StaffSynk - Technical Stack Documentation

1. Core Technologies

- **Frontend**: Flutter 3.19 (Dart 3.0)
- **Backend**: NestJS 10
- **Database**: Supabase PostgreSQL 15 with PostGIS 3.3
- **Authentication**: Firebase Authentication v9
- **Realtime Services**: Supabase Realtime + Firebase Cloud Messaging (FCM)
- **File Storage**: Supabase Storage
- **Geolocation**: Flutter Geolocator 9.0 + Apple CoreLocation

2. Data Layer Architecture

2.1 Supabase PostgreSQL Schema

- **Tables**:
  - `auth.users` (Firebase-mirrored)
  - `profiles` (Extended user attributes)
  - `sites` (PostGIS geofence polygons)
  - `shifts` (Time-bound assignments)
  - `clock_ins` (GPS-validated attendance)
  - `certifications` (Expiry-tracked documents)
- **Row-Level Security (RLS)**:
  - Role-based policies using JWT claims from Firebase
  - Example: Workers can only view their own shifts
- **PostGIS Functions**:
  - Geofence validation stored procedures
  - Proximity-based site queries

2.2 Firebase Collections

- `fcm_tokens` (Device-specific push tokens)
- `auth_events` (Login/logout auditing)

3. Service Layer

3.1 NestJS Microservices

- **Modules**:
  - Auth Sync (Firebase ↔ Supabase user mirroring)
  - Payroll (ABA file generation)
  - Notifications (FCM message routing)
- **Key Endpoints**:
  - POST `/api/payroll/aba` (Generates bank payment files)
  - GET `/api/shifts/conflicts` (Detects scheduling overlaps)

4. Client-Side Implementation

4.1 Flutter Packages

- `supabase_flutter` (Database/Realtime)
- `firebase_auth` + `flutter_secure_storage` (iOS Keychain)
- `geolocator` (Background location tracking)
- `turf` (Client-side geofence math)

4.2 Critical Flows

- **Authentication**:
    1. User signs in via Firebase (Apple/Google)
    2. Token exchanged for Supabase JWT
    3. RLS policies enforced via custom claims
- **Shift Management**:
    1. Managers assign shifts with conflict checks
    2. Real-time updates via Supabase subscriptions
    3. FCM alerts for new assignments
- **Time Tracking**:
    1. GPS position validated against PostGIS geofence
    2. Clock-in records written with device trust score

5. Infrastructure Configuration

5.1 Supabase

- Enabled Extensions:
    - `pg_cron` (Certification expiry checks)
    - `pg_net` (HTTP triggers)
- Connection Pooling: 20 max connections

5.2 Firebase

- APNS/FCM configuration for iOS push
- App Check enforcement

6. Compliance Measures

- **Data Encryption**:
    - AES-256 for sensitive fields (BSB/account numbers)
    - iOS Secure Enclave for payment credentials
- **Audit Logging**:
    - All writes to `audit_logs` table with IP/user metadata

7. Development Standards

- **Code Style**:
    - Dart: `pedantic` linter rules
    - TypeScript: Strict mode + ESLint Airbnb
- **Testing**:
    - Golden tests for UI components
    - Postman collection for API verification

8. Deployment

- **iOS**: TestFlight builds via Fastlane
- **Android**: Firebase App Distribution
- **Backend**: GCP Cloud Run containers

9. Monitoring

- **Frontend**: Sentry error tracking
- **Backend**: Supabase Logs Explorer
- **Performance**: Firebase Performance Monitoring

10. Scalability Parameters

- Current Capacity: 400 concurrent users
- Breakpoint Triggers:
    - 500 users: Enable Supabase Connection Pooling
    - 1,000 users: Vertical scaling to 4vCPU/16GB RAM