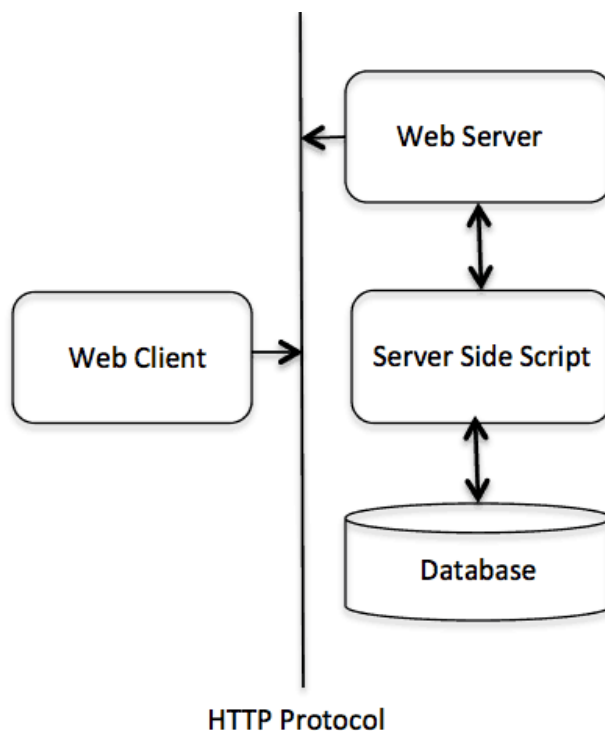


بخش اول - Web & HTTP

- سوال ۱

پروتکل HTTP وظیفه برقراری ارتباط میان سرویس دهنده یا همان سرور و سرویس گیرنده یا همان کلاینت را فراهم می‌کند. پروتکل HTTP در لغت به معنی پروتکل برای انتقال ابرمتن‌ها می‌باشد که منظور از ابرمتن‌ها، متن‌های بزرگ و طولانی است که به صورت HTML نوشته می‌شوند.

به منظور ارسال و دریافت اطلاعات، از Handshaking استفاده می‌شود که برای شروع و پایان عملیات تبادل اطلاعات بین سرور و کلاینت، چندین درخواست و پاسخ رد و بدل می‌شود. پورت ۸۰ به منظور برقراری ارتباط مورد استفاده قرار می‌گیرد.



سرتیترهای HTTP یا همان هدرها به کلاینت و سرور اجازه می‌دهند تا اطلاعات اضافه‌تری انتقال پیدا کند. به صورت کلی هدرها به چهار نوع تقسیم می‌شوند.



- هدرهای عمومی که هم برای درخواست و هم پاسخ مورد استفاده قرار می‌گیرند.
 - هدرهای درخواستی که بیشتر در مورد fetch یا درخواست کلاینت‌ها هستند.
 - هدرهای پاسخ که بیشتر به پاسخ سرور اشاره می‌کنند.
 - هدرهای اطلاعاتی که درباره اطلاعات و جزئیات مربوط به داده هستند، مثل طول داده‌ها و روش کدینگ مورد استفاده.
- برای جزئیات بیشتر دسته بندی‌ها می‌توانیم به [لینک](#) مراجعه کنیم.
- کدهای وضعیت یا همان کدهای استاتوس، پاسخ سرور به درخواست از طرف کلاینت است. این کدها شامل سه رقم است که رقم اول کلاس پاسخ و دو رقم دیگر پاسخ را به صورت یکتا مشخص می‌کند. برای رقم اول پنج وضعیت پیش می‌آید.
- رقم ۱ یا Informational در حالتی است که درخواست دریافت شده و پردازش در حال انجام است.
 - رقم ۲ یا Success در حالتی است که درخواست به صورت صحیح، دریافت، درک و پذیرفته شده است.
 - رقم ۳ یا Redirection در حالتی است که برای تکمیل درخواست، رویدادهای دیگری نیز باید رخ دهد.
 - رقم ۴ یا Client Error در حالتی است که خطای سینتکس رخ داده و به هر صورت درخواست غیرقابل انجام است و مشکل از سمت Client است.
 - رقم ۵ یا Server Error در حالتی است که یک درخواست صحیح، از سمت سرور انجام نشده باشد و مشکل از سمت Server است.

