BANKI

Voice-Powered Banking Assistant Platform

Complete System Architecture & Technical Blueprint

Version 1.0 | February 2026
Status: Production-Ready Architecture
Primary Market: Sri Lanka

Document Purpose

- Define the complete technical architecture for the Banki platform
- Establish the tech stack, database schema, and API structure
- Outline implementation phases with timelines
- Serve as the single source of truth for development

Table of Contents

1. Executive Summary

Banki is a production-grade, multi-tenant, voice-powered banking assistant platform designed for the Sri Lankan market. It enables banks to onboard customers through an AI-driven conversational KYC process powered by Google Gemini's multimodal live API. The platform combines real-time voice interaction, document verification with computer vision, facial recognition with liveness detection, intelligent product recommendations, and a comprehensive admin panel with a visual node-based flow editor.

Key Capabilities

- Real-time voice conversation with Gemini multimodal live API (phone-call style)
- AI-powered ID recognition (Sri Lankan NIC old/new format, passport, driver's license)
- Face matching with liveness detection (blink, head turn, smile challenges)
- Visual node-based flow editor (n8n-style) for customizable KYC workflows
- Intelligent product recommendations based on customer profile
- Multi-tenant architecture with custom branding per bank
- Trilingual support: English, Sinhala, Tamil
- PDF generation with QR codes, bank branding, and digital signatures
- SMS (text.lk) and email notifications
- Role-based admin panel with dashboard, review queue, and analytics
- Full audit trail, encryption at rest, 2FA, and compliance features

Target Users

User Type	Description	Key Actions
Bank Customer	End user applying for banking products	Voice conversation, ID upload, selfie, product selection
Bank Staff / Reviewer	Processes and reviews KYC applications	Review queue, approve/reject, add notes, assign tasks
Bank Admin	Manages bank settings and workflows	Flow editor, product management, branding, settings
Super Admin (Banki)	Platform owner managing all tenants	Create banks, manage subscriptions, system settings

2. System Overview & Architecture

Banki follows a modern microservices-inspired monorepo architecture with clear separation of concerns. The system is divided into four main layers: the customer-facing voice application, the admin panel, the backend API server, and supporting services (AI, verification, notifications).

High-Level Architecture

The system architecture follows this flow:

Architecture Layers

- Layer 1 — Client Apps: Customer Voice WebApp (React/Next.js PWA) + Admin Panel (React/Next.js)
- Layer 2 — API Gateway: Next.js API Routes / Express.js with rate limiting, auth, CORS
- Layer 3 — Core Services: Auth, KYC, Flow Engine, Product Engine, Notification, PDF, Tenant Manager
- Layer 4 — AI Services: Gemini Multimodal Live API, Gemini Vision API, Face Matching (Python)
- Layer 5 — Data Layer: PostgreSQL (primary), Redis (caching/sessions), S3-compatible storage (files)
- Layer 6 — External Integrations: text.lk SMS, Email (SendGrid/SMTP), Gemini API

Data Flow: Customer KYC Journey

1. Customer opens bank-branded URL (e.g., hnb.banki.lk) → Landing page with language selector
2. Agrees to terms & conditions → Voice conversation begins with AI greeting
3. AI follows the bank's custom flow (built in node editor) → Asks questions conversationally
4. Customer uploads NIC/Passport → Camera overlay guides alignment → AI extracts data via Gemini Vision
5. System validates document type, checks legitimacy, extracts personal info, pre-fills form
6. Customer takes selfie → Liveness detection challenges → Face matching against ID photo
7. AI presents extracted info on screen with edit (pencil) icons → Customer confirms or edits
8. AI recommends products based on profile → Visual cards shown → Customer selects
9. Application submitted → PDF generated with branding, QR code → Saved to server

10. Customer ID auto-generated → AI reads it aloud → SMS & email sent to customer

11. Application enters admin review queue → Reviewer approves/rejects/requests re-submission

3. Tech Stack

Frontend

Technology	Purpose	Justification
Next.js 14+	Full-stack React framework	SSR, API routes, great DX, PWA support
TypeScript	Type safety	Production reliability, better tooling
Tailwind CSS	Utility-first styling	Fast development, consistent design, easy theming per tenant
React Flow	Node-based flow editor	Mature library, n8n-like visual editing, extensible
Zustand	State management	Lightweight, simple, perfect for complex UI state
Web Audio API	Voice capture & playback	Browser-native, real-time audio streaming
WebRTC/WebSocket	Real-time voice communication	Low-latency bidirectional audio with Gemini Live API

Backend

Technology	Purpose	Justification
Next.js API Routes + Express	REST API server	Unified codebase, easy deployment, middleware support
PostgreSQL 16	Primary database	ACID compliance, JSONB for flexible schemas, row-level security
Prisma ORM	Database access	Type-safe queries, migrations, multi-tenant support
Redis	Caching & sessions	Session management, rate limiting, real-time data
Bull/BullMQ	Job queue	Background jobs: PDF gen, SMS, email, data cleanup
Socket.IO	Real-time events	Admin notifications, live session tracking, voice streaming

