



(+91) 81976 34433
(+91) 90190 30033

Agentic AI - Introduction

Venkata Reddy Konasani

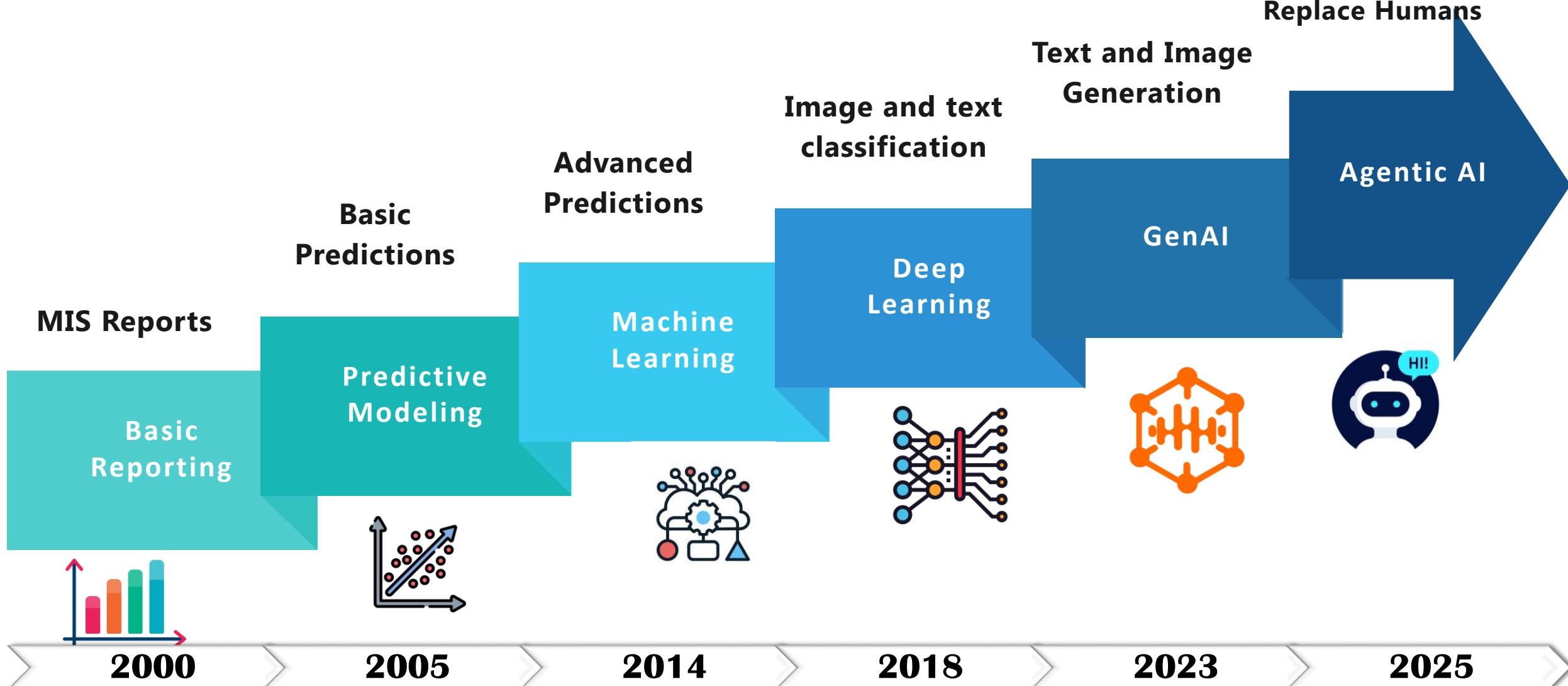
Contents

- GenAI
- AI Agents
- Agentic AI
- Agentic AI Frameworks
- Learning Path

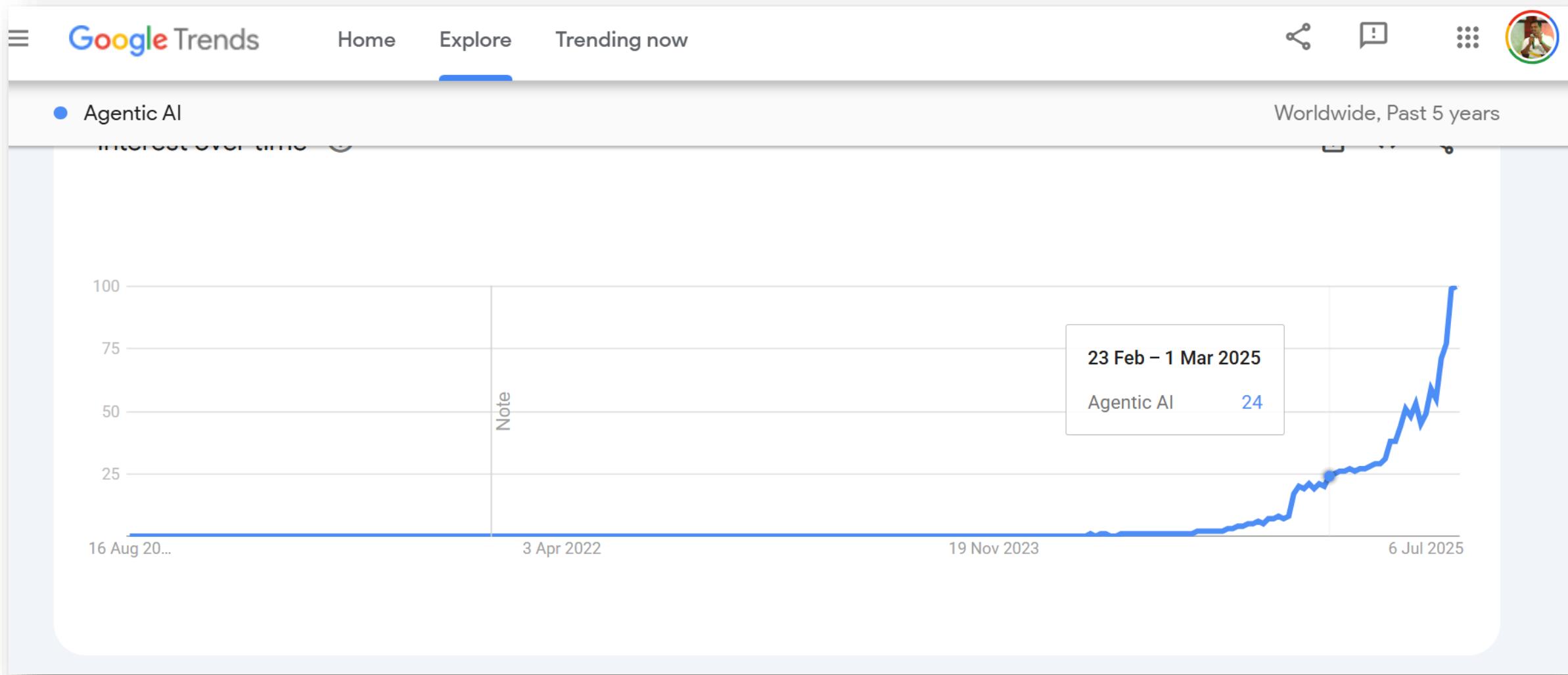


(+91) 81976 34433
(+91) 90190 30033

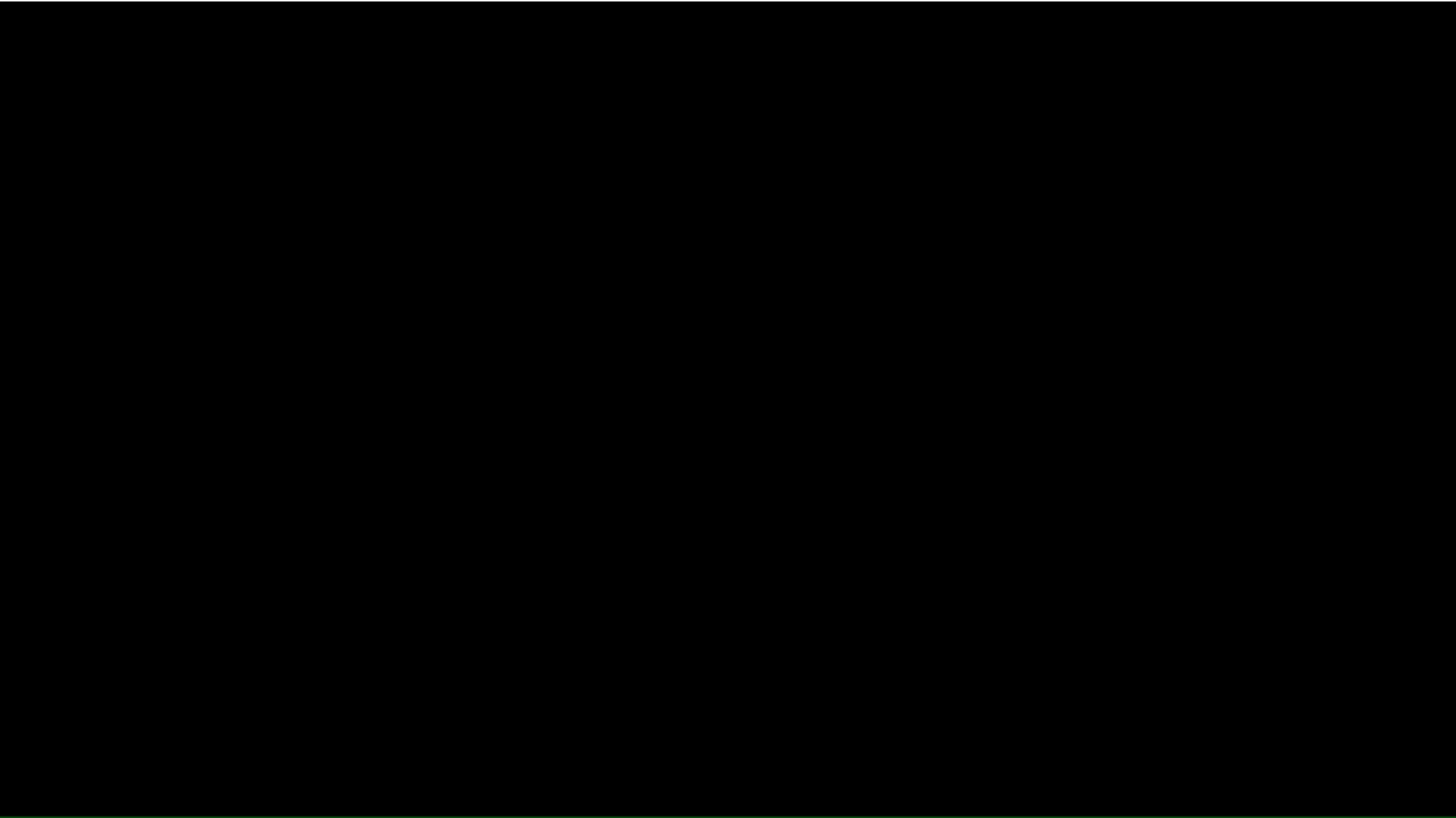
The history of Agentic AI



The history of Agentic AI



Chat GPT Agentic AI System Video



Chat GPT5 – Agentic AI System

ChatGPT 5 ▾

Write end to end code for building an app that can help people to learn spanish language using simple games

Thought

< 2/2 > undo redo copy link up

↓

Write end to end code

Write or code

+ undo redo ✖
∅ ||

ChatGPT can make mistakes. Check important info. See [Cookie Preferences](#)

Spanish Games App ▾

⌚ ↶ ↷ ✖ ⤒ ⤓ ⤔ ⤕ Run code

```

1 import { useState } from "react";
2 import { Button } from "@/components/ui/button";
3 import { Card,CardContent } from "@/components/ui/card";
4
5 // Sample Spanish words
6 const words = [
7   { en: "Apple", es: "Manzana" },
8   { en: "House", es: "Casa" },
9   { en: "Book", es: "Libro" },
10  { en: "Dog", es: "Perro" },
11  { en: "Water", es: "Agua" }
12];
13
