WordPress AI Order Tracker Plugin Structure and Functionality

This document outlines the file structure and describes the functionality of the WordPress AI Order Tracker plugin, based on the provided code and UI images. The goal is to provide a comprehensive overview that allows for easy continuation of development by any AI.

1. Plugin File Structure

The plugin's core files are located in the `/ai-order-tracker` directory. Below is a detailed breakdown of the directory and file organization:

```
/ai-order-tracker
├── admin
│   ├── admin-init.php
│   ├── class-admin-couriers.php
│   ├── class-admin-settings.php
│   ├── class-admin-zones.php
│   ├── css
│   │   └── admin.css
│   ├── js
│   │   ├── admin-couriers.js
│   │   └── admin-zones.js
│   └── partials
│       ├── order-tracking-meta.php
│       ├── settings-fields.php
│       └── zone-map.php
├── assets
│   ├── animations
│   │   ├── processing.json
│   │   └── shipped.json
│   ├── couriers
│   ├── geo
│   │   └── countries.json
│   └── libs
│       ├── leaflet-1.url
│       ├── leaflet.url
│       └── lottie-player.url
├── includes
│   ├── class-courier-manager.php
│   └── class-cron.php
```

```
│   ├── class-database.php
│   ├── class-dependencies.php
│   ├── class-helpers.php
│   ├── class-real-time-api.php
│   ├── class-real-time-tracking.php
│   ├── class-security.php
│   ├── class-tracking-engine.php
│   └── class-zone-manager.php
├── languages
│   └── ai-order-tracker.pot
├── public
│   ├── class-tracking-shortcode.php
│   ├── css
│   │   └── public.css
│   ├── js
│   │   ├── progress-animations.js
│   │   └── public.js
│   └── templates
│       ├── timeline-display.php
│       ├── tracking-display.php
│       └── tracking-form.php
├── tests
│   ├── integration-tests
│   └── unit-tests
├── vendors
├── ai-order-tracker.php
├── README.md
└── uninstall.php
```

2. Plugin Functionality Overview

The AI Order Tracker plugin appears to be a comprehensive solution for managing and displaying order tracking information within a WordPress (likely WooCommerce) environment. It features both 'fake' and 'real' tracking data management, automated tracking updates, and a user-facing tracking interface. The plugin also includes extensive configuration options for delivery zones and courier management.

Key functionalities observed from the images and file structure include:

- **Automated Tracking:** The plugin aims to automate tracking updates without manual intervention from the admin. This involves fetching order data, determining location, and displaying progress based on reliability percentages.
- **Dual Tracking System (Fake vs. Real):** It supports both automatically generated 'fake' tracking IDs for initial display and 'real' tracking IDs that can be manually entered by the admin. The system is designed to prevent users from knowing if an ID is fake.

- **User-Editable Data (Delivery Zones):** Admins can configure delivery timeframes based on countries, states, and major cities, categorized into zones (e.g., North Zone, South Zone). This data is central and can be updated by the user.
- **User-Facing Tracking Interface:** A dedicated front-end page allows users to track their orders, displaying delivery status, package location, and a detailed package journey timeline, similar to an e-commerce giant's tracking page.
- **Admin Management Interfaces:** The backend provides interfaces for managing settings, couriers, and delivery zones.

3. Detailed Component Breakdown

3.1. Core Plugin Files

- `ai-order-tracker.php` : This is the main plugin file, responsible for plugin initialization, loading dependencies, and registering hooks.
- `uninstall.php` : Handles the cleanup process when the plugin is uninstalled, ensuring all data and settings are removed from the database.
- `README.md` : Provides basic information about the plugin.

3.2. Admin Section (`/admin`)

This directory contains all the files related to the WordPress admin interface for managing the plugin's settings and data.

- `admin-init.php` : Likely handles the initialization of the admin area, including menu pages and scripts.
- `class-admin-couriers.php` : Manages the administration of courier companies, allowing admins to add, edit, or delete courier information.
- `class-admin-settings.php` : Handles the general settings of the plugin, possibly including API keys, general tracking options, and reliability percentages for fake tracking.
- `class-admin-zones.php` : Manages the delivery zone configurations, allowing admins to define and update delivery timeframes for different regions.
- `css/admin.css` : Stylesheets for the admin interface.
- `js/admin-couriers.js` : JavaScript for courier management functionalities in the admin area.
- `js/admin-zones.js` : JavaScript for delivery zone management functionalities in the admin area.
- `partials/order-tracking-meta.php` : Likely a template for displaying or editing order tracking metadata within the WordPress order edit screen.

