Spring 2022
Department of Computer Science
Bahria University, Karachi Campus

ARTIFICIAL INTELLIGENCE LAB [CSL-411]

Project Name: E-commerce Chatbot

SEMESTER PROJECT

Maximum Marks: 30

Submission Due Date: 5th July, 2022

Sr.no	Name	Enrolment	Semester
01	Abdullah Abdul Wahid	02-134192-015	6-B
02	Fazeel Zafar	02-134192-010	6-B
03	Zoha Zehra	02-134192-058	6-B

Name	Designation
Ms. Amna Iftikhar	Course Instructor
Ms. Saba Imtiaz	Lab Engineer

Acknowledgement

I would like to express my special thanks of gratitude to my professor, Ms. Amna Iftikhar as well as our lab instructor, Ms. Saba Imtiaz, who gave me the golden opportunity to do this wonderful project on the topic **E-commerce Chatbot**, which also helped me in doing a lot of Research and I came to know about so many new things, I am really thankful to them.

Contents

1. Chapter 1

1.1. Problem Statement

A chatbot that is powered by artificial intelligence is referred to as an AI chatbot (AI). AI chatbots, in contrast to conventional chatbots, can comprehend user inquiries using natural language processing (NLP) and provide insightful responses. This enhances the general user experience. In this project, we will be developing an e-commerce chatbot, specifically designed for websites like daraz.pk which ensure the answers to user's questions related to their purchases, account, payment, tracking and more.

Users can learn about offers and advantages of making payments online from a bot. Chatbots on e-commerce websites provide answers to frequently asked questions, collect user evaluations, and handle challenging client inquiries. These are simply made to reduce the clutter a customer might run into when shopping.

2. Chapter 2

2.1. Literature Review

A. Nursetyo, D. R. I. M. Setiadi and E. R. Subhiyakto, "Smart Chatbot System for E-Commerce Assitance based on AIML," 2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 2018, pp. 641-645, doi: 10.1109/ISRITI.2018.8864349.

Abstract: Currently, the conventional market is starting to be replaced with many online markets. The tight online market competition demands excellent service from sellers to buyers, so many online stores provide full 24-hour service. This service certainly requires a lot of money if done manually. Chatbot can be used as a solution to automatically shop online. Then the bot must be able to provide an accurate and fast response. This study proposes an intelligent chatbot system based on Artificial Intelligence Markup Language (AIML) which can be used as an e-commerce assistant. This Chatbot is applied to the Telegram application. Input questions from users will be processed through three stages, namely, parsing, pattern matching and crawling data using AIML. In the AIML process, user requests are classified into three categories, namely general questions, calculations, and stock checks. Where the calculation request traps the order and payment process. Based on the results of 300 trials, the proposed method can answer all user requests well, with an average response time of 3.4 seconds.
Smart Chatbot System for E-Commerce Assitance based on AIML | IEEE Conference Publication | IEEE Xplore

A. R. D. B. Landim, A. M. Pereira, T. Vieira, E. de B. Costa, J. A. B. Moura, V. Wanick & Eirini Bazaki (2022) Chatbot design approaches for fashion E-commerce: an interdisciplinary review, International Journal of Fashion Design, Technology and Education, 15:2, 200-210, DOI: 10.1080/17543266.2021.1990417

ABSTRAC

Abstract: Chatbots can bring innovation in online assistance and communication with customers. Due to the growth of e-commerce, fashion brands have been adopting chatbots to provide personalised consumer experiences. Research in the area of chatbots for fashion e-commerce has addressed technological advancements and consumer behaviour, but little has been done on analysing chatbot features through a holistic point of view. The aim of this paper is to offer an interdisciplinary review through a comprehensive categorisation of recent studies on the theme and inform future research in the area. To achieve that, a theme-based literature review was carried out through the analysis of specialised research. The collected work was categorised addressing both computational and non-computational perspectives. The findings show that Deep Learning, recommendation systems, audio recognition and integration of chatbots with other fashion applications are a few design opportunities to be applied in both research and practice.

Full article: Chatbot design approaches for fashion E-commerce: an interdisciplinary review (tandfonline.com)

Khan, Mohammad. (2020). Development of An e-commerce Sales Chatbot. 173-176. 10.1109/HONET50430.2020.9322667.

Abstract: This paper presents the development of an e-commerce sales chatbot in order to provide customer support and increase sales. The system uses machine learning for natural language understanding. It is developed on an modular chatbot framework. The system has several components. First, a web based natural language training platform. Second, a microservice to classify input text and extract entities. Finally, a framework which routes user request to specific controller for processing and serves the response.