14 export default function SpanishLearningApp() {
15   const [current, setCurrent] = useState(0);
16   const [score, setScore] = useState(0);
17   const [feedback, setFeedback] = useState("");
18
19   const handleGuess = (guess: string) => {
20     if (guess === words[current].es) {
21       setScore(score + 1);
22       setFeedback("✅ Correct!");
23     } else {
24       setFeedback(`❌ Wrong! Correct answer: ${words[current].es}`);
25     }
26   }
27 }
```

Chat GPT5 – Agentic AI System

The image shows a ChatGPT interface with the following elements:

- Header:** ChatGPT 5 ▾, three dots, X, Spanish Games App ▾, and a set of icons for refresh, search, and file operations.
- Input Area:** A text box containing the instruction: "Write end to end code for building an app that can help people to learn spanish language using simple games".
- Output Area:** A "Thought" section showing the progress: < 2/2 >. Below it is a message with a downward arrow icon: "Write end to end code".
- Game Card Preview:** A card titled "es Spanish Learning Game" with the question "What is "House" in Spanish?". It lists four options: Manzana, Casa, Libro, Perro, and Agua. The "Casa" option is highlighted.
- Bottom Bar:** A toolbar with a plus sign, a blue circular icon with a white checkmark, a red circular icon with an X, a microphone icon, and a small circular icon with a vertical bar.
- Footer:** A note stating "ChatGPT can make mistakes. Check" followed by a link to "https://beta.openai.com/docs/api-reference/introduction".



(+91) 81976 34433
(+91) 90190 30033

Gen AI

Gen AI

Input : Direct prompt →
Output: LLM Generated

How to improve focus and utilize time effectively ? Give me the answer in bullet points

Here's a direct, bullet-point list you can use:

- **Set priorities:** Write down the top 3 tasks for the day and finish them first.
- **Break tasks down:** Divide big goals into smaller, actionable steps.
- **Time-box work:** Use Pomodoro (25–50 min work + 5–10 min break).
- **Remove distractions:** Silence notifications, keep phone away, use site blockers.
- **Create a focused environment:** Clean desk, quiet space, or noise-canceling headphones.
- **Batch activities:** Check emails/messages at fixed times instead of constantly.
- **Take care of energy:** Sleep 7–8 hours, eat light meals, stay hydrated.
- **Exercise regularly:** Even short walks or quick workouts boost concentration.

