INTERN PROJECT REPORT

Azure Blob Storage – Static Website Hosting & Lifecycle Management

Submitted by:
Oleti Raghu Sai Varun

Internship:
Smarted Innovations
(Training by Embrizon Team)

Date: 10-06-2025

Project Report

Azure Blob Storage – Static Website Hosting & Lifecycle Management

Prepared by: Oleti Raghu Sai Varun

Internship: Smarted Innovations (Training by Embrizon Team)

Date: [10-06-2025]

DECLARATION

I, Raghu Sai Varun, hereby declare that the project report titled “Azure Blob Storage – Static Website Hosting & Lifecycle Management” is the result of my own work carried out during my internship with Smarted Innovations (Training by Embrizon Team). This project has been completed under the guidance provided and is submitted as a part of the internship deliverables. The work is original and has not been submitted elsewhere for any academic or professional purpose.

Signature: Oleti Raghu Sai Varun

Date: 10-06-2025

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to Smarted Innovations and the Embrizon Training Team for providing me with the opportunity to undertake this internship project.

I sincerely thank my mentors and trainers at Embrizon for their continuous support, valuable guidance, and timely feedback throughout the duration of this project. Their inputs have helped me gain practical exposure to cloud technologies like Microsoft Azure.

I would also like to extend my appreciation to my family and friends who supported me during this journey.

This internship has been a great learning experience and a significant step toward my professional development.

Raghu Sai Varun
[10-06-2025]

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Cloud computing has become a cornerstone in modern IT infrastructure. Microsoft Azure offers various services, including Azure Blob Storage, which enables scalable and secure storage of unstructured data. This project focuses on using Azure Blob Storage to host a static website and apply data lifecycle management policies. It demonstrates how to use blob containers, configure public access, and apply cost-saving strategies using tiering policies.

CHAPTER 2: OBJECTIVE

The objective of this project is to demonstrate the use of Azure Blob Storage for hosting a static website. The key goals include:

- Creating a storage account
- Enabling static website hosting
- Uploading web content files (index.html and error.html)
- Making the content publicly accessible
- Creating a lifecycle management rule to move blobs to the cool tier after 100 days of inactivity.

CHAPTER 3: TOOLS AND TECHNOLOGIES USED

The project was implemented using the following tools:

- Microsoft Azure Portal: User interface to access and manage Azure resources.
- Azure Blob Storage: Object storage solution for the cloud.
- HTML: Used to create static web content.
- Web browser: To access and validate the hosted website.
- Lifecycle Policy Engine: Built-in Azure feature to automate blob tier transitions.

CHAPTER 4: ARCHITECTURE OVERVIEW

The architecture consists of a single Azure Storage Account with static website hosting enabled. When enabled, a special container named '$web' is created, which stores web files. These files are served via a public endpoint. Azure automatically provisions a

global URL to make these files accessible. Lifecycle policies are then applied to manage blob storage cost by transitioning infrequently accessed blobs to cooler storage tiers.

