# AYUR MIND

The proposed solution is a conversational AI chatbot that provides personalized advice and solutions for Ayurvedic treatments and medical herbs based on specific symptoms. It utilizes a knowledge base for treatment information and incorporates image processing for herbal plant identification. A geometry library maps locations of identified herbs and connects patients with Ayurvedic doctors. A social network allows for health-related discussions, with content contributing to the knowledge base. Auto-machine learning ensures up-to-date data and the involvement of Ayurvedic experts during development. Data collection includes publicly available social network communities and information on herbs and Ayurvedic doctors.

**www.ayurminds.com**

# INTRODUCTION

The Future of Ayurveda: Harnessing the Power of Artificial Intelligence for Personalized Treatment and Diagnosis

Ayurveda recognizes that each person is unique and therefore tailors treatments to each individual's specific needs. Ayurveda emphasizes the importance of self-care and self-awareness to maintain good health. Ayurveda recognizes the connection between the environment and health, and emphasizes the importance of living in harmony with nature.

# MEET OUR TEAM


**MADUWANTHA K.A.I**

IT20069186

**SENARATHNE S.M.A.D**

IT20089436

**DE SILVA A.S.**

IT20166038

**JAYASINGHE J.A.S.C**

IT20216078

# OVERALL
# PROJECT DESCRIPTION

- Modern lifestyle leads to poor health and unhappiness.
- Ayurveda offers alternative solutions for non-communicable diseases.
- Cost of western medicine can be prohibitive, and not all diseases are curable.
- Availability of herbal plants and medicines is limited.
- Finding qualified Ayurvedic doctors can be difficult.
- Identifying necessary herbs and treatments is challenging for people.

# RESEARCH OBJECTIVES

- AI CHATBOT PROVIDING AYURVEDIC RECOMMENDATIONS.
- BUILDING MORE EMPHATICALLY CONNECTION BETWEEN THE PATIENT AND AYURVEDIC DOCTOR
- DEVELOP A SOCIAL NETWORK TO EXCHANGE HEALTH-RELATED INFORMATION.
- IDENTIFYING & MAPPING HERBAL PLANTS FOR AYURVEDIC TREATMENTS.

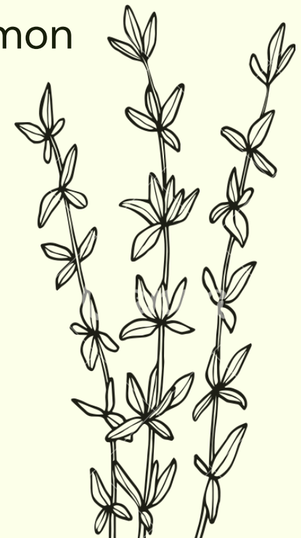

# IT20069186

# MADUWANTHA K.A.I

Software Engineering

# INTRODUCTION
# BACKGROUND

- Identify the need for accessible and personalized healthcare solutions for individuals seeking natural remedies and Ayurvedic treatments.
- growing interest in Ayurvedic practices and the demand for accessible platforms to receive Ayurvedic recommendations.
- what is an AI chatbot
- A healthy lifestyle approach using Ayurveda which recommend basic treatments for conditions and will suggest appropriate Ayurvedic treatments and medical herbs based on known symptoms and common health problems.
- A conversational AI chatbot will be developed as a platform for users to receive solutions through text.



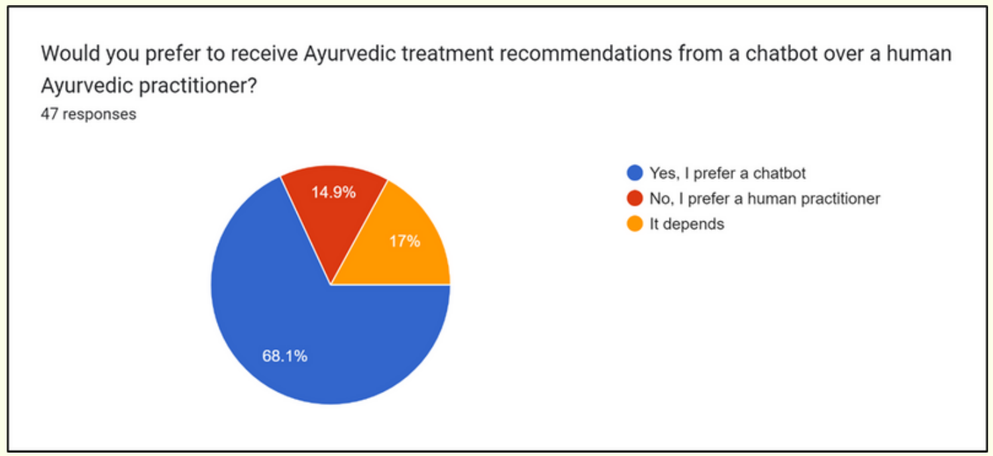
# INTRODUCTION

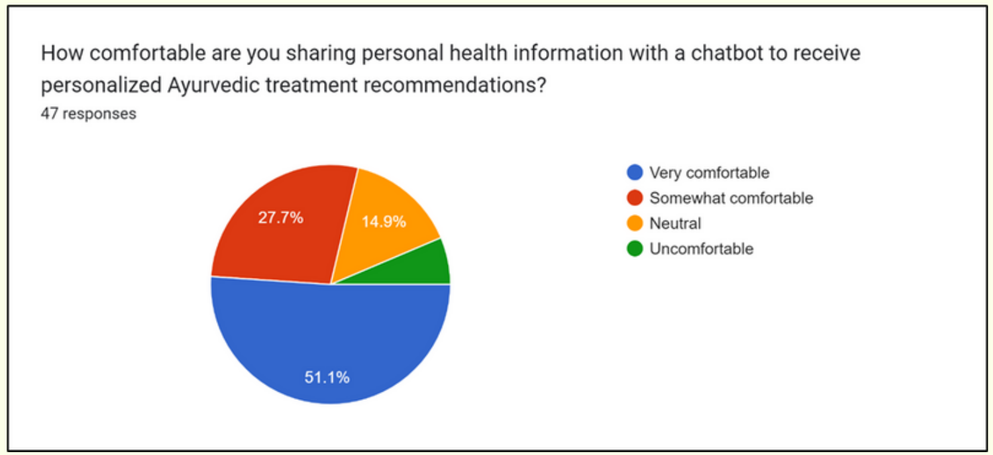
# RESEARCH QUESTION

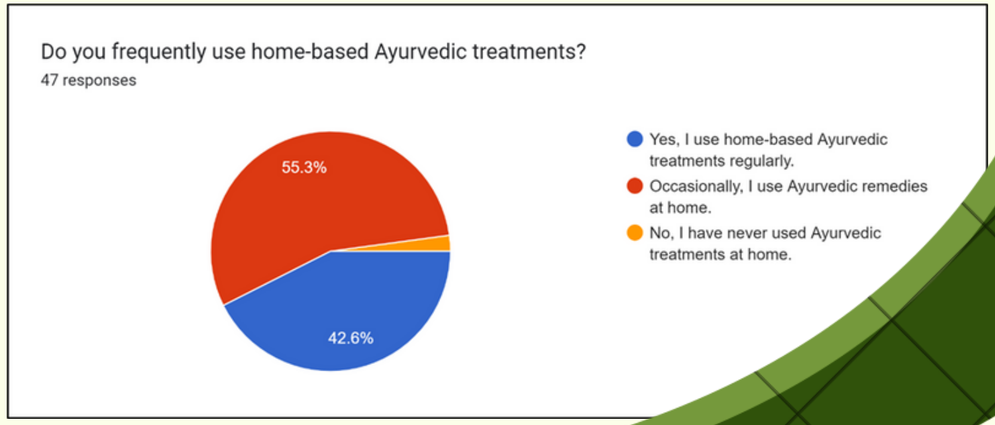
- DO YOU FREQUENTLY USE HOME BASED AYURVEDIC TREATMENTS?
- WOULD YOU PREFER TO RECIEVE AYURVEDIC TREATMENTS RECOMENDATIONS FROM A CHATBOT OVER A HUMAN AYURVEDIC PRACTIONER?

- Based on the survey results, it can be concluded that more than half of the respondents are frequently using ayurvedic treatments.

- The results suggest that there are no clear consequences on whether respondents would prefer to receive Ayurvedic treatment recommendations from a chatbot or a human practitioner. It highlights the importance of considering the preferences and concerns of individuals when implementing AI technology in healthcare.



# RESEARCH GAP

| Features | Research A | Research B | Research C | Proposed System |
|---|---|---|---|---|
| Ayurvedic Chatbot | X | X | X | ✓ |
| Availability of personalized solutions | ✓ | ✓ | X | ✓ |
| User Friendly Platform | X | X | ✓ | ✓ |
| Availability of a knowledge base | ✓ | ✓ | ✓ | ✓ |
| Chatbot framework | X | | | ✓ |
| Ability to ask to follow back questions | ✓ | X | ✓ | ✓ |
| Using RASA | X | X | X | ✓ |
| Updating Knowledgebase Easily | X | X | X | ✓ |
| Technology Used | N-GRAM | - | NLP/RNN | NLP/RNN |

# SPECIFIC & SUB OBJECTIVES

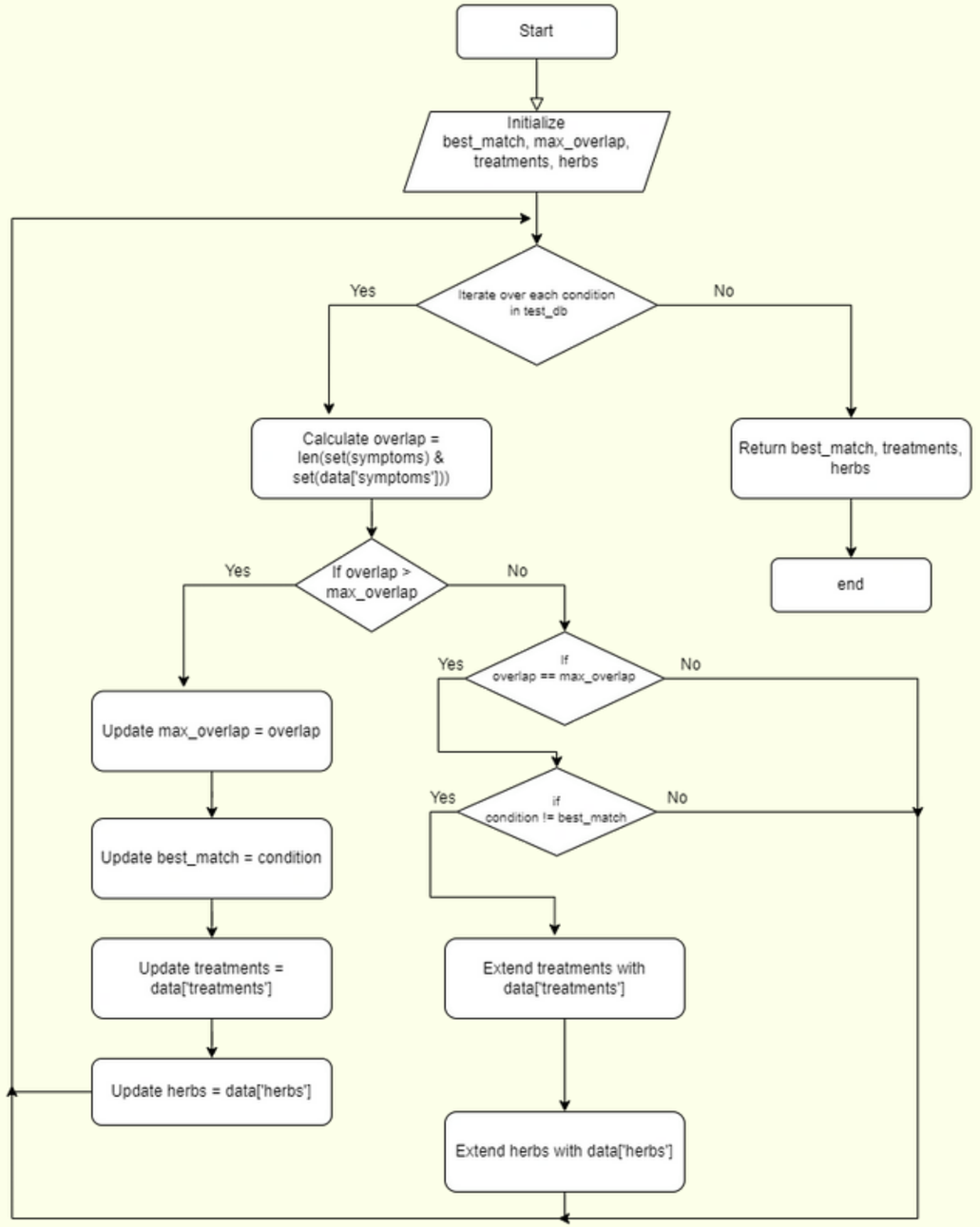
A CONVERSATIONAL AI CHATBOT THAT ALLOWS USERS TO TEXT AND RECEIVE SOLUTIONS FOR NONCOMMUNICABLE DISEASES THROUGH AYURVEDA. THE KNOWLEDGE BASE WILL BE UPDATED AND IMPROVED.

