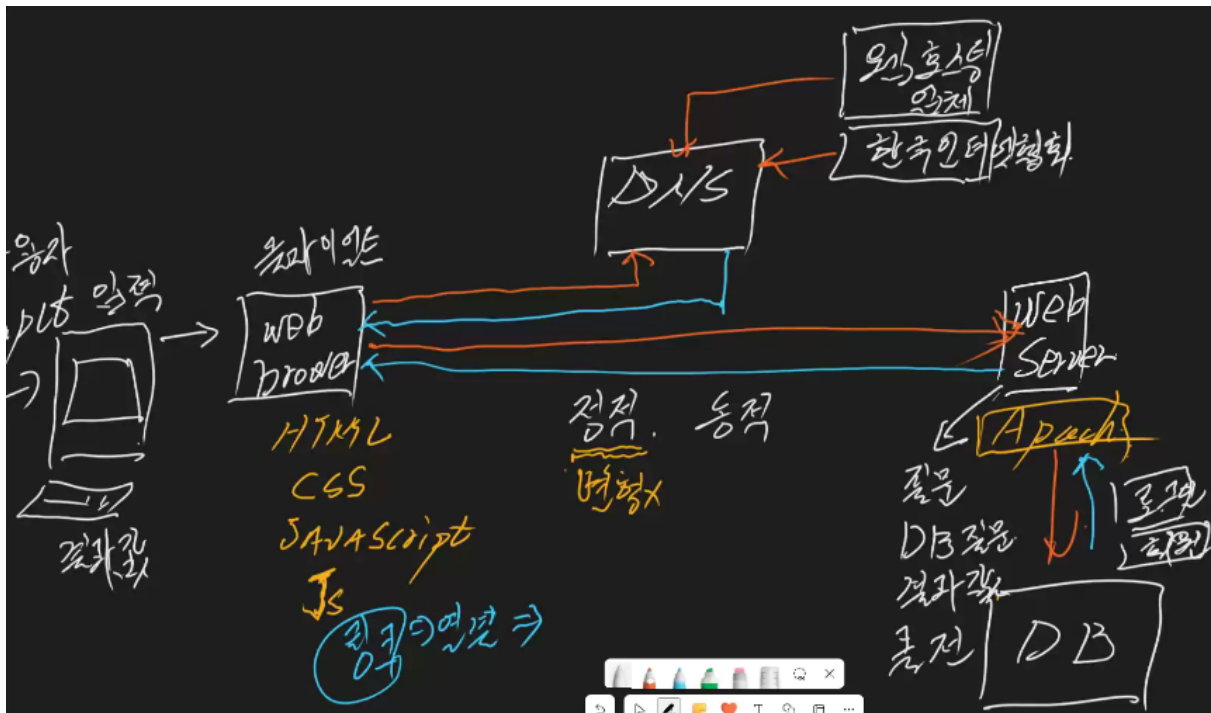
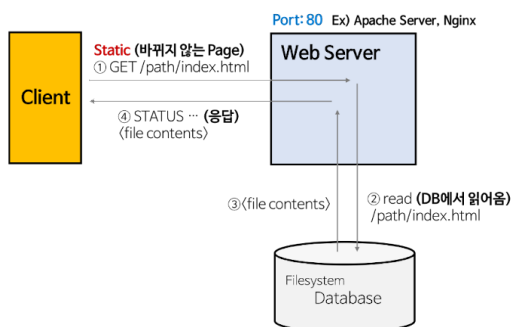


- PC - Web Browser(client) - DNS - Web Server - DB

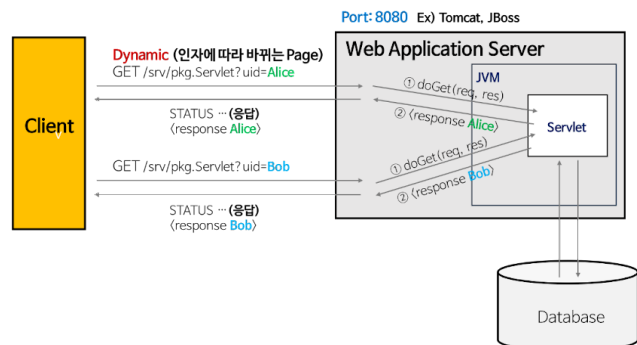


- PC -> Web Browser(client) < - > DNS
  - 웹 호스팅 업체, 한국 인터넷 협회
- PC - Web Browser(client) <---(DNS)---> Web Server
  - Web Browser(client) <-> Web Server
    - 정적(Static Pages) : 변형없는 자료, html, css, javascript 등
    - 동적(Dynamic Pages) : 인자에 따른 변경자료, # 다음 페이지 보충

