SecureED.

User manual

Abstract

SecureED. is a webapp that simulates a university registration system. It allows students to try to break the app, using various vulnerabilities and exploits. The software also resets itself to its initial settings following a session, allowing users to try various things without worrying about destroying their installation.

This document contains the manual of how to use SecureED. The rest of the document is ordered as follows: Chapter 1 describes the system requirements. Chapter 2 contains an explanation of how to start the program. Chapter 3 includes instructions for Professors to enter grades. Chapter 4 includes instructions for Students to search for and register for courses. Chapter 5 includes instructions for admins to add, search and edit non-admin user accounts. Chapter 6 describes vulnerabilities and provides explicit examples of their implementations.

This document describes functionality of SecureED. v. 1.0, dated 7 May 2021.

Table of Contents

1 REQUIREMENTS

1.1 SYSTEM REQUIREMENTS

SecureED. is written in HTML, CSS, Javascript, and PHP, and was built and tested on Google Chrome v89 – thus, Chrome is the recommended browser. Otherwise, all libraries are included in the installation folder. Screen resolutions above 1920x1080, and aspect ratios greater than 16:9, are untested, and thus unsupported.

2 GETTING STARTED

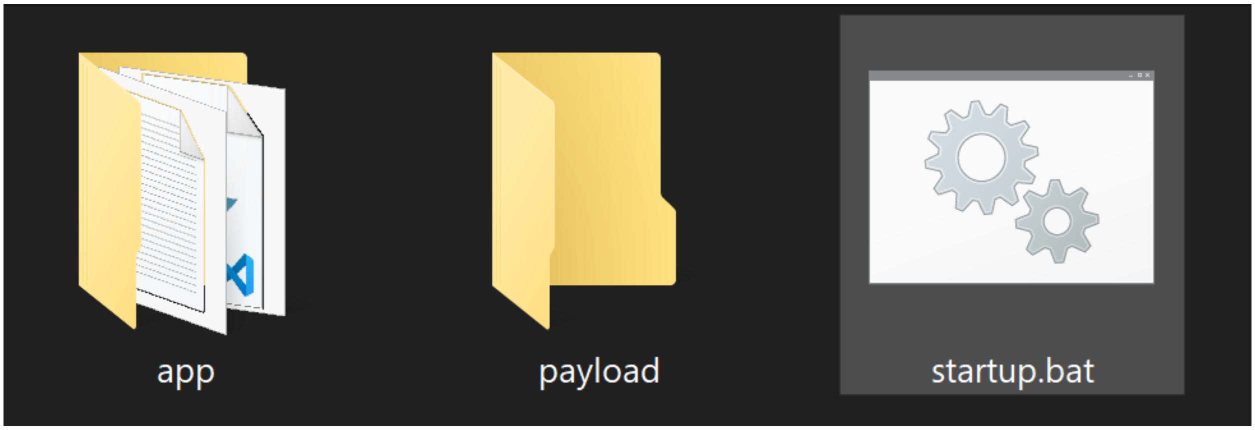
2.1 INSTALLATION

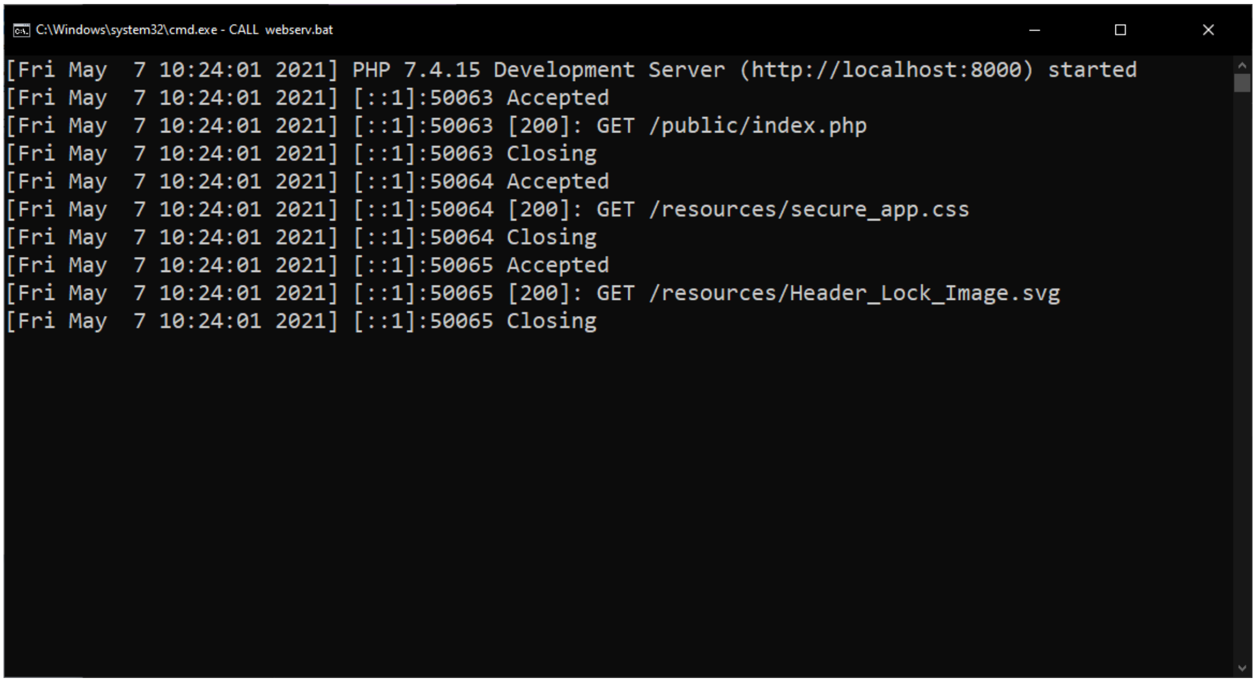
Unzip the file to the folder of your choice. All of the necessary libraries are included – no setup is required.

2.2 START-UP

SecureED. is portable, thus self-contained. To start the webserver, please do the following:

1. Navigate to the root folder of your installation.
2. Execute startup.bat



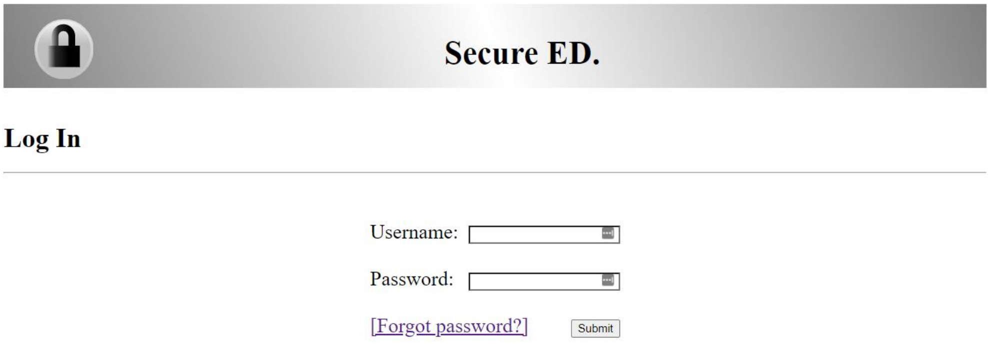


```
[Fri May  7 10:24:01 2021] PHP 7.4.15 Development Server (http://localhost:8000) started
[Fri May  7 10:24:01 2021] [::1]:50063 Accepted
[Fri May  7 10:24:01 2021] [::1]:50063 [200]: GET /public/index.php
[Fri May  7 10:24:01 2021] [::1]:50063 Closing
[Fri May  7 10:24:01 2021] [::1]:50064 Accepted
[Fri May  7 10:24:01 2021] [::1]:50064 [200]: GET /resources/secure_app.css
[Fri May  7 10:24:01 2021] [::1]:50064 Closing
[Fri May  7 10:24:01 2021] [::1]:50065 Accepted
[Fri May  7 10:24:01 2021] [::1]:50065 [200]: GET /resources/Header_Lock_Image.svg
[Fri May  7 10:24:01 2021] [::1]:50065 Closing
```

Your default browser will be opened, and you will be presented with a login screen. Note the command prompt opened in the background – **DO NOT CLOSE THIS** until you want to end your session! Upon closing the command prompt in the background, your session will be terminated, and all data will be reset to their default values.