- RESEARCH AND GATHERING INFORMATION ON THE SELECTED NON-COMMUNICABLE DISEASES
- CREATING A KNOWLEDGE BASE OF SYMPTOMS AND COMMON HEALTH PROBLEMS RELATED TO EACH DISEASE.
- PRE-PROCESSING AND CLEANING THE DATA TO PREPARE IT FOR CHAT BOT
- TRAINING AN AI MODEL ON THE KNOWLADGEBASE TO IDENTIFY PATTERNS AND RELATIONSHIPS BETWEEN SYMPTOMS, COMMON HEALTH PROBLEMS, AND AYURVEDIC TREATMENTS AND MEDICAL HERBS.
- EVALUATING THE PERFORMANCE OF THE AI MODEL ON A VALIDATION SET AND FINE-TUNING THE MODEL TO IMPROVE ITS ACCURACY
- VERIFYING THE ACCURACY AND VALIDITY OF THE SUGGESTIONS MADE BY THE SOFTWARE APPLICATION THROUGH CONSULTATIONS WITH AYURVEDIC EXPERTS.

# METHODOLOGY

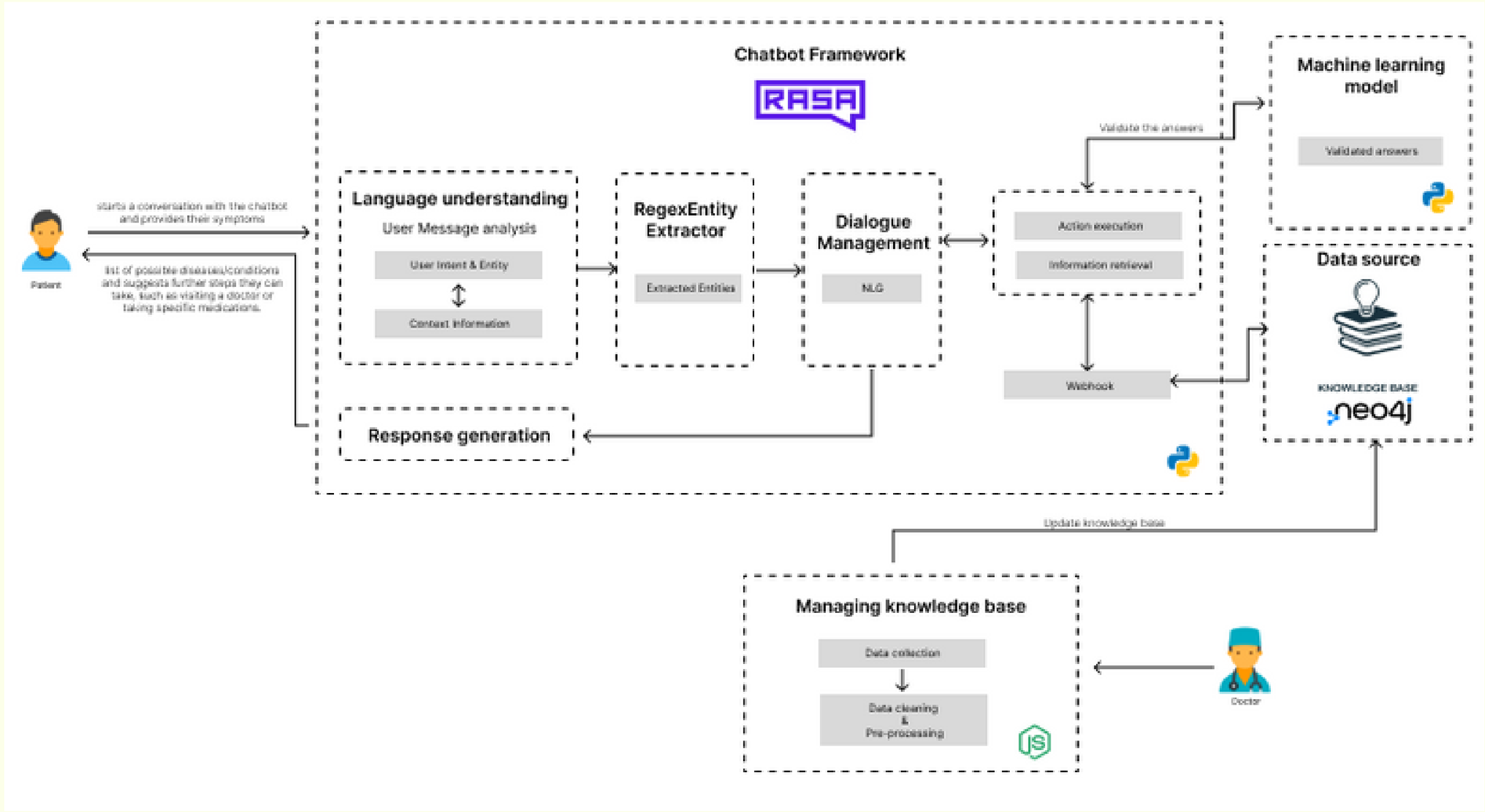
## FLOW CHART OF USED CUSTOM ALGORITHM

THIS FUNCTION EFFICIENTLY IDENTIFIES THE CONDITION THAT BEST MATCHES THE GIVEN SYMPTOMS BY CALCULATING THE OVERLAP BETWEEN THE SYMPTOMS AND THE SYMPTOM SET ASSOCIATED WITH EACH CONDITION IN THE DATABASE. IT TAKES INTO ACCOUNT THE MAXIMUM OVERLAP AND HANDLES CASES WHERE MULTIPLE CONDITIONS HAVE THE SAME OVERLAP, ENSURING THAT THE TREATMENTS AND HERBS LISTS ARE APPROPRIATELY UPDATED.



# METHODOLOGY

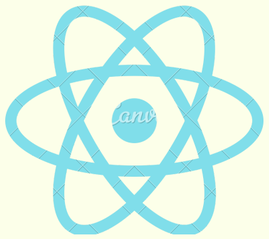
# SYSTEM DIAGRAM



# TECHNIQUES & TECHNOLOGIES

| | |
|---|---|
| Technologies | • React Native<br>• Python<br>• Expo<br>• Node Server |
| Techniques | • Intent Recognition, Entity Extraction , Dialogue Management, Machine Learning Model, Knowledge Base, Sentiment Analysis, Personalization |
| Algorithms | • Recurrent Neural Networks (RNN) |
| Architectures | • RASA |






# METHODOLOGY

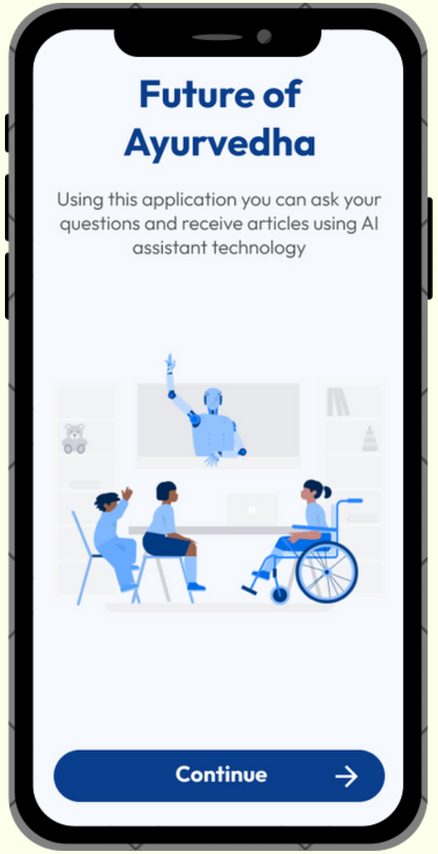
- DATA COLLECTION
- DATA PRE-PROCESSING
- CREATING THE ACTIONS BASED ON THE SYMPTOMS
- VALIDATE THE GENARATE ANSWERS
- CREATE SYMPTOM-BASED KNOWLEDGE BASE
- DATA VISUALISATION
- KNOWLEDGE BASE WILL BE UPDATED AND IMPROVED
- VERIFY SUGGESTIONS WITH AYURVEDIC EXPERTS

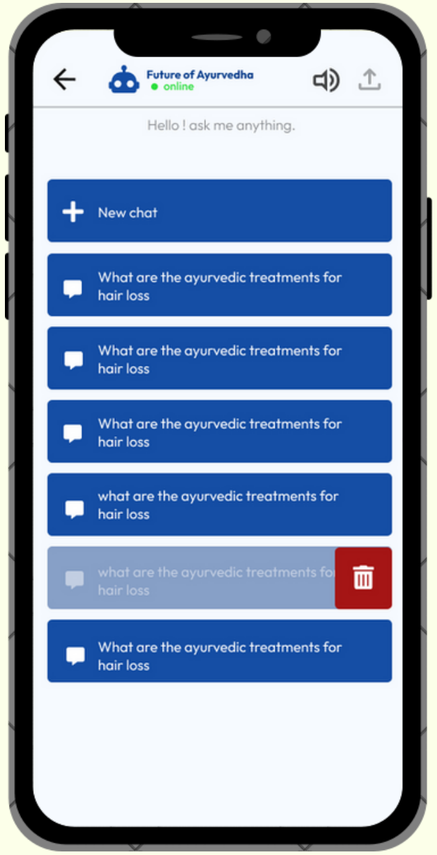
# COMPARISONS OF FRAMEWORK

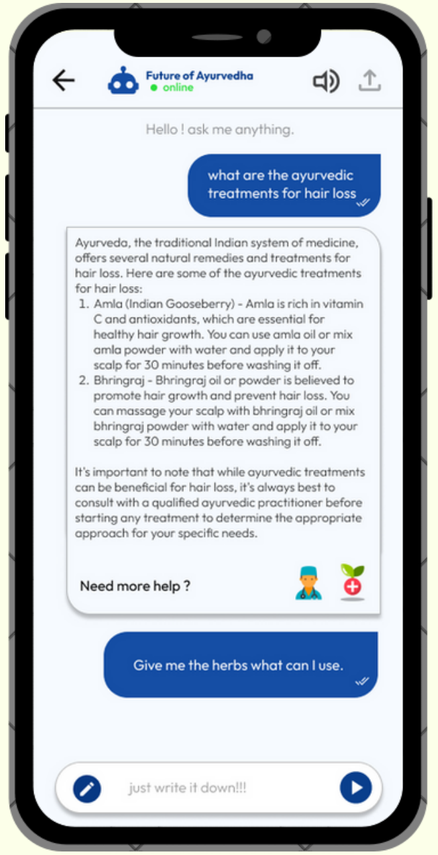
| Framework | Rasa | Microsoft Bot Framework | Dialogflow (formerly API.ai) | IBM Watson Assistant |
|---|---|---|---|---|
| Open-Source | Yes | No | No | No |
| NLU | Highly customizable NLU component with support for complex intents and entities | Limited NLU capabilities with predefined intents and entities | Offers NLU capabilities with predefined intents and entities | Provides NLU capabilities with predefined intents and entities |
| Dialogue Management | Rule-based and machine learning-based dialogue management options | Rule-based dialogue management | Rule-based dialogue management | Rule-based dialogue management |
| Customizability | Highly customizable with open-source codebase | Limited customizability due to closed-source nature | Limited customization options | Limited customization options |
| Machine Learning Capabilities | Supports integration of machine learning models | Limited machine learning capabilities | Limited machine learning capabilities | Provides machine learning capabilities |
| Community and Support | Active and growing community with extensive documentation and resources | Microsoft-backed with good community support | Google-backed with a supportive community | IBM-backed with community support |

# DISPLAY RESULTS IN MOBILE APPLICATIONS

## MOBILE UI PROTOTYPES



# REQUIREMENTS

## FUNCTIONAL REQUIREMENTS

- Patient can ask questions and provide feedback based on their symptoms.
- Able to receive and understand questions related to symptoms from patients.
- Able to identify the disease or condition based on the symptoms provided by the patient and provide suggestions and treatments for the identified disease or condition.
- Provide information about medications, their dosage, and any possible side effects.
- Able to maintain a record of patient symptoms and provide follow-up advice as necessary.

