**Coding Challenge System (CCS)**

**Runbook**

Document Category: RB

Version: 0.2.1

Status: Completed

Last Updated: 2022-04-25

Contents

[Contents 2](#_Toc99309543)

[1 Introduction 4](#_Toc99309544)

[1.1 Purpose 4](#_Toc99309545)

[1.2 Scope 4](#_Toc99309546)

[1.3 Target Audience 5](#_Toc99309547)

[1.4 Related documents 5](#_Toc99309548)

[1.5 Glossary 5](#_Toc99309549)

[2 Installation and Setup 6](#_Toc99309550)

[2.1 Installation 6](#_Toc99309551)

[**2.1.1** Pre-requisites 6](#_Toc99309552)

[**2.1.2** Installation Steps 8](#_Toc99309553)

[2.2 Setup & Configurations 20](#_Toc99309554)

[**2.2.1** Dependent software and service configuration 20](#_Toc99309555)

[**2.2.2** Deployment 21](#_Toc99309556)

[3 Operations Procedure 22](#_Toc99309557)

[3.1 Basic Operations 22](#_Toc99309558)

[**3.1.1** Start, Stopping and Restarting the Application 22](#_Toc99309559)

[3.2 Basic Sanity Test 22](#_Toc99309560)

[3.3 Authorization Errors 23](#_Toc99309561)

[**3.3.1** Adding a User in AMS 23](#_Toc99309562)

[**3.3.2** Updating a User in AMS 26](#_Toc99309563)

[3.4 Monitoring 29](#_Toc99309564)

[**3.4.1** On-Going Monitoring 29](#_Toc99309565)

[**3.4.2** Reactive steps 29](#_Toc99309566)

[**3.4.3** Escalation Steps 29](#_Toc99309567)

[4 Appendix 30](#_Toc99309568)

[4.1 Database Scripts 30](#_Toc99309569)

[4.2 External Documents 30](#_Toc99309570)

[5 Approvals 31](#_Toc99309571)

[6 History 32](#_Toc99309572)

**Table of Figures**

[Figure 1: Deployment Diagram of CCS 5](https://cegepheritageqcca-my.sharepoint.com/personal/1826228_cegep-heritage_qc_ca/Documents/CCS_Runbook.docx#_Toc101774080)

[Figures 2: Verifying installed NuGet packages 9](#_Toc101774081)

[Figure 3 Verifying installed NuGet packages 10](#_Toc101774082)

[Figure 4 Application Management System (AMS) 11](#_Toc101774083)

[Figure 5 How to verify if CCS is already in the added to AMS 11](#_Toc101774084)

[Figure 6 How to verify if CCS is already in the added to AMS 12](#_Toc101774085)

[Figure 7 Checking if the ST (student) role has CCS in it 12](#_Toc101774086)

[Figure 8 Checking if the TE (teacher) role has CCS in it 13](#_Toc101774087)

[Figure 9: Adding a TE (teacher) role 13](#_Toc101774088)

[Figure 10: Editing the TE (teacher) role 13](#_Toc101774089)

[Figure 11: Adding the CCS application to a role 14](#_Toc101774090)

[Figure 12: Confirmation message that appears after successfully adding CCS to a role 15](#_Toc101774091)

[Figure 13: How to install CCS 16](#_Toc101774092)

[Figure 14: How to install CCS 17](#_Toc101774093)

[Figure 15: How to install CCS 18](#_Toc101774094)

[Figure 16: How to install CCS 18](#_Toc101774095)

[Figure 17: SSMS Login Screen 19](#_Toc101774096)

[Figure 18: Success message shown after running CCS\_Script.sql 19](#_Toc101774097)

[Figure 19: Success messages shown after running CCS\_Inserts.sql 19](#_Toc101774098)

[Figure 20: Where to access the Solution file 20](#_Toc101774099)

[Figure 21: How to access the ConnectedService.json file 20](#_Toc101774100)

[Figure 22: Login service link inside ConnectedService.json 20](#_Toc101774101)

[Figure 23: Microsoft WCF Web Service Reference Provider 21](#_Toc101774102)

[Figure 24: Configuring the Login Service 21](#_Toc101774103)

[Figure 25: Database connection string inside appsettings.json 21](#_Toc101774104)

[Figure 26: Folder structure of CCS codebase (minimized) 22](#_Toc101774105)

[Figure 27: Folder structure of CCS codebase (expanded snapshots) 23](#_Toc101774106)

[Figure 28: UXDEV Dummy Client 24](#_Toc101774107)

[Figure 29: IIS window for CSDEV 25](#_Toc101774108)

[Figure 30: Locating the application pool for CCS 26](#_Toc101774109)

[Figure 31: Window that appears upon right-clicking on the CCSAppPool 26](#_Toc101774110)

[Figure 32: Teacher home page 27](#_Toc101774111)

[Figure 33: The student home page 28](#_Toc101774112)

[Figure 34: Logging in as an admin in AMS 29](#_Toc101774113)

[Figure 35: Accessing “Manage Users” view from AMS Home page 29](#_Toc101774114)

[Figure 36: Adding a new user through “Manage Users” view 30](#_Toc101774115)

[Figure 37: Adding a new user 31](#_Toc101774116)

[Figure 38: Logging in as an admin in AMS 31](#_Toc101774117)

[Figure 39: Accessing “Manage Users” view from AMS Home page 32](#_Toc101774118)

[Figure 40: How to search a user and edit their information 32](#_Toc101774119)

[Figure 41: Updating user information 33](#_Toc101774120)

[Figure 42: Searching for a user 33](#_Toc101774121)

[Figure 43: Locating a user by their username 33](#_Toc101774122)

**Tables**

[Table 1‑1 Related Documents 6](#_Toc101774311)

[Table 1‑2 Glossary 6](#_Toc101774312)

# Introduction

## Purpose

The purpose of this document is to provide installation and operational guidelines for the Coding Challenge System and will specifically cover the following topics:

* Installation
* Environment preparation
* Install and post-install verification
* Configuration & setup
* Operations Procedures
* Startup/shutdown/restart
* Anything that the operators might need to do to support the users
* Operational Monitoring
* Basic troubleshooting & Escalation

## Scope

Below is a deployment diagram for the system. It is important to note that the web and database server need to meet the minimum requirements listed in section *2.1 Prerequisites and Dependencies*.

Diagram

Description automatically generated

Figure 1: Deployment Diagram of CCS

This document is specific to the CCS application, and for more detailed information on the system’s architecture, please refer to the System Design document.

In addition, this runbook will also refer to the Application Management System (AMS) and how to set up adding new users.

## Target Audience

This document is intended for use as a reference by infrastructure engineers and system operators responsible for the installation and subsequent operational support of the CCS Web Application and related systems.

## Related documents

These documents contain information related to the information in this document. Refer to [Section 4.2](#_External_Documents) for more details.

| Document Short Name Reference | Document Title |
| --- | --- |
| System | System Documentation |
| User | User Documentation |
| Deployment | Deployment Documentation |

Table 1‑1 Related Documents

## Glossary

| Term/Acronym | Description |
| --- | --- |
| AMS | Application Management System |
| CCS | Coding Challenge System |
| SSMS | SQL Server Management Studio |

Table 1‑2 Glossary

# Installation and Setup

## Installation

### Pre-requisites

Before installing CCS, please make sure the following environment requirements are met.

#### Hardware Requirements:

* Software and hardware architecture of 64 bits
* CPU 2.7GHz (2 processors)
* 4 GB of RAM
* 200 MB of disk space
* Network speed: 5-10 Mb/s

#### Software Requirements:

**NOTE**: Apart from the main “CCS” project, there is also a “CCS Test” project which contains test cases. Thus, they have different requirements.

