**DEVELOPER GUIDE**

**STOCKS PULSE**

**From Anomalies to Forecasts**

**1. Introduction**

The comprehensive technical documentation for **StocksPulse**, a system for detecting and forecasting stock price anomalies, is available in this Developer Guide. To assist developers with maintaining, expanding, or deploying the application, it provides an explanation of the project's architecture, setup guidelines, dependencies, and essential modules.

**2. System Requirements**

* Programming Language: Python 3.9+
* Frameworks: Streamlit, scikit-learn, Prophet, Plotly
* Other Tools: yFinance, TA (Technical Analysis library), BeautifulSoup, Twilio API
* OS Compatibility: Windows, macOS, Linux

**3. Installation & Setup**

**Clone the repository:**

git clone https://github.com/yourusername/stockspulse.git

**Create and activate a virtual environment:**

python -m venv venv

source venv/bin/activate # For Linux/Mac

venv\Scripts\activate # For Windows

**Install dependencies:**

pip install -r requirements.txt

**Add your secrets for email and Twilio in .streamlit/secrets.toml:**

sender\_email="your\_email@gmail.com"

sender\_pass="your\_password"

twilio\_sid="your\_twilio\_sid"

twilio\_auth="your\_twilio\_auth\_token"

twilio\_number="whatsapp:+14155238886"

### 4. Project Structure

stockspulse/

│── app.py # Main application file

│── requirements.txt # Dependencies

│── /data # Sample datasets

│── /assets # Logo and banner images

│── /docs # User & Developer Guides

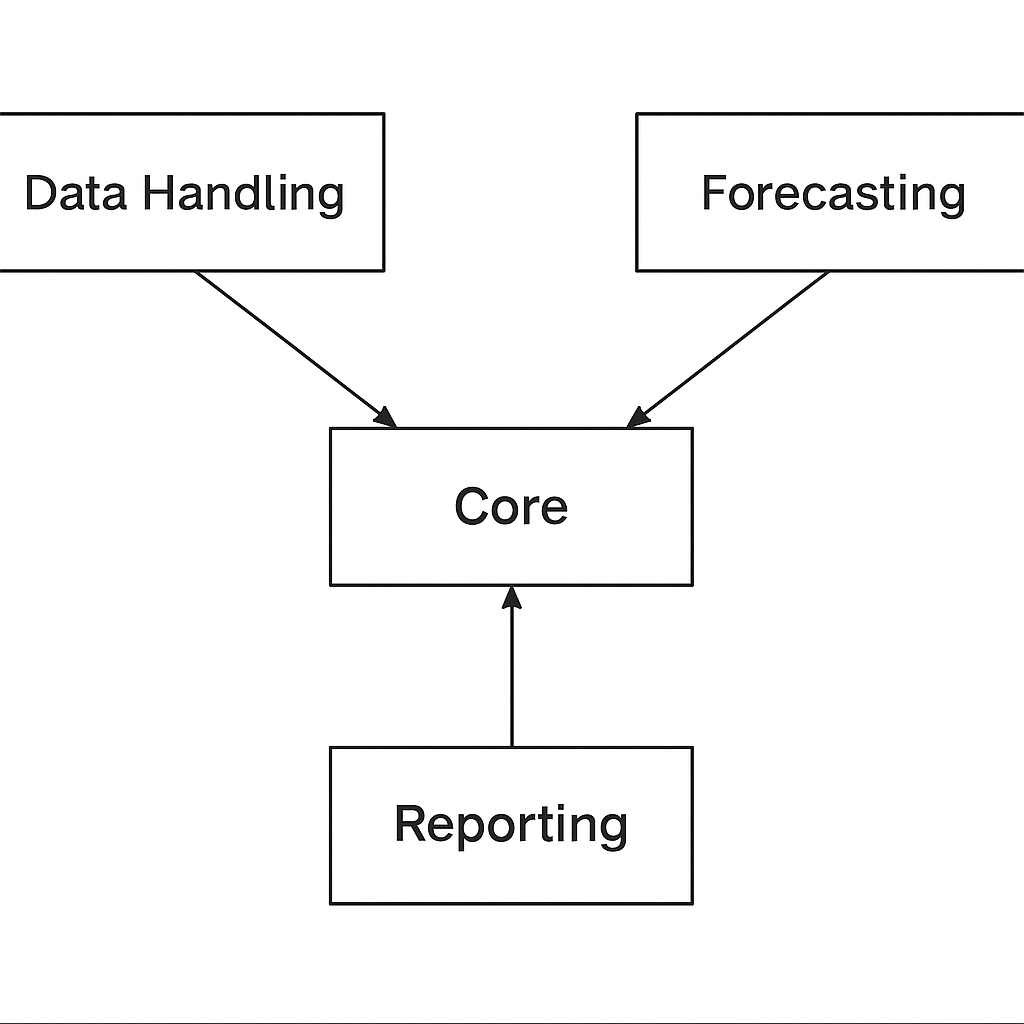
│── /notebooks # Experimental Jupyter notebooks

