

Configuration Manual

MSc Research Project
In Financial Technology

Qingyun Cai

Student ID: 23403519

School of Computing
National College of Ireland

Supervisor: Prof. Victor Del Rosal

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Qingyun Cai

Student ID: 23403519

Programme: MSc in Financial Technology (FinTech)

Year: 2024/2025

Module: Research Project

Supervisor Name: Prof. Victor Del Rosal

Due Date: 11th of August 2025

Project Title: The Role of AI Chatbots in Enhancing Efficiency and Customer Experience in Banking Call Centers

Word Count: 985

Page Count: 8

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Qingyun Cai

Date: 11/08/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

MSc Research Project in Financial Technology

Qingyun Cai

Student ID: 23403519

School of Computing

National College of Ireland

Supervisor: Prof. Victor Del Rosal

1. Introduction / Overview

This configuration manual provides a comprehensive description of the technical configuration and deployment of the PrimeBank AI Assistant, a web-based AI Chatbot for the banking industry. The chatbot was developed using HTML, CSS and JavaScript and integrated with the fine-tuned GPT API from OpenAI to provide real-time, empathetic, domain-specific responses. The purpose of this solution is to automate customer service and reduce call center load, while maintaining a high level of customer experience.

This manual also meets the requirements of Section 5.5.4 of the Practicum Handbook by providing transparency, reproducibility, and clarity to both academic and technical reviewers.

2. System Requirements

Hardware Requirements:

- Operating System: Windows 10+ or macOS Catalina+ or Ubuntu 20.04+
- Processor: Intel Core i5 or similar
- Memory: Minimum 8 GB or more
- Storage: 1 GB or more
- Display: Native 1080p or better

Software Requirements:

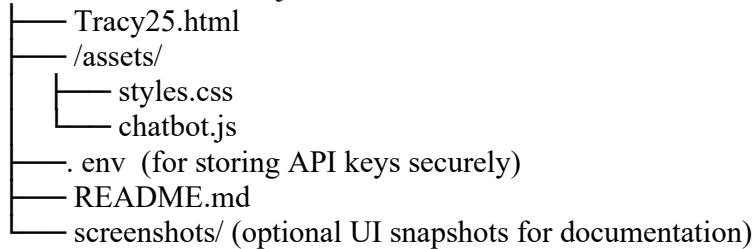
- Browser: Google Chrome v120+, Mozilla Firefox v110+, or Microsoft Edge v100+
- Editor: Visual Studio Code or similar IDE that supports HTML/JS
- API Access: OpenAI account with API key and fine-tuned model's access
- Internet: Stable connection (at least 5 Mbps) for API requests

Dependencies:

- JavaScript enabled in browser
- .env file or browser variable to keep API key secret
- Optional: (Optional) Local web server to serve for testing (e.g. Live Server extension for VS Code)
Tested on macOS and Windows.

3. Project Folder & File Structure

PrimeBankChatbotProject/



- tracy25.html: The main page, which contains the user interface markup as well as code to connect to the API.
- styles.css: The style definitions which contain all of the branding, layout and responsiveness of the web page.
- chatbot.js: The main logic for sending/receiving API messages.
- .env: Not used directly in browser but referenced for secure variable ideas

4. Installation & Setup Instructions

1. Download or Clone the Repository
2. Open the project in Chrome, Firefox, or Edge
3. Insert OpenAI API Key into the JavaScript fetch block within Tracy25.html
4. Test in Browser by entering queries into the Chatbot
5. Capture screenshots for documentation

Security Note: API keys should never be hardcoded in production, Use environment variables or proxy functions

```
// ChatGPT API integration
async function callChatGPTAPI() {
  // IMPORTANT: In a real production environment, you should NEVER expose your API key in client-side code
  // This should be handled through a backend service. This is for demonstration purposes only.

  const apiKey = 'sk-proj-lorOkwRminknLnzqueKxf0gFFC1iLs-z2YgP2ckWYkb-CcIdwKr9WNRSHXMwrohkB0vrLVPh1T3B1bkFJDE0z6ZiGOSLD'

  try {
    const response = await fetch('https://api.openai.com/v1/chat/completions', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
        'Authorization': `Bearer ${apiKey}`
      },
      body: JSON.stringify({
        model: "gpt-3.5-turbo",
        messages: chatSessions[currentChatId]?.messages || [],
        temperature: 0.7,
        max_tokens: 100 // Limit response length
      })
    });
  }
};
```

