MODULE NAME:	MODULE CODE:
APPLICATION DEVELOPMENT SECURITY	APDS7311

ASSESSMENT TYPE: POE (PAPER)

TOTAL MARK ALLOCATION: 100 MARKS

TOTAL HOURS: A minimum of 15 HOURS is suggested to complete this assessment

By submitting this assignment, you acknowledge that you have read and understood all the rules as per the terms in the registration contract, in particular the assignment and assessment rules in The IIE Assessment Strategy and Policy (IIE009), the intellectual integrity and plagiarism rules in the Intellectual Integrity Policy (IIE023), as well as any rules and regulations published in the student portal.

INSTRUCTIONS:

1. ***No material may be copied from original sources, even if referenced correctly, unless it is a direct quote indicated with quotation marks. No more than 10% of the assignment may consist of direct quotes.***
2. ***Make a copy of your assignment before handing it in.***
3. *Assignments must be typed unless otherwise specified.*
4. *Begin each section on a new page.*
5. *Follow all instructions on the PoE cover sheet.*
6. *This is an individual assignment.*

Referencing Rubric

Providing evidence based on valid and referenced academic sources is a fundamental educational principle and the cornerstone of high-quality academic work. Hence, The IIE considers it essential to develop the referencing skills of our students in our commitment to achieve high academic standards. Part of achieving these high standards is referencing in a way that is consistent, technically correct and congruent. This is not plagiarism, which is handled differently.

Poor quality formatting in your referencing will result in a penalty **of a maximum of ten percent being deducted from the percentage awarded**, according to the following guidelines. Please note, however, that **evidence of plagiarism in the form of copied or uncited work (not referenced), absent reference lists, or exceptionally poor referencing, may result in action being taken in accordance with The IIE's Intellectual Integrity Policy (0023)**.

Markers are required to provide feedback to students by indicating **(circling/underlining) the information that best describes the student's work.**

Minor technical referencing errors: 5% deduction from the overall percentage – the student's work contains **five or more errors** listed in the minor error's column in the table below.

Major technical referencing errors: 10% deduction from the overall percentage – the student's work contains **five or more errors** listed in the major error's column in the table below**.**

If both minor and major errors are indicated, then 10% only (and not 5% or 15%) is deducted from the overall percentage. The examples provided below are not exhaustive but are provided to illustrate the error

Required: **Technically correct referencing style**	**Minor errors in technical correctness of referencing style** **Deduct 5% from percentage awarded**	**Major errors in technical correctness of referencing style** **Deduct 10% from percentage awarded**
Consistency • The same referencing format has been used for all in-text references and in the bibliography/reference list.	Minor inconsistencies. • The referencing style is generally consistent, but there are one or two changes in the format of in-text referencing and/or in the bibliography. • For example, page numbers for direct quotes (in-text) have been provided for one source, but not in another instance. Two book chapters (bibliography) have been referenced in the bibliography in two different formats.	Major inconsistencies. • Poor and inconsistent referencing style used in-text and/or in the bibliography/ reference list. • Multiple formats for the same type of referencing have been used. • For example, the format for direct quotes (in-text) and/or book chapters (bibliography/ reference list) is different across multiple instances.
Technical correctness Referencing format is technically correct throughout the submission. Position of the reference: a reference is directly associated with every concept or idea. For example, quotation marks, page numbers, years, etc. are applied correctly, sources in the bibliography/reference list are correctly presented.	**Generally, technically correct with some minor errors.** • The correct referencing format has been consistently used, but there are one or two errors. • Concepts and ideas are typically referenced, but a reference is missing from one small section of the work. • Position of the references: references are only given at the beginning or end of every paragraph. • For example, the student has incorrectly presented direct quotes (in-text) and/or book chapters (bibliography/reference list).	**Technically incorrect.** • The referencing format is incorrect. • Concepts and ideas are typically referenced, but a reference is missing from small sections of the work. • Position of the references: references are only given at the beginning or end of large sections of work. • For example, incorrect author information is provided, no year of publication is provided, quotation marks and/or page numbers for direct quotes missing, page numbers are provided for paraphrased material, the incorrect punctuation is used (in-text); the bibliography/reference list is not in alphabetical order, the incorrect format for a book chapter/journal article is used, information is missing e.g. no place of publication had been provided (bibliography); repeated sources on the reference list.
Congruence between in-text referencing and bibliography/ reference list • All sources are accurately reflected and are all accurately included in the bibliography/ reference list.	**Generally, congruence between the in-text referencing and the bibliography/ reference list with one or two errors.** • There is largely a match between the sources presented in-text and the bibliography. • For example, a source appears in the text, but not in the bibliography/ reference list or vice versa.	**A lack of congruence between the in-text referencing and the bibliography.** • No relationship/several incongruencies between the in-text referencing and the bibliography/reference list. • For example, sources are included in-text, but not in the bibliography and vice versa, a link, rather than the actual reference is provided in the bibliography.
In summary: the recording of references is accurate and complete.	In summary, at least **80%** of the sources are correctly reflected and included in a reference list.	In summary, at least **60%** of the sources are incorrectly reflected and/or not included in reference list.

