Educational Chat-Bot
Using Artificial Intelligence

A PROJECT WORK SYNOPSIS
Submitted by
RISHABH ANAND
19BCS4525

Under the Supervision of:
Aanchal Sharma

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE and ENGINEERING
(Internet of Things)

APEX INSTITUTE OF TECHNOLOGY
CHANDIGARH UNIVERSITY, GHARUAN
Mohali, Punjab

October, 2022

Contents

List of Figures **v**

1 INTRODUCTION **1**

1.1 Need Identification 1

1.2 Problem Identification 1

1.3 Tasks Identification 1

1.4 Timeline 1

1.5 Organization of Report 1

2 LITERATURE SURVEY **3**

2.1 Timeline of the reported problem 3

2.2 Proposed Solutions 3

2.3 Review Summary 3

2.4 Problem Definition 4

2.5 Goals 4

3 METHODOLODY **5**

3.1 Selection of Features 5

3.1.1 Easy Customization 5

3.1.2 Quick chatbot training 5

3.1.3 Emotional intelligence 5

3.1.4 Security & Privacy 6

3.2 Design Constraints 6

3.2.1 Rule-based approach 7

3.2.2 AI-based approach 7

3.2.3 Talking to the audience 7

3.2.4 Understanding AI 8

3.2.5 Defining the bot 8

3.2.6 Defining Sample chat flows 8

3.2.7 Engagement & Attention 9

3.2.8 Clutter 9

3.2.9 Users' time 9

3.2.10 Be honest 9

3.2.11 Be creative 10

3.3 Feature Analysis 10

3.3.1 What makes chatbot successful? 11

3.3.2 Chatbot Analytics 12

3.3.3 Data Security 12

3.4 Design Flow 13

3.5 Design Selection 15

3.6 Implementation Plan 16

4 RESULTS ANALYSIS AND VALIDATION 17

4.1 Implementation 17

4.1.1 File Description 17

4.1.2 Training The Network 17

4.1.3 Model Architecture 18

4.1.4 Functioning 19

5 CONCLUSIONS 21

5.1 Conclusion 21

5.2 Future Work 21

REFERENCES 22

List of Figures

1 Timeline 2

2 Feature Analysis 10

3 System Flow 13

4 User Flow 14

5 Design Flow 14

6 Implementation Plan 16

7 Training the model - 1 18

8 Training the model - 2 18

9 Sample Conversation 20

INTRODUCTION

1.1 Need Identification

A number of students miss out on important information that is being circulated within an institution if they are not paying attention or miss out on classes.

1.2 Problem Identification

Missing out on this information causes a lot of issues for them such as missing::

- Placement drives
- Events
- Tests
- Submissions

1.3 Tasks Identification

The major tasks include ::

- Generating a data-bank using institute's site
- Generating a data-bank using institute's noticeboards
- Regularly updating the data-bank
- Providing a user-friendly way to convey this information to the relevant people.
- Providing a way to personalize the information to the user.

1.4 Timeline

1.5 Organization of Report

1. Literature Survey :: Includes information about similar previousp projects.

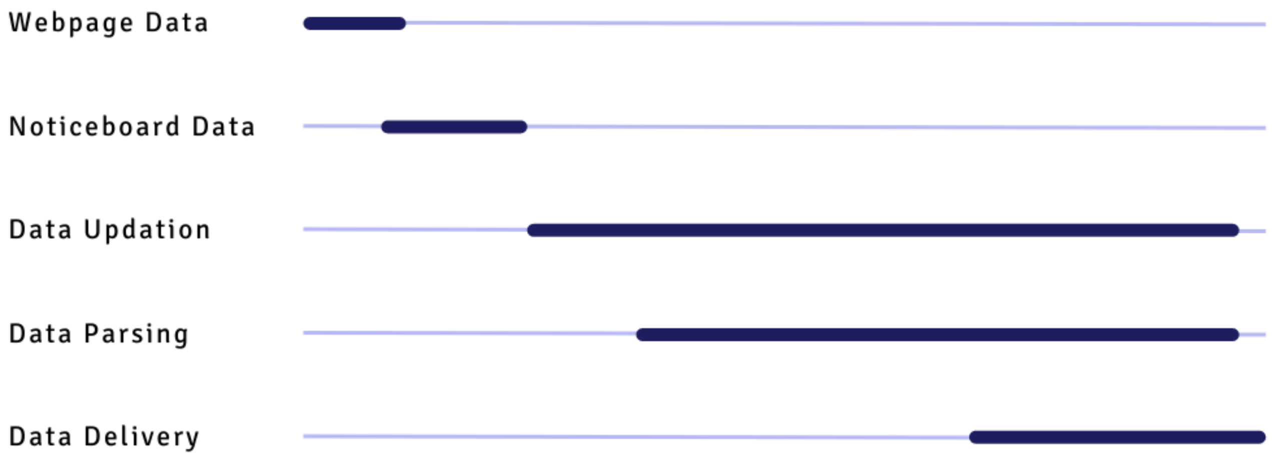


Figure 1: Timeline

2. Design Flow :: Discusses the design decisions taken during the project building.
3. Result Analysis :: Discusses the methodology adopted for the project.
4. Conclusion :: Discusses the conclusion of the project and its future aspect.

LITERATURE SURVEY

2.1 Timeline of the reported problem

The problem starts as soon as we join any institution as that is when most of the important information is passed around that we have to keep a track of. With the abundance of information comes ignorance of knowledge. The real problem starts once we reach our final semesters and it's the time of placements. There are mulltiple companies coming to the campus everyday and conductiong events every hour. In all this chaos, it is hard to keep track of the companies that we might be interested in or the companies that actually are good for our career depending upon the role that we desire.

2.2 Proposed Solutions

There have been attempts to address this problem before as well, from both the institution as well as the students to make it easier to keep track of things and news.

A notice board : where institutions update the students with the news. A low graphic, UX ignorant dashboard with the entireity of information thrown right at your face. Trusted and updated but neither elegant nor effective.
Emails : where institutions update the student with the news and placemnet oppourtinities. A slow, non reliable source of information as many student accidentally put the mail in spam and don't recieve from the sender anymore.

The idea that students came up with is mobile applications. Where they can keep track of information. The news around the campus. This however doesn't cover the emails. It is also not reliable you still need a seperate app in your phone just for this thing.

2.3 Review Summary

From the above solutions we have two scenarios, one where the information is available but not presented properly which makes it hard to convey the information.

The other one where neither the information is complete nor is it easily accessible and isolated. We also saw that we need to tackle with different data sources and present information in a condensed and private manner where the information is personalized for the user and relevant to him. We also saw that trhe solution needs to be platform independent to make it easily accessible.

2.4 Problem Definition

Provide information related to campus activities and placements by compiling different data sources and presenting the data in a elegant and effective layout which is personalized and relevant to different users.

2.5 Goals

- Different Data Sources (Notcie-Board and Emails) Compiled.
- Filtering of Information.
- Personalization of Information.
- Good UX and Content.
- Cross-Platform Solution.

