

A Course End Project  
on  
**Access And Manage Data in Azure Storage Explorer**

Submitted in the Partial Fulfillment of the  
Requirements  
for the Award of the Degree of

**BACHELOR OF TECHNOLOGY**  
**IN**  
**COMPUTER SCIENCE AND ENGINEERING (AI&ML)**

Submitted  
By  
**Team No.: 17**

|                       |                   |
|-----------------------|-------------------|
| <b>J. Monika</b>      | <b>21881A6686</b> |
| <b>P. Sadhika</b>     | <b>21881A66B0</b> |
| <b>G. Shiva Kumar</b> | <b>21881A6685</b> |

Under the Esteemed Guidance of  
**Dr. P. Pavan Kumar**  
Associate Professor



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI&ML)**

**VARDHAMAN COLLEGE OF ENGINEERING**  
(AUTONOMOUS)

Affiliated to **JNTUH**, Approved by **AICTE**, Accredited by **NAAC**, with **A++** Grade, **ISO 9001:2015** Certified  
Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India

**2023- 24**

## ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express my deep sense of gratitude to **Dr. P Pavan Kumar, Associate Professor** for their able guidance and useful suggestions, which helped us in completing the design part of potential project in time.

We are particularly thankful to **Dr. M. A. Jabbar**, Professor & Head, Department of Computer Science and Engineering (AI&ML) for his guidance, intense support and encouragement, which helped us to mould our project into a successful one.

We show gratitude to our honorable Principal **Dr. J. V. R. Ravindra**, for having provided all the facilities and support.

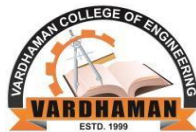
We avail this opportunity to express our deep sense of gratitude and heartfelt thanks to **Dr. Teegala Vijender Reddy**, Chairman and **Sri Teegala Upender Reddy**, Secretary of VCE, for providing congenial atmosphere to complete this project successfully.

We also thank all the staff members of the department of **CSE(AI&ML)** for their valuable support and generous advice. Finally, thanks to all our friends and family members for their continuous support and enthusiastic help.

**J. Monika - 21881A6686**

**P. Sadhika - 21881A66B0**

**G. Shiva Kumar- 21881A6685**



# **VARDHAMAN COLLEGE OF ENGINEERING**

**(AUTONOMOUS)**

Affiliated to **JNTUH**, Approved by **AICTE**, Accredited by **NAAC**, with **A++** Grade, **ISO 9001:2015** Certified  
Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India

## **Department of Computer Science and Engineering (AI&ML)**

### **CERTIFICATE**

This is to certify that the Course End Project report work entitled “**Access Data in Azure Storage Explorer**” carried out by Ms. **J. Monika**, Roll Number **21881A6686**, Ms. **P. Sadhvika**, Roll Number **21881A66B0**, Mr. **G. Shiva Kumar**, Roll Number **21881A6685** towards Course End Project and submitted to the Department of Computer Science and Engineering(AI&ML), in partial fulfillment of the requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering (AI&ML)** during the year 2023-24.

**Name & Signature of the Instructors**

**Dr. P. Pavan Kumar**

**Associate Professor**

**Name & Signature of the HOD**

**Dr. M. A. Jabbar**

**HOD, CSE(AI&ML)**

## ABSTRACT

Azure Storage Explorer is a versatile, user-friendly desktop application that facilitates the management and access of Azure Storage resources. It supports various operating systems, including Windows, macOS, and Linux. To begin, users download and install the application from the Azure website. Upon launching Azure Storage Explorer, users sign in with their Azure credentials, establishing a secure connection to their Azure account. The intuitive interface allows users to easily navigate through their subscriptions and storage accounts.

Within Azure Storage Explorer, users can access different storage services such as Blob Containers, File Shares, Queues, and Tables. By expanding the desired storage account, users can view and interact with the data stored in these services. Key functionalities include uploading and downloading files, which are performed with simple drag-and-drop actions or by using the upload and download buttons. Additionally, users can manage their data by deleting files, viewing file properties, and copying URLs for direct access to blobs.

The tool also offers advanced features such as managing access policies, setting permissions, and generating shared access signatures (SAS). Azure Storage Explorer's powerful search and filtering capabilities enhance data management efficiency, making it an essential tool for developers and IT professionals working with Azure Storage.

**Keyword:** Azure Storage Explorer, Azure Storage, Data management, Blob Containers, File Shares, Queues

## ABBREVIATIONS

| Abbreviation | Expansion                   |
|--------------|-----------------------------|
| VM           | Virtual Machine             |
| BLOB         | Binary Large object         |
| MFA          | Multi Factor Authentication |
| SAS          | Shared Access Signatures    |
| CLI          | Command Line Interface      |

## Table of Contents

| <b>Chapter No.</b> | <b>Title</b>   | <b>Page No.</b> |
|--------------------|--|-----------------|
|                    | Acknowledgements   | ii              |
|                    | Abstract   | iv              |
|                    | Abbreviations  | v               |
| <b>Chapter 1</b>   | <b>Introduction</b>  |                 |
|                    | 1.1 Scope  | 1               |
|                    | 1.2 Objectives   | 1               |
|                    | 1.3 AWS Introduction   | 1-3             |
| <b>Chapter 2</b>   | <b>Problem Definition and Proposed System Methodology</b>            |                 |
|                    | 2.1 Problem Statement  | 4               |
|                    | 2.2 Proposed System Methodology                                      | 4               |
|                    | 2.3 Steps  | 5-18            |
| <b>Chapter 3</b>   | <b>Software Requirements Specification and Hardware Requirements</b> |                 |
|                    | 3.1 Software Requirements Specification                              | 19              |
|                    | 3.2 Hardware Requirements  | 19              |
| <b>Chapter 4</b>   | <b>Results and Discussions</b>                                       | 20              |
| <b>Chapter 5</b>   | <b>Conclusion and Future Scope</b>                                   | 21              |
|                    | <b>References</b>  | 22              |

# CHAPTER 1

## INTRODUCTION

### 1.1 Scope

The scope of this project encompasses the setup and configuration of Azure Storage Explorer for efficient data management in Azure Storage. This includes detailed instructions on downloading, installing, and configuring the application to ensure seamless integration with Azure accounts. The project guides users through navigating storage resources such as Blob Containers, File Shares, Queues, and Tables. It covers essential functionalities like uploading, downloading, and deleting files, as well as viewing file properties. Advanced features such as managing access policies and generating shared access signatures (SAS) are also included, catering to individuals and organizations aiming to streamline their Azure Storage operations.

