Ball Skill - Player Stock Value & Earnings Potential System

Player Stock Portfolio Interface (Robinhood Style)

1. Player Stock Value Model

```javascript
```

```
// Each player becomes a trackable "stock" with performance-based valuation
const PLAYER_STOCK_SYSTEM = {
  stock_valuation_factors: {
    current_rating: { weight: 0.30, calculation: 'rating / 10' },
    win_rate: { weight: 0.25, calculation: 'win_percentage * 100' },
    earnings_velocity: { weight: 0.20, calculation: 'monthly_earnings / 10' },
    activity_level: { weight: 0.15, calculation: 'games_per_week * 5' },
    social_influence: { weight: 0.10, calculation: 'followers + engagement_score' }
  },

  // Stock price calculation
  calculateStockPrice: function(player) {
    let baseValue = 100; // Starting stock price

    // Rating component (1000 rating = $100 stock price)
    const ratingValue = (player.rating / 10);

    // Performance multiplier (70% win rate = 1.4x multiplier)
    const performanceMultiplier = Math.max(0.5, Math.min(2.0, player.win_rate * 2));

    // Earnings momentum (monthly earnings drive stock appreciation)
    const earningsMultiplier = Math.max(0.8, Math.min(1.5,
      1 + (player.monthly_earnings / 1000)
    ));

    // Activity bonus (active players = higher stock price)
    const activityBonus = Math.min(1.3, 1 + (player.weekly_games / 50));

    return Math.round(
      (baseValue + ratingValue) * performanceMultiplier * earningsMultiplier * activityBonus
    );
  },

  // Stock performance tracking
  stock_metrics: {
    current_price: 'calculated_real_time',
    daily_change: 'percentage_and_dollar_amount',
    weekly_performance: '7_day_price_chart',
    monthly_performance: '30_day_price_chart',
    all_time_high: 'highest_stock_price_ever',
    all_time_low: 'lowest_stock_price_ever',

    // Portfolio-style metrics
    market_cap: 'stock_price * total_followers',
    price_to_earnings: 'stock_price / monthly_earnings',
    volatility: 'price_change_standard_deviation',
    beta: 'correlation_to_platform_average'
  }
};

// Robinhood-style portfolio interface
const PORTFOLIO_DASHBOARD = {
  header_display: {
    total_portfolio_value: '$1,247.83',
    daily_change: '+$23.45 (+1.92%)',
    color_coding: 'green_for_gains_red_for_losses',

    // Key metrics prominently displayed
    key_stats: {
      earning_potential: '$2,340/year',
      win_rate: '73.2%',
      current_streak: '5 wins',
      stock_price: '$124.78 (+$2.34 today)'
    }
  },

  // Interactive charts like Robinhood
  performance_charts: {
    timeframes: ['1D', '1W', '1M', '3M', '1Y', 'ALL'],
    chart_types: ['earnings', 'stock_price', 'win_rate', 'activity'],

    // Real-time updates
    live_updates: true,
    price_alerts: 'notify_on_significant_changes',
    milestone_markers: 'show_achievements_on_timeline'
  }
};
```

2. Earnings Potential Projections

javascript

```javascript
// YouTube Analytics style earnings projections
const EARNINGS_PROJECTION_ENGINE = {
  // Run rate calculations based on current performance
  calculateRunRate: function(player, timeframe = 'yearly') {
    const recent_performance = this.getRecentPerformance(player, 30); // Last 30 days

    const metrics = {
      avg_daily_earnings: recent_performance.total_earnings / 30,
      avg_games_per_day: recent_performance.total_games / 30,
      avg_win_rate: recent_performance.wins / recent_performance.total_games,
      avg_earnings_per_win: recent_performance.total_earnings / recent_performance.wins
    };

    // Project forward based on timeframe
    const projections = {
      daily: metrics.avg_daily_earnings,
      weekly: metrics.avg_daily_earnings * 7,
      monthly: metrics.avg_daily_earnings * 30,
      yearly: metrics.avg_daily_earnings * 365,

      // Scenario-based projections
      if_maintains_current_pace: {
        next_30_days: metrics.avg_daily_earnings * 30,
        next_90_days: metrics.avg_daily_earnings * 90,
        next_365_days: metrics.avg_daily_earnings * 365
      },

      if_improves_win_rate: {
        target_win_rate: 0.80, // 80% win rate goal
        projected_earnings_boost: this.calculateWinRateImpact(player, 0.80),
        timeframe_to_goal: this.estimateImprovementTime(player)
      }
    };

    return projections;
  },

  // Scenario-based earnings messages
  motivational_projections: [
    {
      condition: 'win_next_10_of_20_games',
      message: 'Win 10 of your next 20 matchups to earn $245.67 this month',
      probability: 'based_on_current_win_rate',
      timeline: '15 days at your current pace'
    },
    {
      condition: 'maintain_current_streak',
      message: 'Keep your 5-game streak alive to unlock $50 bonus reward',
      probability: '73% chance based on your performance',
      timeline: '2-3 games based on your activity level'
    },
    {
      condition: 'daily_goal_achievement',
      message: 'Complete 3 games today to earn $45 and stay on track for $1,350 monthly goal',
      probability: 'achievable with normal activity',
      timeline: 'by midnight tonight'
    },
    {
      condition: 'performance_milestone',
      message: 'Hit 80% accuracy in next game to unlock Elite tier earning potential (+40% prize multiplier)',
      probability: '67% chance based on recent form',
      timeline: 'next game'
    }
  ]
};
```

