HAMMER SMARTWATCHES WEB SCRAPING

Project for python module

Institute Name- ITvedant Education Pvt. Ltd
Name- Abhishek Rajendra Chavan
Email Address- Abhishekc010@gmail.com

AIM: To retrieve data from the hammer smart watch website in one sheet to analyse the same.

Description: The Hammer Smart Watch Web Scraping Project aims to collect and analyse data from various online sources related to smartwatches. This initiative is driven by the need to aggregate information about smartwatches, their features, user reviews, and pricing from multiple websites. The extracted data will be utilized to enhance market insights, provide users with comprehensive product information, and facilitate informed decision-making when purchasing a smartwatch.

Key Objectives:

1. Data Collection:

Implement web scraping techniques to gather information from leading e-commerce platforms, product review websites, and official manufacturer sites.

Extract relevant details such as product specifications, user ratings, reviews, and pricing.

2. Data Normalization:

Standardize the collected data to ensure consistency across different sources.

Convert and organize data into a structured format for easy analysis.

3. Real-time Updates:

Establish a mechanism for regular updates to keep the information current and reflective of the latest products and reviews in the smartwatch market.

4. Sentiment Analysis:

Employ natural language processing (NLP) tools to conduct sentiment analysis on user reviews.

Classify sentiments into positive, negative, or neutral categories to gauge user satisfaction.

5. Feature Extraction:

Identify and extract key features of smartwatches mentioned in product descriptions and reviews.

Create a comprehensive list of features to aid users in comparing different smartwatch models.

6. Price Tracking:

Monitor and track changes in smartwatch prices over time.

Provide users with historical pricing data and insights into pricing trends.

hammer smart watches web scraping coding command steps:

1. Accessing hammer web site

```
In [3]: import requests
        from bs4 import BeautifulSoup
        import pandas as pd

In [13]: URL='https://hammeronline.in/collections/smart-watch?sort_by=title-ascending'
         URL

Out[13]: 'https://hammeronline.in/collections/smart-watch?sort_by=title-ascending'

In [14]: r = requests.get(URL)
         htmlcontent = r.content
         htmlcontent
```


```
- [count] item left",\n        otherStockLabel: "Low stock - [count] items left",\n        willNotShipUntil: "Ready to ship
[date]",\n        willBeInStockAfter: "Back in stock [date]",\n        waitingForStock: "Inventory on the way",\n        save
Price: "Save [saved_amount]",\n        cartEmpty: "Your cart is currently empty.",\n        cartTermsConfirmation: "You must
agree with the terms and conditions of sales to check out",\n        searchCollections: "Collections",\n        searchPages:
"Pages",\n        searchArticles: "Articles",\n        productFrom: "from ",\n        maxQuantity: "You can only have [quanti
ty] of [title] in your cart."\n      };\n      theme.settings = {\n        cartType: "drawer",\n        isCustomerTemplate: f
alse,\n        moneyFormat: "Rs. {{amount}}",\n        saveType: "dollar",\n        productImageSize: "natural",\n        pro
ductImageCover: false,\n        predictiveSearch: true,\n        predictiveSearchType: null,\n        predictiveSearchVendor:
false,\n        predictiveSearchPrice: false,\n        quickView: false,\n        themeName: \'Impulse\',\n        themeVersi
on: "7.3.3"\n      };\n    </script>\n\n    <script>window.performance && window.performance.mark && window.performance.mark
(\'shopify.content_for_header.start\');</script><meta name="google-site-verification" content="GRwAbg94ghfBPbUZGPvnht83VJyvXK
fyXqZ8mYcnFFY">\n<meta id="shopify-digital-wallet" name="shopify-digital-wallet" content="/33771028616/digital_wallets/dialo
g">\n<link rel="alternate" type="application/atom+xml" title="Feed" href="/collections/smart-watch.atom" />\n<link rel="next"
href="/collections/smart-watch?page=2&amp;sort_by=title-ascending">\n<link rel="alternate" type="application/json+oembed" hre
f="https://hammeronline.in/collections/smart-watch.oembed">\n<script async="async" src="/checkouts/internal/preloads.js?local
e=en-IN"></script><script id="shopify-features" type="application/json">{"accessToken":"5ddd1063d98958f838f6f1a377a5e2cc","be
tas":["rich-media-storefront-analytics"],"domain":"hammeronline.in","predictiveSearch":true,"shopId":33771028616,"smart_payme
nt_buttons_url":"https:\\/\\/hammeronline.in\\/cdn\\/shopifycloud\\/payment-sheet\\/assets\\/latest\\/spb.en.js","dynamic_che
ckout_cart_url":"https:\\/\\/hammeronline.in\\/cdn\\/shopifycloud\\/payment-sheet\\/assets\\/latest\\/dynamic-checkout-cart.e
```