(PDF) Development of An e-commerce Sales Chatbot (researchgate.net)

3. Chapter 3

3.1. Methodology

There are many different kinds of chatbots accessible, but the following categories can be used to group them:

- Text-based chatbot: A text-based chatbot responds to user inquiries using a text-based interface.
- Voice-based chatbot: A voice-based chatbot, also known as a speech-based chatbot, responds to user inquiries using a human voice interface.

The design of chatbots primarily follows two methods, which are as follows:

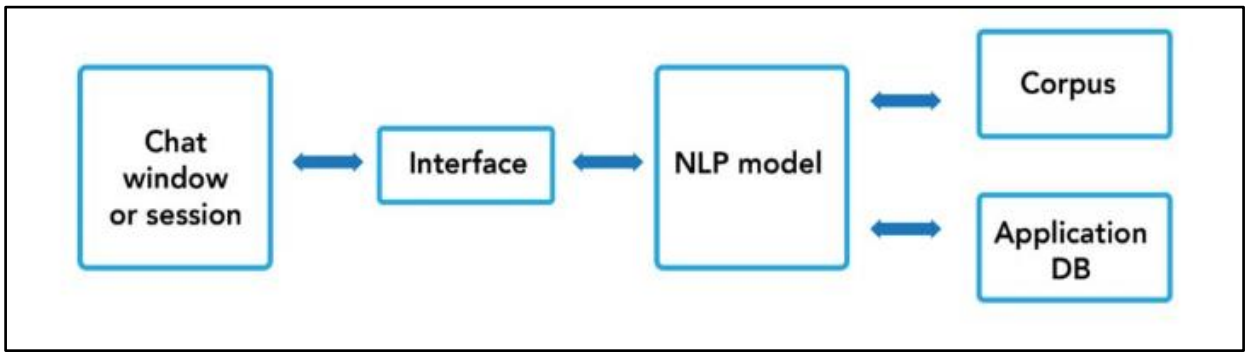
- In **a rule-based method**, a bot responds to queries in accordance with some pre-trained rules. The rules specified might range in complexity from very simple to highly complicated. Simple inquiries are handled by the bots, but complicated ones are not.
- **Self-learning bots**, which employ a few machine learning-based techniques, are unquestionably more effective than rule-based bots.

Depending on their complexity, there are many different types of chatbots accessible. A few of them may be broadly categorised as follows:

- **Traditional chatbot:** Chatbots with basic functionality and the capacity to preserve just system context are known as traditional chatbots. They are powered by automation and systems.
- **Current chatbot:** Chatbots nowadays are powered by two-way communication between the system and people. They possess the capacity to preserve both task and system contexts.
- **Future chatbot:** With automation at the system level, future chatbots can converse at many levels. They are capable of upholding the contexts of the system, the task, and the people. The introduction of master bots and eventually a bot OS is a possibility.

Typical chatbot architecture should consist of the following:

- Chat window/ session/ or front end application interface
- The deep learning model for Natural Language Processing [NLP]
- Corpus or training data for training the NLP model
- Application Database for processing actions to be performed by the chatbot



4. Chapter 4

4.4. Code Snippet

Import and Load data file

```
[ ]  import nltk
     from nltk.stem import WordNetLemmatizer
     lemmatizer = WordNetLemmatizer()
     import json
     import pickle

     import numpy as np
     from keras.models import Sequential
     from keras.layers import Dense, Activation, Dropout
     from keras.optimizers import SGD
     import random

     words=[]
     classes = []
     documents = []
     ignore_words = ['?', '!']
     data_file = open('intents.json').read()
     intents = json.loads(data_file)
```

Pre-process data

Here we iterate through the patterns and tokenize the sentence using nltk.word_tokenize() function and append each word in the words list. We also create a list of classes for our tags.

```
for intent in intents['intents']:
    for pattern in intent['patterns']:

        #tokenize each word
        w = nltk.word_tokenize(pattern)
        words.extend(w)
        #add documents in the corpus
        documents.append((w, intent['tag']))

        # add to our classes list
        if intent['tag'] not in classes:
            classes.append(intent['tag'])
```

lemmatize each word and remove duplicate words from the list

```
# lemmatize, lower each word and remove duplicates
words = [lemmatizer.lemmatize(w.lower()) for w in words if w not in ignore_words]
words = sorted(list(set(words)))
# sort classes
classes = sorted(list(set(classes)))
# documents = combination between patterns and intents
print (len(documents), "documents")
# classes = intents
print (len(classes), "classes", classes)
# words = all words, vocabulary
print (len(words), "unique lemmatized words", words)

pickle.dump(words,open('words.pkl','wb'))
pickle.dump(classes,open('classes.pkl','wb'))
```

```
104 documents
22 classes ['cancel', 'confirm', 'delay', 'delivery', 'goodbye', 'greeting', 'items', 'mobile app', 'more', 'order status', 'payments', 'personal', 'profile info', 'r
145 unique lemmatized words ["'m", "'s", '.', 'a', 'about', 'accept', 'account', 'address', 'agent', 'am', 'an', 'any', 'anyone', 'app', 'are', 'available', 'be', 'be
```

