National College of Ireland

Configuration Manual

MSc Research Project
MSc Cyber Security

Urmila Yelmar
Student ID: X23267992

School of Computing
National College of Ireland

Supervisor: Prof. Joel Aleburu

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Urmila Shridhar Yelmar
Student ID:	X23267992
Programme:	MSc Cybersecurity **Year:** 2024-2025
Module:	Research Project
Lecturer:	Prof. Joel Aleburu
Submission Due Date:	11-08-2025
Project Title:	Smart Data Masking using AI in Banking Transactions.
Word Count:	**544** **Page Count: 3**

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: Urmila Shridhar Yelmar

Date: 11-08-2025

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

Urmila Yelmar
Student ID:x23267992

Smart Data Masking using AI in Banking Transactions
Environment: Flask Development Server
Main Components: Flask APIs, Random Forest, Autoencoder

---

1. What This Project Does

This application uses machine learning to help banks automatically mask sensitive customer data based on the **predicted risk level** of a transaction.
It works like this:

1. You submit transaction details through a simple web form.
2. The AI model predicts whether the risk is **Low**, **Medium**, or **High**.
3. Depending on the risk, the app masks part or all the sensitive data.
4. Every request is logged into a CSV file for review.
5. The system can display real-time stats and let you download the logs.
6. If IP location lookup fails, it assigns a **geolocation** instead of showing "Unknown."

---

2. Tools & Technologies Used

Component	Tool / Service
Programming Language	Python 3.10+
Framework	Flask 2.3
ML Libraries	Scikit-learn 1.3, Joblib 1.3, NumPy 1.26
Data Handling	Pandas 2.1
API Testing	Requests 2.31, Flask's built-in debug mode
Deployment	Flask Development Server (local)
Monitoring	Local CSV logs (~/flask_logs/logs.csv)
Security Add-ons	Bootstrap 5.3 (UI), masking logic

---

3. Project Folder Structure

```
flask_project/
|
 ├── appp.py              # Main Flask app
 ├── masked_risk_model.pkl # Trained ML model
```

```
|
├── templates/          # HTML templates
|   ├── forms.html      # Input form
|   ├── stats.html      # Stats dashboard
|
├── static/             # CSS, JS, Images
|
└── flask_logs/         # Log folder (created automatically)
    └── logs.csv
```

---

4. Requirements

Python version

- Use **Python 3.10+** (3.11 recommended)

Install dependencies

Open a terminal in your project folder and run:

pip install flask==2.3.3 pandas==2.1.3 numpy==1.26.2 scikit-learn==1.3.2 joblib==1.3.2 requests==2.31.0

---

5. Configuration Details

Model File

- masked_risk_model.pkl must be placed in the root folder.
- The app will not start without it.

Log File Location

- Logs are stored in:
  - Windows: C:\Users\<YourName>\flask_logs\logs.csv
  - Linux/Mac: /home/<user>/flask_logs/logs.csv
- Created automatically if missing.

Risk Level Logic

- **High Risk:** Real count from actual logs.
- **Medium Risk:** Real count from actual logs.
- **Low Risk:** Real count from actual logs.

Location Fallback

- If IP lookup fails, location is chosen randomly from a list of real cities and countries.

---

7. How to Run

1. Open your terminal / command prompt.
2. Navigate to the project folder:
3. cd C:\dell\flask_project
4. Start the Flask app:
5. python appp.py
6. Look for this in the terminal:
7. Running on http://127.0.0.1:5000
8. Open that link in your web browser.

---

7. Main Pages

URL	What it does
/	Main form for risk prediction & masking result
/stats	Shows recent logs & risk count summary
/download_csv	Downloads all logs as a CSV

---

8. Logs

The CSV log contains:
timestamp,user_role,device_type,access_time,ip,city,country,ip_risk_score,txn_amount,location_score,field_value,prediction,masked_field,mask_reason,risk_level

---

9. Stopping the App
Press **CTRL + C** in the terminal.

---

10. Common Problems & Fixes

Problem	Cause	Fix
Model file not found	Missing masked_risk_model.pkl	Add the file to root folder
Cannot write logs	No permission to ~/flask_logs	Run terminal as Administrator
Location shows "Unknown"	IP lookup failed	Fallback now gives a random realistic location
Flask server restarts repeatedly	Syntax error in code	Check terminal for error message & fix

---

DEPLOYMENT

1. Prepare a virtual environment

- ✓ python3 -m venv venv
- ✓ source venv/bin/activate
- ✓ pip install -r requirements.txt

2. Run the startup script

- ✓ chmod +x run_server.sh
- ✓ ./run_server.sh
- ✓ The app will run on http://127.0.0.1:5000 by default.