

PRACTICAL

8

Name : Swarnika Singh

Class:D15B

Roll no:63

Aim: To code and register a service worker, and complete the install and activation process for a new service worker for the E-commerce PWA.

Theory:

Service Worker

Service Worker is a script that works on browser background without user interaction independently. Also, It resembles a proxy that works on the user side. With this script, you can track network traffic of the page, manage push notifications and develop "online first" web applications with Cache API.

Things to note about Service Worker:

- A service worker is a programmable network proxy that lets you control how network requests from your page are handled.
- Service workers only run over HTTPS. Because service workers can intercept network requests and modify responses, "man-in-the-middle" attacks could be very bad.
- The service worker becomes idle when not in use and restarts when it's next needed. You cannot rely on a global state persisting between events. If there is information that you need to persist and reuse across restarts, you can use IndexedDB databases.

What can we do with Service Workers?

- You can dominate Network Traffic
You can manage all network traffic of the page and do any manipulations. For example, when the page requests a CSS file, you can send plain text as a response or when the page requests an HTML file, you can send a png file as a response. You can also send a true response too.
- You can Cache
You can cache any request/response pair with Service Worker and Cache API and you can access these online content anytime.

- You can manage Push Notifications

You can manage push notifications with Service Worker and show any information message to the user.

- You can Continue

Although Internet connection is broken, you can start any process with Background Sync of Service Worker.

What can't we do with Service Workers?

- You can't access the Window

You can't access the window, therefore, You can't manipulate DOM elements. But, you can communicate to the window through post Message and manage processes that you want.

- You can't work it on 80 Port

Service Worker just can work on HTTPS protocol. But you can work on localhost during development.

Service Worker Cycle

A service worker goes through three steps in its life cycle:

- Registration
- Installation
- Activation

Registration

To install a service worker, you need to register it in your main JavaScript code.

Registration tells the browser where your service worker is located, and to start installing it in the background. Let's look at an example: main.js

```
if('serviceWorker' in navigator){navigator.serviceWorker.register('/service-worker.js')
.then(function(registration){
console.log('Registrationsuccessful,scopeis:',registration.scope);
})
.catch(function(error){
console.log('Serviceworkerregistrationfailed,error:',erro
r); });
}
```

This code starts by checking for browser support by examining `navigator.serviceWorker`. The service worker is then registered with `navigator.serviceWorker.register`, which returns a promise that resolves when the service worker has been successfully registered. The scope of the service worker is then logged with `registration.scope`. If the service worker is already installed, `navigator.serviceWorker.register` returns the registration object of the currently active service worker.

The scope of the service worker determines which files the service worker controls, in other words, from which path the service worker will intercept requests. The default scope is the location of the service worker file, and extends to all directories below. So if `service-worker.js` is located in the root directory, the service worker will control requests from all files at this domain.

You can also set an arbitrary scope by passing in an additional parameter when registering. For example: `main.js`

```
navigator.serviceWorker.register('/service-worker.js',{scope:'/app/'
});
```

In this case we are setting the scope of the service worker to `/app/`, which means the service worker will control requests from pages like `/app/`, `/app/lower/` and `/app/lower/lower`, but not from pages like `/app` or `/`, which are higher.

If you want the service worker to control higher pages e.g. `/app` (without the trailing slash) you can indeed change the scope option, but you'll also need to set the `Service-Worker-Allowed` HTTP Header in your server config for the request serving the service worker script.

`main.js`

```
navigator.serviceWorker.register('/app/service-worker.js',{scope:'/ap
p' });
```

Installation

Once the browser registers a service worker, installation can be attempted. This occurs if the service worker is considered to be new by the browser, either because the site currently doesn't have a registered service worker, or because there is a byte difference between the new service worker and the previously installed one.

