SMC Forex Predictor - Project Documentation

Prepared by: Nikodimos Elias G/Egziabher

Date: July 30, 2025

Junior Software Developer

# Table of Contents

1. Project Overview

2. File & Folder Structure

3. Concept: Smart Money Concept (SMC)

4. Backend Workflow

5. Technical Terms

6. Frontend Overview

7. Frontend Structure & Styling

8. Additional Notes

# 1. Project Overview

The SMC Forex Predictor is a full-stack application integrating machine learning, Flask-based API backend, and a modern React frontend with Tailwind CSS. The system is designed to analyze Forex market data using Smart Money Concepts (SMC). It predicts market trends such as BUY, SELL, or NEUTRAL with visual feedback, allowing users to make informed decisions based on trends and confidence scores.

# 2. File & Folder Structure

Each file and folder serves a specific role in the backend and frontend logic:

|  |  |
| --- | --- |
| File/Folder (Backend) | Description |
| app.py | Main Flask application for serving frontend/backend logic. |
| bot.py | Telegram bot interface to accept user inputs and return predictions. |
| features/create\_features.py | Handles feature engineering and extraction from uploaded CSVs. |
| model/model.joblib | Serialized trained ML model used for making predictions. |
| model/label\_encoder.joblib | Encodes/decodes class labels for prediction interpretation. |
| uploads/ | Stores user-uploaded CSV files temporarily. |
| utils/ | Houses additional utility functions (if any). |

Frontend File Structure Highlights:

|  |  |
| --- | --- |
| File/Folder (Frontend) | Description |
| src/pages/Home.jsx | Displays all available currency pairs with beautiful grouping and animations. |
| src/pages/Predict.jsx | Handles prediction result rendering, graph visualization, and background logic. |
| src/components/CurrencyCard.jsx | Reusable card component for displaying currency names and click action. |
| src/components/PredictionChart.jsx | Chart component for plotting entry, stop-loss, and take-profit points. |
| src/components/NotFound404.jsx | Beautiful 404 error page with background and a 'Go Home' button. |
| src/css/\*.css | Modular CSS files for styling pages and components. |
| src/services/api.js | API utility functions using Axios for fetching and posting data to backend. |

# 3. Concept: Smart Money Concept (SMC)

Smart Money Concept (SMC) is a trading approach that follows the actions of institutional investors (smart money) to identify market trends and turning points. The concept uses price action, liquidity sweeps, market structure shifts, and order blocks to determine potential buy/sell signals.

# 4. Backend Workflow

1. User selects a currency file from preloaded data (no uploads).  
2. Flask API processes the selected file via `extract\_features` function.  
3. Features are passed into a trained machine learning model.  
4. The model outputs trend prediction, confidence, stop-loss, and take-profit.  
5. Telegram bot and React frontend display results to the user.

# 5. Technical Terms

- CSV: Comma-separated value file containing time-series market data.  
  
- SMA: Simple Moving Average, a trend indicator.  
  
- SMC Trend: Custom trend logic based on SMC trading strategy.  
  
- Label Encoder: Maps string labels (BUY, SELL) to integers for model training.  
  
- Joblib: Library for saving and loading machine learning models.  
  
- API: Application Programming Interface, used to connect frontend with backend.

# 6. Frontend Overview

The frontend is developed using React with Vite for fast build performance. Styling is done using Tailwind CSS. Animations are implemented via Framer Motion. Axios is used to make HTTP requests to the Flask backend.

# 7. Frontend Structure & Styling

- Home Page: Displays currency pairs in groups (A-Z), includes a search bar, and a visually engaging background.  
- Predict Page: Displays prediction result, confidence scores, entry/TP/SL on a graph, and adapts background based on trend.  
- 404 Page: Custom 404 component with themed styling and a back-to-home button.  
- CSS: Modular files for each major component/page, improving maintainability.

# 8. Additional Notes

This system can be extended with:  
  
- Real-time data fetching from brokers or APIs (e.g., Binance, OANDA).  
  
- Integration of historical charting and pattern recognition.  
  
- Multi-timeframe analysis.  
  
- Authentication for private usage.  
  
- Export predictions to CSV or PDF.