Figure 1 Insert OpenAI API Key to code

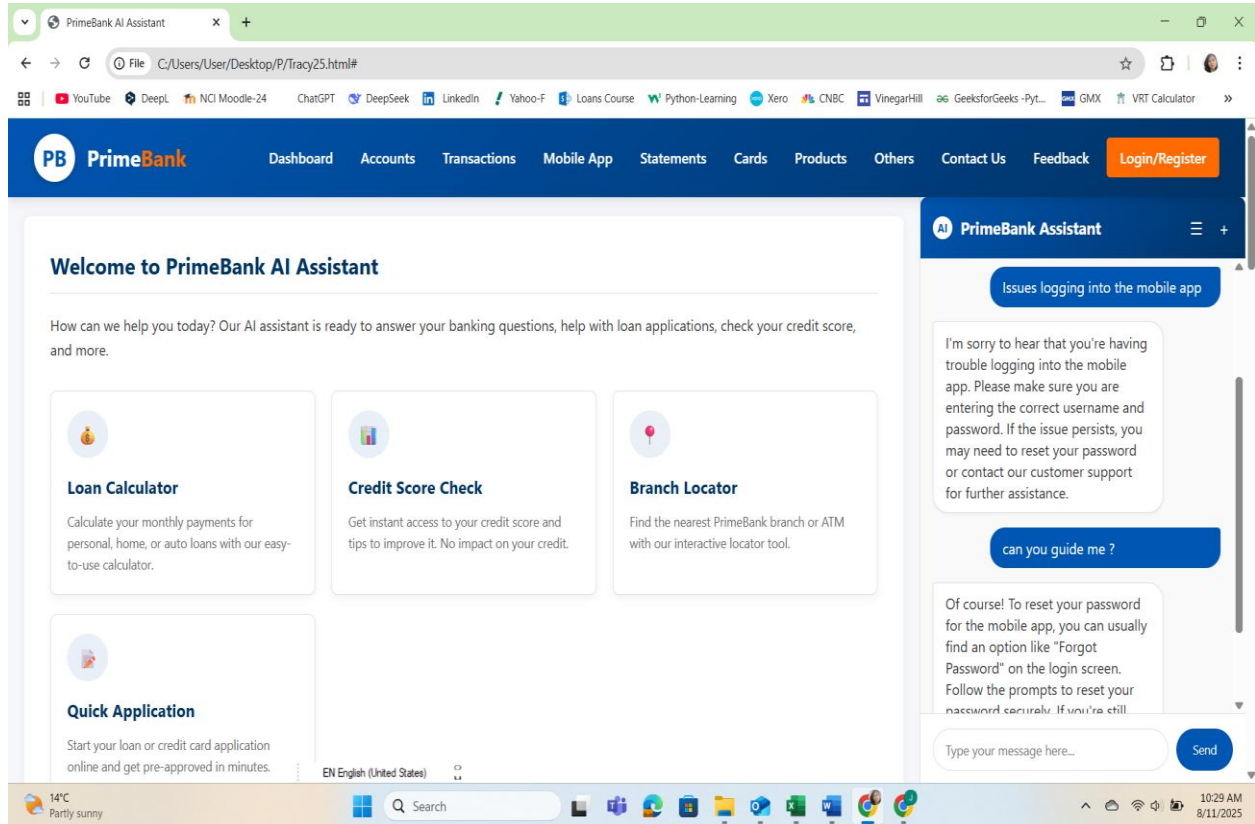


Figure 2 Open the Demo in Chrome

5. Fine-Tuning and Model Integration

- Dataset Preparation: 1,000+ prompt-completion pairs in JSONL format
- Fine-Tuning Parameters: Base model gpt-4-1-mini, 3 epochs, learning rate multiplier 0.1
- Integration Snippet:

```
fetch("https://api.openai.com/v1/chat/completions", {
  method: "POST",
  headers: {
    "Authorization": `Bearer YOUR_API_KEY`,
    "Content-Type": "application/json"
  },
  body: JSON.stringify({
    model: "ft:gpt-4-1-mini:primebank",
    messages: [{ role: "user", content: "I lost my debit card." }]
  })
})
```

