# Nifty50 Chart Labeling AI - Full Stack App Guide

## Nifty50 Screenshot Labeling AI App

Project Goal:
Upload a screenshot of a Nifty50 bar chart, and automatically label it as "Call" or "Put" based on the price movement after a defined interval (15 minutes).

Tech Stack:
- Frontend: React.js
- Backend: FastAPI (Python)
- Screenshot: PIL.ImageGrab / pyautogui
- Price Source: yFinance

Step 1: FastAPI Backend (main.py)
- Capture Screenshot using ImageGrab
- Fetch Nifty50 price from yfinance at capture time and after delay
- Automatically assign label and explanation
- Store results in screenshots/ and metadata.json

APIs:
POST /capture/ -> Captures current screenshot & stores initial price
GET /label/ -> After 15 mins, compares new price, labels screenshot, stores info
GET /metadata/ -> Shows all labeled screenshots and their metadata

Step 2: React Frontend
- Button to capture screenshot and initiate labeling
- Displays labeled result with explanation and preview
- Shows past results using metadata

Component Breakdown:
1. ChartLabeler.js
  - POST to /capture/
  - GET from /label/
  - Display label result and image

2. ScreenshotList.js
  - Fetches and displays all past labeled screenshots from /metadata/

Example Metadata Entry:

```
{
  "file": "call_20250623_1015.png",
  "label": "call",
  "start_price": 23567.10,
  "end_price": 23645.80,
  "explanation": "Price increased after chart snapshot",
  "timestamp": "20250623_1015"
}
```

Tips:
- Replace ImageGrab with pyautogui.screenshot() on Linux.
- Consider scheduling regular captures (e.g., hourly).
- Add CORS in FastAPI for frontend connection.