METHODOLODY

3.1 Selection of Features

The features for this particular project have been chosen carefully and thoughtfully. They are as follows :

3.1.1 Easy Customization

It is important to keep the chatbot easily customizable as the main feature of this model is to provide information that is customized and personalized according to the account and the data that the user has entered during the setup process. This data will later be used to fetch emails and get his/her grades, classes, mentor information, etc.,. Making the chatbot customizable also helps users upload their own custom cssto truly customize their experience and tailor it what looks best to their eyes. And not only look wise, they can also opt in to various other submodules and features that the bot has.

3.1.2 Quick chatbot training

This is one of the most exciting capabilities, where you can train the chatbot to perform complex reasoning on its own, without human interference. The bot uses intents built on keywords which have been derived after scraping the college website. But with abundance of knowledge, it becomes difficult to have a sustained sufficiency of answers. Thus, the abundance of this data has to be cleaned to reduced to instances that have a predefined basic template of a response. And all of this is done automatically. Instead of manually adding and updating FAQs, you can simply load your knowledge base to the chatbot. The chatbot parses through the information and can provide a suitable answer within seconds. Instead of passing simple queries to live agents, this chatbot provides a very good alternative to that.

3.1.3 Emotional intelligence

Emotional connect helps students to engage with each the bot and build a relationship. Since chatbots are the primary interface between the university queries and students, it's vital to enable or design chatbots that can build and foster relationships with students. Moreover, the bot also understands slangs and shorthands that we usually use in our talks to make the conversation easier and interesting.

3.1.4 Security & Privacy

With so many data breaches, your chatbot must be secure. It's a good idea to have a fully transparent policy regarding the data bots, such as what they collect and exactly what it is used for. Users should also be given the option to opt-out of data collection (if desired), but make note that this will inhibit its ability to do progressive profiling.

3.2 Design Constraints

I have been extremely lucky to get a chance on designing a chatbot and the learning in the process has been massive. Most of the notions that I thought were true were discarded by research and a whole new world of possibilities just opened wide.

I feel it safe to assume that being residents of today's digital world, you must be familiar with the concept of chatting via text message. Be it SMS or online messaging, our way of life has been irrevocably pervaded by connectivity. Did you know that 17% of all human interaction happens via text messaging? Well, it should come as no surprise, as there now exist over 4 billion people using messaging applications worldwide.

An indirect result of the above-mentioned 'digital world', is the generation of large amounts (several million terabytes) of data from different sources. The collection of such massive chunks of data or, 'Big Data', is what has given rise to large-scale data analysis. This has in turn given rise to devices as well as entire systems that are capable of dynamic learning, and artificial intelligence (AI).

Given that over 18 billion texts are sent around the world daily, it is clear that we are quite heavily reliant on text messaging as a form of communication. And it is in order to streamline certain modes of such communication, that chatbots are created. We're all aware of the functionality **o**f a chatbot: they're basically messaging systems automated by AI software, or machine, that's got a fixed amount of set data. The framework of the software, as well as its impact on the end-user experience, are necessary in order to create an effective chatbot. There are two ways in which this can be achieved:

3.2.1 Rule-based approach

This is a static approach (relatively) to the creation of chatbot, wherein there is a pre-fixed set of rules that act as guiding parameters, based on which the bot responds to user input (queries, etc). Depending on the requirement, these rules can range from simple, to very complex. This approach, however, does have drawbacks that may affect user experience, if applied in a wrong way. Although this is the more straightforward of the two approaches, there is a lack of efficiency in the overall functionality of the bot.

3.2.2 AI-based approach

This approach enables the bot to be more dynamic in its responses, as well as functionality. The process itself is much more complex as compared to the above, as it requires that the chatbot is connected to an AI. The driving forces behind this approach are advanced data analytics, API (Application Programming Interface) integration, and the subsequent machine learning that takes place. In this way, the bot is able to learn dynamically, and modify its working (responses) in order to provide a more efficient, personalized user experience.

You should note however that, both of the above approaches have their merits, and their applicability is conditional solely to developer requirement(s).

Given that chatbots are a fast-growing concept today, I feel it is necessary to state that with the basic facts related to the subject, and how and why the design of a chatbot is of utmost importance. The following points give a good handle on this chatbot application and what it does to provide the best UX :

3.2.3 Talking to the audience

It’s very important to talk to whom the chatbot is designed for, i.e. the target user. Knowing their requirements, as well as their expectations is crucial. It’s safe to say that if the product that doesn’t fulfil the TG’s requirements is built, it’s an absolute waste. By talking to the audience, an understanding of what their expectations are from the bot is gained.

Also, it is crucial to understand what is the vision of the product owners and what exactly are they willing to achieve from the bot. Once clarity on both the aspects was achieved, the bot was designed in no time.

3.2.4 Understanding AI

To effectively design a chatbot, an understanding of artificial intelligence is also needed. This helped in identifying any limitations of the technology, and also helped in avoiding coming to a roadblock while the bot was designed. Besides identifying its limitations, you may discover possibilities you didn't know of— something that's always helpful. As a matter of fact, every designer should be updated about the technology and platform they are designed for.

3.2.5 Defining the bot

Next comes the stage where working parameters are set for the software. This stage is the establishment of guidelines, or a framework within which the bot is expected to carry out its duty. This step is crucial in defining the functionality, structure, and perceived persona, from the point of view of the developers and users requirements. Based on the target audience and the understanding from the previous stage, definitions of things like whether your chatbot is going to be friendly, or professional and bot-like, or even social was achieved.

3.2.6 Defining Sample chat flows

After this, it's necessary to define the flow of the chat in the bot. This step is actually one of the most important ones, as it defines how the bot will interact with humans. I created as many scenarios as possible depending on the scope of the bot; this particular process will help you define the key terms that bot will take into consideration before replying to the user's query. It is actually a good idea to spend a lot of time on this step to get close to defining the experience for your users.

3.2.7 Engagement & Attention

Now that I had the structure in place and it is time to start with interaction that ensures that the target audience enjoys the offered services and find it extremely useful as well. Here, we're left with the decision of whether to design the chatbot the traditional way, i.e. how every other messaging app like WhatsApp and Messenger looks, or you could decide whether you want to try something that focuses on what's most important.

3.2.8 Clutter

While designing the interaction, it's important to focus on providing answers. Research shows that users go to bots for seeking quick answers or recommendations and do not expect a bot to be human. The users are using the chatbot for one reason, and one reason only: to seek an answer to one of their problems.

3.2.9 Users' time

I find it's always a good idea to present the users with options while they're chatting with the bot. The reason being, it's a huge time-saver, and also allows the conversation to take place easily, and seamlessly. This rings especially true if the bot is rule-based bot, as it may sometimes fail to understand what the user types, which will result in it providing inaccurate information. Only mapping chat flows can solve this problem.

3.2.10 Be honest

When a bot is designed, its important to not make it seem like it's human. It's important to let the bot be a bot. Why? Well, a survey once showed that users found ChatBots pretending to be human is "creepy", and it's also important to be honest to your users, as well. Avoiding the use of indicators such as "is-typing" or artificial delays to give the user the illusion that they're chatting with a human is a good place to start. The style of the bot's messages is set differently so that the bot conveys it isn't a human.