**CCS**:

* OS: Windows 10 or later
* .NET Core version 3.1 (minimum)
* IIS Version 10
* SQL Server Management Studio 18+

**CCS Test**:

* OS: Windows 10 or later
* .NET Core version 3.1 (minimum)

**Both CCS and CCS Test**:

* Packages:

|  |  |
| --- | --- |
| **Package** | **Version** |
| HtmlSanitizer | 7.1.475 |
| Microsoft.AspNetCore.Authentication.Cookies | 2.2.0 |
| Microsoft.AspNetCore.Session | 2.2.0 |
| Microsoft.Data.Sqlite | 6.0.2 |
| Microsoft.EntityFrameworkCore | 3.1.0 |
| Microsoft.EntityFrameworkCore.Design | 3.1.0 |
| Microsoft.EntityFramework.Sqlite | 3.1.0 |
| Microsoft.EntityFramework.SqlServer | 3.1.0 |
| Microsoft.EntityFrameworkCore.Tools | 3.1.0 |
| Microsoft.VisualStudio.Web.CodeGeneration.Design | 3.1.5 |
| Moq | 4.16.1 |
| System.ServiceModel.Duplex | 4.8.1 |
| System.ServiceModel.Http | 4.8.1 |
| System.ServiceModel.NetTcp | 4.8.1 |
| System.ServiceModel.Security | 4.8.1 |

Table 2‑1 Package Requirements for CCS and CCS Test

* Libraries:

|  |  |  |
| --- | --- | --- |
| **Library** | **Version** | **Link** |
| jQuery | 3.5.1 | Should be downloaded locally |
| Bootstrap | 4.3.1 | Should be downloaded locally |
| CodeMirror | 5.64.0 | Should be downloaded locally |
| Popper | 1.12.9 | <https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js> |

Table 2‑2 Library Requirements for CCS

An Excel spreadsheet of all the frameworks and packages is also available and can be accessed from “[/Documentation/Audits](https://csazure.cegep-heritage.qc.ca:8080/F2021-DevProject/_git/Code%20Challenge?path=%2FDocumentation%2FAudits&version=GBmaster)”.

### Installation Steps

#### Pre-install verification steps:

Please make sure the packages and libraries are installed in the order mentioned in the software requirements section.

In order to verify that the packages are installed, use the following steps:

Graphical user interface, text, application

Description automatically generated

Figures 2: Verifying installed NuGet packages

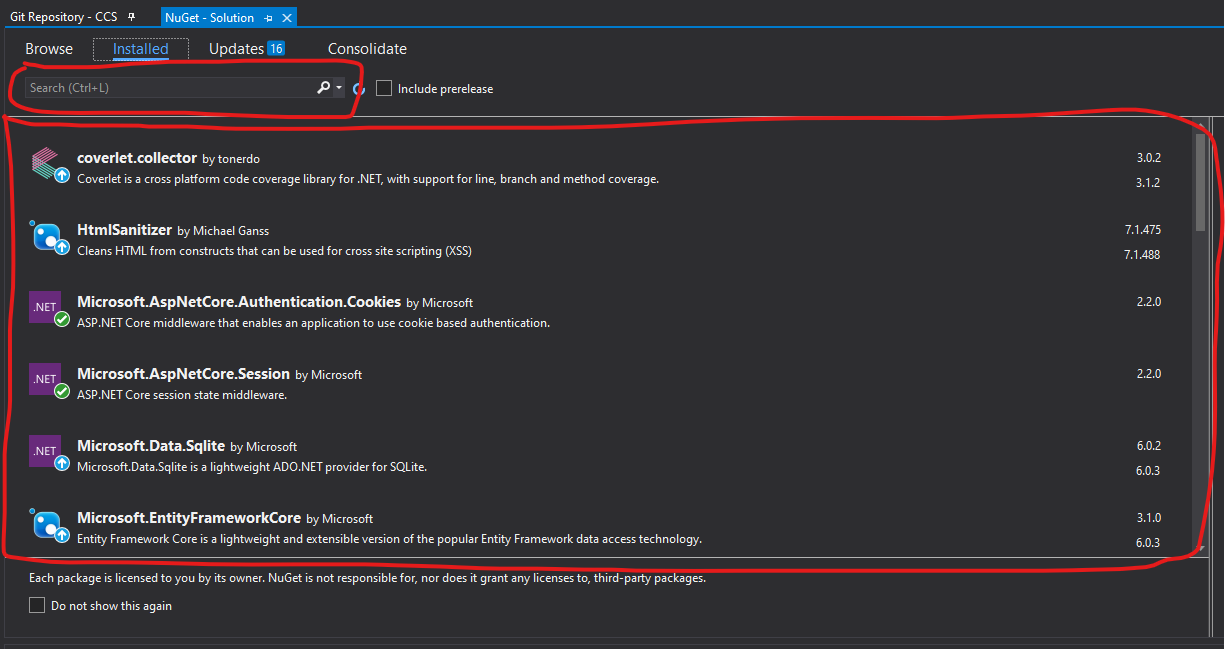


Figure Verifying installed NuGet packages

**Steps**:

1. “Tools”
2. “NuGet Package Manager”
3. “Manage NuGet Packages for Solution…”
4. When the “NuGet – Solution” window opens, you can use the search functionality in order to verify if a package is installed.

For libraries, you can simply look at the codebase and see if there are any folders or CDN links that match the list of libraries mentioned in the Software Requirements section.

#### Installation:

##### **AMS Installation**

1. Login to AMS as an administrator

Username: userad  
password: cs@123test!

**(The password could change. Ask your project manager)**

For CSDEV: <https://csdev.cegep-heritage.qc.ca/Projects/AMS/Login/Index>   
For CSTEST: <https://cstest.cegep-heritage.qc.ca/AMS/>



Figure Application Management System (AMS)

1. In AMS, go to Manage Applications and make sure CCS is there.
2. If CCS is not in the Manage Applications do the following:

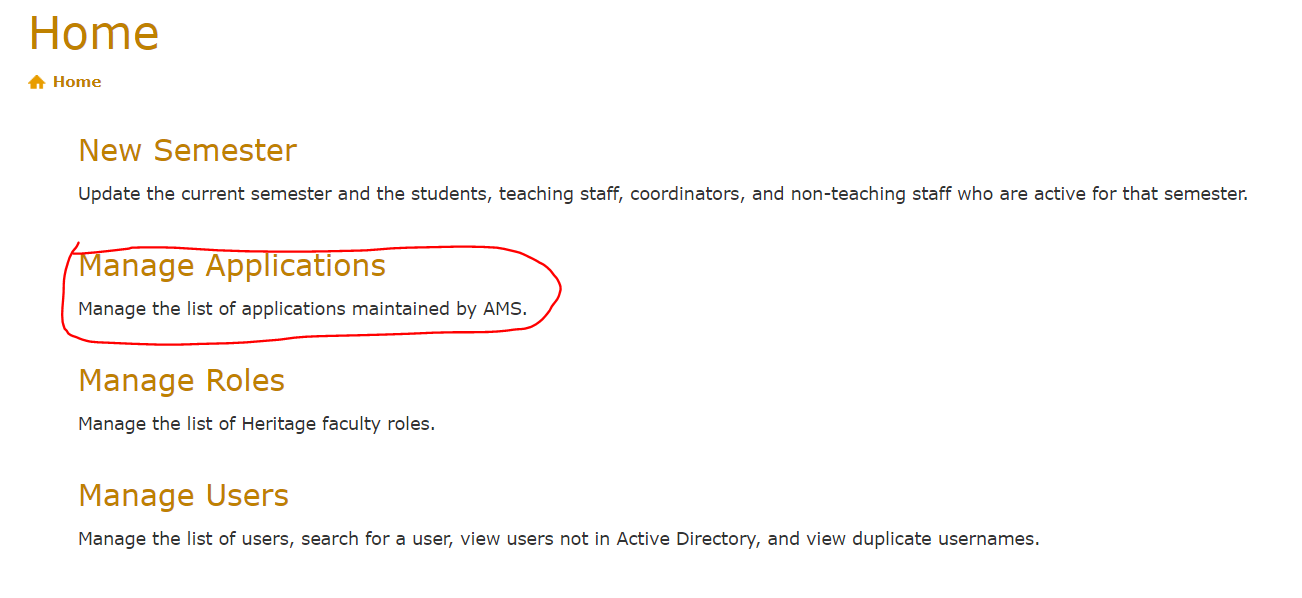


Figure How to verify if CCS is already in the added to AMS

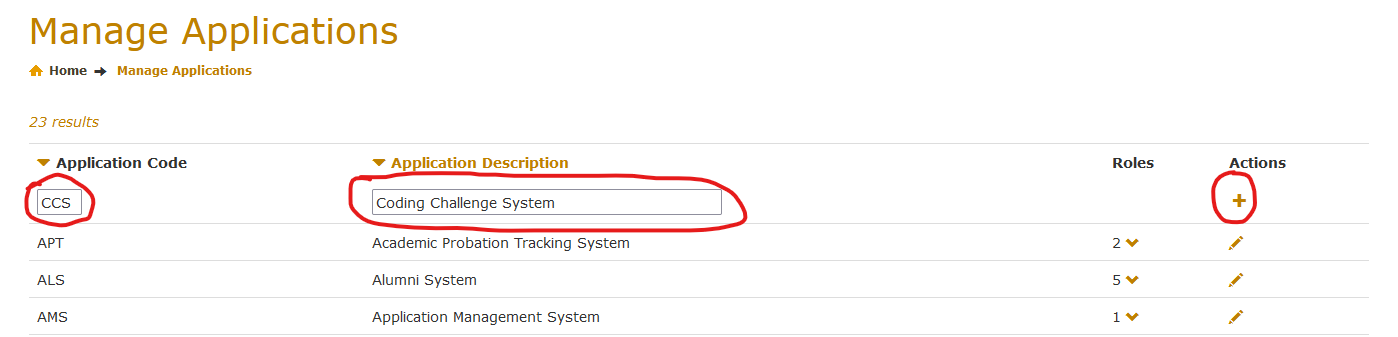


Figure How to verify if CCS is already in the added to AMS

1. In AMS, go to Manage Roles and see if the role code: ST (Student) and TE (Teacher) have CCS in their Applications column.

