Feedback | Group 2

Table of Contents

- Milestone 1 Tasks
- Milestone 1 Feedback
- Milestone 2 Tasks

Milestone 1 Tasks

- 1. Problem Definition (you can learn more about it here)
- 2. Finalizing roles here
- 3. Schedule a call/meeting with me and Garo
- 4. Create a product roadmap and prioritized functionality (items)
- 5. Create a GitHub repository including readme.md and .gitignore (for Python) files
- 6. Create a virtual environment in the above repo and generate requirements.txt (ensure venv is ignored in git)
 - Create venv: python -m venv venv
 - Activate: source venv/bin/activate
 - Install: fastapi
 - Create requirements.txt: pip freeze > requirements.txt
 - Deactivate: deactivate
- 7. Push Problem Definition, GitHub repo setup (readme.md and .gitignore), requirements.txt
- 8. Prototype the UI using Figma or another similar tool
- 9. Create a private Slack channel in our Workspace and name it Group {number}
- 10. Install VS Code (also install the Project Manager extension)

Milestone 1 Feedback

Problem Definition | 10 points

The problem is defined correctly, and the structure is kept.

- Broad Area of Interest
- Preliminary Research
 - o Current trends
 - o Opportunities
- Solution with Methodology
 - Data Collection
 - Analytical Techniques
 - o Implementation Plan
- Expected Outcomes
- Evaluation Metrics

Grade: 10/10

Roadmap | 10 points

The roadmap seems realistic.

Grade: 10/10

UI | 10 Points

In the UI, I could find the endpoint related, except the log in port. Could you please clarify how the endpoints are going to work. What are going to be the actions?

Grade: 5/10

Administrative Tasks | 5 points

- Roles are assigned
- · Preliminary discussion with me was done
- · Slack channel is created
- · Github Repo is created

Grade: 5/5

Technical Tasks | 5 points

- Proper <u>gitignore</u> file is available for Python
- The Requirments.txt file is available with pre-installed packages, indicating that venv was created

Grade: 5/5

Grade

Final Grade: 35/40

Milestone 2 | Tasks

Product and Project Manager | 20 points

- 1. Install mkdocs package to start with the documentation (PSS will be available)
- 2. Database schema: Provide your product database structure (ERD)
- 3. Transform your project file structure according to the below tree.
- 4. Convert .. docx formats into pdf
- 5. instead of ..._link.txt files, provide proper readme.md file
- 6. check all the bellow activities from your team and merge everything

```
PythonPackageProject/ #githhub repo

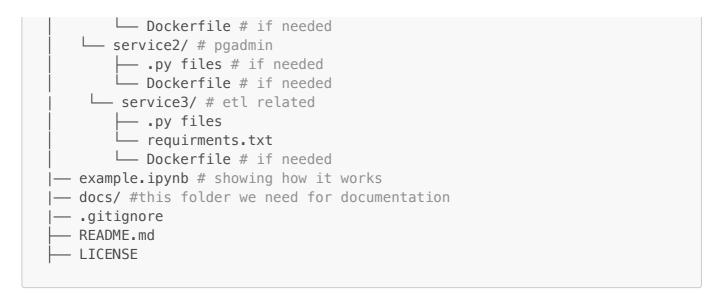
— yourapplications/

— docker-compose.yaml

— .env

— service1/ #postgress

— .py files # if needed
```



Data Scientist and Data Analyst | 20 points

- 1. Create a new git branch and name it ds
- 2. Simulate the data if you need
- 3. Try to use the CRUD functionality done by DB Developer
- 4. Work on modeling part using simple models, conduct extra research
- 5. Push your works to respective branch
- 6. Create pull request for the Product Manager

Database Developer | 30 points

- 1. Create a new git branch and name it db
- 2. Create a DB and respective tables suggested by the Product Manager
- 3. Connect to SQL with Python
- 4. Push data from flat files to DB
- 5. Add extra methods that you might need throughout the project
- 6. Push your works to respective branch
- 7. Create pull request for the Product Manager

API Developer | 30 points

- 1. Create a new git branch and name it back
- 2. Create a new service and name it back
- 3. Communicate with the DB Developer and PM in order to design the API
- 4. You can create dummy endpoints in the beginning (PSS will be available)
- 5. The following endpoints must be available:
 - 1. GET
 - 2. POST
 - 3. PUT
 - 4. DELETE
- 6. Push your works to respective branch
- 7. Create pull request for the Product Manager

Front End Developer | 20

- 1. Create a new git branch and name it front
- 2. Create a container/service and name it front
- 3. Communicate with the PM in order to create the skeleton of the website.
- 4. Push your works to respective branch
- 5. Create pull request for the Product Manager