Smart Recruiter Documentation

Version 1.0

BintyByte

Table of Contents

1. Introduction 3
1.1 Overview 3
1.2 Key Features 3
2. Introduction 4
2.1 Overview 4
3. System Requirements 5
3.1 Hardware Requirements 5
3.2 Software Requirements 5
4. Setting Up Gemini API Key and Model Name 6
4.1 1. Create a Gemini API Key 6
4.1.1 Step 1: Sign in to Google AI Studio and Create an API Key 6
5. Installation & Deployment 10
5.1 1. Download the Binary File and Folder Structure 10
5.2 2. Copy to Deployment Directory 10
5.3 3. Configure Dependencies 11
5.3.1 .env File 11
5.4 4. Verify Configuration 11
6. Configuring Input & Output Folders 12
7. Run the Application 13

1. Introduction

Welcome to the **Smart Recruiter** user guide!!! Smart Recruiter is an AI-powered resume screening tool designed to automate the recruitment process. This guide provides step-by-step instructions on how to deploy and use the software on your laptop, for HR, Hiring Manager or recruitment agencies

1.1 Overview

Smart Recruiter is a Python-based application designed to streamline the recruitment process. It allows you recruitment workflows efficiently.

1.2 Key Features

- **AI-Powered Resume Screening**: Automates the recruitment process by using AI to analyze and match resumes with job descriptions (JDs). Reduces manual effort and improves efficiency in the hiring process.
- **Automated Resume Scoring**: Assigns a score to each resume based on its relevance to the job description.
- **Workflow Automation**: Automate recruitment workflows.

Next: System Requirements →

2. Introduction

Welcome to the **Smart Recruiter** user guide!!! Smart Recruiter is an AI-powered resume screening tool designed to automate the recruitment process. This guide provides step-by-step instructions on how to deploy and use the software on your laptop, for HR, Hiring Manager or recruitment agencies

2.1 Overview

Smart Recruiter is a Python-based application designed to streamline the recruitment process. It allows you recruitment workflows efficiently.

3. System Requirements

Before installing and using **Smart Recruiter**, ensure your system meets the following requirements:

3.1 Hardware Requirements

- **Processor**: Intel Core i5 or equivalent.
- **RAM**: 8 GB or higher.
- **Storage**: 10 GB of free disk space.

3.2 Software Requirements

- **Operating System**: Windows 10, macOS 10.14+, or Linux (Ubuntu 20.04+).
- **Dependencies**: See the Configure Dependencies section.

Next: Installation & Deployment →

4. Setting Up Gemini API Key and Model Name

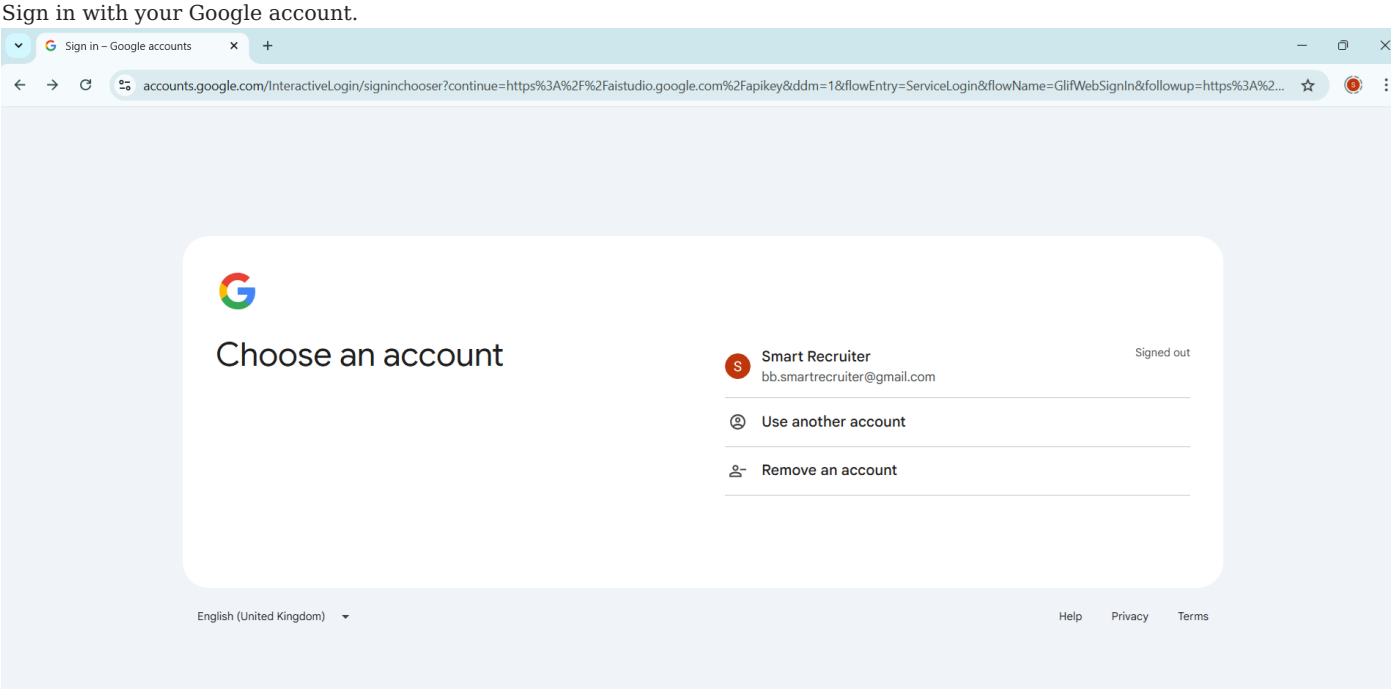
To use **AI Smart Resume Match**, you need to configure the **Gemini API Key** and **Model Name**. Follow the steps below to set up these dependencies.