### 1.2 Objectives

The primary objective of this project is to provide a comprehensive guide for accessing and managing data in Azure Storage using Azure Storage Explorer, targeting users with varying levels of technical expertise. Some key objectives include:

- Downloading and installing Azure Storage Explorer.
- Signing in to Azure Storage Explorer and connecting to Azure accounts.
- Navigating storage resources including Blob Containers, File Shares.
- Uploading, downloading, and deleting files within Azure Storage.
- Viewing and managing file properties and metadata.
- Managing access policies and generating shared access signatures (SAS).

### 1.3 Azure Storage Explorer Introduction

Azure Storage Explorer is a comprehensive and widely adopted desktop application provided by Microsoft. It allows users to manage Azure Storage resources with ease, offering a broad set of functionalities for different types of storage services such as blobs, files, queues, and tables.

#### Key Features and Services of Azure Storage Explorer:

##### 1. Blob Storage:

- **Upload and Download:** Easily upload and download blobs with drag-and-drop functionality or through the interface.
- **Delete and Manage:** Delete blobs and manage blob properties and metadata directly from the application.
- **Copy URL:** Obtain URLs for direct access to blobs for sharing or use in applications.

##### 2. File Storage:

- **File Shares:** Access and manage Azure File Shares.
- **File Operations:** Upload, download, and delete files in your file shares.

3. **Queue Storage:**
  - **Queue Management:** View and manage Azure Queues.
  - **Message Operations:** Add, dequeue, and delete messages from your queues.
4. **Table Storage:**
  - **Table Access:** Browse and manage Azure Tables.
  - **Entity Management:** Add, edit, and delete entities within tables.
5. **Account Management:**
  - **Multiple Accounts:** Connect to multiple Azure accounts and subscriptions.
  - **Access Controls:** Manage access policies and permissions for your storage accounts and containers.
6. **Security and Access:**
  - **Shared Access Signatures (SAS):** Generate SAS tokens to provide controlled access to your storage resources.
  - **Access Policies:** Set and manage access policies for containers and file shares.
7. **Integration and Compatibility:**
  - **Cross-Platform:** Available for Windows, macOS, and Linux, providing a consistent experience across different operating systems.
  - **Azure Integration:** Seamlessly integrates with Azure services, making it easy to manage resources across your Azure ecosystem.
8. **Search and Filtering:**
  - **Advanced Search:** Quickly find specific blobs, files, queues, or tables using the search functionality.
  - **Filtering Options:** Filter and sort your storage resources to efficiently manage large datasets.

### **Benefits of Using Azure Storage Explorer:**

- **Ease of Use:** User-friendly interface that simplifies the management of Azure Storage resources.
- **Efficiency:** Streamlined operations for uploading, downloading, and managing storage data.
- **Security:** Robust security features, including access policies and SAS tokens, to ensure data protection.
- **Flexibility:** Supports a wide range of storage services and operations, catering to various use cases.
- **Accessibility:** Cross-platform availability ensures that users can manage Azure Storage from any device.

Azure Storage Explorer is utilized by a wide range of users, from developers and IT professionals to large enterprises, to efficiently manage their Azure Storage resources, ensuring data is easily accessible, secure, and well-organized.



## **CHAPTER 2**

### **PROBLEM STATEMENT AND PROPOSED METHODOLOGY**

#### **2.1 Problem Statement**

Accessing and managing data in Azure Storage explorer presents various challenges for users, particularly in efficiently navigating storage resources and performing essential data management tasks. Users encounter difficulties in securely connecting Azure Storage Explorer to their Azure accounts, understanding the nuances of different storage services such as Blob Containers, File Shares, Queues, and Tables, and effectively utilizing key functionalities like uploading, downloading, and deleting files. Furthermore, managing access policies, generating shared access signatures (SAS), and optimizing data management workflows pose additional complexities. The lack of a comprehensive guide to streamline these processes often leads to inefficiencies, security vulnerabilities, and suboptimal utilization of Azure Storage resources. Thus, there is a need for a standardized, user-friendly solution to address these challenges and enhance the accessibility and efficiency of data management in Azure Storage.

