PROFESSIONAL TRAINING REPORT

at

Sathyabama Institute of Science and Technology

(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering

By

SANDEEP YALAMANCHILI
(Reg. No – 42111130)

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

CATEGORY- 1 UNIVERSITY BY UGC

Accredited “A++” by NAAC I Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI CHENNAI - 600119

OCTOBER - 2024

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Category - I University by UGC
Accredited "A++" by NAAC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training-1 Report is the bonafide work of **SANDEEP YALAMANCHILI (42111130)** who carried out the Project entitled **“QR Code Generator”** under my supervision from June 2024 to October 2024.

Internal Guide
Dr.G.Kalaiarasi, M.E., Ph.D.,

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.,

Submitted for Professional Training-1 Viva Voce Examination held on ___

Internal Examiner **External Examiner**

DECLARATION

I, **SANDEEP YALAMANCHILI (Reg. No- 42111130),** hereby declare that the Professional Training-1 Report entitled **"QR Code Generator"** done by me under the guidance of **Dr.G.Kalairasi M.E.,Ph.D.,** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE:

PLACE: Chennai **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

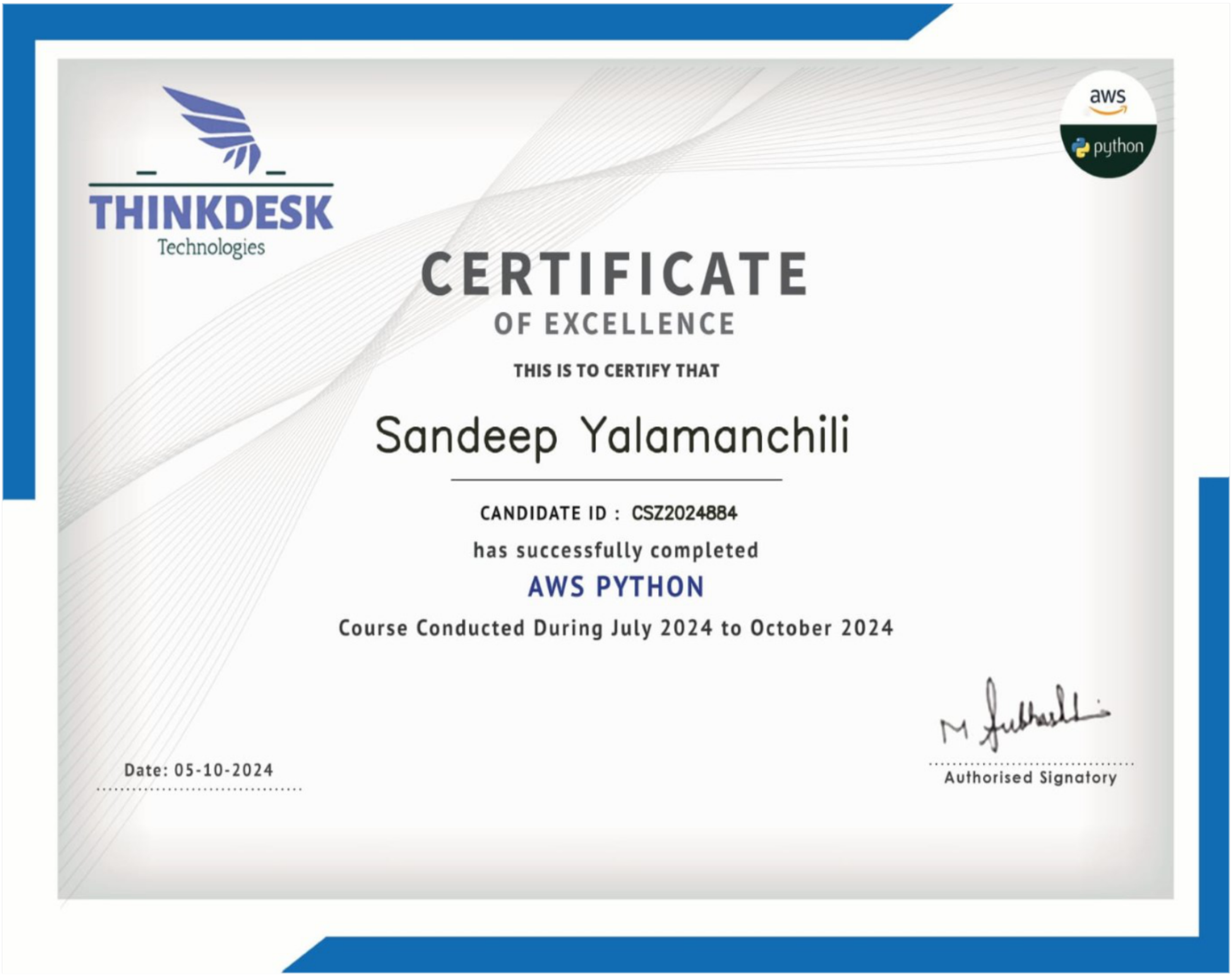
I am pleased to acknowledge my sincere thanks to **BOARD OF MANAGEMENT** of **Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph. D.**, **Dean**, School of Computing, and **Dr. L. Lakshmanan, M.E., Ph.D., Head of the Department** of Computer Science and Engineering for providing me with necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. Kalaiarasi, M.E.,Ph.D.,** for her valuable guidance, suggestions, and constant encouragement paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of
the **Department of Computer Science and Engineering** who were helpful in
many ways for the completion of the project

