Content

General Instructions

Dear Candidates,

We're excited about your interest in joining Nebula9.ai for your internship! Our company is at the forefront of applying generative AI and advanced technologies to solve real-world problems, and we're looking for talented individuals like you to contribute to our mission.

As part of the selection process, you have been assigned a project that showcases your skills, creativity, and ability to apply cutting-edge technologies. The project is designed to be completed within 4 days of focused work, but you will have until Saturday, July 20th, to submit your final work.

We have assigned a specific project to each candidate that best aligns with your chosen workstream, interests, and expertise. The complete list of assignments has been shared with your institution's Student Placement Coordinator (SPOC). Please reach out to your coordinator to learn which project has been assigned to you. The project descriptions for all potential assignments are included in this document for your reference.

If our project is shortlisted we will have the final round through a Video Conference, where we discuss the projects that you have done along with your approach/Technique.

We have assigned a project to each student and shared the list with your Institutions SPOC. Please reach out to your coordinator to know about the project that has been assigned to you.

Important Announcement

By design these are open-ended assignments wherein there will be no guidance and the candidates are expected to leverage public data or create data (if needed) to complete this project.

And in case of lack of clarity please take assumptions to solve the roadblock yourself. Candidates who have taken assumptions to solve a problem will have an edge over others during the final Interview Process.

The idea of this approach is to check the problem solving and decision making ability of the candidate. Also we want to evaluate how independently a candidate can execute projects since this represents the DNA of ownership and result orientation. Feel free to utilize any internet resources available to fulfill these assignments.

How you communicate during the final Interview (Presentations Skills and Project Representation) is also important.

Submission Procedure/Details:

Kindly submit your project using the Google Form shared with you by **Sunday, July 20th, by 10:00 PM**.

Include a link to your GitHub repository containing all code, documentation, and any deployment instructions or in case of non technical workload ensure that you attach the final output in the email. Ensure your repository is public or share a private link with access for our review team.

Machine learning / Generative AI Engineer Internship Assignment - Nebula9.ai

Dear Candidates,

We are thrilled about your interest in the Machine Learning / Gen AI internship position at Nebula9.ai. In this fast-evolving landscape of Generative AI, we aim to challenge your technical skills and creativity by providing assignments that reflect real-world problems and advanced AI technologies. Each assignment is designed to be completed within 4 days, so time management and focus will be key.

List of Assignments:

1. Enterprise Document Q&A system (RAG) with Multi-Modal Input

- **Objective:** Build a system capable of ingesting various document formats (PDFs, spreadsheets, emails) and allowing users to ask questions based on the content. It should support multi-modal input, such as extracting and analyzing both text and images within documents.
- **Key Requirements:**
    - Ingest various types of documents. Use an LLM for natural queries
    - Handle both structured(tables) and unstructured text.
    - Create a RAG chatbot and provide accurate responses with references to relevant sections

2. Graph-Enhanced & Agentic RAG for Corporate Intelligence

- **Objective:** Build a sophisticated Q&A system over a corpus of business documents (e.g., annual reports, SEC filings). The system must go beyond simple semantic search by understanding the relationships between entities.
- **Key Requirements:**
    - Ingest PDF documents. Use an LLM to extract not just text, but also key entities (e.g., people, companies, financial figures) and their relationships.
    - Store unstructured text in a vector database and the extracted structured relationships in a graph database (e.g., Neo4j).

- Create a RAG agent that, given a user query, decides whether to query the vector DB (for semantic questions), the graph DB (for factual/relational questions), or both, to synthesize a comprehensive answer.

3. Self-Correcting Multi-Agent Research Team with Verifiable Citations

- **Objective:** Create an autonomous team of AI agents that can research a complex topic and produce a detailed report with trustworthy, clickable citations.
- **Key Requirements:**
  - Design a multi-agent system (e.g., using LangGraph) with distinct roles: a "Planner," "Web Researchers," a "Fact-Checker/Critic," and a "Final Report Writer."
  - The Critic agent must validate claims made by the Researchers against the content of the source URLs.
  - The final report must embed verifiable citations that link directly to the specific text in the source that supports each claim.

4. Multi-Modal Storyboard and Video Scene Generator

- **Objective:** Build an end-to-end creative pipeline that transforms a simple story idea into a visual storyboard and short video scenes.
- **Key Requirements:**
  - Take a high-level text prompt (e.g., "a detective discovering a clue in a rainy city at night").
  - **Step 1 (Scripting):** Use an LLM to generate a detailed script with scene descriptions and character actions.
  - **Step 2 (Storyboarding):** For each scene in the script, use a text-to-image model (e.g., Midjourney, DALL-E 3) to generate a key storyboard image.
  - **Step 3 (Video Generation):** Use an image-to-video or text-to-video model (e.g., Stable Video Diffusion, Pika) to generate a short, 3-5 second video clip for each scene.

