

# Lesson:

# Useful VS Code extension



# Topics

- Git lens
- Git history
- Git graph

## GitLens - Link

GitLens also known as Git Lens – Git Supercharged is a powerful open-source extension for Visual Studio code that makes it easy to understand and visualize your code history.

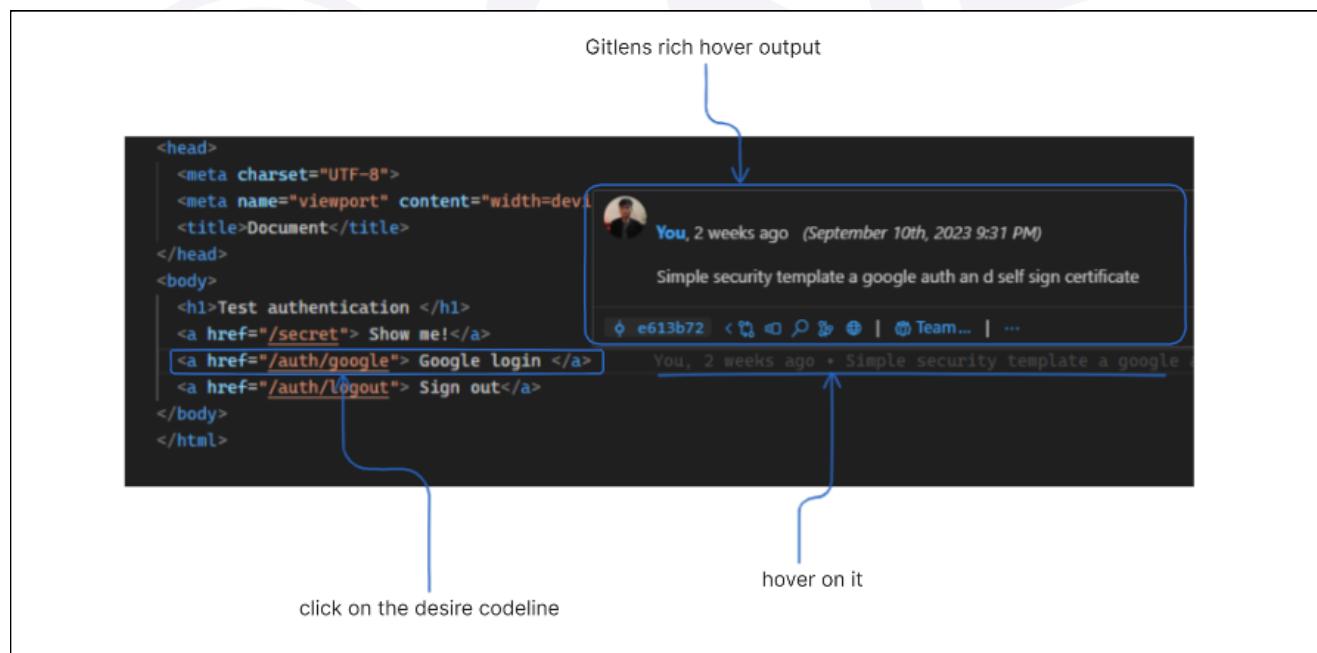
GitLens set itself apart from other Git tools through its deep level of integration, ability to adapt many different functions, and ease of use.

It provides a lot of free and paid features, some of the free features are as follows –

