Job Portal Technical Architecture

Complete System Design Document for Unskilled Workers Platform in India

Version: 1.0
Date: December 2024
Project: Job Portal for Unskilled Workers
Target Market: India
Document Type: Complete Technical Design Document

📋 Table of Contents

---

1. Executive Summary

1.1 Project Overview

The Job Portal is a comprehensive mobile-first platform designed specifically for unskilled workers in India. The system connects job seekers with employers while addressing unique challenges such as low digital literacy, limited smartphone capabilities, language barriers, and regional diversity.

1.2 Key Technical Requirements

- **Mobile-First Design:** Optimized for Android devices with 2-4GB RAM
- **Multi-Language Support:** Hindi, English, and 10+ regional languages
- **Offline Capability:** Core functions work without internet connectivity
- **Low Bandwidth Optimization:** Minimal data usage for 2G/3G networks
- **Voice Integration:** Speech-to-text and text-to-speech capabilities
- **Scalability:** Support for 1M+ users across multiple cities
- **Security:** End-to-end encryption and compliance with Indian data protection laws

1.3 Target Performance Metrics

- **Response Time:** <2 seconds for core operations
- **Uptime:** 99.9% availability
- **Concurrent Users:** 50,000+ simultaneous users
- **Data Usage:** <10MB per user per month
- **Battery Optimization:** <5% battery drain per hour of usage
- **Load Capacity:** 10,000+ requests per second

1.4 Business Objectives

- **Year 1:** 100,000 active users, 5,000 job placements/month
- **Year 2:** 500,000 active users, 25,000 job placements/month
- **Revenue Model:** 5-10% commission on successful job placements
- **Break-even:** Month 18 with 5,000 monthly placements

---

2. System Architecture Overview

2.1 High-Level Architecture

```
┌─────────────────────────────────────────────────────────────────────────
│                    CLIENT TIER                    │
├─────────────────────────────┬─────────────────────────────┬─────────────
│ Mobile Apps  │ Web Portal  │     Admin Panel        │
│ (Android/iOS) │ (Employers)  │     (Internal)         │
│              │              │                        │
│ - Job Seekers │ - Job Posting  │ - User Management       │
│ - Voice UI    │ - Applications │ - Analytics             │
│ - Offline Mode │ - Analytics    │ - Content Moderation      │
```

```
└──────────────┴───────────┴──────────────┴───────────┴────────────────────┴
     │       │         │
     └──────────────────────────┼──────────────────────────────────────┘
                │
┌────────────────────────────────────────────────────────────────────────────
│           API GATEWAY TIER                 │
├────────────────────────────────────────────────────────────────────────────
│  - Load Balancing       - Rate Limiting              │
│  - Authentication       - Request Routing            │
│  - API Versioning       - Response Caching           │
│  - SSL Termination      - Request Logging            │
└──────────────────────────────────────────────┬─────────────────────────────
                │
┌────────────────────────────────────────────────────────────────────────────
│           MICROSERVICES TIER                 │
├──────────────────────┬──────────────────────┬──────────────────────┬───────
│ Auth Service │ Job Service  │ User Service │ Notification Svc  │
│           │           │           │              │
│- OTP Login   │- Job CRUD    │- Profiles    │- Push/SMS        │
│- JWT Tokens  │- Search      │- Skills      │- Email           │
│- Sessions    │- Matching    │- Documents   │- In-App          │
│- Refresh     │- Analytics   │- Verification│- Voice Messages  │
├──────────────────────┼──────────────────────┼──────────────────────┼───────
│Payment Svc   │Message Svc   │Location Svc  │ File Service      │
│           │           │           │              │
│- Razorpay    │- Chat        │- GPS/Maps    │- Upload/Download  │
│- UPI         │- Voice Msgs  │- Geocoding   │- Image Processing │
│- Escrow      │- Push Notif  │- Distance    │- Document OCR     │
│- Commission  │- SMS Backup  │- Radius      │- CDN Integration  │
└──────────────────────┴──────────────────────┴──────────────────────┴───────
                │
┌────────────────────────────────────────────────────────────────────────────
│           DATA TIER                  │
├──────────────────────────┬──────────────────────────┬──────────────────────
│  PostgreSQL   │   MongoDB    │     Redis          │
│            │            │                 │
│- User Data     │- Documents     │- Sessions          │
│- Job Posts     │- Analytics     │- Cache             │
│- Applications  │- Logs          │- Queues            │
│- Transactions  │- Files Metadata│- Real-time Data    │
├──────────────────────────┼──────────────────────────┼──────────────────────
│ Elasticsearch  │  File Storage  │   Message Queue      │
│            │            │                 │
│- Job Search    │- AWS S3        │- Bull Queue          │
│- User Search   │- Images        │- Job Processing      │
│- Analytics     │- Documents     │- Email Queue         │
```

```
|- Logging      |- Voice Files  |- Notification Queue      |
└───────────────────────────────┴───────────────────────────────┴──────────────────────────
```

2.2 Architecture Principles

2.2.1 Microservices Architecture

- **Service Independence:** Each service can be developed, deployed, and scaled independently
- **Domain-Driven Design:** Services organized around business capabilities
- **Fault Isolation:** Failure in one service doesn't bring down the entire system
- **Technology Diversity:** Different services can use different technologies as needed
- **Team Autonomy:** Small teams can own entire service lifecycles

2.2.2 API-First Design

- **Contract-First Development:** APIs defined before implementation
- **Comprehensive Documentation:** Swagger/OpenAPI specifications for all endpoints
- **Versioning Strategy:** Backward compatibility with version management
- **Consistent Standards:** Uniform naming conventions and response formats
- **Rate Limiting:** Protection against abuse and ensuring fair usage

