MODULE NAME:	MODULE CODE:
APPLICATION DEVELOPMENT SECURITY	APDS7311

ASSESSMENT TYPE: POE (PAPER AND MARKING RUBRICS)

TOTAL MARK ALLOCATION: 100 MARKS

TOTAL HOURS: 10 HOURS

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 and Property Rights 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. ***Your assignment must be submitted through SafeAssign.***
3. ***Save a copy of your assignment before submitting it.***
4. *Assignments must be typed unless otherwise specified.*
5. *All work must be adequately and correctly referenced.*
6. *This is a group assignment.*
7. *For group assignments, the group may be at most five members.*

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** according to the following guidelines **a maximum of ten percent being deducted from the overall percentage**. 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 errors 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 errors 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 overall percentage.** **Example: if the response receives 70%, deduct 5%. The final mark is 65%.**	**Major errors in technical correctness of referencing style** **Deduct 10% from the overall percentage.** **Example: if the response receives 70%, deduct 10%. The final mark is 60%.**
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. • The correct referencing format for the **discipline** has been used, i.e., either **APA,** OR **Harvard** OR **Law** • 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:

Assignment Topic **(Marks: 100)**

You work as a developer in the internal development team for an international bank. Your team is working on the internal international payment system. Customers often must make international payments via the bank's online banking site. From here, the payments need to be displayed on the payments portal only accessible by dedicated pre-registered staff. Customers need to register for the system by providing their full name, ID number, account number, and password. As you can imagine, this is quite sensitive information that needs to be appropriately secured. After registering, customers need to log on to the website by providing their username, account number and password. Once logged on, the customer should be able to enter the amount they need to pay, choose the relevant currency and choose a provider to make the payment. In South Africa, we mainly make use of SWIFT. They will next be prompted for the account information and SWIFT code for which they wish the payment to be made. The customer will finalise their process by clicking on Pay Now.

From here, the transaction should be stored in a secured database and appear on the international payments portal. Employees of the bank are pre-registered on the portal when they are employed. No registration is necessary; however, they do need to log on to the system to check transactions and forward them to SWIFT for payment. This is done by checking the payee's account information and verifying that the SWIFT code is correct. The employees complete the transaction by clicking a verified button next to each entry and finally by clicking submit to SWIFT – your job ends when that button is clicked.

You will be asked to do additional research and try out additional tools throughout this POE. The following resources will help you:

CircleCI	https://github.com/VCSOIT/APDS2023/tree/master/.circleci
SonarQube	https://github.com/VCSOIT/APDS2023/tree/master/.circleci
MobSF	https://mobsf.github.io/docs/#/
ScouteSuite	https://github.com/nccgroup/ScoutSuite/wiki/Setup
Setting API CircleCI YT video	https://youtu.be/I4CyzX5rhLU
Example Repo	https://github.com/VCSOIT/apds_dev.git

We also encourage you to use ChatGPT/Co-pilot or any other AI tool should you get stuck.

Task 1: Solutions Architecture: Plan your security and test security tools [80 Marks]

It is very important to ensure that you consider the security aspects of every element in the portal. Use any design tool of your choice (as long as it is electronic) to plan the following:

1. The flow of the data in your system. This should include the flow form when a customer logs on until the transaction is sent to the SWIFT system by an employee of the bank. In your diagrams/designs you need to highlight the following:
    a. How you will secure the information provided as input
    b. How you will secure the data in transit
    c. How you plan to harden this portal against:
        i. Session Jacking
        ii. Clickjacking
        iii. SQL injection attacks
        iv. Cross Site Scripting attacks
        v. Man in the Middle attacks
        vi. DDos attacks
2. You have also been asked to test out two new tools that can assist your team in ensuring that your hosting environment and future mobile application are safe.
    a. Download and configure MobSF: https://mobsf.github.io/docs/#/; Use this tool to analyse your OPSC7311 mobile app submitted in semester one to test the application. Use ChatGPT to write a short report on your findings to either support the use of the tool or to argue against it. This report will be served to the security team at the next CR (Change Request) meeting as it is up for consideration as a tech tool to be used by the organisation.
    b. Download and configure ScoutSuite: https://github.com/nccgroup/ScoutSuite
    c. Watch the following video to set up AWS CLI: https://youtu.be/jCHOsMPbcV0
    d. Use the provided user account to run ScoutSuite against the provided AWS instance

Task 2: Create a secure Customer International Payments Portal. [80 Marks]

You must now develop the customer portal and accompanying API (Application Programming Interface) using React or Angular. Ensure that the following is adhered to:

1. Password security is enforced with hashing and salting.
2. Ensure that you Whitelist all input using RegEx patterns.
3. Ensure that all traffic is served over SSL.
4. Ensure that you protect against all the attacks.
5. Include a video when you hand it in to show everything working to your lecturer. Can use OBS to record the video and upload an unlisted video to YouTube.

Task 3: Finalise the project by adding the secure employee International Payments Portal [80 Marks]

You must now develop the customer portal and accompanying API (Application Programming Interface) using React or Angular. Ensure that the following is adhered to:

1. Users are created as no registration process should be possible.
2. Password security is enforced with hashing and salting.
3. Ensure that you Whitelist all input using RegEx patterns.
4. Ensure that all traffic is served over SSL.
5. Ensure you protect against all the attacks listed in the description.
6. Set up a GitHub repository with a circle-ci pipeline to run a SonarQube scan to check for hotspots and code smells.
7. Include a video when you hand it in to show everything working to your lecturer. You can use OBS to record the video and upload an unlisted video to YouTube.

