REQUIREMENTS OF TECHNOLOGY

Back End

a) **Programming languages:**

- Python (for AI/ML)
- Node.js
- Java

b) **Frameworks:**

- Flask / Django
- Express.js
- Spring Boot

Front End (Cross-platform development)

Mobile:

- React Native + React.js *(most efficient for cross-platform)*
- Flutter + Firebase *(for cross-platform)*

Web:

- Native development (Swift, Kotlin)
- React / Angular / Vue.js

State Management:

- Redux, MobX, Context API

UI Frameworks:

- Material Design, TailwindCSS

Databases

- PostgreSQL / MySQL *(for relational data)*
- MongoDB / Cassandra *(for portfolio data)*
- Redis *(for caching & sessions)*

Cloud Infrastructure:

- AWS, Google Cloud, or Azure

Data Sources & Integration (Market Data Sources)

→ **Real-time stock market data:**

- Alpha Vantage
- Polygon.io
- IEX Cloud
- Yahoo Finance API
- Bloomberg API

Financial News:

- News APIs (NewsAPI, Bloomberg, Reuters)
- Social media feeds & Reddit

Company Fundamentals (Finance Dept):

a) SEC filings data (with help of Tech team)
b) Company financial statements
c) Analyst reports

Integration Methods

- RESTful APIs for data acquisition
- Websockets for real-time updates
- Batch processing for historical data
- ETL pipelines for data transformation

Data Storage & Processing

- Databases for raw financial data
- Data warehouses for processed insights
- Time-series databases for market data
- Caching layer for frequently accessed data

AI & ML Implementation

Key AI Components

Predictive Analytics Models

1. Price Prediction **(LSTM, ARIMA models)**
2. Volatility Forecasting
3. Market Trend Identification

Portfolio Optimization

1. Modern Portfolio Theory implementation
2. Monte Carlo Simulations
3. Genetic Algorithms for allocation

Sentiment Analysis

1. Natural Language Processing for news
2. Social media sentiment extraction
3. Impact correlation with market movements

Personalization Engine

1. User behavior analysis
2. Risk tolerance assessment
3. Goal-based recommendation systems

Anomaly Detection

1. Fraud prevention
2. Market anomaly identification
3. Portfolio risk alerts

ML Training Pipeline

- Data collection & preprocessing
- Feature engineering specific to financial data
- Model training & validation
- Model deployment & monitoring
- Continuous retraining with new data

Data Analysis & Insights:

- Analyze large sets of financial, market, and customer data.
- Identify trends, patterns, and hidden opportunities (like predicting market movements or user behavior).

Model Development:

- Build, test, and refine AI/ML models for things like(Core Features)
    - Portfolio optimization
    - Risk assessment
    - Personalized investment recommendations
    - Fraud detection
- Use statistical techniques to validate model assumptions and performance.

Algorithm Enhancement:

- Work with engineers to improve the core investment algorithms by making them smarter over time based on new data.

Predictive Analytics:

- Forecast stock prices, asset returns, or economic indicators.
- Help create "what-if" scenarios for users' portfolios.

Client Personalization:

- Develop models that understand **user risk appetite, financial goals, and life stages** — to offer hyper-personalized investment strategies.

Data Quality & Governance:

- Ensure the data feeding the AI systems is **accurate, clean, and compliant with regulations.**

Performance Monitoring:

- Set up dashboards and tracking systems to measure **how well investment strategies are working over time.**

Platform-Specific Development

Android Development

- Kotlin or Java with Android Studio
- Material Design components
- Android Jetpack libraries
- Google Play Store requirements

iOS Development

- Swift/UI on UIKit
- Apple Human Interface Guidelines
- App Store compliance

Web Development

- React.js or Angular frontend
- Responsive design with Tailwind CSS or Bootstrap
- Progressive Web app capabilities
- Browser compatibility testing

Cross-Platform Approach

1. Shared business logic with JS API calls
2. Platform specific UI components
3. Platform specific optimizations
4. Feature parity across devices

Security & Compliance

Security Measures

i) End to end encryption for sensitive data
ii) Multi-factor authentication
iii) Biometric authentication options
iv) SSL/TLS for all communications
v) Regular security audits & penetration testing

Financial Regulations

i) KYC/AML compliance implementation
ii) SEC/FINRA requirements (US) GDPR compliance (Europe)
iii) Local financial regulations
iv) Open Banking Standards (where applicable)

