

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه فنی و حرفه ای

دانشگاه فنی و حرفه ای دختران نیشابور

گروه کامپیوتر

مقطع کاردانی

عنوان پروژه:

طراحی وب سایت نحوه انجام و مدیریت لیست کارهای روزانه (To Do List's Website)

استاد راهنما:

مهندس فاطمه بابادی

نام دانشجو:

شکیبا باغیشنی

تاریخ:

تابستان ۱۴۰۱

فهرست

۳	چکیده.....
۱	مقدمه.....
۱	فصل اول.....
۱	نیازسنجی.....
۱	امکان سنجی.....
۲	فصل دوم.....
۲	طراحی ظاهری (UI / UX).....
۲	طراحی.....
۲	دکمه اضافه کردن تسک و پیغام اضافه شدن تسک.....
۳	حذف تسک.....
۳	ریلود شدن و پاک نشدن تسک ها از لوکال استوریج.....
۴	فصل سوم.....
۴	منطق برنامه (syntax).....
۴	Html code.....
۵	Css code (style).....
۶	Javascript.....
۱۱	فصل چهارم.....
۱۱	هاست.....
۱۲	نتیجه گیری.....
۱۳	کارهای آتی.....
۱۴	سپاس گذاری.....
۱۵	منابع.....

چکیده

To Do List با اسم «لیست انجام کار» یا «لیست کارهای روزانه» هم شناخته می‌شود. این لیست مجموعه‌ای از کارهایی است که ما می‌خواهیم در طول یک روز انجام دهیم. عملکرد مغز به گونه‌ای است که هر چقدر تمرکز خود را بر روی یک کار خاص بگذارد، بهره‌وری در آن کار بالا می‌رود، اما در عوض این متمرکز کردن انرژی و افکار باعث می‌شود تا مسائل دیگر به فراموشی سپرده شود. بعضی افراد لیست کارهای روزانه خود را روی کاغذ می‌نویسند و برخی دیگر هم از اپلیکیشن‌ها (TO DO) و نرم افزارهای موجود برای نوشتن To Do List خود استفاده می‌کنند. داشتن لیست کارهای روزانه به شما کمک می‌کند که زندگی خود را طبق برنامه پیش برده و از لحظات زندگی خود بیشترین استفاده را ببرید. زمان، ارزشمندترین دارایی هر انسان است و اگر انسان برای این زمان ارزش و اهمیت قائل شود و برای آن برنامه داشته باشد، می‌تواند زندگی خود را متحول کند.

مقدمه

فرض کنید که صبح از خواب بیدار می‌شوید و هیچ برنامه‌ای هم ندارید. به صورت شانس‌ی هر کاری به ذهنتان رسید انجام می‌دهید و خبر ندارید که کارهای مهم‌تری هست که باید سریع‌تر انجام دهید. همچنین نمی‌دانید که برای انجام یک کار مشخص چقدر زمان دارید. در نتیجه کاری که نهایتاً یک ساعت زمان می‌خواهد، ممکن است ۴ ساعت طول بکشد. بنابراین با استفاده از To Do List می‌توانید با یک ذهن منظم و برنامه دقیق به استقبال روز بروید.

وقتی بدانید که امروز قرار است چه کارهایی انجام دهید، مسلماً ذهنتان نظم می‌گیرد و از وظایف خودش آگاه می‌شود. بنابراین وقتی سراغ انجام کارها می‌روید، خیلی بهتر و سریع‌تر عمل می‌کنید؛ چون ذهنتان آمادگی همه چیز را دارد. به همین خاطر است که نوشتن To Do List بهره‌وری شما را به میزان قابل توجهی افزایش می‌دهد.

دفترهای برنامه ریزی و پلنرهای مختلفی وجود دارد که به صورت فیزیکی می‌توان برای برنامه ریزی از آنها استفاده کرد. اما با پیشرفت تکنولوژی به راحتی می‌توان پلنر خود را به روی گوشی یا در لپ‌تاپ خود داشته باشید.

