🚀 Live Parcel Tracking Setup Guide

📦 Complete Installation

1. Install Dependencies

```bash
# Install Google Maps and WebSocket dependencies
npm install @googlemaps/js-api-loader ws concurrently

# OR using yarn
yarn add @googlemaps/js-api-loader ws concurrently

# OR using pnpm
pnpm add @googlemaps/js-api-loader ws concurrently
```

2. Environment Configuration

Create `.env.local` in your project root:

```bash
# Google Maps API Key
NEXT_PUBLIC_GOOGLE_MAPS_API_KEY=your_actual_api_key_here

# WebSocket Server URL
NEXT_PUBLIC_WS_URL=ws://localhost:3001

# WebSocket Port
WS_PORT=3001
```

3. Google Maps Setup

1. Go to Google Cloud Console
2. Create a new project or select existing
3. Enable these APIs:
   - ✅ Maps JavaScript API
   - ✅ Directions API
   - ✅ Distance Matrix API

- ✅ Geocoding API (optional)
- ✅ Places API (optional)

4. Create credentials → API Key
5. Copy the API key to `.env.local`

4. Project Structure

```
your-project/
├── app/
│   └── api/
│       └── drivers/
│           └── [driverId]/
│               └── location/
│                   └── route.ts          # HTTP API endpoint
├── components/
│   └── SupperAdmin/
│       ├── TrackParcelModal.tsx          # Modal component
│       ├── ParcelMap.tsx                 # Live map component
│       └── DriverLocationSender.tsx      # Driver app component
├── server/
│   └── websocket.js                      # WebSocket server
├── .env.local                            # Environment variables
└── package.json
```

🚀 Running the Application

Development Mode

bash

```
# Terminal 1: Run Next.js app
npm run dev

# Terminal 2: Run WebSocket server
npm run ws:dev

# OR run both together
npm run dev:all
```

Production Mode

```bash
# Build Next.js app
npm run build

# Start both services
npm start & node server/websocket.js
```

📱 How Live Tracking Works

Architecture Overview

```
Driver Device (GPS)
    ↓
WebSocket Connection (/driver/:driverId)
    ↓
WebSocket Server (Node.js)
    ↓
Broadcast to Tracking Clients (/track/:driverId)
    ↓
Admin Dashboard (Map Updates)
```

Data Flow

1. **Driver Side:**
    - Driver's device sends GPS coordinates via WebSocket
    - Location updates every 3-5 seconds
    - Includes: latitude, longitude, speed, heading
2. **Server Side:**
    - WebSocket server receives driver location
    - Validates and broadcasts to tracking clients
    - Maintains active connections map
3. **Admin Side:**
    - Connects via WebSocket when modal opens
    - Receives real-time location updates

- Smoothly animates marker on map
- Calculates ETA and distance

Fallback Mechanism

If WebSocket fails, the system automatically falls back to HTTP polling:

- Polls every 5 seconds via `/api/drivers/:driverId/location`
- Less real-time but ensures tracking continues
- Automatically switches back to WebSocket when available

🔧 Configuration Options

WebSocket Server (server/websocket.js)

```javascript
// Update interval for location updates
const UPDATE_INTERVAL = 3000; // 3 seconds

// Port configuration
const PORT = process.env.WS_PORT || 3001;
```

Map Component (ParcelMap.tsx)

```typescript
// Map zoom level
zoom: 13,

// Animation steps for smooth movement
const steps = 50;

// Polling interval (HTTP fallback)
const intervalId = setInterval(pollLocation, 5000);
```

🎯 Features Implemented

✅ Real-Time Location Updates

- WebSocket connection for instant updates

- HTTP polling as fallback
- Smooth marker animation

✅ **Live Map Tracking**

- Driver position with animated marker
- Route visualization with Directions API
- Pickup and delivery markers
- Distance and ETA calculations

✅ **Visual Indicators**

- Live tracking badge
- Pulsing driver marker
- Last update timestamp
- Connection status indicator

✅ **User Experience**

- Auto-centering on driver
- Recenter button
- Legend for markers
- Responsive design

🧪 Testing

Test Live Tracking

1. **Start servers:**

```bash
npm run dev:all
```

2. **Open parcel tracking modal** for a parcel with status "ongoing"
3. **The WebSocket server will simulate driver movement** (demo mode)
4. **Watch the driver marker move** on the map in real-time

Remove Simulation (Production)

In `server/websocket.js`, remove or comment out:

```javascript
// Remove this function in production
// simulateDriverMovement();
```

📱 Driver Mobile App Integration

For Real GPS Data

1. **Create driver mobile app** (React Native, Flutter, etc.)
2. **Use the DriverLocationSender component** as reference
3. **Connect to WebSocket:** `ws://your-domain/driver/:driverId`
4. **Send GPS coordinates:**

```json
{
  "latitude": 23.8103,
  "longitude": 90.4125,
  "speed": 45,
  "heading": 180,
  "timestamp": "2026-01-10T10:00:00Z"
}
```

Browser Geolocation API

```javascript
```

```
navigator.geolocation.watchPosition(
  (position) => {
    const location = {
      latitude: position.coords.latitude,
      longitude: position.coords.longitude,
      speed: position.coords.speed,
      heading: position.coords.heading,
    };
    ws.send(JSON.stringify(location));
  },
  { enableHighAccuracy: true }
);
```

🔒 Security Considerations

Production Checklist

- ✅ Use WSS (secure WebSocket) in production
- ✅ Implement authentication for WebSocket connections
- ✅ Validate driver IDs before accepting connections
- ✅ Rate limit location updates
- ✅ Encrypt sensitive data
- ✅ Implement proper error handling

Authentication Example

javascript

```
// In WebSocket server
ws.on('connection', (ws, req) => {
  const token = new URL(req.url, 'http://localhost').searchParams.get('token');

  if (!validateToken(token)) {
    ws.close(1008, 'Unauthorized');
    return;
  }

  // Continue with connection...
});
```

🐛 Troubleshooting

Map not loading?

- Check Google Maps API key in `.env.local`
- Verify APIs are enabled in Google Cloud Console
- Check browser console for errors

WebSocket not connecting?

- Ensure WebSocket server is running: `npm run ws:dev`
- Check WebSocket URL in `.env.local`
- Verify port 3001 is not in use

Location not updating?

- Check WebSocket connection status
- Verify driver ID matches
- Check browser/server console logs

Smooth animation not working?

- Clear browser cache
- Check if `google.maps` is loaded
- Verify marker exists before updating

📊 Performance Optimization

Best Practices

1. **Throttle location updates** to 3-5 seconds minimum
2. **Use bounds fitting** instead of constant centering
3. **Implement connection pooling** for WebSocket
4. **Cache map tiles** for offline support
5. **Lazy load map component** with dynamic import

🎨 Customization

Change Marker Colors

```typescript
// In ParcelMap.tsx
fillColor: "#YOUR_COLOR",
```

Adjust Update Frequency

```typescript
// In ParcelMap.tsx
setInterval(pollLocation, YOUR_INTERVAL);
```

Custom Route Styling

```typescript
polylineOptions: {
  strokeColor: "#YOUR_COLOR",
  strokeWeight: 5,
  strokeOpacity: 0.7,
}
```

📞 Support

For issues or questions:

1. Check the troubleshooting section
2. Review browser/server console logs
3. Verify environment variables
4. Test with demo simulation first

🎉 **You're all set!** Your live parcel tracking system is ready to track deliveries in real-time.