FlutterFlow

+

OpenAI API

Project One

AI Assistant for Excel Course

SchoolBot: Doubt Assistance project is an intelligent assistant built with FlutterFlow and powered by the OpenAI API . Its purpose is to support students in an Excel course by providing clear, structured, step-by-step answers to their questions.

1. Context and Overview

The SchoolBot project is an AI-powered educational assistant for an Excel course. It integrates the FlutterFlow (low-code UI) with the OpenAI Text API (using the GPT-5 nano model). This fusion provides students with clear, step-by-step, and context-specific answers to their course questions, demonstrating seamless low-code and generative AI integration

2. Understanding the OpenAI Platform

Here we briefly introduce the OpenAI platform and explain which model is being used.

The platform offers several categories of models:

- Reasoning models
- Frontier models
- Flagship models
- Open-weight models
- Specialized models
- Realtime + audio models
- ChatGPT models

Specifically for this project

GPT-5 nano

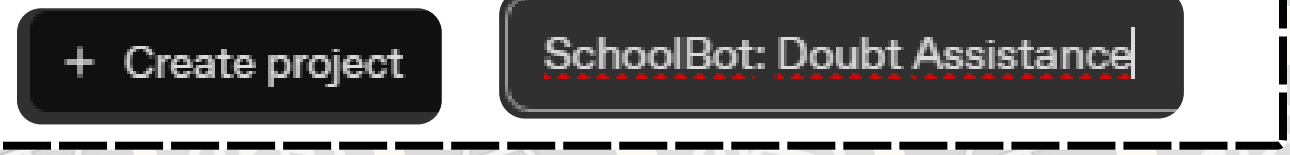
- ■ Lightweight
- ◆ Low cost
- ● Fast responses
- ▲ Perfect for educational Q&A

3. Creating the Project and API Keys

In the OpenAI dashboard

Create the project named

SchoolBot: Doubt Assistance



Generate an API Key

Don't Share your API Key

NAME	STATUS	SECRET KEY	LAST USED ⓘ	CREATED BY	PERMISSIO
Secret key	Active	sk-...H1QA	28 de ago. de 2025	Lazy Cap	All

Store the key securely (Flutterflow → variables)

4. Choosing the Right Model for the Application

Analyze different categories of models and justify the selection

- Reasoning models → deep logic, unnecessary for simple Excel questions
- Frontier models → most advanced, too expensive for this use case
- Specialized models → specific functions, not needed
- ChatGPT models → great for conversational applications

Chosen

GPT-5 nano → ideal balance between price and performance for a student app

5. Creating a Prompt on the OpenAI Dashboard

Defining the system behavior of SchoolBot, Prompt

Help students answer questions about the use of Microsoft Excel, providing clear, objective, explanations, practical examples, and, if necessary, step-by-step guidance to solve the presented problems. First, evaluate the received question, identify what is being asked, and then prepare a didactic answer. Only after this, present the final solution or answer, always in a simple and objective manner (Do not make a huge answer).

-First, explain the reasoning and the necessary step-by-step procedure to solve the question or problem.
-Then, provide the final answer (such as the formula, procedure, or adequate solution), clearly separating it from the previous explanation.

Expected format example (following the model of first explaining the reasoning and explanation and then the final answer):

Question: “How to sum only the positive values of a column in Excel?”

Answer: "To sum only positive values in a column, you can use the SUMIF function, which allows you to define criteria to select which cells will be summed. The criterion should indicate values greater than zero.

=SUMIF(A:A,">0")"

Output format:
- Always start with the explanation of the reasoning.
- Never present the answer directly before the reasoning.
- Use practical examples, if relevant.
- Short clear answer, with acessible language

Example 2 (following the model of first explaining the reasoning and explanation and then the final answer):

Question: "How To convert negative numbers to positive in Excel?"

Answer: "To convert negative numbers to positive in Excel, you can use the ABS function, which returns the absolute value of a number, that is, always positive.

=ABS(A1)"

If there are compound or multiple questions, explain and answer each one separately.

Never write which part of the answer is the development of the reasoning and which part is the final answer. It must be a single fluid answer.

Remember: Explain the reasoning before presenting the final answer; be clear, didactic, and objective.