3. Social Streak & Competition System

javascript

```
// Duolingo-style streak and badge system with social elements
const SOCIAL_STREAK_SYSTEM = {
  daily_streaks: {
    personal_practice: {
      description: 'Practice shots every day',
      current_streak: 23,
      longest_streak: 45,
      streak_rewards: {
        7_days: 'Bronze Practice Badge + $5 credit',
        30_days: 'Silver Dedication Badge + $25 credit',
        100_days: 'Gold Consistency Badge + $100 credit',
        365_days: 'Diamond Year Badge + $500 credit + Custom Badge Design'
      }
    },

    social_battles: {
      description: 'Daily matchups with friends',
      participants: ['User A', 'User B', 'User C'],
      group_streak: 30,
      individual_contributions: {
        current_user: 28,  // Participated 28 of 30 days
        user_a: 30,       // Perfect attendance
        user_b: 25        // Missed 5 days
      },

      streak_rewards: {
        group_milestone: 'All members get bonus when group hits milestones',
        individual_consistency: 'Personal rewards for streak maintenance',
        comeback_bonus: 'Extra rewards for rejoining broken streaks'
      }
    },

    rival_streaks: {
      description: 'Head-to-head daily battles with specific opponents',
      active_rivalries: [
        {
          opponent: 'JordanSlayer23',
          consecutive_days: 47,
          current_leader: 'current_user',
          daily_stakes: '$5 per game',
          streak_bonus: '$10 for every 7-day streak',
          rivalry_level: 'LEGENDARY' // Based on streak length
        }
```

```javascript
      ]
    }
  },

  // Friend engagement mechanics
  friend_challenges: {
    streak_protection: {
      feature: 'Friends can "save" your streak if you miss a day',
      cost: '$2 to save friend streak',
      benefit: 'Builds social bonds and platform engagement',
      limits: 'Maximum 3 saves per month per friendship'
    },

    group_challenges: {
      weekly_team_goals: 'Combined team performance targets',
      month_long_competitions: 'Teams compete for prizes',
      seasonal_tournaments: 'Friend groups compete in brackets'
    },

    social_pressure: {
      streak_notifications: 'Friends see when you miss days',
      leaderboards: 'Friend group performance rankings',
      achievement_sharing: 'Auto-share milestone achievements'
    }
  }
};
```

