Name : Hamza Jamshaid

Project tittle :

Implement a Cloud-Based File Storage System

Introduction:

Intern management systems often require secure storage and controlled access to sensitive documents such as joining letters, certificates, and reports. Traditional local storage solutions lack scalability, security, and remote accessibility. To address these challenges, this project implements a **Cloud-Based File Storage System** using **Microsoft Azure Blob Storage**, providing a secure, scalable, and efficient solution for document management.

Objectives:

The specific objectives include:

- Setting up Azure Blob Storage for cloud file management.
- Organizing documents into structured folders such as joining letters, certificates, and reports.
- Ensuring secure access to files using signed URLs (Shared Access Signatures).
- Providing controlled and temporary access to documents without exposing storage credentials.

Project Structure:

azure-file-storage-system/

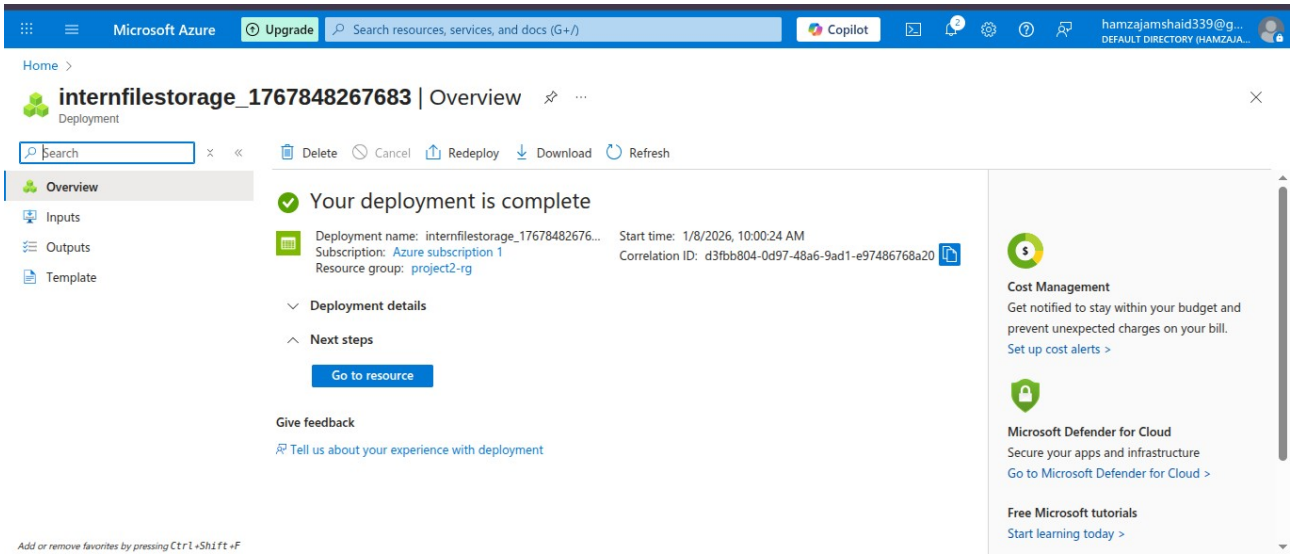
```
├── server.js
│
├── .env
│
├── controllers/
│   └── fileController.js
│
├── routes/
│   └── fileRoutes.js
│
├── package.json
│
└── README.md
```

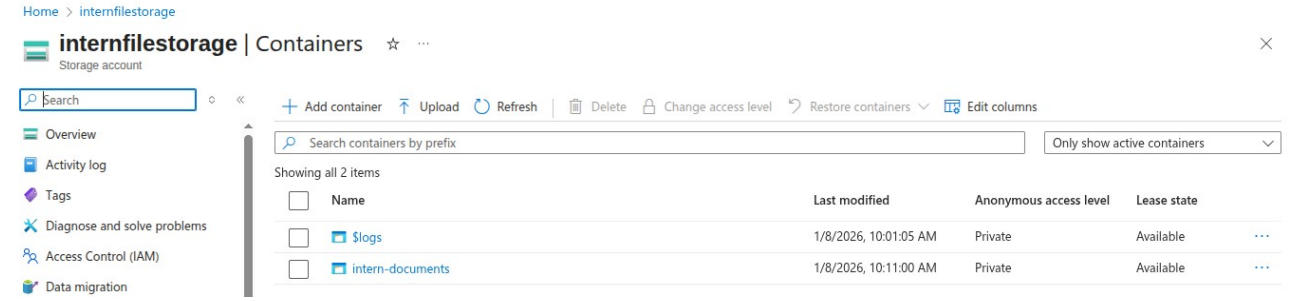
Project Procedure:

