AlphaMarket Connect - Product & Technical Documentation

AlphaMarket Connect - Product & Technical Documentation

Version: 1.0 **Last Updated:** February 2026 **Status:** Live in Production

---

Table of Contents

---

1. Product Overview

AlphaMarket Connect is a SaaS marketplace platform designed to connect SEBI-registered Indian financial advisors (RA/RIA) with investors and brokers. The platform enables:

- Advisors to create and manage investment strategies across various segments (Equity, F&O, Commodity, Baskets)
- Publishing of actionable investment calls and positions with real-time P&L tracking
- Investors to browse, subscribe, and receive recommendations
- Full regulatory compliance including eKYC verification, e-Sign agreements, risk profiling, and performance disclosure with SEBI-mandated disclaimers
- Admin oversight with advisor approval workflows

Live URL: Deployed via Replit's publishing system under a `.replit.app` domain.

---

2. User Roles & Access Control

The platform has three distinct user roles with role-based access control (RBAC):

2.1 Admin

- Approve/reject advisor registrations
- Manage all users (CRUD operations)
- Manage all strategies
- Configure Groww API tokens for market data
- Send broadcast push notifications
- View platform-wide settings

2.2 Advisor (SEBI-Registered RA/RIA)

- Register with SEBI registration number and certificate
- Create and manage investment strategies
- Publish calls (stock recommendations) and positions (F&O/multi-leg trades)
- Create subscription plans with pricing
- View subscribers and revenue analytics
- Manage content (Market Outlook, Learn articles)
- View customer eKYC details, e-Sign agreements, and risk profiles
- Download compliance reports (CSV)
- Answer investor questions
- Toggle risk profiling requirement per advisor

2.3 Investor

- Browse the public marketplace of strategies
- View strategy details, calls, and positions
- Subscribe to strategies (with e-Sign agreement, payment, eKYC, and optional risk profiling)
- Receive push notifications for subscribed strategies
- Maintain a personal watchlist
- View recommendations from subscribed strategies
- Ask advisors questions

---

3. Technology Stack

3.1 Languages & Runtime

Component	Technology	Version
Language	TypeScript	5.6.3
Runtime	Node.js	20.x
Package Manager	npm	Latest

3.2 Frontend

Library	Purpose	Version
React	UI Framework	18.3.x
Vite	Build Tool & Dev Server	7.3.x

Library	Purpose	Version
Tailwind CSS	Utility-first CSS	3.4.x
shadcn/ui (Radix)	Component Library	Latest
wouter	Client-side Routing	3.3.x
TanStack React Query	Data Fetching & Caching	5.60.x
react-hook-form	Form Management	7.55.x
Recharts	Charts & Data Visualization	2.15.x
Framer Motion	Animations	11.13.x
lucide-react	Icon Library	0.453.x
react-icons	Brand/Company Logos	5.4.x
date-fns	Date Utilities	3.6.x
Embla Carousel	Carousel Component	8.6.x
react-helmet-async	SEO Meta Tags	2.0.x
zod	Schema Validation (shared with backend)	3.24.x

3.3 Backend

Library	Purpose	Version
Express.js	HTTP Server & API Framework	5.0.x
express-session	Session Management	1.18.x
connect-pg-simple	PostgreSQL Session Store	10.0.x
Drizzle ORM	Database ORM (type-safe)	0.39.x
drizzle-zod	Schema-to-Zod Validation Bridge	0.7.x
pg	PostgreSQL Client	8.16.x
web-push	Web Push Notifications (VAPID)	3.6.x
cashfree-pg	Cashfree Payment SDK	5.1.x
@sendgrid/mail	Email Service	8.1.x
@google-cloud/storage	Object Storage	7.19.x

3.4 Infrastructure

Service	Purpose
PostgreSQL (Neon)	Primary Database
Replit Object Storage	File Storage (SEBI certificates, uploads)
Replit Deployments	Hosting & Publishing

