I. Written test

*Please explain in your own words

1. What is a JavaScript Framework and explain about Vue.js as one of the JavaScript Framework!
2. What is the use of ellipsis?
3. Explain animation properties below:
   a. @keyframe
   b. animation-name
   c. animation-duration
   d. animation-iteration
   e. animation-direction
4. Please explain how lazy load works in JavaScript!
5. Mention at least 5 git commands and describe each function of them!

Answer

1. JavaScript framework is a collection of JavaScript libraries and tools that help developers build web applications more efficiently. Frameworks typically provide features such as routing, templating, data binding, and event handling.

   Vue.js is a progressive JavaScript framework for building user interfaces. It is designed to be lightweight, flexible, and easy to learn. Vue.js is based on the Model-View-ViewModel (MVVM) architectural pattern, which separates the user interface (view) from the data model (model). This makes it easy to create reusable components and to keep our code organized.

   Vue.js is popular among developers for a number of reasons:
   - It is lightweight and fast.
   - It is easy to learn and use, even for beginners.
   - It is flexible and can be used to build a wide variety of web applications.
   - It has a large and active community of developers.

   Here are some of the key features of Vue.js:
   - Template syntax: Vue.js uses a simple and intuitive template syntax that is easy to learn and use.
   - Data binding: Vue.js provides two-way data binding, which means that changes to the data model are automatically reflected in the view, and vice versa.
   - Components: Vue.js components are reusable pieces of code that can be used to build complex user interfaces.
   - Routing: Vue.js provides a built-in routing system for navigating between different pages in our application.

   - Developer tools: Vue.js provides a variety of developer tools, such as the Vue DevTools, to help developers build and debug their applications.

   Vue.js is a powerful and versatile JavaScript framework that can be used to build a wide variety of web applications, from simple websites to complex single-page applications (SPAs). It is a good choice for developers of all skill levels, from beginners to experienced professionals.

2. The ellipsis (...) is a punctuation mark that's used to show that there's more text that's not being shown. We can use it in CSS to truncate text, which means to shorten it if it's too long to fit in the available space.

   To use the ellipsis in CSS, just add the text-overflow: ellipsis; property to the element that we want to truncate. We can also use the white-space: nowrap; property to prevent the text from wrapping to a new line.

   Here are some tips for using the ellipsis effectively:
   - Use it sparingly.
   - Make sure the truncated text is still meaningful and informative.
   - Consider using a tooltip or other hover effect to show the full text when users hover over the ellipsis.

   Here is a simple example of how to use the ellipsis to **truncate text to a maximum of 3 lines:**

   ```
   .container {
     width: 200px;
     overflow: hidden;
     text-overflow: ellipsis;
     display: -webkit-box;
     -webkit-line-clamp: 3;
     -webkit-box-orient: vertical;
   }
   ```

   The overflow: hidden; property is used to hide any text that overflows the container. The text-overflow: ellipsis; property is used to display the ellipsis at the end of the truncated text. The display: -webkit-box; and -webkit-line-clamp: 3; properties are used to ensure that the text is displayed on 3 lines or less.

3. CSS animations enable the transformation of CSS style setups from one to another through animated transitions. These animations comprise of two main parts: a style that defines the CSS animation itself and a series of keyframes that specify the initial and final styles of the animation, along with any potential intermediate points along the way.

a. @keyframes

   A keyframe is a pivotal point in the animation, whether in CSS or animation software, that defines the state or transformation of an element at a specific point in time during the animation. Keyframes determine how elements move, change, or interact over time within an animation.

b. animation-name

   The animation-name property is used to define the name of an animation. This name is then referenced in the CSS @keyframes rule to associate the animation with a set of keyframes that describe how the animation should behave. The animation-name property essentially links the animation rule to its @keyframes definition, so we can apply the animation to elements by specifying the animation name in other animation-related properties.

c. animation-duration

   The animation-duration property is all about determining how long an animation lasts. We can use values in seconds (s), milliseconds (ms), or even in simpler formats like "2s" (two seconds) or "500ms" (half a second).

d. animation-iteration

   The animation-iteration property is used to control how many times an animation repeats. We can set the value for `animation-iteration` in several ways:

   - infinite: The animation will repeat endlessly.
   - number: The animation will repeat the specified number of times. For example, `animation-iteration-count: 3;` will make the animation repeat three times.
   - alternate: The animation will alternate, meaning that after one iteration is complete, the element will return to its initial state and animate in the opposite direction.

