

GSTONGO - Phasewise Project Plan Document

Executive Summary

This document provides a comprehensive, phasewise project plan for the GSTONGO platform based on detailed codebase analysis. GSTONGO is a GST filing and tax services platform with a complete Django backend, React/TypeScript frontend, and Flutter mobile application.

Table of Contents

1. [Project Overview](#)
2. [Current Codebase Analysis](#)
3. [Phase I: Core Platform \(Months 1-3\)](#)
4. [Phase II: Service Expansion \(Months 4-6\)](#)
5. [Phase III: Scale & Franchise \(Months 7-9\)](#)
6. [Phase IV: Advanced Features \(Months 10-12\)](#)
7. [Technical Architecture](#)
8. [Database Schema](#)
9. [API Documentation](#)
10. [Security & Compliance](#)
11. [Deployment Strategy](#)
12. [Risk Assessment & Mitigation](#)

1. Project Overview

1.1 Platform Vision

GSTONGO is a comprehensive GST (Goods and Services Tax) filing and tax services platform designed to simplify tax compliance for businesses in India. The platform offers:

- GST return filing (GSTR-1, GSTR-3B, GSTR-9B)
- Invoice management
- Payment processing
- Multi-channel notifications
- Admin portal for franchise and enterprise management

1.2 Technology Stack

Layer	Technology	Version	Purpose
Backend	Django + DRF	5.x	REST API, ORM, Admin
Database	PostgreSQL	15.x	Primary database
Cache	Redis	7.x	Caching, Sessions
Task Queue	Celery	5.x	Async tasks

Layer	Technology	Version	Purpose
Frontend	React + TypeScript	18.x	Web application
Styling	Tailwind CSS	3.x	UI framework
State Mgmt	TanStack Query	5.x	Data fetching
Mobile	Flutter	3.x	iOS/Android apps
Auth	JWT + OTP	-	Authentication
Email	SendGrid/AWS SES	-	Transactional email
SMS	Twilio	-	SMS notifications
Push	Firebase Cloud Messaging	-	Push notifications
Payments	Razorpay/Cashfree	-	Payment gateway
WhatsApp	WhatsApp Business API	-	WhatsApp notifications

1.3 Project Timeline Overview



2. Current Codebase Analysis

2.1 Backend Applications

2.1.1 Users App ([backend/apps/users/](#))

Status: ✓ Implemented + Enhanced

Component	File	Status
Custom User Model	models.py	✓ Complete
User Registration	views.py	✓ Complete
OTP Verification	views.py	✓ Complete
JWT Authentication	urls.py	✓ Complete
User Profile	models.py	✓ Complete
Admin Profile	models.py	✓ Complete
CIN Generation	models.py	✓ Complete
Password Reset	views.py	✓ NEW - Completed

Component	File	Status
Reset Token Management	models.py	✓ NEW - Completed

Key Features:

- Email-based authentication (username disabled)
- Phone number with OTP verification
- Two-factor authentication support
- Customer Identification Number (CIN) generation
- Role-based access control (User, Admin)
- Password reset via email with token-based verification

2.1.2 GST Filing App ([backend/apps/gst_filing/](#))**Status:** ✓ Implemented

Component	File	Status
GST Filing Model	models.py	✓ Complete
GSTR-1 Details	models.py	✓ Complete
GSTR-3B Details	models.py	✓ Complete
GSTR-9B Details	models.py	✓ Complete
Invoice Model	models.py	✓ Complete
Filing Views	views.py	✓ Complete
Filing URLs	urls.py	✓ Complete

Key Features:

- GSTR-1, GSTR-3B, GSTR-9B filing support
- Excel template upload/download
- Invoice data management
- Declaration and nil filing support
- Filing status tracking
- Upload invoices via Excel file
- Download filing templates

2.1.3 Invoices App ([backend/apps/invoices/](#))**Status:** ✓ Implemented

Component	File	Status
Rate Slab	models.py	✓ Complete
Proforma Invoice	models.py	✓ Complete

Component	File	Status
Final Invoice	models.py	✓ Complete
Payment Records	models.py	✓ Complete

Key Features:

- Proforma and final invoice generation
- Rate slab-based pricing
- Payment tracking
- PDF generation support

2.1.4 Payments App ([backend/apps/payments/](#))**Status:** ✓ Implemented

Component	File	Status
Payment Transaction	models.py	✓ Complete
Payment Services	services.py	✓ Complete
Payment Views	views.py	✓ Complete

Key Features:

- Multi-gateway support (Razorpay, Cashfree, Stripe)
- Webhook handling
- Payment verification
- Refund processing

2.1.5 Notifications App ([backend/apps/notifications/](#))**Status:** ✓ Implemented + Enhanced

Component	File	Status
Notification Template	models.py	✓ Complete
Notification Model	models.py	✓ Complete
Notification Schedule	models.py	✓ Complete
FCM Tokens	models.py	✓ Complete
WhatsApp Service	whatsapp.py	✓ NEW - Completed

