

Gen-AI Hackathon

Project Title:

Gesture -Controlled nearby places finder

Team Name:

Coders

Team Members:

- Sai Chaithanya
- Kishan
- Karthik
- Harish

Phase-1: Brainstorming & Ideation

Objective:

Develop an AI-powered hand gesture-based tool to help users find nearby places (restaurants, police stations, airports, malls) using **MediaPipe** and **Google Maps API**.

Key Points:

Traditional location searches require typing or voice input, which may not always be convenient. This project enables users to find nearby places through simple hand gestures, making it hands-free and accessible.

Proposed Solution:

An AI-powered **gesture recognition application** that:

- Detects **hand gestures** using a webcam.
- Maps **finger count** to different location types.
- Fetches and displays **nearby places** from Google Maps API.
- Shows a **map with markers** for selected locations.

Target Users:

- **Travelers** who need quick access to locations.
- **People with disabilities** who may benefit from gesture-based input.
- **Emergency responders** needing rapid location searches.

Expected Outcome:

A fully functional AI-powered **gesture-based search tool** that helps users find locations without typing or voice input.

Phase-2: Requirement Analysis

Objective:

Define the technical and functional requirements for the Gesture-Based Location Finder.

Technical Requirements:

- **Programming Language: Python**
- **Backend: Google Maps API**
- **Frontend: OpenCV + MediaPipe**
- **Database: Not required (API-based queries)**

Functional Requirements:

- **Detect hand gestures** using MediaPipe.
- **Map gestures** to different place categories.
- **Fetch locations** from Google Maps API.
- **Display search results** on a map.
- **Show distance** of each location from the user's position

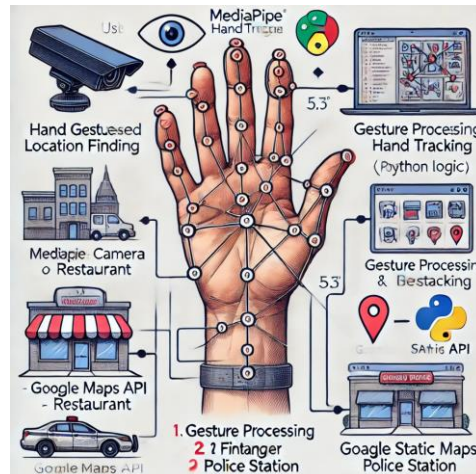
Constraints & Challenges:

- **Ensuring real-time gesture recognition.**
- **Optimizing API calls to avoid exceeding rate limits.**
- **Rendering Google Maps results effectively in OpenCV.**

Phase-3: Project Design

Objective:

Develop the system architecture and user flow.



Key Points:

1. System Architecture:

- User performs hand gesture.
- MediaPipe detects finger count.
- Google Maps API fetches nearby locations.
- Results are displayed as a map and a list.

1. User Flow:

- **Step 1:** User shows **one to four fingers** to indicate the type of place.
- **Step 2:** The system identifies the **gesture**.
- **Step 3:** The API fetches **nearby places** based on the gesture.
- **Step 4:** The **map and list** of places are displayed.

1. UI/UX Considerations:

- **Clear video feed** with hand tracking overlay.
- **Minimal UI** to display search results.
- **Live updates** for better interactivity

Phase-4: Project Planning (Agile Methodologies)

Objective:

Break down development tasks for efficient completion.

Sprint	Task	Priority	Duration	Deadline	Assigned To	Dependencies	Expected Outcome
Sprint 1	Environment Setup & API Integration	<input type="checkbox"/> High	6 hours (Day 1)	End of Day 1	Member 1	Google API Key, Python, OpenCV setup	API connection established & working
Sprint 1	Hand Gesture Detection	<input type="checkbox"/> High	4 hours (Day 1)	End of Day 1	Member 2	MediaPipe setup	Recognizing different hand gestures
Sprint 2	Google Places API Integration	<input type="checkbox"/> High	3 hours (Day 2)	Mid-Day 2	Member 3	API response format finalized	Fetching and displaying places
Sprint 2	Map Rendering	<input type="checkbox"/> Medium	2 hours (Day 2)	Mid-Day 2	Member 4	API response available	Static map with markers
Sprint 3	UI & Performance Optimization	<input type="checkbox"/> Medium	2 hours (Day 2)	End of Day 2	Member 5	Fully working search feature	Smooth UI & real-time responses
Sprint 3	Final Presentation & Deployment	<input type="checkbox"/> Low	1 hour (Day 2)	End of Day 2	Entire Team	Working prototype	Demo-ready project

Phase-5: Project Development

Objective:

Implement core features of the **Gesture-Based Location Finder**.

Technology Stack Used:

- **Frontend:** OpenCV + MediaPipe
- **Backend:** Google Maps API
- **Programming Language:** Python

Development Process:

1. Implement **API key authentication** for Google Maps.
2. Develop **gesture recognition logic** using MediaPipe.
3. Optimize **Google Places API queries** for efficiency.
4. Display **location results on a map** and list format.

Challenges & Fixes:

Challenge	Fix
Unstable hand detection	Fine-tune MediaPipe confidence thresholds
Slow API responses	Cache frequently searched locations
High API request costs	Optimize query radius & request intervals
Poor UI readability	Add high-contrast text overlays

Phase-6: Functional & Performance Testing

Objective:

Ensure that the **Gesture-Based Location Finder** works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	Show 1 finger to find restaurants	List & map of restaurants	✔ Passed	Tester 1
TC-002	Functional Testing	Show 2 fingers to find police stations	List & map of police stations	✔ Passed	Tester 2
TC-003	Performance Testing	API response time under 500ms	API should return results quickly	⚠ Needs Optimization	Tester 3
TC-004	Bug Fixes & Improvements	Fixed incorrect map markers	Correct markers displayed	✔ Fixed	Developer
TC-005	Final Validation	UI should be clear and responsive	Works across all resolutions	✖ Failed - Needs improvements	Tester 4
TC-006	Deployment Testing	Host the app for external access	App should be accessible online	☐ Deployed	DevOps

Final Submission

- **Project Report** based on the template
- **Demo Video (3-5 Minutes)**
- **GitHub/Code Repository Link**
- **Final Presentation**