بسم الله الرحمن الرحیم

سید امیرمهدی میرشریفی

۹۸۳۱۱۰۵

پاسخنامه تکلیف یک درس طراحی وب

استاد علیزاده

۱ بخش HTTP

۱.۱ سوال یک

آ- وارد سایت دیجی‌کالا شوید و URL سایت را با استفاده از fragmentation به گونه‌ای تغییر دهید که مستقیما به تصویر enamad موجود در سایت اشاره کند. این نماد در پایین صفحه و به شکل زیر است:

پاسخ:

با استفاده از لینک : https://www.digikala.com/#sScdOJOzhFxtcEqkjP7P میتوان به قسمت مورد نظر انتقال یافت . بعد از علامت # آی دی عکس مورد نظر قرار دارد تا مرورگر با استفاده از آن به المنت مورد نظر در صفحه برسد.

ب - به ریپازیتوری React وارد شوید. URL را به گونه‌ای تغییر دهید که پاراگراف زیر در README هایلایت شود.

Contributing Guide

Read our contributing guide to learn about our development process, how to propose bugfixes and improvements, and how to build and test your changes to React.

Good First Issues

To help you get your feet wet and get you familiar with our contribution process, we have a list of good first issues that contain bugs that have a relatively limited scope. This is a great place to get started.

License

React is MIT licensed.

پاسخ:

با کپی کردن لینک زیر میتوان به تصویر بالا دست یافت. همانطور که در لینک مشاهده می کنید پس از علامت شارپ آی دی المنت مورد نظر (good first issues) را از inspect به دست آورده و اضافه میکنیم و سپس با استفاده از :~:text=first , last بازه متنی که می خواهیم هایلایت شود را مشخص کنیم.

https://github.com/facebook/react/#user-content-good-first-issues:~:text=Good%20First%20Issues,to%20get%20started

۲.۱ سوال دو

در این سوال قصد داریم با استفاده از ابزار خط فرمان curl درخواست HTTP ارسال کنیم. دستور اجرا شده در هر قسمت و نتیجه اجرای آن را به صورت اسکرین شات نمایش داده و به پرسش‌های مطرح شده پاسخ دهید.

آ- یک درخواست get به دامنه http://google.com ارسال کنید به طوری هدرهای پاسخ را در کنسول مشاهده کنید. کد وضعیت پاسخ چیست و به چه معناست؟ هدر Location چه چیز را مشخص می‌کند؟

پاسخ:

تصویر زیر دستور مربوطه و نتیجه را نمایش میدهد که بخش اول آن هدرهای ریکوئیست و بخش دوم هدر های ریسپانس است:

```
C:\Users\sam>curl -v "http://google.com"
*   Trying 216.239.38.120:80...
* Connected to google.com (216.239.38.120) port 80 (#0)
> GET / HTTP/1.1
> Host: google.com
> User-Agent: curl/7.83.1
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 301 Moved Permanently
< Location: http://www.google.com/
< Content-Type: text/html; charset=UTF-8
< Date: Fri, 17 Mar 2023 15:57:28 GMT
< Expires: Sun, 16 Apr 2023 15:57:28 GMT
< Cache-Control: public, max-age=2592000
< Server: gws
< Content-Length: 219
< X-XSS-Protection: 0
< X-Frame-Options: SAMEORIGIN
<
```

همانطور که در بخش هدرهای ریسپانس مشاهده میشود ، وضعیت با کد ۳۰۱ مشخص شده است که بدین معنا است که منبع درخواستی به محل دیگری مشخص شده است که آدرس جدید آن را میتوان در Location مشاهده کرد.

ب - یک درخواست get به دامنه http://www.google.com ارسال کنید به طوری که کوکی های ست شده توسط این دامنه را در کنسول مشاهده کنید. نام کوکی‌های دریافتی را گزارش کنید.