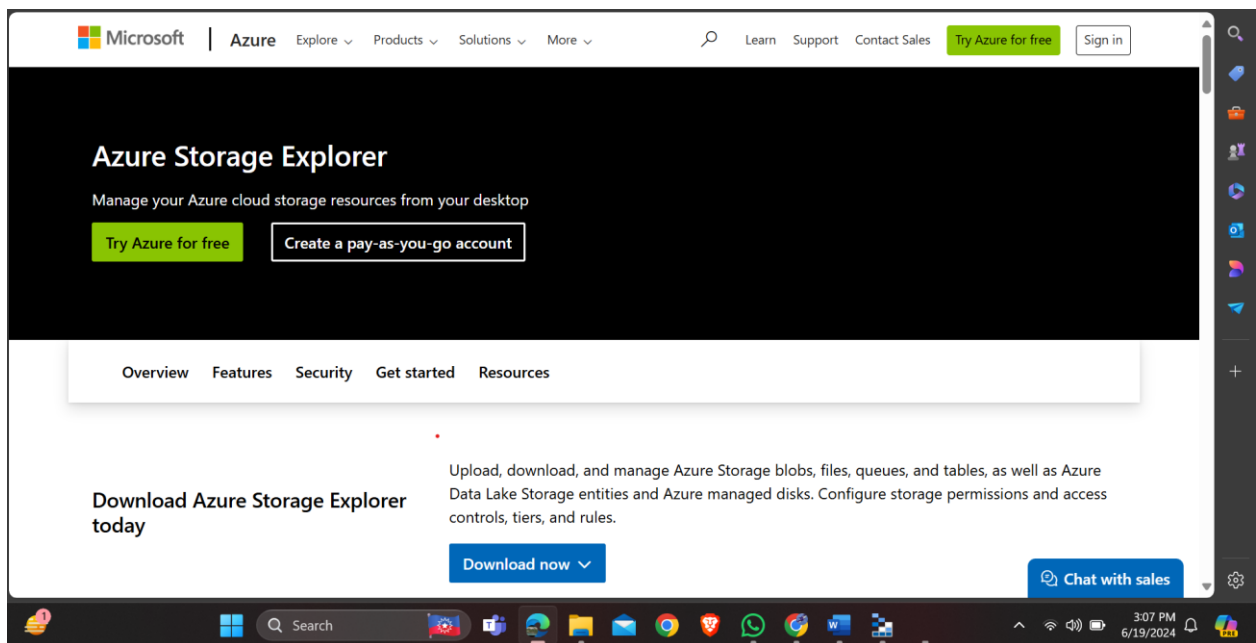
#### **2.2 Proposed System Methodology**

The proposed methodology for accessing and managing data in Azure Storage using Azure Storage Explorer offers a structured approach to streamline operations. It begins with a thorough analysis of user requirements and proceeds with clear, step-by-step instructions for installing and configuring Azure Storage Explorer across different platforms. Comprehensive user training materials are provided to educate users on navigating Azure Storage resources and utilizing key functionalities like file uploads, downloads, and access management. Best practices are integrated for optimizing data workflows, supported by rigorous testing and validation to ensure reliability and performance. Continuous user feedback drives iterative improvements, complemented by extensive documentation and support resources for seamless adoption and efficient data management.

## 2.3 Steps

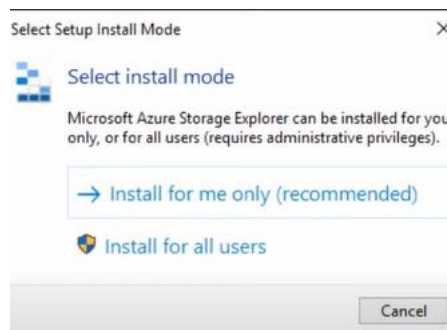
By following these steps, you can both install Azure Storage Explorer and efficiently access and manage data in Azure Storage using the application, simplifying your data management tasks in the Azure cloud environment

