

*Scalable Advanced Software Solutions*

# ***"Video Platform: Development and Deployment"***

**Alireza Foroughi**

**Student ID: 20001307**

**Ulster university's London Campus**



# *Discussion of the problem and identification of the issues related to scalability*

## Issues Encountered in Developing and Scaling the Platform

- **High Concurrent User Load:**

Handling several simultaneous video streaming requests, especially during periods of high demand, to provide buffer-free viewing.

- **Content Delivery Optimization:**

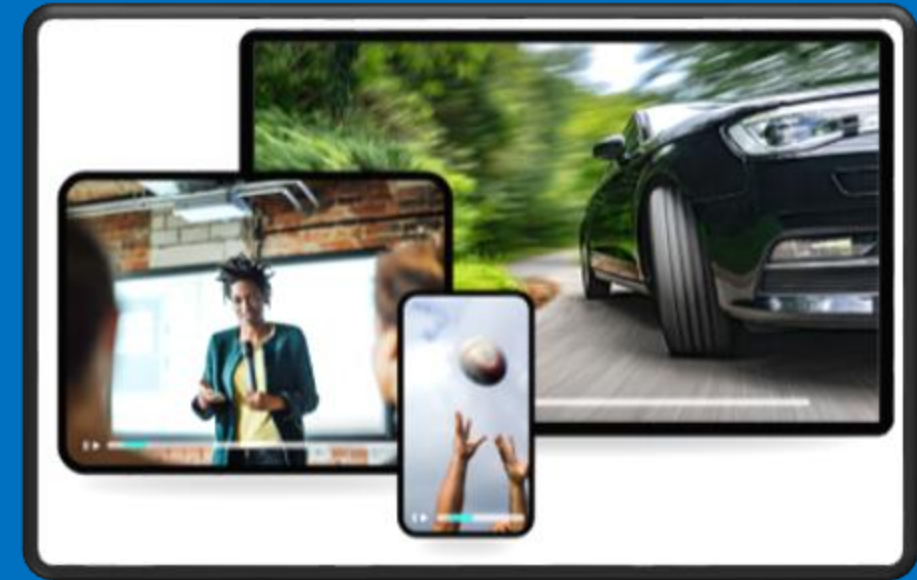
putting into practice effective strategies to guarantee fast video content distribution worldwide while lowering latency and preserving a flawless user experience.

- **Data Security and Compliance:**

making certain that user information, such as viewing preferences and login passwords, is safe and complies with data privacy regulations.

- **Video Storage Scalability:**

overcoming the difficulties associated with keeping a lot of video files accessible and economical.



- **Personalized Recommendations:**

Delivering tailored video suggestions based on user activity and preferences, enhancing user engagement and satisfaction.

- **Adaptability to Growth:**

Ensuring the platform can handle growing user bases, dynamic video collections, and increasing server demands without compromising performance.

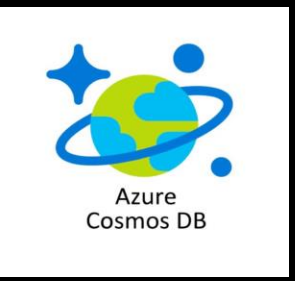
- **Balancing Cost and Performance:**

Strategically utilizing cloud services like Azure Blob Storage and Cosmos DB to achieve high performance while keeping operational costs manageable.



