



Product Requirements Document

Requirements

Functionele requirements

1. Volledige berekening in de browser

Alle bruto-netto berekeningen gebeuren 100% client-side, zonder dat er data naar een server of API wordt gestuurd. Dit zorgt voor privacy en snelheid.

2. Invoer van bruto salaris en extra parameters

Gebruiker kan invoeren:

- bruto maandsalaris
- leeftijd
- arbeidsduur (uren per week)
- vakantiegeld (checkbox, want is altijd 8%)
- pensioenbijdrage (%)
- 13e maand (checkbox)

3. Directe weergave netto maandbedrag

De netto maandwaarde wordt onmiddellijk getoond zodra de gebruiker gegevens invoert of wijzigt, zonder pagina herladen.

4. Validatie en foutmeldingen

Ongeldige invoer (bijv. letters in salarisveld of negatieve getallen) wordt meteen signaleerd met begrijpelijke foutmeldingen, zodat gebruiker weet wat er fout gaat.

5. Responsief ontwerp

De app werkt vloeiend en ziet er goed uit op mobiel (voornamelijk), tablet én desktop.

6. Toekomstbestendig voor SEO en uitbreiding

De app is zo gebouwd dat later gemakkelijk blogpagina's kunnen worden toegevoegd, en dat SEO-elementen (zoals metadata en structured data) eenvoudig kunnen worden uitgebreid.

Non-functionele requirements

Security

- Omdat alle data alleen in de browser verwerkt wordt en **niet wordt opgeslagen of doorgestuurd**, is privacy heel goed gewaarborgd: gebruikersgegevens verlaten nooit hun apparaat.
- De site wordt alleen via **HTTPS** aangeboden om te voorkomen dat iemand tijdens het laden meekijkt met de data (ook al is die data alleen lokaal).
- Er is geen inlog of registratie; iedereen gebruikt de tool anoniem.

SEO

- De site krijgt een goede **meta-titel** en **meta-beschrijving** gericht op relevante zoekwoorden zoals: "bruto netto calculator 2025", "netto salaris berekenen Nederland", "bruto naar netto salaris tool". Dit helpt Google te begrijpen waar de site over gaat.
- De pagina bevat **semantische HTML** (zoals headers `<h1>`, `<h2>`) zodat zoekmachines de inhoud goed kunnen indexeren.
- Er wordt gebruikgemaakt van **structured data** (JSON-LD) om de calculator als tool te markeren voor Google, wat de kans vergroot op speciale zoekresultaten (rich snippets).
- De website is volledig **mobielvriendelijk** en snel, wat een positieve invloed heeft op de ranking (Google let sterk op mobiel gebruik en laadsnelheid).
- De code en content zijn zo opgebouwd dat het makkelijk wordt om later **blogposts** toe te voegen, zonder dat dat nu de MVP vertraagt.

Performance en usability

- Pagina moet in maximaal **2 seconden laden** op gemiddelde mobiele netwerken (4G).
- Resultaten worden binnen **1 seconde** weergegeven na invoer of wijziging.
- De app voldoet aan basale **toegankelijkheidseisen** (zoals contrast en toetsenbordnavigatie) voor zo breed mogelijk gebruik.

Onderhoudbaarheid en schaalbaarheid

- Code is modulair en gedocumenteerd zodat later makkelijk aanpassingen of uitbreidingen gedaan kunnen worden (bijv. nieuwe parameters of rapportages).
- De belastingtarieven en regels zitten in een aparte configuratie zodat ze eenvoudig up-to-date gehouden kunnen worden voor toekomstige jaren (2026+).

Tech-stack

Categorie	Technologie	Motivatie
Frontend framework	React (via Vite)	Moderne, snelle frontendontwikkeling met componentstructuur en herbruikbaarheid.
Styling	Tailwind CSS	Snel stylen, responsief design, consistente huisstijl.
Berekeningen	JavaScript/TypeScript (client-side logica)	Logica voor bruto → netto berekening draait volledig in de browser, geen backend nodig.
Build & hosting	Vercel	Eenvoudige hosting voor static sites, inclusief automatische CI/CD bij push naar GitHub.
Testing	Vitest	Unit tests voor berekeningslogica.
Versiebeheer	GitHub	Samenwerken, versiebeheer, en automatische deploys.