AI & Computer Vision

Technology	Purpose	Justification
Gemini 2.0 Multimodal Live API	Voice conversation	Real-time audio in/out, context-aware, multilingual

Gemini Vision API	ID document extraction	OCR, document type recognition, data extraction
Python (FastAPI)	Face matching microservice	OpenCV, face_recognition lib, dlib — cost-effective
TensorFlow.js / ONNX	Liveness detection (browser)	Client-side blink/head detection, no server round-trip

Infrastructure & DevOps

Technology	Purpose	Justification
Docker + Docker Compose	Containerization	Consistent environments, easy scaling
Nginx	Reverse proxy, SSL	Load balancing, subdomain routing per tenant
GitHub Actions	CI/CD pipeline	Auto-deploy on push, testing, linting
MinIO / AWS S3	File storage	Documents, selfies, PDFs — S3-compatible, self-hostable
Cloudflare	CDN & DDoS protection	Performance, security, subdomain management
Let’s Encrypt	SSL certificates	Free, auto-renewing HTTPS

External Services

Service	Purpose	Integration
text.lk	SMS notifications (Sri Lanka)	REST API — OTP, customer ID, status updates
SendGrid / SMTP	Email notifications	Welcome emails, PDF attachment, status updates
Google Gemini API	AI conversation & vision	Multimodal Live API + Vision API

4. Database Schema Design

The database uses PostgreSQL with row-level security for multi-tenant isolation. All tables include tenant_id for data separation. Sensitive data is encrypted at rest using AES-256.

Core Tables

tenants

Column	Type	Description
id	UUID (PK)	Unique tenant identifier
name	VARCHAR(255)	Bank name
slug	VARCHAR(100) UNIQUE	URL slug (e.g., 'hnb')
subdomain	VARCHAR(100) UNIQUE	Custom subdomain
branding	JSONB	Logo URL, primary/secondary colors, fonts
settings	JSONB	API keys (encrypted), thresholds, toggles
subscription_plan	ENUM	basic, pro, enterprise
subscription_status	ENUM	active, suspended, cancelled
max_applications_monthly	INTEGER	Plan limit
created_at / updated_at	TIMESTAMP	Audit timestamps

users (Admin Users)

Column	Type	Description
id	UUID (PK)	Unique user identifier
tenant_id	UUID (FK)	Bank this user belongs to (NULL for super admin)
email	VARCHAR(255)	Login email
password_hash	VARCHAR(255)	Bcrypt hashed password
role	ENUM	super_admin, bank_admin, bank_reviewer, bank_staff
two_factor_enabled	BOOLEAN	2FA status
two_factor_secret	VARCHAR(255)	TOTP secret (encrypted)
last_login	TIMESTAMP	Last login time
password_changed_at	TIMESTAMP	For 90-day expiry policy

session_timeout_minutes	INTEGER	Auto-logout setting (default 30)
is_active	BOOLEAN	Account active status

applications (KYC Submissions)

Column	Type	Description
id	UUID (PK)	Unique application identifier
tenant_id	UUID (FK)	Bank tenant
customer_id	VARCHAR(50) UNIQUE	Auto-generated (BANKI-2026-00001)
status	ENUM	in_progress, submitted, under_review, approved, rejected, re_submission
flow_version_id	UUID (FK)	Which flow version was used
personal_data	JSONB (encrypted)	Name, DOB, address, phone, email, gender
id_document_type	ENUM	nic_old, nic_new, passport, drivers_license
id_document_path	VARCHAR(500)	S3 path to document scan
id_extracted_data	JSONB	AI-extracted fields from document
id_validation_result	JSONB	Legitimacy check results, confidence scores
selfie_path	VARCHAR(500)	S3 path to selfie
face_match_score	DECIMAL(5,2)	Confidence % from face matching
liveness_passed	BOOLEAN	Liveness detection result
selected_products	JSONB	Array of selected product IDs
recommended_products	JSONB	AI-recommended products with reasoning
conversation_transcript	TEXT	Full voice conversation transcript
pdf_path	VARCHAR(500)	S3 path to generated PDF
pdf_hash	VARCHAR(64)	SHA-256 hash for tamper detection
language	ENUM	en, si, ta
ip_address	INET	Customer IP for fraud detection
session_duration_seconds	INTEGER	Total time spent
assigned_reviewer_id	UUID (FK)	Assigned staff member
reviewed_at	TIMESTAMP	When reviewed
review_notes	TEXT	Internal reviewer notes
rejection_reason	TEXT	Reason if rejected
consent_recorded_at	TIMESTAMP	When T&C was accepted
created_at / updated_at	TIMESTAMP	Audit timestamps

