MOVZZ — Full Product Roadmap

Reliability-Orchestrated Mobility Platform · Chennai, India Last updated: February 2026

Legend

Symbol	Meaning
☑ DONE	Implemented and working
🔄 IN PROGRESS	Partially implemented
☐ TODO	Not yet started

Phase 1 — Core Backend Foundation ☑

#	Task	Status	Key Files
1	Project scaffold (Express + TypeScript + Prisma)	☑ DONE	`backend/src/index.ts`, `prisma/schema.prisma`
2	PostgreSQL schema — Users, Providers, Bookings, BookingAttempts, BookingLogs	☑ DONE	`prisma/schema.prisma`
3	Redis cache (ioredis) with in-memory fallback	☑ DONE	`backend/src/config/redis.ts`
4	OTP authentication (phone + email) with JWT	☑ DONE	`backend/src/controllers/auth.controller.ts`
5	Fare engine — mode-specific rates, surge pricing, Haversine × 1.35 road factor, airport detection	☑ DONE	`backend/src/services/fare.service.ts`
6	Provider scoring — reliability score from historical metrics	☑ DONE	`backend/src/services/provider-scoring.service.ts`
7	Booking state machine — SEARCHING → CONFIRMED → IN_PROGRESS → COMPLETED / FAILED / CANCELLED / MANUAL_ESCALATION	☑ DONE	`backend/src/services/booking.service.ts`
8	Recovery service — 3-level retry, auto-escalation to ops, ₹100 compensation credit	☑ DONE	`backend/src/services/recovery.service.ts`

#	Task	Status	Key Files
9	Quotes API — ranked ride options with reliability rationale	☑ DONE	`backend/src/services/quotes.service.ts`
10	Booking API — create, poll status, cancel	☑ DONE	`backend/src/controllers/booking.controller.ts`
11	Admin dashboard API — overview stats, provider list	☑ DONE	`backend/src/controllers/admin.controller.ts`
12	Rate limiting, CORS, Helmet security headers	☑ DONE	`backend/src/index.ts`

Phase 2 — Frontend Prototype ☑

#	Task	Status	Key Files
13	React + Vite scaffold with Zustand state management	☑ DONE	`frontend/src/App.jsx`
14	5-screen flow: Landing → Auth → Transport → Destination → Results	☑ DONE	`frontend/src/App.jsx`
15	Transport mode selector — CAB (Economy/Comfort/Premium), Bike, Auto, Metro	☑ DONE	`frontend/src/App.jsx`
16	Results screen — ranked quote cards with reliability score, ETA, price, tags (BEST / CHEAPEST / PREMIUM)	☑ DONE	`frontend/src/App.jsx`
17	Auth store — OTP send/verify, JWT storage, auth state	☑ DONE	`frontend/src/stores/authStore.ts`
18	Booking store — fetchQuotes, createBooking, Socket.IO state sync	☑ DONE	`frontend/src/stores/bookingStore.ts`
19	API client (axios) with JWT header injection	☑ DONE	`frontend/src/api/client.ts`

Phase 3 — Production Hardening ☑

#	Task	Status	Key Files
20	WebSockets (Socket.IO) — replaced 5-second polling with server-push `booking:state_changed` events	☑ DONE	`backend/src/config/socket.ts`, `frontend/src/stores/bookingStore.ts`

#	Task	Status	Key Files
21	Google OAuth2 (no Passport.js) — direct REST flow, id_token decode, JWT return	☑ DONE	`backend/src/controllers/oauth.controller.ts`
22	Mapbox geocoding autocomplete — 300ms debounce, suggestion dropdowns	☑ DONE	`frontend/src/App.jsx`
23	Interactive map — react-map-gl with green (pickup) and orange (dropoff) markers, auto-pan	☑ DONE	`frontend/src/App.jsx`
24	Chennai-preset location chips with real coordinates	☑ DONE	`frontend/src/App.jsx`

Phase 4 — Background Jobs ☑

#	Task	Status	Key Files
25	BullMQ queue singletons — booking-timeout, recovery-retry, sms-dispatch	☑ DONE	`backend/src/config/queues.ts`
26	Booking timeout worker — auto-cancel bookings stuck in SEARCHING after 5 minutes + issue compensation	☑ DONE	`backend/src/workers/booking-timeout.worker.ts`
27	Recovery retry worker — async provider retry with 2s delay, delegates to existing 3-level retry logic	☑ DONE	`backend/src/workers/recovery.worker.ts`
28	SMS dispatch worker — retryable OTP delivery (3 attempts, exponential backoff), Twilio-ready	☑ DONE	`backend/src/workers/sms.worker.ts`

