

BACKEND ASSIGNMENT

Research

E Abhishek

190070022

eabhishek001@gmial.com

How network connection works?

IP address, Domain Name and URL a basic view

Each website has a domain name like 'google.com'. The contents of a webpage are stored in a server. A server is in simple terms a computer that is always connected to the network. The concept of website can be explained in terms of our ED lab servers. Each website has a space allocated in a server like we had in our ED lab. The contents of the webpage are stored in the server like we store our drawings in the Drawing hall server. When we want to access a website, we need to access the server i.e., our computer must connect to the server or in simple terms talk to the server. For this connection computers rely on numbers and letters called an IP address. Every device that connects to the Internet has a unique IP address and looks something like this:

22.231.113.64 or 3ffe:1900:4545:3:200:f8ff:fe21:67cf

In order to navigate easily around the web, typing in a long IP address isn't ideal, or realistic, to an online user. This is the reason why domain names were created – to hide IP addresses with something more memorable. You could consider the domain name as a “nickname” to the IP address.

A URL incorporates the domain name, along with other detailed information, to create a complete address (or “web address”) to direct a browser to a specific page online called a web page. In simple terms it's a set of directions and every web page has a unique one.

<https://portal.iitb.ac.in/asc/Login>

<https://portal.iitb.ac.in/asc/Login> is URL (gives the complete path)

portal.iitb.ac.in is Domain Name

How are URL's mapped to website hosted on a server?

Domain Name System (DNS): Contains the information about which domain name is mapped to which IP address. When a URL such as www.iitb.ac.in is typed in a web browser the browser requests the DNS and gets the information of the IP address (like 123.312.121.112) of the server in which the web page is stored. Then the web browser sends a request to the server to send the web page. The server sends the webpage back to the browser and it is displayed to us.

What is the entire cycle of events that follows when you type in the URL of a webpage?

1. You enter a URL into a web browser
2. The browser looks up the IP address for the domain name via DNS
3. The browser sends a HTTP *request* to the server
4. The server sends back a HTTP *response*
5. The browser begins rendering the HTML
6. The browser sends requests for additional objects embedded in HTML (images, CSS, JavaScript) and repeats steps 3-5.
7. Once the page is loaded, the browser sends further requests as needed.

What is TCP and UDP?

Transmission Control Protocol (TCP)

TCP takes messages from an application and divides them into packets, which can then be forwarded by the devices in the network – switches, routers, security gateways – to the destination. Each packet is assigned with the address of the source of the packet and the address of the destination of the packet. Because it is connection-oriented, it ensures a connection is established and maintained until the exchange between the application sending and receiving the message is complete.

User Datagram Protocol (UDP)

User Datagram Protocol (UDP) is a Transport Layer protocol. Unlike TCP, it is **unreliable and connectionless protocol**. So, there is no need to establish connection prior to data transfer. Though Transmission Control Protocol (TCP) is the dominant transport layer protocol used with most of Internet services; provides assured delivery, reliability and much more but all these services cost us with additional overhead and delay. Here, UDP comes into picture. For the **real time services** like computer gaming, voice or video communication, live conferences; we need UDP. Since high performance is needed, UDP permits packets to be dropped instead of processing delayed packets. There is no error checking in UDP, so it also saves bandwidth.

What does setting of server mean?

A server is a computer which is always connected to a network. A server can even be our personal laptop. To set up a server we need networking software and web server software. Networking software handles the networking protocols such as TCP and UDP and web server software responds to the requests of the clients. The web server software responds to protocols such as HTTPS and other such request protocols. The web server software must be informed about the basic settings - default directory or folder, whether to allow visitors to see the contents of a directory or folder, where to store the log file, etc.

How does a server resolve a request?

The job of resolving a request to a server is done by the web server software.

A web server's job is basically to accept requests from clients and send responses to those requests. The URL contains the information about which files to be accessed in the server. A web server gets a URL, translates it to a filename (for static requests), and sends that file back over the internet from the local disk, or it translates it to a program name (for dynamic requests), executes it, and then sends the output of that program back over the internet to the requesting party. If for any reason, the web server was not able to process and complete the request, it instead returns an error message.

So basically, a web server is the software that receives your request to access a web page. It runs a few security checks on your HTTP request and takes you to the web page. Depending on the page you have requested, the page may ask the server to run a few extra modules while generating the document to serve you. It then serves you the document you requested.

Apache and Nginx are two such web server software's.

Apache is an open-source cross-platform web server which is also known as "httpd" and Apache.

Nginx is a web server which can also be used as a reverse proxy, load balancer, mail proxy and HTTP cache. The software was first publicly released in 2004.

The main difference between Apache and NGINX lies in their design architecture. Apache uses a process-driven approach and creates a new thread for each request. Whereas NGINX uses an event-driven architecture to handle multiple requests within one thread.

Q) Suppose you built a html page for the first time, and you are very enthusiastic to show it to your friend on his device. How would you do this if you are not allowed to share your code with him or upload your website.

I will store the html page in a folder and install a web server software such as Apache. I will set the folder with html page as the directory which can be accessed by using the IP address of my laptop. When my friend types the IP address of my laptop in his web browser he will be directed to this directory and he can access the contents in the

directory. In this way the files in the network created using my laptop as server could be accessed by my friend without transferring the files.

Django

It is very difficult for every web designer to develop and maintain the backend database by himself. So, we use some external software's like Django.

Django is a high-level Python Web framework that encourages rapid development and easy design. Built by experienced developers, it takes care of much of the hassle of Web development, so we can focus on writing your app without needing to do whole backend work by ourselves.

Normally we need a database to manage data we get from the user. To communicate with the database standard query language (SQL) is used. We need to learn SQL queries about how to extract data from the database and use those to extract data and send it to our front end which might work in html. By using Django we need not do all these things, Django maintains the database and it works on python programming language. So, by using Django we can type the queries in python and Django will convert them into SQL queries and get the contents from the database and give us the output. In this way Django makes our lives simple to work with database and their queries.

Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Suppose you want to now design the database schema of Book-ed!. This should include the name of the book, author, ISBN number, short description, and any other field you deem necessary. Similarly, you'd have to design the scheme for the users and define the relationship between a book(s) and a user(s). Assume all the data that you need is already available.

Database schema with 2 table's, one with seller information and other with borrower information. The two tables are linked by the name of the book column because this is the most reliable information which the borrower might have about the book.

Seller

ISBN number (primary key)	Int
Book Name	char
Author	Char
Publisher	Char
Genre	Char
Short Description	Char
Name of seller	Char
Contact details of seller	int [10]

Borrower

Name of book	Char
Author	Char
Genre	Char
Name of borrower	Char
Contact details	int [10]

You want to introduce the feature that the books are searchable by the author? What will be the best way to accommodate these. Keep in mind there is additional load on the server when you are querying the tables to get the information.

- a. You want to minimize the number of queries. The simplest way is to fetch all the books and match the author for each of them. Can you do any better?

We can sort the seller information in terms of the author name i.e., we can group the sellers who have books of an author. In this way we need to check the author name of a group only once. This can be done by creating a new table(class) which has columns for author's and in each column of an author there will be information of the seller who is having the book. This way of grouping reduces the number of queries to a large extent.