# Lab 6: Learning Azure Blob Storage Concept

## Objectives

1. Understand the concept of Azure Blob Storage.
2. Create and configure an Azure Storage Account.
3. Enable static website hosting using Blob Storage.
4. Upload and access static web content using Blob endpoints.
5. Configure public access for blob containers.
6. Create and manage containers for data storage.

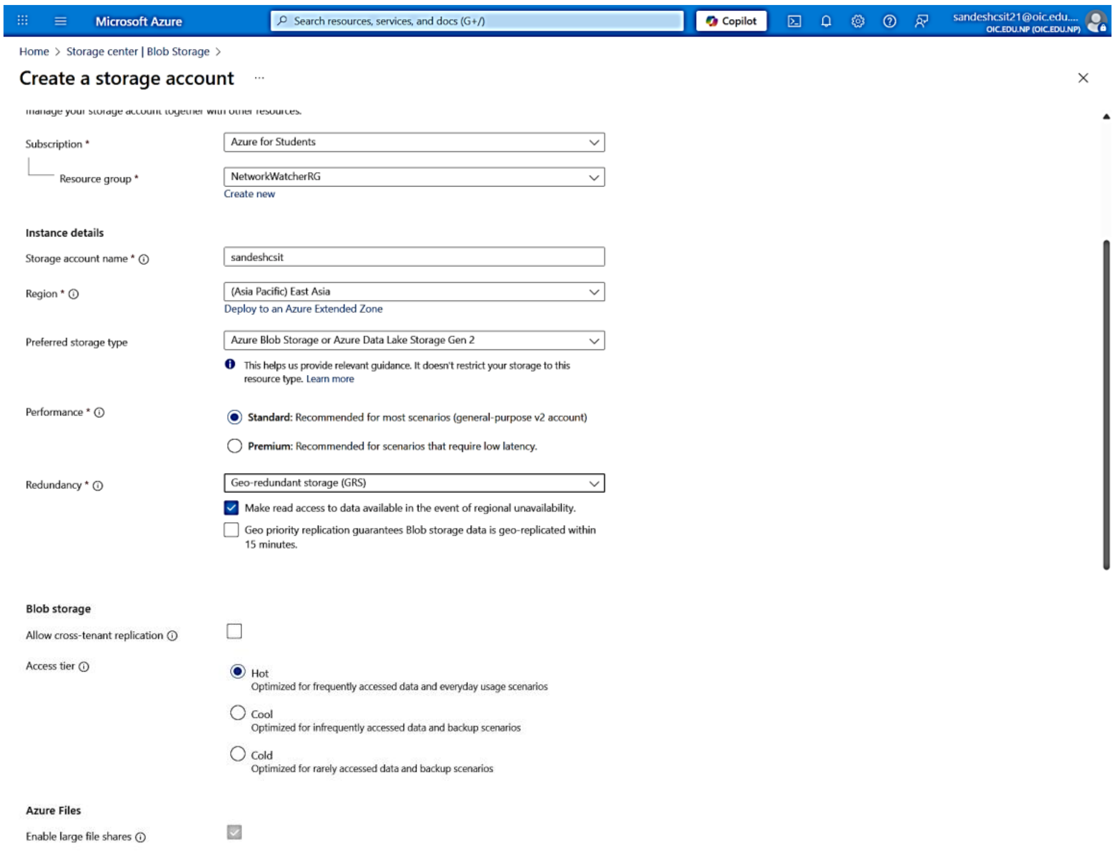
## Tools and Technologies Used

1. Microsoft Azure Portal
2. Azure Storage Account
3. Azure Blob Storage / Azure Data Lake Storage Gen2
4. Static HTML files (index.html, error.html)

## Procedure

### Step 1: Create a Storage Account

1. Log in to the Azure Portal.
2. Navigate to: Home > Storage Accounts
3. Configure the storage account with the following settings:
   a. Storage Account Name: sandeshcsit
   b. Performance: Standard
   c. Preferred Storage Type: Azure Blob Storage or Azure Data Lake Storage Gen2


Home > Storage center | Blob Storage >

Create a storage account

manage your storage account together with other resources.