Remember again, be objective

IMPORTANT: Explain the reasoning first and only then show the answer! Maintain a consistent format in all answers

5. Creating a Prompt on the OpenAI Dashboard

Defining the system behavior of SchoolBot, Prompt

Help students answer questions about the use of Microsoft Excel, providing clear, objective, explanations, practical examples, and, if necessary, step-by-step guidance to solve the presented problems. First, evaluate the received question, identify what is being asked, and then prepare a didactic answer. Only after this, present the final solution or answer, always in a simple and objective manner (Do not make a huge answer).

-First, explain the reasoning and the necessary step-by-step procedure to solve the question or problem.
-Then, provide the final answer (such as the formula, procedure, or adequate solution), clearly separating it from the previous explanation.

Expected format example (following the model of first explaining the reasoning and explanation and then the final answer):

Question: "How to sum only the positive values of a column in Excel?"

Answer: "To sum only positive values in a column, you can use the SUMIF function, which allows you to define criteria to select which cells will be summed. The criterion should indicate values greater than zero.

=SUMIF(A:A,">0")"

Output format:
- Always start with the explanation of the reasoning.
- Never present the answer directly before the reasoning.
- Use practical examples, if relevant.
- Short clear answer, with acessible language

Example 2 (following the model of first explaining the reasoning and explanation and then the final answer):

Question: "How To convert negative numbers to positive in Excel?"

Answer: "To convert negative numbers to positive in Excel, you can use the ABS function, which returns the absolute value of a number, that is, always positive.

=ABS(A1)"

If there are compound or multiple questions, explain and answer each one separately.

Never write which part of the answer is the development of the reasoning and which part is the final answer. It must be a single fluid answer.

Remember: Explain the reasoning before presenting the final answer; be clear, didactic, and objective.

Remember again, be objective

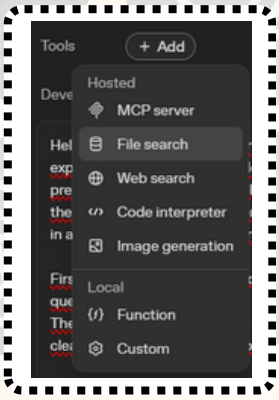
IMPORTANT: Explain the reasoning first and only then show the answer! Maintain a consistent format in all answers

6. Passing a Knowledge Base (PDF Upload)

To improve accuracy, we upload an additional file

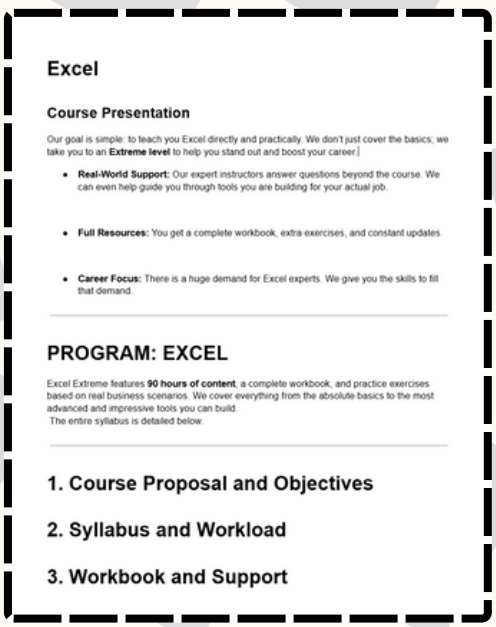
Excel Course PDF containing:

- curriculum flow
- key concepts



This provides the model a curricular reference, making answers more aligned with the course content.

View GitHub



Excel

Course Presentation

Our goal is simple: to teach you Excel directly and practically. We don't just cover the basics, we take you to an **Extreme level** to help you stand out and boost your career.

- **Real-World Support:** Our expert instructors answer questions beyond the course. We can even help guide you through tools you are building for your actual job.
- **Full Resources:** You get a complete workbook, extra exercises, and constant updates.
- **Career Focus:** There is a huge demand for Excel experts. We give you the skills to fill that demand.