products

Column	Type	Description
id	UUID (PK)	Unique product identifier
tenant_id	UUID (FK)	Bank tenant
name	VARCHAR(255)	Product name (e.g., Gold Visa Card)
type	ENUM	savings_account, current_account, credit_card, debit_card, personal_loan, fixed_deposit
description	TEXT	Full product description
features	JSONB	Array of feature strings
interest_rate	DECIMAL(5,2)	Interest rate if applicable
eligibility_rules	JSONB	Min age, min income, required docs, etc.
terms_conditions	TEXT	T&C text for AI to reference
image_path	VARCHAR(500)	Product card/logo image
is_active	BOOLEAN	Currently offered
display_order	INTEGER	Sorting priority

flows (Node Editor Workflows)

Column	Type	Description
id	UUID (PK)	Unique flow identifier
tenant_id	UUID (FK)	Bank tenant
name	VARCHAR(255)	Flow name (e.g., Standard KYC)
description	TEXT	Flow description
is_template	BOOLEAN	Pre-built template flag
is_published	BOOLEAN	Currently active for customers
current_version_id	UUID (FK)	Active version
created_by	UUID (FK)	Admin who created it

flow_versions

Column	Type	Description
id	UUID (PK)	Version identifier
flow_id	UUID (FK)	Parent flow
version_number	INTEGER	Auto-incrementing version
nodes	JSONB	Array of node objects (type, position, config, AI prompt)

edges	JSONB	Array of edge connections between nodes
created_at	TIMESTAMP	When this version was saved
created_by	UUID (FK)	Who saved this version

Additional Tables

Table	Purpose	Key Columns
audit_logs	Full audit trail for compliance	user_id, action, entity_type, entity_id, changes, ip_address, timestamp
notifications	In-app admin notifications	tenant_id, user_id, type, title, message, is_read, created_at
sms_logs	SMS delivery tracking	application_id, phone, message, status, provider_response, sent_at
email_logs	Email delivery tracking	application_id, email, subject, status, sent_at
sessions	Active customer sessions	application_id, tenant_id, flow_node_id, last_activity, is_active
fraud_flags	Anti-fraud detection results	application_id, flag_type (duplicate_id, same_ip), details, resolved
data_deletion_requests	GDPR-style right to be forgotten	application_id, requested_at, completed_at, status
review_assignments	Review queue assignment tracking	application_id, reviewer_id, assigned_at, sla_due_at, status

5. Module Breakdown

5.1 Customer-Facing Voice Application

The customer-facing application is a Progressive Web App (PWA) built with Next.js, designed mobile-first for Sri Lankan users. It provides a real-time voice conversation experience powered by Gemini's Multimodal Live API.

Landing Page

- Bank-branded landing page with logo, colors, and bank name (loaded from tenant config)
- Language selector: English, Sinhala, Tamil
- Terms & Conditions agreement (customizable per bank)
- Bank contact info (phone, email, branches) in footer/sidebar
- "Start Application" button initiating the voice session
- Accessibility mode toggle (larger fonts, high contrast)
- Dark mode toggle

Voice Conversation Interface

- Phone-call style continuous listening (no push-to-talk)
- Real-time text transcript displayed on screen (subtitle-style)
- AI interruption handling — stops speaking when user talks
- Mute button for user microphone
- Progress indicator sidebar showing current step in the flow
- Session timeout after 10 minutes of inactivity
- Internet disconnection handling: auto-save progress, reconnect prompt
- Fallback to text chat if voice fails or user prefers typing
- Empathy responses for natural conversation (e.g., "No worries, take your time!")
- Off-topic handling: brief response then redirect to current step

Document Capture

- Camera overlay with guide rectangle: "Place your NIC here"
- Auto-capture when document is detected in frame
- Manual capture button as fallback

- Gallery upload option for pre-existing photos
- Front and back capture for Sri Lankan NIC

Selfie & Liveness Detection

- Camera preview with face detection overlay
- Multi-challenge liveness: blink, turn left, turn right, smile
- Client-side detection using TensorFlow.js (no server round-trip for liveness)
- Face matching performed server-side (Python microservice)
- Adjustable confidence threshold per tenant (default 85%)
- Handles glasses, hijab, facial hair variations gracefully

Data Confirmation Screen

- All extracted info displayed in editable form
- Pencil icon on each field for inline editing (type or voice)
- AI confirms changes verbally after edit: “Got it, updated your address”
- Visual product recommendation cards with images, features, rates
- Product comparison capability: “Show me Gold vs Platinum”
- Skip/not interested option for products
- Eligibility explanations if customer is ineligible for a product
- Multiple product selection supported in single session

Completion

- Auto-generated customer ID read aloud by AI (format: BANKI-2026-00001)
- SMS sent via text.lk with customer ID and status
- Welcome email sent with PDF application attached
- Session summary displayed on screen
- Session resumability if interrupted — “Welcome back! Shall we continue from step 3?”

