

OCTO: Software Engineer Program and deploy applications (PRDA)	
SAQA Qualification ID	119458
Intake / year	1-2026
Assessment Name	Summative Assessment
Module NQF Level	6
Credits	25
Open date and time	19 June 2026 • Section B: 17:15
Close date and time	19 June 2026 • Section B: 20:15
Assessment Total Marks	100
Pass percentage	65%

Lecturer Information	
Lecturer	Thaveshan Govender
Lecturer E-mail	thaveshang@aie.ac

IMPORTANT NOTICE

Before submitting your assessment, please ensure that it is signed and that all your personal details are accurately filled in the designated space below (Section in red). Work will not be accepted unless the required sections below are fully completed prior to submission, resulting in a 0% grade for this assessment.

The signature must be your valid handwritten signature and NOT your name and surname typed in the space provided.

Student Number:	461476		
Student name and surname:	John Herbst	Date:	2026/19/06
Student cell phone number:	0604944665		
Student Signature:			
Lecturer name and surname:	Thaveshan Govender	Date:	
Lecturer Signature:			
Assessor name and surname:		Date:	
Assessor Signature:			
Moderator name and surname:		Date:	
Moderator Signature:			

Instructions to Candidates	
1. Read each question carefully. 2. You must answer Section B below the space provided named "STUDENTANSWER". Once completed, the answer sheet, with your answers, must be converted to PDF and uploaded to the student portal. Make sure that all additional information is uploaded in a single Zipped folder. 3. QCTO students who achieve between 55-64% may have a further submission opportunity, subject to the departmental rules. 4. NO LATE submissions will be accepted. (System will automatically close when due date and time is reached). If you encounter any difficulty with the student portal, immediately take a screenshot, log a ticket (attention support), and attach your assessment to the ticket. 5. You must follow the departmental formatting guidelines.	

6. You must write your name, surname, and student number on your script. A failure to do so will result in your script not being marked.
7. By submitting your answer sheet for marking, you declare the following:
 - a. You are familiar with the AIE's policies on academic dishonesty and plagiarism;
 - b. Your submitted is your own original work. Where you have made use of someone else's work, you have duly referenced such work making using of the APA style of referencing;
 - c. You have not allowed, and will not allow any other student to copy your work; and
 - d. You did not copy the work of another student.
8. Students that are dishonest will be subjected to disciplinary procedures.

Assessment Objective/s:

The Formative assessment will cover the following assessment topics:

No	PM number	PS Description
1.	PM-03-PS01	Write code
2.	PM-03-PS02	Debug source code
3.	PM-03-PS03	Deploy applications to attain the desired systems' results

Scope:

This module is crafted to immerse you in the practical aspects of software development, focusing on the critical stages of the application lifecycle. By providing hands-on experience in coding, debugging, and deployment, this module aims to equip you with the skills necessary to develop applications that meet user needs and operate efficiently within their intended environments.

Technical Aspects:

Please verify that all pages are included when downloading this assessment by checking the page numbers at the bottom of the document.

Ensure the following font and size is used in this assessment

- a. Font: Arial
- b. Size: 12
- c. Font colour: Black
- d. Ensure clear and adequate screenshots are taken where requested**

Save and upload the report as a .PDF (**No backgrounds**) with the following naming convention:

- a. StudentName_ StudentSurname _Student no _ModuleCode _SA1

Mark allocation for report

See Mark allocation sheet below

Question 1

(35)

<i>PM No</i>	<i>PS No</i>	<i>PS Description</i>
<i>03</i>	<i>01</i>	<i>Write code</i>

Scenario: Create a Flask-based web application that allows users to register, view, and update their profiles. The app should store data in a JSON file or SQLite database.