To Do List یا در بستر وبسایت یا در بستر اپلیکیشن است که اپلیکیشن موبایل و دسکتاپ متفاوت هستند بنابراین داشتن لیست روزانه در وبسایت بهترین انتخاب جهت دسترسی راحت است.

معمولاً از شب قبل لیست کارهای مربوط به روز را ثبت کرده و زمانی که کاری را به پایان می‌رسانید، روی آن کار خط می‌کشید. این کار باعث می‌شود که حس پیشرفت و به نتیجه رساندن چیزی در شما القا شود، و این حس دقیقاً همان چیز است که اگر در انجام کارهای خود عجله داشته باشید و از کارها عقب باشید، احساس نمی‌کنید. و به دست آوردن این احساس موفقیت باعث می‌شود تا انگیزه شما برای پیشرفت و حرکت رو به جلو افزایش پیدا کند.

برای آشنایی بیشتر و مزایای استفاده از to do list تا پایان این پروژه همراه باشید.

فصل اول

نیازسنجی

نوشتن To Do List برای شما نقش یک قرص ضد فراموشی را دارد که پس از مصرف آن دیگر کارهای خودتان را فراموش نمی‌کنید و می‌توانید هر کار را به موقع و درست انجام دهید. بدون نوشتن لیست کارهای روزانه حتماً به مشکل برمی‌خورید و بخشی از کارهای روزانه خود را فراموش خواهید کرد. بنابراین برای جلوگیری از فراموش کردن کارها حتماً لیست کارها را یادداشت کنید.

اگر می‌خواهید به اهداف خود برسید و عملکرد خود را بهبود ببخشید باید لیست کارهای روزانه خودتان را به یک هدف مهم، مشخص و زمان‌دار گره بزنید. مثلاً بگویید که باید در پایان این هفته، فلان مقدار درآمد کسب کنم تا بتوانم فلان کالا را خریداری کنم. البته حتماً سعی کنید اهداف خود را یادداشت کنید چرا که این کار انگیزه شما را برای رسیدن به اهداف چند برابر می‌کند.

به نظر می‌رسد اینکه کارها و اهداف روزانه خود را روی کاغذ پیاده کنید نه تنها کمک بزرگی به تعیین چالش‌های مهم و آزاد کردن فضای ذهن برای تمرکز بیشتر می‌کند بلکه باعث می‌شود تا از اتلاف وقت در تلاش برای شناسایی مهم‌ترین کار بعدی جلوگیری شود و کار یا هدفی را فراموش نکنید. به طور کلی To Do List راهی برای افزایش بهره‌وری ارائه می‌دهد، شما را از فراموش کردن کارها باز می‌دارد، به اولویت بندی کارها، مدیریت موثر وظایف، استفاده عاقلانه از زمان و بهبود مدیریت زمان و همچنین گردش کار کمک می‌کند.

امکان سنجی

امکانات لازمه جهت این برنامه به دو دسته سخت افزاری و نرم افزاری و بستر اینترنت تقسیم میشود.

از جمله امکانات لازمه سخت افزاری به کامپیوتر معمولی و یا گوشی همراه نیاز است.