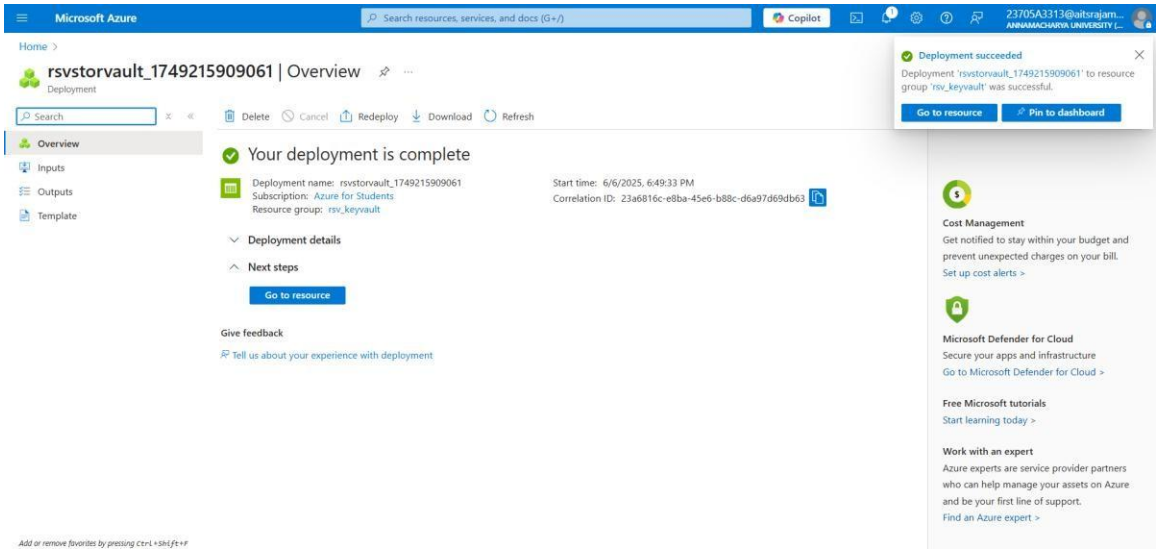


Figure 4.1: Azure Storage Account Dashboard Overview

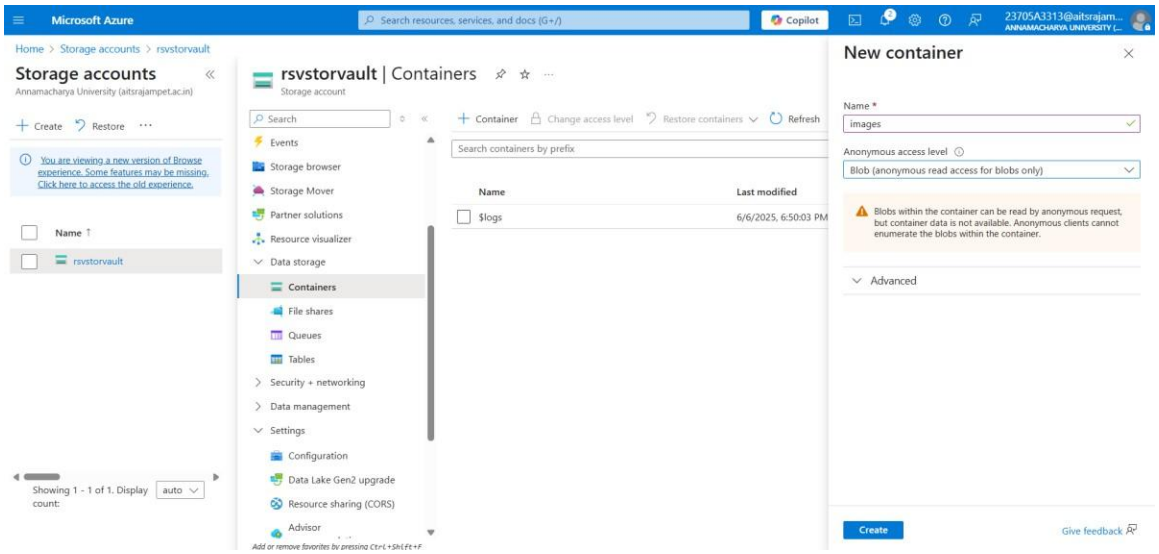


Figure 4.2: Image File Uploaded to Blob Storage Container

CHAPTER 5: IMPLEMENTATION

The implementation followed these steps:

1. Created a new Azure Storage Account.

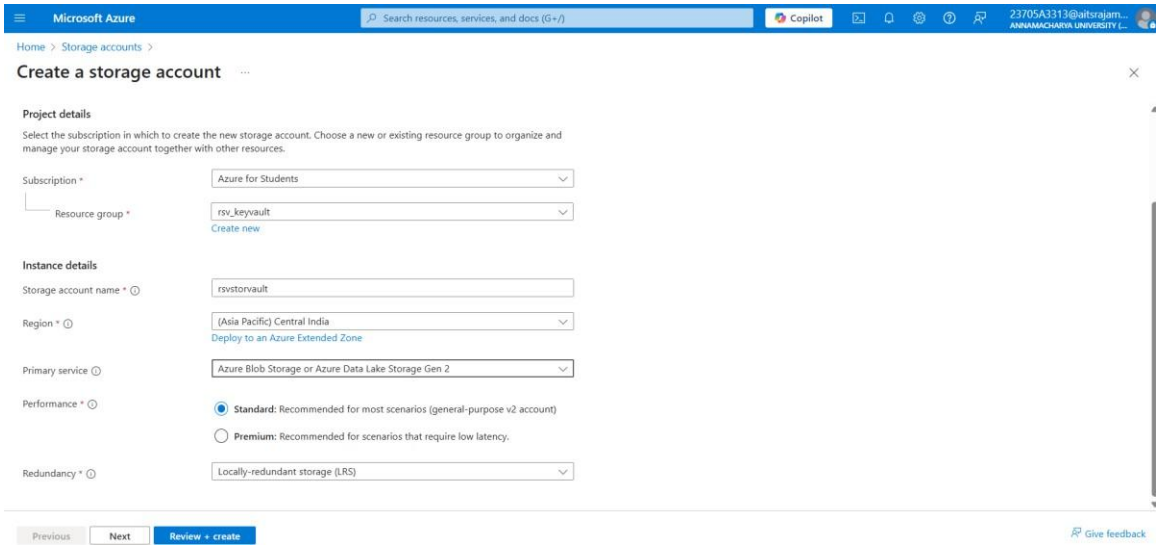


Figure 5.1: Creating a New Azure Storage Account

2. Navigated to 'Static website' in the settings and enabled it.

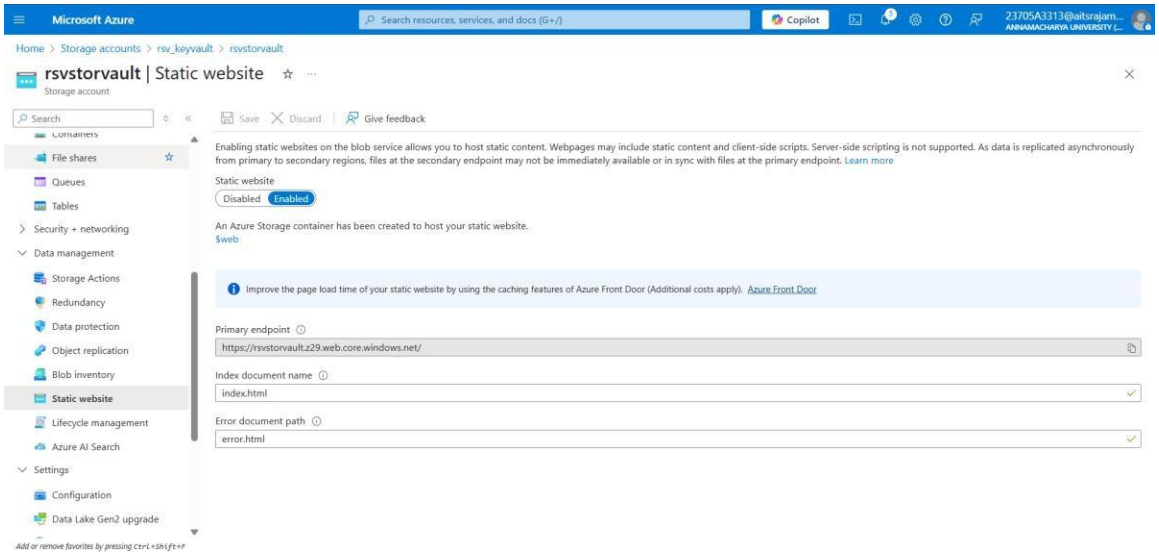


Figure 5.2: Enabling Static Website Feature in Azure Portal

3. Specified 'index.html' and 'error.html' as the document paths.

File Edit View

```
<!DOCTYPE html>
<html>
<head>
    <title>Welcome to My Static Website</title>
</head>
<body style="text-align:center; margin-top:50px;">
    <h1>Welcome to My Azure Static Website!</h1>
    <p>This page is served from an Azure Blob Storage container.</p>
</body>
</html>
```