Instructions:

- Use Python Flask to:
 - Set up routes and templates.
 - Store user data.
 - Add input validation and in-code documentation.

a)	Deliverables and Mark Breakdown • Displays a user registration form • Processes and stores the form data • Displays stored user data dynamically • Preloads user data into an update form • Use Flask-WTF or manual validation for required fields, email format, etc. • Include at least five (5) meaningful inline or block comments explaining logic • Organize code using Flask best practices	[5] [5] [5] [5] [5] [5] [5]
	ANSWER: (Paste all relevant code and screenshots here)	

Question 2

(30)

<i>PM No</i>	<i>PS No</i>	<i>PS Description</i>
03	02	<i>Debug source code</i>

Scenario: After initial development, you (or your instructor) will introduce at least 3 bugs into the Flask app. Your task is to test and debug the code.

Instructions:

- Create test cases (manual or Pytest).
- Identify bugs and fix them.
- Retest after each correction.

a)	Deliverables and Mark Breakdown:: • Document at least five (5) test cases (valid, invalid, edge inputs) • Describe and locate three (3) bugs (e.g., syntax, logic, validation) • Provide before/after code for each fix • Include pass/fail status and screenshots of terminal/browser results • Write a short summary of the debugging process and lessons learned • Submit updated, bug-free source code					[5] [5] [5] [5] [5] [5]
	ANSWER: (Paste all relevant code and screenshots here)					
	Test Case ID	FEATURE COVERED	INPUT DESCRIPTION	EXPECTED RESULTS	STATUS	

	TC-01	Agent Registration	Codename: Agent 007 Username: John Pass-Key:1508 Password:John@1508	Form submits successfully; database saves it; redirects to Login portal.	Passed	
	TC-02 (Invalid)	Boundary Check	Entering a Pass-Key that is too short (e.g., 123 – less than 6 characters).	System catches it; blocks submission; shows red neon flash error.	Passed	
	TC-03 (Invalid)	Boundary Check	Entering a Password that is too short (e.g., hunter - less than 8 characters).	Form blocks it; flashes password length requirement alert.	Passed	
	TC-04 (Boundary)	Field Empty Sets	Leaving the Username field completely blank.	Security Guard catches the empty box; blocks database insertion.	Passed	
	TC-05 (Security)	Route Protection	Trying to visit http://127.0.0 directly without logging in	System blocks page access; kicks user back to Login terminal.	Passed	

```

34         flash("⚡ Operative created successfully. Proceed to authentication")
35         return redirect(url_for('login'))
36     except Exception:
37         flash("⚠ Warning: Username already taken by another operative.", 'error')
38     finally:
39         connection.close()
40
41     return render_template('register.html')
42
43 @app.route('/users')
44 def users():
45     if 'agent_id' not in session:
46         flash("⛔ Unauthorized Access: Log in required.", 'error')
47         return redirect(url_for('login'))
48
49     connection = get_db_connection()
50     # 5. COMMENT: Query the entire user archive table and order the output data
51     agents_list = connection.execute('SELECT * FROM agents ORDER BY id DESC').fetchall()
52     connection.close()
101     return render_template('register.html')
102
103 @app.route('/users')
104 def users():
105     if 'agent_id' not in session:
106         flash("⛔ Unauthorized Access: Log in required.", 'error')

```

PROBLEMS **1** OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

ration_App> python Spy_agency_App.py
PS C:\Users\herbs\OneDrive\Software engineering\Summative\Surveillance\Surveillance_Ap
ration_App> python Spy_agency_App.py
File "C:\Users\herbs\OneDrive\Software engineering\Summative\Surveillance\Surveillance_Ap
egistration_App\Spy_agency_App.py", line 101
    return render_template('register.html')
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
SyntaxError: 'return' outside function
PS C:\Users\herbs\OneDrive\Software engineering\Summative\Surveillance\Surveillance_Ap
. . . . . ■
Accidental extra indentation indentation error!

```

```

34         flash("⚡ Operative created successfully. Proceed to authentication terminal")
35         return redirect(url_for('login'))
36     except Exception:
37         flash("⚠ Warning: Username already taken by another operative.", 'error')
38     finally:
39         connection.close()
40
41     return render_template('register.html')
42
43 @app.route('/users')
44 def users():
45     if 'agent_id' not in session:
46         flash("🚫 Unauthorized Access: Log in required.", 'error')
47         return redirect(url_for('login'))
48

```

Fixed by Shifting the return function left to align with the if statement.

- Error Triggered: TypeError: The view function for 'register' did not return a valid response.
- Status: PASSED after correction.