---

4. Project Structure

```
alphamarket-connect/
|
|-- client/                         # Frontend (React + Vite)
|   |-- public/
|   |   |-- sw.js                   # Service Worker for push notifications
|   |-- src/
|   |   |-- assets/                 # Static assets (images)
```

```
|   |   |-- components/
|   |   |   |-- ui/                 # shadcn/ui base components
|   |   |   |-- navbar.tsx          # Navigation bar (role-aware)
|   |   |   |-- footer.tsx          # Site footer
|   |   |   |-- strategy-card.tsx   # Reusable strategy card
|   |   |   |-- notification-bell.tsx # Push notification bell
|   |   |   |-- ObjectUploader.tsx  # File upload component
|   |   |-- hooks/
|   |   |   |-- use-auth.tsx        # Authentication hook
|   |   |   |-- use-toast.ts        # Toast notification hook
|   |   |   |-- use-mobile.tsx      # Mobile detection hook
|   |   |-- lib/
|   |   |   |-- queryClient.ts     # TanStack Query client setup
|   |   |   |-- utils.ts           # Utility functions
|   |   |-- pages/
|   |   |   |-- home.tsx            # Landing page
|   |   |   |-- auth.tsx            # Login page
|   |   |   |-- strategies-marketplace.tsx  # Public marketplace
|   |   |   |-- strategy-detail.tsx  # Strategy detail with performance reveal
|   |   |   |-- strategy-performance.tsx   # Detailed performance page
|   |   |   |-- subscribe.tsx       # Subscription flow
|   |   |   |-- esign-agreement.tsx  # e-Sign agreement page
|   |   |   |-- payment.tsx         # Payment page
|   |   |   |-- payment-callback.tsx # Payment result handler
|   |   |   |-- ekyc.tsx            # eKYC verification page
|   |   |   |-- risk-profiling.tsx  # Risk profiling questionnaire
|   |   |   |-- investor-dashboard.tsx # Investor dashboard
|   |   |   |-- advisors-listing.tsx # Browse advisors
|   |   |   |-- advisor-detail.tsx  # Advisor profile page
|   |   |   |-- market-outlook.tsx  # Market outlook content
|   |   |   |-- learn.tsx           # Educational content
|   |   |   |-- content-detail.tsx  # Full article view
|   |   |   |-- legal.tsx           # Legal pages
|   |   |   |-- agreements.tsx      # Agreement pages
|   |   |   |-- forgot-password.tsx  # Password reset request
|   |   |   |-- reset-password.tsx  # Password reset form
|   |   |   |-- not-found.tsx       # 404 page
|   |   |   |-- dashboard/          # Advisor Dashboard
|   |   |   |   |-- index.tsx       # Dashboard layout & routing
|   |   |   |   |-- dashboard-home.tsx    # Overview & stats
|   |   |   |   |-- strategy-management.tsx # Strategy CRUD
|   |   |   |   |-- plans.tsx       # Plan management
|   |   |   |   |-- content-page.tsx # Content management
|   |   |   |   |-- questions-page.tsx # Investor questions
|   |   |   |   |-- reports.tsx     # Compliance reports
|   |   |   |   |-- payments-page.tsx # Payment history
|   |   |   |   |-- advisor-profile.tsx # Profile & settings
|   |   |   |-- admin/              # Admin Dashboard
|   |   |   |   |-- index.tsx       # Admin layout & routing
|   |   |   |   |-- admin-advisors.tsx # Advisor approval
|   |   |   |   |-- admin-strategies.tsx # Strategy management
|   |   |   |   |-- admin-settings.tsx # Platform settings
|   |   |-- App.tsx                 # Root component with routing
|   |   |-- main.tsx                # Entry point
|   |   |-- index.css               # Global styles & theme variables
|
|-- server/                         # Backend (Express.js)
|   |-- index.ts                    # Server entry point
|   |-- routes.ts                   # All API route handlers (~2560 lines)
|   |-- storage.ts                  # Database access layer (IStorage interface)
|   |-- db.ts                       # Database connection setup
|   |-- vite.ts                     # Vite dev server integration
|   |-- cashfree.ts                 # Cashfree payment gateway helper
|   |-- email.ts                    # SendGrid email helper
|   |-- push.ts                     # Web push notification helper
```

```
|   |-- groww.ts                  # Groww API market data helper
|   |-- sandbox-kyc.ts           # Sandbox.co.in eKYC/e-Sign helper
|   |-- scheduler.ts             # Intraday auto-square-off scheduler
|   |-- seed.ts                   # Database seed data
|   |-- static.ts                 # Static file serving
|   |-- data/
|   |   |-- nse-symbols.json     # NSE/BSE/MCX symbol list for autocomplete
|
|-- shared/                       # Shared between frontend & backend
|   |-- schema.ts                 # Drizzle ORM schema + Zod types
|
|-- drizzle.config.ts            # Drizzle ORM configuration
|-- tailwind.config.ts           # Tailwind CSS configuration
|-- vite.config.ts               # Vite build configuration
|-- tsconfig.json                # TypeScript configuration
|-- package.json                 # Dependencies & scripts
|-- replit.md                    # Project documentation for AI agents
```

---

5. Database Schema

The application uses PostgreSQL with Drizzle ORM. All primary keys use UUID (`gen_random_uuid()`).

