# ServiceOps Platform Documentation

## 1. Purpose & Scope

ServiceOps is an end-to-end platform that manages on-demand service jobs across administrators, vendors, customers, and guests. This document summarizes every functional area, workflows, and supporting infrastructure so stakeholders can operate or extend the system.

## 2. Technology Stack

- **Frontend:** React (CRA), React Router, Context API, CSS modules
- **Backend:** Node.js/Express, MongoDB via Mongoose
- **Real-time UX:** Context polling for live vendor data, custom notification system, Google Maps + Leaflet maps
- **Tooling:** Babel, ESLint, dotenv, nodemon, npm scripts

## 3. Repository Layout

```
serviceops/ +-- client/ # React frontend � +-- public/ � +-- src/ � +-- components/ � +--
contexts/ � +-- lib/ � +-- pages/ � +-- utils/ � +-- App.jsx / App.css � +-- index.js +--
server/ # Node/Express backend +-- src/ � +-- lib/ � +-- models/ � +-- routes/ � +-- index.js
+-- server.js
```

## 4. Frontend Architecture

### 4.1 Shell & Global Services

- **Topbar (components/Topbar.jsx)**: Responsive navigation, role-aware links, notification badge, mobile drawer. Scroll-aware hide/show logic.
- **Routing (App.jsx)**: Wraps providers (AuthProvider, NotificationsProvider, LiveVendorsProvider) and defines protected routes per role.
- **Contexts:**
  - AuthContext � authentication state and role checks.
  - NotificationsContext � toast queue, dedupe, persistence, audit forwarding.
  - LiveVendorsContext � polls vendor telemetry, surfaces admin notifications, audit entries.
- **Utilities:**
  - utils/geo.js � coordinate derivation, haversine distance helpers.
  - utils/auditLog.js � local audit trail storage with DOM events.

### 4.2 Admin Pages

1. **Dashboard** � KPI cards, quick actions, activity feed, fully responsive.
2. **Jobs** � Search/filter, solo mode, modal job creation, status transitions, vendor reassignment, live notifications.
3. **Live Map** � City filter, active toggle, HQ configuration, vendor focus, Google/Leaflet parity with HQ routing.
4. **Vendors** � Vendor directory, profile drawer, activation controls, performance stats.
5. **Reports** � Filterable analytics, CSV/PDF export, insights, audit trail section.
6. **Financials** � Commission overview, reconciliation tables.
7. **Settings** � Workflow toggles, defaults, poll intervals, compliance editor.

### 4.3 Vendor Experience

- **VendorApp** � Tabs for Open and Assigned jobs, bid management, note translation (Spanish/English), auto-refresh, travel estimates, route guidance.

### 4.4 Customer Experience

- **CustomerLogin** � OTP/password flow.
- **CustomerDashboard** � Live map with route & distance, status timeline, driver card, quick actions, history list.
- **CustomerHome** � Summary of active jobs, shortcuts to key flows.
- **CustomerRequest/Intake** � Job submission wizard (guest intake disabled via backend).

### 4.5 Guest/Public Flows

- **GuestJobTracker** � Token-based status page with map, timeline, vendor updates, bidding, review funnel.
- **PublicVendorBid / PublicCustomerChoose** � Standalone flows for open bidding and customer selection.

### 4.6 Notifications Center

Inbox view listing stored notifications, filters, mark-read controls, deep links.

### 4.7 Landing Page

Marketing content, feature highlights, CTAs, fully responsive.

## 5. Backend Architecture

### 5.1 Express App (server/src/index.js)

- Configures CORS (localhost + production domains), JSON parsing, dotenv, error handler.
- Mounts route modules: /api/jobs, /api/bids, /api/vendors, /api/customers, /api/reports, etc.

### 5.2 Routes & Business Logic

- **Jobs ( outes/jobs.js)**: List/create/update jobs, enforce status transitions via STATUSES, handle completion, share links.
- **Bids ( outes/bids.js)**: Vendor bidding endpoints, selecting winning bid, SMS notifications.
- **Vendors ( outes/vendors.js)**: Admin vendor management API.
- **Reports ( outes/reports.js)**: Aggregated metrics and export endpoints.
- **Auth** routes for admin, vendor, customer login (not detailed here).

### 5.3 Models

- **Job** � Lifecycle tracking, vendor/customer tokens, coordinates, bidding, payments, flags. Indices managed via schema-level .index() calls.
- **Vendor** � Profile, services, runtime telemetry (lat, lng, □ctive, lastSeenAt).
- **Customer** � Profile, contact info, saved vehicles, OTP metadata, guest tokens.
- **Bid**, **Review Funnel**, **Audit** (client-side) support modules.