Create training and testing data

```
# create our training data
training = []
# create an empty array for our output
output_empty = [0] * len(classes)
# training set, bag of words for each sentence
for doc in documents:
    # initialize our bag of words
    bag = []
    # list of tokenized words for the pattern
    pattern_words = doc[0]
    # lemmatize each word - create base word, in attempt to represent related words
    pattern_words = [lemmatizer.lemmatize(word.lower()) for word in pattern_words]
    # create our bag of words array with 1, if word match found in current pattern
    for w in words:
        bag.append(1) if w in pattern_words else bag.append(0)

    # output is a '0' for each tag and '1' for current tag (for each pattern)
    output_row = list(output_empty)
    output_row[classes.index(doc[1])] = 1

    training.append([bag, output_row])
```

```
# shuffle our features and turn into np.array
random.shuffle(training)
training = np.array(training)
# create train and test lists. X - patterns, Y - intents
train_x = list(training[:,0])
train_y = list(training[:,1])
print("Training data created")
```

Create Model

```
[ ] # Create model - 3 layers. First layer 128 neurons, second layer 64 neurons and 3rd output layer contains number of neurons
    # equal to number of intents to predict output intent with softmax
    model = Sequential()
    model.add(Dense(128, input_shape=(len(train_x[0]),), activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(64, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(len(train_y[0]), activation='softmax'))

    # Compile model. Stochastic gradient descent with Nesterov accelerated gradient gives good results for this model
    sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
    model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])

    #fitting and saving the model
    hist = model.fit(np.array(train_x), np.array(train_y), epochs=200, batch_size=5, verbose=1)
    model.save('chatbot_model.h5', hist)

    print("model created")
```

```
Epoch 1/200
21/21 [==============================] - 0s 750us/step - loss: 3.1221 - accuracy: 0.0385
Epoch 2/200
21/21 [==============================] - 0s 750us/step - loss: 3.0018 - accuracy: 0.1154
Epoch 3/200
21/21 [==============================] - 0s 750us/step - loss: 2.9065 - accuracy: 0.1731
Epoch 4/200
21/21 [==============================] - 0s 800us/step - loss: 2.7892 - accuracy: 0.1923
Epoch 5/200
```

```
[ ] from keras.models import load_model
    model = load_model('chatbot_model.h5')
    import json
    import random
    intents = json.loads(open('intents.json').read())
    words = pickle.load(open('words.pkl','rb'))
    classes = pickle.load(open('classes.pkl','rb'))
```

```
[ ] def clean_up_sentence(sentence):
        # tokenize the pattern - split words into array
        sentence_words = nltk.word_tokenize(sentence)
        # stem each word - create short form for word
        sentence_words = [lemmatizer.lemmatize(word.lower()) for word in sentence_words]
        return sentence_words
    # return bag of words array: 0 or 1 for each word in the bag that exists in the sentence
```

```
[ ] def bow(sentence, words, show_details=True):
        # tokenize the pattern
        sentence_words = clean_up_sentence(sentence)
        # bag of words - matrix of N words, vocabulary matrix
        bag = [0]*len(words)
        for s in sentence_words:
            for i,w in enumerate(words):
                if w == s:
                    # assign 1 if current word is in the vocabulary position
                    bag[i] = 1
                    if show_details:
                        print ("found in bag: %s" % w)
        return(np.array(bag))
```

```
[ ]  def predict_class(sentence, model):
        # filter out predictions below a threshold
        p = bow(sentence, words,show_details=False)
        res = model.predict(np.array([p]))[0]
        ERROR_THRESHOLD = 0.25
        results = [[i,r] for i,r in enumerate(res) if r>ERROR_THRESHOLD]
        # sort by strength of probability
        results.sort(key=lambda x: x[1], reverse=True)
        return_list = []
        for r in results:
            return_list.append({"intent": classes[r[0]], "probability": str(r[1])})
        return return_list
```

```
[ ]  def getResponse(ints, intents_json):
        tag = ints[0]['intent']
        list_of_intents = intents_json['intents']
        for i in list_of_intents:
            if(i['tag']== tag):
                result = random.choice(i['responses'])
                break
        return result
```

```
[ ]  def chatbot_response(text):
        ints = predict_class(text, model)
        res = getResponse(ints, intents)
        return res
```