2.3 LOG IN

Enter a username and password into the spaces provided, and then press “Submit”.

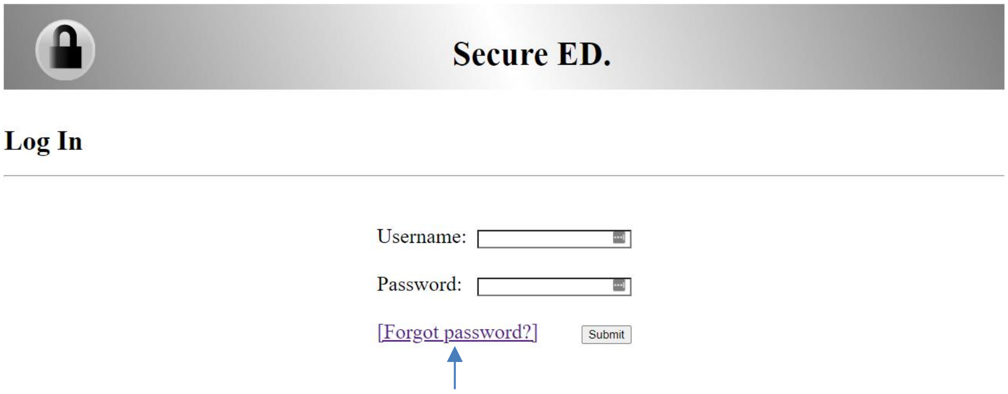


Note: If you have forgotten your credentials, please select “Forgot password?”. See the next section for information regarding how to recover your password.

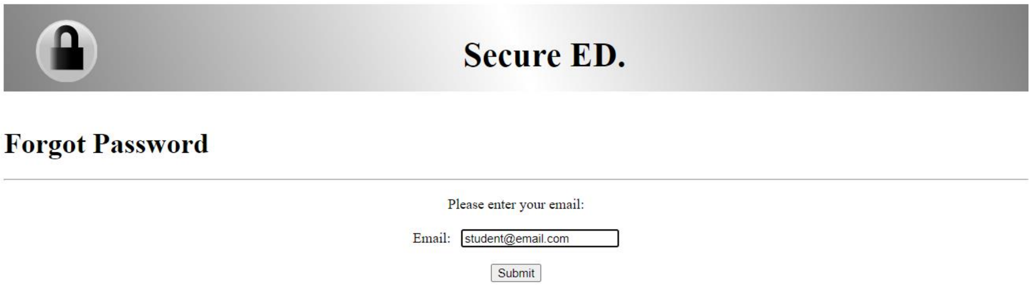
2.4 FORGOT PASSWORD

In order to recover a lost password:

1. Click “Forgot Password?” on the Log In page.



2. On the next page, please enter your email in the provided field, and select “Submit”.



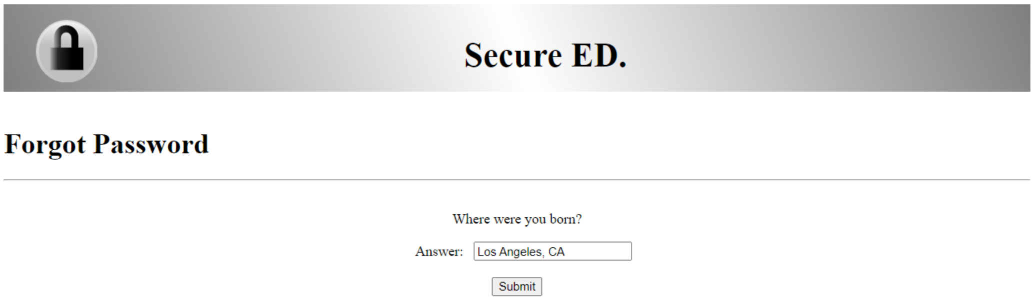
Secure ED.

Forgot Password

Please enter your email:

Email: student@email.com

3. You will then be asked the security question associated with that account.



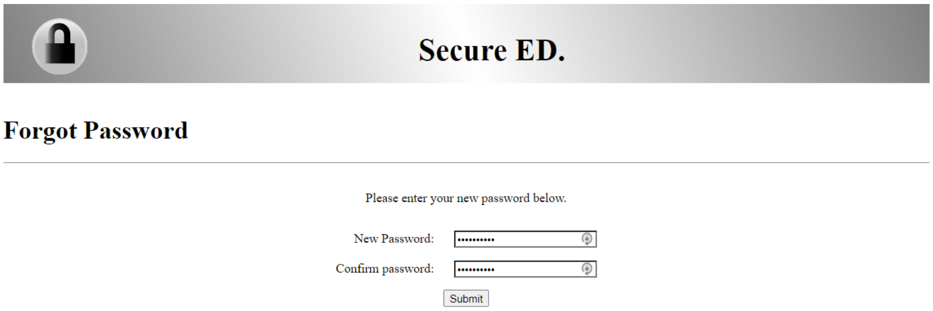
Secure ED.

Forgot Password

Where were you born?

Answer: Los Angeles, CA

4. Enter the correct answer, and you will be taken to a page that will allow you to change your password.



Secure ED.

Forgot Password

Please enter your new password below.

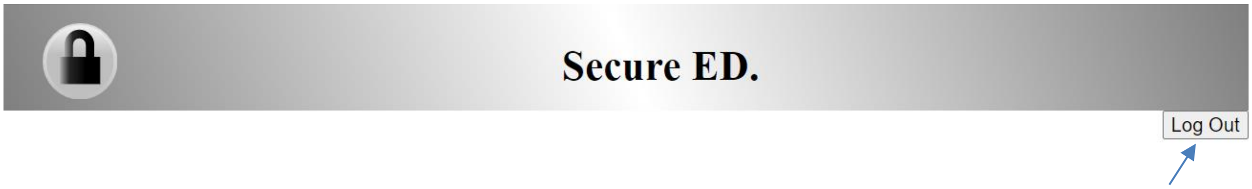
New Password: ••••••••••

Confirm password: ••••••••••

Note: Be sure that the contents of the New Password and the Confirm Password fields match!

2.5 Log Out

After you have successfully logged in, you may log out at any time. To do so, click on the “Log Out” button on the top pane of any page.

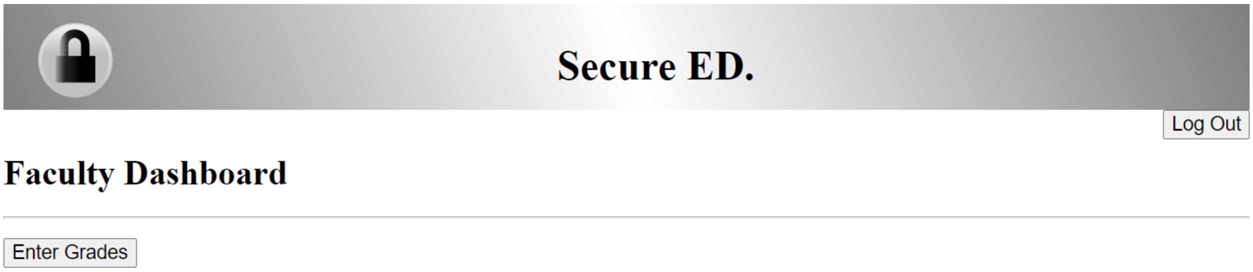


2.6 Closing the Program

To close the program, close your browser and exit the command prompt window. The program will reset itself to its default values the next time you run startup.

