USER-FRIENDLY ENHANCED MACHINE LEARNING-BASED RAILWAY MANAGEMENT SYSTEM

Weerasooriya W.K.M

(IT18085822)

B.Sc. (Hons) Degree in Information Technology Specializing in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology Sri Lanka

October 2021

USER-FRIENDLY ENHANCED MACHINE LEARNING-BASED RAILWAY MANAGEMENT SYSTEM

Weerasooriya W.K.M

(IT18085822)

Dissertation submitted in partial fulfilment of the requirements for the Bachelor of Science (Honours) Degree in Information Technology Specializing in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology Sri Lanka

October 2021

DECLARATION PAGE OF THE CANDIDATE & SUPERVISOR

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to Sri Lanka institute of Information Technology, the nonexclusive right to reproduce and distribute my dissertation, in whole or in part print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: …..... Date: ….....

[Weerasooriya W.K.M.]

The above candidate has carried out research for the bachelor's degree dissertation under my supervision.

Signature of the supervisor: …..... Date: ….....

[Ms. Manori Gamage]

ABSTRACT

There are several places to visit. In Sri Lanka, to be precise. The majority of the locations are well-known and well-liked. There are many wonderful spots to visit in Sri Lanka that are less well-known. It makes no difference whether the locations are well-known or not. People may not have adequate information on the destinations they will visit. One way to get information on a location is to utilize Google search. However, there are occasions when some information is missing or when information about less well-known locations is unavailable. Using the user's input, the chatbot feature assists the user in locating tourist attractions in Sri Lanka. This tool aids in the identification of precise information about locations based on the keywords entered by the user. The chatbot receives the user's input and filters the keywords, then finds and matches the term against the previously trained data set and analyses the outcome. Finally, use normal language to represent the outcomes.

The function's main goal is to search for details using user input and display the results in natural language. Identify the input's particular terms, search the result, represent it, and preserve the input and response for future use. This function entails the use of deep learning and sampling algorithms in the dataset's training.

ACKNOWLEDGEMENT

I'd want to express my heartfelt gratitude to my supervisor, Ms. Manori Gamage, for her unwavering support, advice, and encouragement throughout the project, as well as her vast experience and patience. In addition, I'd like to express my gratitude to Ms. Suriyaa Kumari, my co-supervisor, for her steadfast support and guidance. I'd like to thank the Sri Lanka Institute of Information Technology (SLIIT) for giving undergraduates the opportunity to study through a meaningful research project, as well as my assessment panel for suggesting fresh ideas and pointing out flaws. The support and supervision given by the team of academic experts in the Comprehensive Design and Analysis Project (CDAP) module, directed by Dr. Janaka Wijekoon, is critical to the research's success.

Mr. Premachandra Fernando, previous Senior Officer of Sri Lanka's Colombo Fort Railway Station, and the Colombo Fort Railway Station employees deserve special recognition. Also, I'd like to express my gratitude to Mr. Sanath Wijerathna, Senior Engineer at Sri Lanka Railways' Railway Maintenance Yard in Maradana, for aiding us in reaching Mr. Premachandra Fernando by providing additional information.

Finally, I'd want to express my gratitude to the members of our study group, as well as my family and coworkers, for their constant support and innovative suggestions. Also, a big thank you to everyone who helped make our User-Friendly improved railway administration system research project a success, even if it was just with a word.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
IT	Information Technology
NLP	Natural Language Processing
ALG	AI-based learning algorithms
GUI	Graphical User Interface
API	Application Programming Interface
NLU	Natural Language Understanding
FFNN	Feed-Forward Neural Network

1. INTRODUCTION

1.1. Background & Literature Survey

1.1.1. Background

Sri Lanka's third-largest source of revenue is tourism. The railway is also the most popular and pleasurable mode of transportation. Many foreign visitors and local passengers choose to travel by rail since it is the cheapest and most pleasant mode of public transportation in Sri Lanka. In Sri Lanka, there are many hidden gems and undiscovered, gorgeous spots that tourists are unaware of. Our tourist business will thrive if we can publicize those locations. Unlike in the past, our trains now feature several amenities. As a result, taking the train saves money while also providing a comfortable and safe mode of transportation.

The railway department and the tourism sector have formed an alliance. Details concerning trains and train facilities, as well as popular and less popular places that may be reached by train, will be of tremendous assistance to the tourism sector and the railway department's development. If sufficient attention is paid to this, it will have a positive impact not just on those two sides, but also on a number of ancillary sectors that rely on the railway, as well as tourists.

Tourists are kept away from some destinations due to a lack of knowledge about them. It is extremely beneficial to users if there is a technique that can give users with travel destinations in a certain location as well as more information about those destinations.

People nowadays use Information Technology (IT) to help them in all aspects of their lives. The majority of countries rely on information technology to improve their government services. They have made those services more user-friendly as a result of this. These nations employed information technology to enhance their railway and tourist industries. However, IT is no longer widely used in Sri Lankan railway services.

The railway sector can be enhanced if information technology is employed for the benefit of passengers in the railway industry.

Sri Lanka Railways serve a large portion of the country. They mostly have seven distinct lines. Passengers can travel to a variety of locations utilizing such train lines.

The Main Line runs via Nuwara Eliya and Badulla, while the Mathale Line runs through Kandy and Mathale. Hikkaduwa, Galle, and Kataragama are all part of the Coastal Line. As a result, all highways in Sri Lanka encompass the country's diverse, attractive, and historically significant locations. Those lines span a wide area of Sri Lanka. The historical significance, climatic conditions, and geographical circumstances of such sites change from one to the next. The weather and topography conditions in the Northern province, for example, are vastly different from those in the Southern or Central regions. As a result, the majority of international travelers are unaware of Sri Lanka's travel location.

most popular way to learn about places is to conduct a Google search. However, there will be a scarcity of information, and the information will be unclear. It is highly beneficial for consumers if they can receive precise information using natural language.

Traditional methods for obtaining location and railway information have several limitations. When it comes to uncommon areas, it's very time-consuming. As the number of sites grows, the amount of resources required grows as well. Furthermore, traditional approaches cannot provide complete accuracy and dependability. As a result, it's a good idea to work with AI-based intelligent chatbot solutions that can manage enormous amounts of location data or uncommon places. Because they can efficiently manage and analyse massive data utilizing automated frameworks such as the Hadoop framework, they can overcome the problem of data imbalance.

Furthermore, predicting location information using various algorithms and comparing their accuracies to standard prediction techniques is difficult. As a result, several algorithms may be used to make automated predictions [1]. Furthermore, after evaluating their accuracies, the most effective method for prediction may be chosen.

However, rather of manual functions, an automated railway system might be beneficial for the railway department and the Sri Lanka tourist authority to utilize in a more systematic manner. Because they can automatically identify patterns of traveler behavior that will occur in the future [2], the government can grant steps to improve the facilities.

Tourism is a major source of revenue for many governments throughout the world. As a result, all countries are attempting to grow their tourist industries in a variety of ways. In addition, because railways are the primary mode of transportation in many nations, all governments strive to improve their railway services.

We looked for similar projects that provided answers to those difficulties before processing the project, and we discovered several systems. In Sri Lanka, there is no specific mechanism for dealing with the aforementioned concerns.

1.1.2. Literature Survey

Railbot is a telegram-based chatbot that allows users to receive train information by entering the train's id. [3] The Indian railway service is the subject of this chatbot function. This chatbot is used in railway API for collecting information about trains when a train PNR id is entered and the chatbot provides the train data and delay time. [3] This chatbot is just for Telegram members and can only provide information on trains that have a PNR number.. Its shown in the figure 1. for a better user experience, recommended implementing this with other platforms like Whatsapp, Messenger, etc.

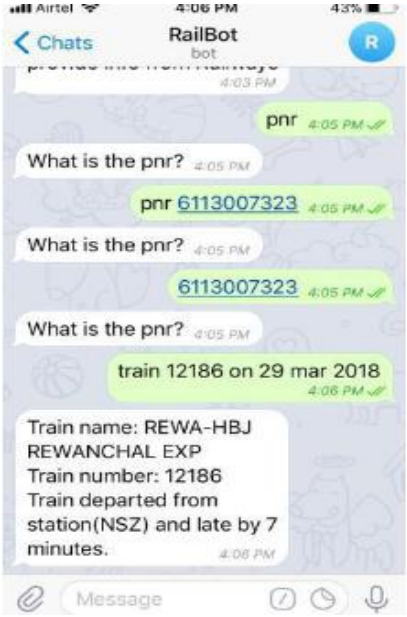


Figure 1:output of railbot function

RailwayBot is another railway-related chatbot research project. This chatbot offers useful train information such as seat availability, ticket purchasing, and ticket cancellation [4]. This is a chatbot for Facebook Messenger. As a result, users can utilize this system in conjunction with Facebook. When a user poses a query, the chatbot will respond with a link to a solution. Figure 2 depicts the situation. In compared to other chatbots, this one provides more railway-related information. It works well as a railway chatbot. It is preferable to offer information directly rather than via a link. It's a great idea for this chatbot [4].