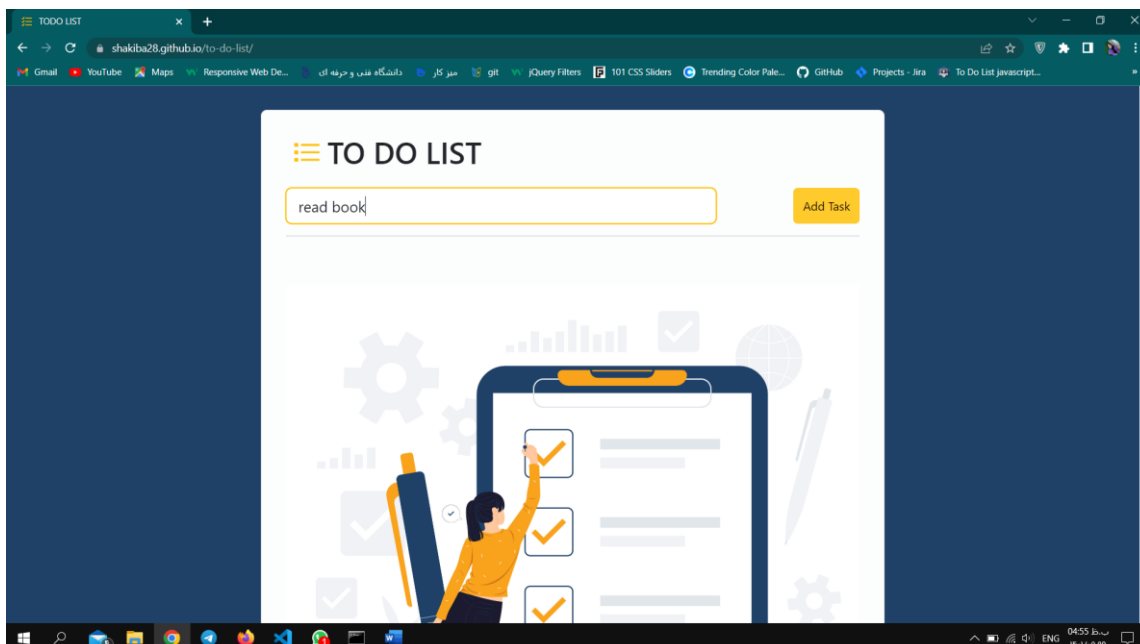
و به نرم افزار های vs code جهت نوشتن کد ها و یک مرورگر جهت اجرای برنامه و دیدن خروجی .

همچنین از سایت گیت هاب برای آپلود پروژه و مدیریت ورژن های پروژه استفاده میشود و به عنوان هاست نیز از گیت هاب استفاده شده.

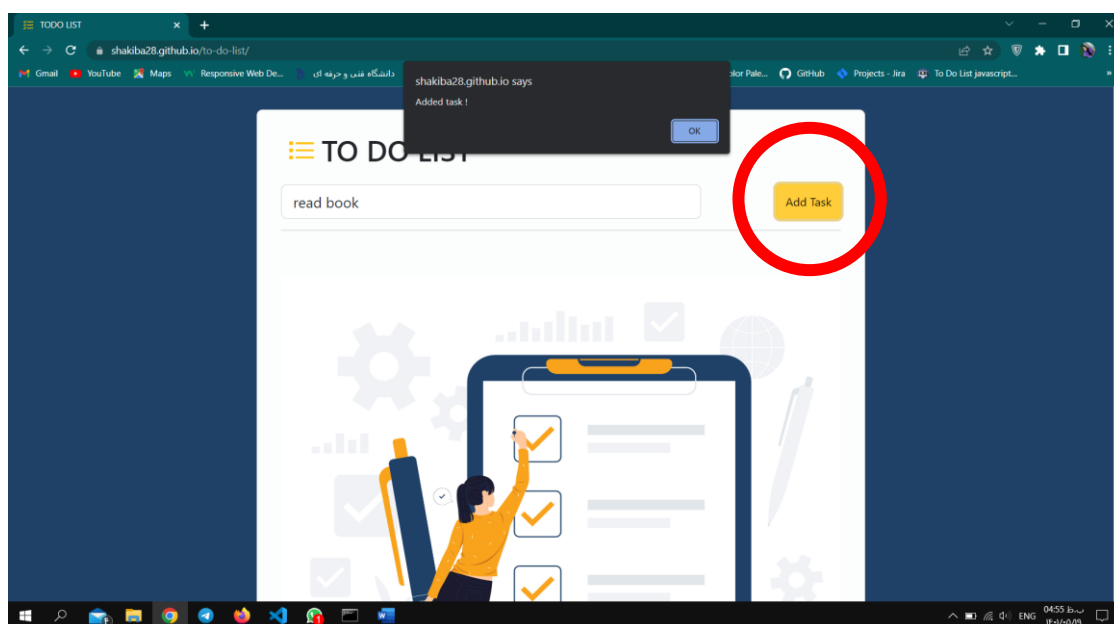
فصل دوم

طراحی ظاهری (UI / UX)