HTTP Status Codes



برای کدهای خاص می‌توان به جداول مربوطه مراجعه کرد که در [لینک](#) آمده است.



– سوال ۲

بخش Inspect مربوط به سامانه آموزش دانشگاه به صورت زیر است.

```
▼ General
Request URL: https://edu.sharif.edu/
Request Method: GET
Status Code: 200
Remote Address: 81.31.168.91:443
Referrer Policy: strict-origin-when-cross-origin

▼ Response Headers
content-type: text/html; charset=utf-8
date: Fri, 28 Jan 2022 18:49:56 GMT
server: nginx/1.10.3 (Ubuntu)
x-content-encoding-over-network: gzip

▼ Request Headers
:authority: edu.sharif.edu
:method: GET
:path: /
:scheme: https
accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
accept-encoding: gzip, deflate, br
accept-language: en-US,en;q=0.9
cache-control: max-age=0
cookie: JSESSIONID=NrCiAWWzMc4hKqndWcUx6zx2eTID1ahdhDSXMEH1Qhb4EiEr8t!104269698
sec-ch-ua: "Not;A Brand";v="99", "Google Chrome";v="97", "Chromium";v="97"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
sec-fetch-dest: document
sec-fetch-mode: navigate
sec-fetch-site: none
sec-fetch-user: ?1
upgrade-insecure-requests: 1
```

در ادامه برخی از هدرها بررسی می‌شود.

Accept

The *Accept* request-header field can be used to specify certain media types which are acceptable for the response. The general syntax is as follows:

```
Accept: type/subtype [q=qvalue]
```



Accept-Encoding

The *Accept-Encoding* request-header field is similar to Accept, but restricts the content-codings that are acceptable in the response. The general syntax is:

```
Accept-Encoding: encoding types
```

Accept-Language

The *Accept-Language* request-header field is similar to Accept, but restricts the set of natural languages that are preferred as a response to the request. The general syntax is:

```
Accept-Language: language [q=qvalue]
```

Authorization

The *Authorization* request-header field value consists of credentials containing the authentication information of the user agent for the realm of the resource being requested. The general syntax is:

```
Authorization : credentials
```

Cookie

The *Cookie* request-header field value contains a name/value pair of information stored for that URL. Following is the general syntax:

```
Cookie: name=value
```



Server

The *Server* response-header field contains information about the software used by the origin server to handle the request. The general syntax is:

```
Server : product | comment
```

A desktop browser would usually send requests with the following header:

```
Sec-CH-UA-Mobile: ?0
```



A browser on a mobile device would usually send requests with the following header:

```
Sec-CH-UA-Mobile: ?1
```



navigate

The request is initiated by navigation between HTML documents.



Sec-CH-UA: "<brand>";v="<significant version>", ...



Directives

<brand>

A brand associated with the user agent, like "Chromium", "Google Chrome", or an intentionally incorrect brand like "Not A;Brand".

<significant version>

The "marketing" version number associated with distinguishable web-exposed features.

Remote Address: 93.184.216.34:80

The IP address of the server

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_4)

Lets server to identify the characteristics of the application, OS, vendor, and versions

Cache-Control: max-age=604800

Set the instruction for caching. Other setting types: no-cache, no-store, no-transform

Content-Encoding: gzip

Specifies the compression algorithm used for the response body

روش استفاده برای HTTP از نوع GET است



GET

The GET method is used to retrieve information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data.

استاتوس کد

Message	Description
200 OK	The request is OK.