File Edit View

```
<!DOCTYPE html>
<html>
<head>
    <title>Page Not Found</title>
</head>
<body style="text-align:center; margin-top:50px; color:red;">
    <h1>404 - Page Not Found</h1>
    <p>The page you're looking for doesn't exist.</p>
</body>
</html>
```

Figure 5.3: 'index.html' and 'error.html'

4. Uploaded HTML files into the '$web' container.

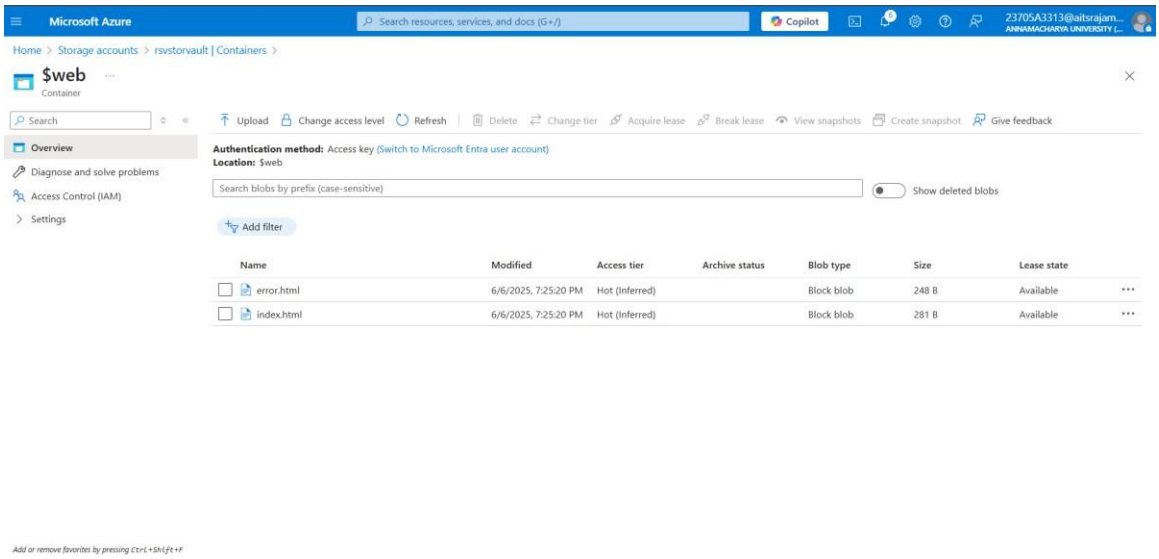