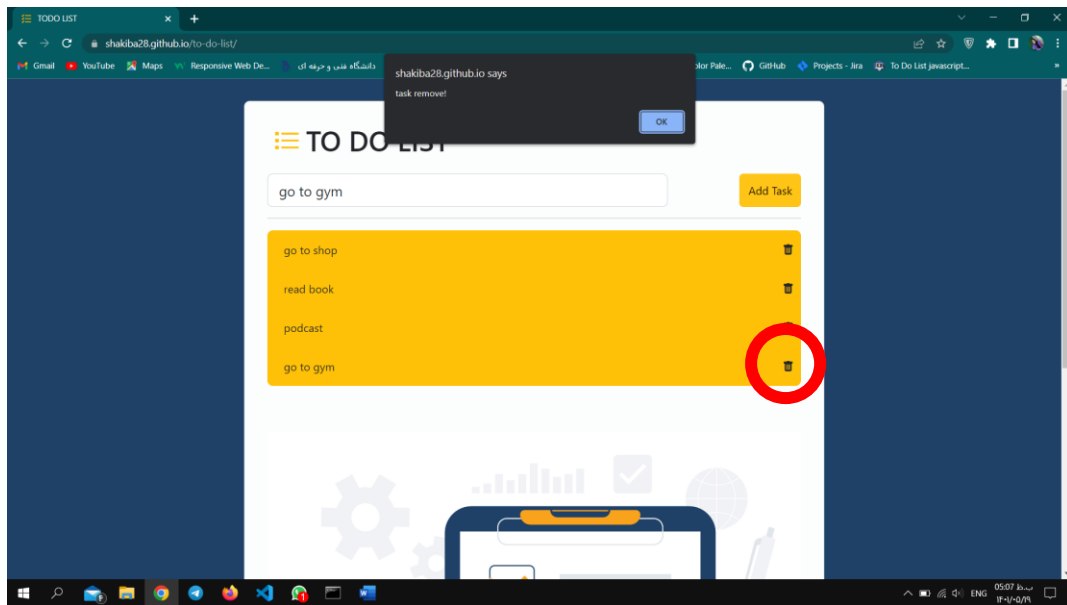
طراحی



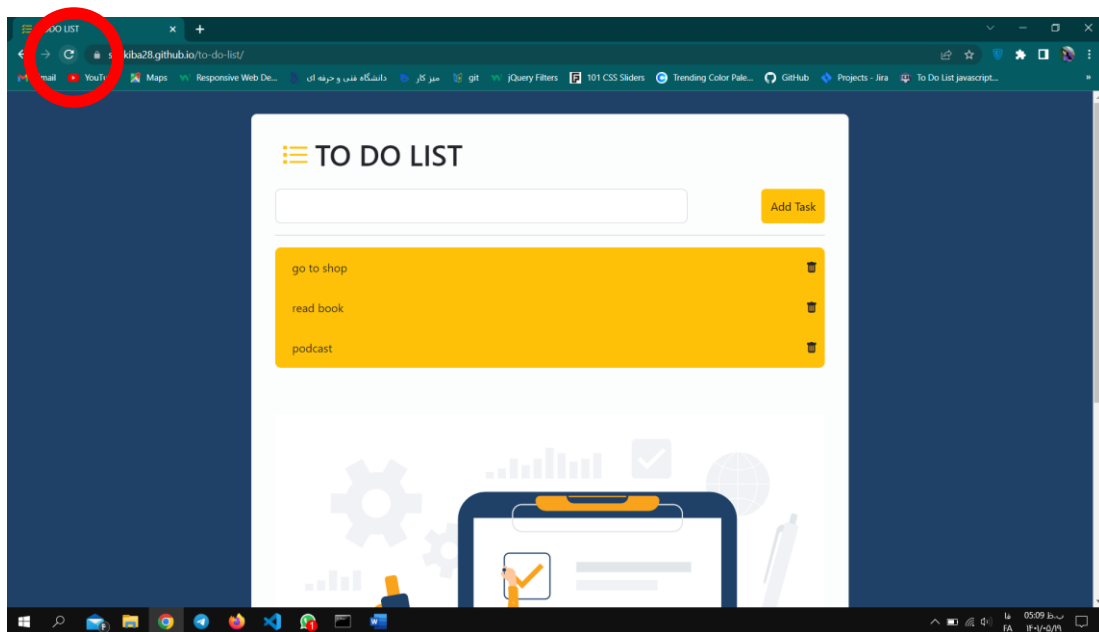
دکمه اضافه کردن تسک و پیغام اضافه شدن تسک