```

if len(password) < 8 or len(password) > 100:
    return "❌ Access Denied: Password must be between 8 and 100 characters."
return None

```

AGENCY REGISTRATION

❌ Access Denied: Password must be between 8 and 100 characters.

CODE NAME (e.g., Agent 007)

USERNAME

PASS KEY

SECRET PASSWORD

Blocks Valid secure passwords over 8 characters!

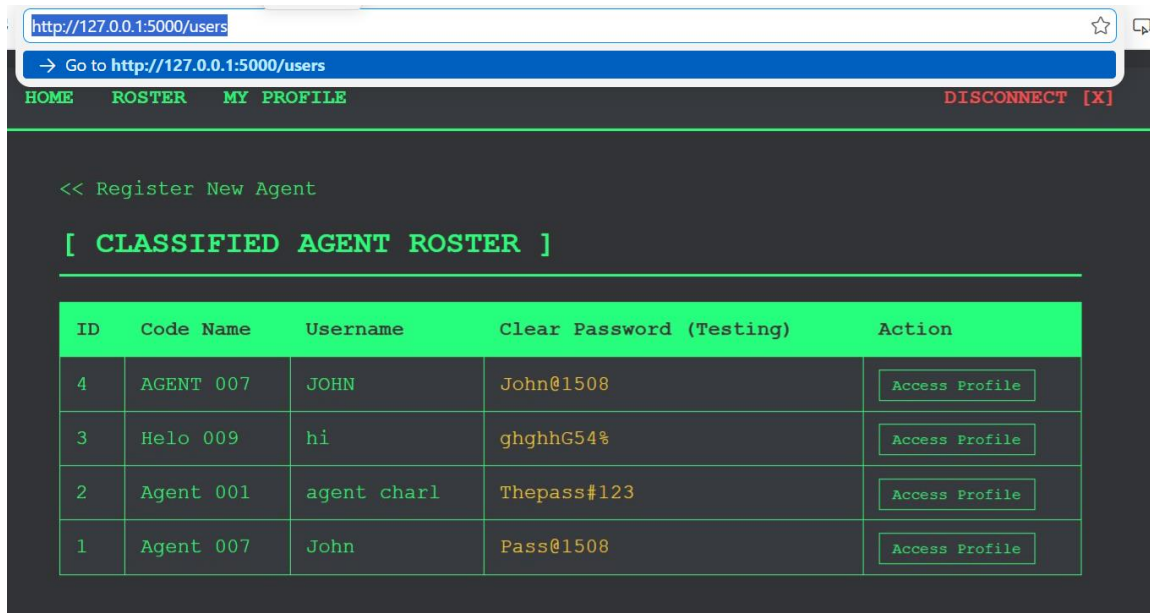
```
if len(password) < 8 or len(password) > 100:
    return "✗ Access Denied: Password must be between 8 and 100 characters."
return None
```

Fixed by turning around, this makes it less than 8 characters not allowed.

- Error: even though I type in a password longer than 8 it still shows a message saying the password must be longer than 8 characters.
- Status: passed after correction

```
@app.route('/users')
def users():

