**Python Mini Project Report**

**Name: SUCHISMITA ROY**

**USN: 22BTRSN053**

**Sem / Sec: 2ND SEM (CSE – SOFTWARE ENGINEERING)**

**Github Link:** **https://github.com/Suchis10/QR-code**

**Blogger:** (Documentation in detail – (i)Abstract, (ii)Problem Statement, (iii)Objectives, (iv)Methodology, (v)Result Analysis, (vi)Conclusion, (vii)References – Base Research Paper, Website Link)

***Abstract :***

This report presents a QR code generation and scanning project implemented in Python. The project aims to create a user-friendly application that allows users to generate QR codes for various types of data and subsequently scan and decode QR codes to retrieve the encoded information. The project utilizes Python's libraries, including qr-code for QR code generation and open cv-python for QR code scanning. This report discusses the project's objectives, design, implementation, and potential future enhancements.

***1. Problem Statement***

The objective of this project is to develop a Python application that generates QR codes for various types of data. The application should be user-friendly, allowing users to input data and create QR codes easily. The generated QR codes should be saved as image files for further use and distribution.

**Requirements:**

**User Input**: The application should prompt the user to input the data they want to encode in the QR code. The data could be a URL, text, contact information, or any other supported format.

**QR Code Generation**: Utilize the **qr code library** in Python to create a QR code image based on the user-provided data. The generated QR code should conform to the QR code standards and support error correction to ensure reliable scanning.

**Image Output:** Save the generated QR code as an image file (e.g., PNG, JPEG) to the local filesystem. The user should be able to specify the output directory and filename for the QR code image.

**Validation:** Implement error handling and data validation to ensure that the user provides valid data for QR code generation. Notify the user in case of invalid input.

**User-Friendly Interface**: Create a user-friendly command-line interface (CLI) to guide the users through the QR code generation process. Display informative messages and instructions to help users understand the application's functionality.

**Cross-Platform Compatibility**: Ensure that the application works on multiple operating systems without any compatibility issues.

**Documentation:** Provide clear and concise documentation explaining how to use the application, including installation instructions and examples of QR code generation for different data types.

1. ***Methodology:***

**QR Code generation:**

The QR code generation part of the project involves the following steps:

* Install the required libraries: **qr code** and **Pillow[PIL]**
* Take user input for the data to be encoded in the QR code (e.g., URL, text, contact information).
* Use the **qr code** library to create a QR code image based on the input data.
* Save the generated QR code image to the local filesystem.

**QR Code Scanning:**

The QR code scanning part of the project involves the following steps:

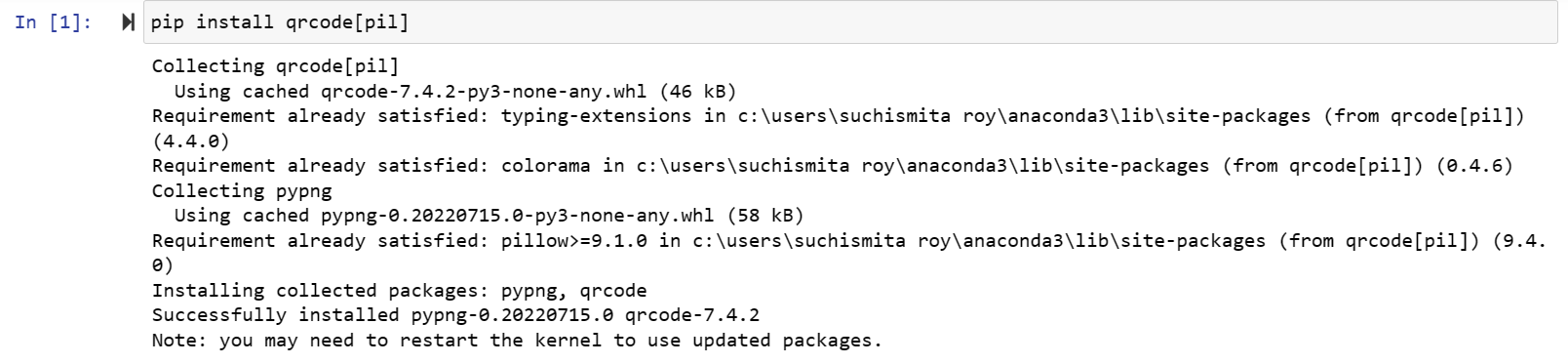
* Install the required libraries: **opencv**-python and **pyzbar.**
* Use the device's camera or an existing image to capture the QR code.
* Utilize the **opencv-python** library to process the image and extract the QR code region.
* Apply the **pyzbar** library to decode the information from the QR code.
* Display the decoded data to the user.