حذف تسک



ریلود شدن و پاک نشدن تسک ها از لوکال استوریج



فصل سوم

منطق برنامه (syntax)

Html code

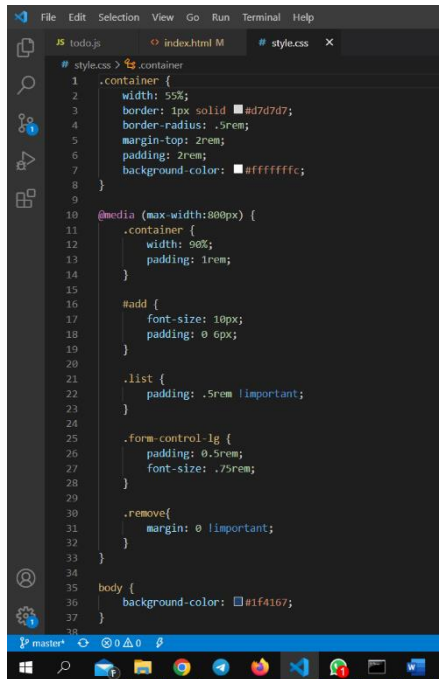
ابتدا نرم افزار vs code را باز کرده و پروژه جدید را باز میکنیم. صفحه index.html کد های html و لینک های مربوطه را وارد میکنیم .

از تگ input برای وارد کردن تسک و یک دکمه جهت افزودن تسک استفاده میکنیم. همچنین از تگ ul جهت لیست شدن تسک ها.

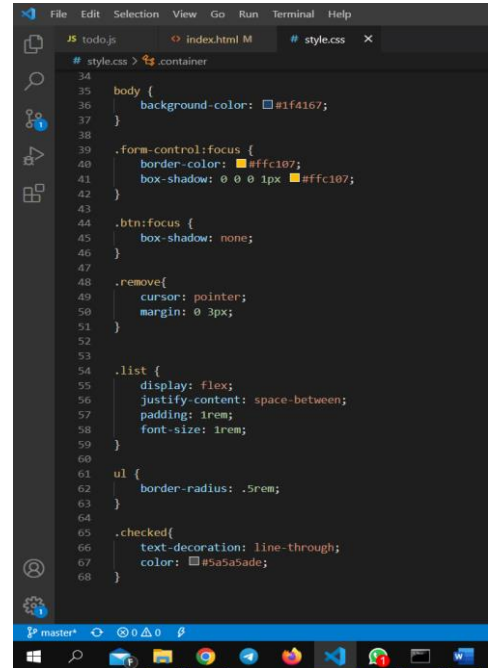
```
1 <html lang="en">
2
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@2.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet"
8     integrity="sha384-0ev442398Q260Zh2oMrBNu69DR0xN0mL666BwOjD6y8FjVab00fYT5znPsdS0e" crossorigin="anonymous">
9
10  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.9.0/font/bootstrap-icons.css">
11
12  <link rel="stylesheet" href="style.css">
13
14  <link rel="icon" type="image/png" href="https://img.icons8.com/ios-filled/50/FAB005/todo-list.png" />
15
16  <title>TODO LIST</title>
17 </head>
18
19 <body>
20   <div class="container">
21     <div class="header d-flex">
22       <h1 class="mb-4"><i class="bi bi-list-task text-warning mx-2"></i>TO DO LIST</h1>
23     </div>
24     <div class="d-flex justify-content-between">
25       <input id="task" class="form-control form-control-lg w-75">
26       <button class="btn btn-warning" id="add">Add Task</button>
27     </div>
28
29     <hr>
30     <div id="todos"></div>
31     <div id="victory" class="text-center">
32       
33     </div>
34     <script src="todo.js">
35   </script>
36 </body>
37 </html>
```

Css code (style)