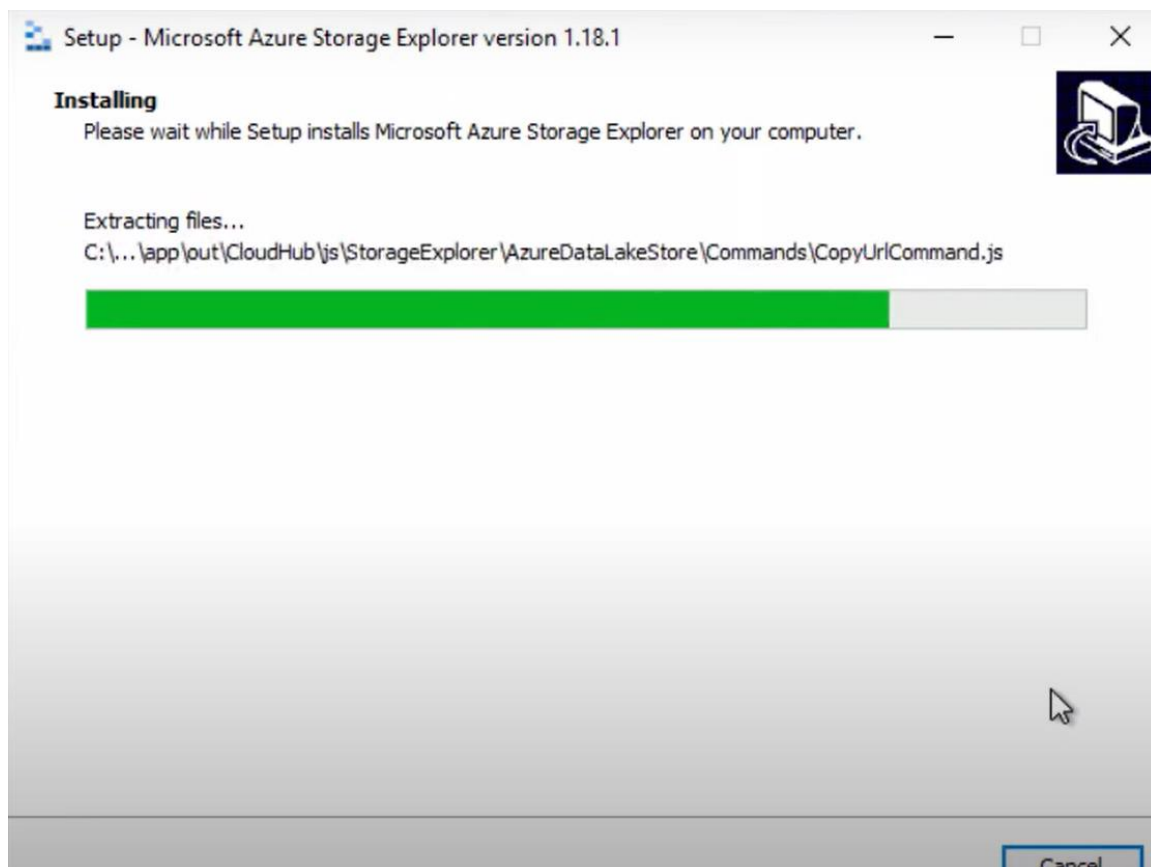
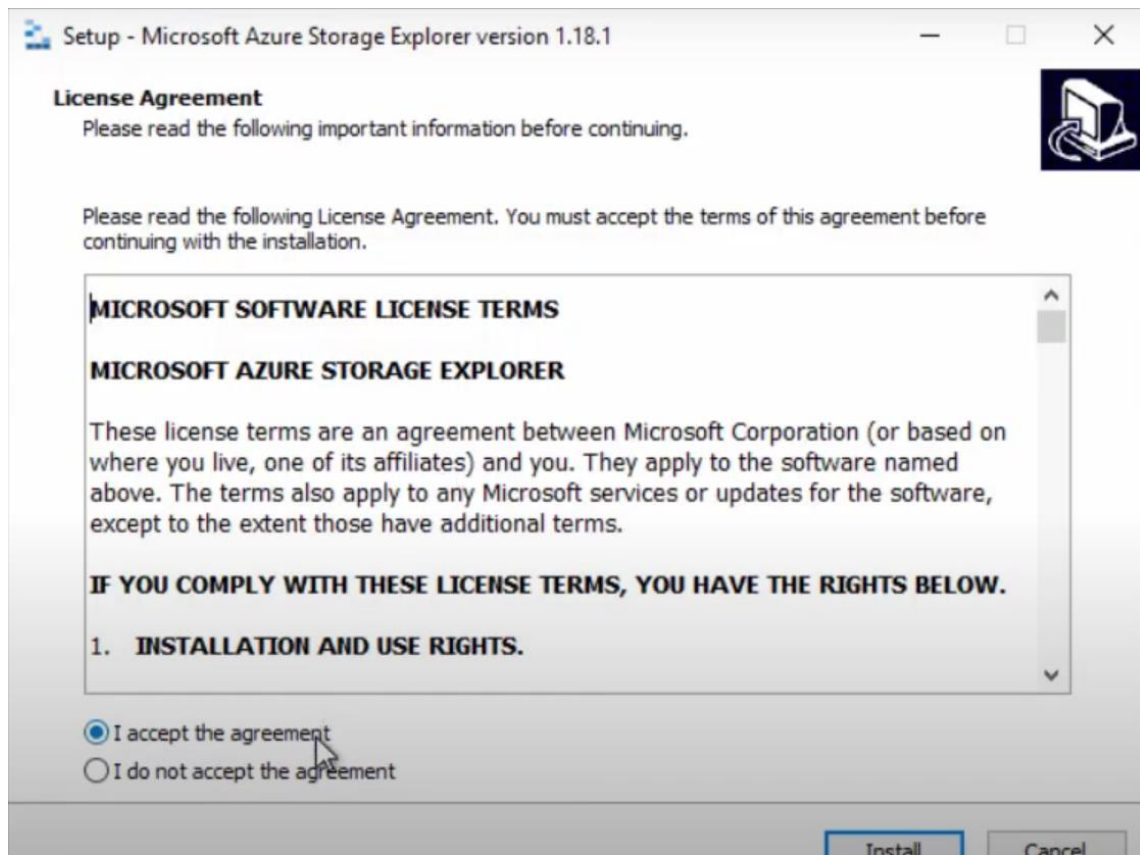
Step-1: Download Azure Storage Explorer from the official Microsoft website.



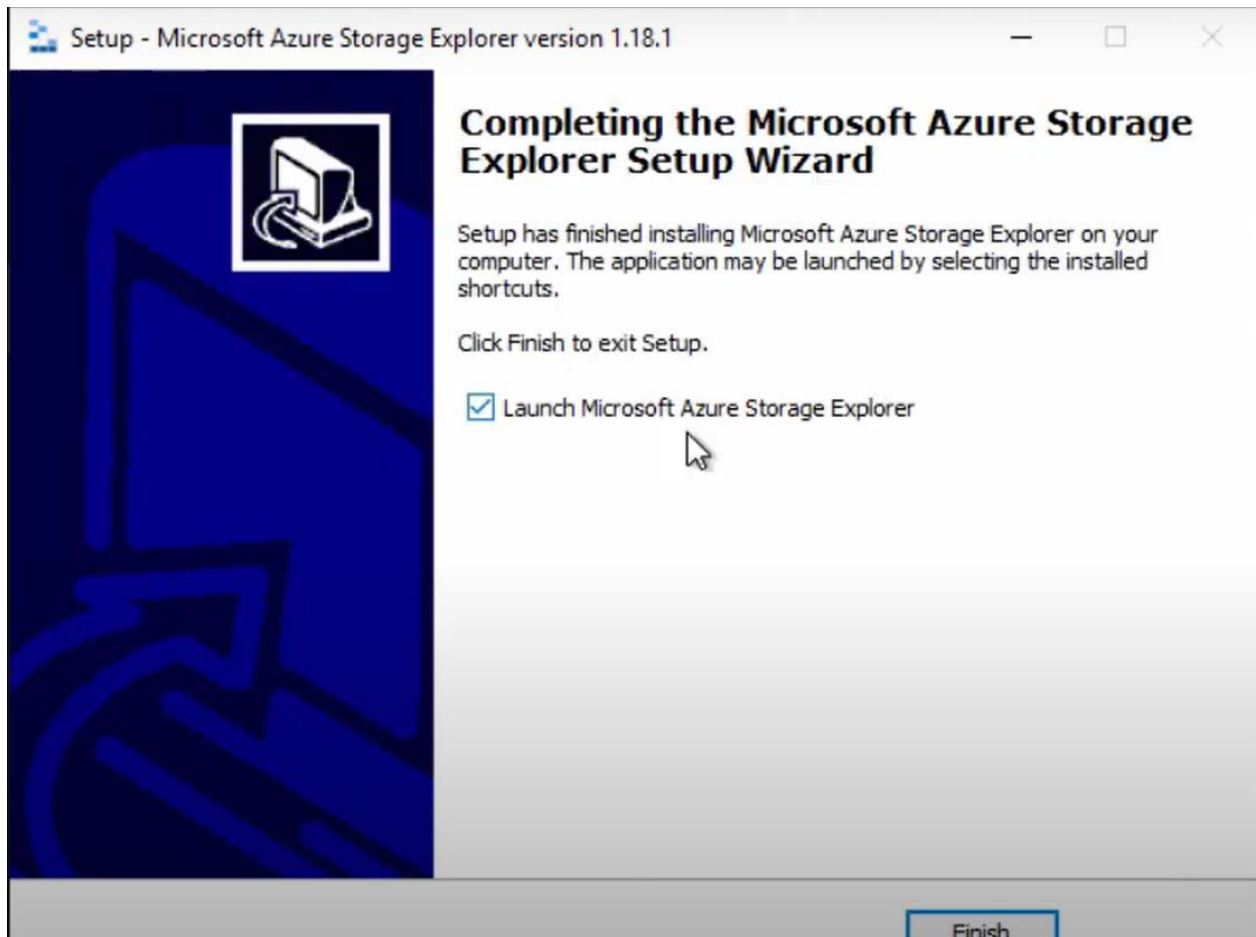
Step-2: Once the download is complete, run the installer file.