5. Generative Code Refactoring and Vulnerability Detection Assistant

- **Objective:** Develop a tool that acts as an AI pair programmer, analyzing existing codebases to suggest and apply improvements automatically.
- **Key Requirements:**
  - The tool should accept a GitHub repository URL as input.
  - Analyze the code to identify code smells (e.g., duplicate code, overly complex functions), performance bottlenecks, or common security vulnerabilities (e.g., OWASP Top 10).

- Use a powerful code generation model (e.g., GPT-4o, Gemini Code Assist, Llama 3) to suggest specific, refactored code patches and explain the reasoning behind the changes.

6. Automated Candidate Interview Agent

- **Objective:** Develop an AI agent that can conduct an initial screening interview with a candidate via a text-based chat interface. For candidates wishing to showcase advanced skills, implement a voice-based conversation mode.
- **Key Requirements:**
  - The agent first analyzes a candidate's resume and the job description.
  - It then dynamically generates relevant questions (technical, behavioral, situational).
  - The agent must engage in a conversational flow, understanding the candidate's live responses and asking intelligent follow-up questions.
  - At the end, it should produce a structured summary of the interview, evaluating the candidate's answers against the job requirements.

7. Real-Time Customer Support Agent Assist & Resolution Bot

- **Objective:** Build a system that supports a human contact center agent in real-time and can take over to resolve common issues automatically.
- **Key Requirements:**
  - The system must listen to a live audio stream of a customer call and perform real-time transcription.
  - In a private "Agent Assist" UI, it should display the transcript, identify the customer's intent, and proactively fetch and display relevant information from a knowledge base (real-time RAG).
  - For predefined, simple issues (e.g., "password reset"), the human agent should be able to trigger a fully automated AI voice agent to take over the call and guide the customer to resolution.

8. The Conversational Time Machine

- **Objective:** Create an interactive application where a user can have a voice-based conversation with a historical figure. The goal is to build a believable and engaging persona that can answer questions based on its historical context.
- **Key Requirements:**
  - Choose a Figure: Select a well-documented historical figure who has voice clippings available

- Knowledge Grounding: The AI's knowledge base should be grounded in fact. To make this achievable, you should use the figure's Wikipedia page, letters, and a list of their most famous quotes as the primary source of information. Use a simple RAG (Retrieval-Augmented Generation) approach by embedding this text data into a vector store for the AI to reference.
- Persona-Driven Chat: The LLM must adopt the persona of the historical figure, responding in their likely tone, style, and from their historical perspective. It should not have knowledge of events after its time.
- Voice Persona (The "Cloning" Aspect): This is a Text-to-Speech task. Implement a TTS engine voice cloning to match the historical figure's voice. If real recordings are unavailable, you can select a synthetic voice (e.g., from Google TTS, ElevenLabs, or other services) that matches the figure's likely age, gender, nationality, and gravitas. The quality of this choice is part of the evaluation.
- Interaction Flow: The user can type a question, and the application responds with the generated voice of the historical figure.

9. Personalized AI News Broadcast Generator

- **Objective:** Instead of just reading a pre-written article, this system will act as an autonomous news production studio. It will gather, write, and produce a complete, personalized video news broadcast from scratch.
- **Key Requirements:**
  - **Personalized Content Curation:** The user selects topics of interest (e.g., "Startups in India," "Telangana Politics," "Latest in GenAI"). An agent must then scour the web for the latest articles on these topics from reputable sources.
  - **AI Scriptwriting:** The system must summarize the gathered information and write a cohesive, balanced news script for a short (2-3 minute) broadcast.
  - **Automated Visual Generation:** For each story in the script, the AI agent must generate relevant visuals. This includes using text-to-image models to create illustrative graphics, charts, or pictures to display alongside the presenter.
  - **Voice and Avatar Synthesis:** Use a professional-sounding Text-to-Speech (TTS) voice to narrate the news script and sync this audio with a digital avatar (e.g., using APIs from D-ID, Synthesia, or an open-source alternative).
  - **Final Video Production:** The system must programmatically combine all elements—the talking head avatar, the generated voice-over, the on-screen visuals, and text headlines—into a single, downloadable MP4 video file that looks like a professional news segment.

