

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**JNANA SANGAMA, BELAGAVI-590018**



**MINI PROJECT SOURCE CODE**

**(21ADMP67)**

**ON**

**“AI-CHATBOT FOR DEPARTMENT”**

*Submitted in partial fulfilment of the requirements for the award of*

*the degree of*

**BACHELOR OF ENGINEERING**

**IN**

**ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

Submitted by

**HARSHITHA N (1KG21AD012)**

**MANYASHREE U J (1KG21AD028)**

**PRAJWAL C L (1KG21AD036)**

**JYOTHISH K S (1KG22AD401)**

Under the Guidance of

**Prof. Harshitha H S**

Assistant Professor

Department of AI&DS

K.S.S.E.M, Bengaluru



**Department of Artificial Intelligence and Data Science**

**K. S. School of Engineering and Management**

**No 15, Mallasandra, off. Kanakapura Road, Bengaluru – 560109**

**2023-2024**

# SOURCE CODE

## HTML Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>KSSEM</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="chat-container">
    <div class="chat-header">
      <h1>Department of AI&DS</h1>
    </div>
    <div class="chat-box" id="chat-box"></div>
    <input type="text" id="user-input" placeholder="Type a message...">
    <button onclick="sendMessage()">Send</button>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

## Java Script Code:

```
function appendMessage(text, className) {
    const chatBox = document.getElementById('chat-box');
    const messageElement = document.createElement('div');
    messageElement.className = `chat-message ${className}`;
    messageElement.textContent = text;
    chatBox.appendChild(messageElement);
    chatBox.scrollTop = chatBox.scrollHeight;
}

async function sendMessage() {
    const userInput = document.getElementById('user-input');
    const message = userInput.value;
    if (message.trim() === "") return;

    appendMessage(`You: ${message}`, 'user-message');
    userInput.value = "";

    const response = await fetch('http://127.0.0.1:5000/chat', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify({ message: message })
    });

    const data = await response.json();
    appendMessage(`AI&DS Chatbot: ${data.response}`, 'bot-message');
}
```

## CSS Code:

```
body {  
  font-family: Arial, sans-serif;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  height: 100vh;  
  background: linear-gradient(to right, #0056a3, #7ed6df);  
  margin: 0;  
}
```

```
.chat-container {  
  background-color: #f5f5f5;  
  width: 500px;  
  box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);  
  border-radius: 12px;  
  overflow: hidden;  
}
```

```
.chat-box {  
  height: 450px;  
  overflow-y: auto;  
  padding: 16px;  
  border-bottom: 1px solid #ddd;  
}
```

```
.chat-message {  
  text-align: left;  
  margin-bottom: 16px;  
  line-height: 1.4;  
  padding: 8px 12px;  
  border-radius: 12px;  
  max-width: 80%;  
  display: inline-block;
```

```
    word-wrap: break-word;
}

.user-message {
    text-align: right;
    color: #fff;
    background-color: #3d8dd3;
    align-self: flex-end; /* Keep messages on the right */
}

.bot-message {
    text-align: left;
    color: #fff;
    background-color: #56ca3e;
    align-self: flex-start; /* Align bot messages on the left */
}

.input-container {
    display: flex;
    border-top: 1px solid #ddd;
    padding: 8px;
}

.chat-header h1 {
    font-size: 32px;
    color: #2c89e0;
    margin-bottom: 10px;
    display: flex;
    align-items: center;
    justify-content: center;
}

.chat-header h1 a {
    text-decoration: none;
    color: #333;
```

```
}
```

```
.chat-header h1 a:hover {  
    text-decoration: underline;  
}
```

```
input[type="text"] {  
    width: calc(100% - 25px);  
    padding: 12px;  
    border: none;  
    outline: none;  
    font-size: 14px;  
    border-top-left-radius: 12px;  
    border-bottom-left-radius: 12px;  
    border-top-right-radius: 12px;  
    border-bottom-right-radius: 12px;  
    box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.1);  
    transition: background-color 0.3s;  
}
```

```
input[type="text"]:focus {  
    background-color: #e0e0e0;  
}
```

```
button {  
    width: 80px;  
    padding: 16px;  
    border: none;  
    background-color: #3585cb;  
    color: white;  
    cursor: pointer;  
    border-top-right-radius: 12px;  
    border-bottom-right-radius: 12px;  
    border-top-left-radius: 12px;  
    border-bottom-left-radius: 12px;
```

```
    transition: background-color 0.3s, transform 0.3s;  
}
```

```
button:hover {  
    background-color: #3585cb;  
    transform: scale(1.05);  
}
```

## Python Code:

```
from flask import Flask, request, jsonify
from flask_cors import CORS
import google.generativeai as genai
app = Flask(__name__)
CORS(app) # This will enable CORS for all routes
GOOGLE_API_KEY = "AIzaSyDbVU4bZ0ypTeennw-0jKdvHFxyd6tDYf0"
genai.configure(api_key=GOOGLE_API_KEY)
model = genai.GenerativeModel('gemini-pro')
def chat_with_gemini(text):
    kb = """
Database(knowledge base)
    reply you give will be directly given to the user so give reply accordingly
    """
    content = f"""
    You are a chatbot for the Department of AI & DS in KS School of Engineering and
    Management.
    If the user requests anything other than department info, reply accordingly.
    Here is the knowledge base: {kb}
    This is the question from the user: {text}
    """
    response = model.generate_content(content)
    response.resolve()
    return response.text
@app.route('/chat', methods=['POST'])
def chat():
    user_input = request.json.get("message")
    response_text = chat_with_gemini(user_input)
    return jsonify({"response": response_text})
if __name__ == '__main__':
    app.run(debug=True)
```