

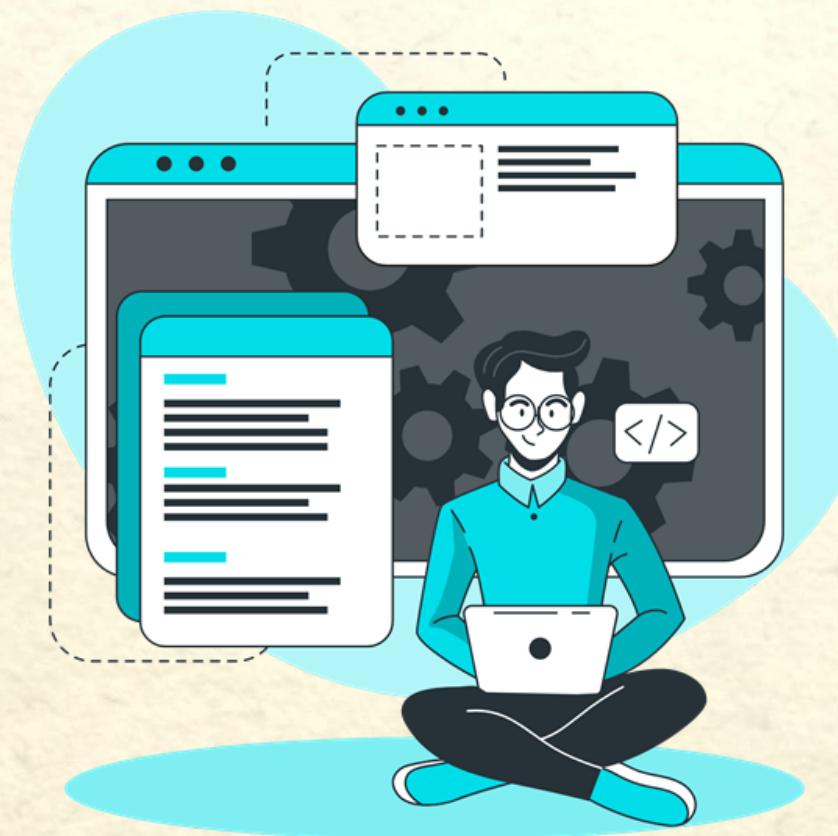
# مکتب شریف

اولین بوت‌کمپ آموزشی-استخدامی ایران



سri چهاردهم

مکتب 88





## Express (1) بخش

1-1. با استفاده از Express یک وب اپلیکیشن بنویسید:

- **GET** /product/get-all-products  
تمامی محصولات را به عنوان response برمیگرداند.
- **GET** /product/get-product/id  
محصول مورد نظر را بر اساس id پیدا میکند و به عنوان response برمیگرداند.
- **POST** /product/create-product  
محصولی جدید به products-data.json اضافه میکند.
- **PUT/PATCH** /product/update-product/id  
محصول مورد نظر را بر اساس id میابد و آپدیت میکند.
- **DELETE** /product/remove-product/id  
محصول مورد نظر را بر اساس id میابد و از product-data حذف میکند.

**نکته:** فایل products-data.json پیوست شده است.

1-2. برای تمرین قبلی (1-1) **کلاینت** پیاده سازی کنید:

**نکته:** فایل products-data.json پیوست شده است.

**GET /products-page**

وقتی به آدرس فوق درخواست زده شد اطلاعات محصولات به صورت یک **جدول** نمایش داده شود.

Products Info

No.	Product ID	Title	Price	Rating	Stock	Brand	Category
1	1	iPhone 9	549	4.69	94	Apple	smartphones
2	2	iPhone X	899	4.44	34	Apple	smartphones
3	3	Samsung Universe 9	1249	4.09	36	Samsung	smartphones
4	4	OPPOF19	280	4.3	123	OPPO	smartphones
5	5	Huawei P30	499	4.09	32	Huawei	smartphones
6	6	MacBook Pro	1749	4.57	83	Apple	laptops
7	7	Samsung Galaxy Book	1499	4.25	50	Samsung	laptops
8	8	Microsoft Surface Laptop 4	1499	4.43	68	Microsoft Surface	laptops
9	9	Infinix INBOOK	1099	4.54	96	Infinix	laptops
10	10	HP Pavilion 15-DK1056WM	1099	4.43	89	HP Pavilion	laptops

- وقتی روی **هر سطری** از جدول کلیک شد **اطلاعات محصول** نمایش داده شود.
- امکان **ویرایش** کردن اطلاعات محصول و **حذف** کردن محصول را فراهم کنید.
- امکان **افزودن** محصولی جدید را فراهم کنید.