Gen AI

- LLMs that generate content (text, images, code, etc.) from prompts
- Generate outputs from input
- Direct prompt → output
- ChatGPT, DALL·E, Gemini
- Low complexity

LLM Limitations

- Standalone LLM
 - Could not do math
 - Could not search internet
 - Could not give answers related to recent facts
 - Could not access current news data
- We need an alternative option

LLM Limitations

```
llm=OpenAI(temperature=0)
print(llm.invoke("Who is the current president of India?"))
```

The current president of India is Ram Nath Kovind.

```
print(llm.invoke("Who is the IPL 2025 Winner?"))
```

It is impossible to predict the winner of IPL 2025 as it



(+91) 81976 34433
(+91) 90190 30033

AI Agents

Software Acting Like Humans

- "Agents" are software programs designed to imitate how humans perform tasks.
- Take creating a PowerPoint presentation as an example.
 - **Thought** - You start by thinking about what needs to be done
 - **Action** - Then search online to gather information.
 - **Observation** - After reviewing the results, you decide if the content is useful.
- If it isn't, you think of a new approach, search again, and evaluate the results.
- This cycle of thinking, acting, and observing is repeated until you achieve your goal.
- **Agents mimic this same** process to complete tasks effectively.

Agents

- Agents are software programs that interact with the
 - Real world (other than LLM training data)
 - External events
 - Current data
 - Beyond LLM trained data
- LangChain offers various types of these interactive agents.
- These agents are designed to automate tasks and handle real-world scenarios.
- Agents are most important and most powerful aspect of Lang chain
- Agents concept got famous because of the ease of use. For enterprise applications we need to use them with caution.



(+91) 81976 34433
(+91) 90190 30033

Agentic AI Systems

Agentic AI System

- A **single AI agent** typically performs a **specific, goal-oriented task** using tools, memory, or reasoning.
 - Example- Access internet, gets the content & context then generates a marketing message
- **Agentic AI** involves multiple specialized AI agents (or components) working together collaboratively or sequentially to achieve a complex, multi-step goal – with planning, decision-making, tool use, and self-reflection.
 - Example - Researches target audience → Writes copy → Designs image → Schedules emails → Posts on social media → Monitors results

Agentic AI System

An **agentic AI system** is more than a plain LLM chatbot. It has:

1. **Autonomy** - it can decide what actions to take without step-by-step human prompting.
2. **Tools / APIs** - it can use external functions like search, code execution, or databases.
3. **Reasoning + Planning** - it can break down tasks into smaller steps and sequence them.
4. **Memory / Context** - often it keeps short- or long-term state for multi-turn tasks.

So, the “*agentic*” part is about the system behaving like an *agent*—taking initiative, choosing actions, and orchestrating tools.

Perplexity AI

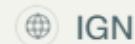
What are the box office collections of the movie super man 2025?

Answer

Images

Sources · 13

Steps



IGN

James Gunn's Superman Inches Towards \$600M at...



The Numbers

Superman (2025) - Box Office and Financial...



Bollymoviereviewz

Superman Box Office Collection Day 35 India...



reddit

Superman 6th Week Worldwide Gross (\$595M)...

James Gunn's **Superman (2025)** has earned approximately **\$594.6 million worldwide** as of August 18, 2025, with a domestic (US/Canada) total of **\$340.9 million** and international gross of **\$253.7 million**. [imdb +2](#) [youtube](#)

LLM Capability +
Search tool +
Summarization Tool

Perplexity AI

Yes, **Perplexity AI** is a good example of an agentic system (though it presents itself as a search engine).

- It uses an LLM but augments it with **Search tool integrations**.
- The system decides when to call search APIs, when to summarize results, and how to cite them.
- That loop—LLM → decide action → fetch data → reason over it → respond—is agentic behavior.
- This is similar to what frameworks like LangChain or AutoGen let you build, but polished into a product.

Vibe coding

- Vibe coding refers to using natural language prompts to guide AI systems to generate working code.
- Developers shift from hand-coding to directing, reviewing, and refining via conversation.
- Karpathy's Take: He summed it up as:
 - *"I just see stuff, say stuff, run stuff, and copy paste stuff, and it mostly works."*
- Most of the vibe coding tools like Lovable, Cursor IDE, GitHub Copilot & Devin (Cognition AI) are Agentic AI Systems

Lovable

App Name: Ghar Ka Doctor ("Doctor at Home")

Concept Summary:

Ghar Ka Doctor is a mobile app designed to make basic healthcare advice and doctor consultations accessible and affordable for everyone, especially in Tier 2 and Tier 3 cities and underserved communities. By offering free or low-cost doctor consultations starting at Rs. 50, the app aims to remove the financial and geographical barriers to everyday medical guidance.