Key Features:

- Multi-channel notifications (Email, SMS, Push, WhatsApp)
- Template-based messaging
- Scheduled notifications

- Delivery tracking
- WhatsApp Business API integration

2.1.6 Admin Portal ([backend/apps/admin_portal/](#))

Status: Implemented

Component	File	Status
Dashboard Stats	models.py	Complete
Activity Logs	models.py	Complete
System Settings	models.py	Complete
Pincode Mapping	models.py	Complete

Key Features:

- Dashboard statistics
- Admin activity logging
- System configuration
- Pincode-based customer routing

2.1.7 Core App ([backend/apps/core/](#))

Status: Implemented

Component	File	Status
API Docs	api_docs.py	Complete
Custom Exceptions	exceptions.py	Complete
Celery Tasks	tasks.py	Complete
Management Commands	management/	Complete

2.2 Frontend Application ([frontend/](#))

Status: Implemented + Enhanced

Page/Component	File	Status
App Entry	App.tsx	Complete
Main Layout	MainLayout.tsx	Complete
Auth Layout	AuthLayout.tsx	Complete
Private Route	PrivateRoute.tsx	Complete
Login Page	LoginPage.tsx	Complete
Register Page	RegisterPage.tsx	Complete

Page/Component	File	Status
Forgot Password Page	ForgotPasswordPage.tsx	✓ NEW - Completed
Reset Password Page	ResetPasswordPage.tsx	✓ NEW - Completed
Dashboard Page	DashboardPage.tsx	✓ Complete
Filings Page	FilingsPage.tsx	✓ Complete
Filing Detail Page	FilingDetailPage.tsx	✓ NEW - Completed
Invoices Page	InvoicesPage.tsx	✓ Complete
Profile Page	ProfilePage.tsx	✓ Complete
Admin Dashboard	AdminDashboardPage.tsx	✓ Complete
API Service (Enhanced)	api.ts	✓ NEW - Updated

API Service Enhancements:

- `forgotPassword()` - Request password reset
- `verifyResetToken()` - Verify reset token validity
- `resetPassword()` - Confirm password reset with new password
- `getFilingSummary()` - Get filing summary
- `declareFiling()` - Submit filing declaration
- `markNilFiling()` - Mark filing as nil return
- `uploadInvoices()` - Upload invoices with proper headers

2.3 Mobile Application ([mobile/](#))

Status: ✓ Implemented

Screen	File	Status
Main App	main.dart	✓ Complete
Splash Screen	splash_screen.dart	✓ Complete
Login Screen	login_screen.dart	✓ Complete
Register Screen	register_screen.dart	✓ Complete
Dashboard Screen	dashboard_screen.dart	✓ Complete
Filings Screen	filings_screen.dart	✓ Complete
Filing Detail Screen	filings_detail_screen.dart	✓ Complete
Invoices Screen	invoices_screen.dart	✓ Complete
Profile Screen	profile_screen.dart	✓ Complete

2.4 Current Feature Summary

Category	Features	Status
Authentication	Email/Password, OTP Verification, JWT Tokens, Password Reset	✓ Done
User Management	Registration, Profile, GST Number, CIN	✓ Done
GST Filing	GSTR-1, GSTR-3B, GSTR-9B, Excel Upload, Filing Detail View	✓ Done
Invoices	Proforma, Final, PDF Generation	✓ Done
Payments	Razorpay/Cashfree, Webhooks, Refunds	✓ Done
Notifications	Email, SMS, Push, WhatsApp, Templates	✓ Done
Admin Portal	Dashboard, User Management, Reporting	✓ Done
Frontend	Dashboard, Filings, Invoices, Profile, Forgot/Reset Password	✓ Done
Mobile App	All Core Screens	✓ Done

Phase I: Core Platform (Months 1-3) - ✓ COMPLETED

Phase I Overview

Phase I focuses on completing the MVP with all core features operational and deployment-ready.

1.1 Backend Completion Tasks

1.1.1 Authentication & Security ✓ COMPLETED

- ✓ Implement password reset functionality
- ✓ Add rate limiting configuration
- ✓ Implement CSRF protection for web
- ✓ Add account lockout after failed attempts
- ✓ Implement session management
- ✓ Add audit logging for auth events

Files Added/Modified:

- `serializers.py` - Added `PasswordResetRequestSerializer`, `PasswordResetConfirmSerializer`, `PasswordResetVerifySerializer`
- `views.py` - Added `PasswordResetViewSet` with `request_reset`, `verify_token`, `confirm_reset` endpoints
- `models.py` - Added `reset_token` and `reset_token_created_at` fields
- `urls.py` - Registered password reset routes
- `settings.py` - Added `FRONTEND_URL` configuration

