

GSoC 2026 Build a GUI Agent with local LLM/VLM for OpenVINO.

Name – Sahil Bhatane

University – Savitribai Phule Pune University

Location – Pune, India (GMT +5:30)

Email – sahilbhatane6@gmail.com

GitHub - [Sahilbhatane](https://github.com/Sahilbhatane)

Mentors -

i. [Ethan Yang](#).

ii. [Zhuo Wu](#).

Prerequisites -

SR. No.	Description	Type w/links	Status
1	Base MVP - The agent follows a strict Perception -> Planning -> Action -> Feedback loop	MVP	Completed/working
2	[Bug]: GPU_ENABLE_LARGE_ALLOCATIONS ignored for large constants during GPU compile (ProgramBuilder before program)	Issue - #34967	Open
3	[GPU] Improve max-allocation error guidance and enable_large_allocations docs	PR - #34894	Approved and queued
4	[GPU] Forward CMAKE_SYSROOT and CMAKE_FIND_ROOT_PATH to oneDNN GPU ExternalProject	PR - #34861	Open
5	[GPU] Apply enable_large_allocations before ProgramBuilder constant GPU uploads	PR - #35022	Open

Short Bio –

My name is Sahil Bhatane and I am Final year Student in Computer science field; I have keen understanding of technologies I use and always keep an interest in learning new technologies (mostly AI based techs). have a firm hand on all the experienced shared below.

The open-source world fascinates me and I have been an active part of this world. In fact, I proudly participated and successfully completed the SWoC 2026, making valuable contributions and got ranked #63. But I'm not limiting it to events, open-source made me contribute to projects with more than 36k stars and millions of downloads, companies like Vercel, tenstorrent, CX, rxresume etc are very famous and most downloaded technologies in their field and I have active contribution in these companies. Getting contributions to OpenVINO will also add great tag on my High-stars contributing companies list.

Experience -

1. Language – Python/C++/JS

- **Python** – Multiple AI/ML, LLM and Interactive APP based Projects most of them are on my GitHub but some are exclusively build from companies (CX GUI agent).
- **C++** - Contributing to `tt-metal` of [Tenstorrent](#) which is a Neural network OP library.
- **JS** – Multiple contribution on web based projects (most notable will be [rxresume](#) and MetaFlow.)

2. Technology (for Project) –

- **OpenVINO** – Used Mistra 7B instruct int4 model when working with CX.
- **AI/ML** – understanding of ML models and Pipeline end-to-end (with experience), Full understanding of AI/LLM working (from data collection, tokenization, RL, training deployment etc).
- **Vision/multimodal** – Experienced with OpenCV (Projects like Sign languages-based text-to-speech model) and UL-TARS (general use), VLM WebRTC project + vision CNN project + GUI agent's VLM path.
- **Agent-style software** - Vercel AI SDK Contribution, MCP/code-index tooling
- **AI agents** – Highly experienced in using AI based products like Cursor, VS copilot, Coderabbit, V0, Claude, OpenClaw etc. AI models like GPT, Gemini, Opus on API based projects and other tools like selenium and my own personal LLM council (inspired

from Andrej Karapathy) which has 4 agents each for different task – 1. GPT for research and Re-prompting, 2. Claude opus 4.6 for coding, 3. Gemini 3.1 for frontend designs and 4. Sonnet 3.5 for security check.

3. Other –

- **Internships** – Softlearn Pvt., Microsoft’s Micorcert, Deloitte Australia Virtual.
- **Freelance** – Local freelance (Have supplied product like bill management, Medicine handling reports through NoSQL, etc) have digitalized nearly all shops in 1km range near me. Also provided 3 products to my own college for placement, event handling and a quiz-based product.
- **Open-source** – Participated in SWoC26 and ranked #63, which made me be a core contributor to CX (formerly Cortex).

- **About Project – Build a GUI Agent with local LLM/VLM and OpenVINO.**

“I use **UL-TARS** on my system before this project, and I always wanted to know how it actually works, I also used **OpenVINO’s** Mistra 7B instruct int4 model when working with CX for showcasing enhanced LLM based reasoning in simple languages for user, and I wanted to participate in GSoC as events like this gets my interest, knowing OpenVINO before and seeing it in GSoC + coincidentally this projects referred **UL-TARS** and aligned with my tech stacks, It made me instantly choose this project.”

- **Time plan to invest in this project.**

“Since its my final year and I don’t have exam till late June/July, I can give **45-60 per week**, I don’t have any restrictions from my university and I can fully focus because of that”

- **Project –**

- **Abstract –**

This project aims to build a local GUI agent powered by OpenVINO that can understand a user’s natural language instruction and execute it directly on the system.

