Spur Steak Ranches Locations CSV Export Scraper

Install Requirements

```
pip install requests beautifulsoup4 pandas lxml
```

Python Script

Save as:

```
spur_scraper.py
```

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import time

BASE_URL = "https://www.spursteakranches.com"

headers = {
    "User-Agent": "Mozilla/5.0"
}

all_data = []

# Change province here if needed
province = "Gauteng"

for page in range(1, 20):
    url = f"https://www.spursteakranches.com/za/find-a-spur/result?
province={province}&brand_id=1&page={page}"

    print(f"Scraping Page {page}...")

    response = requests.get(url, headers=headers)

    if response.status_code != 200:
        print("Failed page:", page)
        continue
```

```
    soup = BeautifulSoup(response.text, "lxml")

    cards = soup.select(".restaurant-block")

    if not cards:
        print("No more restaurants found.")
        break

    for card in cards:
        try:
            name = ""
            address = ""
            phone = ""
            timings = ""
            branch_url = ""

            title = card.select_one("h2 a")
            if title:
                name = title.get_text(strip=True)
                branch_url = BASE_URL + title.get("href")

            address_el = card.select_one(".address")
            if address_el:
                address = address_el.get_text(" ", strip=True)

            phone_el = card.select_one("a[href^='tel']")
            if phone_el:
                phone = phone_el.get_text(strip=True)

            time_el = card.select_one(".trading-hours")
            if time_el:
                timings = time_el.get_text(" ", strip=True)

            # Visit detail page for more info
            email = ""
            facilities = ""

            if branch_url:
                try:
                    detail_response = requests.get(branch_url, headers=headers)
                    detail_soup = BeautifulSoup(detail_response.text, "lxml")

                    email_el = detail_soup.select_one("a[href^='mailto']")
                    if email_el:
                        email = email_el.get_text(strip=True)

                    facility_items = detail_soup.select(".features li")
                    facilities = ", ".join([
```

```
                        item.get_text(strip=True)
                        for item in facility_items
                    ])

                    time.sleep(1)

                except Exception as e:
                    print("Detail page error:", e)

            all_data.append({
                "Branch Name": name,
                "Address": address,
                "Province": province,
                "Phone": phone,
                "Email": email,
                "Timings": timings,
                "Facilities": facilities,
                "Branch URL": branch_url
            })

            print("Added:", name)

        except Exception as e:
            print("Card Error:", e)

# Export CSV

df = pd.DataFrame(all_data)

csv_name = f"spur_{province.lower()}_locations.csv"

# Remove duplicates

df.drop_duplicates(inplace=True)

# Save

df.to_csv(csv_name, index=False, encoding="utf-8-sig")

print("CSV Saved Successfully")
print("Total Branches:", len(df))
print("File:", csv_name)
```

Run Script

```
python spur_scraper.py
```

Output CSV Columns

The CSV will include:

- Branch Name
- Address
- Province
- Phone
- Email
- Timings
- Facilities
- Branch URL

Export All South Africa Provinces

Replace:

```
province = "Gauteng"
```

With:

```
provinces = [
    "Gauteng",
    "Western Cape",
    "KwaZulu-Natal",
    "Eastern Cape",
    "Free State",
    "Limpopo",
    "Mpumalanga",
    "North West",
    "Northern Cape"
]
```

Then loop through all provinces.

CSV Example

Branch Name	Address	Phone
7 Eagles Spur	Greenstone Mall	011-4522897
Alaska Spur	Atlas Mall	011-3953810
Texakhana Spur	Rosebank	011-8800000

Optional Advanced Features

You can also add:

- Google Maps links
- Latitude / Longitude
- Halaal status
- Kids play area
- WiFi availability
- Booking links
- SEO JSON export
- Next.js JSON API export
- MongoDB insertion
- Automatic image download
- Menu linking