2.2.3 Mobile-First Approach

- **Progressive Web App (PWA):** Web app with native-like capabilities
- **Offline-First:** Core functionality available without internet
- **Performance Optimization:** Minimal bundle sizes and lazy loading
- **Device Compatibility:** Support for low-end Android devices
- **Battery Efficiency:** Optimized background processes

2.2.4 Data-Driven Architecture

- **Real-time Analytics:** Immediate insights into user behavior
- **Machine Learning Integration:** AI-powered job matching and recommendations
- **Performance Monitoring:** Continuous optimization based on usage patterns
- **Business Intelligence:** Comprehensive reporting and dashboards
- **A/B Testing:** Data-driven feature development and optimization

---

3. Technology Stack

3.1 Frontend Technologies

3.1.1 Mobile Application (React Native)

Core Framework:

- React Native: 0.72+
- TypeScript: 5.0+
- Metro Bundler: Latest

State Management:

- Redux Toolkit: 1.9+
- RTK Query: Data fetching and caching
- React Redux: 8.0+
- Redux Persist: State persistence

Navigation:

- React Navigation: 6.x
- Stack Navigator: Screen transitions
- Tab Navigator: Bottom tabs
- Drawer Navigator: Side menu

UI Components:

- NativeBase: 3.4+
- React Native Elements: Fallback
- React Native Vector Icons: 10+
- React Native Gesture Handler: 2.12+

Device Features:

- React Native Maps: Location services
- React Native Voice: Speech recognition
- React Native TTS: Text-to-speech
- React Native Camera: Photo capture
- React Native Audio: Voice recording
- React Native Geolocation: GPS access

Offline & Storage:

- SQLite: Local database
- WatermelonDB: Reactive local database
- React Native Async Storage: Key-value storage
- MMKV: Fast key-value storage
- React Native FS: File system access

Push Notifications:

- Firebase Cloud Messaging: Cross-platform
- React Native Notifications: Local notifications
- React Native Push Notification: iOS support

Development Tools:

- Flipper: Debugging
- Reactotron: Development tool
- CodePush: Over-the-air updates
- Sentry: Error tracking
- Jest: Unit testing
- Detox: E2E testing

3.1.2 Web Portal (Employer Dashboard)

Core Framework:

- Next.js: 13+ with App Router
- TypeScript: 5.0+
- React: 18+

Styling:

- Tailwind CSS: 3.3+
- Headless UI: Unstyled components
- Heroicons: Icon library
- Framer Motion: Animations

State Management:

- Zustand: Lightweight state management
- TanStack Query: Server state management

- SWR: Data fetching (alternative)

Authentication:

- NextAuth.js: Authentication library
- JWT: Token handling
- OAuth providers: Google, LinkedIn

Data Visualization:

- Chart.js: Charts and graphs
- D3.js: Custom visualizations
- Recharts: React charts

Forms & Validation:

- React Hook Form: Form handling
- Zod: Schema validation
- Yup: Alternative validation

Development Tools:

- ESLint: Code linting
- Prettier: Code formatting
- Jest: Unit testing
- Cypress: E2E testing
- Storybook: Component documentation

3.1.3 Admin Dashboard

Core Framework:

- React: 18+
- TypeScript: 5.0+
- Vite: Build tool

UI Library:

- Ant Design: 5.0+
- Material-UI: Alternative
- Styled Components: CSS-in-JS

State Management:

- Redux Toolkit: Complex state
- React Query: Server state

Data Visualization:

- D3.js: Advanced visualizations
- Recharts: React charts
- Victory: Alternative charts
- MapBox: Geographic data

Advanced Features:

- React Virtual: Large lists
- React DnD: Drag and drop
- React Helmet: SEO
- React Router: Client-side routing

3.2 Backend Technologies

3.2.1 Core API Services

Runtime Environment:

- Node.js: 18 LTS
- npm: 9+
- TypeScript: 5.0+

Web Framework:

- Express.js: 4.18+
- Helmet: Security middleware
- CORS: Cross-origin requests
- Compression: Response compression

Authentication:

- Passport.js: Authentication middleware
- JWT: JSON Web Tokens
- bcrypt: Password hashing
- speakeasy: Two-factor authentication

API Documentation:

- Swagger: API documentation
- OpenAPI: 3.0 specification
- Redoc: Alternative documentation

Validation & Sanitization:

- Joi: Schema validation
- express-validator: Request validation
- DOMPurify: XSS protection
- Helmet: Security headers

File Handling:

- Multer: File upload middleware
- Sharp: Image processing
- pdf-parse: PDF text extraction
- archiver: File compression

Background Jobs:

- Bull Queue: Job processing
- Redis: Queue backend
- Cron: Scheduled tasks
- node-schedule: Job scheduling

Testing:

- Jest: Testing framework
- Supertest: API testing
- MongoDB Memory Server: Test database
- Factory Girl: Test data generation

3.2.2 Database Systems

Primary Database:

- PostgreSQL: 15+
- PostGIS: Geographic data
- pg: Node.js driver
- Sequelize: ORM (alternative)
- TypeORM: TypeScript ORM

Document Database:

- MongoDB: 6.x
- Mongoose: ODM
- MongoDB Compass: GUI tool

Cache & Session Store:

- Redis: 7.x
- ioredis: Node.js client
- Redis Sentinel: High availability
- Redis Cluster: Horizontal scaling

Search Engine:

- Elasticsearch: 8.x
- Kibana: Visualization
- Logstash: Data pipeline
- APM: Application monitoring

Time Series Database:

- InfluxDB: Metrics storage
- Grafana: Visualization
- Telegraf: Data collection

3.2.3 Additional Backend Services

Email Services:

- SendGrid: Transactional emails
- Amazon SES: Alternative
- Nodemailer: SMTP client
- Email Templates: Handlebars

SMS Services:

- Twilio: International SMS
- MSG91: India-focused SMS
- AWS SNS: Alternative
- Bulk SMS: Mass messaging

Push Notifications:

- Firebase Admin SDK: FCM server
- Apple Push Notifications: iOS
- Web Push: Browser notifications

Payment Processing:

- Razorpay: Indian payment gateway
- Stripe: International payments
- PayPal: Alternative
- UPI: Direct bank transfers

File Storage:

- AWS S3: Object storage
- Google Cloud Storage: Alternative
- Cloudinary: Image optimization
- ImageKit: Image CDN

PDF Generation:

- Puppeteer: PDF from HTML
- jsPDF: Client-side PDF
- PDFKit: Server-side PDF

Media Processing:

- FFmpeg: Video/audio processing
- ImageMagick: Image manipulation
- GraphicsMagick: Alternative

3.3 DevOps & Infrastructure

3.3.1 Containerization & Orchestration

Containerization:

- Docker: 24+
- Docker Compose: Multi-container apps
- Dockerfile: Container definitions
- .dockerignore: Optimization

Container Orchestration:

- Kubernetes: 1.28+
- Helm: Package manager
- Ingress Controller: Traffic routing
- Cert Manager: SSL certificates

Container Registry:

- Docker Hub: Public images
- AWS ECR: Private registry
- GitHub Container Registry: CI/CD integration

3.3.2 CI/CD Pipeline

Version Control:

- Git: Source control
- GitHub: Repository hosting
- GitLab: Alternative platform
- Bitbucket: Atlassian integration

CI/CD Platforms:

- GitHub Actions: Integrated CI/CD
- GitLab CI: Alternative
- Jenkins: Self-hosted
- CircleCI: Cloud-based

Continuous Integration:

- Automated testing: Unit, integration, e2e
- Code quality: ESLint, SonarQube
- Security scanning: Snyk, OWASP
- Dependency checking: npm audit

Continuous Deployment:

- Blue-green deployment: Zero downtime
- Rolling updates: Gradual rollout
- Canary releases: Risk mitigation
- Feature flags: Controlled rollout

3.3.3 Infrastructure as Code

Infrastructure Provisioning:

- Terraform: 1.5+
- AWS CloudFormation: AWS native
- Pulumi: Multi-cloud
- Ansible: Configuration management

Configuration Management:

- Kubernetes ConfigMaps: Application config
- Kubernetes Secrets: Sensitive data
- Vault: Secret management
- AWS Systems Manager: Parameter store

Environment Management:

- Development: Local and cloud
- Staging: Production-like testing
- Production: Live environment
- DR: Disaster recovery

3.3.4 Monitoring & Observability

Application Monitoring:

- Prometheus: Metrics collection
- Grafana: Visualization
- AlertManager: Alert routing
- Jaeger: Distributed tracing

Log Management:

- ELK Stack: Elasticsearch, Logstash, Kibana
- Fluentd: Log forwarding
- CloudWatch: AWS logging
- Papertrail: Log aggregation

Error Tracking:

- Sentry: Error monitoring
- Bugsnag: Alternative
- Rollbar: Error reporting

- LogRocket: Session replay

Performance Monitoring:

- New Relic: APM
- DataDog: Infrastructure monitoring
- Dynatrace: Full-stack monitoring
- Pingdom: Uptime monitoring

Security Monitoring:

- AWS GuardDuty: Threat detection
- Falco: Runtime security
- OWASP ZAP: Security testing
- Nessus: Vulnerability scanning

3.4 Third-Party Services

3.4.1 Communication Services

SMS Gateways:

- Twilio: Global SMS service
- MSG91: India-focused SMS
- TextLocal: UK-based service
- AWS SNS: Amazon's SMS service

Email Services:

- SendGrid: Transactional emails
- Mailgun: Developer-friendly
- Amazon SES: AWS email service
- Postmark: High deliverability

Voice Services:

- Twilio Voice: Voice calls
- Nexmo/Vonage: Voice API
- Amazon Connect: Contact center
- Google Cloud Speech: Speech recognition

3.4.2 Payment & Financial Services

Payment Gateways:

- Razorpay: Indian payment solutions
- Stripe: Global payments
- PayPal: International payments
- Paytm: Indian digital wallet

UPI Providers:

- PhonePe: UPI transactions
- Google Pay: Google's UPI
- Paytm UPI: Wallet integration
- BHIM: Government UPI app

Banking APIs:

- Razorpay X: Banking solutions
- Cashfree: Payment solutions
- InstaMojo: Small business payments
- CCAvenue: Indian payment gateway

3.4.3 Location & Maps Services

Mapping Services:

- Google Maps: Comprehensive mapping
- MapBox: Customizable maps
- Here Maps: Enterprise solutions
- OpenStreetMap: Open-source maps

Geocoding Services:

- Google Geocoding API: Address conversion
- MapBox Geocoding: Alternative
- Nominatim: OSM geocoding
- Here Geocoding: Enterprise

Location Intelligence:

- Google Places API: POI data
- Foursquare Places: Location data
- Yelp API: Business information

- TomTom Places: Location services

3.4.4 Identity & Document Verification

Indian Identity Verification:

- Aadhaar API: Government ID verification
- PAN Verification: Tax ID verification
- Driving License API: DL verification
- Voter ID API: Electoral ID verification

Document Processing:

- Tesseract OCR: Text extraction
- Google Vision API: Document analysis
- Amazon Textract: Document processing
- Azure Cognitive Services: AI services

KYC Services:

- IDfy: Identity verification
- Signzy: Digital KYC
- Perfios: Financial verification
- Bureau ID: Identity solutions

