

1. Code.gs 파일 소스코드

```
// PotensAI 웹앱 서비스
// 카카오톡 스타일 메신저 인터페이스를 통해 PotensAI API와 대화할 수 있는 웹앱

// API 키와 엔드포인트 설정
const API_KEY"iu39Wh2fgwwbpc7WeWCiWtvfoeCqyGMZ"; // 포텐스닷 API 키
const API_ENDPOINT"https://ai.potens.ai/api/chat"; // 포텐스닷 API 엔드포인트

// 웹앱 도메인 설정을 위한 함수
function doGet() {
    return HtmlService.createHtmlOutputFromFile('index')
        .setTitle('PotensAI 챗봇')
        .setXFrameOptionsMode(HtmlService.XFrameOptionsMode.ALLOWALL)
        .addMetaTag('viewport', 'width=device-width, initial-scale=1');
}

// PotensAI API에 요청을 보내는 함수
function callPotensAI(userMessage) {
    try{
        // API 요청 페이로드 구성
        const payload= {
            prompt: userMessage

            // API 요청 옵션 설정
            const options= {
                method: 'post',
                contentType: 'application/json',
                headers: {
                    'Authorization': `Bearer ${API_KEY}`
                },
            },

            payload: JSON.stringify(payload),
            muteHttpExceptions: true

            // API 요청 실행
            const responseUrlFetchApp.fetch(API_ENDPOINT, options);
            const responseCoderesponse.getResponseCode();
            const responseTextresponse.getContentText();

            // 응답 처리
            if(responseCode=== 200) {
                try{
```

```

const responseDataJSON.parse(responseText);
// API 응답 구조에 따라 필요한 데이터 추출
// 실제 API 응답 구조에 맞게 수정 필요
return{
  success: true,
  message: responseData.response || responseData.answer ||
responseData.message || responseText
};
} catch(parseError) {
  // JSON 파싱 오류 처리
  console.error("JSON 파싱 오류:", parseError);
  return{
    success: true,
    message: responseText // JSON이 아닌 경우 텍스트 그대로 반환
  };
}
} else{
  // API 오류 처리
  console.error("API 응답 오류:", responseCode, responseText);
  return{
    success: false,
    message: "죄송합니다. 응답을 처리하는 중에 오류가 발생했습니다. (상태 코드:
"responseCode")"
  };
}
} catch(error) {
  // 예외 처리
  console.error("API 호출 중 오류 발생:", error);
  return{
    success: false,
    message: "죄송합니다. 서비스 연결에 문제가 발생했습니다. 잠시 후 다시 시도해주세요."
  };
}
}

```

2. index.html 파일 소스코드

```
<!DOCTYPEhtml>
<html>
<head>
  <base target="_top">
  <meta charset="UTF-8">
  <title>PotensAI 챗봇</title>
  <style>
    /* 전체 스타일 */
    body{
      font-family: 'Apple SD Gothic Neo', 'Malgun Gothic', sans-serif;
      margin: 0;
      padding: 0;
      background-color: #b2c7da;
      color: #000;
      height: 100vh;
      display: flex;
      justify-content: center;
    }

    /* 채팅 컨테이너 */
    .chat-container{
      width: 100%;
      max-width: 450px;
      height: 100%;
      display: flex;
      flex-direction: column;
      background-color: #b2c7da;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }

    /* 헤더 스타일 */
    .chat-header{
      background-color: #a1b2c3;
      padding: 15px;
      text-align: center;
      font-weight: bold;
      color: #fff;
      position: relative;
      border-bottom: 1px solid #8b9baa;
    }
```

```

/* 메시지 영역 */
.chat-messages{
  flex: 1;
  overflow-y: auto;
  padding: 15px;
  display: flex;
  flex-direction: column;
}

/* 메시지 버블 공통 스타일 */
.message{
  max-width: 70%;
  padding: 10px 15px;
  margin-bottom: 10px;
  border-radius: 15px;
  word-wrap: break-word;
  position: relative;
  animation: fadeIn 0.3s ease;
  white-space: pre-wrap;
}

@keyframes fadeIn{
  from{opacity: 0; transform: translateY(10px); }
  to{opacity: 1; transform: translateY(0); }
}

/* 사용자 메시지 */
.user-message{
  align-self: flex-end;
  background-color: #ffeb33;
  border-top-right-radius: 0;
}

/* AI 메시지 */
.ai-message{
  align-self: flex-start;
  background-color: #ffffff;
  border-top-left-radius: 0;
}

/* 메시지 시간 표시 */

