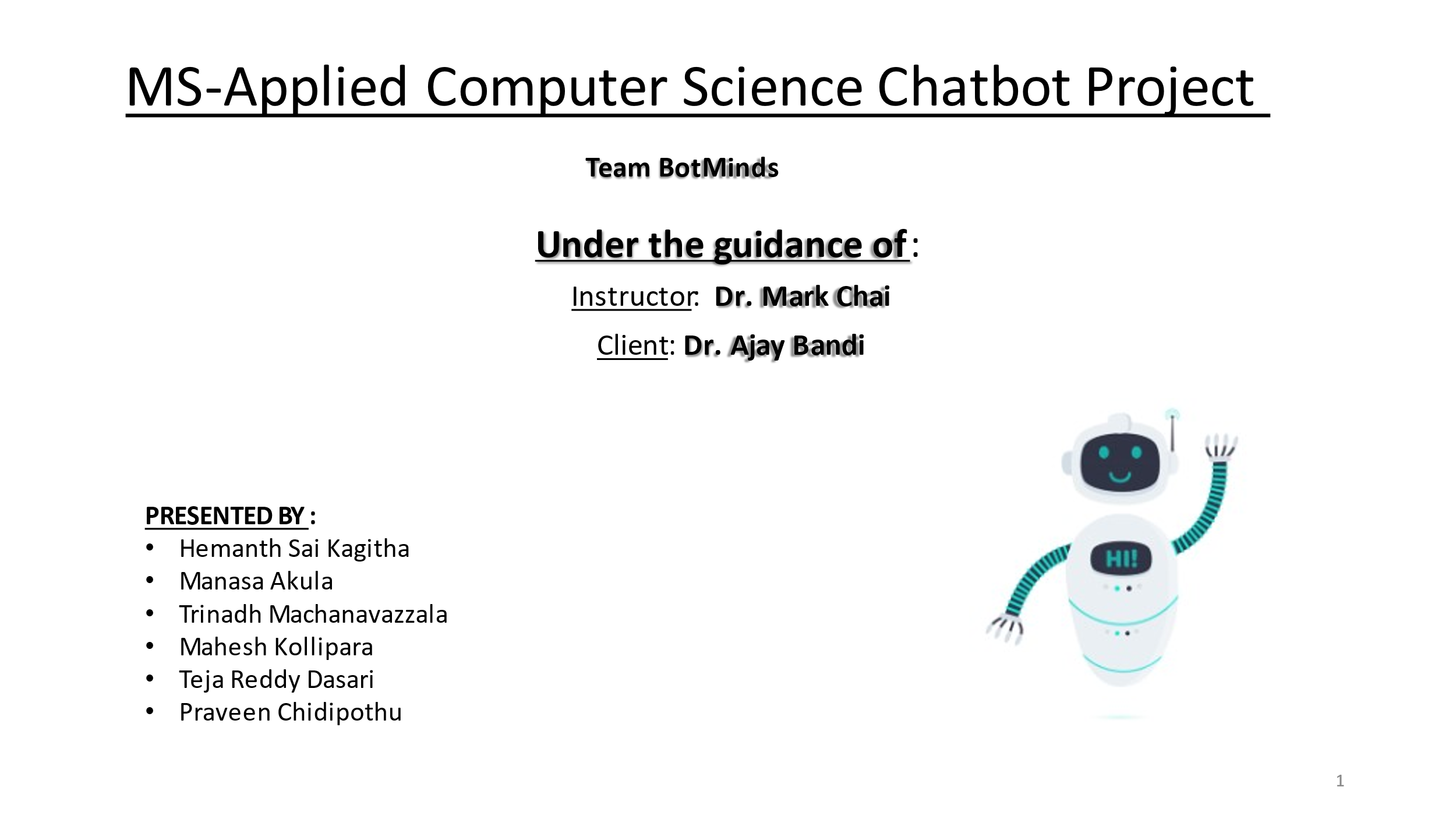
12/01/2023

GitHub URL : <https://github.com/TrinadhM-dev/GDPProject-02>

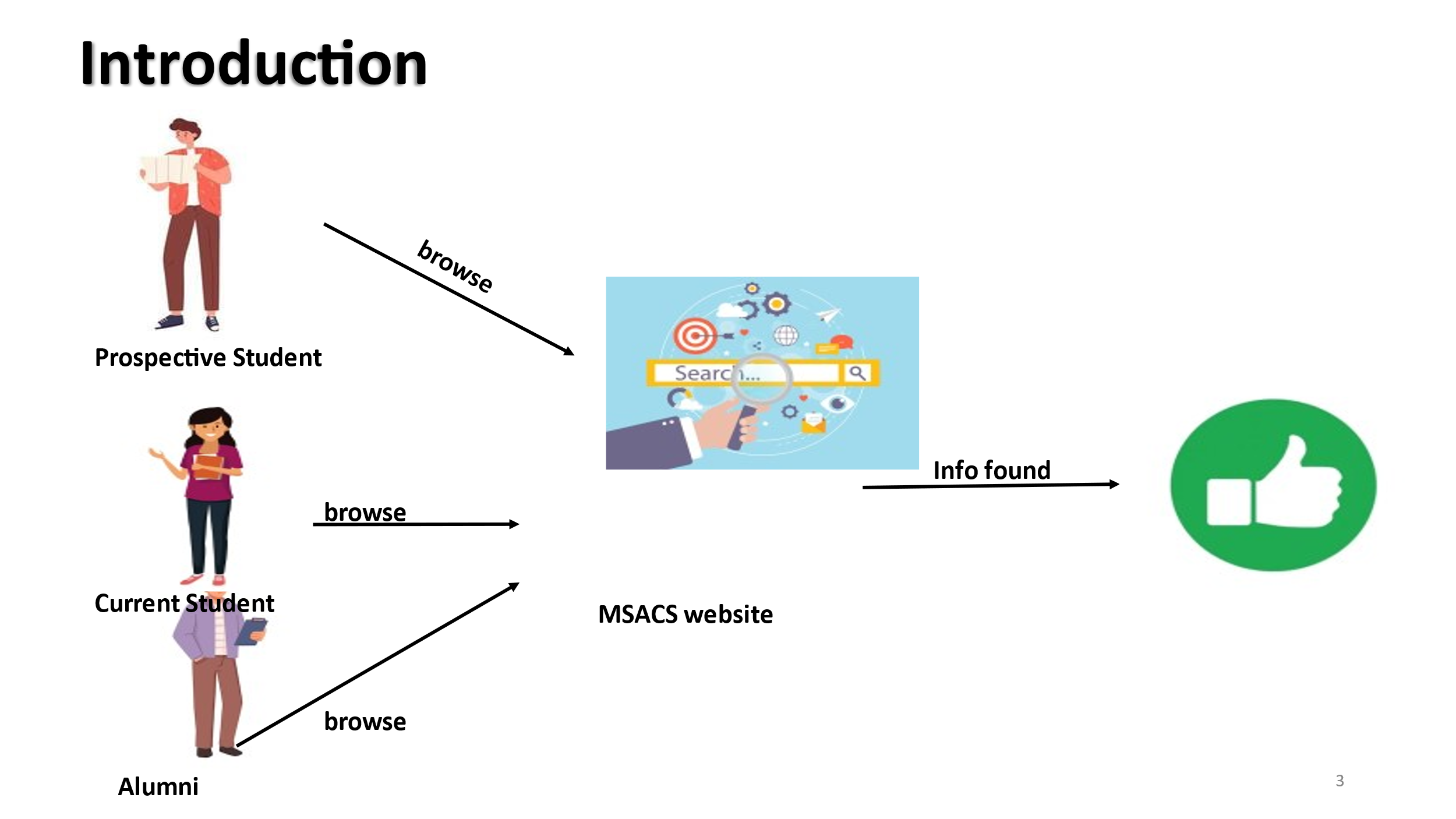
1. Cover paper (Title & Team members)



1. Content

* Introduction
* Problem Statement
* Why Chatbot?
* Proposed Solution
* Use Cases
* Functional Requirements
* Non-Functional Requirements
* Architecture
* Tools and Technologies
* Hosting Strategy and Requirements
* Data Management Plan
* Prototype Demos
* Accomplishments
* Spring 2024 Plan

