Of course. For **Practical 4**, the objective is to demonstrate how you've used JavaScript to make your website interactive and dynamic. Your project has several excellent examples of this, including the collapsible FAQ, the tabbed interface, and the notification popups.

Here is a guide on what to write in your practical file for Practical 4.

Practical 4: Create Dynamic Content

Aim: To enhance the user experience on the ByteBazaar website by creating dynamic content modules such as collapsible FAQs, popups, and content tabs using JavaScript DOM manipulation and event handling.

1. Key Questions & Analysis

1.1. How is the DOM selected and manipulated?

The project uses standard JavaScript methods to select and manipulate Document Object Model (DOM) elements effectively.

- **Selection:** Elements are selected using document.getElementById() for unique IDs (e.g., contactForm) and document.querySelectorAll() for multiple elements (e.g., .faq-item, .tab-btn).
- **Manipulation:**
  - **CSS Classes:** The classList.toggle() and classList.add/remove() methods are used to dynamically add or remove CSS classes, which control the visibility and styling of elements like the collapsible FAQ answers and active tabs.
  - **Content Injection:** The innerHTML property is used to dynamically insert content, such as rendering different product views or displaying error messages.
  - **Event Listeners:** The addEventListener() method is used to listen for user interactions, primarily the click event, to trigger these manipulations.

1.2. Are events and listeners properly handled?

Yes. Event listeners are attached to the relevant elements to create an interactive experience.

- **Collapsible FAQ:** An event listener is attached to each FAQ question (.faq-question) on the contact.html page. When a question is clicked, it toggles an active class on its parent (.faq-item), which then reveals the answer using CSS transitions.

- **Product Details Tabs:** On the product-details.html page, click listeners on the tab buttons (.tab-btn) dynamically show and hide the corresponding content panes (.tab-pane) by managing an active class.

Example of FAQ Event Handling from js/contact.js:

JavaScript

```
// From js/contact.js
function initializeFAQ() {
    const faqItems = document.querySelectorAll('.faq-item');
    faqItems.forEach(item => {
        const question = item.querySelector('.faq-question');
        question.addEventListener('click', () => {
            // Closes other open FAQ items for a clean accordion effect
            faqItems.forEach(otherItem => {
                if (otherItem !== item && otherItem.classList.contains('active')) {
                    otherItem.classList.remove('active');
                }
            });
            // Toggles the current item
            item.classList.toggle('active');
        });
    });
}
```

1.3. How is interactivity enhancing usability?

The interactive elements significantly improve the website's usability.

- **Reduces Clutter:** The collapsible FAQ and tabbed content sections allow a large amount of information to be presented in a compact and organized way, preventing the user from being overwhelmed.
- **Provides Feedback:** Dynamic notifications provide immediate feedback to the user after actions like adding an item to the cart or submitting a form.
- **Improves Navigation:** The mobile hamburger menu is a crucial interactive element that makes the site navigable on small screens.

2. Notification Popup Banner (Supplementary Problem)

The project includes a custom notification system that fulfills the supplementary problem requirement. A JavaScript function, showNotification(), dynamically creates a notification element, appends it to the DOM, and styles it. It is used across the site to confirm actions.

Example of Notification Function from js/script.js:

JavaScript

```
// From js/script.js
function showNotification(message) {
    const notification = document.createElement('div');
    notification.className = 'notification';
    notification.textContent = message;

    // Dynamically sets CSS for animation and appearance
    notification.style.cssText = `...`;

    document.body.appendChild(notification);

    // Animate in and out
    setTimeout(() => {
        notification.style.opacity = '1';
        notification.style.transform = 'translateX(0)';
    }, 100);

    setTimeout(() => {
        // ... code to remove the notification ...
    }, 3000);
}
```

3. Testing of Dynamic Modules (Post-Laboratory Work)

You should include screenshots that demonstrate the interactive modules in different states.

Test Case 1: Collapsed vs. Expanded FAQ

[Insert two side-by-side screenshots of the FAQ section on contact.html, one with all answers collapsed and one with an answer expanded.]

Test Case 2: Active Product Tab

[Insert a screenshot of the product-details.html page, highlighting one of the tabs (e.g., "Specifications") as active and showing its content.]

Test Case 3: Notification Popup

[Insert a screenshot of any page showing the "Product added to cart!" notification popup in the top-right corner.]

This documentation provides strong evidence of your ability to manipulate the DOM and handle events to create a dynamic and user-friendly website, fully covering the requirements of Practical 4.