**توجه کنید** که پس از ساختن، **ویرایش** و **حذف** محصول فایل products-data.json بروزرسانی شود.

**نکته:** طراحی کلاینت **دلخواه** است.

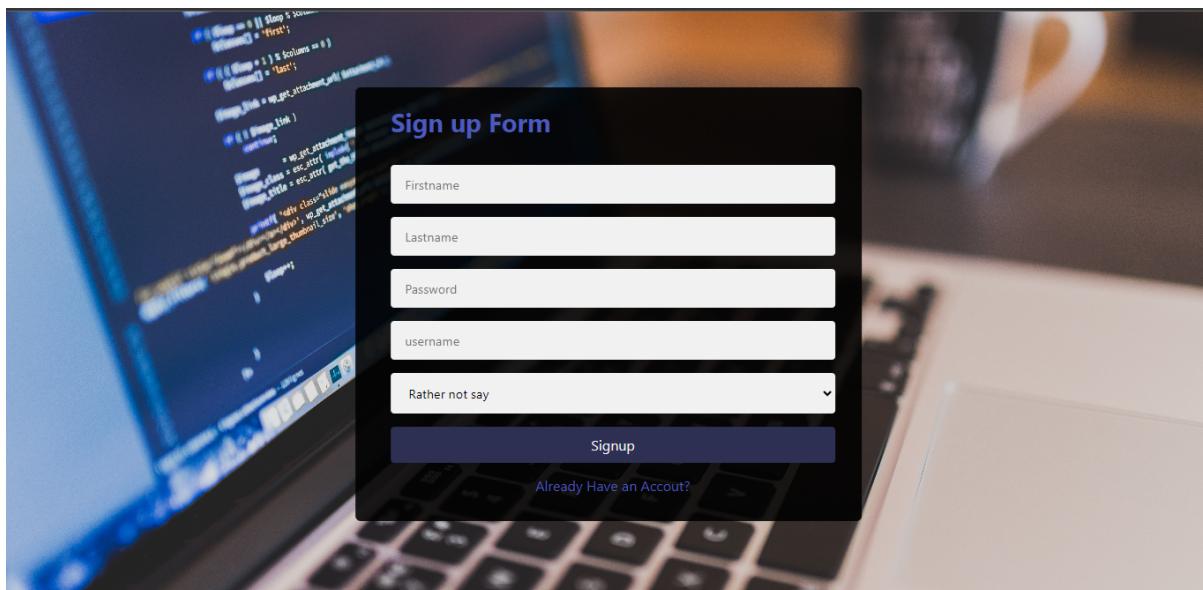
1-3. با استفاده از **Express** یک وب اپلیکیشن بنویسید:

**نکته:** فایل **users-data.json** پیوست شده است.

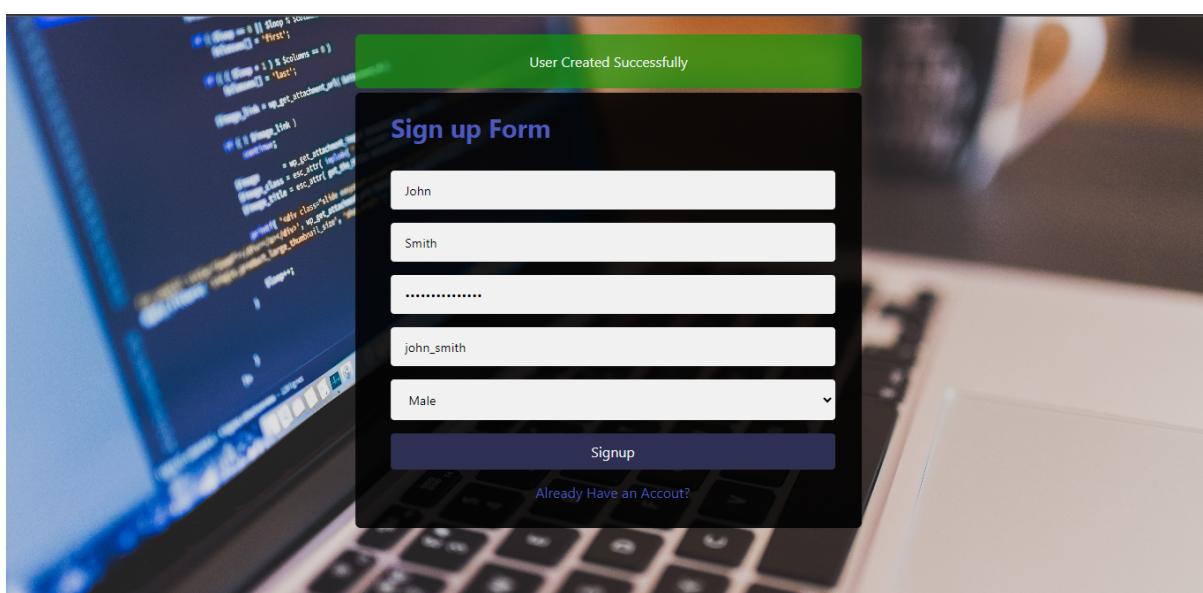
## /auth

- **GET /auth/signup**

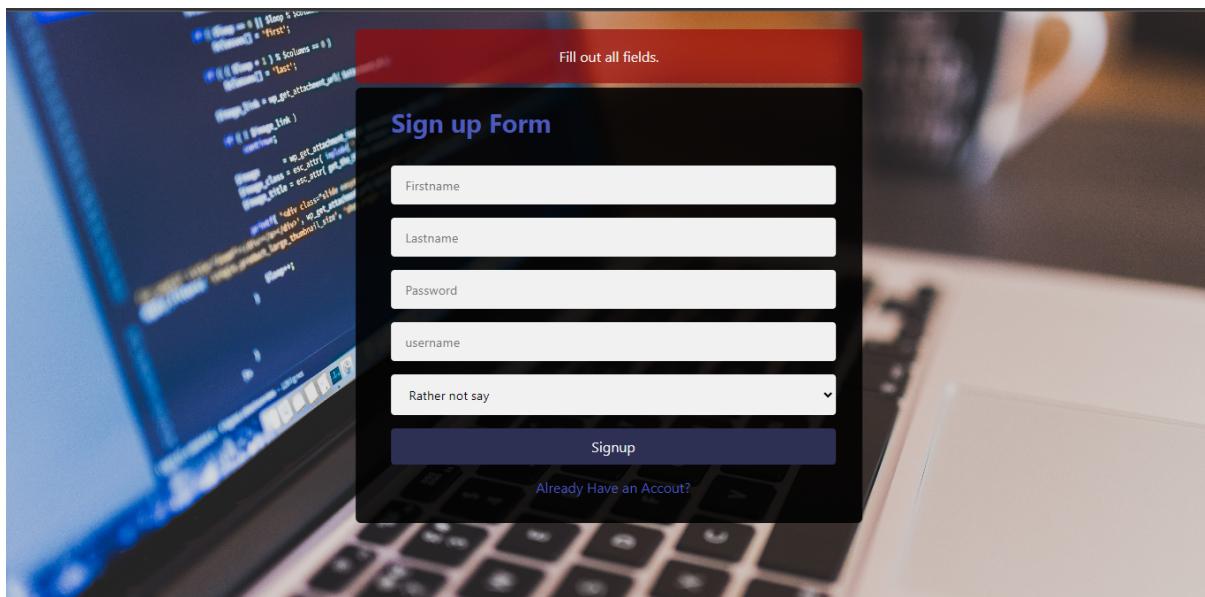
صفحه **signup** را رندر میکند که شامل **فرم ثبت نام** میباشد.



اگر ثبت نام **موفقیت آمیز** بود پیغام موفقیت آمیز برای کاربر نمایش میدهد.



اگر ثبت نام موفقیت آمیز نبود پیغام خطا مناسب برای کاربر نمایش میدهد.