5.1 Tables Overview

Table	Purpose	Key Fields
`users`	All users (admin, advisor, investor)	id, username, email, password, role, companyName, sebiRegNumber, isApproved, requireRiskProfiling
`strategies`	Investment strategies created by advisors	id, advisorId, name, type, status, theme[], horizon, riskLevel, benchmark, planIds[]
`calls`	Stock recommendations within a strategy	id, strategyId, stockName, action, buyRange, targetPrice, stopLoss, status, entryPrice, sellPrice, gainPercent
`positions`	F&O/multi-leg positions within a strategy	id, strategyId, segment, callPut, buySell, symbol, expiry, strikePrice, entryPrice, lots, target, stopLoss, exitPrice, gainPercent
`plans`	Subscription plans created by advisors	id, advisorId, name, code, amount, durationDays
`subscriptions`	Investor subscriptions to strategies	id, planId, strategyId, userId, advisorId, status, ekycDone, riskProfiling
`payments`	Payment transactions	id, orderId, userId, strategyId, planId, amount, status, cfOrderId, paymentSessionId, subscriptionId
`content`	Articles/content published by advisors	id, advisorId, title, type, body, attachments[]

Table	Purpose	Key Fields
`scores`	Advisor compliance scorecards	id, advisorId, month, beginningOfMonth, receivedDuring, resolvedDuring
`watchlist`	Investor favorited items	id, userId, itemType, itemId
`advisor_questions`	Investor questions to advisors	id, advisorId, userId, name, email, question, answer, isRead
`risk_profiles`	Investor risk assessment results	id, subscriptionId, userId, advisorId, (40+ profile fields), capacityScore, toleranceScore, overallScore, riskCategory
`ekyc_verifications`	eKYC verification records	id, subscriptionId, userId, advisorId, verificationType, aadhaarLast4, panNumber, panStatus
`esign_agreements`	Electronic agreement records	id, userId, advisorId, strategyId, planId, subscriptionId, aadhaarLast4, aadhaarName, signedAt
`push_subscriptions`	Browser push notification subscriptions	id, userId, endpoint, p256dh, auth
`notifications`	Notification history	id, type, title, body, data, targetScope, strategyId
`password_reset_tokens`	Password reset tokens	id, userId, token, expiresAt, used
`app_settings`	Platform-wide key-value settings	key, value, updatedAt

5.2 Enums

- **role:** `advisor`, `investor`, `admin`
- **strategy_status:** `Draft`, `Published`
- **call_status:** `Active`, `Closed`
- **strategy_type:** `Equity`, `Basket`, `Future`, `Commodity`, `CommodityFuture`, `Option`

5.3 Key Relationships

```
users (1) ──> (N) strategies
strategies (1) ──> (N) calls
strategies (1) ──> (N) positions
users (advisor, 1) ──> (N) plans
plans (1) ──> (N) subscriptions
users (investor, 1) ──> (N) subscriptions
subscriptions (1) ──> (1) ekyc_verifications
subscriptions (1) ──> (1) risk_profiles
subscriptions (1) ──> (1) esign_agreements
subscriptions (1) ──> (N) payments
users (1) ──> (N) watchlist
users (1) ──> (N) push_subscriptions
```

---

6. Modules & Features

6.1 Authentication & Registration

- Session-based authentication using `express-session` with PostgreSQL session store
- Password hashing with `scrypt` (Node.js built-in crypto)
- Registration with role selection (investor by default, advisor requires approval)
- Password reset flow via email (SendGrid) with tokenized links
- Role-aware navigation and route protection

6.2 Marketplace & Strategy Discovery

- Public marketplace listing all published strategies
- Filter by strategy type (Equity, F&O, Basket, Commodity)
- Live call counts by category
- Strategy detail page with advisor info, calls, positions, plans
- Symbol autocomplete from NSE/BSE/MCX symbol database (~5000+ symbols)

6.3 Strategy Management (Advisor)

- Full CRUD for strategies with Draft/Published lifecycle
- Strategy types: Equity, Basket, Future, Commodity, CommodityFuture, Option
- Strategy metadata: theme tags, horizon, volatility, risk level, benchmark, minimum investment

6.4 Calls Management (Advisor)

- Create stock recommendations with: stock name, action (Buy/Sell), buy range, target price, stop loss, rationale
- Publish modes: Draft, Watchlist, Live
- Close calls with exit price and gain/loss calculation
- Rationale required before publishing (SEBI compliance)
- Push notifications sent to subscribers on publish/update/close

