(front end framework) (fullstack web development)

Unit-4

Git and GitHub:

HTML - Markup lang - develop/design a website
CSS - styling language - look & feel
TS - scripting " - Interactiveness & dynamic
jQuery - js Library - " " - less no.of.Loc.

→ If we want to do a project (HTML, CSS, Js & JQuery)

It can be done: 1) Single:
2) >1 Person:

If we written a source code — Eg: If we lost the file / Deleted the file/program
If, any app (like whatsapp) stop working, Its Version might be changed from Android-Version-12 to 11. As it is a big Company, they will maintain the code. But If we want to get back to our lost/delete files, it is difficult to find the code (in real time)

→ there are multiple people doing a Project in Common, and separate for Places — for Project, all of them needs to collaborate and share source code which is risky in developing large projects

The solution for all these problems is "Git and Github."

Git: Version control system which is used to share the source code.
(Local repository (storage location)) — Linux
↳ one Person can work. - Single. (Desktop Version)

GitHub: Version control system - share source code (Remote Repository) - Microsoft - which is available at Server
multiple People can Collaborate

→ Desktop Version & web version.

Install:

① Google – Download Git for windows – Download Git..

– Latest Version (2.37.1)

→ Github – security purposes – NASA projects (uses Github)
(multiple Git).

→ To work with Github we need to have an account.

→ create account.

→ Create Repository.

→ Repository name: cdd – project.

Desc: Color project description – technologies used are HTML – CSS

If you want to create another – Go to Git

→ for every project you can create one separate repository

→ The option Public – view – download (it), but they can't add & edit the data until they are collaborated.

→ If we create private – they can't view / download – ".

→ You can also add a README file. → create repository.

– Edit the file (If you want)

→ Next commit changes

→ Added description of read me file (md – mark down)

Hash code – one Commit to another Commit

– You can also add folders with simple (/).

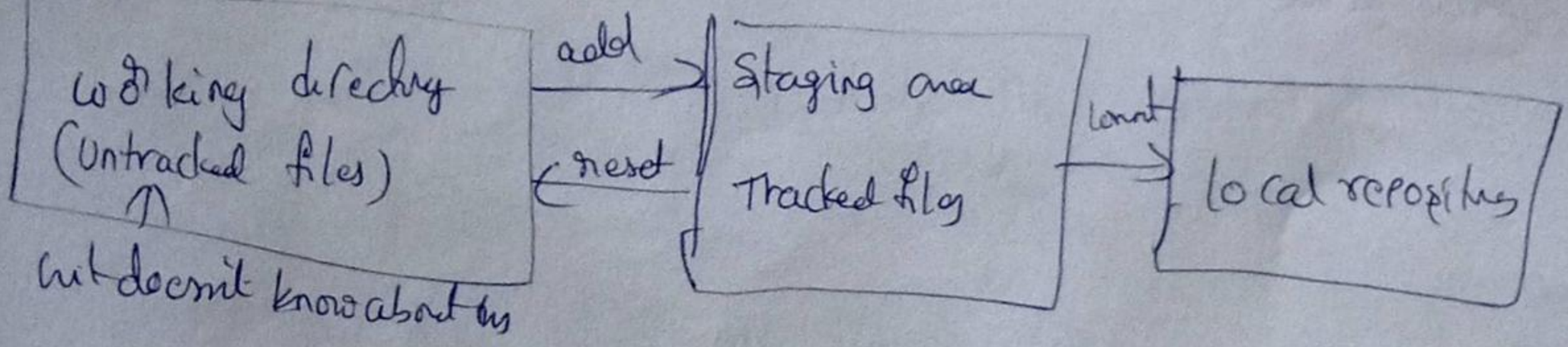
→ lastly go to settings and Create the Page.

Installation git and various Commands: — GIT CMD
GIT GUI
GIT Bash - Linux cmd where we can use Linux commands

Commands: (or) we have GUI Interface.

1) -- Version → To define installed git version
-- help → corresponding list of git commands.
init → to make our repository as local repository directory

① Gil cmd → git --Version.
Git GUI →
Git Bash →
→ can also go with normal cmd prompt.
→ git → cmd → git -init.



1) status: to check the status of code - Untracked / tracked / local.
2) Add: convert untracked files to tracked file
3) Reset: Tracked to untracked.
4) Commit: Permanent save.