    connection = get_db_connection()
    # 5. COMMENT: Query the entire user archive table and order the output dataset
    agents_list = connection.execute('SELECT * FROM agents ORDER BY id DESC').fetchall()
    connection.close()
    return render_template('users.html', agents=agents_list)
```



http://127.0.0.1:5000/users

→ Go to http://127.0.0.1:5000/users

HOME ROSTER MY PROFILE DISCONNECT [X]

<< Register New Agent

[CLASSIFIED AGENT ROSTER]

ID	Code Name	Username	Clear Password (Testing)	Action
4	AGENT 007	JOHN	John@1508	Access Profile
3	Helo 009	hi	ghghhG54%	Access Profile
2	Agent 001	agent charl	Thepass#123	Access Profile
1	Agent 007	John	Pass@1508	Access Profile

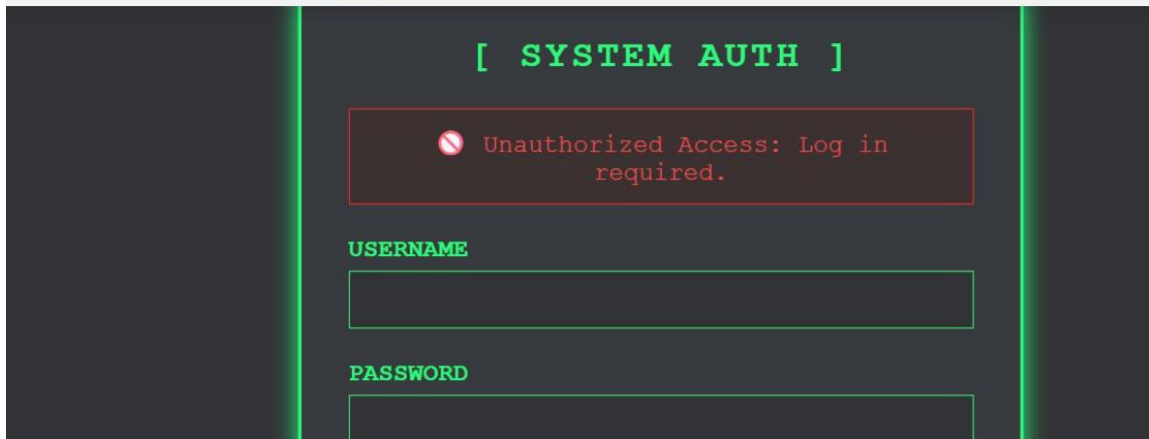
No session validation wall here, Users can just Access the database list without logging in.

```
@app.route('/users')
def users():
    if 'agent_id' not in session:
        flash("⊘ Unauthorized Access: Log in required.", 'error')
        return redirect(url_for('login'))