6.5 Positions Management (Advisor)

- Create F&O positions with: segment, call/put, buy/sell, symbol, expiry, strike price, entry price, lots, target, stop loss
- Support for multi-leg positions (enableLeg flag)
- Percentage-based or absolute target/stop-loss
- Close positions with exit price and P&L calculation
- Intraday auto-square-off scheduler

6.6 Live Market Data

- Real-time Last Traded Price (LTP) via Groww API integration
- Bulk price fetching for multiple symbols
- Option chain data with expiry dates
- Change indicators (price movement arrows)
- Auto-refresh functionality on strategy detail pages
- Live option premium (CE/PE LTP) for F&O P&L calculations

6.7 Subscription Flow

The complete investor subscription flow follows a strict sequence mandated by SEBI:

```
1. Select Plan ——> 2. e-Sign Agreement (Aadhaar OTP) ——> 3. Payment (Cashfree)
       |                                                          |
       v                                                          v
4. eKYC Verification (Aadhaar + PAN) ——> 5. Risk Profiling (Optional per advisor)
```

6.8 e-Sign Agreement Module

- Mandatory electronic agreement signing before payment (SEBI mandate)
- Uses Sandbox.co.in Aadhaar OTP as digital consent mechanism
- Stores aadhaarLast4, aadhaarName, signedAt
- Agreement linked to subscription after successful payment
- Email notifications sent to both investor and advisor
- Advisor can view signed agreements in “Customers Acquired” section
- **Routes:** `/strategies/:id/esign-agreement?plan=planId`
- **APIs:** `POST /api/esign/otp`, `POST /api/esign/verify`, `GET /api/esign/status`, `GET /api/advisor/agreements/:subscriptionId`

6.9 Payment Module (Cashfree)

- Cashfree Payment Gateway integration using `cashfree-pg` SDK v5
- Secure order creation with server-side amount validation
- Payment session creation and redirect to Cashfree checkout
- Payment verification via API callback
- Webhook processing for asynchronous payment confirmation
- Idempotent subscription activation (prevents duplicate processing)
- Payment history for investors and advisors
- **APIs:** `POST /api/payments/create-order`, `POST /api/payments/verify`, `POST /api/webhooks/cashfree`, `GET /api/payments/history`

6.10 eKYC Verification Module

- Mandatory Aadhaar + PAN verification after subscription payment
- Two-step flow: Aadhaar OTP verification, then PAN verification
- Uses Sandbox.co.in API with JWT token caching (22-hour expiry)
- Stores masked PII only (last 4 digits of Aadhaar, masked PAN)
- `subscription.ekycDone` flag set true only when both verifications complete
- Investor dashboard shows “eKYC Pending” banner
- Advisor can view eKYC details per subscriber
- **Helper:** `server/sandbox-kyc.ts`
- **Routes:** `/ekyc?subscriptionId=X`
- **APIs:** `POST /api/ekyc/aadhaar/otp`, `POST /api/ekyc/aadhaar/verify`, `POST /api/ekyc/pan/verify`, `GET /api/ekyc/status`, `GET /api/advisor/ekyc/:subscriptionId`

6.11 Risk Profiling Module

- Optional per-advisor feature (toggle in Profile > Settings)
- 8-section questionnaire covering personal info, financial capacity, risk tolerance, experience, and constraints
- Dual-scoring system:
    - Capacity Score (60% weight): Based on financial metrics (income, assets, liabilities, investible surplus)

    - Tolerance Score (40% weight): Based on experience, market knowledge, behavioral responses
    - Overall Score: 0-100 mapped to 5 risk categories
- Risk Categories: Conservative, Moderately Conservative, Moderate, Aggressive, Very Aggressive
- Advisor can view risk profiles per subscriber
- **Routes:** `/risk-profiling?subscriptionId=X`
- **APIs:** `POST /api/risk-profiles`, `GET /api/risk-profiles/:subscriptionId`, `GET /api/risk-profiling/check`, `PATCH /api/advisor/settings/risk-profiling`

6.12 Performance Reveal Module

- Per-strategy performance disclosure with SEBI-mandated disclaimer
- Replaces CAGR card on strategy detail page with a locked performance metric:
    - **Hit Rate** for F&O/Intraday strategies
    - **Absolute Performance** for Positional/Swing/Basket strategies
- Gated reveal flow: Click lock icon -> Read SEBI disclaimer -> Login required -> Reveal per-strategy (stored in localStorage)
- Detailed performance page at `/strategies/:id/performance` shows:
    - Period-wise breakdown table (1W, 1M, 3M, 6M, 1Y, 3Y, Max)
    - Max profitable call with gain percentage
    - Max drawdown (only from loss-making entries)
    - Win/Loss statistics
    - NIFTY benchmark comparison
- **API:** `GET /api/strategies/:id/performance` (requires authentication)