Subscription *: Azure for Students

Resource group *: NetworkWatcherRG

Create new

**Instance details**

Storage account name *: sandeshcsit

Region *: (Asia Pacific) East Asia

Deploy to an Azure Extended Zone

Preferred storage type: Azure Blob Storage or Azure Data Lake Storage Gen 2

This helps us provide relevant guidance. It doesn't restrict your storage to this resource type. Learn more

Performance *: [✓ **Standard:** Recommended for most scenarios (general-purpose v2 account)] **Premium:** Recommended for scenarios that require low latency.

Redundancy *: Geo-redundant storage (GRS)

[✓] Make read access to data available in the event of regional unavailability.

[ ] Geo priority replication guarantees Blob storage data is geo-replicated within 15 minutes.

**Blob storage**

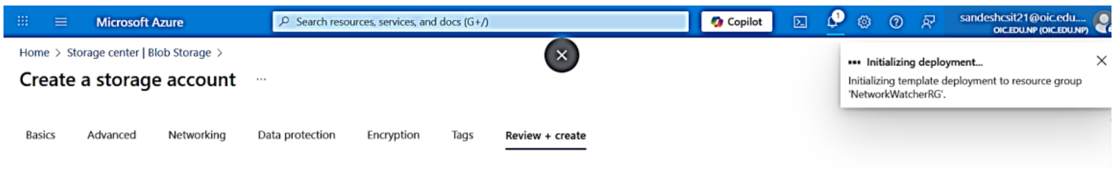
Allow cross-tenant replication: [ ]

Access tier: [✓ Hot — Optimized for frequently accessed data and everyday usage scenarios] Cool — Optimized for infrequently accessed data and backup scenarios; Cold — Optimized for rarely accessed data and backup scenarios

**Azure Files**

Enable large file shares: [✓]

4. Review the configuration and click Deploy.


Home > Storage center | Blob Storage >

Create a storage account

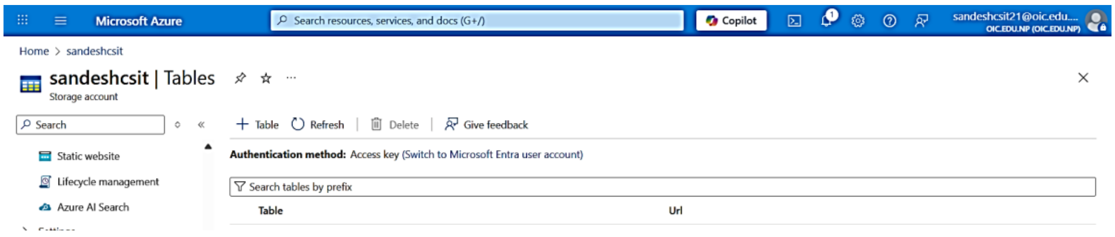
Basics | Advanced | Networking | Data protection | Encryption | Tags | **Review + create**

••• Initializing deployment...

Initializing template deployment to resource group 'NetworkWatcherRG'.

