

Configuration Manual

MSc Research Project
Cybersecurity

Neha Swaminathan
Student ID: 23425741

School of Computing
National College of Ireland

Supervisor: Prof. Eric Gyamfi

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Neha Swaminathan
Student ID: 23425741
Programme: M.Sc. Cybersecurity **Year:** 2025
Module: MSc Research Project
Lecturer: Eric Gyamfi
Submission Due Date: 11/12/2025
Project Title: Automated Cybersecurity Risk Assessment Tool for SMEs
Word Count: 390 **Page Count:** 2

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:

A handwritten signature in black ink, appearing to be "Neha", written over a light blue rectangular background.

Date: 11/12/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Neha Swaminathan
Student ID: 23425741

The configuration manual provides a detailed guide to replicate the **Automated Risk Assessment Tool for SMEs** project which is a user-friendly application to conduct self-assessments and generate risk registers with custom mitigations, visualization dashboard. This document contains details on the tools, structure and working setup to execute this project from the start.

1. System Requirements

- Operating System: Windows 11 was used for this project but can be implemented in Linux, MacOS and Windows.
- Hardware: Minimal specifications with 4GB RAM for database functions and connecting to the local server.
- Internet: Critical for the HTML application to connect with the server and run throughout the process.

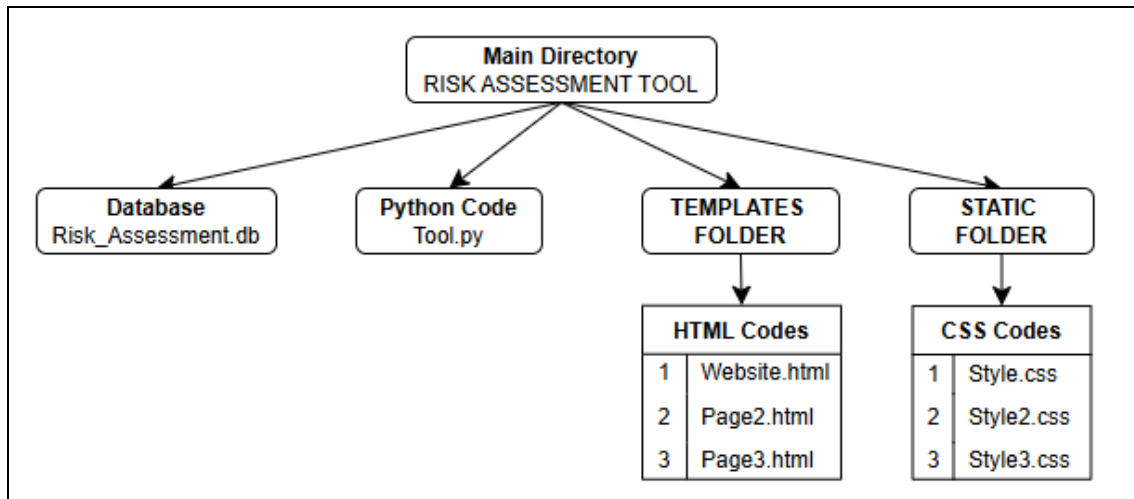
2. Application Setup

Python was used to develop the project for HTML page routing and database processing, running the application in a virtual environment with Flask installed (*pip install flask*). This library is critical for SQL, OS and HTML functions to be integrated in the code. The code is executed to run through python and the risk assessment tool is accessed in the web browser (<http://127.0.0.1/5000/>) with the database and HTML, CSS files stored in the current working directory. Open-source libraries like Chart.js (heatmaps, bar charts), html2canvas (downloadable export to PNG) and SheetJS (downloadable export to XLSX) are incorporated in the code. Application versions used for implementation:

- HTML5
- CSS3
- Python 3.12.7
- SQLite 3.13.1
- Kleopatra GPG 3.3.0

3. Folder Structure

The python code and database are stored in the main directory with the static and templates folder. The templates folder contains the main questionnaire page code as website.html, the automated risk register as page2.html and the mitigation table, visualization dashboard as page3.html. The static folder consists of the CSS style codes associated with the 3 HTML pages. The directory (os.chdir) would have to be modified to the local path for implementation.



4. Database Setup

SQLite was utilised to create the tables within the Risk_Assessment database. Tables such as questions and risk_score contain a fixed number of rows with prepared details, supporting the responses and risk register calculations. Providing an entry through the application will populate a response in the risk_register and responses table.

- a) CREATE TABLE **questions** (question_id INTEGER PRIMARY KEY, question_number TEXT, risk_category TEXT, risk_description TEXT);
- b) CREATE TABLE **responses** (response_id INTEGER PRIMARY KEY AUTOINCREMENT, company_name TEXT, industry TEXT, q1 TEXT, q2 TEXT, q3 TEXT, q4 TEXT, q5 TEXT, q6 TEXT, q7 TEXT, q8 TEXT, q9 TEXT, q10 TEXT, q11 TEXT, q12 TEXT, q13 TEXT, q14 TEXT, q15 TEXT, q16 TEXT, q17 TEXT, q18 TEXT, q19 TEXT, q20 TEXT, submission_date DATETIME DEFAULT CURRENT_TIMESTAMP);
- c) CREATE TABLE **risk_score** (score_id INTEGER PRIMARY KEY AUTOINCREMENT, question_number TEXT, response TEXT, answer TEXT, likelihood INTEGER, impact INTEGER, mitigation TEXT, duration TEXT);
- d) CREATE TABLE **risk_register** (risk_id INTEGER PRIMARY KEY AUTOINCREMENT, question_number TEXT NOT NULL, risk_category TEXT, risk_description TEXT, answer TEXT, likelihood TEXT, impact TEXT, risk_score INTEGER, severity TEXT, mitigation TEXT, duration TEXT, company_name TEXT, id INTEGER, submission_date DATETIME DEFAULT CURRENT_TIMESTAMP, selected_strategy TEXT DEFAULT "", residual_severity TEXT DEFAULT "", residual_score INTEGER DEFAULT 0);
- e) CREATE TABLE **residual_risk** (residual_id INTEGER PRIMARY KEY, question_number TEXT, response TEXT, risk_score INTEGER, accept INTEGER, avoid INTEGER, terminate INTEGER, transfer_risk INTEGER);

5. Application Testing

Setting up all the listed details will prepare the python code for a successful execution. Data entries can be tested to check the workflow, button functionality and validate the accuracy of the risk assessment.