4.1 1. Create a Gemini API Key

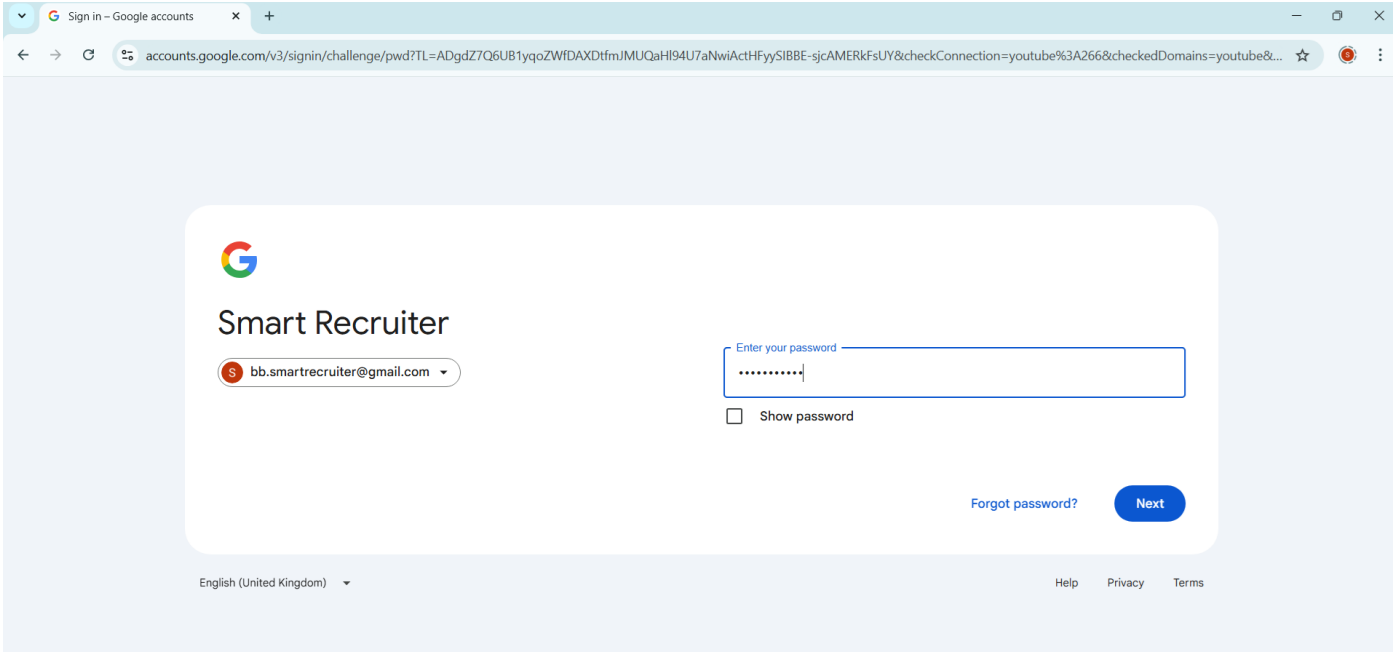
The **Gemini API Key** is required for authentication with the Gemini AI service. Here's how to create one:

4.1.1 Step 1: Sign in to Google AI Studio and Create an API Key

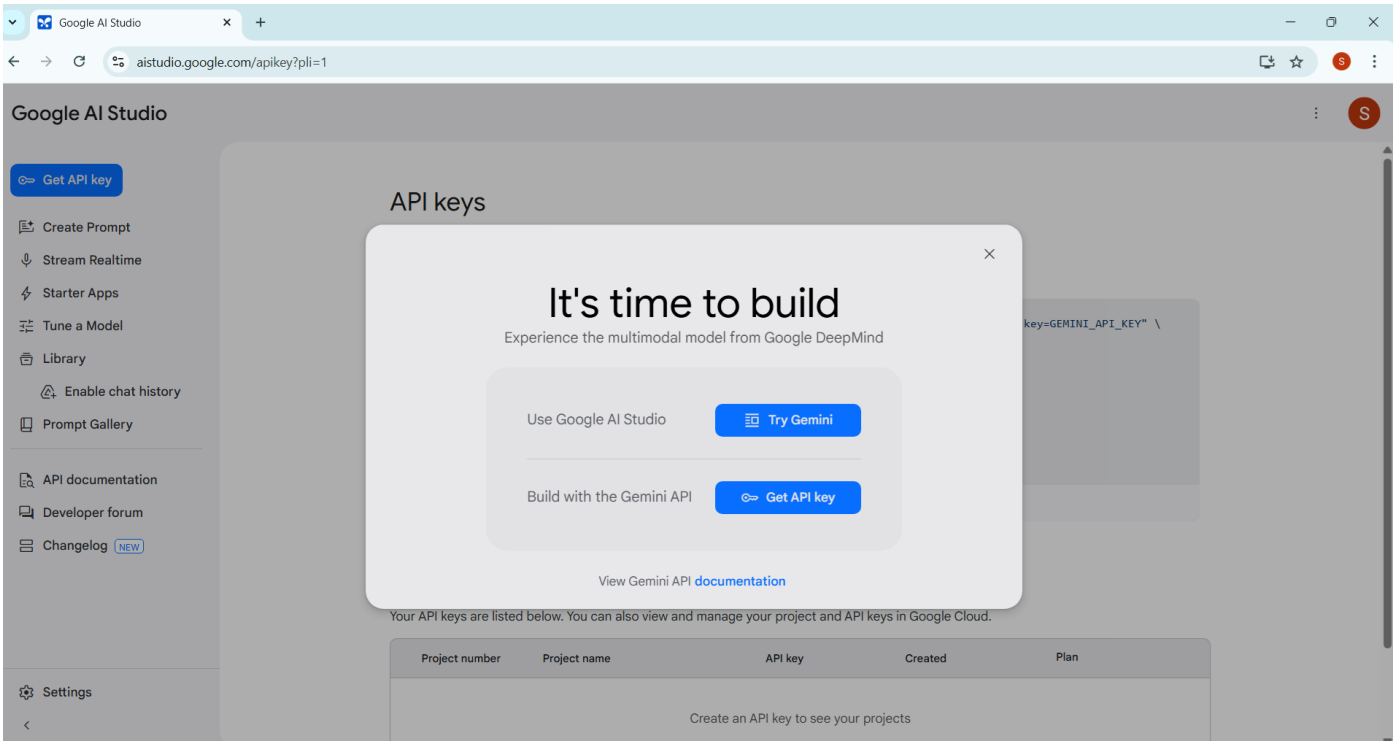
1. Go to Google AI Studio.
2. Sign in with your Google account.