**ST:**

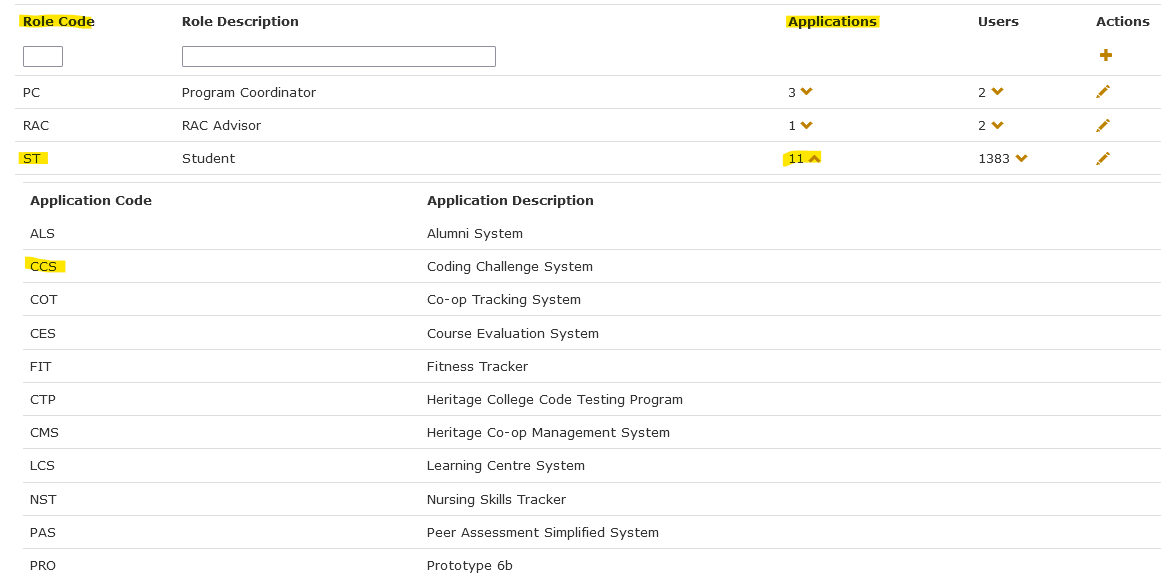


Figure Checking if the ST (student) role has CCS in it

**TE**:

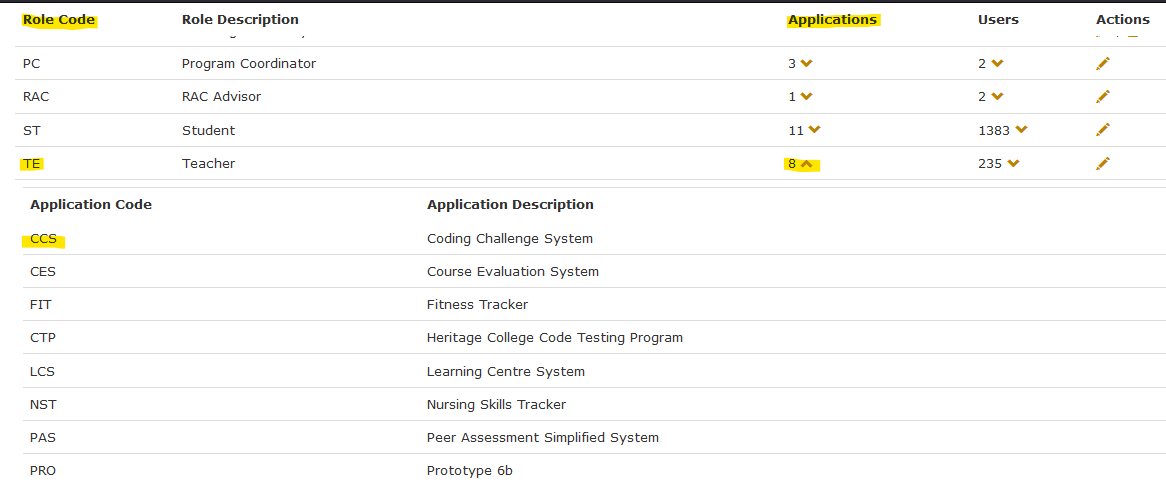


Figure Checking if the TE (teacher) role has CCS in it

1. If ST and/or TE are not there, you simply need to add them using the empty text fields at the top of the table:

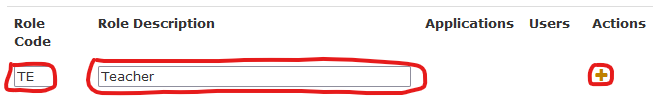


Figure : Adding a TE (teacher) role

1. If CCS is not in the Applications column, do the following:
   1. Edit the role where CCS is missing by clicking the pencil icon



Figure : Editing the TE (teacher) role

* 1. Under applications, search CCS and select “CCS, Coding Challenge System” and click “+”. After clicking “+”, success message should appear:

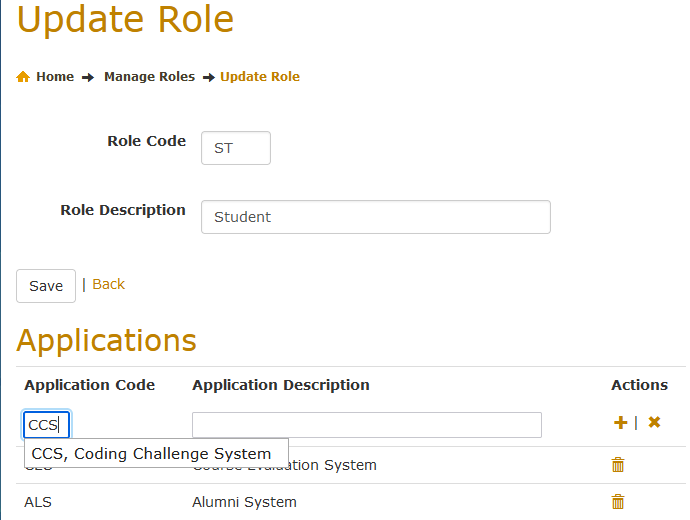


Figure : Adding the CCS application to a role

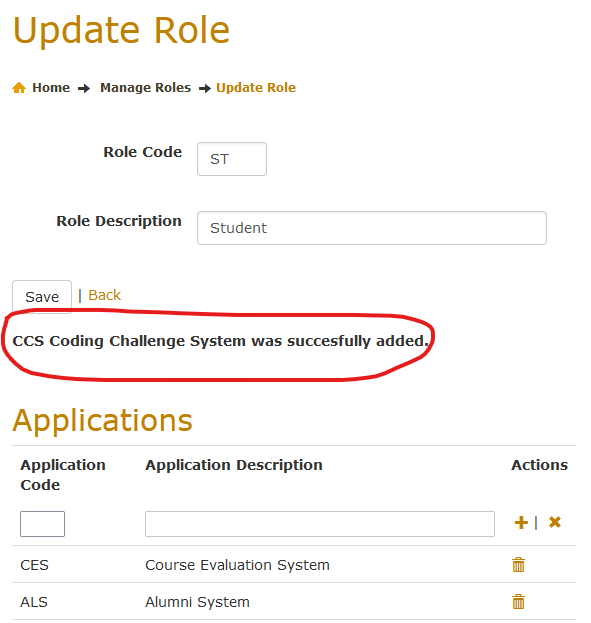
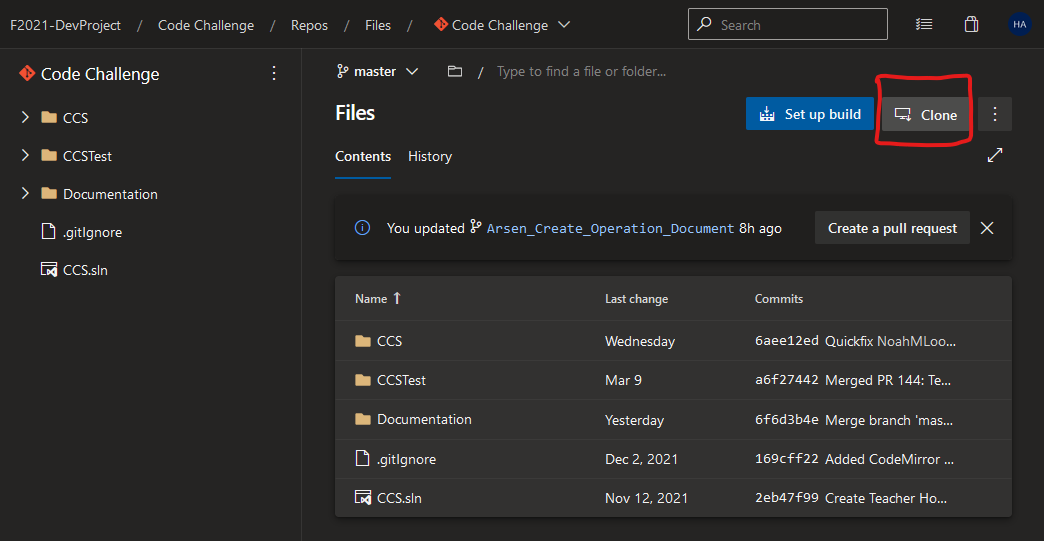


Figure : Confirmation message that appears after successfully adding CCS to a role

* 1. Repeat this step for any other role that’s missing CCS.