5.2 Admin Panel

The admin panel is a full-featured dashboard for bank staff and administrators to manage all aspects of the Banki platform.

Dashboard

- Real-time stats: applications today, pending review, approved, rejected
- Charts/graphs: applications over time, approval rates, conversion funnels
- Live active sessions counter: “3 customers currently in KYC”
- Product popularity analytics
- AI performance metrics: ID recognition accuracy, average face match score
- Average completion time tracking
- Filterable by date range, status, product type
- Export to CSV/Excel
- Monthly PDF reports generation

Application Review Queue

- List of all submitted applications with status filters
- Detailed view: all customer data, ID scan, selfie, face match score, transcript
- Side-by-side ID photo vs. selfie comparison
- Approve / Reject / Request Re-submission actions
- Rejection reason (sent to customer via SMS/email)
- Re-submission: customer receives SMS with link to redo specific step
- Internal notes per application
- Assignment system: assign to specific reviewer
- SLA tracking: overdue application alerts (e.g., pending > 3 days)
- Bulk actions: select multiple and approve/reject
- Re-open rejected applications
- Full audit trail per application: who did what, when

Role-Based Access Control

Role	Permissions
Super Admin	Manage all tenants, create banks, system settings, billing, impersonate bank admins

Bank Admin	All bank settings, flow editor, products, branding, staff management, reports
Bank Reviewer	Review queue, approve/reject, notes, but cannot edit settings or flows
Bank Staff	View-only dashboard, basic application lookup, no actions

Authentication & Security

- Email + password login with bcrypt hashing
- SSO: Google and Microsoft OAuth support
- 2FA: TOTP (authenticator app) and SMS OTP options
- Password policy: min 8 chars, special characters, 90-day expiry
- Session timeout: configurable (default 30 min inactivity)
- Super admin impersonation mode for support
- IP logging for all admin actions

5.3 Node-Based Flow Editor

The visual flow editor is the core configuration tool, allowing bank admins to design the KYC conversation flow using an n8n-style drag-and-drop interface built with React Flow.

Node Types

Node Type	Color	Description	Configuration
Greeting	Green	Conversation opener	Custom greeting text, AI personality prompt
Question	Blue	Voice/text question	Question text, expected answer type, required/optional, AI prompt
File Upload	Purple	Document/selfie capture	Upload type (ID/selfie), camera overlay, accepted formats
Verification	Orange	AI verification step	ID validation, face matching, threshold settings
Condition	Yellow	If/else branching	Field to check, operator, value (e.g., age < 18)
Product Rec.	Teal	Product recommendation	Auto (AI picks) or manual (specific products)
Multiple Choice	Indigo	Selection from options	Options list, allow multiple, display as cards/list
Webhook	Red	External API call	URL, method, headers, body template, response mapping
Delay	Gray	Wait timer	Duration, then trigger (e.g., follow-up SMS after 24h)
Notification	Pink	Send alert	Channel (SMS/email/in-app), template, recipients
AI Prompt	Cyan	Custom AI instruction	System prompt override for that step
Completion	Green	End flow	Generate PDF, create customer ID, send notifications

Editor Features

- Drag-and-drop node placement on canvas
- Connect nodes with edges (arrows showing flow direction)
- Color-coded nodes by type for visual clarity
- Node configuration panel (click node to edit settings)
- Conditional branching with if/else paths
- Timeout logic per node: re-prompt after X seconds of silence
- Required vs. optional flag per node
- Preview/simulate: test the flow as a fake customer conversation

- Version history: every save creates a new version, rollback supported
- Duplicate flows: copy existing flow as starting template
- Pre-built templates: Standard KYC, Quick Account Opening, Loan Application
- Publish/unpublish toggle: set which flow is active

5.4 KYC & Identity Verification Engine

Document Recognition

- Sri Lankan NIC: old format (9 digits + V/X) and new format (12 digits)
- Passport: MRZ reading, data extraction
- Driver's License: front extraction
- Auto-detect document type using Gemini Vision
- Extract: full name, date of birth, gender, NIC number, address
- Calculate age and DOB from old NIC number encoding
- Detect gender from NIC number
- Validate NIC checksum to catch typos
- Flag expired documents
- Front and back scanning for NIC (address on back)

Legitimacy Checks

- Document structure validation (correct layout, expected fields present)
- Image quality assessment (blur detection, lighting, resolution)
- Consistency checks (does extracted data match expected patterns)
- Duplicate document detection: flag if same NIC used in another application
- Multiple applications from same IP flagged for review
- Wrong document handling: "That doesn't look like an NIC. Could you try again?"