3.2.11 Be creative

This bot doesn't use the typical chat layout. Since there was a choice of exploring options, I chose to go with a layout that's different. And Fast. The new layout isn't too complicated, and users are able to get the hang of it quickly.

3.3 Feature Analysis

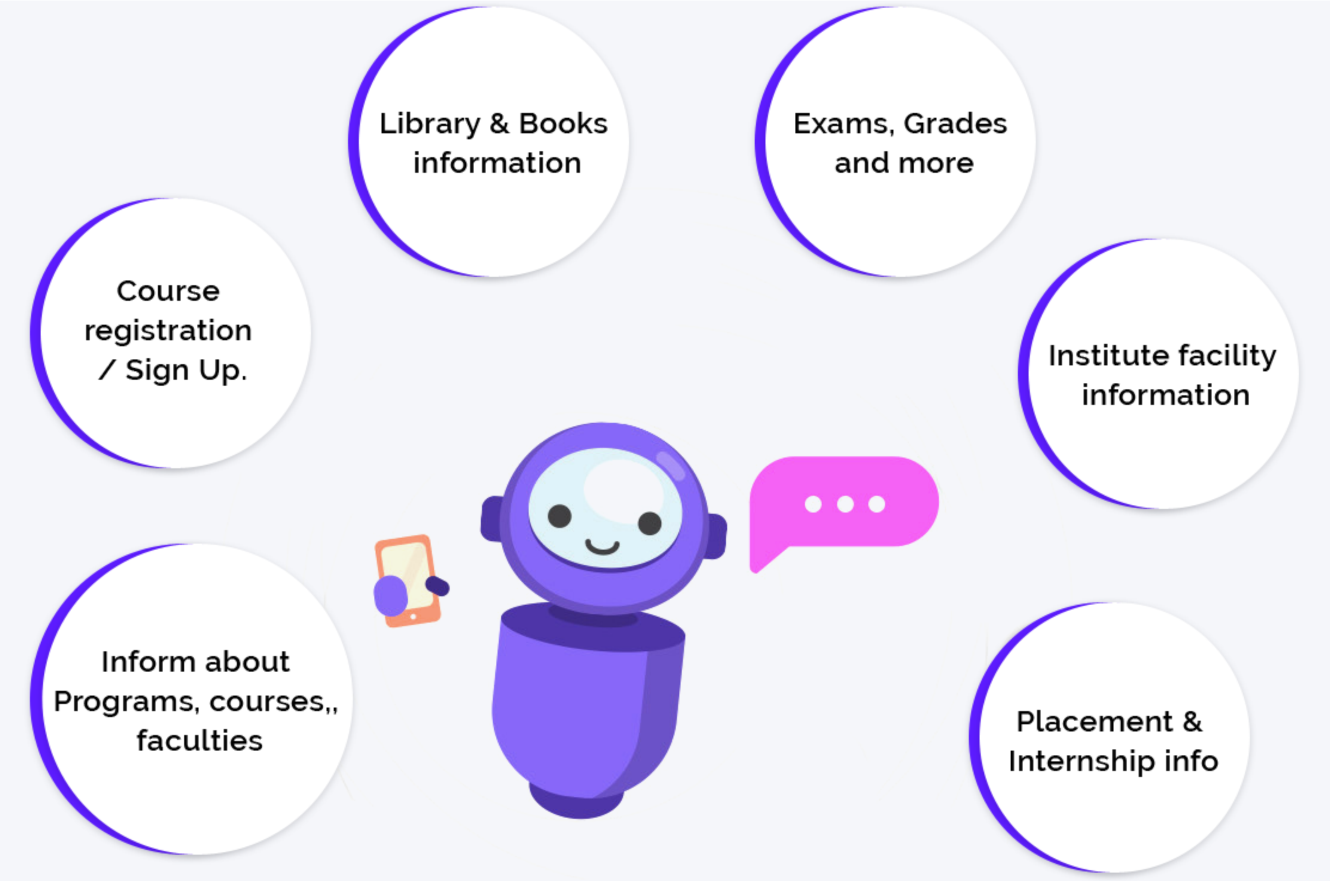

Library & Books information
Exams, Grades and more
Course registration / Sign Up.
Institute facility information
Inform about Programs, courses,, faculties
Placement & Internship info

Figure 2: Feature Analysis

Over 64% of respondents believe that chatbots help them offer a more personalized service experience for students. In fact, chatbot features have evolved so much that organizations find them indispensable to their college communication strategy for enabling real-time responses to consumers.

By deploying bots, colleges can automate interactions as students are also more comfortable interacting with chatbots. However, you need to know those AI chatbot functions that can help meet the student expectations and deliver a prompt answer to their questions.

3.3.1 What makes chatbot successful?

The best chatbots always focus on the quality of the conversation and have features that ensure a high-caliber conversational experience. There are many real-life chatbot examples that combine the key elements of technology, flow, and design in order to prove effective in handling student interactions without requiring any human assistance.

Here are some key features that make chatbots successful.

- **Deliver contextual responses**: Chatbots need to have the ability to understand the context so that students feel like talking to a real person. By leveraging the advancements in natural language processing (NLP), bots can be made to understand context without asking validating questions.
- **Parsing and Summarizing**: This bot chooses sustainable suffiency of answers over abundance of knowledge. This makes it fast and quick with responses with summarized information for students.
- **Allow human handover**: Bots need to be smart to understand the sense of urgency and complexity of a conversation. Even when a chatbot template for online order fails to understand the query, it can still intelligently hand over the conversations to human support.

- **Great UI/UX**: AI chatbots should not be complex or hard to use else they won't be able to make conversations interactive. The design has to be simple and intuitive so that users find it easy to use them for answers.
- **Well-trained with FAQs**: Chatbot benefits are many when they are trained. Regular training can help chatbots become powerful and enable them to smoothly handle questions and interactions.
- **Offer personalized support**: Bots that are designed using AI and machine learning can easily comprehend user conversations and respond in real-time. A student support chatbot template can adjust the tone and language to give personalized experiences.

3.3.2 Chatbot Analytics

Chatbot analytics can help in knowing your students in detail and leading with data. Using this feature, a college can get a deeper understanding of the students and make better decisions.

With the chatbot analytics feature, you can get valuable insights and analyze all the chat conversations handled by your bot. It can help you measure the accuracy of the responses provided by bots to students.

By planning a successful chatbot strategy, you can measure your bot performance and assess the growth of your college. You can leverage chatbot analytics to track relevant chatbot KPIs to make data-driven decisions and better understand the student journey.

Key chatbot metrics to evaluate your chatbot performance:

- **Total number of users**: The total number of users who interacted with chatbots can be traced to get insights on how many students are using your chatbot.
- **Bounce rate**: It denotes the number of users visitors who enter the website and leave without interacting with your chatbot.
- **Interaction rate**: Observing the interaction rate KPI can help you measure user engagement during conversations with your chatbot.
- **Fallback Rate (FBR)**: The fallback rate will capture insights about the scenarios where the bot is unable to gauge the user request and offer a relevant solution.

3.3.3 Data Security

No technology is totally hacking proof and chatbots are not different. They can be at risk due to various reasons including weak coding, poor safeguards, or user error.