Step-3: Follow the installation wizard prompts to install Azure Storage Explorer on your computer.





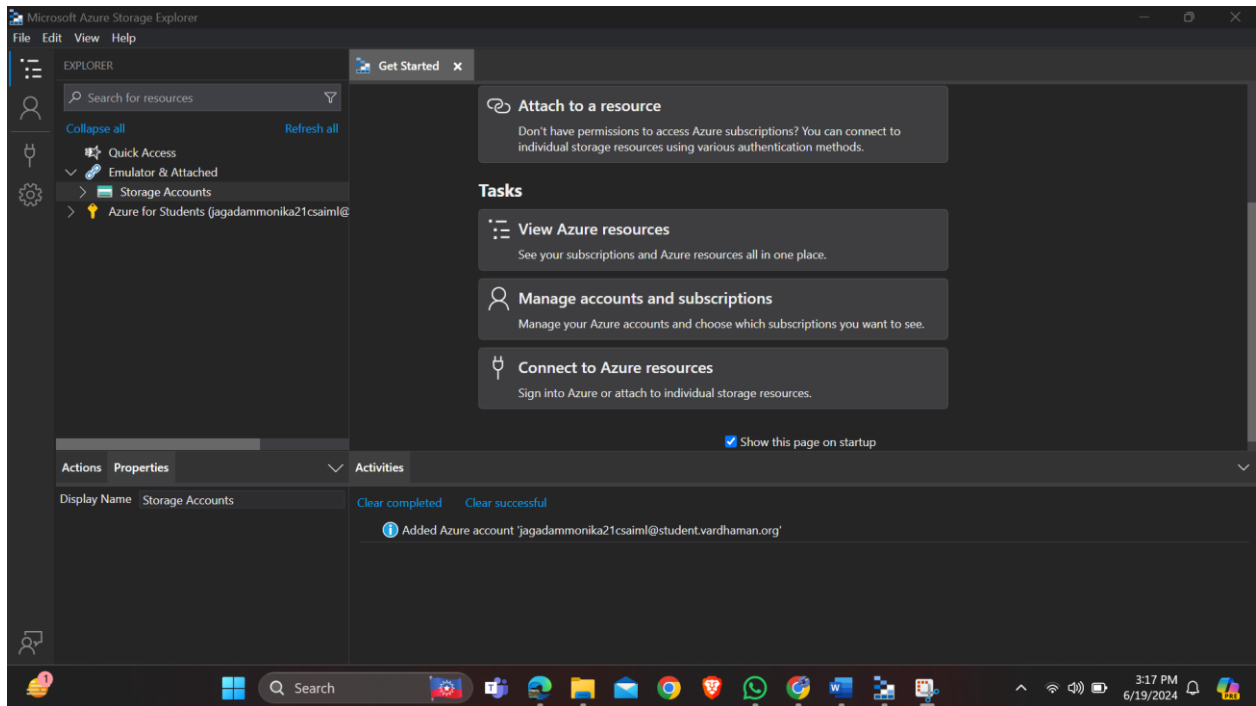
Step-4: After installation, launch Azure Storage Explorer.



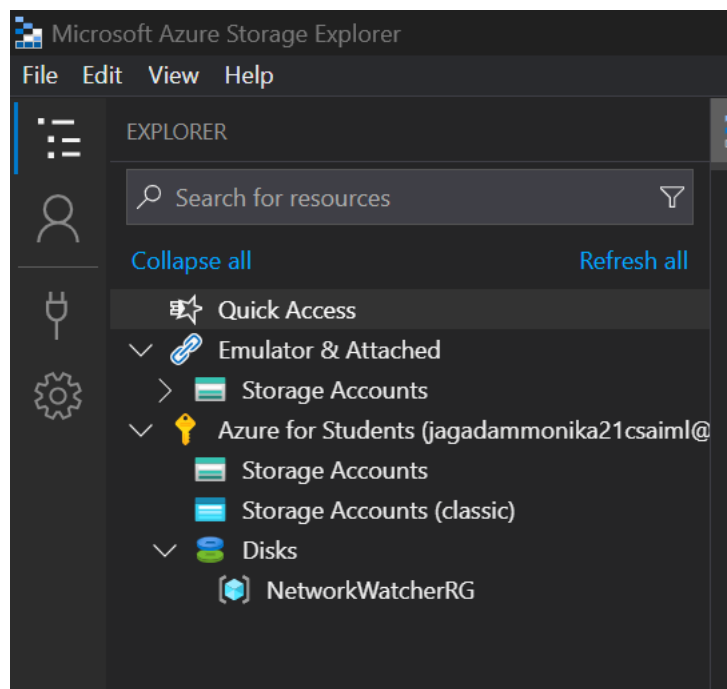
Step-5: In the "Accounts" tab, click on "Add an account."