The core idea is to create a closed-loop agent system that continuously observes the screen, understands UI elements, plans actions, and executes them — similar to how a human interacts with a computer.

So it uses a single Vision-Language Model running entirely through OpenVINO. This keeps the system efficient, consistent, and fully local.

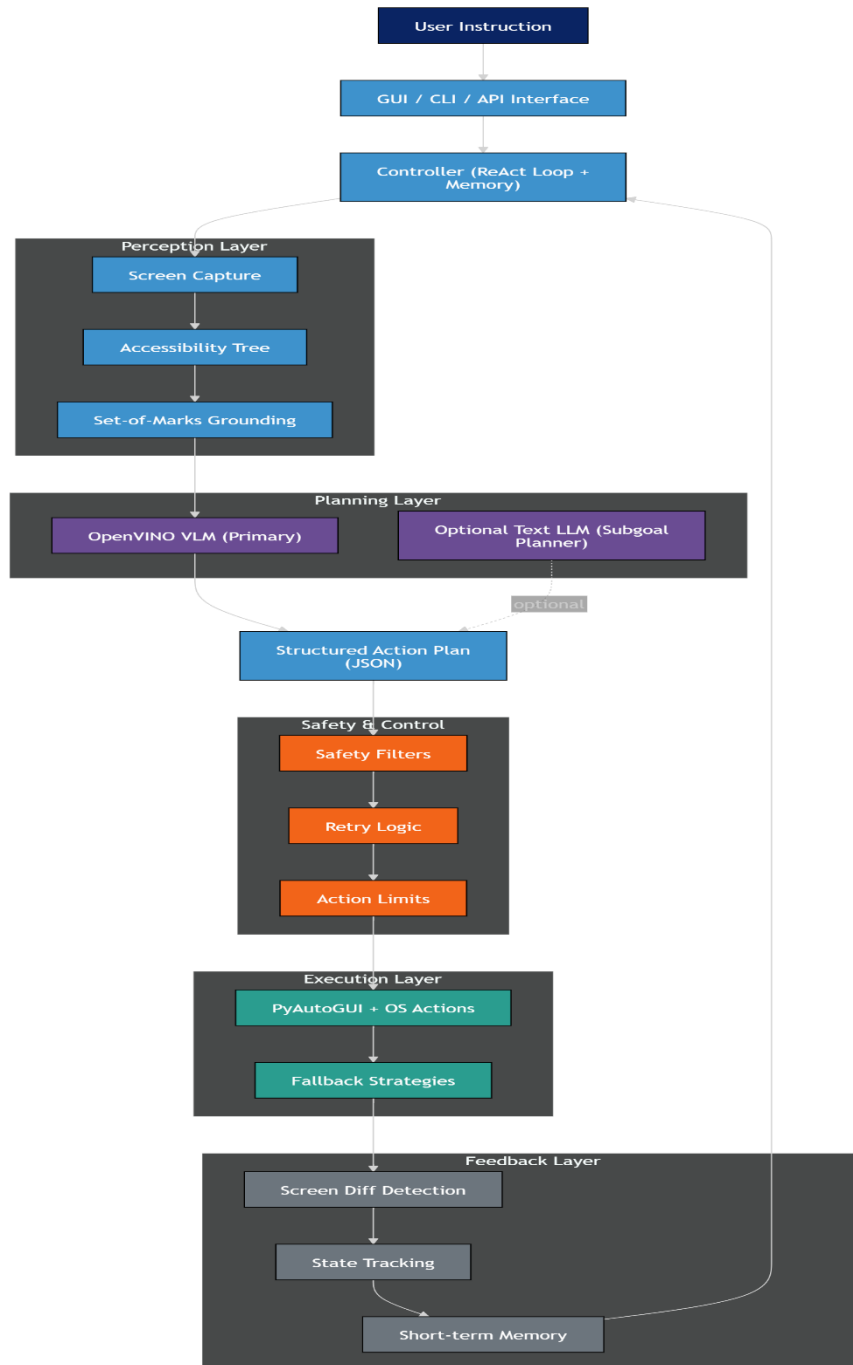
To improve reliability, the agent combines:

- Accessibility APIs for structured UI understanding
- Set-of-Marks overlays for precise grounding
- A safety layer to prevent harmful actions

The final system will support multiple real-world tasks such as:

- Calculator operations
- Text editing in Notepad
- File navigation
- Conditional UI interactions

Intended Flowchart -



Weekly Timeline –

- Community Bonding Period (May 1 – May 25)
- Interact with mentors and finalize project scope
- Study OpenVINO workflows and existing GUI agents
- Finalize architecture and core modules