# Overview of the technical solution developed

## Database Selection



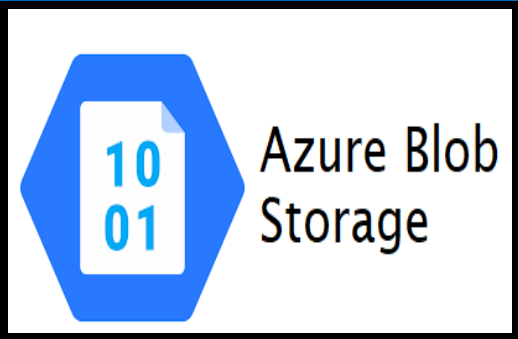
STEP 1

## Frontend Development



STEP 2

## Video Storage



STEP 3

## Backend Development



STEP 4

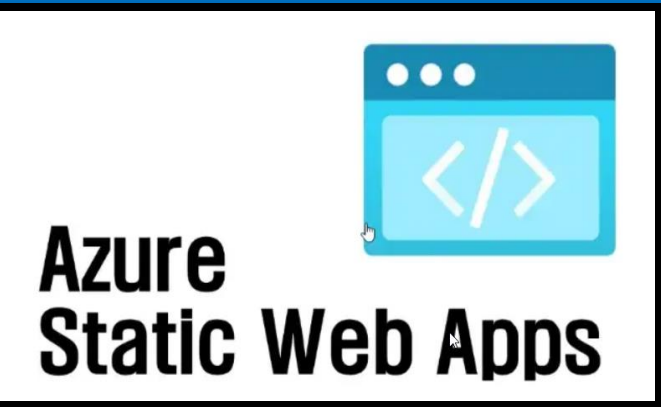


STEP 5



## Integration of Frontend and Backend

STEP 6



## Deployment

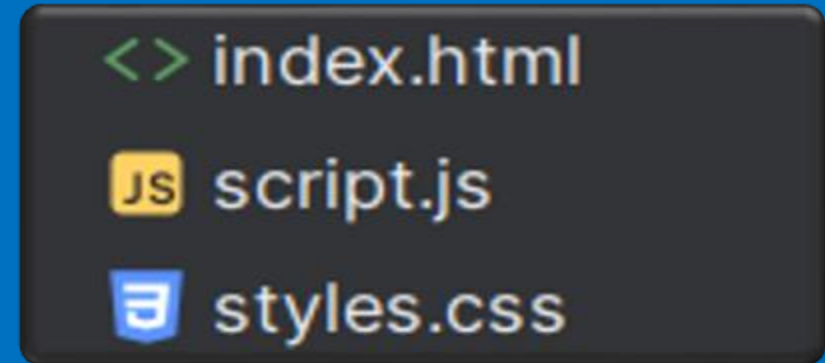
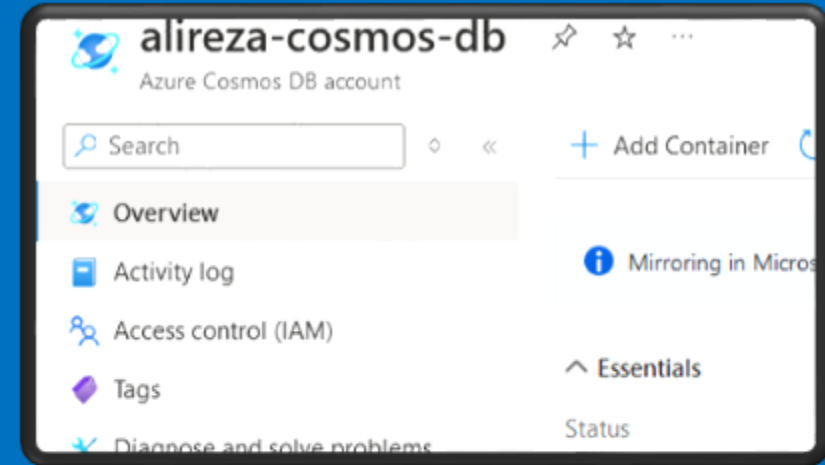
# Database Selection and Frontend Development

## Database Selection

- Choose **Azure Cosmos DB** due of its globally distributed architecture and excellent scalability.
- Used to hold **metadata** about users and videos, including **upload details, likes, and comments**.
- The **NoSQL database** structure facilitates JSON-like document storage and streamlines schema management.

