Deployment Plan for Phishing Simulation Project

Fadys Responsibilities:
Handles all frontend work, including the fake login form, QR code generator, and email template. Also takes care of Docker setup for the frontend, helps with CI pipeline configuration, and writes supporting scripts

Williams Responsibilities:
Builds and manages the backend, including the API, submission logging, and IT alert system. Also handles backend Docker setup, CI/CD deployment, and supports scripting and testing tasks.

Phase 1: Repository Initialization and Version Control ()

Tasks:

- Set up GitHub repo with main, dev, and individual feature branches
- Add .gitignore, README.md, CONTRIBUTING.md, and LICENSE
- Establish clear Git branching strategy and naming conventions
- Create initial project structure: frontend/, backend/, scripts/, docs/, deploy/

Goal: Fully initialized and documented repo with working Git practices

Responsibility:

- Fady: Initial repo setup, documentation
- William: Branching strategy, rules, and structure validation

Phase 2: Frontend and Phishing Flow Prototype ()

Tasks:

- Build HTML/CSS fake login form with basic styling and validation
- Create QR code and phishing link generation tool (scripted)
- Design email template in HTML with embedded phishing link

Goal: User can click link or scan QR code, land on fake form, and submit credentials

Responsibility:

- Fady: Fake form UI, QR code generation, email template
- William: Shared work on scripting tool to generate and encode phishing links

Phase 3: Backend Submission Logger ()

Tasks:

- Build API endpoint to accept POST requests with submitted credentials
- Log submission with timestamp, IP, user agent, and email
- Store logs in file (JSON)

Goal: Successful form submission posts to backend and is logged

Responsibility:

- William: logging logic
- Fady: Assist with testing connection between frontend and backend

Phase 4: Containerization ()

Tasks:

- Write Dockerfile for frontend phishing site
- Write Dockerfile for backend logger
- Compose both using docker-compose.yml with proper networking

Goal: Local environment can be brought up with docker-compose up

Responsibility:

- Fady: Frontend container + docker-compose setup
- William: Backend container + volume setup

Phase 5: IT Alerting System ()

Tasks:

- Implement email alert in backend upon new submission
- Dynamic config of IT destination
- Include detailed alert: timestamp, IP, user agent, email

Goal: Simulated IT alert is triggered live upon new form submission

Responsibility:

- William: Backend alert
- Fady: Format alert content, support testing

Phase 6: CI/CD Automation ()

Tasks:

- Set up GitHub Actions to lint code and build containers on PR
- Add simple test: ensure phishing form loads and API logs correctly
- Auto-deploy to self-hosted test environment when main is updated

Goal: Pushing code to main runs CI and redeploys self-hosted project

Responsibility:

- Fady: Write frontend and integration tests, CI pipeline basics
- William: Deployment script and GitHub Actions runner config

Phase 7: Final Demo Prep ()

Tasks:

- Record logs from simulated submissions to demo system behavior
- Polish frontend: clean layout, realistic copy, responsive behavior
  Write final README with clear instructions to clone and run the system
  Prepare and rehearse demo script: show QR scan, phishing, logging, alert

Goal: Project runs smoothly end-to-end and can be cloned and demoed by others

Responsibility:

- Fady: UI polish, demo scripting
- William: Backend behavior polish, ensure IT alerts simulate clearly

---