```python
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.common.exceptions import TimeoutException, WebDriverException
from bs4 import BeautifulSoup
import concurrent.futures
import json
import time

def get_summary_value(card, data_summary):
    element = card.select_one(f"[data-summary='{data_summary}']")
    if element:
        text = element.get_text(strip=True)
        if not text and element.next_sibling:
            text = element.next_sibling.strip() if isinstance(element.next_sibling, str) else ""
        return text
    return ""

def get_ld_data(card, idx):
    ld_json_data = {}
    ld_script = card.find("script", type="application/ld+json")
    if ld_script and ld_script.string:
        try:
            ld_json_data = json.loads(ld_script.string)
        except Exception as e:
            print(f"Error parsing ld+json for property {idx}: {e}")
    return ld_json_data
```

```python
def get_data(driver):
    soup = BeautifulSoup(driver.page_source, 'html.parser')

    possible_classes = ['mb-srp__card', 'mb-srp_list', 'mb-srpCard', 'srpCard', 'mb-srp']
    cards = []
    for class_name in possible_classes:
        cards = soup.find_all('div', class_=class_name)
        if cards:
            print(f"Found {len(cards)} cards with class '{class_name}'")
            break
    else:
        print("No cards found with known class names.")
        return []

    properties = []
    for idx, card in enumerate(cards, 1):
        try:
            title = ''
            for title_class in ['mb-srp__card--title', 'mb-srp_card--title']:
                title_elem = card.find('h2', class_=title_class)
                if title_elem:
                    title = title_elem.get('title', '').strip() or title_elem.get_text(strip=True)
                    break

            price = ''
            for price_class in ['mb-srp__card__price--amount', 'mb-srp_card_price--amount']:
                price_elem = card.find('div', class_=price_class)
                if price_elem:
                    price = price_elem.get_text(strip=True)
                    break

            area = get_summary_value(card, 'super-area') or get_summary_value(card, 'carpet-area')
            transaction = get_summary_value(card, 'transaction')
            furnishing  = get_summary_value(card, 'furnishing')
            society     = get_summary_value(card, 'society')
            bathroom    = get_summary_value(card, 'bathroom')
            balcony     = get_summary_value(card, 'balcony')
```

```
            data = {
                "title": title,
                "price": price,
                "area": area,
                "transaction": transaction,
                "furnishing": furnishing,
                "society": society,
                "bathroom": bathroom,
                "balcony": balcony
            }

            usp_items = card.select("div.mb-srp_card_usp--item")
            data["usp_details"] = [item.get_text(strip=True) for item in usp_items] if usp_items else []

            ld_json_data = get_ld_data(card, idx)
            if ld_json_data:
                data["numberOfRooms"] = ld_json_data.get("numberOfRooms", "")
                geo_data = ld_json_data.get("geo", {})
                data["latitude"] = geo_data.get("latitude", "")
                data["longitude"] = geo_data.get("longitude", "")

            properties.append(data)
            print(f"Processed property {idx}: {title}")

        except Exception as e:
            print(f"Error on property {idx}: {str(e)}")
            continue
    return properties

def get_house_links(url, retries=3):
    chrome_options = Options()
    #chrome_options.add_argument("--headless")
    chrome_options.add_argument("--disable-gpu")
    chrome_options.add_experimental_option("prefs", {"profile.managed_default_content_settings.images": 2})

    for attempt in range(retries):
        driver = None
```

```
            print(f"\nScraping page {page} for {city}: {url}")
            props = get_house_links(url)
            city_results.extend(props)
        return city_results

    def process_all_cities(cities, start_page=1, end_page=100, max_workers=4):
        all_properties = []
        # Use fewer concurrent threads to reduce resource pressure
        with concurrent.futures.ThreadPoolExecutor(max_workers=max_workers) as executor:
            future_to_city = {
                executor.submit(process_city_pages, city, start_page, end_page): city for city in cities
            }
            for future in concurrent.futures.as_completed(future_to_city):
                city = future_to_city[future]
                try:
                    city_properties = future.result()
                    all_properties.extend(city_properties)
                    print(f"Finished scraping {city}: {len(city_properties)} properties")
                except Exception as exc:
                    print(f"{city} generated an exception: {exc}")
        return all_properties

    if __name__ == "__main__":
        cities = [
            'Gurgaon', 'Noida', 'Ghaziabad', 'Greater-Noida'
        ]
        start_page = 1
        end_page = 4  # Scrape over 90 pages per city
        # Lower max_workers to reduce concurrent load; adjust as needed
        all_properties = process_all_cities(cities, start_page, end_page, max_workers=2)

        print(f"\nTotal properties collected: {len(all_properties)}")

        output_filename = "data.json"
        with open(output_filename, "w", encoding='utf-8') as f:
            json.dump({"properties": all_properties}, f, indent=2, ensure_ascii=False)
        print(f"Data saved successfully to {output_filename}")
```

```
[
    {
      "title": "4 BHK Flat  for Sale in  Adore Presidential World, Sector 84, Faridabad",
      "price": "₹1.25 Cr",
      "area": "Super Area2925 sqft",
      "transaction": "TransactionNew Property",
      "furnishing": "FurnishingUnfurnished",
      "society": "SocietyAdore Presidential World",
      "bathroom": "Bathroom4",
      "balcony": "Balcony4",
      "usp_details": [],
      "numberOfRooms": "",
      "latitude": "",
      "longitude": ""
    },
    {
      "title": "4 BHK Flat  for Sale in  Adore Presidential World, Sector 84, Faridabad",
      "price": "₹1.25 Cr",
      "area": "Super Area2925 sqft",
      "transaction": "TransactionNew Property",
      "furnishing": "FurnishingUnfurnished",
      "society": "SocietyAdore Presidential World",
      "bathroom": "Bathroom4",
      "balcony": "Balcony4",
      "usp_details": [],
      "numberOfRooms": "4",
      "latitude": "28.4047548247827",
      "longitude": "77.3701688183714"
    },
```