



NORTH SOUTH UNIVERSITY

ELECTRICAL AND COMPUTER ENGINEERING

Project Name: Development of a Locally-Run AI Chatbot for University Information Retrieval

Title: NSUChat (An AI-Powered Virtual Assistant)

PROJECT PROPOSAL SUBMITTED BY,

ZAWED BIN TARIQ	ID: 2232811642
NAZMUS SHAKIB	ID: 2312009042
SAZZAD HOSSAIN SHAWON	ID: 2211415042

Course: CSE299 (Junior Design)

Section : 5

Group no: 2

Summer 2025

1. **Introduction:** The rapid advancement of artificial intelligence (AI) has transformed how information is accessed and processed. Large language models like ChatGPT have demonstrated the potential of AI to provide human-like responses to a wide range of queries. However, such models often rely on cloud-based infrastructure, raising concerns about data privacy, accessibility in low-connectivity environments, computational costs and often they are not user friendly as well. To address these challenges, this project proposes the development of a locally-run AI chatbot, designed to operate on a standard computer, using data sourced from university websites and course materials. The chatbot will be implemented using Python, leveraging open-source machine learning libraries to create a privacy-focused, efficient, and accessible tool for university students and faculty.
2. **Problem Statement:** University students and faculty members often face challenges in accessing relevant information from university websites and course materials due to scattered resources, complex navigation, and lack of real-time query resolution. Existing AI solutions like ChatGPT require internet connectivity and raise privacy concerns when handling sensitive academic data. There is a need for a localized AI system that can operate offline, provide accurate responses based on university-specific data, and ensure data privacy by keeping all processing on the user's local machine.
3. **Objectives:** The primary objectives of this project are:
 - To design and implement a locally-run AI chatbot using Python that can process natural language queries related to university information and course content.
 - To collect and preprocess data from university websites and course materials to create a domain-specific dataset for training the AI model.
 - To develop a lightweight machine learning model suitable for running on standard personal computers without requiring high-end hardware.
 - To ensure data privacy by performing all computations locally, eliminating the need for cloud-based processing.
 - To evaluate the chatbot's performance in terms of response accuracy, speed, and user satisfaction.
 - To make our chatbot as user friendly as possible.
4. **Scope:** The project focuses on building an AI chatbot that operates on a local computer, trained on publicly available data from university websites (e.g., departmental pages, academic policies, event schedules) and course materials (e.g., syllabi, lecture notes, FAQs). The system will support natural language queries in English and provide responses relevant to the university context. The project will not involve proprietary datasets, or advanced hardware like GPUs, ensuring accessibility for typical users.

5. Literature Review: Recent progress in natural language processing (NLP) has simplified the creation of AI systems that can understand and respond to human questions in a conversational way. Scikit-learn, a popular Python library, provides user-friendly tools for building machine learning models that process text, such as classifying questions, extracting key information, or grouping similar content. Advanced models like BERT and GPT are highly effective at understanding and generating human-like text, but they require powerful computers and often depend on cloud-based servers, which can be a challenge for local use. In contrast, Scikit-learn's simpler

algorithms, like those for text classification or clustering, are lightweight and can run efficiently on standard personal computers without needing specialized hardware. By using Google Search, we identified valuable open-source resources, including tutorials, datasets, and tools like Hugging Face's Transformers, which offer practical guidance for developing chatbots tailored to specific purposes. This project leverages Scikit-learn to build a chatbot that operates locally on a standard computer, using data from university websites and course materials, with Google Search helping to locate additional tools and references to support development.

6. Methodology: The project will follow a structured methodology to achieve the stated objectives:

6.1 Data Collection

- **Source:** Data will be collected from the university website (e.g., academic pages, faculty profiles, event calendars) and course materials (e.g., syllabi, lecture slides, FAQs).
- **Method:** Web scraping using Python libraries like BeautifulSoup and Scrapy to extract text data. Course materials will be manually curated to ensure relevance and accuracy.
- **Ethics:** Only publicly available data will be used, adhering to the university's data usage policies.