GUI Interface

```
[ ]  import tkinter
     from tkinter import *

     BG_GRAY = "#ABB2B9"
     BG_COLOR = "#c5f0e3"
     TEXT_COLOR = "#000000"

     # BG_GRAY = "#ABB2B9"
     # BG_COLOR = "#1c172a"
     # TEXT_COLOR = "#ffffff"


     FONT = "Helvetica 14"
     FONT_BOLD = "Helvetica 13 bold"


     def send(event):
         msg = EntryBox.get("1.0",'end-1c').strip()
         EntryBox.delete("0.0",END)
         if msg != '':
             ChatLog.config(state=NORMAL)
             ChatLog.insert(END, "You: " + msg + '\n\n')
             ChatLog.config(foreground="#000000", font=("Verdana", 12 ))

             res = chatbot_response(msg)
             ChatLog.insert(END, "Bot: " + res + '\n\n')
```

```
[ ]          ChatLog.config(state=DISABLED)
             ChatLog.yview(END)


     base = Tk()
     base.title("E-Commerce Chatbot")
     base.resizable(width=FALSE, height=FALSE)
     base.configure(width=800, height=800, bg=BG_COLOR)


     #Create Chat window
     ChatLog = Text(base, bd=0, bg=BG_COLOR, fg=TEXT_COLOR, font=FONT_BOLD)
     ChatLog.config(state=DISABLED)

     head_label = Label(base, bg=BG_COLOR, fg=TEXT_COLOR, text="Welcome to E-Commerce Chatbot", font=FONT_BOLD, pady=10)
     head_label.place(relwidth=1)

     line = Label(base, width=450, bg=BG_GRAY)
```

```
#Bind scrollbar to Chat window
scrollbar = Scrollbar(base, command=ChatLog.yview, cursor="heart")
ChatLog['yscrollcommand'] = scrollbar.set
ChatLog.focus()

#Create Button to send message
SendButton = Button(base, font=("Verdana", 12,'bold'), text="Send", width="12", height=15,
                    bd=0, bg="#ed9061", activebackground="#3c9d9b",fg='#ffffff',
                    command=lambda: send)

#Create the box to enter message
EntryBox = Text(base, bg="white",width="29", height="5", font="Arial", background="#dddddd")
EntryBox.focus()
EntryBox.bind("<Return>", send)
#EntryBox.bind("<Return>", send)

scrollbar.place(x=775,y=6, height=800)
line.place(x=0,y=35, height=1, width=770)
ChatLog.place(x=5,y=40, height=700, width=770)
EntryBox.place(x=0, y=740, height=60, width=600)
SendButton.place(x=600, y=740, height=60, width=175)
```

```
[ ]  base.mainloop()
```