Go to
→ visual studio

① open a Program index.html
② And then go to terminal, create a new terminal.

③ c:\ivs git version
② git init
③ git status (untracked files)
④ git add .
⑤ git status
⑥ git reset index.html
⑦ git status
⑧ git commit -m "index file created" -m "index file description"
⑨ git log (commits)
⑩ git --oneline (to display in single line)
⑪ git diff index.html

→ modify the code & Perform (git status)
↓
git add .
↓
git commit -m "anchor tag added"

→ git log --oneline (

→ git checkout (you can add one commit code) & Perform. (get back to the previous Version).

Eg: git checkout hashcode;
git checkout -b name;
Git checkout -d ename;

→ create del.html
→ Rename or move the file (git mv del.html d.html) → It will rename the file
→ git rm d.html (to delete the file).

→ we can see later

1) git add.
2) git commit -m "file deleted"
3) git log --oneline

. (Then we can see the status of all the created, renamed & deleted files).

(~~S~~ynchronize local repository with remote repository:)

1) create new repository (RGUKT – Project describes Push & Pull request).
After creating it copy HTTPS (link) (https://github.com/vishnupriyasklm/rgukt.git)

2) Also create a local repository in laptop by using visual studio
(ctrl + K to → create new folder → (sejv). → create a file index.html.

3) Go to terminal
1) git init
2) git add .
3) git commit -m "index file created"
4) git status.
5) git branch → To show list of branches

→ To transfer data from local repository (Git-admin) to remote repository Authentication is needed. (which location or repository you have to add)

→ To configuration we can use

git config --list

config: system, global & local.

→ git config --global user.name "jv"
git config --global user.email "mail address"

which location of repository you have to add.

Push: Push the data to remote repository (Push -u (upstream) origin master)
Pull: get from remote repository

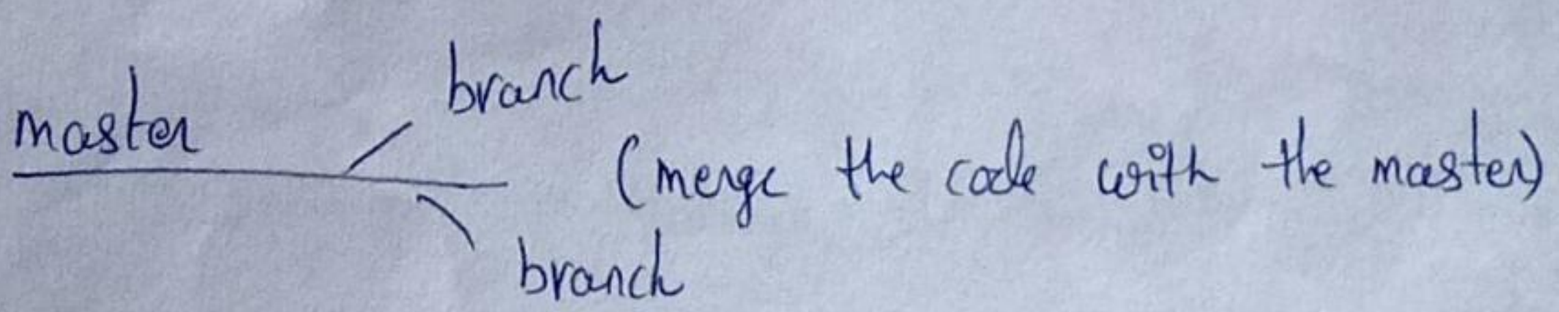
> git push -u origin master. (work code).

① > git remote add origin https://github.com/vighnupriyasklm/rgukt.git

② > git push -u origin master

Ser
② git add.
④ git push -u origin master.
① git commit -m "blue color index file".
② git push origin master

Concept of branches:



> ~~next~~: Execution:

1) git init
→ git add.
→ git commit -m "index file created"
→ git branch
(* main)
→ git config --list
→ git config --global user.name "vighnupriyasklm"
user.email "vighnupriya.j.sai@rgukt.ac.in"

→ git push remote add origin "url"

→ git push -u origin main

→ Then it will ask for authentication

→ no change the code in the program

→ git status

→ git add .

→ git commit -m "updated blue color"

→ git push -u origin main

→ git branch

→ main.