- Set up development environment and dependencies
- Run initial OpenVINO inference test with selected model

Phase 1: Core System Development -

Week 1 (May 26 – June 1)

Environment Setup & Base Modules

- Set up project structure
- Implement screen capture module
- Integrate basic accessibility extraction (Windows)
- Verify UI element extraction

Week 2 (June 2 – June 8)

Grounding + Model Integration

- Implement Set-of-Marks overlay
- Prepare input pipeline (image + elements)
- Integrate OpenVINO VLM inference
- Validate structured output (JSON actions)

Week 3 (June 9 – June 15)

Action Execution Layer

- Implement executor (click, type, scroll, key events)
- Add fallback mechanisms (coordinate + keyboard)
- Introduce safety checks for restricted actions

Week 4 (June 16 – June 22)

Agent Loop (Core Logic)

- Build perception → planning → action loop
- Add step memory and iteration control
- Implement screen change detection

Week 5 (June 23 – June 29)

Initial Scenarios

- Implement Calculator and Notepad tasks
- Debug grounding and execution issues

- Improve action reliability

Week 6 (June 30 – July 6)

Expanded Scenarios + Interfaces

- Add File Explorer and Paint scenarios
- Implement CLI and API interface
- Validate multi-step task execution

Week 7 (July 7 – July 13)

GUI + Midterm Preparation

- Build PySide6 GUI (control + screen + logs)
- Run full system tests
- Record midterm demo

Midterm Evaluation

- Fully working OpenVINO-based agent
- 4–5 scenarios running successfully
- GUI + CLI functional

Phase 2: Optimization & Expansion -

Week 8 (July 14 – July 20)

Performance Benchmarking

- Compare CPU vs GPU performance
- Measure latency and response time
- Optimize input size and token limits

Week 9 (July 21 – July 27)

Prompt & Accuracy Improvements

- Improve action prediction accuracy
- Handle edge cases and failures
- Add retry mechanisms

Week 10 (July 28 – August 3)

Cross-Platform Support

- Add Linux accessibility backend
- Test system on multiple environments
- Abstract platform-specific modules

Week 11 (August 4 – August 10)

Advanced Tasks

- Add multi-step and conditional workflows
- Improve robustness for real-world usage
- Handle unexpected UI states

Week 12 (August 11 – August 17)

Optional Enhancements

- Explore task decomposition (if needed)
- Improve long-horizon task handling
- Evaluate impact on performance

Week 13 (August 18 – August 24)

Final Testing & Documentation

- Run full regression testing
- Finalize benchmarks
- Clean code and improve documentation

Final Week (August 25 – August 31)

- Prepare final report
- Submit code and documentation
- Final evaluation with mentors

• General Q/A –

Q. How do you know OpenVINO?

A. Used OpenVINO's Mistra model during my work with CX. Was subscribed to discussion channel on GitHub after that, then saw it participating in GSoC.

Q. What do you know about OpenVINO?

A. OpenVINO is a toolkit for optimizing and deploying AI inference efficiently on Intel hardware. It supports model conversion to IR format, Quantization, optimized CPU/GPU inference.

Q. Have you already contributed to the OpenVINO project?

A. Yes, all mentioned above in Perquisites with links.

Q. How could you apply it to your professional development?

A. This will counted as a Real-world experience which is always a great tag for any professional jobs when applying, also this will enhance my skill further more than what I know about open-source and tech stack that I use/will use.

Q. Describe any other career development plan you have for the summer in addition to GSoC.

A. Always trying to apply for jobs that are relevant to my skills, giving Aptitude testing, enrolling certification courses, giving interviews to selected jobs etc. I have actively doing this from last year and getting success from this as from 15 job call interviews I have cracked and got selected 9 of them, but rejected as freelancing and open-source provided more freedom and financial stability than those jobs.

Q. Why should we pick you?

A. skill that I provide and experience and time that I will be investing to solve and provide a proper solution to this project and with contribution I did, I have very close alignment with understanding of the project and working of the organization which is what this project needs.

• **Homework –**

“I did create a MVP for this project and here is the working of current MVP –“

LINK - [Openvino-GUI-Agent](#)

To validate the feasibility of this project, I have already implemented and tested a

working MVP of the OpenVINO GUI Agent, along with a structured evaluation pipeline.