3 Faculty

3.1 Dashboard



3.2 Enter Grades

3.2.1 Description

The Enter Grades functionality allows a faculty member to add final grades to a course, using a .csv (comma-separated values) file.

3.2.2 CSV Format

Uploaded files must be comma separated text values, in the following format:

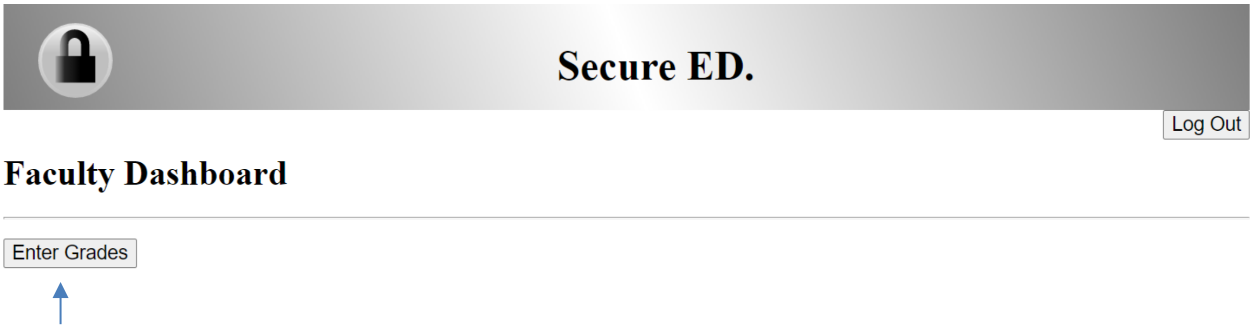
```
USER_ID,GRADE
USER_ID,GRADE
USER_ID,GRADE
…
```

e.g.:

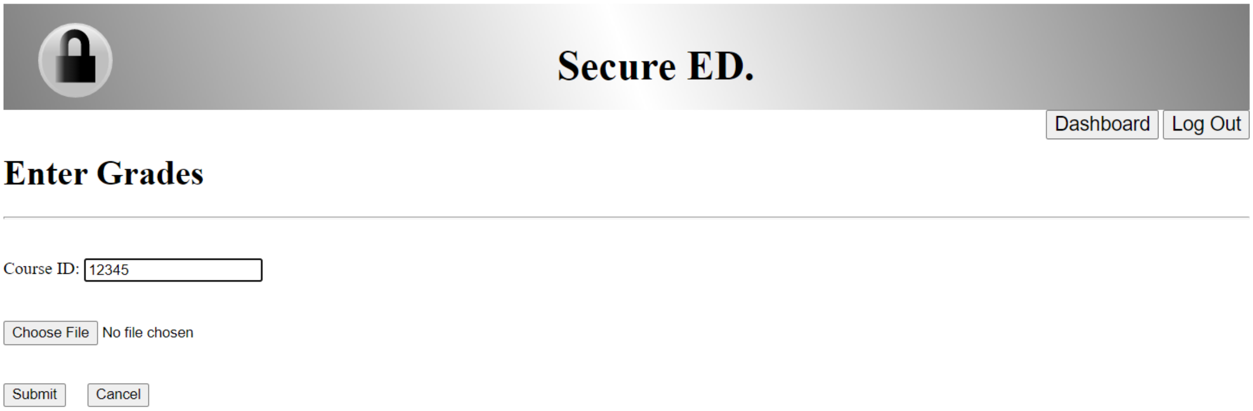
```
927066652,A
927190481,F
927201552,C
927237023,B
927369978,D
927410321,F
927512079,A
927535222,A
927565074,B
927600214,C
927620146,A
927671464,F
927694539,C
927790514,B
```

3.2.3 Instructions

1. From the Dashboard, select “Enter Grades”.



2. On the following page, enter the Course ID (CRN).



Course ID: 12345

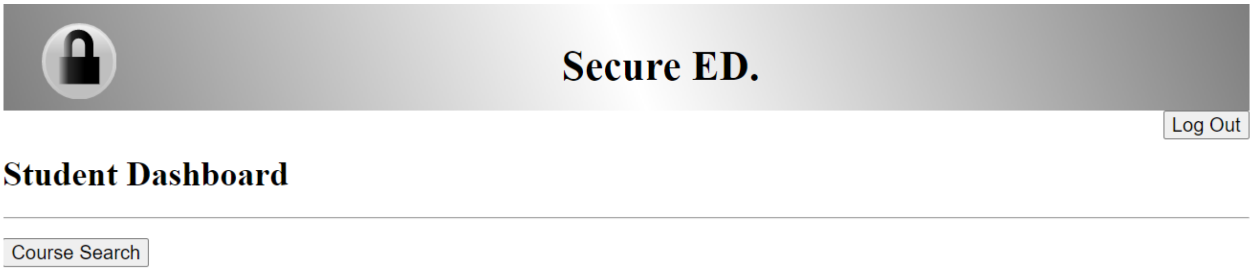
3. Select “Choose File”.

4. Select a .csv file from your computer (see 3.2.2 for correct formatting).

5. Select submit to upload the file.

4 Students

4.1 Dashboard



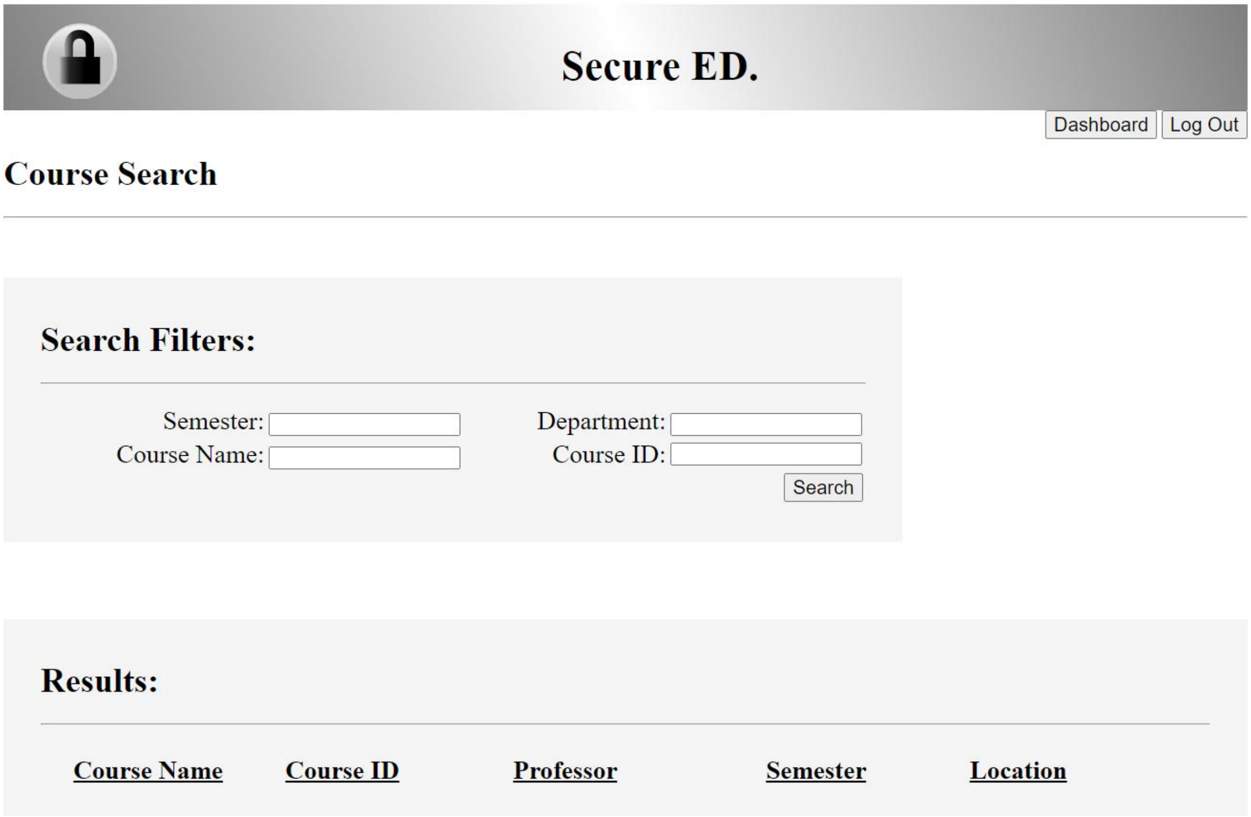
4.2 Course Search

4.2.1 Description

Course search allows students to find current, past and future courses.