Chatbots are now quite common across industries that handle very sensitive data and personally identifiable information. Chatbots need aggregation of data to perform optimally and they need not be vulnerable to hacking attacks.

Threats are one of two types of security risks that chatbots are susceptible to as they include malware and DDoS attacks that can hijack the system and hold you to ransom. Hackers can also expose sensitive student data or use the vulnerabilities in the system to their benefit.

Best practices to ensure chatbot security :

- **End-to-end encryption**: A chatbot design should prevent anyone other than the sender and recipient from accessing the message (comes in v2).
- **Authentication and Authorization**: These are the two main security processes that chatbot needs to use to ensure user identity verification and granting permission to do any task.

3.4 Design Flow

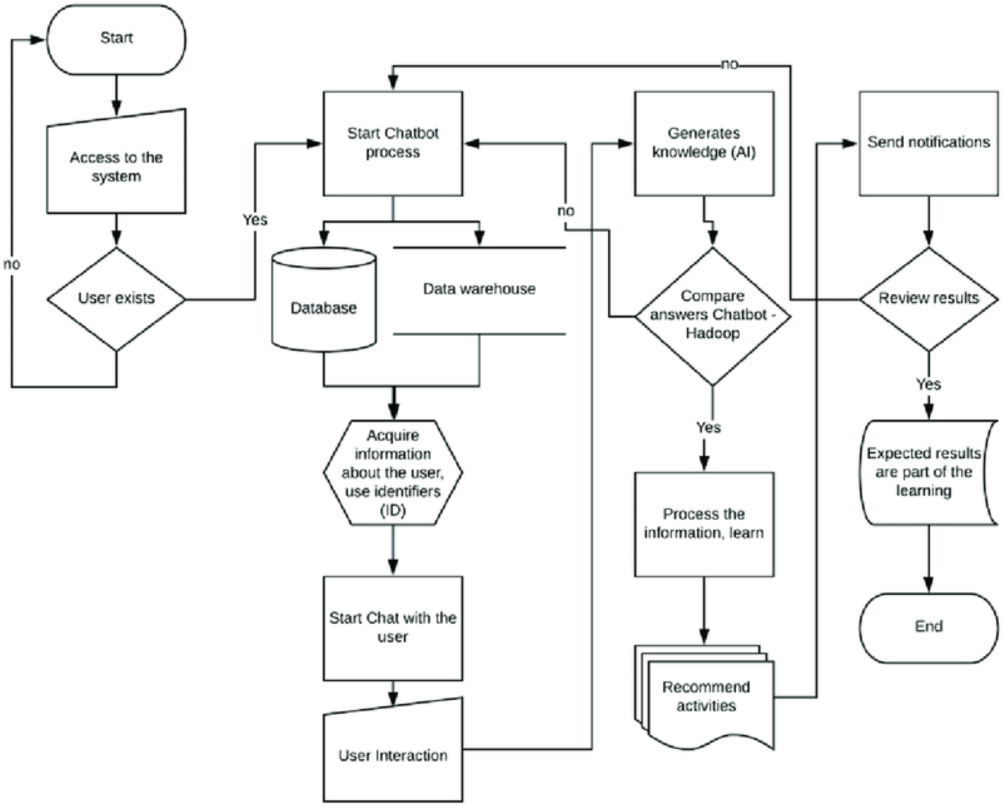


Figure 3: System Flow

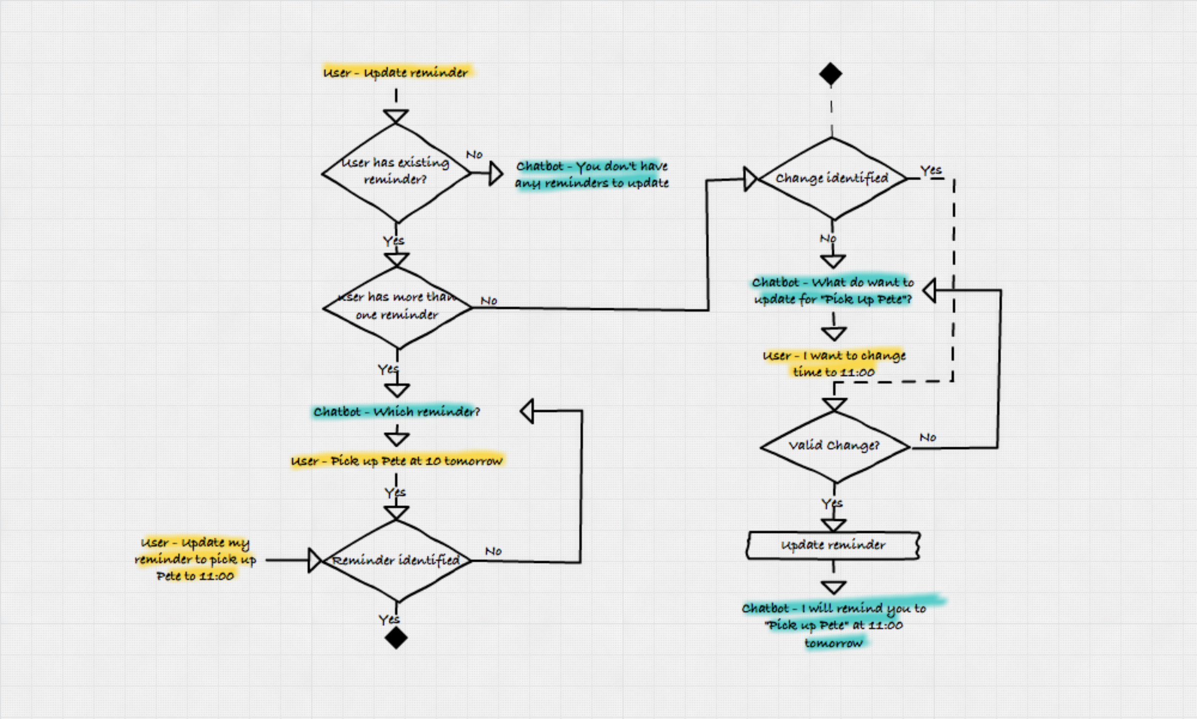

User - Update reminder
User has existing reminder?
No
Chatbot - You don't have any reminders to update
Yes
User has more than one reminder
No
Yes
Chatbot - Which reminder?
User - Pick up Pete at 10 tomorrow
Yes
User - Update my reminder to pick up Pete to 11:00
Reminder identified
No
Yes
Change identified
Yes
No
Chatbot - What do want to update for "Pick Up Pete"?
User - I want to change time to 11:00
Valid Change?
No
Yes
Update reminder
Chatbot - I will remind you to "Pick up Pete" at 11:00 tomorrow

