The motive of this web application is to provide educational chat for people with visual impairments. It will give an answer to the scholarly questions asked by both teachers and students with visual impairments. Teachers can use the chatbot as an assistant who would plan lessons according to the requirements. Furthermore, students can also use the bot as a learning partner or a helper who would answer questions according to their needs. The output will be provided as text.

A chatbot is a conversational agent in which a computer program is designed to simulate an intelligent conversation. It can accept user input in many formats, such as text, voice, feelings, and many more. Various open-source platforms are available for this. The artificial intelligence markup language (AIML) is a derivative of the extensible markup language (XML), which is used to artificially create a conversational agent (chatbot).

### 2.1.6 Natural Language Processing

Natural language processing means developing methods that help us communicate with machines using natural human languages as English

[3.1 The Development of Artificial Intelligence in Education 25](#_Toc42402823)

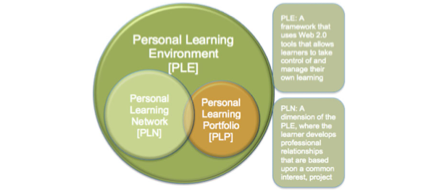
[3.1.1 AI Can Automate Grading 28](#_Toc42402824)

[3.1.2 Supplementary Support from AI Tutors 29](#_Toc42402825)

[3.1.3 AI programs can give a Constructive Feedback 29](#_Toc42402826)

[3.1.4 AI Can Alter the Role of Teachers 29](#_Toc42402827)

[3.1.5 AI makes Trial-and-Error Learning Less Daunting 30](#_Toc42402828)



These will form a part of an individual learning environment in which every individual learner or perhaps life long learner would create and assemble around oneself in an attempt to develop and avail of a continual educational system that has the learner at its center. Personal learning environments or perhaps PLEs are ideal vessels to encapsulate all a learner requires because of their personalization capabilities that empower the same learner. Morrison identifies two essential elements within a PLE as he depicts its anatomy, as shown in overleaf. Each of these components plays a principal role as well as the importance of being investigated individually to make sure they are well set up and compatibility designed to create the expected outcome, an intelligent personal learning environment [38]. These fundamental technologies that source both of them will be justified in terms of academic relevance, general suitability, and pedagogical effectiveness.

5.2.1 It detects the emotional state of the students

That, when identified by the chatbots, can change the response with language adaptation or perhaps even incorporating a joke.

5.2.2 Provides personalized learning

Adapting to the rhythm of the pupil, according to the needs of theirs and specific requirements. This offers a far more direct orientation when sending info or perhaps solving queries about a course

### 5.2.3 It allows the teacher to reduce time

They have invested in the execution and organization of tasks since chatbots provide quick answers, previously predesigned, to the pupils. This time saved could be invested in research and projects pending for the course and also in the inspiration and supervision of the team.

### 5.2.4 Store and analyze data effectively

While reviewing the evaluation and progress of the user, as a consequence of the use of Artificial Intelligence, it can help pupils organize the time of theirs and assign tasks based on the objectives of theirs in an accessible and effective way.

### 5.2.5 Improves access to education

The automatic learning tool is oriented to interaction and formation and doesn't think about the resources, the location, or even the language of the pupil. It may be regarded as a thing such as a "democratization of learning"[57].

Together with the evolution of the chatterbots, researchers observed that the structure of chatbots modeled using the Natural Language Processing (NLP) would represent itself as a complicated job. To be able to help in such tasks, dedicated technologies for chatterbot making have been designed. Among these technologies, Wallace, in collaboration with other software developers' communities, can be noticed. the Artificial Intelligence Markup Language. (AIML) is created, established on the principles of Pattern Recognition, or perhaps Matching Pattern Technique from 1995 to 2000, It is put on to natural language modeling for the dialogue between humans and chatterbots that follow the stimulus-response approach.

For this reason, a set of possible user inputs is modeled. For every among these sentences (stimuli), pre-programmed answers were built to be proven towards the user. The ALICE (Artificial Linguistic Internet Computer Entity) chatbot was the first to work with the AIML language and interpreter. In ALICE, the AIML technology was supervised for pattern matching as well as to connect a user input with a response of the chatbot's Knowledge Base (KB).

In this paper, an AIML language tutorial is presented. Concepts of Pattern Recognition are considered; however, the main plan is to get a reference guide for AIML language's initial studies.

A close up of text on a white background

Description automatically generated

pip install python-aiml

A close up of a logo

Description automatically generated

Figure 6: Installing Python

Simplest Python Program

A screenshot of a cell phone

Description automatically generated

Figure 7: simplest Python program

Adding Python Commands

A screenshot of a cell phone

Description automatically generated

Figure 8: Adding Python commands

When the system begins to have many AIML files, it can have a very long time to master. This is precisely where brain files are available. After the bot learns all of the AIML files, it can save its brain directly to a file that will significantly speed up load times on the subsequent runs.

A screenshot of a cell phone

Description automatically generated

Figure 9: Adding brain file

Sessions and Predicates: By specifying a session, the AIML can customize different conversations to folks that are different. For instance, if one person tells the bot their name is Alice, and someone tells them their name is Bob, the bot can differentiate the folks. In order to specify which session users are making use of the system pass it as another parameter to respond().

sessionId = 12345  
kernel.respond(input(">>>"), sessionId)

In the AIML we can set predicates eith the help of the **set** response in template.

A screenshot of a social media post

Description automatically generated

Figure 10: response templates

This is ideal for having personalized conversations with each client. Users will have to generate their session Id somehow and track them. It has to be kept in mind that saving the brain file does not save all the session values.

A screenshot of a cell phone

Description automatically generated

Figure 11: AIML Matching

Among s Natural Language Processing begins when the client submits an inquiry to Visit bot. This information base was executed in AIML language consisting of labels on class structure.

C. *LSA*: Latent Semantic analysis (LSA) is a thought and approach for separating and speaking to the relevant utilization of words by method for factual calculations actualized to an enormous corpus of literary substance [61].

A screenshot of a cell phone

Description automatically generated

## 6.5 System Overview

Once the application has opened it will give voice instruction to the user, using voice recognization and pyaudio. From that user comes to know about how to use the application. If the record button is clicked, it will give the sound to indicate that the user 's voice is being recorded. Again, the app will give the sound to indicate that the record has been completed. Then the application retrieves the result.

Moreover, what features can AIML-based chatbots have in an educational context? This question has been answered thoroughly in this thesis, but the answer might be summarized as: The features and uses of AIML-based chatbots are many, and they seem only to be limited by what humans might imagine it does. It is a flexible and diverse tool, simultaneously simple and complex, limited only by the inventiveness of its creator. It can work well on its own and provide educational value, but it may reach even better results when combined with other technology.

Python

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment

Spyder is an open source cross-platform integrated development environment for scientific programming in the Python language.

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

AIML, or Artificial Intelligence Markup Language, is an XML dialect for creating natural language software agents

Library for performing **speech recognition**, with support for several engines and APIs, online and offline.

**pyttsx3** is a text-to-speech conversion **library** in Python. Unlike alternative **libraries**, it works offline, and is compatible with both Python 2 and 3

**Jsonify** is a python script that takes a .csv file as input and outputs a file with the same data in .json format