



Debrief: Mostly AI Challenge 48h

Thierry Baudez
AI Content Manager applicant

MOSTLY•AI

Content

1. Before Starting
2. Jupyter Notebook
3. Explainer Video
4. Documentation Snippet
5. Hiccups
6. Take-Aways

MOSTLY•AI

1. Before Starting $T-12h \rightarrow T+1h$

T-12h

Received emails from
Elisa

Started thinking of what
I would need:
templates, structures,
visual cues

Generated true-and-
tested content structure

Watched a few YT
videos by Mostly AI
(already 112 videos!
Congrats!)

- Noticing:
- * Approachable
 - * In person
 - * Tech-centric
- Opportunities:
- * More calls to action
 - * Highlight benefits more

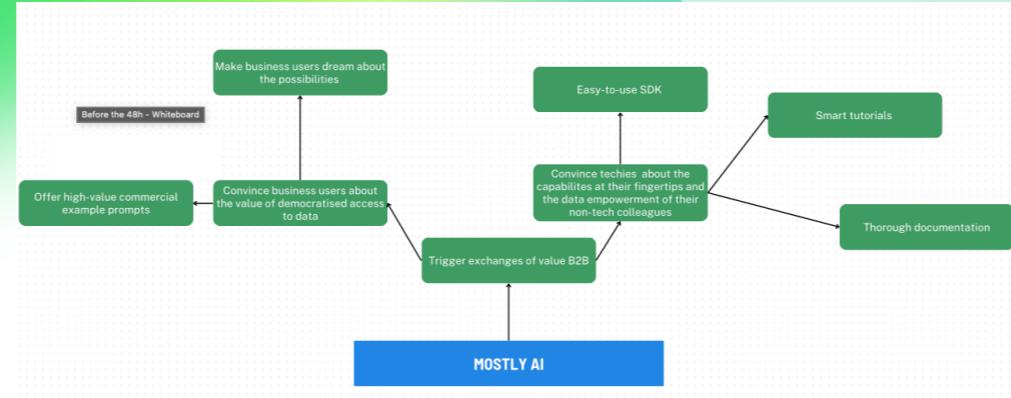
Structures, templates, checks... Things I can still do
before the Mostly AI challenge really starts.

T-0h: Briefing is in!

1. Meet the Players
2. Content Creation Logic
3. ChatGPT
4. Data Set

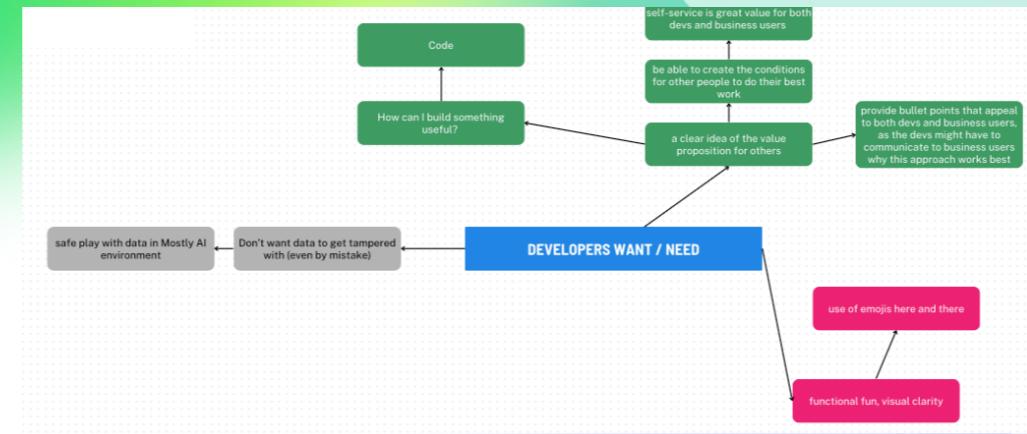
“Create a content asset
that showcases the value
of the AI Assistant + *SQL*
capabilities on multi-table data”

1. Meet the Players: Mostly AI



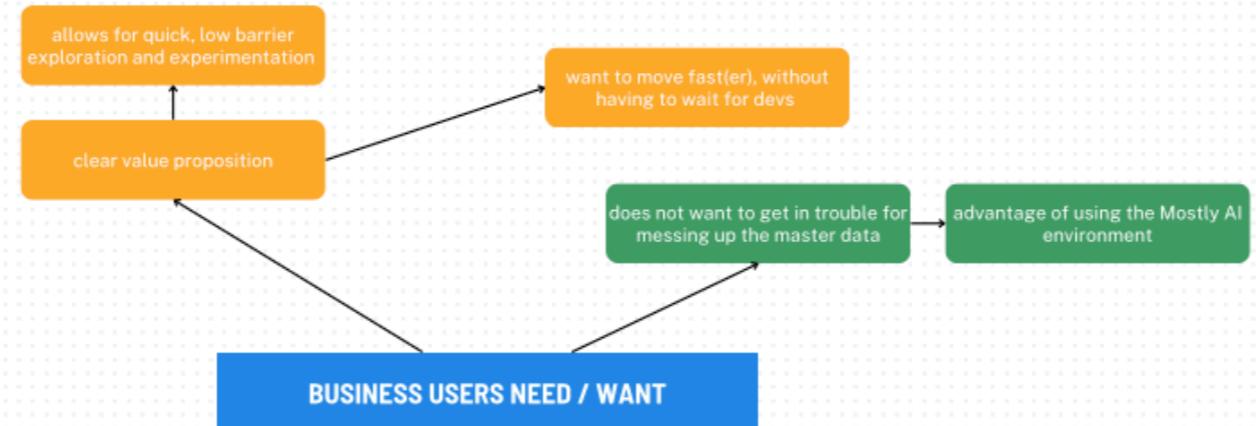
Core question guiding all comms:
How does Mostly AI obtain value from clients?
By facilitating democratised access to data.