## Static Pages



## Dynamic Pages



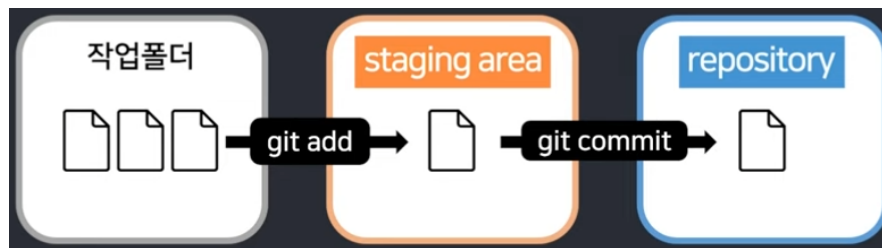
- Web Server : 질문, DB 질문, 결과값, 클라이언트에게 전달
- DB : 로그인, 회원기능 지원 가능

- 초기 웹서버는 DB와 연동되어 즉시 응답했으나, 동적 페이지의 요구 증가



- Git
  - repository name : 한글 불가(DB 구조에 따름), 내부 파일은 한글 내용 가능
- Github
  - snapshot : 만들었던 repository 에 대한 기록은 영구적
- Private
  - 설정시 해당 repository 자체가 안보임
  - 특정 유저 초대
    - invited collaborator > username
    - settings > collaborator > Add people > username
      - 수락시 Top Repositories 목록에 확인됨
- CLI
  - Window : CMD
  - Linux : Git bash
- repository download
  - clone
    - git clone https://github.com/CEB79/exm02.git
    - 전체 내용을 그대로 복사, 용량이 클 경우 오래 걸림
    - local directory에만 적용됨(github upload 연동x)
    - 같은 directory 명 복제 불가
    - private 설정시 복제 불가, 허락 필요,
      - 허락시 github의 메시지 확인해야 적용됨
  - pull
    - git pull origin main
    - 변경된 사항만 다운로드
- repository Push
  - Github의 공용 repository를 Manage로부터 허용 받은 상태
  - C:\git-work\
    - 시작점
  - git clone https://github.com/fullstack230925/1team.git
    - 로컬에 1team 디렉토리가 생성됨
    - 첫 다운로드는 clone or pull
    - 이후 변경사항만 받고 싶을 경우 : git pull origin main
  - cd 1team
    - 해당 디렉토리로 이동
    - C:\git-work\1team : 이동된 상태
  - git init
  - git status
  - C:\git-work\1team 디렉토리에 파일 아무거나 생성
  - git status
  - git add .
  - git status
  - git commit -m "파일업로드 테스트"
  - git branch -M main
    - 만약 branch1 에 넣을 경우 : git branch -M branch1
  - git push -u origin main
    - 만약 branch1 에 넣을 경우 : git push -u origin branch1

Youtube : [쉽게 설명하는 Git 기초 1. add, commit](#)  
[쉽게 설명하는 Git 기초 2. git diff & 응 vscode 쓸거야](#)  
[쉽게 설명하는 Git 기초 3. git branch](#)