## NON-FUNCTIONAL REQUIREMNTS

- Performance
- Security
- Reliability
- Scalability
- Usability
- Accessibility
- Compatibility
- Maintainability
- Compliance
- Performance metrics

## SYSTEM REQUIREMNTS

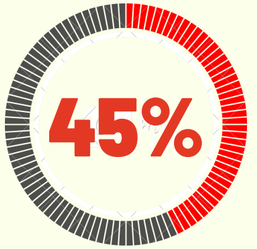
- Server infrastructure
- Software development platform
- Natural language processing (NLP) tools
- Machine learning models
- Database management system
- Integration with third-party systems
- Communication channels
- Security measures
- Monitoring and analytics tools

# COMPLETION AND FUTURE WORKS

## COMPLETION OF COMPONENTS


- DATA COLLECTION PRE-PROCESSING
- CREATE SYMPTOM-BASED KNOWLEDGE BASE
- CREATING THE ACTIONS BASED ON THE SYMPTOMS
- PROVIDING THE TREATMENTS BASED ON THE SYMPTOMS GIVEN BY THE PATIENT

## FUTURE IMPLEMENTATIONS




- VERIFY THE PROVIDED ANSWERS USING ML MODLE
- VERIFY SUGGESTIONS WITH AYURVEDIC EXPERTS
- IMPLEMENTATION OF USER INTERFACES
- HOSTING

# IT20166038

# DE SILVA A.S

Software Engineering

# INTRODUCTION

# BACKGROUND

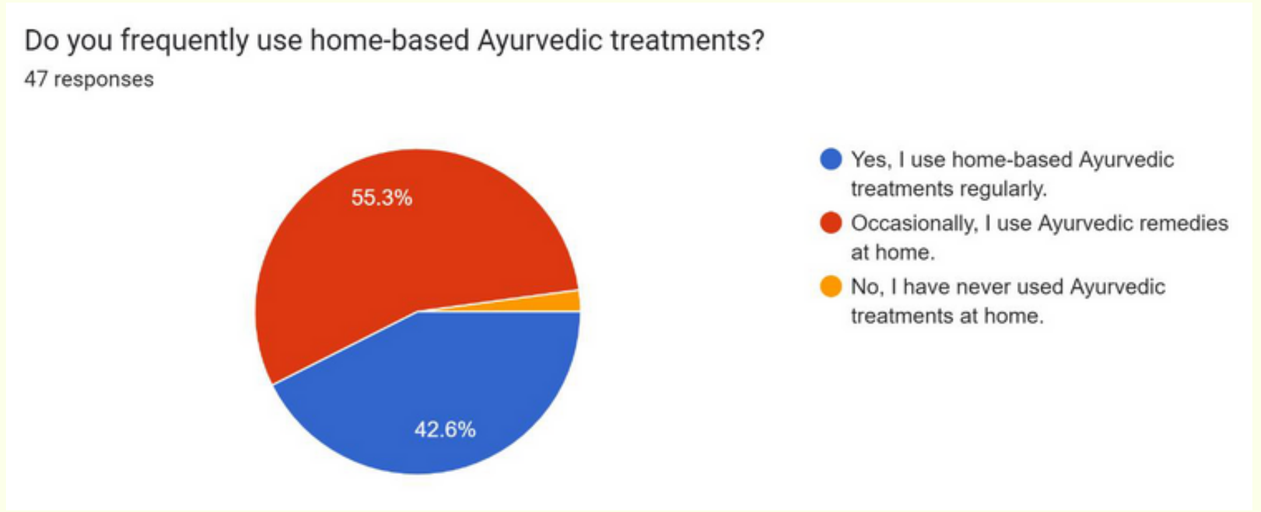
- Ayurveda has become increasingly well-known on a global scale.
- In order to provide individualized and efficient therapy, this has led researchers and practitioners to investigate how Ayurveda principles might be combined with cutting-edge technology like Artificial Intelligence
- Our research indicates that Sri Lankans have a long-standing cultural and traditional connection to Ayurveda.
- We suggest a way for specific problems including arthritis, blood sugar, hair loss, infertility, obesity, paranasal sinusitis, wounds, scrapes, and swellings, our solution attempts to address Ayurvedic therapies as well as general healthy principles.
- Our suggested approach will include an image processing element that can recognize the herbal plants required for treating various conditions.
- The locations of these plants will be mapped out by a geometry library, which will also link patients to Ayurvedic practitioners in a particular geographic region.
- Users will get the ability to add new plants to the model without a high technical knowledge.

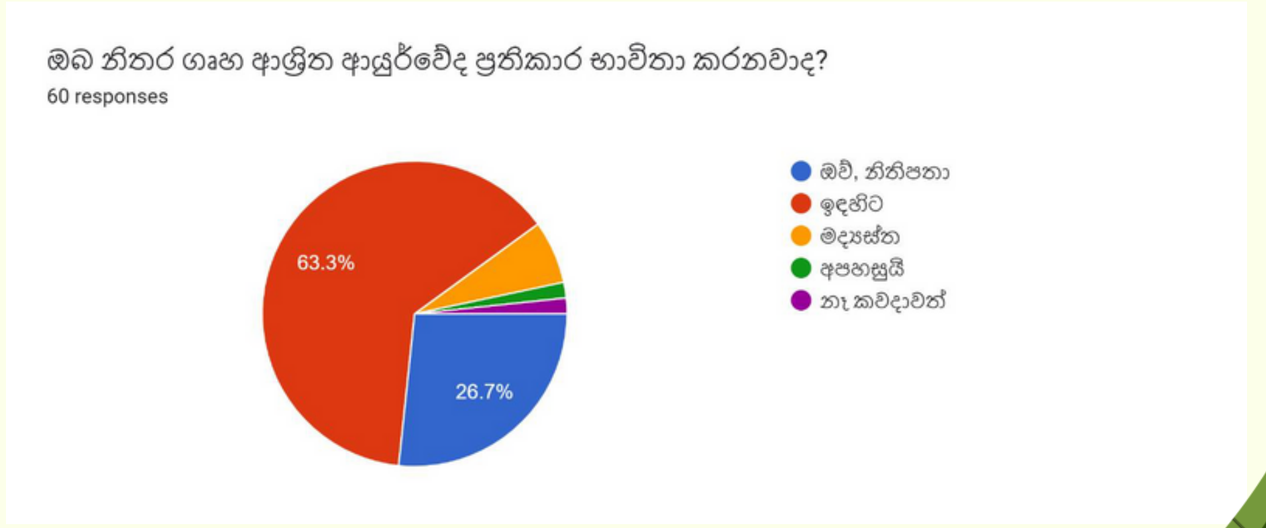
# INTRODUCTION

# RESEARCH QUESTION

## DO YOU FREQUENTLY USE HOME BASED AYURVEDIC TREATMENTS?

- Based on the survey results, it can be concluded that more than half of the respondents are frequently using ayurvedic treatments.
- Due to the presence of Ayurvedic hospitals and clinics all over Sri Lanka, and Ayurveda has a significant influence on the country's healthcare system.



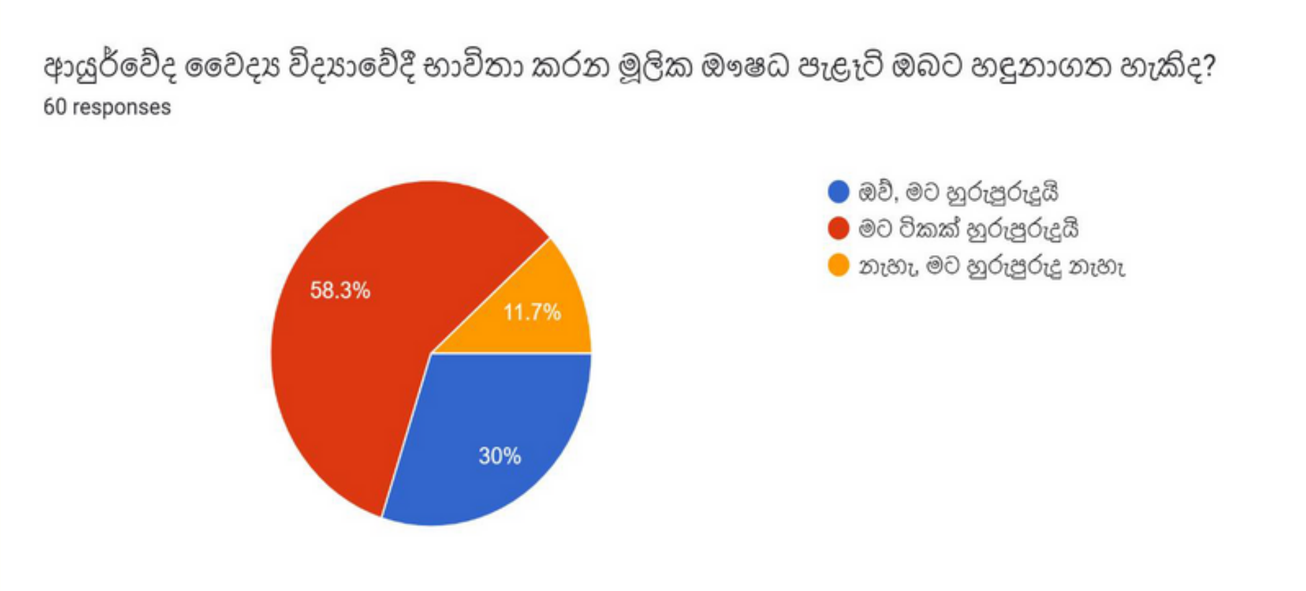



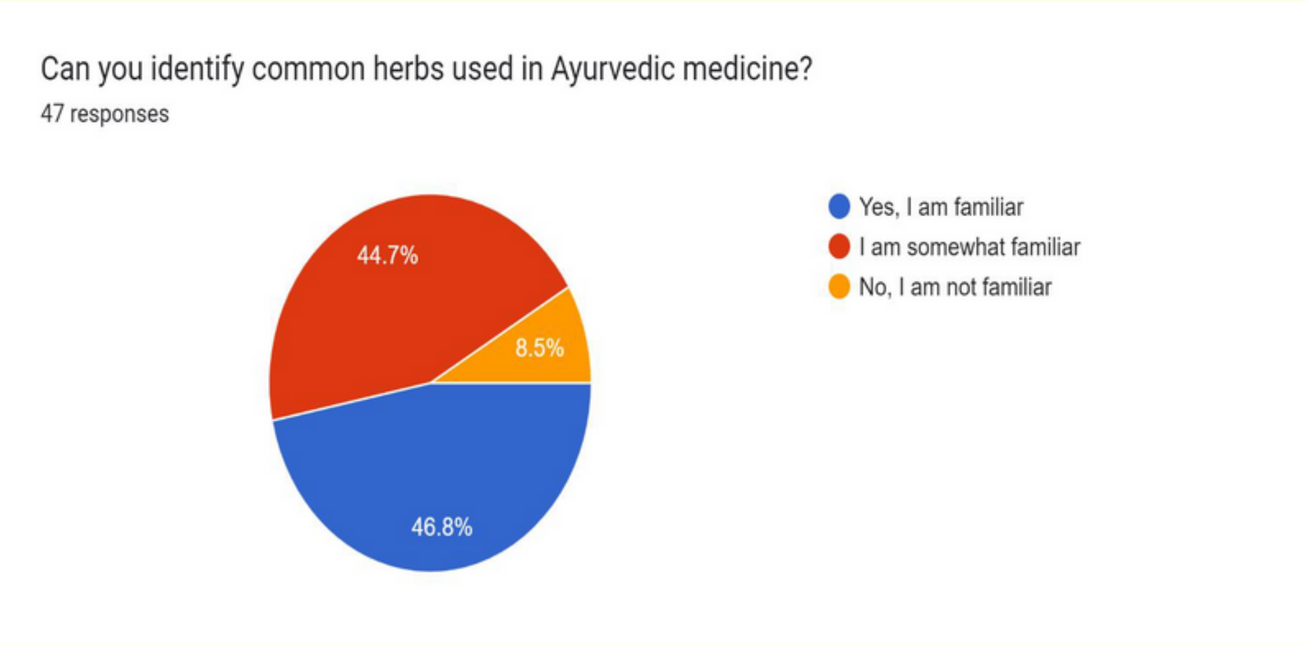

# INTRODUCTION

# RESEARCH QUESTION

## CAN YOU IDENTIFY COMMON HERBS USED IN AYURVEDIC MEDICINE?

- Based on the survey results we can conclude that half of the people who participated to the survey doesn't have the ability to find herbs.
- Also with the limited availability of herbal plants and medicines, combined with the lack of knowledge and resources for identifying and locating these plants, there is a need for an AI-based solution that can accurately identify Ayurvedic medical herbs used for the treatment of non-communicable diseases through image processing, while also mapping their locations.

# SPECIFIC & SUB OBJECTIVE

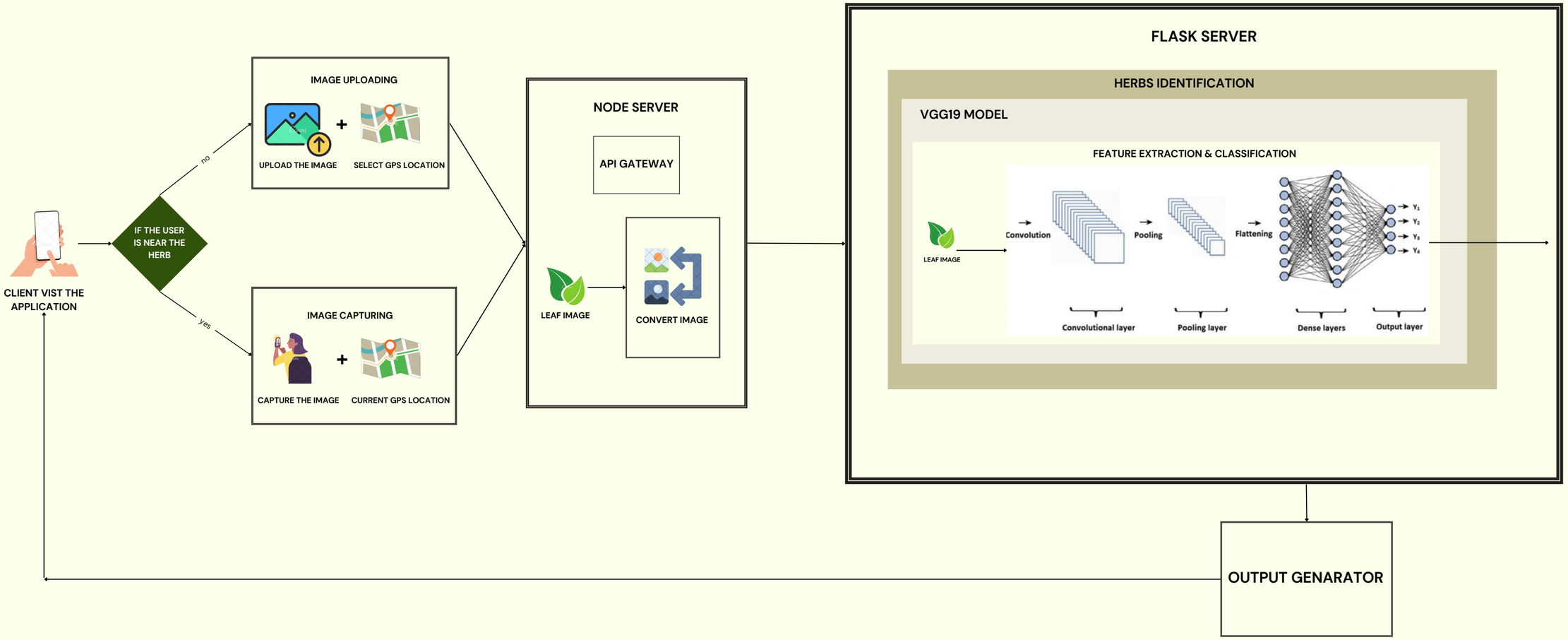
IDENTIFICATION OF AYURVEDIC MEDICAL HERBS WHICH ARE NEEDED FOR TREATMENTS FOR SOME DISEASES USING IMAGE PROCESSING THROUGH TRAINING MACHINE LEARNING MODELS AND MAPPING OF HERBAL PLANTS WITH LOCATIONS.

