ConnectGig – AI-powered Gig Worker Platform (Full-stack Product Specification)

1. Vision & Mission

Vision: Build an intelligent, location-aware, AI-driven platform that connects skilled gig workers with clients needing on-demand services.

Mission: Enable faster hiring, fair pricing, and trust through modern AI, real-time tech, and seamless UX.

2. Target Users

- **Clients:** Individuals or businesses who require on-demand skilled services (plumbing, electrical, repair, tutoring, etc.).

- **Workers:** Skilled professionals seeking local jobs with transparency and trust.

3. Core Features

- Dual App Modes: Client app & Worker app (shared login screen, role selection).

- AI Matching Engine: Recommends best-fit workers based on skills, ratings, location, and availability.

- Smart Pricing Assistant: AI model suggests fair pricing based on historical job data, demand, and locality trends.

- Real-time Location Tracking: Worker discovery, live ETA, geofencing.

- In-app Chat: Client ↔ Worker direct messaging.

- Reviews & Ratings: Star + text-based feedback, AI moderation to filter abuse/spam.

- Identity Verification: KYC for workers, optional verification badge.

- Secure Payments: UPI, cards, wallets integration with escrow system.

- Dashboard & Analytics: Admin console with worker stats, revenue, complaints.

- Goodies for Workers: Onboarding kits (T-shirt, cap, glasses).

4. Modern Tech Stack (AI-first)

Frontend:

- Web: Next.js 15 + Tailwind + Shadcn UI
- Mobile: React Native (Expo)
- State Management: Zustand or Redux Toolkit
- Animations: Framer Motion

Backend (API Layer):

- NestJS + TypeScript
- GraphQL (Apollo) or REST (OpenAPI 3)
- Prisma ORM
- PostgreSQL + PostGIS (geospatial queries)
- Redis (caching, job queues)
- WebSockets (real-time chat, notifications)

AI/ML Services:

- Matching Model: Vector embeddings of worker skills & job requests
- Pricing Model: Regression/transformer-based demand prediction
- Moderation: LLM API for abusive text filtering

Infra & DevOps:

- Monorepo: Turborepo (apps: web, mobile, api, admin)
- Containerization: Docker + Docker Compose (local)
- Cloud: AWS/GCP/Azure
- CI/CD: GitHub Actions → Vercel (Web) + App Store/Play Store builds
- Auth: Clerk/Auth0 (passwordless + OAuth)
- Payments: Stripe/UPI provider

Color Palette (Consistent):

- Primary Blue: `#2563eb`
- Secondary White: `#f9fafb`
- Accent Milky White: `#ffffff`
- Neutral Gray: `#6b7280`

5. Database Schema (Simplified ERD)

Users(id, name, email, password_hash, role[client/worker], phone, kyc_verified, created_at)

Workers(id, user_id FK, skills[], rating_avg, jobs_completed, goodies_sent)

Jobs(id, client_id FK, worker_id FK, title, description, price, status[pending/accepted/completed/cancelled], created_at)

Reviews(id, job_id FK, reviewer_id FK, rating, text, created_at)

Payments(id, job_id FK, client_id FK, worker_id FK, amount, status, method, created_at)

Chats(id, job_id FK, sender_id, receiver_id, message, created_at)

6. Security & Compliance

- End-to-end encryption for chat.
- AES-256 for sensitive DB fields (payment info, KYC).
- GDPR/Indian Data Privacy compliance.
- Role-based access control (admin, client, worker).

7. Development Roadmap (MVP → AI-first Scale)

MVP (3–4 months):

- Auth, Profile, Job posting, Worker acceptance
- Payments & Reviews
- Basic Admin dashboard

Phase 2 (4–6 months):

- AI Matching Engine
- Real-time tracking
- In-app chat + notifications

Phase 3 (6–12 months):

- AI Pricing Assistant
- Abuse detection via LLM moderation
- Worker goodies + loyalty program

Phase 4 (12+ months):

- Advanced analytics dashboard
- Multilingual AI assistants
- Expansion to new cities

8. Cursor/AI Development Notes

- Store this doc in `/docs/connectgig-spec-v1.pdf`
- AI agents (like Cursor) can auto-generate:
- Prisma schema & migrations
- NestJS resolvers/controllers
- React Native screens with Tailwind
- OpenAPI/GraphQL endpoints
- Keep **consistent color scheme** in UI components
- Maintain **modular monorepo** for scalability

End of v1. Ready for code scaffolding & OpenAPI generation.