### *Unique Solution & Action Plan: GenAI Financial Advisor with Stock Prediction (Free Tools/APIs)*

*Project Name: **StockSenseAI*

A *MERN-based platform* combining *Generative AI* for personalized financial advice and *free ML tools/APIs* for historical stock analysis and recommendations.

### *Core Features*

1. *Conversational AI Interface*

   - Users chat with an AI to discuss risk tolerance, goals, and financial queries.

   - *Example*: "Should I invest in tech stocks?" → AI analyzes market trends + user profile → Recommends ETFs or individual stocks.

2. *Historical Stock Prediction*

   - Analyze past stock performance (e.g., 5-year trends) using ML models.

   - *Example*: Predict future trends for Tesla (TSLA) based on historical data.

3. *Personalized Recommendations*

   - Suggest portfolios based on risk appetite (e.g., "Aggressive," "Conservative").

4. *Real-Time Market Data Dashboard*

   - Display stock charts, news, and alerts using free APIs.

### *Tech Stack (100% Free Tools)*

Component	*Tools/APIs*
Frontend	React.js, Chart.js (charts), Material-UI (UI components)
Backend	Node.js, Express.js, Socket.io (real-time chat)
Database	MongoDB Atlas (free tier), Mongoose
AI/ML	- *Chat*: Hugging Face Transformers (free LLMs like GPT-2/Flan-T5)
	- *Stock Prediction*: TensorFlow.js (LSTM model) or scikit-learn (Python)
APIs	- *Stock Data*: Alpha Vantage (free API), Yahoo Finance API
	- *News/Sentiment*: NewsAPI.org (free tier), Reddit API (sentiment analysis)
Deployment	Netlify (frontend), Render/Heroku (backend), GitHub Actions (CI/CD)

### *Action Plan (30 Days)*

Team Roles:

1. *Frontend Dev*: React UI, charts, and chat interface.

2. *Backend Dev*: Node.js API, MongoDB integration.

3. *AI/ML Engineer*: Train stock prediction model + integrate Hugging Face.

4. *DevOps*: Deployment, API integrations, and testing.

#### *Phase 1: Setup & Core Features (Days 1-10)*

1. *Day 1-3*:
   - Set up MERN project + GitHub repo.
   - Create basic UI with:
     - Chat interface (React + Socket.io).
     - Dashboard layout (Chart.js for stock trends).
   - Free API integration: Fetch stock data (Alpha Vantage).

2. *Day 4-7*:
   - Build backend API for:
     - User authentication (JWT + MongoDB).
     - Storing chat history and user preferences.
   - Integrate Hugging Face’s *Flan-T5* (free LLM) for chat responses.

3. *Day 8-10*:
   - Add real-time stock data to the dashboard.
   - Train a *simple LSTM model* (TensorFlow.js) on historical stock data (e.g., S&P 500).

#### *Phase 2: AI/ML Integration (Days 11-20)*

1. *Day 11-14*:
   - *Stock Prediction Model*:

  - Use Python (Google Colab) to train an LSTM model on free datasets (Kaggle/Nasdaq).
  - Export the model to TensorFlow.js for browser/Node.js inference.
- Example Prediction Workflow:

  python

```
# Sample LSTM code (Python)
model = Sequential()
model.add(LSTM(50, return_sequences=True, input_shape=(60, 1)))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
```

2. *Day 15-18*:
   - *Recommendation Engine*:
     - Use *collaborative filtering* (scikit-learn) to suggest stocks based on similar users.
     - Example: If User A (risk-tolerant) likes TSLA and AMZN, recommend NVDA to User B.

3. *Day 19-20*:
   - Integrate sentiment analysis using *Reddit API* (r/stocks, r/investing).
   - Example: Alert users if negative sentiment spikes for a stock.

#### *Phase 3: Testing & Deployment (Days 21-30)*

1. *Day 21-24*:

  - Stress-test APIs (cache data with MongoDB to avoid rate limits).
  - Optimize ML model performance (quantize TensorFlow.js model for speed).

2. *Day 25-27*:
  - Deploy:
    - Frontend → Netlify.
    - Backend → Render (free tier).
    - Python ML model → Hugging Face Spaces (free hosting).

3. *Day 28-30*:
  - Add user tutorials and disclaimers (e.g., "Not financial advice").
  - Final testing + gather beta user feedback.

### *Risks & Mitigation*
- *Risk*: Free API rate limits (e.g., Alpha Vantage allows 5 requests/minute).
  - *Fix*: Cache data in MongoDB and use client-side polling.
- *Risk*: LLM inaccuracies.
  - *Fix*: Use predefined templates for critical advice (e.g., "Diversify your portfolio").

