1. **Subject Name: E-Commerce (BBAA405SE2)**
2. **Semester / Year: Semester IV**
3. **Name of the Teacher: Pinaki Dey**
4. **Name of the topic: Technology in E-Commerce (Part IV)**

The making of World Wide Web

The World Wide Web was designed originally as an interactive world of shared information through which people could communicate with each other and with machines.

Tim Berners-Lee, an English scientist, invented the World Wide Web (WWW) in 1989, while working at **CERN** near **Geneva, Switzerland**. The Web was originally conceived and developed to meet the demand for automated information-sharing between scientists in universities and institutes around the world.

The basic idea of the WWW was to merge the evolving technologies of computers, data networks and hypertext into a powerful and easy to use global information system.

Tim Berners-Lee wrote the first proposal for the World Wide Web in March 1989 and his second proposal in May 1990. Together with Belgian systems engineer **Robert Cailliau**, this was formalized as a management proposal in November 1990. This outlined the principal concepts and it defined important terms behind the Web. The document described a "hypertext project" called "World Wide Web" in which a "web" of "hypertext documents" could be viewed by “browsers”.

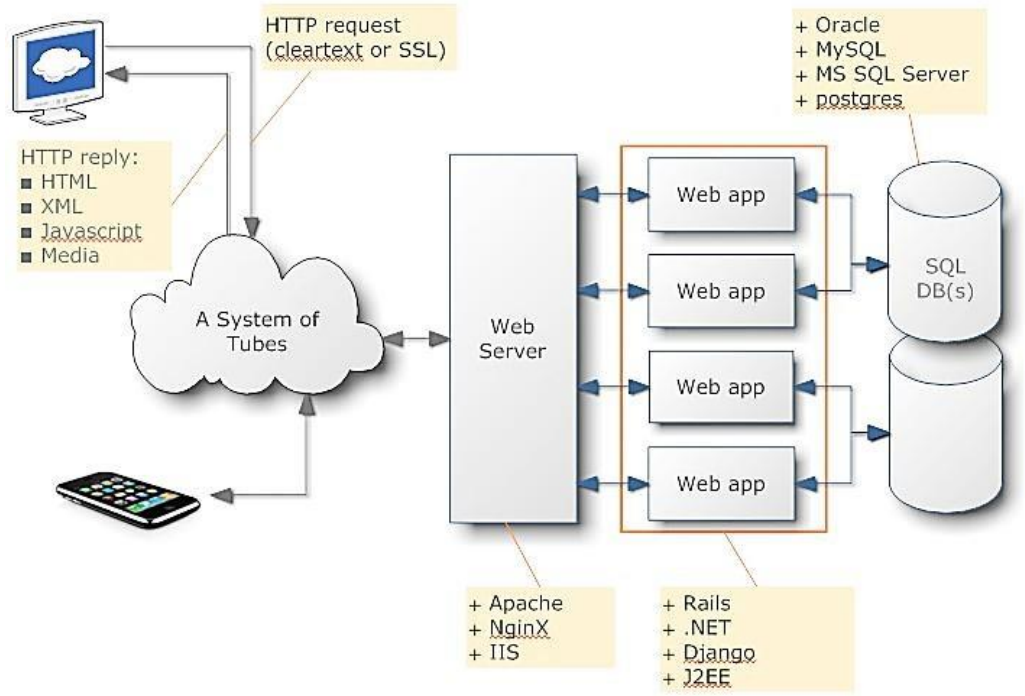
By the end of 1990, Tim Berners-Lee had the first Web server and browser up and running at CERN, demonstrating his ideas. He developed the code for his Web server on a NeXT computer. To prevent it being accidentally switched off, the computer had a hand-written label in red ink: "*This machine is a server. DO NOT POWER IT DOWN!!"*

The goal of the Web was to be a shared information space through which people (and machines) could communicate.

The intent was that this space should span from a private information system to public information, from high value carefully checked and designed material, to off-the-cuff ideas which make sense only to a few people and may never be read again.

Web Application Architecture / Web System Architecture:

The web application architecture describes the interactions between applications, databases, and middleware systems on the web. It ensures that multiple applications work simultaneously.



Let us understand it with a simple example of opening a webpage.