گزارش آزمایش ۴

- سوال ۳

Overview GET edu.sharif.edu + ... No Environment

edu.sharif.edu Save

GET edu.sharif.edu Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (6) Test Results Status: 503 Service Temporarily Unavailable Time: 218 ms Size: 2.84 KB Save Response

KEY	VALUE
Server ⓘ	nginx/1.10.3 (Ubuntu)
Date ⓘ	Fri, 28 Jan 2022 19:27:58 GMT
Content-Type ⓘ	text/html
Content-Length ⓘ	2701
Connection ⓘ	keep-alive
ETag ⓘ	"6138a527-a8d"



– سوال ۴، بخش اول

https://jsonplaceholder.typicode.com/posts

GET https://jsonplaceholder.typicode.com/posts Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (25) Test Results Status: 200 OK Time: 756 ms Size: 27.94 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "userId": 1,
3   "id": 1,
4   "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
5   "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"
6 }
7 ,
8 {
9   "userId": 1,
10  "id": 2,
11  "title": "qui est esse",
```

– سوال ۴، بخش دوم

https://jsonplaceholder.typicode.com/posts?id=1

GET https://jsonplaceholder.typicode.com/posts?id=1 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> id	1			
Key	Value	Description		

Body Cookies Headers (24) Test Results Status: 200 OK Time: 351 ms Size: 1.36 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "userId": 1,
3   "id": 1,
4   "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
5   "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"
6 }
7
8 }
```



گزارش آزمایش ۴

– سوال ۴، بخش سوم

https://jsonplaceholder.typicode.com/posts

POST https://jsonplaceholder.typicode.com/posts

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "userId": 1,
3   "title": "hello",
4   "body": "Hi"
5 }
```

Body Cookies Headers (25) Test Results Status: 201 Created Time: 383 ms Size: 1.21 KB Save Response

Pretty Raw Preview Visualize

```
{
  "userId": 1,
  "title": "hello",
  "body": "Hi",
  "id": 101
}
```

– سوال ۴، بخش چهارم

https://jsonplaceholder.typicode.com/posts/1

PUT https://jsonplaceholder.typicode.com/posts/1

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "userId": 1,
3   "title": "hello",
4   "body": "Hi"
5 }
```

Body Cookies Headers (24) Test Results Status: 200 OK Time: 375 ms Size: 1.11 KB Save Response

Pretty Raw Preview Visualize

```
{
  "userId": 1,
  "title": "hello",
  "body": "Hi",
  "id": 1
}
```



گزارش آزمایش ۴

– سوال ۴، بخش پنجم

GET <https://jsonplaceholder.typicode.com/posts?userId=1> [Send](#)

Params Authorization Headers (8) Body Pre-request Script Tests Settings [Cookies](#)

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	userid	1			
	Key	Value	Description		

Body Cookies Headers (25) Test Results [Status: 200 OK](#) [Time: 417 ms](#) [Size: 3.73 KB](#) [Save Response](#)

[Pretty](#) [Raw](#) [Preview](#) [Visualize](#)

```
{
  "userId": 1,
  "id": 1,
  "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
  "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"
},
{
  "userId": 1,
  "id": 2,
  "title": "qui est esse",
  "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nfugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\nqui aperiam non debitis possimus qui neque nisi nulla"
},
{
  "userId": 1,
  "id": 3,
  "title": "ea molestias quasi exercitationem repellat qui ipsa sit aut",
  "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nfugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\nqui aperiam non debitis possimus qui neque nisi nulla"
}
```



بخش دوم – Introduction to Django

ابتدا از دستور زیر برای ساخت پروژه استفاده می‌کنیم.

```
(Lab4_Section2) C:\Users\Amirehsan\PycharmProjects\Lab4_Section2>django-admin startproject mysite
```

پس از تغییر دایرکتوری، از دستور زیر استفاده می‌کنیم.

```
(Lab4_Section2) C:\Users\Amirehsan\PycharmProjects\Lab4_Section2\mysite>python manage.py runserver
```

بنابراین اولین صفحه ایجاد شد.



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

حال به سراغ ساخت app می‌رویم.

```
C:\Users\Amirehsan\PycharmProjects\Lab4_Section2\mysite>python manage.py startapp test_app
```

در اینجا First View را ایجاد می‌کنیم. برای این کار در دایرکتوری `views`، به ریکوئیسٹ `HTTPS` اینگونه پاسخ می‌دهیم.

```
manage.py × views.py ×
from django.http import HttpResponse

def index(request):
    return HttpResponse("Hello this is a test app")
```

سپس در دایرکتوری اپ، فایل جدیدی با نام `urls` ایجاد می‌کنیم که به `url` های مربوط به خود پاسخ دهد.



