National College of Ireland

Configuration Manual

MSc Research Project

In Financial Technology

Keshav Chhabra

Student ID: 23290684

School of Computing

National College of Ireland

Supervisor: Victor Del Rosal

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Keshav Chhabra

Student ID: 23290684

Programme: MSc in Financial Technology (FinTech)

Year: 2024/2025

Module: Practicum

Supervisor Name: Victor Del Rosal

Due Date: 11th August 2025

Project Title: An AI-Based Artefact to Improve Financial Literacy and Savings in Ireland

Word Count: 977

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: Keshav Chhabra

Date: 11th August 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):	

Configuration Manual

Name: Keshav Chhabra

Student ID: 23290684

Program: MSc in Fintech

1. Introduction

This configuration handbook provides a comprehensive step-by-step instruction for installing, configuring, and running the Savemore artefact which is a web-based AI-powered financial literacy and savings tools designed for the students, freelancers, gig workers, and small business owners in Ireland. The manual highlights the system requirements, installation steps, configuration settings, usage instructions, and troubleshooting tips. If you can follow the steps of this guide thoroughly it will successfully take you to the functional prototype environment ready for load testing and appraisal.

2. System Requirements

- **Operating System:** Windows 10 or above / macOS
- **Browser:** Chrome / Firefox / Edge (latest versions)
- **Backend:** Python Python 3.10+, Google Colab/Visual studio
- **Frontend:** HTML, CSS, JavaScript
- **Database:** Firebase authentication
- **Other:** OpenAI API access, Git, Node.js (optional)

3. Technology Stack

Table 1: Savemore's Technical Stack and Component Functions

Component	Technology Used	Purpose
Frontend design	HTML, CSS, JavaScript	Interactive and user-friendly design
Backend Logic	Python	Routing, logic, and API integration
AI Recommendation	ChatGPT API	Realistic and real-time approach.
Database	Firebase Realtime	Secure user data, goal tracking, activity tracking
Authentication	Firebase Auth	Secure login and access
Survey Engine	Google Forms + Sheets	User feedback suggestions

4. Project Structure

/savemore/

index.html # Single HTML entry point

styles.css # Styling file

app.js # Application logic (UI, Firebase Auth, Charts)

/assets/ # Images, icons, and screenshots

/data/ # Synthetic dataset and survey results

5. Dashboard Preview

To preview the Dashboard.html file locally before deploying, you can use Python's built-in HTTP server module:

1. Open a terminal in the project directory containing Dashboard.html.
2. Run the following command (Python required):

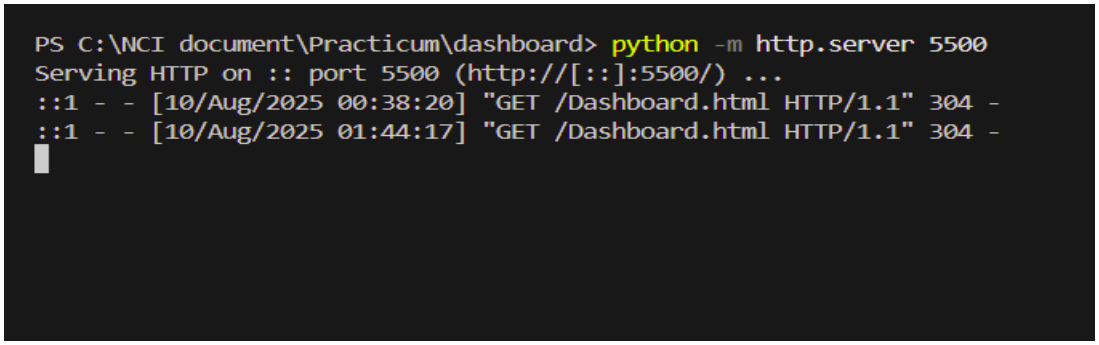


Figure 1: Dashboard preview process

6. Firebase Authentication Setup

Step-1 Create a Firebase project in the Firebase Console.