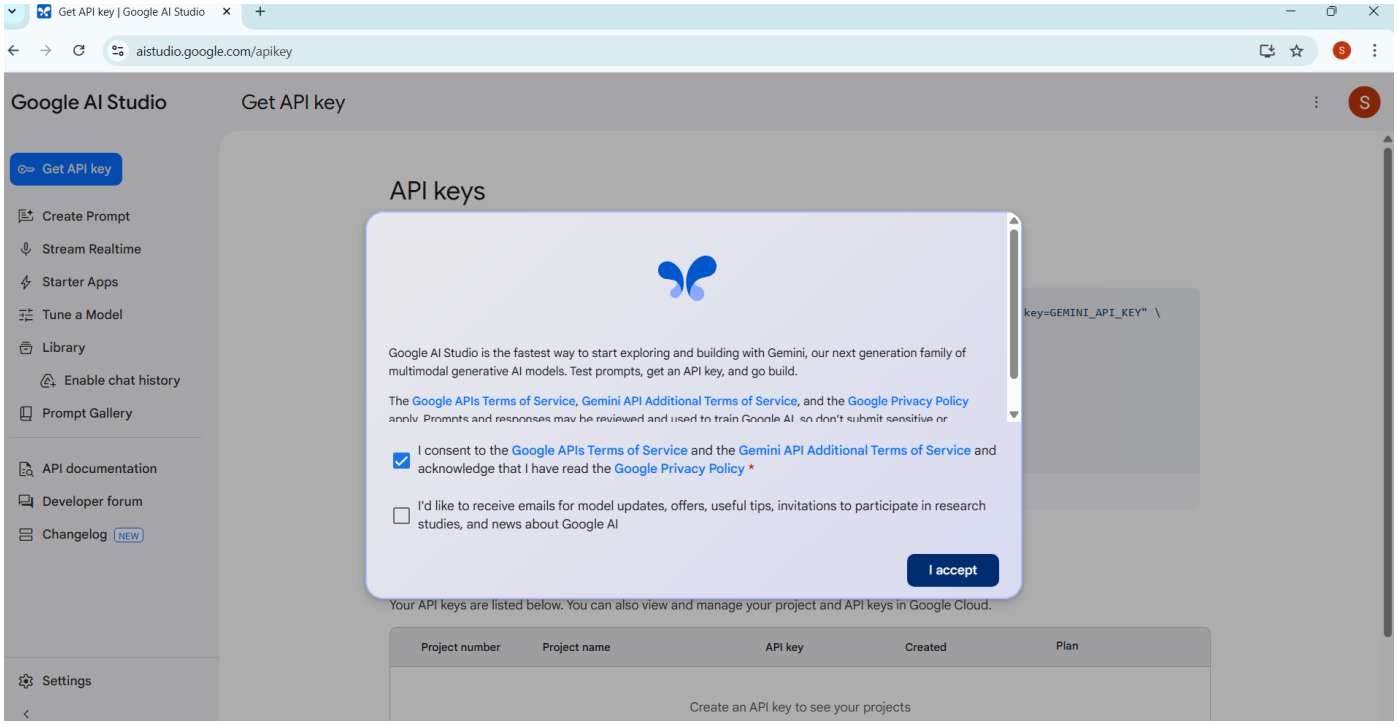
3. Enter your Google account password.