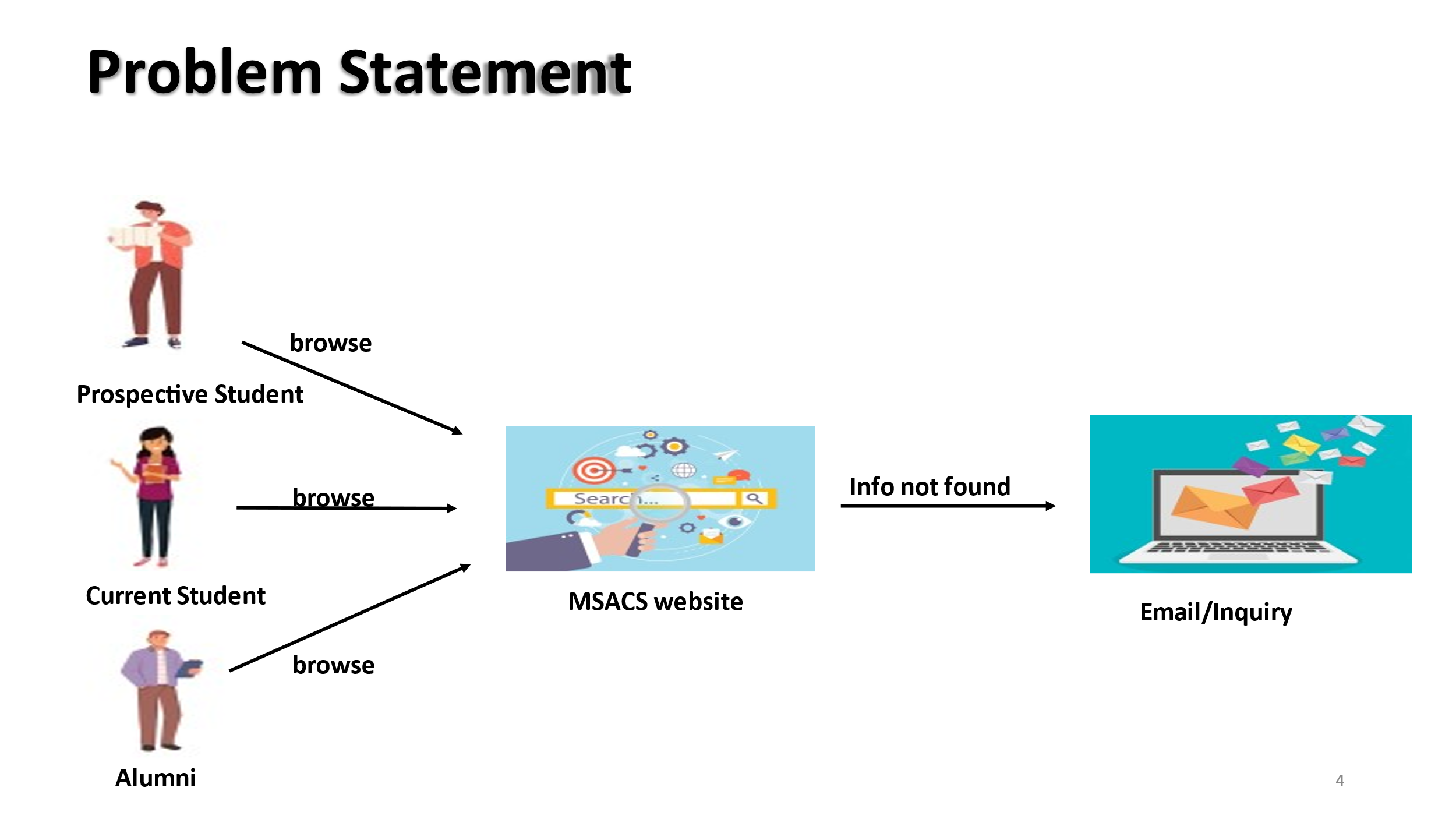
1. Introduction

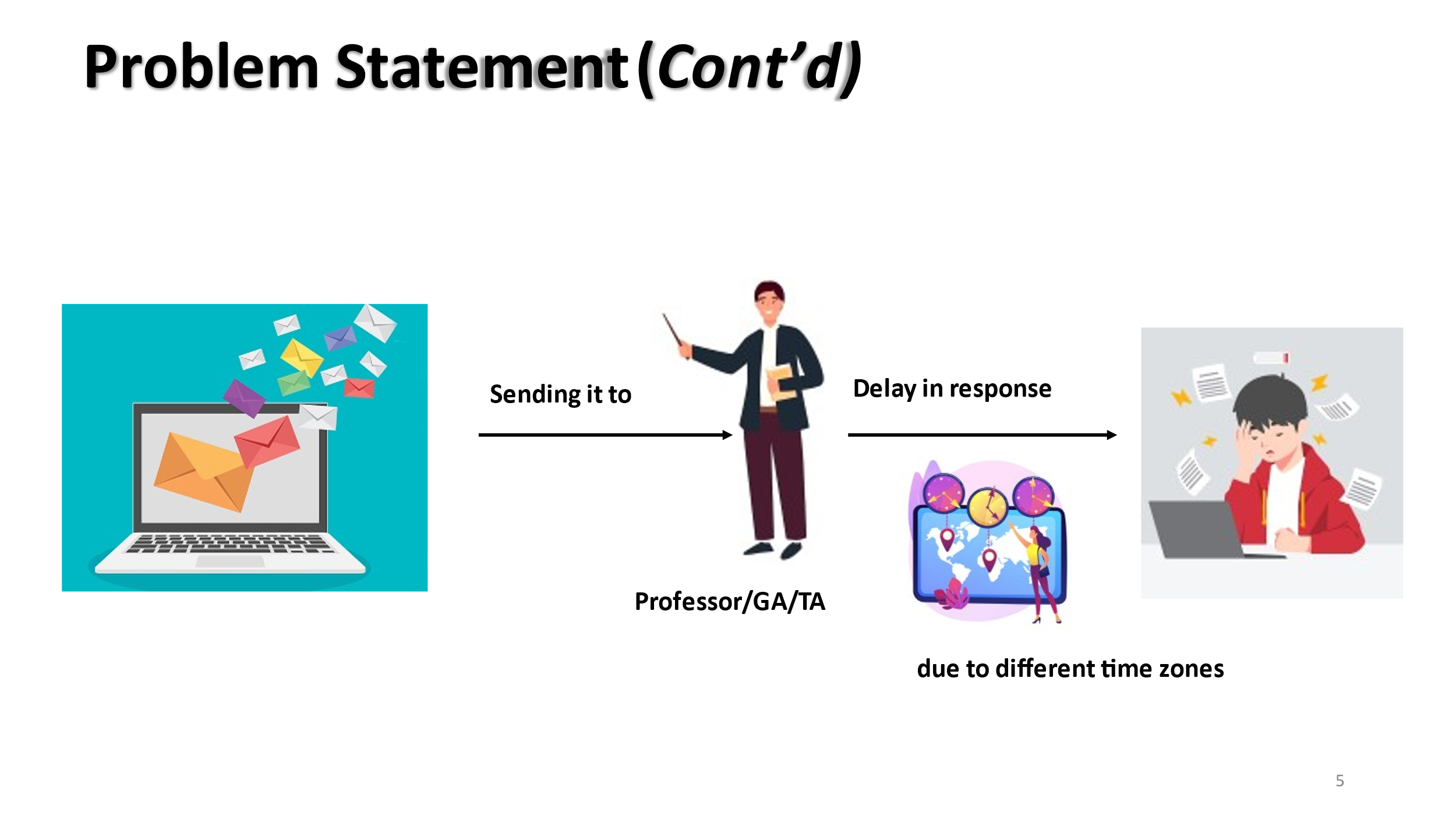


There are three categories of students(Prospective, Current, Alumni) who are trying to request the information from MS-ACS site.