Figure 4: User Flow

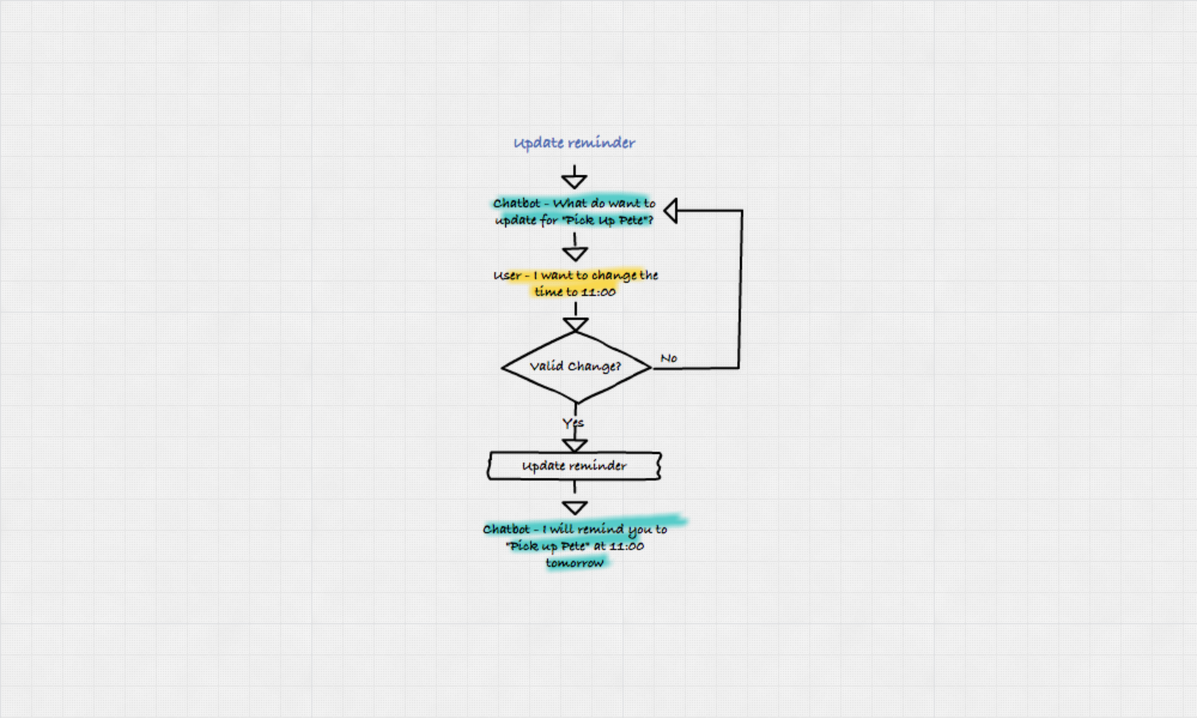

Update reminder
Chatbot - What do want to update for "Pick Up Pete"?
User - I want to change the time to 11:00
Valid Change?
No
Yes
Update reminder
Chatbot - I will remind you to "Pick up Pete" at 11:00 tomorrow

Figure 5: Design Flow

3.5 Design Selection

Effective communication and a great conversational experience are at the forefront when it comes to chatbot design. Chatbots are the technological bridges between collegees and consumers to provide faster and improved online experiences.

According to the research conducted by Grand view global chatbot market size will be $1.25 billion by 2025. With an enhanced focus on student engagement, chatbots in the form of a conversational interface (UI/UX) will be adopted by a huge number of collegees.

Why conversational UI/UX is important for chatbot design?

While building the chatbot user interface (UI), teh end user was kept in mind. They are your students and the fact that can't be denied is : students are judgmental. They have different motivations and look for emotional bonding everywhere, hence creating a first unforgettable impression becomes crucial. They also need speed.

This is how chatbot design gains importance and you should not ignore the key aspect : make it as human as possible. You have to be student-centric while building your chatbot UX design. So, that the impressive UI/UX positively impacts your college and student relationships.

A good chatbot design has a deeper impact on different college functions such as:

- **Consistent student experience** : Well-designed chatbots can help you to anticipate student needs and respond quickly in a personalized way that increases satisfaction. Hence a better student experience can help to establish a better brand.
- **Increased student engagement** : collegees realize how important it is to keep their students engaged. Intelligently designed chatbots to engage the students by understanding their intent and providing relevant answers.

- **Higher lead generation** : When your bots are able to provide guidance and tips during student interaction for successful conversion increases your lead generation.

Designing a chatbot is a mix of both art and science. The art is to understand your target students and their needs and the science is to convert those insights into small steps to deliver a frictionless student experience.

By going through the above principles of chatbot design you can haul your students by engaging them interactively. Thus, with a great chatbot design, you can enhance the overall student experience and build strong college-student relationships.

3.6 Implementation Plan

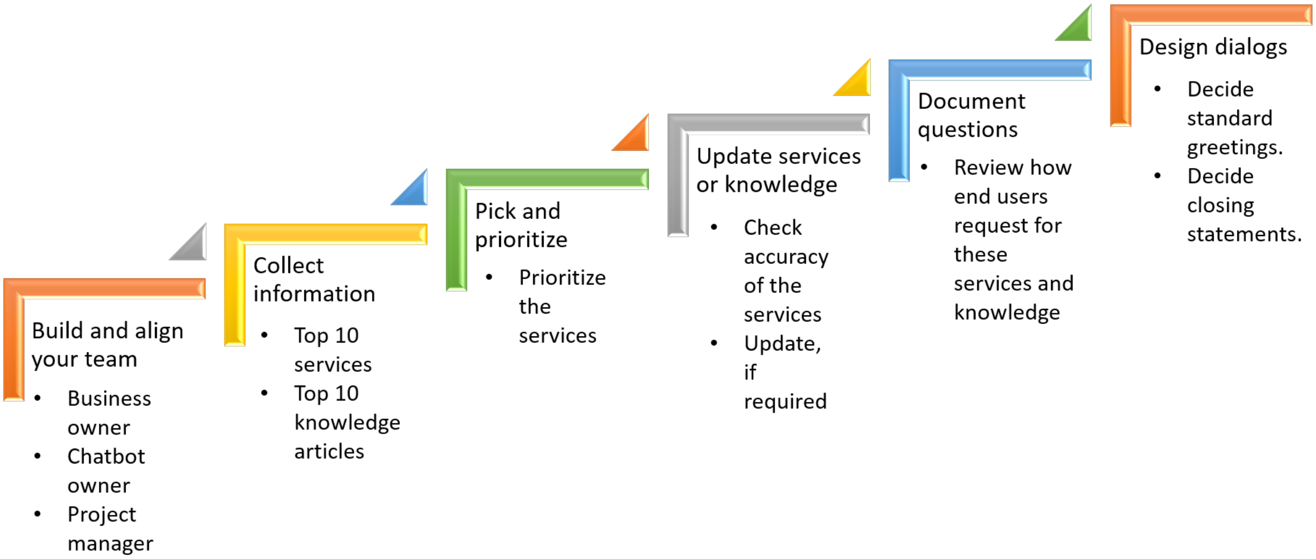

Build and align your team
• Business owner
• Chatbot owner
• Project manager
Collect information
• Top 10 services
• Top 10 knowledge articles
Pick and prioritize
• Prioritize the services
Update services or knowledge
• Check accuracy of the services
• Update, if required
Document questions
• Review how end users request for these services and knowledge
Design dialogs
• Decide standard greetings.
• Decide closing statements.

