

## => Day-2 -> AI -> DevOps with AWS

-> DevOps Introduction -> Linux & Shell scripting  
-> AWS -> DevOps Tools -> projects.

### -> Software Project ->

L> It is a plan to create or improve a software Application to simplify human life.

-> It involves a team working together to design, build, test & deliver the software App within set of time & Budget

-> To develop any software, we use programming language (C, C++, Java, python, JS -----)

ex :- Zepto, Amazon, Facebook, uber -----

### -> Type of software Project

① Stand Alone / Desktop Application (only one user can access)

L> Notepad, OS, Calculator, Paint App, -----

② web Application

L> can be accessed by multiple users at a time with internet (internet is must to access web App)

L> Facebook, Amazon, IRIC, banking APP -----

③ Distributed App (webservice, REST API)

L> One App communicating one more App to re-use business logic

-> Telusro - Paypal

-> MakeMyTrip -> IRIC

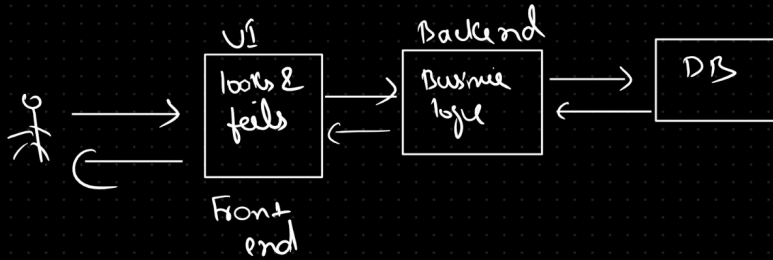
-> RPAY -> SBIBANK APP

④ Mobile App :- Apps which is used / Run in Mobile Devices (smartphones)

We can have both web App & Stand Alone Apps in smartphones (mobile devices)

## Application Arch: wh

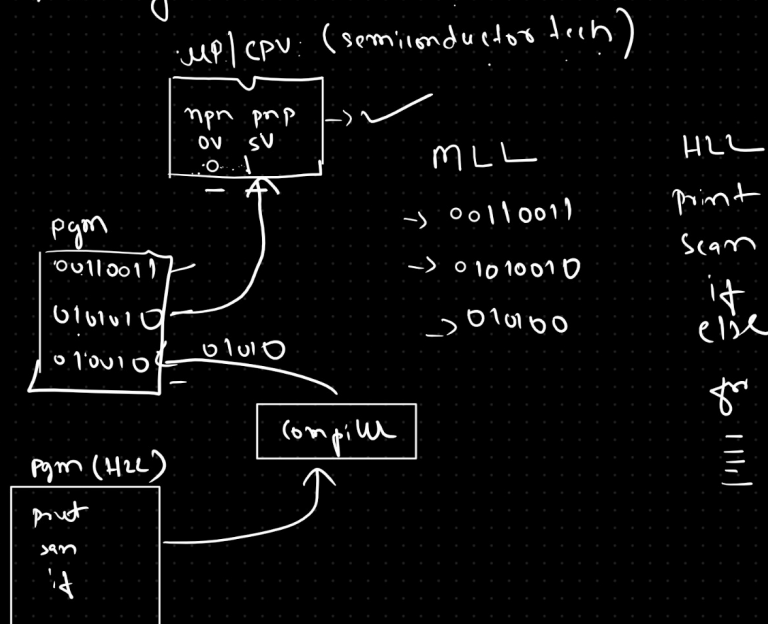
- ① Frontend (UI / Presentation) (HTML + CSS, JS, React, Angular)
- ② Backend (Business Logic) (Java-Springboot, JS-Node.js (ExpressJS) -----
- ③ Database (Storage) -> MySQL, MongoDB ---