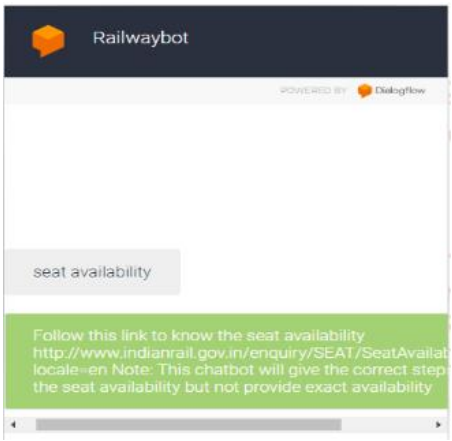


Figure 2: output of railwaybot

Railway Buddy is a chatbot that assists customers in locating information about railways such as train status, trains between stations, train time at the station, train live arrival time at the station, coach position, PNR status, and train timetable. Its shown in the figure 3. It gives information on the Indian railway system. [5]

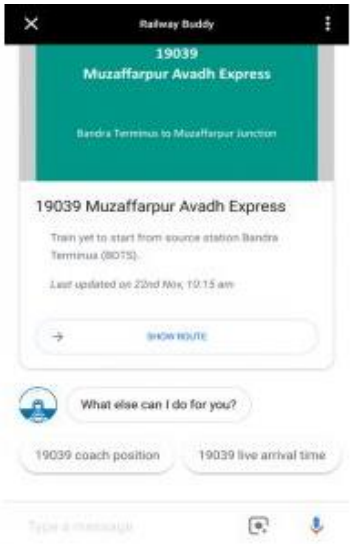


Figure 3:ouput of railwaybuddy

Flow.ai is a chatbot that can help us discover our location. When we tell the chatbot where we are, it locates the nearest retail location [6]. The research focuses mostly on the service-providing firm. Customers may use a chatbot to discover the company's nearest branch. Basically, this chatbot is for WhatsApp or Facebook Messenger. Users may put in the address or share their location with the chatbot, and the chatbot will recognize their location and direct them to the nearest store. It is advantageous for the user to discover the proper place if this feature gives the user with the nearest location rather than the address. Figure 4 depicts the situation. It may also give information about the store; this is a useful function for the chatbot. However, this chatbot is not intended for railway passengers.

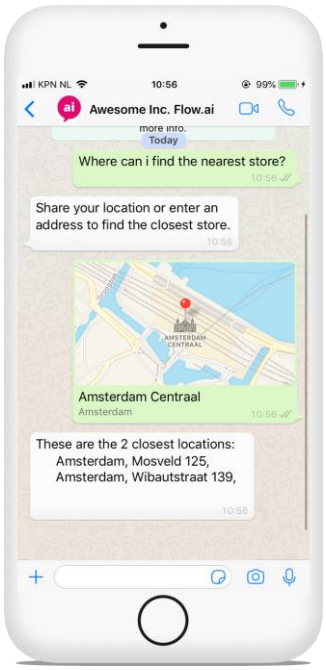


Figure 4:output of flowIo

In the another research. They provide a chatbot for found bus information . The user requests the bus number that runs from the source to the destination from the Chatbot. All of the intentions with bus numbers and routes make up the Chatbot. The user input is then compared to the strings in the intents file, which makes up the Chatbot database. If the input values match the database values, the Chatbot will deliver the outcome by providing the bus number. The user is then presented the output.[7]. this is a different thing when comparing others. chatbot using to the common transport like the bus is a good thing for future. if they can implement this chatbot not only for the bus number but also provide some information about the location, it will be great experience for the users.

Some related projects are very complex. As an example there is a system with the IoT. The system [8] collects user voice through an Amazon Alexa-enabled device and analyzes it using natural language processing (NLP) techniques to figure out what the user is trying to say or ask and answers appropriately. The echo device is the speech recognition engine, the interpretation software is their Alexa Skill, and they've built a knowledge base through data mining and scraping, then combined it with machine learning and AI algorithms to build an intelligent travel engine. It's developed with MongoDB, MySQL, Elasticsearch, Neural Networks, and a successful Restricted

Boltzmann machine implementation for collaborative filtering[8]. this is a very rare and complex chat system compare with others. voice is used as an input is a very good user experience. for the chatbot need special devices. it is one of the cons of the chat bot.

In the another research they build chatbot base robot to interact with users. In creating the Chatbot-based attendance and location guidance system, the suggested prototype includes AI-based learning algorithms (ALGs). The camera analyzes a person's face using Harr Cascade filters and verifies the database, then outputs as a recognized or unknown person via a speaker and display screen. The suggested system's most important features are the attendance database, query, and responding queries from users. The suggested system has the benefit of combining both location identification and facial recognition into one module. The greatest aspect of the proposed prototype is the extra functionality of addressing the inquiry in two distinct languages (Tamil and English). [9]

Some of researchers use some image processing part for the their system. With the use of deep image processing algorithms, the system can locate ancient sites. Archaeologists and other users can interact with the platform by submitting information about archaeological sites using the community platform. To keep the knowledge base constant, deep text summarizing techniques are used. In addition, the platform includes a conversational agent that allows users to quickly and easily obtain the information they need. By employing augmented reality methods, the picture visualization allows virtual visualization of ancient sites. this is a different thing in comparison with others. combine with image processing and chatbot function is good for the modern technologies[10]

1.2. Research Gap

in our system, we implement the deep-learning based chatbot for getting traveling places information for the user.

Questions regarding the location can be submitted by users. Then, after loading the model, determine which question best matches the user input. The model delivered the sample answer as a response if the user input matched the sample question by more than 75%.

Sri Lanka has a diverse range of tourist attractions. Initially, just visiting locations in the Anuradhapura District was considered. We want to visit sites all throughout Sri Lanka in the future as part of our work.

there are some similarities and dissimilarities between our chatbot and other related works

there are some similarities and dissimilarities between our chatbot and other related works

- Paper 1 - telegram-based chatbot call railbot
- Paper 2 - chatbot research related to the railway service Railwaybot
- Paper 3- Flow.ai chatbot which can find the location
- Paper 4 - Interactive Transport Enquiry with AI Chatbot
- Paper 5 - Intelligent travel chatbot for predictive recommendation in echo platform
- Paper 6 - Chatbot - Attendance and Location Guidance system (ALGs)
- Paper 7 - Historical Places & Monuments Identification System

Table 1Different between related work

Feature	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5	Paper 6	Paper 7	Railbot
Information about railways	Yes	Yes	No	No	No	No	No	Yes

Location information	No	No	Yes	No	No	Yes	Yes	Yes
Find location facility	NO	No	Yes	No	No	No	No	Yes
Voice input facility	No	No	No	No	Yes	No	No	No
Image input facility	No	No	No	No	No	Yes	Yes	No
Transport information	Yes	Yes	No	Yes	No	No	No	Yes
Tourism support facility	Yes	Yes	No	Yes	Yes	No	Yes	Yes

1.3. Research Problem

Sri Lanka has a diverse range of tourist attractions. Some areas are more well-known and well-liked than others. However, there are some stunning locations that are less well-known. Whether the location is well-known or not, tourists should be aware of the information available. It is quite beneficial to the user in gaining an understanding of the location and deciding whether or not to visit. Google search may sometimes be useful in locating information about a location. However, it is ineffective for a variety of reasons. Because the traveller has no clue where the destination is, they might obtain worthless information about a site that they will not be able to visit.. As well as sometime some unpopular places information is not clear in the google and lot of time wasting for that process.

We have a way in the percent to locate such type of information. Posting on social media is one of the most common techniques. Although we can obtain information through social media, this procedure takes a long time to complete, and we cannot guarantee that we will receive just correct responses or replies from that time period.

Another approach to obtain geographic information is to use the Google search engine. However, it is not a suitable approach for obtaining that level of information since we cannot always comprehend or locate all of the facts at the same time.

As a result, we want a rapid and responsive platform to obtain information on trains and tourist destinations in Sri Lanka. It is preferable for the user to get information in their native language.

1.4. Research Objectives

The chatbot is being developed to address this issue. Users may ask the chatbot any queries they have about visiting destinations, and the chatbot will respond with straightforward information.

The main goal is to create a chatbot program that allows users to obtain information in natural language. As previously stated, there is no reliable technique for quickly obtaining information on trip destinations. It has resulted in the railway and tourist sectors collapsing. The main goal of the chatbot is to give information about vacation destinations in natural language for a better comprehension.

The chatbot will take the user's input, filter the keywords, and apply a deep learning model to find the best appropriate answer. Then he deconstructs the solution and expresses it in simple terms. It is a fairly short operation, and it can solve the difficulties listed.

There are some particular objectives that must be completed in order to achieve the overall goal.

1. Gather the information

To begin, the author must gather particular information on the visited location. The Department of Archaeology is the finest location to learn about Sri Lanka's historical sites.

2. Prepare the data for analysis

The data collection must be pre-processed before the training model can be created. Text data is not used in the deep learning model. As a result, we must break down the entire text into little chunks, such as words.

3. Make a data set for training.

Following preprocessing, you must generate a data set that the machine can comprehend. The main goal of the data set is to take user input and utilize the training data set to come up with a better solution.

4. Represent the outcome

Fill in the blanks using genuine language. As a consequence, the user can readily comprehend the outcome, and if they have any doubts, they may re-ask the question.

2. METHODOLOGY

2.1. Methodology

The chatbot component of the user-friendly enhanced machine learning-based railway management system project is one of the key components of the project. It is an automated platform that has been developed to raise the user attractiveness of the railway and tourist industries in Sri Lanka. The government has a key necessity to improve the appeal of Sri Lanka's tourism destination with a more user-friendly system using contemporary technology, because the tourism sector has a significant influence on society both economically and socially. The chatbot component has been built as a primary component of the user-friendly improved machine learning-based railway administration system as a response to their main demand and to acquire more facts about going destinations.

This component, on the other hand, has been created to provide information on trip destinations in four key areas.