Face Matching

- Python microservice using face_recognition library (dlib-based)
- Compare selfie against ID photo, return confidence percentage
- Configurable threshold per tenant (default 85%)
- Below threshold: flagged for manual admin review
- Both ID photo and selfie stored for admin side-by-side comparison
- Handles variations: glasses, hijab, facial hair

Liveness Detection

- Runs in browser using TensorFlow.js (no server dependency)
- Multi-challenge sequence: blink → turn head left → turn right → smile
- Admin-configurable number of challenges

- Prevents photo-of-photo attacks
- Results sent to server for audit logging

5.5 Product Recommendation Engine

Gemini analyzes customer profile data (age, income, occupation, etc.) collected during the KYC flow and matches it against the bank's product catalog with eligibility rules.

- AI-driven matching: Gemini evaluates customer profile vs. product eligibility rules
- Visual product cards shown on screen while AI describes them verbally
- Side-by-side comparison on request
- Eligibility explanation: "You qualify for Gold Card because..."
- Ineligibility handling: "The Platinum Card requires a minimum income of..., but I'd recommend..."
- Multi-product selection: customer can choose multiple in one session
- T&C explanation: AI can answer follow-up questions about terms
- Skip option: customer can decline all product recommendations
- Admin product analytics: which products are recommended vs. selected most

5.6 PDF Generation & Document Management

Generated PDF Contents

- Bank branding: logo, colors, name in header/footer
- Customer ID prominently displayed
- All personal data collected during KYC
- ID document scan image
- Selfie image
- Selected products
- Full conversation transcript
- Timestamp for each data point collected
- QR code linking to digital application in admin panel
- Page numbers, headers, footers
- Digital signature/hash (SHA-256) for tamper detection
- Password protection option

Storage & Management

- PDFs saved to S3-compatible storage organized by tenant/date
- Downloadable from admin panel
- Attached to welcome email
- Automatic cleanup for rejected applications (configurable retention period)
- Archival to cold storage after configurable period (default 1 year)

5.7 Notification System

SMS (text.lk Integration)

- OTP verification during KYC
- Customer ID delivery after application
- Status updates: approved, rejected, re-submission requested
- Customizable SMS templates per tenant
- Bulk SMS capability from admin panel
- Delivery tracking and logging
- Settings page: API key, sender name, rate limits

Email (SendGrid/SMTP)

- Welcome email with PDF attachment
- Status update emails
- Customizable email templates with bank branding
- Configurable SMTP settings in admin

In-App Notifications (Admin)

- Real-time alerts when new application submitted
- High-value application alerts
- SLA overdue warnings
- System alerts (API errors, storage issues)

5.8 Multi-Tenant & Branding System

Tenant Isolation

- Each bank is a separate tenant with isolated data
- PostgreSQL row-level security ensures data separation
- Separate S3 buckets/prefixes per tenant
- Subdomain routing: hnb.banki.lk, boc.banki.lk

Branding Customization (Admin Panel)

- Logo upload
- Primary and secondary colors
- Font selection
- Custom greeting message
- AI personality/tone configuration (formal vs. friendly)
- AI name customization (e.g., “Hi, I’m Nila from HNB”)
- Terms & conditions text
- Contact information
- All changes reflected immediately on customer-facing app

Subscription & Billing (Super Admin)

- Plans: Basic (100 apps/month), Pro (1000 apps/month), Enterprise (unlimited)
- Usage tracking per tenant
- Billing dashboard
- Suspend/cancel tenant accounts
- Data export for departing tenants

6. API Structure

The API follows RESTful conventions with JWT authentication. All endpoints are prefixed with /api/v1.

Authentication

Method	Endpoint	Description
POST	/auth/login	Email + password login, returns JWT
POST	/auth/register	Create new admin user (by bank admin or super admin)
POST	/auth/2fa/setup	Enable 2FA, returns QR code
POST	/auth/2fa/verify	Verify 2FA code
POST	/auth/sso/google	Google SSO login
POST	/auth/sso/microsoft	Microsoft SSO login
POST	/auth/refresh	Refresh JWT token
POST	/auth/logout	Invalidate session
POST	/auth/forgot-password	Send password reset email

Applications (KYC)

Method	Endpoint	Description
POST	/applications/start	Begin new KYC session
PATCH	/applications/:id	Update application data
POST	/applications/:id/upload-id	Upload ID document
POST	/applications/:id/upload-selfie	Upload selfie
POST	/applications/:id/verify-face	Trigger face matching
POST	/applications/:id/verify-liveness	Submit liveness results
POST	/applications/:id/submit	Final submission
GET	/applications/:id/status	Check application status
POST	/applications/:id/resume	Resume interrupted session
GET	/applications	List all (admin, with filters)
GET	/applications/:id	Get full details (admin)
PATCH	/applications/:id/review	Approve/reject/request re-submission
POST	/applications/:id/assign	Assign to reviewer
POST	/applications/:id/notes	Add internal note
POST	/applications/bulk-action	Bulk approve/reject