## Frontend Development

- **HTML, CSS, and JavaScript** were used in its construction to provide a responsive, user-friendly interface.
- Upload capabilities and interactive buttons (**Like, Comment, Save, and Dislike**) were designed.
- Centered on clear navigation, responsive design, and accessibility.



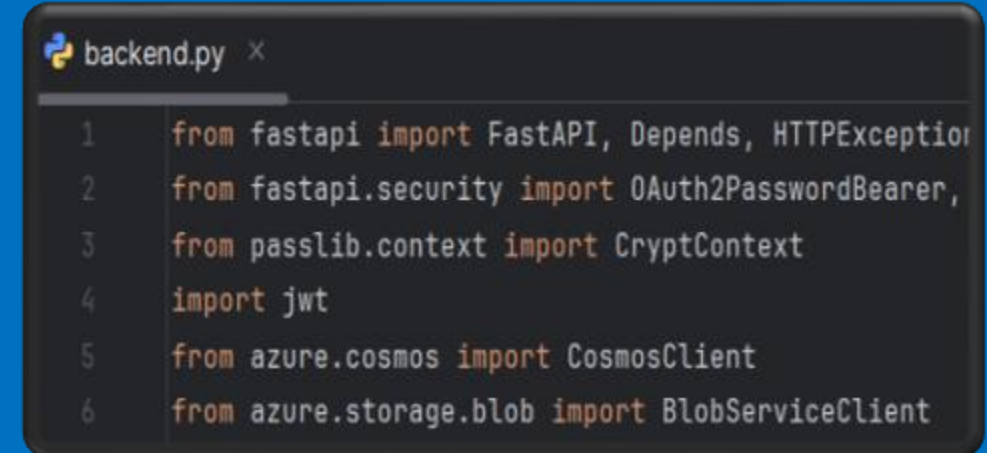
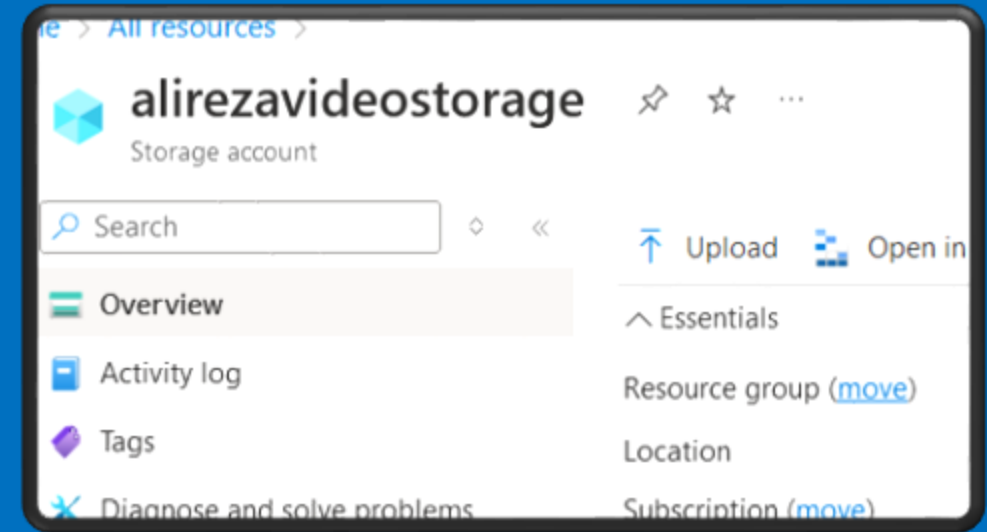
# Video Storage & Backend Development

## Video Storage

- Used **Azure Blob Storage** for storing video files.
- Offers high reliability, security, and low-latency access to uploaded videos.
- Efficient storage with flexible retrieval options ensures scalability.

## Backend Development

- Developed with **FastAPI**, which offers fast, lightweight, and robust RESTful APIs.
- Facilitates user authentication, video uploads, and interactions (e.g., likes/comments).
- APIs tested thoroughly for stability and security.