* 1. Prospective students are the students who are willing to study in MS ACS Course in Northwest Missouri State University.
  2. Current students are those who are already studying this course.
  3. Alumni are people who are already graduated from this university.

1. Problem Statement

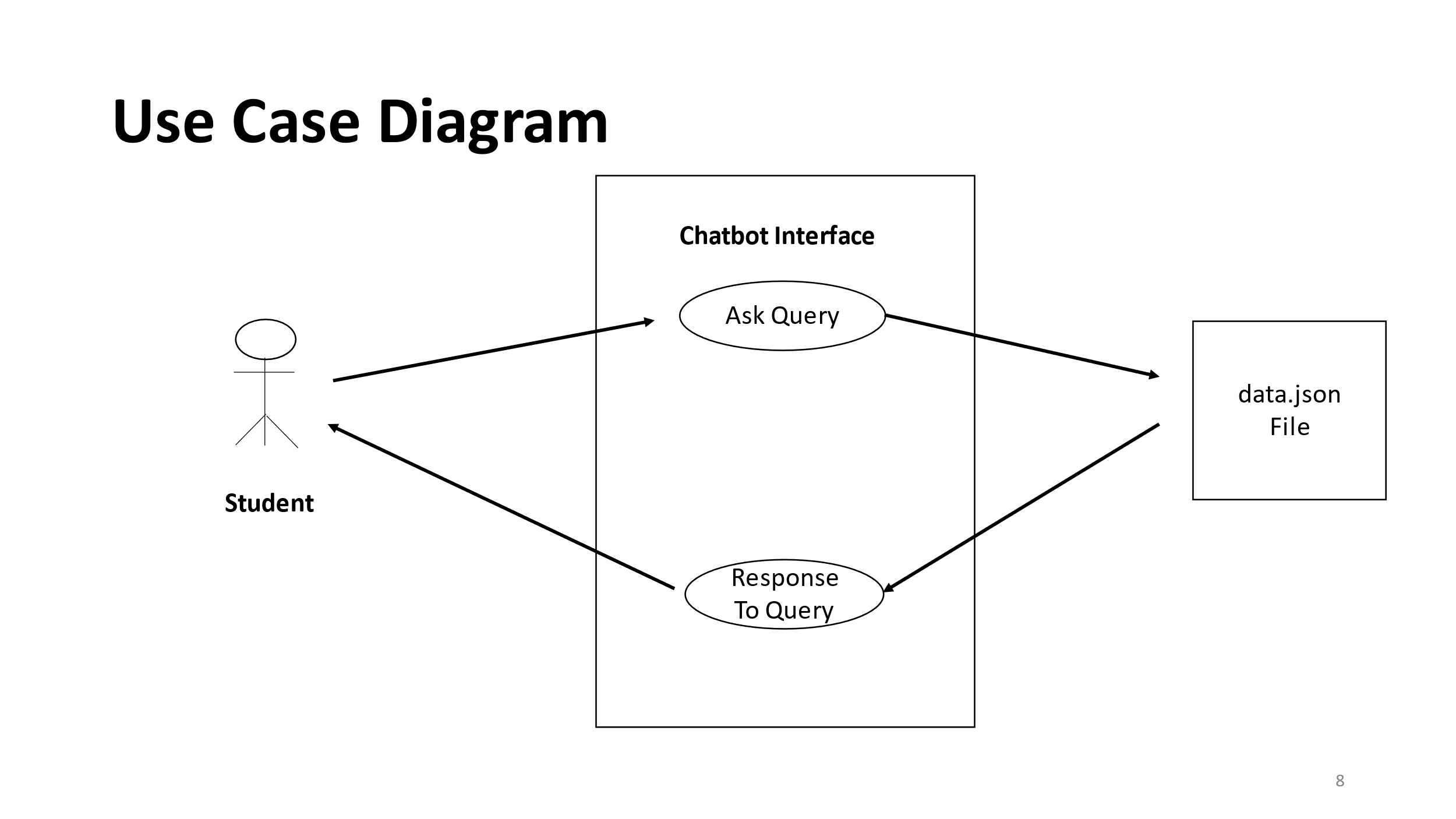




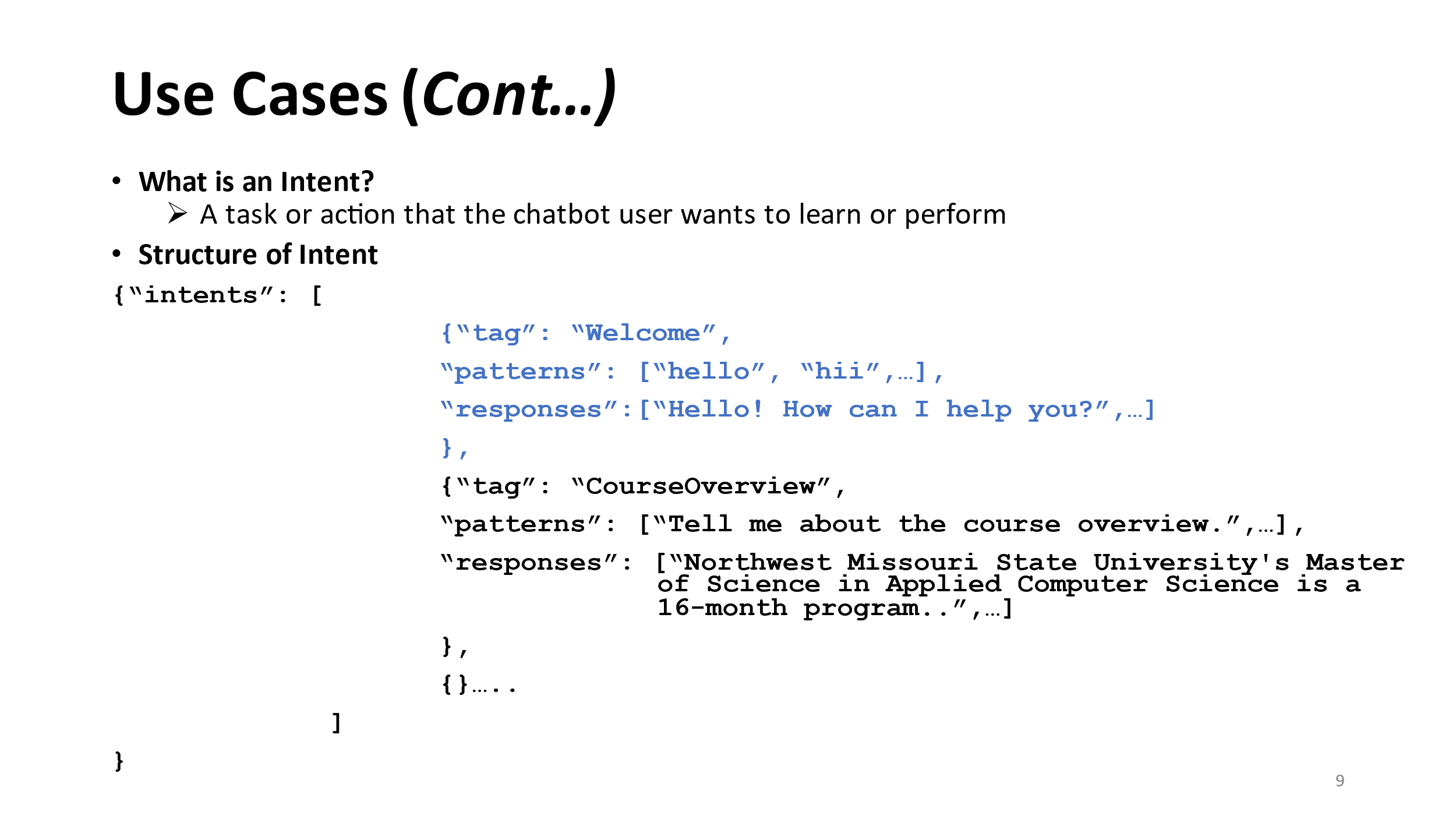
The academic journey for prospective students, current students, and alumni of the MS-Applied Computer Science program faces several challenges:

* Accessibility Challenges: Some users, including students and prospects, struggle to access the website, hindering information retrieval.
* Lack of Personalization: The website lacks personalization, making it difficult for users to find tailored information.
* Irregular Updates: Information is inconsistently updated, causing uncertainty and eroding trust.
* Complex Application Process: The enrollment process is complex, contributing to student frustration and effort.
* Communication Inefficiency: Current course websites have inefficient communication methods, leading to delays in administrative responses and decreased satisfaction.

1. Use Cases :



This is Use Case diagram for a student who requests the chatbot by asking query through it, and the chatbot accesses the data.json file for necessary information and returns the relevant response to the student.



This is how we are storing the Required intents in the data.json file, Here you can examine the Structure of intent.