##### **Getting the Code**

1. Ensure that Visual Studio 2019+ is installed
2. Ensure SQL Server Management Studio 18+ is installed
3. Navigate to <https://csazure.cegep-heritage.qc.ca:8080/F2021-DevProject/_git/Code%20Challenge>
4. Do the following in order to clone the repository:

Figure : How to install CCS

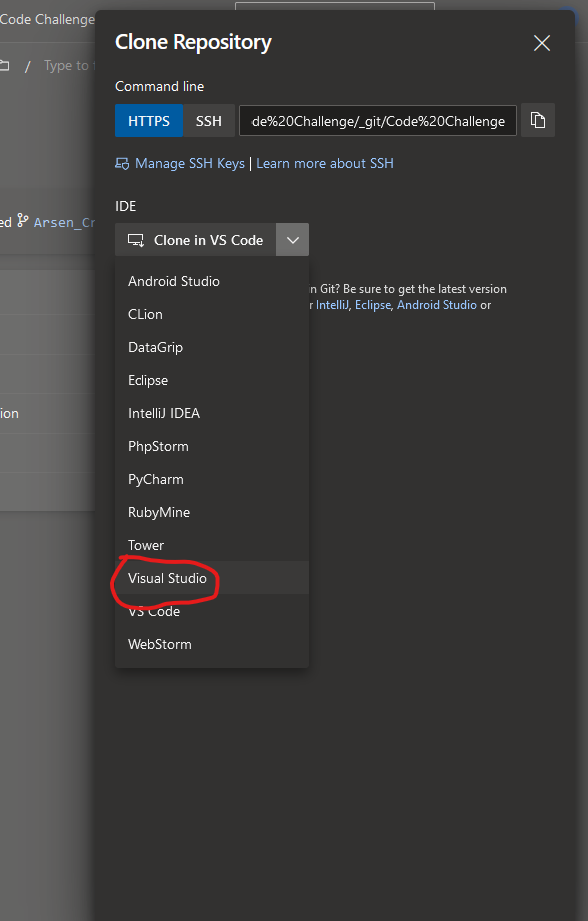


Figure : How to install CCS

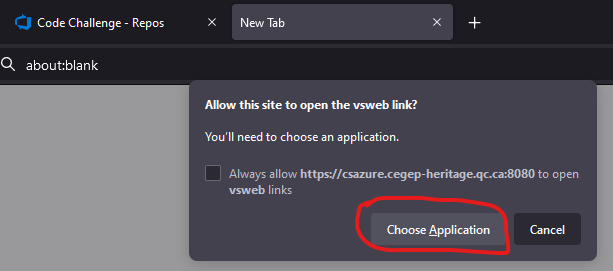


Figure : How to install CCS

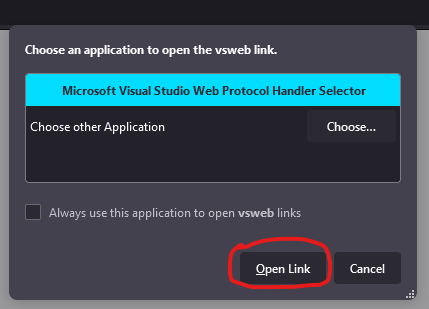


Figure : How to install CCS

##### **Database Setup:**

* + - 1. Launch SSMS
      2. Connect to the appropriate database server (csdev)

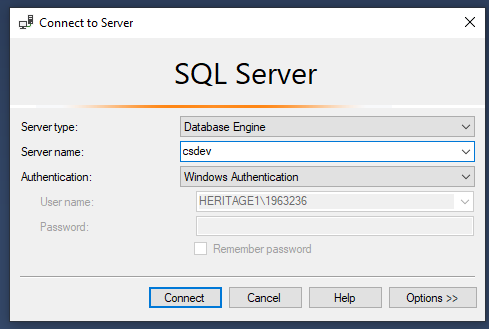


Figure : SSMS Login Screen

* + - 1. Ensure there is an instance of the CCS database.
      2. If there is no instance of the CCS database, perform the following:
         1. Open the CCS\_Script.sql script from the Scripts folder
         2. Ensure the line USE **[master]** is at the top of the file and **[CCS]** in the rest of the file
         3. Execute the file. This will create all the database tables and references needed for the project
         4. The output should show the following:

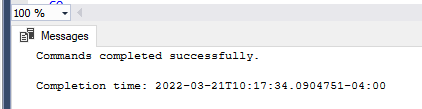


Figure : Success message shown after running CCS\_Script.sql

* + - * 1. Then, open the CCS\_Inserts.sql script from the Scripts folder
        2. Ensure the line **USE [CCS]** is at the top of the file.
        3. Execute file. This will insert sample data into the database tables.
        4. The output should not have any errors in it and show messages such as “(1 row affected)”.

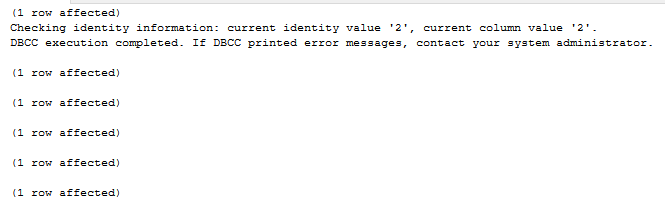


Figure : Success messages shown after running CCS\_Inserts.sql

##### **Connected Services & Application Settings**

Navigate to the CCS folder and double click the solution folder to open the source code in Visual Studio.

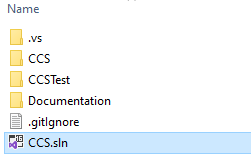


Figure : Where to access the Solution file

Ensure that the service reference for the LoginService is set to the appropriate server by checking the connected services.

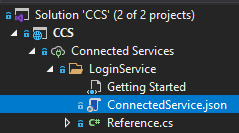


Figure : How to access the ConnectedService.json file

* For **csdev**:

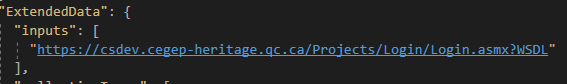


Figure : Login service link inside ConnectedService.json

Do the following:

* Right-click on the existing LoginService folder
* Click on “Delete” and click “Ok” on the warning.
* Double-click on the “Connected Services”, and select **Microsoft WCF Web Service Reference Provider:**



Figure : Microsoft WCF Web Service Reference Provider

The URI should be associated with the appropriate Login service (if unsure of the location, please contact the Operational Manager).