2. Beautifying the html code:

```
In [15]: soup = BeautifulSoup(htmlcontent,"html.parser")
         print(soup)
         soup.prettify
```

```
<!DOCTYPE html>

<html class="no-js" dir="ltr" lang="en">
<head>
<!-- Gokwik Code STARTS HERE -->
<script>

window.merchantInfo  = {
        mid: "3mt5u7y0nwl46pcgso",
        environment: "production",
        type: "merchantInfo",
                storeId: "33771028616",
                fbpixel: "412453419352091",
      }
var productFormSelector = '';
var cart = {"note":null,"attributes":{},"original_total_price":0,"total_price":0,"total_discount":0,"total_weight":0.0,"item_
count":0,"items":[],"requires_shipping":false,"currency":"INR","items_subtotal_price":0,"cart_level_discount_applications":
[]}
var templateName = 'collection'
```

3. Accessing the names of watches

In [16]:
```
names=soup.find_all(class_="grid-product__title grid-product__title--body")
print(names)
```

```
[<div class="grid-product__title grid-product__title--body">ACE 2</div>, <div class="grid-product__title grid-product__title--body">ACE 3</div>, <div class="grid-product__title grid-product__title--body">ACE 4</div>, <div class="grid-product__title grid-product__title--body">ACE PLUS</div>, <div class="grid-product__title grid-product__title--body">ACE Ultra</div>, <div class="grid-product__title grid-product__title--body">Active 2</div>, <div class="grid-product__title grid-product__title--body">Active 3.0</div>, <div class="grid-product__title grid-product__title--body">Active</div>, <div class="grid-product__title grid-product__title--body">Arctic</div>, <div class="grid-product__title grid-product__title--body">Conquer</div>, <div class="grid-product__title grid-product__title--body">Cyclone</div>, <div class="grid-product__title grid-product__title--body">Fit Pro</div>, <div class="grid-product__title grid-product__title--body">Fit +</div>, <div class="grid-product__title grid-product__title--body">Glide</div>, <div class="grid-product__title grid-product__title--body">Luxor</div>, <div class="grid-product__title grid-product__title--body">Polar</div>, <div class="grid-product__title grid-product__title--body">Pulse 2.0</div>, <div class="grid-product__title grid-product__title--body">Pulse 3.0</div>, <div class="grid-product__title grid-product__title--body">Pulse 4.0</div>, <div class="grid-product__title grid-product__title--body">Pulse Ace </div>, <div class="grid-product__title grid-product__title--body">Pulse Ace Pro</div>, <div class="grid-product__title grid-product__title--body">Pulse X</div>, <div class="grid-product__title grid-product__title--body">Robust</div>, <div class="grid-product__title grid-product__title--body">Stroke</div>]
```

click to expand output; double click to hide output

In [17]:
```
Smartwatches_name=[]
for i in range(0,len(names)):
    Smartwatches_name.append(names[i].get_text())
print(Smartwatches_name)
```

```
['ACE 2', 'ACE 3', 'ACE 4', 'ACE PLUS', 'ACE Ultra', 'Active 2', 'Active 3.0', 'Active', 'Arctic', 'Conquer', 'Cyclone', 'Fit Pro', 'Fit +', 'Glide', 'Luxor', 'Polar', 'Pulse 2.0', 'Pulse 3.0', 'Pulse 4.0', 'Pulse Ace ', 'Pulse Ace Pro', 'Pulse X', 'Robust', 'Stroke']
```

