NAME: DHARINI

ROLL NO : ECA2321

SUB : PYTHON PROGRAMMING

SUB CODE : ECA1121

TOPIC: COURSE EVALUATION SYSTEM WITH USER RATING

CONTENT :

INTRODUCTION

PYTHON USED

WORK FLOW

MODULE COMPLETION

PROS CONS

INTRODUCTION

One crucial aspect of this improvement process is gathering rating from students about their courses, instructors, and overall learning journey. To streamline this process and provide valuable insights to educators and administrators, we embark on the journey of creating a Course Evaluation System with User rating Collection using the versatile and powerful programming language, Python.

PYTHON USED

Python is used to create the core functionality of the system, like handling user logins, managing courses, and processing rating feedback submissions. Python interacts with the database to store and retrieve information about users, courses, and rating .

A class is a blueprint for creating objects. Classes encapsulate data for the object and methods to manipulate that data.

1. Defining a Class To define a class, you use the class keyword followed by the class name and a colon. Inside the class, you define methods and attributes.

WORK FLOW

Start Program: Execute the script, and the main function is called.

Create Course: User inputs the course name, and a Course object is created. Display Menu: Show options (Add Rating, View Average Rating, Exit). Get User Choice: User inputs a choice.

Process Choice

Add Feedback: Validate and add user rating feedback.

View Average Rating: Calculate and display the average rating. Exit: Exit the program.

Invalid Choice: Show error message and re-display menu. End Program: Exit the loop and terminate the program.

PROS

Easy to use: The program has a clear menu and simple options.

Reusable: The Course class can be used in other programs.

Validates input: Ensures feedback ratings are between 0 and 10.

Handles multiple rating : Can add many rating and calculates the average.

Deals with no feedback: Shows a message if there is no feedback yet.

Well-organized: Code is clear and easy to follow.

Expandable: New features can be added easily.

Readable: Code is simple and well-commented.

CONS

Only handles numeric rating feedback: It doesn't allow for comments or detailed input.

No data saving: Feedback isn't stored anywhere, so it's lost when you close the program.

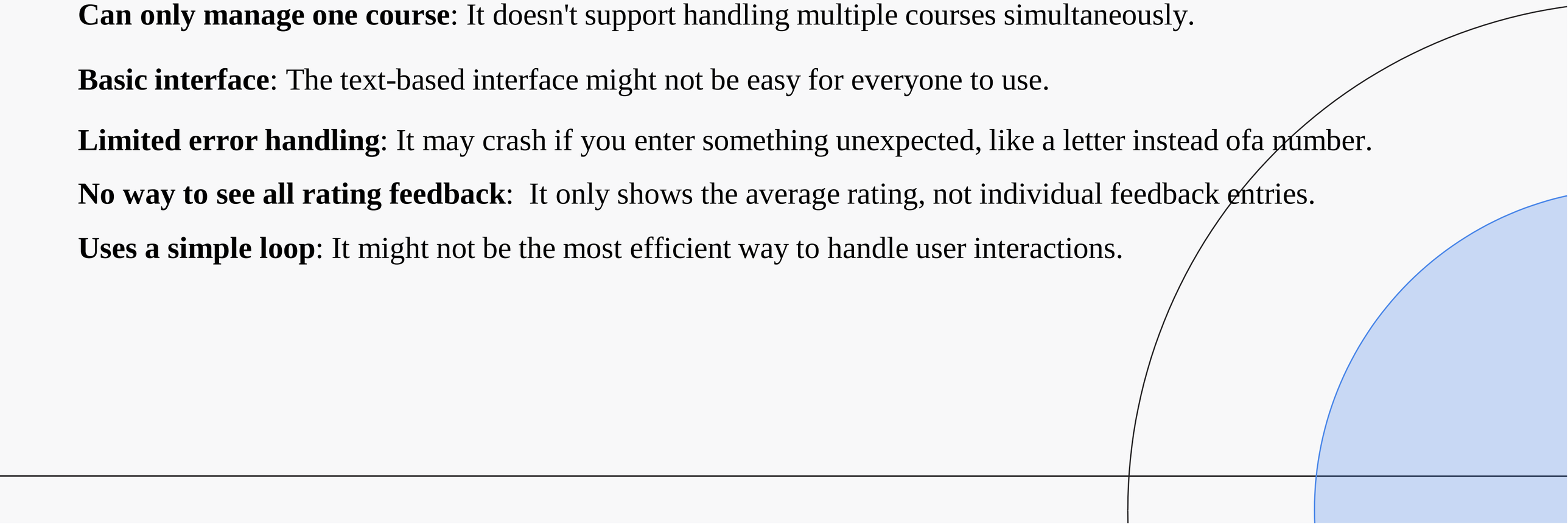
Can only manage one course: It doesn't support handling multiple courses simultaneously.

Basic interface: The text-based interface might not be easy for everyone to use.

Limited error handling: It may crash if you enter something unexpected, like a letter instead ofa number.

No way to see all rating feedback: It only shows the average rating, not individual feedback entries.

Uses a simple loop: It might not be the most efficient way to handle user interactions.



CONCLUSION

The program effectively allows users to add feedback and view average ratings for a single course. However, it has limitations such as basic error handling, lack of data storage, and a simple user interface. It's suitable for basic rating feedback management but would benefit from improvements like handling multiple courses, saving data, and enhancing user interaction

```
class Course:
def _init_(self, name): self.name = name self.feedback = []
def add_feedback(self, feedback): self.feedback.append(feedback)
def get_average_rating(self): if not self.feedback:
return "No feedback yet." total_rating = sum(self.feedback) return
total_rating / len(self.feedback)
def main():
course_name = input("Enter the name of the course: ") course =
Course(course_name)
while True:
print("\n1. Add Feedback")
print("2. View Average Rating") print("3. Exit")
choice = input("Enter your choice: ") if choice == "1":
feedback = float(input("Enter your feedback rating (0-10): "))
```

```
if feedback < 0 or feedback > 10:
print("Invalid feedback rating. Please enter a rating between 0 and 10.") continue
course.add_feedback(feedback) print("Feedback added successfully!")
elif choice == "2":
print(f"Average rating for {course.name}: {course.get_average_rating()}") elif
choice == "3":
print("Exiting program.") break
else:
print("Invalid choice. Please enter a valid option.") if _name_ == "_main_":
main()
```

```
Enter the name of the course: Mathematics
1. Add Feedback
2. View Average Rating
3. Exit
Enter your choice: 1
Enter your feedback rating (0-10): 8
Feedback added successfully!

1. Add Feedback
2. View Average Rating
3. Exit
Enter your choice: 1
Enter your feedback rating (0-10): 7
Feedback added successfully!

1. Add Feedback
2. View Average Rating
3. Exit
Enter your choice: 2
Average rating for Mathematics: 7.5

1. Add Feedback
2. View Average Rating
3. Exit
Enter your choice: 3
Exiting program
```


Thank you