Figure 6: Implementation Plan

Csv file containing training data was loaded into main.py sript. Responses for each intent was created and stored as JSON file which was loaded into main.py script as responses dictionary. Every words in each question from csv data was turned to lower case. Tf-Idf Vectorizer for both monograms and bigrams was fit to the data. Label encoder object was fit to the intents. “predict_tag” function was created within which the input string will be transformed to Tf-idf score vector representation and fed as input to the Network and intent will be predicted by inverse transforming the encoded labels obtained from the predicted probablities returned by the network. The predicted intent will be returned. “start_chat” function was created within which a loop is defined where input will be taken from the user and predicted intent will be obtained by Calling “predict_tag” function on the user input. The predicted intent will be matched with the intents in responses dictionary and the corresponding response is diplayed to the user. “start_chat” function is invoked. So that the user can start interacting with the chatbot.

RESULTS ANALYSIS AND VALIDATION

4.1 Implementation

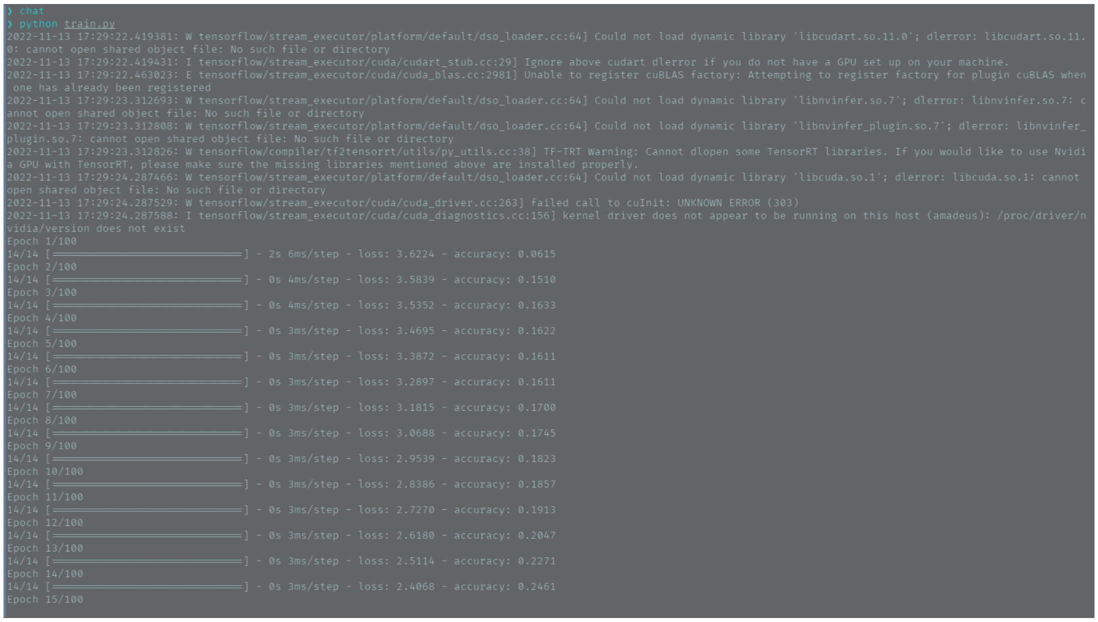
4.1.1 File Description

The project folder “project EduChat” contains the following files

- EduChat - model files (extract it and place it as child folder to the project folder).
- train.py - Python script by which the network was trained and saved.
- main.py - Python script which runs the Chatbot.
- data/data.csv - csv file on which the network was trained.