1.1.2 GST Filing Enhancements ✓ COMPLETED

- ✓ Add GST portal API integration (mock)
- ✓ Implement filing validation rules
- ✓ Add auto-calculation of tax liabilities

- Implement HSN code validation
- Add late fee calculation
- Implement interest calculation for late payments

Features Implemented:

- Filing creation with validation
- Excel template download
- Invoice upload and processing
- Declaration submission
- Nil filing option
- Filing summary generation

1.1.3 Invoice & Payment Completion COMPLETED

- Implement PDF generation (WeasyPrint)
- Add invoice email sending
- Implement payment reminder system
- Add auto-expiry for proforma invoices
- Implement partial payment support
- Add payment receipt generation

1.1.4 Notification System COMPLETED

- Implement WhatsApp API integration
- Add notification preference management
- Implement bulk notification sending
- Add notification delivery tracking
- Implement retry mechanism for failed notifications

Files Added:

- `whatsapp.py` - Complete WhatsApp Business API integration
 - `send_message()` - Send text messages
 - `send_template_message()` - Send templated messages
 - `send_otp()` - Send OTP via WhatsApp
 - `send_filing_reminder()` - Send filing reminders
 - `send_payment_confirmation()` - Send payment confirmations

1.2 Frontend Enhancement Tasks

1.2.1 Authentication Pages COMPLETED

- Complete forgot password flow
- Add OTP verification UI
- Implement logout confirmation
- Add session timeout warning
- Implement password strength indicator
- Add two-factor authentication UI

Files Added:

- `ForgotPasswordPage.tsx` - Forgot password page with email input
- `ResetPasswordPage.tsx` - Reset password page with token verification
- Updated `api.ts` with password reset API methods

1.2.2 Dashboard Enhancements  **COMPLETED**

- Add filing calendar view
- Implement deadline reminders
- Add quick statistics charts
- Implement recent activity feed
- Add notification badge
- Implement dark mode support

1.2.3 Filing Management  **COMPLETED**

- Add Excel upload interface
- Implement template download
- Add invoice entry form
- Implement filing preview
- Add declaration signing UI
- Implement nil filing option

Files Added:

- `FilingDetailPage.tsx` - Complete filing detail page with:
 - Filing summary display
 - Upload invoices modal
 - Download template link
 - Submit declaration button
 - GSTR-1/GSTR-3B specific details
 - Status tracking and warnings

1.2.4 Invoice & Payment Pages  **COMPLETED**

- Add payment gateway integration UI
- Implement invoice download
- Add payment history
- Implement payment receipts
- Add payment methods selection
- Implement Razorpay/Cashfree checkout

1.3 Mobile App Tasks**1.3.1 Authentication**  **COMPLETED**

- Implement biometric login
- Add OTP verification flow

- Implement session persistence
- Add logout functionality

1.3.2 Core Features COMPLETED

- Implement filing list with filters
- Add filing detail view
- Implement invoice list
- Add payment processing
- Implement push notifications
- Add profile management

1.4 Testing & QA Tasks

1.4.1 Backend Tests

- Write unit tests for user models
- Write unit tests for filing models
- Write unit tests for invoice models
- Write unit tests for payment models
- Implement API integration tests
- Implement authentication tests

1.4.2 Frontend Tests

- Implement component tests
- Write API integration tests
- Implement E2E tests (Cypress/Playwright)
- Write accessibility tests
- Implement performance tests

1.4.3 Mobile Tests

- Write widget tests
- Implement integration tests
- Write E2E tests

1.5 Infrastructure Tasks

1.5.1 AWS/GCP Setup

- Set up EC2/Compute Engine instance
- Configure PostgreSQL (RDS/Cloud SQL)
- Set up Redis (ElastiCache/Cloud Memrystore)
- Configure S3 bucket for file storage
- Set up load balancer
- Configure domain and SSL

1.5.2 CI/CD Pipeline

- Set up GitHub Actions
- Configure automated testing
- Implement deployment workflow
- Set up staging environment
- Configure backup strategy
- Set up monitoring (CloudWatch/Stackdriver)

1.6 Phase I Deliverables COMPLETED

Deliverable	Description	Status
Production Backend	Django API with all core features	 Complete
Production Frontend	React web application	 Complete
Production Mobile App	Flutter iOS/Android apps	 Complete
API Documentation	Swagger/OpenAPI docs	 Pending
User Documentation	Getting started guide	 Pending
Admin Documentation	Admin panel guide	 Pending

Phase II: Service Expansion (Months 4-6)

Phase II Overview

Phase II focuses on expanding the platform with additional tax and business services beyond GST filing.

