Final Assignment

Objective

The goal of this assignment is to develop a small Python application that addresses a specific problem or improves efficiency within a business context. You will design and implement an application with basic functionalities such as data input, data editing, reading data from a file, and outputting results. Additionally, you are encouraged to include advanced functionalities to enhance the application's utility.

Assignment Structure

Your assignment will be divided into several stages with specific checkpoints to ensure steady progress. You will be expected to submit your work at each checkpoint for review and feedback.

Project Outline

Title of the Project

- ○ Choose a relevant title that clearly reflects the business problem your application aims to solve.

Introduction:

- ○ Provide a brief introduction to the problem you are addressing.
- ○ Explain why this problem is significant in a business context.
- ○ Mention the scope of the application and the expected impact on business efficiency.

Requirements:

- The application should be developed using Python.
- The application must have a user-friendly interface with a menu.
- The menu should include options to input data, edit data, read data from a file, and output results.
- The application should handle basic error checking and data validation.
- Implement at least one advanced functionality to showcase more sophisticated programming concepts.

Minimal Functionality (xxx points maximum)

- Main Menu:
    - Options to:
        - Input new data
        - Edit existing data
        - Read data from a file
        - Output results to the screen or a file
        - Exit the application
- Input Data:
    - Allow users to enter relevant data through the console.
    - Ensure data is stored in an appropriate data structure (e.g., list, dictionary).
- Edit Data:
    - Provide functionality to modify existing data entries.
- Read Data from File:
    - Implement file reading functionality to load data from a text or CSV file.
- Output Results:
    - Display results in a readable format on the console.
    - Optionally, write results to a file (text or CSV).

Advanced Functionality (additional yyy points):

- Export results to external files (e.g., CSV, Word, Excel)
- Validate data and handle errors
- Implement data visualization (e.g., graphs using matplotlib).
- Integrate a simple database for data storage.

- Implement advanced data analysis (e.g., statistical analysis, forecasting).
- Develop a GUI.
- Implement authentication and user management.

Checkpoints:

Checkpoint 1: Project Proposal (date)

- Describe the problem you aim to solve.
- Outline the main functionalities of your application.
- Sketch the application design
- Submit a brief proposal (1-2 pages) for feedback.

Checkpoint 2: Main Menu Implementation (date)

- Develop the main menu with all the required options.

Checkpoint 3: Core Functionality (date)

- Implement data input, editing, reading from a file, and output functionalities
- Implement the main functionality
- Address the obtained feedback

Checkpoint 4: Advanced Functionality (date)

- Implement the advanced functionality.
- Addressed the obtained feedback

Checkpoint 5: Final Submission (date)

- Finalize the application with all functionalities.
- Prepare a short report (presentation?) (3-5 pages) documenting:
  - The problem statement
  - Application functionality
  - How to use the application
  - Advanced functionality and its implementation

Application Examples:

1. Business Analysis Application:
    - Problem: Analyze sales data to identify trends and improve sales strategies.
    - Main Functionality:
        - Input sales data (e.g., date, product, quantity, price).
        - Edit existing sales records.
        - Read sales data from a CSV file.
        - Output sales summary and trends.
    - Advanced Functionality:
        - Generate sales trend graphs.
        - Implement forecasting algorithms to predict future sales.
2. Mini CRM (Customer Relationship Management):
    - Problem: Manage customer interactions and data to improve customer service.
    - Main Functionality:
        - Input customer data (e.g., name, contact).
        - Input transaction data (date, amount)
        - Edit customer records.
        - Read customer data from a file.
        - Output customer profiles and transaction history.
    - Advanced Functionality:
        - Implement customer segmentation analysis.
        - Integrate a simple SQLite database for customer data storage.
3. Profit Calculation and Analysis:
    - Problem: Calculate and analyze profit margins for various products.
    - Main Functionality:
        - Input costs data
        - Input sales data for products
        - Edit product data
        - Read data from a file
        - Output profit margins and summary.
    - Advanced Functionality:
        - Visualize profit margins using bar charts
        - Implement break-even analysis
4. Inventory Management:

- Problem: Track and manage inventory levels to avoid stockouts and overstock.
- Main Functionality:
  - Input inventory data (products, quantities).
  - Input transactions (sale, restock)
  - Edit inventory records.
  - Read inventory data from a file
  - Output inventory status and alerts for low stock.
- Advanced Functionality:
  - Generate inventory turnover reports.
  - Implement predictive restocking based on sales trends.

5. Profitability, Liquidity, Viability Analysis:
   - Problem: Assess the financial health of a business.
   - Main Functionality:
     - Input financial data (e.g., income, expenses, assets, liabilities).
     - Edit financial records.
     - Read financial data from a file.
     - Output key financial ratios and analysis.
   - Advanced Functionality:
     - Implement graphical financial reports (e.g., pie charts for expense distribution).
     - Conduct scenario analysis to forecast financial outcomes.

Submission Guidelines:

-

Evaluation Criteria:

-