    connection = get_db_connection()
    # 5. COMMENT: Query the entire user archive table and order the output dataset dynamically
    agents_list = connection.execute('SELECT * FROM agents ORDER BY id DESC').fetchall()
    connection.close()
    return render_template('users.html', agents=agents_list)
```

→ Go to <http://127.0.0.1:5000/>

[HQ - Active Operatives <http://127.0.0.1:5000/users>](http://127.0.0.1:5000/users)
[HQ - Agent Recruitment <http://127.0.0.1:5000/users>](http://127.0.0.1:5000/users)
[Agent Profile Dossier \[http://127.0.0.1:5000/my_profile\]\(http://127.0.0.1:5000/my_profile\)](http://127.0.0.1:5000/my_profile)
[HQ - Central Terminal <http://127.0.0.1:5000/dashboard>](http://127.0.0.1:5000/dashboard)
[HQ - Modify Credentials <http://127.0.0.1:5000/update/4>](http://127.0.0.1:5000/update/4)
[Agent Profile Dossier <http://127.0.0.1:5000/profile/1>](http://127.0.0.1:5000/profile/1)



Insert a wall with session which keeps a cookie in the background with the Agent_id and checks here for it before showing the database

- Error: No error triggered but this is a security vulnerability
- Status: passed after correction

--	--	--

Question 3

(35)

<i>PM No</i>	<i>PS No</i>	<i>PS Description</i>
03	03	<i>Deploy applications to attain the desired systems' results</i>

