🗺️ Location Analyzer

A powerful tool for analyzing Google Location History data to generate detailed travel reports, city visits, and location jump analysis.

✨ Features

- 📊 **Comprehensive Analysis**: Analyzes Google Location History JSON files
- 🏙️ **City & Country Detection**: Identifies all cities and countries visited
- ⏱️ **Time Tracking**: Calculates time spent in each location
- 🚀 **Location Jumps**: Tracks significant movements between cities
- 📈 **Distance Calculation**: Computes total travel distance
- 📁 **CSV Exports**: Generates detailed reports in CSV format
- 🌐 **Dual Interface**: Both web application and desktop GUI
- ⚡ **High Performance**: Modern async processing for large datasets

📂 Project Structure

```
location_analyzer/
├── analyzer_bridge.py      # Bridge between interfaces and analyzers
├── app.py                  # Flask web application
├── csv_exporter.py         # CSV export utilities
├── geo_utils.py            # Geocoding utilities
├── gui_app.py              # Desktop GUI application
├── legacy_analyzer.py      # Fallback synchronous analyzer
├── location_analyzer.py    # Modern async analyzer (primary)
├── requirements.txt        # Python dependencies
├── config/                 # Configuration files
│   ├── gui_config.json     # GUI settings
│   ├── web_config.json     # Web app settings
│   └── geo_cache.json      # Geocoding cache
├── templates/              # Web interface templates
│   ├── index.html
│   └── results.html
├── uploads/                # Uploaded files storage
├── outputs/                # Analysis results
└── logs/                   # Application logs
```

🚀 Quick Start

Prerequisites

1. **Python 3.8+** installed on your system
2. **Google Location History data** exported from Google Takeout
3. **Geoapify API key** (free at geoapify.com)

Installation

1. **Clone or download** this project
2. **Install dependencies**:

```
bash

pip install -r requirements.txt
```

3. **Get your Geoapify API key** (required for geocoding):
   - Visit geoapify.com

- Sign up for a free account
- Copy your API key

Export Google Location History

1. Go to Google Takeout
2. Select **"Location History (Timeline)"**
3. Choose **JSON format** (not KML)
4. Download and extract the ZIP file
5. Find the **"Records.json"** file in the Location History folder

💻 Usage

Option 1: Web Interface (Recommended)

1. **Start the web application**:

```
bash

python app.py
```

2. **Open your browser** to: `http://localhost:5000`
3. **Upload and analyze**:
   - Select your Google Location History JSON file
   - Set date range for analysis
   - Enter your Geoapify API key
   - Click "Analyze Location History"
4. **Download results**:
   - View analysis summary
   - Download CSV files with detailed reports

Option 2: Desktop GUI

1. **Start the GUI application**:

```
bash
```

```
python gui_app.py
```

2. **Configure settings**:
   - Browse and select your JSON file
   - Set output directory
   - Enter API keys
   - Set date range
3. **Run analysis** and view progress in real-time

📊 Output Files

The analyzer generates several CSV files:

city_jumps.csv

- **Date/Time** of each location change
- **From/To** cities with distance and duration
- **Travel patterns** and movement analysis

by_city_location_days.csv

- **Time spent** in each city (fractional days)
- **Ranked by duration** of stay
- **City, Country** format for clarity

by_state_location_days.csv

- **Time spent** in each state/country
- **Useful for** international travel analysis
- **Consolidated view** of regional visits

analysis_summary.txt

- **Overview statistics** (total distance, cities, jumps)
- **Top 10 cities** by time spent
- **Top 10 states/countries** by time spent

🔧 Configuration

API Keys

- **Geoapify** (Required): Free geocoding service
- **Google Maps** (Optional): Enhanced accuracy for geocoding

Settings Persistence

- **Web app**: Settings saved in `config/web_config.json`
- **GUI app**: Settings saved in `config/gui_config.json`
- **Geocoding cache**: Saved in `config/geo_cache.json` (speeds up repeated analysis)

Performance Tuning

For large files (>100MB):

- Use the **web interface** (better for large datasets)
- Ensure stable internet connection for geocoding
- Analysis may take 10-30 minutes for very large files

🐛 Troubleshooting

Common Issues

"Analyzer not available" error:

```
bash

pip install aiohttp pandas flask werkzeug
```

"Template not found" error:

- Ensure `templates/` folder exists with `index.html` and `results.html`

Import errors:

- Run from the project root directory
- Check that all `.py` files are in the same folder

Geocoding fails:

- Verify your Geoapify API key is correct
- Check internet connection
- Free tier allows 3,000 requests/day

Performance Tips

1. **Use date ranges** to analyze specific periods
2. **Cache is automatic** - repeated analysis of same locations is faster
3. **Close other applications** for large file processing
4. **Stable internet** required for geocoding API calls

🔄 Updating

To update the project:

1. **Backup your config folder** (contains your settings and cache)
2. **Replace all Python files** with new versions
3. **Restore your config folder**
4. **Run** `pip install -r requirements.txt` to update dependencies

📝 Technical Details

Architecture

- **Modern async analyzer** (`location_analyzer.py`): Primary engine using async/await for efficient API calls
- **Legacy analyzer** (`legacy_analyzer.py`): Fallback synchronous processor
- **Bridge module** (`analyzer_bridge.py`): Automatically selects best analyzer and handles compatibility

Data Processing

1. **Parse** Google Location History JSON (multiple formats supported)
2. **Filter** significant location changes (configurable thresholds)
3. **Geocode** coordinates to city/country names (with intelligent caching)
4. **Calculate** distances, time spent, and location jumps
5. **Export** comprehensive reports in CSV format

Geocoding Cache

- **Automatic caching** reduces API calls and speeds up analysis
- **Precision-based** grouping (nearby coordinates share same result)
- **Persistent storage** in `config/geo_cache.json`
- **Cache survives** between sessions and different analyses

🤝 Support

Getting Help

1. **Check this README** for common solutions
2. **Verify file formats** (Google Location History must be JSON)
3. **Test with smaller date ranges** if having performance issues
4. **Check API key validity** at geoapify.com

File Requirements

- **Google Location History** in JSON format (from Google Takeout)
- **Supported formats**: Timeline objects, activity segments, place visits
- **Date range**: Any period within your location history
- **File size**: Tested up to 500MB+ files

📄 License

This project is provided as-is for personal use in analyzing your own Google Location History data.

🎉 **Enjoy exploring your travel history with detailed analytics!**