Assignment 3

Due: 3rd March 23 05:59 pm

Submission:

1. Github Repo Link

Additional notes:

1. Required attestation and contribution declaration on the GitHub page:
   WE ATTEST THAT WE HAVEN'T USED ANY OTHER STUDENTS' WORK IN OUR ASSIGNMENT AND ABIDE BY THE POLICIES LISTED IN THE STUDENT HANDBOOK
   Contribution:
   - member1: 25%
   - member2: 25%
   - member3: 25%
   - member4: 25%
2. Make sure you do not push anything to your GitHub after submission date (Editing Readme.md is ok but no code pushing after deadline)
3. Create a Codelab document describing everything you did.In your GitHub you should have a readme.md files which would tell what all things are there in this GitHub repository.

Grading Percentage:

1. Each bullet carries 10 points

Carrying forward with Assignment 2, make sure to have Public and Private endpoints with JWT Token based authentication enabled for Private endpoints.

1. Design a service plan as below:
   a. Free - 10 API request limit - reset everyhour
   b. Gold - 15 API request limit - reset everyhour
   c. Platinum - 20 API request limit - reset everyhour
2. Create a user registration page with functionality
   a. Registering as new user and choosing a plan
   b. Changing the password but not fetching the password. Password stored should be hashed
3. To test out please create three different users with each plan
4. Enhance the logging to capture all user activity request this would assist to track and check the user API request count and compare this with the user's enrolled plan
5. Design a dashboard within streamlit accessible by the admin/developers/owner only to track users' activity. Example'
   a. Plotting a line chart of count of request by each user against time (date)
   b. Metric for total API calls the previous day
   c. Metric to show total average calls during the last week.
   d. Comparison of Success (200 response code) and Failed request calls(ie non 200 response codes)
   e. Each endpoint total number of calls
6. Design a dashboard at user level to shows analytics for the logged in user similar to above

7. Use Typer[1] to create a CLI and cover all the functionality
    a. Create a user – command `*yourcli create user someusername*`
    b. Download file by name – command `*yourcli download some123filename.nc*` returns url of the file moved to your s3 location
    c. List all files in a bucket using arguments[2] or prompts[3] or parameters[4]– command `*yourcli fetch GEOS18 ABI-L1b-RadC 2023 005 06*`
8. Create a Python package to access the APIs and create a python package as Wheel[5]
9. Everyday, schedule a run of an airflow dag to update the metadata file
10. Document using Google Codelabs

Other Deliverables

1. All application should be deployed on cloud and accessible to public (No localhost during presentations)
2. Links to Streamlit / Fastapi / Codelab docs in the github README.md file
3. Use Github Issues to log a bug[6] / conversation[7] on your peer repository, following respective templates.
4. Fix for bug should be done using PR's and tagging the the issues[8]

Presentation to cover the following:

1. Demo of application using each of the users from each plans
2. Streamlit dashboard from admin login
3. CLI usage

[1] https://typer.tiangolo.com/
[2] https://typer.tiangolo.com/tutorial/arguments/default/
[3] https://typer.tiangolo.com/tutorial/options/prompt/
[4] https://typer.tiangolo.com/tutorial/parameter-types/
[5] Packaging Python Projects
[6] https://github.com/stevemao/github-issue-templates/blob/master/bugs-only/ISSUE_TEMPLATE.md
[7] https://github.com/stevemao/github-issue-templates/blob/master/conversational/ISSUE_TEMPLATE.md
[8] https://docs.github.com/en/issues/tracking-your-work-with-issues/linking-a-pull-request-to-an-issue