6.13 Web Push Notifications

- Browser push notifications using Web Push API with VAPID keys
- Service worker at `client/public/sw.js`
- Notification bell component in navbar with unread count
- Strategy-specific notifications sent to subscribers on:
    - New call published
    - New position added
    - Stop loss/target update
    - Call/position closed
- Admin broadcast notifications via Settings page
- **Helper:** `server/push.ts`
- **APIs:** `GET /api/notifications/vapid-key`, `POST /api/notifications/subscribe`, `DELETE /api/notifications/subscribe`, `GET /api/notifications/recent`, `POST /api/admin/notifications`

6.14 Content Management

- Advisors publish content under two categories: Market Outlook, Learn
- Rich content with multi-format attachment support (images, videos, audio, PDFs)
- Public content listing and detail pages
- Advisor dashboard for managing own content

6.15 Investor Dashboard

- Active and past subscriptions overview
- eKYC pending banner for unverified subscriptions
- Recommendations feed from subscribed strategies
- Watchlist of favorited strategies/advisors

6.16 Advisor Dashboard

- Dashboard home with stats (subscribers, revenue, strategies)
- Strategy management with call/position CRUD
- Plan management (create/delete pricing plans)
- Customers acquired section with eKYC, e-Sign, risk profile views
- Content management
- Investor questions inbox with unread count
- Compliance reports with CSV download
- Payment history
- Profile & settings (including risk profiling toggle)
- Revenue analytics

6.17 Admin Dashboard

- User management (view all, approve/reject advisors, delete users)
- Strategy oversight (view all, edit status, delete)
- Groww API token management
- Broadcast push notification sender

6.18 Communication

- “Ask a Question” form on advisor profiles for investors
- Advisor inbox with read/unread tracking and reply capability
- Email notifications via SendGrid for:
  - New user registration
  - Password reset links
  - e-Sign agreement copies (to both investor and advisor)

6.19 Reporting & Compliance

- CSV download for calls report with entry/exit details and gain/loss
- Advisor compliance scorecard management (monthly stats)

6.20 SEO

- Dynamic meta tags using react-helmet-async
- Sitemap.xml generation at `/sitemap.xml`
- Open Graph tags for social sharing

---

7. API Reference

7.1 Authentication APIs

Method	Endpoint	Auth	Description
POST	`/api/auth/register`	Public	Register new user
POST	`/api/auth/login`	Public	Login with username/email + password
GET	`/api/auth/me`	Session	Get current user
POST	`/api/auth/logout`	Session	Logout
POST	`/api/auth/forgot-password`	Public	Request password reset email
POST	`/api/auth/reset-password`	Public	Reset password with token

7.2 Public APIs

Method	Endpoint	Description
GET	`/api/advisors`	List all approved advisors
GET	`/api/advisors/:id`	Get advisor profile
GET	`/api/content/public/:type`	Get public content by type
GET	`/api/content/:id`	Get content detail
GET	`/api/strategies/public`	List all published strategies
GET	`/api/strategies/:id`	Get strategy detail
GET	`/api/strategies/:id/calls`	Get strategy calls
GET	`/api/strategies/:id/positions`	Get strategy positions
GET	`/api/strategies/:id/plans`	Get strategy subscription plans
GET	`/api/live-call-counts`	Get live call counts by category
GET	`/api/live-price/:symbol`	Get live price for a symbol
POST	`/api/live-prices/bulk`	Get live prices for multiple symbols
GET	`/api/symbols/search`	Search NSE/BSE/MCX symbols
GET	`/api/option-chain/expiries`	Get option chain expiry dates
GET	`/api/option-chain`	Get option chain data
GET	`/api/notifications/vapid-key`	Get VAPID public key
GET	`/api/ekyc/configured`	Check if eKYC is configured
POST	`/api/advisors/:id/questions`	Submit question to advisor
POST	`/api/notifications/subscribe`	Subscribe to push notifications
DELETE	`/api/notifications/subscribe`	Unsubscribe from push notifications
GET	`/sitemap.xml`	Dynamic sitemap

7.3 Investor APIs (Requires Login)

Method	Endpoint	Description
GET	`/api/strategies/:id/performance`	Get strategy performance data
POST	`/api/strategies/:id/subscribe`	Subscribe to a strategy
POST	`/api/payments/create-order`	Create Cashfree payment order
POST	`/api/payments/verify`	Verify payment
GET	`/api/payments/history`	Get payment history