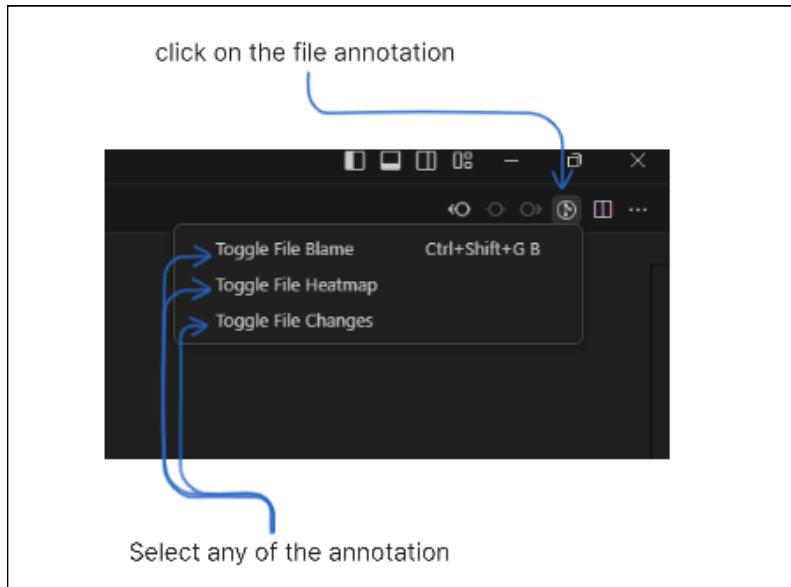
**1. Rich Hovers** – provide detailed information about a line of code, including its auth, commit history, and associated issues. To view a rich hover, simply hover your cursor over a line of code.

The rich hover will display the following information –

- a. The author of the line code
- b. The commit in which the line of code was introduced
- c. The commit message for the commit in which the line of code was introduced.
- d. Any associated issues or pull request

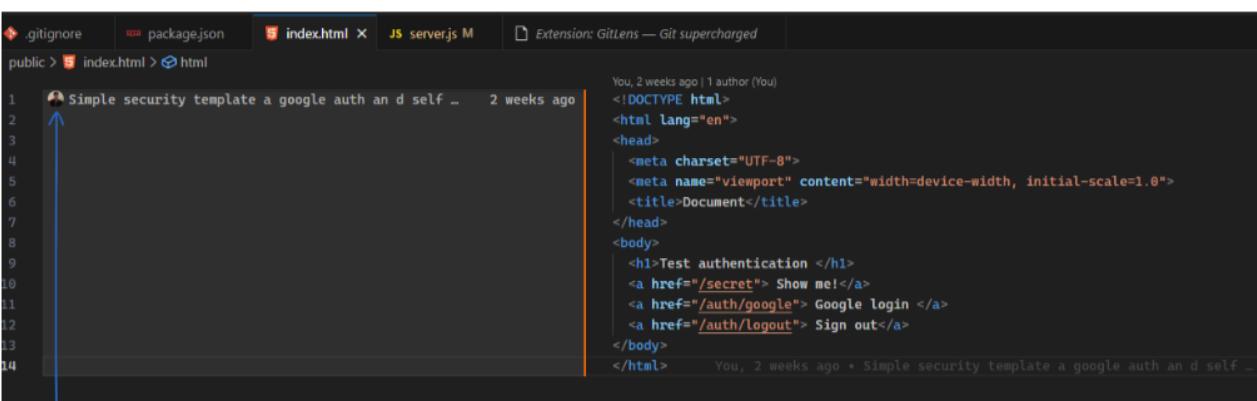


**2. File Annotations** – provide a high-level overview of the authorship and recent changes to a file. To view file annotations, toggle on the file button in the GitLens status bar.



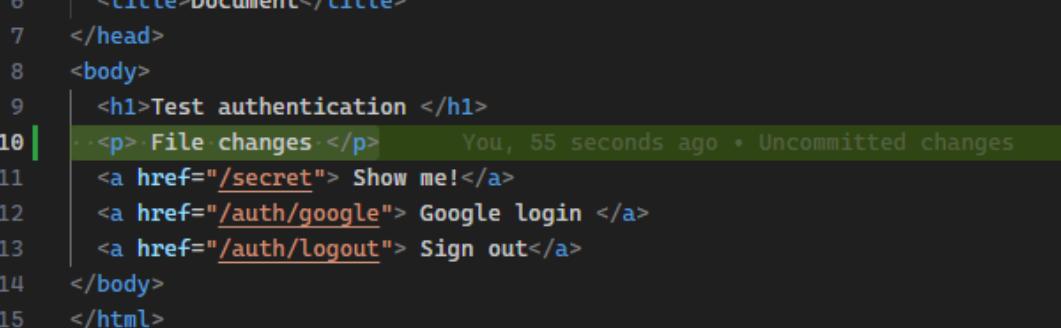
File annotations will display the following -