- `partials/settings-fields.php` : Contains the HTML structure for the plugin's settings fields.
- `partials/zone-map.php` : Possibly a template for visualizing or interacting with delivery zones, perhaps using a map interface.

3.3. Assets (`/assets`)

This directory stores various static assets used by the plugin.

- `animations/processing.json` , `animations/shipped.json` : These are likely Lottie animations (JSON files) used to visually represent different tracking statuses (e.g., processing, shipped) on the user-facing tracking page.
- `couriers/` : This directory is currently empty but is likely intended to store assets related to courier companies, such as logos.
- `geo/countries.json` : Contains geographical data, specifically country information, which is crucial for the delivery zone configuration.
- `libs/leaflet-1.url` , `libs/leaflet.url` , `libs/lottie-player.url` : These files likely contain URLs or references to external JavaScript libraries like Leaflet (for mapping, potentially used in `zone-map.php`) and Lottie Player (for playing JSON animations).

3.4. Includes (`/includes`)

This directory contains core classes and functionalities that are shared across the plugin, or perform background tasks.

- `class-courier-manager.php` : Manages courier-related logic, possibly interacting with external courier APIs or handling courier data.
- `class-cron.php` : Handles scheduled tasks (cron jobs) for the plugin, which could be used for automated tracking updates or data synchronization.
- `class-database.php` : Manages database interactions, including creating tables, inserting, updating, and retrieving data for the plugin.
- `class-dependencies.php` : Handles the checking and loading of plugin dependencies.
- `class-helpers.php` : Contains utility functions and helper methods used throughout the plugin.
- `class-real-time-api.php` : Likely responsible for interacting with external real-time tracking APIs to fetch actual tracking data.
- `class-real-time-tracking.php` : Integrates real-time tracking data into the plugin's system, possibly processing and storing it.

- `class-security.php` : Implements security measures for the plugin, such as nonce verification and data sanitization.
- `class-tracking-engine.php` : The core logic for the tracking system, handling the generation of fake IDs, matching with real data, and calculating reliability.
- `class-zone-manager.php` : Manages the logic for delivery zones, including data storage and retrieval for country, state, and city information.

3.5. Languages (`/languages`)

- `ai-order-tracker.pot` : The Portable Object Template file for internationalization, allowing the plugin to be translated into different languages.

3.6. Public Section (`/public`)

This directory contains files related to the front-end display and user interaction.

- `class-tracking-shortcode.php` : Defines the shortcode that users can use to embed the tracking form and display on their WordPress pages.
- `css/public.css` : Stylesheets for the public-facing tracking interface.
- `js/progress-animations.js` : JavaScript for handling the progress animations (Lottie animations) on the tracking page.
- `js/public.js` : General JavaScript for the public-facing tracking interface, handling form submissions and dynamic content updates.
- `templates/timeline-display.php` : Template for displaying the detailed package journey timeline.
- `templates/tracking-display.php` : The main template for displaying the order tracking results.
- `templates/tracking-form.php` : Template for the user-facing tracking input form.

3.7. Tests (`/tests`)

- `integration-tests/` : Contains files for integration tests.
- `unit-tests/` : Contains files for unit tests.

3.8. Vendors (`/vendors`)

This directory is currently empty but is typically used to store third-party libraries or vendor-specific code that the plugin utilizes.

4. Plugin Functionality in Detail (Based on UI Images)

4.1. How it will work (IMG_3795.jpeg)

This diagram illustrates the core workflow of the plugin:

- **Order Fetching:** The process begins with fetching order data, likely from WooCommerce, via its REST API. This data includes the order and its associated location.
- **Location Matching:** The fetched order location is matched with internal location data. If the admin has set a delivery range (e.g., 5 to 10 days for Delhi), the system starts showing order packed and processed statuses based on a percentage of reliability.
- **Automatic Updates:** The system is designed to work automatically, eliminating the need for manual updates of fake track IDs by the admin.
- **Real Tracking ID:** Users have the option to manually enter a real tracking number, with options for courier companies. This can be updated via order edit and allows for sending email notifications.

