

Course Chatbot

- CSV-based course chatbot (improved dataset)
- `train_model.py` to train the model
- Flask backend (`app.py`)
- HTML UI (`index.html`)
- Using `TfidfVectorizer` for better text handling

□ Project Structure

```
course-chatbot/  
├── intents.csv  
├── train_model.py  
├── app.py  
├── model.pkl  
├── vectorizer.pkl  
└── templates/  
    └── index.html
```

1 □ intents.csv (Improved Dataset)

question,intent,response

What courses do you offer,`course_list`,"We offer Python, Data Science, Web Development, MS Office, Tally with GST, Advanced Excel, Full Stack Development, and Digital Marketing courses."

List all courses,`course_list`,"We offer Python, Data Science, Web Development, MS Office, Tally with GST, Advanced Excel, Full Stack Development, and Digital Marketing courses."

Which courses are available,`course_list`,"We offer Python, Data Science, Web Development, MS Office, Tally with GST, Advanced Excel, Full Stack Development, and Digital Marketing courses."

Tell me about Python course,`python_info`,"Python course covers basics to advanced topics including projects."

Give details of Python course,`python_info`,"Python course covers basics to advanced topics including projects."

What is included in Python course,`python_info`,"Python course covers basics to advanced topics including projects."

What is fee for Python course,`python_fee`,"Python course fee is ₹10,000."

Python course fees,`python_fee`,"Python course fee is ₹10,000."

How much is Python course,`python_fee`,"Python course fee is ₹10,000."

Python course duration,`python_duration`,"Duration of Python course is 3 months."

How long is Python course,`python_duration`,"Duration of Python course is 3

months."

What is duration of Python course,python_duration,"Duration of Python course is 3 months."

Do you offer Data Science course,data_info,"Yes, we offer Data Science course with practical training."

Tell me about Data Science course,data_info,"Yes, we offer Data Science course with practical training."

Data Science course fees,data_fee,"Data Science course fee is ₹25,000."

What is fee for Data Science,data_fee,"Data Science course fee is ₹25,000."

Tell me about MS Office course,ms_info,"MS Office includes Word, Excel, PowerPoint and basic computer skills."

MS Office course details,ms_info,"MS Office includes Word, Excel, PowerPoint and basic computer skills."

MS Office fees,ms_fee,"MS Office course fee is ₹5,000."

What is fee for MS Office,ms_fee,"MS Office course fee is ₹5,000."

MS Office duration,ms_duration,"Duration is 1 month."

How long is MS Office course,ms_duration,"Duration is 1 month."

Tell me about Tally with GST,tally_info,"Tally course covers accounting, GST filing and business management."

Tally course details,tally_info,"Tally course covers accounting, GST filing and business management."

Tally course fees,tally_fee,"Tally course fee is ₹8,000."

What is fee for Tally,tally_fee,"Tally course fee is ₹8,000."

Tally course duration,tally_duration,"Duration is 2 months."

How long is Tally course,tally_duration,"Duration is 2 months."

Tell me about Advanced Excel,excel_info,"Advanced Excel covers formulas, pivot tables, dashboards and automation."

Advanced Excel details,excel_info,"Advanced Excel covers formulas, pivot tables, dashboards and automation."

Advanced Excel fees,excel_fee,"Advanced Excel course fee is ₹7,000."

What is Excel course fee,excel_fee,"Advanced Excel course fee is ₹7,000."

Excel duration,excel_duration,"Duration is 1.5 months."

How long is Excel course,excel_duration,"Duration is 1.5 months."

Tell me about Full Stack Development,fullstack_info,"Full Stack includes frontend and backend technologies like HTML, CSS, JavaScript and Python."

Full stack course details,fullstack_info,"Full Stack includes frontend and backend technologies like HTML, CSS, JavaScript and Python."

Full stack fees,fullstack_fee,"Full Stack course fee is ₹30,000."

What is fee for full stack,fullstack_fee,"Full Stack course fee is ₹30,000."

Full stack duration,fullstack_duration,"Duration is 6 months."

How long is full stack course,fullstack_duration,"Duration is 6 months."