Step-2 Register a new Web app and copy the Firebase configuration snippet.

Step-3 Paste the configuration into index.html before app.js script.

Step-4. Enable Email/Password authentication in the Firebase Console.

Figure 2: Firebase authentication access

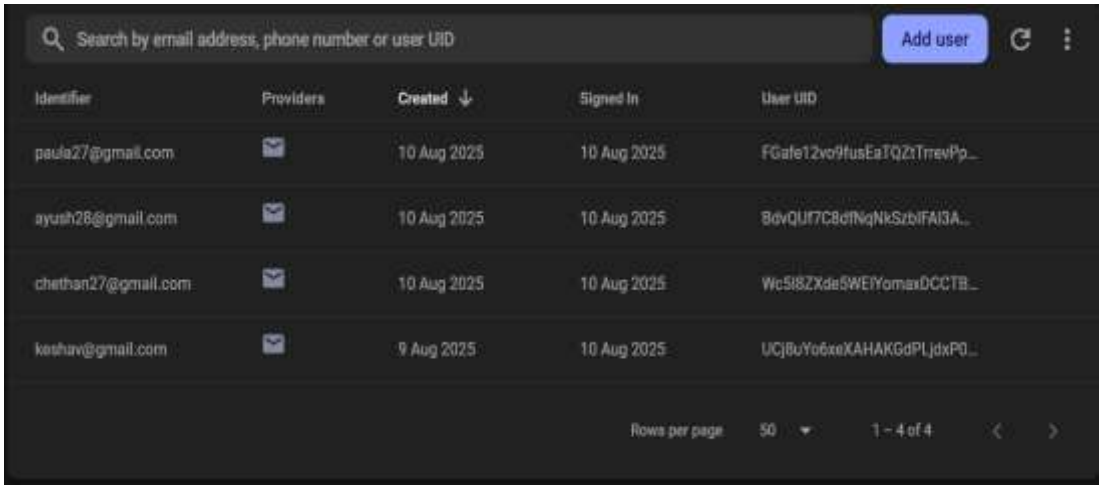


Figure 3: Firebase database access

7. Synthetic Data & Survey Integration

Synthetic dataset (Excel) is converted to JSON or CSV and uploaded it to Google collab for data analysis. Survey results from Google Forms answers are also analyzed from the response section to analyze the requirements of the potential users.

- Synthetic dataset: Smart save (1).xlsx
- Google form survey: https://docs.google.com/forms/d/1ar2qcSPKZqJaH3sIPAL-IdVO1oqNvtu5u2_0I2Z9dHg/edit#responses

8. Synthetic Data analysis with Google collab

Purpose

The synthetic dataset is used to simulate user financial behaviours and evaluate Savemore's savings recommendation logic in a controlled environment before live deployment.

Environment

- **Platform:** Google Colab (Python 3.x)
- **Libraries:** pandas, numpy, matplotlib, seaborn, scikit-learn, scipy
- **Data:** Generated in Excel → exported as CSV/JSON for analysis
- **Reproducibility:** Random seed fixed; library versions recorded in Colab

Analysis:

- Descriptive statistics
- Correlation matrix between income, expenses, and savings variables
- Nudge vs control group outcome comparison
- Occupation-based segmentation