Architectuur – 100% Client-Side

Beschrijving:

- **Geen backend:** Alle logica draait in de browser.

- **Geen database:** Alle berekeningen gebeuren direct op basis van user input.
- **Static hosting:** HTML, CSS en JS worden als statische bestanden geserveerd.
- **Berekeningsmodule:** Afzonderlijke module in JavaScript/TypeScript met Nederlandse belastingtabellen en logica.
- **UI-componenten:** Herbruikbare React-componenten voor invoervelden, knoppen en resultaatweergave.

Voordelen van deze aanpak

- **Supersnel:** Geen netwerkverkeer voor berekeningen → direct resultaat.
- **Privacy:** Gegevens worden niet naar een server gestuurd.
- **Kostenloos:** Geen server- of databasekosten.
- **Schaalbaar:** Onbeperkt aantal gebruikers zonder extra kosten.
- **Eenvoudig te onderhouden:** Geen backendcode of databasebeheer.

User Interface

Mock-up

Bruto naar Netto Rekenaar

Voer je bruto maandsalaris in en bereken direct je nettoloon voor 2025.

Bruto maandsalaris

Ongeldige invoer

Leeftijd

Arbeidsduur

☒ Vakantiegeld

☐ Pensioenbijdrage (%) uren per week

☒ 13e maand

Netto maandbedrag

€ 0

Logica 2025 brutto → netto

```
ts
Kopiëren
// src/utils/calculator.ts
export type CalcInput = {
  brutoMaandsalaris: number; // >0
  leeftijd: number;          // niet gebruikt in MVP; later voor AOW-variant
  urenPerWeek: number;       // 1..60; fulltime = 40
  vakantiegeldAan: boolean;   // 8% over jaarBasis
  pensioenBijdragePct: number; // 0..100 (werknemer); over jaarBasis + 13e
  dertiendeMaandAan: boolean; // 1x bruto maand (geschaald)
};

export type CalcOutput = {
  nettoPerMaand: number;
  tussenstappen: {
```

```

    schaalFactor: number;
    brutoMaandGeschaald: number;
    jaarBasis: number;
    vakantiegeld: number;
    dertiendeMaand: number;
    jaarTotaalBruto: number;
    pensioenAftrek: number;
    belastbaarJaar: number;
    belastingJaar: number;    // vóór heffingskortingen
    heffingskortingen: number; // algemene + arbeids
    belastingNaKorting: number;
    nettoJaar: number;
  };
};

// ---- 2025 parameters (niet-AOW) ----
// Box-1 tarieven 2025: 35,82% / 37,48% / 49,50% met grenzen 38.441 en
// 76.817. (Rijksoverheid) :contentReference[oaicite:4]{index=4}
const BOX1_BANDS_2025 = [
  { upto: 38441, rate: 0.3582 },
  { upto: 76817, rate: 0.3748 },
  { upto: Infinity, rate: 0.495 },
];

// Algemene heffingskorting 2025 (max 3.068; afbouw vanaf 28.406 met 6,
// 337%; bij 76.817 ≈ 0). :contentReference[oaicite:5]{index=5}
export function algemeneHeffingskorting2025(ink: number): number {
  const max = 3068;
  const drempel = 28406;
  const afbouwPct = 0.06337;
  if (ink <= drempel) return max;
  if (ink >= 76817) return 0;
  const k = max - (ink - drempel) * afbouwPct;
  return Math.max(0, Math.min(max, k));
}

// Arbeidskorting 2025 (niet-AOW) – piecewise officiële tabel. :contentRefer
ence[oaicite:6]{index=6}

