SYSTEM: Strict Mode: Answer only based on verified knowledge, especially for AWS and OpenAI usage limits. If any detail is uncertain, say "I don't know" or explain how to verify it. No speculation, filler, or summaries unless explicitly asked. Focus on factual accuracy only.

User:

Pilot Project:

Because I do this project for the first time, this is still no real requirement list. Please tell me what list it is.. I just write what comes to my mind , one by one.. Help me to name it and organize it:

PHASE 1:

1-I aks ChatGTP to tell me about TinyLlama. Then if I have understood it correctly, I copy a TinyLlamda version on my computer.. somewhere

2-I try to access it , untrained with python.. I have 2 GPU but according to ChatGTP analysis, my computer is still too weak to train this. Still, I ask some question via Python and get reply by TinyLlama

3-I write a Simplle GUI, with Tkinter , where I enger a prompt and TinyLlama gives me a reply and show me in a text filed bellow me

4- I try to train my TinyLlama, locally, with my 15 md files. Nothing more.. ChatGTP will use a tool which is able to do this.. Probably it will warn me that a serious training with my computer isn’t possible.

5-Still I trained some md files, for example my CV and after taining. I ask my TinyLlama, “What do you know about Rusbeh Abtahi”? I expect TinyLlama get some information about me, at least I shouldn’t be unknown to it

6-I play with tuning and changing parameters of TinyLlama, and will get a feeling how LLM works.

Ofcourse If My PC is too weak for that, I got annoyed and we cant test TinyLLama completely .

I will give you with this text also a screenshot of my Computer CPU.. Tel me if PHASE 1 with my PC realistic is.. Tell me what you think about it, and you my correct me

SYSTEM: Strict Mode: Answer only based on verified knowledge, especially for AWS and OpenAI usage limits. If any detail is uncertain, say "I don't know" or explain how to verify it. No speculation, filler, or summaries unless explicitly asked. Focus on factual accuracy only.

User:

Pilot Project:

Ok, I changed my mind.

PHASE 1:

1-Identify a Computer in AWS which can work and train TinyLLma wizj **32-bit float (FP32)**

**2-We define which Tools we are going to install inside this computer.. Could we do it remotely from Python or Lambda in AWS? I mean not manual installation but we create everything by codes.**

3-We write a Watchdog in AWS , which shutdownEC2 after a limited time, for example from 15 Minunt until 30 Minutes

4-We must decide , if we are going to store TinyLLama in S3 or direct in EBS. Which make

5-We need a cost, estimation. An Alert, which will alert us per email, for every 5 €costs and we also need a Log-Analyser, which continuesly predicts and inform us the codes… For example measures the time, our EC2 is running and restore it in a log file

6-We have a Python Code with GUI, Locally, which manage everything. It has a log foder, which stores the costs of AWS and OPEN\_AI Tokens.. So we can avoid any high cost surprises

7-Now that we have a strong AWS computer, we can decide to generate Training Text

-md files Technical files

- Other md files

-Our Chat History (Selective, only those which are interesting for me) I have exported all ChatGTP chats as html and Json

- Standard Guidelines: For Example I can feed a Book I bought, as text “AI-Generative Software Development” or another Book “Prompt Engineering wit Python” or AWS -Guide lines , OR ANY thing you sujjest

8- After preparing and of course learning how to prepare these training texts, you teach me how to train my model. Ofcourse I want the best trained model.

9-I test my trained TinyLlama and try to see, how good TinyLlama alone works.

Please tell me how you see this. How realistic is and how you can help me to develop this Idea….. Please make an Audit and use maximal token

Please also make a Cost Estimation, of EC2 Computers you are choosing me.. in your calculation also Include EBS or S3 costs… Tell me how much it will cost, eachtime I use my tool lets say for 30 Minuets (30 Minuets EC2 computer runs)

Cab you Audit this and Tell me:

A-IF it is realistic to implement this system

B-Do you think I am in right track?

C-Am I using enough Amazon-Services, or there are important Services which I have not seen?

D-Am I exaggerating at any part ? (For example having many trained TinyLlama isn’t in your opinion necessary at this stage)

Please give me your detailed AUDIT with maximum token output

Requirement List:

1-Main -Goals:

I-As main based we have a GUI in local PC, which controls an AWS g5.xlarge (24 GB A10). The Main goal is, to send a prompt from local computer to a trained TinyLlama (FPS16), trained with LoRA, and get the response back again to Local computer and show it in a field text

II-A local version GGUF of trained FPS16 TinyLlama, will always be stored in Local PC. The user can use this reduced Model , offline locally or use FPS16 version online in Amazon

III-User can tune many parameters of both models (Developer must still learn what parameters are tuneable, like stric mode vs creative mode)

IV-Python at base either directly controls EC2 computer or it triggers a Lambda function, where this Lambda function controls EC2 computer (ChatGTP must advice me to choose one method, I prefer the more professional one) , User set a time in GUI, during this time EC2 computer will be available for prompting. Default will be 5 Minuets. The computer will be shutfown only if the shut down time has been reached and no new prompt from user has arrived (EC2 most perform all tasks and the shutdown)

V-It must be me cleared, how we manage IAM protocols. Manually or also via Python Programming from Local Computer( ChatGTP must help me to choose one)

VI-We must design a Watchdog, which automatically closes EC2 or other cost generating resources after a timer has been runed out. This Watchdog must also inform us via Email about costs. The costs must also be shown the user online somewhere on GUI

VII- There will be more than one TinyLlama model on S3. In EC2 we have two options. Using a TinyLlama or training a TinyLlama

VIII-In training Mode, which commands comes from GUI. A set of new training data will be send to another S3 bucket. Then a raw or a already trained TinyLlama will be chosen and will be copied *in Training path on S3. A Python or Lambda command,(trigger by python) will send both training set and TinyLlama model on S3 paths to EC2 , EBS at right place. Then on EC2 , the training mode* , will train TinyLlama with Dataset. After training is finished, the trained model will be sent to a new S3 Bucket, with a Tag (given by user in python) and is saved. The model Name with Tag Name, will automatically appears in Local GUI. User can always see a list of available Models, with Tag name, in GUI local.. So in future he can choose any trained TinyLlama he wants.

After training, the EC2 will be cleaned and shut down. Dataset will be deleted from EBS but will be archived in S3 as a pair with trained TinyLlama mode.. So we always now , which Data have trained our TinyLlama. (Although it will be problematic, if our TinyLlama isn’t raw, and we train an already trained TinyLlama with new dataset.. in this case, we must somehow document that TinyLlama wasn’t trained raw, only if it is necessary. Or maybe we only use raw TinyLlama ‘s for training)

2-Infrastructure:

-Local Computer with Python /GUI

-AWS g5.xlarge (24 GB A10)

3-Steps:

-Copy TinyLlamma (FP16) , untrained in S3

-Create Frontend of Local Python GUI. From here we can manage commanding all resources from Python

-in IAM