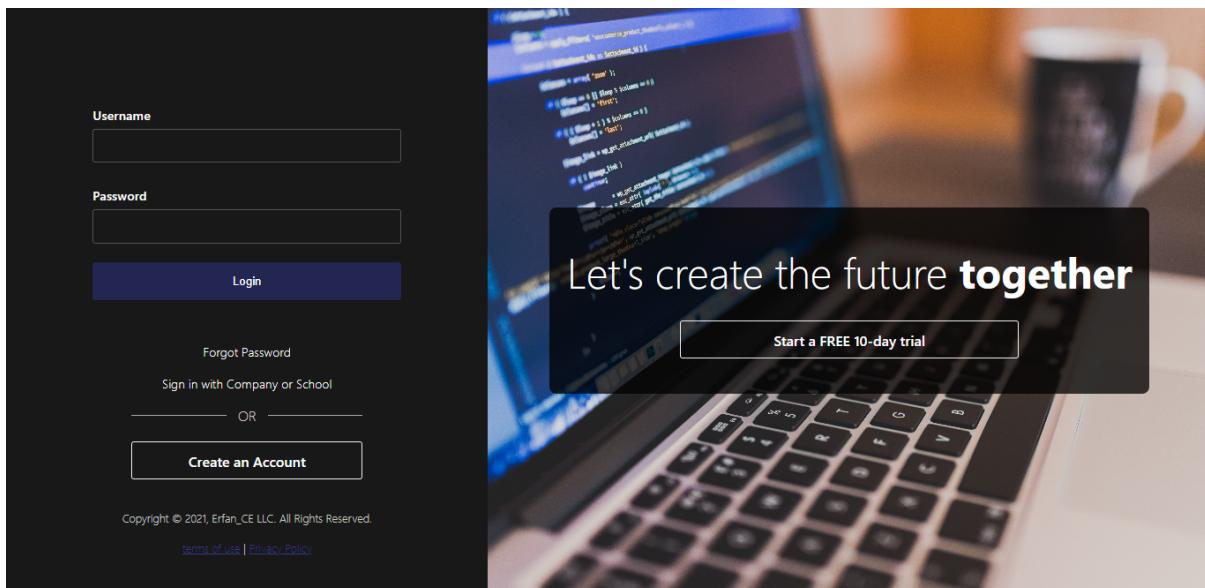


- **POST /auth/signup**

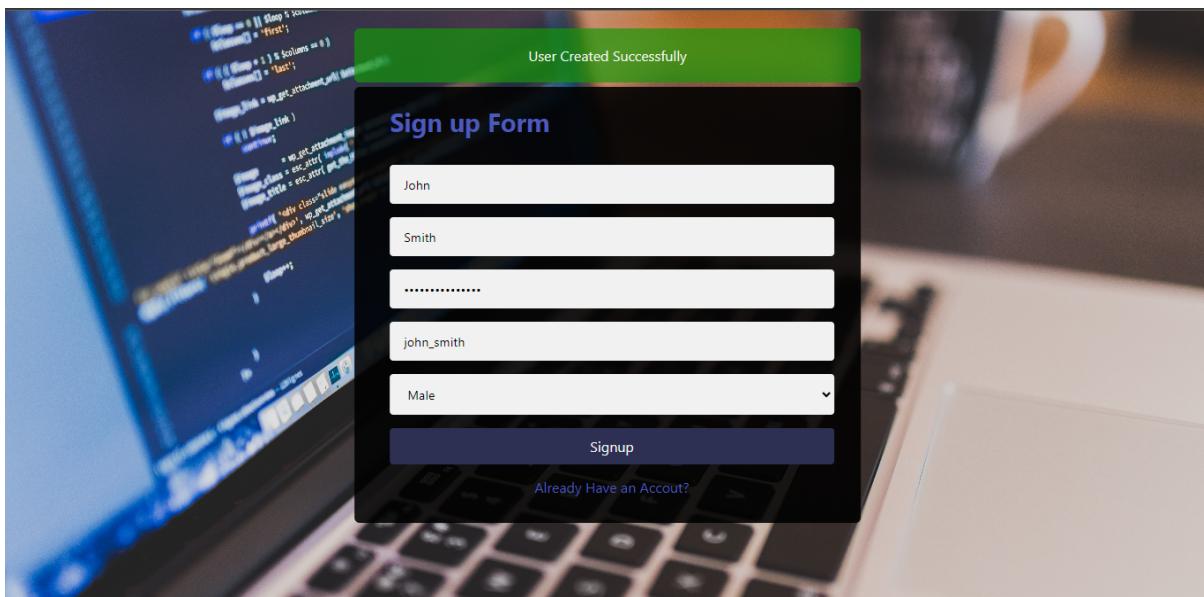
اطلاعات کاربر را که شامل **firstname**, **lastname**, **username**, **password**, **gender** را دریافت میکند و کاربری جدید می سازد و به **users-data.json** اضافه میکند.  
**توجه کنید که username باید یکتا باشد.**