As soon as the user hits the go button after typing a URL in the address bar of a web browser, it requests for that particular web address. The server sends files to the browser as a response to the request made. The browser then executes those files to show the requested page.

Finally, the user is able to interact with the website. The most important thing to note here is the code parsed by the web browser. A web app works in a similar way.

This code might or might not have specific instructions that tell the browser how to respond with respect to the different types of user inputs.

Hence, web application architecture has to include all the sub-components as well as the external applications interchanges for the entire software application, in the aforementioned case, which is a website.

With any typical web application, there are *two* different codes (sub-programs) running side-by-side. These are:

- **Client-side Code -** The code that is in the browser and responds to some user input.
- **Server-side Code -** The code that is on the server and responds to the HTTP requests.

A web developer (team) developing the web application decides as to what the code on the server will do with respect to the code in the browser. For writing **server-side** code, C#, Java, JavaScript, Python, PHP, Ruby, etc. are used.

Any code that is able to respond to HTTP requests has the ability to run on a server. The server-side code is responsible for creating the page that the user requested as well as storing different types of data, including user profiles and user input. It is never seen by the end-user.

A combination of CSS, HTML, and JavaScript is used for writing the **client-side** code. This code is parsed by the web browser. Unlike the server-side code, client-side code can be seen as well as modified by the user. It reacts to user input.

The client-side code communicates only via HTTP requests and is not able to read files off a server directly.

Models of Web Application Components:

Depending on the total number of servers and databases used for a web application, the model of a web app is decided. It can be any of the following *three*:

1. **One Web Server, One Database**: It is the most simple as well as the least reliable web app component model. Such a model uses a single server as well as a single database. A web app builds on such a model will go down as soon as the server goes down. Hence, it isn't much reliable.

   One web server, one database web application component model is not typically used for real web applications. It is mostly used for running test projects as well as with the intent of learning and understanding the fundamentals of the web application.

2. **Multiple Web Servers, One Database**: The idea with this type of web application component model is that the web-server doesn't store any data. When the web-server gets information from a client, it processes the same and then writes it to the database, which is managed outside of the server. This is sometimes also referred to as a stateless architecture.

   At least 2 web servers are required for this web application component model. This is all for avoiding failure. Even when one of the web servers goes down, the other one will take charge.

   All requests made will be redirected automatically to the new server and the web app will continue execution. Hence, reliability is better as compared to the single server with inherent database model. However, if the database crashes the web app will follow to do the same.
3. **Multiple Web Server, Multiple Databases**: It is the most efficient web application component model because neither the web-servers nor the databases have a single point of failure. There are two options for this type of model. Either to store identical data in all the employed databases or distribute it evenly among them.

Types of Web Application Architecture:

Web application architecture is a pattern of interaction between various web application components. The type of web application architecture depends on how the application logic is distributed among the client and server sides.

There are *three* primary types of web Application Architecture. Each one of them is explained as follows:

- **Single-Page Applications (SPAs) –** Instead of loading completely new pages from the server each time for a user action, single page web applications allows for a dynamic interaction by means of providing updated content to the current page.

  AJAX, a concise form of **Asynchronous JavaScript and XML**, is the foundation for enabling page communications and hence, making SPAs a reality. Because single-page applications prevent interruptions in user experience, they, in a way, resemble traditional desktop applications. SPAs are designed in a way so that they request for most necessary content and information elements. This leads to the procurement of an intuitive as well as interactive user experience.

- **Micro-services –** These are small and lightweight services that execute a single functionality. The Micro-services Architecture framework has a number of advantages that allows developers to not only enhance productivity but also speed up the entire deployment process. The components making up an application build using the Micro-services Architecture aren't directly dependent on each other. As such, they don't necessitate to be built using the same programming language. Hence, developers working with the Micro-services Architecture are free to pick up a technology stack of choice. It makes developing the application simpler and quicker.
- **Server-less Architectures –** In this type of web application architecture, an application developer consults a **third-party cloud infrastructure** services provider for outsourcing server as well as infrastructure management.

  The benefit of this approach is that it allows applications to execute the code logic without bothering with the infrastructure-related tasks. The Server-less Architecture is best when the development company doesn't want to manage or support the servers as well as the hardware they have developed the web application for.

A few terminologies of Computer Network:

ISP:

An **Internet Service Provider** (ISP) is a company that provides access to the internet. Wherever we may be, every time we connect to internet, our connection is routed through an ISP.