→ git branch redcolor (or) git checkout -b redcolor. → (Switched to a new branch redcolor)

→ git branch -

→ git push -u origin redcolor.

→ again update redcolor

→ git add .

→ git commit -m "redcolor"

→ git push -u origin redcolor.

→ Merge with pull request.

→ can also delete branch.

→ git branch -d redcolor.

→ See git branch.

And create a new file. (ese.html)

```
<html>
    <b> CSE Html Page </b>
</html>
```

and do git add.
git commit -m "cst file created"
git push (If it shows error then do)
git branch ocolor
git add.
git commit -m "Pushing with cst file"
git status
git push origin ocolor

→ whenever you push some data
first create branch & then push → so that admin can check
& will check merge or not

→ to update pull & update the local repository with remote repo
code & write own source code then go & push or not

① git add.
② git commit -m "end of italic tag in index"
③ git push (error) then
④ git pull
git commit -m "changes"
git pull
git push

> control Panel → user a/cs → Credential manage → windows Credential

synchronize local Repository with remote repository (collaborate):

① Public
② Private

⇒ Create a new repository.

→ create a folder in VS & new file
→ git init
→ git add.
→ git commit -m "file created"
→ git config --global user.name " "
→ git config --global user.email " "
→ git push -u origin master
→ git remote add origin " url "
→ git push -u origin master
→ ~~git~~ add .
→ git ~~push~~ -u origin master
→ git commit -m "Pink color is added"
→ git branch
→ git branch -P color
→ git checkout -b Pcolor
→ git branch
→ git push -u origin Pcolor
→ git commit -m "Pink color is added"
→ git push -u origin Pcolor
→ git status
→ git branch -d Pcolor
→ git checkout master
→ git branch -d Pcolor

→ git add .

→ git status

→ git commit -m "cse file added"

→ git PULL.

→ synchronize local with remote (collaborators)

⇒ open cmd in another system & write

git clone https://github.com/vishnupriyasklm/rguksk.git

→ code .

↳ enter

→ open folder (Rguksk)

→ open a program (new file).

edit changes & in terminal do → git status

git add .

git commit.

→ git push - u origin master

→ go to collaborators → add collaborator.

(in github)

→ again do git push - u origin master

→ git branch

→ git checkout -b rcolo

→ git add.

→ ~~git commit -m~~ ".

→ git reset index.html

→ git status.

→ change in the program.