پاسخ: در ادامه لیستی از کوکی های دریافتی ارائه میشود. همچنین در مقابل هرکدام در صورت نیاز لینکی قرار می گیرد که توضیح مختصری را از آنها ارائه می دهد.

- 1P_JAR (https://replyify.com/cookie-policy.html#:~:text=Purpose-,1P_JAR,-google.com)
- Expires
- Domain
- Samesite(https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie/SameSite#:~:text=See%20also-,SameSite%20cookies,-Secure%20context%3A)
- NID (https://cookiedatabase.org/cookie/google-ads-optimization/nid/#:~:text=With%20Cookiedatabase.Org-,NID,-Summary)

```
< Set-Cookie: 1P_JAR=2023-03-17-20; expires=Sun, 16-Apr-2023 20:04:27 GMT; path=/; domain=.google.com; Secure
< Set-Cookie: AEC=ARSKqsKuo1dL2ZbdzHPHTFILqsvl0xBP7pDjlQqb8IbAvitkgxOpe6Nucg; expires=Wed, 13-Sep-2023 20:04:27 GMT; pat
h=/; domain=.google.com; Secure; HttpOnly; SameSite=lax
< Set-Cookie: NID=511=G98bGuW92Z4h6R6RO4MVRSX8tLIP85NTEgoe1jqJizKuGYrvWu2D_yGvtqJt9V_l8HotMztRvwY0Sa6Erjtms3crsHU4DI3sI3
CBpOjvgRdyFVM-HnpKikMbA12NN76n1KQinl-Smg7xc5w6YZgcy6GYS2V3OdXPg5LRjQ2BWqg; expires=Sat, 16-Sep-2023 20:04:27 GMT; path=/
; domain=.google.com; HttpOnly
```

پ - یک درخواست post به دامنه https://reqres.in/api/users با بدنه JSON زیر ارسال کرده و پاسخ را گزارش کنید.

```
{
    "first_name": "YOUR_FIRST_NAME",
    "last_name": " YOUR_LAST_NAME "
}
```

پاسخ:

برای ارسال فایل جیسون باید به شکل زیر عمل کنیم:

```
>curl -X POST https://reqres.in/api/users
 -H "Content-type:application/json"
 -d "{\"first_name\":\"amirmahdi\",\"last_name\":\"mirsharifi\"}"
```

و پاسخ دریافت شده به شکل زیر است:

```
{"first_name":"amirmahdi","last_name":"mirsharifi","id":"917","createdAt":"2023-03-17T20:48:26.275Z"}
```

ت - امتیازی. یک دامنه تحریم شده مانند https://developer.android.com را انتخاب کرده و یک بار بدون پروکسی و بار دیگر با تنظیم پروکسی درخواستی به آن ارسال کنید و پاسخ‌های دریافتی را مقایسه کنید.

پاسخ: در ابتدا با استفاده از دستور کرل که به این دامنه پیام بزنیم با وضعیت ۴۰۳ که به معنای ممنوع است مواجه میشویم. در این زمان اگر دستور curl https://developer.android.com - - proxy fodev.org:8118:

که proxy– نشانگر استفاده از پراکسی سرور و دامنه و پورتی که در ادامه آمده است نشانگر سرور مدنظر است استفاده کنیم وضعیت ۲۰۰ را دریافت میکنیم. زیرا در مرحله قبلی دامنه تحریم شده بود و به آیدی ما خدمات نمیداد اما پراکسی این خدمات را در اختیار ما قرار داد.