**User Interface:**

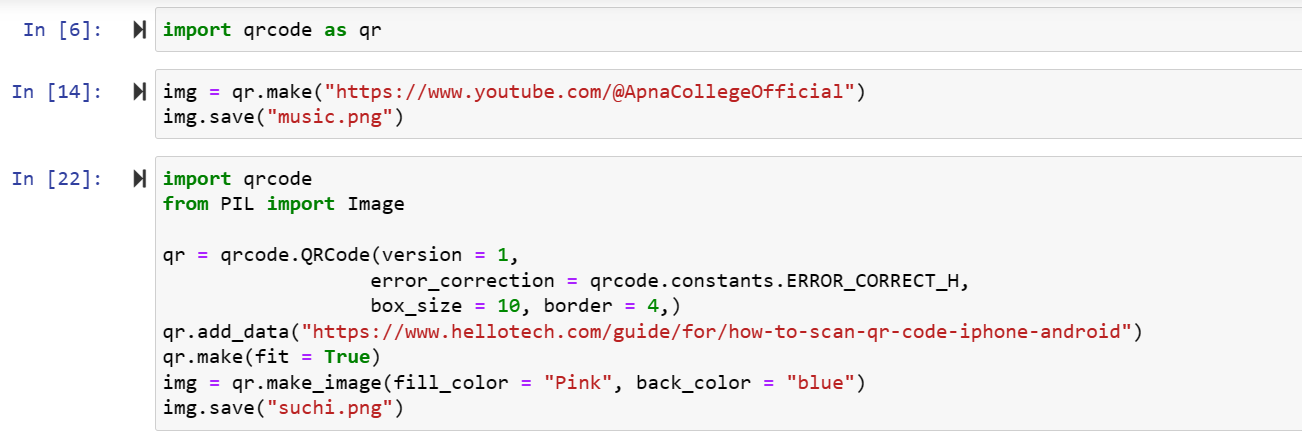
* The project may incorporate a simple command-line interface (CLI) to prompt users for their input during QR code generation and display the scanned results.

1. ***Coding and Results (Snapshot)***

* **Installing the library qrcode and PIL**

****

* **Code for the QR CODE Scanner**

****

* **Output of the QRCODE Scanner**

****

1. ***Result analysis of QR-Code Scanner:***

**Functionality:**

The first aspect to evaluate is the functionality of the QR code generation application. Verify if the application fulfils its primary objectives, such as generating QR codes for different data types and saving them as image files. Ensure that the generated QR codes are scannable and can be decoded to retrieve the original data accurately.

**User-Friendly Interface:**

Assess the user-friendliness of the application's interface. The command-line interface (CLI) should guide users through the QR code generation process with clear instructions and informative messages. If a graphical user interface (GUI) was implemented, evaluate its intuitiveness and ease of use.

1. ***Conclusion***

This report presented a QR code generation and scanning project in Python.

The project aims to provide a simple yet effective tool for generating and scanning QR codes, allowing users to store and retrieve information efficiently.

By leveraging Python libraries for QR code manipulation, the application demonstrates the flexibility and power of Python in working with various data formats.