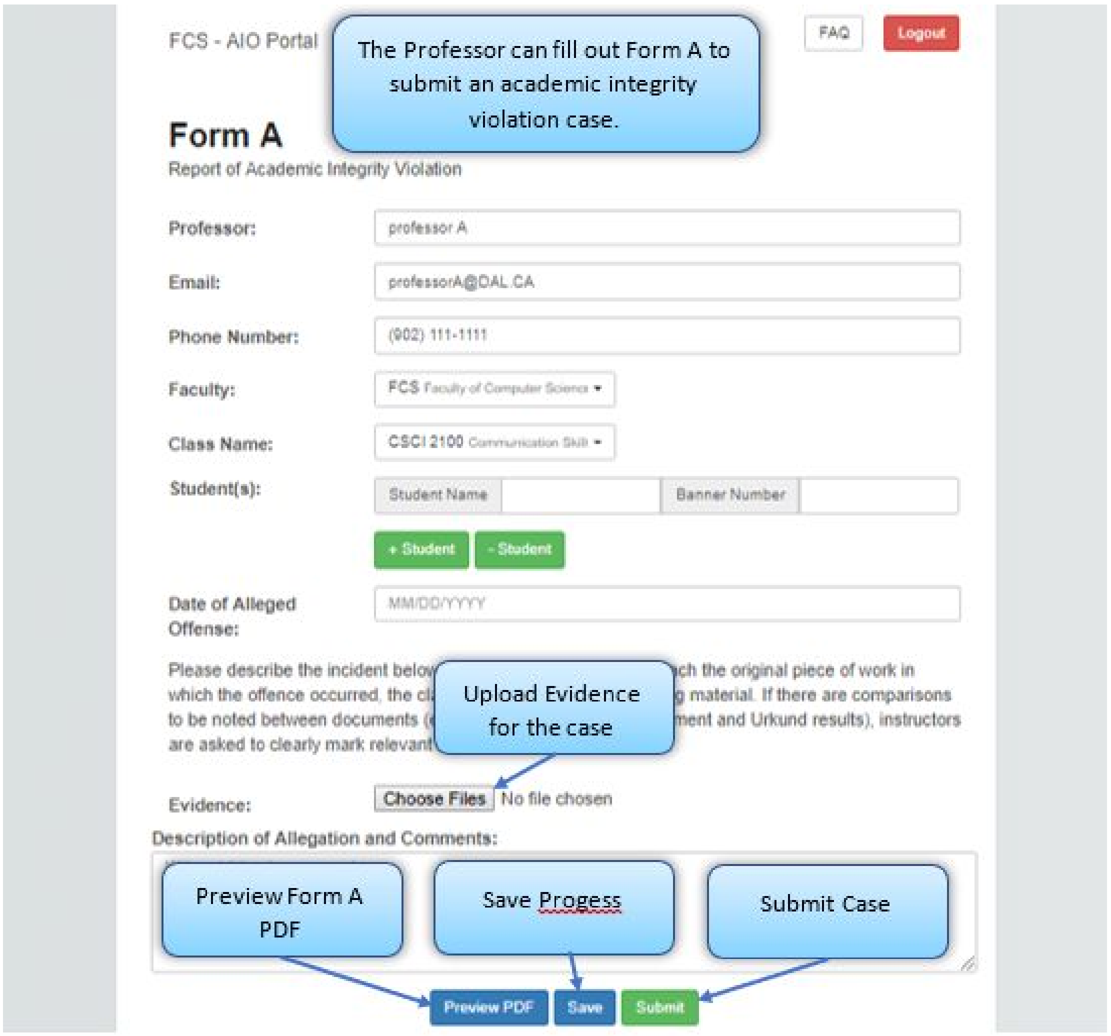
Active Cases

Student Banners	Student Names	AIO	Action required	Status	View
124hi	Sarah		Yes	Submitted	View Case

Use this button to submit a new case.

Use this button to open and view a case that you have submitted.

Submit New Case



FCS - AIO Portal

The Professor can fill out Form A to submit an academic integrity violation case.

Form A

Report of Academic Integrity Violation

Professor:	professor A
Email:	professorA@DAL.CA
Phone Number:	(902) 111-1111
Faculty:	FCS Faculty of Computer Science
Class Name:	CSCI 2100 Communication Skill
Student(s):	Student Name / Banner Number
Date of Alleged Offense:	MM/DD/YYYY

Please describe the incident belo… …ch the original piece of work in which the offence occurred, the cl… …g material. If there are comparisons to be noted between documents (… …ment and Urkund results), instructors are asked to clearly mark relevant…