USING CONTINUAL LEARNING/TRANSFER LEARNING TO IMPROVE ACCURACY. ALSO, CAN INCORPORATE AUTO MACHINE LEARNING TO MAKE IT EASIER TO ADD NEW PLANTS.

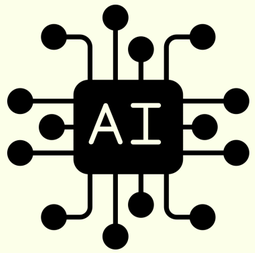
- PRE-PROCESSING OF THE COLLECTED IMAGES, SUCH AS RESIZING AND NORMALIZATION
- SPLITTING THE DATA INTO TRAINING, VALIDATION, AND TESTING SETS.
- TRAINING A MACHINE LEARNING MODEL USING PRE-PROCESSED IMAGE DATA.
- EVALUATING THE MODEL ON THE VALIDATION SET TO IDENTIFY AREAS OF IMPROVEMENT.
- COLLECTING AND PROCESSING LARGE AMOUNTS OF IMAGE DATA FROM VARIOUS GEOGRAPHICAL AREAS TO MAP THE DISTRIBUTION OF HERBAL PLANTS.
- FINE-TUNING THE MODEL BASED ON THE EVALUATION RESULTS.
- TESTING THE FINAL MODEL ON THE TESTING MODEL AND IMPLEMENTING IT IN A SOFTWARE APPLICATION FOR PRACTICAL USE.

# METHODOLOGY

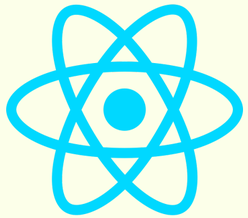
# HIGH LEVEL SYSTEM ARCHITECTURE DIAGRAM

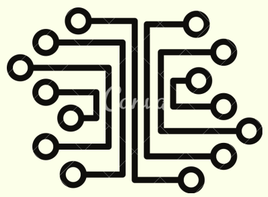


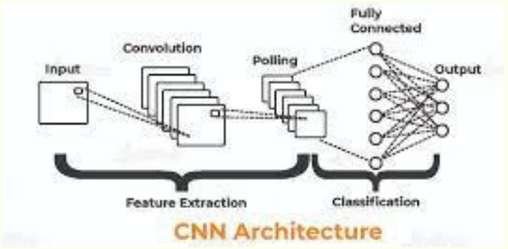
# TECHNIQUES & TECHNOLOGIES




| | |
|---|---|
| Technologies | • React Native<br>• Python<br>• Expo<br>• Node Server<br>• Google Map<br>• TensorFlow<br>• Flask |
| Techniques | • Image processing<br>• Auto Machine Learning (AML)<br>• CNN |
| Model | • VGG19 |





# METHODOLOGY

- DATA COLLECTION
- DATA PRE-PROCESSING
- USE TRANSFER LEARNING BASED CNN ARCHITECTURES FOR TRAINING THE MODELS
- DATA AUGMENTATION AND SELECTED THE BEST TECHNIQUE TO RETRAIN MODELS TO ACHIEVE GOOD FITTING
- TUNING HYPERPARAMETERS AND RETRAINED MODELS TO SELECT THE BEST ARCHITECTURE
- DATA VISUALISATION
- HOST THE FINAL MODEL
- DISPLAYED THE RESULTS IN MOBILE APPLICATION

# DATA COLLECTION AND PRE-PROCESSING

```
Found 1335 images belonging to 29 classes.
Found 457 images belonging to 29 classes.
```

```
Alpinia Galanga (Rasna)
Amaranthus Viridis (Arive-Dantu)
Artocarpus Heterophyllus (Jackfruit)
Azadirachta Indica (Neem)
Basella Alba (Basale)
Brassica Juncea (Indian Mustard)
Carissa Carandas (Karanda)
Citrus Limon (Lemon)
Ficus Auriculata (Roxburgh fig)
Ficus Religiosa (Peepal Tree)
Hibiscus Rosa-sinensis
Jasminum (Jasmine)
Mangifera Indica (Mango)
Mentha (Mint)
Moringa Oleifera (Drumstick)
Muntingia Calabura (Jamaica Cherry-Gasagase)
Murraya Koenigii (Curry)
Nerium Oleander (Oleander)
Nyctanthes Arbor-tristis (Parijata)
Ocimum Tenuiflorum (Tulsi)
Piper Betle (Betel)
Plectranthus Amboinicus (Mexican Mint)
Pongamia Pinnata (Indian Beech)
Psidium Guajava (Guava)
...
Syzygium Cumini (Jamun)
Syzygium Jambos (Rose Apple)
Tabernaemontana Divaricata (Crape Jasmine)
Trigonella Foenum-graecum (Fenugreek)
```

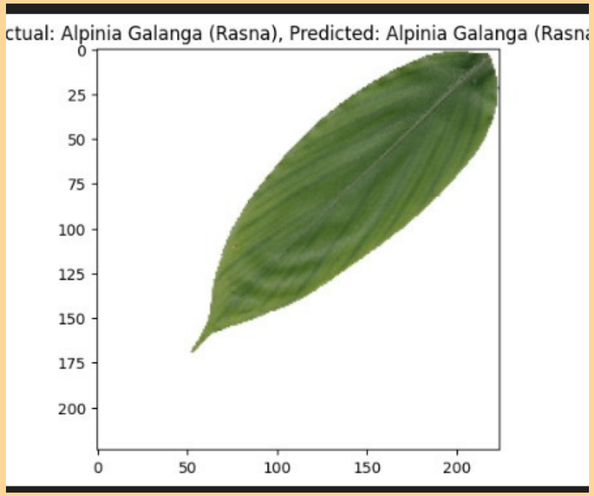
*Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...*

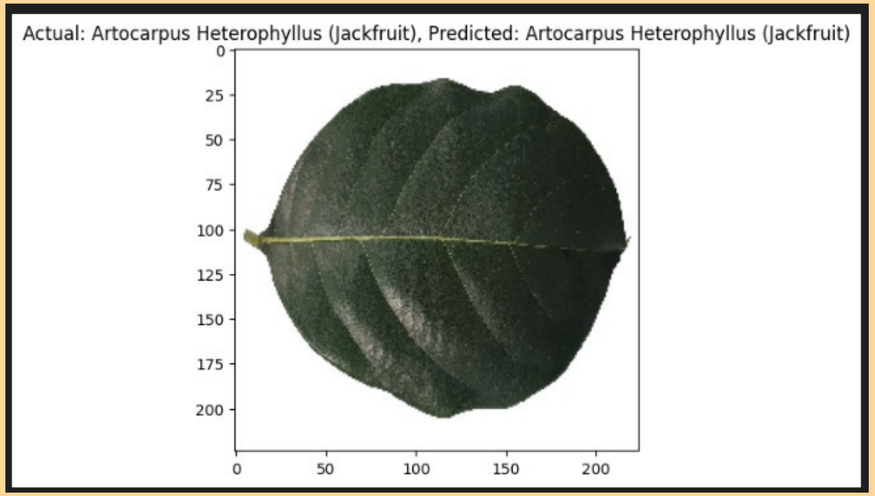
# ACCURACY AND LOSS

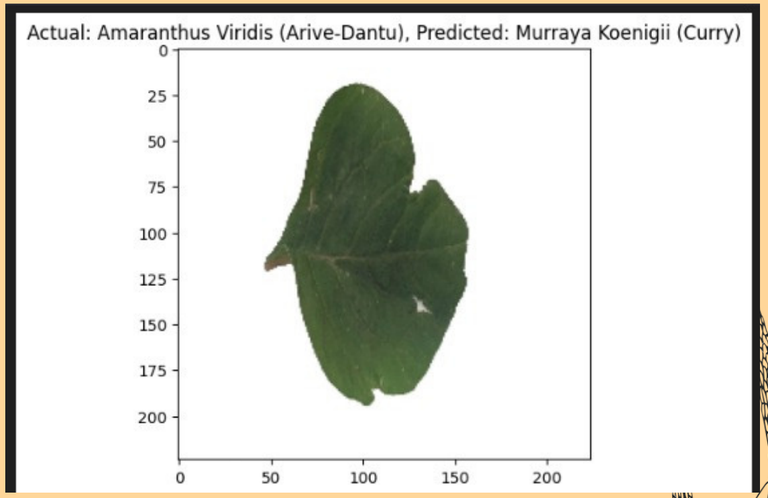
```
  r = model.fit_generator(
Epoch 1/30
27/27 [==============================] - 499s 18s/step - loss: 3.5412 - accuracy: 0.0404 - val_loss: 3.3463 - val_accuracy: 0.0678
Epoch 2/30
27/27 [==============================] - 494s 18s/step - loss: 3.3362 - accuracy: 0.0622 - val_loss: 3.2737 - val_accuracy: 0.0744
Epoch 3/30
27/27 [==============================] - 495s 18s/step - loss: 3.0716 - accuracy: 0.1416 - val_loss: 2.5131 - val_accuracy: 0.3085
Epoch 4/30
27/27 [==============================] - 494s 18s/step - loss: 2.4838 - accuracy: 0.2757 - val_loss: 1.8335 - val_accuracy: 0.4114
Epoch 5/30
27/27 [==============================] - 496s 18s/step - loss: 1.9318 - accuracy: 0.4180 - val_loss: 1.7479 - val_accuracy: 0.5208
Epoch 6/30
27/27 [==============================] - 496s 18s/step - loss: 1.7184 - accuracy: 0.4854 - val_loss: 1.5156 - val_accuracy: 0.5667
Epoch 7/30
27/27 [==============================] - 494s 18s/step - loss: 1.4743 - accuracy: 0.5408 - val_loss: 1.2362 - val_accuracy: 0.6521
Epoch 8/30
27/27 [==============================] - 492s 18s/step - loss: 1.2554 - accuracy: 0.6172 - val_loss: 1.0292 - val_accuracy: 0.6718
Epoch 9/30
27/27 [==============================] - 492s 18s/step - loss: 1.0823 - accuracy: 0.6442 - val_loss: 0.9450 - val_accuracy: 0.6937
Epoch 10/30
27/27 [==============================] - 492s 18s/step - loss: 0.9895 - accuracy: 0.6891 - val_loss: 0.9713 - val_accuracy: 0.6937
Epoch 11/30
27/27 [==============================] - 492s 18s/step - loss: 1.0996 - accuracy: 0.6412 - val_loss: 1.0136 - val_accuracy: 0.6915
Epoch 12/30
27/27 [==============================] - 492s 18s/step - loss: 0.9804 - accuracy: 0.6981 - val_loss: 0.9257 - val_accuracy: 0.7024
Epoch 13/30
...
Epoch 29/30
27/27 [==============================] - 493s 18s/step - loss: 0.3424 - accuracy: 0.8831 - val_loss: 0.6547 - val_accuracy: 0.8315
Epoch 30/30
27/27 [==============================] - 492s 18s/step - loss: 0.3387 - accuracy: 0.8899 - val_loss: 0.5312 - val_accuracy: 0.8512
```

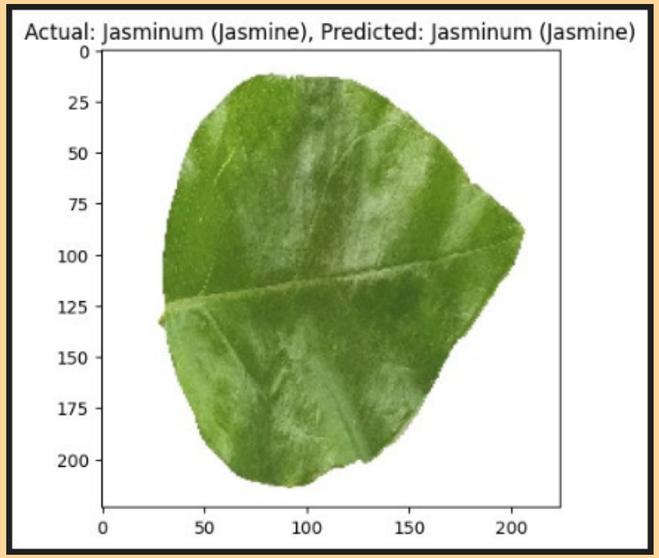
*Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...*