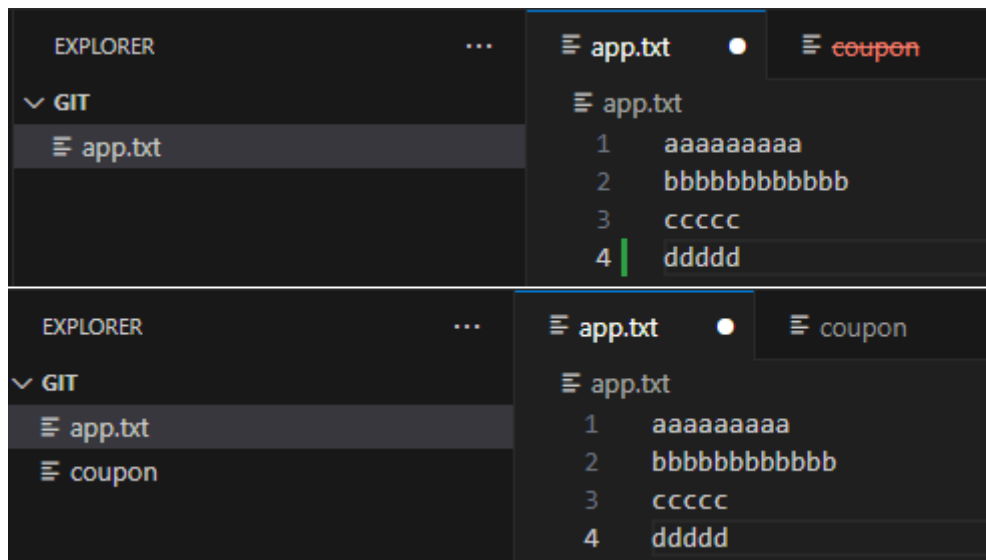
- branch 생성 / 해당 브랜치로 이동
  - 생성 : `git branch branch1`
  - 이동 : `git switch branch1`

```

● PS C:\Users\메리미\Desktop\MH\Git> git branch branch1
● PS C:\Users\메리미\Desktop\MH\Git> git switch branch1
Switched to branch 'branch1'
M      app.txt
● PS C:\Users\메리미\Desktop\MH\Git> git add .
● PS C:\Users\메리미\Desktop\MH\Git> git commit -m '파일만들었음'
[branch1 f99e486] 파일만들었음
 2 files changed, 3 insertions(+), 1 deletion(-)
 create mode 100644 coupon
○ PS C:\Users\메리미\Desktop\MH\Git> 

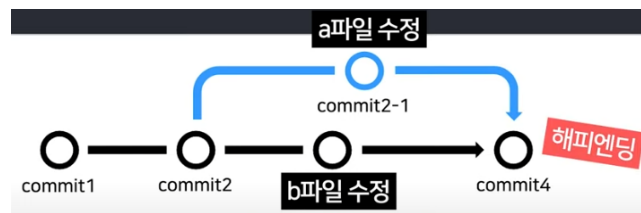
```

VS code 에서 실행한 모습



- main 을 branch 하여 app.txt 파일이 branch1에서도 동일하게 생성된 것을 확인
- branch1 에서는 별도로 coupon 파일을 만듦
- git switch main / git switch branch1 명령어를 통해 이동 가능
  - main 은 app.txt 만 있기에 coupon 파일이 없기에 붉은 색으로 표기됨
  - 정상적으로 branch 됨을 확인

- merge



case A : 다른 파일을 수정시 merge 는 큰 문제없이 정상적인 수정완료



case B : 같은 파일을 수정시 merge 보단 충돌(conflict) 발생

1. git switch main
  - a. 우선 메인으로 이동
2. git merge branch1
  - a. branch1 과의 merge 실행
  - b.

Case A : 아무 문제없이 수정됨(app.txt 와 coupon 파일이 존재함)

Case B : 충돌 부분을 해결을 위해 둘중 하나를 삭제해야함

```

app.txt
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
1 <<<<<< HEAD (Current Change)
2 메인브랜치에서 코드수정함
3 =====
4 쿠폰브랜치에서 수정함
5 >>>>>> branch1 (Incoming Change)
6 bbbbbbbbbbbb
7 ccccc
8 ddddd

Resolve in Merge Editor

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Switched to branch 'branch1'
PS C:\Users\메리미\Desktop\VH\Git> git add .
PS C:\Users\메리미\Desktop\VH\Git> git commit -m '쿠폰브랜치에서 수정'
[branch1 f508225] 쿠폰브랜치에서 수정
1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\메리미\Desktop\VH\Git> git switch main
Switched to branch 'main'
PS C:\Users\메리미\Desktop\VH\Git> git merge branch1
Auto-merging app.txt
CONFLICT (content): Merge conflict in app.txt
Automatic merge failed; fix conflicts and then commit the result.
PS C:\Users\메리미\Desktop\VH\Git>
  
```

main or branch1 부분중 하나를 삭제시 merge 됨

3. git branch -d branch1
  - a. merge 후 남아있는 브랜치는 불필요하므로 삭제