۳.۱ سوال سه

یک کلاینت سادهی HTTP با شرایط زیر پیاده کنید (در استفاده از هر زبان یا کتابخانهای آزاد هستید):
یک آدرس HTTP در ورودی دریافت کنید. اگر آدرس وارد شده معتبر بود، برنامه یکی از متدهای زیر را به عنوان ورودی میگیرد.

```
GET / POST / PATCH/ PUT / DELETE
```

پاسخ:

```python
def url_validator(url):
    regex = re.compile(
        r'^(?:http|ftp)s?://'
        r'(?:(?:[A-Z0-9](?:[A-Z0-9-]{0,61}[A-Z0-9])?\.)+(?:[A-Z]{2,6}\.?|[A-Z0-9-]{2,}\.?)|'
        r'localhost|'
        r'\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})'
        r'(?::\d+)?'
        r'(?:/?|[/?]\S+)$', re.IGNORECASE)

    return re.match(regex, url) is not None
```

با تابع url_validator میتوان کد داده شده را اعتبار سنجی کرد تا در موقع ارسال درخواست با ارور روبه رو نشویم. در ادامه url & method را از کاربر میگیریم و اگر url or method نامعتبر باشد آنگاه از برنامه خارج میشویم. سپس اطلاعات لازم مثل نوع فایل دریافتی و کد و وضعیت را استخراج میکنیم.

```python
url = input("Enter URL:")
method = input("Enter request method:")

if not url_validator(url):
    print("URL is invalid")
    exit()
if not method.lower() in ["get", "put", "post", "patch", "delete"]:
    print("method is invalid")
    exit()
response = requests.request(method=method, url=url)
content_type = response.headers.get("content-type").split(";")[0]
status = response.status_code
```

در آخر با توجه به حالات مختلف اعمال مختلف را اعمال میکنیم.برای مثال تفاوت در ذخیره سازی انواع فایل ها یا پیامی که در مواجه با ارور های مختلف نشان میدهد.همچنین ارور ها صرفا بر اساس نوع که همان عدد شروع شونده آن است تخفیف میشوند یعنی :2xx,3xx,4xx,5xx

```python
if str(status).startswith("2"):
    if content_type == "text/html":
        with open('./page.html', 'wb+') as f:
            f.write(response.content)
    elif content_type == "application/json":
        with open('./file.json', 'wb+') as f:
            f.write(response.content)
    else:
        print(response.content)
elif str(status).startswith("3"):
    print("Resource has been moved.")
elif str(status).startswith("4"):
    print("Client error ")
elif str(status).startswith("5"):
    print("Server error ")
```

۲ بخش Git

بخش اول – amend

برای اصلاح کامیت آخر و اضافه کردن فایل bar.txt

با استفاده از فرمان

```
git add bar. txt
```

فایل bar.txt را stage کرده و سپس کامیت آخر (به همراه پیام آن) را با استفاده از فرمان

```
git commit --amend
```

اصلاح و آپدیت می‌کنیم.

با استفاده از

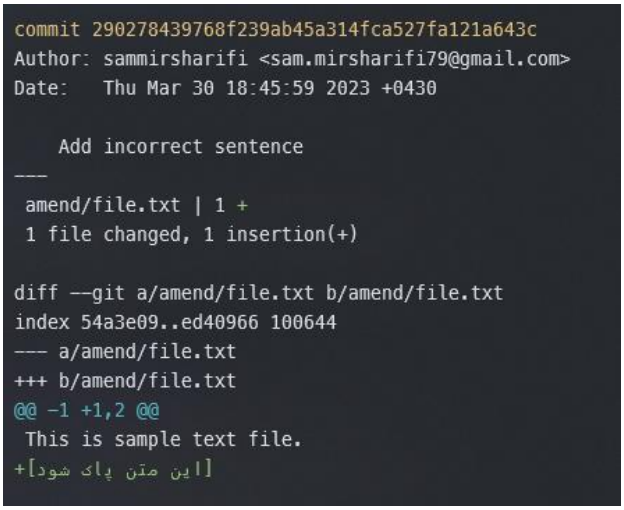
```
git log --stat
```

می‌توان مشاهده کرد که فایل bar.txt به کامیت اضافه شده است. (پیام کامیت نیز قابل مشاهده خواهد بود)

برای اصلاح کامیت قبل‌تر با استفاده از فرمان

```
git log --stat -p
```

و طبق تصویر پایین می‌بینیم که در کامیت محتوای اشتباهی وارد فایل شده که لازم است آنرا پاک کنیم.



