

Pitch Generator ai

AI system that generates startup pitch scripts



01 September 2025

Amay saxena (0827It1014)

Generative AI with AI/ML

**Abstract:**

PitchGenAI is an AI-powered system designed to generate professional and persuasive startup pitch scripts from minimal user input. This project leverages a local instruction-tuned language model, **Phi-3-mini-4k-instruct-q4**, enabling fully offline operation without relying on external APIs or internet connectivity. The system provides an intuitive interface using **Streamlit**, allowing users to input key startup details such as name, problem statement, solution, target market, business model, and preferred pitch tone. Based on these inputs, the AI produces structured, investor-ready pitches suitable for presentations, competitions, or assignments.

The main objective of PitchGenAI is to simplify and accelerate the process of creating effective startup pitches, particularly for students, entrepreneurs, and small teams with limited experience in business communication. The tool also includes features like auto-saving generated pitches with timestamps and downloading them in text format for future use. By combining natural language generation with a user-friendly interface, PitchGenAI demonstrates how AI can assist in creative business writing, reduce manual effort, and enhance the overall quality of startup communication. This project serves as an educational MVP showcasing the potential of local AI models in practical applications.

**Project Details:**

* **Objective:**

The objective of **PitchGenAI** is to develop an AI-powered system capable of generating structured, professional, and persuasive startup pitch scripts from minimal user input. This project aims to simplify and accelerate the process of creating investor-ready pitch content for students, entrepreneurs, and small teams, while operating fully offline using a local instruction-tuned AI model. By providing a user-friendly Streamlit interface, PitchGenAI allows users to input key startup details such as name, problem, solution, target market, business model, and preferred pitch tone, and generate polished pitches efficiently. The system also enables customization of pitch tone, auto-saving of generated pitches, and downloading them in text format for future reference. The ultimate goal is to demonstrate how AI can assist in business communication, reduce manual effort, and enhance the quality of startup presentations in an educational and practical MVP setting.

* **Methodology:**

1. **Tools & Frameworks**
   * **Python 3.10** → Core programming language for implementation.
   * **Streamlit**: Provides a user-friendly web interface for inputting startup details and generating pitch scripts.
   * **GPT4All Python Package**: Enables interaction with local large language models for text generation.
   * **Text Editors / IDE**: VS Code or PyCharm for coding and testing.
2. **Models Used**
   * **Phi-3-mini-4k-instruct-q4.gguf**: A lightweight instruction-tuned large language model optimized for generating coherent and contextually accurate text. The model is hosted locally, ensuring offline functionality and eliminating the need for cloud-based API calls.
3. **Approach**
   * **Input Collection**: Users provide key startup information including name, problem statement, solution, target market, business model, and preferred pitch tone via the Streamlit interface.
   * **Prompt Construction**: The input data is formatted into a structured prompt using predefined templates to guide the model’s generation.
   * **Text Generation**: The GPT4All model processes the prompt and generates a coherent, persuasive pitch script based on the given inputs. Parameters like max\_tokens and pitch tone are configured to control output length and style.
   * **Output Handling**: The generated pitch is displayed in the Streamlit interface, with options to auto-save and download as a text file for offline use.
   * **Testing & Iteration**: Test cases ensure the generator produces complete, readable pitches and the system handles edge cases such as missing or ambiguous input.

* **Implementation:**

**Environment Setup:**

* Install Python 3.10+ and create a virtual environment.
* Install required packages via requirements.txt including Streamlit and GPT4All.

**Model Integration:**

* Place the local model Phi-3-mini-4k-instruct-q4.gguf in the models/ folder.
* Load the model in Python using GPT4All’s LLModel class.

**Input Interface Design:**

* Use Streamlit to create input fields for startup name, problem, solution, target market, business model, and pitch tone.
* Implement real-time input validation for completeness and correctness.

**Text Generation:**

* Pass the constructed prompt to the local GPT4All model.
* Generate the pitch text with configurable parameters such as maximum tokens and pitch tone.

**Output Handling:**

* Display generated pitch on the Streamlit interface.
* Implement functionality to save the output to the outputs/ folder with timestamped filenames.
* Allow downloading the generated pitch as a .txt file.

**Testing & Optimization:**

* Create test scripts to verify pitch generation for different input combinations.
* Ensure proper handling of edge cases like missing or ambiguous inputs.

**Conclusion:**

* **What did you learned?**

The PitchGenAI project successfully demonstrates how a local AI system can generate professional, structured startup pitch scripts from minimal user input. By leveraging a lightweight instruction-tuned model (**Phi-3-mini-4k-instruct-q4.gguf**) and a simple Streamlit interface, the system operates fully offline while producing persuasive, coherent pitches.

**Key Learnings:**

* Gained hands-on experience in integrating **local large language models** into Python applications.
* Learned how to construct effective **prompts** to guide AI output for specific use cases.
* Understood the importance of **user input validation** and structured templates in generating reliable outputs.
* Explored techniques for **offline AI deployment**, avoiding dependence on cloud services.
* Improved skills in Python, Streamlit, and basic project organization for AI-based MVPs.
* **Future improvements:**
* **Support for Multiple Languages:** Expand pitch generation to support languages beyond English, enabling wider accessibility.
* **Enhanced Customization:** Allow users to select additional pitch parameters, such as style, length, and focus (e.g., financial, technical, marketing).
* **Integration with Presentation Tools:** Automatically convert generated pitches into slides or PDF formats for investor-ready presentations.
* **Advanced Model Options:** Incorporate larger or fine-tuned models for more nuanced and creative pitch generation.
* **Collaboration Features:** Enable multiple users to co-create and edit pitches in real-time.
* **Feedback Loop:** Integrate a feedback system where users can rate generated pitches, allowing iterative improvement of outputs.
* **Template Library Expansion:** Include industry-specific templates to generate pitches tailored for different sectors, such as healthcare, fintech, or renewable energy.
* **Mobile Support:** Develop a lightweight mobile interface for pitch generation on-the-go.

**References:**

* **Frameworks & Tools:**
* Python 3.10+ — Core programming language for the project.
* Streamlit — For building the interactive web interface.
* GPT4All Python Package — For interacting with local large language models.
* **Models:**
* Phi-3-mini-4k-instruct-q4.gguf — Instruction-tuned local LLM used for text generation.
* **Prompting Techniques & Research References:**
* Nomic AI GPT4All Documentation — Guidelines on using and integrating local LLMs.
* **Additional Resources:**
* OpenAI Research on GPT-based Text Generation for Structured Outputs.
* StackOverflow & GitHub community forums for troubleshooting Python and Streamlit integration.

**GitHub:** <https://github.com/amaysaxena02/PitchGenAI>