Scenario: Deploy the working Flask app. Evaluate user feedback through a basic acceptance test.

Instructions:

- Choose a deployment platform.
- Push your Flask app via Git.
- Share a live URL and collect feedback from two users.

a)	Deliverables and Mark Breakdown: • Choose Heroku, Render, or another and explain why • Use Git, Profile, and requirements.txt to deploy Flask app • Submit link to working web app online • Use Google Forms or feedback form to collect user responses • Summarize two (2) users' feedback and suggest 2 enhancements • Provide screenshots of platform dashboard and live app	[5] [10] [5] [5] [5] [5]
	ANSWER: (Paste all relevant code and screenshots here)	

Selected Platform: Render (<https://render.com>)

Justification for Selection:

1. **Native Flask Support:** Render provides specialized Web Service environments that handle Python frameworks out of the box without requiring complex configurations.
2. **Git-Driven Deployment:** It connects directly to GitHub. Whenever new code updates are pushed to the main repository branch, Render automatically runs a clean build and redeploys the site instantly.
3. **Embedded Database Support:** Since SQLite3 relies on a local file architecture (spy_vault.db), Render allows the tiny database file to compile right alongside my application files inside a free standalone container.

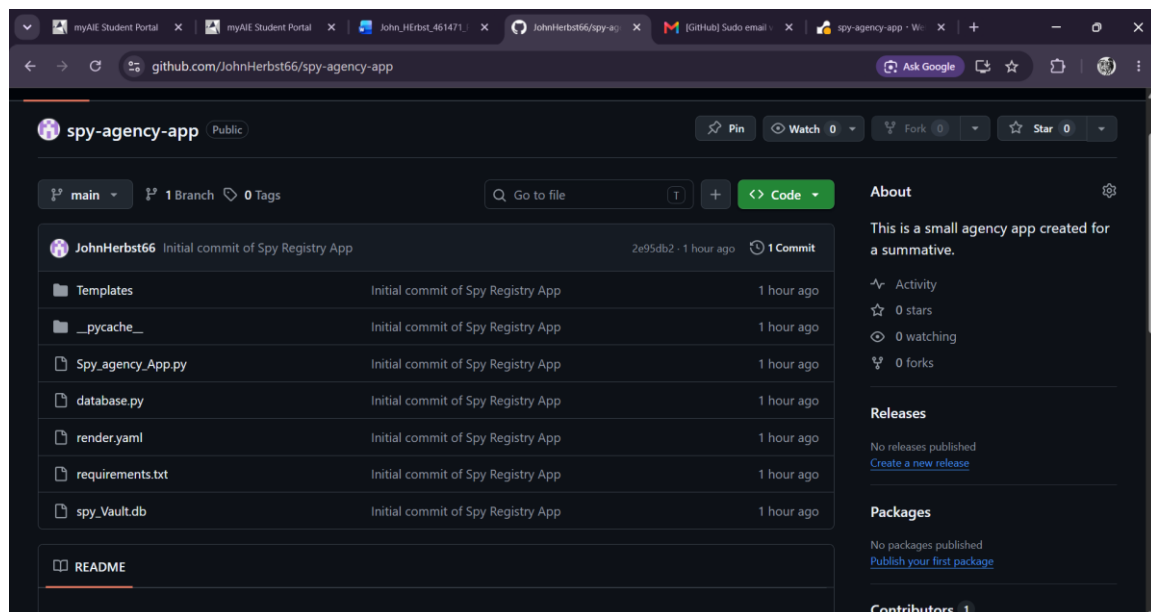