Tell me about Digital Marketing,dm_info,"Digital Marketing includes SEO,

```
social media marketing and ads."
Digital marketing course details,dm_info,"Digital Marketing includes SEO,
social media marketing and ads."
```

```
Digital marketing fees,dm_fee,"Digital Marketing course fee is ₹20,000."
What is fee for digital marketing,dm_fee,"Digital Marketing course fee is
₹20,000."
```

```
Digital marketing duration,dm_duration,"Duration is 3 months."
How long is digital marketing course,dm_duration,"Duration is 3 months."
```

```
Hi,greeting,"Hello! How can I help you?"
Hello,greeting,"Hi there! Ask me anything about courses."
Hey,greeting,"Hello! How can I help you?"
```

2 train_model.py (Train Model)

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
import joblib

# Load CSV
data = pd.read_csv('intents.csv')

# Drop empty rows
data = data.dropna(subset=['question'])
data = data.reset_index(drop=True)

# Features & labels
X = data['question'].astype(str)
y = data['intent']

# Vectorizer
vectorizer = TfidfVectorizer()
X_vec = vectorizer.fit_transform(X)

# Model
model = LogisticRegression(max_iter=1000)
model.fit(X_vec, y)

# Save
joblib.dump(model, 'model.pkl')
joblib.dump(vectorizer, 'vectorizer.pkl')

print("Model trained successfully!")
```

Chatbot into a **ChatGPT-style web UI** with:

- **A scrollable chat window**
- **User and bot messages** clearly separated
- **Typing animation** for bot responses
- **Fallback response** if the bot doesn't understand

I'll give the **full updated HTML + minor Flask tweaks**.

1 app.py (Add fallback & JSON response)

```
from flask import Flask, render_template, request, jsonify
import joblib
import pandas as pd

app = Flask(__name__)

# Load model and vectorizer
model = joblib.load('model.pkl')
vectorizer = joblib.load('vectorizer.pkl')

# Load CSV responses
data = pd.read_csv('intents.csv')

@app.route('/')
def home():
    return render_template('chat.html') # renamed template

@app.route('/get', methods=['POST'])
def chatbot():
    user_input = request.form['message']

    # Transform input
    input_vec = vectorizer.transform([user_input])

    # Predict intent
    intent = model.predict(input_vec)[0]

    # Get response
    responses = data[data['intent'] == intent]['response'].values

    if len(responses) > 0:
        response = responses[0]
    else:
        response = "Sorry, I didn't understand. Can you rephrase?"

    return jsonify({'response': response})
```

```
if __name__ == "__main__":
    app.run(debug=True)
```

2 New `templates/index.html` (ChatGPT-style UI)

```
<!DOCTYPE html>
<html>
<head>
  <title>Course Enquiry Chatbot</title>
  <style>
    body { font-family: Arial; background: #f0f0f0; }
    .chat-container { width: 400px; margin: 50px auto; background: #fff;
border-radius: 8px; padding: 20px; box-shadow: 0 0 10px #aaa; }
    .chat-box { height: 400px; overflow-y: auto; border: 1px solid #ddd;
padding: 10px; }
    .user-msg { text-align: right; margin: 10px 0; }
    .bot-msg { text-align: left; margin: 10px 0; }
    .msg { display: inline-block; padding: 8px 12px; border-radius: 15px;
max-width: 80%; }
    .user-msg .msg { background: #0084ff; color: white; }
    .bot-msg .msg { background: #e5e5ea; color: black; }
    #userInput { width: 80%; padding: 10px; }
    #sendBtn { width: 18%; padding: 10px; background: #0084ff; color:
white; border: none; cursor: pointer; }
  </style>
</head>
<body>

<div class="chat-container">
  <h3>Course Enquiry Bot</h3>
  <div class="chat-box" id="chatBox"></div>
  <input type="text" id="userInput" placeholder="Ask something...">
  <button id="sendBtn">Send</button>
</div>

<script>
const chatBox = document.getElementById('chatBox');
const input = document.getElementById('userInput');
const btn = document.getElementById('sendBtn');

function appendMessage(sender, message) {
  const div = document.createElement('div');
  div.classList.add(sender + '-msg');
  div.innerHTML = `<span class="msg">${message}</span>`;
  chatBox.appendChild(div);
  chatBox.scrollTop = chatBox.scrollHeight;
}

function sendMessage() {
  const message = input.value;
  if (message.trim() === '') return;

  appendMessage('user', message);
```

```
input.value = '';

