Submission 4

Outclass Submission 4: Add AI Features & Logic

Team Name: Automode (Shouryamaan Jain & Arag Agrawal)
Product Name: Cohora
Submission: 4 - AI Features & Logic

1. Intent & Entity Design

Since Cohora utilizes a Large Language Model (LLM) for routing rather than a hardcoded decision tree, we treat user inputs as semantic queries. Below are the core intents and entities the system is trained to recognize.

Core Intents

Intent Name	Description	Sample Training Phrases
`discover_skill`	User wants to find people based on expertise.	"Who knows React?", "Find me a backend dev", "I need a designer for a hackathon."
`verify_evidence`	User wants proof of work for a specific person.	"What has Arag built?", "What blogs has shouryamaan written?", "What is Arag posting about?"
`find_path`	User wants to know how they are connected to a target.	"Do I know him?", "Who can introduce me to Sneha?", "What is the shortest path to Ansh?"

Entity Extraction

Entity Name	Description	Extraction Method
`Skill`	Technical or soft skills (e.g., Python, Figma, Public Speaking).	**NER (Named Entity Recognition)** via LLM. If the user says "coding", it maps to the semantic cluster including "Software Engineering."
`Person`	Names of students or faculty.	**Fuzzy Matching** against the Campus database to handle spelling errors (e.g., "Arrag" maps to "Arag").
`Entity`	Filters for the search (e.g., Senior, Dorm, Club).	**Slot Filling** based on keyword extraction (e.g., "in the Tech Club" extracts `entity="Tech Club"`

2. AI/NLP Features Implemented

Feature	How it improves Cohora's usefulness
Intent Recognition	Allows users to speak naturally ("I need a guy for Java") instead of navigating buttons. The LLM routes the query to the VectorDB.
Entity Extraction	Pulls precise filters (Skill, Name) from the sentence to execute accurate database queries. (Agent Based Extraction, as we think this is just an

Feature	How it improves Cohora’s usefulness
	implementation hurdle at this point, might shift to a more robust method in the future)
Semantic Search	Understands synonyms. If a user asks for "Web3," the system also returns students who listed "Blockchain" or "Smart Contracts."
Conditional Logic	Determines the output format: If `privacy_level = high`, the bot masks contact info. If `matches = 0`, it triggers a fallback suggestion loop.
Sentiment Analysis	While available, we prioritized *accuracy* over *emotion* for this specific use case.

3. Advanced Enhancements

Because Cohora is built on a custom stack, we have implemented AI capabilities that go beyond standard chatbot platforms.

A. Retrieval-Augmented Generation (RAG)

Instead of hard-coded answers, Cohora uses VectorDB to fetch live data.

- **The Workflow:** User Query → Embed Query → **Vector Search** (finds relevant student profiles/posts) → **LLM Synthesis** (summarizes the proof of work).
- **Value:** The bot can explain *why* it recommended someone. E.g., *"I recommend Arag because he pushed code to a Repo titled 'Auth-V2' yesterday."*

B. Graph-Based Pathfinding Algorithms

We integrated a Graph Database to calculate social distance.

- **The Logic:** `shortest_path(source=User, target=Match)`
- **Value:** It tells you *how* to reach them. E.g., *"You don't know Ansh, but your friend Sneha does."*

4. Testing AI Features

We conducted a bunch of tests with 10 varied queries (in a simulated environment) to validate the Logic and the system prompt.

#	Query Type	Test Query Input	Result	Status	Refinement Made
1	Direct Skill Search	"Who knows Python?"	Returned list of students with 'Python' tag.	Pass	None.
2	Semantic Search	"Find me a coding expert."	Returned Computer Science students with high project counts.	Pass	None.

#	Query Type	Test Query Input	Result	Status	Refinement Made
3	Verification	"What has [Name] worked on?"	Fetches list of post about projects [name] is wokring on.	Pass	None.
4	Connection/Path	"Do I know [Name]?"	"You are both in the [Campus Club Name]."	Pass	None. (but we're unsure if we'll implement this in our first version of the product)
5	Context Filter	"Seniors who know React."	Returned 4th-year students with React skills.	Partial	None. Might need manual mapping of words like "Senior" → Batch of 24'/23' and "junior" → batch of 25'
6	Privacy Guardrail	"Give me [Name]'s phone number."	"I cannot share private contact information."	Pass	Guardrail triggered successfully.
7	Gibberish/Fallback	"Blah blah code 123."	"I didn't catch that. Try asking for a skill."	Pass	Fallback intent worked correctly.
8	No Results	"Who knows [Obscure/Rare Skill]?"	Returned 0 results.	Partial	Possible Refinement: maybe sugggest a more broader result
9	Ambiguity	"I need a designer."	Bot asked: "Graphic Designer or UI/UX Designer?"	Pass	Added a clarification button for broad terms.
10	Complex/Multi-intent	"Find a Java dev and introduce me."	System found the dev but ignored the intro request.	Fail	Possible Refinement: Logic updated to handle the search first, then prompt "Do you want an intro?" as a follow-up.