- Collect data
- Data preprocessing
- Train the model
- Represent the output

However, these outputs assist the tourist in planning their vacation, making it simple to decide what to observe and the worth of the locations. The chatbot's results are disseminated using a web-based application known as a user-friendly enhanced machine learning-based railway management system. As a result, the target consumer for this user-friendly enhanced machine learning-based railway management system online application is travelers who wish to learn more about Sri Lanka's vacation destinations. Users can also read information about the place in their own language. The chatbot's high-level system design is shown in Figure 5. The first stage, as shown in the diagram, is data gathering, in which the system consumes information on tourism destinations in Sri Lanka. The data is then cleaned and standardized using data preprocessing. Then, using the location data gathered in the previous phase, train a deep learning model called a feed-forward neural network.

The trained model can then recognize the keywords in the user input and reflect the correct response to the user's query. The user-friendly enhanced machine learning-based railway management system Graphical User Interface (GUI) of the front-end program retrieves the anticipated answer and other user inputs from a database.

Some model extremities are extensively considered in order to achieve the most realistic results.

- Accuracy
- Performance
- Efficiency
- Effectiveness

- Security

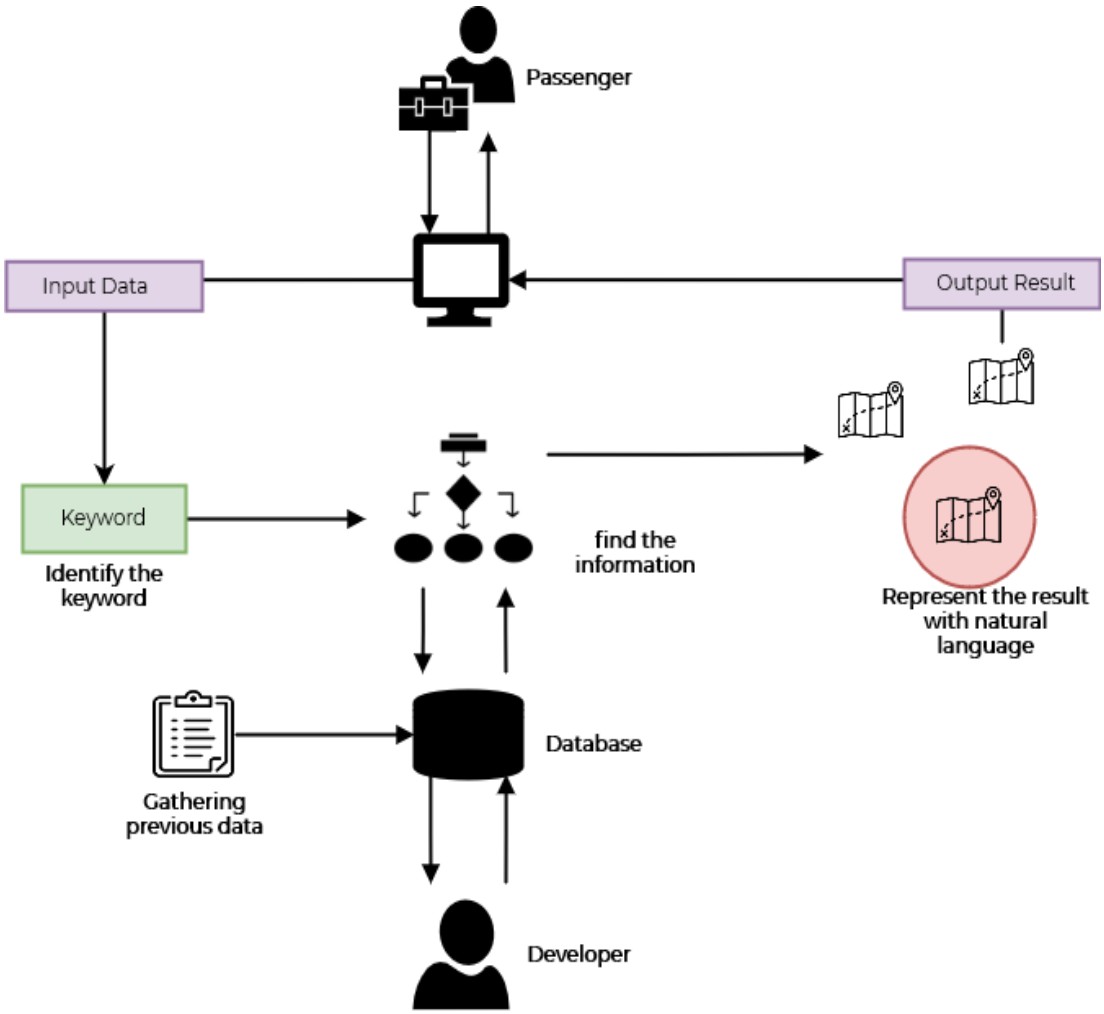


Figure 5: The chatbot's high-level system design

As a summery, the user may ask the chatbot any questions about the location they are visiting, and the chatbot will respond in natural language. Natural language is a form of communication. It is really beneficial to the user in terms of improved understanding. The keywords are identified using the Natural Language Processing (NLP) technique. The user's inquiries can be recognized by the chatbot. Similar

questions can be asked in a variety of ways. Those queries will be recognized by the chat bot as being the same. Then he finds and represents the solution. The response is solely for the user's query.

The methodology can be divided into the four steps

- Feasibility studies
- Functions
- Application
- Resources

2.1.1 Feasibility study

Since the user-friendly enhanced machine learning-based railway management system has been developed as a solution to problems gained for travellers when traveling on the train and who find information about tourism places. it is a huge requirement accomplishment from the railway department as well as the tourism industry of Sri Lanka.

As a result, after evaluating the project's resources, development, time, and cost, the creation and maintenance of the application is determined to be viable. Although the application's hosting costs money, the application's initial features are free; nevertheless, future updates and further functionality will cost money. As a result, the budget is deemed acceptable. Initially, the Chabot would be aimed at the Anuradhapura district's tourism attractions. However, according to the project plan, the system implementation date is also achievable.

Furthermore, the dataset and other fundamental information resources are easily accessible, data gathering is practical, and the programming skills required are reasonable. In addition, during the initial phase, the tools and technology utilized for application development are free.

the feasibility study can be divide into four

- Economical

- Technical
- Operational
- Organizational

2.1.1.1. Economical

Costs are a major consideration when developing Deep Learning-based applications. Because the method is based on mechanics, the data set utilized here must be handled with caution. The dataset determines the overall accuracy and user-friendliness of the system. As a result, creating a highly accurate dataset will take more time and money. To put up a full database, we had to recruit the aid of travel agents while compiling the dataset. When hosting the system, I had to purchase a live domain and server. Consider a few aspects of the server while purchasing it. The PHP and Python languages must be executed on the server, and the amount of disk space, speed, and bandwidth available must all be considered.

This method is very beneficial for international travellers. Because the majority of international travellers are unfamiliar with Sri Lanka (SL) and its travel destinations. As a result, such software is critical for improving train service and the tourism sector. As a result, the finest service provider must take accuracy into account in that system and must host for.

2.1.1.2. Technical

The Railwaytour project is being developed as a web application. Backend development languages, frontend development languages, and frameworks were utilized to create the online application. The developed machine learning components were written in Python.

- **Frontend Languages**
  - Html
  - CSS

- **Backend Languages**
  - Php
  - Python
  - JavaScript
- Framework
  - Bootstrap
  - Flask

2.1.1.3. Operational

The Railwaytour online application is primarily aimed towards international visitors to Sri Lanka. Local passengers can also use the online application to schedule their journeys. As a result, that system used both domestic and international travelers for their travel needs. As a result, the software's smoothness and user-friendliness must be considered. Machine learning-based functions were utilized to construct that system. As a result, this system may be brought even closer to the user. Because, based on past data, machine learning models can anticipate appropriate suggestions for passengers. In addition, JavaScript and AJAX were used to create the system. As a result, most functions may be done without refreshing the website. The majority of the work is done on the client's end. This is extremely beneficial to the entire system's enhanced smoothness.

2.1.1.4. Organizational

Following the development of the Railwaytour system, a study was performed to assess the system's value. Many people have suggested this system as an essential system, according to the user poll. Local and international travellers were the majority of those who used the method. Following the launch, the system should be improved in accordance with the demands and thoughts of those passengers. The majority of users believe that these kind of solutions work better as a web application. Because consumers did not utilize the system on a regular basis. As a result, they believe that developing this system as a web application rather than a phone application is more appropriate. Some passengers, on the other hand,

recommended that the system be set up as a mobile application. As a result, that application may be built as a mobile application in the future. As a result, the system should be further improved in accordance with the users' thoughts and demands.

2.1.2 Functions

- Get user input from the front end
- Transfer the input value to model via API
- Find the answer related to user input
- Transfer the answer value to front end via API
- Display the answer

2.1.3 Application literacy

The user-friendly enhanced machine learning-based railway management system and the chatbot function are both written in English. However, future versions will include Sinhala and Tamil languages. Because the program is web-based, a reliable internet connection is required to use it. As a result, an internet connection is required to get data from servers and to access essential Application Programming Interface (API) in order to maintain the application's functioning.

2.1.4 Resources

2.1.4.1. Software requirements

1. **PyCharm**

PyCharm is a JetBrains-developed Integrated Development Environment (IDE) for computer programming, specifically for the Python programming language. It has grown in popularity as a result of its efficient and intelligent help in the development of software with nearly all required tools. However, it has a graphical debugger, code analysis, an integrated unit tester, version control system integration, and Anaconda Navigator support. PyCharm, on the other hand, was utilized in the user-friendly

improved machine learning-based railway management system's backend development[11].