## Step 2: Enable Static Website Hosting

1. After deployment, open the storage account.
2. Navigate to: Storage account > data management > static website


Home > sandeshcsit

sandeshcsit | Tables

Storage account

**Authentication method:** Access key (Switch to Microsoft Entra user account)

Search tables by prefix

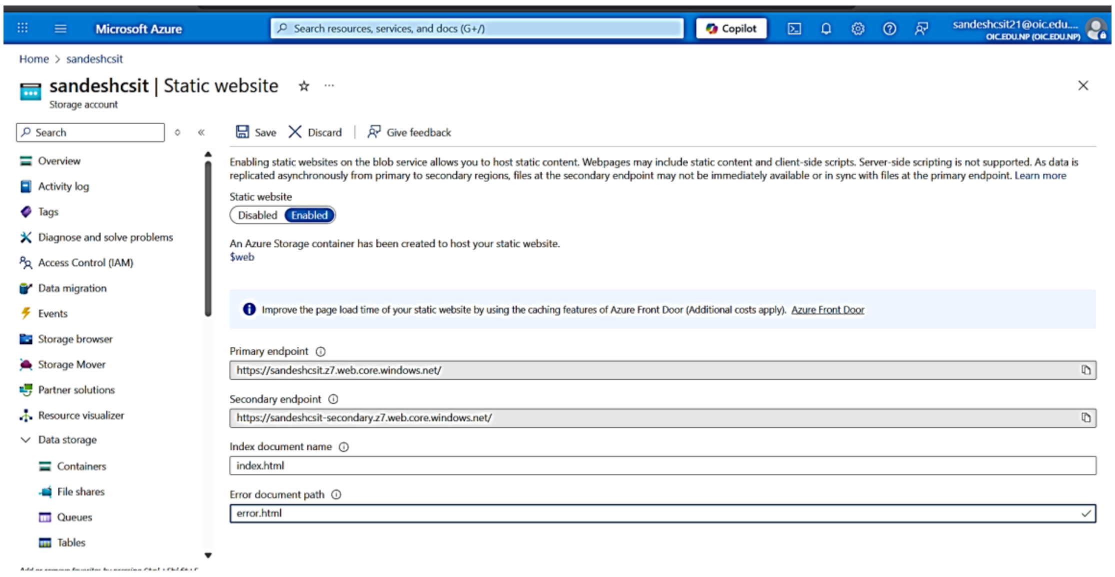
| Table | Url |
|---|---|

3. Enable Static Website.
4. Specify:

   Index document: index.html

Error document: error.html

5. Save the configuration.



sandeshcsit | Static website

Enabling static websites on the blob service allows you to host static content. Webpages may include static content and client-side scripts. Server-side scripting is not supported. As data is replicated asynchronously from primary to secondary regions, files at the secondary endpoint may not be immediately available or in sync with files at the primary endpoint. Learn more

Static website: Disabled [✓ Enabled]

An Azure Storage container has been created to host your static website.
$web

Improve the page load time of your static website by using the caching features of Azure Front Door (Additional costs apply). Azure Front Door

Primary endpoint: https://sandeshcsit.z7.web.core.windows.net/

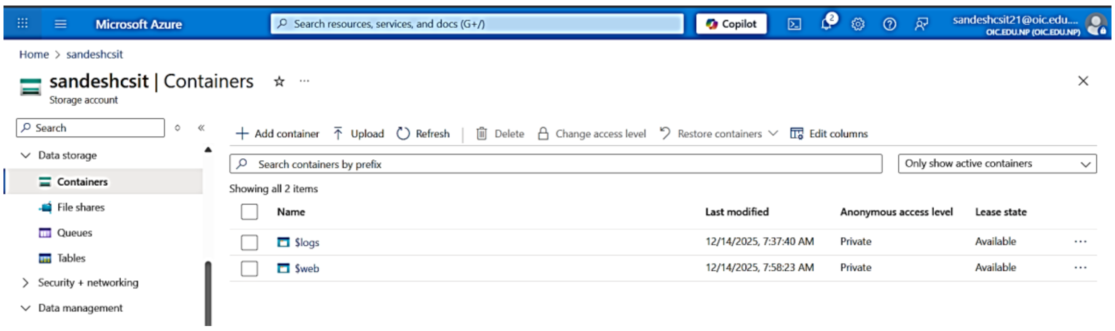
Secondary endpoint: https://sandeshcsit-secondary.z7.web.core.windows.net/

Index document name: index.html

Error document path: error.html

## Step 3: Upload Static Website Files

1. Open the $web container.
2. Upload the following files:
    a. index.html
    b. error.html
3. Once uploaded, access the website using the provided Primary Endpoint URL.
4. The static website is now live and accessible through the browser.



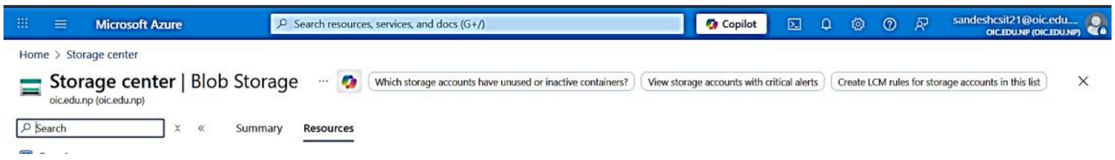
sandeshcsit | Containers

Showing all 2 items

| Name | Last modified | Anonymous access level | Lease state |
|---|---|---|---|
| $logs | 12/14/2025, 7:37:40 AM | Private | Available |
| $web | 12/14/2025, 7:58:23 AM | Private | Available |

**Step 4: Working with Blob Containers**

1. Navigate to: Data storage > Containers
2. Create a new container.
3. Upload files (documents, images, or HTML content).
4. Each uploaded blob generates a unique Blob Endpoint URL.