### *Outcome*

A scalable, AI-driven platform where users can:

1. Chat with a financial AI assistant.

2. Analyze historical stock trends.

3. Get free, data-backed investment recommendations.

No-code alternative: Use Voiceflow + Airtable for prototyping, but MERN offers full customization.

##DEEPseek

Here’s a comprehensive list of *free APIs* you can use for your project, along with their specific use cases:

### *1. Stock Data APIs*

| *API* | *Use Case* | *Free Tier Details*
|

Alpha Vantage	Real-time and historical stock data, technical indicators, and forex rates.	Free tier: 5 API requests per minute, 500 requests/day.
Yahoo Finance API	Stock prices, historical data, and company financials.	Free and open-source (via yfinance Python library).
Finnhub	Real-time stock prices, news, and financial statements.	Free tier: 60 API calls/minute, limited historical data.
IEX Cloud	Stock quotes, historical data, and company information.	Free tier: 50,000 messages/month, limited features.
Tiingo	End-of-day stock prices, crypto data, and forex rates.	Free tier: 500 API requests/day, limited historical data.

### *2. News & Sentiment Analysis APIs*

API	*Use Case*	*Free Tier Details*
NewsAPI	Fetch financial news articles for market sentiment analysis.	Free tier: 100 API requests/day, limited sources.
Reddit API	Scrape Reddit (e.g., r/stocks, r/investing) for sentiment analysis.	Free and open-source (requires OAuth for authentication).
GDELT Project	Global news sentiment and trends.	Free and open-source (real-time news data).
Aylien Text Analysis	Sentiment analysis of news articles and social media.	Free tier: 1,000 API requests/day.

### *3. AI/ML APIs*

API	*Use Case*	*Free Tier Details*
Hugging Face	Pre-trained LLMs (e.g., Flan-T5, GPT-2) for conversational AI.	Free tier: Access to many open-source models.
OpenAI API	GPT-3.5/4 for advanced conversational AI (paid but affordable).	Free trial: $18 credit for new users.
TensorFlow.js	Run ML models (e.g., LSTM for stock prediction) in the browser.	Free and open-source.
FastAPI	Host ML models (e.g., LSTM) as APIs for backend integration.	Free and open-source.

### *4. Financial APIs*

API	*Use Case*	*Free Tier Details*
**FRED (Federal Reserve)	Economic data (e.g., interest rates, inflation) for macro analysis.	Free and open-source.
World Trading Data	Stock market data, ETFs, and mutual funds.	Free tier: 250 API requests/day, limited features.
CoinGecko	Cryptocurrency prices, market caps, and trends.	Free tier: No rate limits for basic data.
Financial Modeling Prep	Financial statements, ratios, and stock metrics.	Free tier: 250 API requests/day, limited features.

### *5. Visualization & Charting APIs*

API	*Use Case*	*Free Tier Details*
Chart.js	Interactive stock charts for the frontend.	Free and open-source.
Plotly	Advanced financial charts (e.g., candlestick charts).	Free and open-source.
Highcharts	Professional-grade stock charts.	Free for non-commercial use.

### *6. Deployment & Monitoring APIs*

API	*Use Case*	*Free Tier Details*
Netlify	Deploy frontend (React) with CI/CD.	Free tier: 100GB bandwidth/month, 300 build minutes/month.
Render	Deploy backend (Node.js) and databases.	Free tier: Unlimited static sites, limited backend services.
Heroku	Host backend and ML models.	Free tier: Limited dyno hours, sleep after 30 mins of inactivity.
GitHub Actions	Automate CI/CD pipelines.	Free for public repositories.

### *7. Miscellaneous APIs*

API	*Use Case*	*Free Tier Details*
OpenWeatherMap	Economic impact analysis (e.g., weather on agriculture stocks).	Free tier: 1,000 API requests/day.
Google Trends API	Analyze search trends for stocks or financial terms.	Free and open-source (unofficial Python libraries available).
Twilio	Send SMS alerts for stock price changes or recommendations.	Free trial: $15 credit for testing.

### *How to Use These APIs Together*

1. *Stock Data: Use **Alpha Vantage* or *Yahoo Finance* for real-time and historical stock data.

2. *Sentiment Analysis: Combine **NewsAPI* and *Reddit API* to gauge market sentiment.

3. *AI Chat: Use **Hugging Face* or *OpenAI* for conversational AI.

4. *Stock Prediction: Train an LSTM model using **TensorFlow.js* or *scikit-learn* and host it with *FastAPI*.

5. *Visualization: Use **Chart.js* or *Plotly* for interactive stock charts.

6. *Deployment: Host frontend on **Netlify, backend on **Render, and ML models on **Hugging Face Spaces*.

This list ensures you can build a *fully functional platform* using *100% free tools and APIs*!