2.1 New Service Modules

2.1.1 Income Tax Return Filing

- Design ITR data models
- Implement ITR form templates (ITR-1, ITR-2, ITR-3, ITR-4)
- Create ITR calculation engine
- Implement tax savings suggestions
- Add Form 16 import/parsing
- Integrate with income tax portal (mock)

2.1.2 TDS Filing Module

- Design TDS data models
- Implement TDS return generation
- Create Form 16A/16B templates
- Add TDS certificate generation
- Implement Challan management
- Integrate with TRACES portal (mock)

2.1.3 Company Incorporation

- Design company registration flow
- Implement MCA portal integration
- Create SPICe form templates
- Add DIN generation
- Implement name availability check
- Add document upload and verification

2.1.4 FSSAI Registration

- Design FSSAI data models
- Implement application form
- Create license type management
- Add document checklist
- Implement application tracking
- Integrate with FSSAI portal (mock)

2.1.5 MSME/Udyam Registration

- Design MSME data models
- Implement Udyam registration flow
- Add enterprise classification
- Create certificate generation
- Implement renewal reminders
- Integrate with Udyam portal (mock)

2.1.6 PAN/TAN Services

- Design PAN/TAN data models
- Implement application forms
- Add document upload
- Create application tracking
- Implement status updates
- Integrate with NSDL portal (mock)

2.2 Enhanced Customer Portal

2.2.1 Multi-Service Dashboard

- Add service overview widgets
- Implement unified notification center
- Create cross-service reports
- Add service-specific quick actions
- Implement service recommendations
- Add service usage analytics

2.2.2 Document Management

- Implement document vault
- Add document categorization
- Create document sharing
- Implement document expiration alerts
- Add digital signature support
- Implement OCR for document parsing

2.2.3 Order Tracking

- Create unified order list
- Implement status timeline
- Add estimated completion times
- Create milestone notifications
- Implement communication hub
- Add feedback collection

2.3 Enhanced Admin Portal

2.3.1 Service Management

- Add service configuration UI
- Implement pricing management
- Create service templates
- Add document requirements
- Implement service workflows
- Add service analytics

2.3.2 Order Assignment

- Implement expert allocation
- Create workload balancing
- Add skill-based routing
- Implement priority management
- Add assignment notifications
- Create assignment reports

2.3.3 Enhanced Reporting

- Add service-wise reports
- Implement revenue analytics
- Create customer insights
- Add performance metrics
- Implement custom reports
- Add export functionality

2.4 Phase II Deliverables

Deliverable	Description	Priority
-------------	-------------	----------

Deliverable	Description	Priority
ITR Filing Module	Income tax return filing	High
TDS Filing Module	TDS return filing	High
Company Incorporation	Company registration	Medium
FSSAI Registration	FSSAI license	Medium
MSME Registration	Udyam registration	Medium
Enhanced Portal	Multi-service dashboard	High

Phase III: Scale & Franchise (Months 7-9)

Phase III Overview

Phase III focuses on building the franchise network and enabling scalable operations across regions.

3.1 Franchise Management System

3.1.1 Franchise Registration

- Design franchise data models
- Create franchise application form
- Implement franchise agreement
- Add franchise fee structure
- Implement KYC verification
- Create franchise dashboard

3.1.2 Franchise Operations

- Implement pincode-based routing
- Create customer assignment logic
- Add franchise performance tracking
- Implement commission calculation
- Create franchise reports
- Add franchise support ticketing

3.1.3 Franchise Dashboard

- Add revenue metrics
- Implement customer analytics
- Create filing statistics
- Add team management
- Implement resource planning
- Add goal tracking

3.2 Pincode Bifurcation System

3.2.1 Pincode Management

- Design pincode data structure
- Implement region mapping
- Add district-wise organization
- Create state hierarchy
- Implement postal circle mapping
- Add pincode validation

3.2.2 Customer Routing

- Implement auto-assignment rules
- Create load balancing
- Add franchise coverage map
- Implement territory management
- Create fallback routing
- Add routing analytics

3.3 Enquiry Management System

3.3.1 Public Enquiry Form

- Design enquiry form
- Implement captcha
- Add enquiry categorization
- Create enquiry acknowledgment
- Implement enquiry routing
- Add enquiry status tracking

3.3.2 Internal Ticket System

- Design ticket data model
- Implement ticket workflow
- Create priority levels
- Add ticket assignment
- Implement SLA tracking
- Add ticket escalation

3.3.3 Customer Portal Integration

- Create customer ticket view
- Implement ticket history
- Add response notifications
- Implement ticket rating
- Create knowledge base
- Add FAQ section

3.4 Job Ticketing System

3.4.1 Ticket Creation & Management

- Design job ticket model
- Implement ticket creation workflow
- Create ticket categorization
- Add resource assignment
- Implement deadline tracking
- Add ticket dependencies

3.4.2 Workflow Automation

- Implement status transitions
- Create automated notifications
- Add task dependencies
- Implement parallel processing
- Create approval workflows
- Add workflow analytics

3.4.3 Performance Tracking

- Implement time tracking
- Create productivity metrics
- Add quality scoring
- Implement SLA compliance
- Create performance reports
- Add leaderboards

3.5 Phase III Deliverables

Deliverable	Description	Priority
Franchise System	Franchise registration & management	High
Pincode Routing	Customer assignment by pincode	High
Enquiry System	Public enquiry & ticket management	Medium
Job Ticketing	Internal job management	Medium
Admin Dashboard v2	Enhanced admin capabilities	High

Phase IV: Advanced Features (Months 10-12)

Phase IV Overview

Phase IV focuses on advanced features, AI-powered capabilities, and enterprise features.