Method	Endpoint	Description
GET	`/api/strategies/:id/subscription-status`	Check subscription status
GET	`/api/investor/subscriptions`	Get investor’s subscriptions
GET	`/api/investor/recommendations`	Get recommendations feed
GET	`/api/investor/watchlist`	Get watchlist items
POST	`/api/investor/watchlist`	Add to watchlist
DELETE	`/api/investor/watchlist`	Remove from watchlist
GET	`/api/investor/watchlist/check`	Check if item is in watchlist
GET	`/api/investor/watchlist/ids`	Get watchlist item IDs
POST	`/api/risk-profiles`	Submit risk profile
GET	`/api/risk-profiles/:subscriptionId`	Get risk profile
GET	`/api/risk-profiling/check`	Check risk profiling status
POST	`/api/esign/otp`	Request e-Sign Aadhaar OTP
POST	`/api/esign/verify`	Verify e-Sign Aadhaar OTP
GET	`/api/esign/status`	Check e-Sign status
POST	`/api/ekyc/aadhaar/otp`	Request eKYC Aadhaar OTP
POST	`/api/ekyc/aadhaar/verify`	Verify eKYC Aadhaar OTP
POST	`/api/ekyc/pan/verify`	Verify PAN
GET	`/api/ekyc/status`	Check eKYC status
GET	`/api/notifications/recent`	Get recent notifications

7.4 Advisor APIs (Requires Advisor Role)

Method	Endpoint	Description
GET	`/api/advisor/strategies`	Get advisor’s strategies
POST	`/api/strategies`	Create strategy
PATCH	`/api/strategies/:id`	Update strategy
DELETE	`/api/strategies/:id`	Delete strategy
POST	`/api/strategies/:id/calls`	Create call
POST	`/api/strategies/:id/positions`	Create position
GET	`/api/advisor/strategies/:id/calls`	Get all calls (including drafts)
GET	`/api/advisor/strategies/:id/positions`	Get all positions (including drafts)
PATCH	`/api/calls/:id`	Update call
POST	`/api/calls/:id/close`	Close call
PATCH	`/api/calls/:id/exit`	Update call exit
POST	`/api/calls/:id/publish`	Publish call
POST	`/api/positions/:id/publish`	Publish position
PATCH	`/api/positions/:id`	Update position
POST	`/api/positions/:id/close`	Close position
PATCH	`/api/positions/:id/exit`	Update position exit

Method	Endpoint	Description
GET	`/api/advisor/plans`	Get advisor's plans
POST	`/api/plans`	Create plan
DELETE	`/api/plans/:id`	Delete plan
GET	`/api/advisor/subscribers`	Get subscribers list
GET	`/api/advisor/subscriptions`	Get all subscriptions
GET	`/api/advisor/content`	Get advisor's content
POST	`/api/content`	Create content
DELETE	`/api/content/:id`	Delete content
GET	`/api/advisor/scores`	Get compliance scores
POST	`/api/advisor/scores`	Create compliance score
PATCH	`/api/advisor/profile`	Update advisor profile
GET	`/api/advisor/reports/download`	Download calls report (CSV)
GET	`/api/advisor/payments`	Get advisor payment history
GET	`/api/advisor/revenue`	Get revenue analytics
GET	`/api/advisor/questions`	Get investor questions
GET	`/api/advisor/questions/unread-count`	Get unread question count
PATCH	`/api/advisor/questions/:id`	Answer question
PATCH	`/api/advisor/settings/risk-profiling`	Toggle risk profiling
GET	`/api/advisor/settings/risk-profiling`	Get risk profiling setting
GET	`/api/advisor/agreements/:subscriptionId`	Get e-Sign agreement
GET	`/api/advisor/ekyc/:subscriptionId`	Get eKYC details

7.5 Admin APIs (Requires Admin Role)

Method	Endpoint	Description
GET	`/api/admin/users`	Get all users
PATCH	`/api/admin/users/:id`	Update user (approve/reject)
DELETE	`/api/admin/users/:id`	Delete user
GET	`/api/admin/strategies`	Get all strategies
PATCH	`/api/admin/strategies/:id`	Update strategy
DELETE	`/api/admin/strategies/:id`	Delete strategy
GET	`/api/admin/groww-token-status`	Get Groww API token status
POST	`/api/admin/groww-token`	Update Groww API token
POST	`/api/admin/notifications`	Send broadcast notification

7.6 Webhooks

Method	Endpoint	Description
POST	`/api/webhooks/cashfree`	Cashfree payment webhook