Microsoft Azure | Search resources, services, and docs (G+/) | Copilot | sandeshcsit21@oic.edu... OIC.EDU.NP (OIC.EDU.NP)

Home > Storage center

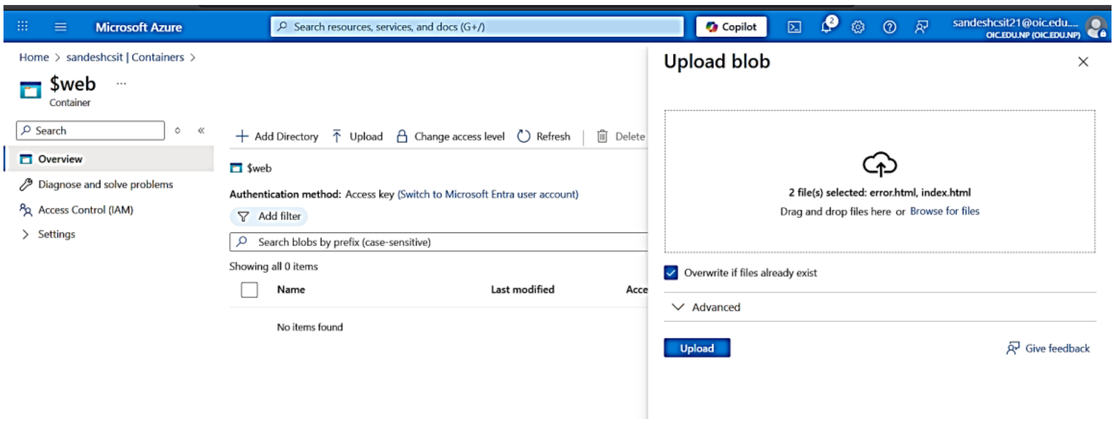
Storage center | Blob Storage
oic.edu.np (oic.edu.np)

Which storage accounts have unused or inactive containers? | View storage accounts with critical alerts | Create LCM rules for storage accounts in this list

Search | Summary | Resources

**Step 5: Access Uploaded Content**

1. Copy the blob URL of an uploaded file.
2. Paste it into a browser.
3. The content should load successfully if public access is enabled.
4. This confirms correct container and access configuration.


Microsoft Azure | Search resources, services, and docs (G+/) | Copilot | sandeshcsit21@oic.edu... OIC.EDU.NP (OIC.EDU.NP)

Home > sandeshcsit | Containers >

$web
Container

Search | Overview | Diagnose and solve problems | Access Control (IAM) | Settings

Add Directory | Upload | Change access level | Refresh | Delete

$web

Authentication method: Access key (Switch to Microsoft Entra user account)

Add filter

Search blobs by prefix (case-sensitive)

Showing all 0 items

| Name | Last modified | Acce |
|---|---|---|
| No items found | | |

Upload blob

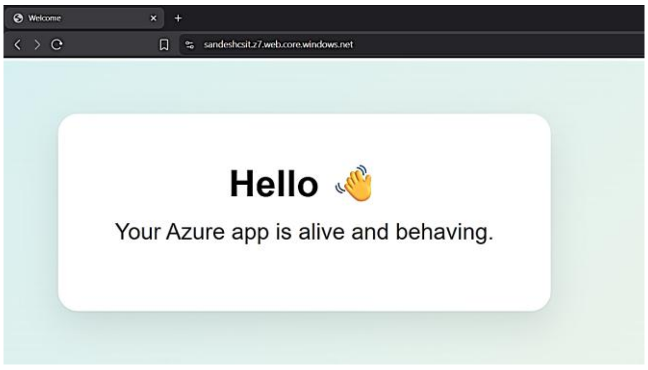
2 file(s) selected: error.html, index.html
Drag and drop files here or Browse for files

☑ Overwrite if files already exist

Advanced

Upload | Give feedback

Endpoint (index.html):

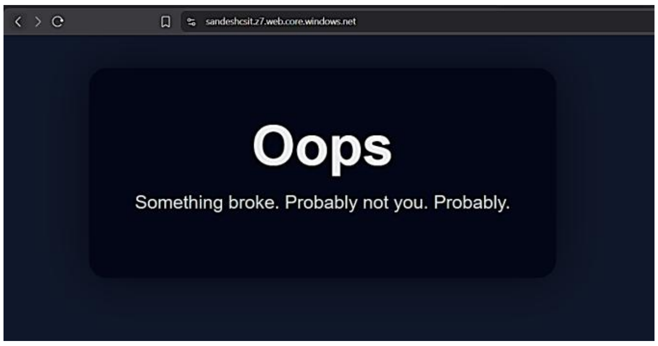

Welcome

sandeshcsit.z7.web.core.windows.net

Hello 👋

Your Azure app is alive and behaving.