4. Accessing Original prices of watches

In [22]:
```
original_price=soup.find_all(class_="grid-product__price--original")
print(original_price)
```

```
[<span class="grid-product__price--original">Rs. 3,999.00</span>, <span class="grid-product__price--original">Rs. 4,999.00</span>, <span class="grid-product__price--original">Rs. 4,999.00</span>, <span class="grid-product__price--original">Rs. 4,999.00</span>, <span class="grid-product__price--original">Rs. 4,999.00</span>, <span class="grid-product__price--original">Rs. 5,999.00</span>, <span class="grid-product__price--original">Rs. 5,999.00</span>, <span class="grid-product__price--original">Rs. 9,999.00</span>, <span class="grid-product__price--original">Rs. 8,999.00</span>, <span class="grid-product__price--original">Rs. 8,999.00</span>, <span class="grid-product__price--original">Rs. 4,999.00</span>, <span class="grid-product__price--original">Rs. 8,999.00</span>, <span class="grid-product__price--original">Rs. 6,999.00</span>, <span class="grid-product__price--original">Rs. 8,999.00</span>, <span class="grid-product__price--original">Rs. 8,999.00</span>, <span class="grid-product__price--original">Rs. 5,999.00</span>, <span class="grid-product__price--original">Rs. 13,330.00</span>, <span class="grid-product__price--original">Rs. 9,999.00</span>, <span class="grid-product__price--original">Rs. 8,999.00</span>, <span class="grid-product__price--original">Rs. 3,999.00</span>, <span class="grid-product__price--original">Rs. 4,999.00</span>, <span class="grid-product__price--original">Rs. 4,999.00</span>, <span class="grid-product__price--original">Rs. 8,999.00</span>, <span class="grid-product__price--original">Rs. 6,999.00</span>]
```

In [23]:
```
Smartwatches_price=[]
for i in range(0,len(original_price)):
    Smartwatches_price.append(original_price[i].get_text())
print(Smartwatches_price)
```

```
['Rs. 3,999.00', 'Rs. 4,999.00', 'Rs. 4,999.00', 'Rs. 4,999.00', 'Rs. 4,999.00', 'Rs. 5,999.00', 'Rs. 5,999.00', 'Rs. 9,999.00', 'Rs. 8,999.00', 'Rs. 8,999.00', 'Rs. 4,999.00', 'Rs. 8,999.00', 'Rs. 6,999.00', 'Rs. 8,999.00', 'Rs. 8,999.00', 'Rs. 5,999.00', 'Rs. 13,330.00', 'Rs. 9,999.00', 'Rs. 8,999.00', 'Rs. 3,999.00', 'Rs. 4,999.00', 'Rs. 4,999.00', 'Rs. 8,999.00', 'Rs. 6,999.00']
```

5. Accessing the discounted prices of watches

```
In [40]: Discounted_price=soup.find_all(class_="grid-product__price")
         Discounted_price
```

```
<span class="grid-product__price--original">Rs. 8,999.00</span>
<span class="visually-hidden">Sale price</span></div>,
<div class="grid-product__price">

          Rs. 1,299.00
<span class="visually-hidden">Regular price</span>
<span class="grid-product__price--original">Rs. 4,999.00</span>
<span class="visually-hidden">Sale price</span></div>,
<div class="grid-product__price">

          Rs. 3,599.00
<span class="visually-hidden">Regular price</span>
<span class="grid-product__price--original">Rs. 8,999.00</span>
<span class="visually-hidden">Sale price</span></div>,
<div class="grid-product__price">

          Rs. 2,299.00
<span class="visually-hidden">Regular price</span>
<span class="grid-product__price--original">Rs. 6,999.00</span>
<span class="visually-hidden">Sale price</span></div>,
```

```
In [44]: Discounted_Price=[]
         discounted_price=[]
         for i in range(0,len(Discounted_price)):
             Discounted_Price.append(Discounted_price[i].get_text())
         #print(Discounted_Price)
         for i in range(0,len(Discounted_price)):
             discounted_price.append(Discounted_Price[i].strip())
         #print(discounted_price)
         Discounted_Price=[i[:+12] for i in discounted_price]
         print(Discounted_Price)
```

```
['Rs. 1,299.00', 'Rs. 1,299.00', 'Rs. 1,299.00', 'Rs. 1,299.00', 'Rs. 2,499.00', 'Rs. 2,399.00', 'Rs. 2,499.00', 'Rs. 3,599.0
0', 'Rs. 3,299.00', 'Rs. 2,999.00', 'Rs. 1,299.00', 'Rs. 3,599.00', 'Rs. 2,299.00', 'Rs. 2,699.00', 'Rs. 2,399.00', 'Rs. 1,799.
00', 'Rs. 2,399.00', 'Rs. 2,099.00', 'Rs. 3,399.00', 'Rs. 1,299.00', 'Rs. 1,999.00', 'Rs. 1,299.00', 'Rs. 2,699.00', 'Rs. 1,19
9.00']
```