Figure 5.4: HTML Files Uploaded to Azure Blob Storage

5. Accessed the public URL to test static website functionality.

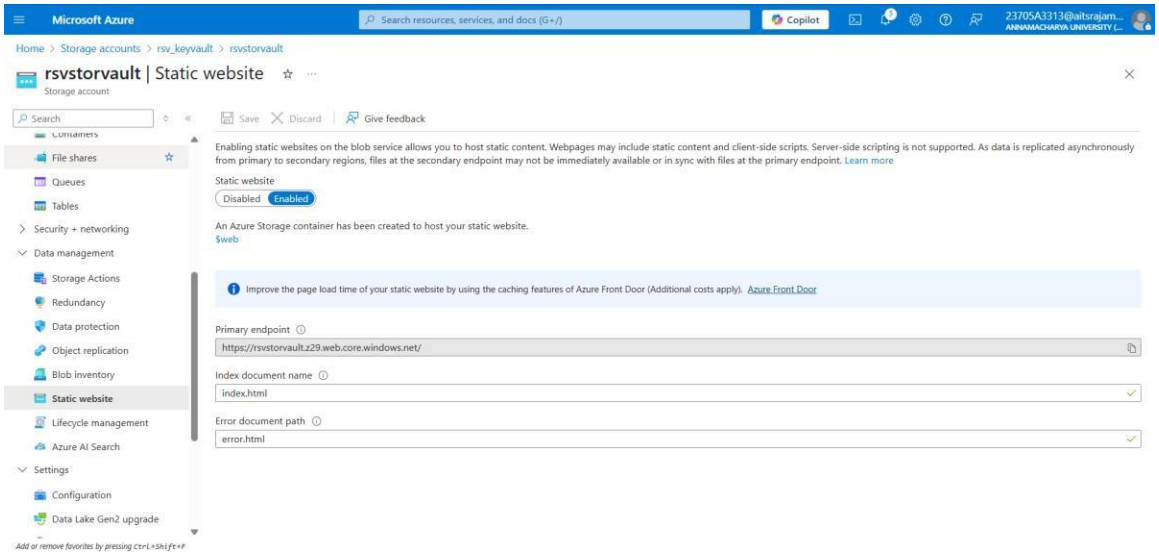


Figure 5.5: Static Website Successfully Hosted and Accessed via URL

6. Configured lifecycle rules to move blobs to the cool tier if not accessed in 100 days.

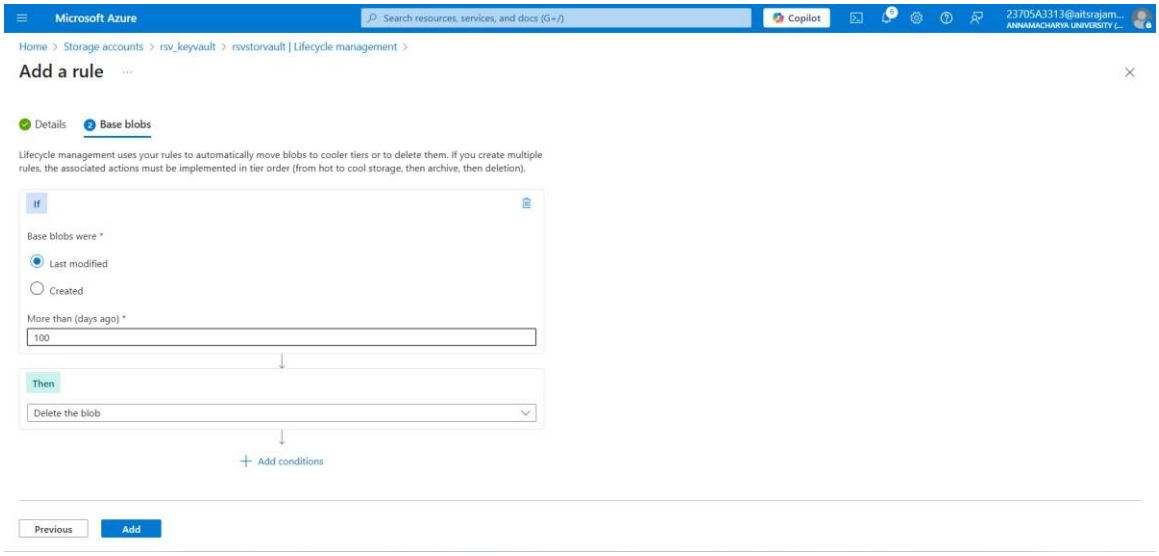


Figure 5.6: Lifecycle Rule Configuration – Final Rule Summary