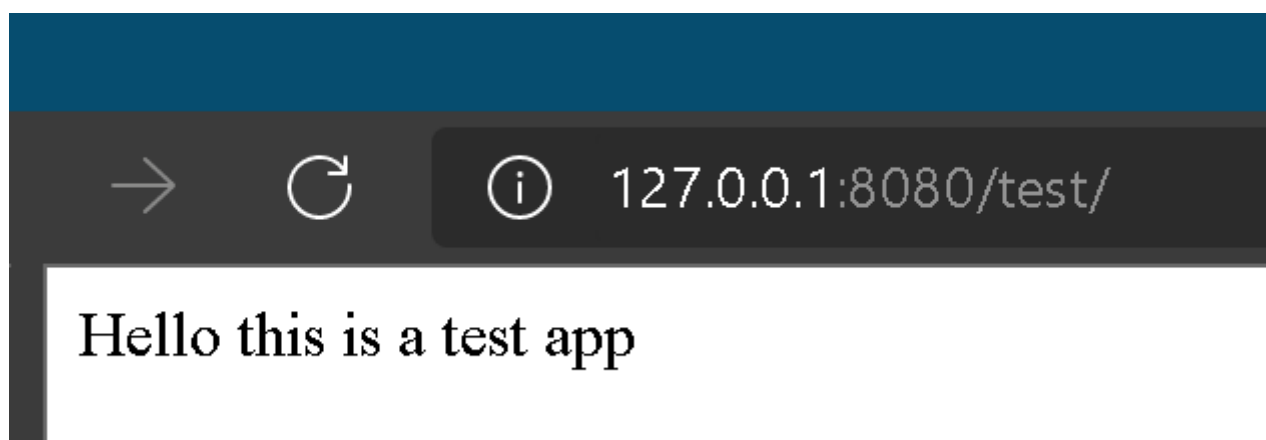
```
age.py × views.py × urls.py ×  
from django.urls import path  
  
from . import views  
  
urlpatterns = [  
    path('', views.index, name='index'),  
]
```

در بخش url مربوط به پروژه نیز تنظیم مربوط به اپلیکیشن test را وارد می‌کنیم.

```
from django.contrib import admin  
from django.urls import include, path  
  
urlpatterns = [  
    path('test/', include('test_app.urls')),  
    path('admin/', admin.site.urls),  
]
```

The **include()** function allows referencing other URLconfs. Whenever Django encounters **include()**, it chops off whatever part of the URL matched up to that point and sends the remaining string to the included URLconf for further processing.

```
C:\Users\Amirehsan\PycharmProjects\Lab4_Section2\mysite>python manage.py runserver:8080
```





بخش سوم – Introduction to Database

– سوال ۱

دیتابیس پیش فرض جنگو، SQLite است. برای اتصال دیتابیس دلخواه نیز نیاز است تا آن را نصب کرده و سپس تنظیمات مربوطه را تغییر دهیم.

If you wish to use another database, install the appropriate database bindings and change the following keys in the DATABASES 'default' item to match your database connection settings:

- ENGINE – Either `'django.db.backends.sqlite3'`, `'django.db.backends.postgresql'`, `'django.db.backends.mysql'`, or `'django.db.backends.oracle'`. Other backends are also available.
- NAME – The name of your database. If you're using SQLite, the database will be a file on your computer; in that case, NAME should be the full absolute path, including filename, of that file. The default value, `os.path.join(BASE_DIR, 'db.sqlite3')`, will store the file in your project directory.

If you are not using SQLite as your database, additional settings such as USER, PASSWORD, and HOST must be added. For more details, see the reference documentation for DATABASES.

حال با استفاده از دستور اشاره شده، جداول مربوط به اپ‌های موجود را ایجاد می‌کنیم.