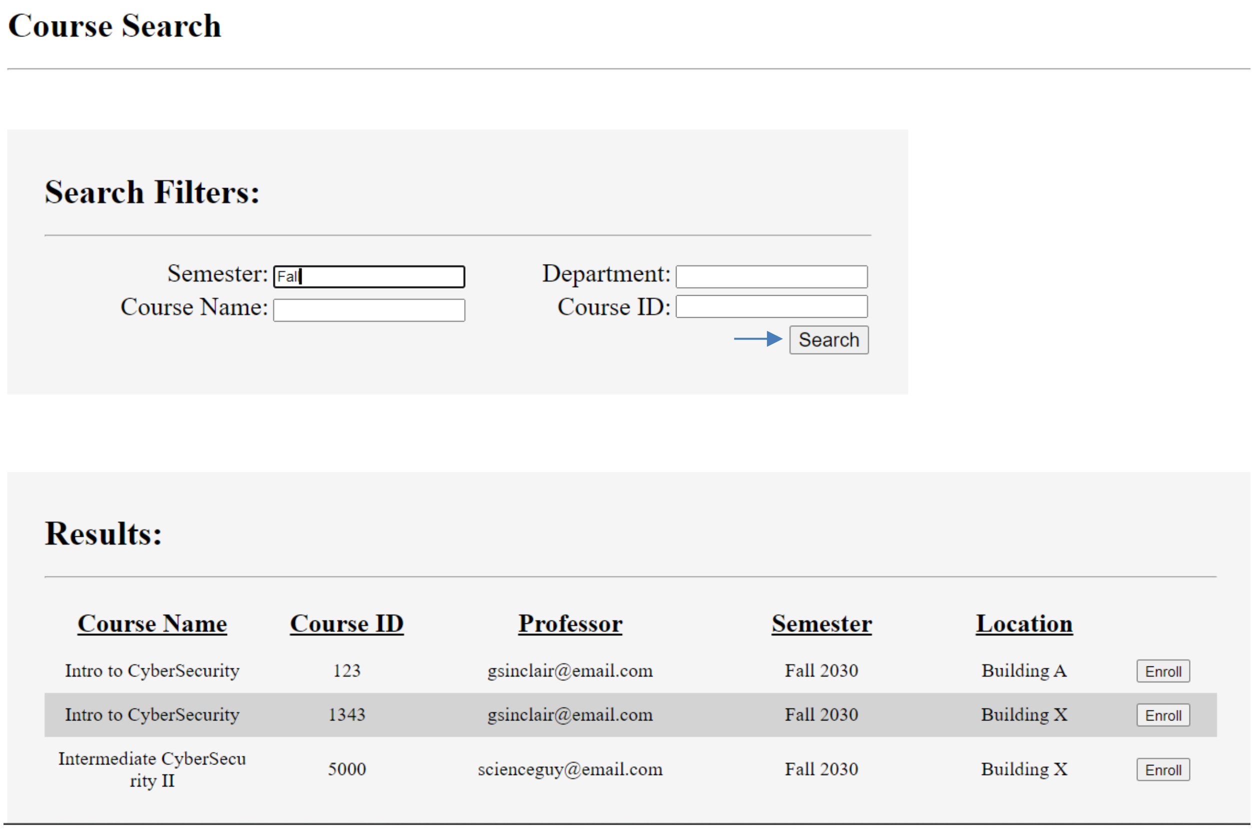
4.2.2 Instructions

1. From the Dashboard, select “Course Search”.
2. On the following page, enter information relevant to the course you may be interested in.



Course Name	Course ID	Professor	Semester	Location

3. Select search.



Course Search

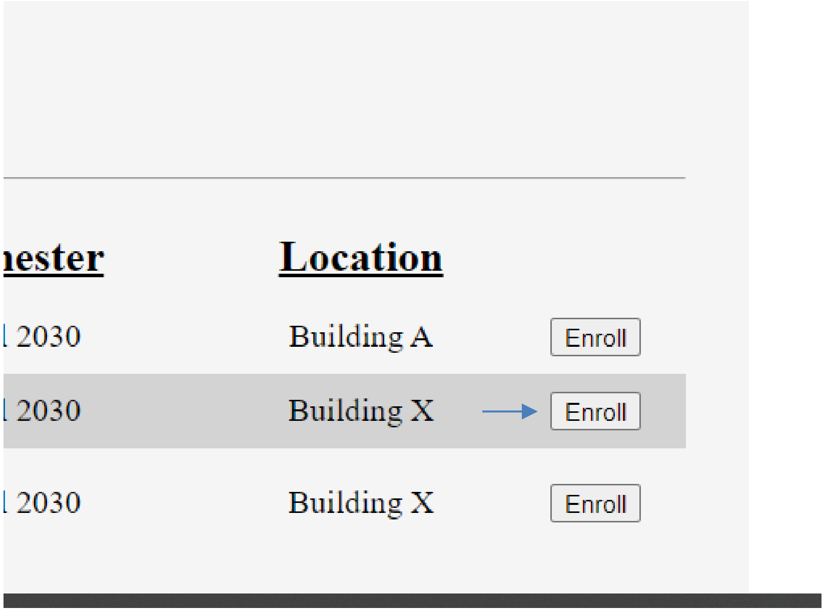
Search Filters:

Semester: Fal
Course Name:
Department:
Course ID:

Results:

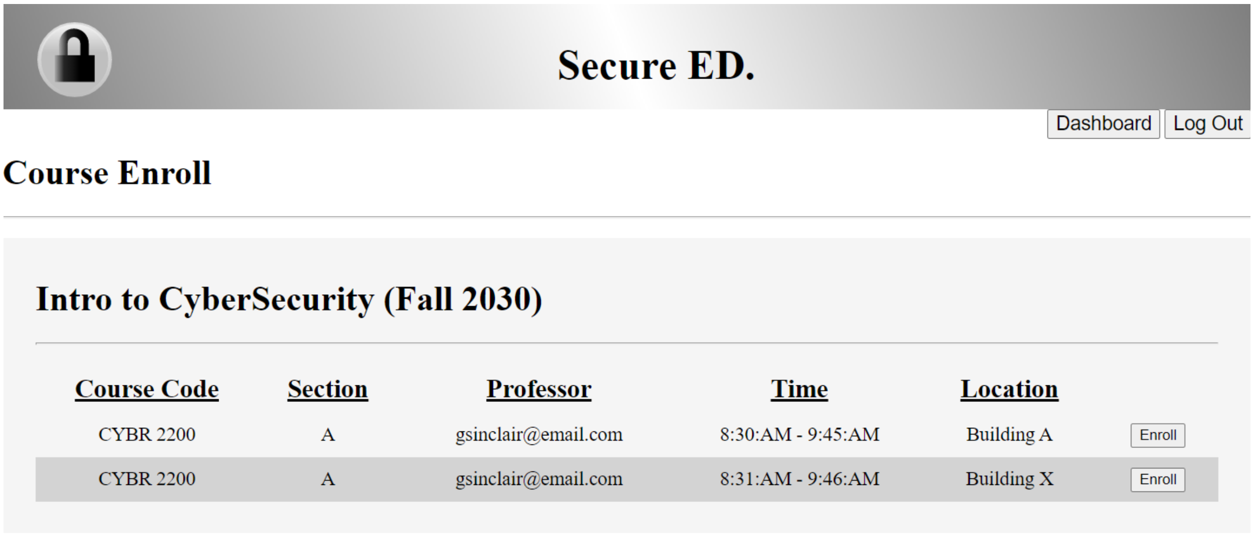
Course Name	Course ID	Professor	Semester	Location	
Intro to CyberSecurity	123	gsinclair@email.com	Fall 2030	Building A	Enroll
Intro to CyberSecurity	1343	gsinclair@email.com	Fall 2030	Building X	Enroll
Intermediate CyberSecurity II	5000	scienceguy@email.com	Fall 2030	Building X	Enroll

4. For a list of sections, select enroll.



ester	Location	
l 2030	Building A	Enroll
l 2030	Building X	Enroll
l 2030	Building X	Enroll

5. You will be presented with a list of sections to browse.



Secure ED.