│── /models # Saved ML models (optional)

│── .streamlit/secrets.toml # Credentials and secrets

**5. Fundamental Modules**

1. Data Management  
     
   fetch\_ticker\_df: Uses yFinance to download stock data.  
     
   basic\_preprocess: Sorts, interpolates, and cleans up missing values.  
     
   feature\_engineer: Incorporates features such as volatility, moving averages, and RSI.
2. Finding Anomalies  
     
   supports the Local Outlier Factor, One-Class SVM, and Isolation Forest.  
     
   utilizes StandardScaler to normalize features.  
     
   uses Plotly to provide visualization (heatmap, candlestick, and line chart).
3. Making predictions  
     
   Prophet-powered with adjustable seasonality.  
     
   offers evaluation metrics, projections, and confidence ranges.  
     
   CSV download option for export.
4. Dashboard for the Portfolio  
     
   manages several tickers.  
     
   shows current anomalies, short forecasts, and price history.
5. Integration of News  
     
   retrieves RSS feeds for Google News.  
     
   displays designed news cards and parses using BeautifulSoup.
6. Reporting  
   Forecasts and anomalies are exported to CSV and PDF files.  
     
   Reports can be sent by WhatsApp (Twilio API) or email (SMTP).



1. **Continuing the Project**

* Include fresh models for anomaly detection (autoencoders, LSTMs, etc.).
* ARIMA or hybrid models can be used to improve forecasting.
* Utilize interactive dashboards to improve reporting.
* Link to APIs for real-time stock data.