- The author of each line of code in the file (click on Toggle File Blame)



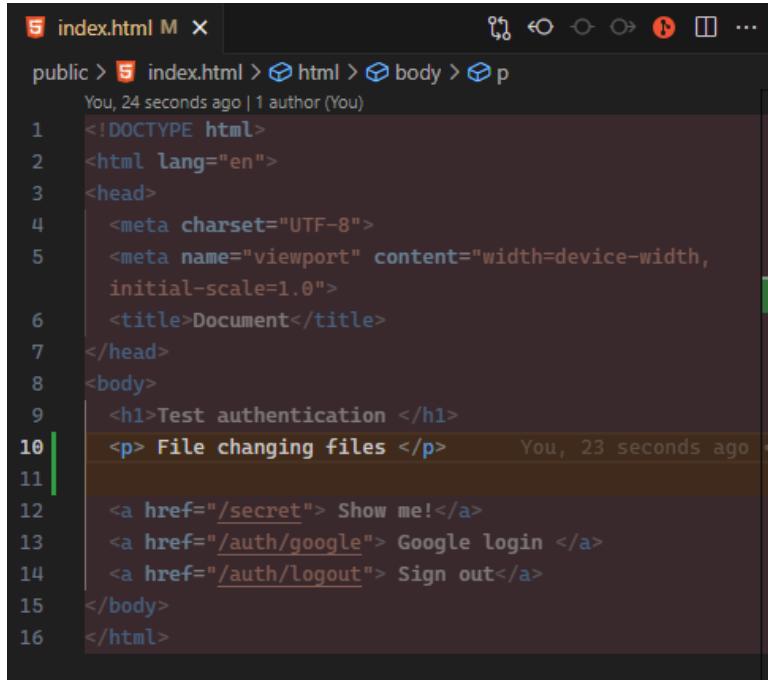
The code is written by only person with a commit message of - "Simple security template...."

- The date and time of the last change to each line of code in the file



File changes will be shown by green highlighting

c. A heatmap of the file shows the most frequently changed areas

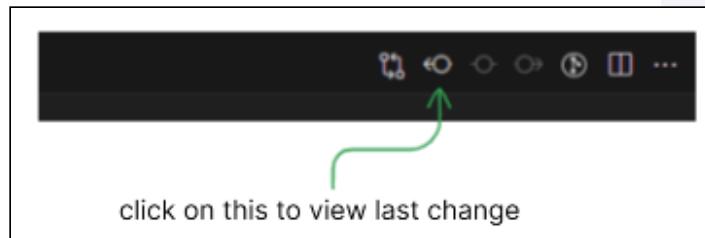
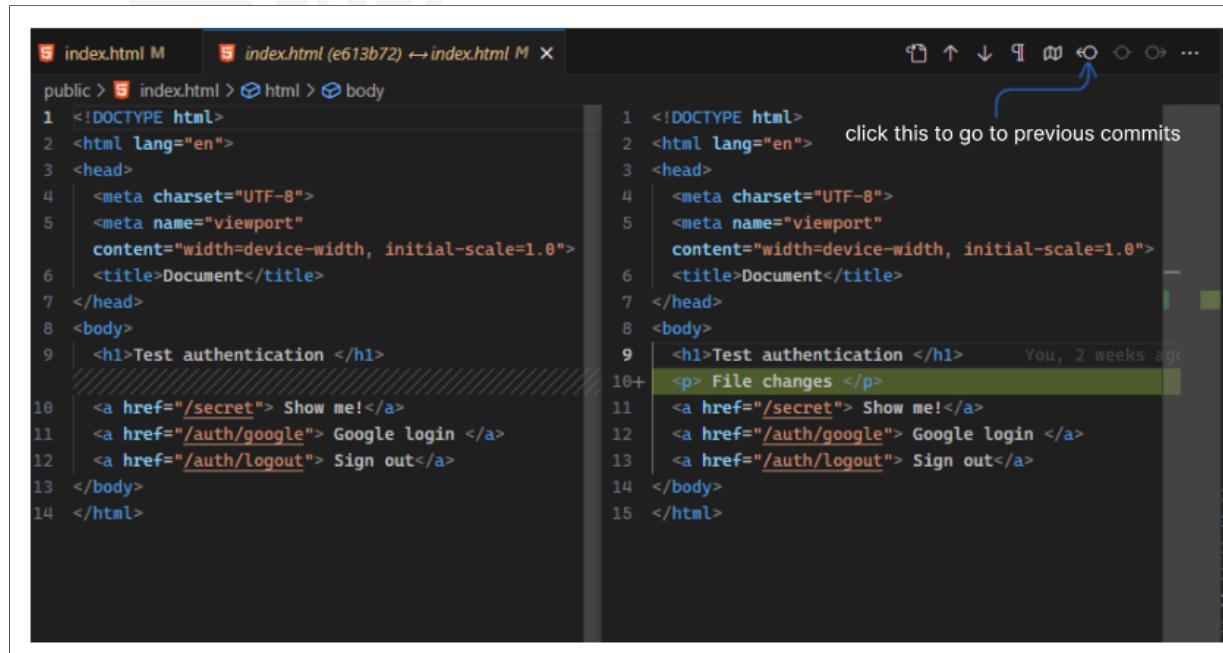


```

index.html M X
public > index.html > html > body > p
You, 24 seconds ago | 1 author (You)
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width,
initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <h1>Test authentication </h1>
10  <p> File changing files </p> You, 23 seconds ago .
11
12  <a href="/secret"> Show me!</a>
13  <a href="/auth/google"> Google login </a>
14  <a href="/auth/logout"> Sign out</a>
15 </body>
16 </html>

