You

I have the following python code but I want to amke it so that I can share the code with my friends andthey can still use the Gemini instance:

```
#Importing the modules
import vertexai
from vertexai.generative_models import GenerativeModel
import google.auth

credentials, project_id = google.auth.default()

vertexai.init(project="generalpurposeai", location="us-central1")

model = GenerativeModel(model_name="gemini-1.0-pro")

while True:
  prompt = input("Enter a prompt for the Ai to process: ")
  if prompt != "":
    responses = model.generate_content(prompt, stream=True)
    for response in responses:
    print(response.text)
  else:
    exit()
```

Copilot

Certainly! It looks like you've got some Python code there for using the Gemini instance. Sharing code with friends is a great idea, especially when you're working on collaborative projects or just exploring cool AI models together. Let's make sure your friends can use it too!

First off, I'll provide a brief explanation of what your code does:

1. **Imports and Setup**:
    - You're importing necessary modules: vertexai, GenerativeModel, and google.auth.
    - You're initializing your Vertex AI project with a specific project ID and location (in this case, "generalpurposeai" and "us-central1").
    - You're creating an instance of the GenerativeModel class, specifically using the "gemini-1.0-pro" model.
2. **User Interaction Loop**:
    - Your code enters an infinite loop (while True) where it prompts the user to input a prompt for the AI to process.
    - If the input prompt is not empty, it generates content using the Gemini model and prints the response.
    - If the input prompt is empty, the loop exits.

Now, let's make your code more shareable! To ensure your friends can use it without any issues, follow these steps:

1. **Package Dependencies**:
    - Make sure your friends have the necessary Python packages installed. They'll need vertexai and any other dependencies used in your code.
    - You can create a requirements.txt file listing the required packages, which your friends can then use to install everything easily:
    - vertexai==<version>
    - # Add other dependencies here
2. **Authentication and Project Setup**:
    - Ensure that your friends have the appropriate Google Cloud credentials set up. They might need to authenticate using their own credentials or set up a service account.
    - Make sure they replace "generalpurposeai" with their own project ID if they're using a different project.
3. **Sharing the Code**:
    - Share the Python script with your friends. You can create a separate file (e.g., gemini_ai.py) and copy your existing code into it.
    - Encourage them to run the script and interact with it by entering prompts.

Remember to provide clear instructions alongside the code so your friends know how to set it up and use it effectively. Happy coding, and enjoy exploring the Gemini instance together! If you have any more questions or need further assistance, feel free to ask. 😊🚀

By the way, have you tried any other interesting AI models or projects lately? I'm always curious to hear about cool tech experiments!

You

THe thing is my frineds don't have google cloud workspace or access to gemini themselves but obviously it's not good pactice to sahre my secret credentials with them. What else can I do instead?

Copilot

Ah, I see! You're absolutely right—it's definitely not a good idea to share your secret credentials with friends. Security first! Let's find a more elegant solution that doesn't involve revealing your precious access keys.

Here are a couple of alternative approaches you can consider:

1. **Host a Web Service or API Wrapper**:
    - Instead of sharing your credentials directly, you could create a simple web service or API wrapper around your Gemini model.
    - Set up a server (could be on a cloud provider or even your local machine) that exposes an API endpoint. When your friends want to generate content, they can send a request to this endpoint with their prompts.
    - Your server, which has the credentials, can then make the Gemini API call and return the response to your friends.

    - This way, your friends won't need direct access to Google Cloud or the Gemini instance—they'll just interact with your API.

2. **Deploy the Model on a Platform**:
    - Explore platforms that allow you to deploy machine learning models without exposing your credentials.
    - For example, you could use platforms like Heroku, AWS Lambda, or Google Cloud Run. These services allow you to deploy code (including models) as serverless functions.
    - You'd set up your code to run when an HTTP request is made to a specific endpoint. Again, your friends would send prompts via this endpoint, and your deployed function would handle the Gemini API calls.

3. **Create a Docker Container**:
    - Docker containers encapsulate your code and dependencies, making it easy to share.
    - You could create a Docker image containing your Python script, the necessary dependencies, and your credentials (loaded from environment variables).
    - Your friends would then run this container locally, and it would handle the interactions with Gemini.

Remember, whichever approach you choose, document it well for your friends. Provide clear instructions on how to use your solution, including any necessary setup steps. And hey, you're being a responsible AI enthusiast by not sharing those secret keys—kudos! 😄

Have you ever dabbled in deploying models or creating APIs before? It's a fascinating world out there! Let me know if you'd like more details on any of the options above or if you have other questions—I'm here to help! 🌟🚀

You

How could I deploy it on google run?