Phase 5 — Payments ☐

#	Task	Status	Notes
29	Razorpay — create order endpoint	☐ TODO	`POST /api/v1/payments/create-order` · needs `RAZORPAY_KEY_ID` + `RAZORPAY_KEY_SECRET`
30	Razorpay — verify payment endpoint + HMAC-SHA256 signature check	☐ TODO	`POST /api/v1/payments/verify` · transitions booking SEARCHING → CONFIRMED on success
31	Frontend — Razorpay Web Checkout modal (CDN script, no new npm)	☐ TODO	Opens after `createBooking()` succeeds; uses test card 4111 1111 1111 1111

#	Task	Status	Notes
32	Provider payout tracking (T+2 terms)	☐ TODO	Prisma schema addition — `ProviderPayout` table

Phase 6 — Notifications ☐

#	Task	Status	Notes
33	Twilio SMS — replace mock in sms.worker.ts	☐ TODO	`TWILIO_ACCOUNT_SID`, `TWILIO_AUTH_TOKEN`, `TWILIO_PHONE_NUMBER` env vars
34	SendGrid transactional email — booking confirmation, cancellation, compensation	☐ TODO	`npm install @sendgrid/mail` · `SENDGRID_API_KEY` · ~50 lines
35	FCM push notifications — booking state changes, driver ETA updates	☐ TODO	Requires React Native or PWA service worker
36	WhatsApp API notifications via Twilio	☐ TODO	Same Twilio credentials, different endpoint

Phase 7 — Error Monitoring & Infrastructure ☐

#	Task	Status	Notes
37	Sentry — error tracking for backend + frontend	☐ TODO	`npm install @sentry/node @sentry/react` · 3 lines of init code each
38	Neon.tech — migrate from local Postgres to cloud	☐ TODO	Just swap `DATABASE_URL` in `.env` — zero code change
39	AWS S3 — profile photos, document uploads	☐ TODO	`npm install @aws-sdk/client-s3` · `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`
40	CDN — static asset delivery for frontend	☐ TODO	CloudFront or Cloudflare in front of S3

Phase 8 — Provider Integrations ☐

#	Task	Status	Notes
41	Fast Track Cabs API integration	☐ TODO	Real provider dispatch instead of mock scoring
42	Chennai Call Taxi API integration	☐ TODO	Local fleet operator
43	Uber/Ola API fallback (when all MOVZZ providers fail)	☐ TODO	Aggregator fallback layer

#	Task	Status	Notes
44	Provider onboarding portal — document upload, KYC	☐ TODO	Admin-side flow

Phase 9 — Admin Panel ☐

#	Task	Status	Notes
45	Full admin dashboard UI	☐ TODO	Current: API only (`GET /api/v1/admin/dashboard`)
46	Live booking map — all active bookings on a map	☐ TODO	Socket.IO + Mapbox
47	Provider management — add/edit/deactivate, reliability history	☐ TODO	CRUD on Provider table
48	Manual escalation queue — ops team resolves MANUAL_ESCALATION bookings	☐ TODO	Already tracked in DB, needs UI
49	Analytics dashboard — daily rides, revenue, reliability trends	☐ TODO	Aggregate queries on Booking + ProviderMetric tables

Phase 10 — Mobile Apps ☐

#	Task	Status	Notes
50	React Native user app — port existing 5-screen web flow	☐ TODO	Expo recommended; reuse existing API + WebSocket
51	React Native driver app — accept/reject rides, navigation	☐ TODO	Separate app with driver-specific Socket.IO events
52	Driver location tracking — real-time GPS updates	☐ TODO	New `driver:location_update` Socket.IO event
53	In-app FCM push notifications	☐ TODO	Firebase + React Native Push Notification library
54	App Store + Play Store deployment	☐ TODO	Expo EAS build

Summary

Phase	Description	Tasks	Status
1	Core Backend Foundation	12	☑ Complete
2	Frontend Prototype	7	☑ Complete

Phase	Description	Tasks	Status
3	Production Hardening	5	☑ Complete
4	Background Jobs (BullMQ)	4	☑ Complete
5	Payments (Razorpay)	4	☐ Not started
6	Notifications	4	☐ Not started
7	Error Monitoring & Infrastructure	4	☐ Not started
8	Provider Integrations	4	☐ Not started
9	Admin Panel	5	☐ Not started
10	Mobile Apps	5	☐ Not started
Total		**54**	**28 done / 26 remaining**

Recommended Next Steps (by effort)

Priority	Task	Effort	Value
1	Sentry (#37)	30 min	Error visibility in dev now
2	Neon.tech (#38)	15 min	Cloud DB, shareable
3	SendGrid (#34)	2 hrs	Real email notifications
4	Razorpay (#29–31)	1 day	Core revenue flow
5	Twilio SMS (#33)	2 hrs	Real OTP delivery
6	Admin Panel UI (#45–49)	3–5 days	Ops tooling
7	React Native (#50–51)	2–3 weeks	Full mobile product