TRAINING CERTIFICATE



THINKDESK
Technologies

aws
python

CERTIFICATE
OF EXCELLENCE

THIS IS TO CERTIFY THAT

Sandeep Yalamanchili

CANDIDATE ID : CSZ2024884

has successfully completed

AWS PYTHON

Course Conducted During July 2024 to October 2024

Date: 05-10-2024

Authorised Signatory

ABSTRACT

This mini project focuses on the development of a “QR Code Generator” using Python and the PyQRCode library. The tool allows users to generate QR codes from any text or URL input, offering features such as handling single-line or multiline text and saving the QR codes in PNG format. Additionally, it supports the modification of the QR code’s content after initial creation, as well as accepting existing PNG files for regeneration. The project emphasizes ease of use and flexibility, ensuring users can create and manage their QR codes efficiently.

A key aspect of the project is its integration with “Amazon Web Services (AWS) Simple Storage Service (S3)”, where all generated QR codes are securely stored for future access. By utilizing AWS S3, the project ensures reliable cloud-based storage and retrieval, enabling users to maintain and share QR codes with ease. This project, developed as part of an AWS Python course for a bachelor's degree, demonstrates practical applications of cloud computing alongside Python programming, illustrating how these technologies can enhance digital asset management.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

In today’s technology-driven world, QR codes have emerged as a highly efficient tool for sharing information quickly and easily. From marketing campaigns and digital payments to event management and contactless data exchange, QR codes are used across various sectors to allow instant access to websites, product details, contact information, and more. Their ability to store a wide range of data types in a simple, scannable format has made them a valuable asset in both personal and professional contexts.

This project aims to build a QR Code Generator using Python and the PyQRCode library, offering a streamlined way to generate QR codes from any form of text or URLs. The tool is designed with flexibility in mind, allowing users to input single lines or multiline text, making it suitable for everything from basic web links to more detailed text-based content. The generated QR codes are saved in PNG format, ensuring high-quality output for both digital and print applications. This ensures that users can conveniently store and use these QR codes across a variety of platforms.

A unique feature of this project is the ability to modify the content embedded in the QR code even after it has been generated. This capability adds a layer of adaptability, enabling users to update or revise their QR codes as needed without having to start over from scratch. Whether it’s updating a website URL or adjusting information, users have full control over the data their QR codes store.

What makes this project stand out further is its integration with Amazon Web Services (AWS) Simple Storage Service (S3). By utilizing AWS S3, users are not limited to saving QR codes locally; they can also securely upload and store their generated QR codes in the cloud. This cloud storage ensures that QR codes are accessible from anywhere, backed up, and easily shareable. AWS S3’s scalable storage solutions provide long-term security, and users can benefit from features such as encryption and access controls to ensure their data remains safe. This feature is especially useful for businesses or organizations managing large volumes of QR codes.

This project is part of an AWS Python course designed to showcase how Python programming and cloud technologies can be used together to create practical, scalable applications. By combining Python’s simplicity with the power of AWS cloud services, this project demonstrates the potential of building tools that are not only functional but also secure and adaptable to real-world needs.

The QR Code Generator tool is a perfect example of how modern technologies can be applied to solve everyday challenges. It offers an efficient, user-friendly solution for generating and managing QR codes, with the added benefit of cloud integration for enhanced accessibility and security. Through this project, the use of Python and AWS S3 is demonstrated in a practical way, showcasing their potential to deliver powerful, scalable digital solutions.

PURPOSE

The purpose of this project is to develop a **user-friendly QR Code Generator** that simplifies the creation and management of QR codes using **Python**. QR codes have become essential in many industries for sharing information quickly, whether it's for marketing, payments, or accessing web links. This project is designed to provide a flexible tool for generating QR codes from various inputs, including text or URLs. A notable feature of this generator is its ability to support **multiline text**, allowing users to include more detailed information, making it suitable for use in a wide range of applications where more than just a simple URL is required.