4. Record Breaking & First Achievement System

javascript

```
// First-to-achieve and record-breaking reward system
const RECORD_BREAKING_SYSTEM = {
  platform_firsts: {
    // One-time achievements for platform firsts
    first_perfect_game: {
      achievement: 'First player to score 10/10 shots',
      reward: '$100 + Legendary Badge + Profile Highlight',
      claimant: null, // Available to be claimed
      timestamp: null
    },

    first_speed_demon: {
      achievement: 'First to complete tournament in under 30 seconds',
      reward: '$50 + Speed Demon Badge + Custom Animation',
      claimant: null,
      verification_required: 'video_proof_with_timestamp'
    },

    first_win_streak_10: {
      achievement: 'First 10-game win streak',
      reward: '$75 + Streak Master Badge + News Feature',
      claimant: null,
      difficulty_scaling: 'rewards_increase_with_streak_length'
    }
  },

  daily_firsts: {
    // Reset daily - competition for first achievements each day
    first_game_of_day: {
      reward: '$5 + Early Bird Badge',
      resets: 'daily_at_midnight_user_timezone',
      competition: 'first_player_to_complete_game_wins'
    },

    first_perfect_score_daily: {
      reward: '$15 + Daily Perfect Badge',
      requirement: '100%_accuracy_in_any_game_mode',
      resets: 'daily'
    },

    first_tournament_completion: {
      reward: '$25 + Tournament Pioneer Badge',
      requirement: 'complete_any_tournament_first_each_day',
      resets: 'daily'
    }
  },

  ongoing_records: {
    // Continuously tracked records with leaderboards
    highest_single_game_earnings: {
      current_record: '$156.78',
      record_holder: 'ProShooter_Mike',
      date_set: '2025-08-15',
      verification: 'blockchain_verified',
      reward_for_breaking: '10%_of_new_record_amount + Record Breaker Badge'
    },

    longest_active_win_streak: {
      current_record: 23,
      record_holder: 'StreakKing_2025',
      live_tracking: true,
      pressure_factor: 'everyone_watches_when_approaching_record',
      reward_for_breaking: '$500 + Streak Legend Status'
    },

    fastest_tournament_completion: {
      current_record: '00:00:47',
      record_holder: 'SpeedBaller',
      game_mode: 'standard_3PT_tournament',
      verification: 'video_required + anti_cheat_validation',
      reward_for_breaking: '$200 + Speed Record Badge'
    }
  },

  // Milestone achievement system
  milestone_rewards: {
    games_played: {
      100: '$10 + Century Player Badge',
      500: '$50 + Veteran Badge',
      1000: '$150 + Legend Badge',
      5000: '$500 + Hall of Fame Badge'
    },

    total_earnings: {
      100: '$10 bonus + Earner Badge',
      1000: '$100 bonus + Profit Master Badge',
      5000: '$500 bonus + Elite Earner Badge',
      10000: '$1000 bonus + Millionaire Club Badge'
    },

    accuracy_milestones: {
      first_90_percent_game: '$25 + Precision Badge',
      maintain_80_percent_over_100_games: '$100 + Consistency Master',
      perfect_month: '$250 + Perfect Player Badge'
    }
  }
};
```

5. Event Suggestion Engine

javascript

```
// AI-powered event suggestions based on performance and earning potential
const SMART_EVENT_SUGGESTIONS = {
  performance_based_matching: {
    algorithm: 'analyze_player_performance_patterns',

    suggestions: {
      optimal_earning_events: {
        analysis: 'Events where player has highest win probability × prize pool',
        example: 'Based on your 85% win rate in 3PT events, enter the $200 3PT Championship (87% win proba
        expected_value: 'calculated_ev_with_confidence_interval'
      },

      skill_improvement_events: {
        analysis: 'Events that target player weaknesses for growth',
        example: 'Your free throw percentage is 67% - enter FT Challenge to improve and earn $45',
        development_value: 'skill_improvement_potential + earning_opportunity'
      },

      streak_protection_events: {
        analysis: 'Lower-risk events to maintain winning streaks',
        example: 'Protect your 7-game streak with this 95% win probability event ($20 entry)',
        risk_assessment: 'calculated_risk_vs_streak_value'
      }
    }
  },

  // Personalized event recommendations
  recommendation_engine: {
    time_based: {
      peak_performance_hours: 'suggest_events_during_player_best_performance_times',
      quick_games: 'suggest_short_events_when_player_has_limited_time',
      marathon_sessions: 'suggest_tournament_series_when_player_has_extended_time'
    },

    social_based: {
      friend_events: 'suggest_events_where_friends_are_participating',
      rival_challenges: 'suggest_head_to_head_with_established_rivals',
      group_tournaments: 'suggest_team_events_for_friend_groups'
    },

    financial_based: {
      budget_optimization: 'suggest_events_that_maximize_expected_earnings_per_dollar_spent',
      bankroll_management: 'suggest_appropriate_stakes_based_on_available_balance',
            investment_opportunities: 'suggest_higher_stakes_events_when_player_is_hot'
        }
    }
};
```