In the beginning, dial-up modems were used by ISPs to grant internet connection. Regular phone lines were utilized during this time and speed was up to 56 Kbps only. During the late 1990s, cable modems and Digital Subscriber Line (DSL) were used by ISPs to offer faster internet access.

Today, fiber optic cables are used by some ISPs to offer high-performance fiber connections. Cable connections are being provided by popular companies like *Alliance Broadband, Tikona Infinet Ltd, Hathway Digital Cable TV, Siti Broadband, GTPL KCBPL* etc in India.

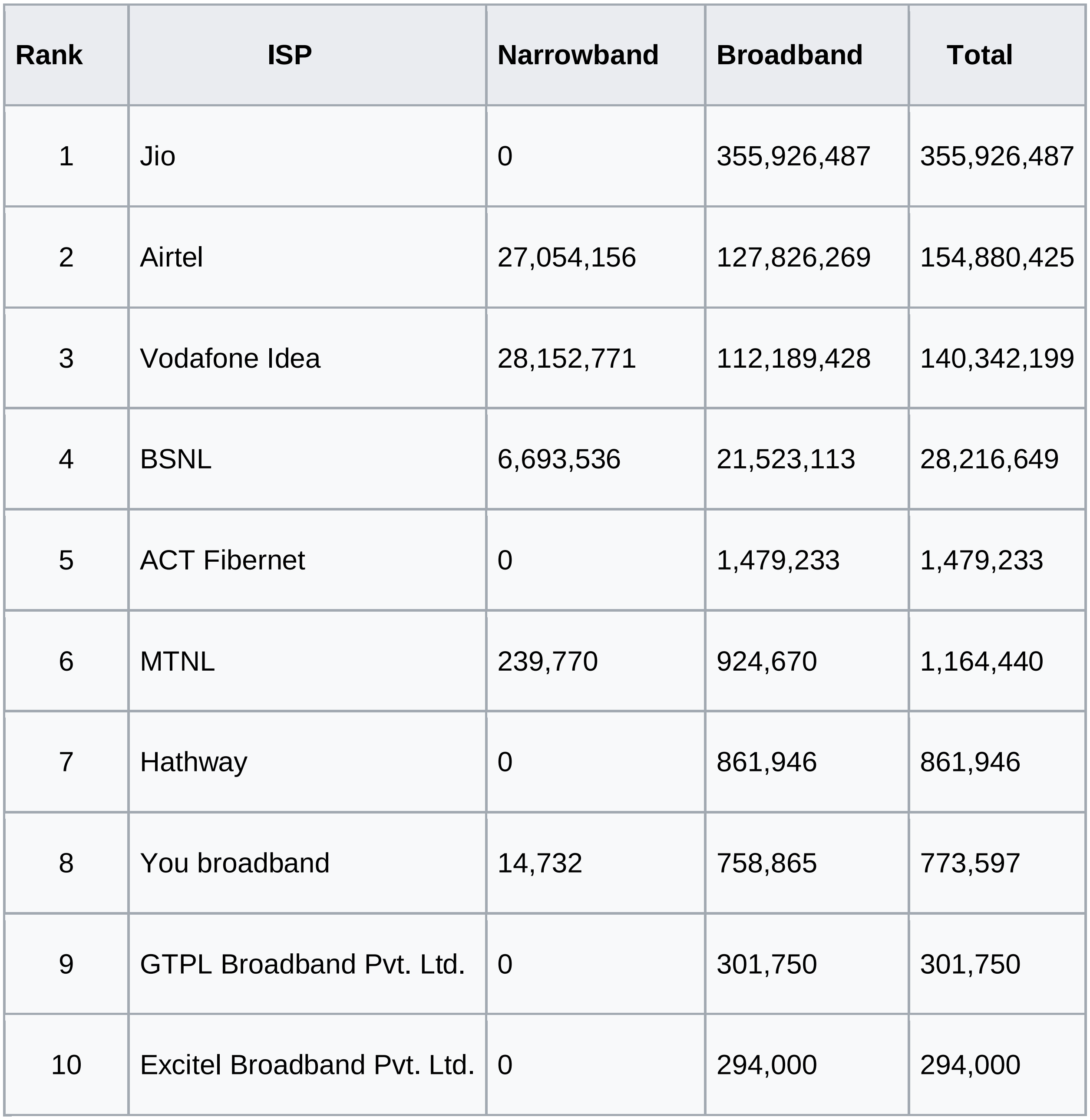
To link to an ISP, we need an active account with our ISP and a working modem. Connect the modem to the telephone line or cable outlet, depending on our ISP, in order to link with our ISP.