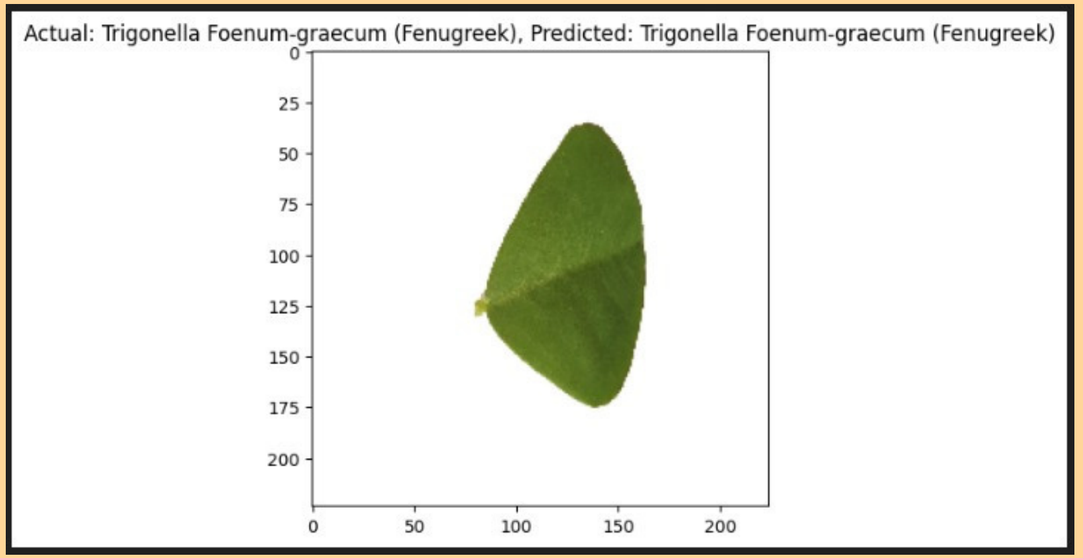
# LEAF IDENTIFICATION ACCURACIES



# EVIDENCE OF COMPLETION

## COMPARISONS OF ARCHITECTURES

### VGG16

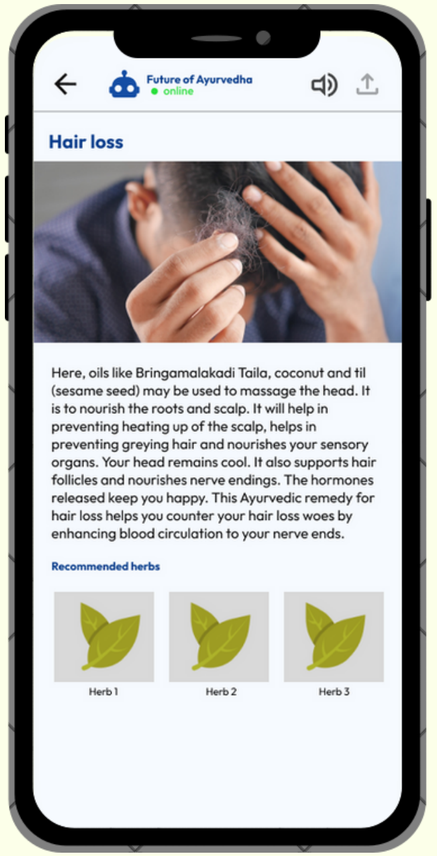
1. **Architecture: VGG16 consists of 16 layers, including 13 convolutional layers and 3 fully connected layers.**
2. **Filter Sizes: The convolutional layers use small 3x3 filters, which are the smallest possible size to capture both local and global features effectively.**
3. **Depth: VGG16 has a relatively shallower architecture compared to VGG19, which can make it easier to train and computationally less expensive.**
4. **Parameters: VGG16 has approximately 138 million trainable parameters, making it a relatively large model.**
5. **Performance: VGG16 performs well on various image classification tasks and has achieved competitive results in benchmark datasets like ImageNet.**
6. **Generalization: VGG16 tends to generalize well to new datasets due to its depth and the use of smaller filters**.

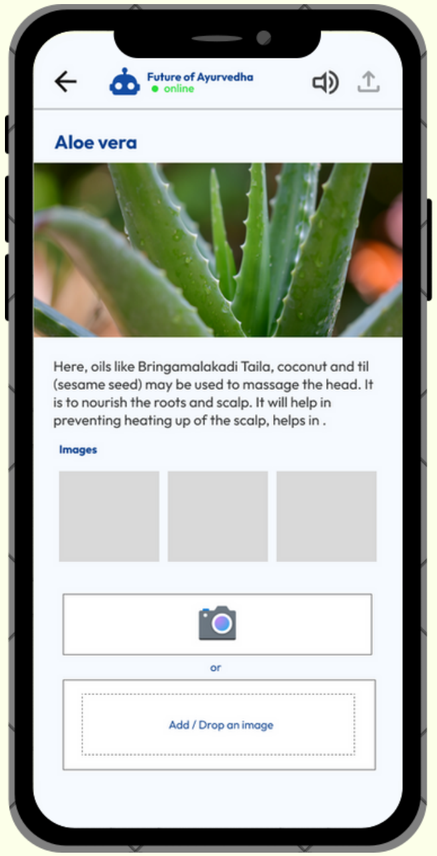
### VGG19

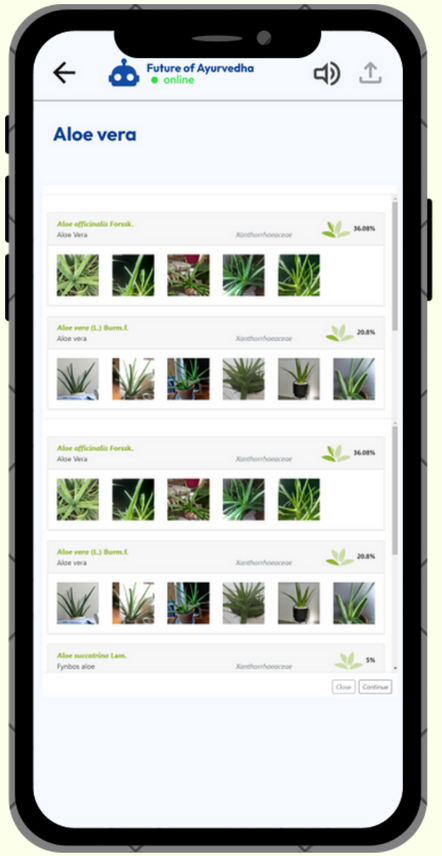
1. **Architecture: VGG19 has a deeper architecture compared to VGG16, with 19 layers, including 16 convolutional layers and 3 fully connected layers.**
2. **Filter Sizes: Similar to VGG16, VGG19 uses 3x3 filters in its convolutional layers.**
3. **Depth: VGG19's deeper architecture allows it to capture more complex and abstract features compared to VGG16.**
4. **Parameters: VGG19 has more parameters than VGG16, with approximately 143 million trainable parameters.**
5. **Performance: Due to its increased depth, VGG19 has the potential to capture more intricate details in images and may achieve slightly better performance than VGG16, although the difference may not always be significant.**
6. **Computational Cost: The increased depth of VGG19 makes it computationally more expensive to train and use compared to VGG16**

# DISPLAY RESULTS IN MOBILE APPLICATIONS

## MOBILE UI PROTOTYPES



# REQUIREMENTS

## FUNCTIONAL REQUIREMENTS

- Allow users to upload images of plants for identification.
- Allow users to capture images of plants for identification.
- Able to identify plants based on the uploaded images and provide information about the plant's name, family, genus, species, and other relevant details.
- The system should be able to identify plants based on the uploaded images and provide information about the plant's location.
- High accuracy rate in identifying plants.
- Should integrate with a database
- The system should allow us to make it easier to add new plants.

## NON-FUNCTIONAL REQUIREMNTS

- Performance
- Usability
- Reliability
- Security
- Compatibility
- Maintainability
- Scalability
- Accessibility

## SYSTEM REQUIREMNTS

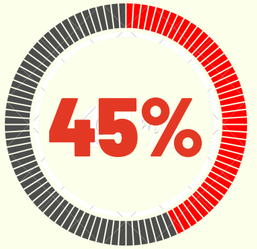
- Image processing capabilities
- Machine learning algorithms
- Database integration
- Mobile compatibility
- Scalability
- Performance
- Usability
- Security
- Reliability
- Integration with third-party APIs
- System backups and data recovery

# COMPLETION AND FUTURE WORKS

## COMPLETION OF COMPONENTS


- DATA GATHERING AND PREPROCESSING
- DATASET APPROVAL
- TRAINING THE MACHINE LEARNING MODEL
- INITIAL LEAF RECOGNITION

## FUTURE IMPLEMENTATIONS




- EVALUATING AND FINE TUNING OF THE MODEL
- GEOGRAPHIC LOCATION MAPPING
- AUTO MACHINE LEARNING IMPLEMENTATION TO ADD NEW PLANTS EASILY

# IT20089436

# SENARATHNE S.M.A.D

Software Engineering

# INTRODUCTION

# BACKGROUND

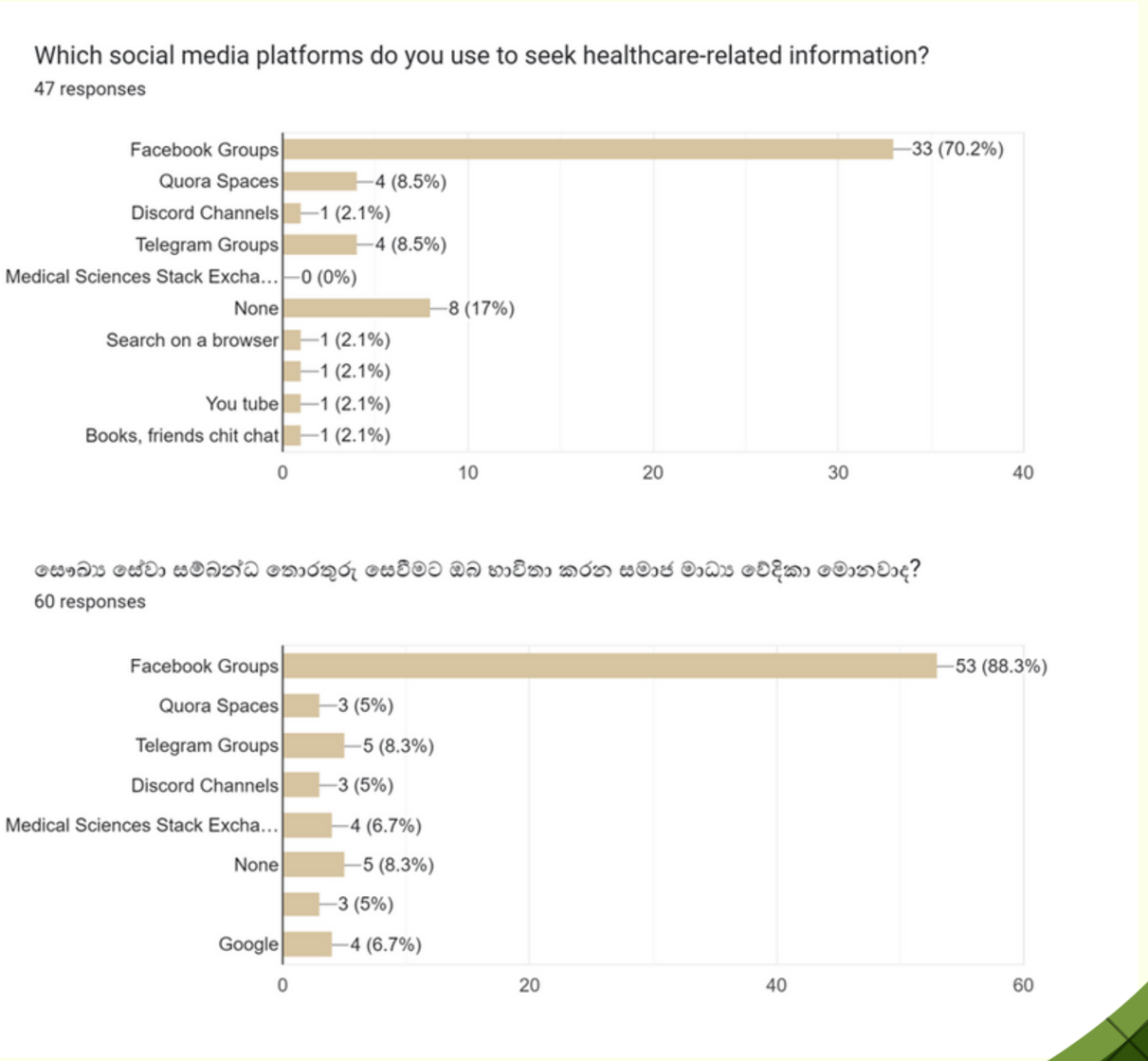
- Ayurveda, a traditional system of medicine, originated in India over 5,000 years ago.
- Ayurveda emphasizes the interconnectedness of mind, body, and spirit for overall wellness.
- Sri Lanka has a long-standing tradition of incorporating Ayurveda into its healthcare system.
- Challenges faced by individuals seeking Ayurvedic treatments include identifying the right practitioners.
- Consulting qualified Ayurvedic practitioners can be costly and time-consuming.
- The proposed solution is a user-friendly platform for personalized Ayurvedic treatment and diagnosis.
- The platform helps users find qualified doctors based on their specific symptoms.
- Users can also share ideas and experiences with each other through the platform.
- While some research on Ayurvedic treatments exists, there is a lack of information and accessibility, particularly outside of India.
- Sri Lanka has seen a growing interest in Ayurvedic medicine, but there is a need for a stronger connection between patients and doctors.

# INTRODUCTION

# RESEARCH QUESTION

## SURVEY FINDINGS ON INFORMATION-SEEKING BEHAVIOR FOR HEALTHCARE ON SOCIAL MEDIA PLATFORMS:

- 80.4% of respondents used Facebook groups.
- 6.4% of respondents used Quora spaces.
- 3.2% of respondents used Discord channels.
- 8.4% of respondents used Telegram groups.
- 2.1% of respondents used Medical Sciences Stack Exchange.
- 11.2% of respondents reported using no social media platform for healthcare-related information.

