

Amazon Web Scrapping

Name : Srinivasan H

Reg No : 20MIS1078

PostgreSQL Database :

```
CREATE TABLE products (  
    rank INT,  
    title TEXT,  
    price TEXT,  
    asin VARCHAR(10),  
    review_count INT,  
    stars DECIMAL(2, 1),  
    category_node TEXT,  
    PRIMARY KEY (asin)  
);
```

Scrapping Script :

```
const puppeteer = require('puppeteer');  
  
const { Client } = require('pg');  
  
const client = new Client({  
    user: 'your_pg_user',  
    host: 'localhost',  
    database: 'amazon_scraper',  
    password: 'your_pg_password',  
    port: 5432,  
});
```

```

const scrapeAmazonBestSellers = async () => {

  const browser = await puppeteer.launch();

  const page = await browser.newPage();

  // URL for Amazon Best Sellers

  const url = 'https://www.amazon.com/Best-Sellers/zgbs';

  await page.goto(url, { waitUntil: 'networkidle2' });

  // Function to extract product data

  const extractProductData = async () => {

    const products = await page.evaluate(() => {

      const items = Array.from(document.querySelectorAll('.zg-item-immersion'));

      return items.map(item => {

        const rank = parseInt(item.querySelector('.zg-badge-text')?.textContent || '0');

        const title = item.querySelector('.p13n-sc-truncated')?.textContent?.trim() || '';

        const price = item.querySelector('.p13n-sc-price')?.textContent?.trim() || '';

        const asin = item.getAttribute('data-asin') || '';

        const reviewCount = parseInt(item.querySelector('.a-size-small .a-link-normal')?.textContent?.replace(/^[^0-9]/g, '') || '0');

        const stars = parseFloat(item.querySelector('.a-icon-alt')?.textContent?.split(' ')[0] || '0');

        const categoryNode = window.location.pathname;

        return { rank, title, price, asin, reviewCount, stars, categoryNode };

      });

    });

    return products;

  };

  // Function to scroll and load more products

```

```

const loadMoreProducts = async () => {
  let products = [];
  while (true) {
    products = [...products, ...(await extractProductData())];
    const loadMoreButton = await page.$('.a-button-text');
    if (loadMoreButton) {
      await loadMoreButton.click();
      await page.waitForTimeout(2000); // wait for more products to load
    } else {
      break;
    }
  }
  return products;
};

```

// Scrape the first two pages

```

let allProducts = [];
for (let i = 0; i < 2; i++) {
  const products = await loadMoreProducts();
  allProducts = [...allProducts, ...products];
  if (i === 0) {
    await page.click('.a-pagination .a-last a'); // go to next page
    await page.waitForTimeout(2000); // wait for next page to load
  }
}

```

```

await browser.close();
return allProducts;
};

```

```

const saveToDatabase = async (products) => {

```

```
await client.connect();
```

```
for (const product of products) {
```

```
  await client.query(
```

```
    `INSERT INTO products (rank, title, price, asin, review_count, stars, category_node)
```

```
    VALUES ($1, $2, $3, $4, $5, $6, $7)
```

```
    ON CONFLICT (asin) DO NOTHING`,
```

```
    [product.rank, product.title, product.price, product.asin, product.review_count,  
    product.stars, product.category_node]
```

```
  );
```

```
}
```

```
await client.end();
```

```
};
```

```
const main = async () => {
```

```
  try {
```

```
    const products = await scrapeAmazonBestSellers();
```

```
    await saveToDatabase(products);
```

```
    console.log('Data scraped and saved successfully.');
```

```
  } catch (error) {
```

```
    console.error('Error:', error);
```

```
  }
```

```
};
```

```
main();
```