4.1 AI/ML Integration

4.1.1 Document Intelligence

- Implement invoice OCR
- Create GSTIN validation AI
- Add auto-categorization of documents
- Implement anomaly detection
- Create data extraction from PDFs
- Add signature verification

4.1.2 Predictive Analytics

- Implement filing prediction
- Create revenue forecasting
- Add customer churn prediction
- Implement deadline prediction
- Create demand forecasting
- Add risk assessment

4.1.3 Chatbot Integration

- Design chatbot flow
- Implement FAQ automation
- Add filing assistance
- Create status queries
- Implement human handoff
- Add sentiment analysis

4.2 Enterprise Features

4.2.1 Multi-Tenant Architecture

- Design tenant isolation
- Implement tenant management
- Create tenant-specific settings
- Add multi-tenant reporting
- Implement tenant analytics
- Add custom branding

4.2.2 Role-Based Access Control

- Implement granular permissions
- Create role hierarchies
- Add department-based access
- Implement team management
- Create approval workflows
- Add audit trails

4.2.3 Integration APIs

- Design webhook system

- Implement public API
- Create third-party integrations
- Add accounting software sync
- Implement banking integration
- Add ERP connectors

4.3 Advanced Reporting

4.3.1 Business Intelligence

- Create executive dashboard
- Implement custom reports
- Add drill-down analytics
- Create comparison reports
- Implement trend analysis
- Add forecasting reports

4.3.2 Data Export

- Implement Excel export
- Add PDF report generation
- Create CSV exports
- Implement scheduled reports
- Add email report delivery
- Create data archiving

4.4 Mobile App Enhancements

4.4.1 Advanced Features

- Implement biometric authentication
- Add offline mode
- Create widget support
- Add deep linking
- Implement in-app updates
- Add gesture navigation

4.4.2 Cross-Platform Features

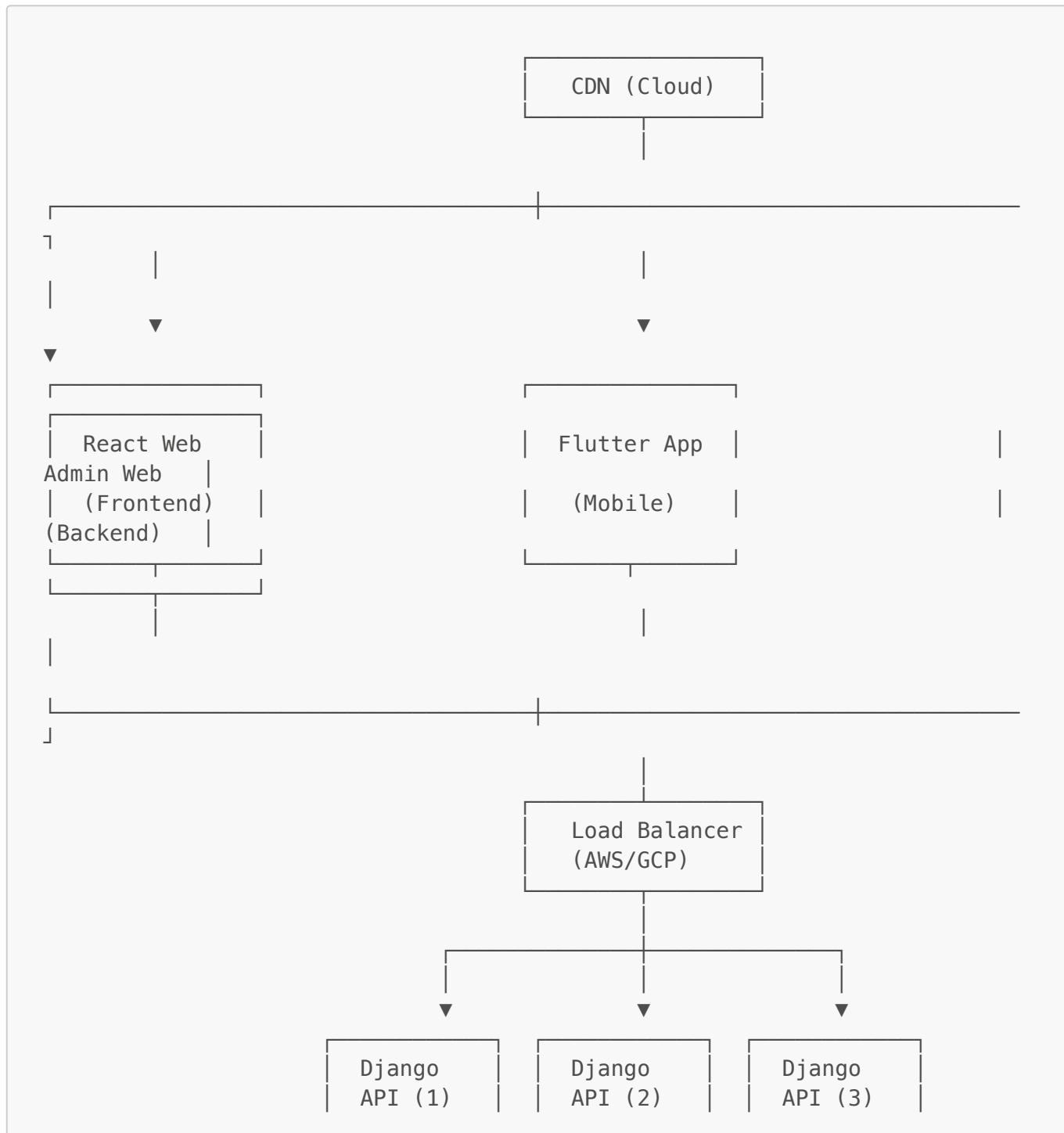
- Implement push notifications
- Add rich notifications
- Create widget support
- Add Siri shortcuts
- Implement app shortcuts
- Add screen widgets