```

```
.message-time{
  font-size: 10px;
  color: #888;
  margin-top: 5px;
  display: block;
}

/* 입력 영역 */
.chat-input-container{
  display: flex;
  padding: 10px;
  background-color: #ffffff;
  border-top: 1px solid #ddd;
}

/* 텍스트 입력 */
.chat-input{
  flex: 1;
  padding: 10px;
  border: none;
  border-radius: 20px;
  background-color: #f0f0f0;
  outline: none;
  resize: none;
  font-family: inherit;
  margin-right: 10px;
  min-height: 20px;
  max-height: 100px;
}

/* 전송 버튼 */
.send-button{
  background-color: #ffeb33;
  color: #000;
  border: none;
  border-radius: 50%;
  width: 40px;
  height: 40px;
  display: flex;
  align-items: center;
  justify-content: center;
  cursor: pointer;
}
```

```

    font-size: 16px;
    transition: background-color 0.2s;
}

.send-button:hover{
    background-color: #e6d430;
}

.send-button:disabled{
    background-color: #f0f0f0;
    color: #aaa;
    cursor: not-allowed;
}

/* 로딩 애니메이션 */
.loading{
    display: flex;
    align-items: center;
    padding: 10px 15px;
    background-color: #ffffff;
    border-radius: 15px;
    border-top-left-radius: 0;
    align-self: flex-start;
    margin-bottom: 10px;
}

.loading-dot{
    width: 8px;
    height: 8px;
    background-color: #aaa;
    border-radius: 50%;
    margin: 0 3px;
    animation: bounce 1.4s infinite ease-in-out both;
}

.loading-dot:nth-child(1) { animation-delay: -0.32s; }
.loading-dot:nth-child(2) { animation-delay: -0.16s; }

@keyframes bounce{
    0%, 80%, 100%transform: scale(0); }
    40%transform: scale(1); }

```

```

    /* 반응형 디자인 */
    @media(max-width:600px) {
    .chat-container {
        max-width: 100%;
        height: 100vh;
    }

    .message{
        max-width: 80%;
    }
    }
    </style>
</head>
<body>
    <div class="chat-container">
        <div class="chat-header">
PotensAI 챗봇
        </div>
        <div class="chat-messages" id="chatMessages">
            <!-- 초기 AI 메시지 -->
            <div class="message ai-message">
안녕하세요! PotensAI입니다. 무엇을 도와드릴까요?
                <span class="message-time">지금</span>
            </div>
        </div>
        <div class="chat-input-container">
            <textarea class="chat-input" id="userInput" placeholder="메시지를 입력하세요..." rows="1"></textarea>
            <button class="send-button" id="sendButton">
                <span>↑</span>
            </button>
        </div>
    </div>

    <script>
        // DOM 요소
        const chatMessages=document.getElementById('chatMessages');
        const userInput=document.getElementById('userInput');
        const sendButton=document.getElementById('sendButton');

        // 메시지 전송 함수

```

```

function sendMessage() {
  const message = userInput.value.trim();
  if(!message) return;

  // UI에 사용자 메시지 추가
  addMessageToUI('user', message);
  userInput.value = '';
  userInput.style.height = 'auto';

  // 전송 버튼 비활성화
  sendButton.disabled = true;

  // 로딩 표시 추가
  const loadingElement = document.createElement('div');
  loadingElement.className = 'loading';
  loadingElement.innerHTML = `
<div class="loading-dot"></div>
<div class="loading-dot"></div>
<div class="loading-dot"></div>
`;
  chatMessages.appendChild(loadingElement);
  chatMessages.scrollTop = chatMessages.scrollHeight;

  // API 호출
  google.script.run
  .withSuccessHandler(function(result) {
    // 로딩 표시 제거
    loadingElement.remove();

    // 전송 버튼 활성화
    sendButton.disabled = false;

    // 응답 처리
    if(result.success) {
      addMessageToUI('ai', result.message);
    } else{
      addMessageToUI('ai', "죄송합니다. 오류가 발생했습니다. 다시 시도해주세요.");
    }

    // 입력 필드에 포커스
    userInput.focus();
  });
}

```



```

}))
.withFailureHandler(function(error) {
    // 로딩 표시 제거
    loadingElement.remove();

    // 전송 버튼 활성화
    sendButton.disabled=false;

    // 오류 처리
    console.error("Error:", error);
    addMessageToUI('ai', "죄송합니다. 서비스 연결에 문제가 발생했습니다.");

    // 입력 필드에 포커스
    userInput.focus();
})
.callPotensAI(message);
}

// UI에 메시지 추가 함수
function addMessageToUI(sender, text) {
    const messageElement=document.createElement('div');
    messageElement.className`message ${sender===
'user'?'user-message':'ai-message'}`;

    // 메시지 내용 설정
    messageElement.innerTexttext;

    // 시간 표시 추가
    const timeElement=document.createElement('span');
    timeElement.className'message-time';
    timeElement.innerTextgetCurrentTime();
    messageElement.appendChild(timeElement);

    // 메시지 추가 및 스크롤
    chatMessages.appendChild(messageElement);
    chatMessages.scrollTopchatMessages.scrollHeight;
}

// 현재 시간 포매팅 함수
function getCurrentTime() {
    const nownew Date();
    let hoursnow.getHours();

```

```

const minutesnow.getMinutes();
const ampmhours>= 12'오후' '오전';

hourshours12;
hourshourhours12; // 0시는 12시로 표시

return `${ampm} ${hours}:${minutes < 10'0'minutesminutes}`;
}

// 이벤트 리스너
sendButton.addEventListener('click', sendMessage);

userInput.addEventListener('keypress', function(e) {
  // Enter 키 누를 때 메시지 전송 (Shift+Enter는 줄바꿈)
  if(e.key=== 'Enter'&& !e.shiftKey) {
    e.preventDefault();
    sendMessage();
  }
});

// 텍스트 영역 자동 높이 조절
userInput.addEventListener('input', function() {
  this.style.height'auto';
  const newHeightMath.min(this.scrollHeight, 100); // 최대 높이 제한
  this.style.heightnewHeight'px';
});

// 페이지 로드 시 입력 필드에 포커스
window.onloadfunction() {
  userInput.focus();
}
</script>
</body>
</html>

```