6. Accessing the number of reviews on the watches

```
In [39]: reviews=soup.find_all(class_="jdgm-prev-badge")
         print(reviews)
```

```
[<div class="jdgm-prev-badge" data-average-rating="4.84" data-number-of-questions="0" data-number-of-reviews="25" style="displa
y:none"> <span aria-label="4.84 stars" class="jdgm-prev-badge__stars" data-score="4.84" role="button" tabindex="0"> <span class
="jdgm-star jdgm--on"></span><span class="jdgm-star jdgm--on"></span><span class="jdgm-star jdgm--on"></span><span class="jdgm-
star jdgm--on"></span><span class="jdgm-star jdgm--on"></span> </span> <span class="jdgm-prev-badge__text"> 25 reviews </span>
</div>, <div class="jdgm-prev-badge" data-average-rating="4.86" data-number-of-questions="0" data-number-of-reviews="43" style
="display:none"> <span aria-label="4.86 stars" class="jdgm-prev-badge__stars" data-score="4.86" role="button" tabindex="0"> <sp
an class="jdgm-star jdgm--on"></span><span class="jdgm-star jdgm--on"></span><span class="jdgm-star jdgm--on"></span><span clas
s="jdgm-star jdgm--on"></span><span class="jdgm-star jdgm--on"></span> </span> <span class="jdgm-prev-badge__text"> 43 reviews
</span> </div>, <div class="jdgm-prev-badge" data-average-rating="4.76" data-number-of-questions="0" data-number-of-reviews="1
7" style="display:none"> <span aria-label="4.76 stars" class="jdgm-prev-badge__stars" data-score="4.76" role="button" tabindex
="0"> <span class="jdgm-star jdgm--on"></span><span class="jdgm-star jdgm--on"></span><span class="jdgm-star jdgm--on"></span><
span class="jdgm-star jdgm--on"></span><span class="jdgm-star jdgm--on"></span> </span> <span class="jdgm-prev-badge__text"> 17
reviews </span> </div>, <div class="jdgm-prev-badge" data-average-rating="4.75" data-number-of-questions="0" data-number-of-rev
iews="16" style="display:none"> <span aria-label="4.75 stars" class="jdgm-prev-badge__stars" data-score="4.75" role="button" ta
bindex="0"> <span class="jdgm-star jdgm--on"></span><span class="jdgm-star jdgm--on"></span><span class="jdgm-star jdgm--on"></
span><span class="jdgm-star jdgm--on"></span><span class="jdgm-star jdgm--on"></span> </span> <span class="jdgm-prev-badge__tex
t"> 16 reviews </span> </div>, <div class="jdgm-prev-badge" data-average-rating="4.72" data-number-of-questions="0" data-number
-of-reviews="43" style="display:none"> <span aria-label="4.72 stars" class="jdgm-prev-badge__stars" data-score="4.72" role="but
ton" tabindex="0"> <span class="jdgm-star jdgm--on"></span><span class="jdgm-star jdgm--on"></span><span class="jdgm-star jdgm-
-on"></span><span class="jdgm-star jdgm--on"></span><span class="jdgm-star jdgm--half"></span> </span> <span class="jdgm-prev-b
adge__text"> 43 reviews </span> </div>, <div class="jdgm-prev-badge" data-average-rating="4.95" data-number-of-questions="0" da
ta-number-of-reviews="20" style="display:none"> <span aria-label="4.95 stars" class="jdgm-prev-badge__stars" data-score="4.95"
role="button" tabindex="0"> <span class="jdgm-star jdgm--on"></span><span class="jdgm-star jdgm--on"></span><span class="jdgm-s
tar jdgm--on"></span><span class="jdgm-star jdgm--on"></span><span class="jdgm-star jdgm--on"></span> </span> <span class="jdgm
-prev-badge__text"> 20 reviews </span> </div>, <div class="jdgm-prev-badge" data-average-rating="4.75" data-number-of-questions
="0" data-number-of-reviews="24" style="display:none"> <span aria-label="4.75 stars" class="jdgm-prev-badge__stars" data-score
```

```
In [41]: reviews_given=[]
         for i in range(0,len(reviews)):
             reviews_given.append(reviews[i].get_text())
         print(reviews_given)
```

```
['     25 reviews  ', '     43 reviews  ', '     17 reviews  ', '     16 reviews  ', '     43 reviews  ', '     20 reviews  ',
'     24 reviews  ', '     7 reviews  ', '     2 reviews  ', '     22 reviews  ', '     6 reviews  ', '     7 reviews  ', '
3 reviews  ', '     1 review  ', '     5 reviews  ', '     5 reviews  ', '     110 reviews  ', '     104 reviews  ', '     9 re
views  ', '     68 reviews  ', '     58 reviews  ', '     11 reviews  ', '     5 reviews  ', '     14 reviews  ']
```