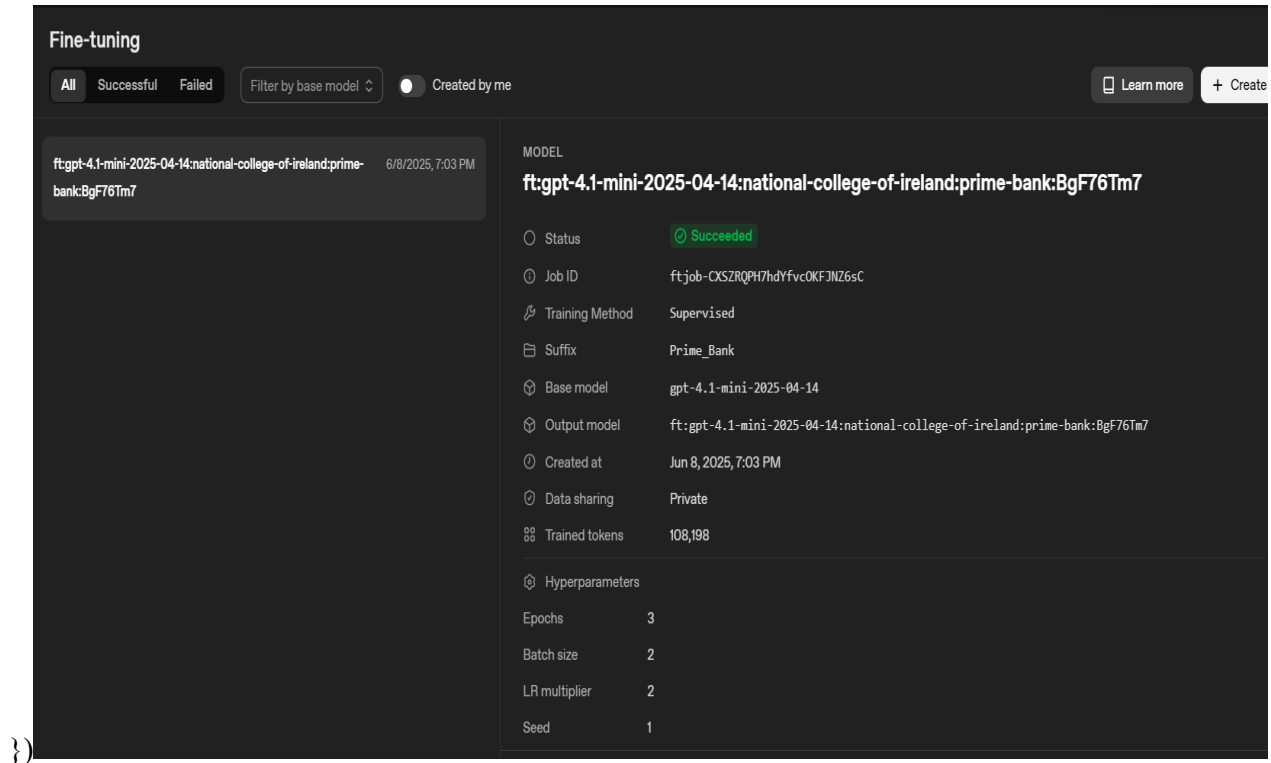


Figure 3 Fine Tuning

6. Interface and User Experience (UX)

- Layout: Header, central chatbox, input field with send button
- Responsive Design: Uses flexbox and media queries for desktop/mobile
- UX Enhancements: Color-coded chat bubbles, typing indicator, auto-scroll