# Testing, Integration, and Deployment

- **Testing Backend API**

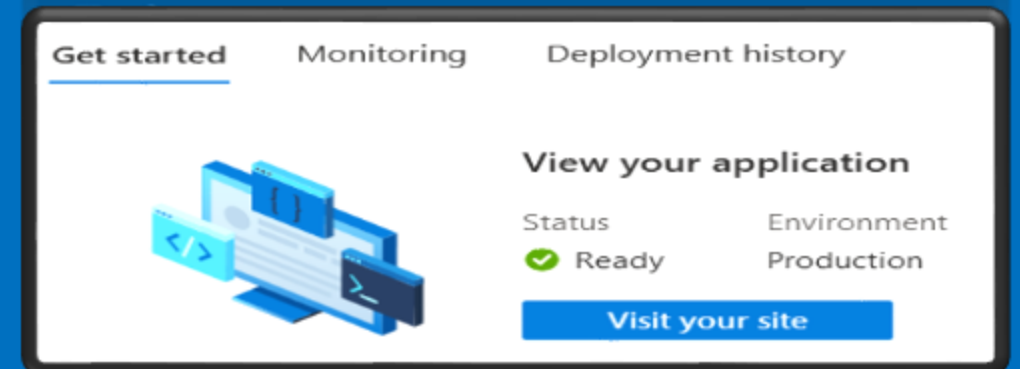
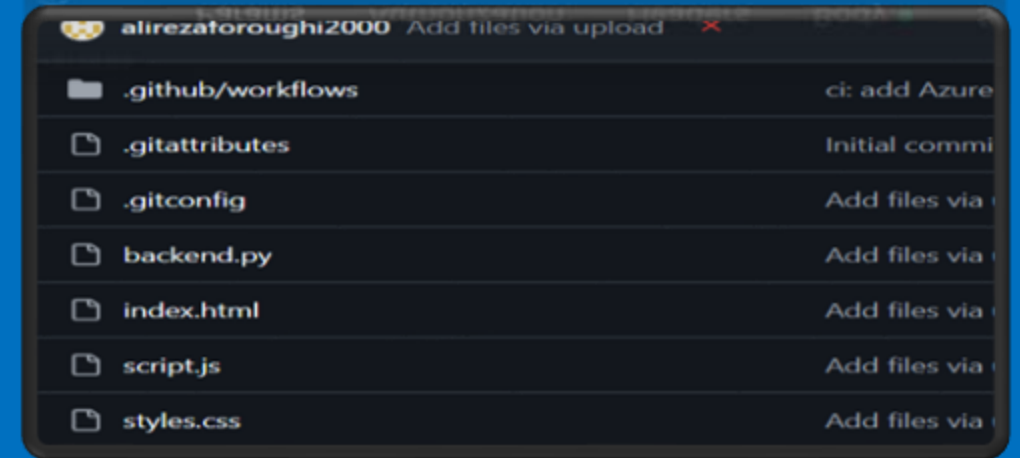
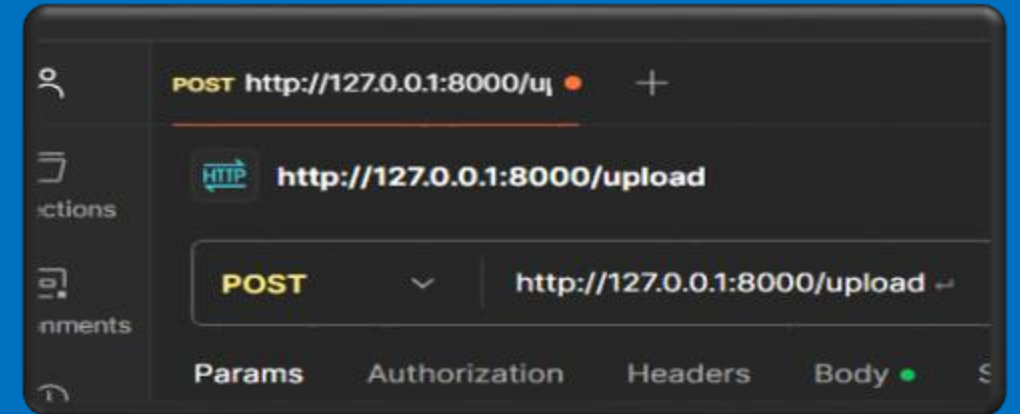
- Used **Postman** for API testing to verify endpoints and data flow between the backend and database.
- Ensured secure access for sensitive operations like user login and video uploads.