1. Meet the Players: DEVs



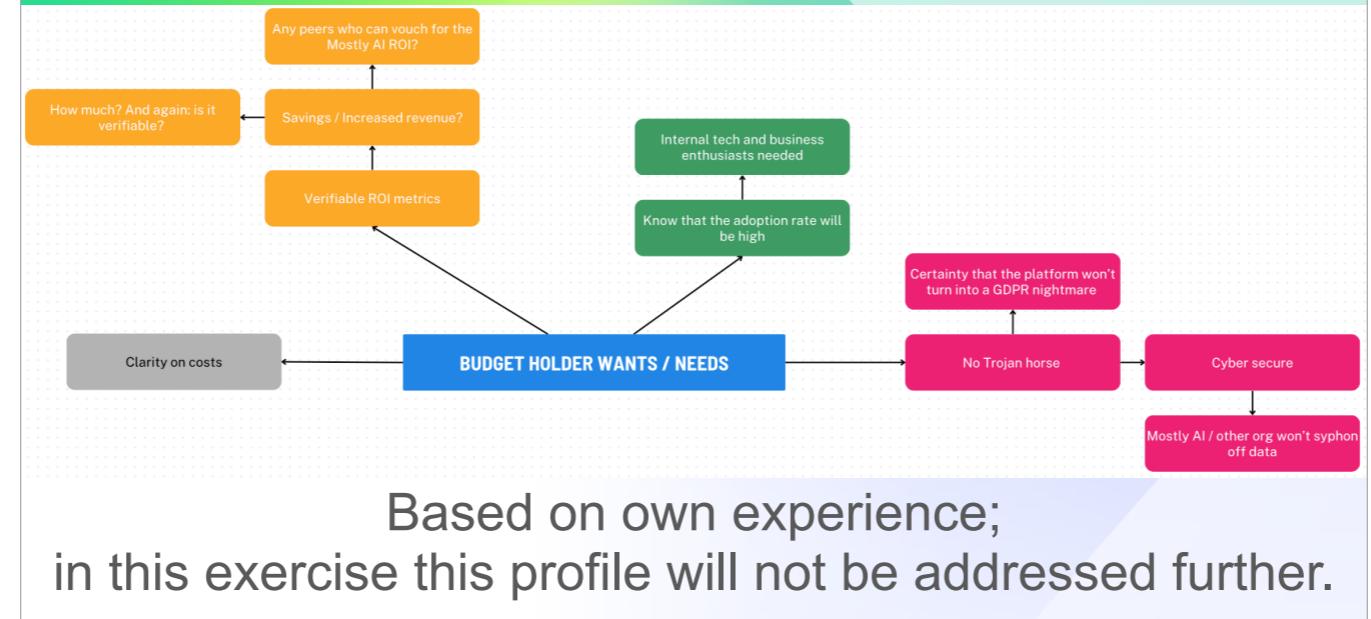
Based on briefing, own experience
+ insight: techies don't operate in a vacuum;
-> add selling points towards business colleagues.

1. Meet the Players: Business Users



Based on briefing + own experience;
in line with regular Mostly AI marketing copy.

1. Meet the Players: Budget Holder

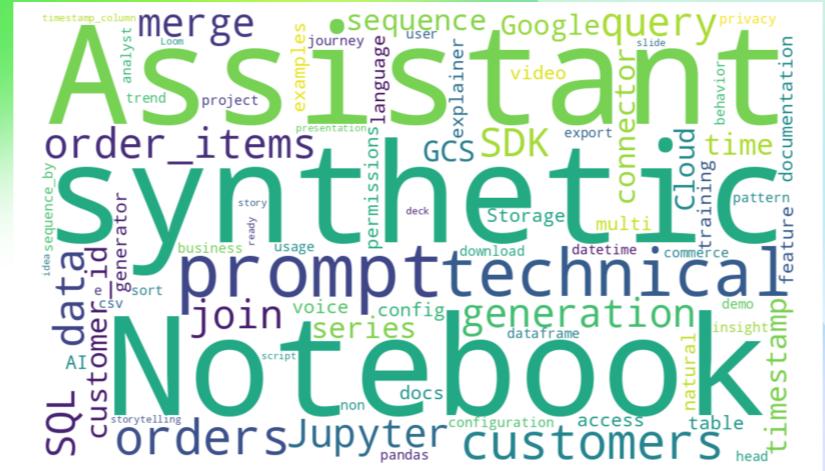


2. Content Creation Logic

1. understand what's happening technically, thus creating materials for the devs
2. Play around with the Assistant, then prepare script for the video
3. Distill materials to be added to the documentation

For this 1st time: Tech, Business, Sell
In reality: Business, Tech, Business, Sell

3. ChatGPT



Clarifying my partnership with LLMs: I lay out the content, direction and early thoughts. GPT answers. I refine, push back. We discuss my approach and focus.

4. Data Set

kaggle

Search

DILLON MYRICK · UPDATED 2 YEARS AGO

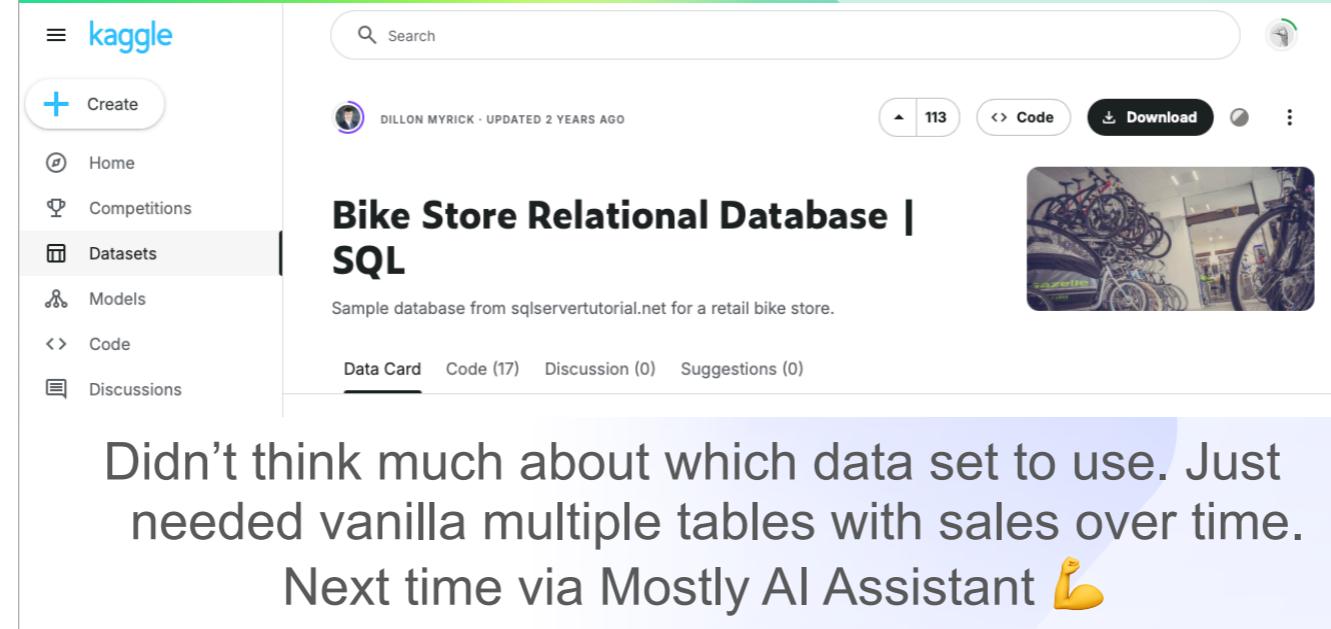
113 Code Download :

Bike Store Relational Database | SQL

Sample database from sqlservertutorial.net for a retail bike store.

