









# PHP Shipment Tracking System with Database Integration

A comprehensive shipment tracking system built with PHP, MySQL, and JavaScript that allows real-time tracking of packages with interactive maps and database management.

## Features

-  Interactive map visualization using Leaflet.js
-  Real-time shipment tracking with database integration
-  Click-to-update location functionality
-  Responsive design for mobile and desktop
-  Admin panel for managing shipments
-  Geocoding with location caching
-  Shipment history tracking
-  Modern, animated UI with smooth transitions

## Requirements

- PHP 7.4 or higher
- MySQL 5.7 or higher (or MariaDB 10.2+)
- Apache/Nginx web server
- Internet connection for map tiles and geocoding

## Installation

### Step 1: Clone or Download Files

Create the following directory structure in your web server root:

```
shipment-tracking/  
├── config/  
│   └── database.php  
├── classes/  
│   └── Shipment.php  
├── api/  
│   └── handler.php  
├── index.php  
├── admin.php  
└── shipment_tracking.sql
```

## Step 2: Database Setup

### 1. Create the database and tables:

```
sql  
  
mysql -u root -p < shipment_tracking.sql
```

Or import through phpMyAdmin:

- Open phpMyAdmin
- Create a new database named `shipment_tracking`
- Import the `shipment_tracking.sql` file

### 2. Configure database connection: Edit `config/database.php` and update the database credentials:

```
php  
  
private $host = 'localhost';    // Your database host  
private $db_name = 'shipment_tracking'; // Database name  
private $username = 'your_username'; // Your database username  
private $password = 'your_password'; // Your database password
```

## Step 3: File Permissions

Ensure proper permissions for the web server:

```
bash  
  
chmod 755 /path/to/shipment-tracking/  
chmod 644 /path/to/shipment-tracking/*.php  
chmod 644 /path/to/shipment-tracking/config/*.php  
chmod 644 /path/to/shipment-tracking/classes/*.php  
chmod 644 /path/to/shipment-tracking/api/*.php
```

## Step 4: Web Server Configuration

### Apache (.htaccess)

Create a `.htaccess` file in the root directory:

```
apache

RewriteEngine On

# Allow API requests
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^api/(.*)$ api/handler.php [QSA,L]

# Security headers
<IfModule mod_headers.c>
    Header always set X-Content-Type-Options nosniff
    Header always set X-Frame-Options DENY
    Header always set X-XSS-Protection "1; mode=block"
</IfModule>
```

### Nginx

Add to your server block:

```
nginx

location /api/ {
    try_files $uri $uri/ /api/handler.php?$query_string;
}

location ~ \.php$ {
    fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    include fastcgi_params;
}
```

## Usage

### Admin Panel

1. **Access the admin panel:** Navigate to `http://yoursite.com/admin.php`

## 2. Create a new shipment:

- Enter tracking number (must be unique)
- Set origin location (e.g., "Dubai, UAE")
- Set destination location (e.g., "Los Angeles, CA, USA")
- Add descriptions for better context

## 3. Update shipment locations:

- Select an existing shipment
- Choose location type (origin, current, or destination)
- Enter new location name
- Add optional description

## Public Tracking

1. **View shipment map:** Navigate to `http://yoursite.com/index.php?tracking=YOUR_TRACKING_NUMBER`

## 2. Interactive features:

- Click "Enable Click Mode" to update location by clicking on map
- Use dropdown to select from predefined locations
- Enter custom location names in the text field

## API Endpoints

The system provides RESTful API endpoints:

### GET Requests:

- `GET /api/handler.php?action=get_shipment&tracking=TRACKING_NUMBER`
- `GET /api/handler.php?action=get_coordinates&location=LOCATION_NAME`
- `GET /api/handler.php?action=get_history&tracking=TRACKING_NUMBER`

### POST Requests:

javascript

```
// Update location
fetch('/api/handler.php', {
  method: 'POST',
  headers: {'Content-Type': 'application/json'},
  body: JSON.stringify({
    action: 'update_location',
    tracking_number: 'SH001234567890',
    location_type: 'current',
    location: 'New York, USA',
    description: 'Package arrived at sorting facility'
  })
});

// Create new shipment
fetch('/api/handler.php', {
  method: 'POST',
  headers: {'Content-Type': 'application/json'},
  body: JSON.stringify({
    action: 'create_shipment',
    tracking_number: 'SH001234567891',
    origin: 'Tokyo, Japan',
    destination: 'London, UK',
    origin_description: 'Dispatched from Tokyo warehouse',
    destination_description: 'Delivery to London office'
  })
});
```

## Database Schema

### Tables:

1. **shipments** - Main shipment information
2. **location\_cache** - Cached geocoding results for performance
3. **shipment\_history** - Historical tracking updates

### Sample Data:

The installation includes sample data:

- Tracking number: SH001234567890
- Route: Dubai → Toronto → Los Angeles

# Customization

## Adding New Predefined Locations:

Edit the locations in both files:

### 1. Admin panel (admin.php):

php

```
<option value="Berlin, Germany">Berlin, Germany</option>
```

### 2. Main page (index.php):

javascript

```
<option value="Berlin, Germany">Berlin, Germany</option>
```

## Styling:

The CSS is embedded in each file for easy customization. Key color variables:

- Primary green:  #059669
- Secondary orange:  #fcb529
- Background gradient:  #667eea to  #764ba2

## Map Configuration:

Modify map settings in the JavaScript section:

javascript

```
const map = L.map("map", {  
  zoomControl: true,  
  scrollWheelZoom: true,  
  // Add more options here  
}).setView([30, 0], 2); // [latitude, longitude], zoom level
```

## Troubleshooting

### Common Issues:

#### 1. Database Connection Error:

- Check database credentials in config/database.php

- Ensure MySQL service is running
- Verify database user permissions

## 2. Geocoding Not Working:

- Check internet connection
- Nominatim API might be temporarily unavailable
- Consider implementing API key for better reliability

## 3. Map Not Loading:

- Check browser console for JavaScript errors
- Ensure Leaflet CDN is accessible
- Verify HTTPS if using HTTPS site

## 4. Permission Errors:

- Check file permissions
- Ensure web server can read all PHP files
- Check SELinux settings if applicable

## Debug Mode:

Enable PHP error reporting for development:

```
php
<?php
error_reporting(E_ALL);
ini_set('display_errors', 1);
// Rest of your code
?>
```

## Security Considerations

1. **Input Validation:** All user inputs are sanitized using prepared statements
2. **SQL Injection Protection:** PDO with parameter binding is used throughout
3. **XSS Prevention:** All output is escaped using `htmlspecialchars()`
4. **Rate Limiting:** Consider implementing rate limiting for API endpoints
5. **HTTPS:** Use HTTPS in production for secure data transmission

## License

This project is open source. Feel free to modify and distribute according to your needs.

## Support

For issues and questions:

1. Check the troubleshooting section
2. Review the code comments for implementation details
3. Test with the provided sample data first

## Version History

- **v1.0** - Initial release with full functionality
  - Interactive mapping
  - Database integration
  - Admin panel
  - Responsive design
  - API endpoints