### 5.4 Notifications & Audit

- Notifications emitted through NotificationsContext and SMS (stubbed via lib/notifier.js).
- Audit events persisted in client local storage; surfaced on Admin Reports page.

### 5.5 Environment & Config

- .env (server): MONGO_URI, API_PORT, SMS credentials, base URLs.
- .env (client): REACT_APP_API_URL, REACT_APP_GOOGLE_MAPS_KEY.
- config/env.js resolves base URLs and map keys with sensible fallbacks.

## 6. Detailed Workflows

### 6.1 Job Lifecycle

1. Creation by admin or customer.
2. Vendors view open jobs (bid mode open), submit bids.
3. Admin/customer selects bid ? status Assigned, live vendor notified.
4. Vendor transitions status (OnTheWay, Arrived, Completed), enabling live routing.
5. Completion triggers completeJobWithPayment (commission, payment status).
6. Audit logs record transitions; reports reflect totals.

### 6.2 Vendor Note Translation

- Uses MyMemory API for English ? Spanish.
- State caches translation per job with loading/error states.

### 6.3 Live Mapping

- Google Maps (with key) renders directions and markers for vendors/HQ/customer.
- Leaflet fallback draws polylines and distance chips even without Google key.
- Admin HQ stored in local storage for route calculations.

### 6.4 Notifications & Audit Trail

- Notification events: new job, status change, vendor assigned/unassigned, vendor online/offline, vendor removed.
- Audit log accessible via Admin Reports; clearable per admin instance.

## 7. User Guides

### 7.1 Administrator

- **Login:** /admin/login
- **Dashboard:** Monitor KPIs, recent actions.
- **Jobs:** Manage job lifecycle, use Create Job modal, observe live notifications.
- **Live Map:** Set HQ (manual coords or geolocation), focus vendors, monitor routes.
- **Reports:** Filter insights, export CSV/PDF, review audit trail.
- **Settings:** Toggle workflow features, defaults, and compliance.

### 7.2 Vendor

- **Login:** /vendor/login
- **Open Jobs:** Review cards, translate notes, place/update bids.
- **Assigned:** Track progress, navigate to pickup, manage status.
- **Auto-refresh:** Enabled by default; manual refresh button available.

### 7.3 Customer

- **Login:** /customer/login
- **Dashboard:** Live job view, share/copy link, call driver, view history.
- **Requests:** Submit new jobs (customer intake) or use admin-created invitations.

### 7.4 Guest

- **Tracking:** /status/:token (UI ready; server currently disables guest endpoints).
- **Vendor Bid:** /bid/:vendorToken
- **Customer Selection:** /choose/:customerToken

## 8. APIs & Environment

- **Key Endpoints:**
  - GET /api/jobs � filter by status/search string.
  - POST /api/jobs � create job (optional vendor assignment).
  - PATCH /api/jobs/:id � update priority/status/vendor.
  - POST /api/jobs/:id/complete � finalize job with payment.
  - POST /api/bids/:bidId/select � choose winning vendor.
  - GET /api/reports/range � analytics data.
- **Environment Setup:**
  - Install dependencies ( pm install in client/ and server/).
  - Configure .env files.
  - Start backend ( pm run dev in server/), frontend ( pm start in client/).

## 9. Known Warnings / Cleanup

- Lint: unused variables (e.g., □endorsError, □boutHighlights), BOM markers in some page files.
- Translation API throttling not implemented�monitor usage for production.

## 10. Recent Updates

- Admin nav alignment adjusted.
- Live map fallback now renders HQ routes & distance overlays.
- Vendor note translation toggle with caching/error handling.
- Admin notifications + audit entries for job & vendor events.
- Admin live map HQ configuration persisted locally.
- Job schema index warnings resolved.
- Status transitions unified via STATUSES list.

## 11. Appendices

### 11.1 Troubleshooting

- **STAGES is not defined** � ensure updated outes/jobs.js using STATUSES.
- **Duplicate index warnings** � re-run with updated schema, drop duplicate indexes if present.
- **Google Maps fallback** � Leaflet route preview used when key absent.

### 11.2 Future Enhancements

- Persist audit logs server-side.
- Move from polling to WebSockets for live updates.
- Provide admin-accessible notification templates.
- Integrate richer analytics dashboards.

---