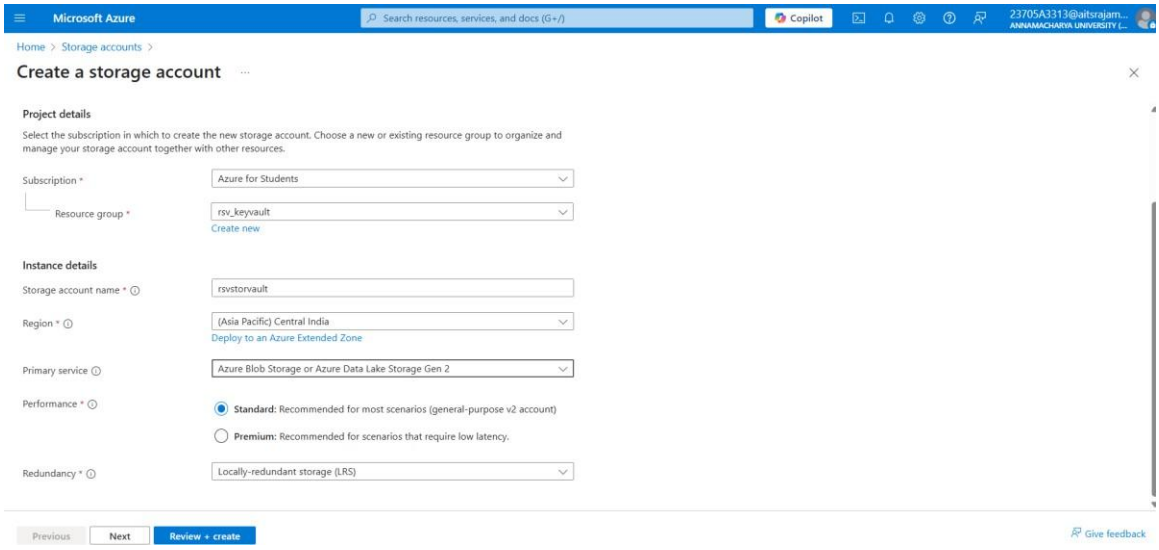
CHAPTER 6: RESULTS

The static website became publicly accessible through the Azure-generated endpoint. Visitors could load the main page and see the error page when navigating to invalid paths. Lifecycle management rules were successfully implemented, allowing for cost optimization over time.

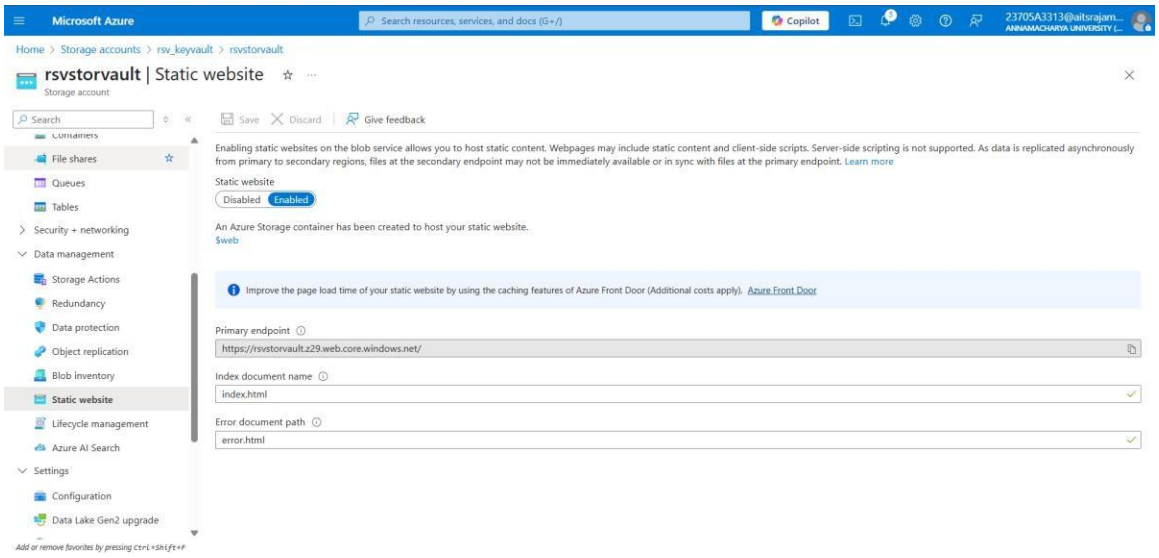
CHAPTER 7: SCREENSHOTS

Screenshots include:

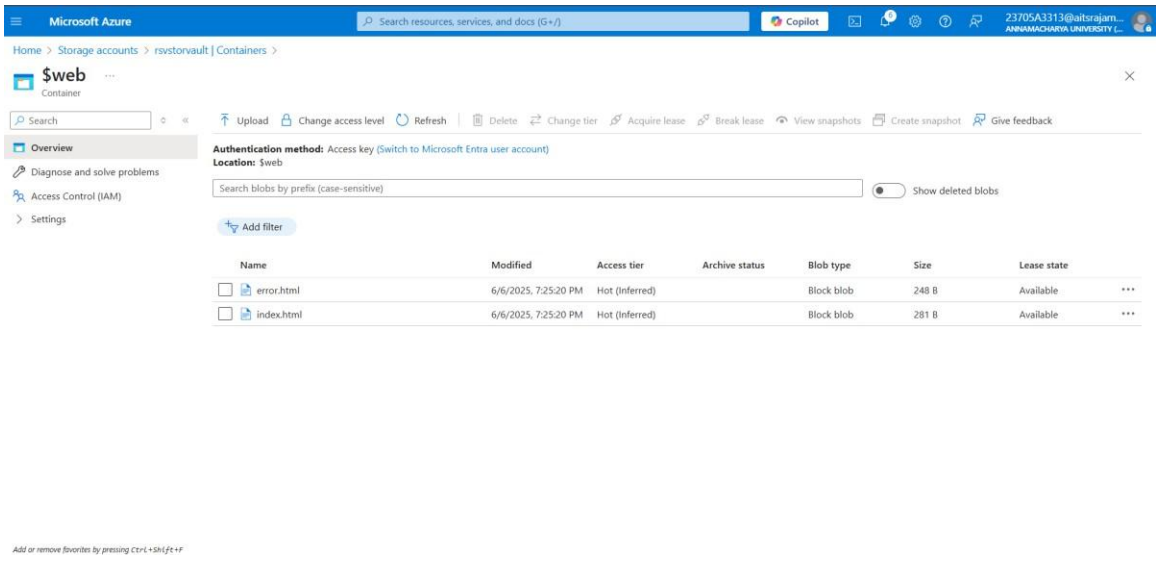
1. Storage Account Overview



2. Static Website Settings



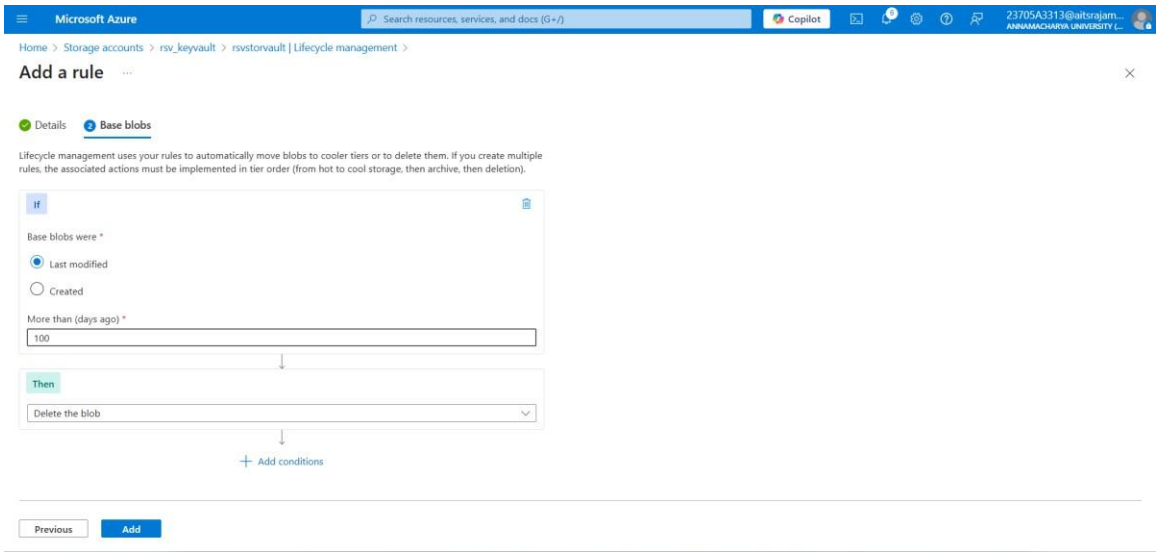
3. Uploaded Files in '$web' Container



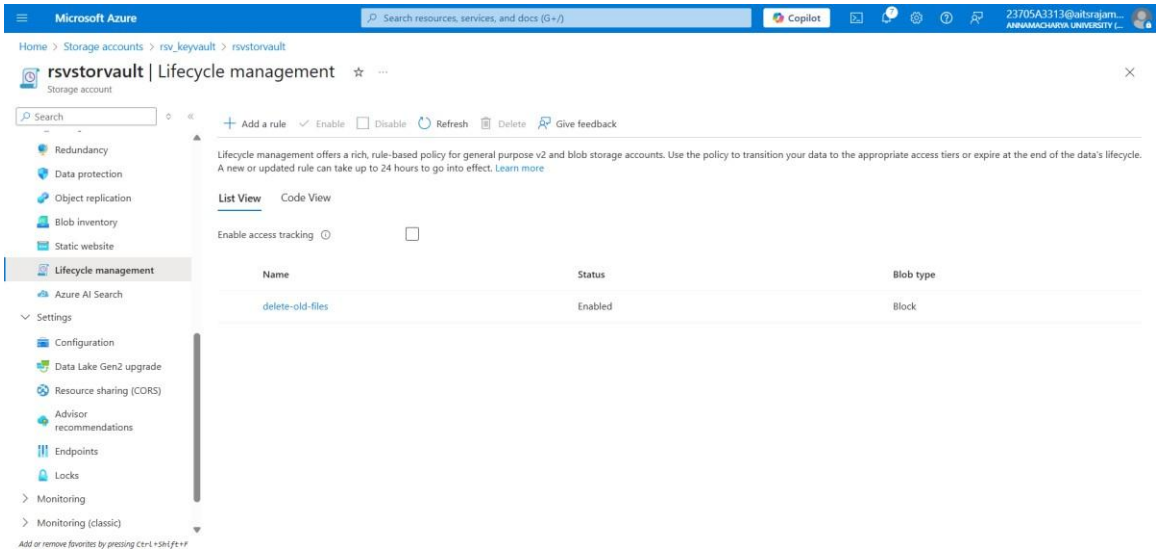
4. Website Loading in Browser

5. Error Page Display

6. Lifecycle Rule Configuration – Conditions



7. Lifecycle Rule Configuration – Actions



CHAPTER 8: CONCLUSION

This project provided hands-on experience with Azure Blob Storage and lifecycle management. It helped reinforce concepts of cloud storage, public access configuration, and resource optimization. The project is applicable for small-scale static web hosting with cost control features.

CHAPTER 9: FUTURE ENHANCEMENTS

Possible future improvements include:

- Adding JavaScript and CSS for dynamic content.
- Integrating with Azure CDN for faster delivery.
- Securing the endpoint with HTTPS and custom domains.
- Automating file uploads and version control through DevOps pipelines.

REFERENCES

1. Microsoft Learn – Quickstart: Create a storage account in the Azure portal
https://learn.microsoft.com/en-us/azure/storage/common/storage-account-create?tabs=azure-portal
2. Microsoft Learn – Host a static website in Azure Storage
https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blob-static-website-how-to?tabs=azure-portal
3. Microsoft Learn – Configure Azure Blob Storage lifecycle management

https://learn.microsoft.com/en-us/azure/storage/blobs/lifecycle-management-policy-configure?tabs=azure-portal

4. Azure Documentation – Azure Storage Account Overview https://learn.microsoft.com/en-us/azure/storage/common/storage-account-overview
5. Azure Documentation – Public access to blob data https://learn.microsoft.com/en-us/azure/storage/blobs/anonymous-read-access-configure?tabs=portal