Leeds Trinity University

Introduction to Assignments

Dr. Xin Lu

x.lu@leedstrinity.ac.uk

COM7033 Secure Software Development

Leeds Trinity University

Assessment Learning Outcomes

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.

Assessments

Leeds Trinity University

- Assessment 1: Software Artefact (70%)
- Assessment 2: Demonstration (30%)

Assessment 1: Software Artefact

- Develop a Secure Web App for a hospital to store patient information (Patient registration information & Stroke Dataset)
- Stroke dataset can be download from Kaggle:

https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset/data

Dataset

A	B	C	D	E	F	G	H	I	J	K	L
id	gender	age	hypertensi	heart_dise	ever_marri	work_type	Residence_	avg_glucos	bmi	smoking_st	stroke
9046	Male	67	0	1	Yes	Private	Urban	228.69	36.6	formerly sn	1
51676	Female	61	0	0	Yes	Self-emplo	Rural	202.21	N/A	never smok	1
31112	Male	80	0	1	Yes	Private	Rural	105.92	32.5	never smok	1
60182	Female	49	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
1665	Female	79	1	0	Yes	Self-emplo	Rural	174.12	24	never smok	1
56669	Male	81	0	0	Yes	Private	Urban	186.21	29	formerly sn	1
53882	Male	74	1	1	Yes	Private	Rural	70.09	27.4	never smok	1
10434	Female	69	0	0	No	Private	Urban	94.39	22.8	never smok	1
27419	Female	59	0	0	Yes	Private	Rural	76.15	N/A	Unknown	1
60491	Female	78	0	0	Yes	Private	Urban	58.57	24.2	Unknown	1
12109	Female	81	1	0	Yes	Private	Rural	80.43	29.7	never smok	1
12095	Female	61	0	1	Yes	Govt_job	Rural	120.46	36.8	smokes	1
12175	Female	54	0	0	Yes	Private	Urban	104.51	27.3	smokes	1
8213	Male	78	0	1	Yes	Private	Urban	219.84	N/A	Unknown	1

Leeds Trinity University

Tasks:

Build a Web application using Python Flask

- User-friendly interface
- Support single or multiple databases (SQLite and/or MongoDB)
- Implement multiple secure programming concepts and techniques (such as input validation, hash passwords etc.)
- Unit tests
- Version Control

Leeds Trinity University

Submission:

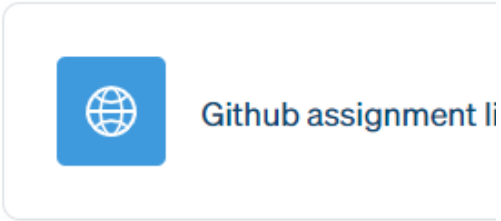
- 29th November 2024
- GitHub repository from Github classroom

https://github.com/CS-LTU/com7033-assignment-XXXX

Github link

Leeds Trinity University

Assessments



Join the classroom:

CS-LTU-classroom-COM7033-SecureSoftwareDevelopment

To join the GitHub Classroom for this course, please select yourself from the list below to associate your GitHub account with your school's identifier (i.e., your name, ID, or email).

Can't find your name? Skip to the next step →

Leeds Trinity University

Github link

GitHub Classroom GitHub Education

You're ready to go!

You accepted the assignment, **COM7033_Assignment**.

Your assignment repository has been created:

https://github.com/CS-LTU/assignment-xinluleeds

We've configured the repository associated with this assignment.

Note: You may receive an email invitation to join CS-LTU on your behalf. No further action is necessary.

Marking Scheme

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.

Assessment 2: Demonstration

- 15-minute recorded presentation to demonstrate your developed secure web application for Assessment 1.
- 29th November 2024
- Using **Panopto** to record your video.
- Submission files
  - **Panopto link** with open access to Moodle
  - **PPT slides** used in your presentation

Key requirements

PPT slides:

- System design overview
- Security features
- Ethical and professional development related to secure software development

System Demonstration:

- System walkthrough
- Security implementations
- Database handing

Highlight Professional development practice

- Version control
- Unit tests

Marking Scheme

To obtain a PASS mark (50%), you must have:	○ Present clear but simple slides explaining the system design. ○ Show a basic understanding of secure software development principles and key security practices. ○ Explain ethical considerations and how professional practices were integrated into development. ○ Demonstrate the web application with at least one core feature working. ○ Explain the basic structure of the database and how data is stored and accessed. ○ Explain the use of version control (e.g., GitHub) and show a basic understanding of its importance.
To obtain a MERIT mark (60%), you must have (in addition to the above):	○ Fully demonstrate the system with all key features working correctly. ○ Provide well-organized slides explaining system architecture, security features, and user interface. ○ Offer a good discussion of ethical considerations and data privacy. ○ Provide a detailed explanation of the database design, showing an understanding of data relationships. ○ Explain the code to demonstrate a good understanding of its functionality and structure ○ Show a good understanding of version control with multiple, well-structured commits on GitHub. ○ Demonstrate a basic understanding of unit testing practices.
To obtain a DISCTINCTION mark (70%), you must have (in addition to the above):	○ Provide well-organized slides with detailed explanations of system architecture, security features, and user interface design. ○ Provide a detailed discussion of ethical concerns and compliance with industry standards. ○ Demonstrate professional use of version control. ○ Explain key code sections clearly, describing their purpose and contribution to the system. ○ Explain unit tests, their importance in software development, and how they are applied to ensure system reliability.
To obtain an EXCEPTIONAL DISTINCTION mark (80%), you must have (in addition to the above):	○ Deliver professional slides that clearly cover system design, architecture, security, and database management. ○ Provide a comprehensive discussion of ethical and legal concerns, referencing industry standards. ○ Offer an in-depth explanation of code sections and their contributions. ○ Provide a detailed discussion of unit tests, their validation role, and contribution to software quality.

Leeds Trinity University

Thank you!