УДК 004.89

Я.А. Игнаткова, ст. преп;
Н.П. Шутько, доц., канд. техн. наук (БГТУ, г. Минск)

ОСОБЕННОСТИ КАСТОМИЗАЦИИ ЭЛЕМЕНТОВ HTML-ФОРМ СРЕДСТВАМИ CSS

Веб-формы долгое время были проблемой для фронтенд-разработчиков, стремящихся создать эстетические и функциональные интерфейсы. Нативные элементы, такие как чекбоксы и выпадающие списки, сложно кастомизировались с помощью CSS, что требовало применения JavaScript или замены семантических HTML-элементов на несемантические. Современный CSS, однако, предоставил мощные инструменты для стилизации форм, не жертвуя доступностью и производительностью.

Цель данного исследования заключается в анализе возможностей современных инструментов CSS для разработки сложных, эстетически привлекательных и интерактивных элементов форм, которые ранее не могли быть реализованы без обязательного применения JavaScript.

Суть современной кастомизации форм на CSS заключается в эффективном использовании семантической HTML-разметки, псевдоклассов (:checked, :focus, :hover и др.) [1] и псевдоэлементов (::before, ::after) [2] для создания нового визуального слоя, а также продвинутых CSS-свойств (appearance, clip-path, transition и transform). Основной принцип состоит в обоснованном скрытии нативных элементов и их визуальной «подмене», что сохраняет функциональность и доступность для ассистивных технологий. Данный подход принципиально отличается от устаревших методов.

Можно выделить следующие способы и техники использования CSS для стилизации элементов форм:

– использование тегов <input> и <label> с соседними селекторами;

– сокрытие исходного элемента с сохранением функциональности при помощи свойств opacity и position;

– использование псевдоэлементов(::before, ::after) для создания нового элемента;

– отслеживание состояния через псевдоклассы :checked, :focus, :focus-within;

– сброс и переопределение системного вида элемента;

– создание сложных анимаций и переходов;

– работа с свойством clip-path для нестандартных форм.

Данные способы могут быть использованы как самостоятельно, так и в различных комбинациях. Далее кратко рассмотрим каждый из них.

Связывание элементов <input> и <label> с использованием соседних селекторов соседства основой кастомизации форм. Связь элемента <label> с элементом <input> может осуществляться либо через вложение, либо с помощью атрибутов for и id. При активации элемента <label> браузер переводит фокус на связанный элемент <input> или изменяет его состояние.

Такие CSS-селекторы позволяют применять стили к элементам в зависимости от их положения относительно <input>. Эти селекторы позволяют модифицировать визуальное представление кастомных компонентов в зависимости от состояния скрытого нативного элемента формы (рис. 1).

Рисунок 1 – Стилизованный чекбокс

Методы визуального скрытия нативных элементов форм. После установления связи между элементами требуется устранить видимое представление нативного элемента формы при сохранении его функциональных характеристик. Наиболее распространенным методом является применение комбинации CSS-свойств opacity: 0 и position: absolute (рис. 2).

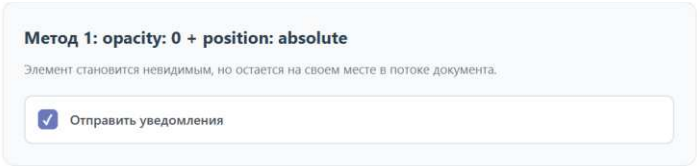

Рисунок 2 – Методы визуального скрытия нативных элементов форм

Свойство opacity: 0 обеспечивает полную прозрачность элемента, сохраняя его присутствие в DOM-дереве и все интерактивные возможности. Элемент остается доступным для устройств ввода, программ чтения с экрана и сохраняет возможность получения фокуса. Свойство position: absolute исключает элемент из обычного потока документа, предотвращая влияние невидимого элемента на компоновку интерфейса.

Альтернативные методики включают установку нулевых значений для свойств width и height, применение свойства clip-path для гео-

метрического обрезания элемента или использование свойства visibility: hidden. Выбор конкретного метода определяется требованиями к доступности и особенностями компоновки интерфейса.

