Building a comprehensive AI trading system like this involves multiple complex components working together. Here's how to approach this ambitious project:

**System Architecture Overview**

Your AI needs several interconnected modules:

* **News Processing Engine** (NLP for sentiment and event extraction)
* **Market Data Pipeline** (real-time and historical price/volume data)
* **Behavioral Prediction Models** (psychology-based investor behavior)
* **Stock Screening Filter** (your custom criteria implementation)
* **Decision Fusion System** (combining all inputs for recommendations)
* **Risk Management Layer** (position sizing, stop losses, portfolio balance)

**Phase 1: Foundation and Data Infrastructure**

**Data Collection Systems** Start by building robust data pipelines for news feeds (Reuters, Bloomberg APIs, or web scraping financial news sites), market data (Yahoo Finance, Alpha Vantage, or paid services like Quandl), and historical datasets for backtesting. You'll need to handle real-time data streams and store everything in a structured database.

**News Processing Pipeline** Use natural language processing to extract sentiment, identify key events (earnings, mergers, regulatory changes), and categorize news by relevance and impact. Libraries like NLTK, spaCy, or transformer models can help analyze text for market-moving information.

**Phase 2: Core AI Models**

**Behavioral Psychology Models** This is perhaps the most challenging aspect. You'll need to model concepts like herd mentality, fear/greed cycles, overreaction to news, and pattern recognition in investor behavior. Consider using reinforcement learning or neural networks trained on historical market reactions to similar events.

**Predictive Models** Build separate models for different time horizons (short-term sentiment-driven moves vs long-term fundamental trends). Consider using ensemble methods that combine multiple approaches like LSTM networks for time series, random forests for feature-based predictions, and transformer models for processing complex news narratives.

**Phase 3: Integration and Decision Making**

**Stock Filtering Engine** Implement your custom criteria as a scoring system that can be easily modified. This might include financial ratios, technical indicators, sector classifications, or any other metrics important to your strategy.

**Decision Fusion Algorithm** Create a system that weighs inputs from news sentiment, predicted investor behavior, technical analysis, and your filtering criteria. This could use weighted scoring, machine learning meta-models, or rule-based expert systems.

**Technical Implementation Strategy**

**Start Simple, Scale Gradually** Begin with a basic version focusing on one asset class or sector. Use existing APIs and libraries rather than building everything from scratch initially. Python with pandas, scikit-learn, and TensorFlow/PyTorch provides a solid foundation.

**Backtesting Framework** Build comprehensive backtesting capabilities to validate your models against historical data. This is crucial for understanding performance and avoiding overfitting to recent market conditions.

**Paper Trading Phase** Before risking real money, run your system in simulation mode for several months to identify bugs, edge cases, and performance characteristics across different market conditions.

**Critical Considerations**

**Market Efficiency Challenges** Remember that markets are highly efficient, and many sophisticated institutions are trying to do similar things with massive resources. Your edge will likely come from unique data sources, novel approaches to behavioral modeling, or specialized domain knowledge.

**Risk Management** Build in strict risk controls including position limits, correlation checks, and circuit breakers. Many promising trading systems have failed due to inadequate risk management rather than poor predictions.

**Regulatory Compliance** Ensure your system complies with relevant financial regulations, especially if you plan to manage significant capital or provide advice to others.

This is a multi-year project that combines elements of quantitative finance, machine learning, behavioral psychology, and software engineering. Start with a minimal viable version focusing on one specific use case, then expand capabilities over time as you gain experience and validate your approaches.