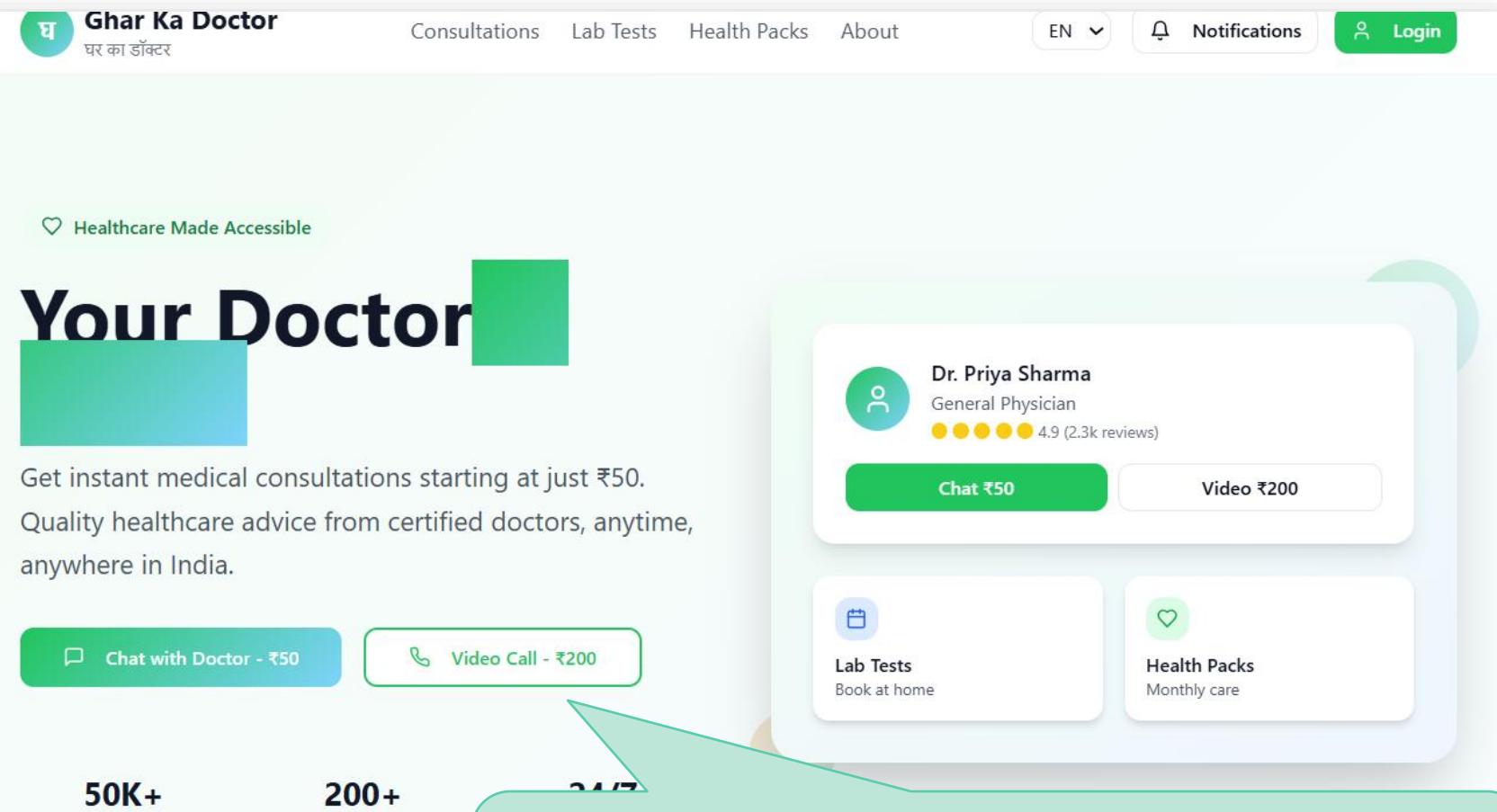
Key Objectives:

Democratize access to healthcare by making consultations affordable or free.

Offer proactive health tracking through subscription-based diagnostic services.

Build a healthcare ecosystem around primary consultations that leads to monetization via lab tests, pharmacy orders, referrals, and subscriptions.

Core Features:



The screenshot shows the Ghar Ka Doctor app's home screen. At the top, there is a navigation bar with the app's logo, "Ghar Ka Doctor" in English and Hindi, and links for "Consultations", "Lab Tests", "Health Packs", and "About". There are also buttons for "EN", "Notifications", and "Login". Below the navigation bar, a large green banner with the text "Healthcare Made Accessible" and a large "Your Doctor" heading. A call-to-action button "Chat with Doctor - ₹50" is visible. To the right, a profile card for "Dr. Priya Sharma, General Physician" with a 4.9 rating (2.3k reviews) and buttons for "Chat ₹50" and "Video ₹200". Below the profile card are two more service cards: "Lab Tests Book at home" and "Health Packs Monthly care". At the bottom of the screen, there are two large numbers: "50K+" and "200+", likely representing user counts or consultation counts. A green callout bubble points from the bottom right towards the "Health Packs" card.

a complex agentic AI system, because it exhibits planning, tool use (codegen, hosting, integrations), and autonomy in creating production-ready apps.

Lovable

- Lovable = the YC startup that lets you describe an app in natural language and then generates the whole application.
- It takes a natural language prompt → uses an LLM to generate **specs, frontend code, backend code**, and even sets up deployment.
- It doesn't just “answer your question”—it **autonomously coordinates multiple subtasks**: design, code generation, scaffolding, wiring, and refinement.
- That orchestration of multiple roles makes it a **complex agentic system**, not just an “AI code assistant.”