---

4. Database Design

4.1 PostgreSQL Schema Design

4.1.1 Core User Management Tables

```sql
```

```sql
-- Enable necessary extensions
CREATE EXTENSION IF NOT EXISTS "uuid-ossp";
CREATE EXTENSION IF NOT EXISTS "postgis";
CREATE EXTENSION IF NOT EXISTS "pg_trgm";
CREATE EXTENSION IF NOT EXISTS "btree_gin";

-- Custom types for better data integrity
CREATE TYPE user_type_enum AS ENUM ('job_seeker', 'employer', 'admin', 'support');
CREATE TYPE user_status_enum AS ENUM ('active', 'suspended', 'pending', 'deactivated');
CREATE TYPE gender_enum AS ENUM ('male', 'female', 'other', 'prefer_not_to_say');
CREATE TYPE business_type_enum AS ENUM ('individual', 'small_business', 'company', 'startup', 'ngo');
CREATE TYPE verification_status_enum AS ENUM ('pending', 'in_progress', 'verified', 'rejected', 'expired');
CREATE TYPE experience_level_enum AS ENUM ('fresher', 'beginner', 'intermediate', 'experienced', 'expert');
CREATE TYPE availability_enum AS ENUM ('available', 'busy', 'not_available', 'partially_available');

-- Users table (both job seekers and employers)
CREATE TABLE users (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    phone_number VARCHAR(15) UNIQUE NOT NULL,
    country_code VARCHAR(5) DEFAULT '+91',
    email VARCHAR(255) UNIQUE,
    email_verified BOOLEAN DEFAULT FALSE,
    user_type user_type_enum NOT NULL,
    status user_status_enum DEFAULT 'pending',
    preferred_language VARCHAR(10) DEFAULT 'hi',
    timezone VARCHAR(50) DEFAULT 'Asia/Kolkata',
    last_login_at TIMESTAMP,
    login_count INTEGER DEFAULT 0,
    terms_accepted_at TIMESTAMP,
    privacy_accepted_at TIMESTAMP,
    marketing_consent BOOLEAN DEFAULT FALSE,
    created_at TIMESTAMP DEFAULT NOW(),
    updated_at TIMESTAMP DEFAULT NOW(),
    deleted_at TIMESTAMP,

    -- Constraints
    CONSTRAINT users_phone_format CHECK (phone_number ~ '^[0-9]{10,15}$'),
    CONSTRAINT users_email_format CHECK (email ~ '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$')
);

-- Create indexes for performance
CREATE INDEX CONCURRENTLY idx_users_phone ON users(phone_number) WHERE deleted_at IS NULL;
CREATE INDEX CONCURRENTLY idx_users_email ON users(email) WHERE deleted_at IS NULL;
CREATE INDEX CONCURRENTLY idx_users_type_status ON users(user_type, status) WHERE deleted_at IS NULL
CREATE INDEX CONCURRENTLY idx_users_created_at ON users(created_at DESC);
```

Job Seeker Profiles:

```
sql
```

```
sql
```

```sql
CREATE TABLE job_seeker_profiles (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  user_id UUID REFERENCES users(id) ON DELETE CASCADE,
  full_name VARCHAR(255) NOT NULL,
  display_name VARCHAR(100),
  date_of_birth DATE,
  age INTEGER GENERATED ALWAYS AS (EXTRACT(YEAR FROM AGE(date_of_birth))) STORED,
  gender gender_enum,
  profile_photo_url VARCHAR(500),
  cover_image_url VARCHAR(500),

  -- Contact Information
  whatsapp_number VARCHAR(15),
  emergency_contact_name VARCHAR(255),
  emergency_contact_phone VARCHAR(15),

  -- Address Information
  current_location POINT, -- PostGIS geography type
  address_line1 VARCHAR(255),
  address_line2 VARCHAR(255),
  landmark VARCHAR(255),
  city VARCHAR(100) NOT NULL,
  district VARCHAR(100),
  state VARCHAR(100) NOT NULL,
  country VARCHAR(100) DEFAULT 'India',
  pincode VARCHAR(10),

  -- Work Preferences
  preferred_work_radius INTEGER DEFAULT 10, -- in kilometers
  max_commute_time INTEGER DEFAULT 60, -- in minutes
  availability availability_enum DEFAULT 'available',
  available_days JSONB DEFAULT '["monday","tuesday","wednesday","thursday","friday","saturday"]',
  preferred_shift JSONB DEFAULT '{"morning": true, "afternoon": true, "evening": false, "night": false}',
  willing_to_relocate BOOLEAN DEFAULT FALSE,

  -- Financial Information
  expected_salary_min INTEGER,
  expected_salary_max INTEGER,
  current_salary INTEGER,
  salary_currency VARCHAR(3) DEFAULT 'INR',
  salary_type VARCHAR(20) DEFAULT 'monthly', -- hourly, daily, weekly, monthly

  -- Personal Information
  languages_known JSONB DEFAULT '[]', -- Array of language codes
  education_level VARCHAR(50),
  marital_status VARCHAR(20),
    family_members INTEGER,
    dependents INTEGER,

    -- Verification Status
    verification_status JSONB DEFAULT '{"identity": "pending", "address": "pending", "phone": "pending"}',
    verification_score INTEGER DEFAULT 0, -- 0-100
    trust_score DECIMAL(3,2) DEFAULT 0.0, -- 0.00-5.00

    -- Profile Completion
    profile_completion_percentage INTEGER DEFAULT 0,
    onboarding_completed BOOLEAN DEFAULT FALSE,
    onboarding_step VARCHAR(50) DEFAULT 'basic_info',

    -- Activity Tracking
    last_active_at TIMESTAMP DEFAULT NOW(),
    job_search_count INTEGER DEFAULT 0,
    applications_sent INTEGER DEFAULT 0,
    profile_views INTEGER DEFAULT 0,

    created_at TIMESTAMP DEFAULT NOW(),
    updated_at TIMESTAMP DEFAULT NOW(),

    -- Constraints
    CONSTRAINT age_valid CHECK (age >= 16 AND age <= 80),
    CONSTRAINT salary_valid CHECK (expected_salary_min <= expected_salary_max),
    CONSTRAINT radius_valid CHECK (preferred_work_radius >= 1 AND preferred_work_radius <= 100)
);
```

