**1. Problem Overview**

**Project Title: Smart Print Automation System**

**Domain & Sub-Domain: AI - Open Innovative**

**Problem Statement:**

* **What is the problem you are solving?** The current process of getting document printouts from a Xerox shop is time-consuming as it requires sending files via WhatsApp or email, which may lead to privacy risks.
* **Who are the affected stakeholders?** Students, professionals, illiterates, and people in urban areas who face difficulties or delays in printing documents.
* **What is the real-world impact of this issue?** Time delay, privacy risks, and dependency on shopkeepers for response.

**Objectives:**

* Enable quick document submission through QR code scanning.
* Eliminate the need for WhatsApp/email communication.
* Ensure secure and centralized document submission.

**Target Audience:**

* Students, professionals, and individuals with limited technical knowledge.

**2. Solution Design & Implementation**

**Technology Stack:**

* **Programming Language:** Python
* **Framework:** Flask
* **Libraries:**
  + Flask-Uploads (for file upload)
  + CUPS (for direct printing)
  + ReportLab (for PDF invoice generation)
  + Qrcode (for QR code generation)
  + OS, Shutil (for file management)
  + SQLite3 (for print history tracking)
* **Other Tools:** Tkinter (for GUI in desktop app)

**Architecture Diagram:**

1. User uploads file via QR code.
2. File is sent to Flask API.
3. Flask API communicates with CUPS for printing.
4. File gets printed directly without WhatsApp or Email.

**Core Features:**

1. **QR Code Generation:** Shopkeeper generates a unique QR code linked to their shop.
2. **File Upload:** Users upload files by scanning the QR code.
3. **Direct Printing:** The uploaded file is directly sent to the printer.
4. **Print History:** Track previous print requests using SQLite3.
5. **Secure Communication:** No third-party involvement ensuring privacy.

**Implementation Details:**

* **Step 1:** Develop a desktop application using Tkinter for the shopkeeper.
* **Step 2:** Build Flask API for handling file uploads and printing.
* **Step 3:** Integrate CUPS with Flask to facilitate direct printing.
* **Step 4:** Use SQLite3 to store print history.
* **Step 5:** Generate QR codes that contain shop details and server URL.

**Code Snippets:**

import qrcode

qr = qrcode.make("http://localhost:5000/upload")

qr.save("shop\_qr.png")

**3. Testing & Validation**

**Testing Strategy:**

* Unit testing for API endpoints.
* Integration testing for file upload and direct printing.
* User trials for QR code scanning and printing.

**Metrics for Validation:**

* **Accuracy:** Successful file printing without manual intervention.
* **Response Time:** Minimal time between file upload and print.
* **User Feedback:** Ease of use for shopkeepers and users.

**Results & Improvements:**

* Reduced time consumption from file upload to print.
* Improved data privacy by eliminating third-party communication.

**4. Deployment & User Guide**

**Deployment Process:**

* Install Python, Flask, and CUPS on the desktop system.
* Run the Flask server (python app.py).
* Generate QR code using the desktop application.
* Connect the printer via CUPS.

**User Manual:**

* **Shopkeeper:** Generate QR code and display it in the shop.
* **User:** Scan QR code, upload file, and collect printout.

**Security Measures:**

* No personal data is shared via WhatsApp/Email.
* Direct file transfer to the printer reduces interception.

**5. Future Enhancements & Impact**

**Scalability Plans:**

* Implement cloud storage for file access.
* Provide mobile app for file submission.

**Future Features:**

* Payment gateway integration.
* Notification to users once the print is ready.

**Societal Impact:**

* Time-saving solution in emergencies (like exam hall entry).
* Reduced dependency on shopkeeper’s response.
* Enhanced privacy and security for document submission.

**Conclusion**

The Smart Print Automation System offers a fast, secure, and efficient way to print documents without relying on WhatsApp or email. This solution significantly reduces time consumption, ensures data privacy, and simplifies the document submission process. With future enhancements like cloud storage, payment integration, and mobile app support, this solution has the potential to revolutionize document printing services, making them faster, safer, and more accessible. The impact on society will be highly beneficial, particularly for students and professionals who face urgent document printing needs.