

Page: Scan

Page Spec: Scan QR

Route

/scan

Architecture Status

Core

Figma Reference

scan.png

Visual Description from Figma

The scan page is a full-screen dark camera-style view for QR code scanning:

- **Top bar:** Left side has a white back arrow (←). Center has "Scan QR" in white bold text. No right-side elements.
- **Camera viewport:** Full-screen dark background (#1A1A1A or near-black) simulating a camera feed.
- **Scanner viewfinder:** Large centered square/rectangle with:
 - Dashed gray border (#6B7280 approx) forming the full perimeter
 - Four green corner brackets (thick, #0B6E35) at each corner — top-left, top-right, bottom-left, bottom-right
 - A horizontal green scanning line across the middle of the viewfinder (animated or static)
- **Instruction text:** Near the bottom, white/light text: "Pek kameraet mot butikkens QR-kode" — muted, centered.

- **Simulate button:** Full-width green button at the very bottom: "Simuler skanning" — dark green background (`#0B6E35`), white text, rounded corners. This is a demo-only button for simulating a QR scan without a real camera.
- **No bottom navigation** — this is a full-screen overlay/modal page.

Page Layout

```
Full-screen dark (no bottom nav – camera overlay)
├─ Top Bar (over dark bg)
|   ├─ Left: Back arrow (white)
|   └─ Center: "Scan QR" title (white, bold)
├─ Camera Viewport (dark bg, full screen)
|   └─ Scanner Viewfinder (centered)
|       ├─ Dashed border (gray)
|       ├─ Green corner brackets (4 corners)
|       └─ Green scan line (horizontal, centered)
├─ Instruction Text
|   └─ "Pek kameraet mot butikkens QR-kode"
└─ Simulate Button (demo only)
    └─ "Simuler skanning" (green, full width)
```

Components

Top Bar

```
<div className="flex items-center justify-between px-6 pt-6">
  <button onClick={() => router.back()} className="rounded-lg p-2 hover:bg-white/10">
    <ArrowLeft className="h-5 w-5 text-white" />
  </button>
  <h1 className="text-lg font-bold text-white">Scan QR</h1>
  <div className="w-9" /> { /* spacer for centering */ }
</div>
```

Scanner Viewfinder

```

<div className="relative mx-auto aspect-square w-[75%] max-w-[320px]">
  {/* Dashed border */}
  <div className="absolute inset-0 rounded-2xl border-2 border-dashed border-[#6B7280]" />

  {/* Green corner brackets – top-left */}
  <div className="absolute left-0 top-0 h-10 w-10 rounded-tl-2xl border-l-4 border-t-4 border-
[#0B6E35]" />
  {/* Green corner brackets – top-right */}
  <div className="absolute right-0 top-0 h-10 w-10 rounded-tr-2xl border-r-4 border-t-4
border-[#0B6E35]" />
  {/* Green corner brackets – bottom-left */}
  <div className="absolute bottom-0 left-0 h-10 w-10 rounded-bl-2xl border-b-4 border-l-4
border-[#0B6E35]" />
  {/* Green corner brackets – bottom-right */}
  <div className="absolute bottom-0 right-0 h-10 w-10 rounded-br-2xl border-b-4 border-r-4
border-[#0B6E35]" />

  {/* Scan line (centered horizontal) */}
  <div className="absolute left-4 right-4 top-1/2 h-0.5 -translate-y-1/2 bg-[#0B6E35]" />
</div>

```

Instruction Text

```

<p className="text-center text-sm text-[#9CA3AF]">
  Pek kameraet mot butikkens QR-kode
</p>

```

Simulate Button (Demo Only)

```

<button
  onClick={handleSimulateScan}
  className="mx-6 mb-8 h-12 w-[calc(100%-48px)] rounded-xl bg-[#0B6E35] text-sm font-semibold
text-white transition-colors hover:bg-[#095C2C]"
>
  Simuler skanning
</button>

```

Data Displayed

Data	Source	API
None pre-loaded	—	—
On scan success	QR payload (merchant ID, amount)	POST <code>/api/transactions</code> (type: <code>qr_payment</code>)

User Interactions

Element	Action	Result
Back arrow	Click	Navigate back (<code>router.back()</code>)
Camera viewfinder	Scan QR code	Parse QR payload → navigate to payment confirmation
"Simuler skanning" button	Click	Simulate QR scan with mock merchant data → navigate to payment confirmation

QR Scan Flow (Demo)

1. User taps "Simuler skanning"
2. App generates mock QR payload: `{ merchantId: "REMA1000-001", merchantName: "Rema 1000", amount: 189 }`
3. Navigate to confirmation screen (inline or modal) showing merchant name + amount
4. User confirms → POST `/api/transactions` with type `qr_payment`, PISP initiates payment from linked bank account
5. On success → redirect to `/dashboard` with success toast

QR Scan Flow (Production — Future)

1. Camera activates, user points at merchant QR code
2. QR library decodes payload (merchant ID, amount, reference)
3. Same confirmation + PISP flow as above

Norwegian Labels

Element	Norwegian Text
---------	----------------

Page title	Scan QR
Instruction	Pek kameraet mot butikkens QR-kode
Simulate button	Simuler skanning
Confirm prompt	Bekreft betaling
Merchant label	Butikk
Amount label	Belop
Pay button	Betal nå
Cancel button	Avbryt
Success toast	Betaling fullfort!
Error toast	Betaling feilet. Prov igjen.

Design Tokens

Token	Value
Page bg	#1A1A1A (dark/camera)
Viewfinder border	border-dashed border-[#6B7280]
Corner brackets	border-[#0B6E35] (4px)
Scan line	bg-[#0B6E35]
Primary	#0B6E35
Primary hover	#095C2C
Text white	#FFFFFF
Text muted (on dark)	#9CA3AF
Button radius	rounded-xl
Viewfinder radius	rounded-2xl
Button height	h-12

Bottom Navigation

No — this is a full-screen camera overlay, no bottom nav.