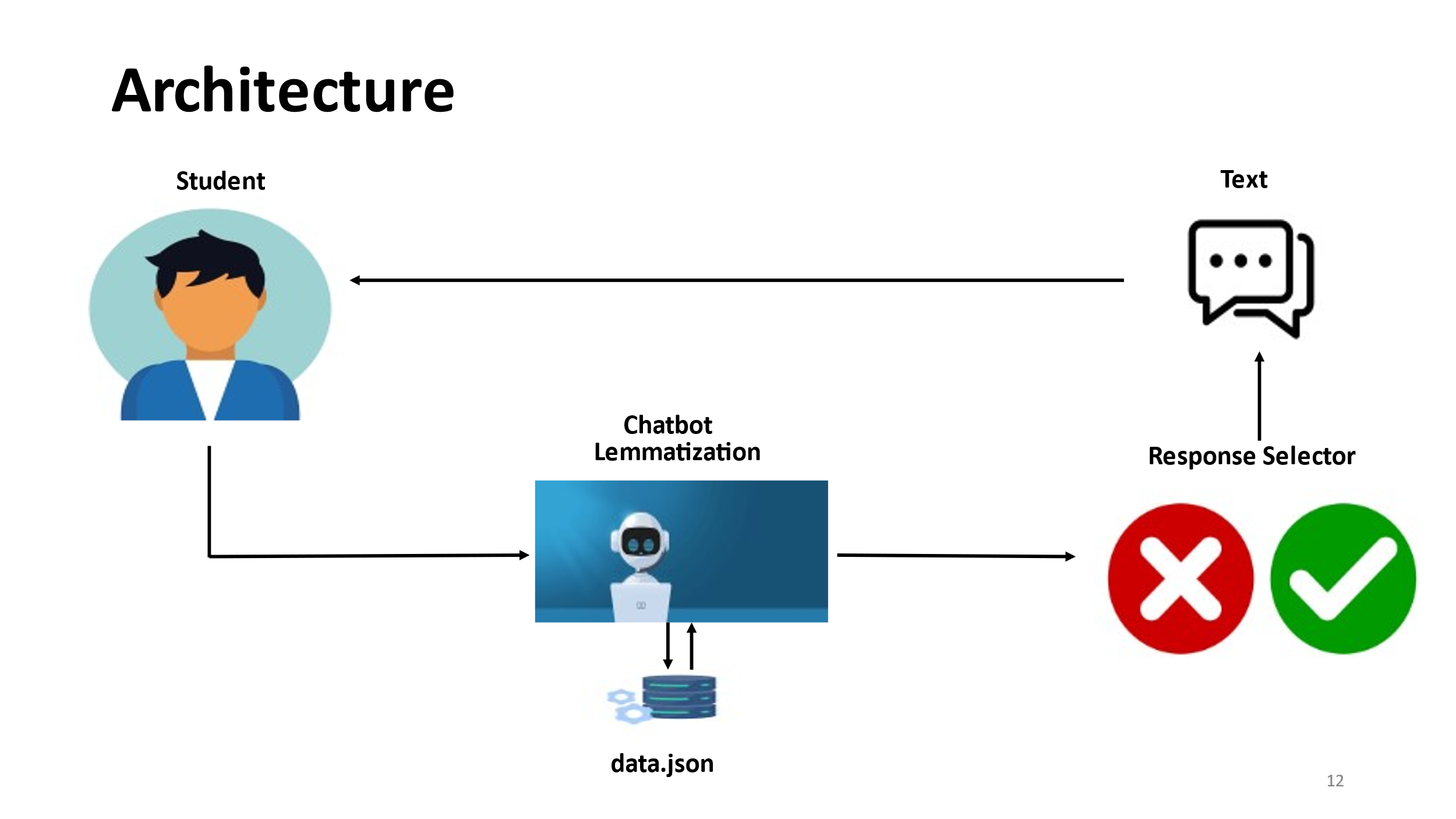
1. Functional Requirements :

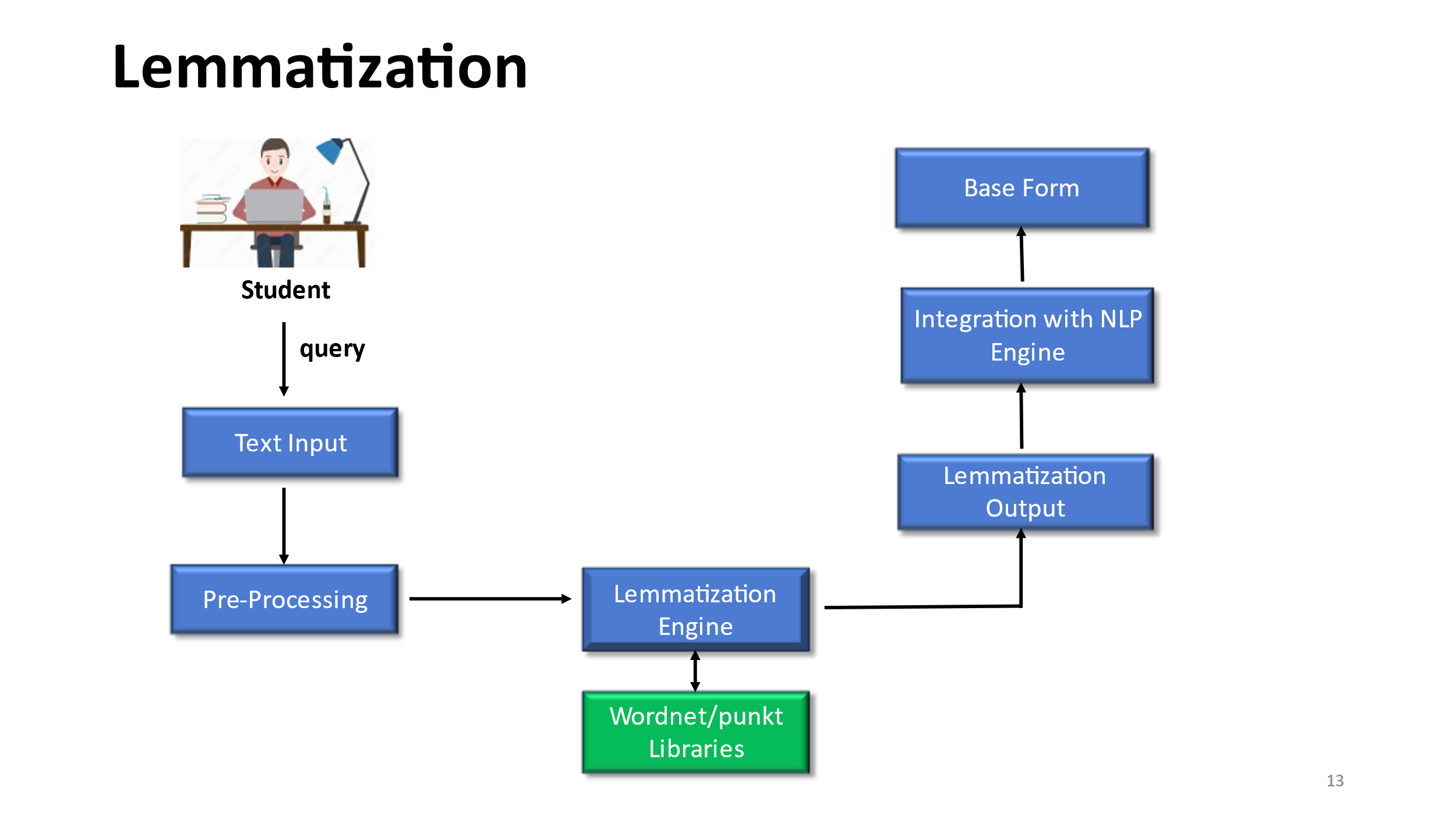
* The chatbot must store chat history and feedback history.
* The chatbot should answer the questions related to three categories.
* The chatbot must process and handle user queries, requests, and commands effectively.
* The chatbot must recognize user intents to determine the user's purpose or request.
* The chatbot must generate contextually relevant and accurate responses to user queries.
* The chatbot must retrieve information from databases, APIs, or external sources as needed to provide answers and services.
* The chatbot shall handle errors, exceptions, and unexpected user inputs, providing clear error messages and suggestions.