2. **Visual Studio Code (VS Code)**

Microsoft created Visual Studio Code, which is available for Windows, Linux, and macOS. It combines a source code editor's clarity with powerful developer tools like IntelliSense code completion and debugging [12]. Basically, it's an editor that speeds up the edit-build-debug cycle in the present context, giving developers more time to execute their ideas. VS Code, on the other hand, was utilized to create the project's frontend.

3. **Python**

Python is a general-purpose programming language that is object-oriented, high-level, and high-level [13]. This can be put to a lot of different uses. Modules and packages, exceptions, dynamic data types, and classes are all consolidated in this application. The Python interpreter, as well as a large standard library, are freely available in source and binary formats. The backend implementation of this project, however, was done using Python 3.8.

4. **Flask**

A micro web framework developed in Python is known as Flask. Certain tools and libraries are not required. It also works without a database abstraction layer, form validation, or other components such as third-party libraries that offer basic and common functionality. In addition, flask extensions for object-relational mappers, numerous authentication mechanisms, form validation, and common framework-related utilities are being created. As a result, in the user-friendly enhanced machine learning-based railway management system, Flask is utilized as the server-side Python framework.[14]

5. **MySQL**

As the database utilized in the project is MySQL, the main storage unit is MySQL.

6. **HTML and CSS**

Frontend technologies are those that use HTML and CSS. They were used to produce papers for the user-friendly improved machine learning-based railway management system on the World Wide Web.

7. **Bootstrap4**

Bootstrap4 is the most recent version, and it supports frontend UI design. It is the most used HTML, CSS, and JavaScript framework for creating responsive and efficient websites. It is completely free to use.

8. **Anaconda Navigator and Jupyter Notebook**
9.

Anaconda Navigator is a desktop or graphical user interface (GUI) provided by the Anaconda distribution for launching software applications. It also handles conda packages, channels, and environments flexibly rather than utilizing instructions in a command line. The navigator, on the other hand, is available for Windows, macOS, and Linux. Furthermore, Jupyter Notebook is a software program that can be accessed through the Anaconda Navigator and has been used to develop and test road accident prediction models in a user-friendly enhanced machine learning-based railway management system[15].

2.1.4.2. Hardware requirements

- For the backend, a server computer with a lot of computing power is required.
- Any computer (PC) or laptop with an internet connection can be used.

2.1.4.3. Memory requirements

Because the user-friendly enhanced machine learning-based railway management system web application requires ML and contains a huge quantity of data, the backend requires more RAM, but the frontend does not.

Back End

- Ram- 8 GB minimum
- Storage – 2GB

Front End

- Ram- 1GB
- Storage- 500mb

2.2. Commercialization Aspect of the Product

By definition, commercialization is the process of presenting new products or services to the market [16]. As a result, when examining the chatbot component, which is one of the key components of the user-friendly enhanced machine learning-based railway management system, it emphasizes the commercialization element, which encourages high-quality manufacturing, distribution, and customer service. Because some geographical information is quite scarce, the chatbot component is particularly creative. Typically, a railway-related system will only provide one type of service, such as ticket booking or railway schedule information, but the user-friendly enhanced machine learning-based railway management system will provide four main services, including trip planning with train schedules, smart chatbot for location information, seat prediction system, and places category prediction system. As a result, the user may utilize the system to get a general concept of their trip locations and time allotment for their journey. Users may also utilize the system to determine which seat position is ideal for them. More crucially, the system's end users are both travellers and railway users, resulting in the system's contribution to the development of both the tourist and

railway industries. As a result, this component is extremely inventive, increasing its economic worth.

As a result, it boosts the tourism sector while also lowering the cost of travel, which is extremely beneficial to the country's economy. This platform, in particular, makes it simple to go by rail, which is a significant aspect from a societal standpoint. Indirectly, it benefits a variety of other industries, therefore boosting the economy.

- People who provide accommodation can improve their economy.
- People who provide food and beverage can improve their economy.
- People who provide recreation and entertainment can improve their economy.
- People who provide travel services can improve their economy
- Railway related industries are improved

The user-friendly enhanced machine learning-based railway management system may be provided as a collection of packages with premium features in terms of commercialization. The chatbot component of this initial package version includes capabilities such as retrieving all relevant information about the sought region in a timely manner and retrieving station information about the relevant area.

Then, as part of the premium package, elements like trip planning and seat prediction can be included. The railway automation system, on the other hand, has a primary page that displays all railway details and location information, which is critical for users.

As a result, together with the premium package, certain advertising about railway and tourist industry development techniques, economic aid from growing railway and tourism industry, and other relevant data may be presented. New and expanded features, as well as a premium package that may be purchased, may be released in the future. It will also bring value to the existing user-friendly machine learning-based railway management system.

2.3. Testing and Implementation

2.3.1. Testing

The testing step of the software development life cycle is crucial. If mistakes are made during the testing process, the system may fail. As a result, testing is a crucial part of software development since it uncovers problems or defects before the program is delivered, ensuring its quality [17]. A fully tested program also verifies a software application's dependability, quality, and high performance. There are numerous testing steps in this implementation.

- Design testing
- Unit testing
- Module testing
- Component testing
- Integration testing
- User acceptance testing

Because the chatbot component is a subpart of the railway automation system, system testing is performed after all of the application's subparts have been combined.

- Design testing
  - During the chatbot component's design phase, design testing was carried out. It is determined whether the component design meets the needed functional criteria. The needed UI designs are also put to the test to see if they provide the optimum user experience. During the design testing phase, the flow of UIs and user controls are also examined.
- Unit testing
  - Unit testing is done when a piece of code, a unit, or a module has been implemented in the component and is done separately. The primary objective of this testing is to guarantee that the implemented software unit is functional. If there were any bugs, they were addressed at this phase.

- Module testing
    - Certain subcomponents of the chatbot component are checked during module testing. In addition, methods, functions, and connection controls are all examined in this section.
- Component testing and Integration testing
    - After integrating all subcomponents such as API functions, model creation, and database connection, the entire chatbot component is tested. If any bugs were discovered, they were fixed during the same testing phase. The functionality of the integrated component, in particular, is tested to ensure that it performs as expected.
- User acceptance testing
    - During this phase, the entire component is examined to see if it meets the product's business value. The main aim is to meet the customer's needs as well as their degree of acceptance of the functional component.

The following are some of the test cases that are run. The implementation of test cases is split into two portions here.

- Frontend testing
- Backend testing

Frontend testing

Table 2 Test case 1

Test case ID	0001
Scenario	Test chatbot hide when close button click
Test steps	Find close button Click close button
Test data	-
Expected result	Close chat
Actual result	Close chat

Status	Pass

Table 3 Test case 2

Test case ID	0002
Scenario	Test chatbot not hide when click outside area
Test steps	Click outside of the chat
Test data	-
Expected result	Not close chat
Actual result	Not close chat
Status	Pass

Table 4 Test case 3

Test case ID	0003
Scenario	Test sending message align to the right side
Test steps	Type message on the input area Click send button
Test data	hi
Expected result	Message display right side on the display
Actual result	Message display right side on the display
Status	Pass

Table 5 Test case 5

Test case ID	0004
Scenario	Test receiving message align to the left side
Test steps	Type message on the input area Click send button Wait for response
Test data	hi

Expected result	Receiving message display left side on the display
Actual result	Receiving message display left side on the display
Status	Pass

Table 6 Test case 6

Test case ID	0005
Scenario	Test chat remove when page reload
Test steps	Reload the page
Test data	-
Expected result	Chatbot reset when page reload
Actual result	Chatbot reset when page reload
Status	Pass

Backend testing

Table 7Test case 7

Test case ID	0006
Scenario	Test bot response for various input
Test steps	Ask same question in various way
Test data	Who create sigiriya? Who made sigiriya? Who built sigiriya?
Expected result	Sigiriya was built by King Kashyapa
Actual result	Sigiriya was built by King Kashyapa
Status	Pass

2.3.2. Implementation

As the second component of the user-friendly enhanced machine learning-based railway management system, a chatbot has been created. As a result, the backend is built utilizing the Python programming language and the Flask server-side Python framework. In addition, MySQL was used as the database to record anticipated outcomes as well as key functional data. The frontend, on the other hand, is made up of HTML, and Bootsrap4. Furthermore, HTML and CSS have been utilized to produce documents for the World Wide Web. Because travellers and railway passengers are the target users of the user-friendly improved machine learning-based railway management system, the UIs are designed to provide the best user experience with user-friendly features. In addition, the chatbot's model was created using DL and the Python programming language. In addition, Jupiter Notebook was used in the early stages of the model development process. Furthermore, RESTful APIs for communication between the server and the application have been created.

Model implementation

Data Collection

For the chatbot, I gathered information from Wikipedia and tourism sources. I gathered all relevant information, such as the founder's name, the year, the significance of the location, and so on. After gathering data, construct example question patterns and sample responses according to the location's information. The example questions and answers were then saved in a JSON file. There were several intentions in the file. There is a tag, a question pattern, and a sample response for each intent.

Data pre-processing

used some NLP techniques to preprocess the data.

- Tokenization
- Stemming
- Bag of Word

Tokenization

Tokenization is the process of breaking down a phrase, sentence, paragraph, or complete text into smaller pieces, such as individual words or concepts. A token is the name given to each of these smaller components [18]. The JSON file's questions were tokenized and saved as an array.

Stemming

Shortening a word to its stem, which is connected to suffixes and prefixes, or the origins of words, is known as stemming. Natural Language Understanding (NLU) and Natural Language Processing (NLP) both need stemming [19].

