## This is the prompt to generate an investment report summary.

```
print("""
Input Data Structure:
1. Client Profile:
   - Name: [Client Name]
   - Investment Goals: [Investment Objectives]
   - Risk Tolerance: [Risk Profile]

2. Portfolio Performance Metrics:
   - Year-to-Date Returns: [Value]
   - Annualized Returns: [Value]
   - Asset Allocation:
     - Equities: [Percentage]
     - Bonds: [Percentage]
     - Cash: [Percentage]
   - Risk Metrics:
     - Sharpe Ratio: [Value]
     - Standard Deviation: [Value]

3. Benchmarks:
   - Equities Benchmark: [Benchmark Name and Return Value]
   - Bonds Benchmark: [Benchmark Name and Return Value]

Required Output Format:
- Header: Include the client’s name and report date.
- Section 1: Portfolio Performance Summary.
  - Compare portfolio metrics with benchmarks.
  - Highlight alignment with the client’s goals and risk tolerance.
- Section 2: Actionable Insights.
  - Suggest portfolio adjustments to improve performance or manage risks.
- Section 3: Compliance Disclosures.
  - Standard financial disclaimers:
    - "Past performance is not indicative of future results."
    - "This report is for informational purposes only and does not constitute financial advice.""")
```

```
Input Data Structure:
1. Client Profile:
   - Name: [Client Name]
   - Investment Goals: [Investment Objectives]
   - Risk Tolerance: [Risk Profile]

2. Portfolio Performance Metrics:
   - Year-to-Date Returns: [Value]
   - Annualized Returns: [Value]
   - Asset Allocation:
     - Equities: [Percentage]
     - Bonds: [Percentage]
     - Cash: [Percentage]
   - Risk Metrics:
     - Sharpe Ratio: [Value]
     - Standard Deviation: [Value]

3. Benchmarks:
   - Equities Benchmark: [Benchmark Name and Return Value]
   - Bonds Benchmark: [Benchmark Name and Return Value]

Required Output Format:
- Header: Include the client’s name and report date.
- Section 1: Portfolio Performance Summary.
  - Compare portfolio metrics with benchmarks.
  - Highlight alignment with the client’s goals and risk tolerance.
- Section 2: Actionable Insights.
  - Suggest portfolio adjustments to improve performance or manage risks.
- Section 3: Compliance Disclosures.
  - Standard financial disclaimers:
    - "Past performance is not indicative of future results."
    - "This report is for informational purposes only and does not constitute financial advice.
```

**This is code to implement the data of the client into the prompt get an investment report summary**

---

If the client have necessary data as an single report they can easly upload the file in json format to get the report summary of their investment

```
import json
from datetime import datetime

class FinancialReportGenerator:
    def __init__(self, prompt_template, input_data):
        self.prompt_template = prompt_template
        self.input_data = input_data

    def generate_header(self):
        return f"""Investment Portfolio Report
Client: {self.input_data['client_profile']['name']}
Date: {datetime.now().strftime('%B %d, %Y')}
------------------------------------------"""

    def generate_performance_summary(self):
        portfolio = self.input_data['portfolio_data']
        benchmarks = self.input_data['benchmarks']

        return f"""Portfolio Performance Summary
-------------------------------------------------
Year-to-Date Return: {portfolio['year_to_date_returns']}%
Annualized Return: {portfolio['annualized_returns']}%

Asset Allocation:
- Equities: {portfolio['asset_allocation']['equities']}%
- Bonds: {portfolio['asset_allocation']['bonds']}%
- Cash: {portfolio['asset_allocation']['cash']}%

Risk Metrics:
- Sharpe Ratio: {portfolio['risk_metrics']['sharpe_ratio']}
- Standard Deviation: {portfolio['risk_metrics']['standard_deviation']}%

Benchmark Comparison:
- S&P 500: {benchmarks['equities']['S&P 500']}%
- Bloomberg US Aggregate Bond Index: {benchmarks['bonds']['Bloomberg US Aggregate Bond Index']}%"""

    def generate_insights(self):
        portfolio = self.input_data['portfolio_data']
        client = self.input_data['client_profile']
        insights = []
        if client['risk_tolerance'] == 'Moderate' and portfolio['asset_allocation']['equities'] > 70:
            insights.append("Consider reducing equity exposure to better align with your moderate risk tolerance.")