- **Integration of Frontend & Backend**

- Connected the frontend with backend APIs, enabling real-time interaction for uploading and streaming videos.
- **GitHub** used for version control and seamless collaboration during development.

- **Deployment on Azure**

- Deployed the project on **Azure Static Web Apps**, leveraging its tight integration with Azure services.
- Ensured global accessibility with scalable infrastructure.



# *An overview of advanced features within the developed solution*

- **Instantaneous Video Communication:**

- Like, comment, save, and dislike buttons allow users to engage with videos.
- allows for smooth backend updates for user engagement metrics.



- **Optimized Video Streaming:**

- Azure Blob Storage is used to store videos, guaranteeing low latency and excellent playback. Adaptive bitrate streaming to accommodate various network circumstances.

- **Scalable Architecture:**

- The system can effectively manage growing loads thanks to Azure Cosmos DB and Blob Storage. For an increasing number of users, high availability is guaranteed by the cloud-based solution.



- **User-Friendly Interface:**

- For device compatibility, responsive design ideas were used in its development. Interactive elements increase user retention and engagement.



- **Backend Optimization:**

- FastAPI guarantees effective asynchronous processing of API queries. Endpoints that facilitate quick user interactions and are optimized for performance.

- **Secure Data Transactions:**

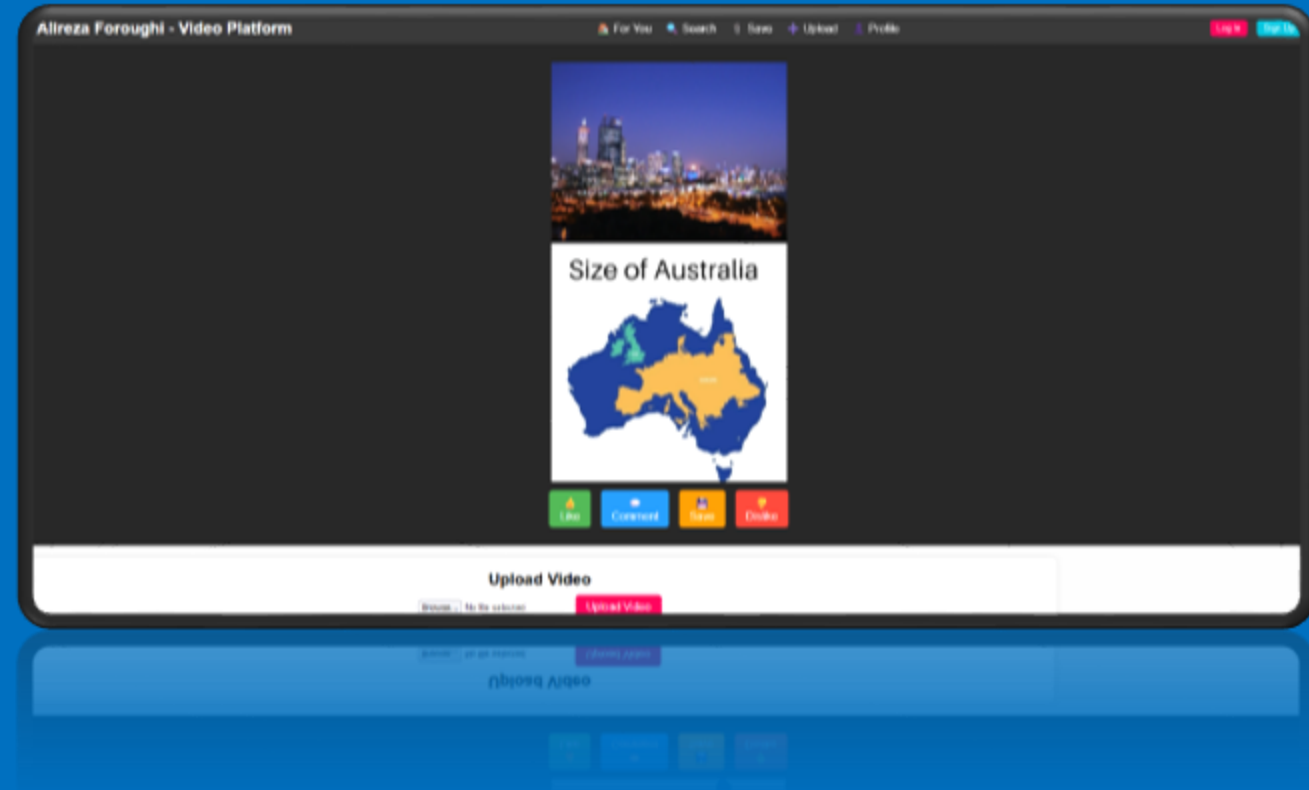
- Including secure authentication for the access and uploading of videos guarantees privacy and data integrity for every user transaction.

