

Table of Contents

1. [Milestone 1](#)
2. [Milestone 2](#)
 - [Fixes from Milestone 1](#)
 - [Feedback on Milestone 2](#)
 - [Feedback on Milestone 2 Revised](#)
3. [Milestone 3](#)
4. [Milestone 4](#)
5. [Final Feedback](#)
6. [Grade](#)

Feedback | Group 3

Milestone 1 | 2Oct-13Oct

1. **Define the problem:** not done
 - Please change the file format from .txt to .md , as it is not readable
 - I couldn't understand the link between your problem and Marketing.
2. **Finalizing roles:** done
3. **Create a product roadmap and prioritize functionality (items):** done
4. **Creating the GitHub repository included readme.md and .gitignore (for Python) files:** done
5. **Create a virtual environment in the above repo and generate requirements.txt (venv must be ignored in git)** done
6. **Push *point 1, point 3, point 5 (requirements.txt)*.** done:
7. **Complete the first chapter of Developing Python Packages:** done
8. **Create a private Slack channel in our Workspace and name it Group-{number}** done
9. **Schedule a call with me and Garo or come during the office hours:** done

By the end of the Milestone 2, you must complete the tasks mentioned above. Feel free to reach out if you have any questions.

Explain the Problem and the Solution

Grade: 8/10 (I hope you will explain me your project)

Milestone 2 | 16Oct-27Oct

Fixes From the Milestone 1

I can see that you have changed the topic into book recommendation. In the problem definition I'd like clearly see the recommendation approach.

Milestone 2

1. DB developer:

- Design the database using Star schema (provide ERD): **done**
- Insert Sample to data **done**
- **the structure is wrong**

2. Data Scientist:

- Complete data generation/acquisition/research: **done**
- Select data from DB: **done**
- Insert data to DB: **done**

3. API developer:

- Select data from DB **done**
- Insert data to DB **done**
- Update data in DB **wrong arguments**

4. Finish the second chapter of Datacamp course **done**

5. Finalize file/folder structure: relative imports must work properly **not done**

- docs folder: putting all the documents there **not done**
- models folder: putting modeling-related classes, functions **not done**
- api folder: api related stuff **not done**
- db folder: db related stuff **not done**
- initialize `__init__.py` files accordingly (see Datacamp assignment chapter 1 and chapter 2) **not done**
- logger folder: I will provide this module **done**

I can see multiple contributors

In order to improve you performance I would recommend:

- approach the datacamp course seriously (it is obvious You are just taking the hints and completing it)
- start to work on group project before the deadline

Remember you are building a package, like in the Datacamp you must have following file structure:

```
| GitHubRepo
|   | PackageName
|   |   | SubPackage_1
|   |   |   module1
|   |   |   __init__.py
|   |   | SubPackage_2
|   |   |   module2.py
```

```
__init__.py
__init__.py
utils.py
setup.py
example.py/ipy nb (from PackageName import SomeModule)
```

By the end of the 3rd Milestone you must **fix folders and their relationships**

If you manage to complete the above points by Friday (before the class) you will get

Grade: 10/20

Milestone 2 | Revision

You have definitely improved however the structure still is not the same as the above schema (check one more time, I added `utils.py` place as well).

- Create package and insert modules there
- Please remove `_Folder` suffix from your directories.
- Keep Data_Folder (Data) folder out the package
- No need to `__init__.py` in Docs folder

Grade: 15/20

Milestone 3 | 30Oct-10Nov

1. Complete things from *Milestone 2* **done**
2. Finish the **third** chapter of Datacamp course (please complete only the 3rd one) **done by everyone**
3. **API Developer:** **partially done**
 - Create a **run.py** file for an API (find the minimum workable example [here](#))
 - **your run.py file should be out of the package. api is simply another module**
 - **utils.py must be out of the **api** folder, see at the file structure section**
 - Test it on swagger
 - following request types must be available to test (GET, POST, PUT), will provide more details on Friday. **done**
4. **DB developer:** **done**
 - complete/fix the methods from **SQLHandler()** class **done**
 - finalize the documentation for **schema.py** by using **pyment** package **done**
 - finalize the documentation for **SQLHandler()** by using **pyment** package **done**
5. **Data Scientist:** start working on modeling part, by selecting the date from SQL DB **done**
 - we just need to run sample model and store the output to sql **done**

Grade: 20/30

Milestone 4 | 26 Nov-6 Dec

1. Complete things from *Milestone 3*
 - making sure `__init__.py` are working properly
 - complete the restructuring
 - Create `.db` file out of the package
2. Complete the final (4th) chapter of Datacamp course
3. Make API work with the database instead of `csv`
4. In `readme.md` provide exact steps on running the package (you have tried to do that already.)

You have all the functionality guys, now we need to work on making it work as a package. Following steps must support:

1. `from bookstore.models import dataprep`: to create a db
 2. `from bookstore.models import model1 etc.:` to test the model
 3. `from bookstore import api`: to run an api, and the endpoints must recommend books
-

Optional for the Milestone 4 , mandatory for the group project:

1. Create an `example.ipynb` file (out of the package) and implement all the package's functionality (make sure to make it chunk by chunk to convert it `reveal.js` presentation). This is going to be part of the demo.
2. As soon as you finish the documentation us `Mkdocs` in order to generate `docs.html` file, which is going to be hosted on GitHub
3. Publish your package to `pypi.org`

Final Feedback

Group Project Scope

- Finding a Marketing related problem
- Understanding the methodology of the analysis
- Building a Python package with following mandatory modules:
 - Predictive Model (component)
 - DB
 - API
 - Logging (provided by me)
- Post to [Pypi.org](https://pypi.org)

Submission format

In the Github Repository, the following structure must be available

```
| GitHubRepo
| Docs
| PackageName
|   SubPackage_1
|       module1
|       __init__.py
|   SubPackage_2
|       module2.py
|       __init__.py
|   __init__.py
|   utils.py
other files (.gitignore, *config files)
readme.md
requirements.txt
setup.py
example.py/ipynb (Demonstrate all the functionality)
```

Submission format is correct.

Grading Methodology

Group Project is going to be graded according to the following points:

1. **Topic Relevancy:** [matched but not correctly demonstrated](#)
2. **Team Work:** [I can see the contributions from each member](#)
3. **Availability of Documentation:** [Perfect](#)
 - Description of each [function\(\)/method\(\)](#):
 - Parameters: description/docstrings
 - Returns: what do you expect as a return?

- Description of Classes:
 - Use *dunder methods*: `__repr__`, `__str__`, for nice Class formulation
 - Describe the class
 - converting into a webapp using `mkdocs` or any alternative
4. **The code must run without any errors:** OK
 - logical
 - syntax
 - runtime error
 5. **The availability of a Predictive Element** CLV
 6. **Endpoints solving/touching the business problem** OK
 7. **Successfully hosted on Pypi** Done

Final Feedback

Technically you have done everything which was required. However you could have done way better by simply trying to understand what do you need to do at first.

It is obvious that there was a communication problem between **product/project manager** and the **team**. There was a gap between business/marketing and package logic.

Grade

- **Grade from the Milestones:** 78
- **Grade from the Presentation:** 300/300
- **Final Grade:** 378