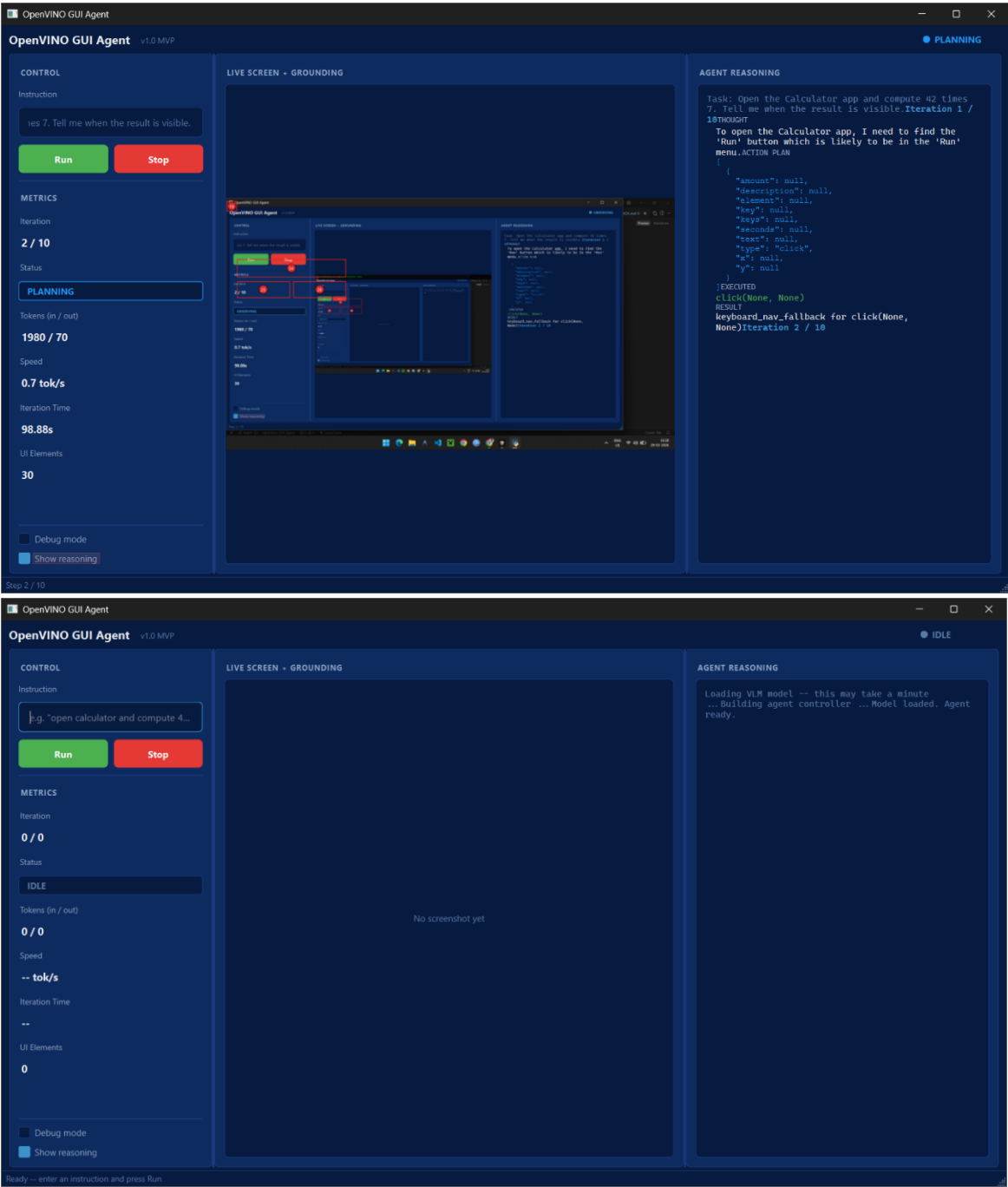
The MVP follows a complete **Perception → Planning → Action → Feedback loop** and has been tested using automated tests, real-world scenarios, and benchmarks.

Features – this also aligns (nearly) with project -

- **Cross-platform:** Windows, Linux, macOS accessibility backends
- **Local inference:** Runs locally via OpenVINO (GPU when available)
- **Observable:** 3-panel UI shows live screen, agent reasoning, and control metrics
- **Safety system:** Blocks dangerous commands (rm -rf, format, shutdown), hotkeys (Alt+F4), and executables (powershell, cmd)
- **Structured errors:** Every error is categorized (model/execution/os/grounding/safety) with recovery info
- **Fallback strategies:** Element click -> coordinate fallback -> keyboard navigation
- **Debug mode:** Saves raw/annotated screenshots, JSON plans, and action logs per iteration

ScreenShots –

GUI working -



Current Flowchart of working MVP -

