Development Phase Documentation

1. Frontend Documentation

1.1 Major Functionalities

- Authentication (Signup/Login/Logout) – Secure login and signup forms with validation.
- Signal Management – View signals list, details, and allow traders to post/edit signals.
- Engagement Features – Like/unlike, comment, and watchlist management.
- Subscription & Payments – Subscribe to traders, manage plans, and handle renewals.
- Reports & Leaderboard – Admin report generation and leaderboard display.
- Admin Tools – Signal verification and user management.

1.2 Pseudo Code Snippets

User Signup Flow

```
FUNCTION handleSignup(formData):
    VALIDATE formData.email, formData.password
    SEND POST request → /auth/register with formData
    IF response.success:
        SHOW success notification
        REDIRECT user to login page
    ELSE:
        SHOW error message
```

User Login Flow

```
FUNCTION handleLogin(credentials):
    VALIDATE credentials
    SEND POST request → /auth/login
    IF response includes token:
        STORE token in localStorage
        UPDATE auth state to loggedIn
        REDIRECT to /signals
    ELSE:
        SHOW error message
```

Post Signal (Trader)

```
FUNCTION handlePostSignal(signalData):
    CHECK user.role == 'Trader'
    VALIDATE signalData (symbol, price range, stopLoss, targets)
    SEND POST request → /signals with signalData
    IF success:
        SHOW "Signal Posted" notification
        REFRESH signals list
    ELSE:
        SHOW error message
```

Like/Unlike Signal

```
FUNCTION toggleLike(signalId):
    SEND POST request → /signals/{id}/like
    IF success:
        UPDATE like count in UI
    ELSE:
        SHOW error message
```

Subscribe to Trader

```
FUNCTION subscribeTrader(traderId, planId):
    SEND POST request → /subs/checkout with {traderId, planId}
    REDIRECT to payment gateway (checkoutUrl)
    ON success webhook → update subscription state
```

2. Backend Documentation

Section A: Simplified Endpoint Map

Auth

• POST /auth/register – create user

• POST /auth/login – get JWT/session

• POST /auth/logout – end session

• GET /users/me – current user profile & roles

Tickers & Market

• GET /tickers?query=SYM – search symbols

• GET /market/{symbol}/prices?range=1d|5d|1m – candles/snapshots (cached)

Signals

• GET /signals?symbol=SYM&visibility;=public|all&cursor;=… – feed/list

• POST /signals – create (trader only)

• GET /signals/{id} – details (access-guarded)

• POST /signals/{id}/close – close & compute outcome

• POST /signals/{id}/targets/hit – mark target hit (system/secure)

Engagement

• POST /signals/{id}/like – toggle like

• GET /signals/{id}/comments – list comments

• POST /signals/{id}/comments – add comment

Watchlist

• GET /watchlist – my symbols

• PUT /watchlist/{symbol} – add/update

• DELETE /watchlist/{symbol} – remove

Subscriptions & Payments

• GET /subs/status?traderId=… – check access

• POST /subs/checkout – create pending sub + payment intent

• POST /payments/webhook – provider → confirm

• GET /subs/my – my active/expired subs

Leaderboard & Profiles

• GET /leaderboard?period=7d|30d|all – top traders

• GET /traders/{traderId} – trader profile & KPIs

Notifications

• GET /notifications?cursor=… – inbox

• POST /notifications/{id}/read – mark read

Admin (secured)

• POST /admin/signals/{id}/hide – moderate

• POST /admin/users/{id}/block – block/unblock user

• GET /admin/audit?cursor=… – audit log

Health/Meta

• GET /health – liveness

• GET /ready – readiness

• GET /version – build/version info

Minimal DTOs:

• SignalCreate: { symbol, buyFrom, buyTo, stopLoss, risk, visibility, rationale?, targets:[{seq, price}] }

• SignalView: { id, trader:{id,name}, symbol, buyRange:{from,to}, stopLoss, postedAt, status, targets:[{seq, price, hit, hitAt?}], metrics:{likes, comments}, consensus?:{count} }

• CheckoutReq: { traderId, planId } → { checkoutUrl, paymentId }

• WebhookEvent: { id, type, meta:{subId,paymentId}, amount, currency }

Section B: Backend Pseudocode Flows

1) Post Signal

• Authenticate trader; validate ranges and symbol.

• Insert Signal (Status='Open', PostedAt=UTC now).

• Insert ordered Targets; assert seq unique per signal.

• Update consensus group within ±48h; set consensus flag if threshold met.

• Fan-out notifications to followers; enqueue async jobs.

• Audit: Create Signal.

2) Can View Signal (Access control)

• If Public → allow.

• If requester is owner → allow.

• Else require active subscription where now ∈ [StartAt, EndAt].

• If invite-only plan, must include valid InviteCode.

• If not, deny with 403.

3) Close Signal & Outcome

• Authenticate trader; ensure ownership & Status='Open'.

• Fetch latest price; compute entry = avg(buyFrom,buyTo).

• Compute pctReturn and maxDrawdown; insert SignalOutcome; set Status='Closed'.

• Recompute TraderProfile KPIs; invalidate caches.

• Audit: Close Signal.

4) Update Consensus

• Find or create ConsensusGroup for symbol overlapping ±48h.

• Increment SignalCount or create new group.

• Flag consensus if group count ≥ threshold.

5) Leaderboard Recalculation (Nightly)

• For each trader: join closed Signals with Outcomes.

• Compute successRate, avgReturn, totalSignals.

• Optionally recency-weight results.

• Update TraderProfile KPIs; cache leaderboard.

6) Subscription Checkout → Activation

• Create Subscription(Status='Pending'); create PaymentIntent with meta{subId}.

• On webhook payment.succeeded: set Subscription=Active with StartAt, EndAt.

• Update Payment to Paid; notify subscriber.

• Audit: Subscription activation.

7) Market Data ETL

• Scheduler fetches quotes; normalize & upsert MarketPrices.

• Check open signals: mark targets hit if crossed.

• Trigger notifications.

8) Engagement (Likes, Comments, Watchlist)

• Like: upsert unique (UserId, SignalId).

• Comment: insert sanitized; notify owner.

• Watchlist: upsert unique (UserId, Symbol).

9) Notifications Fan-Out

• Determine recipients (subs, watchlist).

• Insert Notification; enqueue async delivery.

• Track delivery results.

10) Admin & Compliance

• Admins can hide signals, block users, refund via provider.

• All actions AuditLogged.

• Exports possible for compliance requests.

11) Security & Reliability

• JWT auth, secure cookies, 2FA optional.

- Policy-based authorization checks.
- Input validation, output encoding, SQL parameterization.
- Rate limiting & idempotency for webhooks.
- Logs, metrics, backups, migrations with compatibility.