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'test_app.apps'  
]
```

```
C:\Users\Amirehsan\PycharmProjects\Lab4_Section2\mysite>python manage.py migrate
```

– سوال امتیازی

ORM یا Object-Relational Mapping روشی است که برنامه نویس به جای اینکه در برنامه، مستقیم روی دیتابیس کوئری اجرا کند این کار را از طریق فریم ورک انجام می‌دهد. یعنی جنگو یک سری توابع دارد که به واسطه آن می‌توانیم درخواست‌های خودمان را به دیتابیس ارسال و جواب را دریافت کنیم. بنابراین دیگر در برنامه خبری از کوئری‌های دیتابیس مثل `select * from` نیست. فواید زیادی برای ORM وجود دارد.



۱. امنیت بسیار بالا: با استفاده از ORM حملاتی مثل SQL injection عملاً منتفی خواهد بود.
۲. سرعت بسیار بالاتر: توابع بهینه سازی شده و برای هر منظور مشخص است که قرار است از کدام تابع استفاده شود. البته همیشه امکان استفاده از ORM ها وجود نداشته و اگر کوئری شما قرار است کار پیچیده و خاصی را انجام دهد می بایست این کوئری را به صورت دستی ایجاد و اجرا کنیم.

- سوال ۲

با استفاده از SQLite می توان به دیتابیس متصل شد. با استفاده از دستور table می توان لیست table ها را مشاهده کرد.

```
SQLite version 3.8.6 2014-08-15 11:46:33
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .table
auth_group                                auth_user_user_permissions
auth_group_permissions                    django_admin_log
auth_permission                           django_content_type
auth_user                                 django_migrations
auth_user_groups                          django_session
sqlite>
```

- سوال ۳

ایجاد کلید اصلی به ما کمک می کند تا بر اساس خاصیت یکتا بودن مقادیر یک ستون یا ترکیبی از ستون های مشخص، به طور دقیق یک ردیف خاص را در جدول پیدا و بازیابی کنیم.

```
sqlite> create table teammates_info(
...> ID INT NOT NULL PRIMARY KEY,
...> LastName TEXT NOT NULL,
...> FirstName TEXT,
...> Email TEXT,
...> DateJoined TEXT,
...> DateUpdated TEXT
...> );
```



```
sqlite> select * from teammates_info
...> ;
sqlite> INSERT INFO teammates_info (name,lastname,email,std_num,date_joined,date_update)
...> VALUES("Amirehsan","Alizadeh","Gmail",96109844,"DEC28","JAN28");
Error: near "INFO": syntax error
sqlite> INSERT INTO teammates_info (name,lastname,email,std_num,date_joined,date_update)
...> VALUES("Amirehsan","Alizadeh","Gmail",96109844,"DEC28","JAN28");
sqlite> select * from teammates_info
...> ;
Amirehsan|Alizadeh|Gmail|96109844|DEC28|JAN28
sqlite> INSERT INTO teammates_info (name,lastname,email,std_num,date_joined,date_update)
...> VALUES("Amirehsan","Alizadeh","Gmail",96209844,"DEC28","JAN28");
sqlite> INSERT INTO teammates_info (name,lastname,email,std_num,date_joined,date_update)
...> VALUES("Amirehsan","Alizadeh","Gmail",96309844,"DEC28","JAN28");
sqlite> INSERT INTO teammates_info (name,lastname,email,std_num,date_joined,date_update)
...> VALUES("Amirehsan","Alizadeh","Gmail",96409844,"DEC28","JAN28");
sqlite> select * from teammates_info
...> ;
Amirehsan|Alizadeh|Gmail|96109844|DEC28|JAN28
Amirehsan|Alizadeh|Gmail|96209844|DEC28|JAN28
Amirehsan|Alizadeh|Gmail|96309844|DEC28|JAN28
Amirehsan|Alizadeh|Gmail|96409844|DEC28|JAN28
sqlite>
```

– سوال ۴

داخل DB Browser این کار را انجام می‌دهیم.

Database Structure	Browse Data	Edit Pragmas	Execute SQL			
Table: teammates_info_GUI						
	name	lastname	email	std_num	date_joined	date_update
	Filter	Filter	Filter	Filter	Filter	Filter
1	amirehsan	alizadeh	gmail	96109844	dec28	jan28
2	sina	bolouki	gmail	96209844	dec28	jan28
3	amirsina	alizbolo	yahoo	96309844	dec28	jan28
4	sinamir	bolotaliz	ee.sharifmail.edu	96409844	dec28	jan28