A service worker installation triggers an `install` event in the installing service worker. We can include an `install` event listener in the service worker to perform some task when the service worker installs. For instance, during the `install`, service workers can

precache parts of a web app so that it loads instantly the next time a user opens it (see caching the application shell). So, after that first load, you're going to benefit from instant repeat loads and your time to interactivity is going to be even better in those cases. An example of an installation event listener looks like this:

service-worker.js

```
//Listenforinstallevent,setcallback
self.addEventListener('install',function(event){
//Performsometask
});
```

Activation

Once a service worker has successfully installed, it transitions into the activation stage. If there are any open pages controlled by the previous service worker, the new service worker enters a waiting state. The new service worker only activates when there are no longer any pages loaded that are still using the old service worker. This ensures that only one version of the service worker is running at any given time.

When the new service worker activates, an activate event is triggered in the activating service worker. This event listener is a good place to clean up outdated caches (see the [Offline Cookbook](#) for an example).

service-worker.js

```
self.addEventListener('activate',function(event){
//Performsometask
});
```

Once activated, the service worker controls all pages that load within its scope, and starts listening for events from those pages. However, pages in your app that were loaded before the service worker activation will not be under service worker control. The new service worker will only take over when you close and reopen your app, or if the service worker calls `clients.claim()`. Until then, requests from this page will not be intercepted by the new service worker. This is intentional as a way to ensure consistency in your site.

CODE:

1) Index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Panda Ecommerce</title>
  <link rel="stylesheet" href="main.css">
  <!-- Reference to the manifest file -->
  <link rel="manifest" href="manifest.json">
  <!-- bootstrap stylesheet -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet"

integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWFspD3yD65Voh
hpuuCOmLASjC" crossorigin="anonymous">
  <script>
    if ('serviceWorker' in navigator) {
      window.addEventListener('load', function() {
        navigator.serviceWorker.register('/service-worker.js')
        .then(function(registration) {
          console.log('ServiceWorker registration successful with scope: ',
registration.scope);
        }, function(err) {
          console.log('ServiceWorker registration failed: ', err);
        });
      });
    }
  </script>

</head>

<body>
  <!-- header start -->
  <header>
```

```

<!-- navigation -->
<nav class="navbar fixed-top navbar-expand-lg navbar-light bg-light">
  <div class="container">
    <a class="navbar-brand" href="#"></a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarScroll"
    aria-controls="navbarScroll" aria-expanded="false" aria-label="Toggle
navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarScroll">
      <ul class="navbar-nav me-auto my-2 my-lg-0 navbar-nav-scroll"
style="--bs-scroll-height: 100px;">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="#">Home</a>
        </li>
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown"
role="button"
          data-bs-toggle="dropdown" aria-expanded="false">
            Product
          </a>
          <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
            <li><a class="dropdown-item" href="#">Tv</a></li>
            <li><a class="dropdown-item" href="#">Watch</a></li>
            <li><a class="dropdown-item" href="#shoes">Shoes</a></li>
            <li><a class="dropdown-item" href="#backpack">Backpack</a></li>
          </ul>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">About Us</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#subscribe">Subscribe Us</a>
        </li>
      </ul>
      <form class="d-flex">
        <input class="form-control me-2" type="search" placeholder="Search"
aria-label="Search">
        <button class="btn btn-outline-success" type="submit">Search</button>

```

```

        </form>
    </div>
</div>
</nav>
<!-- image slider -->
<section class="container ">
    <div id="carouselExampleControls" class="carousel slide"
data-bs-ride="carousel">
        <div class="carousel-inner">
            <div class="carousel-item active">
                <div class="row mt-5 sliding-image p-5 d-flex align-items-center">
                    <div class="col-lg-7">
                        <h1>Cool Dude Headphone</h1>
                        <p>This is the best headphone in the world for people who just
want to waste time in
                            front
                            of funky world.</p>
                        <h1>$420</h1>
                        <button class="panda-btn-buy-now">Buy Now</button>
                    </div>
                    <div class="col-lg-5">
                        
