

Home Assignment

AI Task: Create a Text to Sequence of Nodes

Objective: Create a PoC of an AI system capable of Text to Sequence of Nodes. We expect an AI pipeline to take as input a text prompt and output a sequence of nodes. You are free to decide on the format of the output.

Examples

Prompt: Navigate to a new page after a delay of 3 seconds when the user clicks a button.

Sequence of Nodes:

1. [OnClick]
2. [Delay]
3. [Navigate]

Prompt: Fetch user data and display it in a modal when a button is clicked.

Sequence of Nodes:

1. [OnClick]
2. [FetchData]
3. [DisplayModal]

Prompt: Reduce a list of scores to find the highest score and log the result.

Sequence of Nodes:

1. [Reduce]
2. [Log]

Prompt: Cache fetched data to improve performance and display the data on the screen.

Sequence of Nodes:

1. [FetchData]
2. [CacheData]
3. [Show]

Prompt: Log a message when a key is pressed and display the key value on the screen.

Sequence of Nodes:

1. [OnKeyPress]
2. [Log]
3. [Show]

Prompt: Highlight an element when the mouse enters it and remove the highlight when the mouse leaves.

Sequence of Nodes:

1. [OnMouseEnter]
2. [Highlight]
3. [OnMouseLeave]
4. [Show]

Prompt: Filter out items that are out of stock and sort the remaining items by price before displaying them on the screen.

Sequence of Nodes:

1. [Filter]
2. [Sort]
3. [Show]

List of possible Nodes:

Event Nodes

- **[OnVariableChange]:** Triggered when a specified variable changes value.
- **[OnKeyRelease]:** Triggered when a key is released.
- **[OnKeyPress]:** Triggered when a key is pressed.
- **[OnClick]:** Triggered when an element is clicked.
- **[OnWindowResize]:** Triggered when the window is resized.
- **[OnMouseEnter]:** Triggered when the mouse pointer enters an element.
- **[OnMouseLeave]:** Triggered when the mouse pointer leaves an element.
- **[OnTimer]:** Triggered at specified time intervals.

Action Nodes

- **[Console]:** Prints a message to the console.
- **[Alert]:** Displays an alert message.
- **[Log]:** Logs information for debugging purposes.
- **[Assign]:** Assigns a value to a variable.
- **[SendRequest]:** Sends a network request.
- **[Navigate]:** Navigates to a different URL or page.
- **[Save]:** Saves data to local storage or a database.
- **[Delete]:** Deletes specified data or records.
- **[PlaySound]:** Plays an audio file.
- **[PauseSound]:** Pauses an audio file.
- **[StopSound]:** Stops an audio file.

Transformation Nodes

- **[Branch]:** Conditional node that branches based on a true/false evaluation.
- **[Map]:** Transforms data from one format to another.
- **[Filter]:** Filters data based on specified criteria.
- **[Reduce]:** Reduces a list of items to a single value.
- **[Sort]:** Sorts data based on specified criteria.

- **[GroupBy]**: Groups data by a specified attribute.
- **[Merge]**: Merges multiple datasets into one.
- **[Split]**: Splits data into multiple parts based on criteria.

Display Nodes

- **[Show]**: Displays information on the screen.
- **[Hide]**: Hides information from the screen.
- **[Update]**: Updates the display with new information.
- **[DisplayModal]**: Displays a modal dialog.
- **[CloseModal]**: Closes an open modal dialog.
- **[Highlight]**: Highlights an element on the screen.
- **[Tooltip]**: Shows a tooltip with additional information.
- **[RenderChart]**: Renders a chart with specified data.

Data Nodes

- **[FetchData]**: Fetches data from an API or database.
- **[StoreData]**: Stores data in a variable or storage.
- **[UpdateData]**: Updates existing data.
- **[DeleteData]**: Deletes specified data.
- **[CacheData]**: Caches data for performance improvement.

Context & Resources:

- Incari Studio: [Incari Studio](#)
- Logic Editor: [Logic Editor](#) | [Incari Studio](#)

Requirements:

- **LLM Setup:**
 - Set up a Lightweight Language Model to generate a sequence of nodes. Example: [google/codegemma-7b · Hugging Face](#) or any other 7b model you judge appropriate.
- **UI Setup:**
 - Enable users to interact with it.
- **Write Tests:**
 - Create unit tests and integration tests to validate the functionality and accuracy of your AI system. Ensure that it correctly retrieves information from the paper.
- **Dockerization:**
 - Dockerize your AI system for easy deployment and isolation.
- **GitHub Repository:**
 - Create a GitHub repository to host your code.
 - Provide the repository link for review.

Evaluation Criteria:

- **LLM Implementation:**

- Use small model
- Fast inference
- **Evaluate the system performances:**
 - Define metrics to evaluate the quality of the generated output
- **Simple UI implementation:**
 - Users can interact with the tool
- **Tests:**
 - Well-documented and comprehensive unit tests and integration tests to ensure the system's correctness.
- **Clean Code:**
 - Maintain clean and well-documented code with appropriate comments and code organization.
- **Dockerization:**
 - The ability to build and run the Docker container successfully.
- **Focus on Topic:**
 - Ensure that the AI system remains on-topic and doesn't engage in unrelated node generation.

Time estimation to complete the task:

- 4hrs

We will discuss your PoC during the interview :).