Intents:

```
"intents": [
    {
      "tag": "greeting",
      "patterns": [
        "Hi",
        "Hey",
        "Is anyone there?",
        "Hello",
        "Good day"
      ],
      "responses": [
        "Hey :-) My name is Daz!",
        "Hello, thanks for visiting. My name is Daz!",
        "Hi there, My name is Daz!. What can I do for you?",
        "Hi there, My name is Daz! How can I help?"
      ]
    },
    {
      "tag": "personal",
      "patterns": [
        "How are you?",
        "How are you doing?",
        "How is your day?"
      ],
      "responses": [
        "I'm good, all's good, thanks. How about you?"
      ]
    },
            {
      "tag": "user response",
      "patterns": [
        "I'm good",
        "Im good",
        "Im doing good",
        "I am good",
        "I am okay"
      ],
      "responses": [
        "Great to hear you are doing good."
      ]
    },
    {
      "tag": "goodbye",
```

```
      "patterns": ["Bye", "See you later", "Goodbye"],
      "responses": [
        "See you later, thanks for visiting. Hope I was able to help!",
        "Have a nice day. Hope I was able to help!"
      ]
    },
    {
      "tag": "thanks",
      "patterns": ["Thanks", "Thank you", "That's helpful", "Thank's a lot!",
"thx", "thnks"],
      "responses": ["Happy to help!", "Any time!", "My pleasure"]
    },
    {
      "tag": "items",
      "patterns": [
        "Which items do you have?",
        "What kinds of items are there?",
        "What do you sell?",
        "What do you offer?",
        "What can I buy?"
      ],
      "responses": [
        "Search your preference in our flagship store's search bar to see all
available products. "
      ]
    },
    {
      "tag": "payments",
      "patterns": [
        "Do you take credit cards?",
        "Do you accept Mastercard?",
        "Can I pay with Cash?",
        "Are you cash only?",
        "What are your payment methods?",
        "How do I pay?"
      ],
      "responses": [
        "We accept VISA, Mastercard and Cash"
      ]
    },
    {
      "tag": "track",
      "patterns": [
        "How can I track my order",
        "I want to track my order",
```

```
        "Can I track my order",
        "Track order"
      ],
      "responses": [
        "Visit the order page, click on the specific order, select 'track my
order', and check the status"
      ]
    },
    {
      "tag": "delivery",
      "patterns": [
        "How long does delivery take?",
        "How long does shipping take?",
        "Please Tell me about my delivery",
        "When do I get my delivery?",
        "Why is my order not delivered yet"
      ],
      "responses": [
        "Delivery takes 2-4 days. Please bear with us!",
        "Shipping takes 2-4 days.  Please bear with us!"
      ]
    },
    {
      "tag": "confirm",
      "patterns": [
        "My payment was processed successfully but I didn't get any order
confirmation. What should I do?",
        "Payment was completed but no confirmation",
        "Order was not confirmed",
        "Money deducted but order is not confirmed"
      ],
      "responses": [
        "Daraz sends you an Email & SMS upon confirmation of your order. If it
requires manual confirmation, our team will contact you within 24 hours after
order placement. Delay in confirmation SMS may occur due to network error, you
may receive it with a delay.",
        "It takes upto 24 hours for confirmation, please bear with us! Type
Need more help for more assistance. "
      ]
    },
    {
      "tag": "delay",
      "patterns": [
        "Why is my order delayed? ",
        "Did my order get delayed?",
```

```
        "Why is my order not delivered yet?",
        "When do I get my delivery?"
      ],
      "responses": [
        "We are really sorry if there has been a delay in your order! If your
order is within delivery period, We recommend you to please wait during this
period since our Delivery Heroes make 3 attempts to deliver your order!",
        "If your order is past up to 3 days after delivery period, There may
be logistic issues causing a delay in delivery. Please type 'Common reasons
for delivery delay' to know more about this.",
        "We appreciate if you could wait for your items as most orders are
delivered successfully within this period.",
        "If your order is past more than 3 days, Since there may be unexpected
issues causing delivery delays, you can click on 'Need more Help' for further
assistance."
      ]
    },
    {
      "tag": "reasons",
      "patterns": [
        "Common reasons for delivery delay",
        "common reasons for delivery delay",
        "reasons for delay",
        "delivery delay"
      ],
      "responses": [
        "Reasons include Seller Sourcing Issues, Courier Issues, Cross Border
shipment delay, Wrong Address or Phone Number, and Unavailability of Customer.
"
      ]
    },
    {
      "tag": "more",
      "patterns": [
        "Need more help",
        "help me more",
        "can I talk to an agent",
        "can I call customer service",
        "customer support number",
        "how to contact customer service",
        "customer service number",
        "contact number for help",
        "helpline number",
        "How to become a seller",
        "How to contact a seller"
```