```

**3. Revision Navigation** – This allows you to navigate through the history of a file easily. To view the revision history of a file, right-click on the file in the GitLens sidebar and select Revision History.

```

index.html M index.html (e613b72) ↔ index.html M X
public > index.html > html > body
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport"
content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <h1>Test authentication </h1>
10  <a href="/secret"> Show me!</a>
11  <a href="/auth/google"> Google login </a>
12  <a href="/auth/logout"> Sign out</a>
13 </body>
14 </html>

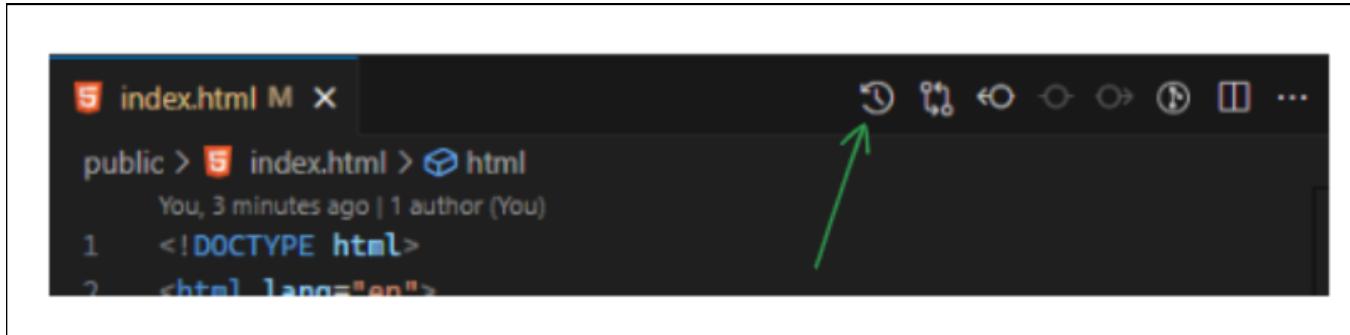
```

click this to go to previous commits

## Git history - [Link](#)

Git History provides a comprehensive set of features. Using Git History, users can access the Git log with a graphical representation and detailed information. As the name implies, it allows users to view and search the history, as well as compare branches, commits, and files across different commits, alongside various other features. In the latest update of Visual Studio Code, you'll find many of these miscellaneous Git History features already integrated into the Source Control section.