1. Non Functional Requirements:

* The chatbot interface should be user-friendly and intuitive, allowing users to interact with it effectively.
* The chatbot should offer a user feedback mechanism for users to report issues or provide input on their training needs.
* The chatbot should continuously improve its training materials and adapt to changing user needs and feedback.

1. Architecture Diagram:

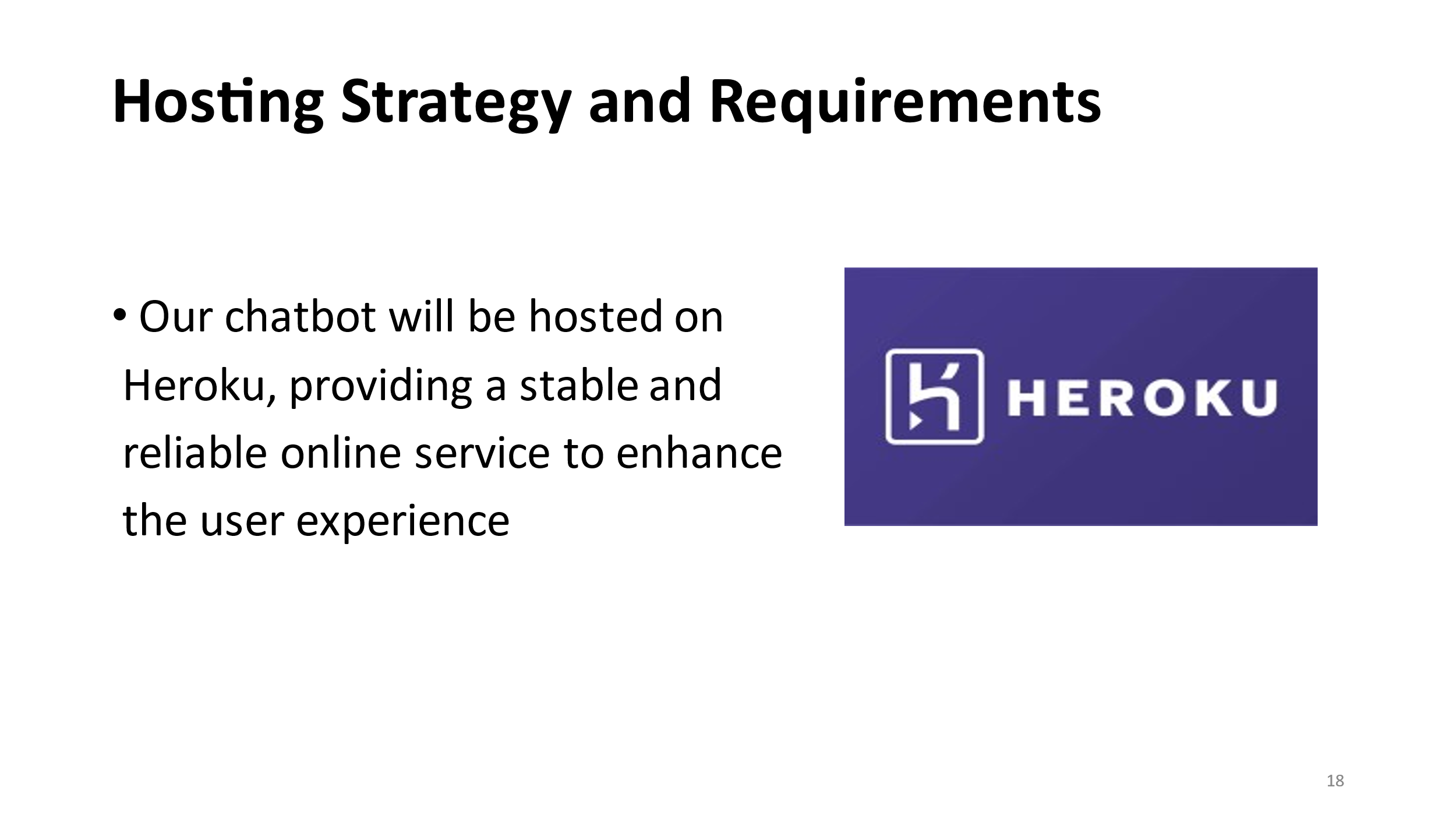


The Architecture Diagram shows that the student sends a query to chatbot, In chatbot the Lemmatization takes place where the query is breakdown into individual tokens and compared with the Third-Party Libraries and make the necessary changes to tokens with utilization of libraries and Response is sent back to the student. 

* This is the detailed process that takes place in Lemmatization, i.e., input by the student in text format is preprocessed (cleaning, tokenization, and POS tagging) ,
* And then Lemmatization Engine efficiently utilizes the Wordnet/punkt library to find the lemma of word in student’s query. Then return the output of base form of word.
* The lemma is then used for further natural language processing tasks (Intent Recognition).
* Platforms and Technologies :

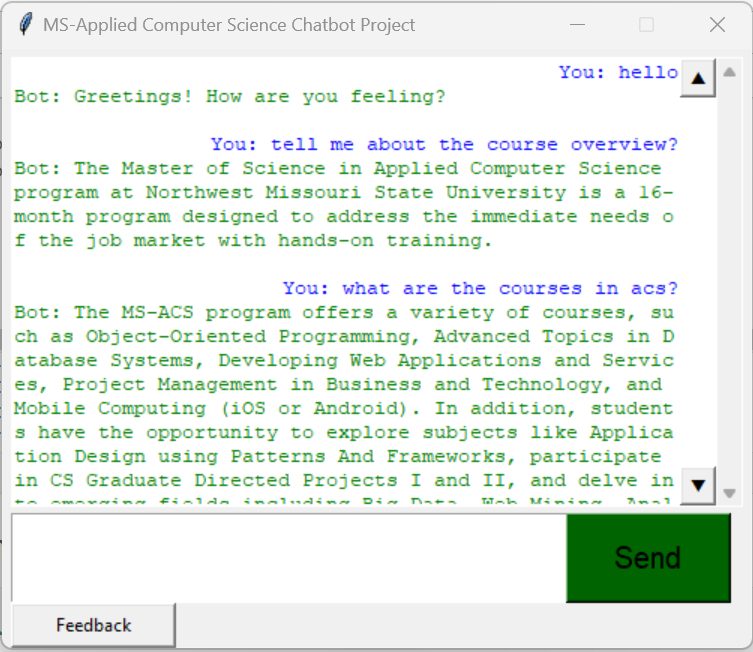


* Hosting Strategy and Requirements :



* Third party libraries to provide functionality
  + Wordnet
    - Contains 155,327 words organized in 175,979 synsets
    - A synset or synonym set is defined as a set of one or more synonyms
    - Example : WordNet synsets for the word “hello”
    - Synset : hello, hullo, hi, howdy, how-do-you-do
    - Definition: An expression of greeting.
    - Example Usage: "She greeted him with a cheerful hello."
  + Punkt
    - punkt is designed to learn parameters in Natural Language Processing
    - It identify the boundaries between sentences (multiple) in text
    - Enables to analyze the sentence for tasks like sentiment analysis, intent recognition, or information extraction
    - Example :
    - text = “This is a sample sentence. And here is another one!”