<https://spy-agency-app.onrender.com>

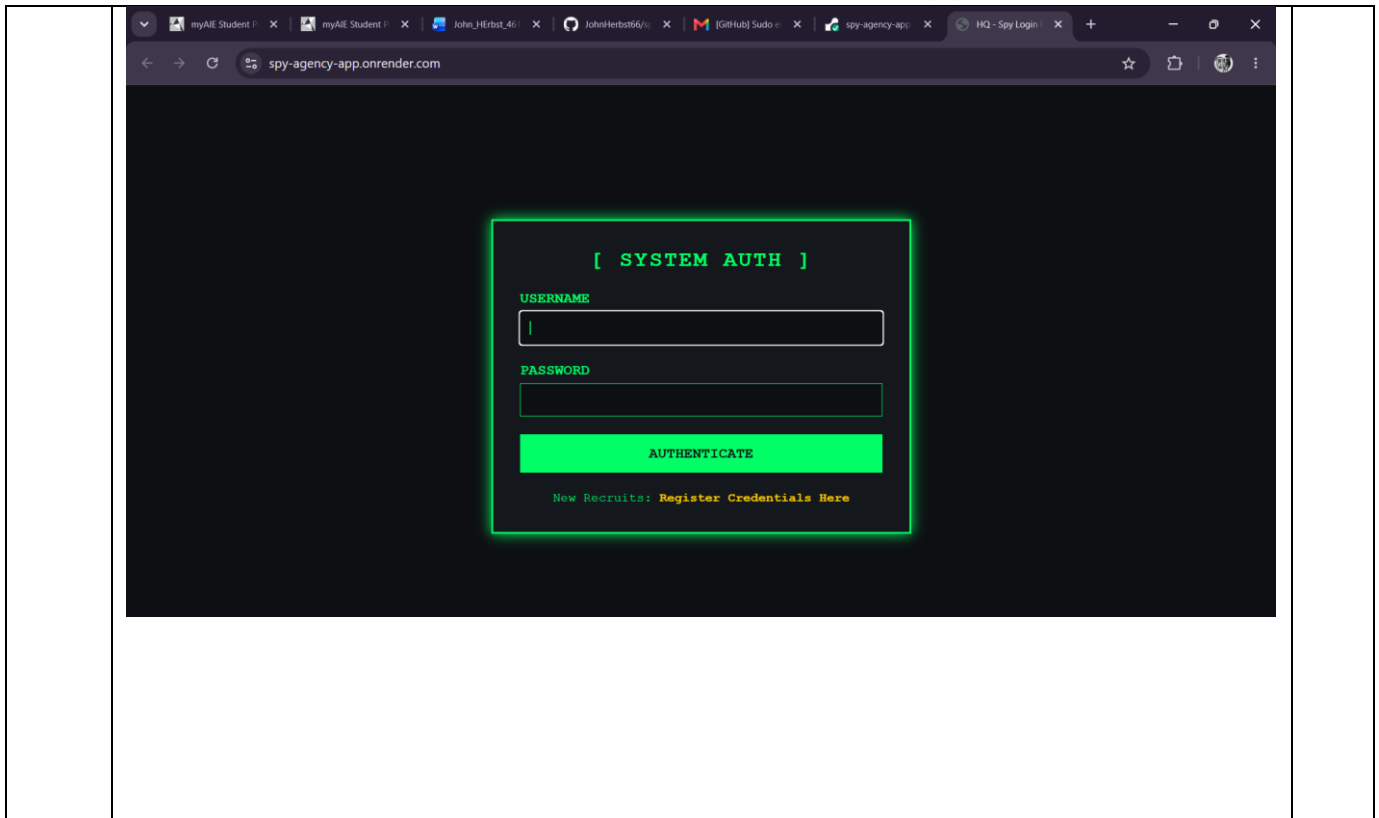
User responses:

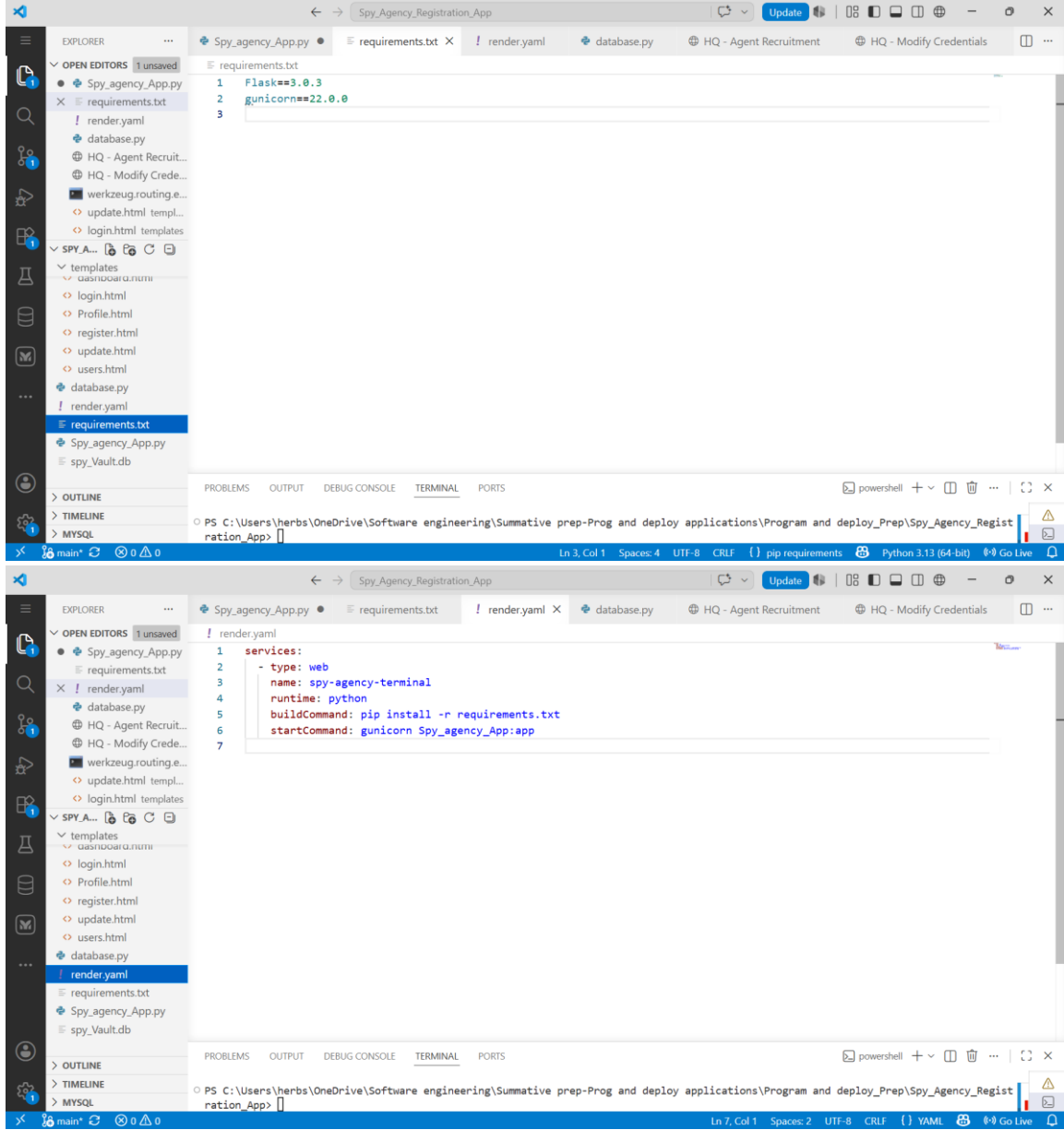
[19:25, 19/06/2026] Albert: When i click on : "register credentials here" it says : "internal server error"

[19:26, 19/06/2026] Albert: If i click on anything it says internal server error

This happned becuae the app expects the create table switch at the bottom of the app.py to build the actual layout boxes, we have to force the app to build the table automaticly



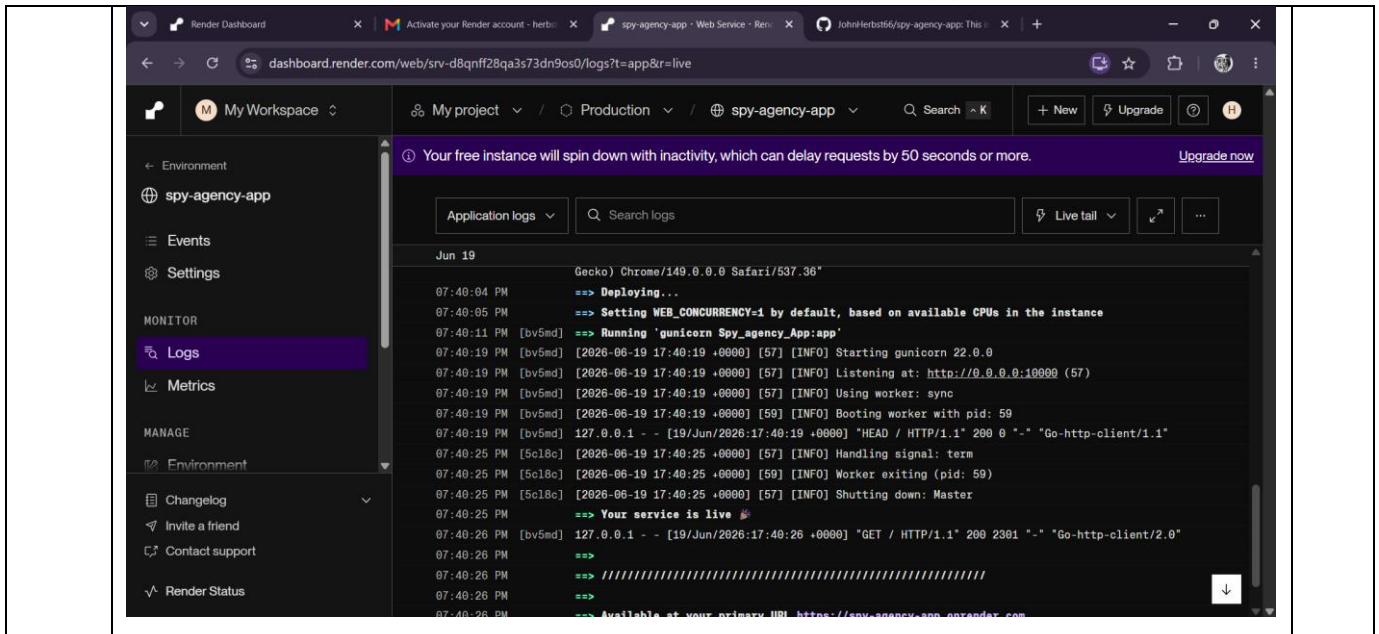
A screenshot of a web browser window. The browser has multiple tabs open, including 'myAIE Student', 'John_H_Erbs1_46', 'JohnHerbst66', '[GitHub] Sudo', 'spy-agency-app', and 'HQ - Spy Login'. The address bar shows 'spy-agency-app.onrender.com'. The main content area is dark-themed and displays a login form titled '[SYSTEM AUTH]' in green. The form has two input fields: 'USERNAME' and 'PASSWORD', both with green borders. Below the fields is a green 'AUTHENTICATE' button. At the bottom of the form, there is a link that says 'New Recruits: Register Credentials Here' in green. The entire form is highlighted with a green rectangular border.



