**Requirements:**

1. **Model Development**
   * Use **TensorFlow 2.14.0** to build a **Transformer-based model** suitable for time series data prediction.
   * The input features are **OHLCV (Open, High, Low, Close, Volume)** from 15-minute time frames.
   * The output is the **predicted return** for the next candle.
2. **Data Preparation**
   * Collect and preprocess ES futures data using **Yahoo Finance (yfinance)** API.
   * Handle missing data appropriately (e.g., weekends or holidays without trading).
   * Implement sliding windows for sequence generation to convert time series into input features for the model.
   * Normalize the input features (OHLCV) using **StandardScaler**.
3. **Model Architecture**
   * Design a Transformer model with key components such as **multi-head attention**, **feedforward layers**, and **residual connections**.
   * Experiment with various hyperparameters like:
     + **Number of layers**
     + **Embedding dimensions**
     + **Attention heads**
     + **Feedforward dimension (FFN)**
   * Ensure the model structure supports sequential time series processing.
4. **Training & Optimization**
   * Train the model using a **Mean Squared Error (MSE)** loss function.
   * Utilize the **Adam optimizer** with learning rate scheduling.
   * Use **early stopping** to prevent overfitting and ensure efficient training.
5. **Backtesting**
   * Implement backtesting using the **Vectorbt** library to evaluate the model's predicted signals in a real market context.
   * Run the backtest on ES futures data, focusing on:
     + **Entry and exit signals** based on the model's predictions.
     + Generate performance metrics such as **equity curve**, **trade distribution**, and **sensitivity analysis**.
6. **Deliverables**
   * Provide the trained model in **Keras SavedModel** format.
   * Present a **comprehensive backtest report**, including:
     + Performance metrics
     + Equity curve
     + Trade analysis

**Key Evaluation Criteria:**

* Model's ability to predict ES futures returns accurately.
* Quality and readability of code (following **PEP 8 standards**).
* Thoroughness of backtesting and analysis.