APPENDIX A – Assignment Marking Rubrics

MODULE NAME:	MODULE CODE:
APPLICATION DEVELOPMENT SECURITY	APDS7312

STUDENT NAME :	
STUDENT NUMBER:	

Task 1: Project Planning and Security: Data flow [80 Marks]			
Marking Criteria	**Does not meet the required standard**	**Meets the required standard**	**Exceeds the required standard**
Password Security	• There is little to no discussion regarding hashing and salting algorithms.	• The student provides a basic discussion on how hashing and salting will be applied	• An in-depth explanation that shows evidence of additional research and effort is provided.
[10 Marks]	**0 – 4 Marks**	**5 – 7 Marks**	**8 – 10 Marks**
Data in Transit	• There is little to no discussion regarding the use of TLS and SSL	• The student provides a basic discussion on how TSL and SSL will be used	• An in-depth explanation that shows evidence of additional research and effort is provided.
[10 Marks]	**0 – 4 Marks**	**5 – 7 Marks**	**8 – 10 Marks**
Hardening against attacks	• There is little to no discussion regarding the steps and tools that will be used to ensure that the web application is not vulnerable to the listed attacks	• The student provides a basic discussion on the tools and methods that will be used to harden the web app against the listed attacks.	• An in-depth explanation that shows evidence of additional research and effort is provided.
[30 Marks]	**0 – 9 Marks**	**10-20 Marks**	**20-30 Marks**

Task 1: MobSF and ScouteSuite			
Marking Criteria	**Does not meet the required standard**	**Meets the required standard**	**Exceeds the required standard**
MobSF Implementation **[10 Marks]**	• There is little to no discussion regarding hashing and salting algorithms. **0 – 4 Marks**	• The student provides a basic discussion on how hashing and salting will be applied **5 – 7 Marks**	• An in-depth explanation that shows evidence of additional research and effort is provided. **8 – 10 Marks**
ScouteSuite Implementation **[20 Marks]**	• There is little to no discussion regarding the steps and tools that will be used to ensure that the web application is not vulnerable to the listed attacks **0 – 9 Marks**	• The student provides a basic discussion on the tools and methods that will be used to harden the web app against the listed attacks. **10 – 14 Marks**	• An in-depth explanation that shows evidence of additional research and effort is provided. **15 – 20 Marks**

Task 2: Project Customer Portal			
Marking Criteria	**Does not meet the required standard**	**Meets the required standard**	**Exceeds the required standard**
Password Security	• No or limited password hashing and salting is applied	• Basic password hashing and salting is applied.	• The provided software shows additional research to provide an exceptional implementation
[10 Marks]	**0 – 4 Marks**	**5 – 7 Marks**	**8 – 10 Marks**
Input Whitelisting	• No or limited whitelisting security is applied •	• Basic Rex Patterns that restrict the use of characters known to be used for injection attacks are applied	• The provided software shows additional research to provide an exceptional implementation
[10 Marks]	**0 – 4 Marks**	**5 – 7 Marks**	**8 – 10 Marks**
Securing Data in Transit with SLL	• No or limited SSL is applied	• A valid certificate and key is generated and used to serve web traffic over SSL for the web app	• The provided software shows additional research to provide an exceptional implementation
[20 Marks]	**0 – 9 Marks**	**10-14 Marks**	**15-20 Marks**
Protecting against attacks	• No or limited tools are applied	• Tools such as Express Brute, Helmet etc. present and correctly configured and fully functional.	• The provided software shows additional research to provide an exceptional implementation
[30 Marks]	**0 – 9 Marks**	**10-20 Marks**	**20-30 Marks**
DevSecOps pipeline	• No or a limited pipeline is configured	• A basic DevSecOps pipeline is configured and triggered whenever code is pushed	• The provided software shows additional research to provide an exceptional implementation
[10 Marks]	**0 – 4 Marks**	**5 – 7 Marks**	**8 – 10 Marks**

Task 3: Project Employee Portal			
Marking Criteria	**Does not meet the required standard**	**Meets the required standard**	**Exceeds the required standard**
Password Security	• Lack of general security needed for both portals	• Basic security is applied to both portals.	• The provided software shows additional research to provide an exceptional implementation
[20 Marks]	**0 – 9 Marks**	**10 – 14 Marks**	**15 – 20 Marks**
DevSecOps Pipeline	• No or limited static login information is applied	• Accounts are preconfigured and functional; no registration process is possible.	• The provided software shows additional research to provide an exceptional implementation
[30 Marks]	**0 – 9 Marks**	**10 – 20 Marks**	**20-30 Marks**
Static login	• No or limited static login information is applied	• Accounts are preconfigured and functional; no registration process is possible.	• The provided software shows additional research to provide an exceptional implementation
[10 Marks]	**0 – 4 Marks**	**5 – 7 Marks**	**8 – 10 Marks**
The overall functioning of the web app	• The web app is not functioning or only partially functioning.	• The web application is correctly configured and secured. Information processed on the customer portal appears in the staff portal correctly.	• The provided software shows additional research to provide an exceptional implementation
[20 Marks]	**0 – 9 Marks**	**10 – 14 Marks**	**15 – 20 Marks**

[TOTAL MARKS WILL BE CALCULATED AS A PERCENTAGE OUT OF 100 FOR EACH TASK]