Technical Implementation

1. Real-time Stock Price Database

```
sql
```

```sql
-- Player stock price tracking (updates every game completion)
CREATE TABLE player_stock_prices (
  id SERIAL PRIMARY KEY,
  user_id INTEGER REFERENCES users(id),

  -- Stock metrics
  current_stock_price DECIMAL(8,2),
  daily_change_amount DECIMAL(8,2),
  daily_change_percentage DECIMAL(5,2),

  -- Performance factors affecting stock price
  current_rating INTEGER,
  win_rate DECIMAL(5,2),
  monthly_earnings DECIMAL(10,2),
  weekly_activity_score INTEGER,
  social_influence_score INTEGER,

  -- Historical tracking
  all_time_high DECIMAL(8,2),
  all_time_low DECIMAL(8,2),

  -- Market metrics
  market_cap DECIMAL(12,2),     -- stock_price * followers
  price_to_earnings DECIMAL(8,2), -- stock_price / monthly_earnings

  updated_at TIMESTAMP DEFAULT NOW(),
  price_date DATE DEFAULT CURRENT_DATE,

  UNIQUE(user_id, price_date)
);

-- Historical price tracking for charts
CREATE TABLE stock_price_history (
  id SERIAL PRIMARY KEY,
  user_id INTEGER REFERENCES users(id),
  stock_price DECIMAL(8,2),
  timestamp TIMESTAMP DEFAULT NOW()
);

-- Index for fast chart queries
CREATE INDEX idx_stock_history_user_time ON stock_price_history(user_id, timestamp);
```

2. Streak Tracking Database

sql

```sql
-- Comprehensive streak tracking
CREATE TABLE user_streaks (
  id SERIAL PRIMARY KEY,
  user_id INTEGER REFERENCES users(id),
  streak_type VARCHAR(30), -- 'daily_practice', 'win_streak', 'social_battle', 'rival_challenge'

  -- Streak metrics
  current_streak INTEGER DEFAULT 0,
  longest_streak INTEGER DEFAULT 0,
  streak_start_date DATE,
  last_activity_date DATE,

  -- Social streak details (if applicable)
  partner_user_id INTEGER REFERENCES users(id),
  group_streak_id VARCHAR(50),

  -- Rewards and milestones
  total_rewards_earned DECIMAL(10,2) DEFAULT 0,
  next_milestone INTEGER,
  next_milestone_reward VARCHAR(100),

  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW(),

  UNIQUE(user_id, streak_type, partner_user_id)
);

-- Record tracking
CREATE TABLE platform_records (
  id SERIAL PRIMARY KEY,
  record_type VARCHAR(50),
  record_value DECIMAL(10,4),
  record_holder_id INTEGER REFERENCES users(id),

  -- Verification
  verified BOOLEAN DEFAULT FALSE,
  verification_method VARCHAR(30), -- 'video_proof', 'blockchain', 'live_witness'
  verification_data JSONB,

  -- Context
  game_mode VARCHAR(20),
  event_id INTEGER REFERENCES events(id),
    record_date TIMESTAMP DEFAULT NOW(),
    previous_record_value DECIMAL(10,4),
    previous_record_holder INTEGER REFERENCES users(id)
);
```

3. Earnings Projection Engine

javascript

```
// Real-time earnings calculation and projection
class EarningsProjectionEngine {
  constructor(userId) {
    this.userId = userId;
    this.performanceHistory = this.loadPerformanceHistory(30); // Last 30 days
  }

  // Calculate current earnings velocity
  calculateEarningsVelocity() {
    const recentEarnings = this.performanceHistory.total_earnings;
    const dayspan = this.performanceHistory.dayspan;

    return {
      daily_rate: recentEarnings / dayspan,
      weekly_rate: (recentEarnings / dayspan) * 7,
      monthly_rate: (recentEarnings / dayspan) * 30,
      annual_projection: (recentEarnings / dayspan) * 365
    };
  }

  // Generate motivational earning projections
  generateMotivationalProjections() {
    const velocity = this.calculateEarningsVelocity();
    const performance = this.analyzePerformancePatterns();

    return {
      // Short-term achievable goals
      today_potential: this.calculateDayEarningsPotential(),
      week_potential: this.calculateWeekEarningsPotential(),
      month_potential: this.calculateMonthEarningsPotential(),

      // Scenario-based projections
      if_win_next_10: this.calculateScenarioEarnings('win_10_of_20'),
      if_maintain_streak: this.calculateStreakValue(),
      if_improve_accuracy: this.calculateAccuracyImpact(0.05), // 5% improvement

      // Milestone-based goals
      next_earnings_milestone: this.getNextEarningsMilestone(),
      time_to_milestone: this.estimateTimeToMilestone()
    };
  }
}
```

This system transforms basketball performance into an engaging financial game while building addictive social mechanics. Users become "stock traders" of their own athletic performance with clear earning projections and social competition elements.