                    </div>
                </div>
            </div>
            <div class="carousel-item">
                <div class="row mt-5 sliding-image p-5 d-flex align-items-center">
                    <div class="col-lg-7">
                        <h1>Mega LCD TV For Sports</h1>
                        <p>This is the best headphone in the world for people who just
want to waste time in
                            front
                            of TV.</p>
                        <h1>$1200</h1>
                        <button class="panda-btn-buy-now">Buy Now</button>
                    </div>
                    <div class="col-lg-5">
                        

```

```

        </div>
    </div>
</div>
<div class="carousel-item">
    <div class="row mt-5 sliding-image p-5 d-flex align-items-center">
        <div class="col-lg-7">
            <h1>X-Box for your living room</h1>
            <p>This is the best X-Box in the world for people who just want to
waste time in
            front
            of fake sports.</p>
            <h1>$600</h1>
            <button class="panda-btn-buy-now">Buy Now</button>
        </div>
        <div class="col-lg-5">
            
        </div>
    </div>
</div>
</div>
</div>
</div>
<button class="carousel-control-prev" type="button"
data-bs-target="#carouselExampleControls"
data-bs-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Previous</span>
</button>
<button class="carousel-control-next" type="button"
data-bs-target="#carouselExampleControls"
data-bs-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Next</span>
</button>
</div>
</section>
</header>

<!-- main start -->
<main>

```



```

<!-- catagories -->
<section class="container">
  <div class="overflow-hidden my-5 categories">
    <div class="row gx-5">
      <h1>Categories</h1>
      <div class="col-lg-4 col-md-6">
        <div class="p-3 bg-warning d-flex align-items-center justify-content-around rounded-3">
          <h1>Watch</h1>
          
        </div>
      </div>
      <div class="col-lg-4 col-md-6">
        <div class="p-3 bg-danger d-flex align-items-center justify-content-around rounded-3">
          <h1>Bag</h1>
          
        </div>
      </div>
      <div class="col-lg-4 col-md-6">
        <div class="p-3 bg-primary d-flex align-items-center justify-content-around rounded-3">
          <h1>Shoes</h1>
          
        </div>
      </div>
    </div>
  </div>
</section>
<section class="container" id="shoes">
  <h2>Shoes</h2>
  <div class="row row-cols-1 row-cols-md-2 row-cols-lg-3 g-4">
    <div class="col">
      <div class="card border-0 shadow-lg h-100">
        
        <div class="card-body">
          <h5 class="card-title">Reebok Dart Men's Shoes</h5>
          <p class="card-text">Amet minim mollit non deserunt ullamco est sit aliqua dolor do amet sint. </p>

```

```

        <h2>$1289</h2>
    </div>
    <div>
        <button class="panda-btn-buy-now">Buy Now</button>
    </div>
</div>
</div>
<div class="col">
    <div class="card border-0 shadow-lg h-100">
        
        <div class="card-body">
            <h5 class="card-title">Reebok Pump</h5>
            <p class="card-text">Amet minim mollit non deserunt ullamco est sit
aliqua dolor do amet
                sint. </p>
            <h2>$799</h2>
        </div>
        <div>
            <button class="panda-btn-buy-now">Buy Now</button>
        </div>
    </div>
</div>
<div class="col">
    <div class="card border-0 shadow-lg h-100">
        
        <div class="card-body">
            <h5 class="card-title">Reebok All Terrain</h5>
            <p class="card-text">Amet minim mollit non deserunt ullamco est sit
aliqua dolor do amet
                sint. </p>
            <h2>$699</h2>
        </div>
        <div>
            <button class="panda-btn-buy-now">Buy Now</button>
        </div>
    </div>
</div>
</div>
</section>
<section class="container my-5" id="backpack">

```

```

<h2>Backpack</h2>
<div class="row row-cols-1 row-cols-md-2 row-cols-lg-3 g-4">
  <div class="col">
    <div class="card border-0 shadow-lg h-100">
      