One of the most useful aspects of this project is the ability to **modify the content** embedded in a QR code after it has been created. This allows users to update information, such as changing a website link or revising text, without needing to generate a new QR code. The ability to edit existing QR codes makes the tool adaptable to changing user needs, ensuring that the codes remain up-to-date and functional in the long term.

A key component of the project is its integration with **Amazon Web Services (AWS) Simple Storage Service (S3)**. By using AWS S3, users are able to store their generated QR codes securely in the cloud, in addition to saving them locally. This ensures that the codes are **accessible from anywhere** and can be easily shared with others. Storing QR codes in the cloud also provides the benefit of **data security**, as AWS S3 includes features like encryption and access controls, ensuring that the codes are protected from unauthorized access.

The use of AWS S3 adds **scalability** to the project, allowing users to store and manage large numbers of QR codes without worrying about running out of space or losing access to their files. As the project grows, AWS S3's infrastructure will seamlessly support the increased storage needs, making it an ideal solution for long-term use.

The project is intended to demonstrate practical skills in both **Python programming** and **cloud computing**, with a particular emphasis on using **AWS services** to enhance the storage and management of digital assets. By combining the simplicity and power of **Python** with the secure and scalable infrastructure of **AWS S3**, this project highlights how these technologies can work together to build efficient, real-world solutions. This project serves as part of the academic requirements for an **AWS Python course**, showcasing the integration of software development and cloud computing in building modern digital tools.

By developing this QR Code Generator, the project illustrates how Python can be used to create a functional and adaptable tool, while also demonstrating the importance of cloud services like AWS in improving accessibility, security, and scalability. The goal is to showcase the practical applications of **cloud-based storage** and **Python programming** to solve real-world challenges in a flexible and efficient manner.

USER INPUT FORMAT

- **Enter the text or link(s) for the QR code:** Users can input either a single line or multiple lines of text, or even links, which will be encoded into the QR code. The tool supports multiline input, offering flexibility for various use cases, such as embedding URLs, contact information, or longer messages.

- **After input is given, press the Enter key twice to complete the input:** This allows users to signal the completion of their input and ensures the tool properly handles multiline entries, processing them as a complete block of text.

- **Enter the local filename to save the QR code:** Users must provide a name for saving the QR code as a PNG file on their local device. The filename must be in the format name_qr.png, ensuring the QR code is properly formatted for later use.

- **Enter the S3 filename for the QR code:** This is the name given to the QR code file to store it in **AWS S3**, the cloud storage service. The file must also follow the format name_qr.png, allowing for easy retrieval and consistent naming conventions when accessing it from the cloud.

Each field is carefully designed to provide a **user-friendly experience**, ensuring that users can easily generate and manage their QR codes. Additionally, by saving the QR code both locally and on **AWS S3**, users can access their codes from any device, anywhere in the world, at any time. This enhances the flexibility and accessibility of the tool, making it a practical solution for personal and professional use.

python implementations

The provided Python implementation is designed to generate a QR code from user input, handle multiline text, and upload the generated QR code to an AWS S3 bucket. Here's the full Python implementation :

.code:

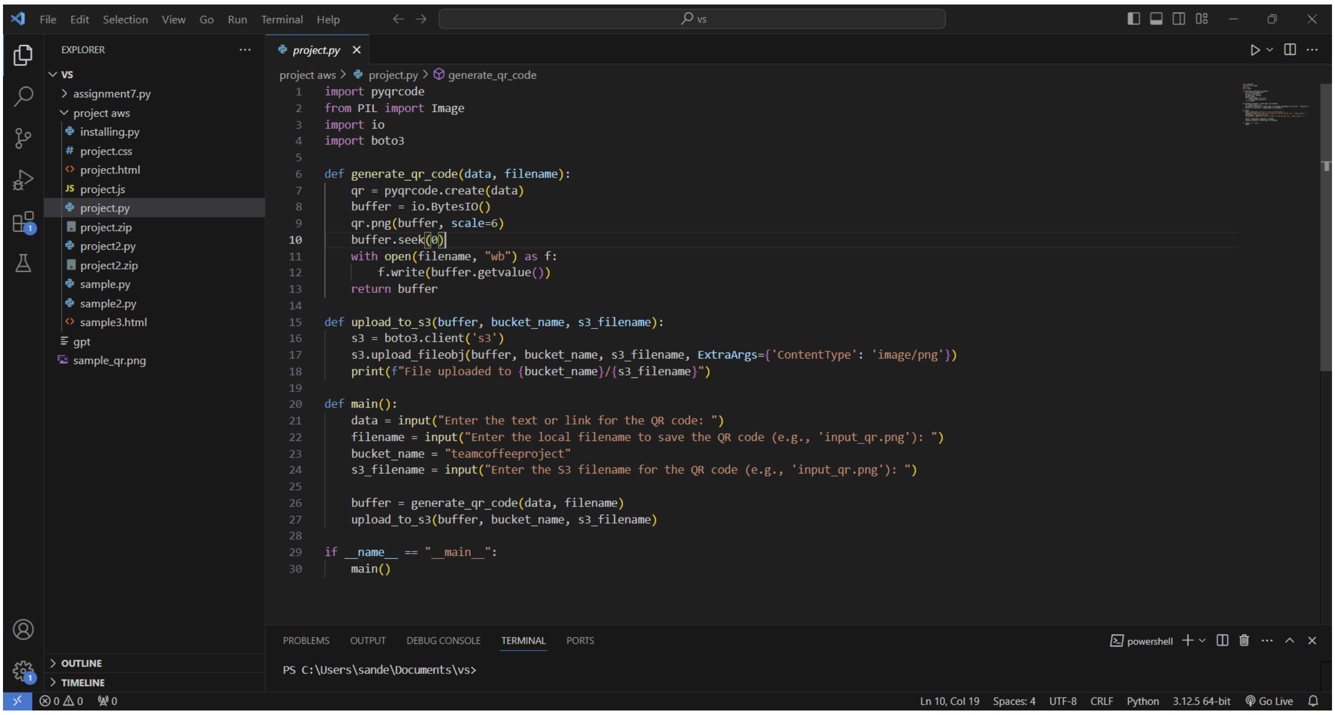
```
import pyqrcode

from PIL import Image

import io

import boto3


def generate_qr_code(data, filename):

   qr = pyqrcode.create(data)

   buffer = io.BytesIO()

   qr.png(buffer, scale=6)

   buffer.seek(0)

   with open(filename, "wb") as f:

      f.write(buffer.getvalue())

   return buffer


def upload_to_s3(buffer, bucket_name, s3_filename):

   s3 = boto3.client('s3')

         s3.upload_fileobj(buffer,    bucket_name,    s3_filename,    ExtraArgs={'ContentType':
'image/png'})

   print(f"File uploaded to {bucket_name}/{s3_filename}")


def main():

   data = input("Enter the text or link for the QR code: ")

   filename = input("Enter the local filename to save the QR code (e.g., 'input_qr.png'): ")

   bucket_name = "teamcoffeeproject"

   s3_filename = input("Enter the S3 filename for the QR code (e.g., 'input_qr.png'): ")


   buffer = generate_qr_code(data, filename)

   upload_to_s3(buffer, bucket_name, s3_filename)
```

```
if __name__ == "__main__":
    main()
```

- **extensions to be installed:**

1. **pyqrcode**: This library is used to create QR codes from the given data.
2. **Pillow**: This is a Python imaging library (PIL fork) used for working with image files. It's necessary for handling the QR code PNG file output.
3. **Boto3**: The AWS SDK for Python, used here to upload the generated QR code to an S3 bucket.

