

Technischer Bericht

Bugfix: Android — Animationen & neues CSS-System fuer Links

Projekt: Digital & Mobil in Deutschland

Datum: 21. April 2026

1. Problembeschreibung

1.1 Rechteckiger Highlight bei Android-Geraeten

Beim Antippen von Schaltflaechen (.nav-btn, .btn-main) auf Android-Geraeten wurde ein rechteckiger nativer Highlight-Rahmen angezeigt, der die border-radius-Rundungen der Elemente ignorierte. Auf iOS trat dieses Problem nicht auf.

1.2 Fehlende Tap-Animation bei a.btn-main auf Android

Bei Schaltflaechen vom Typ <a href> mit Klasse .btn-main wurde auf Android beim Antippen keine :active-Animation ausgeloeset. Der Browser navigierte sofort zur Zielseite. Auf iOS funktionierte die Animation korrekt (mit 180ms Verzoegerung, dokumentiert im vorherigen Bericht).

1.3 Animation beim Scrollen durch das Navigationsmenue

Beim Scrollen durch das horizontale Navigationsmenue wurden .nav-btn-Schaltflaechen auf Android unbeabsichtigt hervorgehoben, obwohl kein gezielter Tap erfolgte.

1.4 Keine einheitliche Tap-Rueckmeldung bei einfachen Links

Einfache <a href>-Links im Projekt hatten individuelle Inline-Styles, aber kein einheitliches visuelles Feedback beim Antippen und keinen Schutz vor dem rechteckigen Android-Highlight.

2. Durchgefuehrte Korrekturen

2.1 Entfernung des nativen Android-Highlights (.nav-btn, .btn-main)

In navbar.css und components.css wurden folgende Eigenschaften hinzugefuegt:

- -webkit-tap-highlight-color: transparent
- isolation: isolate
- -webkit-mask-image: -webkit-radial-gradient(white, black)
- user-select: none

2.2 Shine-Sweep-Animation: Umstellung von left auf transform

Die bisherige Implementierung nutzte left: -100% fuer die Glanzlicht-Animation (::before / ::after). Ersetzt durch:

```
opacity: 0 + transform: translateX(-100%) /* Ausgangszustand */
transform: translateX(160%) + opacity: 1 /* Aktiv-Zustand */
border-radius: inherit /* auf ::before */
```

transform arbeitet auf einem bereits beschnittenen GPU-Layer und vermeidet den rechteckigen Artefakt.

2.3 JavaScript-Fix fuer a.btn-main auf Android

Das :active-Pseudoelement wird von Android Chrome bei <a href>-Links nicht zuverlaessig ausgeloeset. Loesung: Klasse .is-active via touchstart-Event in tabs.js:

```
touchstart → this.classList.add('is-active') // Animation sofort
starten
touchmove → this.classList.remove('is-active') // Scrollen:
Animation stoppen
touchend → 180ms Verzoeigerung, dann Navigation + Klasse entfernen
```

In components.css wurde ergaenzt:

```
.btn-main:active, .btn-main.is-active { transform: scale(0.97)
translateY(0); }
.btn-main.is-active::before { transform: translateX(160%); opacity:
1; }
```

2.4 Neuer CSS-Klasse .btn-link und .btn-link-group

Fuer einfache <a href>-Links wurde ein einheitliches CSS-System in components.css eingefuehrt:

- .btn-link: overflow: hidden, isolation: isolate, -webkit-mask-image, -webkit-tap-highlight-color: transparent, user-select: none, transition: opacity 0.15s ease, transform 0.15s ease
- .btn-link.is-active: opacity: 0.75, transform: scale(0.96)
- .btn-link-group: display: flex, gap: 10px, flex-wrap: wrap

Individuelle Farben und box-shadow werden weiterhin per style="" im HTML gesetzt.

2.5 JavaScript fuer .btn-link in tabs.js

Analog zu a.btn-main wurde fuer a.btn-link folgende Logik ergaenzt:

```
touchstart → classList.add('is-active')
touchmove (> 8px Verschiebung) → classList.remove('is-active')
touchend → setTimeout 180ms, dann classList.remove('is-active')
```

Kein e.preventDefault() — Navigation erfolgt nativ ohne Verzoeigerung.

