**I'm going to create one web application, which will be deployed to AWS cloud for end users.**

**Note:** I’m doing this project to learn stuff. So, make me understand all the tools, technologies and mainly, the process. So, treat me as a beginner and teach me from scratch (whatever it is). Try to include as many tools/technologies/cloud services as possible so that I can get an idea.

**Objective:** Article/Blog Generation ( Input: keyword/statement, output: Article/blog)

I need a simple and attractive UI where user will give the input:

1. the blog topic (keyword/statement),
2. tone (Formal, Informal),
3. length (small, medium or large),
4. language (only English for now, I will implement Telugu in next version)

and it should give the output - Article

**Frontend:** Use HTML/CSS, tailwind/Bootstrap/JavaScript/PHP, ReactJS etc.

**Database:** MySQL, MongoDB

**CI/CD:** Git, GitHub, GitHub Actions, Terraform, Docker, ECR, ECS/EKS, Prometheus/Grafana

**Backend:** Python, Llama model, FastAPI, Prompt Engineering, AWS services

1. I want the user should be able to register by sign-up and login after successful registration.
2. As part of sign-up, collect Full Name, username, password, email ID. While logging in we will take username and password to verify the user.
3. Implement the Authentication using OAuth2.0 or JWT.
4. Let’s develop as many APIs as we can, as I want to learn.
5. End user should be able to download the generated article (UI should have download button/symbol).
6. In the Article History section, I want to showcase last 5 generated articles list along with View and Download option. When user clicks on view It should get expanded.

Let’s figure out where we can fit all these tools/cloud services if required we will take the additional use cases to cover them.

Wherever I mentioned multiple tools for one purpose (like Prometheus/Grafana and ECS/EKS etc.), choose the best which suits our project/scenario.

If you get confused on any requirement do not hesitate to ask me but make sure you should not hallucinate on anything.

Version-2:

Language – Telugu implementation

Authorization Implementation:

1. Protected Route: /generate-article – implemented in V0.1
2. User-Specific Article History: /article-history – implemented in V0.1
3. Download Article: /download/{article\_id} – implemented in V0.1
4. Role-Based Access: /admin/users
5. Token Expiry Handling
6. Account Info Endpoint: /me