        if portfolio['year_to_date_returns'] < self.input_data['benchmarks']['equities']['S&P 500']:
            insights.append("Portfolio is currently underperforming the S&P 500 benchmark. Review individual holdings for optimization opportu
        if portfolio['asset_allocation']['cash'] > 15:
            insights.append("High cash position may be dragging on returns. Consider deploying capital into investment opportunities.")
        return """Actionable Insights
-----------------------------------------------------
""" + "\n".join(f"- {insight}" for insight in insights)

    def generate_disclosures(self):
        return """Compliance Disclosures
-----------------------------------------------------
1. Past performance is not indicative of future results.
2. This report is for informational purposes only and does not constitute financial advice.
3. Investment values will fluctuate, and you may lose money.
4. Always consult with a qualified financial advisor before making investment decisions."""

    def generate_report(self):
        sections = [
            self.generate_header(),
            self.generate_performance_summary(),
            self.generate_insights(),
            self.generate_disclosures()
        ]
        return "\n\n".join(sections)

prompt_template = """You are an expert financial analyst tasked with generating tailored investment reports for clients..."""
file_path = "data.json"
with open(file_path, "r") as file:
    test_data = json.load(file)
```

```
report_generator = FinancialReportGenerator(prompt_template, test_data)
generated_report = report_generator.generate_report()
print(generated_report)
```

```
Investment Portfolio Report
Client: John Doe
Date: December 20, 2024
----------------------------------------

Portfolio Performance Summary
--------------------------------------------------
Year-to-Date Return: 8.5%
Annualized Return: 7.2%

Asset Allocation:
- Equities: 60%
- Bonds: 30%
- Cash: 10%

Risk Metrics:
- Sharpe Ratio: 1.5
- Standard Deviation: 12.3%

Benchmark Comparison:
- S&P 500: 9.0%
- Bloomberg US Aggregate Bond Index: 2.5%

Actionable Insights
----------------------------------------------------------
- Portfolio is currently underperforming the S&P 500 benchmark. Review individual holdings for optimization opportunities.

Compliance Disclosures
----------------------------------------------------------
1. Past performance is not indicative of future results.
2. This report is for informational purposes only and does not constitute financial advice.
3. Investment values will fluctuate, and you may lose money.
4. Always consult with a qualified financial advisor before making investment decisions.
```

**This will also generate the same report summary of the investment but if the client doesn't have proper report for their investment, they can fill the every detail's individually**

```
def get_client_input():
    """Get client portfolio data from user input"""
    print("\n=== Client Profile Input ===")
    client_profile = {
        "name": input("Enter client name: "),
        "investment_goals": input("Enter investment goals (e.g., Long-term growth): "),
        "risk_tolerance": input("Enter risk tolerance (Conservative/Moderate/Aggressive): ")
    }

    print("\n=== Portfolio Data Input ===")
    portfolio_data = {
        "year_to_date_returns": float(input("Enter year-to-date returns (%): ")),
        "annualized_returns": float(input("Enter annualized returns (%): ")),
        "asset_allocation": {
            "equities": float(input("Enter equity allocation (%): ")),
            "bonds": float(input("Enter bond allocation (%): ")),
            "cash": float(input("Enter cash allocation (%): "))
        },
        "risk_metrics": {
            "sharpe_ratio": float(input("Enter Sharpe ratio: ")),
            "standard_deviation": float(input("Enter standard deviation (%): "))
        }
    }

    print("\n=== Benchmark Data Input ===")
    benchmarks = {
        "equities": {
            "S&P 500": float(input("Enter S&P 500 return (%): "))
        },
        "bonds": {
            "Bloomberg US Aggregate Bond Index": float(input("Enter Bond Index return (%): "))
        }
    }

    return {
        "client_profile": client_profile,
        "portfolio_data": portfolio_data,
```

```
            "benchmarks": benchmarks
        }

    def create_report(client_data):
        from datetime import datetime
        today = datetime.now().strftime("%B %d, %Y")

        report = []
        report.append(f"""
    Investment Portfolio Report