// Typing animation
appendMessage('bot', 'Typing...');
const typingDiv = chatBox.lastChild;

fetch("/get", {
  method: "POST",
  headers: { "Content-Type": "application/x-www-form-urlencoded" },
  body: "message=" + encodeURIComponent(message)
})
.then(response => response.json())
.then(data => {
  typingDiv.remove(); // remove typing
  appendMessage('bot', data.response);
});
}

btn.addEventListener('click', sendMessage);
input.addEventListener('keypress', function(e) {
  if(e.key === 'Enter') sendMessage();
});
</script>

</body>
</html>
```

3 Features Added

- Scrollable chat window
 - User and bot messages in bubbles
 - “Typing...” animation before bot responds
 - Enter key also sends message
 - Fallback message if intent not found
-

4 Run Steps

1. Make sure your **model is trained**:

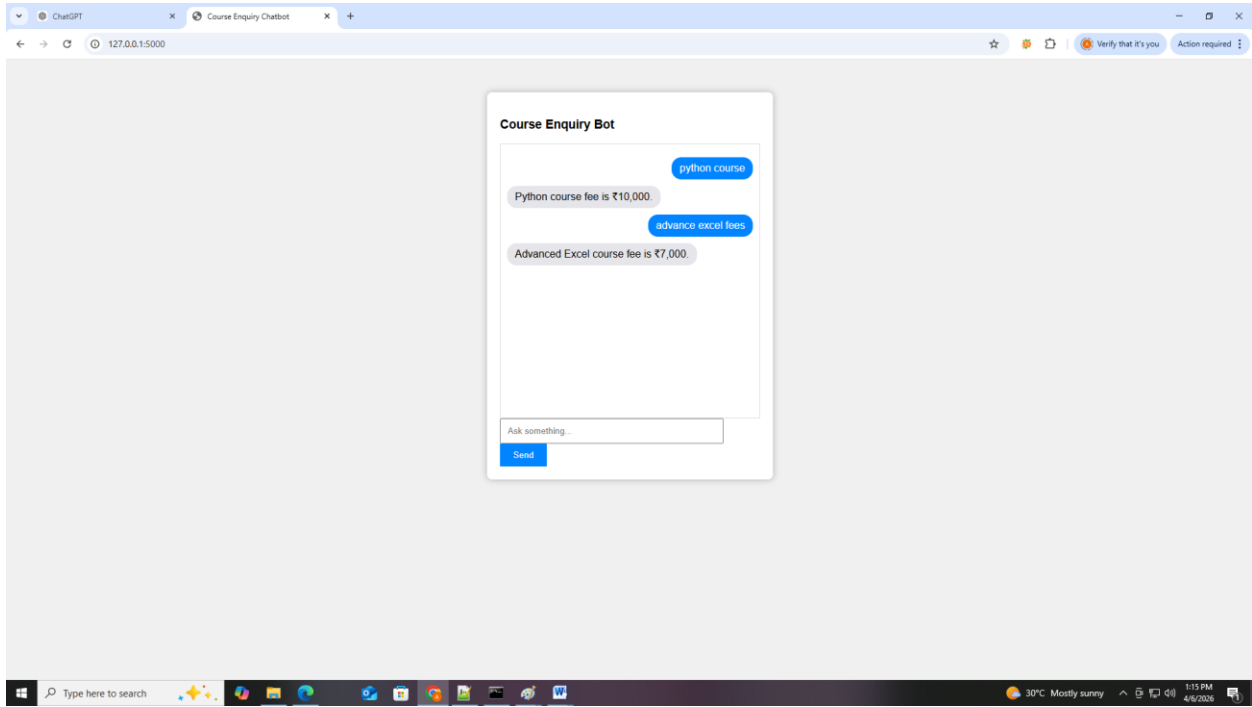
```
python train_model.py
```

2. Run Flask:

```
python app.py
```

3. Open browser:

http://127.0.0.1:5000/
Output:-



Locally:-

```
pip install -r requirements.txt
```

1. Push project on github using git:

```
git init
git add .
git commit -m "initial commit"
git push -u origin main
```

Deploy to Render

1. Sign up at render.com
2. Connect your GitHub repo
3. Set start command: `gunicorn app:app`
4. Deploy → Get public URL