Below is a high-level breakdown of all procedures performed in the project. Each step mentions only actions and file names.

1. Create Storage account:

- Create a Storage account
- Create Container
- Get Storage Credentials



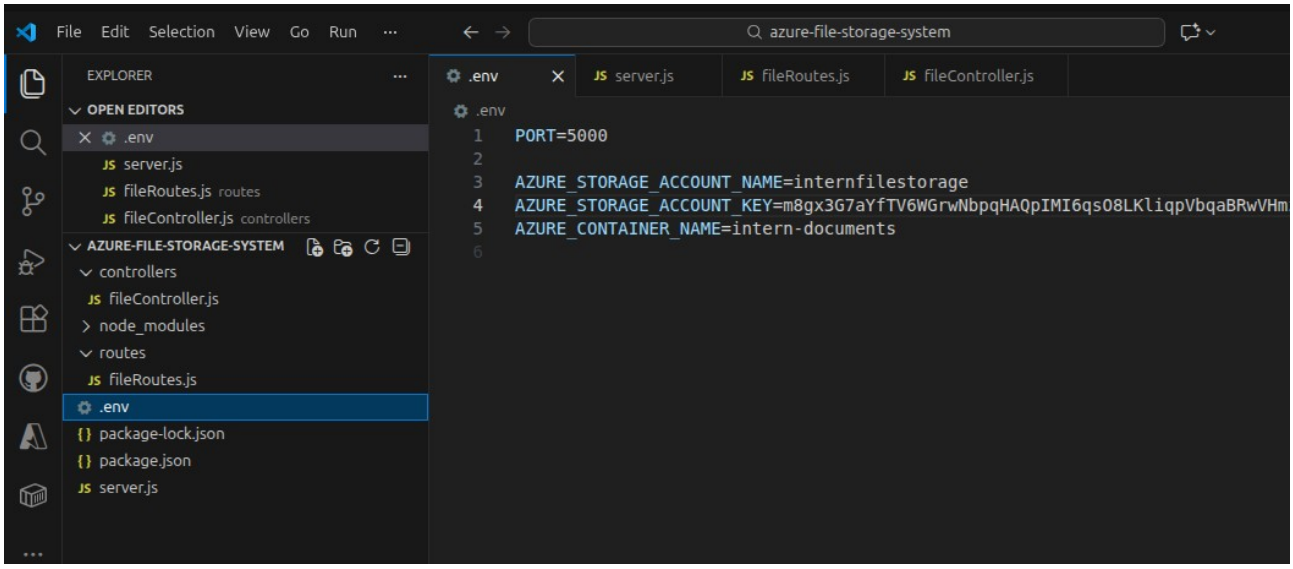


2. Backend Project Setup:

- Create Project folder
- Initialize Node.js Project

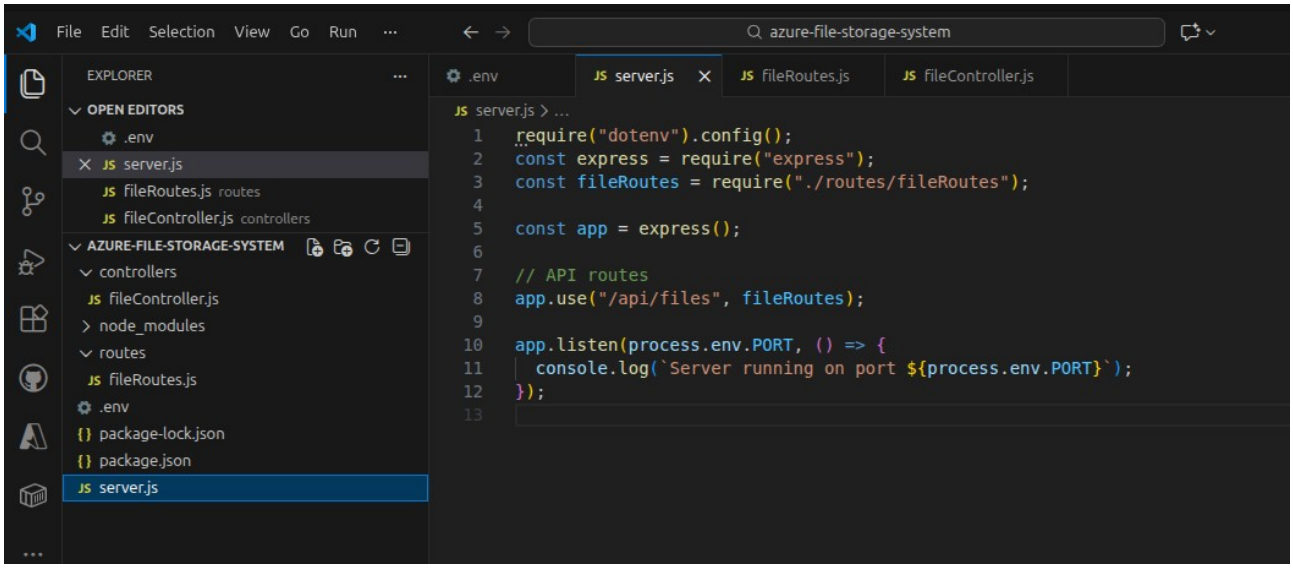
```
hamza@hamza-HP-EliteBook-840-G4:~$ mkdir azure-file-storage-system
hamza@hamza-HP-EliteBook-840-G4:~$ cd azure-file-storage-system
hamza@hamza-HP-EliteBook-840-G4:~/azure-file-storage-system$ npm init -y
Wrote to /home/hamza/azure-file-storage-system/package.json:

{
  "name": "azure-file-storage-system",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}



hamza@hamza-HP-EliteBook-840-G4:~/azure-file-storage-system$
```