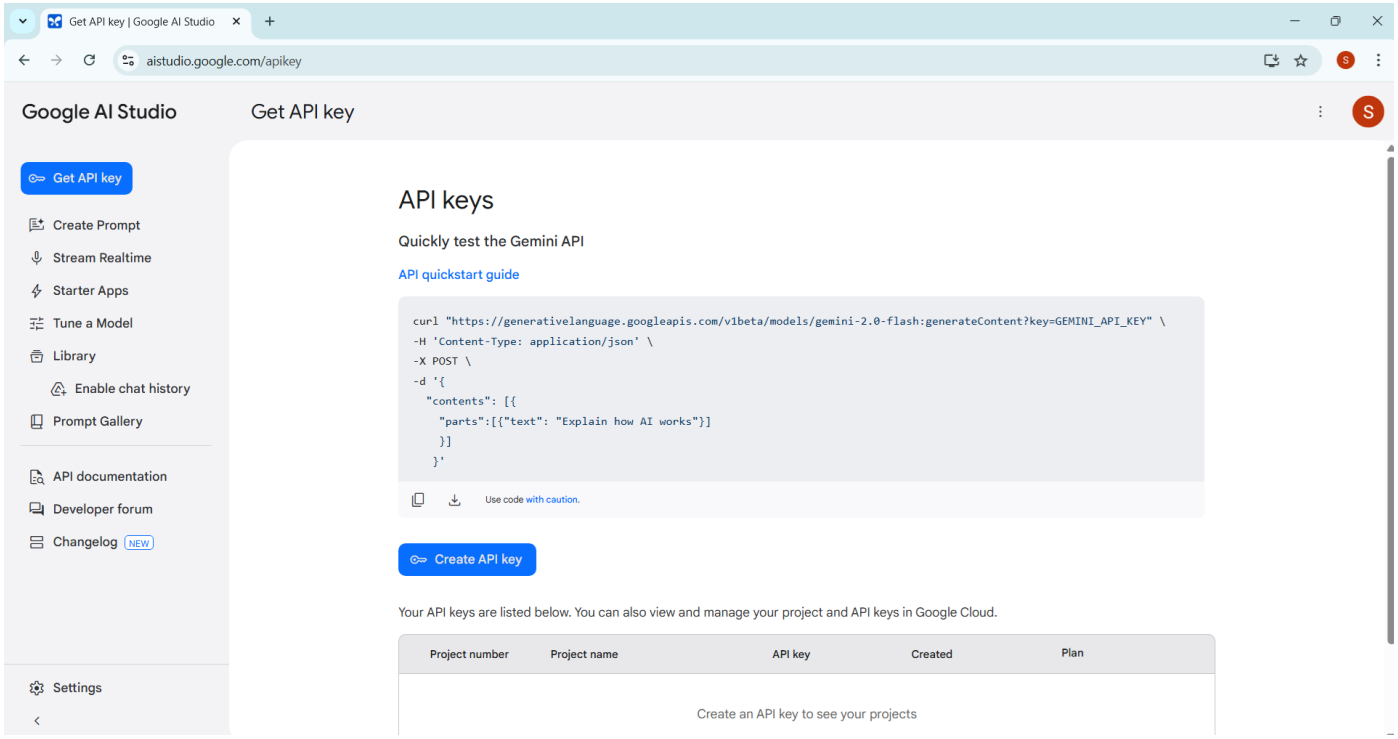
4. After Successful Google account Sign in. You will get below screen then click on **Get API Key** button.



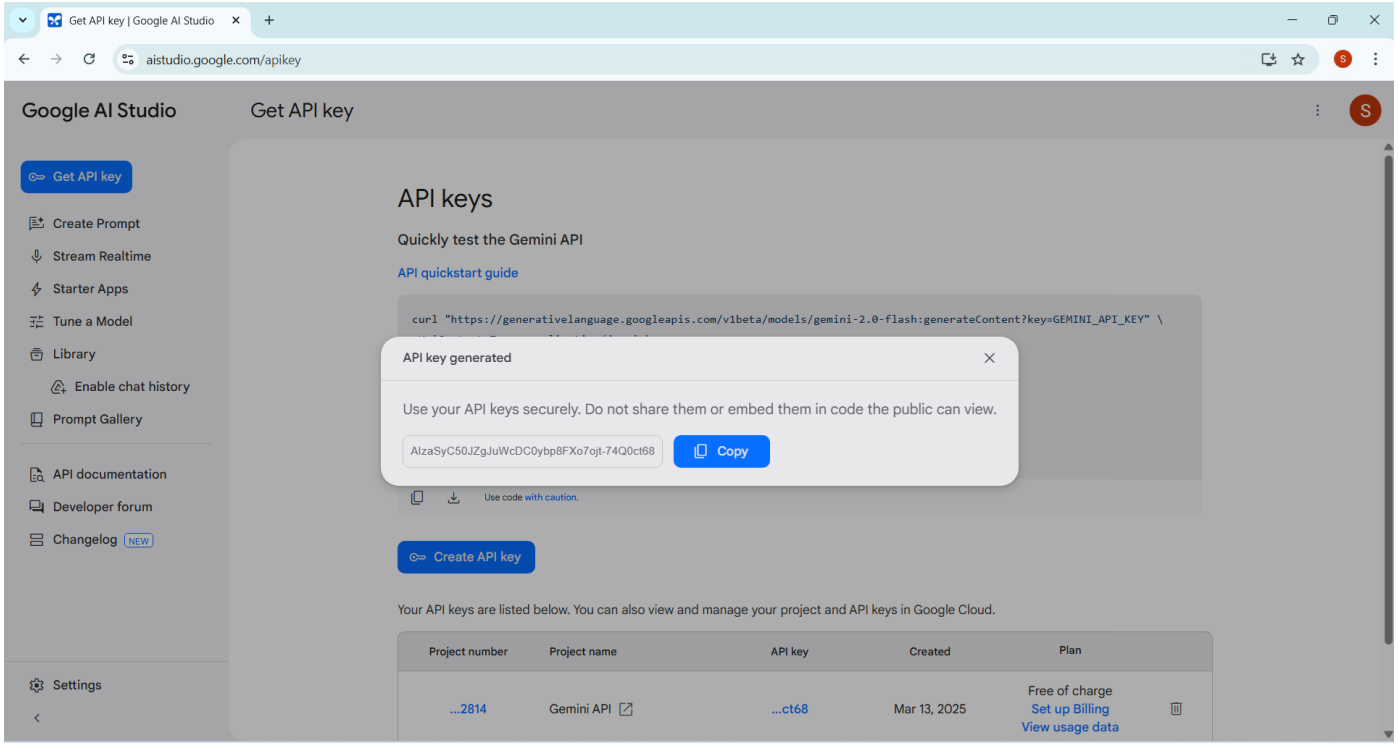
5. After you get below screen then select **I consent to the Google APIs Terms of Service ...** and then click **I accept** button.