4.1.2 Skills and Categories System

```sql
```

```sql
-- Skill Categories (Construction, Delivery, etc.)
CREATE TABLE skill_categories (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    name VARCHAR(100) NOT NULL,
    name_hindi VARCHAR(100),
    name_regional JSONB DEFAULT '{}', -- {"te": "Telugu name", "ta": "Tamil name", ...}
    slug VARCHAR(100) UNIQUE NOT NULL,
    description TEXT,
    icon_url VARCHAR(500),
    color_code VARCHAR(7) DEFAULT '#6B7280', -- Hex color for UI

    -- Hierarchy Support
    parent_category_id UUID REFERENCES skill_categories(id),
    level INTEGER DEFAULT 1, -- 1 = main category, 2 = subcategory
    sort_order INTEGER DEFAULT 0,

    -- Requirements
    requires_verification BOOLEAN DEFAULT FALSE,
    min_age INTEGER DEFAULT 16,
    max_age INTEGER,
    physical_requirements JSONB DEFAULT '[]',

    -- Metadata
    is_active BOOLEAN DEFAULT TRUE,
    is_featured BOOLEAN DEFAULT FALSE,
    job_count INTEGER DEFAULT 0, -- Updated via triggers
    worker_count INTEGER DEFAULT 0,
    avg_salary DECIMAL(10,2),

    created_at TIMESTAMP DEFAULT NOW(),
    updated_at TIMESTAMP DEFAULT NOW(),
    created_by UUID REFERENCES users(id)
);

-- Individual Skills under categories
CREATE TABLE skills (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    category_id UUID REFERENCES skill_categories(id) NOT NULL,
    name VARCHAR(100) NOT NULL,
    name_hindi VARCHAR(100),
    name_regional JSONB DEFAULT '{}',
    slug VARCHAR(100) UNIQUE NOT NULL,
    description TEXT,

    -- Skill Properties
    difficulty_level experience_level_enum DEFAULT 'beginner',
    learning_time_days INTEGER, -- Estimated days to learn
    requires_certification BOOLEAN DEFAULT FALSE,
    requires_verification BOOLEAN DEFAULT FALSE,
    verification_type VARCHAR(50) DEFAULT 'self_declared',

    -- Related Skills
    prerequisite_skills JSONB DEFAULT '[]', -- Array of skill IDs
    related_skills JSONB DEFAULT '[]',
    alternative_names JSONB DEFAULT '[]',

    -- Market Data
    demand_score INTEGER DEFAULT 0, -- 0-100 based on job postings
    supply_score INTEGER DEFAULT 0, -- 0-100 based on workers
    avg_salary_min INTEGER,
    avg_salary_max INTEGER,
    job_count INTEGER DEFAULT 0,
    worker_count INTEGER DEFAULT 0,

    -- Content
    training_resources JSONB DEFAULT '[]',
    video_url VARCHAR(500),
    documentation_url VARCHAR(500),

    is_active BOOLEAN DEFAULT TRUE,
    is_trending BOOLEAN DEFAULT FALSE,
    created_at TIMESTAMP DEFAULT NOW(),
    updated_at TIMESTAMP DEFAULT NOW(),
    created_by UUID REFERENCES users(id)
);
```

4.2 MongoDB Collections for Document Storage

4.2.1 User Documents Collection

```javascript
```

```
// User Documents Collection - Stores uploaded documents
{
  _id: ObjectId(),
  userId: "uuid", // Reference to users table
  profileId: "uuid", // Reference to job_seeker_profiles or employer_profiles

  // Document Information
  documentType: "aadhaar|pan|driving_license|educational_certificate|work_certificate|business_license|gst_certi
  category: "identity|education|work_experience|business_registration|address_proof",

  // File Information
  originalName: "aadhaar_card.pdf",
  fileName: "doc_uuid_encrypted.pdf", // Encrypted filename for security
  fileUrl: "https://storage.example.com/documents/doc_uuid_encrypted.pdf",
  thumbnailUrl: "https://storage.example.com/thumbnails/doc_uuid_thumb.jpg",
  mimeType: "application/pdf",
  fileSize: 1234567, // in bytes
  fileSizeFormatted: "1.2 MB",

  // Verification Details
  verificationStatus: "pending|in_progress|verified|rejected|expired",
  verificationDetails: {
    method: "ocr|manual|api", // How it was verified
    confidence: 0.95, // 0.0 to 1.0
    ocrText: "extracted text from document",
    extractedData: {
      name: "John Doe",
      documentNumber: "1234-5678-9012",
      issueDate: "2020-01-01",
      expiryDate: "2030-01-01",
      address: "123 Main St, City, State, PIN",
      dateOfBirth: "1990-01-01",
      fatherName: "Father Name"
    },
    verificationDate: ISODate("2024-01-15T10:30:00Z"),
    verifiedBy: "uuid", // User ID of verifier
    verificationNotes: "Document appears authentic",
    rejectionReason: "Unclear image quality"
  },

  uploadedAt: ISODate("2024-01-15T10:30:00Z"),
  updatedAt: ISODate("2024-01-15T10:35:00Z")
}
```