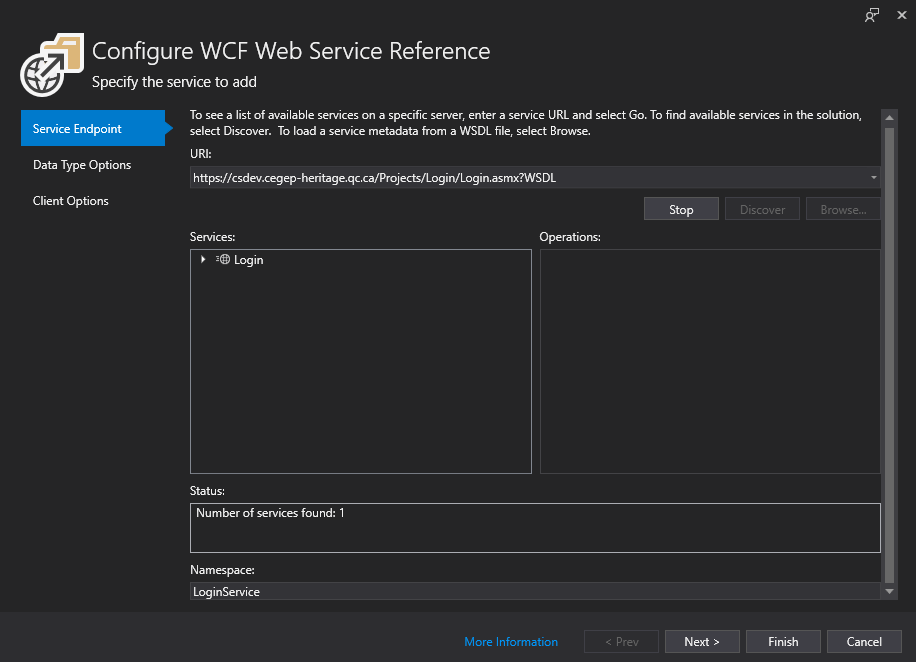


Figure : Configuring the Login Service

Make sure to change the connection string to the appropriate database server. For example, this is how the appsettings.json file should look like:

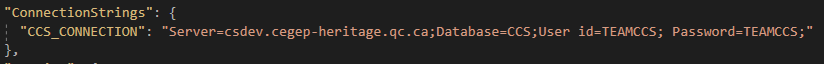
****

Figure : Database connection string inside appsettings.json

#### Post Installation verification steps:

Verify the installation is correct.

1. check if the directory structure is populated with the right folders and files.

Sample directory structure snapshots provided below:

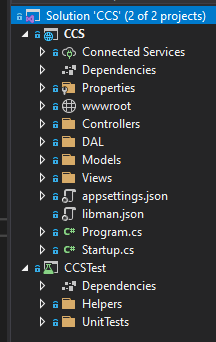


Figure : Folder structure of CCS codebase (minimized)

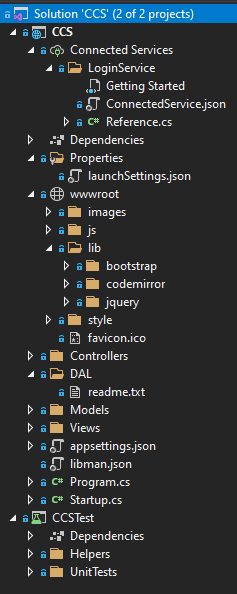


Figure : Folder structure of CCS codebase (expanded snapshots)

## Setup & Configurations

### Dependent software and service configuration

The system needs a compiler called Judge0. It was deployed to the UXDEV server on the Heritage College Network and should be found at the following URL: [**http://uxdev.cegep-heritage.qc.ca:2358/submissions**](http://uxdev.cegep-heritage.qc.ca:2358/submissions).

In the event where the link above does not work try the following URL: [**http://uxdev.cegep-heritage.qc.ca:2358/dummy-client**](http://uxdev.cegep-heritage.qc.ca:2358/dummy-client)

Clicking those URL should bring you to a web page where you can test Judge0.

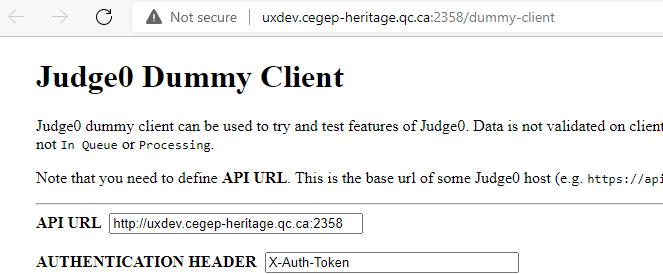


Figure : UXDEV Dummy Client

If there is a problem setting it up, the contact the Computer Science IT Technician.

### Deployment

In order to deploy the application to a server, please refer to the Deployment Document that is located in “[/Documentation/Reports/CCS\_Deployment\_Document.docx](https://csazure.cegep-heritage.qc.ca:8080/F2021-DevProject/_git/Code%20Challenge?path=%2FDocumentation%2FReports%2FCCS_Deployment_Document.docx&version=GBmaster)” ([Section 4.2](#_External_Documents))

There are two deployed versions, one on CSTEST server and one on CSDEV server hosted at Heritage College.

# Operations Procedure

## Basic Operations

### Start, Stopping and Restarting the Application

1. Remote Desktop connect to the server CCS is deployed on.
2. Open the search bar and enter “Internet Information Services (IIS) Manager”
3. Click on the server on the left side under “Connections” and click the dropdown.
   1. Example for CSDEV:

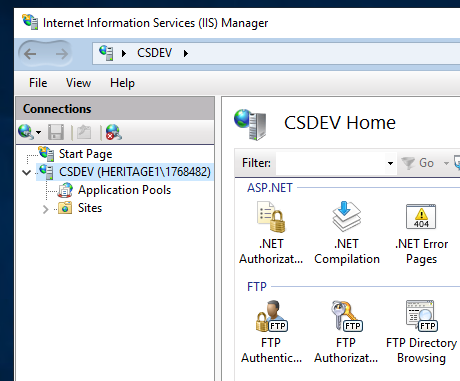


Figure : IIS window for CSDEV

1. Click on the “Application pools” tab
2. Find the “CCSAppPool”:
   1. Example for CSDEV:

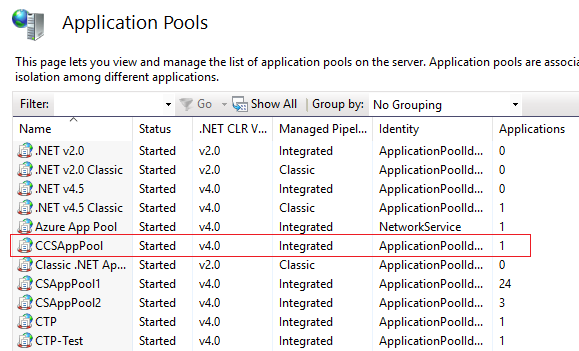


Figure : Locating the application pool for CCS

1. Right click on CCSAppPool and select the option you want.
   1. Example for CSDEV:

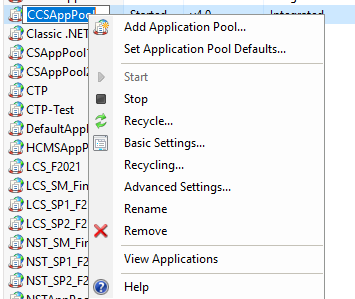


Figure : Window that appears upon right-clicking on the CCSAppPool

## Basic Sanity Test

After deployment is done, a basic sanity test should be done. This sanity test can be performed any time, it is required to test if the deployed application is working as intended.

In order to perform valid tests on the application, make sure all prerequisites and dependencies are met before running the tests. The following steps are required to perform a sanity test on the CCS application:

1. For this test you will need the credentials for the user types teacher and student.
2. Open the login screen of the application by typing the [URL](https://csdev.cegep-heritage.qc.ca/Projects/CCS/Login) of the deployed application. The system should redirect you to the login page by default.
3. Try introducing valid credentials from a pre-defined teacher or student. The system should redirect you to the home page (This tests the connection of the web application with AMS).
4. Once in any home page, check that no errors are presented. In addition, try clicking on different menu items to see that all data is loaded. This tests that the connection between the web application and the database is working properly. The screenshot below shows how the homepage for a teacher should look like.

A screenshot of a computer

Description automatically generated

Figure : Teacher home page

A screenshot of a computer

Description automatically generated

Figure : The student home page

## Authorization Errors

If the following error occurs when logging in (**You are not authorized to access this system**), this is an AMS error, meaning that there is an error in the way that the user is configured in AMS.  
  
If the following error occurs (**You are not authorized to access this system. Please contact your system administrator)**, this is a CCS error, meaning that the user was properly authenticated in AMS but it is not recognized as a valid user.

If the following error occurs **(The username or password you have entered is incorrect.)**, this is a user error, meaning that the user has either entered their username/password incorrectly or they’re simply not registered as a user in AMS.

If the errors occur with credentials that should be valid, please follow the steps below.

### Adding a User in AMS

1. Login as administrator in AMS.  
     
   Username: userad  
   password: cs@123test!