استایل های مربوط به هر المنت را در فایل style.css وارد میکنیم همچنین برخی از استایل ها توسط فریمورک بوت استرپ داده شده است.



```
# style.css > .container
1 .container {
2   width: 55%;
3   border: 1px solid #d7d7d7;
4   border-radius: .5rem;
5   margin-top: 2rem;
6   padding: 2rem;
7   background-color: #ffffff;
8 }
9
10 @media (max-width:800px) {
11   .container {
12     width: 90%;
13     padding: 1rem;
14   }
15
16   #add {
17     font-size: 10px;
18     padding: 0 6px;
19   }
20
21   .list {
22     padding: .5rem 1important;
23   }
24
25   .form-control-lg {
26     padding: 0.5rem;
27     font-size: .75rem;
28   }
29
30   .remove{
31     margin: 0 1important;
32   }
33 }
34
35 body {
36   background-color: #f4f4f4;
37 }
```



```
# style.css > .container
34
35 body {
36   background-color: #f4f4f4;
37 }
38
39 .form-control:focus {
40   border-color: #ffc107;
41   box-shadow: 0 0 0 1px #ffc107;
42 }
43
44 .btn:focus {
45   box-shadow: none;
46 }
47
48 .remove{
49   cursor: pointer;
50   margin: 0 3px;
51 }
52
53
54 .list {
55   display: flex;
56   justify-content: space-between;
57   padding: 1rem;
58   font-size: 1rem;
59 }
60
61 ul {
62   border-radius: .5rem;
63 }
64
65 .checked{
66   text-decoration: line-through;
67   color: #5a5a5a;
68 }
```

همانطور که این تابع توسط همه توابع دیگر استفاده می شود، ابتدا تابع `get_todos` را ببینیم. هیچ پارامتری دریافت نمی کند. فقط با استفاده از روش `getItem` محتویات کلید `todo` را در `localStorage` واکشی می کند. اگر این اولین باری است که تابع فراخوانی می شود، ورودی `localStorage` خاص خالی خواهد بود و فراخوان `localStorage.getItem` پوچ خواهد شد. در آن صورت آرایه خالی تازه ایجاد شده را برمی گردانیم. اگر مقدار برگشتی تهی نباشد، باید داده های رشته ای باشد که قبلاً ذخیره کرده بودیم. ما از `JSON.parse` برای تبدیل رشته `JSON` به داده های جاوا اسکریپت و برگرداندن آن استفاده می کنیم.

```

JS todo.js  X  <> index.html M  # style.css

JS todo.js > checked
1  function get_todos() {
2      var todos = new Array;
3      var todos_str = localStorage.getItem('todo');
4      if (todos_str !== null) {
5          todos = JSON.parse(todos_str);
6      }
7      return todos;
8  }
9

```

دومین تابعی که ممکن است بخواهیم به آن نگاهی بیندازیم، تابعی به نام add است که زمانی فراخوانی می شود که کاربر روی دکمه Add کلیک کرده باشد. در ابتدا، با استفاده از `getElementById` از HTML را با وظیفه id که کادر ورودی است، قرار می دهد و سپس مقداری را که کاربر تایپ کرده است، بازیابی می کند. سپس، با فراخوانی `get_todos`، فهرست موجود از موارد TODO را از "پایگاه داده" بازیابی می کنیم. همانطور که در بالا توضیح داده شد، در اولین بار این تابع یک آرایه خالی را برمی گرداند.

تسک جدید را با استفاده از روش `push` به آرایه اضافه می کنیم و سپس لیست جدید موارد TODO را در "پایگاه داده" ذخیره می کنیم. برای این کار ابتدا آرایه را با استفاده از متد `JSON.stringify` و سپس رشته برگشتی را با استفاده از متد `localStorage.setItem` ذخیره می کنیم. در مرحله بعد تابع `show()` را فراخوانی می کنیم که لیست TODO های نمایش داده شده در صفحه وب را به روز می کند. سرانجام ما `false` را برمی گردانیم. برای جلوگیری از هرگونه اقدام بعدی که توسط رویداد "کلیک" ایجاد می شود.

```
10 //Add the task to the array
11
12 function add() {
13     var task = document.getElementById('task').value;
14     if (task === "") {
15         alert("enter your task !")
16     } else {
17         var todos = get_todos();
18         todos.push(task);
19         alert("Added task !");
20         localStorage.setItem('todo', JSON.stringify(todos));
21
22         show();
23
24         return false;
25     }
26 }
27
28
```

