

- ◆ Project Title

# Team-Based Full-Stack Notes & Weather App using GitHub

---

- ◆ Project Overview

This project allows a team to collaboratively build a simple web application that:

- Lets users add & save notes
- Shows current weather using an API
- Uses GitHub for teamwork (branches, pull requests)
- ◆ Technologies Used
  - **Frontend:** HTML, CSS, JavaScript
  - **Backend:** Node.js (Express) (optional for notes)
  - **API:** OpenWeatherMap
  - **Version Control:** GitHub
- ◆ Team Roles (Example)

MEMBER	ROLE
STUDENT 1	Frontend (UI, Notes page)
STUDENT 2	Weather API integration
STUDENT 3	Backend / GitHub manager



## STEP 1: Create GitHub Repository

Go to [github.com](https://github.com)

Click New Repository

Name it: notes-weather-app



Select Public

Click Create repository

The screenshot shows the GitHub interface for creating a new repository. At the top, there are several icons: a search bar, a plus sign for creating a new item, a circular icon, a gear icon, a car icon, and a GitHub logo. A dropdown menu is open, showing options: 'New repository' (which is highlighted with an orange border), 'Import repository', 'New codespace', 'New gist', and 'New organization'. Below this, the 'issue' button is visible. The main form for creating a repository is shown, with fields for 'Owner' (set to 'PUBLIC' and 'ben'), 'Repository name' ('iOSApp'), and a note about repository names being short and memorable. There is a text area for 'Description (optional)' containing 'iOS project for our mobile group'. Below this, there are two radio buttons for 'Visibility': 'Public' (selected) and 'Private'. A note under 'Public' says 'Anyone can see this repository. You choose who can commit.' A note under 'Private' says 'You choose who can see and commit to this repository.' There is also a checkbox for 'Initialize this repository with a README', which is unchecked. At the bottom of the form is a large green 'Create repository' button.

in

New repository

Import repository

New codespace

New gist

New organization

issue

Owner

PUBLIC ben

Repository name

iOSApp

Great repository names are short and memorable. Need inspiration? How about [drunken-dubstep](#).

Description (optional)

iOS project for our mobile group

**Public**  
Anyone can see this repository. You choose who can commit.

**Private**  
You choose who can see and commit to this repository.

**Initialize this repository with a README**  
This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

Add .gitignore: **None** Add a license: **None**

Create repository



## STEP 2: Clone Repository (All Team Members)

```
git clone https://github.com/username/notes-weather-app.git
```



```
cd notes-weather-app
```

```
cuttlefish@ubuntu: ~
File Edit View Search Terminal Help
cuttlefish@ubuntu:~$ git --version
git version 2.19.1
cuttlefish@ubuntu:~$ git clone
https://github.com/cameromcnz/rock-paper-scissors.git
Cloning into 'rock-paper-scissors'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 776 (delta 2), reused 0 (delta 0), pack
Receiving objects: 100% (776/776), 29.08 MiB | 1.82 MiB
Resolving deltas: 100% (151/151), done.
cuttlefish@ubuntu:~$
```

Go to file

Add file ▾

< > Code ▾

Local

Codespaces

Clone



HTTPS    SSH    GitHub CLI

<https://github.com/github/docs.git>



Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP



## STEP 3: Create Branches (Team Work)

Each member creates their own branch:

```
git checkout -b notes-feature
```

```
git checkout -b weather-feature
```



```
# How to Create a Branch in GIT?  
  
# Syntax  
$ git checkout -b <new-branch>  
  
# Example  
$ git checkout -b feature-2  
Switched to a new branch 'feature-2'
```



## STEP 4: Build Frontend (Notes Page)

Create files:

index.html

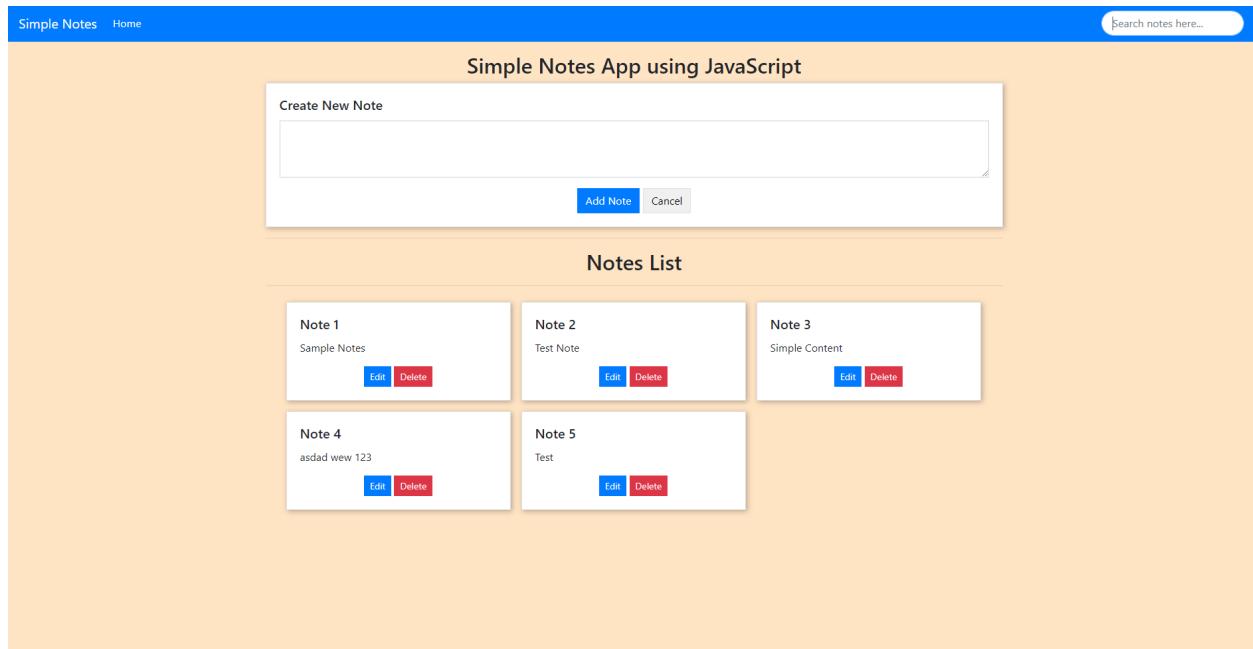
style.css

script.js

Simple Notes UI

```
<textarea id="note"></textarea>  
  
<button onclick="saveNote()">Save</button>
```





The screenshot shows the FormatMagic UI interface for building forms. On the left, there's a sidebar with various input field types: Text, Link, Form, Input, Textarea, Select, Checkbox, Radio, Switch, Datepicker, and Message. The "Forms" section is expanded, showing a preview of a form titled "Form title". The form contains fields for Customer Name, Income, Gender, Contact methods (Email or SMS), a date selector for Date of birth, a newsletter subscription toggle, and a large Textarea field with placeholder text "Some text here". A "Submit" button is at the bottom. To the right of the form preview, there are sections for "Textarea" (Process data key: textKey), "Properties" (Label, Input label, Placeholder, Helpertext), "Datasource" (Default value: eg. Name), and "Validators" (with a "Add a validator" button). The top of the screen shows the project name "docs\_form | Version 1 | draft", a language dropdown set to "Romanian", and status indicators for "Saved" and "Start".



## STEP 5: Weather API Integration

Get free API key from OpenWeatherMap

Use JavaScript fetch

```
fetch(`https://api.openweathermap.org/data/2.5/weather?q=Delhi&appid=API_KEY`)
```



```
{  
  cod: "200",  
  message: 0.0047,  
  cnt: 3,  
  - list: [  
    - {  
      dt: 1511881200,  
      - main: {  
          temp: 275.82,  
          temp_min: 275.82,  
          temp_max: 277.894,  
          pressure: 994.71,  
          sea_level: 1026.39,  
          grnd_level: 994.71,  
          humidity: 61,  
          temp_kf: -2.07  
        },  
      - weather: [  
        - {
```