**(The password could change. Ask your project manager)**  
  
For CSDEV: <https://csdev.cegep-heritage.qc.ca/Projects/AMS/Login/Index>   
For CSTEST: <https://cstest.cegep-heritage.qc.ca/AMS/>



Figure : Logging in as an admin in AMS

1. On the home page, select Manage Users

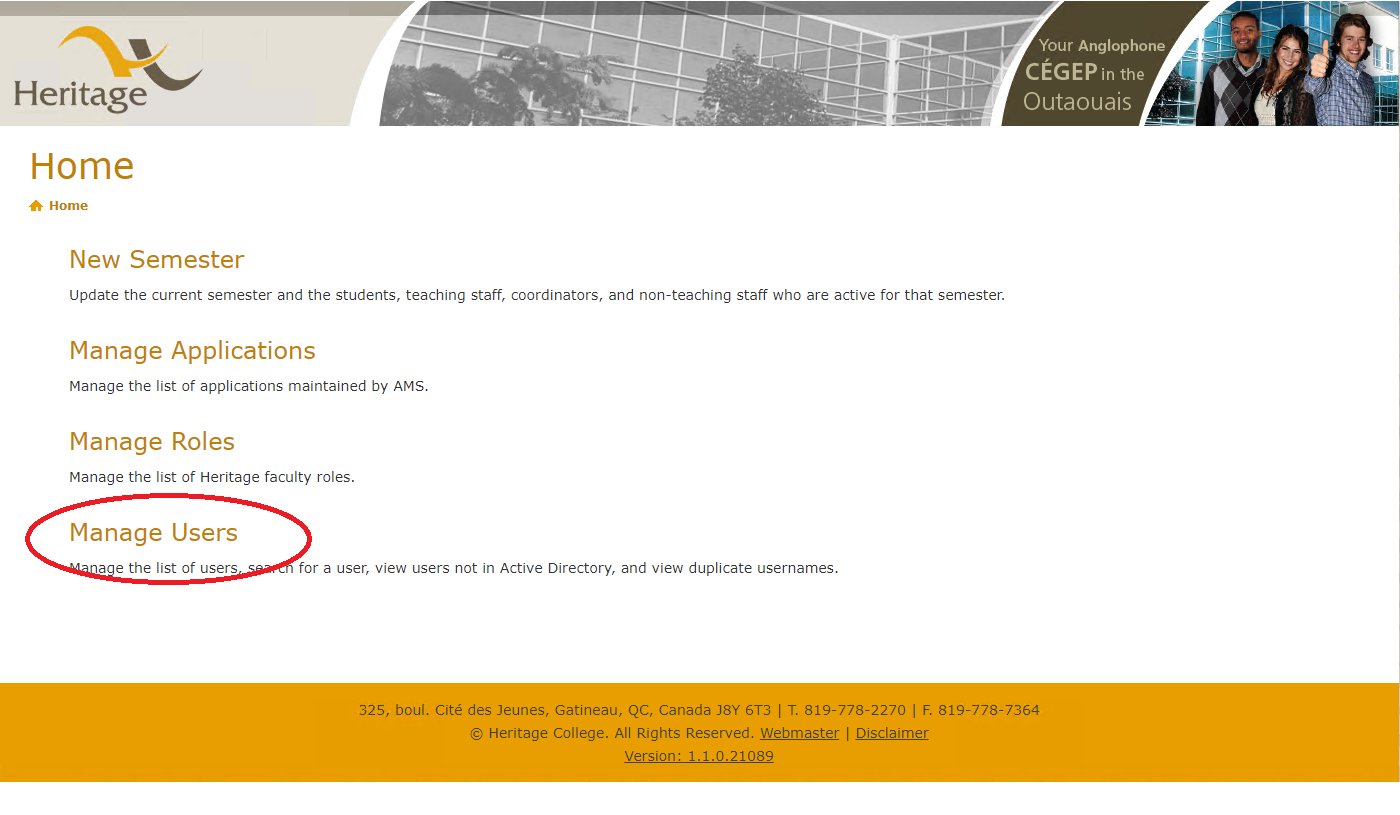


Figure : Accessing “Manage Users” view from AMS Home page

1. Select Add User.

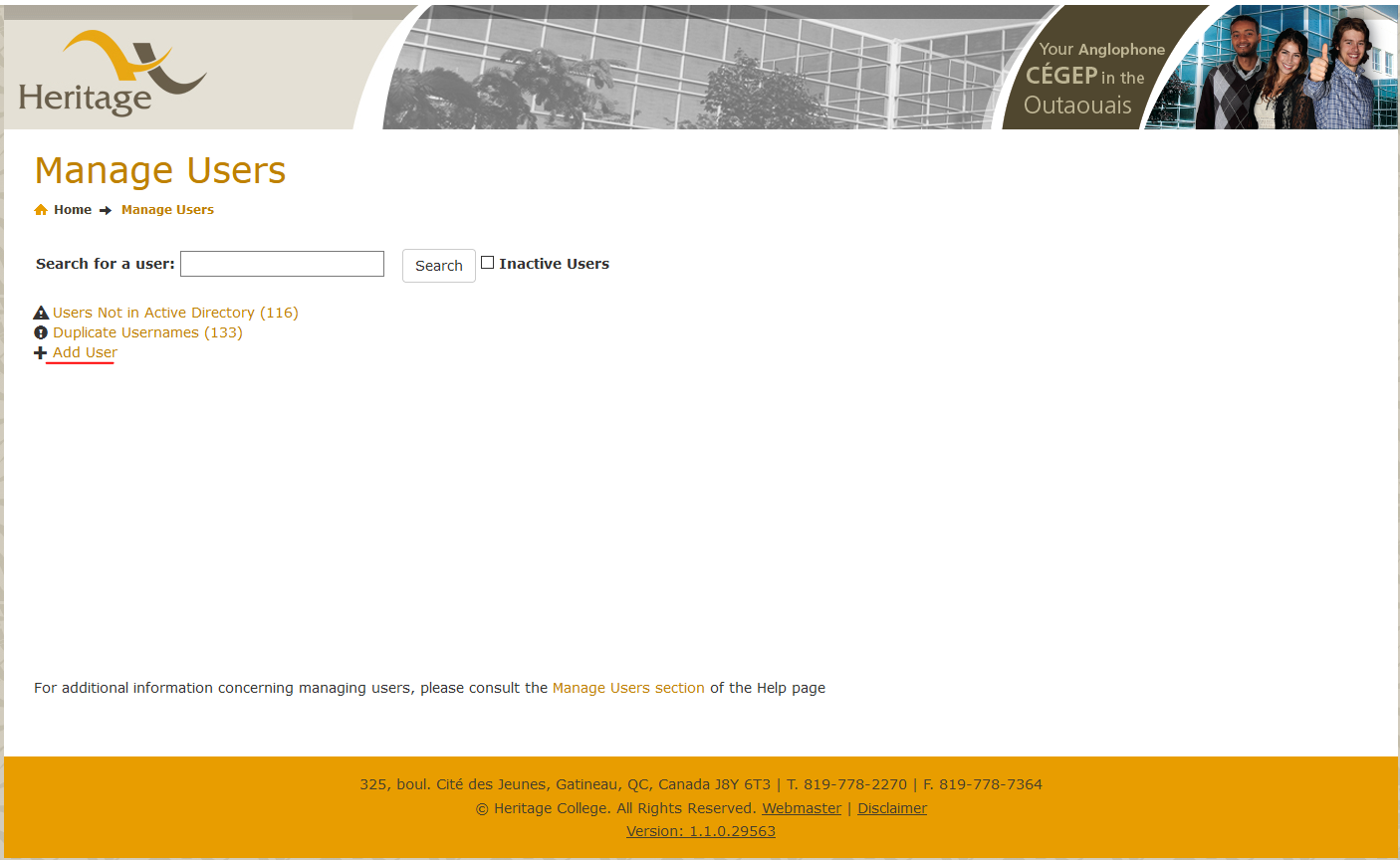


Figure : Adding a new user through “Manage Users” view

1. Enter a first name, last name and username. Select the user type and click Add.

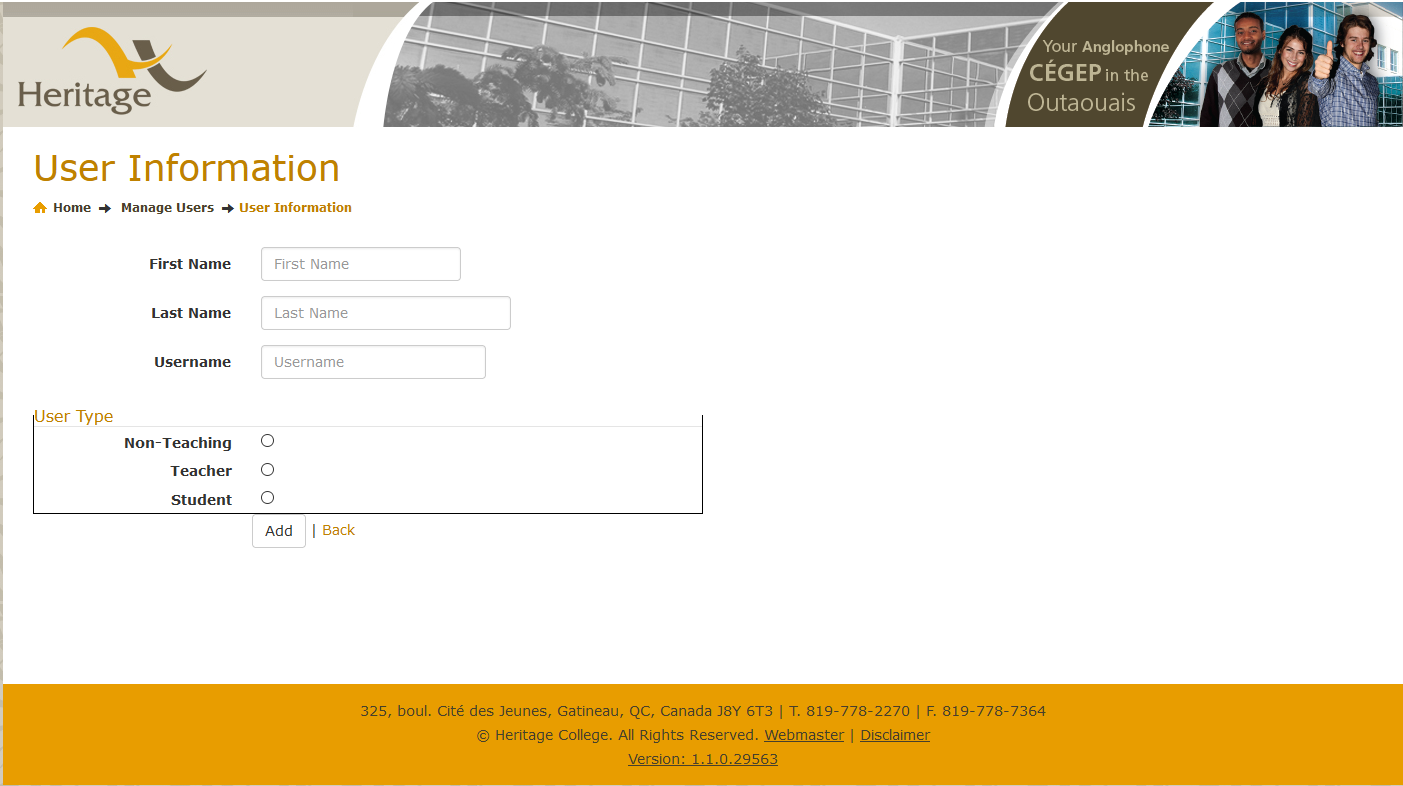


Figure : Adding a new user

### Updating a User in AMS

This workflow can be used to update information about a user in AMS such as their roles, their login username, and others.

1. Login as administrator in AMS.  
     
   Username: userad  
   password: cs@123test!

**(The password could change. Ask your project manager)**  
  
For CSDEV: <https://csdev.cegep-heritage.qc.ca/Projects/AMS/Login/Index>   
For CSTEST: <https://cstest.cegep-heritage.qc.ca/AMS/>



