MOVZZ

Frontend ↔ Backend Integration Roadmap

Version 3.0 February 28, 2026 *Updated after Phase 4 Session*

Progress Overview

14 of 15 roadmap tasks are complete. WebSockets and Google OAuth are fully functional. Google Maps integration code is implemented (Places Autocomplete, Distance Matrix) but blocked on Google Cloud API activation — the Maps JavaScript API and Places API must be enabled in GCP Library and billing must be attached before the autocomplete dropdown becomes functional.

Phase	Tasks	Completed	Status
Phase 1: Backend	5 tasks	5/5	☑ Complete
Phase 2: Frontend API	4 tasks	4/4	☑ Complete
Phase 3: Screen Wiring	3 tasks	3/3	☑ Complete
Phase 4: Production	3 tasks	**2/3**	◍ **In Progress**

Task-by-Task Status

Phase 1 — Backend Modifications

#	Task	Owner	Est.	Status	Completion Notes
1	Add transportMode to Prisma schema + migration	Backend	2 hrs	☑ DONE	TransportMode enum: CAB, BIKE, AUTO, METRO. BIKE_TAXI renamed to BIKE.
2	Build mode-specific fare estimation	Kevin	4 hrs	☑ DONE	fare.service.ts (562 lines). 3 cab tiers, time-aware surge, 1.35x road factor, paise precision. 52/52 tests. PR #6.
3	Build POST /quotes endpoint	Backend	8 hrs	☑ DONE	quotes.controller.ts + quotes.service.ts. Individual quote caching, tag enums (BEST/CHEAPEST/PREMIUM), quoteId wiring.
4	Metro-specific quote logic	Kevin	4 hrs	☑ DONE	metro.service.ts (147 lines). Extracted from fare.service.ts. Chennai Blue + Green lines, station-based fares. PR #8.
12	Wire real Redis (replace MemoryCache)	Backend	1 hr	☑ DONE	ioredis with graceful fallback. Integrated in Session 002.

Phase 2 — Frontend API Layer

#	Task	Owner	Est.	Status	Completion Notes
5	Frontend: API client + auth store	Backend	4 hrs	☑ DONE	Zustand authStore + axios client with JWT interceptor. Built in Session 002.
6	Frontend: Phone OTP input UI	Backend	3 hrs	☑ DONE	Phone input + 6-digit OTP flow. Built in Session 002.
7	Frontend: Wire auth screen to OTP flow	Both	3 hrs	☑ DONE	Auth response format fixed (PR #9) — aligned { success, data } pattern. OTP flow now works end-to-end.
8	Frontend: Booking store (Zustand)	Backend	3 hrs	☑ DONE	Booking store with mode, locations, quotes, polling. Built in Session 002.

Phase 3 — Screen-by-Screen Wiring

#	Task	Owner	Est.	Status	Completion Notes
9	Wire Results screen to POST /quotes	Backend	4 hrs	☑ DONE	App.jsx helper extraction (getToneClass, getTagLabel). Quote flow wired. Session 002.
10	Wire Confirm screen to POST /bookings	Backend	4 hrs	☑ DONE	quoteId wiring (Zod → controller → service → Redis). Booking fast-path fixed (isActive bug, PR #8).
11	Frontend: Booking status polling	Backend	2 hrs	☑ DONE	Polling implemented in Session 002. ride.controller.ts deprecated (410 Gone).

Phase 4 — Production Hardening

#	Task	Owner	Est.	Status	Completion Notes
13	WebSockets / Socket.IO for live updates	Both	6 hrs	☑ DONE	Replaced 5-second HTTP polling with server-push events. Per-user rooms via socket.join(userId). JWT auth on handshake.
14	Google OAuth	Backend	8 hrs	☑ DONE	Direct REST flow, no Passport.js. "Continue with Google" alongside OTP. Users stored as phone = "oauth_google_<sub>".
15	Google Maps integration	Frontend	6 hrs	▨ IN PROGRESS	Code complete. Blocked on GCP — Maps JavaScript API + Places API must be enabled in Library and billing must be activated. "This page can't load Google Maps correctly" error until resolved.

Phase 4 — Technical Details

Task #13 — WebSockets / Socket.IO

Problem: Frontend was polling `GET /bookings/:id` every 5 seconds. 95% of requests returned no state change.

