The software design architecture is shown in the `react-architecture.png` file.
The main approach is to keep components in the ReAct implementation decoupled, flexible, and extendable.
The core component is the **Reasoner**, which combines reasoning and acting.
Acting is handled through plugins. The program currently supports two types of plugins: `WikipediaPlugin` and `WolframAlphaPlugin`

- The `WikipediaPlugin` connects to the Wikipedia API and sends queries to search or look up information.
- The `WolframAlphaPlugin` connects to the WolframAlpha API and queries for mathematical calculations.

To add another plugin to enhance ReAct, simply implement the `ReActPlugin` class and configure it via `appsettings.json` by toggling the corresponding plugin name. Below is an example of how to configure plugins:

```
"plugins": {
  "wikipedia": "on",
  "wolfram": "on",
  "other": "off"
}
```

Each plugin includes a prompt along with a few-shot set of examples to guide behavior, and defines its own action space. For the **Wikipedia plugin**, the action space is `['search', 'lookup', 'finish']:`

- `search` – searches Wikipedia using the keyword provided by the LLM.
- `lookup` – looks up keywords provided by the LLM if the search does not return an answer.
- `finish` – returns the final answer.

For the **Wolfram Alpha plugin**, the action space is `['compute', 'finish']:`

- `compute` – performs mathematical computations using the WolframAlpha API.
- `finish` – returns the final answer.

Note that, to use Wolfram Alpha Plugin, you need to add WOLFRAM_API_KEY in env variables.

You can use this WOLFRAM_API_KEY = `WQ9U7Q-HK5W5UPEPV.`

Context persistence:

The program supports chat context persistence through two different options: **MongoDB**, and a **Dummy DB**. Below is an example of how to configure persistence:

```
"persistence": "mongodb",
"mongodb": {
  "uri": "mongodb://localhost:27017/",
  "db": "chat_db",
  "collection": "sessions"
}
```

By setting the **persistence** value to `"dummy"`, the program will skip storing chat context.
If you choose the **MongoDB** option, make sure you have a MongoDB server up and running.
For a serverless option, use **Dummy**.

When choosing MongoDB persistence, simply pass the conversation session_id to continue within the same context. Program will load data with the provided session_id.

Intelligence Provider:

Currently, the program only supports **Gemini**: model": "gemini". If you want to change the underlying LLM, you simply need to implement the `LLMModel` class and configure it in the program settings.

Potential improvements not implemented due to time constraints:

- Asynchronous execution for external connections (e.g., database operations or plugin API calls) is a potential improvement to enhance performance.
- Implement a more sophisticated search mechanism that retrieves actual information from a knowledge base
- Optimize prompt structure to reduce the number of tokens used, improving efficiency.
- Overall code quality may be improved.