Upload Evidence for the case

Evidence: Choose Files No file chosen

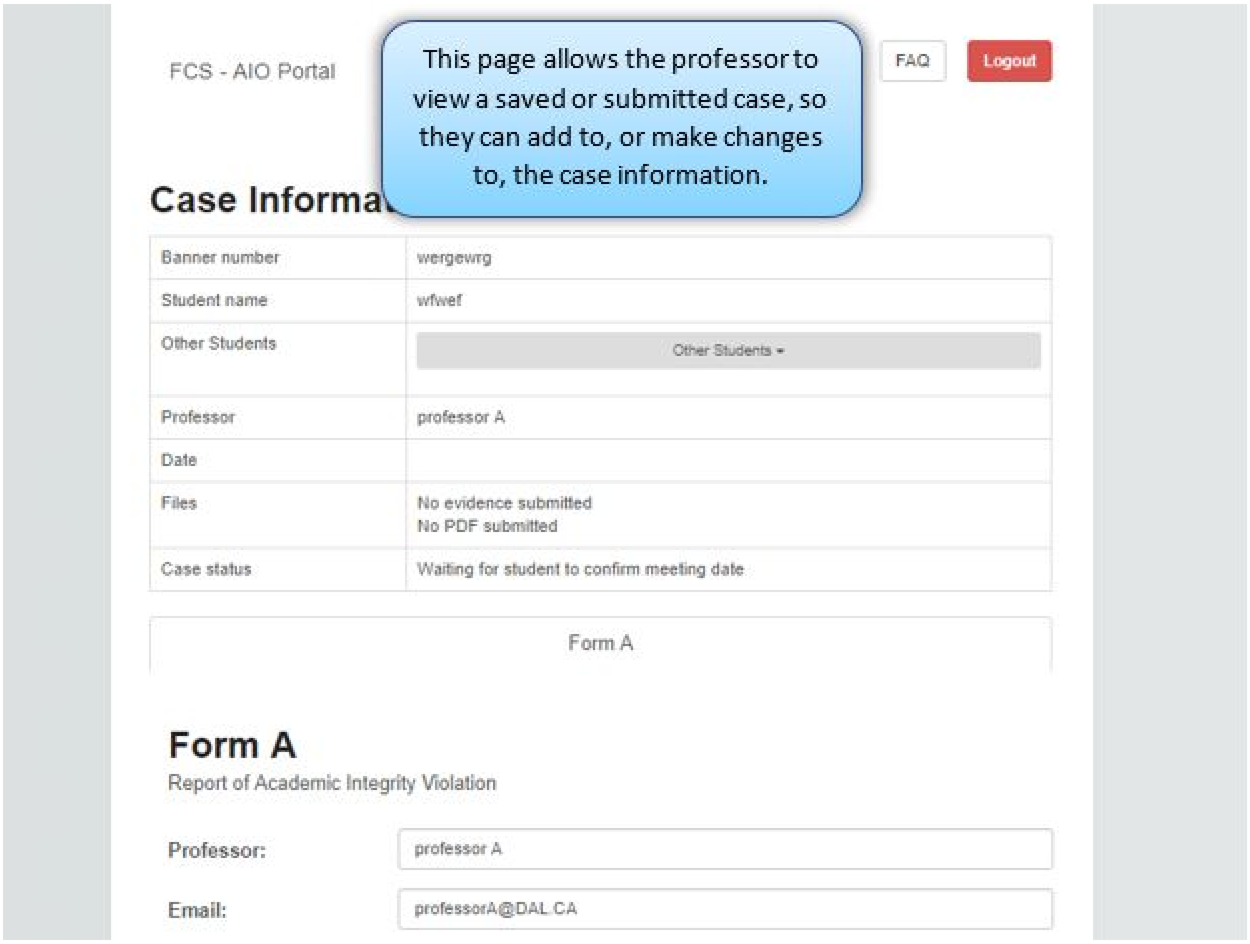
Description of Allegation and Comments:

Preview Form A PDF

Save Progess

Submit Case

View Case



This page allows the professor to view a saved or submitted case, so they can add to, or make changes to, the case information.

Case Informa

Banner number	wergewrg
Student name	wfwef
Other Students	Other Students ▾
Professor	professor A
Date	
Files	No evidence submitted No PDF submitted
Case status	Waiting for student to confirm meeting date

Form A

Form A

Report of Academic Integrity Violation

Professor: professor A

Email: professorA@DAL.CA

List of Training Material

The next group to take on development of the AIO portal should invest some time researching the following technologies and how they relate to this project:

- Apache
- Bootstrap
- CSS
- FileZilla
- HTML
- JavaScript
- LaTeX
- MAMP
- MySQL
- PHP
- PuTTy

Project Sign-off

By signing below, the client, team leader, and team members have all reviewed this report and agree to conclude this project with the progress that was made over the summer term of 2018. All parties have a full understanding of what has been done and what remains to be done by the future teams on this project.

Team Leader: ___________________________

Client: ___________________________

Date: ____________________________