Bag of Word

The textual representation depicts the creation of words in the JSON file throughout this procedure. Ignore the grammatical intricacies and word order in favor of the word count. Because it does not regard the sequence or structure of words, the document is referred to as a batch of words (figure 6). The model simply considers if a document contains well-known terms. [20].

```
# stem each word
sentence_words = [stem(word) for word in tokenized_sentence]
# initialize bag with 0 for each word
bag = np.zeros(len(words), dtype=np.float32)
for idx, w in enumerate(words):
    if w in sentence_words:
        bag[idx] = 1

return bag
```

Figure 6:bag of word function

Feature selection

Sri Lanka has a diverse range of tourist attractions. In practice, gathering all of the facts will take a long time. As part of the research, the region was narrowed down and location information was gathered in the Anuradhapura district.

Anuradhapura offers a variety of travel options, including lake, historical, and traditional. We take into account all categories and gather data.

Model generation, training, and testing

In order to cover all chatbot processes, the model was created using the DL method. First, a pre-processed data intent file and the specified areas indicated in the feature selection section were used to construct the chatbot's model. The Feed-Forward Neural Network technique was used to create the model (FFNN). Through the function evaluated from x, the intermediate computations needed to calculate f, and finally the output y, these models are known as feed-forward. As the mode output returns to itself, there are no feedback loops. Recurrent neural networks [21] are immune neural networks that have expanded to include feedback. Because it maximizes Mean Square Error, it's not a good idea to directly maximize that number of accurate outputs.

FFNN with two hidden layers was utilized in the chatbot component (figure 7). Get the bag of words as an input for the fully linked main layer in the model. The inputs are subsequently processed by the hidden layers, which anticipate the answer as the output. There is an activation layer for each layer. applied SoftMax to each output's obtain probability

```
class NeuralNet(nn.Module):
    def __init__(self, input_size, hidden_size, num_classes):
        super(NeuralNet, self).__init__()
        self.l1 = nn.Linear(input_size, hidden_size)
        self.l2 = nn.Linear(hidden_size, hidden_size)
        self.l3 = nn.Linear(hidden_size, num_classes)
        self.relu = nn.ReLU()
    def forward(self, x):
        out = self.l1(x)
        out = self.relu(out)
        out = self.l2(out)
        out = self.relu(out)
        out = self.l3(out)
        # no activation and no softmax at the end
        return out
```

Figure 7:FFNN Model

The model will subsequently be loaded and the training process will begin. The learning rate was set to 0.001 and the number of loop epochs was set at 1000.

Finally, load the trained model and get the output for the user input. In that process, user inputs were gone through the feed-forward net and check matching percentage with each intent (Fugure 8). If found intent with more the 75% match with the user input, model return the response.

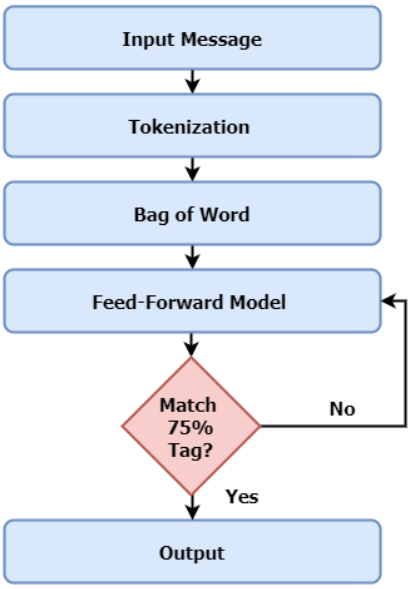


Figure 8:Flowchart of chatbot

Database connection

The data repository for the Railway automation web application is a MySQL database. As a result, the Mysql Database contains all of the application's data. Figure 9 shows how to create a database on the PhpMyAdmin server.

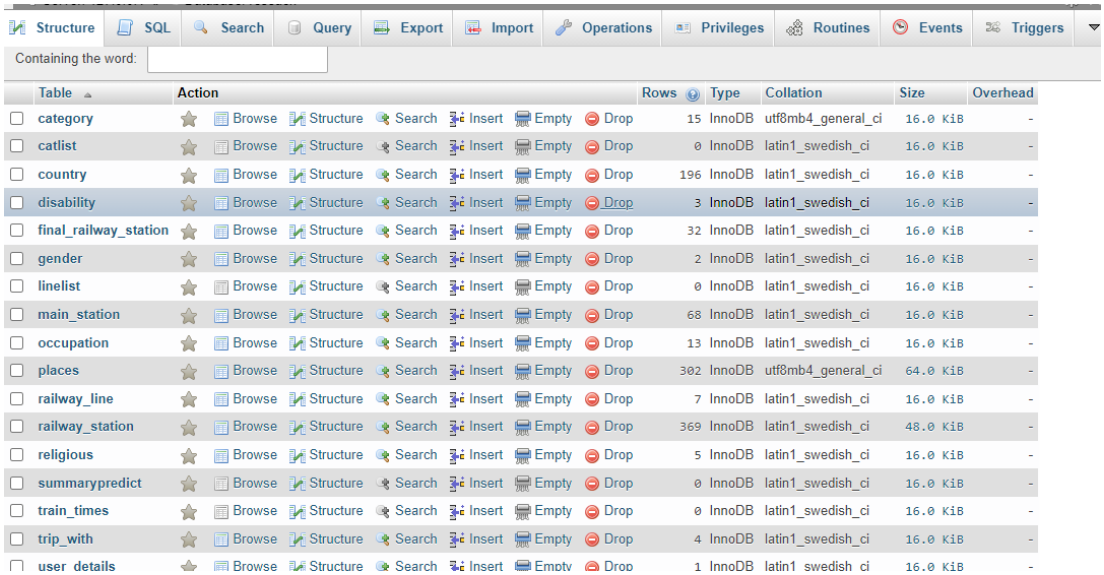


Figure 9:PhpMyAdmin database

API generation

For routing, APIs with a RESTful design are utilized. The fully functioning APIs handle Create, Read, Update, and Delete (CRUD) activities in particular. Figure 10 depicts one of the GET procedures on chatbot as an example. On the backend, APIs are also used to manage server connections.

```
@app.route('/<string:name>/')
def hello(name):
    device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

    with open('intents.json', 'r') as json_data:
        intents = json.load(json_data)

    FILE = "data.pth"
    data = torch.load(FILE)

    input_size = data["input_size"]
    hidden_size = data["hidden_size"]
    output_size = data["output_size"]
    all_words = data['all_words']
    tags = data['tags']
    model_state = data["model_state"]

    model = NeuralNet(input_size, hidden_size, output_size).to(device)
    model.load_state_dict(model_state)
    model.eval()
```

Figure 10: GET procedures on chatbot

Functionality in interfaces

The appropriate interface for the chatbot component in the user-friendly enhanced machine learning-based railway management system is shown in Figure 11 below.

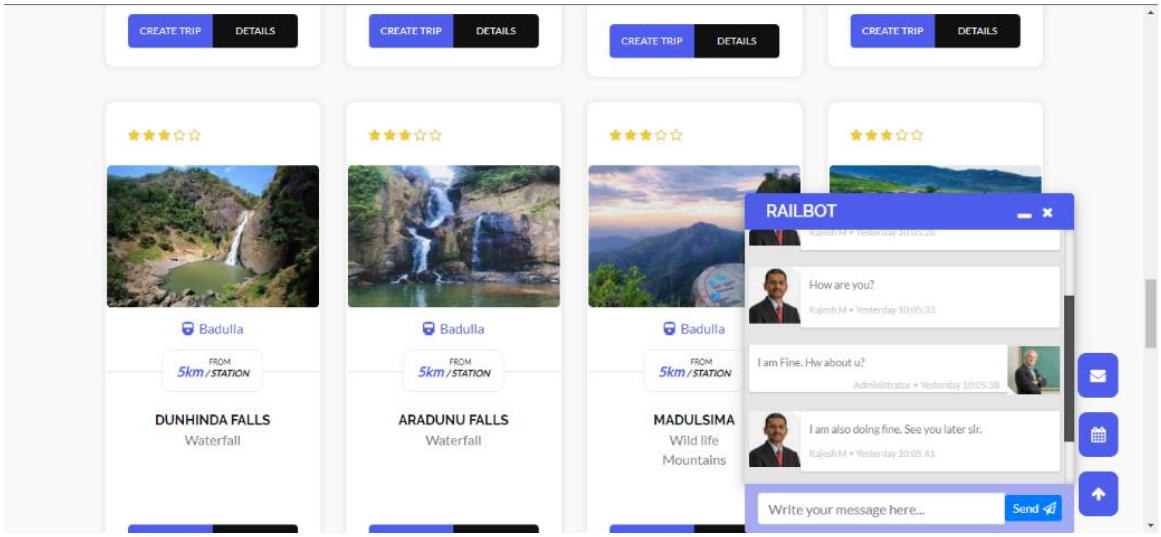


Figure 11:interface for the chatbot component

In addition, the Appendix section contains more comprehensive code for the whole Chatbot component. It provides the features that help with the prediction process and component development.

3. RESULT AND DISCUSSION

3.1. Results

The results and observations of the chatbot component in the user-friendly improved machine learning-based railway management system are detailed in this part. The findings were produced using the methodology's prediction procedure. The chatbot's output determined the accuracy and efficacy of the prediction procedure. The outcomes, on the other hand, are mostly determined by the dataset's aim and the DL methods used.