تابع نمایش لیست TODO فعلی ذخیره شده در "پایگاه داده" را نمایش می دهد. اولین چیزی که `get_todos` را فراخوانی می کند تا آرایه (احتمالاً خالی) آیتم های TODO را دریافت کند. سپس به صورت دستی یک قطعه HTML در متغیر `html` ایجاد می کنیم. این یک عنصر `ul` (و لیست نامرتب)، با یک `li` (مورد لیست) برای هر ورودی TODO است. علاوه بر محتوای آرایه `todos`، ما همچنین یک دکمه را به هر آیتم لیست اضافه می کنیم. هر دکمه متعلق به کلاسی به نام "حذف" است و هر دکمه دارای شناسه ای است که حاوی نمایش مورد انجام کار در لیست بازبینی شده از "پایگاه داده" است. ما از این دکمه ها استفاده می کنیم تا به کاربر اجازه دهیم یک مورد را از لیست حذف کند.

فراخوانی `document.getElementById('todos').innerHTML = html`; قطعه HTML در واقع محتوای عنصر را با شناسه "todos" جایگزین می کند. این بدان معناست که در تماس های بعدی فقط لیست جدید را بدون توجه به آنچه قبلاً وجود داشت نشان می دهد.

در ۴ خط بعدی از متد `getElementsByClassName` برای واکنشی همه دکمه هایی که در کلاس «حذف» هستند استفاده می کنیم. اینها دکمه هایی هستند که ما به تازگی به هر مورد `todo` اضافه کرده ایم. به هر دکمه یک شنونده رویداد اختصاص می دهیم که اگر کاربر روی هر یک از آن دکمه ها کلیک کند، فراخوانی می شود. فراخوانی به `addEventListener` رویداد 'کلیک' را به تابع `remove` متصل می کند.

```
54
55 function show() {
56     var todos = get_todos();
57
58     var html = '<ul class="bg-warning list-unstyled" id="ulist">';
59     for (var i = 0; i < todos.length; i++) {
60         html += '<div class="d-flex justify-content-between align-items-center mx-2"><li class="list">
61             | + todos[i] + '</li><i class="bi bi-trash-fill remove" id="' + i + '"></i></div>';
62     };
63     html += '</ul>';
64
65     document.getElementById('todos').innerHTML = html;
66
67     var buttons = document.getElementsByClassName('remove');
68     for (var i = 0; i < buttons.length; i++) {
69         buttons[i].addEventListener('click', remove);
70     };
71 }
72
73 document.getElementById('add').addEventListener('click', add);
74
75 show();
76
```

با استفاده از CSS کلاسی به اسم checked تعریف شده که استایل text-decoration و زمانی این کلاس به تسک افزوده میشود که کاربر روی تسک مد نظر کلیک کند. (به معنای انجام دادن تسک) این تابع به محض لود شدن صفحه اجرا میشود.

```
43 window.onload = function () {
44     checked();
45 };
46 function checked() {
47     let list = document.querySelector('ul');
48     list.addEventListener('click', function (ev) {
49         if (ev.target.tagName === 'LI') {
50             ev.target.classList.toggle('checked');
51         }
52     }, false);
53 }
54
```

در نهایت به تابع `remove` می‌رسیم که با کلیک کاربر روی هر یک از دکمه‌های حذف فراخوانی می‌شود. این شیء DOM فعلی را نشان می‌دهد که دکمه حذفی است که کاربر روی آن کلیک کرده است. مقدار ویژگی `id` آن را با استفاده از متد `getAttribute` بازیابی می‌کنیم. این نمایه مورد خاص `TODO` در میان موارد `TODO` در "پایگاه داده" است.