Upon verification of our account, the ISP assigns our modem an IP address. We are connected to the Internet once we have an IP address. We can use a router to link several devices to the Internet.

A router can be separate device or a built-in router in the modem. Since the devices are routed through the same modem, they will all utilize the same public IP address allocated by the ISP.

An ISP is also sometimes called an **IAP (Internet Access Provider)**. At times, ISP is used as an acronym for independent service provider to differentiate an independent, separate service provider from a telephone company.

The following table shows the top 10 ISPs in India by total subscriber base as of 30 September 2019 *(surveyed by, Telecom Regulatory Authority of India. 8th January 2020, "The Indian Telecom Services Performance Indicators July – September, 2019"*) .



Rank	ISP	Narrowband	Broadband	Total
1	Jio	0	355,926,487	355,926,487
2	Airtel	27,054,156	127,826,269	154,880,425
3	Vodafone Idea	28,152,771	112,189,428	140,342,199
4	BSNL	6,693,536	21,523,113	28,216,649
5	ACT Fibernet	0	1,479,233	1,479,233
6	MTNL	239,770	924,670	1,164,440
7	Hathway	0	861,946	861,946
8	You broadband	14,732	758,865	773,597
9	GTPL Broadband Pvt. Ltd.	0	301,750	301,750
10	Excitel Broadband Pvt. Ltd.	0	294,000	294,000

HTTP (Hypertext Transfer Protocol):

It allows for the retrieval of linked resources from across the web. As protocols went for accessing remote data, a standard did exist in the File Transfer Protocol (FTP). However, this was not optimal for the web, in that it was too slow and not sufficiently rich in features, so a new protocol designed to operate with the speed necessary for traversing hypertext links, Hyper Text Transfer Protocol, was designed.

HTTP uses a server-client model. A client, for example, may be a home computer, laptop, or mobile device. The HTTP server is typically a web host running web server software, such as Apache or IIS. When we access a website, our browser sends a request to the corresponding web server and it responds with an HTTP status code. If the URL is valid and the connection is granted, the server will send our browser the webpage and related files.

Cookies:

Cookies are files, generally from the visited web pages, which are stored on a user's computer. They hold a small amount of data, specific to a particular client and website, and can be accessed either by the web server or the client computer.

It can store a wide range of information, including personally identifiable information (such as our name, home address, email address, or telephone number)

This allows the server to deliver a page personalized to a particular user, or the page itself can contain some script which is aware of the data in the cookie and so is able to carry information from one visit to that website.

Cookies cannot be used to spread viruses and they cannot access our hard drive. This does not mean that cookies are not relevant to a user's privacy and anonymity on the Internet. In only this way are cookies a threat to privacy. The cookie will only contain information that we freely provide to a Web site.

In some browsers, each cookie is a small file but in Firefox browser, all cookies are stored in a single file, located in the Firefox profile folder. Cookies often store our settings for a website, such as our preferred language or location.

In Google Chrome browser, all cookies are stored in single file called Cookies. The file is located at the following path: "C:\Users\Your User Name\AppData\Local\Google\Chrome\User Data\Default."
A simple example of Cookies is shown here: #Example Cookie. # Recorded 2020-04-18. username=xyz.

Types of Cookies:

Generally, there are *three* types of computer cookies: Session, Persistent and Third-party. These are virtually invisible text files but all are very different. Each with their own mission, these cookies are made to track, collect and store any data that companies request.

- *Session Cookies* − These are mainly used by online shops and allows us to keep items in our basket/ tray when shopping online. These cookies expire after a specific time or when the browser is closed.
- Permanent Cookies − These remain in operation, even when we have closed the browser. They remember our login details and password so we don’t have to type them in every time to visit the site. It is recommended that we delete these types of cookies after a specific time.
- Third-Party Cookies − These are installed by third parties for collecting certain information. For example: Google Maps.