7. Collecting the details of watches

In [43]:
```
detail=soup.find_all(class_="collection_tags")
print(detail)
```

```
[<div class="collection_tags">
        BT Calling | 1.83" TFT Display
      </div>, <div class="collection_tags">
        BT Calling | 1.85" Display
      </div>, <div class="collection_tags">
        BT Calling | 1.85" Display
      </div>, <div class="collection_tags">
        BT Calling | Metallic Alloy Body
      </div>, <div class="collection_tags">
        BT Calling | 1.96" Display | AOD
      </div>, <div class="collection_tags">
        BT Calling | 1.95" Screen
      </div>, <div class="collection_tags">
        BT Calling | 1.39" Display
      </div>, <div class="collection_tags">
        BT Calling | 1.32" TFT Display
      </div>, <div class="collection_tags">
        2.04" Super Amoled | BT Calling
      </div>, <div class="collection_tags">
        2.02" Amoled | BT Calling
      </div>, <div class="collection_tags">
        1.39" Screen | BT Calling
      </div>, <div class="collection_tags">
        1.43" Super Amoled | Calling
      </div>, <div class="collection_tags">
        BT Calling | 1.85" TFT Display
```

In [61]:
```
products_detail=[]
product_detail=[]
for i in range(0,len(detail)):
    products_detail.append(detail[i].get_text())
#print(product_detail)
for i in range(0,len(detail)):
    product_detail.append(products_detail[i].strip())
print(product_detail)
```

```
['BT Calling | 1.83" TFT Display', 'BT Calling | 1.85" Display', 'BT Calling | 1.85" Display', 'BT Calling | Metallic Alloy Bod
y', 'BT Calling | 1.96" Display | AOD', 'BT Calling | 1.95" Screen', 'BT Calling | 1.39" Display', 'BT Calling | 1.32" TFT Disp
lay', '2.04" Super Amoled | BT Calling', '2.02" Amoled | BT Calling', '1.39" Screen | BT Calling', '1.43" Super Amoled | Callin
g', 'BT Calling | 1.85" TFT Display', '1.43" Amoled Display | AOD', '1.45" Amoled | BT Calling', 'BT Calling | 2.01" AOD', 'BT
Calling | 1.69" HD Display', 'BT Calling | 1.69" Display', 'BT Calling | 1.28" Display', 'BT Calling | 1.69" TFT Display', 'BT
Calling | 1.81" TFT Display', 'BT Calling | 1.83" Display', '1.96" Amoled display| AOD', 'BT Calling | 1.96" TFT Display']
```

8. Printing the data in form of table using Pandas library

In [65]:
```
hammer_Smartwatches=pd.DataFrame({"Smartwatches_name":Smartwatches_name,"product_detail":product_detail,"reviews_count":reviews_c
print(hammer_Smartwatches)
```

	Smartwatches_name	product_detail	reviews_count
1	ACE 2	BT Calling | 1.83" TFT Display	25 reviews
2	ACE 3	BT Calling | 1.85" Display	43 reviews
3	ACE 4	BT Calling | 1.85" Display	17 reviews
4	ACE PLUS	BT Calling | Metallic Alloy Body	16 reviews
5	ACE Ultra	BT Calling | 1.96" Display | AOD	43 reviews
6	Active 2	BT Calling | 1.95" Screen	20 reviews
7	Active 3.0	BT Calling | 1.39" Display	24 reviews
8	Active	BT Calling | 1.32" TFT Display	7 reviews
9	Arctic	2.04" Super Amoled | BT Calling	2 reviews
10	Conquer	2.02" Amoled | BT Calling	22 reviews
11	Cyclone	1.39" Screen | BT Calling	6 reviews
12	Fit Pro	1.43" Super Amoled | Calling	7 reviews
13	Fit +	BT Calling | 1.85" TFT Display	3 reviews
14	Glide	1.43" Amoled Display | AOD	1 review
15	Luxor	1.45" Amoled | BT Calling	5 reviews
16	Polar	BT Calling | 2.01" AOD	5 reviews
17	Pulse 2.0	BT Calling | 1.69" HD Display	110 reviews
18	Pulse 3.0	BT Calling | 1.69" Display	104 reviews