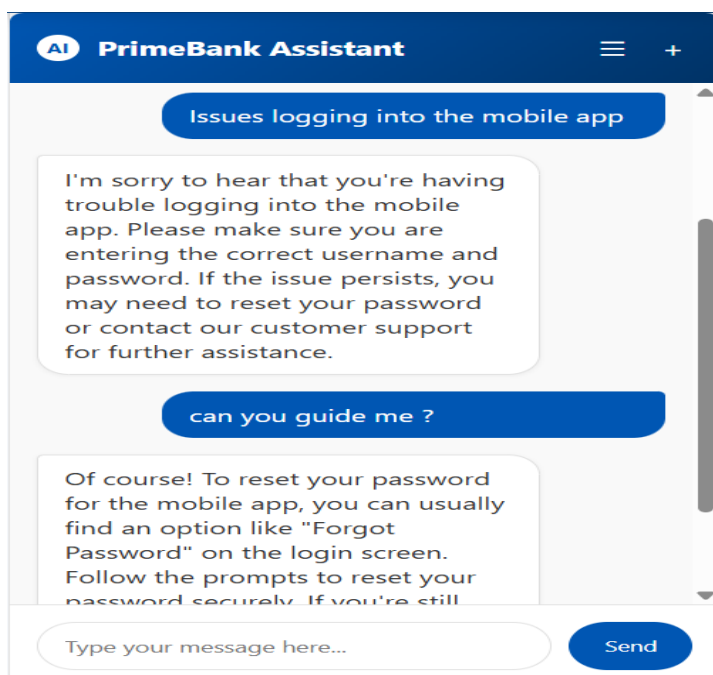


Figure 4 Chatbot chat 1

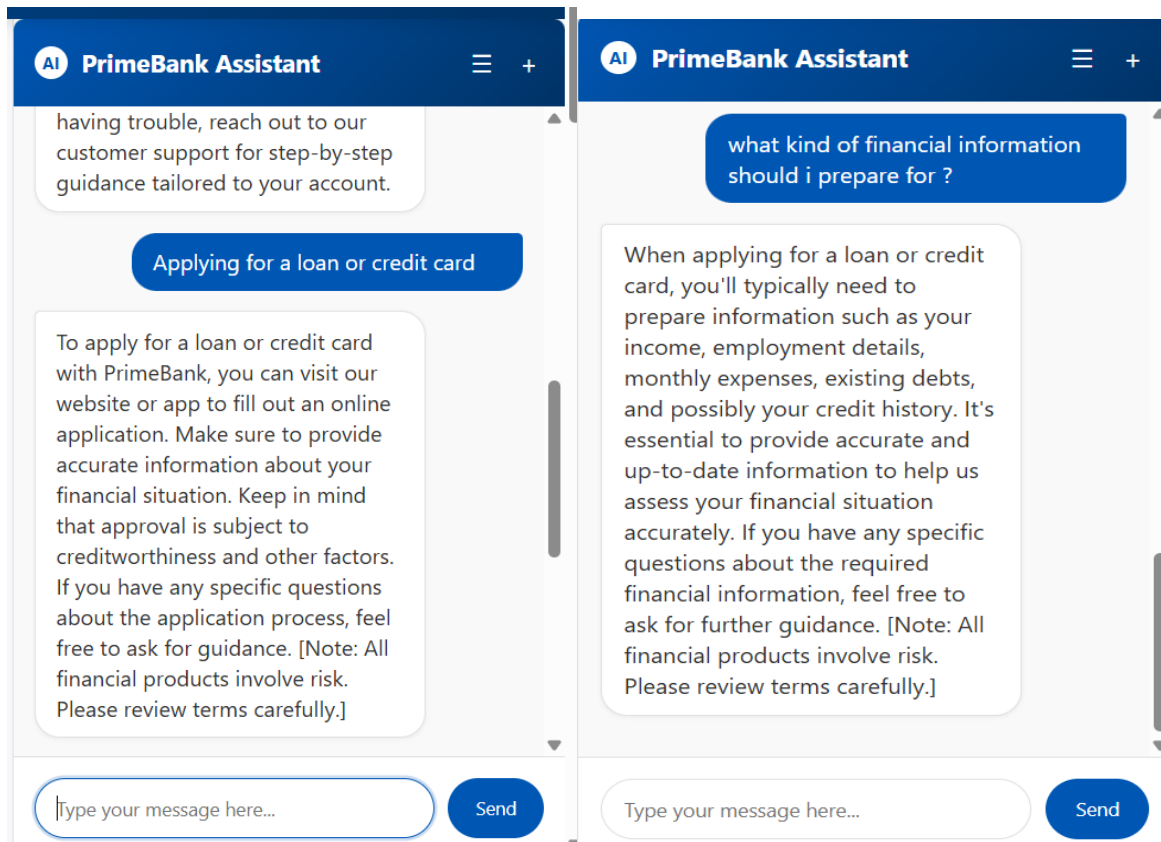


Figure 5 Chatbot chat 2

7. Testing and Evaluation

- Functional Testing: Chrome/Firefox under different network conditions
- User Testing: 10 participants, 5–10 scenarios each
- Evaluation Metrics: Accuracy, tone & empathy, contextual relevance