Step-6: Sign in with your Azure credentials and select the desired subscription and storage account.

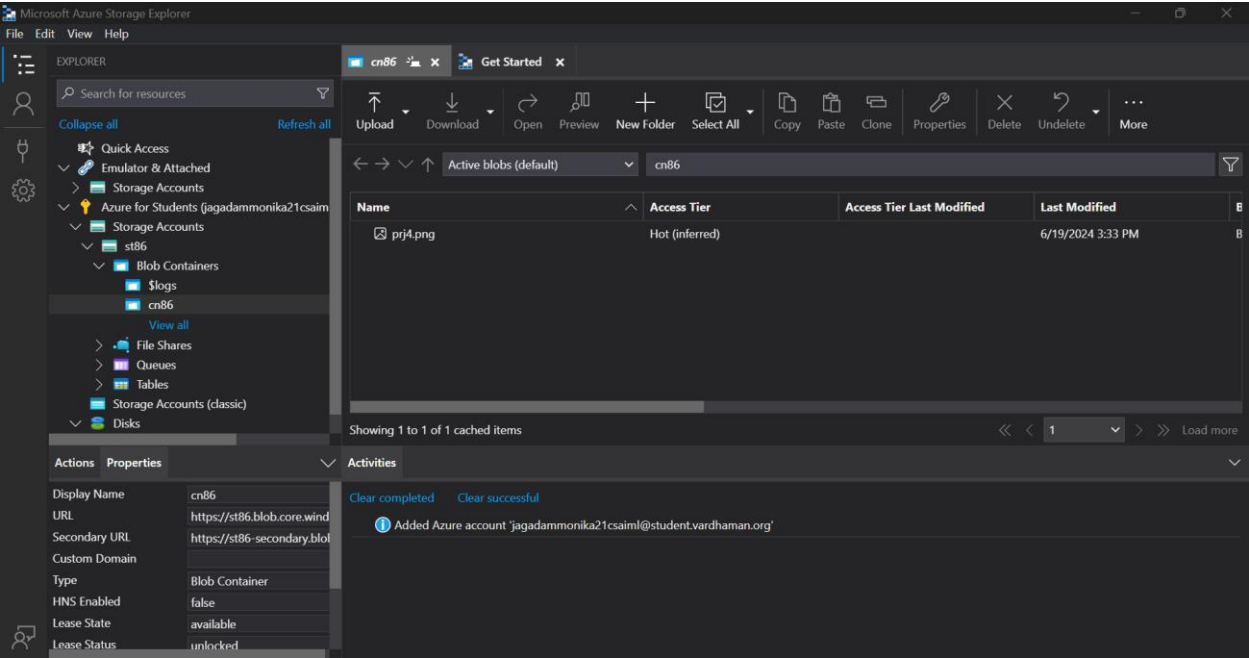
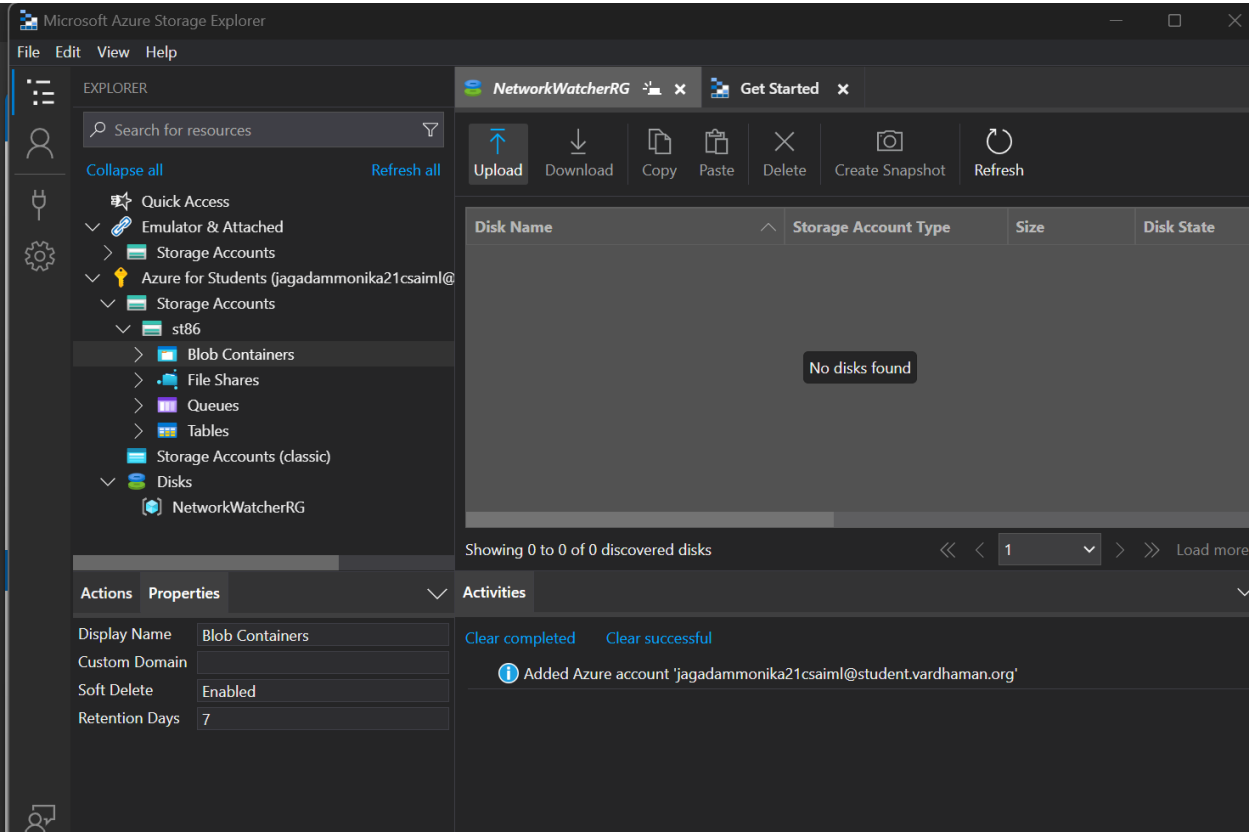
Step-7: Once connected, you will see the list of storage resources under your account.