10. The Proactive Work-Life Assistant

- **Objective:** Elevate the virtual assistant from a simple command-taker to a proactive, goal-oriented agent that can independently plan and execute complex, multi-step tasks by integrating with various real-world services.
- **Key Requirements:**
    - **High-Level Goal Interpretation:** The agent must accept a high-level goal in natural language, not just specific commands. For example: "Organize a celebratory team dinner for my 6-person team in Hyderabad next week. We want to go somewhere with great Hyderabadi biryani near our Gachibowli office."
    - **Autonomous Planning and Research:** The agent must break the goal into a logical plan. This includes using tools to search external services (e.g., Google Maps, Zomato) to find suitable options based on the user's constraints like cuisine, location, and reviews.
    - **Multi-Tool Coordination:** The agent must integrate with multiple tools to gather all necessary information. A key requirement is to integrate with a calendar service (e.g., Google Calendar API) to check the team's availability and find a common free slot.
    - **User Interaction and Action:** The agent should present 2-3 well-reasoned options to the user. Once the user makes a choice, the agent must take the final action, such as drafting and sending the calendar invitations to all team members.
    - **Web Automation (Advanced Bonus):** For candidates wishing to demonstrate advanced capabilities, implement a web automation component. After the user confirms the plan, the agent uses a framework like Selenium or Playwright, driven by the LLM, to navigate the chosen restaurant's website and attempt to complete the reservation form automatically.

Technologies to Focus On:

- **Python:** Core development language.
- **Generative AI Models:** Leverage cutting-edge models like GPT-4, Gemini, Falcon, and other open-source models for text and media generation.
- **Multi-Agent Systems:** Use frameworks like LangChain, AutoGen or custom multi-agent architectures to build collaborative AI systems.
- **AI Automation:** Implement automation tools and techniques to streamline workflows and decision-making processes.
- **Media Generation Models:** Explore tools like DALL-E or other open source models for image-to-video or text-to-video tasks.

Submission Guidelines:

Your submission should include a GitHub repository link containing all source code, a README file with setup instructions, and a brief overview of your project.

Please ensure your project is publicly accessible or share a private link with access permissions for our review team.

Please ensure the project you submit is solely your work and is not borrowed / picked up from internet sources.

Evaluation Criteria:

- **Creativity:** Innovative approaches to problem-solving and model use.
- **Technical Proficiency:** Effective use of AI models, frameworks, and programming techniques.
- **Functionality:** How well the solution meets the project requirements.
- **Time Management:** Ability to deliver a functional project within the 4-5 day timeframe.
- **Documentation:** Clear and thorough documentation of code and decisions.

FAQ for Internship Candidates:

Q1: Where can I find public data sources for the projects?

A: There are several reputable public data sources available for different types of projects. For textual data, consider sources like Kaggle, Google Dataset Search, and governmental databases (e.g., data.gov, europa.eu/data). For image datasets, look into ImageNet, Pexels, or Unsplash. Always ensure the data you use is open for public use and cite your sources.

Q2: What infrastructure can I use for developing and deploying my project?

A: You're encouraged to use free tiers of cloud providers such as Google Cloud Platform (GCP), Amazon Web Services (AWS), and Microsoft Azure for computational resources. For development and deployment, platforms like Heroku, Vercel, and Netlify offer free tiers that are sufficient for small-scale projects. Remember to monitor usage to stay within free tier limits.

Q3: How can I access LLMs or API keys for my project?

A: For projects requiring Large Language Models, consider using open-source models or platforms offering free tiers or developer API keys. Gemini Pro API, OpenAI's GPT-3 (with limited free credits for new users), and Hugging Face's APIs are good starting points. Ensure you register and review the documentation for API usage guidelines.

Q4: Can I use external libraries or frameworks not mentioned in the project descriptions?

A: Yes, you're encouraged to use any libraries or frameworks that you find suitable for completing your project, as long as they adhere to the project's objectives. Please document any external tools you use and provide reasoning for your choices in your project report.

Q5: What if my project exceeds the free tier limits on cloud platforms or APIs?

A: Plan your project to minimize resource usage and stay within free tier limits. If you anticipate exceeding these limits, consider simulating parts of your project or using mocked data to demonstrate your concept. Always communicate any limitations or assumptions in your project documentation.

Q6: How should I handle data privacy and security in my project?

A: Treat all data with respect to privacy and security best practices. If your project involves personal data, use anonymized or synthetic datasets. Ensure any application you develop follows security principles, like securing API keys and sensitive information.

Q7: What documentation is expected with the project submission?

A: Your submission should include a README file with an overview of the project, setup and running instructions, a brief discussion of the challenges faced and how you addressed them, and any assumptions made during the development. Additionally, include comments in your code for clarity.

Q8: Are collaborations with other candidates allowed?

A: While we encourage you to complete the project independently to showcase your individual skills, discussing ideas and sharing knowledge with peers can be beneficial. However, the final submission should be your work. Clearly state any collaborations or external help in your project documentation.

Q9: What happens if I encounter a problem I can't solve?

A: Researching and troubleshooting are key skills in software development and AI. Use resources like Stack Overflow, GitHub discussions, and official documentation for help. If an issue persists, document your problem-solving process and the hurdles encountered in your project report. This demonstrates your critical thinking and problem-solving skills, which are valuable to us.

We're looking forward to seeing your innovative solutions and how you tackle the challenges presented by these projects. This is your chance to shine and show us what you're capable of!

Best of luck,

Nebula9.ai