      <div class="card-body">
        <h5 class="card-title">Ison Backpack</h5>
        <p class="card-text">Amet minim mollit non deserunt ullamco est sit
aliqua dolor do amet
          sint.</p>
          <h2>$1289</h2>
        </div>
        <div>
          <button class="panda-btn-buy-now">Buy Now</button>
        </div>
      </div>
    </div>
    <div class="col">
      <div class="card border-0 shadow-lg h-100">
        
        <div class="card-body">
          <h5 class="card-title">Biaowang Backpack</h5>
          <p class="card-text">Amet minim mollit non deserunt ullamco est sit
aliqua dolor do amet
            sint. </p>
            <h2>$799</h2>
          </div>
          <div>
            <button class="panda-btn-buy-now">Buy Now</button>
          </div>
        </div>
      </div>
    <div class="col">
      <div class="card border-0 shadow-lg h-100">
        
        <div class="card-body">
          <h5 class="card-title">Dxyizu WS54 Smart</h5>
          <p class="card-text">Amet minim mollit non deserunt ullamco est sit
aliqua dolor do amet
            sint. </p>

```

```

        <h2>$699</h2>
    </div>
    <div>
        <button class="panda-btn-buy-now">Buy Now</button>
    </div>
</div>
</div>
</div>
</section>

<!-- subscribe -->
<section style="background-color: #FEEAE9; border-radius: 10px; height: 300px;"
    class="container d-flex justify-content-center my-5 align-items-center"
id="subscribe">
    <div>
        <h1>LET'S STAY IN TOUCH</h1>
        <p>Get updates on sales specials and more</p>
        <input class="form-control" type="email" name="" id="" placeholder="Your
Email">
        <br>
        <button class="panda-btn-buy-now">SUBMIT</button>
    </div>
</section>
</main>

<footer class="container footer text-center">
    <p><small>Panda Ecommerce. All rights reserved. Dhaka Bangladesh</small></p>
</footer>

<!-- bootstrap script -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"

integrity="sha384-MrcW6ZMFYlzcLA8NI+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+Jc
Xn/tWtIaxVXM"
    crossorigin="anonymous"></script>
</body>

</html>

```

2) service-worker.js

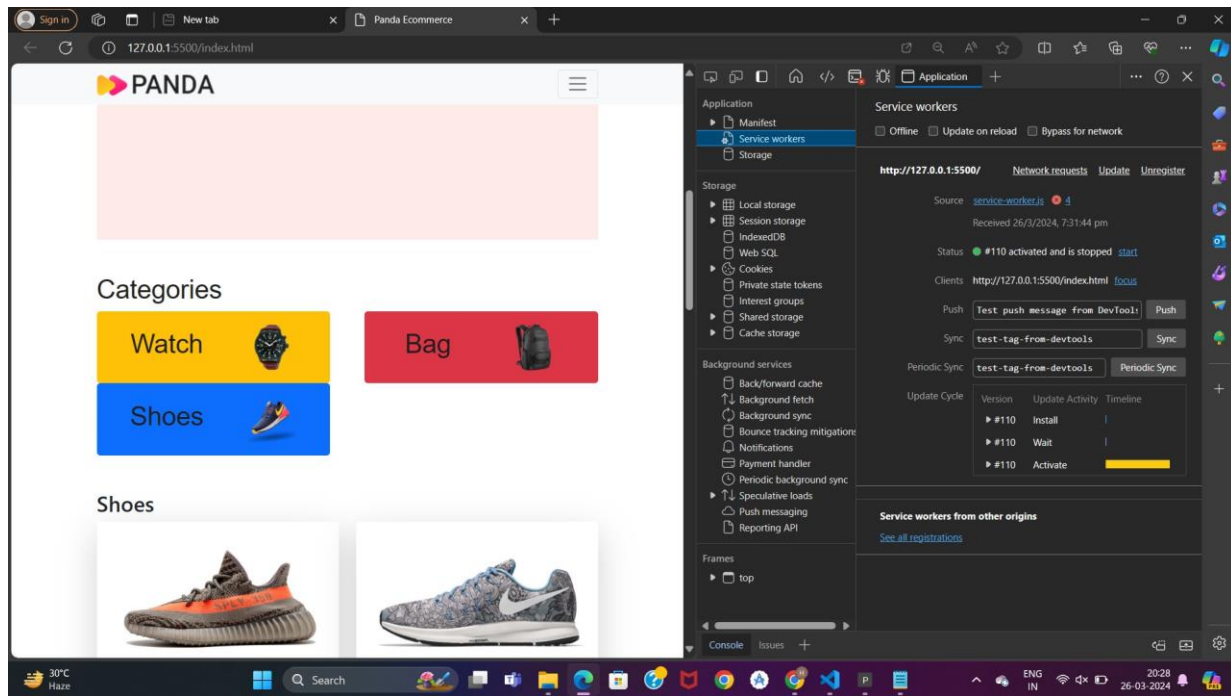
```
const cacheName = "panda-ecommerce-v1";
const assetsToCache = [
  '/',
  '/index.html',
  '/main.css',
  '/app.js',
  '/manifest.json',
  '/images/logo.png',
  '/images/banner-images/headphone.png',
  '/images/banner-images/tv.png',
  '/images/banner-images/xbox.png',
  '/images/categories/watch.png',
  '/images/categories/bag.png',
  '/images/categories/shoes.png',
  '/images/shoes/shoe-1.png',
  '/images/shoes/shoe-2.png',
  '/images/shoes/shoe-3.png',
  '/images/bags/bag-1.png',
  '/images/bags/bag-2.png',
  '/images/bags/bag-3.png'
];

