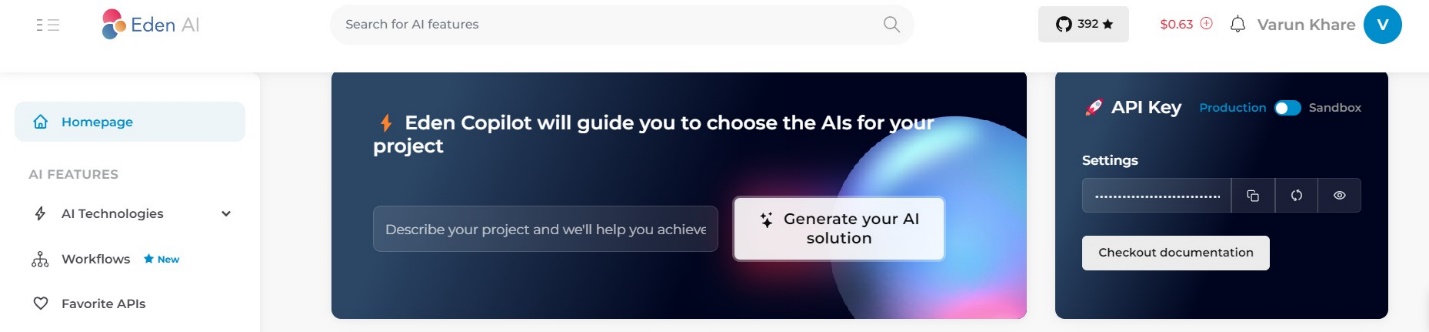
***1. Setup Instructions for Eden AI API Key,*** [***Eden AI***](https://www.edenai.co/) ***:***

* Log in using google account.
* Eden AI provides free 1 US $ to run it’s LLM API
* Go to “API Settings”
* Click on “Create a new API Key”



* Go to home page
* API keys are clearly visible on right side, copy them



***2. User guide to run streamlit app :***

**Prerequisites**

Before running the Streamlit app, make sure you have the following installed on your system:

* **Python**: Ensure you have Python 3.7 or later installed. You can download it from [python.org](https://www.python.org/downloads/).
* **pip**: This is usually included with Python, but if not, you can install it from [here](https://pip.pypa.io/en/stable/installation/)
* Install the following libraries using pip  
    
  ***pip install PyPDF2 scikit-learn requests streamlit***
* Clone the source code
* Run the command

***python -m streamlit run source\_code\_file\_name.py***

**Using the App**

1. Enter Eden AI API Key:
   * When the app loads, you will see a text input field to enter the Eden AI API key. Enter your API key here. If you don't have an API key, you can sign up for one at [Eden AI](https://www.edenai.co/).
2. Upload PDF Documents:
   * Use the "Upload PDF Documents" button to select and upload one or more PDF files. The uploaded files will be processed and displayed in the app.
3. Ask Questions:
   * Enter your question in the "Ask a question about the PDF content:" text input field and click the "Send" button. The app will process your question, find relevant text in the PDF, and query the Eden AI API to generate a response.
4. View Responses and Highlights:
   * The app will display the response from the Eden AI API along with the highlighted relevant parts of the PDF text. Scroll through the chat history to view previous questions and responses.
5. Interact with Multiple PDFs:
   * If multiple PDFs are uploaded, the app will consider all of them when processing your query and finding relevant text. You can ask questions related to any of the uploaded PDFs.

***3. Explanation of approach and design decisions:***

**Overview**

The project aims to develop a web app that lets users interact with PDF documents via a chat interface. Key tasks include PDF text extraction, text preprocessing, query processing, AI API integration, and creating an interactive interface using Streamlit.

**Key Components**

1. **PDF Text Extraction**:
   * **Library**: Used PyPDF2 for reliable text extraction.
   * **Function**: extract\_text\_from\_pdf reads and extracts text from uploaded PDFs.
2. **Text Preprocessing**:
   * **Purpose**: Ensures clean text for processing.
   * **Implementation**: preprocess\_text removes extra whitespace and handles line breaks.
3. **Text Chunking**:
   * **Rationale**: Splits text into manageable chunks for efficient query matching.
   * **Function**: split\_text\_into\_chunks divides text into 500-word segments.
4. **Query Processing**:
   * **Method**: Uses TF-IDF vectorization and cosine similarity.
   * **Function**: find\_relevant\_text matches user queries with the most relevant text chunk.
5. **Eden AI API Integration**:
   * **API Choice**: Selected for its simplicity and effectiveness.
   * **Function**: query\_edenai\_api sends queries to the API and retrieves responses.
6. **Streamlit Interface**:
   * **Framework**: Chosen for ease of creating interactive web apps.
   * **Features**: File uploader, chat interface, and PDF display.
7. **Multiple PDFs Support**:
   * **Functionality**: Processes multiple PDF uploads in a single session to find relevant text for each query.

**Design Considerations**

* **Modularity**: Ensures ease of testing, maintenance, and future enhancements.
* **User Experience**: Provides a clean, interactive interface.
* **Error Handling**: Informs users of issues and ensures graceful recovery.
* **Scalability**: Allows for future enhancements like support for more document formats and advanced text processing.