Copilot

Great choice! Deploying your Python app on Google Cloud Run is a smart move—it's a serverless platform that automatically scales your containerized applications. Let's get you started:

1. **Prerequisites**:
    - Make sure you have a Google Cloud account. If you're new to Google Cloud, you can create an account and even get $300 in free credits to explore.
    - Install the Google Cloud CLI if you haven't already.

2. **Initialize the gcloud CLI**: Run the following command to set up your default project (replace PROJECT_ID with your actual Google Cloud project ID):

```
gcloud init
```

Enable APIs: Enable the necessary APIs by running these commands:

```
gcloud services enable run.googleapis.com cloudbuild.googleapis.com
```

Build and Deploy:

- First, build a Docker container for your app and publish it to Container Registry:

```
gcloud builds submit --tag gcr.io/PROJECT_ID/bookshelf
```

□ Replace PROJECT_ID with your actual project ID.

□ Next, deploy your container to Cloud Run:

```
gcloud run deploy bookshelf --image gcr.io/PROJECT_ID/bookshelf
```

4.
   - Again, replace PROJECT_ID.
5. **Access Your Deployed Service**: Once deployed, you'll receive a URL. Copy it into your web browser to view your app. Note that this URL is SSL-protected and useful for development.

And that's it! 🚀 Your Python app is now running on Google Cloud Run. You've avoided sharing secret credentials, and your friends can access the service without needing direct access to Google Cloud or Gemini. Plus, you're being a responsible sharer—kudos to you!

If you have any more questions or need further guidance, feel free to ask. Happy deploying! 😊 🌟 Learn more about deploying Python services on Cloud Run. It's like giving your app wings! ✨

Learn more: 1. cloud.google.com 2. cloud.google.com 3. cloud.google.com 4. cloud.google.com

You

Reduce this paragraph by aboiut 10%:

A dedicated Computer Science student at the University of Bradford, with a strong academic background including an A* in AQA Computer Science A-levels. Demonstrates exceptional organisational and technical skills through volunteer work at The Children's Society and Boots Pharmacy. Also has demonstrated technical and workplace skills in a 10 week internship working as an AI Researcher for the Bradford Children's and Families Trust. Proficient in web development and software project management, with a passion for creating innovative programming projects. Overcomes challenges with resilience and drive, as evidenced by a successful recovery from a personal health setback. Engages in sports and personal fitness, showcasing a well-rounded character with a commitment to continuous learning and community service.

Copilot

Certainly! Let's trim down that paragraph while keeping the essence intact:

"As a dedicated Computer Science student at the University of Bradford, I hold an impressive academic record, including an A* in AQA Computer Science A-levels. My organizational and technical skills shine through my volunteer work at The Children's Society and Boots Pharmacy. During a 10-week internship as an AI Researcher with the Bradford Children's and Families Trust, I honed my technical and workplace abilities. Proficient in web development and software project management, I'm passionate about creating innovative programming projects. My resilience and determination were evident in my successful recovery from a personal health setback. Beyond academics, I engage in sports and personal fitness, embodying a well-rounded character committed to continuous learning and community service."

Feel free to adjust this further if needed! 😊 🌟

You

Can you reduce it by a further 20%?

Copilot

Certainly! Let's distill that paragraph even further:

"As a dedicated Computer Science student at the University of Bradford, I excel academically, including an A* in AQA Computer Science A-levels. My organizational and technical skills shine through volunteer work at The Children's Society and Boots Pharmacy. During a 10-week AI Researcher internship with the Bradford Children's and Families Trust, I honed my abilities. Proficient in web development and software project management, I'm passionate about innovative programming projects. My resilience and determination were evident in my successful recovery from a personal health setback. Beyond academics, I engage in sports and personal fitness, embodying a well-rounded character committed to learning and community service."

Feel free to tweak this further if needed! 😊🌟

You

Summarise this keeping teh format similar but removing any reference to exam boards and including teh Engineering BTEC as partr of GCSEs:

Copilot

Certainly! Let's create a concise summary while maintaining the format:

"As a dedicated Computer Science student at the University of Bradford, I excel academically. My

---

Conversation by Fifa Utuber with Copilot on Thursday, Oct 3 2024 + Friday, Oct 4 2024