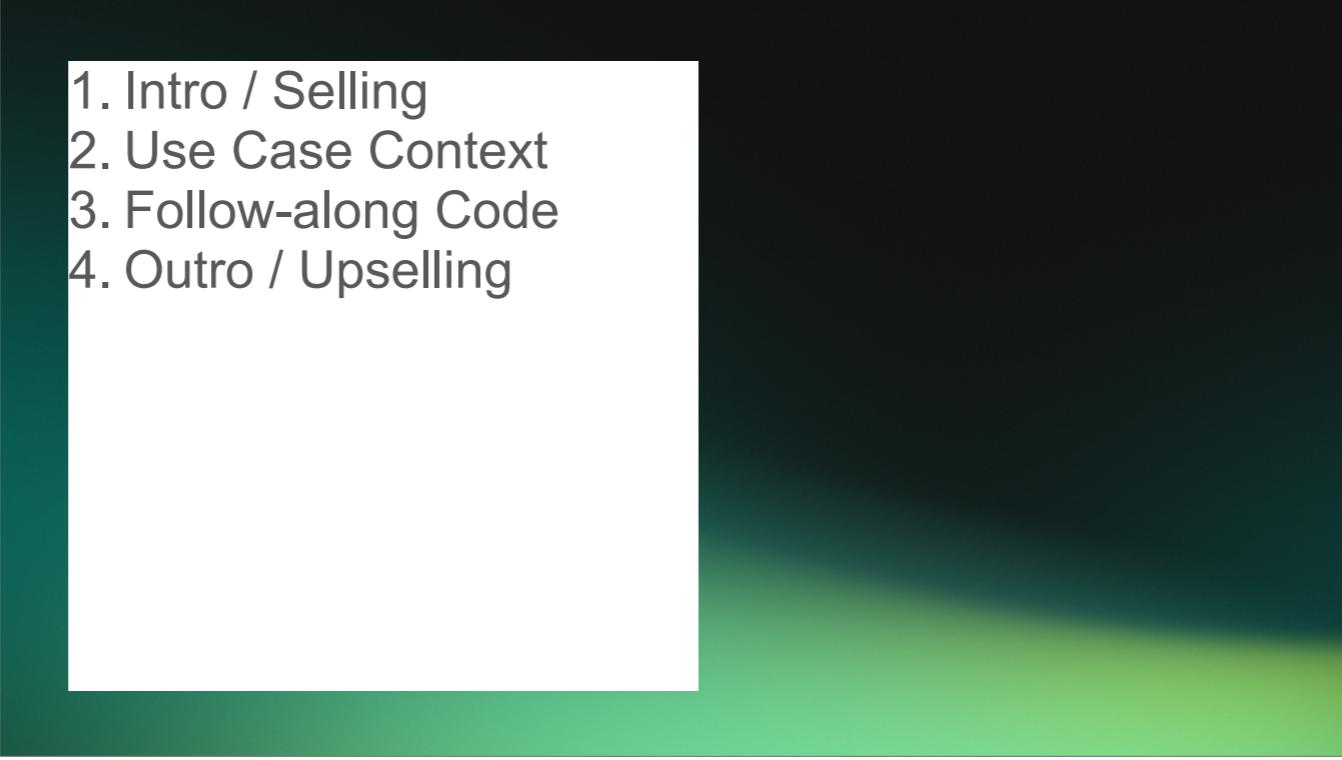
Data Card Code (17) Discussion (0) Suggestions (0)

Didn't think much about which data set to use. Just needed vanilla multiple tables with sales over time.
Next time via Mostly AI Assistant 💪



MOSTLY•AI

2. Jupyter Notebook $T+1 \rightarrow T+5$

- 
1. Intro / Selling
 2. Use Case Context
 3. Follow-along Code
 4. Outro / Upselling

1. Intro / Selling

The screenshot shows a slide with a dark background and white text. At the top, it says 'Querying & Preparing Time-Series Data with MOSTLY AI SDK'. Below that is a section titled 'Using AI Assistant to Query Connected Data Sources'. The main content is a 'Welcome to this tutorial!' message. It includes a question '💡 What if you could enable non-technical users to explore real data safely using natural language queries?'. A list of goals follows: 'After this tutorial you'll be able to:' which includes 'Select subsets of data before training a generator', 'Pre-aggregate or transform data before training', 'Connect your tables of choice to the Mostly AI Assistant', and 'Empower non-technical users to explore real data safely'. Another question '💡 Which connectors work for this approach? MOSTLY AI supports running custom SQL queries on:' is followed by a list: 'Cloud storage: AWS S3, Azure Blob, GCP buckets', 'Data warehouses: BigQuery, Snowflake, Databricks', and 'Databases: MySQL, PostgreSQL, Oracle, MariaDB, Microsoft SQL Server, Apache Hive'. At the bottom is a 'Let's go!' button.

Based on DEV mind map: offer clear value proposition for themselves & for business users, highlighting compatibility, safety and empowerment

2. Use Case Context

Use Case Context

This dataset represents a **typical customer lifecycle**: a customer places one or more orders, each containing one or more items. It's ideal for demonstrating **multi-sequence, time-series data generation** with MOSTLY AI.

Key columns:

- `customer_id` for identifying sequences
- `order_date` for time alignment

This setup simulates a customer behavior dataset that can be explored, queried, and synthesized.