    ========================
    Client: {client_data['client_profile']['name']}
    Date: {today}
    """)
        portfolio = client_data['portfolio_data']
        report.append(f"""
    Portfolio Overview
    ----------------
    Investment Goals: {client_data['client_profile']['investment_goals']}
    Risk Tolerance: {client_data['client_profile']['risk_tolerance']}

    Performance:
    - Year to Date Return: {portfolio['year_to_date_returns']}%
    - Annualized Return: {portfolio['annualized_returns']}%

    Current Asset Allocation:
    - Stocks: {portfolio['asset_allocation']['equities']}%
    - Bonds: {portfolio['asset_allocation']['bonds']}%
    - Cash: {portfolio['asset_allocation']['cash']}%
    """)
        benchmarks = client_data['benchmarks']
        report.append(f"""
    Risk and Benchmark Analysis
    -------------------------
    Risk Metrics:
    - Sharpe Ratio: {portfolio['risk_metrics']['sharpe_ratio']}
    - Volatility (Standard Deviation): {portfolio['risk_metrics']['standard_deviation']}%

    Benchmark Comparison:
    - S&P 500: {benchmarks['equities']['S&P 500']}%
    - Bond Index: {benchmarks['bonds']['Bloomberg US Aggregate Bond Index']}%
    """)
        report.append("""
    Investment Insights
    -----------------""")

        insights = []
        if portfolio['year_to_date_returns'] < benchmarks['equities']['S&P 500']:
            insights.append("- Your portfolio is currently underperforming the S&P 500. Consider reviewing your stock selection.")
        else:
            insights.append("- Your portfolio is performing well compared to the S&P 500 benchmark.")
        if portfolio['asset_allocation']['cash'] > 15:
            insights.append("- Your cash position is relatively high. Consider investing excess cash to potentially improve returns.")
        if client_data['client_profile']['risk_tolerance'].lower() == "moderate":
            if portfolio['asset_allocation']['equities'] > 70:
                insights.append("- Your stock allocation might be too high for your moderate risk tolerance. Consider rebalancing.")
            elif portfolio['asset_allocation']['equities'] < 40:
                insights.append("- Your stock allocation might be too low to meet your long-term goals. Consider increasing equity exposure.")

        report.append("\n".join(insights))

        report.append("""
    Important Disclosures
    -------------------
    - Past performance does not guarantee future results
    - This report is for informational purposes only
    - Please consult your financial advisor before making investment decisions
    """)
        final_report = "\n".join(report)
        return final_report

    def main():
        print("Welcome to the Investment Report Generator!")
        print("Please enter the following information to generate a report.")

        try:
            client_data = get_client_input()
```

```
            report = create_report(client_data)
            print("\nGenerating report...\n")
            print(report)

        except ValueError as e:
            print("\nError: Please enter valid numerical values for portfolio data.")
        except Exception as e:
            print(f"\nAn error occurred: {str(e)}")

        print("\nThank you for using the Investment Report Generator!")

    if __name__ == "__main__":
        main()
```

```
Welcome to the Investment Report Generator!
Please enter the following information to generate a report.

=== Client Profile Input ===
Enter client name: Boopal
Enter investment goals (e.g., Long-term growth): Long term growth
Enter risk tolerance (Conservative/Moderate/Aggressive): Moderate

=== Portfolio Data Input ===
Enter year-to-date returns (%): 8
Enter annualized returns (%): 7
Enter equity allocation (%): 60
Enter bond allocation (%): 30
Enter cash allocation (%): 10
Enter Sharpe ratio: 1.5
Enter standard deviation (%): 10.5

=== Benchmark Data Input ===
Enter S&P 500 return (%): 8.0
Enter Bond Index return (%): 1.5

Generating report...


Investment Portfolio Report
==========================
Client: Boopal
Date: December 20, 2024


Portfolio Overview
----------------
Investment Goals: Long term growth
Risk Tolerance: Moderate

Performance:
- Year to Date Return: 8.0%
- Annualized Return: 7.0%

Current Asset Allocation:
- Stocks: 60.0%
- Bonds: 30.0%
- Cash: 10.0%


Risk and Benchmark Analysis
-------------------------
Risk Metrics:
- Sharpe Ratio: 1.5
- Volatility (Standard Deviation): 10.5%

Benchmark Comparison:
- S&P 500: 8.0%
- Bond Index: 1.5%


Investment Insights
-----------------
```