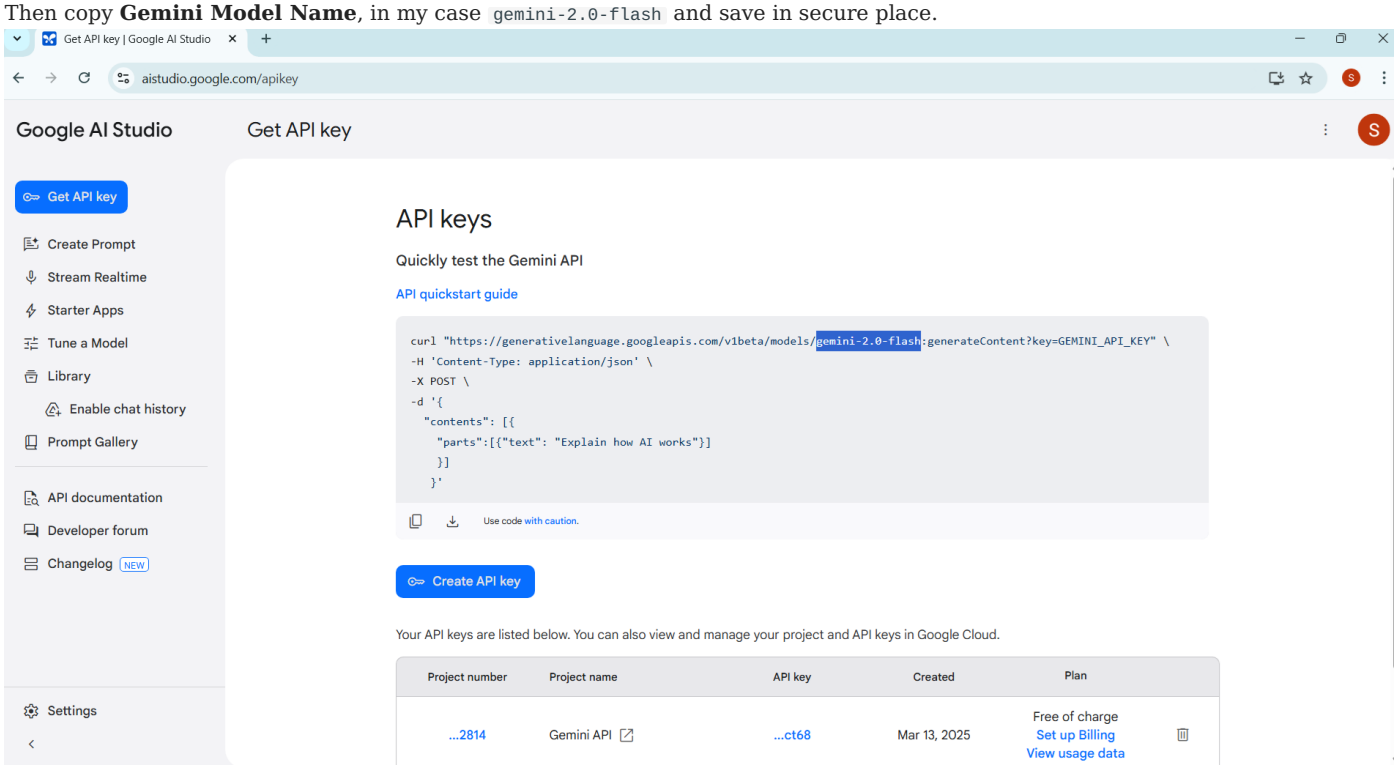
6. Then you will get **Google AI Studio Get API Key** screen then click **Create API key** button.