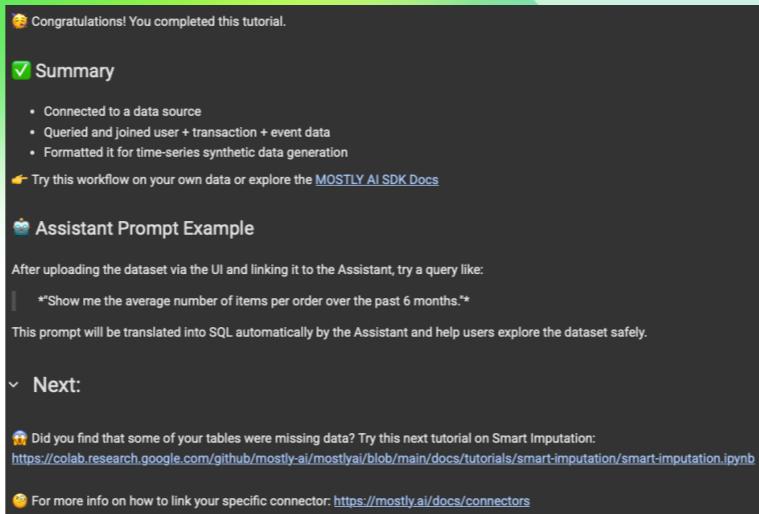
The data set details are not important to the tutorial.
-> Minimal use case context, getting to the code ASAP.

3. Follow-along Code

```
[ ] # Install and import the MOSTLY AI SDK  
!pip install mostlyai  
  
from mostlyai.sdk import MostlyAI  
  
[ ] # Authenticate with your API key (replace with actual key)  
# --- In case you don't have an API key yet: https://app.mostly.ai/settings/api-keys  
mostly = MostlyAI(  
    api_key='your_api_key',  
    base_url='https://app.mostly.ai'  
)
```

Regular tutorial notebook, with anticipatory helpful notes in the description(e.g. API key).

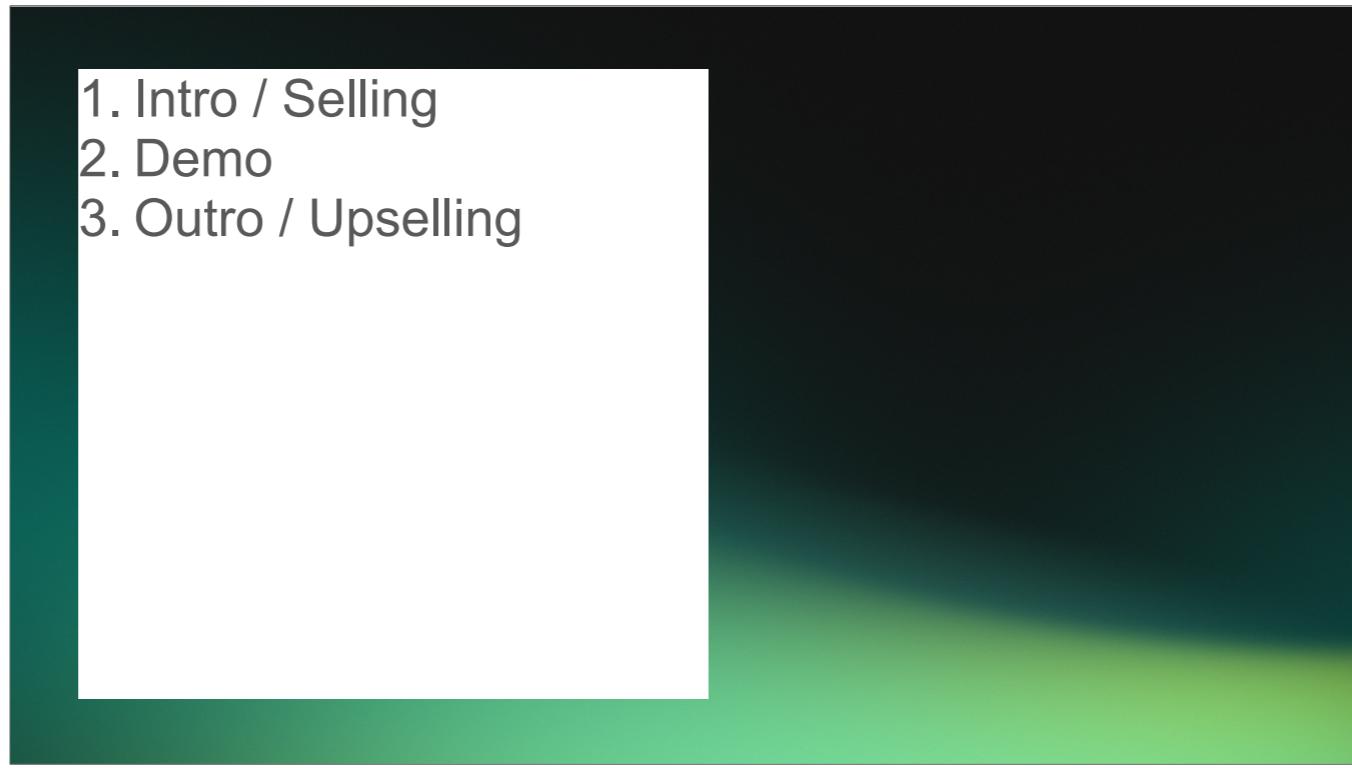
4. Outro / Upselling



Positive reinforcements with congrats, recap, real-life action, upselling to related tutorials and documentation.

MOSTLY•AI

3. Video Explainer $T+5 \rightarrow T+26$

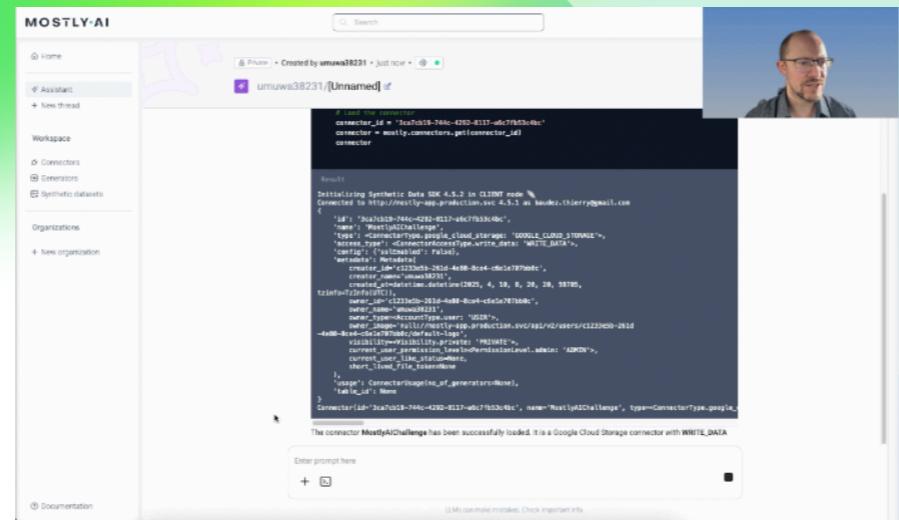
- 
1. Intro / Selling
 2. Demo
 3. Outro / Upselling

1. Intro / Selling (00:00 - 00:25)



Get straight into business users' pain and offer relief
within first 20 sec for fast value proposition.

