Price Tracker Setup Guide (Indian Rupees Support)

Overview

This is a complete web-based price tracking system that monitors product prices in Indian Rupees (₹), sends email alerts, and provides scheduling capabilities. Optimized for Indian e-commerce sites.

Features

- ✅ Real-time price tracking from Indian websites (Flipkart, Amazon.in, Myntra, etc.)
- 💰 Full Indian Rupee (₹) support with proper formatting
- 📧 Email alerts when prices drop below target
- 📅 Automated scheduling (hourly, daily, weekly)
- 📊 Price history tracking with CSV storage
- 🎯 Indian e-commerce specific CSS selectors
- 🔧 Easy Gmail configuration
- 📱 Responsive web interface

Supported Indian E-commerce Sites

- **Flipkart**: Automatic price detection
- **Amazon India**: Built-in selectors
- **Myntra**: Fashion price tracking
- **Snapdeal**: Marketplace support
- **Paytm Mall**: Integrated selectors
- **Nykaa**: Beauty products
- **FirstCry**: Kids products
- **BigBasket**: Grocery tracking
- **Any Indian website**: Custom selector support

Installation

1. Clone or Download Files

Save the following files in your project directory:

- `app.py` (Backend Python code)

- `requirements.txt` (Dependencies)
- Create a `templates` folder and save `index.html` inside it

2. Install Dependencies

```
bash

pip install -r requirements.txt
```

3. Gmail Setup (Important!)

For Gmail alerts to work:

1. Enable 2-factor authentication on your Google account
2. Go to Google Account Settings > Security > App passwords
3. Generate an app password for "Mail"
4. Use this app password (not your regular password) in the configuration

4. Run the Application

```
bash

python app.py
```

The application will start on `http://localhost:5000`

Usage

1. Configure Email Settings

- Go to the "Email Config" tab
- Enter your Gmail address and app password
- Set recipient email for alerts
- Save configuration

2. Track a Price

- Go to "Track Price" tab
- Enter product name, URL, and target price in ₹
- Optionally provide a CSS selector for price extraction
- Click "Track Price Now"

Example URLs:

- Flipkart: `https://www.flipkart.com/product-name/p/itm...`
- Amazon India: `https://www.amazon.in/product-name/dp/...`
- Myntra: `https://www.myntra.com/product-name/...`

3. Schedule Automatic Tracking

- Go to "Schedule Tracking" tab
- Fill in product details with ₹ prices
- Choose monitoring interval (hourly/daily/weekly)
- The system will automatically check prices and send alerts

4. View Price History

- Go to "Price History" tab
- View all tracked prices over time
- Filter by specific products
- See when alerts were sent

How It Works

Backend (Python)

- **Web Scraping**: Uses BeautifulSoup and requests to extract prices
- **Email Alerts**: SMTP integration for Gmail notifications
- **Scheduling**: Background thread runs scheduled price checks
- **Data Storage**: CSV file stores all price history
- **API**: Flask REST API for frontend communication

Frontend (HTML/CSS/JS)

- **Modern UI**: Responsive design with animations
- **Real-time Updates**: AJAX calls to backend API
- **Interactive Forms**: Easy product tracking setup
- **Data Visualization**: Price history tables and status indicators

Price Extraction

The system uses multiple strategies to find Indian prices:

1. **Indian e-commerce selectors**: Built-in support for Flipkart, Amazon.in, Myntra, etc.
2. **Currency detection**: Recognizes ₹, Rs., INR formats
3. **Indian number formats**: Supports 1,23,456 (Indian comma system)
4. **Custom CSS selectors**: For any website
5. **Fallback patterns**: Regex-based price detection

Indian Price Format Support

- ₹1,23,456.78 ✅
- Rs. 1,23,456 ✅
- INR 1,23,456 ✅
- 1,23,456 ₹ ✅
- Handles lakhs and crores formatting ✅

File Structure

```
project/
├── app.py              # Main Python backend
├── requirements.txt    # Python dependencies
├── templates/
│   └── index.html      # Frontend interface
├── price_data.csv      # Auto-generated price history
└── config.json         # Auto-generated email config
```

Troubleshooting

Common Issues:

1. **Email not sending**
   - Ensure you're using an app password, not your regular Gmail password
   - Check that 2-factor authentication is enabled
   - Verify SMTP settings (Gmail: smtp.gmail.com:587)
2. **Price not detected**
   - Try providing a custom CSS selector
   - Check if the website blocks automated requests
   - Verify the URL is accessible
3. **Scheduling not working**

- Ensure the Flask app keeps running
- Check console for error messages
- Restart the application if needed

Advanced Price Selectors for Indian Sites

For better price extraction, inspect the website and find the price element:

Common Indian E-commerce Selectors:

- **Flipkart**: `._30jeq3`, `.CEmiEU`, `._1_WHN1`
- **Amazon India**: `.a-price-whole`, `.a-price-fraction`
- **Myntra**: `.pdp-price`, `.discount-price`
- **Snapdeal**: `.payBlkBig`, `.product-price`
- **Paytm Mall**: `._1kMS`, `.price`
- **Nykaa**: `.css-1d0jdb`, `.product-price`

How to find selectors:

- Right-click on price → "Inspect Element"
- Copy the CSS selector
- Test with the system

Security Notes

- Keep your email credentials secure
- Don't share the config.json file
- Use environment variables for production deployment
- Consider using a VPN for scraping if blocked

Deployment

For production deployment:

1. Use a production WSGI server (gunicorn)
2. Set up proper logging
3. Use environment variables for sensitive data
4. Consider using a task queue (Celery) for background jobs
5. Add rate limiting to prevent IP blocking

Legal Considerations

- Respect robots.txt files
- Don't overload servers with requests
- Use reasonable delays between requests
- Comply with website terms of service

Support

This is a educational project. For production use, consider:

- Error handling improvements
- Database instead of CSV
- User authentication
- Rate limiting
- Proxy rotation
- More robust email handling