پس از بازیابی لیست فعلی موارد `TODO`، از روش `splice` برای حذف یک عنصر خاص از آرایه جاوا اسکریپت استفاده می‌کنیم و سپس لیست جدید را در پایگاه داده ذخیره می‌کنیم.

سپس، همانطور که در تابع `add`، تابع `show` را فراخوانی می‌کنیم تا لیست در مرورگر نیز به روز شود و `false` را برمی‌گردانیم. برای جلوگیری از انتشار رویداد 'کلیک'.

```
28
29 function remove() {
30     var id = this.getAttribute('id');
31     var todos = get_todos();
32     todos.splice(id, 1);
33     alert("task remove!");
34     localStorage.setItem('todo', JSON.stringify(todos));
35
36     show();
37
38     return false;
39 }
40
```


فصل چهارم

هاست

گیت هاب یک پلتفرم برای برنامه نویسان و توسعه دهندگان است. افراد در گیت هاب می توانند پروژه های خود را به اشتراک بگذارند و با هم تیمی های خود به صورت مشترک روی آن کار کنند. این سایت امکانات مختلفی را مانند سیستم کنترل نسخه در اختیار توسعه دهندگان قرار می دهد. یکی از ویژگی های دیگری که گیت هاب دارد، امکان میزبانی رایگان صفحات وب است و شما می توانید یک سایت کاملا رایگان برای خودتان درست کنید. از این سایت میتوانید برای نشان دادن رزومه خود یا معرفی یک پروژه، محصول، شرکت یا ساخت یک وبلاگ استفاده کنید.

برای شروع، اگر از قبل در گیت هاب اکانت ندارید ابتدا باید یک اکانت در سایت گیت هاب درست کنید. برای این کار به آدرس سایت [github](https://github.com) رفته و اکانت خود را درست کنید. پس از اینکه اکانت خود را ساختید، باید یک ریپازیتوری برای خودتان ایجاد کنید. برای این کار روی علامت + کلیک کرده و گزینه **New repository** را انتخاب کنید. در این قسمت Repository name همان آدرس و نام وبسایت شماست مانند to-do-list.ir

حتما گزینه Public عمومی را انتخاب کنید تا وبسایتی که در هاست گیت هاب می سازید نمایش داده شود. سپس بر روی گزینه Create repository کلیک کنید تا پروژه ساخته شود.

برای قرارگیری بر روی هاست به سربرگ setting ریپازیتوری مد نظر رفته و تب page و لینک مربوطه را در مرورگر خود کپی کنید و به صورت زنده صفحه وب خود را مشاهده کنید.

نتیجه گیری

فرض کنید صبح از خواب بیدار شده اید و برنامه هایی که در طول روز قرار به انجام است را در ذهن خود دارید.

بنظر شما اکتفا کردن به ذهن تا چه اندازه درست و منطقی است؟

ممکن است که بارها برای شما پیش آمده باشد برنامه های خود را فراموش کرده باشید بنابراین نمیشود برنامه های روزانه را در ذهن خود مدیریت و دسته بندی کرد. برای حل این مشکل برنامه to do list به کمک شما آمده تا کارهای مدنظر خود را وارد و در طول روز انجام دهید بدون اینکه از یاد ببرید یا اینکه برنامه ای را جا بیندازید.

کارهای آتی

از جمله طرح های توسعه ی این برنامه میشود به :

- دسته بندی اولویت تسک ها
- ویرایش تسک مد نظر
- فیلتر تسک (مثلا تسک های انجام شده و انجام نشده)
- تاریخ و ساعت برای ایجاد و انجام تسک
- گذاشتن یادآور جهت انجام تسک

اشاره کرد.

سپاس گذاری

در پایان ممنونم از استاد مربوطه ، مهندس بابادی از دوست عزیزم و گروه توسعه دهندگاه جاوا اسکریپت همچنین سایت های مربوطه و یوتیوب.

- <https://code-maven.com/todo-in-html-and-javascript>
- <https://sabzlearn.ir/save-information-with-js/>
- <https://github.com/>
- <https://www.w3schools.com/js/>
- https://www.w3schools.com/howto/howto_js_todolist.asp
- <https://codepen.io/tag/todolist>
- <https://youtube.com/>