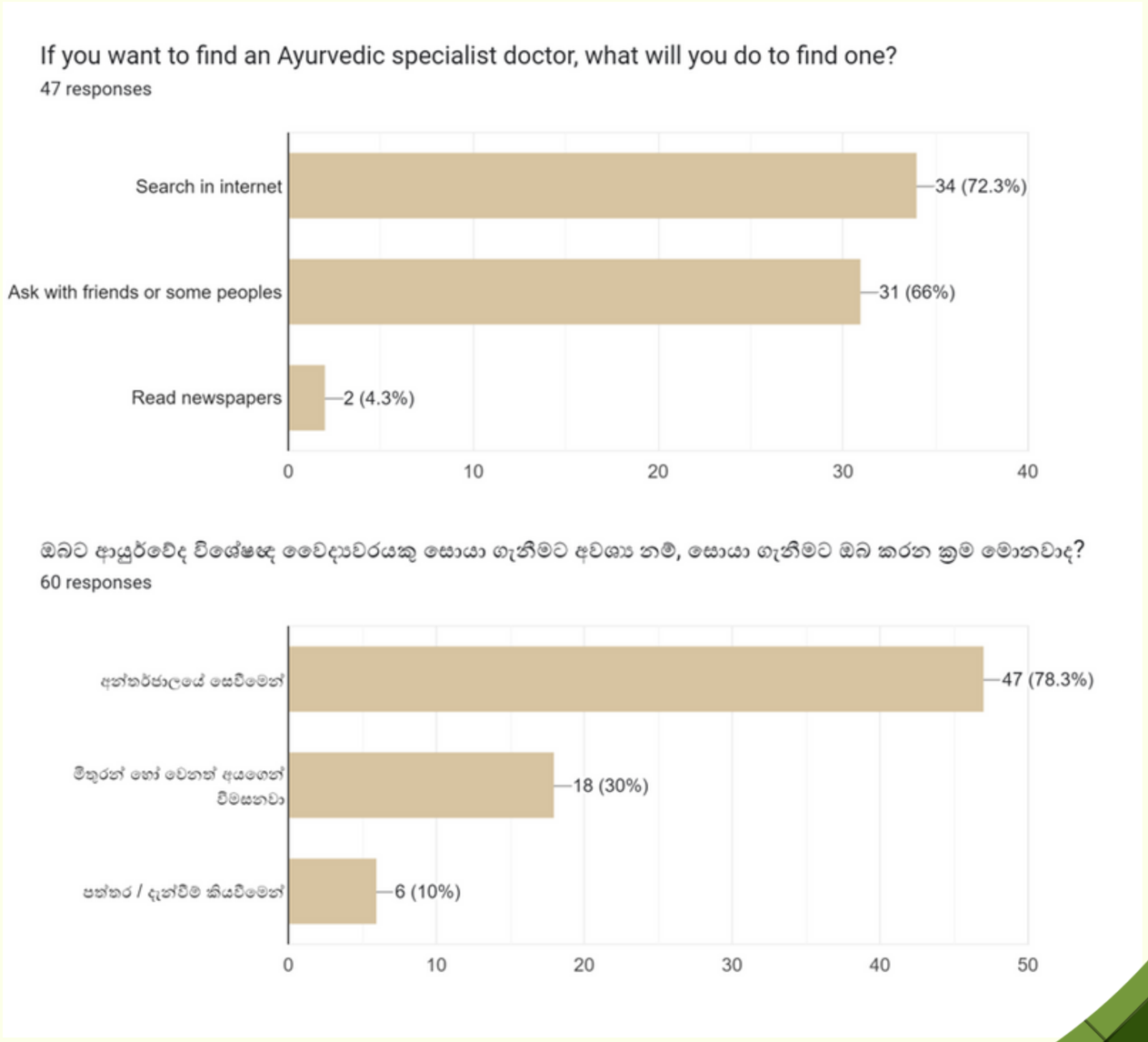


# INTRODUCTION

# RESEARCH QUESTION

**SURVEY RESULTS SHOW THAT 75.2% OF PEOPLE SEEKING AN AYURVEDIC SPECIALIST DOCTOR USE THE INTERNET FOR THEIR SEARCH.:**

- Survey results show that 75.2% of people seeking an Ayurvedic specialist doctor use the internet for their search.
- Only 48.6% of respondents rely on asking friends or people they know to find an Ayurvedic specialist.
- A small percentage of 7.9% still resort to traditional methods such as reading newspapers for finding Ayurvedic doctors.



# SPECIFIC & SUB OBJECTIVE

IMPLEMENTING A MACHINE LEARNING MODEL AND ALGORITHEMS THAT ANALYZES A PATIENT'S SYMPTOMS AND SUGGESTS THE MOST SUITABLE AYURVEDIC DOCTORS BASED ON THEIR SPECIALIZATION, RATINGS, AND PROXIMITY TO THE PATIENT'S LOCATION

DEFINE A RETRAIN PIPELINE TO TRAIN MODEL AGAIN WITH UPDATE THE DATA OF THE DOCTORS OR NEWLY ADDED DATA

- PRE-PROCESSING OF THE COLLECTED DOCTOR DATA
- SPLITTING THE DATA INTO TRAINING, VALIDATION, AND TESTING SETS.
- TRAINING A MACHINE LEARNING MODEL USING PRE-PROCESSED IMAGE DATA.
- EVALUATING THE MODEL ON THE VALIDATION SET TO IDENTIFY AREAS OF IMPROVEMENT.
- TESTING THE FINAL MODEL ON THE TESTING MODEL AND IMPLEMENTING IT IN A SOFTWARE APPLICATION FOR PRACTICAL USE.
- IMPLEMET THE MOBILE AND WEB APPLICTIONS

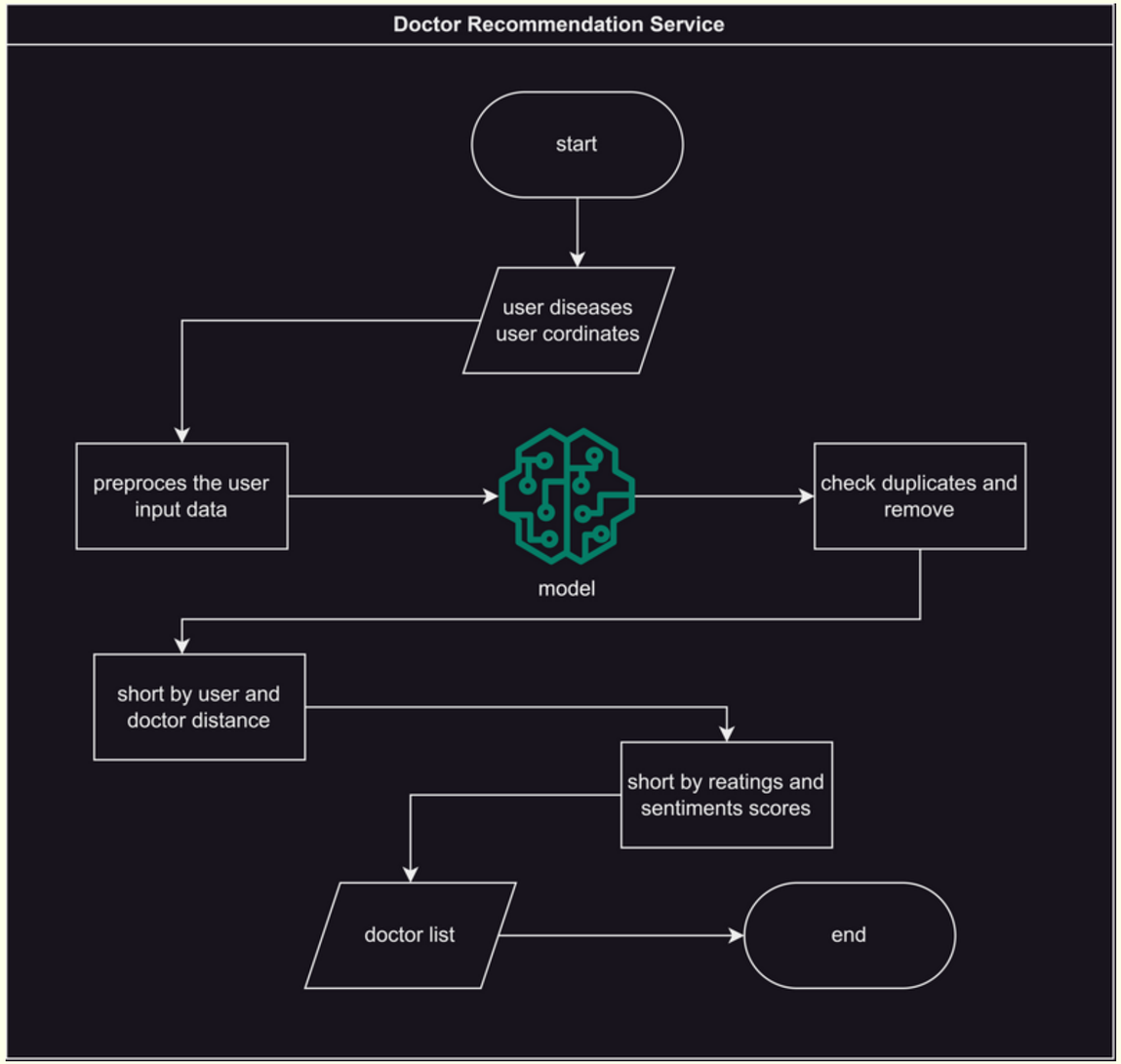
# OTHER OBJECTIVE

- INTEGRATING A FEEDBACK AND RATING SYSTEM INTO THE A
- DEVELOPING A PRIVATE CHAT FEATURE
- TRBUILDING A REGISTRATION SYSTEM THAT ALLOWS PATIENTS AND AYURVEDIC DOCTORS
- ENABLES PATIENTS TO VIEW THE AVAILABILITY OF AYURVEDIC DOCTORS AND BOOK APPOINTMENTS BASED ON THEIR SCHEDULE
- TESTING THE FINAL MODEL ON THE TESTING MODEL AND IMPLEMENTING IT IN A SOFTWARE APPLICATION FOR PRACTICAL USE.

# METHODOLOGY

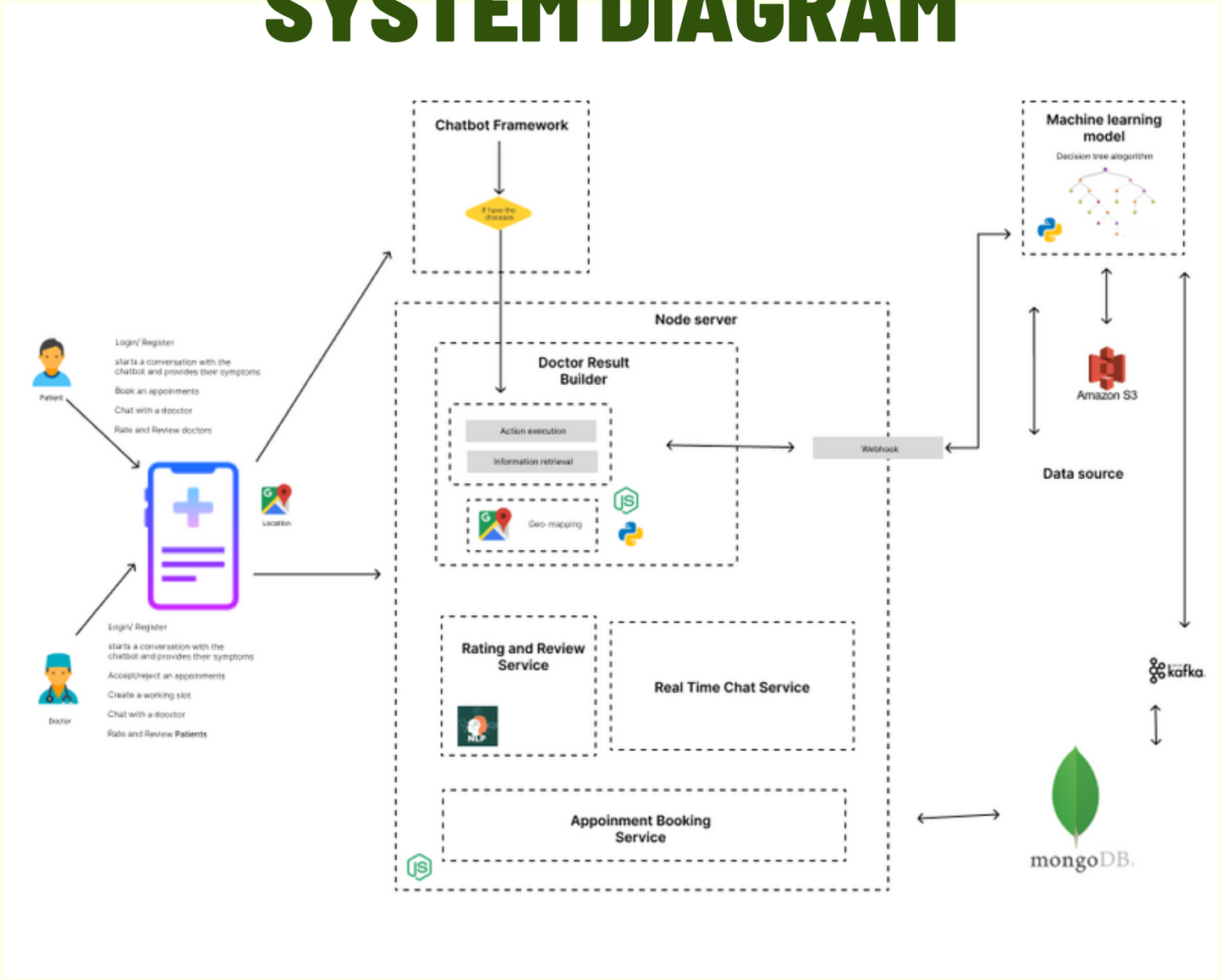
## FLOW CHART OF USED CUSTOM ALGORITHM

THE METHODOLOGY FOR THIS RESEARCH PROJECT WILL INVOLVE THE DEVELOPMENT OF A MOBILE APPLICATION TO HELP INDIVIDUALS SEEKING AYURVEDIC MEDICINE TO IDENTIFY THE MOST SUITABLE AYURVEDIC DOCTOR FOR THEIR SPECIFIC DISEASES. THE DEVELOPMENT OF THE APPLICATION WILL INVOLVE THE USE OF MACHINE LEARNING TECHNIQUES TO RECOMMEND THE BEST AYURVEDIC DOCTORS BASED ON RATINGS AND FEEDBACK PROVIDED BY OTHER USERS. THE APPLICATION WILL ALSO INCLUDE A FEATURE FOR USERS TO RATE AND GIVE FEEDBACK TO THE SELECTED AYURVEDIC DOCTORS AND BOOK APPOINTMENTS WITH THEM. ADDITIONALLY, THE APPLICATION WILL ENABLE USERS TO HAVE PRIVATE CHATS WITH DOCTORS.