Data Privacy

i) Transparent data usage policies
ii) User consent management
iii) Data minimization practices
iv) Right to be forgotten implementation

Development & Testing Strategies

→ Automated Testing

i) Unit testing for all components
ii) Integration testing for API connectivity
iii) UI automation testing
iv) Performance testing under load

→ Financial Accuracy Testing

i) Portfolio calculation verification
ii) Tax calculation validation
iii) Historical backtracking of recommendations
iv) Risk Assessment Accuracy

→ User Testing

i) Beta testing program
ii) Usability testing sessions
iii) A/B testing for features
iv) Focus Groups for feedback

User Experience Design:

Create an intuitive Interface Design

1. Visualizes complex financial data simply
2. Provides actionable insights
3. Features progressive Onboarding
4. Offers customizable dashboards
5. Includes educational resources

Deployment & Launch

→ Infrastructure Setup

i) CI/CD pipeline configuration
ii) Container orchestration with Kubernetes
iii) Auto-scaling policies
iv) Disaster recovery planning

→ Monitoring & Analytics

i) Application performance monitoring

ii) User behaviour analytics

iii) Error tracking & alerting

iv) Conversion & retention metrics

→ Launch Strategy

i) Phased rollout (Beta, soft launch, full launch)

ii) Platform specific launch considerations

iii) Marketing & PR co-ordination (marketing team)

iv) Support team preparation

Post-Launch Optimization

→ Performance Optimization

i) API response time improvements
ii) App startup time reduction
iii) Battery usage optimization (For mobile)
iv) Memory footprint reduction

→ Iterative Improvement

i) Feature prioritization based on user feedback

ii) A/B testing for UX refinements

iii) ML model accuracy improvements

iv) Regular Algorithm updates

→ Scaling Strategy

i) Horizontal scaling for growing user base

ii) Geographic expansion considerations

iii) Additional financial product integration

iv) Advanced feature roadmap

Payment Gateway

- **Core Payment Gateway Requirements**

i) Payment Processing capabilities

a) Fund deposits (bank transfers, cards, digital wallets)

b) Withdrawals to user accounts

c) Subscription payments for premium features

d) Investment transactions

e) Fee collection

ii) Security & Compliance considerations

Implementation Options

1. **Third-Party Payment Processors Advantage**:

   a) Faster implementation

   b) Established security infrastructure

   c) Handles compliance requirements

   d) Reduced liability Potential Providers: **a) Stripe b) PayPal c) Plaid d) Adyen e) Dwolla (specialized in ACH transfers)**

2. **Custom Payment Gateway Advantage:**

   a) Complete control over user experience

   b) Potentially lower transaction fees at scale

   c) Customized for investment-specific needs Consider:

   i) Requires significant security expertise

   ii) Lengthy compliance certification process

   iii) Higher development & maintenance costs

Steps:

a) Requirements Gathering

i) Define transaction types & workflows

ii) Map user payment journeys

iii) Determine regulatory requirements in target markets

b) Design

i) Payment Processing flow

ii) Encryption Strategy

iii) Database Schema for transaction records

iv) Reconciliation systems

c) Development Approach

• Create Secure API endpoints

• Implement encryption for sensitive data

• Build transaction monitoring systems

• Develop automated reconciliation

d) Testing & Security

• Penetration testing

• Transaction flow verification

• Edge case handling (failed payments, partial refunds)

• Load testing for high volume periods

e) Integration with App Components

• Connect to user accounts system

• Link to portfolio management system

• Integrate with AI recommendation engine

Specific Platform Consideration

Android	iOS	Web
i) Use Secure storage for payment tokens	i) Leverage Apple's Secure enclave	i) Implement tokenization for card details
ii) Implement biometric authentication when available	ii) Consider Apple Pay integration	ii) Use 3D Secure for additional protection
iii) Follow Google Play billing policies for in-app purchases	iii) Adhere to App Store guidelines for financial apps	iii) Ensure responsive design works across devices

Key Technical Considerations

API Security

→ OAuth 2.0 for authorization

→ HTTPS for all communications

→ Rate limiting to prevent attacks

→ Input validation to prevent injection attacks

Reconciliation & Monitoring

→ Real-time transaction monitoring

→ Automated reconciliation processes

→ Fraud detection systems

→ Reporting dashboard for financial oversight

User Experience

→ Minimize friction in payment flows

→ Clear transaction receipts & history

→ Transparent fee disclosures

→ Intuitive deposit/withdrawal processes