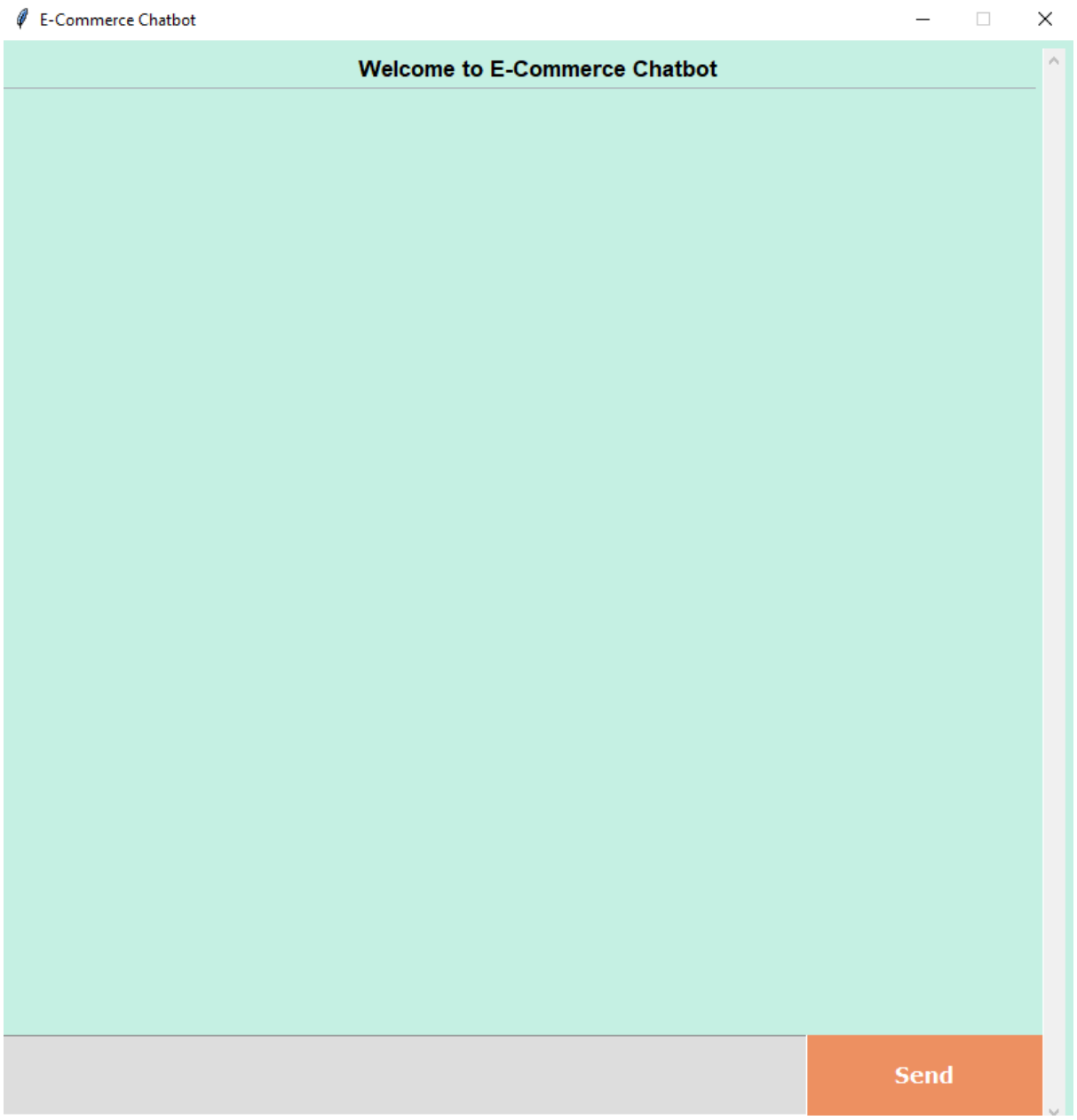
```
      ],
      "responses": [
        "Contact us for further information here: Phone: 021-111-132729.
Timings are from 09:00 AM to 05:00 PM from Monday to Saturday."
      ]
    },
    {
      "tag": "cancel",
      "patterns": [
        "Can I cancel my order?",
        "Why is my order cancelled?",
        "How do I can cancel my order",
        "Can I cancel an order"
      ],
      "responses": [
        "Order can only be cancelled within 7 days of placement. Digital goods
do not qualify for refund.",
        "Visit my orders page to check status of specific orders."
      ]
    },
    {
      "tag": "refund",
      "patterns": [
        "Can I refund an item.",
        "I want to refund an item",
        "can I refund my order",
        "Are refunds available"
      ],
      "responses": [
        "Refund can only be issued within 7 days of placement. Digital goods
do not qualify for refund.",
        "Visit my orders page to check for specific orders."
      ]
    },
    {
      "tag": "refund status",
      "patterns": [
        "Why is the status Refunded when it's not credited?",
        "No refund even though status is refunded",
        "No refund when status says refunded",
        "I did not receive my refund money",
        "Refund money not received"
      ],
      "responses": [
```

```
        "Please be patient as refunds take upto 30 days to receive into bank.
"
      ]
    },
    {
      "tag": "refund",
      "patterns": [
        "Can I refund an item.",
        "I want to refund an item",
        "can I refund my order",
        "Are refunds available"
      ],
      "responses": [
        "Refund can only be issued within 7 days of placement. Digital goods
do not qualify for refund.",
        "Visit my orders page to check for specific orders."
      ]
    },
    {
      "tag": "return",
      "patterns": [
        "How long do return items take",
        "When will my items be returned",
        "When can I get a refund for return items?",
        "Are refunds available for return?"
      ],
      "responses": [
        "Return takes 6 to 10 business days. There are 2 parts to the return
process. In the first part, we check the items received from you to verify the
issue highlighted in your return claim. Once we've verified and accepted your
return, we initiate the refund process which takes 1 business day for Daraz
wallet and 2-10 business days for bank or card. "
      ]
    },
    {
      "tag": "withdraw refunds",
      "patterns": [
        "How to withdraw refunds from wallet?",
        "Withdraw funds",
        "Can I withdraw refunds"
      ],
      "responses": [
        "Withdrawing refunds from your wallet is very easy. Please follow the
steps below: Step 1: Click on the 'Wallet' icon on the top right corner of the
Daraz App, or click here. Step 2: Click on the 'Balance Details' Icon. Step 3:
```

