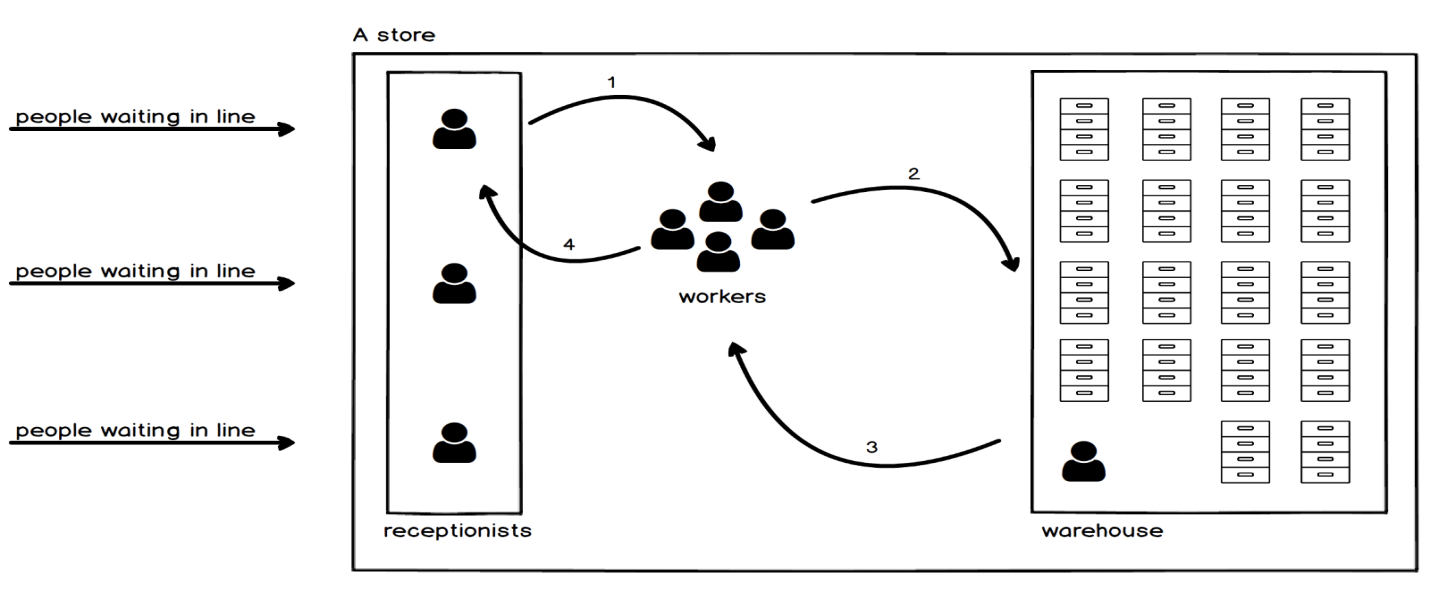
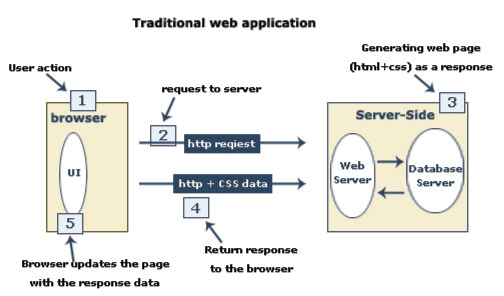
**How a Web Application Works**

Working of a web application is explained with the help of an IKEA store analogy. IKEA store is a store that sells furniture.



In IKEA store there are 3 lines in which people are waiting & each line has its receptionists. People give a list to the receptionists about the furniture they need. Receptionists give this list to workers & they give it to the warehouse manager. Warehouse has many cabinets & drawers in which there are parts of furniture. Warehouse manager takes the part according to the list & give it to the workers. Workers give them to Receptionists & then back to the customers. Here Workers speak different language with each other & warehouse manager speak a different language. In the same way a web application works.



In a web application, user searches for a url in browser. This is a http request which goes to a DNS server. DNS server has a table which contains domain name & its ip address. DNS fetch the ip address of the domain & search it in the cloud & display the information in the browser.

Now let’s map this with IKEA store. There are servers. There are many people sending http request to the server so they are lined up in the server. There are web servers (receptionists in IKEA store analogy) in the server which take the request & send it to the works who knows different language like **Ruby, php, Python, Java, nodeJS**, etc. It doesn’t matter which language the workers uses. Workers go to the database (eg sqlite, MYSQL, Oracle, Hadoop, MongoDB) (warehouse in IKEA store analogy), where different news are stored. Database managers (Warehouse manager) organize the inventories the way they think it should be organized. There are different databases because different people organize data differently. Databases may have small or large data. **Hadoop & Oracle** is an example of large data. Databases comes to the workers, they arrange it & gives back to the web server. Examples of Web servers are **Apache Tomcat (works with Java), WEBrick (works with Ruby), Ngnix. Web application produces HTML, CSS, Javascript, XML, JSON**. Workers with different languages come with different frameworks (toolboxes). There is an MVC framework ie Model (handles anything related to databases), Views (anyone that puts together html, css & javascript) & Controller (who controls information in between models & the views). MVC is the standard framework for creating the web application. Some frameworks for php are **Yii, Laravel, cakephp**, etc. some frameworks for Ruby are **Rail, Sinatra**, etc. For Python are **Django, Flask, pylon**, etc. Toolboxes organize javascript code to make it neater & here comes **Angular, React, Ember, Backbone, vue.**