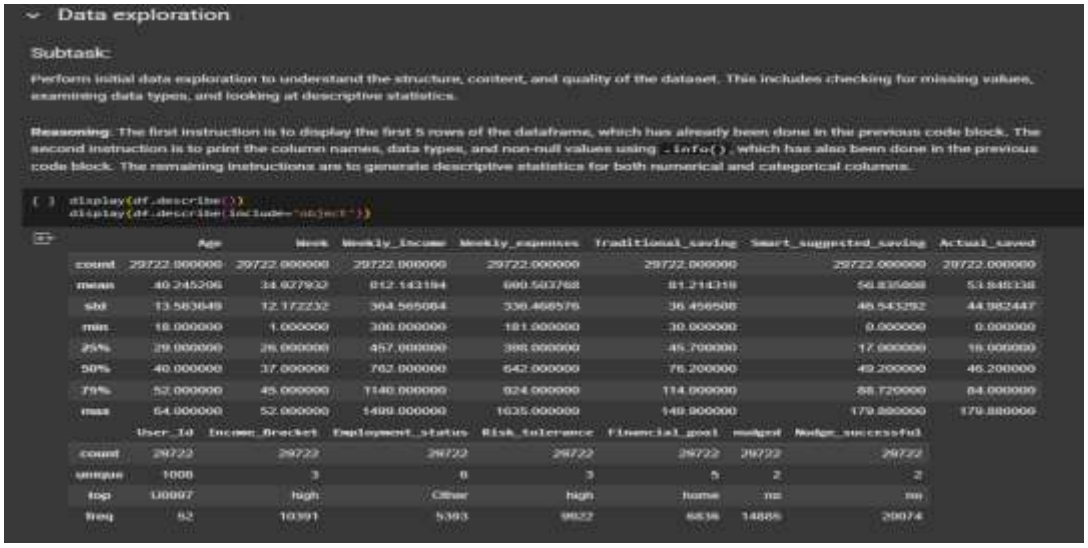
Visualisation Outputs:

- Income/expense distribution plots
- Spending by category charts
- Correlation heatmap
- Nudge impact bar chart

Google collab link for visualisation:
https://colab.research.google.com/drive/107NpfONiMZDeiYog-3ENfZg-MeuVs4ad?usp=sharing

```python
import sys, platform, pandas as pd, numpy as np, matplotlib, seaborn as sns
import sklearn, scipy
print("Python:", sys.version.split()[0])
print("OS:", platform.platform())
print("pandas:", pd.__version__)
print("numpy:", np.__version__)
print("matplotlib:", matplotlib.__version__)
print("seaborn:", sns.__version__)
print("scikit-learn:", sklearn.__version__)
print("scipy:", scipy.__version__)
```

Figure 4: Python libraries



Data exploration

Subtask:

Perform initial data exploration to understand the structure, content, and quality of the dataset. This includes checking for missing values, examining data types, and looking at descriptive statistics.

Reasoning: The first instruction is to display the first 5 rows of the dataframe, which has already been done in the previous code block. The second instruction is to print the column names, data types, and non-null values using `.info()`, which has also been done in the previous code block. The remaining instructions are to generate descriptive statistics for both numerical and categorical columns.

```
display(df.describe())
display(df.describe(include='object'))
```

	Age	Week	Weekly_Income	Weekly_Expenses	Traditional_Saving	Smart_Suggested_Saving	Actual_Saved
count	29722.000000	29722.000000	29722.000000	29722.000000	29722.000000	29722.000000	29722.000000
mean	40.245206	14.027932	812.143194	690.503768	81.214359	56.835808	53.848338
std	13.563648	12.172232	364.565064	330.468576	36.456508	46.543292	44.982447
min	18.000000	1.000000	300.000000	181.000000	30.000000	0.000000	0.000000
25%	29.000000	26.000000	457.000000	388.000000	45.700000	17.000000	16.000000
50%	40.000000	37.000000	762.000000	642.000000	76.200000	49.200000	46.200000
75%	52.000000	45.000000	1140.000000	924.000000	114.000000	88.720000	84.000000
max	64.000000	52.000000	1499.000000	1035.000000	149.900000	179.880000	179.880000

	User_Id	Income_Bracket	Employment_Status	Risk_Tolerance	Financial_goal	nudged	Nudge_successful
count	29722	29722	29722	29722	29722	29722	29722
unique	1000	3	6	3	5	2	2
top	U0097	high	Other	high	home	no	no
freq	62	10391	5383	9822	6636	14885	20074

Figure 5: Data Exploration

Identify trends and patterns

Subtask:

Analyze the data to identify any notable trends or patterns. This could involve visualizing the data or grouping it to look for relationships between columns.

