Project Initialization and Planning Phase in Crude Oil Price Prediction

The **Project Initialization and Planning Phase** lays the foundation for the **Crude Oil Price Prediction** project. It involves defining objectives, identifying requirements, and planning tasks to ensure the project is executed efficiently and achieves its goals.

**1. Problem Statement**

The volatile nature of crude oil prices poses a challenge for industries and investors relying on oil for business or financial decisions. The goal is to leverage machine learning techniques to forecast short-term crude oil prices using historical data, thereby enabling informed decision-making and reducing uncertainty.

**2. Objectives**

1. **Primary Objective**:
   * Develop a machine learning model to predict crude oil prices for the next day.
2. **Secondary Objectives**:
   * Understand and analyze historical trends in crude oil prices.
   * Create a web-based application to provide predictions to end users.

**3. Scope of the Project**

* **Data Collection**:
  + Obtain reliable historical crude oil price data.
* **Data Preprocessing**:
  + Handle missing values, normalize data, and convert it into a suitable format for time-series forecasting.
* **Model Building**:
  + Design and train an LSTM neural network for accurate prediction.
* **Application Development**:
  + Deploy a user-friendly web application using Flask for real-time predictions.
* **Evaluation**:
  + Assess the model's performance using metrics like MSE and RMSE.

**4. Key Deliverables**

1. A trained LSTM model capable of predicting crude oil prices.
2. A functional web application for users to input historical prices and receive predictions.
3. Comprehensive documentation of the methodology, implementation, and results.

**5. Requirements Analysis**

**5.1 Data Requirements**

* **Dataset**: Historical crude oil price data (e.g., daily closing prices).
* **Source**: Trusted repositories like Kaggle or government/industry data portals.
* **Features**:
  + Date
  + Closing Price

**5.2 Technical Requirements**

* **Hardware**:
  + A system with sufficient processing power (CPU focus, as user prefers CPU-based TensorFlow.
* **Software**:
  + Python (with libraries like TensorFlow, Flask, Pandas, NumPy.
  + IDEs or development environments like Jupyter Notebook or PyCharm.

**5.3 Human Resources**

* Data analysts for preprocessing and visualization.
* Developers for model building and application development.

**6. Planning and Timeline**

**6.1 Milestones & developer**

1. **Data Preparation** - developed by - aryaman singh
   * Timeframe: Week 1
   * Tasks: Data collection, cleaning, and preprocessing
2. **Model Development** - developed by - viraj yadav
   * Timeframe: Week 2–3
   * Tasks: Build, train, and tune the LSTM model.
3. **Application Development** - developed by - Ansh Jaiswar
   * Timeframe: Week 4
   * Tasks: Develop and test the Flask web application.
4. **Testing and Deployment** - developed by - Vinit Pawar
   * Timeframe: Week 5
   * Tasks: Test the complete system, deploy locally, and refine based on feedback.

**6.2 Tools for Project Management**

* **Task Management**: Tools like Trello, Jira, or Microsoft Excel for tracking progress.
* **Version Control**: GitHub for maintaining and collaborating on code.

**7. Risks and Challenges**

1. **Data Challenges**:
   * Missing or inconsistent historical price data.
   * Limited external features affecting model accuracy.
2. **Technical Challenges**:
   * Ensuring the LSTM model trains efficiently on a CPU.
   * Handling potential overfitting or underfitting.
3. **Deployment Challenges**:
   * Ensuring the web application runs seamlessly for end users.
   * Addressing scalability for future cloud deployment.

**8. Success Criteria**

1. The model achieves acceptable accuracy, with RMSE below a predefined threshold.
2. The web application provides fast and user-friendly predictions.
3. The project is completed within the planned timeline and meets the stated objectives.