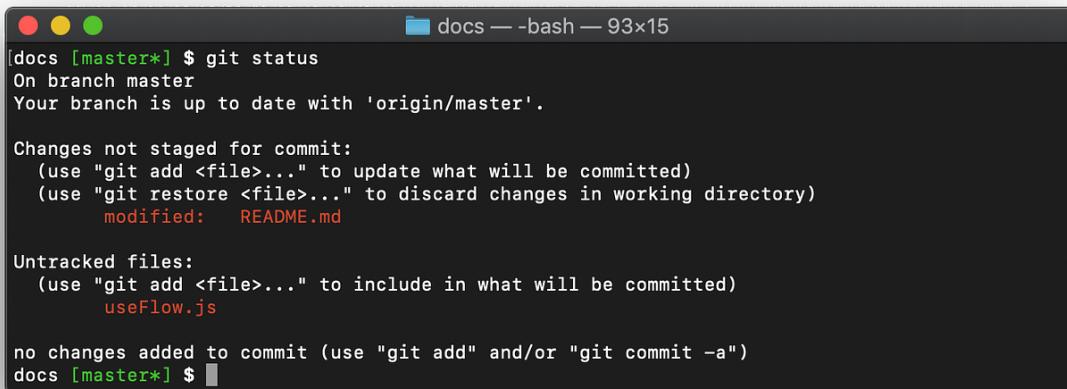


## STEP 6: Commit & Push Changes

```
git add .
```

```
git commit -m "Added notes feature"
```

```
git push origin notes-feature
```



```
[docs [master*] $ git status  
On branch master  
Your branch is up to date with 'origin/master'.  
  
Changes not staged for commit:  
  (use "git add <file>..." to update what will be committed)  
  (use "git restore <file>..." to discard changes in working directory)  
    modified: README.md  
  
Untracked files:  
  (use "git add <file>..." to include in what will be committed)  
    useFlow.js  
  
no changes added to commit (use "git add" and/or "git commit -a")  
docs [master*] $ ]
```



```
Ravish Kumar@Ravish MINGW64 /e/Yug/Programming/New folder/Linked_List (master)
$ git commit --m "2nd commit"
[master b12a692] 2nd commit
 1 file changed, 1 insertion(+), 1 deletion(-)

Ravish Kumar@Ravish MINGW64 /e/Yug/Programming/New folder/Linked_List (master)
$ git push
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin master

Ravish Kumar@Ravish MINGW64 /e/Yug/Programming/New folder/Linked_List (master)
$ git status
On branch master
nothing to commit, working tree clean
```

## STEP 7: Pull Request & Merge

Open GitHub repository

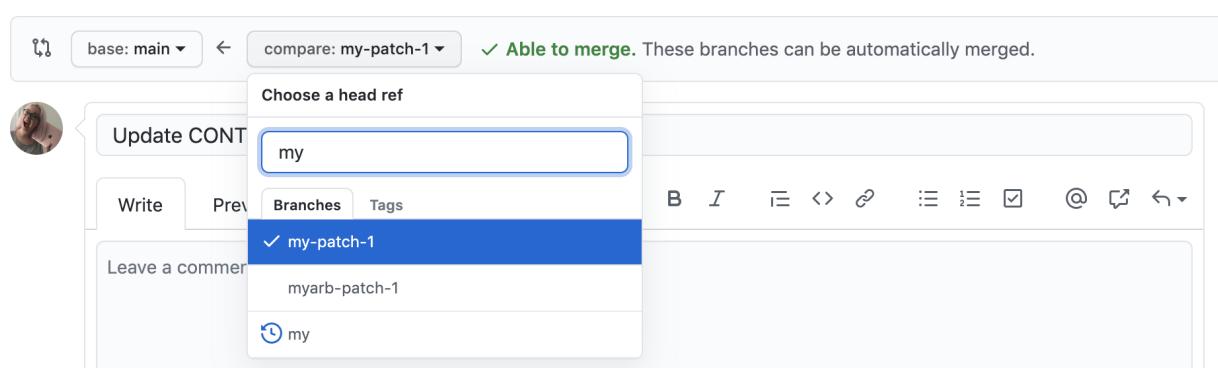
Click Compare & Pull Request

Team leader reviews

Click Merge

### Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).



Merge pull request



You can also [open this in GitHub Desktop](#)

✓ **Create a merge commit**

All commits from this branch will be added to the base branch via a merge commit.

**Squash and merge**

The 1 commit from this branch will be added to the base branch.

**Rebase and merge**

The 1 commit from this branch will be rebased and added to the base branch.

astng them.



## STEP 8: Final Output

- ✓ Notes section working
- ✓ Weather info displayed
- ✓ Team collaboration done





- ◆ Advantages of This Project

- Real team experience
- Learn GitHub collaboration
- Practical full-stack skills