2. Demo (00:26 - 01:15)



For the sake of speed and flow, the demo gets a post-fact voice-over. Speaker lowkey highlights Assistant benefits.

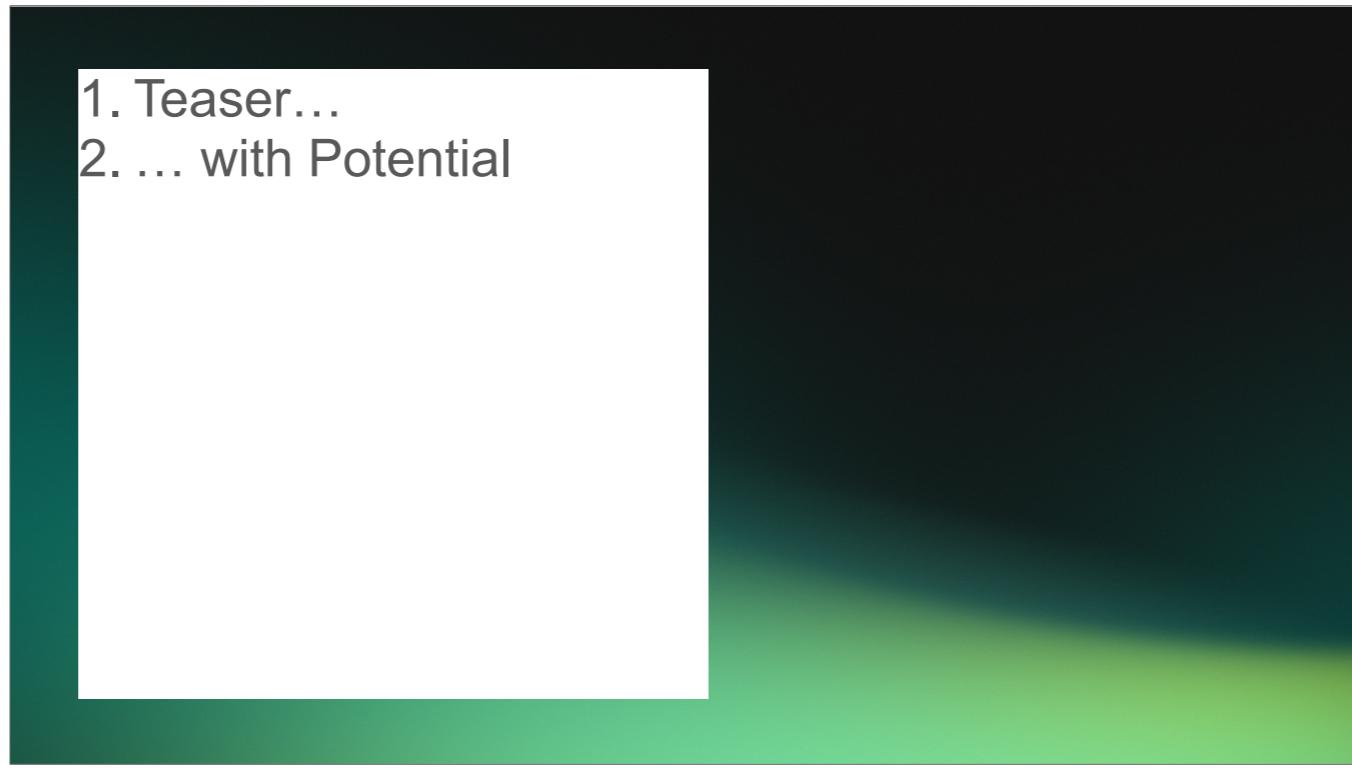
3. Outro / Upselling (01:16 - 01:29)



Recap the plus points of the approach, plus upsell to personal demo for more.

MOSTLY•AI

4. Documentation Snippet $T+26 \rightarrow T+28$

- 
1. Teaser...
2. with Potential

1. Teaser...

Explore complex datasets without writing SQL

Say goodbye to waiting on data teams to explore datasets spread across multiple tables. Business users can now independently uncover valuable trends, ask follow-up questions, and make data-driven decisions - faster and without writing a single line of SQL.

By simply connecting the Assistant to the tables of your choice, you can explore layered, time-based relationships instantly using natural language. The Assistant understands how your tables relate - it joins customer and order data, filters for repeat purchases, aggregates revenue, and surfaces insights automatically.

Example prompts:

- "Which product categories generated the most revenue from returning customers last quarter?"
- "Who were our top clients last year?"

Standard teaser targeting business users, with commercially triggering questions as example prompts to showcase immediate value.

2. ... with Potential

Example prompts:

- "Which product categories generated the most revenue from returning customers last quarter?"
- "Who were our top clients last year?"