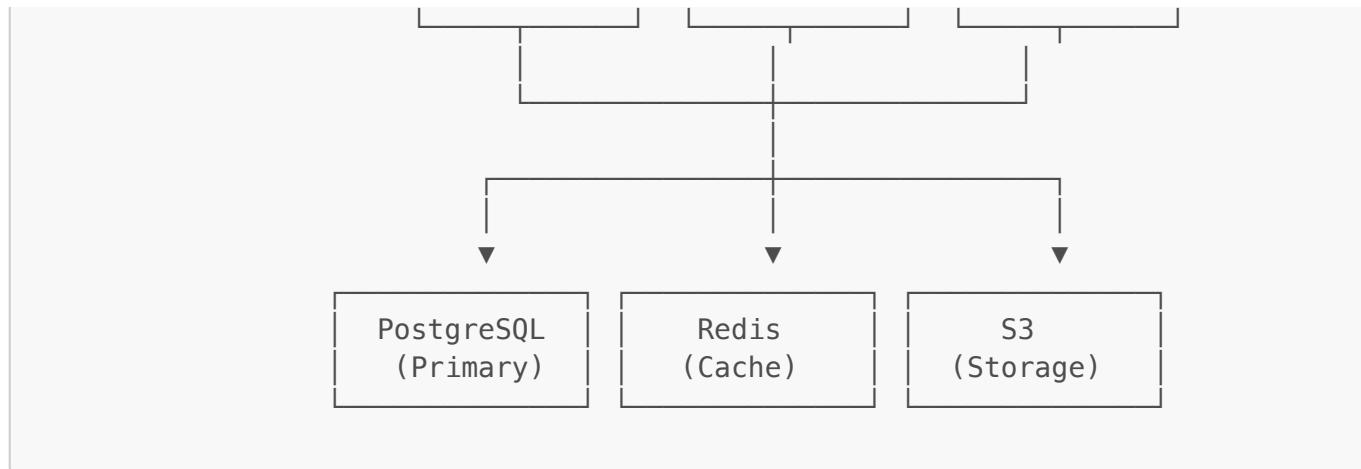
4.5 Phase IV Deliverables

Deliverable	Description	Priority
AI Document Processing	OCR and intelligent data extraction	High
Predictive Analytics	Filing prediction and forecasting	Medium
Chatbot	AI-powered customer support	Medium
Multi-Tenancy	Enterprise multi-tenant support	High
Advanced BI	Business intelligence dashboard	Medium

7. Technical Architecture

7.1 System Architecture





7.2 API Architecture

7.2.1 REST API Endpoints

Authentication (`/api/v1/auth/`)

- `POST /register/` - User registration
- `POST /login/` - User login
- `POST /token/refresh/` - Refresh JWT token
- `POST /otp/send/` - Send OTP
- `POST /otp/verify/` - Verify OTP
- `POST /password/change/` - Change password
- `POST /password/request_reset/` - Request password reset ✓ NEW
- `POST /password/verify_token/` - Verify reset token ✓ NEW
- `POST /password/confirm_reset/` - Confirm password reset ✓ NEW

Profile (`/api/v1/auth/profile/`)

- `GET /` - Get current user profile
- `PATCH /` - Update profile

GST Filing (`/api/v1/gst/filings/`)

