OpenAI + servicenow

SNGPTBOT

By: Noah Samuel Markus and Tyler Distefano

GitHub Repository https://github.com/therealNoahSamuelMarkus/SNGPTBOT

Project Summary

SNGPTBOT is an AI-driven IT Service Desk assistant designed to streamline internal support operations within companies. Built using OpenAI's APIs, ServiceNow integrations, and deployed via a Streamlit web interface, it allows employees to:

- Ask company policy questions by pulling from the ServiceNow Knowledge Base (KB)
- Submit and track IT service incidents through ServiceNow
- Perform password resets.

The long-term vision for SNGPTBOT is deployment as a mobile app through Mobile Device Management (MDM) platforms, allowing credential auto-login via company-managed smartphones. However, for this course project (EGR 404), the app functions via a **Streamlit website** where users **manually input** their credentials.

Our motivation was based on real-world problems we observed:

- Many users submit support tickets for questions easily answered in the Knowledge Base.

- Users often encounter login problems and cannot easily reset their passwords.

- Tickets often have vague or incomplete short descriptions, leading to operational inefficiencies.
  We recognized that integrating GPT technology with ServiceNow could significantly reduce ticket volume, improve ticket quality, and streamline password recovery.

Methodology

The solution architecture included four main components:

1. **Natural Language Processing (NLP) with OpenAI GPT API**:
   We used OpenAI's API to interpret natural language queries and generate appropriate responses.

2. **ServiceNow API Integration**:
   The bot can search ServiceNow KB articles, submit incident tickets, and trigger password reset workflows via ServiceNow REST APIs.

3. **Streamlit Frontend**:
   Streamlit provided a lightweight, rapid development platform for the web UI, ensuring ease of deployment for the class project.

4. **Manual User Credential Input**:
   Since MDM credential pulling was out-of-scope, users manually enter their username and password at session start for authentication.

The core logic flow is:

- **Knowledge Base Search**: When a user asks a company-specific question, the app queries ServiceNow's Knowledge Base. If relevant articles are found, the answer is based on article content.

- **Incident Creation**: If the user reports an IT issue, the app helps create a ServiceNow incident with a properly structured and detailed description.

- **Password Reset Flow**: If a user mentions password problems, the app offers the option to trigger a reset request through ServiceNow.

Tools Used

- **OpenAI API (GPT-4-mini)**: For interpreting and responding to user inputs.
- **ServiceNow REST API**: For ticket creation, password reset, and KB search functionalities.
- **Python**: Core backend logic for API handling and UI interactions.
- **Streamlit**: To build the web-based front-end interface.
- **dotenv**: For managing environment variables securely.

Project Results

The final deliverable successfully demonstrated:

- Seamless integration with ServiceNow's Knowledge Base and incident management system.
- Correct submission of ServiceNow incidents with a GPT-generated detailed short description.
- Password reset initiation functionality for users experiencing account lockouts.
- Basic user authentication (manual for now).
- A clean, usable Streamlit web app interface for interaction.

Limitations:

- **No automatic credential detection**: Users manually enter credentials.
- **No mobile app yet**: It remains a Streamlit website, not an MDM-deployed app.
- **Single Knowledge Base**: Only queries articles in one designated ServiceNow KB.

Despite these limitations, **core functionality works** reliably and fulfills the goals we set for the project scope.

Design Choices and Challenges

Design Choices:

- **Streamlit for MVP**: Streamlit was ideal for rapid prototyping due to its minimal setup requirements and seamless UI components.
- **OpenAI for interpretation**: GPT’s ability to parse user intent (e.g., "forgot my password" vs. "need new laptop") made it well-suited for dynamic IT support.
- **ServiceNow integration first**: Since our real-world pain points involved ServiceNow, integration with it was prioritized over adding multi-platform support.

Challenges Encountered:

- **Authentication Complexity**: ServiceNow authentication required careful token management, and manual credential entry had to be implemented securely.
- **Parsing User Intent**: Although GPT is powerful, fine-tuning prompt design was crucial to reliably distinguish between questions needing KB search, incident creation, or password reset.
- **Streamlit UI Limitations**: Certain UI features (e.g., pop-ups, multi-page flows) were harder to implement compared to a traditional mobile app framework like Flutter or React Native.

Future Work

- **Mobile App Version (Post-Project Goal)**: Build a native iOS/Android app with Single Sign-On (SSO) using MDM credentials.

- **Access Level Management**: Adjust bot responses based on user's role and access level fetched dynamically from ServiceNow.

- **Proactive Ticket Suggestions**: Recommend tickets for common issues (like VPN setup, software downloads) based on detected intents.

- **Advanced Logging**: Integrate detailed logging and monitoring into ServiceNow dashboards for IT admins to track bot usage and impact.