Name: Shivang Bhut

NSID: ayw932

Inspection Checklist for Team_23

Product: Retail Billing System

Checklist to check for code's correctness, organization, security, performance, documentation, and functionality, to meet system requirements and standard.

General:

- Does the code function as intended?
  Answer: Yes**,** all tests pass successfully so code works well.

- Is the code well-organized and easy to understand?
  Answer: Yes**,** it is easy to understand.

- Is there any duplicated code that could be refactored?
  Answer: No, I didn't find any code that needs to be refactored.

- Have all unnecessary or commented-out code blocks been removed?
  Answer: Yes**,** all unnecessary code was removed.

- Is the code modular and promotes re-usability?
  Answer: Yes, code is broken down into functions so whenever you need it you call or access anywhere.

- Are loop conditions and termination conditions well-defined and clear?
  Answer: There are no loops, here basically it is one part of GUI call the other part.

- Are all variable names meaningful and clear?
  Answer: Yes**,** I especially like this as naming was done precisely and I don't have to find and think jut by reading you come to know.

- Does the code handle edge cases correctly?
  Answer: Not Applicable, because in GUI either it works or doesn't.

Python-Specific:

- Does the code pass pylint?
  Answer: Yes, it gave 10/10 ratings.

- Is the code properly formatted using black?
  Answer: Yes, it is.

Security:

- Is user authentication and authorization handled correctly?
  Answer: Yes, it handles good.

- If SQL queries are used are proper placeholders used to prevent SQL injection?
  Answer: Not Applicable, because it has nothing to do with SQL here.

Documentation:

- Does each module have a docstring explaining its purpose and functionality?
  Answer: Yes, there is proper docstring.

- Does each class have a docstring explaining its purpose and functionality?
  Answer: Yes, it has.

- Does each method have a docstring explaining its purpose and functionality?
  Answer: Yes, it has.

- Is any special case behavior documented?
  Answer: Yes, here there is no special case.

- Are inline comments used to provide clarity where necessary?
  Answer: Yes, it was there.

Performance:

- Is the code optimized for performance, with no obvious bottlenecks or inefficiencies?
  Answer: Yes, the code is optimized.

- Can any computational or database operations be further optimized?
  Answer: Not Applicable, as no database operation is required.

Testing:

- Is the code structured to be easily testable, with well-defined functionality?
  Answer: Yes, code is structured well.

- Are there comprehensive tests covering different features and scenarios?
  Answer: No, there are no comprehensive tests covering different features and scenarios

- Do the test cases adequately check the code's functionality and edge cases?
  Answer: No Edge cases required here because either it will work or not.