6.2 Data Preprocessing

- Clean and preprocess the collected data to remove noise (e.g., HTML tags, irrelevant text).
- Tokenize and structure the data into a format suitable for training (e.g., question answer pairs, text chunks).
- Create a dataset of approximately 10,000-20,000 text entries, depending on the volume of available data.

6.3 Model Selection and Training

- **Model:** Use a lightweight transformer model from the Hugging Face Transformers library or Scikit learn optimized for local execution.
- **Training:** Fine-tune the model on the university-specific dataset using supervised learning for question-answering tasks.
- **Tools:** Python, PyTorch, Hugging Face Transformers, and NLTK for text processing.
- **Hardware:** Training will be performed on a standard laptop (e.g., 16GB RAM, Intel i3 processor) to ensure compatibility with typical user systems.

6.4 Chatbot Implementation

- Develop a command-line or GUI-based interface using Python libraries like Tkinter or Flask for user interaction.
- Integrate the trained model to process natural language queries and generate responses.
- Implement a retrieval-augmented generation (RAG) approach to combine the model's knowledge with direct access to the dataset for improved accuracy.

6.5 Evaluation

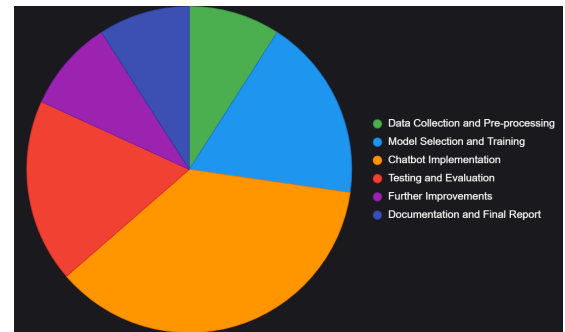
- **Metrics:** Evaluate the chatbot using accuracy (correctness of responses), response time, and user satisfaction (via surveys).
- **Testing:** Conduct user testing with 10-15 students and faculty, collecting feedback on usability and performance.
- **Benchmarks:** Compare performance with a baseline model (e.g., untuned Distil BERT) to quantify improvements from fine-tuning.

7. Expected Outcomes:

- A fully functional AI chatbot that runs locally on a standard computer, capable of answering university-related queries with at least 80% accuracy.
- A documented dataset of university and course information, reusable for future projects.
- A user-friendly interface for interacting with the chatbot.
- A technical report detailing the model's performance, limitations, and future improvements.

8. Timeline:

	Phase	Duration
1.	Data Collection and Pre-processing	1 week
2.	Model Selection and Training	2 weeks
3.	Chatbot Implementation	4 weeks
4.	Testing and Evaluation	2 weeks
5.	Further Improvements	1 week
5.	Documentation and Final report	1 week
	Total	11 week



9. Resources:

- Hardware: Personal computers with at least 16GB RAM and Intel i3 processors.
- Software:
 - Python3.8+,
 - PyTorch,
 - Hugging Face Transformers, Scikit Tools, Tokenizers
 - BeautifulSoup, Scrapy, Tkinter/Flask, Scikit learn
 - PyCharm Community Edition, Visual Studio Code
- Data: University website and course materials (publicly available).
- Human Resources: Project team of 3 CSE students, supervised by a faculty advisor.

10. Budget Estimation:

- The project will utilize open-source tools and publicly available data, minimizing costs. Estimated expenses include:
 - Personal Computer
 - Hardware Support
 - Printing and Documentation

11. Conclusion: This project aims to develop a locally-run AI chatbot that enhances access to university information while prioritizing data privacy and accessibility. By leveraging Python and open-source NLP tools, the project will deliver a practical solution for students and faculty, with potential applications in other educational institutions. The proposed methodology ensures a feasible implementation within the given timeframe and resources