---

8. Third-Party Integrations

8.1 Cashfree Payment Gateway

- **SDK:** `cashfree-pg` v5
- **Purpose:** Subscription payment processing
- **Flow:** Server creates order -> Client redirects to Cashfree checkout -> Cashfree sends webhook/callback -> Server verifies and activates subscription
- **Environment Variables:** `CASHFREE_APP_ID`, `CASHFREE_SECRET_KEY`
- **Helper File:** `server/cashfree.ts`

8.2 SendGrid Email Service

- **SDK:** `@sendgrid/mail`
- **Purpose:** Transactional emails
- **Use Cases:** Registration confirmation, password reset links, e-Sign agreement copies
- **Helper File:** `server/email.ts`
- **Integration:** Installed via Replit's SendGrid integration

8.3 Groww API (Market Data)

- **Purpose:** Real-time market prices (LTP), option chain data
- **Features:** Single and bulk price queries, option chain with expiry dates
- **Token Management:** Admin can update API tokens via admin dashboard
- **Environment Variables:** `GROWW_API_KEY`, `GROWW_API_SECRET`
- **Helper File:** `server/groww.ts`

8.4 Sandbox.co.in (eKYC & e-Sign)

- **Purpose:** Aadhaar OTP verification and PAN verification for KYC, Aadhaar OTP for e-Sign agreements
- **Authentication:** JWT token with 22-hour caching
- **Data Storage:** Only masked PII stored (last 4 digits Aadhaar, masked PAN)
- **Environment Variables:** `SANDBOX_API_KEY`, `SANDBOX_API_SECRET`
- **Helper File:** `server/sandbox-kyc.ts`

8.5 Replit Object Storage

- **Purpose:** File storage for SEBI certificates and advisor document uploads
- **SDK:** `@google-cloud/storage`
- **Integration:** Installed via Replit's object storage integration
- **Environment Variables:** `DEFAULT_OBJECT_STORAGE_BUCKET_ID`, `PUBLIC_OBJECT_SEARCH_PATHS`, `PRIVATE_OBJECT_DIR`

8.6 Web Push (VAPID)

- **Library:** `web-push`
- **Purpose:** Browser push notifications for real-time strategy updates
- **Architecture:** Service worker (`sw.js`) + server-side push delivery
- **Helper File:** `server/push.ts`

---

9. Authentication & Security

9.1 Authentication Mechanism

- **Type:** Session-based authentication
- **Session Store:** PostgreSQL via `connect-pg-simple`
- **Password Hashing:** `scrypt` with random salt (Node.js crypto)
- **Session Secret:** Stored in `SESSION_SECRET` environment variable

9.2 Authorization Middleware

Three middleware functions control access:

```
requireAuth(req, res, next)     // Any logged-in user
requireAdvisor(req, res, next)  // Only users with role="advisor"
requireAdmin(req, res, next)    // Only users with role="admin"
```

9.3 Security Practices

- Passwords never returned in API responses (filtered with destructuring)
- Masked PII storage for eKYC (only last 4 digits of Aadhaar, masked PAN)
- Server-side payment amount validation
- Idempotent webhook processing (prevents duplicate subscription activation)
- SEBI performance disclaimer required before viewing performance data
- All sensitive credentials stored as environment secrets (never in code)

---

10. Frontend Architecture

10.1 Routing

Client-side routing with `wouter`. All routes defined in `client/src/App.tsx`:

Route	Page	Access
`/`	Home / Landing Page	Public
`/login`	Login	Public
`/register`	Registration	Public
`/forgot-password`	Password Reset Request	Public
`/reset-password`	Password Reset Form	Public
`/strategies`	Strategy Marketplace	Public
`/strategies/:id`	Strategy Detail	Public
`/strategies/:id/performance`	Performance Detail	Authenticated
`/strategies/:id/subscribe`	Subscription Flow	Authenticated
`/strategies/:id/esign-agreement`	e-Sign Agreement	Authenticated
`/strategies/:id/payment`	Payment	Authenticated
`/payment-callback`	Payment Result	Authenticated
`/ekyc`	eKYC Verification	Authenticated

Route	Page	Access
`/risk-profiling`	Risk Profiling	Authenticated
`/advisors`	Browse Advisors	Public
`/advisors/:id`	Advisor Detail	Public
`/market-outlook`	Market Outlook Content	Public
`/learn`	Educational Content	Public
`/content/:id`	Content Detail	Public
`/investor-dashboard`	Investor Dashboard	Investor
`/dashboard/*`	Advisor Dashboard	Advisor
`/admin/*`	Admin Dashboard	Admin
`/terms-and-conditions`	Terms	Public
`/privacy-policy`	Privacy Policy	Public
`/cancellation-policy`	Cancellation Policy	Public
`/legal-agreement`	Legal Agreement	Public
`/shipping-and-delivery`	Shipping & Returns	Public
`/contact-us`	Contact Us	Public