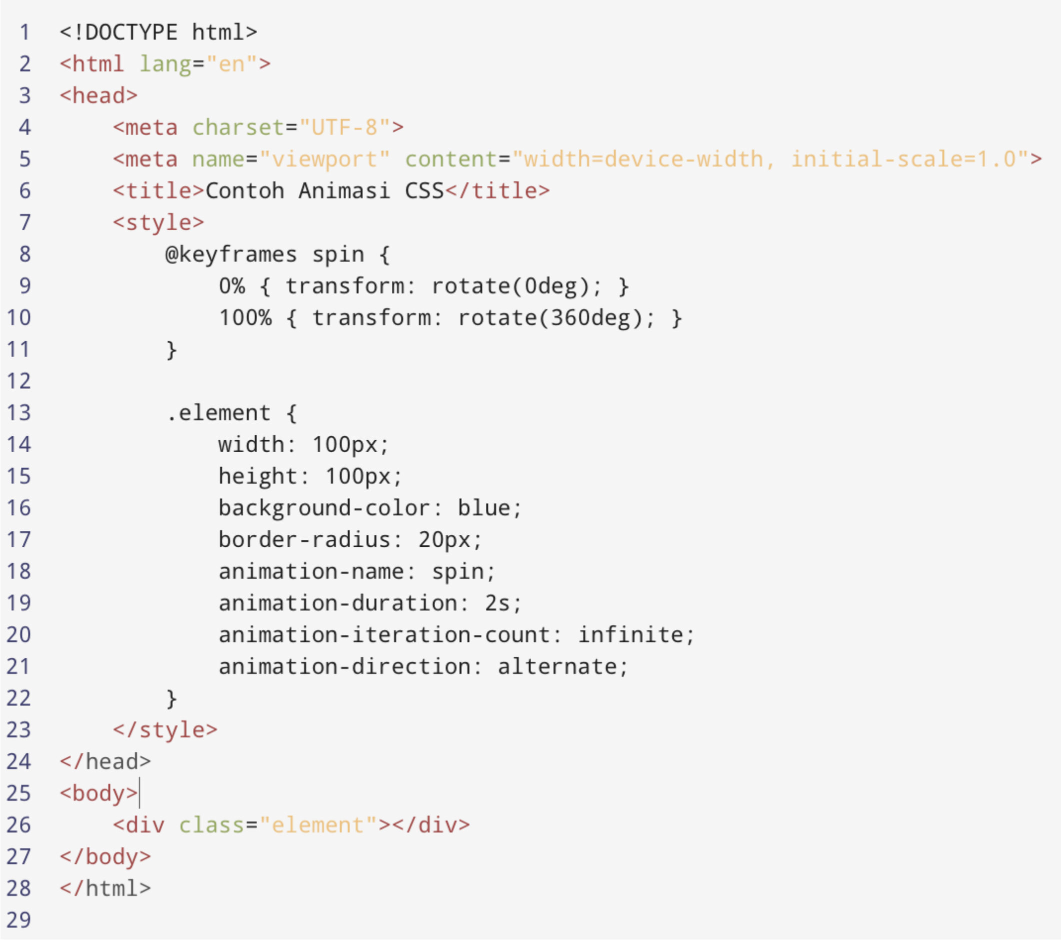
e. animation-direction

   The animation-direction property determines how the animation moves. Here are the options:

   - normal: The animation runs from start to finish.
   - reverse: The animation goes backward, from finish to start.
   - alternate: It changes direction every time it reaches the end, like a back-and-forth motion.
   - alternate-reverse: Similar to "alternate," but it starts in reverse and then alternates.

In addition to using "pure CSS," we can also utilize various CSS libraries like Animate.css to facilitate the creation of animations more easily and quickly. Furthermore, there's the lightweight and user-friendly Lottie JSON animation format for crafting more complex animations. With these tools, we have the flexibility to create animations that suit our web project needs more efficiently.