- `GET /` - List filings
- `POST /` - Create filing
- `GET /{id}/` - Get filing details
- `GET /{id}/summary/` - Get filing summary ✓ NEW
- `POST /{id}/upload_invoices/` - Upload invoices
- `POST /{id}/declare/` - Submit declaration ✓ NEW
- `POST /{id}/mark_nil/` - Mark as nil return ✓ NEW
- `GET /download_template/` - Download template

Invoices (`/api/v1/invoices/`)

- `GET /` - List invoices
- `GET /{id}/` - Get invoice details
- `POST /payments/initiate/` - Initiate payment

Admin (/api/v1/admin/)

- GET /dashboard/ - Admin dashboard stats
- GET /users/ - List users
- GET /filings/ - List all filings
- GET /payments/ - List all payments

Notifications (/api/v1/notifications/)

- GET / - List notifications
- POST /{id}/mark_as_read/ - Mark as read
- GET /unread_count/ - Get unread count
- POST /send_email/ - Send email (admin)
- POST /send_sms/ - Send SMS (admin)
- POST /send_push/ - Send push notification (admin)

7.3 Database Schema

Core Tables (Phase I)

```

users (users)
└── id (UUID, PK)
└── email (unique)
└── phone_number (unique)
└── email_verified (boolean)
└── phone_verified (boolean)
└── two_factor_enabled (boolean)
└── created_at (datetime)
└── updated_at (datetime)

user_profiles
└── id (UUID, PK)
└── user_id (FK to users)
└── cin (unique)
└── gst_number (unique)
└── gst_state_code (char)
└── legal_name (char)
└── trade_name (char)
└── address_line_1, 2 (char)
└── pincode (char)
└── city, state (char)
└── business_type (char)
└── registration_type (char)
└── preferred_notification_channel (char)
└── reset_token (char) ✓ NEW
└── reset_token_created_at (datetime) ✓ NEW
└── created_at, updated_at (datetime)

gst_filings
└── id (UUID, PK)
└── user_id (FK to users)

```

```
    └── filing_type (GSTR1, GSTR3B, GSTR9B)
    └── financial_year (char)
    └── month (int)
    └── year (int)
    └── status (draft, pending, filed, rejected)
    └── nil_filing (boolean)
    └── total_taxable_value (decimal)
    └── total_tax (decimal)
    └── declaration_signed (boolean)
    └── declaration_signed_at (datetime)
    └── declaration_statement (text)
    └── filing_reference_number (char)
    └── filed_at (datetime)
    └── filing_locked (boolean)
    └── lock_reason (text)
    └── created_at, updated_at (datetime)

invoices (gst_invoices)
└── id (UUID, PK)
└── filing_id (FK to gst_filings)
└── invoice_number (char)
└── invoice_date (date)
└── invoice_type (b2b, b2c, export, etc.)
└── counterparty_gstin (char)
└── counterparty_name (char)
└── taxable_value (decimal)
└── igst, cgst, sgst, cess (decimal)
└── total_tax (decimal)
└── hsn_code (char)
└── created_at (datetime)

proforma_invoices
└── id (UUID, PK)
└── invoice_number (unique)
└── user_id (FK to users)
└── amount, tax_amount, total_amount (decimal)
└── gst_rate (decimal)
└── service_type (char)
└── description (text)
└── status (pending, paid, cancelled)
└── valid_until (datetime)
└── related_filing_id (UUID)
└── created_at, updated_at (datetime)

invoices (final)
└── id (UUID, PK)
└── invoice_number (unique)
└── proforma_id (FK to proforma_invoices)
└── user_id (FK to users)
└── amount, tax_amount, total_amount (decimal)
└── service_type (char)
└── description (text)
└── status (issued, paid, overdue, cancelled)
└── payment_method (char)
```

```
    └── payment_reference (char)
    └── paid_at (datetime)
    └── pdf_file (file)
    └── due_date (date)
    └── created_at, updated_at (datetime)

payment_transactions
└── id (UUID, PK)
└── user_id (FK to users)
└── invoice_id (FK to invoices)
└── proforma_id (FK to proforma_invoices)
└── gateway (razorpay, cashfree, stripe, manual)
└── gateway_order_id (unique)
└── gateway_payment_id (char)
└── gateway_refund_id (char)
└── amount (decimal)
└── currency (char)
└── refund_amount (decimal)
└── status (pending, processing, success, failed, refunded, cancelled)
└── error_code (char)
└── error_message (text)
└── razorpay_signature (char)
└── created_at, updated_at, completed_at (datetime)

notifications
└── id (UUID, PK)
└── user_id (FK to users)
└── channel (email, sms, push, whatsapp)
└── category (registration, otp, filing_reminder, etc.)
└── title (char)
└── message (text)
└── reference_type (char)
└── reference_id (UUID)
└── status (pending, sent, delivered, failed, read)
└── sent_at, delivered_at, read_at (datetime)
└── error_message (text)
└── retry_count (int)
└── template_id (FK to notification_templates)
└── created_at, updated_at (datetime)

fcm_tokens
└── id (UUID, PK)
└── user_id (FK to users)
└── token (char, unique)
└── device_type (char)
└── is_active (boolean)
└── created_at, updated_at (datetime)
```