برای اینکار از git rebase به شکل تعاملی (interactive) استفاده می‌کنیم.

فرمان

git rebase -i '290278439768f239ab45a314fca527fa121a643c^'

را اجرا کرده و rebase را از گره پدر این کامیت آغاز می‌کنیم.

پس از اجرای فرمان در ادیتور، فرمانِ اولین کامیت را از pick به edit تغییر می‌دهیم و ذخیره کرده و از ادیتور خارج می‌شویم. با اینکار عملیات rebase به شکل ناتمام (سر کامیتی که edit قرار دادیم) باقی‌مانده تا فرمان rebase --continue اجرا شود.

در این لحظه آخرین کامیت همان کامیت اشتباه می‌باشد بنابراین مانند بخش اول ابتدا فایل‌های اصلاح شده را وارد مرحله staging کرده و سپس amend می‌کنیم.

به ترتیب

- edit file.txt
- git add -A file.txt
- git commit --amend --allow-empty --no-edit
با توجه به اینکه با حذف خط اشتباه، دیگر کامیت خالی از تغییرات خواهد بود، برای مجاز کردن این فرایند فلگ --allow-empty اضافه شده است
- git rebase --continue
با اینکار عملیات rebase ادامه می‌یابد ولی در ادامه و در کامیت‌های بعدی کانفلیکت ایجاد می‌شود و نیاز است کانفلیکت را برطرف کرده و دوباره rebse --continue را اجرا کنیم.

- در نهایت با استفاده از git push -f origin amend تغییرات را روی ریپازیتوری خود قرار می‌دهیم. استفاده از f- یا force-- به این علت است که پس از rebase برنچ ریموت قابلیت fast-forward به کامیت آخر را نخواهد داشت (مثلا در صورتی که کامیت بخش قبلی را push کرده باشیم)

با توجه به اینکه گفته می‌شود بعد از تغییرات ابتدا git commit کنیم و سپس rebase --continue را اجرا کنیم، در این صورت به نظر کامیت خالی حذف می‌شود. برای جلوگیری از این اتفاق، بدون git commit اینکار را انجام می‌دهیم و git rebase خود این کامیت را انجام می‌دهد. به نظر این اتفاق مربوط به حالت interactive می‌باشد.
در نهایت از روش دوم استفاده شد و پس از stage کردن تغییرات مستقیما rebase --continue صدا زده شد.

۲.۳ بخش دوم – conflict merge

برای explicit-merge از وارد برنچ merge-explicit شده و سپس فرمان

git merge --no-ff merge-conflict

را اجرا می‌کنیم، conflictها را برطرف کرده و سپس کامیت می‌کنیم. درخت برنچ به شکل زیر می‌شود.

```
└>» git log --graph --oneline merge-explicit
*   6b17228 (origin/merge-explicit, merge-explicit) Merge branch 'merge-conflict' into merge-explicit
|\
| * 43cda63 (HEAD -> merge-conflict) Add an equation
* | db91237 Add an equation
|/
* 005a3b4 (origin/merge-conflict) Add setup
* c4bd85e Fix question
* a48ea53 Add exercise readme
* f5ed9c8 Add amend
* 5088c6a Add merge-conflict
* 15d462e first commit
```

همانطور که می‌بینیم برای مرج، یک کامیت جدید ایجاد شده است.

حال برای implicit-merge وارد برنچ merge-implicit شده و سپس فرمان

git rebase merge-conflict

را اجرا می‌کنیم. کانفلیکت‌ها را برطرف کرده، stage و git rebase --continue

را اجرا می‌کنیم. درخت به شکل زیر خواهد بود که یعنی کامیت‌های برنچ ما در ادامه کامیت‌های merge-conflict اضافه شده‌اند (merge-conflict به عنوان base برنچ فعلی قرار گرفته است).