3. Ergebnis

- Kein rechteckiger Highlight auf Android bei `.nav-btn` und `.btn-main`
- Tap-Animation bei `a.btn-main` funktioniert auf Android und iOS
- Scrollen loest keine ungewollten Animationen aus
- Einheitliches CSS/JS-System fuer alle einfachen Links via `.btn-link`

4. Geaenderte Dateien

- `css/navbar.css` — `-webkit-tap-highlight-color`, `isolation`, `-webkit-mask-image`, transform-basierte Shine-Animation
- `css/components.css` — gleiche Korrekturen fuer `.btn-main`, neue Klassen `.btn-link` und `.btn-link-group`
- `js/tabs.js` — `touchstart/touchmove/touchend` fuer `a.btn-main` (`.is-active`) und `a.btn-link` (`.is-active`)

Технический отчёт

Исправление: Android — анимации & новая CSS-система для ссылок

Проект: *Digital & Mobil in Deutschland*

Дата: 21 апреля 2026

1. Описание проблем

1.1 Прямоугольный highlight на Android

При нажатии на кнопки (.nav-btn, .btn-main) на Android-устройствах браузер отображал нативный прямоугольный highlight поверх элемента, игнорируя border-radius. На iOS проблема отсутствовала.

1.2 Анимация a.btn-main не срабатывала на Android

При нажатии на кнопки типа <a href> с классом .btn-main на Android анимация :active не воспроизводилась — браузер сразу переходил на новую страницу. На iOS анимация работала корректно.

1.3 Анимация при прокрутке меню

При скролле через навигационное меню кнопки .nav-btn на Android подсвечивались случайно, хотя пользователь просто листал.

1.4 Отсутствие единой визуальной реакции у обычных ссылок

Простые <a href>-ссылки имели индивидуальные inline-стили, но не имели единой визуальной реакции на нажатие и защиты от прямоугольного highlight на Android.

2. Выполненные исправления

2.1 Удаление нативного highlight на Android (.nav-btn, .btn-main)

Добавлены следующие CSS-свойства в navbar.css и components.css:

- -webkit-tap-highlight-color: transparent
- isolation: isolate
- -webkit-mask-image: -webkit-radial-gradient(white, black)
- user-select: none

2.2 Shine-sweep-анимация: замена left на transform

Прежняя реализация использовала left: -100% для анимации блика (::before / ::after).
Заменено на:

```
opacity: 0 + transform: translateX(-100%) /* начальное состояние */  
transform: translateX(160%) + opacity: 1 /* активное состояние */  
border-radius: inherit /* на ::before */
```

transform работает на уже обрезанном GPU-слое и исключает прямоугольный артефакт.

2.3 JavaScript-исправление для a.btn-main на Android

Android Chrome ненадёжно запускает :active для <a href>-ссылок. Решение: класс .is-active через touchstart в tabs.js:

```
touchstart → this.classList.add('is-active') // анимация стартует  
мгновенно  
touchmove → this.classList.remove('is-active') // скролл: анимация  
отменяется  
touchend → 180ms задержка, затем навигация + класс удаляется
```

Добавлено в components.css:

```
.btn-main:active, .btn-main.is-active { transform: scale(0.97)  
translateY(0); }  
.btn-main.is-active::before { transform: translateX(160%); opacity:  
1; }
```

2.4 Новые CSS-классы .btn-link и .btn-link-group

Для простых <a href>-ссылок введена единая система в components.css:

- .btn-link: overflow: hidden, isolation: isolate, -webkit-mask-image, -webkit-tap-highlight-color: transparent, user-select: none, transition: opacity 0.15s ease, transform 0.15s ease
- .btn-link.is-active: opacity: 0.75, transform: scale(0.96)
- .btn-link-group: display: flex, gap: 10px, flex-wrap: wrap

Индивидуальные цвета и box-shadow по-прежнему задаются через style="" напрямую в HTML.

2.5 JavaScript для .btn-link в tabs.js

Аналогично a.btn-main добавлена логика для a.btn-link:

```
touchstart → classList.add('is-active')  
touchmove (> 8px) → classList.remove('is-active')  
touchend → setTimeout 180ms, затем classList.remove('is-active')
```

Без e.preventDefault() — навигация происходит нативно без задержки.

3. Результат

- Нет прямоугольного highlight на Android для .nav-btn и .btn-main
- Тап-анимация a.btn-main работает на Android и iOS
- Скролл не запускает случайные анимации
- Единая CSS/JS-система для всех обычных ссылок через .btn-link

4. Изменённые файлы

- css/navbar.css — -webkit-tap-highlight-color, isolation, -webkit-mask-image, transform-базирующая Shine-анимация
- css/components.css — те же исправления для .btn-main, новые классы .btn-link и .btn-link-group
- js/tabs.js — touchstart/touchmove/touchend для a.btn-main (.is-active) и a.btn-link (.is-active)