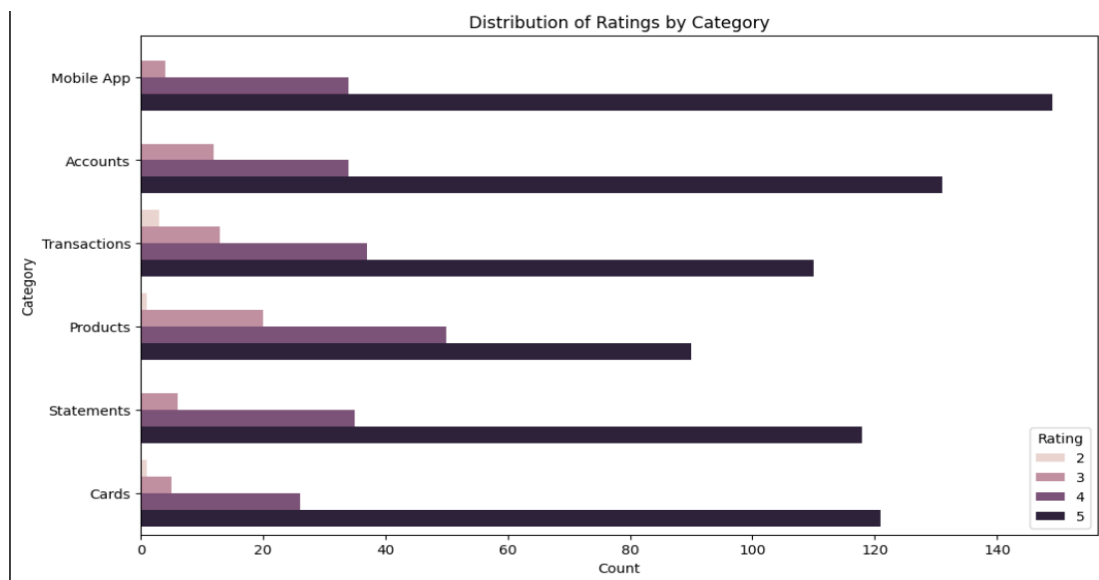


Figure 6 Distribution of Ratings by Category

8. Troubleshooting and Debugging

Issue	Cause	Solution
API not responding	Invalid or expired OpenAI API key	Regenerate key from OpenAI dashboard and update in code
Chatbot interface not loading	Incorrect HTML/CSS/JavaScript file paths	Verify file paths and ensure all assets are in correct folders
Irrelevant chatbot responses	Wrong model ID or fine-tuned model not linked	Update API request to use the correct fine-tuned model ID
UI not displaying properly	Browser compatibility issues	Use the latest version of Chrome or Firefox
Slow or delayed responses	Network latency or API rate limits reached	Check internet connection and monitor API usage quota
CORS errors when calling API	Cross-origin request blocked by browser	Enable CORS in server settings or use proxy for API requests

9. API key Configuration

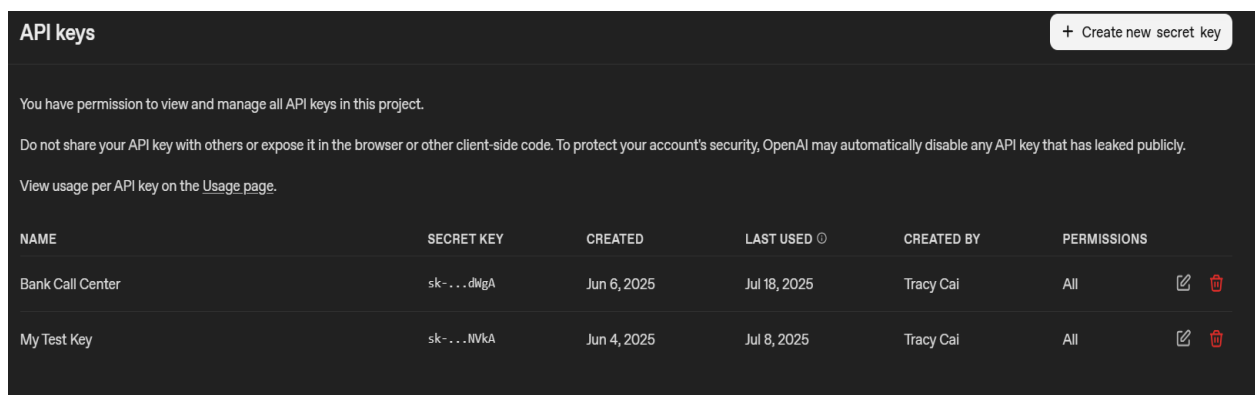
-Go to OpenAI developer platform: (<https://platform.openai.com>) Login with your account credentials.

-Accessing API Keys: Click on the user icon in the top right corner after logging in, then go to Settings> API Keys.

-Create a New Key: Click. Generate a new secret key. OpenAI will never show this key again, so name it (such as “Savemore Integration”) and copy it immediately.

-Secure Your Key: Store it securely in your backend (such as in a private configuration file or Firebase environment variables).

-Use the key in your app: Use the key to authenticate requests to OpenAI’s API in your backend (e.g. Python, Node.js, etc.).



API keys						+ Create new secret key
You have permission to view and manage all API keys in this project.						
Do not share your API key with others or expose it in the browser or other client-side code. To protect your account's security, OpenAI may automatically disable any API key that has leaked publicly.						
View usage per API key on the Usage page.						
NAME	SECRET KEY	CREATED	LAST USED	CREATED BY	PERMISSIONS	
Bank Call Center	sk-...dwgA	Jun 6, 2025	Jul 18, 2025	Tracy Cai	All	 
My Test Key	sk-...NVkA	Jun 4, 2025	Jul 8, 2025	Tracy Cai	All	 

Figure 7 Set up API Key