Products

Method	Endpoint	Description
GET	/products	List all products (filtered by tenant)
POST	/products	Create new product
PATCH	/products/:id	Update product
DELETE	/products/:id	Deactivate product
GET	/products/recommend/:applicationId	Get AI recommendations for customer

Flows

Method	Endpoint	Description
GET	/flows	List all flows for tenant
POST	/flows	Create new flow
GET	/flows/:id	Get flow with current version
PATCH	/flows/:id	Update flow metadata
POST	/flows/:id/versions	Save new version (nodes + edges)
GET	/flows/:id/versions	List all versions
POST	/flows/:id/publish	Set as active flow
POST	/flows/:id/duplicate	Duplicate flow
GET	/flows/templates	List pre-built templates
POST	/flows/:id/simulate	Test flow simulation

Tenants (Super Admin)

Method	Endpoint	Description
GET	/tenants	List all bank tenants
POST	/tenants	Create new tenant/bank
PATCH	/tenants/:id	Update tenant settings
PATCH	/tenants/:id/branding	Update branding
POST	/tenants/:id/suspend	Suspend tenant
GET	/tenants/:id/usage	Get usage stats/billing
POST	/tenants/:id/export	Export all tenant data

Voice / AI

Method	Endpoint	Description

WS	/voice/session	WebSocket for real-time voice streaming to/from Gemini
POST	/ai/extract-id	Extract data from ID document image
POST	/ai/validate-id	Validate document legitimacy
POST	/ai/face-match	Compare selfie with ID photo

Notifications, Analytics, Settings, Audit

Method	Endpoint	Description
POST	/notifications/sms	Send SMS via text.lk
POST	/notifications/email	Send email
GET	/notifications/admin	Get admin in-app notifications
GET	/analytics/dashboard	Dashboard stats
GET	/analytics/funnel	Conversion funnel data
GET	/analytics/products	Product popularity stats
GET	/analytics/ai-performance	AI accuracy metrics
GET	/analytics/export	Export as CSV/Excel/PDF
GET	/settings	Get tenant settings
PATCH	/settings	Update settings (API keys, thresholds, etc.)
GET	/audit-logs	Query audit trail (with filters)

7. Security & Compliance

Data Protection

- AES-256 encryption at rest for all sensitive personal data
- TLS 1.3 for all data in transit
- PostgreSQL row-level security for tenant isolation
- S3 server-side encryption for documents and selfies
- Encrypted API keys in tenant settings (never stored in plaintext)
- Password hashing with bcrypt (cost factor 12)

Access Control

- JWT tokens with short expiry (15 min) + refresh tokens
- Role-based access control (RBAC) with four roles
- 2FA enforcement option per tenant
- Session timeout (configurable, default 30 min)
- IP-based admin action logging
- Rate limiting: 100 requests/min for API, 5 requests/min for auth endpoints
- CORS restricted to tenant subdomains

Compliance Features

- Full audit trail: every action logged with who, what, when, where
- Consent recording with timestamp
- Data deletion requests (right to be forgotten)
- Configurable data retention periods
- Automatic cleanup of rejected application data
- Data export capability for regulatory requests
- Terms & conditions per tenant
- Privacy policy pages (editable per bank)

Anti-Fraud

- Duplicate NIC detection across applications
- Multiple applications from same IP flagging

- Liveness detection to prevent photo spoofing
- Document legitimacy scoring
- Anomaly alerts to admin panel

Infrastructure Security

- Cloudflare DDoS protection
- Bot protection and rate limiting at edge
- Automated daily backups (database + files)
- Disaster recovery: point-in-time recovery for PostgreSQL
- System health monitoring with alerts (uptime, error rates, API latency)
- Centralized logging (structured JSON logs)

8. Deployment Architecture

Container Architecture (Docker)

Container	Technology	Purpose
banki-web	Next.js	Customer app + Admin panel + API routes
banki-face	Python/FastAPI	Face matching microservice
banki-db	PostgreSQL 16	Primary database
banki-redis	Redis 7	Cache, sessions, rate limiting
banki-queue	BullMQ Worker	Background jobs (PDF, SMS, email, cleanup)
banki-storage	MinIO	S3-compatible file storage (self-hosted option)
banki-nginx	Nginx	Reverse proxy, SSL termination, subdomain routing

Scaling Strategy

- Horizontal scaling: multiple banki-web containers behind Nginx load balancer
- Database: read replicas for analytics queries
- Redis cluster for session distribution across web containers
- Queue workers: scale independently based on job volume
- File storage: MinIO for self-hosted, AWS S3 for cloud deployment
- Target: 100+ concurrent users per instance, horizontally scalable to 1000+

CI/CD Pipeline (GitHub Actions)

- On push to main: lint → test → build → deploy to staging
- On release tag: deploy to production
- Automated database migrations
- Docker image building and registry push
- Health check after deployment