Examples of Complex Agentic AI Systems

- **Devin (by Cognition AI)**
 - Can plan, search docs, write code, debug, test, and deploy autonomously.
- **ChatGPT with Advanced Tools (OpenAI)**
 - When ChatGPT has browsing, code interpreter, and file upload enabled, it chooses when to invoke each.
 - That makes it an agentic AI system, because the LLM itself decides: “*Should I search the web? Should I run Python? Should I summarize a file?*”
- **Copilot for Microsoft 365**
 - Goes beyond text generation. It decides when to pull data from Outlook, Teams, Excel, or Word and coordinates across them.
- **Replit’s Ghostwriter / Cursor IDE Agents**
 - Not just code completion—they can run refactorings, tests, and multi-file updates by planning actions.



(+91) 81976 34433
(+91) 90190 30033

Agentic AI Frameworks

Agentic AI Frameworks

- Agentic AI frameworks = LLMs + tools + memory + planning + collaboration.
- They help build autonomous systems that can think, act, and work together like digital teams.
 - AutoGen (by Microsoft)
 - LangGraph (by LangChain)
 - CrewAI
 - MetaGPT
 - OpenAgents (by OpenAI)
 - SuperAgent
 - n8n

Frameworks

If you want to...

Use this

Build multi-agent pipelines easily

CrewAI, LangGraph, AutoGen

Simulate teams with roles

MetaGPT, ChatDev, AutoGen

Focus on tools and plugin use

LangChain, AutoGen, OpenAgents

Have visual workflow control

Autogen Studio, n8n, Superagent

Build agents that code

MetaGPT, Devin (AI Engineer)

Frameworks Comparison

If you want to...

Use this

Build multi-agent pipelines easily

CrewAI, LangGraph, AutoGen

Simulate teams with roles

MetaGPT, ChatDev, AutoGen

Focus on tools and plugin use

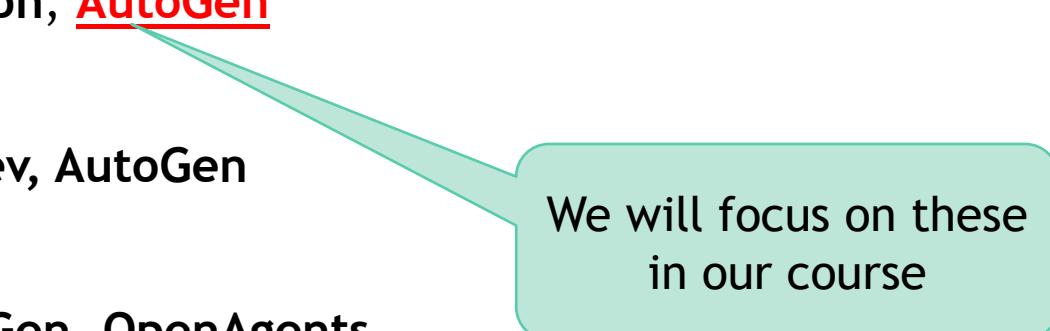
LangChain, AutoGen, OpenAgents

Have visual workflow control

Autogen Studio, n8n, Superagent

Build agents that code

MetaGPT, Devin (AI Engineer)



We will focus on these
in our course

AutoGen

- Designed for building LLM-driven agents that can collaborate through structured conversations and tool use. Strong planning + self-correction.
- Good at Multi-agent chat orchestration



CrewAI

- Lightweight and intuitive framework where agents take on different roles (e.g., researcher, writer) to complete complex tasks.
- Role-based multi-agent collaboration



n8n

- n8n (pronounced "n-eight-n") is a powerful open-source workflow automation tool that lets you connect apps, APIs, and services with no-code or low-code logic.
- Drag-and-drop workflow builderConnects to 350+ services (APIs, databases, CRMs, AI tools)