Figure : Logging in as an admin in AMS

1. On the home page, select Manage Users

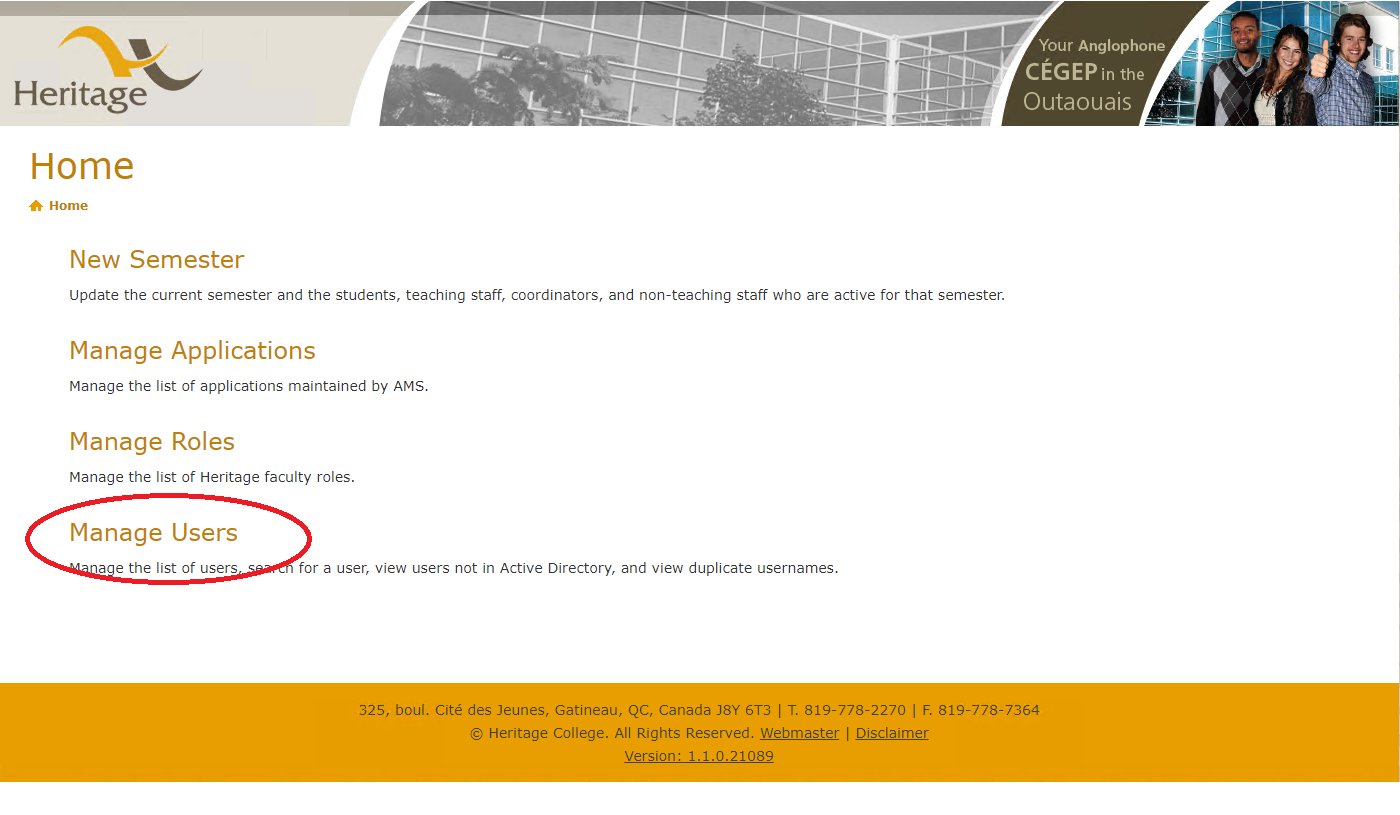


Figure : Accessing “Manage Users” view from AMS Home page

1. Search for the user you are trying to update and select the edit action button.

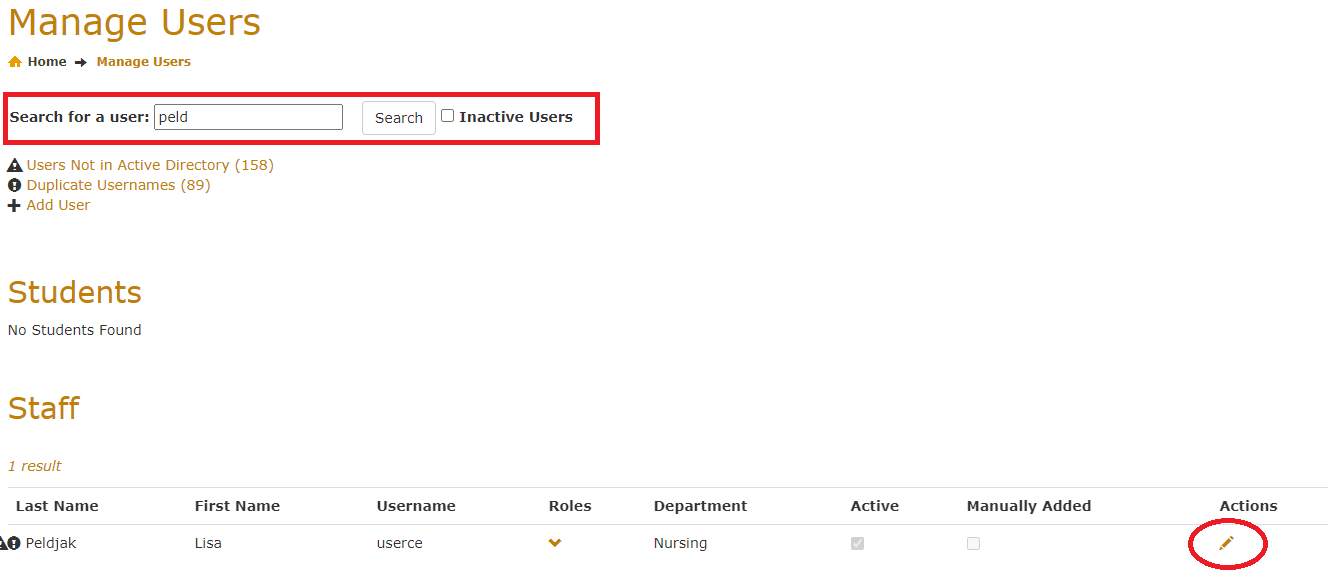


Figure : How to search a user and edit their information

1. Now you may update the user’s information or update their roles. When you are done, click Save.

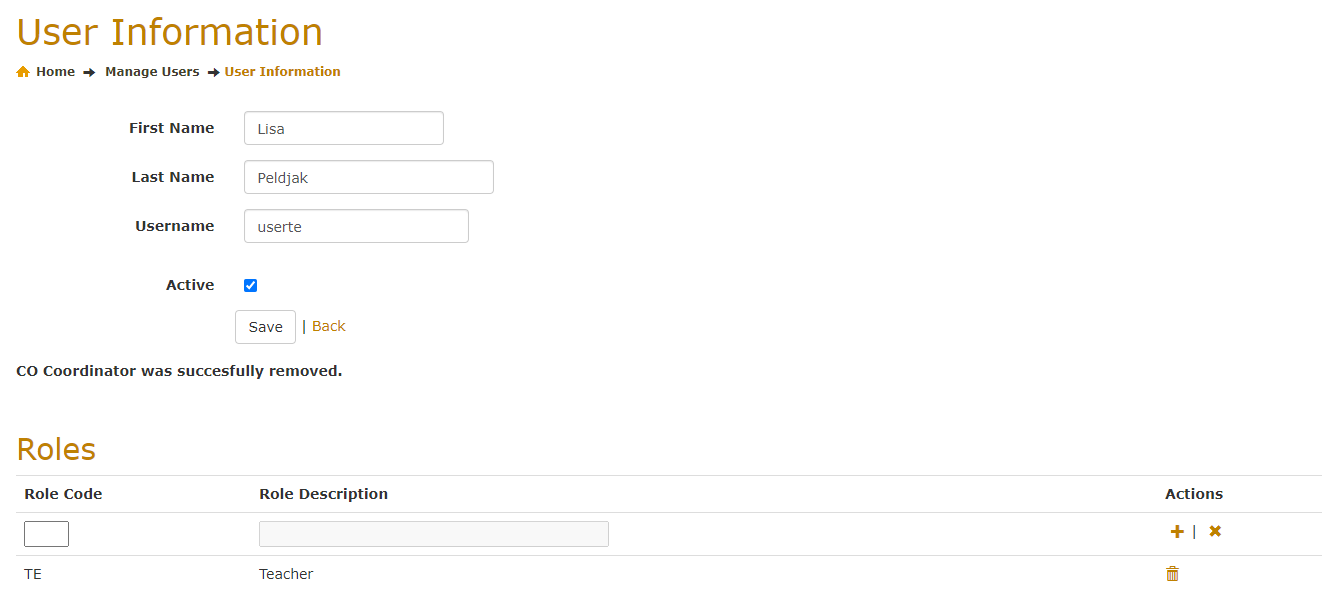


Figure : Updating user information

1. If you decide to update the username, ensure that no other user has this username by searching the username and updating the users with the same username. For example, if we search userte:

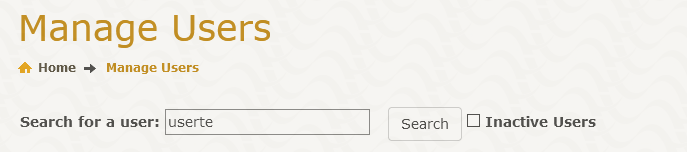


Figure : Searching for a user

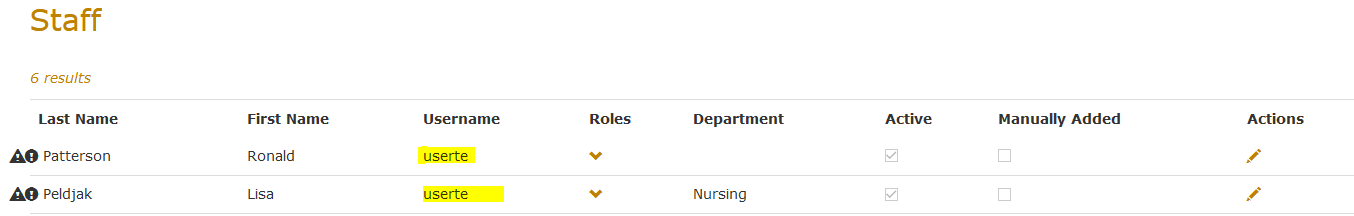


Figure : Locating a user by their username

Each user must have a unique username, so ensure that if two users share a username at any time, one of the usernames is changed. This can occur if, for example, you would like Lisa Peldjak to have the userte username, but Ronald Patterson already holds this username. In this case, you must change Ronald Patterson’s username before moving on.

## Monitoring

### On-Going Monitoring

The following needs to be monitored throughout the day to maintain the proper health of the component:

* CPU
* Memory
* Disk space
* File system
* File descriptors
* Pending message size are within the acceptable range.

### Reactive steps

In the case of an issue such as low network connections, external service disconnects or power outages please reference these steps to solve the issue. A basic sanity test ([Section 3.2](#_Basic_Sanity_Test)) must be done at the end of each step to assure that CCS is running properly.

Reactive Steps:

1. If the issue is known and awaiting to be fixed. Once it is properly resolved, run CCS again and proceed to complete any tasks that were left uncompleted.
2. Stop and re-run the deployed application: reference [Section 3.1.1](#_Start,_Stopping_and). **It is important to note that contacting the business client is necessary before stopping the system**. Refer to [Section 3.4.3.1](#_Stakeholder_Contacts) to the contact information of the business client.
3. Stop deployed CCS application, delete the mapped solution file from your local machine and re-deploy: reference [Section 2.1](#_Installation) and Deployment Document.

### Escalation Steps

#### Stakeholder Contacts

Stakeholders contact to notify an outage or any other issue (impending or occurred)