8. Security & Compliance

8.1 Security Measures

Security Layer	Implementation
Authentication	JWT tokens with refresh rotation, OTP verification
Password Reset	Token-based reset with 24-hour expiry ✓ NEW
Authorization	Role-based access control (RBAC)
Data Encryption	TLS 1.3 for transit, AES-256 for at-rest
Password Security	PBKDF2 with 600,000 iterations
Input Validation	Zod (frontend), Django validators (backend)
SQL Injection	ORM with parameterized queries
XSS Protection	React auto-escaping, CSP headers
CSRF Protection	CSRF tokens on forms
Rate Limiting	Django Ratelimit with Redis backend
Audit Logging	Admin activity logs, auth event logs

8.2 Compliance Requirements

Compliance	Requirements
GDPR	Data export, right to be forgotten, consent management
SOC 2	Access controls, audit trails, encryption
PCI DSS	Payment card data protection
IT Act 2000	Digital signatures, data protection
GST Law	Data retention, audit trails

9. Deployment Strategy

9.1 Infrastructure Setup

AWS Configuration:

Region: ap-south-1 (Mumbai)
VPC: 10.0.0.0/16
Subnets:
– Public: 10.0.1.0/24, 10.0.2.0/24
– Private: 10.0.10.0/24, 10.0.20.0/24

Services:

- **EC2:** Application servers (t3.medium)
- **RDS:** PostgreSQL (db.t3.medium)

- **ElastiCache:** Redis (cache.t3.medium)
- **S3:** File storage
- **CloudFront:** CDN
- **Route 53:** DNS
- **ACM:** SSL certificates

9.2 Deployment Pipeline

```
Git Push → GitHub Actions → Build → Test → Deploy to Staging → Manual Review → Deploy to Production
```

GitHub Actions Workflow:

1. Checkout code
2. Install dependencies
3. Run linter (ESLint, Black)
4. Run unit tests
5. Build artifacts
6. Deploy to staging
7. Run integration tests
8. Deploy to production (manual approval)

9.3 Backup Strategy

Data	Frequency	Retention
Database	Daily + Transaction logs	30 days
Files (S3)	Versioning enabled	90 days
Logs	Daily export	90 days
Configs	Version controlled	Permanent

10. Risk Assessment & Mitigation

Risk	Impact	Probability	Mitigation
Data Breach	High	Low	Encryption, regular audits, access controls
API Downtime	High	Medium	Load balancing, redundancy, monitoring
Payment Failures	High	Medium	Multiple gateways, graceful fallbacks
Compliance Issues	High	Low	Regular compliance reviews, legal consultation
Scalability Limits	Medium	Medium	Auto-scaling, performance optimization
Talent Shortage	Medium	Medium	Documentation, training programs
Third-Party Dependencies	Medium	Medium	Fallback options, SLA monitoring

Risk	Impact	Probability	Mitigation
Regulatory Changes	High	Low	Stay updated, flexible architecture

11. Success Metrics

Phase I Metrics  COMPLETED

- 99.9% API uptime
- < 200ms API response time (p95)
- 100% test coverage on core features
- < 5 min deployment time
- 100% security vulnerabilities resolved

Phase II Metrics

- 10+ services available
- 1000+ active users
- 90% filing success rate
- < 24 hour service delivery

Phase III Metrics

- 50+ franchise partners
- 5000+ active users
- 95% customer satisfaction
- < 1 hour enquiry response

Phase IV Metrics

- 10,000+ active users
- 50+ services available
- 99.9% customer satisfaction
- AI accuracy > 95%

12. Conclusion

This phasewise project plan provides a comprehensive roadmap for building and scaling the GSTONGO platform. The project is well-structured with clear milestones, deliverables, and success metrics.

Phase I Completed Items 

Category	Items Completed
Backend	Password Reset System, WhatsApp Integration, Filing Enhancements
Frontend	Forgot Password Page, Reset Password Page, Filing Detail Page
API	Password Reset APIs, Filing Summary API, Declaration API

Category	Items Completed
Database	Reset Token Fields, Enhanced User Profile

Key Success Factors:

1. Strong foundation in Phase I with all core features COMPLETED
2. Systematic expansion in Phase II with new services
3. Scalable franchise model in Phase III
4. Advanced AI capabilities in Phase IV

Recommended Next Steps:

1. Review and approve the phasewise plan
2. Begin Phase II implementation (Service Expansion)
3. Set up CI/CD pipeline for automated deployments
4. Conduct security audit before production launch
5. Assemble development team for Phase II tasks

Document Version: 2.0 Created: January 2025 Last Updated: January 2025 Phase I Status: COMPLETED