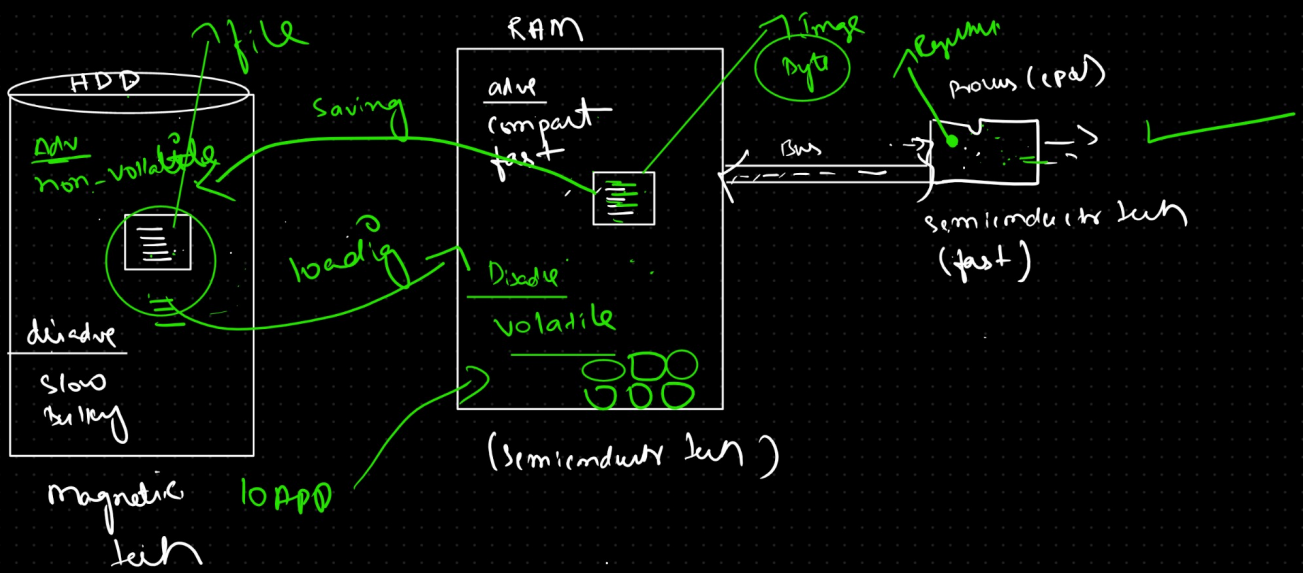
## Application Tech stack ->

Angular + Java-Springboot + MySQL  
ReactJS + Java-springboot + MySQL / MongoDB / Oracle  
Angular + Python + MongoDB  
ReactJS + NodeJS-ExpressJS + MongoDB

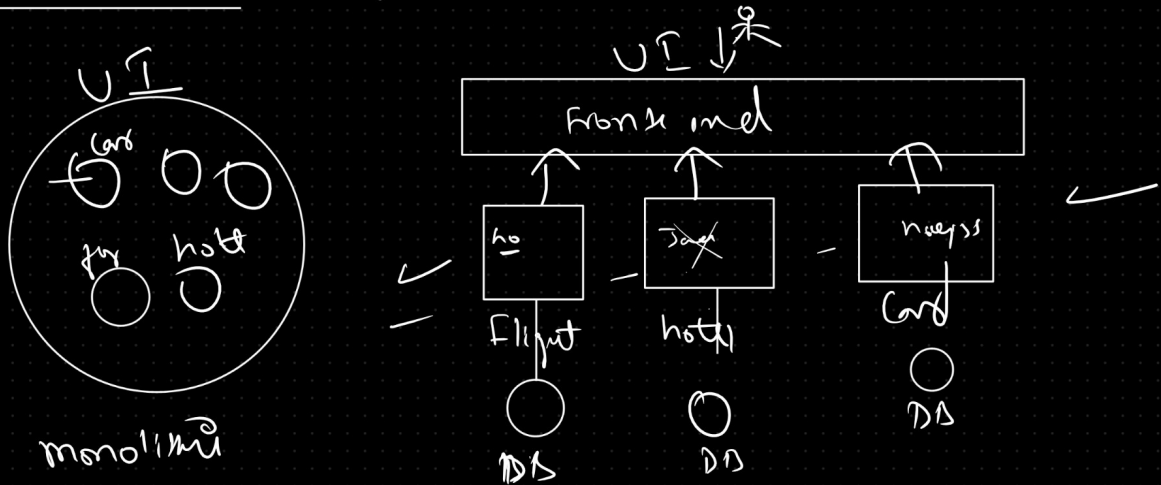
-> Programming -> coding ->



HDD / SSD, RAM, processor (CPU)



## Monolithic Architect microservice Architech (trending)



Monolithic Architecture --> If we develop all functionalities / services in a single app

### Drawbacks/Limitations:

- > Everything at one place
- > Single Point of failure
- > Burden on server
- > Maintenance is difficult
- > Re-Deploy Entire app which increases down time of app

Microservice --> Small Services which are independently deployable and executable  
(Developing app as multiple services/api depends on functionalities)

### Benefits

- > Loose coupled
- > High Availability
- > If there is prob in one service need not to re deploy all apis
- > Tech independence

{ SDLC, Agile, scrum → Intro Linux }