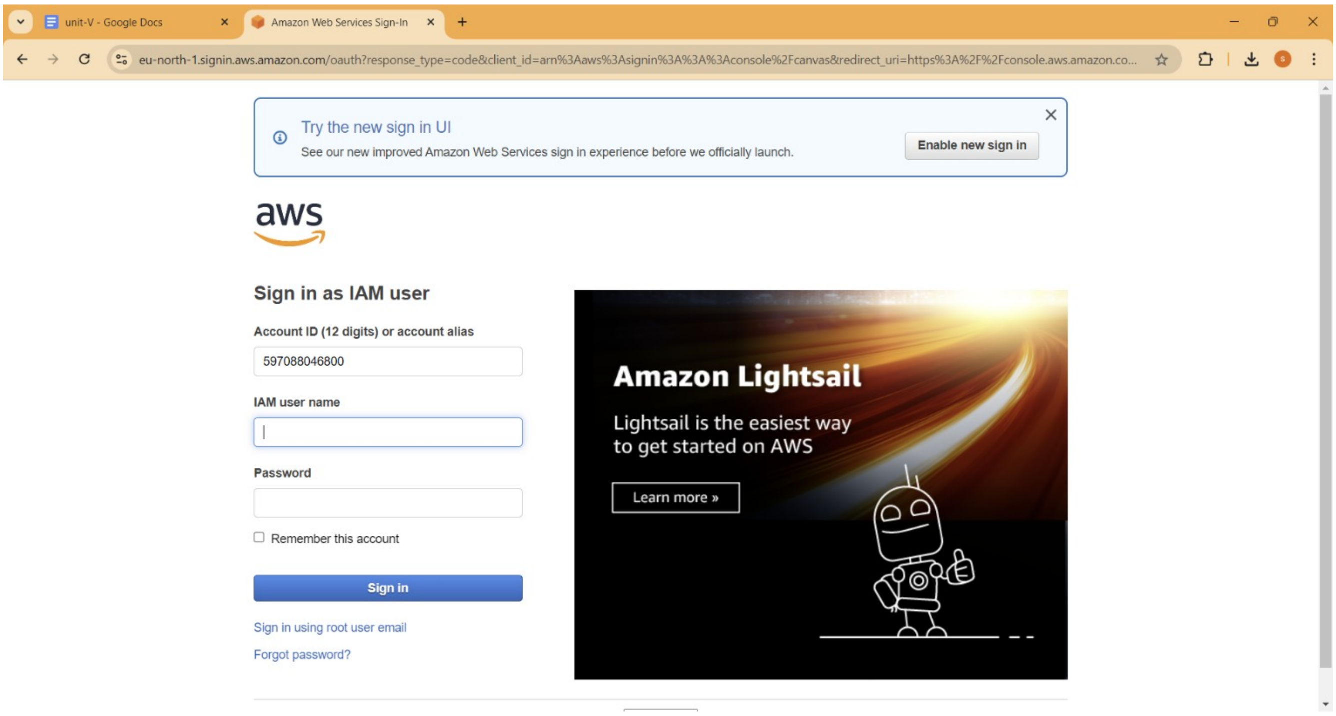
Once these dependencies are installed, you can execute the program



```python
import pyqrcode
from PIL import Image
import io
import boto3

def generate_qr_code(data, filename):
    qr = pyqrcode.create(data)
    buffer = io.BytesIO()
    qr.png(buffer, scale=6)
    buffer.seek(0)
    with open(filename, "wb") as f:
        f.write(buffer.getvalue())
    return buffer

def upload_to_s3(buffer, bucket_name, s3_filename):
    s3 = boto3.client('s3')
    s3.upload_fileobj(buffer, bucket_name, s3_filename, ExtraArgs={'ContentType': 'image/png'})
    print(f"File uploaded to {bucket_name}/{s3_filename}")

def main():
    data = input("Enter the text or link for the QR code: ")
    filename = input("Enter the local filename to save the QR code (e.g., 'input_qr.png'): ")
    bucket_name = "teamcoffeeproject"
    s3_filename = input("Enter the S3 filename for the QR code (e.g., 'input_qr.png'): ")

    buffer = generate_qr_code(data, filename)
    upload_to_s3(buffer, bucket_name, s3_filename)

if __name__ == "__main__":
    main()
```

```
PS C:\Users\sande\Documents\vs>
```

Fig no. 5.1


unit-V - Google Docs
Amazon Web Services Sign-In
eu-north-1.signin.aws.amazon.com/oauth?response_type=code&client_id=arn%3Aaws%3Asignin%3A%3A%3Aconsole%2Fcanvas&redirect_uri=https%3A%2F%2Fconsole.aws.amazon.co...
aws
Amazon Lightsail
Lightsail is the easiest way to get started on AWS
Learn more »

Try the new sign in UI

See our new improved Amazon Web Services sign in experience before we officially launch.

Enable new sign in

Sign in as IAM user

Account ID (12 digits) or account alias: 597088046800

IAM user name:

Password:

□ Remember this account

Sign in

Sign in using root user email

Forgot password?

Fig no. 5.2

Data analysis:

1. Functionality Overview

- The implementation generates QR codes using the pyqrcode library and uploads them to AWS S3 via the boto3 library. It allows users to input text or URLs, save the QR code locally in PNG format, and store it in the cloud for easy access.

2. Code Structure

- The code is modular, with separate functions for QR code generation (generate_qr_code) and S3 uploading (upload_to_s3). This structure improves readability and maintenance, and the use of if __name__ == "__main__": enhances reusability.

3. User Interaction

- The script prompts users clearly for input, ensuring ease of use. However, enforcing filename conventions could further improve user experience.

4. Performance and Efficiency

- Utilizing io.BytesIO for in-memory image handling improves performance by reducing disk I/O. However, performance may degrade with bulk operations, suggesting that asynchronous uploads could enhance efficiency.

5. Error Handling

- The implementation currently lacks robust error handling. Incorporating try-except blocks to manage potential exceptions during QR code generation and S3 uploads
- would enhance reliability.

6. Security Considerations

- The script assumes AWS credentials are correctly configured. It is crucial to avoid hard-coded credentials, following best practices for security. Setting the ContentType during S3 uploads is a good practice for proper file recognition.

7. Potential Improvements

- Error Handling: Implement try-except blocks for better user feedback and reliability.
- Filename Validation: Add checks to ensure filenames meet expected formats, reducing user errors.
- Logging: Incorporate logging to track actions, successes, and failures for easier debugging.
- Bulk Processing: Allow users to generate and upload multiple QR codes simultaneously.
- User Interface: Develop a GUI using Tkinter or PyQt to enhance accessibility for users unfamiliar with command-line tools.