# METHODOLOGY

# SYSTEM DIAGRAM



# METHODOLOGY (EVIDENCE OF COMPLETION)

- DATA COLLECTION
- DATA PRE-PROCESSING
- USE TRANSFER LEARNING BASED CNN ARCHITECTURES FOR TRAINING THE MODELS
- DATA AUGMENTATION AND SELECTED THE BEST TECHNIQUE TO RETRAIN MODELS TO ACHIEVE GOOD FITTING
- TUNING HYPERPARAMETERS AND RETRAINED MODELS TO SELECT THE BEST ARCHITECTURE
- DATA VISUALISATION
- HOST THE FINAL MODEL
- DISPLAYED THE RESULTS IN MOBILE APPLICATION

# METHODOLOGY (EVIDENCE OF COMPLETION)

## EVALUVATE THE DESITION TREE MODEL

```
Classification Report:
              precision    recall  f1-score   support

        1000       0.67      1.00      0.80         2
        1001       1.00      1.00      1.00         1
        1002       0.00      0.00      0.00         1
        1004       1.00      1.00      1.00         1

    accuracy                           0.80         5
   macro avg       0.67      0.75      0.70         5
weighted avg       0.67      0.80      0.72         5

Confusion Matrix:
[[2 0 0 0]
 [0 1 0 0]
 [1 0 0 0]
 [0 0 0 1]]
Accuracy Score:
0.8
```

**OUTPUT**

```
Doctor Id list :
 [1001 1000]
```

# METHODOLOGY (EVIDENCE OF COMPLETION)

## EVALUVATE THE K-NEAREST NEIGHBORS ALGORITHM

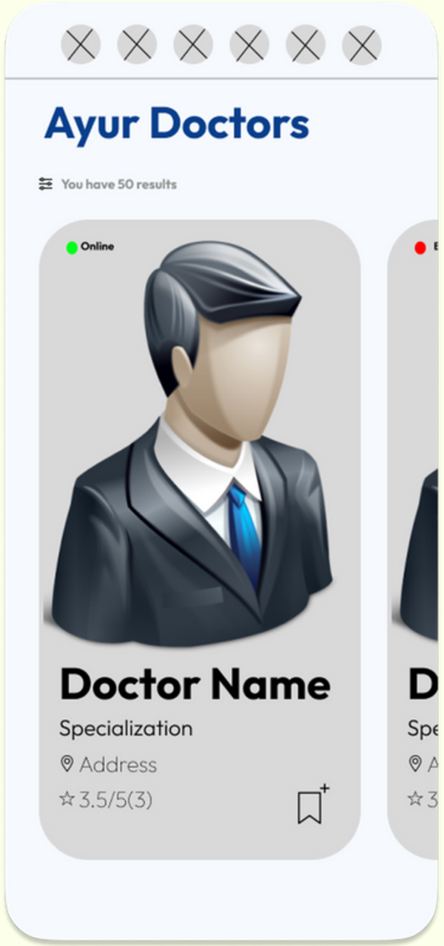
```
Classification Report:
              precision    recall  f1-score   support

        1000       0.40      1.00      0.57         2
        1001       0.00      0.00      0.00         1
        1002       0.00      0.00      0.00         1
        1004       0.00      0.00      0.00         1

    accuracy                           0.40         5
   macro avg       0.10      0.25      0.14         5
weighted avg       0.16      0.40      0.23         5

Confusion Matrix:
[[2 0 0 0]
 [1 0 0 0]
 [1 0 0 0]
 [1 0 0 0]]
Accuracy Score:
0.4
```

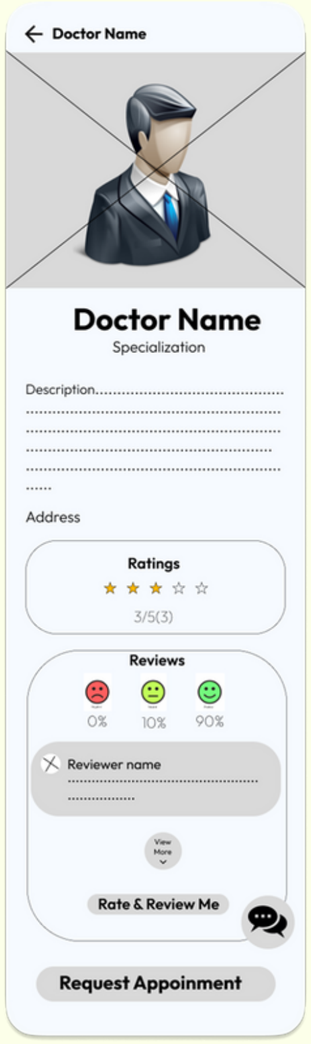
**OUTPUT**

```
Doctor Id list :
 [1000 1000]
```

# DISPLAY RESULTS IN MOBILE APPLICATIONS

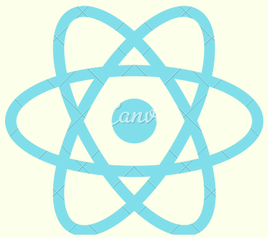
## MOBILE UI PROTOTYPES



# TECHNIQUES & TECHNOLOGIES

| | |
|---|---|
| Technologies | • React Native<br>• Python<br>• Expo<br>• Node Server |
| Techniques | • Machine Learning Model<br>• Sentiment Analysis<br>• Flask<br>• Apache Kafka |
| Algorithms | • Desition tree algorithm, Haversine formula |

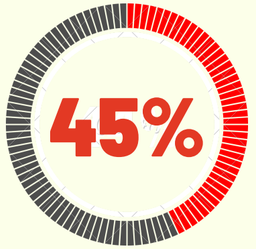



# COMPLETION AND FUTURE WORKS

## COMPLETION OF COMPONENTS

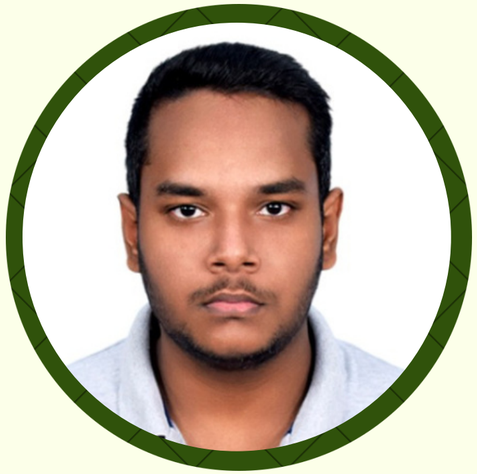

- DATA COLLECTION PRE-PROCESSING -
- CREATE DATA SETS
- CREATING THE ACTIONS BASED ON PREDICTIONS
- PROVIDING THE DISEASES BASED ON THE DOCTOR LIST

## FUTURE IMPLEMENTATIONS




- VERIFY THE PROVIDED ANSWERS USING ML MODLE
- VERIFY SUGGESTIONS WITH AYURVEDIC EXPERTS
- IMPLEMENTATION OF USER INTERFACES
- OTHER OBJECTIVES

- HOSTING - IN PROGRESS



# IT20216078

# JAYASINGHE J.A.S.C

Software Engineering

# BACKGROUND

- Implementation of a social network intended to share only health-related content.
- Allowed to ask questions and share articles.
- Allowed to leave any responses regarding posted content.
- Much information about the health domain may circulate throughout the social network.

# RESEARCH PROBLEM

- Most present social networks rely on services such as ElasticSearch for improved search functionality.
- Can it be achieved following a semantic search approach?
- Chatbot application uses a knowledge base to store information on various diseases, symptoms, and available treatments in Ayurvedic practice.
- How to explore and compare existing knowledge against real-world users' experiences sharing in a social network?

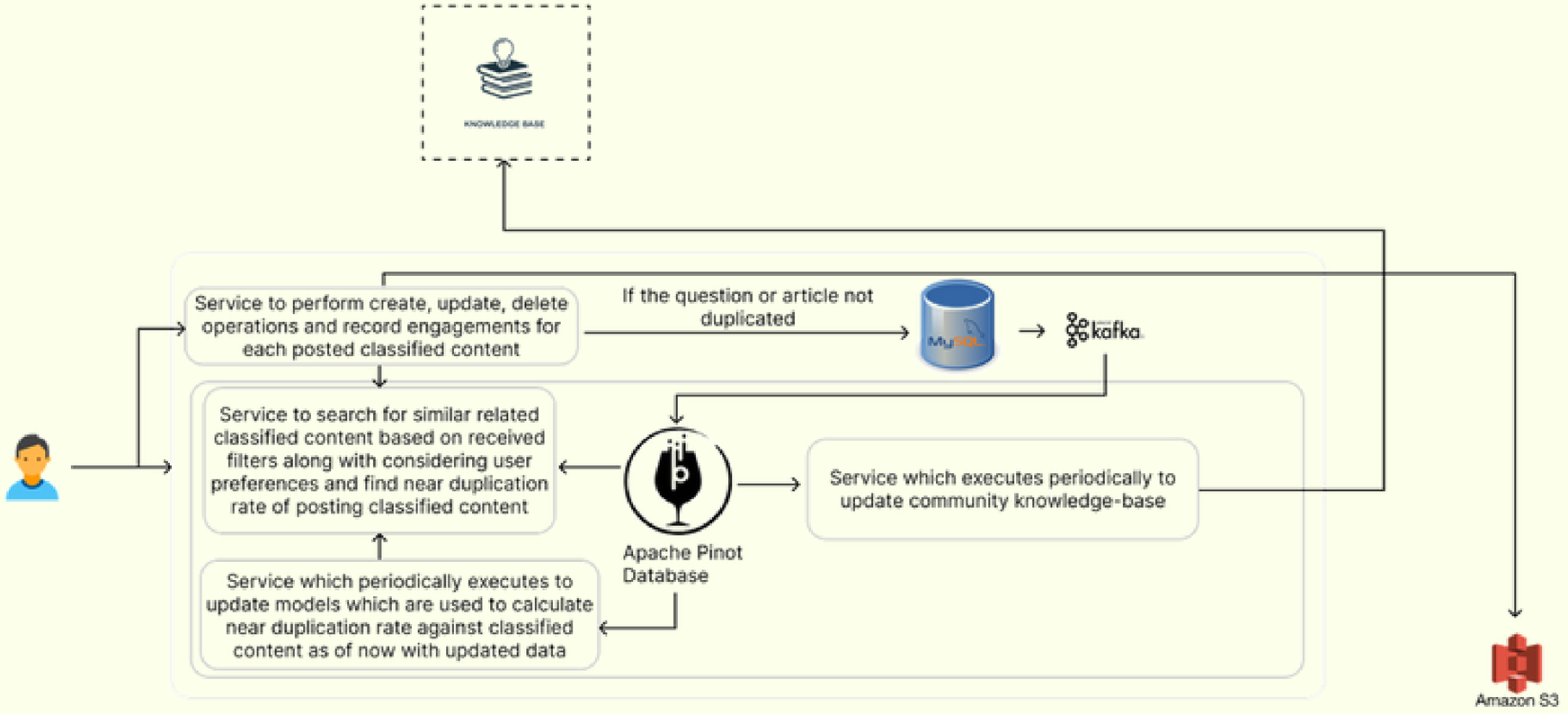
# TASK BREAKDOWN

- Implement a service with a REST API to save, update, and delete the content. – 10%
- Implement a service with a REST API to search the content.
  - The content should be able to be filtered based on user identification, tag (disease and symptoms), and user engagement rate. – 5%
  - The content should be processed to find near duplications against existing content considering semantic textual similarity and acknowledge the user. – 25%

# TASK BREAKDOWN

- Implement a scheduler service to update the existing knowledge base using extracted content in the social network and label it as community knowledge. – 30%
- Design user experience. – 15%
- Integration of developed models with APIs, deployment, and performance improvements for scheduler and content searching services. – 15%

# HIGH-LEVEL SYSTEM ARCHITECTURE DIAGRAM OF SOCIAL NETWORK




# ENHANCED CONTENT SEARCH BASED ON SEMANTIC SIMILARITY

- Most social networks, such as Stack Overflow, use Elasticsearch, a search engine based on the Apache Lucene library.
- Existing studies available that evaluate different machine learning and deep learning approaches to detect duplicating content in social networks such as Stack Overflow, Quora, and Twitter.
- In this project, it is expected to propose a solution based on a deep learning approach to detect semantic similarities in real-time.