To view Git History click on the clock icon as shown below -



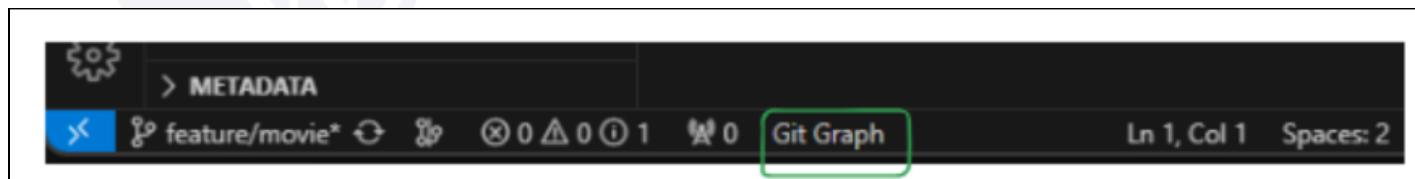
## Git graph - [Link](#)

Git Graph is a Visual Studio Code extension that provides a visual representation of the commit history of your code. It allows you to easily navigate through the commit history, compare commits, and view the diff for a commit. It is a powerful tool that can help you to understand your codebase better and to work more efficiently.

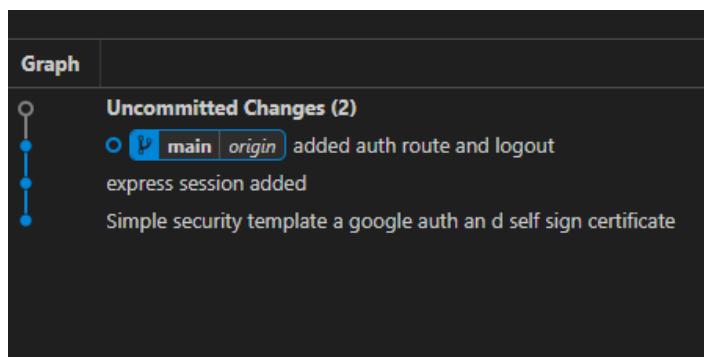
It stores history as a graph of snapshots of the entire repository. These snapshots, called commits in Git, can have multiple parents, creating a history that looks like a graph instead of a straight line.

It is a must-have for any developer who uses Visual Studio Code.

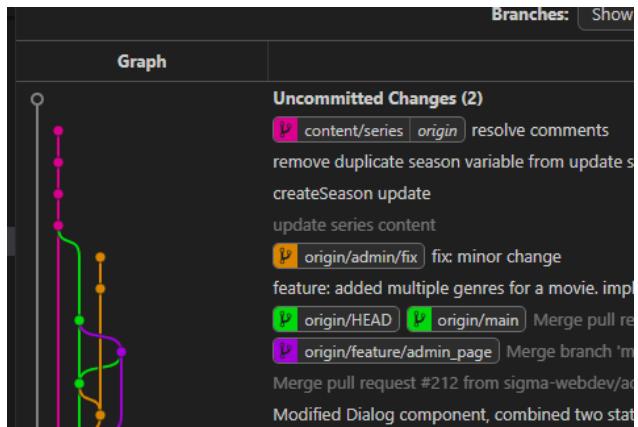
To use Git Graph click on the "Git Graph" icon on the Visual Studio Code activity bar as shown below -



If you have a single branch it will look something like this as shown below



If you have multiple branches it will look something like this as shown below



The screenshot shows a GitHub repository interface. At the top, there's a "Branches" dropdown with a "Show" button. Below it, a "Graph" tab is selected. On the left, a vertical timeline shows a series of commits represented by colored dots (purple, green, orange) connected by lines. To the right of the timeline, a list of "Uncommitted Changes (2)" is displayed:

- content/series | origin resolve comments  
remove duplicate season variable from update s  
createSeason update
- update series content  
origin/admin/fix fix: minor change  
feature: added multiple genres for a movie. imp  
origin/HEAD | origin/main Merge pull re  
origin/feature/admin\_page | Merge branch 'm  
Merge pull request #212 from sigma-webdev/ac  
Modified Dialog component, combined two stat