

## Тест-кейсы

Примечания к тестам:

- Используется библиотека `@testing-library/react` для рендеринга и взаимодействия с компонентами.
- В тестах используется `jest.mock` для эмуляции обработчика событий.

Тест-кейс для AllNews компонента:

<b>Цель тест-кейса:</b> Проверить корректность отображения новостей в компоненте AllNews, а также функционал загрузки дополнительных новостей.	
renders without crashing	<p>Действие: Рендер компонента AllNews с пустым списком новостей.</p> <p>Ожидаемый результат: Компонент успешно рендерится без ошибок.</p>
fetches articles and renders NewsPreviewList	<p>Действие: Подмена fetch-запроса для возврата моковых новостей. Рендер компонента AllNews с заполненным списком новостей.</p> <p>Ожидаемый результат: Компонент успешно загружает и отображает новости с соответствующими заголовками, изображениями и датами публикации.</p>
renders LoadMoreButton if there are more articles	<p>Действие: Подмена fetch-запроса для возврата большого количества новостей (15).</p> <p>Ожидаемый результат: Компонент отображает кнопку "Load More", так как количество новостей превышает лимит для её отображения.</p>
does not render LoadMoreButton if there are no more articles	<p>Действие: Нажатие на кнопку "Load More" после загрузки большого количества новостей.</p> <p>Ожидаемый результат: Компонент загружает и отображает дополнительные новости, а также скрывает</p>

	кнопку "Load More", если больше нет новостей для загрузки.
--	--

## Тест-кейс для компонента Button

<b>Цель тест-кейса:</b> Проверить корректность отображения и функциональность компонента Button.	
renders with default primary color	<p>Действие: Рендер компонента Button без указания цвета.</p> <p>Ожидаемый результат: Кнопка успешно рендерится с классами "button" и "primary".</p>
renders with specified color (secondary)	<p>Действие: Рендер компонента Button с указанным цветом (secondary).</p> <p>Ожидаемый результат: Кнопка успешно рендерится с классами "button" и "secondary".</p>
calls onClick handler when clicked	<p>Действие: Рендер компонента Button с обработчиком события onClick. Эмуляция клика на кнопку.</p> <p>Ожидаемый результат: Обработчик события onClick вызывается один раз.</p>

Тест-кейс для компонента Footer

<b>Цель тест-кейса:</b> Проверить корректность отображения и функциональность компонента Footer.	
renders without crashing	Действие: Рендер компонента Footer.  Ожидаемый результат: Компонент успешно отрисовывается без ошибок.
sets body dataset theme on theme change	Действие: Эмуляция клика на кнопку изменения темы.  Ожидаемый результат: Значение dataset.theme элемента body устанавливается в "dark".
changes theme on button click	Действие: Эмуляция клика на кнопку изменения темы.  Ожидаемый результат: Значение текста на кнопке изменяется с "light" на "dark".

Тест-кейс для компонента LoadMoreButton

<b>Цель тест-кейса:</b> Проверить корректность отображения и функциональность компонента LoadMoreButton.	
renders without crashing	Действие: Рендер компонента LoadMoreButton.  Ожидаемый результат: Компонент успешно отрисовывается без ошибок.
calls onClick handler when clicked	Действие: Эмуляция клика на кнопку "Load More".  Ожидаемый результат: Обработчик события (onClickMock) вызывается ровно один раз.

## Тест-кейсы для компонента Main

<b>Цель тест-кейса:</b> Проверить корректность отображения и функциональность компонента Main.	
renders without crashing	<p>Действие: Рендер компонента Main.</p> <p>Ожидаемый результат: Компонент успешно отрисовывается без ошибок.</p>
displays default content if no search or sort value provided	<p>Действие: Рендер компонента Main без предоставления значений для поиска и сортировки.</p> <p>Ожидаемый результат: Компонент отображает контент по умолчанию с текстом "Default Content".</p>
updates search value on input change	<p>Действие: Эмуляция ввода текста "Bitcoin" в поле поиска.</p> <p>Ожидаемый результат: Значение поля поиска обновляется и содержит текст "Bitcoin".</p>
updates sort value on select change	<p>Действие: Эмуляция выбора опции "Latest" в поле сортировки.</p> <p>Ожидаемый результат: Значение поля сортировки обновляется и содержит значение "latest".</p>


