AI AGENT WORKSHOP:
Cross-App Autonomous Agent

1. What is the problem you are trying to solve?

While Large Language Models (LLMs) show impressive text processing capabilities, they still struggle with performing autonomous real-world tasks that require interaction with external interfaces (APIs). The central problem is the inability of current agents to perform Long-horizon planning and Cross-app reasoning to reliably complete complex end-to-end tasks.

---

2. Describe briefly, in high level your presumed solution

The solution is the development of an Autonomous AI Agent embedded within the AppWorld environment. The agent will receive natural language instructions and use a reasoning engine to decompose the task into sub-actions. It will execute API calls to various applications (such as email, banking, and calendar) within the simulated environment, analyze system responses, and update its actions until the task is complete.

---

3. Are there other approaches?

- **RPA (Robotic Process Automation):** A rule-based approach that lacks the flexibility and natural language understanding required for dynamic tasks.
- **Single-App Agents:** Agents designed for a specific application that cannot integrate information from multiple sources.
- **Standard Chatbots:** Models that can suggest a textual plan of action but lack the capability to execute it against digital interfaces.

---

4. Who are the expected users of the application?

- **AI Researchers:** Users who utilize the system to test and measure the performance of language models in operational environments.
- **Autonomous Systems Developers:** Those seeking a framework to build digital "personal assistants" capable of managing complex administrative tasks.
- **Day to day users:** Those who wish to manage cross-app tasks with an AI personal assistant

---

5. What will be the main features and flows of the user?

- **Natural Language Input Interface:** The user defines a goal (e.g., "Schedule a meeting based on the hotel availability found in my email").
- **Planning and Control Loop:** The agent breaks the request into steps, executes them, and verifies success or failure at each stage.
- **Multi-System Integration:** The ability to retrieve data from one app (Email) and input it into another (Calendar) via APIs.
- **Trace Dashboard:** A display showing the sequence of actions performed by the agent for monitoring and transparency.

---

6. Are there any external dependencies?

- **AppWorld Engine:** The execution and simulation environment for the applications.
- **LLM API:** Access to powerful language models (such as GPT-4 or Claude) acting as the agent's "brain".
- **Development Libraries:** Frameworks for building agents, such as LangChain or CrewAI.

Additional Project Details:

- **Technology Stack:** Python, LangChain/AutoGPT for agent management, OpenAI/Anthropic APIs, and the AppWorld SDK.
- **Measurable Outcomes:** Achieving a high Success Rate on AppWorld benchmark tests compared to established baselines.
- **Challenges Faced:** Managing long context windows and ensuring accuracy when extracting parameters for API calls.

Created by Lior Rusanovsky, Omer Zilberberg, Gilad Polikar.