Project Brief: "Living CV" AI Chatbot (7-Day MVP)

1. Project Overview & 7-Day Goal

Project: A conversational AI chatbot for college students to build their CVs.
Timeline: 7 days.
Primary Goal: A "working demo" that proves the core AI features. The product does not need to be perfect, but it must work. The focus is on data capture, jargon learning, and granular conversational validation.
What "Working" Means in 7 Days:

1. A user can send a message in natural language (e.g., "I interned at Swiggy as a PM").
2. The bot saves this to the correct database table (internship).
3. A user can *teach* the bot a new term (e.g., "A POR is a Position of Responsibility").
4. The bot uses that new term to correctly save data (e.g., "I got a new POR at ETMS" is saved to the por table).
5. **Crucially:** The bot must follow the **custom conversational scripts** defined in **Section 3c** to validate all entries. This means asking for the *type* of experience (e.g., Tech vs. Business) and then asking for the specific, granular fields for that type.

Out of Scope for 7-Day MVP:

- **Phase 4 (PDF Generation):** This feature is **discarded**.
- **Adaptive Coaching:** The bot does *not* need to learn what a "good" entry is. Its coaching is limited to validating required fields based on the scripts in Section 3c.

2. Core Philosophy & Non-Negotiable Constraints

1. **100% Data Privacy (Self-Hosted):** This is the most important rule. **No 3rd-party APIs** (OpenAI, Google, etc.). The entire stack must run locally.
2. **$0 Software Budget (Open-Source):** The entire stack must be built using Free and Open-Source Software (FOSS).
3. **Technology Stack:** The choice of FOSS stack (backend language, DB, AI framework) is up to the developer, as long as it meets the self-hosting and privacy constraints.

3. 7-Day MVP: Key Features & Deliverables

Phase 1: The "CV Vault" (Database) - (Required)

Goal: A robust database schema.
Deliverable: A database (e.g., SQLite) with all necessary tables.
Tables:

- education

- internship
- project
- por (Positions of Responsibility)
- competition
- certification
- skills
- jargon (Critical)

Phase 2: Conversational Data Extraction - (Required)

Goal: The bot must understand natural language and save data to the correct table.
Deliverable: A /process-entry API endpoint.
User Story:

1. **User says:** "I just finished a new project called 'CV Bot POC'..."
2. **System saves:** A new row in the project table.
3. **Bot replies:** "Successfully saved: Project"

Phase 3: Adaptive Learning & Validation

Phase 3a: Jargon Learning (RAG) - (Required)

Goal: The bot must learn college-specific terminology.
Deliverable: The /process-entry endpoint must first check if the user is defining a new term.
User Story:

1. **User says:** "A 'GC' means 'General Championship'."
2. **System saves:** This new entry into the jargon table.
3. **Bot replies:** "Got it! I've learned a new term: GC"
4. **Later, User says:** "I won the Ad Design GC."
5. **System does:** The backend first loads jargon from the DB, augments its prompt, and correctly understands "GC".
6. **System saves:** A new row in the competition table.

Phase 3b: Granular Conversational Validation - (Required)

Goal: The bot must run the correct "conversational script" (defined in Section 3c) to ensure all required fields are filled before saving.
Deliverable: A "stateful" conversational logic that can ask multiple follow-up questions.
User Story (Illustrating the new "Dispatcher" logic):

1. **User says:** "I want to add my new project."
2. **Bot recognizes intent:** add_project.
3. Bot replies (Dispatcher Question): "Sounds good. What type of project was it? Please enter a number:
   (1) Tech / Software
   (2) Research / Data Science
   (3) Business / Product
   (4) Core Engineering"

4. **User replies:** "1"
5. **System does:** The bot now triggers the **Tech/Software** script from Section 3c.
6. **Bot replies (Script Question 1):** "Okay, a Tech project. What was the project title and the dates? (e.g., 'Socialley Media App, [May 2022-Present]')"
7. **User replies:** ...
8. **Bot replies (Script Question 2):** "Great. Now for your first bullet point: What did you **Build**? (e.g., 'A social media web app')"
9. **User replies:** ...
10. **Bot replies (Script Question 3):** "What was the **Tech Stack**? (e.g., 'React.js, Firebase, and Tailwind CSS')"
11. **User replies:** ...
12. **Bot replies (Script Question 4):** "What was the main **Feature** or **Result**? (e.g., 'Users can create, edit, and save images...')"
13. **User replies:** ...
14. **System does:** The bot combines these into a single bullet point and saves the complete project entry.
15. **Bot replies:** "Saved. ... Do you want to add another bullet point for this project?"

