Slide 0 (title page) (15 secs)

- If I were to give you shoebox filled with my personal diaries from the last 22 years & say, “find entry where I wrote about my decision to apply for the MSIM program?”
    - Where would you start?
    - How long do you think it’d take you?
- I suspect you’d give up pretty quickly.

Slide 1 (45 secs)

- I call this the **shoebox problem**.
    - Ppl store meaningful personal info in handwritten, unstructured formats – hard to access, share, analyze.
- For my project, decided I wanted to try tackle this problem.
- **Motivation**: develop strategy to convert diary entries into portable, accessible digital structure - enables search, analysis, long-term preservation.
- **Aim**: focus on process rather than building ‘perfect’ info product.

Slide 2 (1 min)

- **When I first started mapping out my strategy:**
    - Envisioned using real diary entries.
    - Already taken photos of 8,000+ entries, saved in JPEG.
    - I started playing w/ Optical Character Recognition – OCR – tools, Python & AI to experiment with extracting text.
    - But realized it was more important to focus on testing end-to-end process (rather than spending time extracting text).
- To be clear on what’s **in scope** for The Diary Project:
    - Using AI to create 300 dummy diary entries
    - Developing sample JSON structure, taxonomy, metadata.
    - Building info system to host entries via web-based API.
    - Interactive filters & simple user interface.
- **Key requirements**
    - Will eventually use personal data: Strong emphasis on security.
    - Easy & cost effective to maintain.
    - An enjoyable user experience.

Slide 3 (45 secs)

- In terms of existing info structure, a quick snapshot of what I've got to work with.
- **22 hard copy diaries**:
    - Poor quality: messy writing, inconsistent layout
    - Stored in shoeboxes (mine in Australia, can't access at all)
    - Means it's basically impossible to retrieve info from them, or to move them around.
    - Concerned they'll be damaged, lost, read by prying eyes.
- **Digital files**.
    - An improvement: they're high quality & stored in cloud.
    - But current folder structure is terrible.
    - Although it's structured by year, how am I supposed to find anything beyond that?

Slide 4 (30 secs)

- So I started thinking about what transformation would look like:
    - I wanted my new info product to reflect a thoughtfully designed taxonomy, with relevant metadata & controlled vocabulary.
- I designed this personalized taxonomy
    - 4 top-level categories
        - Time, Theme, Event, Place.
    - Range of sub-categories and specific terms.
    - **Goal**: support filtering functionality later.
    - An iterative process, terms evolving as I test.

Slide 5 (30 secs)

- So a quick recap on what the transformation process involves:
    - Let's pretend for a sec I'm not using the dummy data.
- **Information** itself remains the same
- But the **format & structure** are changing: unstructured hard copies – JPEG image – highly structured JSON w/ defined fields
- **Access methodology** evolving from manual – digital access (Dropbox) – web-based API

Slide 6 (45 secs)

- Let's talk a bit more about the transformation process
- One of most important steps is to convert messy, unstructured data into machine-readable JSON format.
    - [Click to show JSON structure]
- This structure meets my requirements
    - As it reflects my new taxonomy & every metadata field ties back to those goals in my info story
- Now that we've got our new structure
    - Next steps: build a simple web-based API to host our data, with multiple endpoints & a front-end user interface
    - Accessible to users on their personal devices via a public web link, served using Flask and nGrok.
    - Enable them to access, filter & retrieve data interested in.
- So let's see what my first attempt at doing this looked like….

Slide 7 (45 secs)

- [Show video demo]
- Our first test is **endpoint accessibility**.
    - Got Flask running in the background.
    - Returned a ton of dummy diary entries – great!
- Now I'm experimenting with different filters: people mentioned, country where entry was written, date range, min OCR confidence.
- Just to explain OCR confidence: represents certainty in accuracy of extracted text
    - I've added this field to account for later use of OCR tools in data extraction process.
- Last action was to put a numerical entry into one of the open-text fields, to check it didn't return any data.
    - It didn't – great!

Slide 8

- But how can I be sure that data being returned is correct & complete?
- What I'm looking for in terms of **completeness:**
    - That my GET request returns a full list of entries in JSON
    - These entries include all expected metadata fields: date, year, text, people, topics, etc.
    - And my use of filters refines the data as intended.
- In terms of **accuracy**, I'll be looking for:
    - An OCR confidence score ideally over 0.95
    - *(95% certain in accuracy of diary entry – great!)*
    - A manual scan of the text looks good.
- I'll be concerned if any of those conditions are not met.

Slide 9

- So let's summarize those ideas into some **key quality goals**:
    - JSON schema validated.
    - Metadata fields populated consistently.
    - Returned data in line with expectations.
    - OCR level > 0.95
- I'm identifying any **issues** by
    - Flagging low-confidence OCR for review
    - Conducting manual spot checks
    - A range of functional & performance tests
- I've demonstrated some of these **functional tests** already
- At the moment, my info product has not yet passed all tests
    - Can access end point & return all data
    - Single filters are working fine
    - Abnormal inputs don't return any data
    - But combinations of filters are not yet working well – need to remediate.
- I'm also interested in **performance tests** to ensure response time is reasonable & filters are working efficiently
- Security testing is also on my radar, and that's a future priority.

Slide 10

- Reminder of 3 key requirements mentioned at outset:
    - Strong security emphasis
    - Easy & cost effective
    - Enjoyable UX
- I've made good process on those aspects, but there's still much more work to be done.
- Next steps:
    - Address filtering issue.
    - Robust data security plan, inc. user authentication & tests.
    - Explore cost-effective cloud hosting options.
    - More sophisticated UI – I'd love it to look like this
    - Transition to use of real data.
- Later: Explore advanced sentiment analysis and synthesis, leveraging AI tools
    - But requires further thinking re: risk management.
- Really excited to take it forward beyond scope of our class!
- Thank you - any questions?