Course Enroll

Intro to CyberSecurity (Fall 2030)

Course Code	Section	Professor	Time	Location	
CYBR 2200	A	gsinclair@email.com	8:30:AM - 9:45:AM	Building A	Enroll
CYBR 2200	A	gsinclair@email.com	8:31:AM - 9:46:AM	Building X	Enroll

Note: If you don't wish to enroll at this time, you can always select "Dashboard" to return to your landing page without enrolling in a class!

4.3 Course Enroll

4.3.1 Description

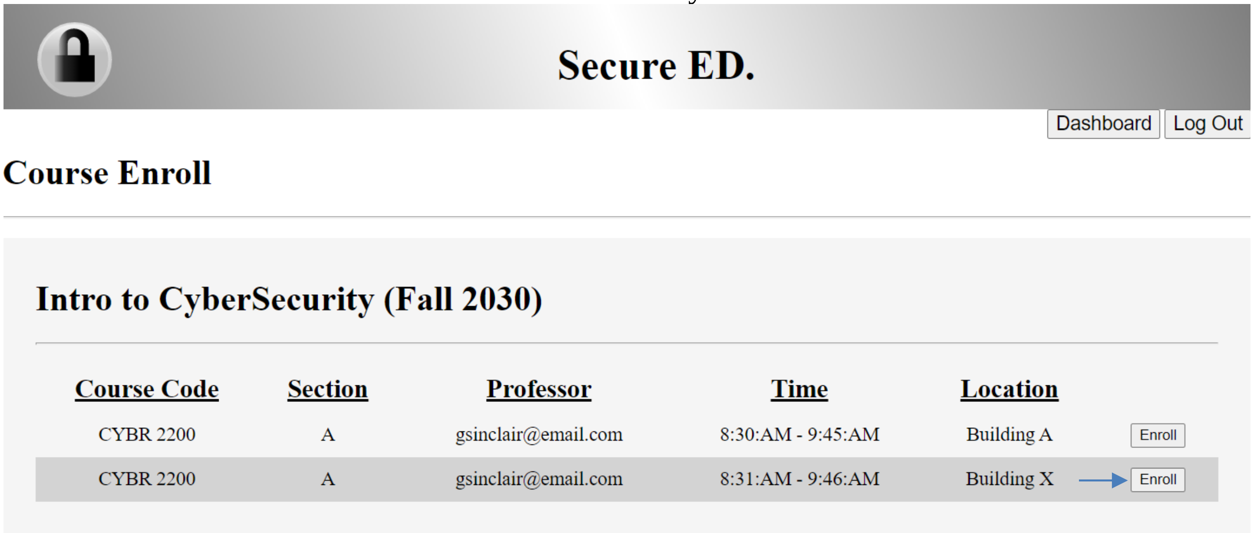
After finding a course using Course Search, a student can enroll in a section of the course.

4.3.2 Instructions

1. Find a course per the instructions in 4.2.2.

2. Select "Enroll" to bring up a list of sections.

3. Select "Enroll" next to the section you wish to enroll in.



Secure ED.

Course Enroll

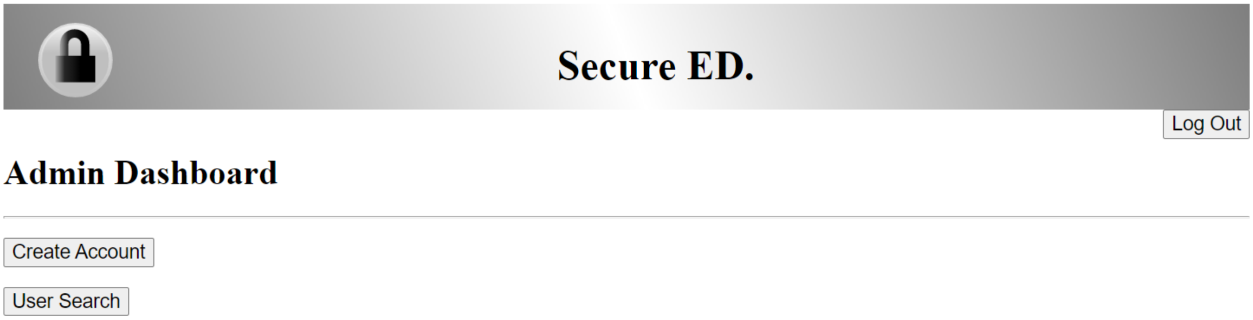
Intro to CyberSecurity (Fall 2030)

Course Code	Section	Professor	Time	Location	
CYBR 2200	A	gsinclair@email.com	8:30:AM - 9:45:AM	Building A	Enroll
CYBR 2200	A	gsinclair@email.com	8:31:AM - 9:46:AM	Building X	→ Enroll

You will automatically be returned to your dashboard.

