**Chapter 1: Introduction**

**1.1 Problem Statement: PS ID – SIH1705**

**Overview:** The proposed solution is an AI-powered employee chatbot tailored for a large public sector organization. Its primary objective is to provide accurate and timely responses to employee queries related to HR policies, IT support, project status, holidays, company events, and other organizational matters. The chatbot is designed to enhance internal communication and reduce the load on support teams.

Key components of the chatbot include:

* Natural Language Processing (NLP) for intent recognition
* Deep learning-based intent classification
* Parallel handling of at least 5 users with response times under 5 seconds
* Two-Factor Authentication (2FA) via email
* Bad language filtering based on a predefined dictionary
* Document analysis capabilities for summarizing and extracting keywords from uploaded documents

A demonstration video has been provided on YouTube for understanding the scope: <https://youtu.be/Q3pP7mRk5Qk>

**Why We Chose This Problem Statement:** Efficient communication is critical in any large organization. Employees often struggle to find quick answers to administrative or IT-related queries. This chatbot eliminates unnecessary delays, promotes automation, and ensures smoother HR and IT operations. The added feature of document summarization further improves organizational productivity.

**Project Objectives**

* To build an intelligent chatbot capable of understanding and responding to real-life employee queries across departments.
* To implement a scalable architecture that supports a minimum of 5 concurrent users.
* To enable secure authentication through 2FA.
* To develop document summarization and keyword extraction features for employee-uploaded documents.
* To ensure chatbot security through offensive language filtering.
* To train the system using publicly available datasets and simulated HR/IT data.
* To use only free and open-source technologies.

**Applications**

* **HR Department**: Answer queries on leaves, salaries, training, promotions, and holidays.
* **IT Support**: Provide assistance on login issues, software installations, or connectivity problems.
* **General Admin**: Deliver updates on company policies and events.
* **Employees**: Quick self-service assistance for project status, work-from-home rules, etc.
* **Document Processing**: Allow staff to upload files for keyword extraction and summarization.
* **Security & Compliance**: Restrict offensive language using dictionary-based filtering.

**Chapter 2: Exploratory Data Analysis**

**2.1 Dataset Overview**

* **Format**: JSON file with patterns, intents, and responses
* **Fields**: patterns, responses, intent
* **Purpose**: Used to train the chatbot to identify user queries and map them to predefined answers.

**2.2 Key EDA Insights**

* **Intent Frequency Analysis**: Identified most frequent query types (e.g., leave\_policy, salary\_info, project\_query).
* **Word Cloud Generation**: Highlighted common keywords such as *leave*, *salary*, *holiday*, *training*.
* **Intent Length Distribution**: Most user queries range between 5–12 words, optimal for BERT-based input processing.
* **Balanced Data Check**: Ensured all intents had a reasonable number of examples to prevent training bias.

**2.3 Tools Used**

* **Pandas**: For data manipulation
* **Matplotlib/Seaborn**: For plotting distribution of intents
* **NLTK/Regex**: For text tokenization and cleaning
* **WordCloud**: For visualizing keyword frequencies

**Chapter 3: Methodology**

**3.1 Hardware and Software Requirements**

|  |  |
| --- | --- |
| **Component** | **Specification** |
| Processor | Intel i5 or AMD Ryzen 5 |
| RAM | 8 GB |
| Storage | 256 GB SSD |
| OS | Windows 11 |
| Backend | Python (Flask) |
| Frontend | HTML, CSS, JavaScript |
| Libraries | TensorFlow, scikit-learn, NLTK |
| Editor | VS Code |
| Dataset | jioemployee\_dataset.json |

**3.2 System Design Overview**

* **Frontend**: Web-based interface for employee interactions.
* **Backend (Flask)**: Handles logic, API calls, and connects to the NLP model.
* **NLP Engine**: Trained using custom JSON data; intent classifier + response selector.
* **Document Processor**: Summarizes and extracts key info from PDFs/text files.
* **Security Layer**: Includes email-based 2FA and bad language filtering.
* **Scalability**: Supports multiple users with optimized async processing.

**3.3 Algorithms**

* **Text Preprocessing**: Stopword removal, tokenization, stemming
* **Intent Classification**: LSTM/BERT model for predicting intents
* **Document Summarization**: TextRank/BERT Summarizer
* **Keyword Extraction**: TF-IDF based feature selection
* **2FA**: SMTP-based email verification module
* **Language Filter**: Pattern matching against a predefined offensive words list

**3.4 Outcome of Module 1**

* Dataset prepared and cleaned for model training.
* Intent frequency and keyword patterns identified.
* Frontend interface connected to backend using Flask APIs.
* Foundational insights used for building the scalable chatbot model with NLP and deep learning.