The user asks the chatbot a query. The inputs are then transmitted to the backend via the PHP API (figure 12). Then, using the flask API, capture the user input. The Feed-Forward neural network predicts the outcome of user inputs. The user's input is used to transform the input words into a bag of words.

```php
<?php
$msg= $_GET['msg'];
$str = str_replace(' ', '%', $msg);
$requestingURL='http://127.0.0.1:5000/'.$str;

        $curl = curl_init();

        curl_setopt_array($curl, array(
        CURLOPT_URL => $requestingURL,
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_ENCODING => '',
        CURLOPT_MAXREDIRS => 10,
        CURLOPT_TIMEOUT => 0,
        CURLOPT_FOLLOWLOCATION => true,
        CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
        CURLOPT_CUSTOMREQUEST => 'GET',
        CURLOPT_HTTPHEADER => array(
            'Content-Type: application/json'
        ),
        ));




        $response = curl_exec($curl);
        print_r($response);

?>
```

Figure 12:Php API

The pre-processed inputs are then fed into a trained DL model, which looks for similarities between the inputs and the sample questions (Figure 13). The model delivered the sample answer as a response if the user input matched the sample question by more than 75%.

```
{
  "tag": "anuradhapura-places",
  "patterns": ["what are the places i can visit in anuradhapura?", "what are the places I can see in Anuradhap
  "responses": ["You can see Ruwanweliseya, Thuparamaya, Isurumuniya and many other historical places in Anura
},

 {
  "tag": "create-sigiriya",
  "patterns": ["who made sigiriya?", "who created sigiriya?", "who built sigiriya?","who built lion rock?"],
  "responses": ["Sigiriya was built by King Kashyapa"]
},
```

Figure 13: Sample questions and answers

Sri Lanka has a diverse range of tourist attractions. Initially, just visiting locations in the Anuradhapura District was considered. We want to visit sites all throughout Sri Lanka in the future as part of our work.

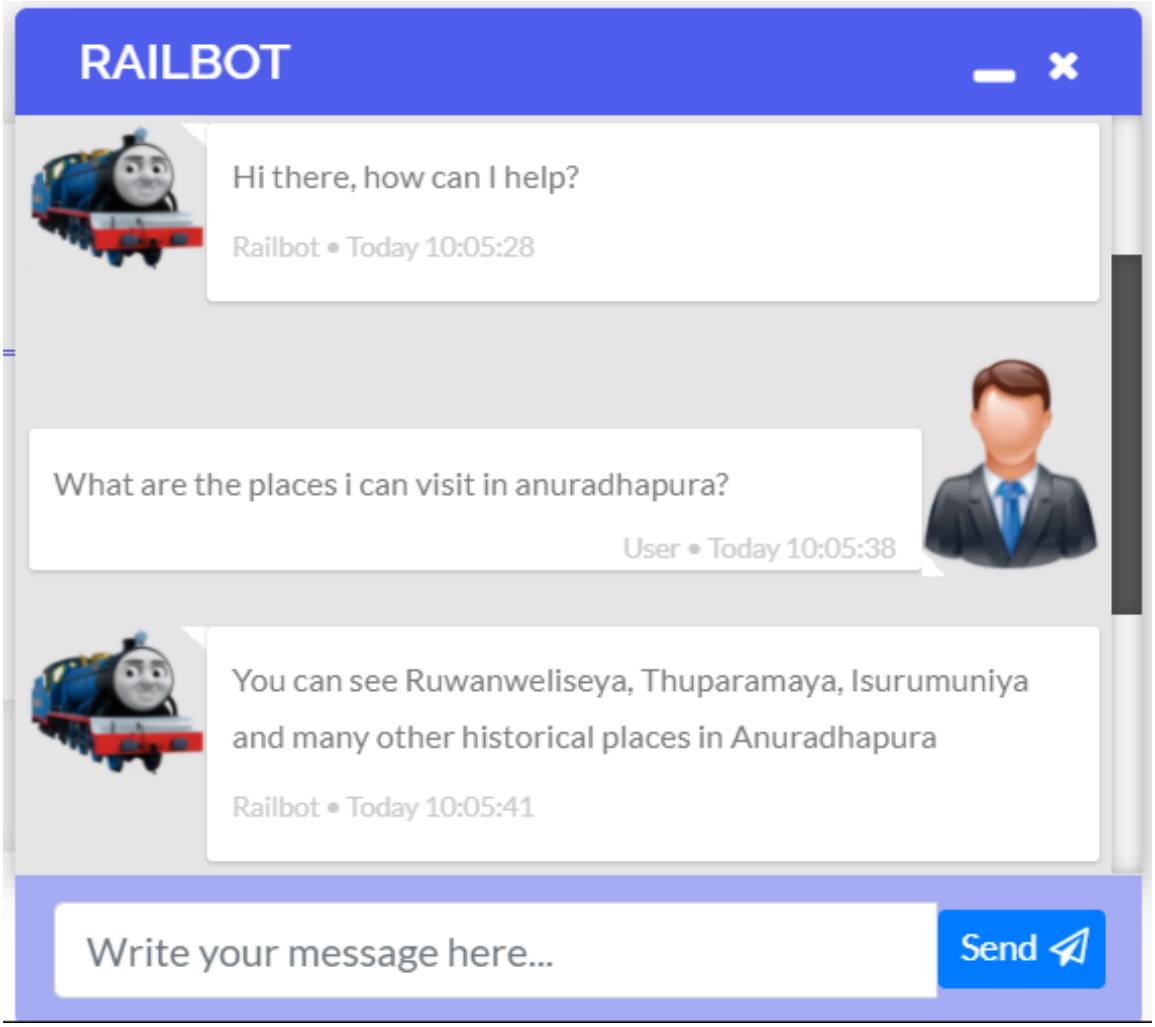


Figure 14:sample conversation

3.2. Research Findings

The user-friendly enhanced machine learning-based railway management system provides an automated platform with the primary goal of developing Sri Lanka's railway service and tourism business. As a result, the system offers the following four features: trip planning with train information, train seat prediction, location prediction, and intelligent chatbot. These are associated with both the railway and tourist industries. The notion is novel since these four functions have never been supplied

from a single system previously, as explained in this report's literature study. In general, researchers only investigate one function at a time.

As a result, the chatbot component also has a depth of weighted prediction. This component also includes station data, in addition to location information (Figure 15).

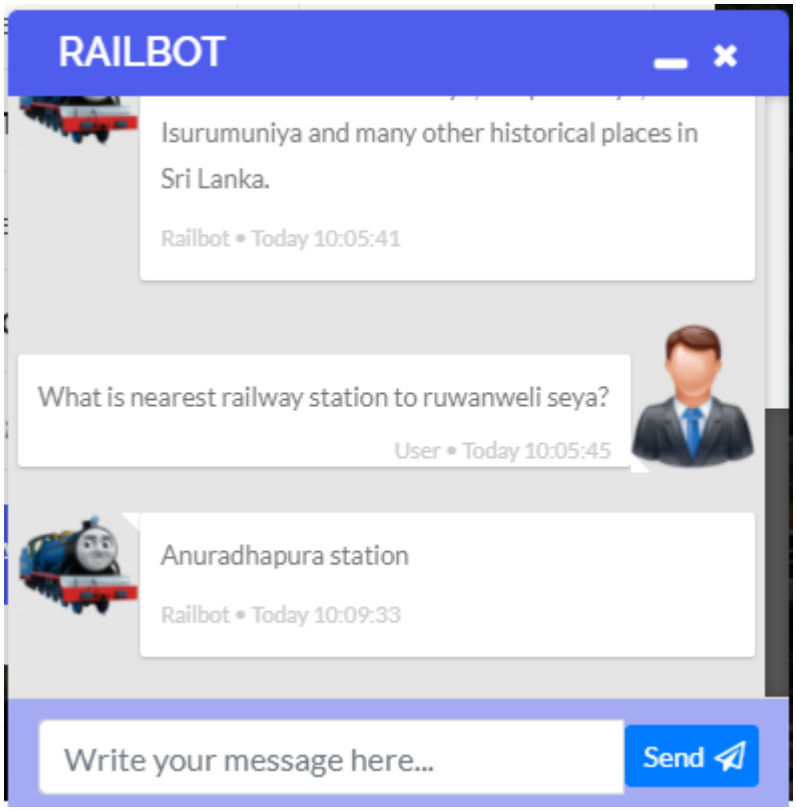


Figure 15:Station details in the chatbot

Furthermore, it is discovered that model training several times is necessary to improve the model's performance. In order to create a model with high performance and dependability, it is also necessary to use a genuine dataset for predictions. Furthermore, when it comes to improving a model's accuracy, it's important to employ input features wisely. As a result, the most relevant feature selection technique for the prediction should be picked.

The feature importance and study on relevant work techniques were used to choose the best features set in this chatbot component, which included beautiful sites in Sri Lanka and railway information. In addition, the Bag Of Word method was utilized to choose the most appropriate intent from the recognized inputs. Data preparation, on the other hand, is critical for obtaining the most accurate prediction results since it cleans and normalizes the dataset. These data aid in the creation of an accurate and efficient prediction.

3.3. Discussion

The user-friendly enhanced machine learning-based railway management system developed as part of the research project is an automated platform that has been deployed to help the railway and tourist industries grow. As a result, this platform has four major features: trip planning, intelligent chatbot, train seat prediction, and location prediction. Furthermore, the user-friendly improved machine learning-based railway management system is a web-based program that includes tourist and railway information.

As a result, the chatbot component has two major sides, which can also be referred to as subcomponents, such as providing location and providing station information. The results are also evaluated and supplied to the end user using the Railway automation web application. The primary functions that this component supports are also given below.