For more on this:  [Easy, Safe Multi-Table Analysis](#) |  [Multi-Table Time Series](#)

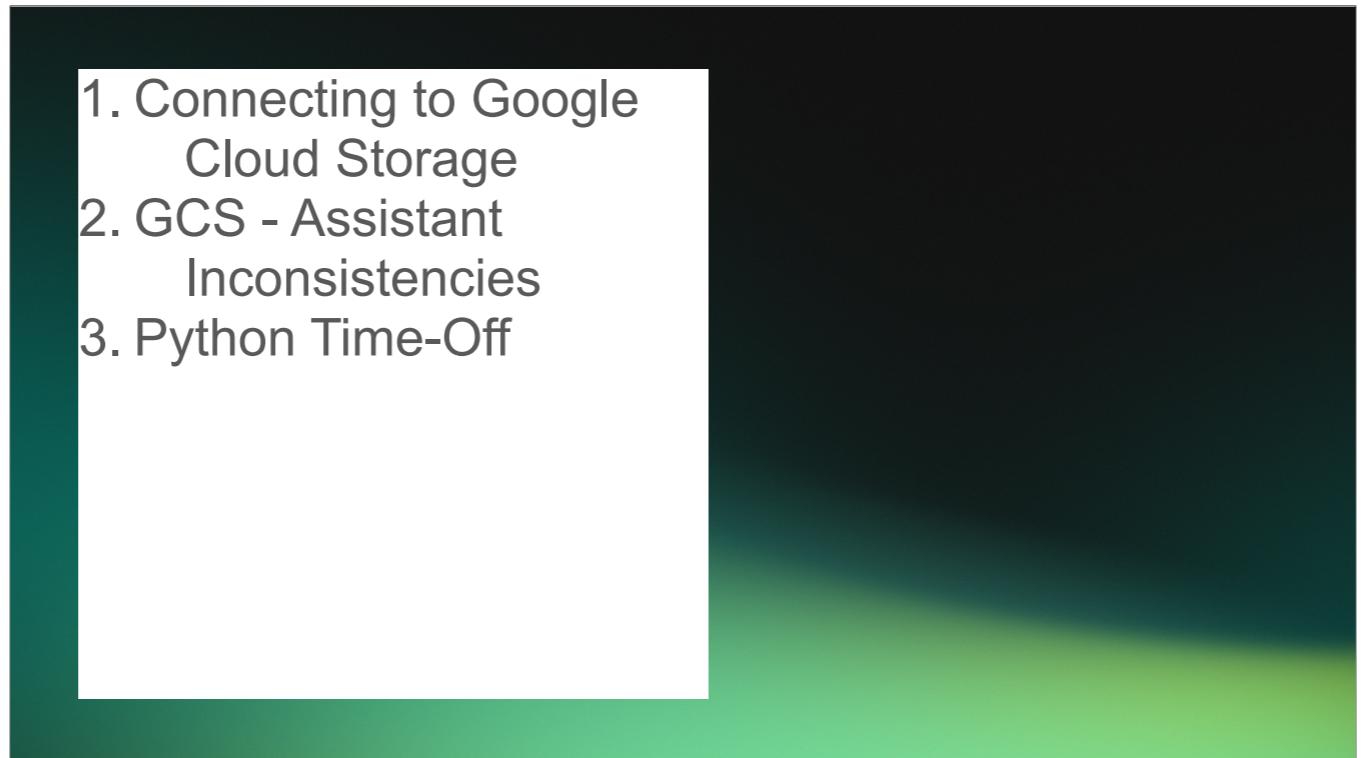
Upselling idea: we could elegantly link other resources for business users to consult or forward (YT, Jupyter NB).

MOSTLY•AI

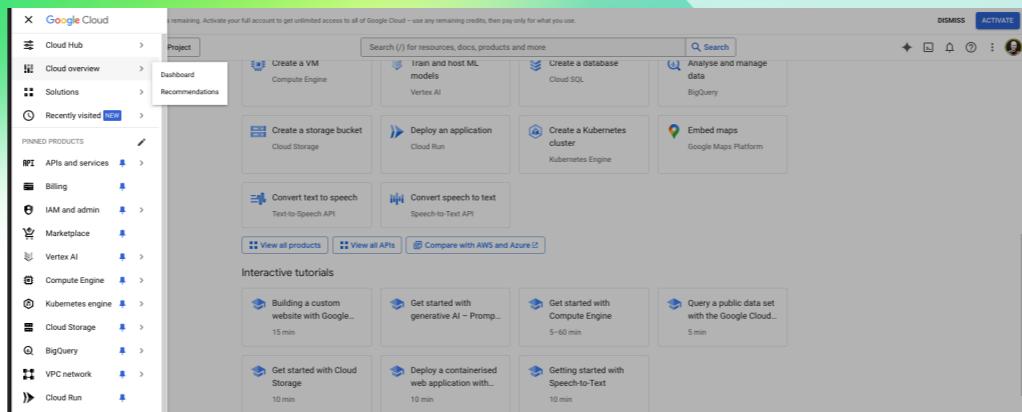
5. Hiccups

T+1 -> T+3,5

T+30 -> T+38

- 
1. Connecting to Google Cloud Storage
 2. GCS - Assistant Inconsistencies
 3. Python Time-Off

1. Connecting to Google Cloud Storage T+1h -> T+3,5h



First time using a connector taught me a lot about connecting in theory vs IRL. Stumped for a few hours on day 1. Troubleshooting with LLMs didn't help.

1. Connecting to Google Cloud Storage T+1h -> T+3,5h

```
HTTP 400: Invalid JSON structure for field type | b53c4bc [WARNNING:modtuya.sdk.client.base.Ignoring unrecognized fields for GeneratorConfig: connector_id, target_table] created generator file@/3-199-444d-bd69-770c2/216 Started generator training

APIStatusError Traceback (most recent call last)
cipython-input-15-2514592ebe39-in cell line: 0>
  3 # Skip passing 'location' if it's already part of the connector setup
  4
  5 --> 6 connector = mostly.connect()
  6     config=(
  7         'name': 'archive',
  8
  9 frames
[WARNNING:modtuya.sdk.client.base.Ignoring unrecognized fields for GeneratorConfig: connector_id, target_table] created generator file@/3-199-444d-bd69-770c2/216 Started generator training

HTTP 400: Cannot start training for a generator without tables | b19-744c-4 [WARNNING:modtuya.sdk.client.base.Ignoring unrecognized fields for GeneratorConfig: connector_id, target_table] created generator file@/3-199-444d-bd69-770c2/216 Started generator training
```