3c. Schema Qualifying Criteria (The "Conversational Scripts")

This is the core logic for the Phase 3b validator. The bot must follow these scripts.

Script for: education

Bot must collect all four:

1. **Year** (e.g., 2027, 2021)
2. **Degree/Exam** (e.g., "M.TECH Dual Degree 5Y", "BOARD EXAMINATION")
3. **Institute** (e.g., "IIT Kharagpur", "HPBOSE")
4. **CGPA/Marks** (e.g., "8.38/10", "94.4%")

Script for: internship and project

*First, bot must ask for **Basic Fields**:*

- title (e.g., "Product Management Intern")
- company / organization / project_name
- dates (e.g., "[May' 25-Present]")

*Second, bot must ask for the **Experience Type** (Dispatcher):*

> "What type of experience was it?
> (1) Business / Product / Consulting
> (2) Tech / Software
> (3) Research / Data Science
> (4) Core Engineering / Design"

*Third, bot runs a loop for bullet points, using the **script for that type**:*

- **(Type 1) Business/Product Script:** (For each bullet point, ask for...)
  1. **Action:** "What was your main action or accomplishment?" (e.g., "Launched a co-pilot," "Applied bottom-up costing")
  2. **Methodology:** "How did you do it?" (e.g., "...through MVP feature scoping," "...to analyze market trends")
  3. **Impact (Quantifiable):** "What was the measurable result?" (e.g., "...reducing onboarding by 35%," "...boosting profitability")
- **(Type 2) Tech/Software Script:** (For each bullet point, ask for...)
  1. **Action/Product:** "What did you build, develop, or implement?" (e.g., "Developed a social media web app," "Created an agentic workflow")
  2. **Tech Stack:** "What languages, libraries, or frameworks did you use?" (e.g., "React.js, Firebase, HTML, Tailwind," "LLaMA-3.1-8B, FastAPI")
  3. **Feature/Result:** "What was the main feature or outcome?" (e.g., "Users can post, create, and save images," "dynamic query routing")
- **(Type 3) Research/Data Science Script:** (For each bullet point, ask for...)
  1. **Objective/Problem:** "What problem were you analyzing?" (e.g., "Analyzed supply-demand gaps," "Uncover patterns in the ag sector")
  2. **Model/Methodology:** "What models, algorithms, or tools did you use?" (e.g., "SVM algorithms, linear regression," "CAPM and Fama-French models")
  3. **Finding/Outcome:** "What was the key finding or result?" (e.g., "...develop a robust GTM strategy," "...tested for validity of assumptions")
- **(Type 4) Core Eng. Script:** (For each bullet point, ask for...)
  1. **Objective:** "What were you trying to design or analyze?" (e.g., "Design a Current Mirror Circuit," "Analyze transient frequency response")
  2. **Action:** "What was the main task you performed?" (e.g., "Analyzed an NPN transistor," "Cascoded the current mirror")
  3. **Tools Used:** "What software or tools did you use?" (e.g., "LT spice," "Tina-TI")

Script for: por (Positions of Responsibility)

Bot must collect all three:

1. **Role & Organization** (e.g., "Governor | Technology Aquatics Society
2. **Dates** (e.g., "[April'24 - Present]")

For each bullet point, bot must ask for:

1. **Key Action/Responsibility:** (e.g., "Organized 10+ events," "Successfully coordinated placement drive")
2. **Scale / Impact:** (e.g., "...managing a 4.5 lakh budget," "...impacting 2800+ students," "...achieving 100+ PPOs")

Script for: competition

Bot must collect all three:

1. **Event & Award:** (e.g., "Ad Design | General Championship | IIT Kharagpur (GOLD)")
2. **Date:** (e.g., "[March'25]")
3. **Description:** (Optional, but if given, follow the **Business/Product Script** for bullet points).

Script for: certification & skills

These are simple. Just collect the basic fields:

- **Certification:** title, organization, dates
- **Skills:** category (e.g., "Languages"), skill_list (e.g., "Python, C++")

4. Summary of 7-Day Deliverables

1. A single, well-commented backend codebase (main.py or similar) containing the complete API, including all logic for Phases 1, 2, and 3 (a & b).
2. A single index.html file (or similar simple frontend) that provides the chat interface.
3. A requirements.txt (or equivalent) for dependencies.
4. A README.md file explaining how to initialize the database and start the server.