Overall Feedback about the consistency, technical correctness and congruence between in-text referencing and bibliography:

……

……

INFORMATION FOR DEVELOPERS

A PoE is a purposeful collection of student work that exhibits the student's efforts, progress and achievements in one or more areas. The collection of work typically reflects student participation in selecting content, the criteria for judging merit, and evidence of student self-reflection.

PoEs can be a series of separate tasks in different areas of performance, or, can be a series of tasks that progressively build into a single piece of work. The type of PoE used depends on how best to achieve the module outcomes.

When building a series of tasks into a single piece of work, a PoE must be both formative and summative in nature. Students receive feedback on the initial tasks (formative) in order to complete a final task (summative). PoEs also provide students with opportunities to reflect on their learning and this can be included in the PoE tasks by including a reflective component such as asking students how feedback on initial tasks was incorporated into the final task.

All PoEs require students to engage in independent reading/research, to collate, select and synthesise appropriate information, and to produce a well-crafted written piece in which they demonstrate sufficient levels of critical thinking and/or reflection for their respective NQF level.

Please use this information to guide your PoE conceptualisation and development within the context of the Developer Brief you have been given.

Instructions

Background

You have been employed by the National Government to create an inter departmental bulletin board. The government aims to use this bulletin board to post issues that need to be solved by more than one department in conjunction. The issues that will be posted here are of a highly confidential nature – thus this system must:

- **Be developed with very strong security;**
- **Only allow authorised users to utilise the system.**

Part 1 — Plan a secure logon (Marks: 60)

Create a proposal in which you explain the security features that you will employ for your logon. This proposal must be no more than 2000 words and must include a detailed description on how you would implement the following:

1. The registration of new users and your login process in terms of:

    a. HTTP requests and traffic security;
    b. Input validation;
    c. Storing and hashing of passwords;
    d. Maintaining authentication state;
    e. Credential security;
    f. The overall flow of your login process.

2. How you plan to protect your application against (state definition and protection for each):

    a. User name harvesting;
    b. Brute force attacks;
    c. Session jacking;
    d. Session fixation.

List the features you will include as well as the rationale for each feature. You are welcome to use images to better explain your thinking but remember to refer to and reference the image as appropriate.

Part 2 — Develop the backend (Marks: 120)

Develop the backend API as per the lab guide – part 2.

This includes the following:

- Setup MongoDB in the cloud
- Generate SSL certificate and private key
- Get/Create/Delete posts
- Register new user / Login existing user

Ensure

- All calls to the database and API use SSL
- Cross-Origin resource sharing (CORS) is catered for
- Passwords are not stored or compared using free text
- Separate routes for posts and users are implemented and protected
- Login Information is persisted after authentication

You must submit the following:

- A zip file containing your Visual Studio Code source code – remove the "node_modules" folder before zipping as this can be very large in size.

- A Word document containing screenshots for backend API testing using the tool utilised in the lab guide – part 2.

  Ensure each testing case is clearly labelled.

  More details and instructions on how to use the testing tool can be found here https://marketplace.visualstudio.com/items?itemName=humao.rest-client

POE — Develop the frontend (Marks: 100)

Develop the frontend as per the lab guide

This includes the following:

- Display/Create/Delete Posts
- Register new user / Login existing user

Ensure

- Services are used to communicate with the backend API in a secure fashion
- You consider input field validation, password obscuring and sanitization
- Error messages are displayed using a custom component
- Login Information is persisted after authentication
- Additional security packages are implemented – express-brute; helmet; morgan

For the POE you must submit the following:

1. A word document using the following headings, in the order they appear:
    - Cover Page.
    - Table of Contents.
    - Name and student number
    - List of figures (if any)
    - List of references – if you have used any new sources not already referenced in Part 1 (using The IIE Referencing Guidelines).

- A zip file containing your Visual Studio Code source code – remove the “node_modules” folder before zipping as this can be very large in size.

Appendix A - PoE Marking Rubrics

Markers – Please note that the rubrics below must be used to evaluate the students' responses to the relevant assignment questions. Please clearly indicate the specific mark you allocate for each rubric criterion to show how you reached the question total. Also, please provide constructive feedback to ensure students and moderators can follow your marking logic based on the rubric criteria. The most important point is that many markers across different campuses will be marking. The rubric needs to promote the validity and reliability of their marking practices. In addition, there is a separate memorandum that provides additional marking guidance – please ensure you get this from your relevant campus administrator.

