FinAgent Hackathon

Conducted by Techfest, IIT-Bombay in Collaboration with JIO Financial Services

Problem Statement

Objective:

Build a **"Browser-Based Action Agent"** proof-of-concept; an intelligent system that automates end-to-end financial workflows on a web interface. The agent should accept natural language commands from a user (e.g., "Pay my electricity bill") and autonomously navigate a banking website to execute the task; clicking buttons, filling forms, and validating details; just like a human would. The main goal is to demonstrate the future of **"Agentic AI"** in finance, where systems don't just answer questions but actively perform tasks to reduce friction and simplify complex user journeys.

Guiding Principles

- **Action Over Chat:** Your primary goal is to build an agent that *does* things. It shouldn't just tell the user how to pay a bill; it should open the browser and pay it for them.
- **The "Conscious Pause" (Intelligent Friction):** Speed is good, but safety is paramount. Your agent must know when to stop. It should intelligently identify high-stakes moments (like hitting "Transfer") and consciously pause to seek human confirmation.
- **Resilience & Error Handling:** The web is unpredictable. Your agent should handle slow loading pages, pop-ups, or changed button positions gracefully without crashing.

Scope & Required Deliverables

Your solution should be a "Web Action Agent" that automates a set of financial tasks on a simulated banking website (provided by you).

1. The "Action Brain" (Intent Engine)

- Your system must accept natural language commands (Text or Voice) from a user.

- It must correctly interpret the user's intent (e.g., "Buy Gold" vs. "Pay Bill") and extract key details (Amount: ₹500, Biller: Adani Power).

2. The "Digital Hand" (Web Automation Module)

- Your agent must launch a web browser and interact with a "Dummy Banking Website" (a simple HTML simulation you will create).
- It must autonomously navigate menus, select options, and input data into fields based on the user's command.

3. The "Conscious Pause" Mechanism

- Implement a mandatory **"Stop-and-Confirm"** protocol.
- Before clicking any "Pay", "Transfer", or "Submit" button, the agent must pause and display a confirmation prompt to the user (e.g., *"I am ready to transfer ₹500 to Mom. Shall I proceed?"*).
- It should only proceed after receiving explicit user approval.

4. The User Dashboard (Command Center)

- A simple interface where the user can:
  - Type or speak their command.
  - Watch a "Live Feed" or log of what the agent is doing.
  - Approve or Reject critical actions during the "Conscious Pause."

5. Deliverables

- A working prototype of the Action Agent.
- Source code with clear documentation.
- A "Dummy Bank" website (HTML/Localhost) used for the demo.
- A demo video showcasing the agent executing 3 distinct financial workflows.
- A report detailing the architecture and design decisions.

Key Features to Demonstrate

End-to-End Workflow Execution: Demonstrate a single user command (e.g., "Invest ₹100 in Digital Gold") resulting in the automatic navigation to the Gold page, entry of the amount, and preparation of the payment.

The "Conscious Pause": Show the agent stopping exactly at the payment confirmation screen. Demonstrate that it *cannot* proceed until the user clicks "Approve" on your dashboard.

Error Handling & Correction: Simulate a scenario where the agent tries to enter an invalid amount (e.g., negative number). Show the agent detecting the error message on the website and asking the user for the correct amount.

Technical Considerations & Stack (Guidance)

We encourage you to move beyond basic scripts and explore the future of AI automation. Surprise us with efficiency!

- **The Brain (AI Models):**
  - *Standard:* OpenAI/Gemini APIs for text parsing.
  - *Explorer Mode:* Try **Vision-Language Models (VLMs)** (like GPT-4o or Gemini 1.5 Pro) that can "see" the website screenshot and decide where to click, rather than reading HTML code. This is how the next generation of "Computer Use" agents work.
- **The Hands (Automation):**
  - *Standard:* Selenium (Reliable but heavy).
  - *Explorer Mode:* Try **Microsoft Playwright** or **Puppeteer** for faster, more stable execution. Look into **"Large Action Models" (LAMs)** concepts or frameworks like **LangGraph** or **AutoGen** to make your agent smarter at planning multi-step tasks.
- **Target Environment:** You do **not** need to automate the real Jio app. You should build a simple **"Dummy Bank"** website (HTML/CSS) running on localhost.