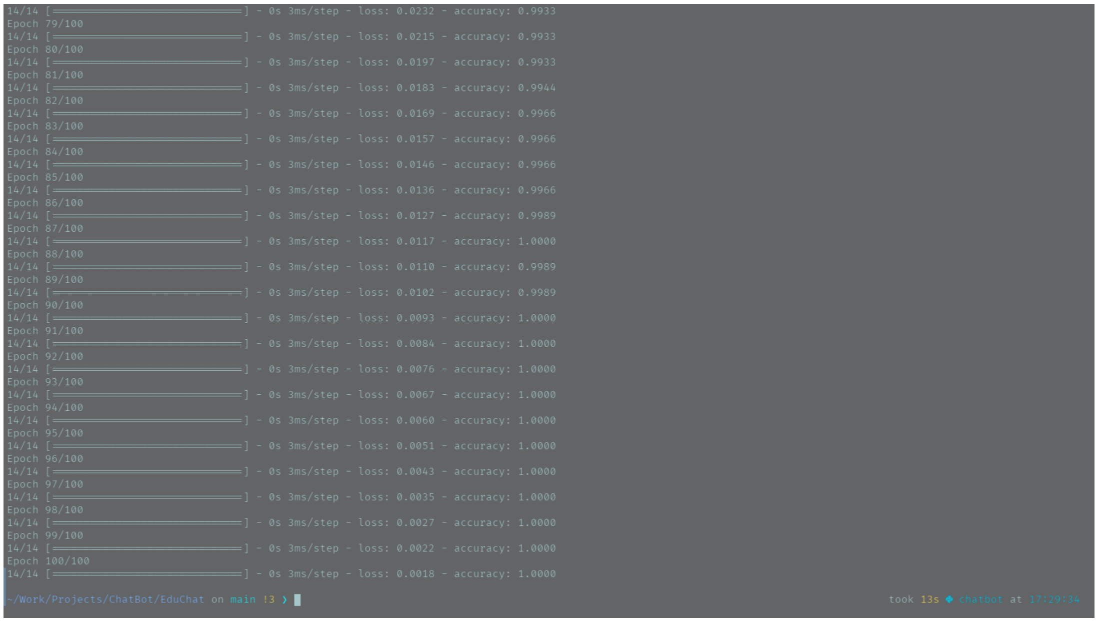
4.1.2 Training The Network

- CSV file containing the questions and their intents was created aftre scrpaing the website.
- The csv file was loaded in the training script.
- Every words in each question was turned to lower case.
- The data was turned into sparse matrix containing Tf-Idf scores of words in each question.
- The sparse matrix was converted into array format to be fed to the network for training.
- Intents were one hot encoded.
- The network was created and trained with Tf-Idf score array and one hot encoded intents array.
- The model was saved.



```
❯ chat
❯ python train.py
2022-11-13 17:29:22.419381: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcudart.so.11.0'; dlerror: libcudart.so.11.
0: cannot open shared object file: No such file or directory
2022-11-13 17:29:22.419431: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2022-11-13 17:29:22.463023: E tensorflow/stream_executor/cuda/cuda_blas.cc:2981] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when
 one has already been registered
2022-11-13 17:29:23.312693: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libnvinfer.so.7'; dlerror: libnvinfer.so.7: c
annot open shared object file: No such file or directory
2022-11-13 17:29:23.312808: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libnvinfer_plugin.so.7'; dlerror: libnvinfer_
plugin.so.7: cannot open shared object file: No such file or directory
2022-11-13 17:29:23.312826: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Cannot dlopen some TensorRT libraries. If you would like to use Nvidi
a GPU with TensorRT, please make sure the missing libraries mentioned above are installed properly.
2022-11-13 17:29:24.287466: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot
open shared object file: No such file or directory
2022-11-13 17:29:24.287529: W tensorflow/stream_executor/cuda/cuda_driver.cc:263] failed call to cuInit: UNKNOWN ERROR (303)
2022-11-13 17:29:24.287588: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (amadeus): /proc/driver/n
vidia/version does not exist
Epoch 1/100
14/14 [==============================] - 2s 6ms/step - loss: 3.6224 - accuracy: 0.0615
Epoch 2/100
14/14 [==============================] - 0s 4ms/step - loss: 3.5839 - accuracy: 0.1510
Epoch 3/100
14/14 [==============================] - 0s 4ms/step - loss: 3.5352 - accuracy: 0.1633
Epoch 4/100
14/14 [==============================] - 0s 3ms/step - loss: 3.4695 - accuracy: 0.1622
Epoch 5/100
14/14 [==============================] - 0s 3ms/step - loss: 3.3872 - accuracy: 0.1611
Epoch 6/100
14/14 [==============================] - 0s 3ms/step - loss: 3.2897 - accuracy: 0.1611
Epoch 7/100
14/14 [==============================] - 0s 3ms/step - loss: 3.1815 - accuracy: 0.1700
Epoch 8/100
14/14 [==============================] - 0s 3ms/step - loss: 3.0688 - accuracy: 0.1745
Epoch 9/100
14/14 [==============================] - 0s 3ms/step - loss: 2.9539 - accuracy: 0.1823
Epoch 10/100
14/14 [==============================] - 0s 3ms/step - loss: 2.8386 - accuracy: 0.1857
Epoch 11/100
14/14 [==============================] - 0s 3ms/step - loss: 2.7270 - accuracy: 0.1913
Epoch 12/100
14/14 [==============================] - 0s 3ms/step - loss: 2.6180 - accuracy: 0.2047
Epoch 13/100
14/14 [==============================] - 0s 3ms/step - loss: 2.5114 - accuracy: 0.2271
Epoch 14/100
14/14 [==============================] - 0s 3ms/step - loss: 2.4068 - accuracy: 0.2461
Epoch 15/100
```

Figure 7: Training the model - 1



```
14/14 [==============================] - 0s 3ms/step - loss: 0.0232 - accuracy: 0.9933
Epoch 79/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0215 - accuracy: 0.9933
Epoch 80/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0197 - accuracy: 0.9933
Epoch 81/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0183 - accuracy: 0.9944
Epoch 82/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0169 - accuracy: 0.9966
Epoch 83/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0157 - accuracy: 0.9966
Epoch 84/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0146 - accuracy: 0.9966
Epoch 85/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0136 - accuracy: 0.9966
Epoch 86/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0127 - accuracy: 0.9989
Epoch 87/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0117 - accuracy: 1.0000
Epoch 88/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0110 - accuracy: 0.9989
Epoch 89/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0102 - accuracy: 0.9989
Epoch 90/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0093 - accuracy: 1.0000
Epoch 91/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0084 - accuracy: 1.0000
Epoch 92/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0076 - accuracy: 1.0000
Epoch 93/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0067 - accuracy: 1.0000
Epoch 94/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0060 - accuracy: 1.0000
Epoch 95/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0051 - accuracy: 1.0000
Epoch 96/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0043 - accuracy: 1.0000
Epoch 97/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0035 - accuracy: 1.0000
Epoch 98/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0027 - accuracy: 1.0000
Epoch 99/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0022 - accuracy: 1.0000
Epoch 100/100
14/14 [==============================] - 0s 3ms/step - loss: 0.0018 - accuracy: 1.0000

~/Work/Projects/ChatBot/EduChat on main !3 ❯                                                                                          took 13s ◆ chatbot at 17:29:34
```

Figure 8: Training the model - 2

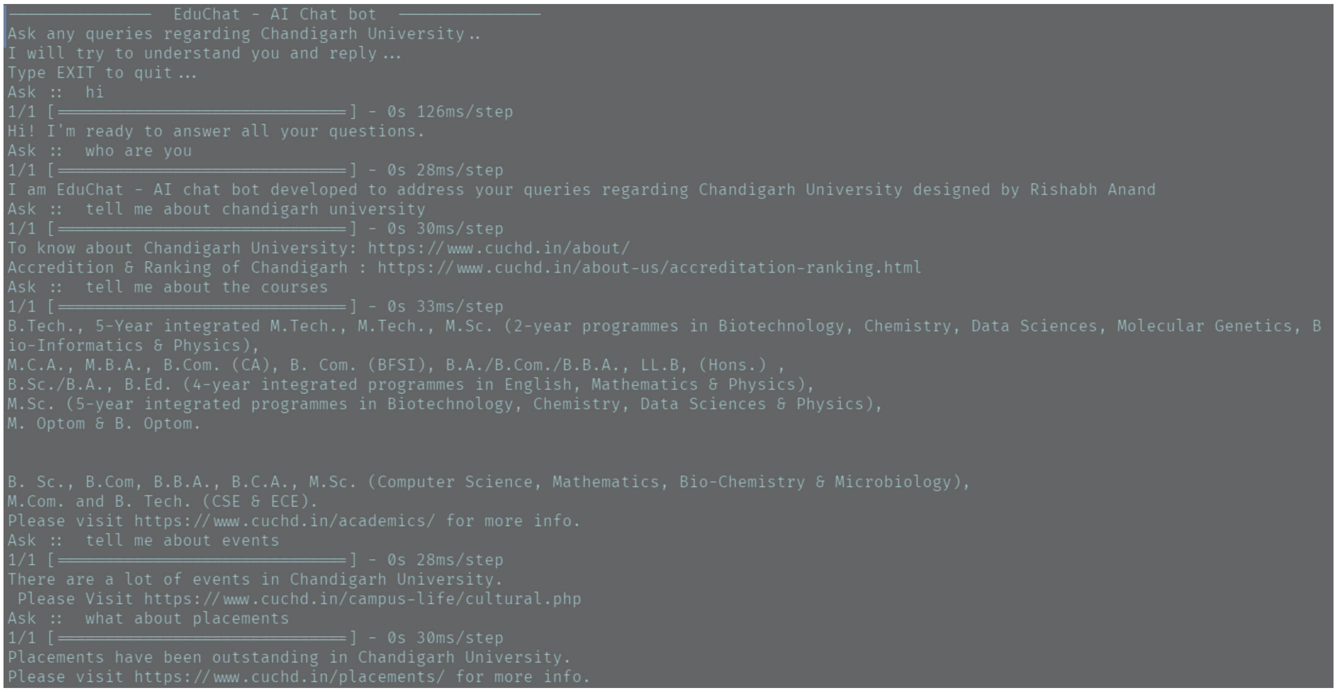
4.1.3 Model Architecture

- The network has input_shape = len(training_data_tfidf[0])
- The 1st dense layer contains 10 nodes.

- The second and third dense layers contains 8 nodes each.
- The fourth dense layer contains 6 nodes.
- The output dense layer have number of nodes = len(training_data_tags_dummy_encoded[0]) and “softmax” as activation function.
- The network was compiled with “rmsprop” as optimizer and “categorical_crossentropy” as loss function.