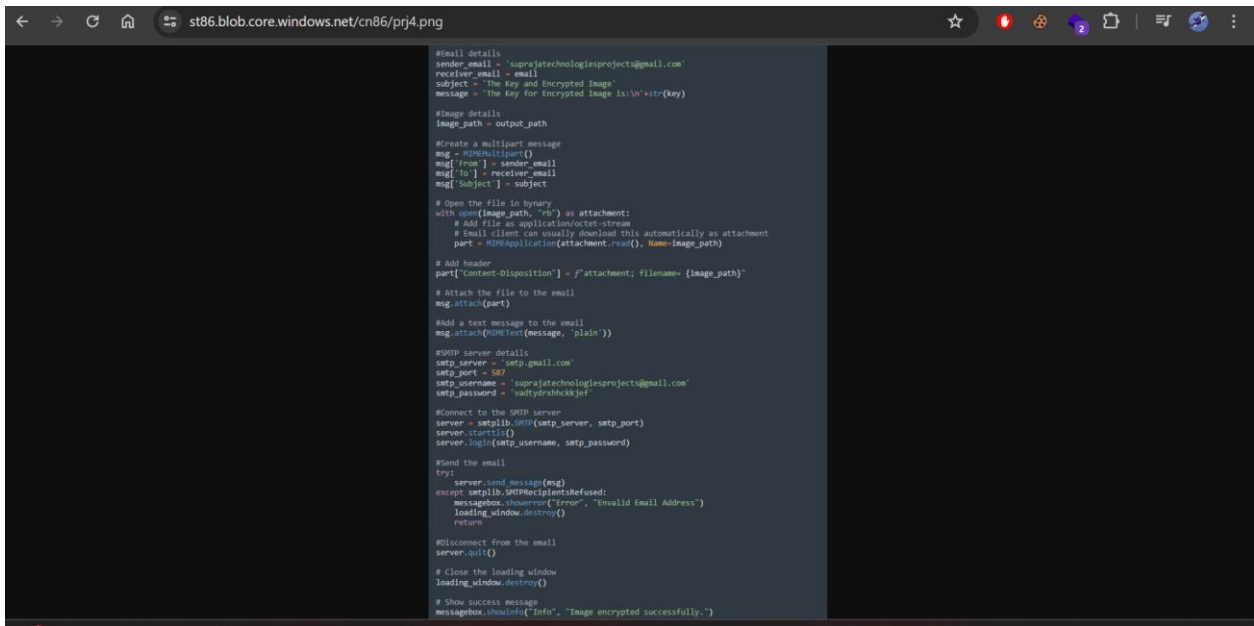
Step-8: Navigate to the specific storage resource you want to access, such as Blob Containers, File Shares, Queues, or Tables.



Step-9: To upload files, right-click on the container or folder where you want to upload and select "Upload."



Step-10: Select the file and click on provided URL to open the file to access the files and data.



The screenshot shows a web browser window with the address bar displaying 'st86.blob.core.windows.net/cn86/prj4.png'. The main content area contains a Python script for email encryption. The script defines email details, image details, and SMTP server details. It creates a multipart message, adds the image as an attachment, and sends the email. The script also includes error handling and a success message.

```
#Email details
sender_email = 'suprateltechnologiesproject@gmail.com'
receiver_email = 'mail'
subject = 'The Key and Encrypted Image'
message = 'The Key for Encrypted Image is:\n'+str(key)

#Image details
image_path = output_path

#Create a multipart message
msg = MIMEText(message)
msg['from'] = sender_email
msg['to'] = receiver_email
msg['subject'] = subject

# Open the file in binary
with open(image_path, "rb") as attachment:
    # Add file as application/octet-stream
    # Email client can usually download this automatically as attachment
    part = MIMEApplication(attachment.read(), name=image_path)

# Add header
part['Content-Disposition'] = f'attachment; filename={image_path}'

# Attach the file to the email
msg.attach(part)

# Add a text message to the email
msg.attach(MIMEText(message, 'plain'))

#SMTP server details
smtp_server = 'smtp.gmail.com'
smtp_port = 587
smtp_username = 'suprateltechnologiesproject@gmail.com'
smtp_password = 'suprprshbckjpf'

#Connect to the SMTP server
server = smtplib.SMTP(smtp_server, smtp_port)
server.starttls()
server.login(smtp_username, smtp_password)

#Send the email
try:
    server.send_message(msg)
except smtplib.SMTPRecipientsRefused:
    messagebox.showerror("Error", "Invalid Email Address")
    loading_window.destroy()
    return

#Disconnect from the email
server.quit()

# Close the loading window
loading_window.destroy()

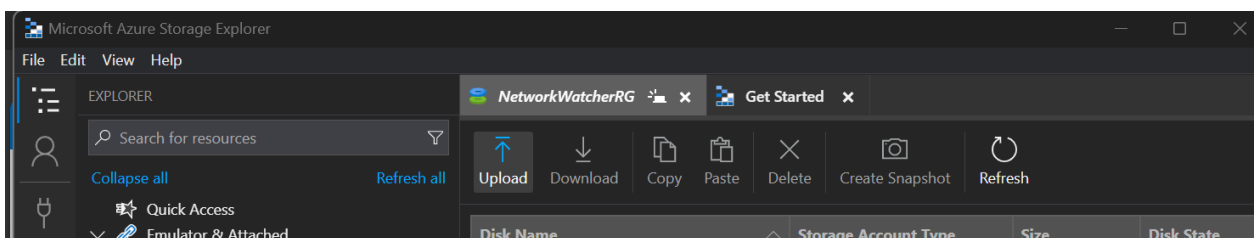
# Show success message
messagebox.showinfo("Info", "Image encrypted successfully.")
```