Example animation CSS:



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Contoh Animasi CSS</title>
    <style>
        @keyframes spin {
            0% { transform: rotate(0deg); }
            100% { transform: rotate(360deg); }
        }

        .element {
            width: 100px;
            height: 100px;
            background-color: blue;
            border-radius: 20px;
            animation-name: spin;
            animation-duration: 2s;
            animation-iteration-count: infinite;
            animation-direction: alternate;
        }
    </style>
</head>
<body>
    <div class="element"></div>
</body>
</html>

```

To view a live preview, you can also access this code at the following URL: https://bb.fauzanmustofa.my.id/animation-example/

4. Lazy loading is a way to load images and other content on a web page only when it's needed. This can make our web pages load faster, especially on mobile devices or when people have slow internet connections.

   Here's how it works:

   1. When a web page first loads, only the most important content is loaded. This could be things like the text and layout of the page.

2. As we scroll down the page, the browser loads the rest of the content, such as images and videos. This happens only when the content is close enough to be seen.

Lazy loading has a few benefits:

1. Faster loading times: Web pages with lazy loading load faster than pages without it. This is because the browser doesn't have to load all of the content at once.
2. Improved user experience: People are more likely to visit a website that loads quickly. Lazy loading can help us keep our visitors happy.
3. Reduced bandwidth usage: Lazy loading can save us money on bandwidth costs. This is because the browser only loads the content that people actually see.

There are a few things to keep in mind when using lazy loading:

1. Make sure our content is still accessible: Even though the content isn't loaded right away, it should still be accessible to people who use screen readers or other assistive technologies.
2. Test our website thoroughly: Make sure that our website works properly on all devices and browsers, even with lazy loading enabled.
3. Be careful with SEO: Search engines may have different ways of indexing lazily loaded content. Make sure that our content is still visible to search engines.

Overall, lazy loading is a great way to improve the performance and efficiency of our web pages. Just be sure to implement it carefully and test our website thoroughly.

Here is a simple example of how to lazy load images using JavaScript and the Fetch API:

HTML

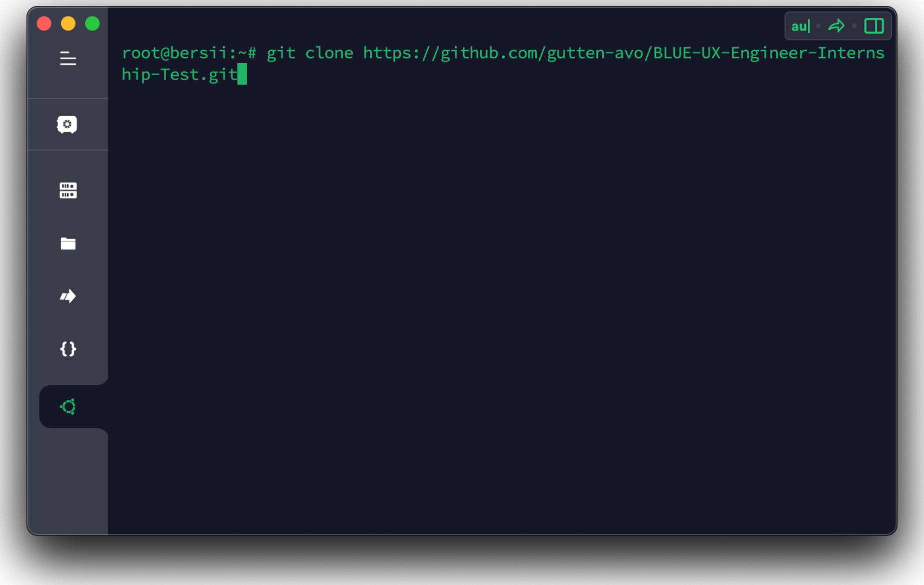
```
<img src="" data-src="https://example.com/image.jpg"
alt="Image">
```

JavaScript

```
window.addEventListener('scroll', function() {
  // Get all of the images with the data-src attribute.
                      const           images           =
document.querySelectorAll('img[data-src]');

   // Loop through the images and load any that are now
visible.
  for (const image of images) {
    // Check if the image is now visible in the viewport.
              if    (image.getBoundingClientRect().top    <
window.innerHeight) {
      // Load the image using the Fetch API.
      fetch(image.getAttribute('data-src'))
        .then(response => response.blob())
        .then(blob => {
          // Create a new Image object.
          const newImage = new Image();

          // Set the src attribute of the new Image object
to the blob.
          newImage.src = blob;