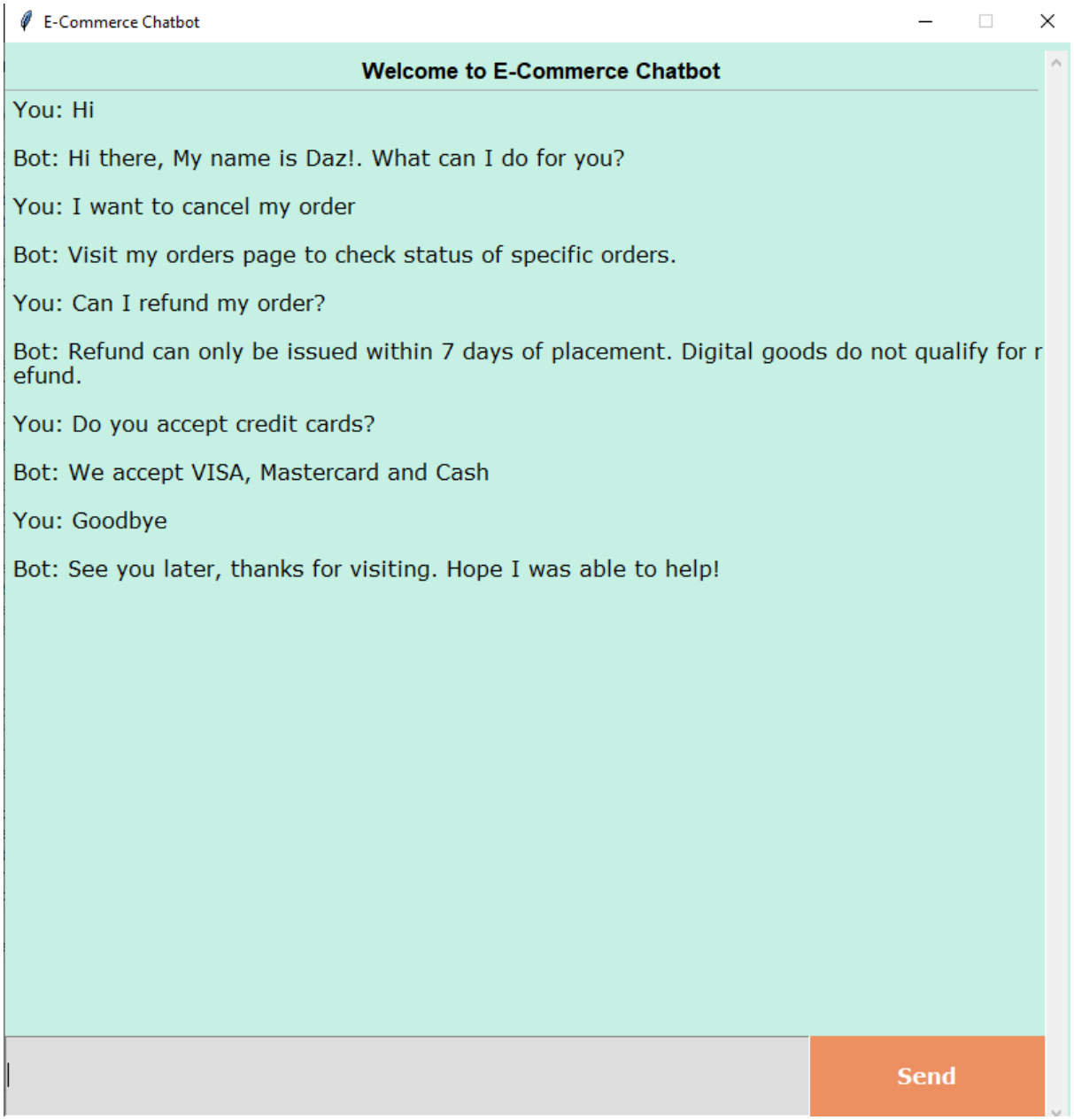
```
Click on 'Withdraw' button. Step 4: Enter your bank account details and click
'Withdraw'. Step 5: You will receive a verification email, click confirm to
initiate your withdrawal request."
      ]
    },
    {
      "tag": "order status",
      "patterns": [
        "What is my order status",
        "I want to know my return status",
        "How to return status"
      ],
      "responses": [
        "Please visit the My Orders page for a list of your confirmed orders."
      ]
    },
    {
      "tag": "use voucher",
      "patterns": [
        "How to use a voucher?",
        "Can I use a voucher?",
        "How to use a voucher?"
      ],
      "responses": [
        "You can add a voucher by clicking on My Cart > Check Out > Enter
Voucher Code > APPLY. "
      ]
    },
        {
      "tag": "profile info",
      "patterns": [
        "How can I change my profile information",
        "I want to change my password",
        "I want to change my phone number",
        "I want to change my address",
        "I want to Reset my password",
        "I want to delete my account",
        "delete my account"
      ],
      "responses": [
        "You can easily add or change your account details by following the
steps below: Step 1: Click on 'Account', Step 2: Click on 'Manage my account'
from the icon, Step 3: You change or edit your name, address, email address,
mobile number, etc., Step 4: Fill in the required details, and click on Save.
```

```
Note: You can also change your delivery address from the Checkout page before
proceeding to pay."
      ]
    },
        {
      "tag": "mobile app",
      "patterns": [
        "Can I download a mobile app for this?",
        "Mobile app",
        "Download mobile app"
      ],
      "responses": [
        "You can download the Daraz Mobile App from both IOS App Store, and
Google Play Store. "
```