5 Administrators

5.1 Dashboard



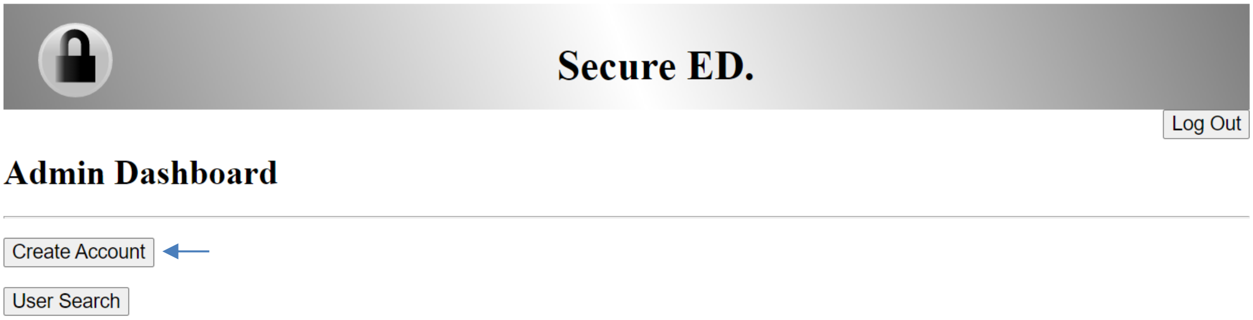
5.2 Create Account

5.2.1 Description

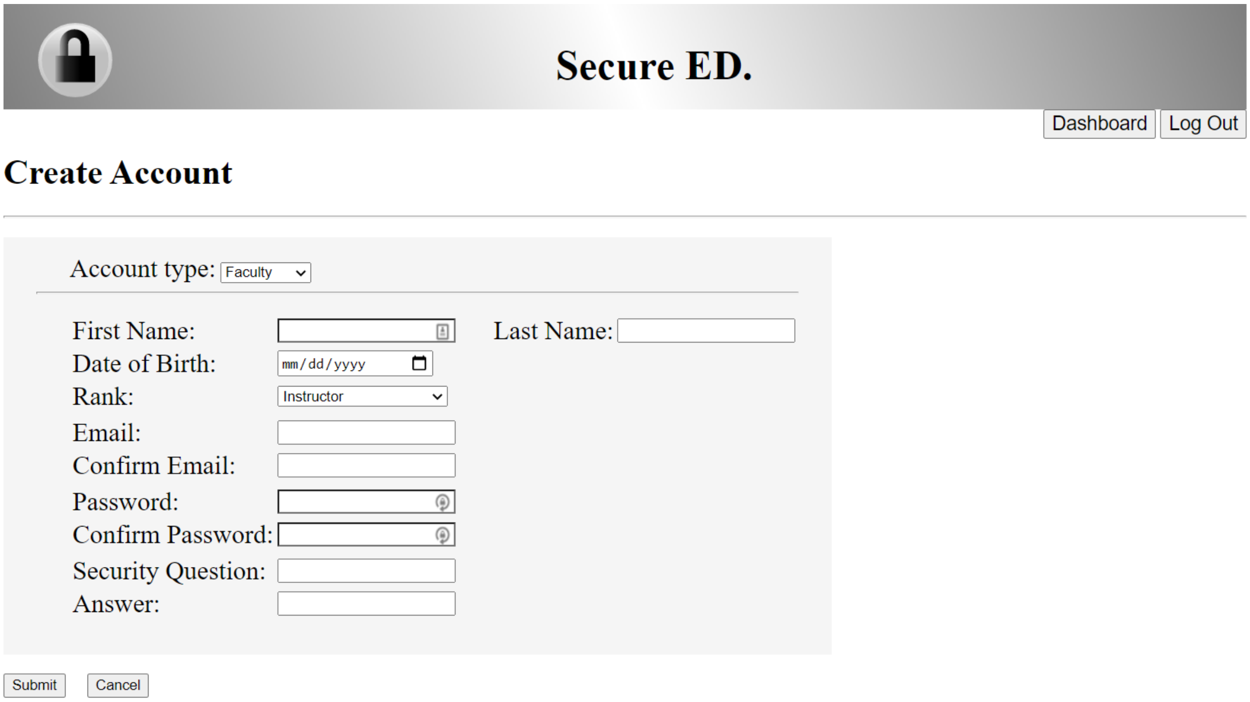
Allows the administrator account to create new accounts for new users. This is the only way to create a new account!

5.2.2 Instructions

1. From the Dashboard, select “Create Account”.



2. On the following page, select the account type.
   Note: Admin accounts cannot be created!



Create Account

Account type: Faculty

First Name:
Last Name:
Date of Birth: mm/dd/yyyy
Rank: Instructor
Email:
Confirm Email:
Password:
Confirm Password:
Security Question:
Answer:

3. Enter the user's:
   - First and Last Name
   - Date of Birth
   - Rank (faculty) or Year (students)
   - Email
   - Password
   - Security Question
   - Answer (to Security Question)

4. Confirm the user's e-mail and password, and re-enter them in their respective confirmation fields.

5. Select "Submit".

5.3 User Search

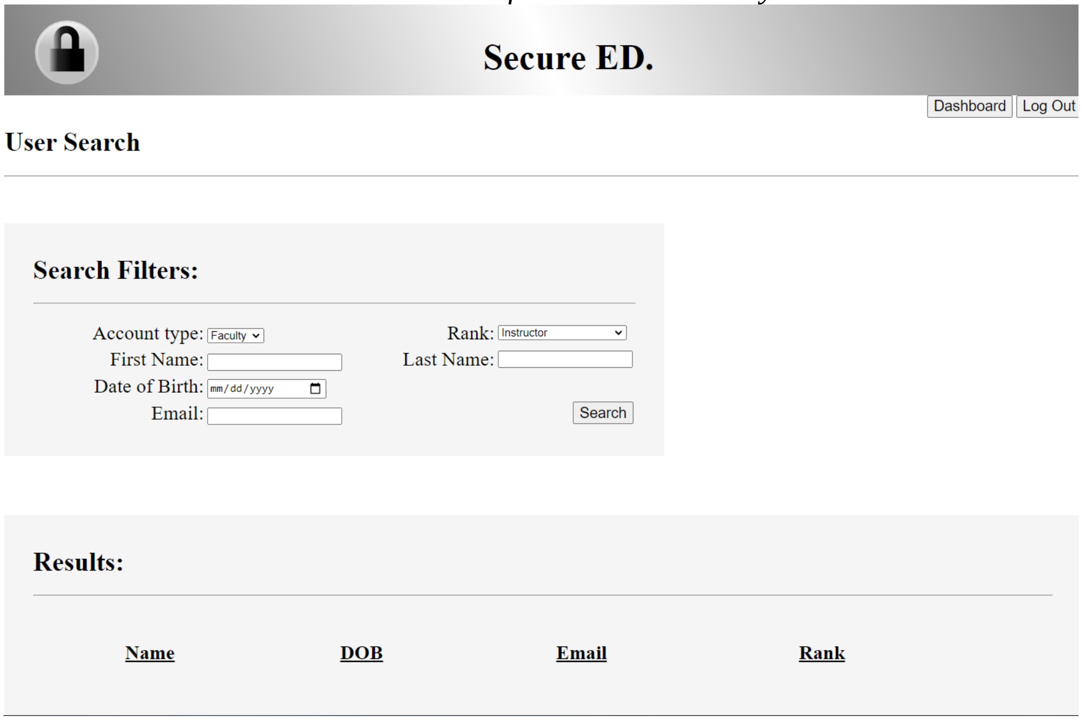
5.3.1 Description

Allows administrators to search for users using filters.

5.3.2 Instructions

1. From the Dashboard, select “User Search”.

2. Select an account type, and a Rank (faculty) or Year (students).
   Note: These two fields are mandatory!



User Search

Search Filters:

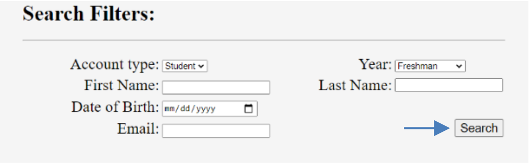
Account type: Faculty
Rank: Instructor
First Name:
Last Name:
Date of Birth: mm/dd/yyyy
Email:

Results:

Name	DOB	Email	Rank

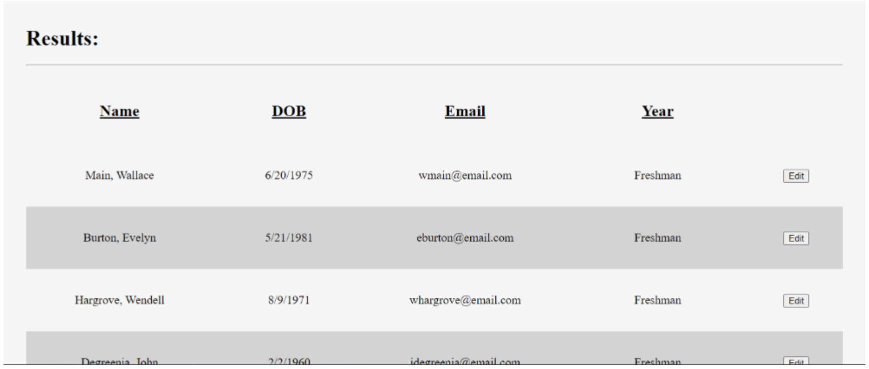
3. Enter any more relevant information into the fields provided.

4. Select “Search”.



Search Filters:

Account type: Student
Year: Freshman
First Name:
Last Name:
Date of Birth: mm/dd/yyyy
Email:



Results:

Name	DOB	Email	Year	
Main, Wallace	6/20/1975	wmain@email.com	Freshman	Edit
Burton, Evelyn	5/21/1981	eburton@email.com	Freshman	Edit
Hargrove, Wendell	8/9/1971	whargrove@email.com	Freshman	Edit
Degreenia, John	2/2/1960	jdegreenia@email.com	Freshman	Edit

A list of matching accounts will appear below.

