# Incident Management AI Chatbot

## 1. Project Overview

### 1.1 Project Description

- \*\*Name\*\*: Incident Management AI Chatbot

- \*\*Purpose\*\*: Intelligent IT incident classification and response generation

- \*\*Key Features\*\*:

- Automatic incident status prediction

- AI-powered incident response generation

- Transformer-based machine learning model

- Streamlit web interface

### 1.2 Technology Stack

- \*\*Core Libraries\*\*:

- PyTorch

- Transformers (Hugging Face)

- LangChain

- Streamlit

- \*\*Machine Learning\*\*:

- BERT-based sequence classification

- Text generation pipeline

- \*\*Data Processing\*\*:

- Pandas

- scikit-learn

## 2. System Architecture

### 2.1 Component Diagram

A[Incident Text Input] --> B[Tokenization]

B --> C[Incident Classification Model]

C --> D[Status Prediction]

D --> E[LangChain Response Generator]

E --> F[Streamlit UI]

### 2.2 Key Components

- \*\*Incident Classification Model\*\*

- Uses transformer-based architecture

- Fine-tuned on incident management dataset

- Predicts incident status

- \*\*Response Generation\*\*

- Uses GPT-2 text generation pipeline

- Generates contextual responses based on incident details

- \*\*User Interface\*\*

- Streamlit web application

- Real-time interaction

- Responsive design

## 3. Model Training Process

### 3.1 Data Preparation

- \*\*Data Source\*\*: CSV file with incident details

- \*\*Text Features\*\*:

- Issue Summary

- Department

- Category

- \*\*Label Encoding\*\*:

- Converts categorical status to numerical labels

- Uses scikit-learn's LabelEncoder

### 3.2 Training Pipeline

- \*\*Model\*\*: BERT-base-uncased

- \*\*Training Parameters\*\*:

- Epochs: 10

- Batch Size: 4

- Learning Rate: 2e-5

- \*\*Optimization\*\*:

- AdamW Optimizer

- Linear Learning Rate Scheduler

- Gradient Clipping

### 3.3 Model Evaluation

- \*\*Metrics\*\*:

- Training Loss

- Validation Loss

- Validation Accuracy

## 4. Installation Guide

### 4.1 Prerequisites

- Python 3.8+

- CUDA (recommended for GPU acceleration)

### 4.2 Dependencies

# Install dependencies

pip install -r requirements.txt

### 4.3 Required Dependencies

- langchain\_openai

- langchain\_core

- python-dotenv

- streamlit

- torch

- transformers

- scikit-learn

- pandas

- langchain

- accelerate

## 5. Configuration

### 5.1 Environment Setup

- Set up `.env` file for sensitive configurations

- Configure model paths

- Set up Hugging Face authentication if required

### 5.2 Model Configuration

python

model\_config = {

'model\_name': 'bert-base-uncased',

'max\_length': 128,

'num\_labels': 4, # Based on incident statuses

'learning\_rate': 2e-5

}

## 6. Deployment

### 6.1 Local Deployment

# Run Streamlit application

streamlit run incidentchatbot.py

### 6.2 Deployment Options

- Local server

- Cloud platforms (AWS, GCP)

- Docker containerization

## 7. Usage Example

### 7.1 Incident Input

python

incident\_text = "Network connectivity issues in IT department"

predicted\_status = predict\_incident\_status(

model, tokenizer, incident\_text, label\_encoder

)

## 8. Limitations and Considerations

### 8.1 Model Constraints

- Dependent on training data quality

- Limited to predefined incident categories

- Potential bias in response generation

### 8.2 Performance Factors

- Model accuracy varies with training data

- Computational resources impact inference speed

- Requires periodic retraining

## 9. Future Roadmap

- Expand incident categories

- Implement more advanced response generation

- Add multi-language support

- Enhance model interpretability

## 10. Ethical Considerations

- Ensure bias-free incident classification

- Maintain data privacy

- Provide transparent AI-generated responses

## 11. Troubleshooting

### 11.1 Common Issues

- CUDA out of memory

- Model loading failures

- Tokenization errors

### 11.2 Debugging Strategies

- Check GPU compatibility

- Verify model and tokenizer paths

- Monitor system resources

## 12. Contributing Guidelines

- Fork the repository

- Create feature branches

- Submit pull requests

- Follow PEP 8 style guide