10.2 State Management

- **Server State:** TanStack React Query (caching, refetching, mutations)
- **Auth State:** Custom `useAuth` hook with React Context
- **Local UI State:** React `useState` / `useCallback`
- **Persistent UI State:** localStorage (e.g., performance reveal state)

10.3 Design System

- **Component Library:** shadcn/ui (built on Radix UI primitives)
- **Styling:** Tailwind CSS with custom theme variables
- **Colors:**
  - Primary: `hsl(10, 72%, 48%)` (warm red)
  - Accent: `hsl(145, 45%, 42%)` (green)
- **Dark Mode:** Supported via class-based toggle
- **Icons:** lucide-react for UI icons, react-icons for brand logos
- **Charts:** Recharts for data visualization

---

11. Deployment & Environment

11.1 Environment Variables

Variable	Purpose	Required
`DATABASE_URL`	PostgreSQL connection string	Auto-managed
`SESSION_SECRET`	Express session encryption key	Yes
`CASHFREE_APP_ID`	Cashfree API app ID	Yes

Variable	Purpose	Required
`CASHFREE_SECRET_KEY`	Cashfree API secret	Yes
`GROWW_API_KEY`	Groww market data API key	Yes
`GROWW_API_SECRET`	Groww market data API secret	Yes
`SANDBOX_API_KEY`	Sandbox.co.in API key (eKYC/e-Sign)	Yes
`SANDBOX_API_SECRET`	Sandbox.co.in API secret	Yes
`DEFAULT_OBJECT_STORAGE_BUCKET_ID`	Replit object storage bucket	Auto-managed
`PUBLIC_OBJECT_SEARCH_PATHS`	Public object paths	Auto-managed
`PRIVATE_OBJECT_DIR`	Private object directory	Auto-managed

11.2 Build & Run Scripts

```
{
  "dev": "tsx server/index.ts",           // Development server
  "build": "vite build",                  // Production build
  "start": "NODE_ENV=production tsx server/index.ts",  // Production start
  "db:push": "drizzle-kit push",          // Push schema to database
  "db:push --force": "drizzle-kit push --force"  // Force push schema
}
```

11.3 Hosting

- **Platform:** Replit Deployments
- **Domain:** `.replit.app` (custom domain configurable)
- **Port:** Frontend and backend served on port 5000
- **Vite Integration:** Backend serves Vite dev server in development, static build in production

---

12. Scope of Work Summary

Completed Modules

1. User Authentication & Registration (session-based, 3 roles)
2. Advisor Approval Workflow (admin reviews, approves/rejects)
3. Strategy Management (CRUD, Draft/Published lifecycle)
4. Calls Management (create, publish, close, P&L tracking)
5. Positions Management (F&O, multi-leg, intraday auto-square-off)
6. Subscription Plans & Pricing (advisor creates plans)
7. e-Sign Agreement (Aadhaar OTP-based, SEBI mandate)
8. Payment Processing (Cashfree gateway, webhooks)
9. eKYC Verification (Aadhaar + PAN via Sandbox.co.in)
10. Risk Profiling (8-section questionnaire, dual scoring)
11. Performance Reveal (SEBI disclaimer gated, per-strategy)
12. Live Market Data (Groww API, LTP, option chain)
13. Web Push Notifications (VAPID, strategy-specific)

14. Content Management (Market Outlook, Learn articles)
15. Investor Dashboard (subscriptions, recommendations, watchlist)
16. Advisor Dashboard (strategies, subscribers, revenue, reports)
17. Admin Dashboard (users, strategies, settings)
18. Communication (advisor questions, email notifications)
19. Compliance Reports (CSV export)
20. SEO (sitemap, meta tags, Open Graph)
21. Legal Pages (terms, privacy, cancellation, shipping)

Key Technical Metrics

- **Backend Code:** ~2,560 lines (routes.ts) + ~700 lines (storage.ts)
- **Database Schema:** ~375 lines, 15 tables, 4 enums
- **Frontend Pages:** 25+ pages across 3 role dashboards
- **API Endpoints:** 90+ REST endpoints
- **Third-Party Integrations:** 6 (Cashfree, SendGrid, Groww, Sandbox.co.in, Object Storage, Web Push)

This document provides a comprehensive overview of the AlphaMarket Connect platform. For implementation-level details, refer to the source code files listed in the Project Structure section.