4.2.2 Job Analytics Collection

javascript

```
// Job Performance Analytics Collection
{
 _id: ObjectId(),
 jobPostingId: "uuid", // Reference to job_postings table
 employerId: "uuid", // Reference to employer_profiles table

 // Time Period
 analyticsPeriod: "daily|weekly|monthly",
 startDate: ISODate("2024-01-01T00:00:00Z"),
 endDate: ISODate("2024-01-31T23:59:59Z"),

 // View Analytics
 viewsData: {
  totalViews: 1250,
  uniqueViews: 980,
  averageViewDuration: 45, // seconds
  bounceRate: 0.35, // percentage of single-page sessions

  // Daily breakdown
  dailyViews: [
   {
    date: "2024-01-01",
    views: 45,
    uniqueViews: 38,
    averageDuration: 42,
    topViewingHours: [9, 10, 14, 15, 18, 19] // Peak hours
   }
  ],

  // Geographic distribution
  viewsByLocation: [
   {
    city: "Bangalore",
    state: "Karnataka",
    views: 345,
    uniqueViews: 287
   },
   {
    city: "Mumbai",
    state: "Maharashtra",
    views: 198,
    uniqueViews: 165
   }
  ]
 },
  // Application Analytics
  applicationsData: {
    totalApplications: 87,
    qualifiedApplications: 45,
    shortlistedApplications: 12,
    interviewedCandidates: 8,
    selectedCandidates: 2,

    // Conversion funnel
    conversionRates: {
      viewToApplication: 0.0696, // 87/1250
      applicationToShortlist: 0.138, // 12/87
      shortlistToInterview: 0.667, // 8/12
      interviewToSelection: 0.25 // 2/8
    }
  },

  lastUpdated: ISODate("2024-02-01T00:00:00Z"),
  dataSource: "automated_collection",
  version: "1.0"
}
```

4.3 Redis Data Structures and Caching Strategy

4.3.1 User Session Management

```
redis
```

```
# User sessions with detailed information
HSET user:session:{sessionId}
  userId "uuid"
  userType "job_seeker"
  loginTime "2024-01-15T10:30:00Z"
  lastActivity "2024-01-15T14:25:00Z"
  deviceInfo '{"platform":"android","model":"Samsung Galaxy A12","appVersion":"1.2.3"}'
  location '{"city":"Bangalore","state":"Karnataka","coordinates":[77.5946,12.9716]}'
  permissions '["location","camera","notifications"]'

EXPIRE user:session:{sessionId} 86400 # 24 hours

# Active user tracking
SADD online_users {userId}
EXPIRE online_users 300 # 5 minutes

# User preferences cache
HSET user:preferences:{userId}
  language "hi"
  notifications_enabled "true"
  job_alert_radius "15"
  salary_alerts "true"
  push_notifications "true"
  sms_notifications "false"

EXPIRE user:preferences:{userId} 7200 # 2 hours
```

4.3.2 Job Search and Matching Cache

```redis
```

```
# Job search results cache with complex key generation
SET jobs:search:{hashOfQueryParams} '[
  {"id":"job_uuid_1","title":"Construction Worker","location":"Bangalore","salary":18000},
  {"id":"job_uuid_2","title":"Mason Required","location":"Bangalore","salary":20000}
]'
EXPIRE jobs:search:{hashOfQueryParams} 1800 # 30 minutes

# Popular searches cache
ZINCRBY popular_searches 1 "construction worker bangalore"
ZINCRBY popular_searches 1 "delivery boy mumbai"

# Job recommendations for users (personalized)
SET user:recommendations:{userId} '[
  {"jobId":"uuid1","score":0.95,"reason":"skill_match"},
  {"jobId":"uuid2","score":0.87,"reason":"location_proximity"}
]'
EXPIRE user:recommendations:{userId} 3600 # 1 hour

# Job view counts (real-time)
HINCRBY job:stats:{jobId} views_count 1
HINCRBY job:stats:{jobId} applications_count 1
```

4.3.3 Application Rate Limiting and Security

```redis
# API rate limiting by user
INCR api:rate_limit:{userId}:job_search
EXPIRE api:rate_limit:{userId}:job_search 3600 # Reset every hour

# OTP rate limiting
INCR otp:attempts:{phoneNumber}
EXPIRE otp:attempts:{phoneNumber} 900 # 15 minutes

# Failed login attempt tracking
INCR login:failed:{phoneNumber}
EXPIRE login:failed:{phoneNumber} 1800 # 30 minutes
```

---

5. API Architecture

5.1 RESTful API Design Principles

5.1.1 API Standards and Conventions

Base URL Structure:

- Production: `https://api.jobportal.com/v1`
- Staging: `https://api-staging.jobportal.com/v1`
- Development: `https://api-dev.jobportal.com/v1`

HTTP Methods:

- `GET`: Retrieve resources
- `POST`: Create new resources
- `PUT`: Update entire resources
- `PATCH`: Partial updates
- `DELETE`: Remove resources

Response Format:

```json
{
  "success": true,
  "data": {...},
  "message": "Operation completed successfully",
  "timestamp": "2024-01-15T10:30:00Z",
  "requestId": "req_550e8400-e29b-41d4-a716-446655440000"
}
```

Error Response Format:

```json
{
  "success": false,
  "error": {
    "code": "VALIDATION_ERROR",
    "message": "The provided data is invalid",
    "details": [
      {
        "field": "phoneNumber",
        "message": "Phone number is required"
      }
    ]
  },
  "timestamp": "2024-01-15T10:30:00Z",
  "requestId": "req_550e8400-e29b-41d4-a716-446655440000"
}
```

5.1.2 Authentication and Authorization

JWT Token Structure:

```json
{
  "header": {
    "alg": "HS256",
    "typ": "JWT"
  },
  "payload": {
    "userId": "550e8400-e29b-41d4-a716-446655440000",
    "userType": "job_seeker",
    "phoneNumber": "+919876543210",
    "permissions": ["read_jobs", "apply_jobs", "manage_profile"],
    "iat": 1642271400,
    "exp": 1642357800,
    "iss": "jobportal.com"
  }
}
```

Authorization Headers:

```http
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
X-API-Key: api_key_for_third_party_integrations
X-Client-Version: 1.2.3
X-Platform: android
```