Reasoning: Analyze and visualize the distribution of categorical columns and the 'Age' column as per the instructions.

```
import matplotlib.pyplot as plt
import seaborn as sns

categorical_cols = ['Income_Bracket', 'Employment_Status', 'Risk_Tolerance', 'Financial_goal', 'nudged', 'Nudge_successful']
numerical_cols = ['Age', 'Weekly_Income', 'Weekly_Expenses', 'Traditional_Saving', 'Smart_Suggested_Saving', 'Actual_Saved']

# Plotting distributions for categorical columns
plt.figure(figsize=(15, 10))
for i, col in enumerate(categorical_cols):
    plt.subplot(2, 3, i + 1)
    sns.countplot(data=df, y=col, order=df[col].value_counts().index, palette='viridis')
    plt.title(f'Distribution of {col}')
    plt.xlabel('Count')
    plt.ylabel(col)
plt.tight_layout()
plt.show()

# Plotting distribution for Age
plt.figure(figsize=(8, 5))
sns.histplot(data=df, x='Age', kde=True, bins=20, color='skyblue')
plt.title('Distribution of Age')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```

Figure 6:Analysing trend and patterns

9. API key configuration

- Visit the OpenAI developer platform: (https://platform.openai.com) Use your account information to log in.
- To access the API Keys, click on your profile icon in the upper right corner after logging in, and then select API Keys from the Settings menu.
- Create a New Key: Click. Make a fresh secret key. OpenAI won't display the key again, so give it a name (for example, "Savemore Integration") and copy it right away.
- Keep Your Key Safe: Keep it securely in your backend (for example, in a secure configuration file or Firebase environment variables).
- Use the key in your app: Utilize the key to authenticate queries to OpenAI's API in your backend (such as Python, Node.js, etc.).

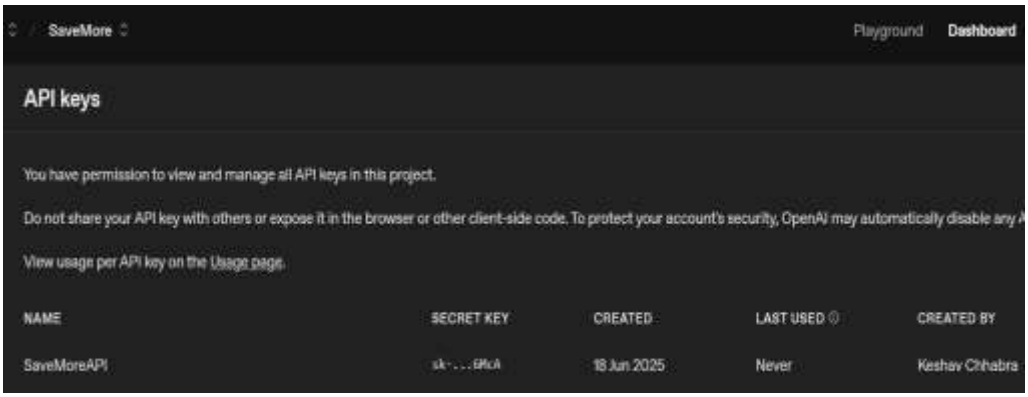


Figure 7: API Key creation

10. Usage Instructions

Web Interface

- Sign up or log in
- Set savings goals
- View spending analysis

- Receive AI nudges and suggestions

11. Troubleshooting

Table 2 Troubleshooting

Issue	Cause	Solution
Firebase login fails	Incorrect config or credentials	Check Firebase keys and rules
API not responding	Invalid or expired key	Regenerate key from OpenAI
Charts not loading	Missing dataset	Upload savemore_data.csv
UI not displaying properly	Browser compatibility	Use Chrome or Firefox
Colab notebook errors	Missing libraries	Run pip install commands again