3. Environment Configuration:

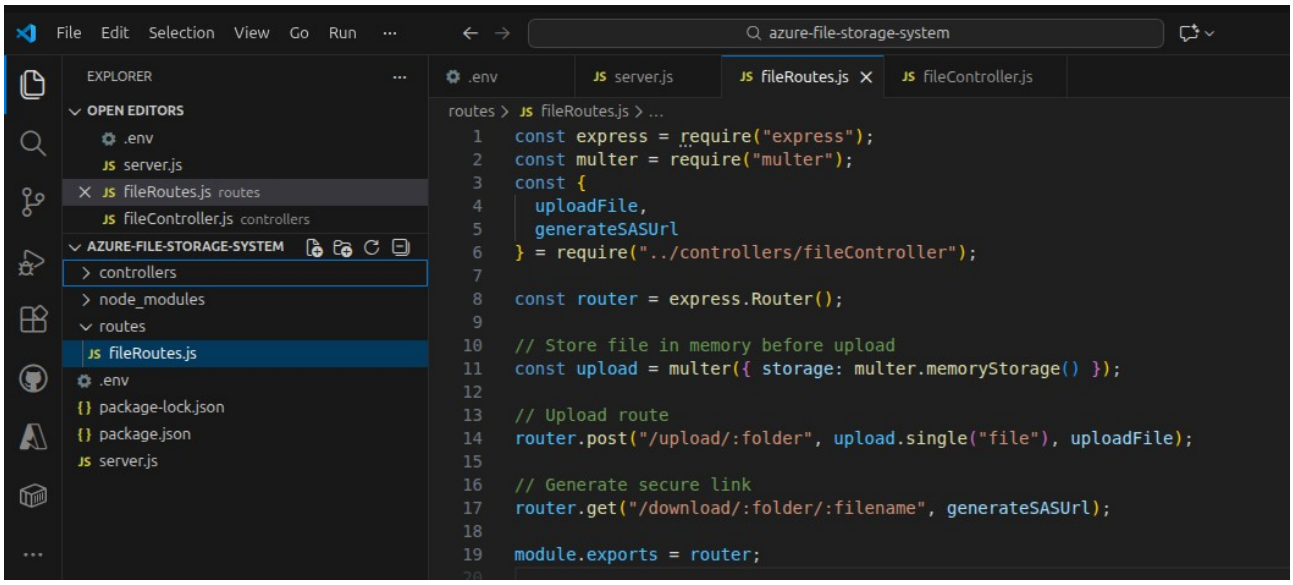


4.Backend Coding:

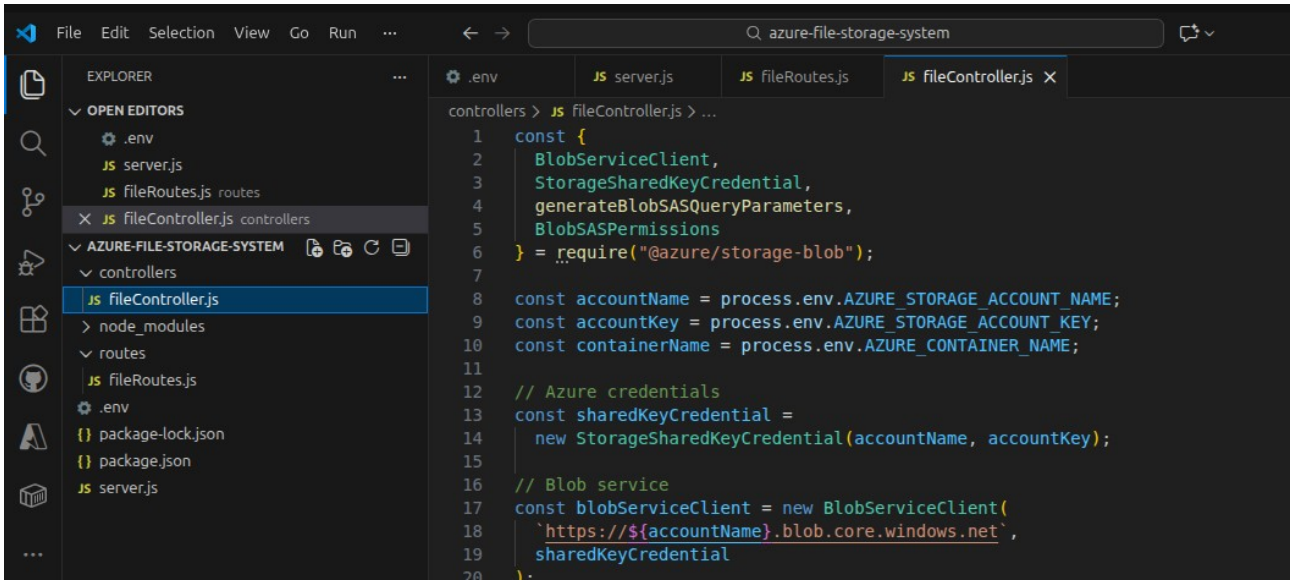
4.1 Server.js:-



4.2 routes/fileRoutes.js :-



4.3 controllers/fileController.js :-



5: Run & Test :

```
1745  node server.js
```

Result :-

The project was successfully implemented using Azure Blob Storage. Separate containers and folders were created for different types of intern documents, ensuring proper organization and easy retrieval. Files were uploaded securely to the cloud, and **signed URLs** were generated to allow temporary and permission-based access to specific documents. The system demonstrated reliable performance, secure data handling, and efficient file access from remote locations.

Conclusion :-

The Cloud-Based File Storage System effectively fulfills its objective of secure and organized document storage using Azure services. By leveraging Azure Blob Storage and signed URLs, the system ensures data security, scalability, and controlled access. This solution reduces the risks associated with local storage, improves accessibility, and provides a practical foundation for enterprise-level intern or document management systems. The project highlights the importance of cloud technologies in modern application development and can be extended further with role-based access control and automation features