- **GET /auth/login**

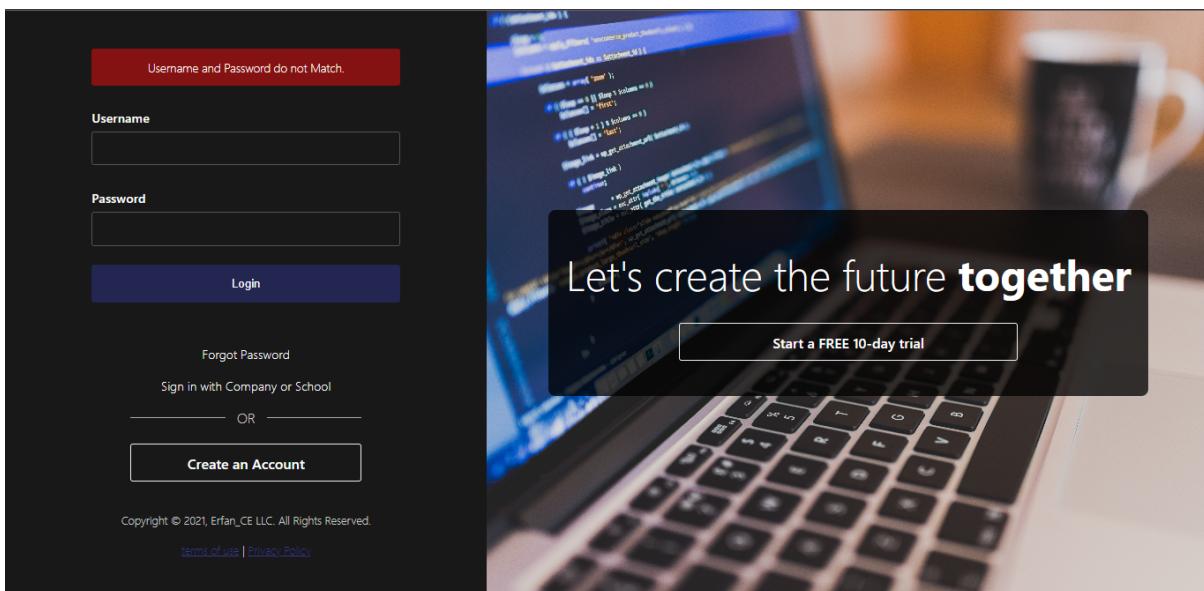
صفحه login-page.html را رندر میکند که شامل **فرم ورود** میباشد.



اگر عملیات ورود **موفقیت آمیز** بود پیغام موفقیت آمیز برای کاربر نمایش میدهد.



اگر عملیات ورود **موفقیت آمیز نبود** پیغام خطا مناسب برای کاربر نمایش میدهد.



- **POST /auth/login**

اطلاعات کاربر را که شامل username, password را دریافت میکند و اگر کاربری با این مشخصات در فایل **users-data.json** موجود بود response مناسب را ارسال میکند.

## /admin

- **GET /admin/get-all-users**

اطلاعات تمامی کاربران را به عنوان response برمیگرداند.

- **GET /admin/get-user/username**

اطلاعات کاربر مورد نظر را بر اساس username میابد و به عنوان response برمیگرداند.

- **DELETE /admin/remove-user/username**

کاربر مورد نظر را بر اساس username میابد و از users-data.json حذف میکند.

- **GET /admin/panel**

صفحه admin-panel.html را رندر میکند که در آن اطلاعات کاربران به صورت یک جدول نمایش داده شده است.



### Users Info

Username	First name	Last name	Gender
linus_torvalds	Linus	Torvalds	male
richard_stallman	richard	stallman	male
mararet_hamilton	Margaret	Hamilton	female
neil_druckman	Neil	Druckmann	male
ellie	ashley	johson	female
joel	troy	baker	male



- وقتی روی **هر سطری** از جدول کلیک شد **اطلاعات کاربر** نمایش داده شود.
  - امکان **حذف کردن کاربر** نیز وجود دارد.
- توجه کنید** که پس از حذف کاربر فایل **users-data.json** بروزرسانی شود.
- نکته:** طراحی کلاینت در **همه بخش ها** دلخواه است.

## Linux (2) بخش

در بخش لینوکس دستورات را داخل یک فایل word قرار دهید.

