M.A.Shanika Sewwandi

Shanika.uok2@gmail.com

Abstract

The GradieBears Chatbot is an AI-powered web application designed to automate customer interactions for GradieBears, a Sri Lankan business specializing in customizable graduation teddy bears. The project combines natural language processing (NLP) and neural networks to create an intuitive, responsive chatbot that handles product inquiries, pricing, and customization requests in real time.

AI Chat bot

GradieBears Chatbot Documentation

**Content**

1. **Project overview**
2. **Technical architecture**
3. **Key components**
4. **Work flow explanation**
5. **Customization guide**
6. **Deployment note**
7. **Maintenance**
8. **Project overview**

**Business problem :**

GradieBears needed an automated solution to handle customer inquiries about their graduation teddy products 24/7 . including :

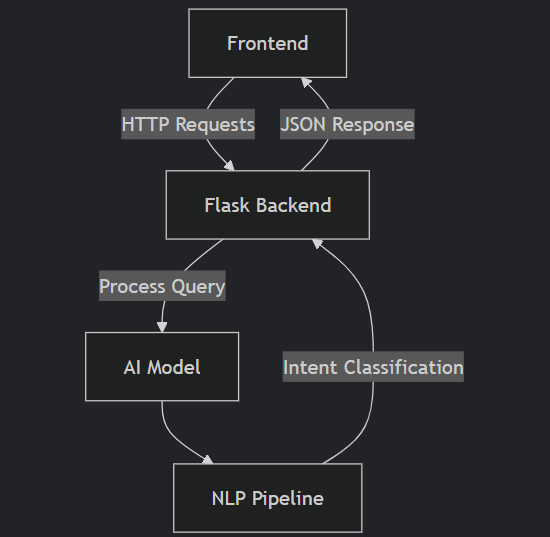
* Product pricing
* Customization options
* Delivery queries

**Solution :**

An AI – powered web chatbot that :

* Understanding natural language questions
* Provides instant responses
* Maintains conversation context
* Displays product in a compact interface

1. **Technical architecture**

****

1. **Key components**

**Natural network implementation**

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

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

dense (Dense) (None, 128) 2432

dropout (Dropout) (None, 128) 0

dense\_1 (Dense) (None, 64) 8256

dense\_2 (Dense) (None, 6) 390

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

Total params: 11,078

Trainable params: 11,078

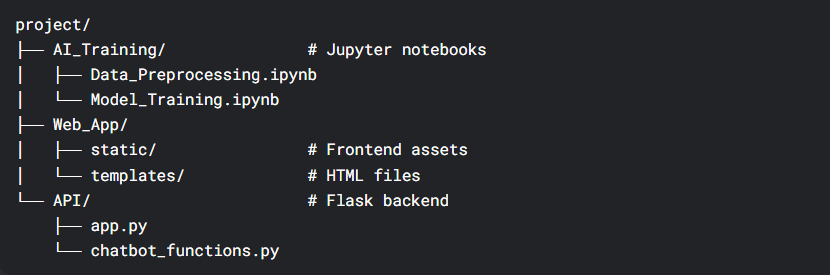
**Training process :**

1. Input : 200 + labeled text patterns
2. Preprocessing : tokenization 🡪 Lemmatization 🡪 Bag of words
3. Training : 200 epochs with early stopping

**Data files :**

|  |  |  |
| --- | --- | --- |
| File | Purpose | Sample content |
| Intents.json | Training patterns/ responses | json {"tag":"greeting","patterns":["Hello"],"responses":["Hi!"]} |
| words.pkl | Vocabulary bank | ["hello", "how", "much"] |
| classes.pkl | Intent categories | ["greeting", "price"] |

**Code structure :**



1. **Work flow explanation**

**User interaction flow :**

1. Input phase :

* User types query : “how much for premium pack ?”
* Frontend sends POST request to /ask endpoint

1. Processing phase :

def predict\_intent(text):

tokens = clean\_text(text) # Tokenization

bow = bag\_of\_words(tokens) # Convert to numbers

prediction = model.predict([bow]) # Neural network inference

return get\_response(prediction) # Match to intent

1. Output phase :

* Returns : “premium pack costs 2500 LKR (Hat + scrall + teddy + medal)”
* Renders in chat bubble with product cads visible

1. **Customization guide**

**Adding new products**

Edit intnets.json :

{

"tag": "new\_product",

"patterns": ["Do you have X?"],

"responses": ["We now offer X for Y LKR!"]

}

**Retrain model :**  python train.ipynb

**Modify prices**

1. Update response messages in intents.json
2. No retraining needed for simple price changes
3. **Deployment note**

Requirements:

* Python 3.8+
* TensorFlow 2.6+
* Flask 2.0+

Run Locally:

flask run --host=0.0.0.0 --port=5000

1. **Maintenance**

**Monitoring:**

* Log all conversations to chat\_logs.csv
* Weekly accuracy checks with test phrases

**Improvement Roadmap:**

1. Add Sinhala/Tamil support
2. Integrate with WhatsApp Business API
3. Implement order tracking system