4.2. Tracking System (IMG_3794.jpeg)

This image clarifies the distinction between

the 'Fake Tracking data' and 'Real Tracking data' mechanisms:

Fake Tracking Data:

1. **Auto-generated ID:** A fake tracking ID is automatically generated, appearing real but being distinct from an actual tracking ID.
2. **Location Check:** The plugin checks the customer's order location.
3. **Admin End Data:** It fetches admin-defined data regarding the number of days for a particular location.
4. **International/Domestic:** Supports two types of users: international businesses and domestic ones.
5. **Progressive Display:** The system displays progress based on set delivery ranges and processing days. For example, if a website has a 5-7 day delivery range and 2 days processing to Uttarakhand, the order will show as processed for 2 days, then in transit for 5-7 days, and so on.
6. **Reliability Percentage:** An admin-set percentage of reliability determines the display. For instance, if 50% reliable, the order won't be shown as delivered until 7 + 4 days.

7. **UI Emulation:** The UI is designed to mimic big e-commerce tracking interfaces like Amazon.
8. **Automatic Operation:** The system is fully automatic, requiring no manual intervention from the admin.

Real Tracking Data:

1. **Admin Option:** Admins can manually add real tracking IDs.
2. **Courier List:** A list of global couriers with their tracking page URLs is available.
3. **URL Click:** When a URL is clicked, the real tracking ID uploaded by the admin is copied, allowing the user to paste and track.
4. **Admin Real ID Option:** Admins have the option to add the real tracking ID.

Crucially, the user is prevented from knowing if the tracking ID is fake.

4.3. User Editable Data (IMG_3793.jpeg)

This image details the data that can be configured by the user, primarily focusing on delivery zones:

- **Centralized Data:** All country, state, and major city data is centralized and fetchable via a license code.
- **Zone-based Delivery:** The world is divided into zones (North, East, West, Central, South), each with an editable date range.
- **International Sellers:** For international sellers, country data is linked to zones.
- **Domestic Sellers:** For domestic sellers, state data is linked to zones.
- **Major City Data:** Major cities within states are centrally stored, with an option for users to add more.
- **Estimated Delivery:** Admins can add estimated delivery dates for these zones. The system progresses hourly from 10% after 3 hours.
- **Dynamic Fetching:** After entering a license key, the admin chooses international or domestic, and the system automatically fetches data for that country/state, divided into zones and major cities.
- **Example:** If an admin sets North Zone to 5-7 days, and an order is from a specific state, the system fetches details based on the order's location and displays it on the track order page.

4.4. User's End View When Plugin is Installed (IMG_3791.jpeg)

This image shows the front-end user interface for tracking an order, which closely resembles Amazon's tracking page:

- **Search Bar:** A prominent search bar allows users to enter their order ID for real-time tracking.
- **Order Details:** Displays the order number, tracking ID, and courier.
- **Delivery Status:** Shows a progress bar with confidence percentage and expected delivery date/window.
- **Package Location:** Indicates the current package location and the next destination.
- **Package Journey:** A detailed timeline of the package's journey, including status updates (Order Confirmed, Processing, Shipped, In Transit, Out for Delivery, Delivered) with dates and confidence percentages.
- **Help Section:** Provides options for support (e.g., phone number, live chat).
- **Delivery Notes:** Displays additional notes related to delivery.

4.5. Delivery Zone Configuration (IMG_3784.jpeg)

This image illustrates the admin interface for configuring delivery zones:

- **Zone Overview:** Displays cards for each zone (North, South, East, West) with the number of states and average delivery days.
- **Zone Summary:** Provides a detailed overview of each zone, including the states within it and a description (e.g., "Northern states with good connectivity").
- **Bulk Update:** An option for bulk updating zone configurations.

5. Conclusion

The WordPress AI Order Tracker plugin is a sophisticated solution designed to enhance the e-commerce experience by providing advanced order tracking capabilities. Its dual tracking system, automated updates, and highly configurable delivery zones make it a powerful tool for online businesses. The detailed file structure and functional breakdown provided in this document should enable any AI to understand, maintain, and further develop this project efficiently.

Ok, reduce lookup will help consume less cpu.
Also now we have latitude and longitude data.
So now can we do one thing,
First the plugin fetch state city data of seller, ask seller to set up it’s country>state>city, now once this set the plugin fetch the latitude and longitude of sellers city location, then someone order’s it starts showing order received the order packed, order processed,then order dispatched, then order in transit, then ‘seller’s city name’ warehouse, then seller’s near city name hub, then more if long days delivery range selected, then buyer’s state fetched up, then order in ‘buyer’s random city name’ warehouse, now we have latitude and longitude data, now random city name can be picked up from near latitude and longitude city name, which will make another possibility like now this fake track can show map with where’s there order. Map can be loaded from leaflet if possible.
Now suggest me all this and continue