Solution: Server pushes `booking:state_changed` the moment a state transition occurs.

Files changed:

- `backend/src/config/socket.ts` *(new)* — Singleton `setIo()` / `getIo()` to avoid circular imports between `index.ts` and `booking.service.ts`.
- `backend/src/index.ts` — Replaced `app.listen()` with `http.createServer(app)` + Socket.IO. JWT auth middleware on the socket handshake. `socket.join(userId)` for per-user rooms. `io.close()` added to graceful shutdown.
- `backend/src/services/booking.service.ts` — After every `prisma.booking.update()` in `transitionState()`, emits `booking:state_changed` with `{ id, state, previousState, metadata, updatedAt }` to the user's room.
- `frontend/src/stores/bookingStore.ts` — Added `socket: Socket | null` to store state. `connectSocket(token)` creates the client, authenticates, listens for `booking:state_changed`. `disconnectSocket()` for cleanup.
- `frontend/src/App.jsx` — Removed `setInterval` polling useEffect. `connectSocket` called after OTP verify and after OAuth login. Mount-time effect reconnects on page reload if token in `localStorage`.

Task #14 — Google OAuth 2.0

Problem: Auth was phone OTP only. No social login option.

Solution: Direct OAuth2 REST flow. No Passport.js — uses Node 22 built-in `fetch`. Google users stored with `phone = "oauth_google_<googleId>"` as placeholder (no schema change required).

Flow:

```
User clicks "Continue with Google"
  → window.location = GET /api/v1/auth/google
    → backend redirects to Google consent screen
      → Google redirects to GET /api/v1/auth/google/callback?code=xxx
        → backend exchanges code, decodes id_token, upserts user
        → redirects to frontend: http://localhost:5173?token=<jwt>
          → frontend reads ?token, calls loginWithOAuthToken, connects socket
```

Files changed:

- `backend/src/controllers/oauth.controller.ts` *(new)* — `googleRedirect` and `googleCallback`. Code exchange via `fetch('https://oauth2.googleapis.com/token')`. id_token decoded with `Buffer.from(payloadB64, 'base64url')`.

- `backend/src/routes/auth.routes.ts` — Added `GET /api/v1/auth/google` and `GET /api/v1/auth/google/callback`.
- `backend/.env` — Added `GOOGLE_CLIENT_ID`, `GOOGLE_CLIENT_SECRET`, `OAUTH_CALLBACK_URL`, `FRONTEND_URL`.
- `frontend/src/stores/authStore.ts` — Added `loginWithOAuthToken(token)` — writes to `localStorage` and sets `isAuthenticated: true`.
- `frontend/src/App.jsx` — "Continue with Google" button in `!otpSent` branch. Mount-time `useEffect` reads `?token=` after callback redirect, clears URL param with `history.replaceState`.

Task #15 — Google Maps Integration

Problem: Coordinates were hardcoded (Chennai Airport → Velachery). Distance used Haversine × 1.35. Location inputs were plain text.

Solution: Google Places Autocomplete on both inputs. Google Distance Matrix API for real road distances. Chips carry actual coordinates.

Files changed:

- `backend/src/services/fare.service.ts` — Added `getDistanceMatrix(originLat, originLng, destLat, destLng): Promise<number>` using native `fetch`. Returns road distance in km. Falls back to Haversine × 1.35 if API key absent or call fails. Added `distanceKmOverride?` parameter to `estimateFares()`.
- `backend/src/controllers/quotes.controller.ts` — Calls `getDistanceMatrix()` when lat/lng present; passes result as override to `estimateFares()`. Falls back to Haversine when coordinates missing (no breaking change).
- `backend/.env` — Added `GOOGLE_MAPS_API_KEY`.
- `frontend/.env` — Added `VITE_GOOGLE_MAPS_API_KEY` (Vite requires `VITE_` prefix for browser access via `import.meta.env`).
- `frontend/src/stores/bookingStore.ts` — Removed hardcoded mock coordinates. `fetchQuotes()` and `createBooking()` now accept optional `pickupLat`, `pickupLng`, `dropoffLat`, `dropoffLng` and forward them to the API.
- `frontend/src/App.jsx` — `useJsApiLoader` hook loads Maps JS SDK. `<Autocomplete>` wrappers on both inputs (India restriction). `sourceCoords` / `destCoords` state. Chip arrays changed from plain strings to `{ label, lat, lng }` objects. `findRides()` and `bookRide()` pass real coordinates.