## Тест-кейсы для компонента NewsPreview

<b>Цель тест-кейса:</b> Проверить, что NewsPreview правильно рендерится и взаимодействует с модальным окном.	
renders without crashing	<p>Действие: Рендер компонента NewsPreview с моковыми данными новостей.</p> <p>Ожидаемый результат: Компонент NewsPreview успешно рендерится без ошибок.</p>
opens and closes the modal on click	<p>Действие: Проверить, что модальное окно с текстом "Хотите перейти по ссылке?" изначально не отображается. И успешно закрывается</p> <p>Ожидаемый результат: Модальное окно с текстом "Хотите перейти по ссылке?" успешно открывается. Модальное окно не отображается после закрытия.</p>

## Тест-кейсы для компонента NewsPreviewList

<b>Цель тест-кейса:</b> Проверить корректность отображения и передачу свойств компоненту NewsPreviewList.	
renders news list correctly	<p>Действие: Рендер компонента NewsPreviewList с моковыми данными новостей.</p> <p>Ожидаемый результат: Компонент успешно отрисовывается, количество элементов списка соответствует количеству моковых новостей.</p>
passes correct props to NewsPreview component	<p>Действие: Рендер компонента NewsPreviewList с моковыми данными новостей.</p> <p>Ожидаемый результат: Каждый элемент списка (компонент NewsPreview) получает корректные свойства, такие как url, urlToImage, id, title, и publishedAt.</p>
renders news list correctly	<p><b>БАГ</b></p> <p>Ключи в React должны быть уникальными внутри списка элементов. В данном случае, в качестве ключа используется news.title, что может стать проблемой, если будут новости с одинаковыми заголовками.</p> <p>Например:</p> <pre>const mockNews = [   {     title: "Test News 1",     url: "https://example.com/news1",     urlToImage: "https://example.com/image1.jpg",     publishedAt: "2023-01-01",   },   {</pre>



	<pre>title: "Test News 1", // Одинаковый заголовок url: "https://example.com/news2", urlToImage: "https://example.com/image2.jpg", publishedAt: "2023-01-02", }, ];</pre> <p>Эти данные при тестировании компонента дадут ошибку.</p> <div> Warning: Each child in an array or iterator should have a unique "key" prop. <a href="#">bundle.js:22874</a></div>
--	---

## Тест-кейсы для компонента TabItem

**Цель тест-кейса:** Проверить, что компонент TabItem взаимодействует с пользовательским вводом и вызывает функцию onTabClick с правильными параметрами.

calls onTabClick  
when the button  
is clicked

Действие: Рендер компонента TabItem с моковыми данными.

Ожидаемый результат: При клике на вкладке "Test Tab" вызывается функция onTabClick с моковыми данными вкладки.

calls onTabClick  
when the button  
is clicked

Действие: Рендер компонента TabItem с моковыми данными.

Ожидаемый результат: При клике на вкладке "Test Tab" с использованием `userEvent.click`, функция onTabClick вызывается с моковыми данными вкладки.

## Тест-кейсы для компонента Typography

<b>Цель тест-кейса:</b> Проверить, что компонент Typography правильно рендерит текст с учетом заданных входных параметров.	
renders default variant and color correctly	<p>Действие: Рендер компонента Typography с текстом "Hello, World!" без указания варианта и цвета.</p> <p>Ожидаемый результат: Текст "Hello, World!" отображается внутри элемента <code>&lt;p&gt;</code> с классом "primary".</p>
renders specified variant correctly	<p>Действие: Рендер компонента Typography с текстом "Heading" и вариантом "h2".</p> <p>Ожидаемый результат: Текст "Heading" отображается внутри элемента <code>&lt;h2&gt;</code>.</p>
renders specified color correctly	<p>Действие: Рендер компонента Typography с текстом "Secondary Text" и цветом "secondary".</p> <p>Ожидаемый результат: Текст "Secondary Text" отображается с классом "secondary".</p>
applies custom className correctly	<p>Действие: Рендер компонента Typography с текстом "Custom Text" и дополнительным классом "customClass".</p> <p>Ожидаемый результат: Текст "Custom Text" отображается с классом "customClass".</p>
renders children correctly	<p>Действие: Рендер компонента Typography с дочерним элементом <code>&lt;span&gt;</code> содержащим текст "Child Span".</p> <p>Ожидаемый результат: Текст "Child Span" отображается внутри элемента <code>&lt;span&gt;</code>.</p>