Success Metrics (Evaluation Criteria)

Your prototype will be evaluated based on its ability to meet the sponsor's key objectives and functional requirements.

- **Task Completion Rate:** Measures the percentage of attempts where the agent successfully completes the entire workflow (from command to final success screen) without crashing.
- **Safety Adherence: Critical Metric.** Did the agent successfully execute the "Conscious Pause" before every sensitive action?
- **Intent Accuracy:** Assessment of how well the agent understood complex or vague commands.
- **Innovation in Automation:** Did you use standard brute-force coding, or did you explore intelligent UI understanding (Vision AI/LAMs) to make the agent more adaptable?

Competition Structure

This competition will be conducted in two rounds. Each Team can have a maximum of 4 members.

Round 1: Abstract & Design Strategy (Virtual)

Objective: To demonstrate your conceptual understanding of the "Action Agent" problem and present a robust architectural blueprint. This round focuses on your logic, security approach, and technical feasibility before you start building. **Goal:** To convince the judges that your team has a clear, workable plan to build a secure and intelligent financial agent. **Submission Method:** All details must be submitted via the official Google Form. **What to Submit (via Google Form):**

- **Team Details:** Name, ID, and Member contacts.
- **Abstract (PDF Upload):** A single PDF (max 5 pages) containing:
    - **Architecture Diagram:** A visual representation of how your LLM (Brain) will communicate with the Web Automation script (Hands).
    - **Workflow Strategy:** A detailed flowchart for **one** sample task (e.g., "Buying Gold"), showing every logical step, decision node, and error check the agent will perform.
    - **"Conscious Pause" Mechanism:** A detailed explanation of how your agent will detect high-stakes actions (e.g., payment confirmation) and how the "Stop-and-Confirm" UI will function to ensure user safety.
    - **Tech Stack Selection:** A justification of the specific tools (e.g., Playwright vs. Selenium, GPT-4o vs. Llama 3) you plan to use. Explain why your choice offers the best balance of speed, cost, and accuracy.

Submission Link for Round 1: https://forms.gle/Zjrkk1UZdjBBjbD99

Round 2: The Grand Finale - Live Demo & Presentation (At IIT Bombay)

Objective: To present the fully built, working "Action Agent" to the panel of judges at Techfest, IIT Bombay. **Goal:** Showcase the reliability, speed, and real-time adaptability of your agent in a live environment.

Format:

- **10-minute Presentation & Live Demo:** You must bring your laptop with the "Dummy Bank" website and your Agent installed. You will run live demos

of the agent executing **3 distinct financial workflows** (e.g., Bill Pay, Gold Buy, Profile Update) on the spot.

- **The "Surprise Command":** During the demo, judges may ask you to execute a variation of a task (e.g., *"Try paying ₹1000 instead of ₹500"* or *"The biller website is loading slowly, does your agent crash?"*) to test if your agent is truly dynamic or just hard-coded.
- **5-minute Q&A:** A technical defense of your code structure, focusing on latency, security, and how your agent handles "hallucinations" (wrong actions).

Timeline

Date	Event
Last date for Round 1 submission	10th Decmber 2025
Round 1 Results	13th December 2025
Round 2: Final Presentation at IIT Bombay	22nd-24th December 2025

Prize Money

Prize money will be awarded to the top three teams per theme via NEFT by the latest May 2026. Winners must email the following to **devam@techfest.org** immediately after results:

Subject: "FinAgent Hackathon, [Team ID] – [Position]" (e.g., "FinAgent Hackathon, TF-250245 – 1st Position")

Body:

1. Account Holder's Name
2. Account Number
3. Bank Name and Branch
4. IFSC Code
5. A photograph of the Bank Passbook as proof