The screenshot displays the Visual Studio Code interface for a project named 'Spy_Agency_Registration_App'. The Explorer sidebar on the left shows the project structure, including files like 'Spy_agency_App.py', 'requirements.txt', 'render.yaml', 'database.py', and various HTML templates. The main editor area shows the 'requirements.txt' file with the following content:

```
1 Flask==3.0.3
2 gunicorn==22.0.0
3
```

Below the editor, the TERMINAL panel is open, showing a PowerShell prompt at the path 'PS C:\Users\herbs\OneDrive\Software engineering\Summative prep-Prog and deploy applications\Program and deploy_Prep\Spy_Agency_Regisration_App>'. The status bar at the bottom indicates the file is 'render.yaml' at line 7, column 1, using UTF-8 encoding and CRLF line endings.



rk allocation for student		
Question	Maximum Mark	Learner mark
Question 1.a	35	
Question 2.a	30	
Question 3.a	35	
Total:	100	

STUDENT ASSESSMENT FEEDBACK

Qualification Name:	OCTO: Software Engineer
Qualification SAQA Number:	119458
Subject Name:	Program and deploy applications
Subject Code:	PRDA
Assessment Name:	Summative Assessment
Assessment Code:	PRDA_SA

Assessment Decision	Competent		Not yet competent	
----------------------------	------------------	--	--------------------------	--

Feedback report	1st Attempt		2nd Attempt	
	C	NYC	C	NYC
Practical module/Practical skill				
PM03				
PS01				
PS02				
PS03				

General feedback to learner Supply comprehensive feedback why learner is found NYC