5.2 Core API Endpoints

5.2.1 Authentication APIs

User Registration:

```http
```

```
POST /auth/register
Content-Type: application/json

{
  "phoneNumber": "+919876543210",
  "userType": "job_seeker",
  "referralCode": "REF123456",
  "deviceInfo": {
    "platform": "android",
    "model": "Samsung Galaxy A12",
    "version": "11",
    "appVersion": "1.2.3"
  }
}

Response:
{
  "success": true,
  "data": {
    "userId": "550e8400-e29b-41d4-a716-446655440000",
    "otpSent": true,
    "expiresIn": 300,
    "nextStep": "verify_otp"
  }
}
```

OTP Verification:

```
http
```

```
POST /auth/verify-otp
Content-Type: application/json

{
  "phoneNumber": "+919876543210",
  "otp": "123456",
  "deviceToken": "fcm_device_token_here"
}

Response:
{
  "success": true,
  "data": {
    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
    "refreshToken": "refresh_token_here",
    "user": {
      "id": "550e8400-e29b-41d4-a716-446655440000",
      "phoneNumber": "+919876543210",
      "userType": "job_seeker",
      "isNewUser": true,
      "profileCompleted": false
    },
    "tokenExpiry": "2024-01-15T22:30:00Z"
  }
}
```

5.2.2 Job Search APIs

Search Jobs:

```
http
```

```
GET /jobs/search
Authorization: Bearer {token}
Query Parameters:
 ?query=mason
 &location=bangalore
 &radius=10
 &salaryMin=800
 &salaryMax=1500
 &jobType=full_time
 &urgency=immediate
 &skills=masonry,construction
 &page=1
 &limit=20
 &sort=relevance

Response:
{
 "success": true,
 "data": {
  "jobs": [
   {
    "id": "job_550e8400",
    "title": "Experienced Mason Required",
    "shortDescription": "We need an experienced mason for residential construction project",
    "employer": {
     "id": "emp_123",
     "name": "ABC Construction Pvt Ltd",
     "logo": "https://cdn.example.com/logos/abc_construction.jpg",
     "rating": 4.2,
     "totalReviews": 89,
     "verificationStatus": "verified"
    },
    "location": {
     "city": "Bangalore",
     "state": "Karnataka",
     "address": "Whitefield, Bangalore",
     "distance": 5.2,
     "coordinates": [77.7500, 12.9698]
    },
    "salary": {
     "min": 800,
     "max": 1200,
     "type": "daily",
     "currency": "INR",
     "negotiable": false
    },
    "requirements": {
      "skills": ["masonry", "construction"],
      "experience": "intermediate",
      "minAge": 21,
      "maxAge": 45
    },
    "benefits": ["food_provided", "transport_allowance"],
    "urgency": "immediate",
    "postedAt": "2024-01-15T10:30:00Z",
    "expiresAt": "2024-02-15T23:59:59Z",
    "applicationsCount": 23,
    "viewsCount": 156,
    "matchScore": 0.94,
    "featured": true,
    "premium": false
  }
],
"filters": {
  "availableFilters": {
    "jobTypes": ["full_time", "part_time", "contract"],
    "urgencyLevels": ["immediate", "within_24_hours", "within_week"],
    "salaryRanges": [
      {"min": 0, "max": 500, "count": 45},
      {"min": 500, "max": 1000, "count": 120},
      {"min": 1000, "max": 1500, "count": 89}
    ],
    "skills": [
      {"id": "masonry", "name": "Masonry", "count": 67},
      {"id": "construction", "name": "Construction", "count": 156}
    ]
  }
},
"pagination": {
  "page": 1,
  "limit": 20,
  "total": 245,
  "pages": 13,
  "hasNext": true,
  "hasPrev": false
},
"searchInsights": {
  "totalResults": 245,
  "searchTime": 0.045,
  "popularSearches": ["mason bangalore", "construction worker"],
  "suggestedsearches": ["carpenter bangalore", "plumber bangalore"]
}
```

```
  }
}
```