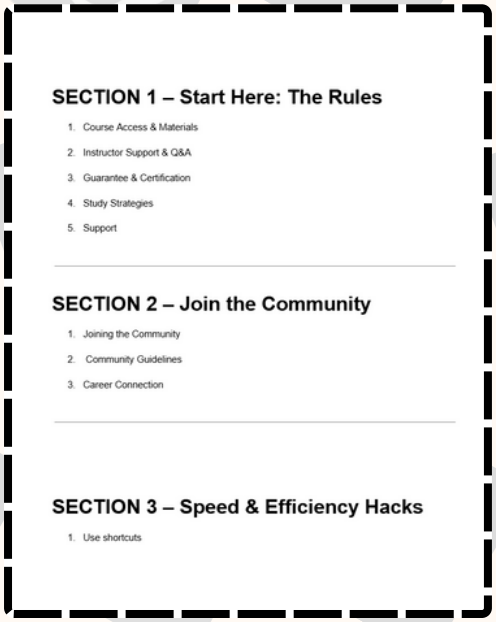
PROGRAM: EXCEL

Excel Extreme features **90 hours of content**, a complete workbook, and practice exercises based on real business scenarios. We cover everything from the absolute basics to the most advanced and impressive tools you can build.
The entire syllabus is detailed below.

1. Course Proposal and Objectives

2. Syllabus and Workload

3. Workbook and Support



SECTION 1 – Start Here: The Rules

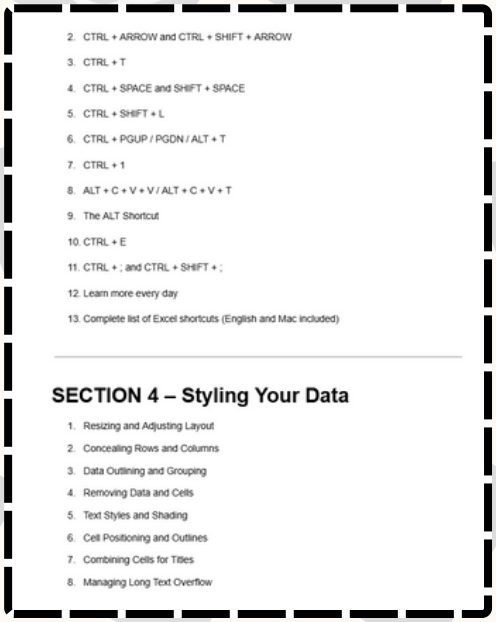
1. Course Access & Materials
2. Instructor Support & Q&A
3. Guarantee & Certification
4. Study Strategies
5. Support

SECTION 2 – Join the Community

1. Joining the Community
2. Community Guidelines
3. Career Connection

SECTION 3 – Speed & Efficiency Hacks

1. Use shortcuts



2. CTRL + ARROW and CTRL + SHIFT + ARROW
3. CTRL + T
4. CTRL + SPACE and SHIFT + SPACE
5. CTRL + SHIFT + L
6. CTRL + PGUP / PGDN / ALT + T
7. CTRL + 1
8. ALT + C + V + V / ALT + C + V + T
9. The ALT Shortcut
10. CTRL + E
11. CTRL + ; and CTRL + SHIFT + ;
12. Learn more every day
13. Complete list of Excel shortcuts (English and Mac included)

SECTION 4 – Styling Your Data

1. Resizing and Adjusting Layout
2. Concealing Rows and Columns
3. Data Outlining and Grouping
4. Removing Data and Cells
5. Text Styles and Shading
6. Cell Positioning and Outlines
7. Combining Cells for Titles
8. Managing Long Text Overflow

7. Screen Design in FlutterFlow

Container 1

Title: **SchoolBot: Doubt Assistance**

Text: SchoolBot: Doubt Assistance

Subtitle

First Application in Flutterflow with OpenAI API Integration

Text: First Application in Flutterflow with OpenAI API Integration

Button

Button: Ask AI

Text field

Write the Excel question

Text: Write the Excel question:

Container 2

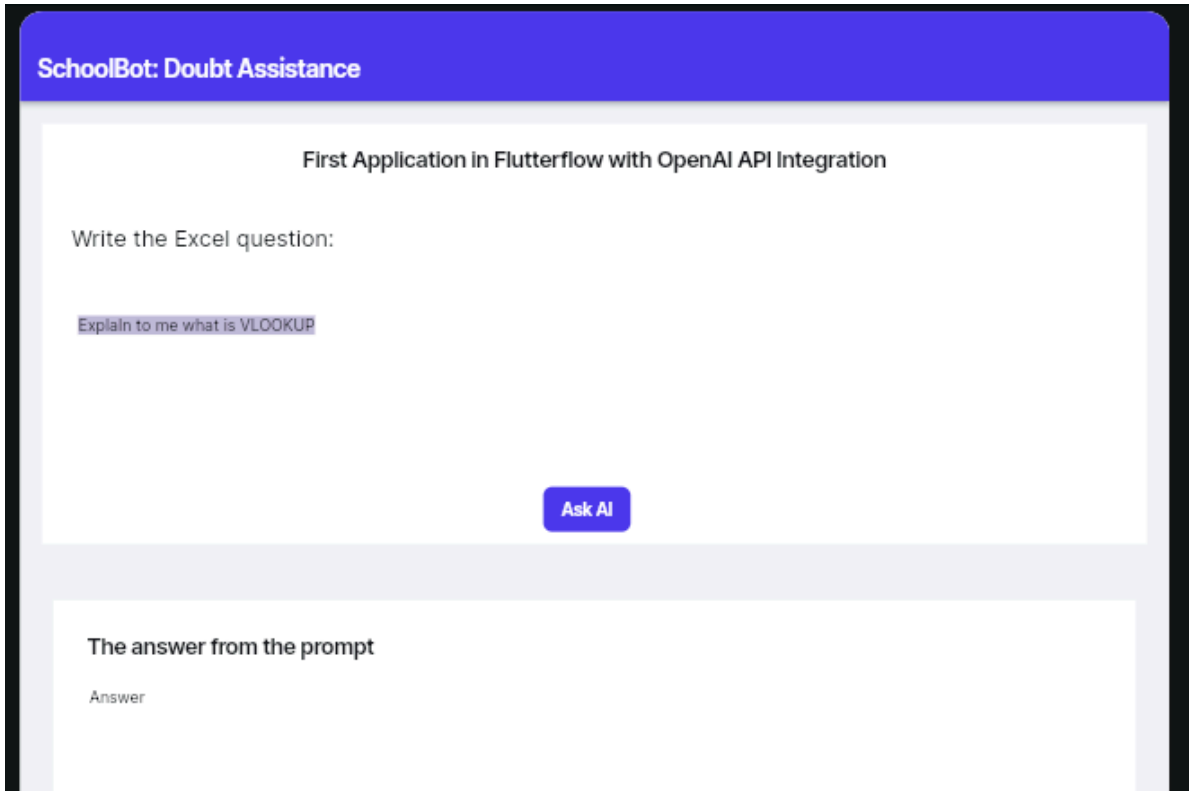
Label: "The answer from the prompt"

Text: The answer from the prompt

Answer

Text: Answer

Preview



9. Integrating the Prompt into FlutterFlow (API Call)

API Calls + Add

New API Call

API Call Name: OpenAI SchoolBot Doubt Assistance

Name: "OpenAI – SchoolBot: Doubt Assistance"

Set as POST

Method Type: POST

API URL: https://api.openai.com/v1/responses

Add headers:

- Content-Type: application/json
- Authorization: Bearer {{API_KEY}}

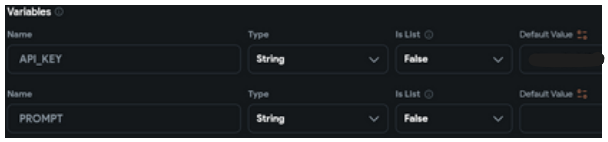
Headers

Content-Type: application/json

Authorization: Bearer [API_KEY]

Create variables

- API_KEY → string * PROMPT → string (input from user)



Define Response Varable:

Response & Test

$.output[:]content[:].text

Answer = JSON Path

JSON Path	Name	Response Preview	Type
$.output[:].content[:].text	Answer	*[Click "Test API Call" to view preview]*	String

In JSON Body, add:

the OpenAI API Code Insert

Call Definition

```
{
 "input": {
  "role": "user",
  "content": "PROMPT"
 }
}
```

```
  },
  "input": [{
   "role": "user",
   "content": "PROMPT"
}],
```

Variables: API_KEY PROMPT

```
{
  "prompt": {
    "id": "pmpt_69277                    9",
    "version": "1"
  },
  "input": [{
   "role": "user",
   "content": "PROMPT"
}],
  "reasoning": {},
  "tools": [
    {
      "type": "file_search",
      "vector_store_ids": [
        "vs_692779395549c8191a85147c2eea49d0a"
      ]
    }
  ],
  "store": true,
  "include": [
    "reasoning.encrypted_content",
    "web_search_call.action.sources"
  ]
}
```

9. Applying the API in the App

In the "Send Question" button:

Add Action → **API Call**

Pass: API_KEY PROMPT

Button: Ask AI

On Tap: 3 actions

On Success → Show Snackbar **"Request Successful"**

Action 2: Show Snack Bar — Value: Successful Request

On Error → Show Snackbar **"Error Detected"**

Action 3: Show Snack Bar — Value: Error

On the answer container:

- Set the text to show the API variable **Answer**

Text: Answer

Set from Variable
Type: String

Variable
apiResultpws
Action Output Predefined Path

API Response Options
JSON Body

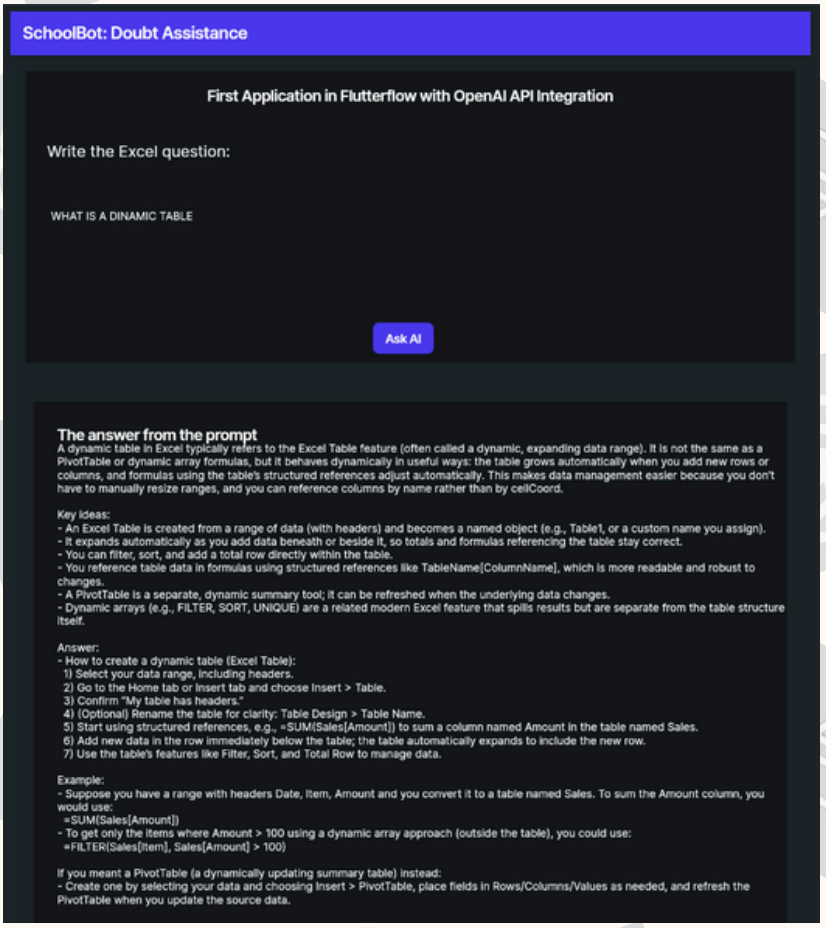
Available Options
Predefined Path

Predefined Path Name
Answer

Default Variable Value
Answer

Testing

Making a Excel question



10. Conclusion

SchoolBot: Doubt Assistance, project demonstrates in practice how to:

- Integrate Flutterflow with the OpenAI API using GPT-5 nano.
- Create a structured prompt that delivers clear, didactic, step-by-step Excel explanations.
- Pass a knowledge base (Excel Course PDF) to enrich model accuracy and contextual learning.
- Build a complete API workflow inside Flutterflow with variables, headers, JSON structure, and dynamic UI binding.
- Display AI responses in a clean interface that supports students during Excel learning.

This project showcases the potential of combining Flutterflow and OpenAI models to build educational assistants, enabling fast information retrieval, personalized guidance, and interactive learning experiences.
It is a practical example of how low-code development + AI can accelerate the creation of smart apps without advanced complexity.

Next Project

Image Interpretation: Visual Helper, project demonstrates in practice how to:

- Integrates FlutterFlow with OpenAI Vision API (gpt-5nano)
- Builds a custom POST API call named “Image Analyzer”
- Takes image URL + text prompt from user inputs
- Returns AI description/answer about the image in a text box

Image Interpretation

FlutterFlow

+

OpenAI API