```
import requests
from prettytable import PrettyTable
import json

# API Configuration
API_KEY = "your_alphavantage_api_key"
BASE_URL = "https://www.alphavantage.co/query"

# Portfolio Data
portfolio = {}

# Fetch stock price from Alpha Vantage API
def fetch_stock_price(symbol):
    try:
        url = f"{BASE_URL}?function=GLOBAL_QUOTE&symbol={symbol}&apikey={API_KEY}"
        response = requests.get(url)
        data = response.json()

        if "Global Quote" in data:
            stock_price = float(data["Global Quote"]["05. price"])
            return stock_price
        else:
            print(f"Error: Unable to fetch data for {symbol}. Check the stock symbol or API key.")
            return None
    except Exception as e:
        print(f"An error occurred: {e}")
        return None

# Add a stock to the portfolio
def add_stock(symbol, quantity, purchase_price):
    if symbol in portfolio:
        portfolio[symbol]["quantity"] += quantity
        portfolio[symbol]["purchase_price"] = purchase_price
    else:
        portfolio[symbol] = {"quantity": quantity, "purchase_price": purchase_price}

# Display the portfolio
def display_portfolio():
    table = PrettyTable(["Stock", "Quantity", "Purchase Price", "Current Price", "Gain/Loss (%)",
"Value"])
    total_value = 0

    for symbol, data in portfolio.items():
        current_price = fetch_stock_price(symbol)
        if current_price:
            quantity = data["quantity"]
            purchase_price = data["purchase_price"]
            current_value = current_price * quantity
            gain_loss = ((current_price - purchase_price) / purchase_price) * 100
            total_value += current_value

            table.add_row([
                symbol,
                quantity,
                f"${purchase_price:.2f}",
                f"${current_price:.2f}",
                f"{gain_loss:.2f}%",
                f"${current_value:.2f}"
            ])
    print(table)
```

```
    print(f"Total Portfolio Value: ${total_value:.2f}")

# Save portfolio to file
def save_portfolio(filename="portfolio.json"):
    with open(filename, "w") as file:
        json.dump(portfolio, file)
    print(f"Portfolio saved to {filename}")

# Load portfolio from file
def load_portfolio(filename="portfolio.json"):
    global portfolio
    try:
        with open(filename, "r") as file:
            portfolio = json.load(file)
        print(f"Portfolio loaded from {filename}")
    except FileNotFoundError:
        print(f"File {filename} not found. Starting with an empty portfolio.")

# Main Menu
def main():
    load_portfolio()

    while True:
        print("\nStock Portfolio Tracker")
        print("1. Add Stock")
        print("2. View Portfolio")
        print("3. Save Portfolio")
        print("4. Exit")
        choice = input("Enter your choice: ")

        if choice == "1":
            symbol = input("Enter stock symbol (e.g., AAPL): ").upper()
            quantity = int(input("Enter quantity: "))
            purchase_price = float(input("Enter purchase price: "))
            add_stock(symbol, quantity, purchase_price)
        elif choice == "2":
            display_portfolio()
        elif choice == "3":
            save_portfolio()
        elif choice == "4":
            save_portfolio()
            print("Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.")

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