    - [‘This is a sample sentence.’, ‘And here is another one!’]
    - It returns list of sentences from input query

1. Data Management Plans :
   * Data Storage and Organization:
     + Using JSON files for initial intents and response storage.
     + Plan to transition to a dynamic database
     + (SQL/NoSQL) for scalable data management.
   * Data Processing and Optimization:
     + Regular data cleaning and preprocessing to
     + maintain quality.
   * Backup and Recovery:
     + Routine data backups to prevent loss.
2. Prototype Demos :
   1. TKinter is primarily used for creating GUI (Graphical User Interface) applications in Python, and these applications are generally designed to run on a local machine.



* 1. The source code can be saved as python script (.py), and can be run in console or terminal, But the jupyter notebook facilitate this to run the code for the Command Line Interface.

A screenshot of a computer

Description automatically generated

* 1. This is a web interface developed using Flask library, that serves as a background framework that serves user-centric experience.

A screenshot of a computer

Description automatically generated

1. What have been accomplished?
   * Functional Chatbot: Successfully developed a functional chatbot that can understand user queries and provide relevant responses
   * User Interface Design: Created a user-friendly interface for the chatbot, making it easy for users to interact with and receive information
   * Natural Language Processing (NLP): The chatbot demonstrates NLP capabilities by understanding and responding to natural language inputs from users
   * GUI and Web Interface Implementation using Flask: Successfully implemented a graphical user interface (GUI) and a web interface for the chatbot using Flask
2. Sprig 2024 plan :
   1. In GDP-2, we will be integrating Neural Network Model with Generative AI

* With this, The Chatbot can be coherent, contextually relevant, and varied responses
* The combination of neural networks and generative AI models empower the chatbot to engage in small talk, making interactions more dynamic