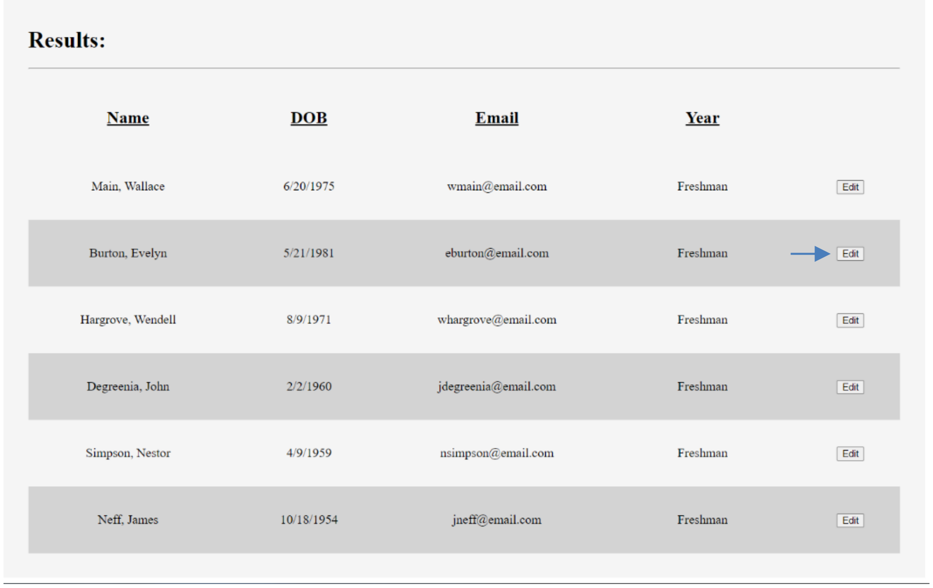
5.4 Edit Account

5.4.1 Description

Allows an administrator to edit an non-administrator account.

5.4.2 Instructions

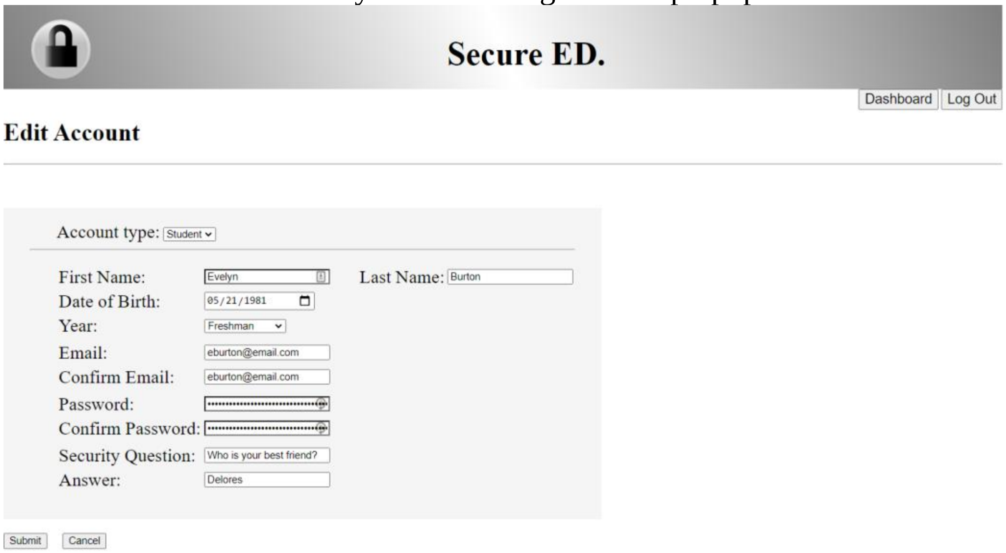
1. Find a user per the instructions in 5.3.2.
2. Select “Edit” to the right of the user’s information.



Results:

Name	DOB	Email	Year	
Main, Wallace	6/20/1975	wmain@email.com	Freshman	Edit
Burton, Evelyn	5/21/1981	eburton@email.com	Freshman	→ Edit
Hargrove, Wendell	8/9/1971	whargrove@email.com	Freshman	Edit
Degreenia, John	2/2/1960	jdegreenia@email.com	Freshman	Edit
Simpson, Nestor	4/9/1959	nsimpson@email.com	Freshman	Edit
Neff, James	10/18/1954	jneff@email.com	Freshman	Edit

3. Make any desired changes to the prepopulated data in the fields provided.



Edit Account

Account type: Student

First Name: Evelyn Last Name: Burton
Date of Birth: 05/21/1981
Year: Freshman
Email: eburton@email.com
Confirm Email: eburton@email.com
Password: ••••••••••••••••••••••••••
Confirm Password: ••••••••••••••••••••••••••
Security Question: Who is your best friend?
Answer: Delores

Note: Don’t forget to confirm any changes to the e-mail or password!

4. Select “Submit”.

6 VULNERABILITIES

6.1 IMPROPER INPUT VALIDATION

<table>
<tr><td colspan="3">*Improper Input Validation* occurs when input or data is not properly checked to ensure that it can be processed correctly.</td></tr>
<tr><td>##</td><td>Instructions</td><td>Result</td></tr>
<tr><td>Ex.1</td><td>1. Go to the Login page and enter the following:
Username: ' OR 1=1;
2. Leave the *Password* field blank.
3. Select “Submit”</td><td>This bypasses the login using a Boolean attack, allowing you to access the admin account.</td></tr>
</table>

6.2 PATH TRAVERSAL

<table>
<tr><td colspan="3">*Path Traversal* occurs when users are able to access parts of the web app that they are not supposed to be able to access, due to a lack of proper pathname restrictions. Potentially, methods can be employed which will allow an attacker to access directories that are not normally public, such as the root folder.</td></tr>
<tr><td>##</td><td>Instructions</td><td>Result</td></tr>
<tr><td>Ex.1</td><td>1. Navigate to Forgot Password
2. Enter the following:
Email: student@email.com
3. Leave the security answer field blank
4. Select “Submit”
5. Retrieve the database’s path from the output: http://localhost:8000/db/persistentconndb.sqlite
6. Navigate to the above link.</td><td>This will download the database, which should not be directly accessible.
Cf. 6.5 – Exposure of Sensitive Information</td></tr>
<tr><td>Ex.2</td><td>1. Navigate to http://localhost:8000/config/config.php</td><td>This will dump debugging info to the screen
Cf. 6.5 – Exposure of Sensitive Information</td></tr>
<tr><td>Ex.3</td><td>1. Navigate to http://localhost:8000/public/create_account.php</td><td>This will allow you to access Create Account without being logged in.
Cf. 6.7 – Improper Access Control</td></tr>
</table>

6.3 CROSS-SITE SCRIPTING

<table>
<tr><td colspan="3"><i>Cross-Site Scripting</i> is a type of code injection-based attack whereby malicious code is injected into a site to be run on the client’s machine.</td></tr>
<tr><td>##</td><td>Instructions</td><td>Result</td></tr>
<tr><td>Ex.1</td><td>1. Go to the Login page and enter:
<i>Username: admin@email.com’--</i>
Leave the password field blank.
2. Select “Submit” and bypass the login via Code Injection.
3. Navigate to User Search
4. Search for all students who are Junior.
5. Click edit to change the user’s information.
6. Set the user’s First Name to the following:
<i>&lt;iframe width="100%" height="166" src="https://www.youtube.com/embed/dQw4w9WgXcQ?autoplay=1" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe&gt;</i>
7. Search for all students who are Junior again</td><td>This puts a video on the search results.</td></tr>
<tr><td>Ex.2</td><td>1. Complete Ex.3 of Section 6.4 – SQL Injection
2. Go to the Login page and enter:
<i>Username: student@email.com’--</i>
3. Leave the password field blank
4. Select “Submit” and bypass the login via Code Injection
5. Navigate to Course Search
6. Leave all input fields blank
7. Select “Search”</td><td>This will return all available courses, including the video script added in 6.4 Ex.3.</td></tr>
</table>