2-1. یک shell script بنویسید به طوری که نحوه استفاده آن به صورت زیر باشد:

```
$ request.sh <url> <method>
```

- \$ ./request.sh <https://reqres.in/api/users> GET

درخواست <https://reqres.in/api/users> به `get` میشود و آن در فایل <https://reqres.in/api/users> ذخیره میشود.

```
erfan@Erfans-MBP exercise % ./request.sh https://reqres.in/api/users GET
Input: {"name":"morpheus","job":"leader"}
Output: {"page":1,"per_page":6,"total":12,"total_pages":2,"data":[{"id":1,"email":"george.bluth@reqres.in","first_name":"George","last_name":"Bluth","avatar":"https://reqres.in/img/faces/1-image.jpg"}, {"id":2,"email":"janet.weaver@reqres.in","first_name":"Janet","last_name":"Weaver","avatar":"https://reqres.in/img/faces/2-image.jpg"}, {"id":3,"email":"emma.wong@reqres.in","first_name":"Emma","last_name":"Wong","avatar":"https://reqres.in/img/faces/3-image.jpg"}, {"id":4,"email":"eve.holt@reqres.in","first_name":"Eve","last_name":"Holt","avatar":"https://reqres.in/img/faces/4-image.jpg"}, {"id":5,"email":"charles.morris@reqres.in","first_name":"Charles","last_name":"Morris","avatar":"https://reqres.in/img/faces/5-image.jpg"}, {"id":6,"email":"tracey.ramos@reqres.in","first_name":"Tracey","last_name":"Ramos","avatar":"https://reqres.in/img/faces/6-image.jpg"}], "support": {"url": "https://reqres.in/#support-heading", "text": "To keep ReqRes free, contributions towards server costs are appreciated!"}}
```

- \$ ./request.sh <https://reqres.in/api/users> POST

از فایل `input.json` محتوای `request body` خوانده میشود و درخواست `post` به آن در فایل <https://reqres.in/api/users> ذخیره میشود.

```
erfan@Erfans-MBP exercise % ./request.sh https://reqres.in/api/users POST
Input: {"name":"morpheus","job":"leader"}
Output: {"name":"morpheus","job":"leader","id":321,"createdAt":"2023-02-25T17:58:32.061Z"}
```

- \$ ./request.sh
- \$ ./request.sh <https://reqres.in/api/users>

وقتی url یا method مشخص نشده باشد خروجی به صورت زیر خواهد بود.

```
erfan@Erfans-MBP exercise % ./request.sh
Bad params
Usage:
request.sh <url> <method>
erfan@Erfans-MBP exercise % ./request.sh https://reqres.in/api/users
Bad params
Usage:
request.sh <url> <method>
erfan@Erfans-MBP exercise %
```

- \$ ./request.sh <https://reqres.in/api/users> PUT

وقتی متد از نوع get یا post نباشد خروجی به صورت زیر خواهد بود.

```
erfan@Erfans-MBP exercise % ./request.sh https://reqres.in/api/users PUT
Bad method
```

2-2. با استفاده از **محیط CLI** خروجی مد نظر بدست بیاورید.

- محتوای دایرکتوری </usr/share/dict/word> در ترمینال نمایش دهید.
- در مرحله بعد کلمات مورد نظر را lowercase کنید و نمایش دهید.
- در مرحله بعد کلماتی که طول کاراکتر های آنها سه است را نمایش دهید.
- در مرحله بعد کلماتی که حروف "a e i n s" را ندارند، نمایش دهید.
- در مرحله آخر کلماتی که را با "tr" شروع میشوند و حرف آخر آن ها "y" است را نمایش دهید.



## نکات

- مهلت ارسال تمرین تا پایان روز **چهارشنبه 1401/12/10** است. پاسخ تمارین را در **گیت هاب (github)** قرار دهید و **لینک مربوطه را در کارتابل** خود ارسال بفرمایید.
- به ازای هر روز تأخیر در ارسال تمارین ۵ درصد نمره کسر خواهد شد.
- نام **ریپازیتوری** تمرین را به این صورت قرار دهید: Name\_hw14\_maktab88 به عنوان .Mohammad\_Ali\_Kargar\_hw14\_maktab88 مثال:
- ارسال پاسخ تمارینی که در کلاس تحويل داده شده، **الزامی** است.
- در صورت لزوم یک فایل word به عنوان توضیح در کنار کدهای خود قرار دهید.
- در صورتی که سوالی دارید **در گروه راکت چت** بپرسید.