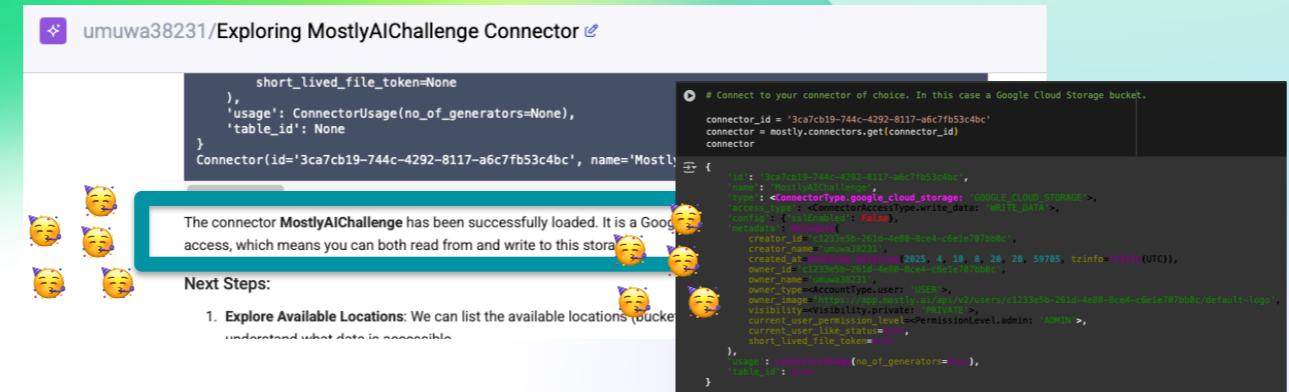
It's still not working. Screw it. I'm losing too much time. Let's just use the files from my local system:

TypeError:	→ 128 connector = self.request[129 verb+POST, 130 path][0] AssertionError: [Errno 3] File exists: '/tmp/mostly/api/client/base.py' by request[self, path, verb, 131 response_type, new_response, is_api_call, do_lame_case, case, 132 do_response, dcl_case, do_include, include, extra_key_values, **kwargs] 133 error_msg = exc.response_content 134 raise HTTPError(error_code=HTTPStatus.error_code) 135 raise HTTPStatusError(error_code=HTTPStatus.error_code, error_msg) from None
AttributeError:	12 13 # Step 3: Start the training using the training data
AttributeError: '_MostlyConnectorsClient'	[Errno 3] File exists: '/tmp/mostly/api/client/generators.py' 3034 *** 3035 rich.print("Started generator training") 3036 self.generator_client_training_start(self.generator_id) 3037 3038 def cancel_training(): >>> None [Errno 3] File exists: '/tmp/mostly/api/client/generators.py' 3039 train(self.generator_id) 3100 3101 def train(self, generator_id): >>> None

Decided in the interest of time to focus on the main deliverables: illustrating the capabilities of the Assistant to query without SQL. Using local files.

1. Connecting to Google Cloud Storage

T+30h -> T+36,5h



Revisiting the connectors issue. The problem was that the correct IAM roles (storage bucket viewer/creator) were not available. Creating a new bucket made it work.

1. Connecting to Google Cloud Storage

T+30h -> T+36,5h



Impact of the connector now working:

- rework the Jupyter Notebook tutorial
- record new session with Assistant

1. Connecting to Google Cloud Storage T+30h - T+36,5h

But...

The Assistant and GCS were having other issues,
unrelated to my original blocker.

2. GCS - Assistant Inconsistencies T+36,5h -> T+38h

```
Let's start by loading the data from the CSV files.
```

```
python
# Load customers.csv into a DataFrame
customers_df = connector.query("SELECT * FROM read_csv_auto('gs://mostlyai-challenge-bucket/customers.csv')")
customers_df.head()
```

```
Result
```

```
HTTP 400: Error occurred while calling Mostly Core API detail: {message=400: IO Error: No files found that match the pattern "gs://mostlyai-challenge-bucket/customers.csv"}
```

```
python
# Explore the contents of the 'mostlyai-challenge-bucket/' location
bucket_contents = connector.locations(prefix='mostlyai-challenge-bucket/')
```

```
Result
```

```
[]
```

The location mostlyai-challenge-bucket/ appears to be empty, as there are no files or subdirectories listed within it.

The connector failed to connect to GCS, my bucket or the files within. This complicated recording for the explainer video.

3. Python Time-Off
T+36,5h -> T+38h

Notice the blue dot instead of the usual green

Private · Created by umuwa38231 · just now · ⚙

umuwa38231/[Unnamed] ↗

Let's use connector MostlyAIChallenge.

First load the connector via `mostly.connectors.get('3ca7cb19-744c-4292-8117-a6c7fb53c4bc')` and then prompt me for next steps.

```
python
from mostlyai.sdk import MostlyAI

# Initialize the MostlyAI SDK
mostly = MostlyAI()

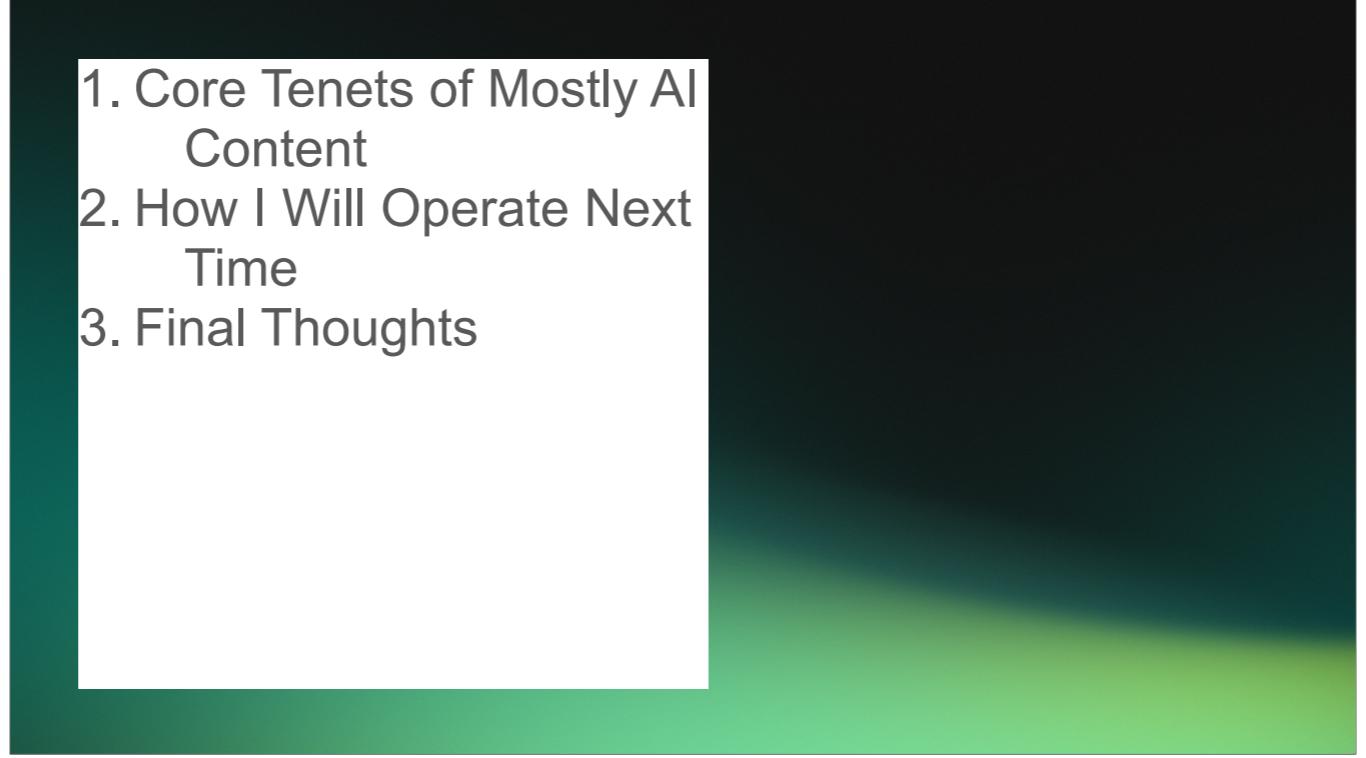
# Load the connector with the given ID
connector_id = '3ca7cb19-744c-4292-8117-a6c7fb53c4bc'
connector = mostly.connectors.get(connector_id)

# Display the connector details
connector
```