- **Automated Testing Workflow:**

- Postman was used to test the backend API for dependable and consistent performance. A bug-free user experience is ensured by early issue discovery.

- **Integration with Azure Static Web Apps:**

- smooth deployment integration that effectively combines frontend and backend. GitHub steps taken to reduce manual errors through automated deployment.



# *An assessment of limitations of the solution and evaluation of its ability to scale*

## Challenges and Constraints in the Developed Solution

- **High Initial Setup Cost:**
  - It costs a lot of money to deploy Azure services like Cosmos DB and Blob Storage. It might not be appropriate for modestly funded enterprises.
- **Learning Curve for FastAPI and Azure Integration:**
  - More learning time was required for the FastAPI interaction with Azure services. During the frontend and API integration process, there were necessary debugging challenges.
- **Video Upload Limitations:**
  - When uploading huge videos, Blob Storage's performance is reliant on network bandwidth. Users who submit high-resolution videos may encounter difficulties due to file size restrictions.
- **Real-time Updates for User Interactions:**
  - Although like/dislike and commenting tools are in place, real-time data syncing has to be improved in order to scale with high user traffic.

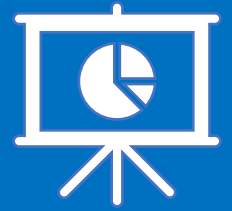


# Solution's Capability to Scale

- **Azure's Scalable Infrastructure:**
  - Cosmos DB and Blob Storage ensure high scalability for data and media storage.
  - Supports global content delivery without significant delays.
- **Backend Resilience with FastAPI:**
  - The asynchronous nature of FastAPI allows handling multiple requests efficiently.
  - Suitable for managing sudden spikes in user activity.
- **Future Scalability Improvements:**
  - Implementing load balancers and auto-scaling groups for better performance.
  - Using Content Delivery Networks (CDNs) to optimize media delivery across regions.
- **Challenges in Scaling:**
  - Increased cost with scaling operations, especially when handling millions of users.
  - Requires constant monitoring and optimization to ensure performance at scale.