Google Cloud APIs required: Maps JavaScript API, Places API, Distance Matrix API.

Integration Issues Found & Fixed (Phase 4 Session)

#	Issue	Root Cause	Resolution
1	"Continue with Google" button not visible	Button rendered below phone frame's overflow boundary	Moved button inside `!otpSent` branch, directly below Send OTP button

#	Issue	Root Cause	Resolution
2	OAuth login didn't redirect to transport screen	`loginWithOAuthToken` action missing from authStore	Added action; writes token to localStorage, sets isAuthenticated
3	Stale `movzz_token` blocked auth screen	Previous session token kept user in authenticated state	User clears with `localStorage.clear(); location.reload()` — documented as known fix
4	Circular import: booking.service ↔ index.ts	booking.service needed `io` from index.ts which imports booking.service	Solved via `config/socket.ts` singleton (`setIo` / `getIo`)
5	"This page can't load Google Maps"	Maps JavaScript API and Places API not yet enabled in GCP Library	Enabled both APIs; confirmed Distance Matrix API also enabled
6	Chip onClick broke after chip data change	Chips changed from `string[]` to `{ label, lat, lng }[]` but JSX still used `chip` as string	Updated JSX to use `chip.label`, `chip.lat`, `chip.lng`
7	Manual typing in Autocomplete input didn't update state	`defaultValue` makes input uncontrolled — `destination` state never updated	Changed to `value` + `onChange` handler; manual typing clears coords (triggers Haversine fallback)

Key Technical Deliverables

Fare Engine (`fare.service.ts`)

562 lines of production-grade fare estimation. Three cab tiers (Economy ₹12/km, Comfort ₹15/km, Premium ₹18/km) with time-of-day surge pricing: morning rush 1.2x, evening rush 1.3x, late night 1.15x, airport supplement +0.05x. Now accepts optional `distanceKmOverride` to use real road distance from Distance Matrix instead of Haversine approximation.

Socket.IO Architecture

Per-user rooms (`socket.join(userId)`). Singleton pattern avoids circular imports. `booking:state_changed` emitted on every state transition — payload matches `GET /bookings/:id` response shape for drop-in compatibility. Polling kept as a reconnect fallback.

Google OAuth (No Passport.js)

Direct REST implementation using Node 22 native `fetch`. id_token decoded by base64url-decoding the JWT middle segment — no verification needed as Google already verified it. No new npm packages added.

Google Maps Autocomplete

`@react-google-maps/api <Autocomplete>` component with `componentRestrictions: { country: 'in' }`. Falls back to plain `<input>` when SDK not loaded (no API key = no broken UI). Coordinates flow:

select place → `onPlaceChanged` → `setSourceCoords` / `setDestCoords` → `fetchQuotes` → `getDistanceMatrix` → `estimateFares(distanceKmOverride)`.

Current Codebase Health

Metric	Current State
Fare tests	52/52 passing (3ms runtime)
TypeScript errors	Zero across all modified files
POST /quotes endpoint	Functional — Distance Matrix wired, Haversine fallback active
OTP authentication	Functional — end-to-end working
Google OAuth	Functional — "Continue with Google" tested end-to-end
Quote-to-booking	Functional — coordinates wired, falls back to Haversine when Maps unavailable
Booking state updates	Real-time via Socket.IO — polling removed
Places Autocomplete	▨ Blocked — code complete, awaiting GCP API enablement + billing
API response consistency	All controllers use `{ success, data }` pattern

Pull Request Log

PR	Description	Date	Status
#6	Restore mode-specific fare engine + BIKE enum correction	Feb 26	☑ Merged
#8	Integration cleanup: isActive fix, fare wiring, metro extraction, tag unification, dead code removal	Feb 26	☑ Merged
#9	Fix auth controller response format — align `{ success, data }` convention	Feb 26	☑ Merged
#10	Phase 4: WebSockets (Socket.IO), Google OAuth, Google Maps	Feb 28	☑ Merged

MOVZZ | Integration Roadmap v3.0 | Feb 28, 2026 | ***14/15 tasks complete — Task #15 blocked on GCP***