Combined Project Report – Phases 1, 2 & 3

Phase 1 – Problem Identification & Data Acquisition

Problem Statement

Build an AI-based solution to predict stock prices using time series analysis, enabling short-term and long-term forecasting, trend detection, and model interpretability.

Data Collection

- **Sources Used**:

  - Yahoo Finance via `yfinance` (for live, real-time stock data)

  - Kaggle datasets:

    - Nifty 50 Stock Market Data

    - Financial Analysis Stocks Dataset

    - Nifty 100 5-min Interval Dataset

Data Overview

- Multiple tickers, daily and intraday data

- Data attributes include Open, High, Low, Close, Volume

- Combined data covered Indian and global stocks

Challenges

- Varying formats and symbols

- Missing values and inconsistent column names

- Need to unify static and live formats

Phase 2 – Preprocessing, Feature Engineering, and Storage

Cleaning & Standardization

- Removed nulls and fixed column inconsistencies
- Standardized time indexes and fixed timezone errors
- Renamed confusing columns like "Unnamed: 4" to "Close"

Feature Engineering

- Added RSI, EMA, MACD indicators to static datasets
- Calculated rolling averages and volatility zones

Dataset Preparation

- Saved all enriched files into `data/processed/enriched/`
- Scaled closing prices using MinMaxScaler
- Generated 60-step sequences for training

Output

- Stored processed model-ready datasets as `.npz` files
- File naming organized by ticker and source
- Separate storage for static, live, and global categories

Phase 3 – Model Building, Evaluation, and Deployment

Models Used

- **Prophet** (additive time series forecasting)
- **LSTM** (deep learning for sequence data)
- **GRU** (lightweight deep learning alternative)

Evaluation Strategy

- RMSE calculated between predicted and actual prices

- Evaluation and comparison visualized as line plots

- Plots saved under `/plots/`, models under `/models/`

Deployment

- Streamlit app with dropdowns for stock & model

- Real-time prediction from trained models

- CSV download for Prophet forecasts

- UI served from `/streamlit_app/app.py`

Features in Deployment

- User chooses stock and model

- Model runs on preprocessed `.npz` or `.csv`

- Outputs chart and RMSE, downloadable data

Conclusion

All 3 phases were executed in a modular, scalable manner—covering a complete AI pipeline: from acquisition, enrichment, modeling, and deployment. The app can now be scaled and extended to new data or integrated into financial tools.