```

```

export function arbeidskorting2025(arbInk: number): number {
  const a = 12169;
  const b = 26288;
  const c = 43071;
  const d = 129078;

  if (arbInk <= 0) return 0;

  if (arbInk <= a) return 0.08053 * arbInk;

  if (arbInk <= b) return 980 + 0.3003 * (arbInk - a);

  if (arbInk <= c) return 5220 + 0.02258 * (arbInk - b);

  if (arbInk < d) {
    const k = 5599 - 0.06510 * (arbInk - c);
    return Math.max(0, k);
  }
  return 0;
}

// ----- helpers -----
function berekenBelasting(jaarInk: number): number {
  let resterend = jaarInk;
  let begin = 0;
  let totaal = 0;
  for (const band of BOX1_BANDS_2025) {
    const grens = band.upto;
    const deel = Math.max(0, Math.min(resterend, grens - begin));
    totaal += deel * band.rate;
    resterend -= deel;
    begin = grens;
    if (resterend <= 0) break;
  }
  return totaal;
}

function clamp(n: number, min: number, max: number) {

```

```

return Math.min(max, Math.max(min, n));
}

function assertInputs(i: CalcInput) {
  if (!(i.brutoMaandsalaris > 0)) throw new Error("Voer een geldig bruto maandsalaris (> 0) in.");
  if (!(i.urenPerWeek > 0 && i.urenPerWeek <= 60)) throw new Error("Uren per week moet tussen 1 en 60 liggen.");
  i.pensioenBijdragePct = clamp(i.pensioenBijdragePct, 0, 100);
}

// ----- kern -----
export function calculateNetMonthly(i: CalcInput): CalcOutput {
  assertInputs(i);
  const FULLTIME = 40;
  const VAK_PCT = 0.08; // 8%

  const schaalFactor = i.urenPerWeek / FULLTIME;
  const brutoMaandGeschaald = i.brutoMaandsalaris * schaalFactor;

  const jaarBasis = brutoMaandGeschaald * 12;
  const vakantiegeld = i.vakantiegeldAan ? jaarBasis * VAK_PCT : 0;
  const dertiendeMaand = i.dertiendeMaandAan ? brutoMaandGeschaald : 0;

  const jaarTotaalBruto = jaarBasis + vakantiegeld + dertiendeMaand;

  // MVP-aanname: pensioen over jaarBasis + 13e (niet over vakantiegeld)
  const pensioenAftrek = (jaarBasis + dertiendeMaand) * (i.pensioenBijdragePct / 100);

  const belastbaarJaar = Math.max(0, jaarTotaalBruto - pensioenAftrek);

  // Box-1 belasting vóór kortingen
  const belastingJaar = berekenBelasting(belastbaarJaar);

  // Heffingskortingen — MVP: op basis van belastbaarJaar als benadering van arbeids-/verzamelinkomen

```



```

const alg = algemeneHeffingskorting2025(belastbaarJaar);
const arb = arbeidskorting2025(belastbaarJaar);
const heffingskortingen = Math.max(0, alg + arb);

const belastingNaKorting = Math.max(0, belastingJaar - heffingskortingen);
const nettoJaar = belastbaarJaar - belastingNaKorting;
const nettoPerMaand = nettoJaar / 12;

return {
  nettoPerMaand,
  tussenstappen: {
    schaalFactor,
    brutoMaandGeschaald,
    jaarBasis,
    vakantiegeld,
    dertiendeMaand,
    jaarTotaalBruto,
    pensioenAftrek,
    belastbaarJaar,
    belastingJaar,
    heffingskortingen,
    belastingNaKorting,
    nettoJaar,
  },
};
}

```

Bestandstructuur

bruto-netto/

```

├─ public/
│   └─ favicon.ico          # Favicon en eventuele statische assets
├─ src/
│   ├─ components/         # React-componenten
│   │   └─ SalaryForm.tsx  # Form voor invoer bruto, uren, pensioen, etc.
│   │   └─ NetResult.tsx   # Toont netto bedrag + breakdown

```

- | | └─ Layout.tsx # Header/Footer wrapper
- | └─ utils/ # Logica & helpers
- | | └─ calculator.ts # Berekeningslogica 2025
- | | └─ validators.ts # Inputvalidatie
- | └─ data/
- | | └─ tax-2025.json # Belastingtarieven & kortingen
- | └─ styles/
- | | └─ index.css # Tailwind entry point
- | └─ App.tsx # Hoofdcomponent
- | └─ main.tsx # React entry point
- | └─ vite-env.d.ts
- └─ index.html # HTML-template
- └─ package.json
- └─ tsconfig.json
- └─ tailwind.config.cjs
- └─ postcss.config.cjs