**Scaling the web application**

Now again consider the IKEA store. What if the number of customer increases daily & then the customer has to wait for hours in the line. The customer gets pissed off & leaves the store without purchasing anything. Who’s mistake is this the Receptionist or the workers or the warehouse manager? What happens is warehouse manager has organized all the different inventories in a certain way but when the workers arrive & ask for something, the manager opens up the 1st drawer of the 1st cabinet, searches for that part & if not found search for the 2nd drawer & still not found in any drawer of 1st cabinet, he goes to the 2nd cabinet & so on. There are millions of drawers. So it took few hours to find an inventory. But suppose we have a database index which has all the parts & their place in the database. So the warehouse manager will look at the index & then will search for the part needed. So it becomes faster to find the inventory by looking at the index. Also in the case of web application, we can have the database index to find anything present in the database. But suppose now the workers are fast but they perform a million of step to assemble simple IKEA furniture. This is the fault of the person who created the instruction. In the case of web application, it is the fault of the developer. So it’s not the problem of language used to code it. It’s the problem of developer so he has to learn about Big O notation & how to write a efficient code & rewrite it. Now consider the workers go 500 times to the warehouse for taking parts of the furniture. They take the 1st part & perform operation, they again to the warehouse, ask for the 2nd part & perform operation & so on. So it’s better to go once ask for all the part needed & then assemble or perform operation on them. Now suppose we type a URL in the browser but it takes 2 sec for the http request to go to the web server & 2 sec from web server to the browser. When it takes 3 or more sec to load a page, the user will close the URL. It may happen because the server (say Africa) & the customer (say USA) may be far away from each other. So in this case move the server close to the customer or maybe have two servers i.e one at Africa & other at USA. Now suppose the data i.e. Html, css & javascript is huge & the internet is slow. Suppose we have 10 megabytes of data & internet could transfer only 1 megabyte of data per sec. In this case the customer will not think that their internet is slow but will think that our site is taking time to load. So in this case we can compress some data like images or combine some images into one or get rid of empty spaces or zip it.

The web server used is Thread based web servers, 1st http requests will get the data first. **Apache HTTP** is a **Thread based web server**. So in this case the other http request has to wait until the 1st request is completed. But in Event based system this doesn’t happens. In this the Customers fills the form, give it to receptionist. When the order is ready they give it them. **Ngnix** is an **Event based web server.**

Now suppose many users want the home page of yahoo.com which has top 20 news of the day. The web server gets the request from a user then it goes to the database, get the top 20 news & gives to that user. Again another users request for the same yahoo.com home page & this happens many times. So instead of this we can use caching as the top 20 news is not changing. So database will give the copy of that page & also the web server will give the copy of that page to the users.

To improve the scalability of the web application, we can scale up the web server or the database server when there is a huge traffic. **DevOps** set up the servers, monitor, scale up & down & write scripts. **Amazon, Microsoft & Google** now manages the database, scale up & down.

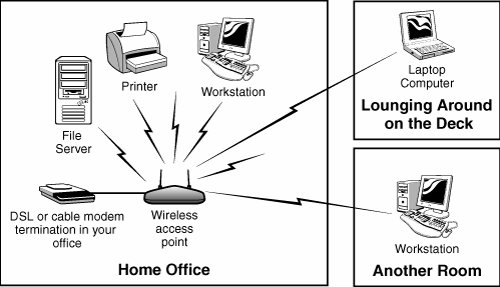
**Introduction to Networking**

What is Network?

At work or home, we have many devices like computers, printers, TVs & so on. We connect these by means of network. When the devices are connected they are able to share data. Eg sending a print job to a printer, sending an email or streaming a video. This can also be used for sharing an internet connection. For all these devices to communicate, they need to be connected somehow. One way is to plug the cables into the devices & connect them to another device called **Switch.** Eg School’s computer lab. In this case, computer in the lab connects to a wall socket. Another cable runs through the wall which comes out at a patch panel. The port on the patch panel connects to the switch. Later we can also connect these devices wirelessly, for this use a wireless access point. WIFI is an example of wireless network. The goal of the network is to move information from one device to another. For this, the sender & receiver must speak the same language which means that the devices agree on how data is sent, received, organized & handled. They agree on a process they follow called **Protocol (Ethernet or TCP or HTTP for accessing the web & SMTP used for email).** Network, Software, hardware is designed according to these protocol.

Network Types –

1. SOHO (small office home office) network – This has some computers, printers, some phones & a tablet. A router is used to connect to the internet.



People often say Hub when they mean switch because a switch look like a central hub in a SOHO network. But switches & hubs are different things.

1. Enterprise network – It covers several floor in the building or several office building in different cities or even across different countries. An internet provider has a very large internet provider called service provider network. It also offers services to connect their customers

A LAN (Local Area Network) is a collection of network devices in a local area like a single building.

1. WAN (Wide Area Network) – If banks are all over the country or world, then they can be joined together using WAN.