6.4 SQL INJECTION

SQL Injection is an insertion of an SQL query in a field or file where other inputs are expected, potentially revealing sensitive data or modifying the database.		
##	Instructions	Result
Ex.1	1. Navigate to Forgot Password 2. Enter the following: *Email: ‘OR 1=1;* 3. Select “Submit” 4. Enter the following: *Answer: ’ OR 1=1;* 5. Select “Submit” 6. Enter the following: *New Password: '; DROP TABLE User;* *Confirm Password: '; DROP TABLE User;* 7. Select “Submit”	This will drop the User table.
Ex.2	1. Navigate to step 6 of the previous example. Enter the following into both the New Password and Confirm Password fields: *'='a'; INSERT INTO User (UserID, Email, AccType, Password, FName, LName, DOB, Year, Rank, SQuestion, SAnswer) VALUES ('10000000', 'hackerman@getrekt.com','1', '111', 'Hacker', 'Man', '1111-11-11', NULL, NULL, 'get', 'rekt');'-- = ' '='a'; INSERT INTO User (UserID, Email, AccType, Password, FName, LName, DOB, Year, Rank, SQuestion, SAnswer) VALUES ('10000000', 'hackerman@getrekt.com','1', '111', 'Hacker', 'Man', '1111-11-11', NULL, NULL, 'get', 'rekt');'--';*	This will add a new admin account named Hacker Man.
Ex.3	1. Navigate to step 6 of the previous example. 2. Enter the following into both the New Password and Confirm Password fields:	This will add a Section and Course where the course name is a script.

	'='a'; INSERT INTO Section (CRN, Instructor, Course, Semester, SectionLetter, StartTime, EndTime, Year, Location)
	VALUES ('987654321', '927000002', 'CYBR 2201' , 'Fall', 'F', '08:30:00', '11:11:11', '2030','Building A');
	INSERT INTO Course (Code, CourseName)
	VALUES ('CYBR 2201', '<iframe width="100%" height="166" src="https://www.youtube.com/embed/dQw4w9WgXcQ?autoplay=1" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>');	

Ex.4	1. Log into a student account 2. Navigate to Course Search 3. In CRN field, type 123’ OR 1=1;	This will return all courses regardless of input in other fields.
Ex.5	*See Ex.1 of 6.6 – Unrestricted File Upload*	

6.5 Exposure of Sensitive Information

Exposure of Sensitive Information, as the name suggests, is any information in a web app’s code, pages, or file structure, which is sensitive in nature. This can include things like private user information, business information, or any other data that should otherwise be considered confidential.

##	Instructions	Result
Ex.1	1. Using ex.1 of section 6.2 – Path Traversal, acquire the following link from the debugging info dump: *http://localhost:8000/db/persistentconndb.sqlite* 2. Navigate to the above link	This will download the database. *Cf. 6.2 – Path Traversal*
Ex.2	1. Try to login without entering a username or password	This will dump debugging info to the screen.
Ex.3	1. Using ex.1 of section 6.1 – Improper Input Validation, sign in as admin 2. Navigate to Create Account 3. Enter admin@email.com, or some other account that already exists 4. Select “Submit”	This will dump debugging info to the screen.
Ex.4	*See Ex.4 of 6.2 – Path Traversal*	

6.6 UNRESTRICTED FILE UPLOAD

<table>
<tr><td colspan="3"><i>Unrestricted File Upload</i> is an exploit whereby files of an unrestricted type can be uploaded, especially types which were not intended to be uploaded by the web designers.</td></tr>
<tr><td>##</td><td>Instructions</td><td>Result</td></tr>
<tr><td>Ex.1</td><td>1. Go to the Login page and enter:
<i>Username: scienceguy@email.com’</i>--
2. Leave the password field blank
3. Select “Submit” and bypass the login via Code Injection
4. Select “Enter Grades”
5. In notepad, create a txt file with the following:
<i>lol,always remember to sanitize your inputs'); DROP TABLE User;--</i>
6. Change the txt file’s extension to .csv
7. Enter whatever you like for the Course Number, and upload the file
8. Select “Submit”</td><td>This drops the user table.</td></tr>
<tr><td>Ex.2</td><td>1. In the ISWA root folder, navigate to the payloads folder
2. Find the file named HelloWorld.php
3. Go to the Login page and enter:
<i>Username: scienceguy@email.com’</i>--
4. Leave the password field blank
5. Select “Submit” and bypass the login via Code Injection
6. Navigate to EnterGrades
7. Upload HelloWorld.php
8. Navigate to the following: http://localhost:8000/uploads/HelloWorld.php</td><td>This will run HelloWorld.php, which prints “Hello world” to the screen.</td></tr>
</table>

6.7 IMPROPER ACCESS CONTROL

<table>
<tr><td colspan="3"><i>Improper Access Control</i> is when the app does not provide access to a resource appropriate to the access level of the user.</td></tr>
<tr><td>##</td><td>Instructions</td><td>Result</td></tr>
<tr><td>Ex.1</td><td>1. Open 2 instances of the site.
2. Navigate to Forgot Password on both instances.
3. On the first instance, enter the following:
<i>Email: scienceguy@email.com</i>
4. On the first instance, answer the security question:
<i>Answer: Charity Nye</i>
5. On the second instance, enter the following:
<i>Email: student@email.com</i>
6. Back on the first instance, complete the new password</td><td>student@email.com’s password will now be the password you entered for scienceguy@email.com, bypassing student@email.com’s security question entirely. This is because the password is stored in a txt file instead of being session data/post data.</td></tr>
</table>

6.8 IMPROPER AUTHENTICATION

<table>
<tr><td colspan="3"><i>Improper Authentication</i> is when an app does not properly check the claimed identity of an actor.</td></tr>
<tr><td>##</td><td>Instructions</td><td>Result</td></tr>
<tr><td>Ex.1</td><td>1. Go to the Login page and enter:
<i>Username: scienceguy@email.com</i>
<i>Password: 1e031774109ee2e6ac244e778ca579d5199e94fa3753848a3180e9d2e27e8ff7</i>
2. Select “Submit”</td><td>This logs in with a hashed password, instead of requiring the actual password itself.</td></tr>
</table>

6.9 CROSS-SITE REQUEST FORGERY

<table>
<tr><td colspan="3"><i>Cross-Site Request Forgery</i> is an attack whereby a user can be made to do things on their account that they don’t want to do, often via phishing or social engineering.</td></tr>
<tr><td>##</td><td>Instructions</td><td>Result</td></tr>
<tr><td>Ex.1</td><td>1. Set up a link/site that causes a form to submit with post as its method to one of the pages in ISWA/src/ (must supply appropriate data)
2. Issue a request to the server using their session (the user must be logged in for this)
3. This vulnerability is due to not validating where a request is coming from (i.e. from within the web app)

<i>Note: A good example might be the following form:</i>

<i><form action="http://localhost:8000/src/CourseEnrollInsertLogic.php" method="POST" ></i>
<i><input name="courseid" id="courseid" value="111', '1'); DROP TABLE Section;--"></i>
<i><button type="submit">Drop Section</button></i>
<i></form></i></td><td>The provided example form uses an SQL Injection to drop the Section table.</td></tr>
</table>

6.10 CODE INJECTION

<table>
<tr><td colspan="3"><i>Code Injection</i> is any injection of malicious code, whether by Unrestricted File Upload, Cross-Site Scripting, SQL Injection, or any other method, which was not intended by the developers. This is caused by Improper Input Validation.</td></tr>
<tr><td>##</td><td>Instructions</td><td>Result</td></tr>
<tr><td>Ex.1</td><td>See any of the above mentioned in the description.</td><td>—</td></tr>
</table>

7 USER TABLE

Account Type	**Username**	**Password**	**Security Question**	**Security Answer**
Faculty	scienceguy@email.com	Password2	How many siblings do you have?	0
Student	student@email.com	Password5	Where were you born?	Los Angeles, CA
Admin	admin@email.com	Password1	Favorite Relative?	Charity Nye