4.1.4 Functioning

- Csv file containing training data was loaded into main.py sript.
- Responses for each intent was created and stored as JSON file which was loaded into main.py script as responses dictionary.
- Every words in each question from csv data was turned to lower case.
- Tf-Idf Vectorizer for both monograms and bigrams was fit to the data.
- Label encoder object was fit to the intents.
- “predict_tag” function was created within which the input string will be transformed to Tf-idf score vector representation and fed as input to the
- Network and intent will be predicted by inverse transforming the encoded labels obtained from the predicted probablities returned by the network.
- The predicted intent will be returned.
- “start_chat” function was created within which a loop is defined where input will be taken from the user and predicted intent will be obtained by
- Calling “predict_tag” function on the user input. The predicted intent will be matched with the intents in responses dictionary and the
- corresponding response is diplayed to the user.
- “start_chat” function is invoked. So that the user can start interacting with the chatbot.



```
────────────────  EduChat - AI Chat bot  ───────────────
Ask any queries regarding Chandigarh University..
I will try to understand you and reply...
Type EXIT to quit...
Ask ::  hi
1/1 [==============================] - 0s 126ms/step
Hi! I'm ready to answer all your questions.
Ask ::  who are you
1/1 [==============================] - 0s 28ms/step
I am EduChat - AI chat bot developed to address your queries regarding Chandigarh University designed by Rishabh Anand
Ask ::  tell me about chandigarh university
1/1 [==============================] - 0s 30ms/step
To know about Chandigarh University: https://www.cuchd.in/about/
Accredition & Ranking of Chandigarh : https://www.cuchd.in/about-us/accreditation-ranking.html
Ask ::  tell me about the courses
1/1 [==============================] - 0s 33ms/step
B.Tech., 5-Year integrated M.Tech., M.Tech., M.Sc. (2-year programmes in Biotechnology, Chemistry, Data Sciences, Molecular Genetics, B
io-Informatics & Physics),
M.C.A., M.B.A., B.Com. (CA), B. Com. (BFSI), B.A./B.Com./B.B.A., LL.B, (Hons.) ,
B.Sc./B.A., B.Ed. (4-year integrated programmes in English, Mathematics & Physics),
M.Sc. (5-year integrated programmes in Biotechnology, Chemistry, Data Sciences & Physics),
M. Optom & B. Optom.


B. Sc., B.Com, B.B.A., B.C.A., M.Sc. (Computer Science, Mathematics, Bio-Chemistry & Microbiology),
M.Com. and B. Tech. (CSE & ECE).
Please visit https://www.cuchd.in/academics/ for more info.
Ask ::  tell me about events
1/1 [==============================] - 0s 28ms/step
There are a lot of events in Chandigarh University.
 Please Visit https://www.cuchd.in/campus-life/cultural.php
Ask ::  what about placements
1/1 [==============================] - 0s 30ms/step
Placements have been outstanding in Chandigarh University.
Please visit https://www.cuchd.in/placements/ for more info.
```

Figure 9: Sample Conversation

CONCLUSIONS

5.1 Conclusion

This project was a great learning experience for m. It also addresses the given issue greatly and provides a solution that is actually eliminating the problem to a good extent. With more time and effort put into the solution might land us somewhere that solves it completely.

This project was able to provide a solution to most of users queries and help its users keeps you their data bank up to date with the latest information. It was able to filter out the required information effectively and provide students with the specific personalized result, which were summarized, directly from the instutute's webpages.

5.2 Future Work

Something that I would like to add to this bot is multi-channel support for whatsapp, telegram, or other apps that Gen-Z uses so that they can find all the information wherever they are and won't have to rely on a separate app or interface. This would make the entire experience more seamless and easy to use and would potentially increase the user base.

Also, the email reader functions and the custom CSSs that people might opt in for will be added in later versions. Currently the function is a lot slower as I am using scraping up until now to excess the emails and provide users with the desired information. In future versions I would like to add an outlook or GMAIL API that users might opt into for faster access to theor data. Another reason, I didn't add this feature now was beacuse it takes alot to secure this personal data from all the users.

REFERENCES

- Designing an Educational Chatbot: A Case Study of CikguAIBot NA Nasharuddin, NM Sharef, EI Mansor... - 2021 Fifth ..., 2021 - ieeexplore.ieee.org ... Therefore, this paper introduces a chatbot for teaching artificial intelligence (AI) through Malay language, namely CikguAIBot. We emphasize on the development and evaluation design ...

- Opportunities and challenges in using AI chatbots in higher education S Yang, C Evans - Proceedings of the 2019 3rd International ..., 2019 - dl.acm.org ... in the areas of educational simulation, educational software' training, and helpdesk support, to support our study. In each case study, an AI virtual chatbot prototype has been proposed. ...

- Android based educational Chatbot for visually impaired people MN Kumar, PCL Chandar, AV Prasad... - 2016 IEEE ..., 2016 - ieeexplore.ieee.org ... In this paper we proposed the Educational Chatbot specifically for visually impaired people and it can also be used by normal people. This Chatbot uses the voice recognition for input ...

- A framework to implement AI-integrated chatbot in educational institutes B Debnath, A Agarwal - Journal of Student Research, 2019 - jsr.org ... of chatbot in educational institutes such as schools and colleges and to propose a chatbot ... This paper aims to provide an artificial intelligence(AI) integrated chatbot framework that can ...

- Artificial Intelligence (AI) chatbot as language learning medium: An inquiry N Haristiani - Journal of Physics: Conference Series, 2019 - iopscience.iop.org ... Chatbot for general purposes and for educational purposes have been developed [4,10,11]. However, despite chatbots' unlimited possibility to enhance language teaching and learning...

- Learning analytics for investigating the mind map-guided AI Chatbot approach in an EFL flipped speaking classroom CJ Lin, H Mubarok - Educational Technology & Society, 2021 - JSTOR . . . map-guided AI chatbot approach (MM-AI) promoted the students' English speaking performances more than did the conventional AI chatbot approach (C-AI). Moreover, the MMAI also . . .

- A system for educational and vocational guidance in Morocco: Chatbot E-Orientation O Zahour, A Eddaoui, H Ouchra, O Hourrane - Procedia Computer . . . , 2020 - Elsevier . . . These positive aspects of chatbots can be beneficial in the educational sector. They . . . our article, we set up a chatbot in the field of educational and professional guidance which is based . . .

- "AskBot": The AI Chatbot that Enhances the Learning Process K El Azhari, I Hilal, N Daoudi, R Ajhoun - International Conference on . . . , 2022 - Springer . . . The purpose of this article is to propose an architecture of an educational chatbot that can offload the teachers by answering in real time the massive and repetitive students' questions . . .

- Cognitive Services Applied as Student Support Service Chatbot for Educational Institution L Mrsic, T Mesic, M Balkovic - International Conference on Innovative . . . , 2020 - Springer . . . Domain specific bots like AI driven Support Center Automation . . . in Educational (EDU) environment like any kind of EDU Support System. This paper cover experience in building Chatbot . . .