FOR DEVELOPERS

What follows are examples of three-level, four-level and five-level marking rubrics. Please choose the typing that best suits this specific assignment and its question(s) and follow that throughout to ensure instrument validity and marking consistency and reliability. Please complete and populate the rubric(s) according to the following requirements:

- **create as many criteria as needed to ensure each rubric aligns directly to the question type, question instructions, case study/scenario, and required mark allocations.**
- **each rubric criterion must be fit-for-purpose in terms of the specific NQF level of this assignment and application of the principles of Bloom's taxonomy (see presentation on creating rubrics).**
- **There are three steps to rubric design. These are (i) a list of criteria for assessing the important goals of the task, (ii) a scale for grading at the different levels of achievement and (iii) a description of each qualitative level.**
- **(i) the components of the task must appear in the rubric.**
- **(ii) there must be a clear delineation of a fail, pass and distinction within both the mark allocation and descriptors of each rubric criterion.**

- **(iii) each rubric criterion must include clear and specific descriptors of all the levels of achievement to ensure these levels are differentiated appropriately – i.e., describe what exactly constitutes a 'poor' versus a 'good' versus an 'excellent' answer for each criterion in the context of this assessment.**
- **The mark allocation ranges of each rubric criterion must be fair and equitable, based on the weighting of each criterion within the assignment.**

Assessment Sheet (Marking Rubric)

Please note: Tear off this section and **attach** it to your work when you submit it/ If this is an online submission, then this information needs to be included in the online submission.

MODULE NAME:	**MODULE CODE:**
APPLICATION DEVELOPMENT SECURITY	**APDS7311**
STUDENT NAME:	
STUDENT NUMBER:	

Marking Criteria	**Poor**	**Developing**	**Good**	**Excellent**	**Feedback**
PART 1					
The registration of new users and your login process in terms of: **(Students should explain why they plan to implement each one.)**	0	1-2	3-4	5	
HTTP requests and traffic security. Student discussed the correct Action method and HTTP security to be implemented to create a more secure login form.	0	1-2	3-4	5	
Input validation. White listening is applied by using RegEx/ any other framework that would successfully validate user input. Empty fields are not allowed.	0	1-2	3-4	5	
Storing and hashing of passwords. A strong cypher is selected, and Salt is added to this cypher.	0	1-2	3-4	5	

Maintaining authentication state. Authentication state is maintained between HTTP requests.	0	1-2	3-4	5	
Credential security. Password Storage: Passwords are stored in a cypher and checked in a cypher. Salt: Salt is added to each password.	0	1-2	3-4	5	
The overall flow of your login process. The flow of the registration process follows the process depicted in the text.	0	1-2	3-4	5	
How you plan to protect your application against: **(State definition and protection for each)**	0	1-2	3-4	5	
User name harvesting Error messages are implemented in a safe and secure manner to protect from User Name Harvesting.	0	1-2	3-4	5	
Brute Force Attacks Log on attempts are limited to protect from Brute Force Attacks.	0	1-2	3-4	5	
Session Jacking Sessions are created when the user logs on. Authentication information is persisted.	0	1-2	3-4	5	
Session Fixation Sessions are created when the user logs on. Authentication information is persisted.	0	1-2	3-4	5	

Final Mark =[Student Mark/60 *100]

Marking Criteria	Poor	Developing	Good	Excellent	Feedback
PART 2					
API: Get all posts	0-4	5-6	7-8	9-10	
API: Create a new post	0-4	5-6	7-8	9-10	
API: Delete a post	0-4	5-6	7-8	9-10	
API: Register new user	0-4	5-6	7-8	9-10	
API: Logon	0-4	5-6	7-8	9-10	
SECURITY: Implement SSL including private key and certificate generation	0-4	5-6	7-8	9-10	
SECURITY: Cater for CORS	0-4	5-6	7-8	9-10	
SECURITY: Store encrypted password	0-4	5-6	7-8	9-10	
SECURITY: Protect routes	0-4	5-6	7-8	9-10	
SECURITY: Persist authentication information	0-4	5-6	7-8	9-10	
DOCUMENTATION: API Testing document	0-9	10-13	14-17	18-20	

Final Mark = [Student Mark /120* 100]

Marking Criteria	Poor	Developing	Good	Excellent	Feedback
POE					
COMPONENT: Display posts including delete	0-4	5-6	7-8	9-10	
COMPONENT: Create Post	0-4	5-6	7-8	9-10	
SERVICE: Posts	0-4	5-6	7-8	9-10	
COMPONENT: Register new user	0-4	5-6	7-8	9-10	
COMPONENT: Logon	0-4	5-6	7-8	9-10	
SERVICE: Auth	0-4	5-6	7-8	9-10	
SECURITY: Input field validation, password obscuring and sanitisation	0-4	5-6	7-8	9-10	
SECURITY: Display custom error message	0-4	5-6	7-8	9-10	
SECURITY: Persist authentication information	0-4	5-6	7-8	9-10	
SECURITY: Implement additional packages – express-brute; helmet; morgan	0-4	5-6	7-8	9-10	

[TOTAL MARKS: 100]