Excellent—let’s formally establish a **research plan** for your FYP on **FTSE 100 Short-Term Forecasting**. Below is a comprehensive structure that includes your evaluation standards, model selection strategy, and focus priorities.

**📌 FTSE 100 Forecasting – Final Year Project (FYP) Plan**

**🎯 1. Project Objective**

Develop and evaluate a system that predicts **short-term movements** in the FTSE 100 index using both **traditional time-series models** and **advanced deep learning techniques**, ultimately integrating the best-performing model into a web-based dashboard.

**⚙️ 2. Evaluation Standards (Metrics)**

Your project will be evaluated on both **accuracy** and **practical usability**. Use the following metrics:

**📈 Forecast Accuracy**

| **Metric** | **Purpose** |
| --- | --- |
| **RMSE (Root Mean Squared Error)** | Penalizes large errors more heavily; good for magnitude accuracy |
| **MAPE (Mean Absolute Percentage Error)** | Intuitive % error measure; useful for business interpretation |
| **MAE (Mean Absolute Error)** | Robust to outliers, useful for model comparison |
| **R² (Coefficient of Determination)** | Measures how well predictions capture variance in data |
| **Directional Accuracy (DA)** | Measures how often model correctly predicts the **direction** (up/down) |

**🧪 Robustness / Generalization**

| **Technique** | **Use** |
| --- | --- |
| **Expanding Window Cross-Validation** | Mimics real-time forecasting; prevents lookahead bias |
| **Rolling Predictions Evaluation** | Track RMSE/MAPE over time to detect drift or overfitting |

**🧠 3. Model Strategy**

**⚖️ Baseline Models (Benchmark Level)**

These models are simple, interpretable, and serve as reference points:

| **Model** | **Why It’s Chosen** |
| --- | --- |
| **ARIMA** | Standard for univariate time series; captures seasonality & trend |
| **Prophet (Facebook)** | Useful for time series with trend + holidays; handles missing data gracefully |
| **Simple Moving Average (SMA)** | Naive benchmark; easy to beat |
| **Linear Regression (Lag Features)** | For early feature-to-target experimentation |

**🚀 Advanced Techniques (Core Focus)**

| **Model** | **Why It’s Chosen** |
| --- | --- |
| **CNN-LSTM** | Captures short-term temporal + spatial dependencies in indicators |
| **XGBoost** | Ensemble tree model; robust and usually high-performing on tabular time series data |
| **Hybrid (e.g., LSTM + XGBoost Ensemble)** | Leverages strengths of both sequence models and gradient boosting |
| *(Optional)*: **Transformer or TCN** | For exploratory/bonus work if time permits |

**🛠️ 4. Technical Components to Focus On**

**📊 Data Engineering**

* Time series resampling (if needed)
* Technical indicators: SMA, EMA, MACD, RSI, Bollinger Bands
* Lagged features, rolling stats

**🧪 Model Training & Validation**

* GridSearch / RandomSearch for tuning
* Walk-forward (expanding window) validation
* Error trend tracking over time

**🧠 Model Interpretability**

* SHAP (for XGBoost feature importance)
* Visualize model attention (if Transformer or attention-based LSTM is explored)

**💻 Web Deployment**

* **Flask** backend: Load model and serve predictions
* **React** or **Streamlit** frontend: Input window, charts, metrics
* Deployment options: Heroku / Streamlit Cloud / local

**🗂️ 5. Suggested Project Workflow (Phases)**

| **Phase** | **Description** |
| --- | --- |
| **Week 1–2** | Finalize literature review and metrics; complete EDA and feature engineering |
| **Week 3–5** | Baseline modeling (ARIMA, Prophet, Regression) + evaluation |
| **Week 6–8** | Build & train CNN-LSTM and XGBoost models |
| **Week 9** | Ensemble modeling and alignment testing |
| **Week 10** | Full evaluation on expanding window + results analysis |
| **Week 11–12** | Web app development and integration |
| **Week 13–14** | Write final report, test app, polish everything |
| **Week 15** | Final submission and presentation prep |

**✅ What You Should Focus On Next**

1. **Finalize Baseline Results**:
   * ARIMA, SMA, and/or Prophet → Done or repeat with current cleaned data
2. **Train and Compare Advanced Models**:
   * Re-run **CNN-LSTM** and **XGBoost** using **Expanding Window**
   * Focus on error stability and direction accuracy
3. **Select Best Model** (or ensemble) based on:
   * RMSE, MAPE, DA consistency over time
   * Deployment feasibility
4. **Begin App Design**:
   * Define data I/O and chart types
   * Decide whether React or Streamlit is better for your tech stack

Would you like a **printable or markdown version** of this research plan for your report or supervisor submission?  
And do you want help **finalizing your baseline evaluation notebook next**?