| Question Group | Technique | Recall-Rate (%) | | |
|---|---|---|---|---|
| | | Top-5 | Top-10 | Top-20 |
| Java | SVM | 50.30 | 50.34 | 50.45 |
| | LR | 52.74 | 52.77 | 53.05 |
| | RF | 46.95 | 46.98 | 47.18 |
| | Xgboost | 53.79 | 53.90 | 53.98 |
| | WV-CNN | 81.27 | 81.27 | 81.30 |
| | WV-RNN | 65.17 | 65.20 | 65.25 |
| | WV-LSTM | **82.06** | **82.15** | **82.20** |
| C++ | SVM | 51.65 | 51.75 | 51.79 |
| | LR | 49.82 | 50.09 | 50.37 |
| | RF | 47.70 | 47.75 | 47.93 |
| | Xgboost | 52.76 | 52.90 | 53.08 |
| | WV-CNN | 80.01 | 80.06 | 80.10 |
| | WV-RNN | 60.16 | 60.20 | 60.25 |
| | WV-LSTM | **80.15** | **80.19** | **80.28** |
| Python | SVM | 56.47 | 56.59 | 56.99 |
| | LR | 56.30 | 56.41 | 56.82 |
| | RF | 54.41 | 54.58 | 55.15 |
| | Xgboost | 56.41 | 56.53 | 56.70 |
| | WV-CNN | 79.50 | 79.67 | 79.84 |
| | WV-RNN | 58.36 | 58.48 | 58.65 |
| | WV-LSTM | **79.61** | **79.78** | **80.01** |
| Ruby | SVM | 51.82 | 52.06 | 53.27 |
| | LR | 53.75 | 54.96 | 56.17 |
| | RF | 40.19 | 41.40 | 43.10 |
| | Xgboost | 54.00 | 54.72 | 56.42 |
| | WV-CNN | 76.76 | 77.00 | 77.72 |
| | WV-RNN | 61.26 | 61.50 | 62.71 |
| | WV-LSTM | **76.76** | **77.24** | **77.97** |
| Html | SVM | 58.59 | 58.76 | 59.17 |
| | LR | 56.68 | 56.93 | 57.26 |
| | RF | 52.03 | 52.20 | 53.20 |
| | Xgboost | 54.11 | 54.36 | 54.85 |
| | WV-CNN | 81.41 | 81.49 | 81.66 |
| | WV-RNN | 67.47 | 67.63 | 67.72 |
| | WV-LSTM | **81.58** | **81.74** | **81.91** |
| Objective-C | SVM | 41.59 | 41.69 | 41.80 |
| | LR | 55.05 | 55.27 | 55.59 |
| | RF | 41.15 | 41.48 | 41.59 |
| | Xgboost | 54.51 | 54.61 | 55.16 |
| | WV-CNN | 75.90 | 76.11 | 76.22 |
| | WV-RNN | 55.70 | 55.81 | 55.92 |
| | WV-LSTM | **78.39** | **78.61** | **78.94** |

# ENHANCED CONTENT SEARCH BASED ON SEMANTIC SIMILARITY

- Semantic search seeks to improve search accuracy by understanding the content of the search query. In contrast to traditional search engines, which only find documents based on lexical matches, semantic search can also find synonyms.
- The idea behind semantic search is to embed all entries in your corpus (existing records), whether sentences, paragraphs, or documents, into a vector space.
- At search time, the query is embedded into the same vector space, and the closest embeddings from your corpus are found. These entries should have a high semantic overlap with the query.
- There are two types of semantic searches.

# ENHANCED CONTENT SEARCH BASED ON SEMANTIC SIMILARITY

- For symmetric semantic search, your query and the entries in your corpus are about the same length and have the same amount of content. An example would be searching for similar questions: Your query could be “How to learn Python online?” and you want to find an entry like “How to learn Python on the web?”. For symmetric tasks, you could potentially flip the query and the entries in your corpus.
- For asymmetric semantic search, you usually have a short query (like a question or some keywords) and want to find a longer paragraph answering the query. An example would be a query like “What is Python,” and you want to see the section “Python is an interpreted, high-level and general-purpose programming language. Python’s design philosophy …”. For asymmetric tasks, flipping the query and the entries in your corpus usually does not make sense.
- Here it is used asymmetric semantic search, considering the solution’s nature.

# ENHANCED CONTENT SEARCH BASED ON SEMANTIC SIMILARITY

- Followed a transformers-based deep learning approach using BERT (Bidirectional Encoder Representations from Transformers) open-source framework. Several reasons encouraged the selection of this approach.
  - BERT models generate context-dependent embeddings that enable us to have various numeric representations for the same word, depending on its context. As a result, BERT embeddings capture contextual nuances that context-independent embeddings cannot.
  - BERT model considers the position of each word in a sentence as a direct input before computing its embedding.
  - BERT generates contextual embeddings that require a sentence as input instead of a single word. The BERT model requires knowledge of the surrounding terms to create a word vector, necessitating the presence of the trained model to produce embeddings based on input and context.

# ENHANCED CONTENT SEARCH BASED ON SEMANTIC SIMILARITY

- Following algorithmic steps have been followed.
  - Importing Required Libraries
  - Initializing FastAPI and Loading Sentence Transformer all-MiniLM-L6-v2 Model
  - Establishing a Connection to the Database
  - Defining Query Parameter Model
  - Caching Decorator for Encoded Embeddings. This decorator is used to cache the encoded corpus embeddings, avoiding redundant encoding operations
  - Retrieving and Caching Encoded Embeddings
  - Checking for Updates and Re-encoding Affected Records
  - Updating the Cached Embeddings with Affected Records
  - Endpoint for Retrieving Similarities
  - Starting the Update Thread

```python
# Cache decorator for caching encoded corpus embeddings
def cache_encoded_embeddings(func):
    encoded_embeddings = None
    corpus = None
    lock = threading.Lock()

    @functools.wraps(func)
    def wrapper(*args, **kwargs):
        nonlocal encoded_embeddings, corpus
        if encoded_embeddings is None:
            with lock:
                if encoded_embeddings is None:
                    corpus, encoded_embeddings = func(*args, **kwargs)
        return corpus, encoded_embeddings

    return wrapper
```

```python
# Update the cached embeddings with affected records
def update_embeddings(records):
    global corpus, corpus_embeddings

    # Extract the affected records
    affected_records = [record for record in records if record not in cached_records]

    # Re-encode the affected records
    affected_corpus = [record[1] for record in affected_records]
    affected_embeddings = embedder.encode(affected_corpus, convert_to_tensor=True)

    # Update the cached embeddings with the affected records
    corpus.extend(affected_corpus)
    corpus_embeddings = torch.cat([corpus_embeddings, affected_embeddings], dim=0)
```

# ENHANCED CONTENT SEARCH BASED ON SEMANTIC SIMILARITY

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/similar_sentences/?query=headache' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/similar_sentences/?query=headache
```

Server response

| Code | Details |
| --- | --- |
| 200 | Response body |

```
{
  "query": "headache",
  "top_sentences": [
    {
      "sentence": "I have a headache. Can it be Covid-19",
      "score": 0.6261075735092163
    }
  ]
}
```

Download

Response headers

```
content-length: 118
content-type: application/json
date: Tue,23 May 2023 07:05:18 GMT
server: uvicorn
```

# ENHANCED CONTENT SEARCH BASED ON SEMANTIC SIMILARITY

- Used hyperparameters
  - The SentenceTransformer model "all-MiniLM-L6-v2" is loaded, representing a pre-trained model based on the MiniLM architecture with six transformer layers. It is a relatively more minor model than the full version, balancing the model size and performance.
  - The top 5 similar sentences are returned by default, but the code ensures that the number of returned sentences does not exceed the length of the corpus.
  - The similarity threshold of 0.5 is used to assign predicted labels.
- It was able to retrieve an average recall value of 86.67 that the relevant sentences are those with a cosine similarity score above the similarity threshold. This is subjected to training and test data.

# ENHANCED CONTENT SEARCH BASED ON SEMANTIC SIMILARITY

```
Query: A child is doing their homework in the living room.

Top 5 most similar sentences in corpus:
A child is doing their homework. (Score: 0.8035)
A child is playing basketball. (Score: 0.3011)
A child is riding a bike. (Score: 0.2325)
A cat is playing with a mouse. (Score: 0.1685)
A cat is playing with a ball of yarn. (Score: 0.1637)


======================


Query: A dog is sleeping on the couch in the living room.

Top 5 most similar sentences in corpus:
A dog is sleeping on the couch. (Score: 0.8424)
A dog is playing fetch. (Score: 0.2586)
A cat is sitting on the windowsill. (Score: 0.2551)
A dog is chasing a cat. (Score: 0.2297)
A cat is playing with a mouse. (Score: 0.2081)


======================


Query: A cat is sitting on the windowsill in the bedroom.

Top 5 most similar sentences in corpus:
A cat is sitting on the windowsill. (Score: 0.8388)
A cat is playing with a mouse. (Score: 0.5189)
A cat is playing with a ball of yarn. (Score: 0.4085)
A dog is chasing a cat. (Score: 0.4001)
A dog is sleeping on the couch. (Score: 0.3167)
Recall value: 0.8666666666666667
```

```python
total_queries = len(queries)
relevant_sentences = 0
retrieved_sentences = 0

for query, query_embedding in zip(queries, query_embeddings):
    cos_scores = util.cos_sim(query_embedding, corpus_embeddings)[0]

    # Retrieve sentences with cosine similarity above the threshold
    relevant_indices = [
        idx for idx, score in enumerate(cos_scores) if score > threshold
    ]

    if len(relevant_indices) > 0:
        relevant_sentences += 1

    top_results = torch.topk(cos_scores, k=top_k)

    retrieved_indices = [idx.item() for idx in top_results[1]]

    retrieved_sentences += len(set(relevant_indices) & set(retrieved_indices))

recall = retrieved_sentences / relevant_sentences if relevant_sentences > 0 else 0.0

print("Recall:", recall)
```

# UPDATE THE EXISTING KNOWLEDGE BASE USING AVAILABLE CONTENT IN THE SOCIAL NETWORK

- Existing libraries and models available with higher F1 scores to extract diseases, symptoms, and treatments from a given sentence.
  - scispaCy
    - en_ner_craft_md
    - en_ner_jnlpba_md
    - en_ner_bc5cdr_md
    - en_ner_bionlp13cg_md
  - BioBERT
  - Apache cTAKES
  - MetaMap

```
Diseases: pneumonia
Symptoms: fever, cold, headache
```

```python
def extract_symptoms_diseases(text):
    nlp = spacy.load("en_ner_bc5cdr_md")
    doc = nlp(text)

    diseases = []
    symptoms = []

    for ent in doc.ents:
        if ent.label_ == "DISEASE":
            diseases.append(ent.text)
        elif ent.label_ == "SYMPTOM":
            symptoms.append(ent.text)

    return diseases, symptoms
```

# UPDATE THE EXISTING KNOWLEDGE BASE USING AVAILABLE CONTENT IN THE SOCIAL NETWORK

- Checking for existing disease and symptom nodes.

```
existing_diseases = session.run("MATCH (d:Disease) RETURN d.name AS name")

existing_symptoms = session.run("MATCH (s:Symptom) RETURN s.name AS name")

existing_diseases = set(record["name"] for record in existing_diseases)

existing_symptoms = set(record["name"] for record in existing_symptoms)
```

- Checking if the disease node already exists and creating it if not.

```
if disease not in existing_diseases:

    session.run("CREATE (d:Disease {name: $name})", name=disease)

    existing_diseases.add(disease)
```

# UPDATE THE EXISTING KNOWLEDGE BASE USING AVAILABLE CONTENT IN THE SOCIAL NETWORK

- Creating symptom nodes and relationships.
- Checking if the symptom node already exists and creating it if not

```
if symptom not in existing_symptoms:
    session.run("CREATE (s:Symptom {name: $name})", name=symptom)
    existing_symptoms.add(symptom)
```

- Creating a relationship between the disease and symptom nodes

```
session.run("MATCH (d:Disease {name: $disease}), (s:Symptom {name: $symptom}) "
            "CREATE (d)-[:HAS_SYMPTOM]->(s)",
            disease=disease, symptom=symptom)
```

# UPDATE THE EXISTING KNOWLEDGE BASE USING AVAILABLE CONTENT IN THE SOCIAL NETWORK

- Known limitations yet to be addressed
  - Existing libraries cannot extract Ayurvedic treatments and need to improve them.

# REMAINING WORK – 50%

- Implement a scheduler service to update the existing knowledge base using extracted content in the social network and label it as community knowledge.
  - Need to address the limitations stated in the previous slide
- Design user experience.
- Integration of developed models with APIs, deployment, and performance optimizations (caching) for scheduler and content searching services.

# REFERENCES


1. L. Wang, L. Zhang, and J. Jiang, "Duplicate Question Detection With Deep Learning in Stack Overflow," IEEE Access, vol. 8, pp. 25964 - 25975, 2020.
2. K. Denecke, "Extracting Medical Concepts from Medical Social Media with Clinical NLP Tools: A Qualitative Study," 2014.
3. S. Luca and N. Goharian, "QuickUMLS: a fast, unsupervised approach for medical concept extraction," 2014.
4. K. Denecke, "Extracting Medical Concepts from Medical Social Media with Clinical NLP Tools: A Qualitative Study," 2014


# Q & A

# THANK YOU