8. Use Cases:

The QR Code Generator can be used for promotional materials, event management, and educational resources, facilitating easy sharing and access to online content.

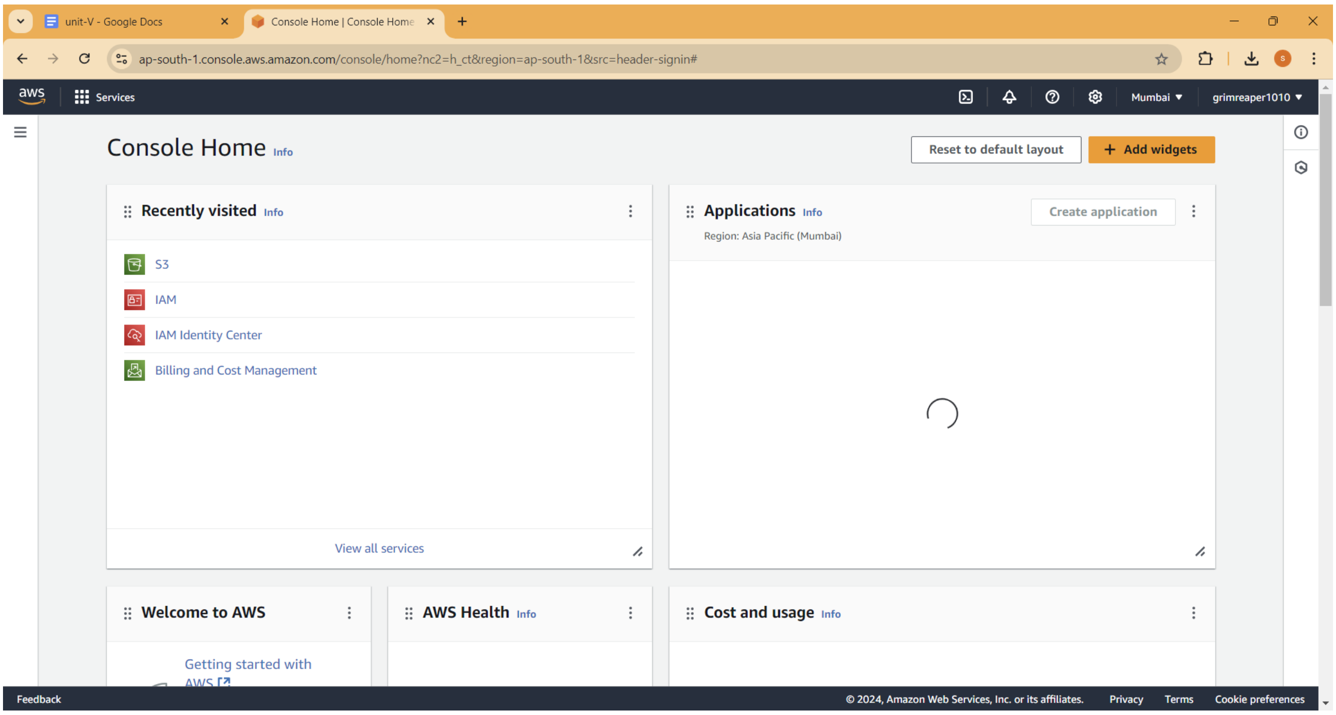


Fig no.6.1

Conclusion

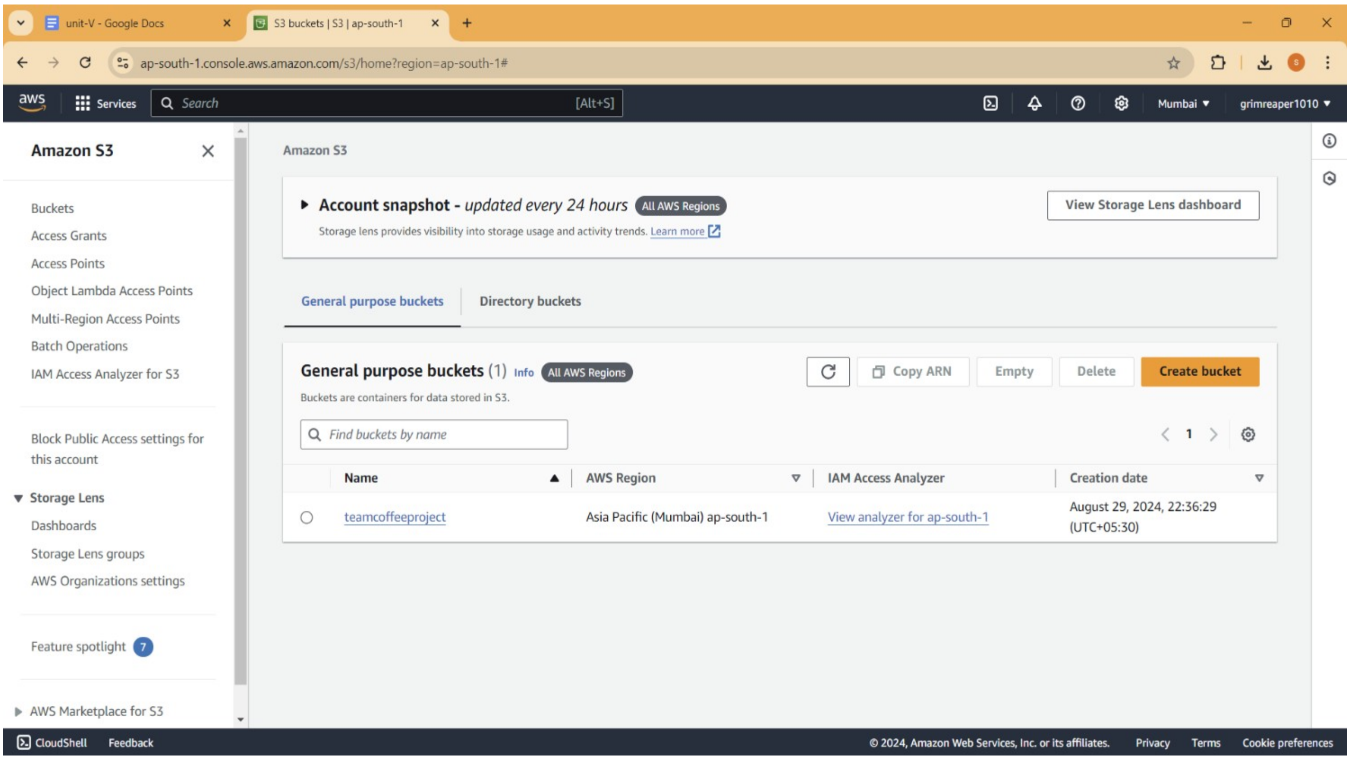
The QR Code Generator project successfully demonstrates the integration of Python programming with cloud computing through AWS. It provides users with an efficient and straightforward tool for generating QR codes from text or URLs and securely storing them in AWS S3. The implementation showcases key functionalities, such as handling multiline text input, saving QR codes in PNG format, and facilitating cloud storage, all of which enhance the overall user experience.

While the current implementation meets the primary objectives, there are numerous opportunities for continuous improvement. Enhancements such as robust error handling, user input validation, and detailed logging can significantly increase the application's reliability and user-friendliness. Implementing a graphical user interface (GUI) could further broaden the accessibility of the tool, allowing users who are less familiar with command-line operations to benefit from its features.

Security considerations are paramount in any application that involves cloud storage. Ensuring that AWS credentials are managed securely, along with employing best practices for data protection, will help build user trust and ensure the safety of their generated QR codes.