7. Then you will get **API key generated** screen then copy **API key** and save in secure place.



8. Then copy **Gemini Model Name**, in my case `gemini-2.0-flash` and save in secure place.



5. Installation & Deployment

Follow these steps to install and deploy **Smart Recruiter** on your system.

5.1 1. Download the Binary File and Folder Structure

1. Download the **Folder Structure** and **Smart_Recruiting** executable file from a Git repository or a zipped archive.
2. Extract the contents to a directory of your choice.

5.2 2. Copy to Deployment Directory

1. Place the downloaded files in a preferred working directory.
2. Ensure the folder structure is intact after extraction.

5.3 3. Configure Dependencies

5.3.1 .env File

The `.env` file contains environment variables that configure the application. Follow these steps to set it up:

1. Open the `.env` file in a text editor.
2. Modify the following variables according to your setup:

Variable	Description
`GEMINI_API_KEY`	Enter your Google Gemini API key.
`GEMINI_MODEL_NAME`	The model is set to `gemini-1.5-flash`. You probably don't have to change this.
`BASE_JD_DIR`	Specify the directory where your Job Description (JD) files are located.
`SUMMARY_OUTPUT_FILE`	Specify the file path where the results will be saved.
`JSON_INPUT_DIR`	This is a temporary directory for JSON input files.
`TEMPLATE_DIR`	Specify the directory where the resume templates are located.
`LOGO_PATH`	Specify the path where the company logo is located.
`INPUT__LLM_PROMPT`	Specify the prompt file name (without the extension) for the Gemini model. For example, `gemini_1` or `gemini_2`. Ensure the prompt files are located in the correct directory.
`OUTPUT__RESUME_FORMAT`	Specify the desired resume output formats (comma-separated): `docx,html,md,pdf`.
`USERNAME`	Enter your username for license validation.
`EMAIL`	Enter your email for license validation.
`SECRET_KEY`	**Do not modify manually.** This key is automatically generated during the license generation step.
`LICENSE_KEY`	**Do not modify manually.** This key is automatically generated during the license generation step.
`TESSERACT_MODE`	Choose `ON` if you want to parse only `.pdf` formats in your JD directories with the help of `TESSERACT_CMD`. Choose `OFF` if you want to parse both `.pdf` and `.docx` formats and use the file data directly without `TESSERACT_CMD`.

5.4 4. Verify Configuration

After editing the `.env` file, verify that all variables are correctly configured. Ensure the paths and keys are accurate.

6. Configuring Input & Output Folders

AI Smart Resume Match requires predefined folders for input and output within the `DATA` folder.

Input Folders:

- **Job Descriptions (JDs) Upload Path:**

  `/DATA/INPUT/JDS/JD-00001`

  (Replace `JD-00001` with the actual Job ID). Each job description should reside in its own uniquely named folder.

- **Resumes Upload Path:**

  `/DATA/INPUT/JDS/JD-00001/Resumes`

  Upload resumes to the respective `Resumes` subfolder within the corresponding JD folder.

Output Folder:

- **Results & Formatted Resumes:**

  `/DATA/OUTPUT`

7. Run the Application

To start the application: ```bash python src/main.py