TDS-GA2-Q2-Notes (By - moon)

I believe in Teaching not Cheating

Answer `https://github.com/moonejs/TDS-GA-2-Q1`

> Note
>
> Don't copy the above answer ,everyone has different zip files Let's understand and solve it for you

Case Study: Removing .env from Git History

Problem

A .env file containing secret keys was accidentally committed to the repository.
We need to remove it completely from Git history and prevent it from happening again.

> Note
>
> You have to connect your github account with wsl (I have written the commands using wsl so wsl or bash terminal)

Do it first

Let's first learn some concepts

Important Concepts

1. Git History

Git stores project history as a chain of commits.
Each commit saves a snapshot of files at that time.

If a file was committed once, it stays inside history even if you delete it later.

So deleting .env normally is not enough.
We must rewrite history.

---

2. git log --all -- .env

- `git log` shows commit history.
- `--all` checks all branches.
- `-- .env` tells Git to search only for changes related to the .env file.

This helps us find when .env was added.

---

3. --diff-filter=A

Command:

```
git log --all --diff-filter=A -- .env
```

`A` means "Added".

This shows the commit where .env was first added.

---

4. Rewriting History

To remove a file from every commit, we use:

```
git filter-repo --path .env --invert-paths --force
```

Explanation:

- `--path .env` selects the file
- `--invert-paths` removes that file
- `--force` allows rewriting

This changes commit hashes but keeps the rest of the history.

Step-by-Step Procedure

Step 1: Go into the repository folder (use **WSL**)

```
cd repository-folder
```

Check that commits exist:

```
git log --oneline
```

You should see many commits.

Step 2: Find when .env was added

```
git log --all --diff-filter=A -- .env
```

Note the commit. This is only for checking.

Step 3: Remove .env from entire history

```
git filter-repo --path .env --invert-paths --force
```

This removes .env from all commits.

Step 4: Verify it is removed

```
git log --all -- .env
```

If nothing appears, the file is removed from history.

Also confirm history still exists:

```
git log --oneline
```

You should still see many commits.

Step 5: Create .gitignore

```
touch .gitignore
```

Open it and add:

```
vim .gitignore
```

after this press `i` and enter the below text

```
.env
```

This prevents .env from being committed again.

press `esc` and enter `:wq` and hit the enter

> **Important**
>
> Remember this step , this helps to write into the file using cli I will not teach this furthur

Step 6: Create .env.example

```
touch .env.example
```

Add placeholder values like: (recall the previous step i have used to add into a file)

```
API_KEY=your_api_key_here
DB_PASSWORD=your_database_password_here
SECRET_KEY=your_secret_key_here
```

This is good practice because it shows required variables without exposing real secrets.

Step 7: Commit the changes

```
git add .gitignore .env.example
git commit -m "Add .gitignore and .env.example to prevent secret leaks"
```

Step 8: Push to GitHub

Since history was rewritten, force push is required:

```
git push --force --all
git push --force --tags
```

Final Understanding

- Deleting a file does not remove it from history.
- Rewriting history is required to remove sensitive data.
- .gitignore prevents future tracking.
- .env.example is used instead of committing real secrets.

I skipped some commands is so that you can also explore and practice on your own instead of just copying everything blindly.

Follow this repo for the notes of other question ----

``

TDS-GA2-Notes	Link
`git commands` repo	Link