

Aivestor: MVP and Workplan Document

May 21, 2025

1 Introduction

Aivestor is a personal investment advisor platform designed to empower users with AI-driven portfolio management, real-time nudges, and brokerage integration. This document outlines the Minimum Viable Product (MVP) features and a phased workplan to develop Aivestor, tailored to the current development stage as of May 21, 2025.

2 Minimum Viable Product (MVP)

The MVP focuses on core functionality to validate the concept, targeting individual investors. Key features include:

- **User Dashboard:** A simple interface displaying portfolio overview, asset allocation, and performance metrics.
- **AI Recommendations:** Basic investment suggestions based on user risk tolerance and market data.
- **Real-Time Nudges:** Automated alerts for price changes or portfolio rebalancing opportunities.
- **Brokerage Integration:** Initial connection to a mock brokerage for portfolio updates.
- **Authentication:** Secure login with email and password.

The MVP prioritizes usability and core AI functionality, with enhancements planned in subsequent phases.

3 Workplan

The development is divided into phases, each with specific tasks and deliverables. The current progress is through Phase 1, with Phase 2 underway.

3.1 Phase 1: Project Setup

- **Task 1:** Install development tools (Node.js, Python, VS Code, Git).
- **Task 2:** Create Next.js frontend with Turbopack and Tailwind CSS.
- **Task 3:** Set up Express backend with a test endpoint.
- **Task 4:** Develop Flask AI service with a basic endpoint.
- **Status:** Completed (frontend at localhost:3000, backend at localhost:5000, AI at localhost:5001).

3.2 Phase 2: Database and Backend Integration

- **Task 1:** Set up PostgreSQL database (local installation completed).
- **Task 2:** Connect backend to PostgreSQL (in progress).
- **Task 3:** Implement CRUD operations for user and portfolio data.
- **Status:** Task 1 completed, Task 2 underway.

3.3 Phase 3: Authentication and User Management

- **Task 1:** Integrate Firebase Authentication.
- **Task 2:** Create user registration and login endpoints.
- **Task 3:** Secure API routes with JWT.
- **Status:** Not started.

3.4 Phase 4: AI Model Development

- **Task 1:** Develop a simple AI model for recommendations.
- **Task 2:** Integrate AI with backend API.
- **Task 3:** Test AI accuracy with mock data.
- **Status:** Not started.

3.5 Phase 5: Frontend Development

- **Task 1:** Build dashboard UI with portfolio and nudge displays.
- **Task 2:** Integrate frontend with backend APIs.
- **Task 3:** Add real-time updates with WebSockets.
- **Status:** Not started.

3.6 Phase 6: Brokerage Integration

- **Task 1:** Set up mock brokerage API.
- **Task 2:** Sync portfolio data with brokerage.
- **Task 3:** Implement trade execution simulation.
- **Status:** Not started.

3.7 Phase 7: Testing and Optimization

- **Task 1:** Unit and integration testing.
- **Task 2:** Performance optimization.
- **Task 3:** User acceptance testing.
- **Status:** Not started.

3.8 Phase 8: Deployment

- **Task 1:** Deploy frontend to Vercel.
- **Task 2:** Deploy backend and AI to a cloud server (e.g., Heroku).
- **Task 3:** Configure CI/CD pipeline.
- **Status:** Not started.

4 Conclusion

This workplan provides a structured approach to building Aivestor's MVP, starting with foundational setup and progressing to advanced features. Current efforts focus on connecting the backend to PostgreSQL, with subsequent phases enhancing functionality and deployment.