Get Job Details:

```
http
```

```
GET /jobs/550e8400-e29b-41d4-a716-446655440000
Authorization: Bearer {token}

Response:
{
  "success": true,
  "data": {
    "job": {
      "id": "job_550e8400",
      "title": "Experienced Mason Required - Residential Project",
      "description": "We are looking for an experienced mason for our residential construction project in Whitefield a
      "shortDescription": "Experienced mason needed for 3-month residential project",
      "category": {
        "id": "construction",
        "name": "Construction",
        "icon": "https://cdn.example.com/icons/construction.png"
      },
      "employer": {
        "id": "emp_123",
        "name": "ABC Construction Pvt Ltd",
        "description": "Leading construction company in Bangalore with 15+ years of experience",
        "rating": 4.2,
        "totalReviews": 89,
        "responseRate": 0.89,
        "avgResponseTime": "4 hours",
        "logo": "https://cdn.example.com/logos/abc_construction.jpg",
        "website": "https://abcconstruction.com",
        "location": {
          "city": "Bangalore",
          "state": "Karnataka"
        },
        "otherActiveJobs": 12,
        "totalHires": 89,
        "hiringSuccessRate": 0.78
      },
      "location": {
        "coordinates": [77.7500, 12.9698],
        "address": {
          "line1": "Whitefield Main Road",
          "line2": "Near ITPL",
          "city": "Bangalore",
          "state": "Karnataka",
          "pincode": "560066",
          "landmark": "Opposite Forum Mall"
        },
        "description": "Easy access by public transport",
    "distance": 5.2,
    "travelTime": "25 minutes by bus"
  },
  "salary": {
    "min": 800,
    "max": 1200,
    "type": "daily",
    "currency": "INR",
    "negotiable": false,
    "paymentTerms": "Weekly payment on Saturdays"
  },
  "benefits": [
    {
      "type": "food_provided",
      "name": "Food Provided",
      "description": "Lunch and tea provided"
    },
    {
      "type": "transport_allowance",
      "name": "Transport Allowance",
      "description": "₹100 per day transport allowance"
    }
  ],
  "requirements": {
    "skills": [
      {
        "id": "masonry",
        "name": "Masonry",
        "level": "intermediate",
        "required": true,
        "minExperience": 2
      }
    ],
    "experience": "intermediate",
    "minAge": 21,
    "maxAge": 45,
    "physicalRequirements": "Ability to lift 25kg, work at heights",
    "educationRequirements": "Primary education preferred",
    "languageRequirements": ["hindi", "kannada"],
    "additionalRequirements": "Should have own basic tools"
  },
  "workDetails": {
    "jobType": "full_time",
    "startDate": "2024-01-22",
    "endDate": "2024-04-22",
    "duration": "3 months",
    "urgency": "immediate",
    "workingHours": {
      "start": "08:00",
      "end": "17:00",
      "break": 60
    },
    "workingDays": ["monday", "tuesday", "wednesday", "thursday", "friday", "saturday"],
    "shiftDetails": "Single day shift only"
  },
  "applicationProcess": {
    "deadline": "2024-01-25T23:59:59Z",
    "instructions": "Please bring work samples and references",
    "contactPerson": "Rajesh Kumar",
    "contactPhone": "+919876543210",
    "screeningQuestions": [
      {
        "question": "How many years of masonry experience do you have?",
        "type": "number",
        "required": true
      }
    ]
  },
  "metrics": {
    "applicationsCount": 23,
    "viewsCount": 156,
    "uniqueViewsCount": 134,
    "shortlistCount": 5,
    "averageApplicationTime": "2.5 days",
    "competitionLevel": "moderate"
  },
  "timeline": {
    "postedDate": "2024-01-15T10:30:00Z",
    "expiresAt": "2024-02-15T23:59:59Z",
    "startDate": "2024-01-20",
    "expectedDuration": "3 months",
    "urgency": "immediate"
  },
  "insights": {
    "similarJobsCount": 15,
    "averageSalaryInArea": 950,
    "demandTrend": "increasing",
    "bestTimeToApply": "morning",
    "successRate": "68% of applications get response"
  },
  "userContext": {
    "matchScore": 0.94,
    "canApply": true,
    "applicationStatus": null,
      "missingRequirements": [],
      "strengthAreas": ["skill_match", "location_preference"],
      "recommendationReason": "Perfect match for your masonry skills and location preference"
     }
    }
  }
}
```

5.3 Job Application APIs

5.3.1 Apply for Job

```
http
```

```
POST /jobs/550e8400-e29b-41d4-a716-446655440000/apply
Authorization: Bearer {token}
Content-Type: multipart/form-data

Form Data:
- coverMessage: "I am interested in this mason position..."
- expectedSalary: 1000
- availabilityStartDate: "2024-01-22"
- resume: [file upload]
- voiceMessage: [audio file upload]
- screeningAnswers: {"experience_years": 3, "can_work_heights": true}

Response:
{
  "success": true,
  "data": {
    "application": {
      "id": "app_550e8400",
      "jobId": "job_550e8400",
      "status": "applied",
      "appliedAt": "2024-01-16T14:30:00Z",
      "applicationNumber": "APP-2024-001234"
    },
    "nextSteps": [
      {
        "step": "employer_review",
        "description": "Your application will be reviewed by the employer",
        "expectedTime": "2-3 business days"
      },
      {
        "step": "possible_interview",
        "description": "You may be contacted for an interview",
        "tips": ["Keep your phone available", "Prepare work samples"]
      }
    ],
    "trackingInfo": {
      "applicationId": "app_550e8400",
      "trackingUrl": "/applications/app_550e8400/track"
    }
  }
}
```

5.3.2 Track Application Status

```
http
```

```
GET /applications/550e8400-e29b-41d4-a716-446655440000
Authorization: Bearer {token}

Response:
{
  "success": true,
  "data": {
    "application": {
      "id": "app_550e8400",
      "job": {
        "id": "job_550e8400",
        "title": "Experienced Mason Required",
        "employer": {
          "name": "ABC Construction Pvt Ltd",
          "logo": "https://cdn.example.com/logos/abc_construction.jpg"
        }
      },
      "status": "shortlisted",
      "timeline": [
        {
          "status": "applied",
          "timestamp": "2024-01-16T14:30:00Z",
          "description": "Application submitted successfully"
        },
        {
          "status": "viewed",
          "timestamp": "2024-01-17T09:15:00Z",
          "description": "Employer viewed your application"
        },
        {
          "status": "shortlisted",
          "timestamp": "2024-01-17T11:30:00Z",
          "description": "You have been shortlisted for this position"
        }
      ],
      "interviewDetails": {
        "scheduled": true,
        "dateTime": "2024-01-20T10:00:00Z",
        "type": "in_person",
        "location": {
          "address": "ABC Construction Office, Whitefield",
          "coordinates": [77.7500, 12.9698]
        },
        "contactPerson": "Rajesh Kumar",
        "contactPhone": "+919876543210",
        "instructions": "Please bring your ID proof and work samples"
      },
      "messages": [
        {
          "id": "msg_123",
          "from": "employer",
          "message": "We are impressed with your profile. Can you join for an interview tomorrow?",
          "timestamp": "2024-01-17T11:35:00Z",
          "read": true
        }
      ],
      "feedback": {
        "strengths": ["Good experience", "Local candidate"],
        "concerns": [],
        "overallRating": 4.5
      }
    }
  }
}
```

This completes the comprehensive formatting of your System Design Document. The document now has:

- Clean, consistent formatting with proper headings and structure
- Well-organized code blocks with syntax highlighting
- Professional table of contents with working links
- Consistent spacing and typography
- Clear section breaks and visual hierarchy
- Properly formatted JSON, SQL, and Redis examples

The document maintains all the technical detail from your original while presenting it in a much more readable and professional format suitable for technical reviews, stakeholder presentations, or development team reference.