Output:




E-Commerce Chatbot
Welcome to E-Commerce Chatbot
You: Hi
Bot: Hi there, My name is Daz!. What can I do for you?
Send


E-Commerce Chatbot
Welcome to E-Commerce Chatbot
You: Hi
Bot: Hi there, My name is Daz!. What can I do for you?
You: I want to cancel my order
Bot: Visit my orders page to check status of specific orders.
You: Can I refund my order?
Bot: Refund can only be issued within 7 days of placement. Digital goods do not qualify for r
efund.
You: Do you accept credit cards?
Bot: We accept VISA, Mastercard and Cash
You: Goodbye
Bot: See you later, thanks for visiting. Hope I was able to help!
Send

5. Chapter 6

5.1. Conclusion

With a chatbot, your organization can easily offer high-quality support and conflict resolution any time of day, and for a large quantity of customers simultaneously. According to me, chatbots and other intelligent digital assistants are fundamentally altering the corporate landscape. Numerous chatbot development platforms are available for a variety of businesses, including e-commerce, retail, banking, leisure, travel, healthcare, and so on. On messaging applications, chatbots may reach a larger audience and are more efficient than people. They could soon become an effective tool for acquiring information.

5.2. Future Work

- **Voice Recognition Support:** We can implement voice recognition support for this bot for people with language or literacy barrier. A voice chatbot is an artificial intelligence (AI) conversational communication tool that can record, decode, and analyse vocal input from the speaker to answer in a manner that is akin to natural language. A voice AI chatbot allows users to communicate with it via voice commands and hear contextually appropriate replies.

- **Call Scheduling:** We can implement calls between customer and support assistant where needed, using our chatbot. AI makes ensuring that all established business rules are followed by schedules. AI will take your decisions into account while optimizing the timetable using previous data. For instance, if an employee has a history of working shifts related to a specific client, it can indicate that they are a great fit for that customer.

- **Online Assistant Support Connectivity:** Chatbots improve the support experience by providing rapid replies and automated responses to help requests, while live chat allows human-to-human connection and adds empathy to support chats. To give your customers a top-notch customer care experience, live chat and chatbots collaborate.

- **Order Management:** Compared to human workforces, chatbots can distribute the shipment and delivery information more quickly. Additionally, chatbots might quickly link internally, enabling simple communications amongst staff members, and effectively handle client inquiries in comparison to others who do not have the technology.

6. Chapter 6

6.1. References

AI Chatbot | Complete Guide to build your AI Chatbot with NLP in Python (analyticsvidhya.com)

Introduction to Chatbot | Artificial Intelligence Chatbot Tutorial (mygreatlearning.com)

8 Steps to Build a Successful Chatbot Strategy (revechat.com)

Chatbot Implementation in 8 Steps - Helpful Expert Tips for Getting Started - MoreThanDigital

Schedule.cc - AI-powered Appointment Scheduling Software - Agile CRM Blog

Voice AI, Voice Chatbots, Voicebots: The Future of Contact Centres (verloop.io)

MarekLani/E-Commerce-Chat-Bot: This repo contains description of Microsoft and SmartBase common project focused on development of chat bot feature for SmartBase's e-commerce solution. Bot is built on top of Bot Framework using Node.js (github.com)