Endpoint (Error.html):

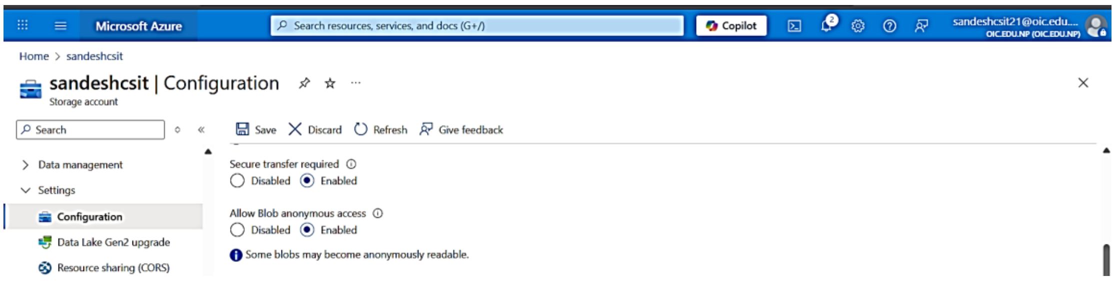


Oops

Something broke. Probably not you. Probably.

## Step 6: Enable Public Access to Blob Storage

1. By default, blob access may be restricted.
2. Go to: Settings > Configuration
3. Enable: Allow Blob anonymous access
4. Save the configuration.
5. This allows public access to blobs via URLs.
6. Change storage account to public



sandeshcsit | Configuration
Storage account

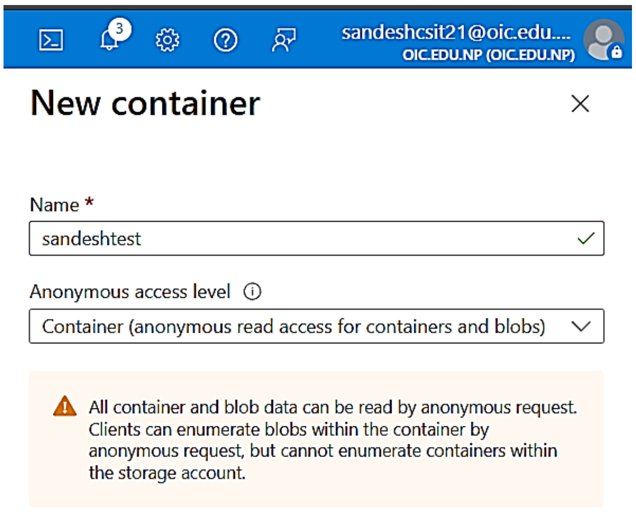
Secure transfer required
Disabled [✓ Enabled]

Allow Blob anonymous access
Disabled [✓ Enabled]

Some blobs may become anonymously readable.

## Step 7: Create public static file

a. Create Container



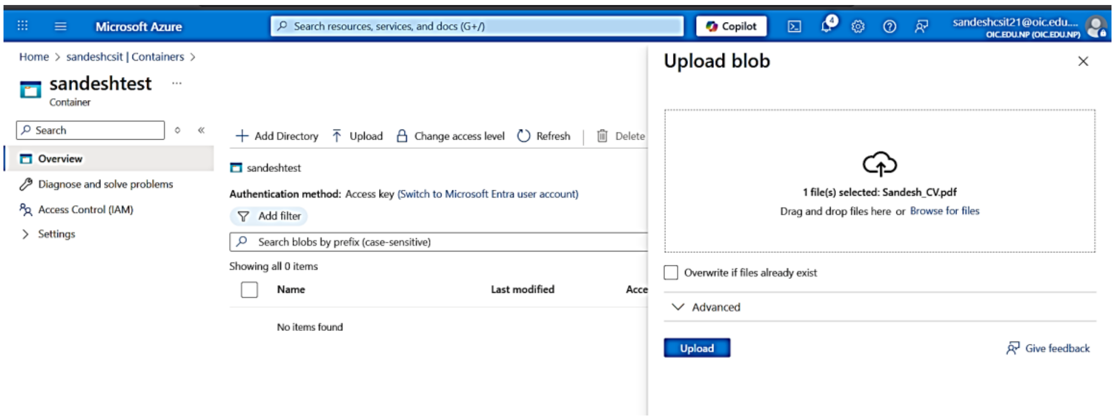
New container

Name *
sandeshtest

Anonymous access level
Container (anonymous read access for containers and blobs)