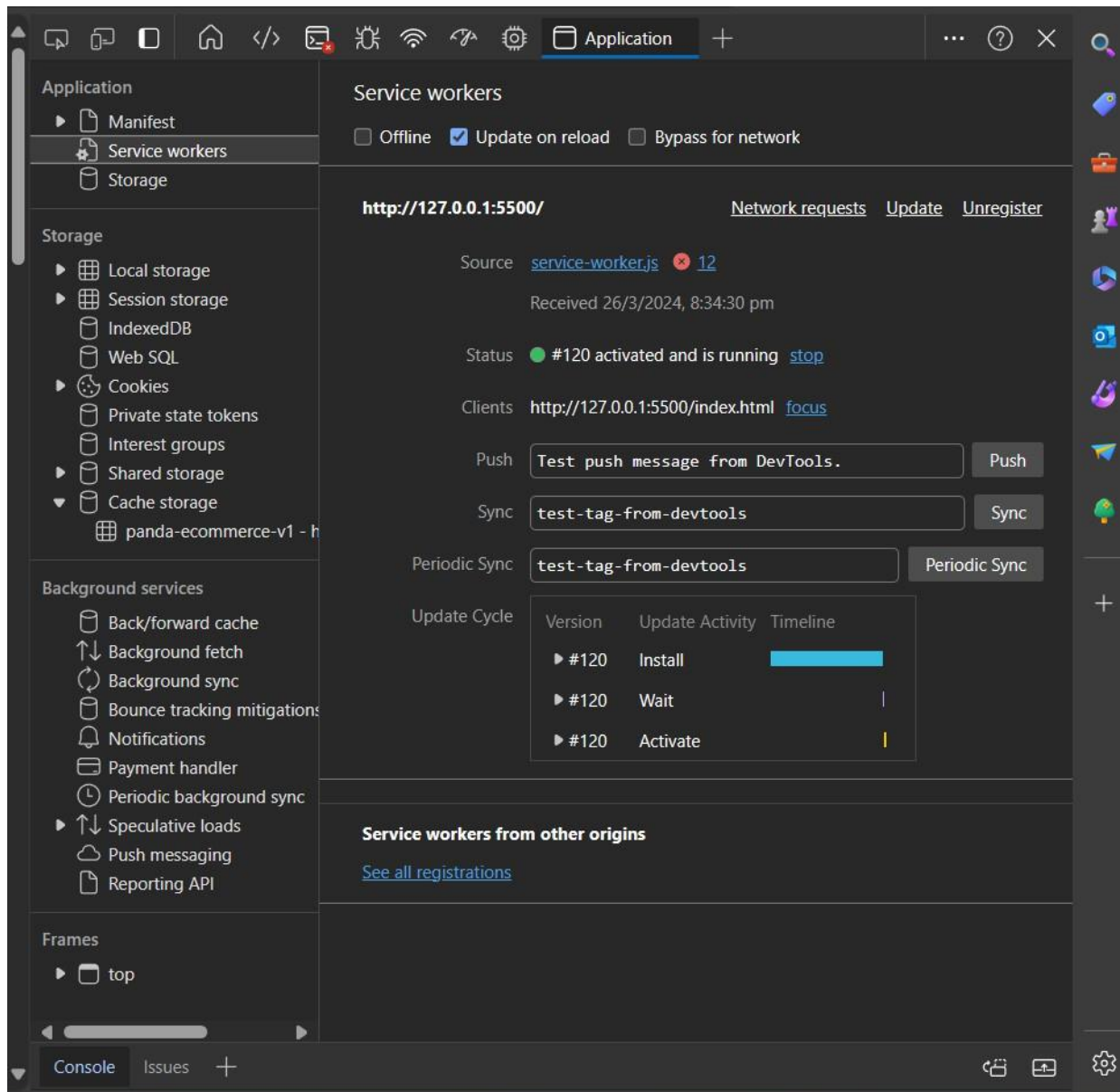
self.addEventListener('install', event => {
  event.waitUntil(
    caches.open(cacheName)
      .then(cache => {
        return cache.addAll(assetsToCache);
      })
  );
});