             // Replace the old image with the new Image
object.
          image.replaceWith(newImage);
        });
    }
  }
});
```

5. Several git commands
   - git clone
     Function: Clones an existing Git repository from a remote server (like GitHub, GitLab, or Bitbucket) to our local machine. It creates a copy of the entire repository, including its history and branches.

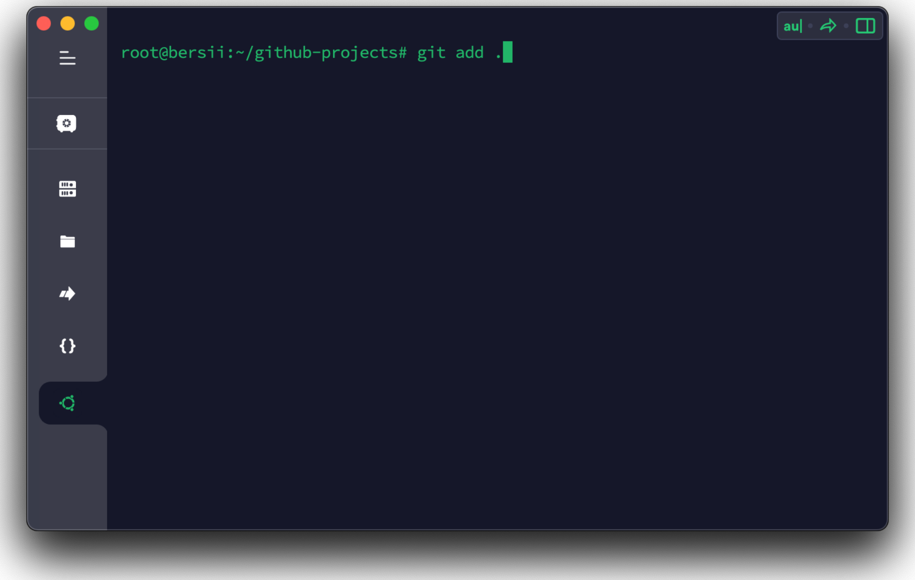
     Example to clone a git project on a Virtual Private Server using Termius:



```
root@bersii:~# git clone https://github.com/gutten-avo/BLUE-UX-Engineer-Internship-Test.git
```

   - git add
     Function: Stages changes in our working directory for commit. We can use this command to specify which files or changes we want to include

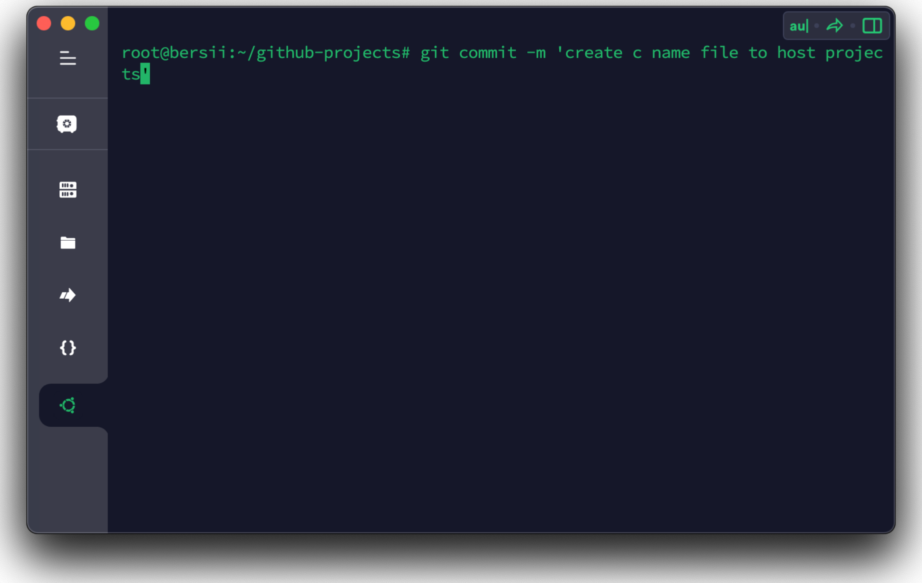
in the next commit. For example, git add filename stages a specific file, and git add. stages all changes in the current directory.
Example to add all changes to a git project on a Virtual Private Server using Termius:



```
root@bersii:~/github-projects# git add .
```

- git commit
  Function: Records a snapshot of the staged changes in our Git repository with a commit message. Each commit represents a point in the project's history, and the commit message describes the purpose or context of the changes.

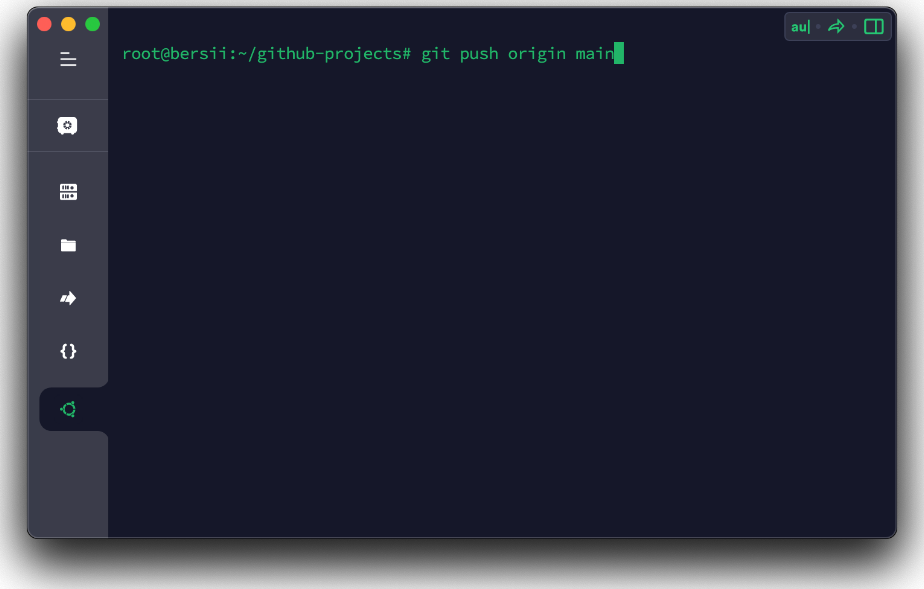
  Example to commit a git project on a Virtual Private Server using Termius:



```
root@bersii:~/github-projects# git commit -m 'create c name file to host projec
ts'
```

- git push
  Function: Pushes our local commits to a remote repository. This command is typically used to upload our changes to a central Git server, making them available to others who collaborate on the same project.

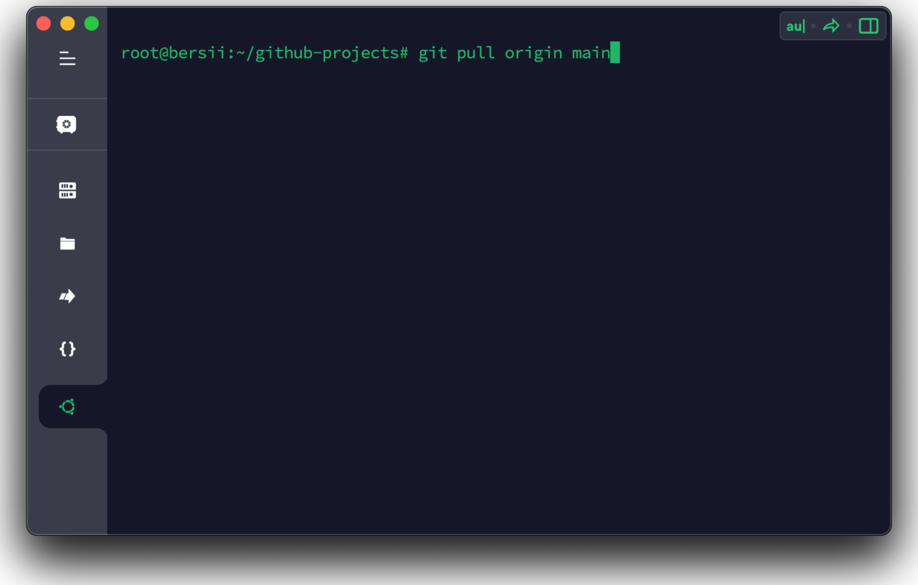
  Example to push all changes to a git project on a Virtual Private Server using Termius:

  ```
  root@bersii:~/github-projects# git push origin main
  ```

- git pull
  Function: Updates our local repository with changes from a remote repository. It fetches changes from the remote repository and merges them into our current branch. This command is useful for keeping our local repository up-to-date with the latest changes made by others in a collaborative environment.

  Example to add all changes from a git project to local on a Virtual Private Server using Termius:



```
root@bersii:~/github-projects# git pull origin main
```