|  |  |  |  |
| --- | --- | --- | --- |
| Name of Contact | Primary Hotline | Email | Relation to CCS |
| Christopher Elliott | 613 608-1132 | celliott@cegep-heritage.qc.ca | Project Manager |
| Allan McDonald | 819 778-2270 ext. 2083 | amcdonald@cegep-heritage.qc.ca | Business Client |

**Table 3‑1 Stake Holders Contact**

### Performance Testing

To make sure the application is performing properly you must do the following tests:

* Performance Tests: Performance tests are tests that aims to demonstrate that the system is meeting its performance criteria. To do this type of test you must go through each page in the web application and time how long it takes for the page to load. The worst acceptable time is 2 seconds.
* Sanity Tests: Sanity tests are tests that ensures the system is working as intended. An example of what we would need to do are in the following section: [Section 3.2](#_Basic_Sanity_Test).

# Appendix

## Database Scripts

|  |  |  |
| --- | --- | --- |
| **Reference Name** | **Location** | **Description** |
| CCS\_Script.sql | /Documentation/Scripts/ | This will create all the database tables and references needed for the project |
| CCS\_Inserts.sql | /Documentation/Scripts/ | This will insert sample data into the database tables created by the CCS\_Script.sql script |

**Table 4‑1 Information about the Database Scripts**

## External Documents

|  |  |  |
| --- | --- | --- |
| **Reference Name** | **Location** | **Description** |
| CCS\_K40\_A08\_System\_Documentation.docx | /Documentation/Reports/ | Outlines all the requirements set into place for the CCS project |
| CCS\_User\_Documentation.docx | /Documentation/User Documentation/ | A user guide briefly explaining how the system works |
| CCS\_Deployment\_Document.docx | /Documentation/Reports/ | Contains steps on how to deploy the application to a server |

**Table 4‑2 Information about related documents**

# Approvals

This document has been read and approved by the following people, responsible for its implementation.

| Name | Approval |
| --- | --- |
| Emmanuelle Fontaine | Approved |
| Nahom Haile | Approved |
| Abraham Getachew | Approved |
| Zhaojiang Zhong | Approved |
| Alexander Dionne | Approved |
| Noah Loomis | Approved |
| Arsen Hakobyan | Approved |

**Table 5‑1 Approvals**

# History

| Version | Status | Date | Author | Reason for changes |
| --- | --- | --- | --- | --- |
| 0.2.1 | Completed | 2022-03-21 | Arsen Hakobyan | Creating the document |

**Table 6‑1 History of the changes made to the runbook**