Neal Anders
CTO Interview
Recorded Responses
2025-11-03

#1

I notice you've recently transitioned from Deputy Chief Engineer at Innovative Defense Technologies to starting your own consulting firm, CyberSavant. What motivated this entrepreneurial move, and how do you see your previous experience in leading software development teams translating to our need for migrating from a .NET/Excel platform to a cloud-first solution?

#2

Your resume shows extensive experience across different technology leadership roles. Looking at our need to migrate from a .NET/Excel platform to a cloud-first solution, could you describe a specific migration project you've led that involved similar technical challenges, and what key lessons you learned that would apply to our situation?

#3

Could you walk me through your professional journey and how it has prepared you for this CTO/Lead Software Architect role?

#4

What aspects of transitioning a product from desktop-based to cloud-native SaaS excite you the most?

#5

How do you balance strategic technical leadership with hands-on development work?

#6

Describe a situation where you had to pivot a technical strategy based on new information or changing business requirements. How did you handle it?

#7

How would you approach migrating a .NET/Excel-based platform to a cloud-first solution using Azure and Python microservices while ensuring minimal disruption to existing users?

#8

What strategies would you implement for integrating AI document processing capabilities with Microsoft Office tools like Excel, Word, and PowerPoint?

#9

Describe your experience with scaling distributed systems and how you would apply those principles to this platform as it grows.

#10

Your resume mentions experience with AI Foundations and Machine Learning. Given our platform's focus on AI automation for document processing, could you share specific examples of how you've implemented AI/ML technologies in previous roles, particularly any work related to document scanning or data extraction?

Code Example

Looking at this code snippet for an Azure Function that processes document data, what improvements would you suggest to make it more scalable and maintainable?

```python
import azure.functions as func
import json
import pandas as pd
from typing import List

def main(req: func.HttpRequest) -> func.HttpResponse:
    try:
        req_body = req.get_json()
        document_data = req_body.get('document_data')

        # Process document data
        df = pd.DataFrame(document_data)

        # Perform calculations
        results = []
        for _, row in df.iterrows():
            # Some complex business logic
            processed_value = row['value'] * 1.5 if row['type'] == 'A' else row['value'] * 0.8
            results.append({
                'id': row['id'],
                'processed_value': processed_value
            })

        # Return results
        return func.HttpResponse(
            json.dumps({"results": results}),
            mimetype="application/json"
        )
    except Exception as e:
        return func.HttpResponse(
            json.dumps({"error": str(e)}),
            status_code=500
        )
```

Thoughts:

Initial Recommendations:

- Avoid locking via async
- Handle input/payload errors
- Ensure all responses have status codes
- Make processed values configurable vs hard-coded
- Row iteration vs lambda call - PD Vectors
- Add Logging/Metrics

Thinking Bigger:

- Request Data Input Limitations
- Separate Business Logic from Request Processing
- Larger Documents & Memory Limits for AZ Funcs
- Split Request/Response, Parsing, Processing, Etc
- Queues / Kafka / Flink / Etc
- Cloud-provider Agnostic

* These are all quick & easy fixes where AI tools can help.

Thank You

Notes:

Desktop:

- Video Window Disappears
- Choppy / Laggy Response
- Audio Volume Too Low
- Not Capturing at all?

Mobile:

- Better video quality
- Couldn’t see questions