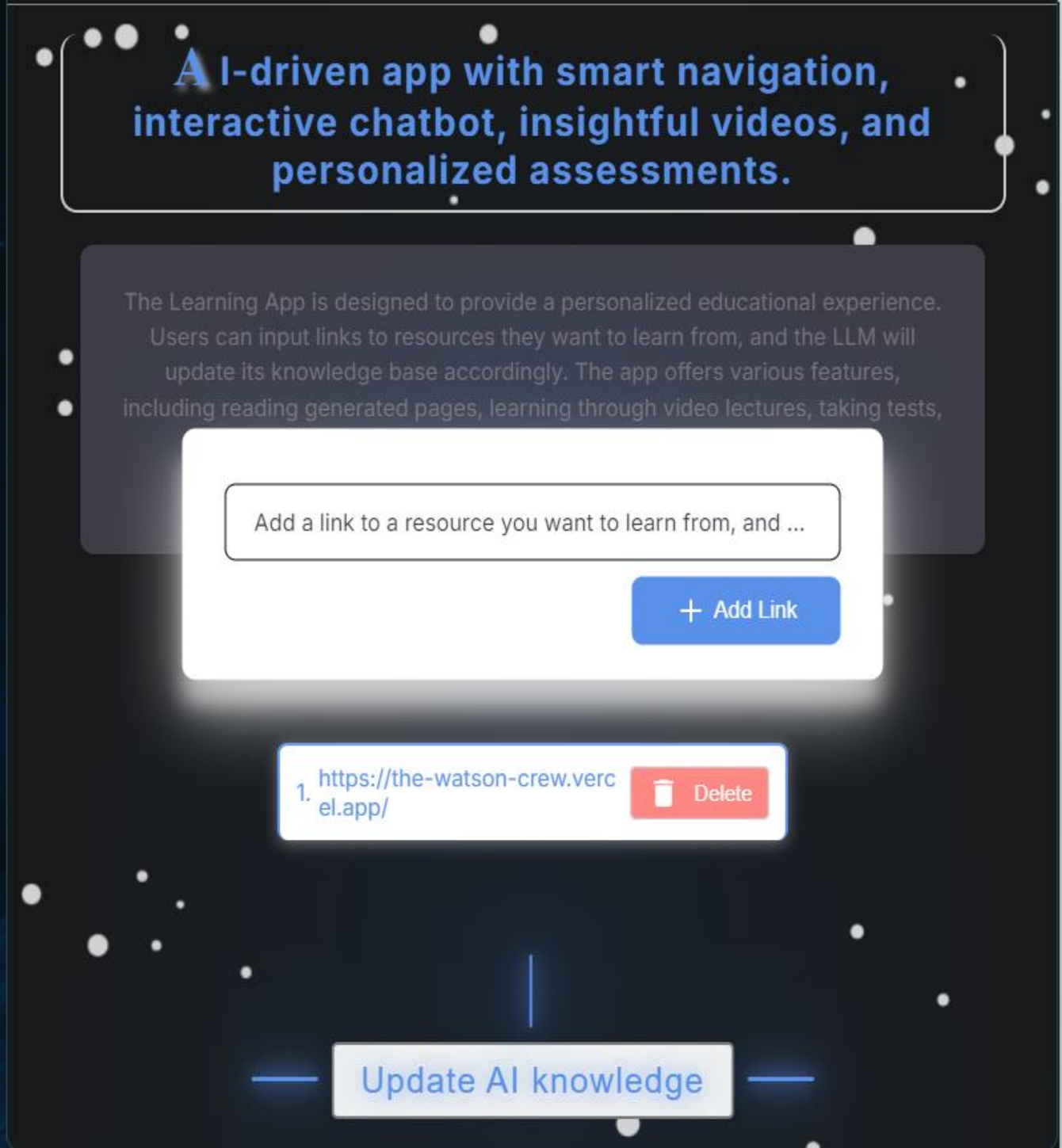
THE WATSON CREW

AI-driven app with smart navigation, interactive chatbot, insightful videos, and personalized assessments.



Problem

1. **Un-Interactive Bots**

   Many service bots today only respond in plain text, which can make interactions feel less engaging.

2. **Upper-Bound on Agent Tools**

   Having too many tools available can overwhelm agents and potentially degrade their performance.

3. **Reading is Tough**

   Users often prefer engaging content formats, such as videos or interactive media, over lengthy text-based content like blogs.

4. **Maintenance of Tools**

   Agents are provided with a static list of tools for each query, which can lead to inefficiencies if the tools are not updated or optimized based on the query's context.

Solution

Directed Acyclic Graph (DAG) for Queries

- Each query is a directed acyclic graph (DAG) in itself.
- Certain parts of the query contribute to answering other parts.
- This idea allows segmenting the query into directed sub-tasks.
- Solve and get the final response for each task topologically, enriching each segment with metadata like associated images, web links, etc.
- Serve the final response as a concatenation of enriched responses from each node in the graph.

Approach

Agent Hierarchy

- ✓ Each agent has a set of agents and tools available to it, similar to how a prime minister has a set of ministers under them.
- ✓ This hierarchical structure helps manage tasks efficiently.

DAG Module

- ✓ Each agent has a module for creating a directed acyclic graph.
- ✓ The DAG module establishes the relationship between the input/output of tools and agents.

Agent Runner

- ✓ After exploring connectivity using the DAG module, each tool is run in a topological manner.
- ✓ The output of a parent tool node is available to children as context in their prompt.

Node Level

- ✓ Each node in the graph is either a tool or another agent itself.
- ✓ Each node handles contrastive parts of the original query and returns a response along with metadata

App Features

Knowledge Ingestion

- Provide the sources to build knowledge from, PDF or web links

Making Lecture Notes

- Performs chunking and embedding on documents. Then apply PCA on embeddings and do clustering of reduced embeddings. I am ordering clusters to maintain the flow of thoughts.

Video Lecture

- It also provides a video lecture assisting in understanding concepts

Organizing Quiz

- An objective quiz to test your understanding of the concept.

Technologies Used

- Watsonx Models
  - ibm/granite-13b-chat-v2
- Backend
  - Django
- Frontend
  - Next JS

Summary

The Learning App is designed to provide a personalized educational experience. Users can input links to resources they want to learn from, and the LLM will update its knowledge base accordingly. The app offers various features, including reading generated pages, learning through video lectures, taking tests, and chatting with the AI.

THANK YOU