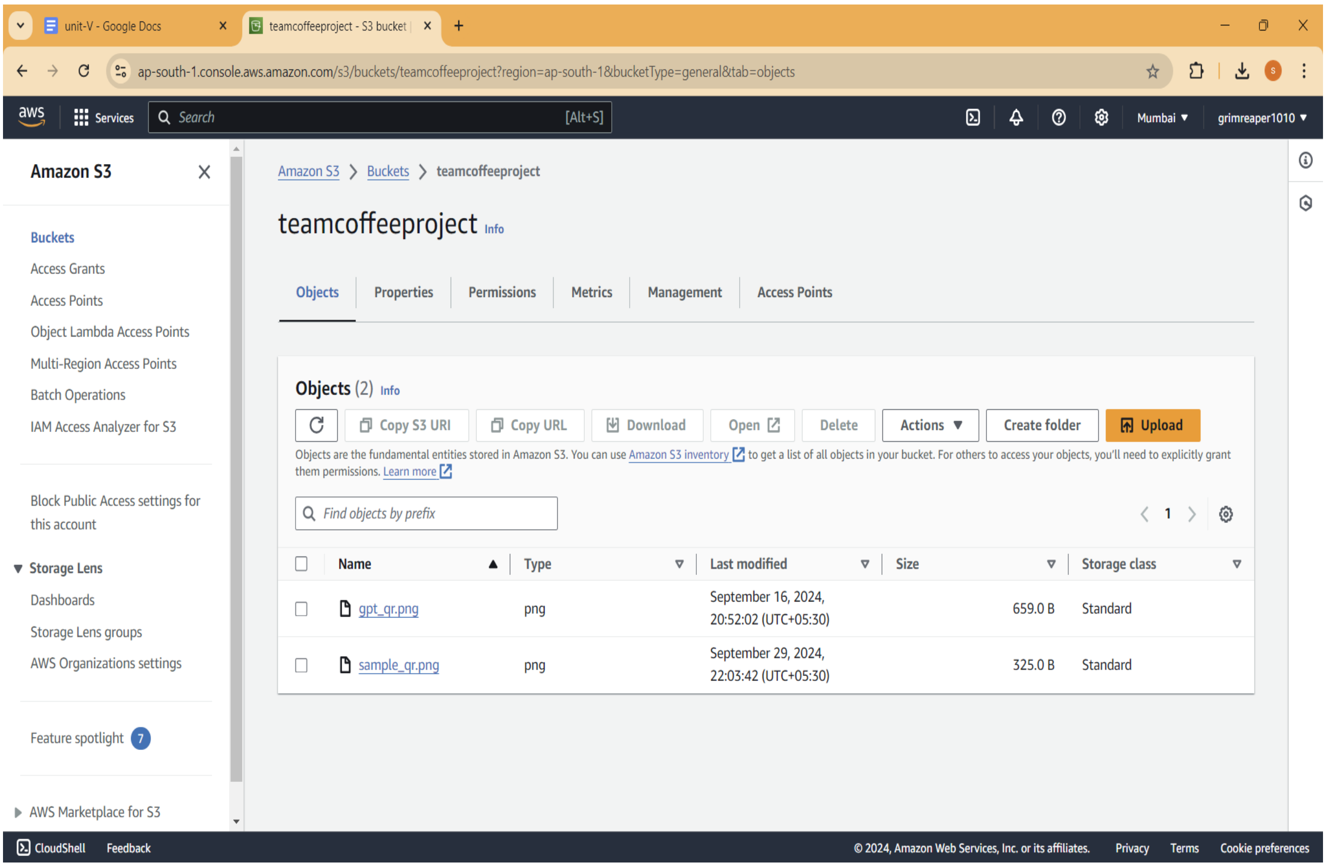
The potential use cases for this project are vast, ranging from marketing materials and event ticketing to educational resources, making it a versatile solution for various industries. With ongoing user feedback and iterative enhancements, this QR Code Generator can evolve into a comprehensive tool that meets the dynamic needs of its users.

In summary, the QR Code Generator not only serves its intended purpose effectively but also lays a strong foundation for future development. By focusing on continuous improvement, security, and user experience, this project can adapt to changing technological landscapes and remain relevant in an increasingly digital world.


unit-V - Google Docs
S3 buckets | S3 | ap-south-1
ap-south-1.console.aws.amazon.com/s3/home?region=ap-south-1#
aws
Services
Search
[Alt+S]
Mumbai
grimreaper1010
Amazon S3
Buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 7
AWS Marketplace for S3
Amazon S3
Account snapshot - updated every 24 hours
All AWS Regions
View Storage Lens dashboard
Storage lens provides visibility into storage usage and activity trends. Learn more
General purpose buckets
Directory buckets
General purpose buckets (1)
Info
All AWS Regions
Copy ARN
Empty
Delete
Create bucket
Buckets are containers for data stored in S3.
Find buckets by name
1
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Name	AWS Region	IAM Access Analyzer	Creation date
○ teamcoffeeproject	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	August 29, 2024, 22:36:29 (UTC+05:30)

Fig no. 7.1



Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

	Name	Type	Last modified	Size	Storage class
□	gpt_qr.png	png	September 16, 2024, 20:52:02 (UTC+05:30)	659.0 B	Standard
□	sample_qr.png	png	September 29, 2024, 22:03:42 (UTC+05:30)	325.0 B	Standard

Fig no. 7.2

References

1. **AWS Training and Certification**:
    - For further learning, you can refer to the official AWS training and certification resources: AWS Training and Certification
    - From the pride course which was completed and certified
2. **Python Libraries**
    - **PyQRCode Documentation**:
        - PyQRCode is a Python library for generating QR codes. Official documentation can be found here: PyQRCode Documentation
    - **Pillow (PIL) Documentation**:
        - Pillow is a fork of the Python Imaging Library (PIL) and is used for opening, manipulating, and saving image files. Official documentation is available at: Pillow Documentation
3. **AWS S3**
    - **Boto3 Documentation**:
        - Boto3 is the Amazon Web Services (AWS) SDK for Python. It allows Python developers to write software that makes use of Amazon services like S3. Documentation can be found here: Boto3 Documentation
    - **Amazon S3 User Guide**:
        - The official AWS documentation for using S3, including best practices and security considerations: Amazon S3 User Guide
4. **QR Code Technology**
    - **Wikipedia - QR Code**:
        - An overview of QR code technology, its history, and applications can be found on the Wikipedia page: QR Code - Wikipedia
    - **How QR Codes Work**:
        - A detailed article explaining the mechanics of QR codes and their applications can be found here: How QR Codes Work