

Feedback | Group 4

Milestone 1

Problem Definition | 20 points

The problem is clearly defined however the solution is quite vague

- Broad Area of Interest
- Preliminary Research
 - Current trends
 - Opportunities
- Solution with Methodology
 - Data Collection: **how are you going to get the data**
 - Analytical Techniques **analytical techniques wasn't defined**
 - Implementation Plan
- Expected Outcomes: **the API is still unclear**
- Evaluation Metrics

Grade: 10

Roadmap | 10 points

Grade: 10

Administrative Tasks | 5 points

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

Grade: 5

Technical Tasks | 5 points

- Proper **.gitignore** file is available, however python track wasn't added
- The Requirments.txt file is available, indicating that **venv** was created
- The first chapter of the Package Development course is done by **everyone**

Grade: 4

Grade

Overall you did really great job during the M1. Keep it like that!

Final Grade: 39/40

Milestone 2 | Tasks

From the milestone 1:

- Clarify data collection
- Analytical technique (how are you going to deal with the images)
- How the API is going to look like?

Prudoct and Project Manager | 40 points

1. Name your Python package: register to [pypi](#)
2. Install [mkdocs](#) package to start with the documentation
3. Database schema: Provide your product database structure (ERD)
4. Transform your project file structure according to the below tree

```
PythonPackageProject/ #github repo
├── yourpackagename/
│   ├── __init__.py
│   ├── submodule1/ #database related
│   │   ├── __init__.py
│   │   └── submodule1_1.py
│   ├── submodule2/ #model related
│   │   ├── __init__.py
│   │   └── submodule1_2.py
│   └── submodule3/ # api related
│       ├── __init__.py
│       └── submodule1_2.py
├── tests/
│   ├── __init__.py
│   ├── test_module1.py
│   └── test_module2.py
├── example.ipynb # showing how it works
├── run.py # in order to run an API
├── docs/ #this folder we need for documentation
├── .gitignore
├── requirments.txt
├── README.md
├── LICENSE
└── setup.py
```

Data Scientist and Data Analyst | 20 points

1. Simulate the data if you need
2. Try to use the CRUD functionality done by DB Developer
3. Work on modeling part using simple models

```
from yourpackage.submodule2 import modelname
```

Database Developer | 30 points

1. Create a DB and respective tables suggested by the Product Manager
2. Connect to SQL with Python
3. Push data from flat files to DB
4. Test the code provided [here](#) and complete the missing components
5. Add extra **methods** that you might need throughout the project:
 1. Communicate with PM and API Developer for custom functionality

```
from yourpackage.submodule1 import sqlinteractions
```

API Developer | 30 points

1. Communicate with DB Developer and PM in order to design the API
2. You can create dummy endpoints in the beginning, then communicate with PM as well
3. The following endpoints must be available:
 1. GET
 2. POST
 3. UPDATE

Check out this [this repo](#).

```
from yourpackage.submodule2 import api
```