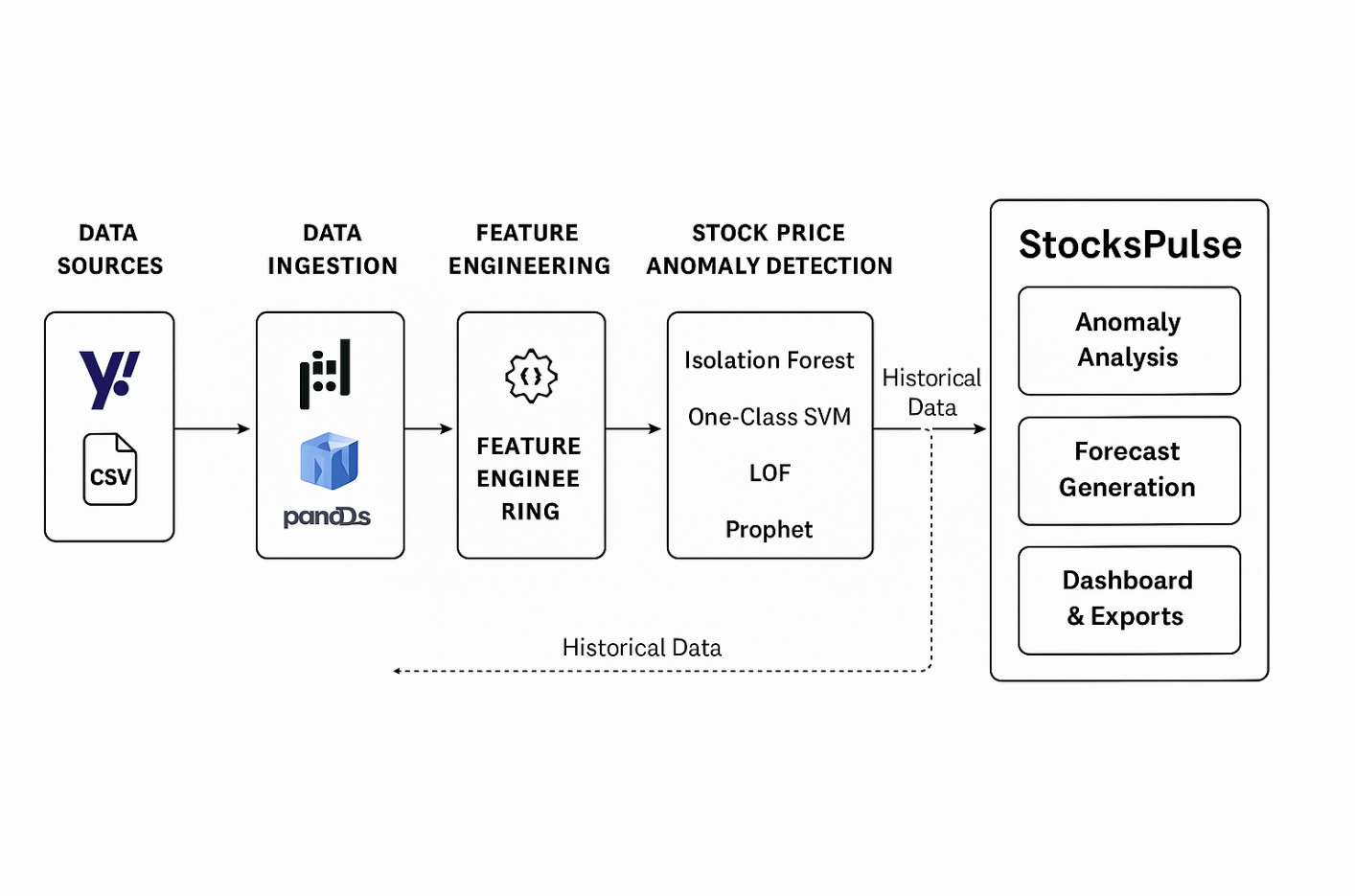
1. **Troubleshooting**

* BeautifulSoup parser problem -> install lxml.
* Problems installing Prophet: install Pystan first.
* Twilio number problem ← Verify that WhatsApp has the sandbox number enabled.
* Streamlit cache problems → use streamlit cache clean to clear the cache

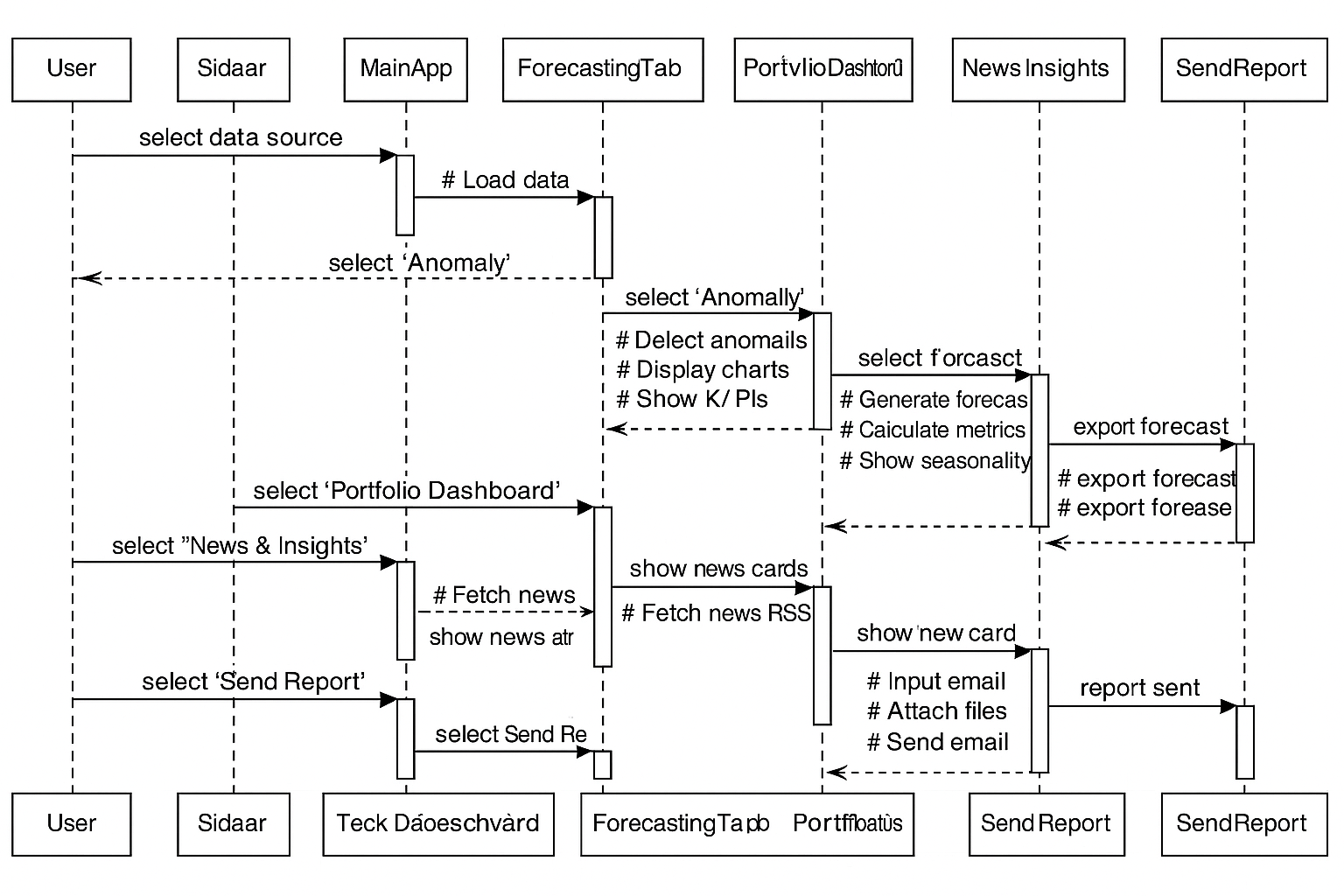
1. **System Architecture**

****

1. **Data Flow/Pipeline**

****

1. **Sequence Flow**

****