Применение псевдоэлементов ::before и ::after для построения интерфейсных компонентов. Данные псевдоэлементы представляют собой виртуальные элементы, генерируемые исключительно средствами CSS. Каждый элемент DOM может содержать до двух таких псевдоэлементов, которые располагаются соответственно перед и после его основного содержимого. Стилизация данных элементов осуществляется стандартными CSS-свойствами, включая возможности трансформации и анимации.

Обработка состояний элементов посредством CSS-псевдоклассов. CSS-псевдоклассы состояния позволяют применять стили к элементам при наступлении определенных условий без необходимости использования JavaScript. Данный механизм является критически важным для создания интерактивных компонентов форм.

Псевдокласс :checked активируется при установке флажка в чекбоксах или выборе варианта в радиокнопках. Данное состояние используется для изменения визуального представления кастомных компонентов, ассоциированных с данными элементами.

Псевдокласс:focus применяется к элементам, получившим фокус ввода. Обеспечение видимого индикатора фокуса является обязательным требованием доступности. Данный псевдокласс позволяет кастомизировать отображение индикатора в соответствии с дизайн-системой.

Псевдокласс :focus-within применяется к элементам, содержащим вложенные элементы в состоянии фокуса. Это позволяет изменять стилизацию элементов при активации содержащихся в них полей ввода.

Дополнительные псевдоклассы включают :hover (наведение указателя), :disabled (неактивное состояние), :valid и :invalid (состояния валидации), :placeholder-shown (отображение текста-заполнителя) (рис. 3).

Отключение системного оформления элементов посредством свойства appearance. Свойство appearance предоставляет контроль над применением системно-зависимых стилей оформления к элементам форм. Значение none полностью отключает браузерное оформление элемента, позволяя осуществлять его полную кастомизацию средствами CSS. Свойство accent-color представляет собой альтернативный подход, позволяющий изменить только акцентные цвета элементов форм без полного переопределения их оформления.

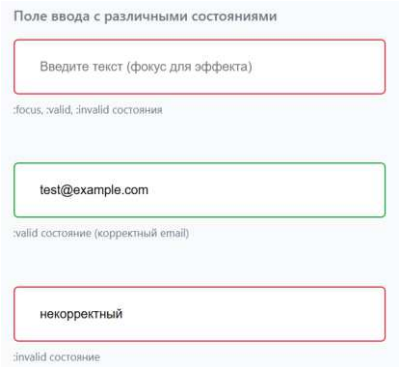

Рисунок 3 – Различные состояния полей ввода

Реализация анимационных эффектов. Свойство transition определяет параметры плавного перехода между значениями CSS-свойств. Оно специфицирует свойства для анимации, продолжительность перехода, функцию временной шкалы и возможную задержку начала. Данный механизм оптимизирован браузерами для аппаратного ускорения.

Типичные применения включают плавное изменение цвета фона при наведении указателя, постепенное появление индикаторов состояния, анимированное перемещение текстовых меток.

Использование расширенных CSS-свойств для создания сложных визуальных эффектов. Свойство clip-path определяет область отображения элемента путем задания контура отсечения. Контур может быть задан посредством координат точек, базовых геометрических фигур или функций path() для произвольных SVG-путей. Данное свойство позволяет создавать элементы нестандартной геометрии без применения растровых изображений.

Таким образом, возможно создавать уникальные элементы интерфейса, соответствующие дизайн-системам, без избыточного использования JavaScript, сохраняя семантику и доступность. Это обеспечивает не только эстетические преимущества, но и практическую выгоду: снижение объема кода, улучшение производительности и упрощение поддержки. Методы, представленные в исследовании, свидетельствуют о размывании границ между стандартными и кастомными элементами.

ЛИТЕРАТУРА

1. CSS Basic User Interface Module Level 3: W3C Candidate Recommendation Draft, 21 February 2023. – URL: https://www.w3.org/TR/css-ui-3/ (дата доступа: 18.02.2026).

2. Styling web forms: руководство / Mozilla Developer Network. – URL: https://developer.mozilla.org/en-US/docs/Learn/Forms/Styling_web_forms (дата доступа: 19.02.2026).