In progress...

At 9-10pm Python would not initialise, making the Assistant unable to proceed. Perhaps due to peak traffic? After a while Python was responsive again.

6. Take-Aways

- 
1. Core Tenets of Mostly AI Content
 2. How I Will Operate Next Time
 3. Final Thoughts

1. Core Tenets of Mostly AI Content

SDK, Assistant, tutorials, marketing
copy, documentation, YT presence

Already strong foundations

Already there:

Qualitative, accessible and fun communication

Big focus on tech and business users

Tools in place to be taken seriously (SDK, Assistant, tutorials, marketing copy, YT presence)

-> strong foundation to build upon

1. Core Tenets of Mostly AI Content

Opportunities ++

Potential for upselling and audience understanding.
++ Wild card: user incentives?

Opportunity:

Use the existing and future materials to upsell more:
related features,
documentation,
marketing copy,
equip tech people with talking points towards business users,
demo reservation...

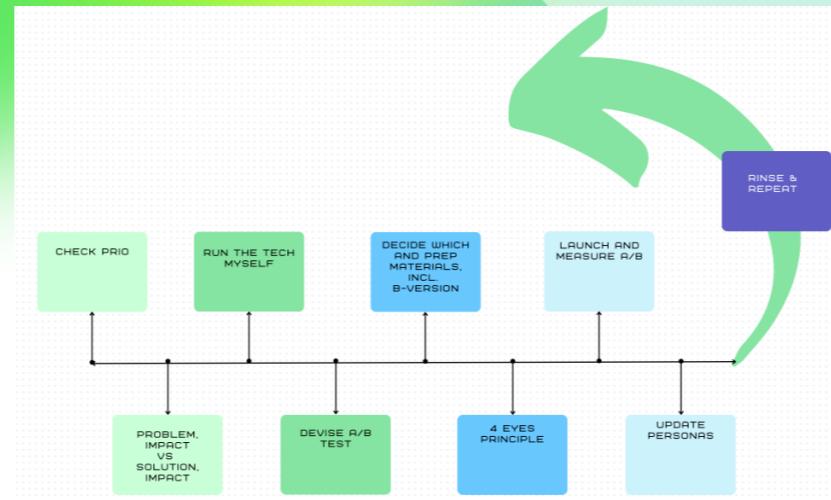
Understand our various audiences thoroughly:

business need that drives our development and content, which drives business and tech interest
(in that order?), which leads to sales

! Wild card: Do we want to incentivise to explore the full range of features on business and tech side?

some sort of automated certification track that people work towards
personal investment of time and effort, plus familiarity with the product create real ambassadors within organisations

2. How I Will Operate Next Time



More learning, more testing, then more learning

1. In case of reworking existing materials:

Check the prioritisation of the feature for which content is being reviewed / redone.

I.e. are our top features (tbd on which metric: popularity, conversion, perceived value) completely covered?

2. Get a clear view on the (business) problem statement and its ramifications. Then how our solution works to resolve the issue and the impact it may have on our clients' business.

! For this point it would be great to have a few Mostly AI ambassadors on speed dial: people who can get excited by our features and are willing to share what that will mean in their work / company.

3. Run the tech by myself, taking detailed notes of what makes me pause or does not work. Those points may offer valuable insights into what additional info to highlight (upsell potential). Or even to file bug tickets to our developers if needed. The key is to get things to work as the end-user would.

4. What can be A/B-tested so we learn as an organisation?

I.e. we test assumptions around the preferences / characteristics of our target audiences.

5. Combine the learnings from points 2 and 3 to prepare tech (e.g. tutorial) and marketing content (video / event / ...)

6. Prepare the B-version of the A/B test

7. If we have the luxury of a 4 eyes principle: have a Mostly AI colleague look at the materials with fresh eyes

8. Launch and measure the A/B split results

9. Communicate findings internally to adjust personas and eliminate the less attractive option

10. Rinse and repeat

3. Final Thoughts

1st:



Regardless of the outcome of this conversation, I want to thank you for having given me this challenge. It was fun for me.

3. Final Thoughts

2nd:



I appreciate the variety of skills I could bring to this task:
my recently acquired Data Science skills and knowledge
my experience in writing copy for developers and business users
my hobby-honed video creating skills
strategic thinking about the players in this game of value exchange and then tactically driving for outcomes
and the opportunity to reflect on my experience in this challenge and discuss these with you via this presentation.

3. Final Thoughts

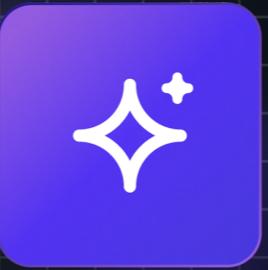
3rd:



If this is the kind of strategic, thoughtful content approach you want more of at Mostly AI — I'd love to help shape it.

MOSTLY•AI

Q&A



|Enter prompt here

+

↑