Monitoring & Alerting

- Application health endpoints (/health, /ready)
- Uptime monitoring with alerts (email/SMS)
- Error rate tracking and alerting

- API response time monitoring
- Database connection pool monitoring
- Gemini API usage and cost tracking
- Storage usage tracking per tenant

9. Project Folder Structure

The project follows a monorepo structure with clear separation between frontend, backend, and services.

```
banki/
├── docker-compose.yml
├── docker-compose.prod.yml
├── .github/workflows/
│   ├── ci.yml
│   └── deploy.yml
├── nginx/
│   ├── nginx.conf
│   └── ssl/
├── web/                          (Next.js monorepo)
│   ├── package.json
│   ├── next.config.js
│   ├── tailwind.config.js
│   ├── prisma/
│   │   ├── schema.prisma
│   │   ├── migrations/
│   │   └── seed.ts
│   ├── src/
│   │   ├── app/                    (Next.js App Router)
│   │   │   ├── (customer)/         Customer-facing routes
│   │   │   │   ├── [tenant]/        Tenant-specific pages
│   │   │   │   │   ├── page.tsx      Landing page
│   │   │   │   │   ├── apply/        Voice KYC app
│   │   │   │   │   └── status/       Check status
│   │   │   ├── (admin)/            Admin panel routes
│   │   │   │   ├── dashboard/
│   │   │   │   ├── applications/
│   │   │   │   ├── products/
│   │   │   │   ├── flows/
│   │   │   │   ├── settings/
│   │   │   │   ├── users/
│   │   │   │   └── analytics/
│   │   │   ├── (super-admin)/      Super admin routes
│   │   │   │   ├── tenants/
│   │   │   │   ├── billing/
│   │   │   │   └── system/
│   │   │   └── api/                API routes
│   │   │       ├── v1/
│   │   │       │   ├── auth/
│   │   │       │   ├── applications/
│   │   │       │   ├── products/
│   │   │       │   ├── flows/
│   │   │       │   ├── tenants/
│   │   │       │   ├── notifications/
│   │   │       │   ├── analytics/
│   │   │       │   ├── voice/
```

```
│   │   │       │   ├── ai/
│   │   │       │   └── audit/
│   │   │       └── webhooks/
│   │   ├── components/
│   │   │   ├── customer/           Customer UI components
│   │   │   │   ├── VoiceChat.tsx
│   │   │   │   ├── Transcript.tsx
│   │   │   │   ├── CameraCapture.tsx
│   │   │   │   ├── LivenessCheck.tsx
│   │   │   │   ├── ProductCards.tsx
│   │   │   │   ├── EditableForm.tsx
│   │   │   │   └── ProgressSidebar.tsx
│   │   │   ├── admin/              Admin UI components
│   │   │   │   ├── Dashboard/
│   │   │   │   ├── ReviewQueue/
│   │   │   │   ├── FlowEditor/
│   │   │   │   ├── ProductManager/
│   │   │   │   └── BrandingEditor/
│   │   │   └── shared/             Shared components
│   │   ├── lib/
│   │   │   ├── gemini/             Gemini API integration
│   │   │   ├── voice/              Voice streaming logic
│   │   │   ├── auth/               Auth utilities
│   │   │   ├── db/                 Prisma client
│   │   │   ├── storage/            S3/MinIO client
│   │   │   ├── sms/                text.lk integration
│   │   │   ├── email/              SendGrid/SMTP
│   │   │   ├── pdf/                PDF generation
│   │   │   ├── encryption/         AES-256 utilities
│   │   │   ├── flow-engine/        Flow execution logic
│   │   │   └── liveness/           TF.js liveness detection
│   │   ├── hooks/                Custom React hooks
│   │   ├── stores/               Zustand stores
│   │   ├── types/                TypeScript types
│   │   └── utils/                Utility functions
│   └── public/
│       ├── icons/
│       ├── sounds/               Chimes, notifications
│       └── manifest.json         PWA manifest
├── face-service/                 Python microservice
│   ├── Dockerfile
│   ├── requirements.txt
│   ├── main.py               FastAPI entry
│   ├── face_match.py         Face comparison logic
│   ├── id_validator.py       Document validation
│   └── models/               Pre-trained models
├── queue-worker/                 Background job worker
│   ├── Dockerfile
│   ├── jobs/
│   │   ├── pdf-generator.ts
│   │   ├── sms-sender.ts
│   │   ├── email-sender.ts
│   │   ├── data-cleanup.ts
│   │   └── archive.ts
│   └── index.ts
├── scripts/
```

```
│   ├── backup.sh
│   ├── restore.sh
│   └── seed-templates.ts
└── docs/
    ├── architecture.md
    ├── api-reference.md
    └── deployment-guide.md
```

10. Implementation Phases & Timeline

The project is divided into 6 phases, each building on the previous. Estimated total timeline: 16–20 weeks for a team of 2–3 developers.