All container and blob data can be read by anonymous request. Clients can enumerate blobs within the container by anonymous request, but cannot enumerate containers within the storage account.

b. Upload content:



sandeshtest

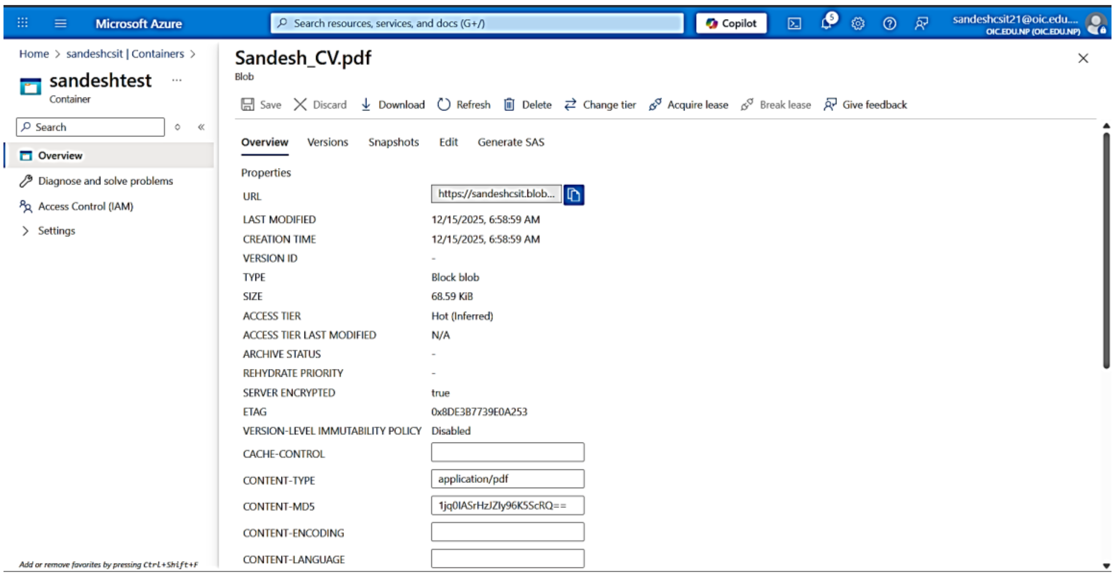
Authentication method: Access key (Switch to Microsoft Entra user account)

Showing all 0 items

No items found

1 file(s) selected: Sandesh_CV.pdf

Uploaded file:



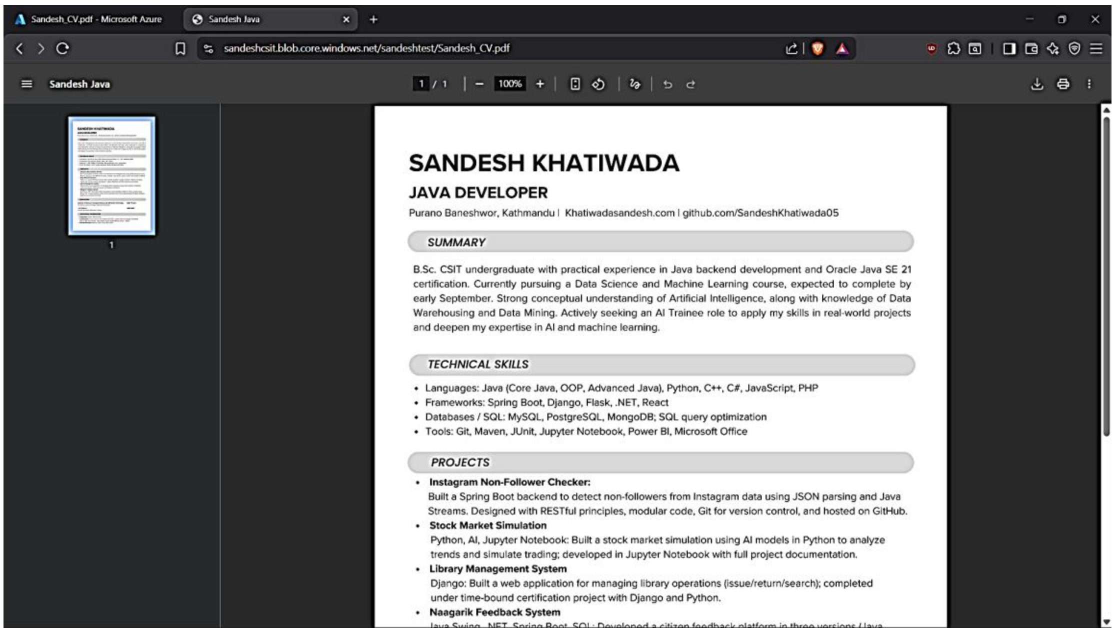
Sandesh_CV.pdf

Blob

Properties

| | |
|---|---|
| URL | https://sandeshcsit.blob... |
| LAST MODIFIED | 12/15/2025, 6:58:59 AM |
| CREATION TIME | 12/15/2025, 6:58:59 AM |
| VERSION ID | - |
| TYPE | Block blob |
| SIZE | 68.59 KiB |
| ACCESS TIER | Hot (Inferred) |
| ACCESS TIER LAST MODIFIED | N/A |
| ARCHIVE STATUS | - |
| REHYDRATE PRIORITY | - |
| SERVER ENCRYPTED | true |
| ETAG | 0x8DE3B7739E0A253 |
| VERSION-LEVEL IMMUTABILITY POLICY | Disabled |
| CACHE-CONTROL | |
| CONTENT-TYPE | application/pdf |
| CONTENT-MD5 | 1jq0lASrHzJZly96K5ScRQ== |
| CONTENT-ENCODING | |
| CONTENT-LANGUAGE | |

c. Accessing the url:



sandeshcsit.blob.core.windows.net/sandeshtest/Sandesh_CV.pdf

SANDESH KHATIWADA

JAVA DEVELOPER

Purano Baneshwor, Kathmandu | Khatiwadasandesh.com | github.com/SandeshKhatiwada05

*SUMMARY*

B.Sc. CSIT undergraduate with practical experience in Java backend development and Oracle Java SE 21 certification. Currently pursuing a Data Science and Machine Learning course, expected to complete by early September. Strong conceptual understanding of Artificial Intelligence, along with knowledge of Data Warehousing and Data Mining. Actively seeking an AI Trainee role to apply my skills in real-world projects and deepen my expertise in AI and machine learning.

*TECHNICAL SKILLS*

- Languages: Java (Core Java, OOP, Advanced Java), Python, C++, C#, JavaScript, PHP
- Frameworks: Spring Boot, Django, Flask, .NET, React
- Databases / SQL: MySQL, PostgreSQL, MongoDB; SQL query optimization
- Tools: Git, Maven, JUnit, Jupyter Notebook, Power BI, Microsoft Office

*PROJECTS*

- **Instagram Non-Follower Checker:**
  Built a Spring Boot backend to detect non-followers from Instagram data using JSON parsing and Java Streams. Designed with RESTful principles, modular code, Git for version control, and hosted on GitHub.
- **Stock Market Simulation**
  Python, AI, Jupyter Notebook: Built a stock market simulation using AI models in Python to analyze trends and simulate trading; developed in Jupyter Notebook with full project documentation.
- **Library Management System**
  Django: Built a web application for managing library operations (issue/return/search); completed under time-bound certification project with Django and Python.
- **Naagarik Feedback System**
  Java Swing, .NET, Spring Boot, SQL: Developed a citizen feedback platform in three versions (Java

## Conclusion:

This lab demonstrated the core concepts of Azure Blob Storage, including storage account creation, static website hosting, container management, and public blob access. By enabling static website hosting and uploading HTML files, Azure Blob Storage was successfully used to host web content without a traditional web server. This lab highlights Blob Storage as a scalable, cost-effective solution for storing and serving unstructured data and static websites.