## Concluding comments



This project effectively used cutting-edge technologies including **Microsoft Azure**, **FastAPI**, and **Cosmos DB** to create and implement a **scalable video platform**. The platform tackles important issues with scalability, performance, and user engagement by combining reliable backend technologies, cloud storage, and a responsive frontend. In order to deliver a flawless user experience, the solution exhibits sophisticated features like effective video uploads, real-time interactivity, and streamlined API testing.

Technical know-how in database administration, cloud computing, and API design was used throughout the development process to guarantee a strong basis for upcoming improvements. This project demonstrates how the platform could be expanded to include cross-platform interoperability, real-time analytics, and tailored recommendations.

# References

- [1] Microsoft Azure Documentation, "Azure Cosmos DB Overview." [Online]. Available: <https://learn.microsoft.com/en-us/azure/cosmos-db/introduction>
- [2] Microsoft Azure Documentation, "Azure Blob Storage: Scalable and Secure Cloud Storage." [Online]. Available: <https://learn.microsoft.com/en-us/azure/storage/blobs/>
- [3] Microsoft Azure Documentation, "Azure Static Web Apps Overview." [Online]. Available: <https://learn.microsoft.com/en-us/azure/static-web-apps/>
- [4] FastAPI Documentation, "FastAPI Framework for APIs with Python." [Online]. Available: <https://fastapi.tiangolo.com/>
- [5] GitHub Documentation, "Using GitHub for Code Collaboration and CI/CD." [Online]. Available: <https://docs.github.com/en>
- [6] Postman Documentation, "Postman API Testing Guide." [Online]. Available: <https://www.postman.com/api-documentation/>
- [7] Mozilla Developer Network (MDN), "HTML Basics: Structure of Web Pages." [Online]. Available: [https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/HTML\\_basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics)
- [8] Mozilla Developer Network (MDN), "CSS Basics: Styling Web Pages." [Online]. Available: [https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/CSS\\_basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics)
- [9] Mozilla Developer Network (MDN), "JavaScript: Dynamic Functionality in Web Pages." [Online]. Available: [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps)
- [10] Netflix Technology Blog, "Scalability Challenges and Best Practices in Video Streaming Platforms." [Online]. Available: <https://netflixtechblog.com/>
- [11] Stack Overflow, "Using Azure Services for Scalable Web Development." [Online]. Available: <https://stackoverflow.com/>
- [12] W3Schools, "JavaScript Event Handling." [Online]. Available: [https://www.w3schools.com/js/js\\_events.asp](https://www.w3schools.com/js/js_events.asp)
- [13] DigitalOcean Blog, "Integrating FastAPI with Cloud Storage Services." [Online]. Available: <https://www.digitalocean.com/community>
- [14] Coursera, "Cloud Application Development with Microsoft Azure." [Online]. Available: <https://www.coursera.org/>
- [15] Udemy, "Full-Stack Web Development Bootcamp." [Online]. Available: <https://www.udemy.com/>
- [16] IEEE Xplore Digital Library, "Scalability and Performance Analysis in Cloud-Based Applications." [Online]. Available: <https://ieeexplore.ieee.org/>
- [17] ResearchGate, "Advanced Techniques in Cloud Deployment and API Testing." [Online]. Available: <https://www.researchgate.net/>
- [18] Azure Storage Performance Whitepaper, "Optimizing Azure Blob Storage for Large-Scale Applications." [Online]. Available: <https://learn.microsoft.com/en-us/azure/storage/blobs/storage-performance-checklist>
- [19] Microsoft Learn, "Azure DevOps for Continuous Integration." [Online]. Available: <https://learn.microsoft.com/en-us/devops/>
- [20] Codecademy, "Frontend Development: HTML, CSS, and JavaScript Foundations." [Online]. Available: <https://www.codecademy.com/>
- [21] YouTube, "Tutorial: Building a Scalable Web Application on Azure." [Online]. Available: <https://www.youtube.com/>
- [22] Mendeley Ltd, "Reference Management and Citation for Academic Projects." [Online]. Available: <https://www.mendeley.com/>