self.addEventListener('fetch', event => {
  event.respondWith(
    caches.match(event.request)
      .then(response => {
        return response || fetch(event.request);
      })
  );
});
```

OUTPUT:

Now Right click → Inspect → Application → Service workers





Now scroll down to Cache Storage

The screenshot shows a web browser window displaying a product page for 'PANDA' headphones. The page features a pink background with the text 'Cool Dude Headphone' and a price of '\$420'. A 'Buy Now' button is visible. The browser's address bar shows the URL '127.0.0.1:5500/index.html'. The Chrome DevTools 'Cache Storage' panel is open on the right side of the browser window. The panel shows the origin 'http://127.0.0.1:5500' and a list of cached resources. The 'Storage' tab is selected, and the 'Cache storage' section is expanded. The list of cached resources includes various image files and a manifest file. The 'Background services' section is also visible, showing a list of background tasks. The 'Frames' section shows the 'top' frame. The 'Console' and 'Issues' tabs are also visible at the bottom of the DevTools panel.

Cache Storage Panel Data:

#	Name	Resp...	Cont...	Cont...	Tim...	Vary...
1	/	basic text/...	15.8...	26/3...	Origin	
2	/images/bags/bag-1.png	basic ima...	2.29...	26/3...	Origin	
3	/images/bags/bag-2.png	basic ima...	621...	26/3...	Origin	
4	/images/bags/bag-3.png	basic ima...	1.98...	26/3...	Origin	
5	/images/banner-images/hea...	basic ima...	498...	26/3...	Origin	
6	/images/banner-images/tv.p...	basic ima...	629...	26/3...	Origin	
7	/images/categories/bag.png	basic ima...	162...	26/3...	Origin	
8	/images/categories/shoes.png	basic ima...	21.4...	26/3...	Origin	
9	/images/categories/watch.png	basic ima...	25.5...	26/3...	Origin	
10	/images/logo.png	basic ima...	27.5...	26/3...	Origin	