Step-10: To download files, right-click on the file you want to download and select "Download."

Step-11: To delete files, right-click on the file you want to delete and select "Delete."

Step-12: To view file properties, right-click on the file and select "Properties."

To filter storage resources, use the filter options provided. To refresh the storage resources, click on the "Refresh" button.



Step-17: To disconnect from an account, click on your account name and select "Disconnect."

Step-18: To exit Azure Storage Explorer, simply close the application.

## Chapter 3

# SOFTWARE REQUIREMENTS SPECIFICATION AND HARDWARE REQUIREMENTS

### 3.1 Software Requirements:

1. **Azure Account:**
  - An active Azure account with sufficient permissions to create and manage storage accounts, containers, and other Azure resources.
2. **Operating System:**
  - Windows 7 or later, macOS 10.12 or later, or Linux distributions such as Ubuntu, Fedora, and Debian.
3. **Azure Storage Explorer:**
  - Latest version of Azure Storage Explorer installed on the local machine.
4. **Azure CLI and SDKs:**
  - Azure Command Line Interface (CLI) and relevant Software Development Kits (SDKs) for scripting and automation.
5. **Security Software:**
  - SSL certificates for secure data transfer, firewall software, and anti-malware tools for enhanced security.

### 3.2 Hardware Requirements:

1. **Local Computer:**
  - **CPU:** Dual-core processor or better.
  - **Memory:** Minimum 4 GB RAM; 8 GB or more recommended for optimal performance.
  - **Storage:** Minimum 100 GB free disk space.
2. **Network Configuration:**
  - Reliable internet connection with a minimum upload/download speed of 5 Mbps.
3. **Security:**
  - Properly configured firewalls and antivirus software on the local machine to ensure secure access to Azure resources.

By meeting these software and hardware requirements, users can effectively utilize Azure Storage Explorer to manage their Azure storage resources.



## **CHAPTER 4**

### **RESULTS AND DISCUSSIONS**

The implementation of Azure Storage Explorer for data access and management has shown considerable improvements in operational efficiency and user satisfaction. Users quickly and securely connected to their Azure storage accounts, managing data seamlessly across Blob Containers, File Shares, Queues, and Tables.

The intuitive interface facilitated easy navigation and reduced the time required for data transfers. The ability to directly upload and download files improved transfer speeds compared to traditional methods. Managing access policies and permissions directly from the tool simplified security administration and user access management.

Security integration was robust, with SSL certificates ensuring secure data transfers and comprehensive firewall and anti-malware tools protecting data throughout the process. The use of Azure CLI and SDKs enabled automation and scripting, further enhancing efficiency.

Some initial setup and configuration challenges were noted, mainly due to unfamiliarity with Azure services. This was mitigated by providing comprehensive documentation and training resources.

In summary, Azure Storage Explorer proved to be a valuable tool for managing Azure storage resources, offering speed, security, and user-friendly features. It streamlined operations, improved data management practices, and enhanced security, making it a highly recommended solution for organizations using Azure storage services.

## CHAPTER 5

### CONCLUSION AND FUTURE SCOPE

#### Conclusion:

The use of Azure Storage Explorer has greatly improved data access and management within Azure storage services. Its user-friendly interface and robust functionality have streamlined operations, enhanced data transfer speeds, and simplified security management. Despite initial setup challenges, comprehensive documentation and training facilitated smooth adoption. Overall, Azure Storage Explorer is a valuable tool for managing Azure storage resources, offering efficiency, security, and ease of use.

#### Future Scope:

Future enhancements for Azure Storage Explorer include:

1. **Advanced Automation:** Integrating more sophisticated automation scripts using Azure CLI and SDKs.
2. **Enhanced Security:** Implementing multi-factor authentication (MFA) and advanced encryption methods.
3. **Cross-Platform Compatibility:** Expanding support for more operating systems and devices.
4. **Integration with Azure Services:** Deeper integration with services like Azure Data Factory and Azure Synapse Analytics.
5. **User Training:** Developing extensive training programs and support resources.

These improvements will ensure Azure Storage Explorer continues to provide significant value to its users.

## References

- [1] Microsoft. "Azure Storage Explorer Documentation." Microsoft Docs.  
<https://docs.microsoft.com/en-us/azure/storage/common/storage-explorer>
- [2] Microsoft. "Download Azure Storage Explorer." Microsoft Azure.  
<https://azure.microsoft.com/en-us/features/storage-explorer/>
- [3] Microsoft. "Azure Command-Line Interface (CLI) Documentation." Microsoft Docs.  
<https://docs.microsoft.com/en-us/cli/azure/>
- [4] Microsoft. "Azure Security Documentation." Microsoft Docs.  
<https://docs.microsoft.com/en-us/azure/security/>
- [5] Microsoft. "Integrate with Azure Services." Microsoft Azure.  
<https://azure.microsoft.com/en-us/services/>