9. **Then storing all data in CSV file and opening the file in application.**

```
In [47]: import os
         print(os.getcwd())
         hammer_Smartwatches.to_csv("hammer smart watch.csv",index=False)
         a=pd.read_csv("hammer smart watch.csv")
         a
```

Out[47]:

	Smartwatches_name	product_detail	reviews_count	original_price	discounted_price
0	ACE 2	BT Calling | 1.83" TFT Display	25 reviews	Rs. 3,999.00	Rs. 1,299.00
1	ACE 3	BT Calling | 1.85" Display	43 reviews	Rs. 4,999.00	Rs. 1,299.00
2	ACE 4	BT Calling | 1.85" Display	17 reviews	Rs. 4,999.00	Rs. 1,299.00
3	ACE PLUS	BT Calling | Metallic Alloy Body	16 reviews	Rs. 4,999.00	Rs. 1,299.00
4	ACE Ultra	BT Calling | 1.96" Display | AOD	43 reviews	Rs. 4,999.00	Rs. 2,499.00
5	Active 2	BT Calling | 1.95" Screen	21 reviews	Rs. 5,999.00	Rs. 2,399.00
6	Active 3.0	BT Calling | 1.39" Display	24 reviews	Rs. 5,999.00	Rs. 2,499.00
7	Active	BT Calling | 1.32" TFT Display	7 reviews	Rs. 9,999.00	Rs. 3,599.00
8	Arctic	2.04" Super Amoled | BT Calling	2 reviews	Rs. 8,999.00	Rs. 3,299.00
9	Conquer	2.02" Amoled | BT Calling	22 reviews	Rs. 8,999.00	Rs. 2,999.00
10	Cyclone	1.39" Screen | BT Calling	6 reviews	Rs. 4,999.00	Rs. 1,299.00
11	Fit Pro	1.43" Super Amoled | Calling	7 reviews	Rs. 8,999.00	Rs. 3,599.00
12	Fit +	BT Calling | 1.85" TFT Display	3 reviews	Rs. 6,999.00	Rs. 2,299.00
13	Glide	1.43" Amoled Display | AOD	1 review	Rs. 8,999.00	Rs. 2,699.00
14	Luxor	1.45" Amoled | BT Calling	5 reviews	Rs. 8,999.00	Rs. 2,399.00
15	Polar	BT Calling | 2.01" AOD	4 reviews	Rs. 5,999.00	Rs. 1,799.00
16	Pulse 2.0	BT Calling | 1.69" HD Display	110 reviews	Rs. 13,330.00	Rs. 2,399.00
17	Pulse 3.0	BT Calling | 1.69" Display	104 reviews	Rs. 9,999.00	Rs. 2,099.00
18	Pulse 4.0	BT Calling | 1.28" Display	9 reviews	Rs. 8,999.00	Rs. 3,399.00
19	Pulse Ace	BT Calling | 1.69" TFT Display	68 reviews	Rs. 3,999.00	Rs. 1,299.00
20	Pulse Ace Pro	BT Calling | 1.81" TFT Display	58 reviews	Rs. 4,999.00	Rs. 1,999.00
21	Pulse X	BT Calling | 1.83" Display	11 reviews	Rs. 4,999.00	Rs. 1,299.00
22	Robust	1.96" Amoled display| AOD	5 reviews	Rs. 8,999.00	Rs. 2,699.00
23	Stroke	BT Calling | 1.96" TFT Display	15 reviews	Rs. 6,999.00	Rs. 1,199.00

Conclusion

In conclusion, the Hammer Smart Watch Web Scraping Project successfully harnesses web scraping technologies to aggregate, normalize, and present valuable information about smartwatches. With a user-friendly interface, real-time updates, and a commitment to privacy and ethics, the project empowers consumers to make informed decisions while providing insights into market trends. Its scalability and documentation ensure adaptability and transparency, making it a robust tool for navigating the dynamic landscape of smartwatch offerings.