- brows to the user-friendly enhanced machine learning-based railway management system via https://railwaytoure.tk/
- find railbot on the web application
- type the question related to locations or railway station
- chatbot will provide the answer

Because the railway and tourist industries are both locally and globally affected by the present pandemic crisis and have a significant economic and social impact, the Department of Railways and the Sri Lanka Tourism Development Authority have a key responsibility to mitigate the repercussions. Many solutions have been discovered

in order to obtain information on tourist destinations and stations, as stated in the literature. However, they are all in different nations. However, the majority of similar efforts have focused on a single kind, such as railway or region information. In order to achieve reliable findings, some statistical analysis is also used. As a result, the chatbot component was created to address the identified need and to provide a solution to consumers' most pressing needs. In addition, the findings advocate decreasing the waste of a country's capital through promoting public transportation and saving human time.

This component was also built using cutting-edge technologies like Python and PHP, as well as tools like PyCharm and VS Code. Many publications, personal blogs, and technical and academic knowledge were heavily involved in the selection process. However, there were several difficulties encountered during the creation of the chatbot component. Finding accurate information on the locations was a big problem. After extensive study, accurate genuine information was discovered in a variety of sources. In addition, the Sri Lanka Tourism Development Authority has provided a great deal of assistance in locating the material. As a result, they have a thorough real dataset with a broad understanding of location information. Furthermore, choosing the best algorithm for a chatbot was a difficult task. As a result, numerous related works and publications were consulted in order to pick some acceptable algorithms, with the most appropriate algorithm being chosen mostly after comparing their accuracies. The other major issue was to choose appropriate input characteristics, which was achieved using a variety of approaches including a bag of words, tokenization, and referencing to related publications.

Nonetheless, a fully functional chatbot component was successfully created and deployed alongside the user-friendly improved machine learning-based railway administration system. Initially, the application evaluates the Anuradhapura district's sites. In the future, location information for the entire island will be available, allowing for a variety of premium services. The user-friendly improved machine learning-based railway management system, on the other hand, may persuade both railway and tourism growth.

4. CONCLUSIONS

The Railway Automated Platform was created with the goal of improving the railway and tourist industries. A web-based application is included in this platform. As a result, this paper has been written on the chatbot component, which is the second component of the user-friendly improved machine learning-based railway management system.

Tourism has long been the country's third-largest source of foreign currency. Due to the aftermath of the April 2019 Easter Terror Attacks, Sri Lanka had over 1.9 million visitor visits in 2019, a 21% decrease from the previous year. In 2019, the industry is expected to generate about $3.5 billion. In 2020, about 570,000 tourists are expected[22]. The tourist sector appears to be in a major downturn. As a result, strengthening the tourist business is a significant priority for the Sri Lankan tourism administration.

According to the literature review, many automated solutions have been implemented in the tourist and railway industries, but they are all foreign applications, which are insufficient to ensure the efficacy of the Sri Lanka tourism business. The user-friendly enhanced machine learning-based railway management system chatbot component was created to solve that gap.

However, this component contains two types of questions.

- Location base questions
- Station base questions

However, this component contains two types of questions. chatbot has been implemented using the Feedforward neural network algorithm and classified the questions as into the intent.

The user-friendly enhanced machine learning-based railway management system interface displays the results of the responses. There are also user-friendly features that allow users to get the info they need from the app.

The user-friendly enhanced machine learning-based railway management system may be provided as a series of packages with premium features in terms of

commercialization. Overall, the user-friendly enhanced machine learning-based railway management system chatbot component ensures that the needed service for travellers is given more effectively and efficiently.

The chatbot will be developed in other languages in the future, including Sinhala and Tamil. It will be extremely beneficial to Sri Lankan consumers. Because Sinhala is Sri Lanka's primary language and Tamil is the country's secondary language. In the beginning, this system was also built as a web application. However, we plan to deploy the system with Android and iOS smartphones in the future. It will become more user-friendly as a result. Mobile applications are easier to use than online applications since everyone has a mobile device.

5. REFERENCE

[1] V. M. Ramachandiran, P. N. K. Babu, and R. Manikandan, "Prediction of road accidents severity using various algorithms," *International Journal of Pure and Applied Mathematics*, vol. 119, pp. 16663-16669, 01/01 2018.

[2] K. R. Sumana and D. Phaneendra, *Smart Automated Modelling using Eclat Algorithm for Traffic Accident Prediction*. 2019.

[3] M. R. Afsar, M. R. Haque, M. Dooley, and X. Shen, "RailBot: A Novel Assistive Device for Stair Climbing," Journal of Medical Devices, vol. 15, no. 1, 2021.

[4] “чатбот для пошуку та моніторингу залізничних квитків,” RailwayBot. [Online]. Available: https://railwaybot.com/en/. [Accessed: 26-Feb-2021].

[5] Railwaybuddy.site. 2021. *Railway Buddy Chatbot*. [online] Available at: <https://www.railwaybuddy.site/> [Accessed 11 October 2021].

[6] Flow.ai, "Create a location-aware chatbot and find nearest store locations around the user: Flow.ai Chatbot Knowledge Base and Blog," AI chatbot software. [Online]. Available: https://flow.ai/blog/create-a-location-aware-chatbot-and-find-nearest-store-location-around-user. [Accessed: 26-Feb-2021].

[7] Ieeexplore.ieee.org. 2021. *Interactive Transport Enquiry with AI Chatbot*. [online] Available at: <https://ieeexplore.ieee.org/document/9120905> [Accessed 11 October 2021].

[8] Ieeexplore.ieee.org. 2021. Intelligent travel chatbot for predictive recommendation in echo platform. [online] Available at: <https://ieeexplore.ieee.org/document/8301732> [Accessed 11 October 2021].

[9] Ieeexplore.ieee.org. 2021. *Chatbot - Attendance and Location Guidance system (ALGs)*. [online] Available at: <https://ieeexplore.ieee.org/document/9451793> [Accessed 11 October 2021].

[10] Ieeexplore.ieee.org. 2021. Chatbot - Attendance and Location Guidance system (ALGs). [online] Available at: <https://ieeexplore.ieee.org/document/9451793> [Accessed 11 October 2021].

[11] JetBrains. (1-April-2020). PyCharm. Available: https://www.jetbrains.com/pycharm/

[12](1- June -2021). Visual Studio Code. Available: https://code.visualstudio.com/docs/editor/whyvscode

[13] (1-June-2021). What is Python? Executive Summary. Available: https://www.python.org/doc/essays/blurb/

[14] (23- June -2021). Flask (web framework). Available: https://en.wikipedia.org/wiki/Flask_(web_framework)

[15] (25- June -2021). *Anaconda Navigator*. Available: https://docs.anaconda.com/anaconda/navigator/#:~:text=Anaconda%20Navigator%20is%20a%20desktop,without%20using%20command%2Dline%20commands.&text=To%20get%20Navigator%2C%20get%20the%20Navigator%20Cheat%20Sheet%20and%20install%20Anaconda.

[16] (2021, 12-July-2021). *Commercialization*. Available: https://www.investopedia.com/terms/c/commercialization.asp#:~:text=Commercialization%20is%20the%20process%20of,the%20new%20product%20or%20service.

[17] (23-July-2021). *Why is Testing Necessary?* Available: https://www.toolsqa.com/software-testing/istqb/why-is-testing-57 necessary/#:~:text=The%20testing%20is%20important%20since,and%20high%2Dperformance%20software%20operation.

[18] Tokenisation, H., 2021. What is Tokenisation | Methods to Perform Tokenisation. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2019/07/how-get-started-nlp-6-unique-ways-perform-tokenization/> [Accessed 24 July 2021].

[19] 2021. stemming.. [online] Available at: <https://searchenterpriseai.techtarget.com/definition/stemming.> [Accessed 24 July 2021].

[20] GreatLearning Blog: Free Resources what Matters to shape your Career!. 2021. An Introduction to Bag of Words in NLP using Python | What is BoW?. [online] Available at: <https://www.mygreatlearning.com/blog/bag-of-words/> [Accessed 24 July 2021].

