ASSESSMENT 1 BRIEF

MODULE CODE	COM7033
MODULE TITLE	Secure Software Development
MODULE LEADER	Xin Lu
ASSESSMENT TITLE	Software Artefact
WEIGHTING	70%

ASSESSMENT LEARNING OUTCOMES

Upon successful completion of this assessment, you will be able to:

1. Demonstrate an understanding of secure programming concepts and techniques.
2. Apply programming skills to manipulate and analyse data using popular libraries and frameworks.
3. Demonstrate an understanding of the importance of developing software in an ethical, secure, and professional manner.
4. Develop technical software solutions for complex problems.

INSTRUCTIONS

According to the World Health Organization (WHO), stroke is the second leading cause of death globally. In this assignment, you are tasked with developing a secure web application for a local hospital to manage a Stroke Prediction dataset. This system will help doctors handle patient data, including patient registration data, medical data, and lifestyle factors, to predict the likelihood of a stroke.

Dataset

You will work with the Stroke Prediction dataset available on Kaggle: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset/data).The dataset includes the following attributes:

- id: unique identifier
- gender: “Male”, “Female” or “other”
- age: age of the patient
- hypertension: 0 if the patient doesn’t have hypertension, 1 if the patient has hypertension
- ever_married: “No” or “Yes”
- work_type: "Children", "Govt_jov", "Never_worked", "Private" or "Self-employed"
- Residence_type: "Rural" or "Urban"
- avg_glucose_level: average glucose level in blood
- bmi: body mass index
- smoking_status: "Formerly smoked", "Never smoked", "Smokes" or "Unknown"*
- stroke: 1 if the patient had a stroke or 0 if not

*Note: "Unknown" in smoking_status means that the information is unavailable for this patient

Your tasks are:

1. Build a secure web application using Python Flask to store, manage, and retrieve patient data from the Stroke Prediction dataset. The application should:
    - Include a fully functioning web server with a user-friendly interface.
    - Support single or multiple databases (SQLite and MongoDB) for the secure storage and retrieval of patient records.
    - Implement multiple secure programming concepts and techniques such as encryption for patient registration data to protect sensitive information and input validation to ensure data integrity and prevent vulnerabilities etc.
2. Integrated secure development techniques and professional programming skills into your system design to demonstrate proficiency in areas such as:
    - Use data encryption, such as password encryption.
    - Apply input validation methods to prevent security risks.
    - Implement multiple databases for improved data management and security (e.g., SQLite for user data, MongoDB for patient records).
    - Develop unit tests to ensure functionality and code reliability.
    - Showcase version control through GitHub, documenting the progress of your project and demonstrating professional programming practices.

SUBMISSION DETAILS

RELEASE DATE	15 October 2024
SUBMISSION DATE	29 November 2024
DELIVERABLES	Upload your project code to the **GitHub repository** created in the module's GitHub Classroom. You will find your personal repository link on Moodle. https://github.com/CS-LTU/com7033-assignment-XXXX Replace XXXX with your GitHub username. Once created, the repository is ready for your submission.
SUBMISSION DETAILS	Submit your assignment by uploading it to Moodle before midday on the submission date. This deadline will be automatically and strictly enforced. If your submission is late, your grade may be affected. If you have any issues submitting your work, you must email the assessment team and copy in the module leader before the assessment due time. Do not leave your submission until the last minute to avoid any penalties due to problems with the submission portal. Assessment Team: assessment@leedstrinity.ac.uk Module Leader: x.lu@leedstrinity.ac.uk We may ask for a demonstration of your work following the submission. If needed, this will be communicated to you individually via email. Please check your emails regularly.

Your feedback / feed forward and mark for this assessment will be provided within 15 working days.

MARKING CRITERIA

Marks are awarded based on the following criteria. Within each part, aim to complete the work for each section before moving on to the next. The following banded marking scheme is used:

Exceptional Distinction	100/95/92		*Pass*	58/55/52
	88/85/82		*Bare Fail*	48/45/42
Distinction	78/75/72		*Fail*	38/35/32
Merit	68/65/62			

If you have completed all the preparatory exercises and attended your classes, the estimated additional time required to PASS this assessment is approximately 15 hours.

To obtain a PASS mark (50%), you must have:	○ Develop a basic web application with a basic user interface ○ Use a Single database (Store user registration data in either SQLite or MongoDB). ○ Apply at least one security feature, such as basic input validation or password encryption. ○ Use GitHub to track code progress with at least one commit.
To obtain a MERIT mark (60%), you must have (in addition to the above):	○ Develop fully functional web app with a more advanced interface. ○ Implement multiple databases (SQLite for user data e.g., registration details and MongoDB for patient-related data e.g., medical records). ○ Demonstrate the ability to add, update, and delete records in both databases. ○ Implement two security features (e.g., input validation and password hashing). ○ Four GitHub commits with meaningful messages. ○ Partial code comments. ○ Implement a single unit test.
To obtain a DISCTINCTION mark (70%), you must have (in addition to the above):	○ Develop a fully working web application with a customized, professional user interface. ○ Use multiple databases with interconnected data structures to enhance data security and query efficiency. ○ Implement more than two secure development techniques to show excellent understanding of secure programming concepts. ○ Make at least 8 Github commits with detailed messages to demonstrate the development process over time. ○ Implement multiple tests across different features. ○ Fully documenting code.
To obtain an EXCEPTIONAL DISTINCTION mark (80%), you must have (in addition to the above):	○ Demonstrate professional code development with a clear focus on code efficiency and scalability. ○ Use of third-party APIs or libraries for additional security. ○ Compréhensive documentation (installation, usage, API references). ○ Fully test the entire application using unit tests, integration tests, and end-to-end tests to ensure correctness and reliability. ○ Regularly use GitHub with a well-maintained commit history, clear branches, and pull requests to demonstrate collaboration readiness and continuous development.

ACADEMIC MISCONDUCT

Academic Misconduct includes all forms of academic dishonesty, whether intentional or accidental, that compromise the integrity of the University's assessment processes. It is essential that you review our Student Academic Misconduct Policy to understand the guidelines and the serious consequences that may arise if they are not followed.

HELP AND SUPPORT

- Please use the module handbook and the School of Computer Science Community Teams site as a source of information. Do try and find the answer out yourself before reaching out for help.
- Support will be provided via Microsoft Teams and email during office working hours. You can also ask questions during your timetabled sessions. You may request a one-to-one meeting with a tutor during their published office hours.
- The Student Support team are available seven days a week to support you in all aspects of student life. This could be for support relating to your course, your accommodation or for more general advice such as relationships or your wellbeing. Log in to the LTU app to access support services.
- The full set of university guidelines on assessments, deadlines, and extensions is available on the LTU app, please familiarise yourself with the documentation.