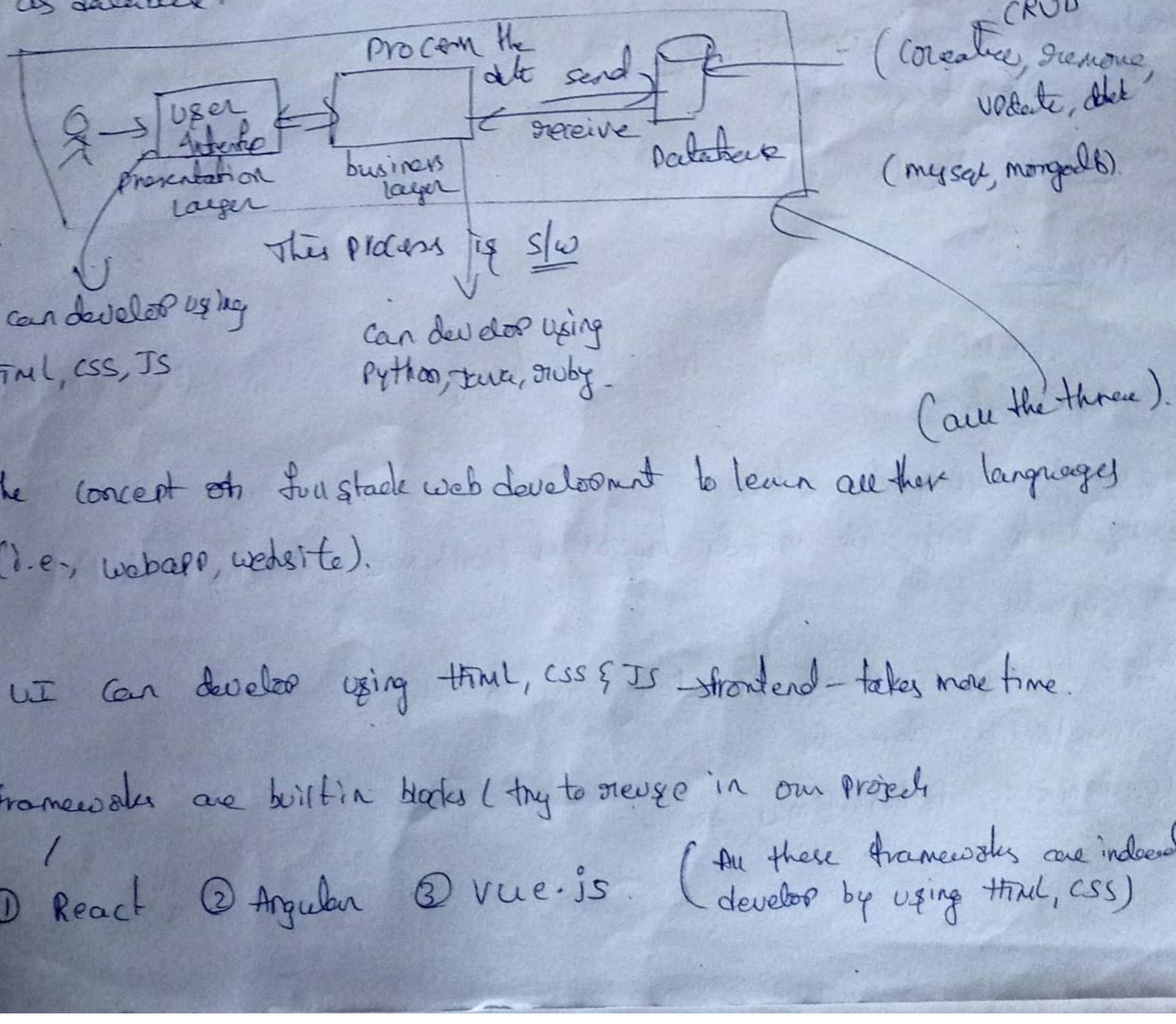
→ git add .

→ git commit -m "color added"

→ git push -u origin master newcolor.

Full stack web development & frontend frameworks:

→ If you want to develop a software, we should have a UI user will interact with UI & to process the data we require some kind of logic (This is called business logic) after processing we will store in some location & this location can be called as database.



→ The concept of full stack web development to learn all other languages (i.e., webapp, website).

→ UI can develop using HTML, CSS & JS → frontend - takes more time.

→ frameworks are built in blocks (try to reuse in our project

① React ② Angular ③ Vue.js (All these frameworks are indeed develop by using HTML, CSS)

UI Framework

→ Grid system need to be followed
→ font size & styles
→ navigation bar
→ buttons, Panel etc.

→ Frontend framework:

① React framework:

→ developed by facebook in 2013, going to use lang. JS (HTML+CSS)
→ used for Single Page / cross Platform / social nlws / web apps / messaging apps.

Adv: → less code / Reusable
Drawbacks: continuous Update (eg: Version 1, V2..)
→ Poor documentation

Companies: Uber, fb, netflix, Instagram etc.

② Angular:

→ To develop single Page applications / using model View Control (MVC) reduce the developing time
→ by google.
→ used typescript as base.
→ used in large scale projects / websites (with more Interactive content)
→ Upgrades

Adv: (large community / developing time is less.
→ customize elements
Companies: forbes, BMW etc.

Vue.js:

→ Developed in 2014.

→ To implement light weight Parts uses angular

→ less developing time / open source & uses larg. JS

Adv: open source / customization

Disadv: less community support, language barrier (coz developed by chinese developers)

→ limited Plugin

Companies: Google, Zoom, BMW, Trivago etc.

Build & Deployment:

→ how to deploy a particular website (so that it available to all users)

→ develop (HTML + CSS + JS)

→ After developing this website

→ It needs to available to all ppl.

→ For eg orggkt website is developed to a server, which is available to all users

→ Deployment is Push code to the server { local — XAMPP, apache (other ppl can't connect with & can access this) ↓ can't be accessed by 3rd Party ppl. / Remote

develop website upto a location i.e., university level

→ If we want global wide, need to connect remote Server (i.e, 3rd Party Server)

Eg: godaddy, netlify, AWS, digital ocean, github (

→ It needs some kind of Payment

→ 000webhostlogin → login → create web design