Project Title: Pythonic

1. Introduction

- Briefly describe the key features of your game.

Answer random interactive python questions via the terminal and get prompted if 👍 correct or 💩 incorrect. One point gets added users total score if correct. An, incorrect answer creates a prompt displaying the correct answer. In addition, correct answers are written to a txt file for later review.

2. Design and Implementation

I wanted something vibrant and semi-interactive so I utilized Rich the python package, and configured the input method to allow users to press enter without triggering a submit.

- Give a detailed description of the design and implementation of your project.

I wanted the correct, incorrect, writing to file, and spacing, and other methods separated from the main application. I imported everything in via modules

- In particular, this section should contain:

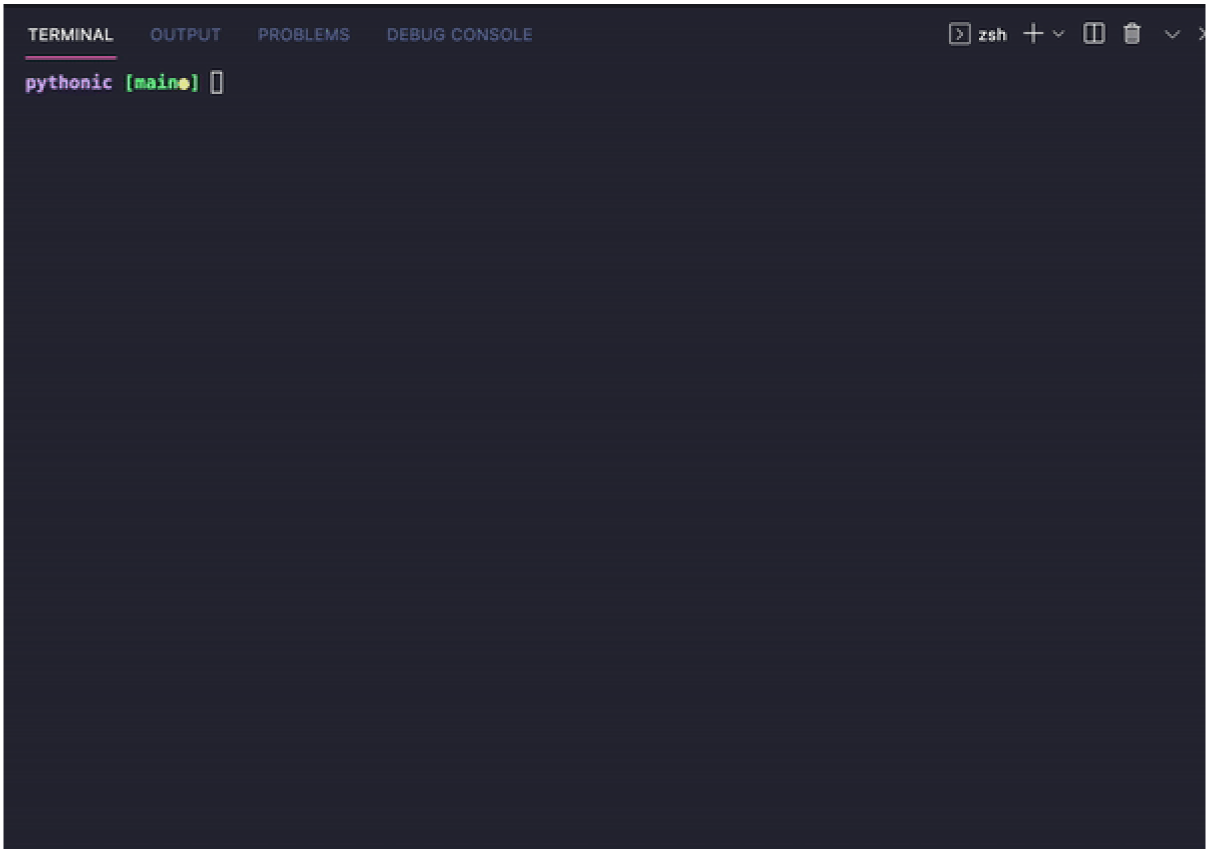
- Details of how you converted from design to the actual realization of your project, in terms of implementing the code.

I wanted to have a clean layout but not just a standard terminal app, so I added some visual appeal to it, via Rich. I kept i simple, a hero with a title and instructions. The instructions only vanish after answering your first question.

- Any choices that you made, and any modifications that you made to the design, in response to difficulties that you might have encountered while implementing the project.

No not really. Terminal apps seem to be straight forward

- Include relevant screenshots of your game at different stages of play in the report.



```
pythonic [main●]
```

3. Conclusions

- Discuss what you personally learned from your project.

I should have use classes and make the main method more clean. I learned how to use a package and pyenv virtual env

- Discuss the best features and the shortcomings of the project.

I wanted to add a timer and audio off and on switch and allow audio to work for windows and linux. In addition a harder level after 10 beginner questions but I decided to just randomize the questions and keep it simple

- Discuss any choices that you might have made differently, in hindsight after completing the project.

The use of classes. Classes seem more structured than procedural and or functional programing

- Describe any additional features you may want to add in the future.

Timer, audio option, click effect, harder level, ability to indent multiple lines of code, use tabs instead of spaces