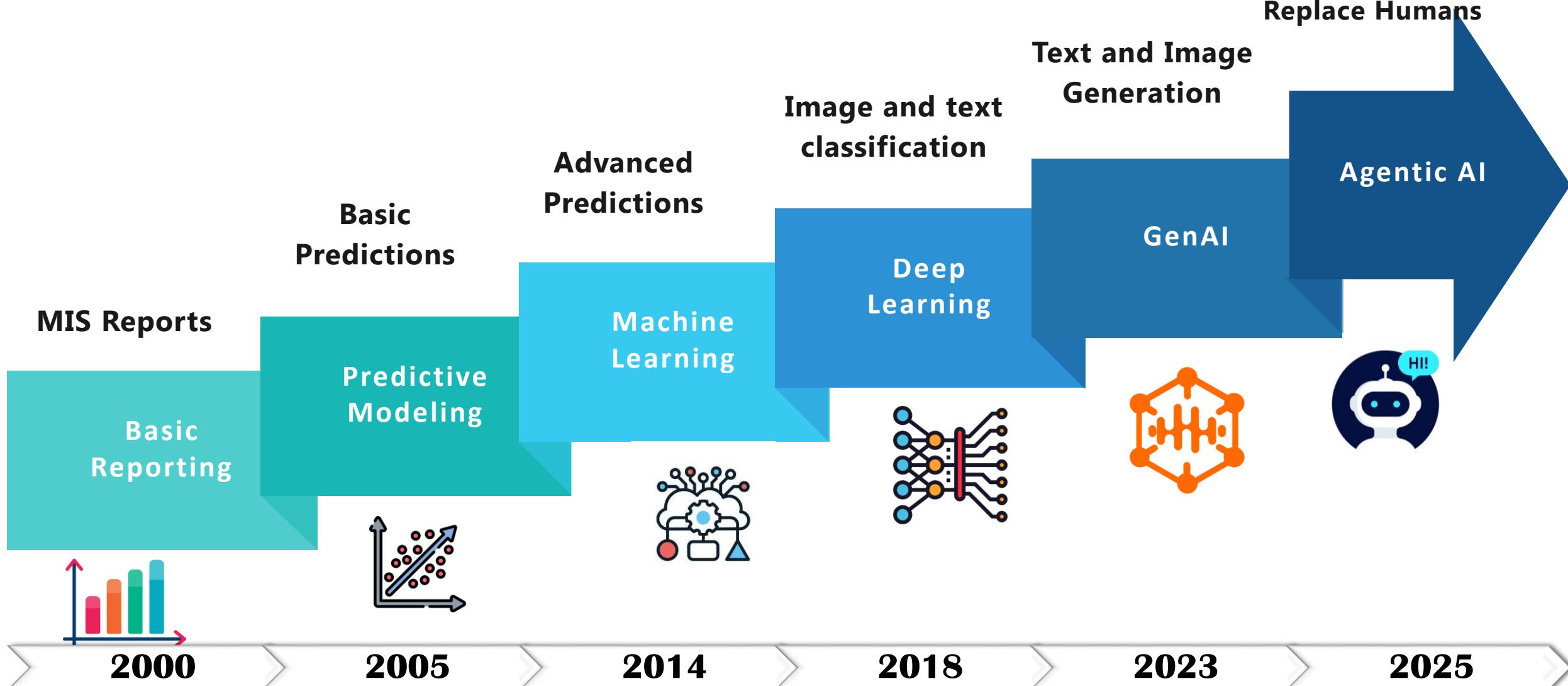




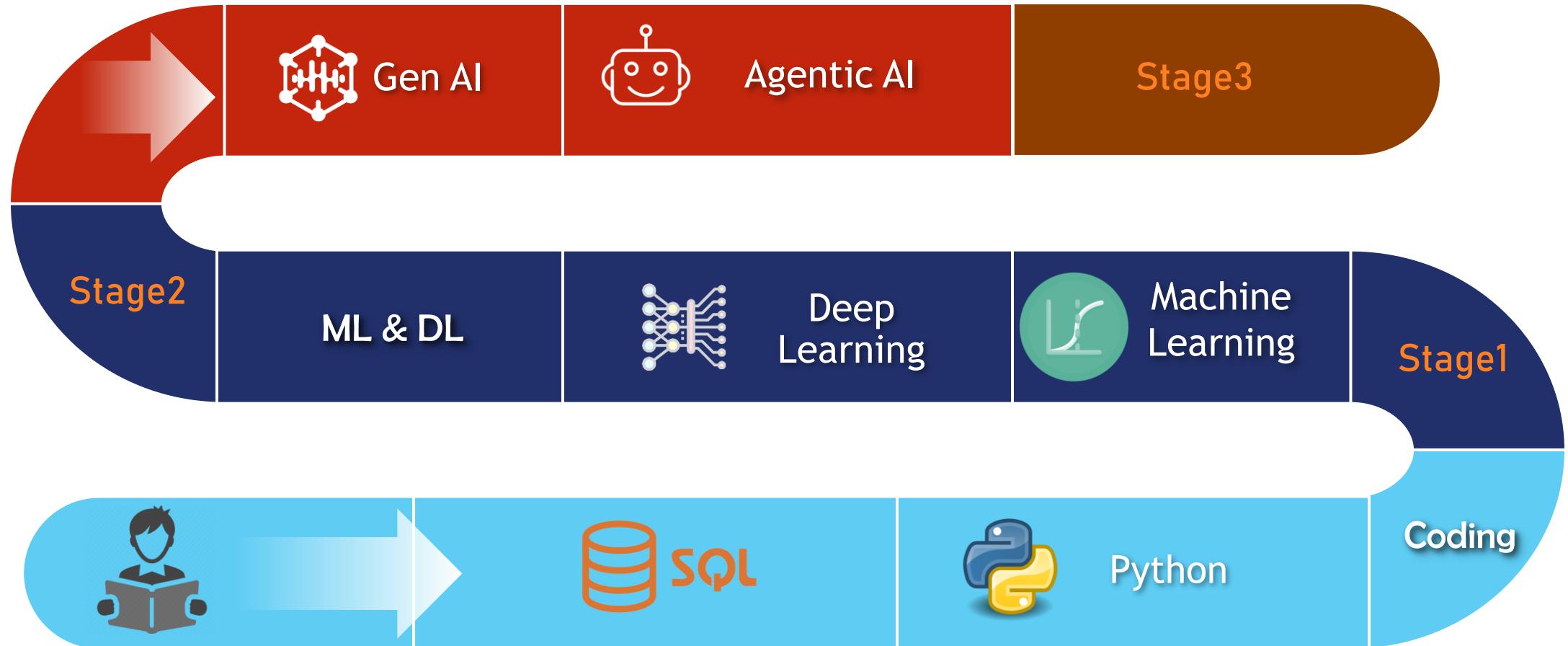
(+91) 81976 34433
(+91) 90190 30033

Learning Path

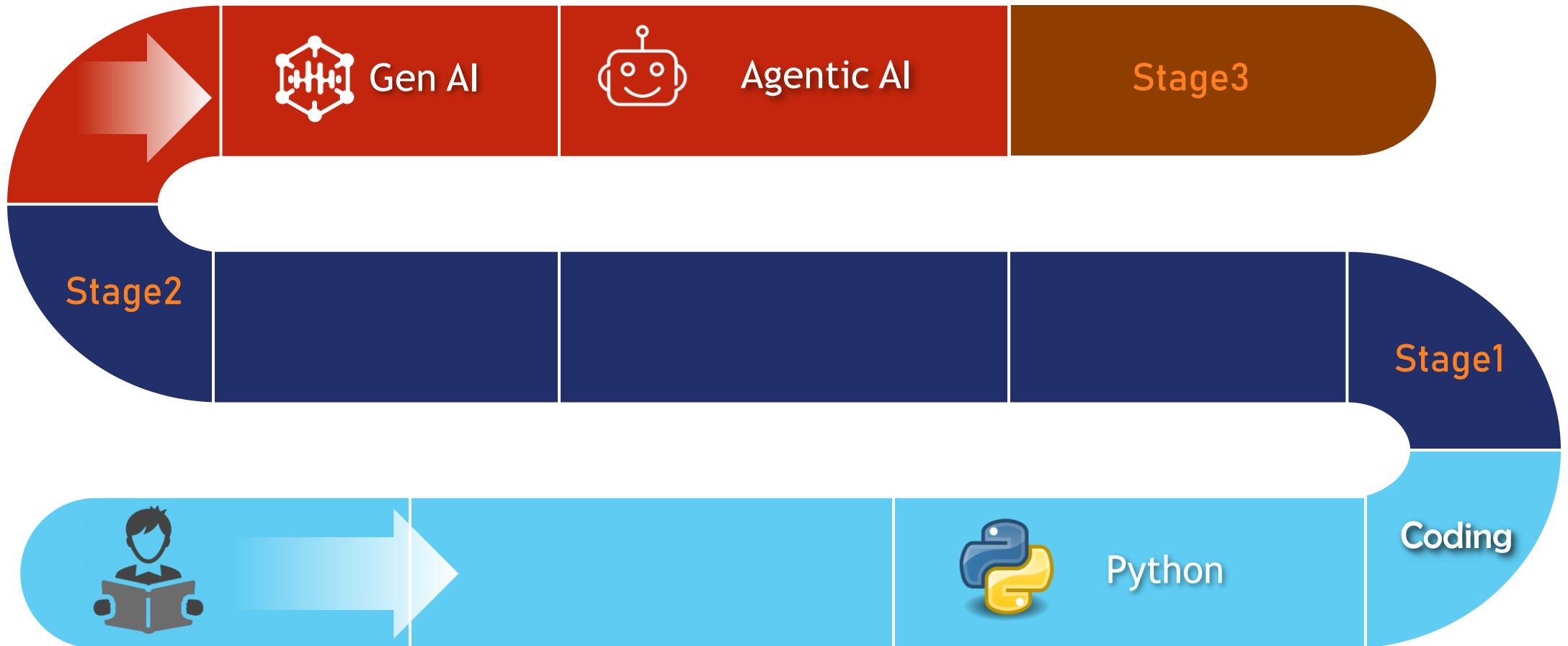
The history of Agentic AI



Learning Path – Full Fledged



Learning Path – The essentials



Course Contents

- Introduction
- LangChain Agents (Acts as recap of GenAI)
- AutoGen
- n8n Agents
- Agentic AI Applications
- MCP
- CrewAI



(+91) 81976 34433
(+91) 90190 30033

Assignment

1. Create an App in lovable for creating a social networking app only for lawyers
2. Create an App for planning a trip by taking number of days, origin, destination, accommodation and number of travellers as input.



(+91) 81976 34433
(+91) 90190 30033

Thank you