Phase 1: Foundation (Weeks 1–3)

Phase 1 Deliverables

- Project setup: Next.js, TypeScript, Tailwind, Prisma, Docker Compose
- Database schema: all core tables with migrations
- Authentication system: login, register, JWT, 2FA, SSO, password policies
- Multi-tenant foundation: tenant model, subdomain routing, row-level security
- Role-based access control (RBAC) implementation
- Basic admin panel layout: sidebar navigation, dashboard skeleton
- Super admin panel: tenant CRUD, billing page skeleton
- CI/CD pipeline setup with GitHub Actions

Phase 2: Admin Panel Core (Weeks 4–6)

Phase 2 Deliverables

- Dashboard with real-time stats, charts, and filters
- Product management: CRUD, eligibility rules, images, T&C
- User management: invite staff, assign roles, manage access
- Settings page: API keys, branding, SMS/email templates, thresholds
- Branding editor: logo upload, colors, fonts, preview
- Notification system (in-app): real-time alerts via Socket.IO
- Audit log viewer with filters
- Export functionality (CSV/Excel/PDF)

Phase 3: Flow Editor (Weeks 7–9)

Phase 3 Deliverables

- Node-based visual editor using React Flow
- All 12 node types with configuration panels
- Conditional branching logic
- Edge connections with validation
- Version history and rollback

- Flow duplication and templates
- Publish/unpublish workflow
- Preview/simulate conversation mode
- Timeout logic per node
- Webhook and delay nodes

Phase 4: Voice AI & KYC Engine (Weeks 10–13)

Phase 4 Deliverables

- Gemini Multimodal Live API integration (real-time voice)
- WebSocket voice streaming pipeline
- Customer-facing PWA: landing page, voice interface, progress sidebar
- Trilingual support: English, Sinhala, Tamil (language detection)
- Camera capture with document guide overlay and auto-detect
- Gemini Vision: ID extraction, document type recognition, legitimacy checks
- Sri Lankan NIC parsing: old/new format, age, gender, checksum validation
- Python face matching microservice (FastAPI + face_recognition)
- TensorFlow.js liveness detection (blink, head turn, smile)
- Editable data confirmation screen with pencil icons
- Flow engine: execute admin-built flow as voice conversation
- Session management: save/resume interrupted sessions
- Fallback to text chat

Phase 5: Products, PDF & Notifications (Weeks 14–16)

Phase 5 Deliverables

- Product recommendation engine (Gemini + eligibility rules)
- Visual product cards with comparison feature
- PDF generation: branded, QR code, digital hash, password protection
- Customer ID auto-generation (BANKI-YYYY-NNNNN format)
- text.lk SMS integration: OTP, customer ID, status updates
- Email integration: SendGrid/SMTP, welcome email with PDF
- Review queue: full workflow (approve/reject/re-submit/assign/notes/SLA)
- Bulk actions in review queue
- Customer status page: check application status
- Returning customer recognition

Phase 6: Polish, Security & Launch (Weeks 17–20)

Phase 6 Deliverables

- AES-256 encryption at rest for all sensitive data
- Anti-fraud: duplicate NIC detection, IP flagging, anomaly alerts
- Data deletion requests feature
- Data retention policies and automated cleanup
- Backup and disaster recovery setup
- Cloudflare integration: DDoS protection, CDN
- System health monitoring and alerting
- Conversion funnel analytics
- AI performance metrics dashboard
- Monthly report generation
- Accessibility mode (larger fonts, high contrast)
- Dark mode
- PWA optimization (offline indicator, install prompt)
- Load testing and performance optimization
- Security audit and penetration testing
- Documentation: API reference, deployment guide
- Production deployment

11. Risk Assessment & Mitigation

Risk	Impact	Probability	Mitigation
Gemini API downtime	High	Low	Fallback to text-based form; queue and retry; monitor API status
Face matching inaccuracy	High	Medium	Adjustable threshold; admin manual review queue; continuous model improvement
Sri Lankan NIC format changes	Medium	Low	Configurable extraction rules; Gemini Vision adapts to new formats
High concurrent voice sessions	High	Medium	WebSocket connection pooling; horizontal scaling; session queuing
Data breach	Critical	Low	Encryption at rest/transit; 2FA; audit logs; regular security audits; penetration testing
text.lk SMS delivery failures	Medium	Medium	Retry logic; fallback to email; delivery status tracking; admin alerts
Large file storage costs	Medium	Medium	Automatic archival; compression; configurable retention; tenant storage limits
Browser compatibility (voice)	Medium	Medium	Feature detection; fallback to text chat; supported browser list
Regulatory changes	High	Low	Configurable flows; audit trail; data export; compliance documentation
Tenant data isolation breach	Critical	Low	PostgreSQL RLS; application-level checks; security audits; penetration testing

End of Architecture Document

Banki v1.0 — February 2026

Ready for implementation upon approval.