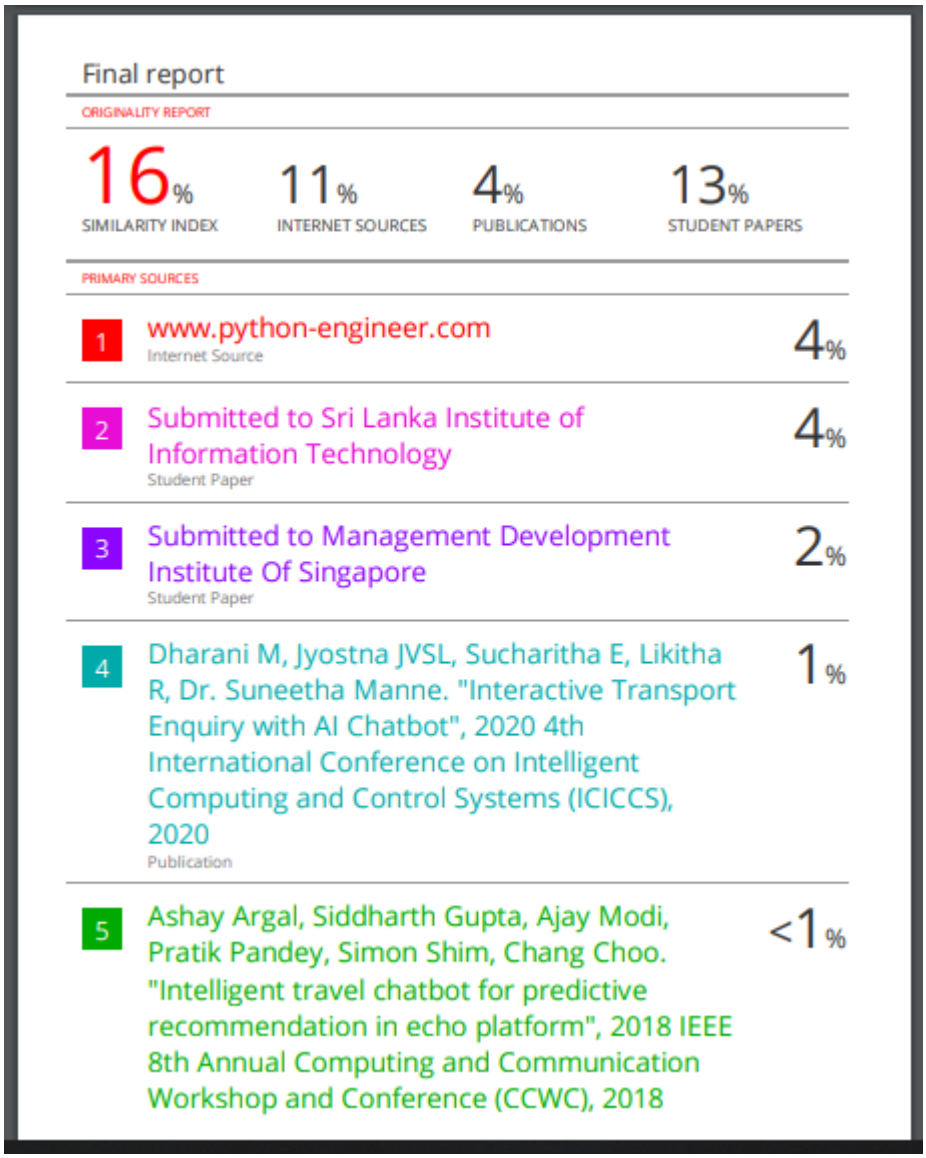
[21] Medium. 2021. Deep Learning: Feedforward Neural Network. [online] Available at: <https://towardsdatascience.com/deep-learning-feedforward-neural-network-26a6705dbdc7> [Accessed 24 July 2021].

[22] Ieeexplore.ieee.org. 2021. Chatbot - Attendance and Location Guidance system (ALGs). [online] Available at: <https://ieeexplore.ieee.org/document/9451793> [Accessed 11 October 2021].

[23] Python-engineer.com. 2021. Chat Bot With PyTorch - NLP And Deep Learning | Python Engineer. [online] Available at: <https://www.python-engineer.com/posts/chatbot-pytorch/> [Accessed 12 October 2021].

6. APPENDIX

Summary of plagiarism report



Final report

ORIGINALITY REPORT

16%	11%	4%	13%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.python-engineer.com Internet Source	4%
2	Submitted to Sri Lanka Institute of Information Technology Student Paper	4%
3	Submitted to Management Development Institute Of Singapore Student Paper	2%
4	Dharani M, Jyostna JVSL, Sucharitha E, Likitha R, Dr. Suneetha Manne. "Interactive Transport Enquiry with AI Chatbot", 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 2020 Publication	1%
5	Ashay Argal, Siddharth Gupta, Ajay Modi, Pratik Pandey, Simon Shim, Chang Choo. "Intelligent travel chatbot for predictive recommendation in echo platform", 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), 2018	<1%

Source code of the chatbot

```
from flask import Flask
app = Flask(__name__)
import random
import json

import torch

from model import NeuralNet
from nltk_utils import bag_of_words, tokenize

@app.route('/<string:name>/')
def hello(name):
    device = torch.device('cuda' if torch.cuda.is_available() else
'cpu')

    with open('intents.json', 'r') as json_data:
        intents = json.load(json_data)

    FILE = "data.pth"
    data = torch.load(FILE)

    input_size = data["input_size"]
    hidden_size = data["hidden_size"]
    output_size = data["output_size"]
    all_words = data['all_words']
    tags = data['tags']
    model_state = data["model_state"]

    model = NeuralNet(input_size, hidden_size, output_size).to(device)
    model.load_state_dict(model_state)
    model.eval()

    bot_name = "TravellChatty"

    while True:
    # sentence = "do you use credit cards?"
        sentence =name
        if sentence == "quit":
            break

        sentence = tokenize(sentence)
        X = bag_of_words(sentence, all_words)
        X = X.reshape(1, X.shape[0])
        X = torch.from_numpy(X).to(device)
```

```
        output = model(X)
        _, predicted = torch.max(output, dim=1)

        tag = tags[predicted.item()]

        probs = torch.softmax(output, dim=1)
        prob = probs[0][predicted.item()]
        if prob.item() > 0.75:
            for intent in intents['intents']:
                if tag == intent["tag"]:
                    return (f" {random.choice(intent['responses'])}")

                    #print(f"{bot_name}:
{random.choice(intent['responses'])}")
        else:
            return (f" I do not understand...")
            #print(f"{bot_name}: I do not understand...")

app.run()

import torch
import torch.nn as nn

class NeuralNet(nn.Module):
    def __init__(self, input_size, hidden_size, num_classes):
        super(NeuralNet, self).__init__()
        self.l1 = nn.Linear(input_size, hidden_size)
        self.l2 = nn.Linear(hidden_size, hidden_size)
        self.l3 = nn.Linear(hidden_size, num_classes)
        self.relu = nn.ReLU()
    def forward(self, x):
        out = self.l1(x)
        out = self.relu(out)
        out = self.l2(out)
        out = self.relu(out)
        out = self.l3(out)
        # no activation and no softmax at the end
        return out

import numpy as np
import nltk
nltk.download('punkt')
```

```
from nltk.stem.porter import PorterStemmer
stemmer = PorterStemmer()

def tokenize(sentence):
    """
    split sentence into array of words/tokens
    a token can be a word or punctuation character, or number
    """
    return nltk.word_tokenize(sentence)


def stem(word):
    """
    stemming = find the root form of the word
    examples:
    words = ["organize", "organizes", "organizing"]
    words = [stem(w) for w in words]
    -> ["organ", "organ", "organ"]
    """
    return stemmer.stem(word.lower())


def bag_of_words(tokenized_sentence, words):
    """
    return bag of words array:
    1 for each known word that exists in the sentence, 0 otherwise
    example:
    sentence = ["hello", "how", "are", "you"]
    words = ["hi", "hello", "I", "you", "bye", "thank", "cool"]
    bog   = [  0 ,    1 ,    0 ,   1 ,    0 ,    0 ,      0]
    """
    # stem each word
    sentence_words = [stem(word) for word in tokenized_sentence]
    # initialize bag with 0 for each word
    bag = np.zeros(len(words), dtype=np.float32)
    for idx, w in enumerate(words):
        if w in sentence_words:
            bag[idx] = 1

    return bag

import json
import numpy as np

import torch
import torch.nn as nn
```

```
from torch.utils.data import Dataset, DataLoader
from nltk_utils import bag_of_words, tokenize, stem
from model import NeuralNet

with open('intents.json', 'r') as f:
    intents = json.load(f)

all_words = []
tags = []
xy = []

for intent in intents['intents']:
    tag = intent['tag']
    # add to tag list
    tags.append(tag)
    for pattern in intent['patterns']:
        # tokenize each word in the sentence
        w = tokenize(pattern)
        # add to our words list
        all_words.extend(w)
        # add to xy pair
        xy.append((w, tag))

ignore_words = ['?', '.', '!']
all_words = [stem(w) for w in all_words if w not in ignore_words]
all_words = sorted(set(all_words))
tags = sorted(set(tags))
print(tags)

X_train = []
y_train = []

for (pattern_sentence, tag) in xy:
    # X: bag of words for each pattern_sentence
    bag = bag_of_words(pattern_sentence, all_words)
    X_train.append(bag)
    # y: PyTorch CrossEntropyLoss needs only class labels, not one-hot
    label = tags.index(tag)
    y_train.append(label)

X_train = np.array(X_train)
y_train = np.array(y_train)

class ChatDataset(Dataset):

    def __init__(self):
```

```
        self.n_samples = len(X_train)
        self.x_data = X_train
        self.y_data = y_train

    # support indexing such that dataset[i] can be used to get i-th
sample
    def __getitem__(self, index):
        return self.x_data[index], self.y_data[index]

    # we can call len(dataset) to return the size
    def __len__(self):
        return self.n_samples

num_epochs = 1000
learning_rate = 0.001
batch_size = 8
input_size = len(X_train[0])
hidden_size = 8
output_size = len(tags)
print(input_size, output_size)

dataset = ChatDataset()
train_loader = DataLoader(dataset=dataset,
                          batch_size=batch_size,
                          shuffle=True,
                          num_workers=0)

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model = NeuralNet(input_size, hidden_size, output_size).to(device)

criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)

for epoch in range(num_epochs):
    for (words, labels) in train_loader:
        words = words.to(device)
        labels = labels.to(dtype=torch.long).to(device)

        # Forward pass
        outputs = model(words)
        # if y would be one-hot, we must apply
        # labels = torch.max(labels, 1)[1]
        loss = criterion(outputs, labels)

        # Backward and optimize
        optimizer.zero_grad()
```

```
        loss.backward()
        optimizer.step()
    if (epoch+1) % 100 == 0:
        print (f'Epoch [{epoch+1}/{num_epochs}], Loss:
{loss.item():.4f}')


print(f'final loss: {loss.item():.4f}')

data = {
"model_state": model.state_dict(),
"input_size": input_size,
"hidden_size": hidden_size,
"output_size": output_size,
"all_words": all_words,
"tags": tags
}

FILE = "data.pth"
torch.save(data, FILE)

print(f'training complete. file saved to {FILE}')
```