ChatGPT

Uitbreiding Ontwikkelplan: Module Kinderopvangtoeslag

In het oorspronkelijke ontwikkelplan voor de kinderopvang rekentool is de berekening van de **brutokosten** per kind uitgewerkt. Hier voegen we een module toe voor het berekenen van de **kinderopvangtoeslag**, zodat de tool ook de **nettokosten** voor ouders kan tonen. Deze uitbreiding beschrijft de benodigde **data** (toeslagtabel), de **architectuur** voor beheer van toeslagtabel en eventuele gemeentelijke toeslagen, de **berekeningslogica**, de **impact op de frontend**, wijzigingen in **database/API** en een **testplan**.

Benodigde databronnen

De toeslagmodule gebruikt jaarlijks veranderende gegevens van de Belastingdienst. Per **toeslagjaar** publiceren zij een tabel met inkomensklassen en bijbehorende vergoedingpercentages, plus maximum uurtarieven per opvangsoort [1]. Deze data wordt door een superuser in het systeem geladen als een JSON-bestand. Belangrijke onderdelen van dit JSON-formaat zijn:

- **Jaar van toepassing** – Bijv. `\"year\": 2024` om het toeslagjaar aan te duiden.
- **Maximum uurtarieven per opvangsoort** – Object met tarieven, bijv. `\"max_hourly_rates\": { \"dagopvang\": 10.25, \"bso\": 9.12, \"gastouder\": 7.53 }` voor 2024 [2]. Dit zijn de hoogste tarieven per uur die vergoed worden; hoger tarief wordt niet vergoed boven dat maximum [3].
- **Inkomensklassen** – Een array `\"income_brackets\"` met entries die de toeslagpercentages bevatten. Elke entry heeft:
- `min` en `max` gezamenlijk toetsingsinkomen (inkomensgrens van die klasse, in euro's per jaar).
- `perc_first_child` – toeslagpercentage voor het **eerste kind** in opvang binnen die inkomensklasse.
- `perc_other_children` – toeslagpercentage voor het **tweede en volgende kind** binnen die klasse.

Een vereenvoudigd voorbeeld van de JSON-struktuur voor een toeslagtabel is:

```
{
  "year": 2024,
  "max_hourly_rates": {
    "dagopvang": 10.25,
    "bso": 9.12,
    "gastouder": 7.53
  },
  "income_brackets": [
    {
      "min": 0,
      "max": 22346,
      "perc_first_child": 96.0,
      "perc_other_children": 96.0
```

```
    },
    {
      "min": 22347,
      "max": 23834,
      "perc_first_child": 96.0,
      "perc_other_children": 96.0
    },
    {
      "...": "... (verdere klassen) ..."
    },
    {
      "min": 207680,
      "max": null,
      "perc_first_child": 33.3,
      "perc_other_children": 67.1
    }
  ]
}
```

Hierbij duidt `"max": null` aan dat de laatste klasse doorloopt tot oneindig (\"207.680 en hoger\" in de toeslagtabel). De percentages variëren per inkomen: lage inkomens krijgen ~96% vergoeding, terwijl hoge inkomens afbouwen naar ~33% voor het eerste kind en ~67% voor latere kinderen, volgens het model van de Belastingdienst [4] [5]. Deze JSON-tabel dient jaarlijks te worden geüpdatet door de superuser (bij nieuwe overheidsgegevens) en vormt de bron voor alle toeslagberekeningen dat jaar.

Daarnaast kan de tool optioneel rekening houden met **gemeentelijke toeslagen**. Sommige gemeentes bieden een extra tegemoetkoming voor de eigen bijdrage van ouders (bijvoorbeeld Amsterdam en Den Haag vergoeden ~4% extra van de kosten tot aan het maximumtarief) [6] [7]. Deze lokale toeslagpercentages zullen per organisatie configureerbaar zijn (zie verder bij Architectuur). Er is geen aparte externe databron; de organisatie voert zelf de relevante gemeentelijke toeslag in (bijv. 4% extra subsidie).

Architectuur en beheer van toeslagtabel en gemeentelijke toeslagen

Toeslagtabel beheer: We introduceren een nieuw gegevensobject **Toeslagtabel** in de applicatie. Dit kan gerealiseerd worden als een database-entiteit met velden voor het jaar en de JSON-structuur, of door het JSON-bestand op de server op te slaan en in te lezen. De voorkeur gaat uit naar opslaan in de database voor versiebeheer en eenvoudige koppeling. Een superuser (applicatiebeheerder) krijgt een admin-interface om jaarlijks de nieuwe toeslagtabel te uploaden (in het gespecificeerde JSON-formaat). Bij het uploaden wordt de JSON gevalideerd op structuur (controle op aanwezigheid van `year`, `max_hourly_rates` en `income_brackets` velden en correcte datatypes). Na goedkeuring wordt de nieuwe toeslagtabel opgeslagen onder het betreffende jaar.

Keuze actief toeslagjaar per organisatie: Indien meerdere toeslagjaren in het systeem aanwezig zijn, kan elke organisatie kiezen welke tabel (welk jaar) gebruikt moet worden voor de berekeningen. In de organisatiesettings (beheerportal voor een organisatie) komt een instelling **Actief toeslagjaar**, waar de admin van de organisatie een jaar selecteert. Standaard zal dit het meest recente jaar zijn, maar voor scenario's als vooruitberekenen of achterliggende jaarvergelijking kan men een ander jaar kiezen. Intern wordt deze keuze bijvoorbeeld opgeslagen als een relatie (foreign key) van de organisatie naar

de betreffende Toeslagtabel in de database. Alle berekeningen binnen die organisatie raadplegen dan de juiste toeslagpercentages van het gekozen jaar.

Gemeentelijke toeslagen beheer: Per organisatie komt er tevens een optie om een **gemeentelijke toeslag** in te stellen. In de organisatie-instellingen kan men een extra percentage invoeren dat als lokale tegemoetkoming geldt op de resterende kosten. Dit percentage (bijvoorbeeld 4.0% extra vergoeding) wordt opgeslagen in een veld, bijvoorbeeld `org.municipal_allowance_percentage`. Er is ook een toggle/vinkje **Gemeentelijke toeslag toepassen** om de functie aan of uit te zetten. Als de organisatie deze instelling activeert, zal de rekentool bij elke berekening ook deze gemeentelijke subsidie meerekenen. (Indien een organisatie geen gemeentelijke regeling heeft, blijft dit op 0% en wordt het genegeerd.)

De architectuur voorziet in een flexibele omgang met de toeslagtabeldata: - Bij een berekening haalt de backend het juiste **Toeslagtabel JSON** op voor de organisatie (aan de hand van het gekozen toeslagjaar). - De businesslogica interpreteert deze data (lees: bepaalt toeslagpercentages en max. uurprijzen) zonder hardgecodeerde waarden. Zo blijft de tool bruikbaar voor elk nieuw toeslagjaar door enkel de JSON te updaten. - Gemeentelijke toeslagpercentages zijn gescheiden per organisatie, zodat verschillende organisaties hun eigen lokale regeling kunnen voeren.

Beveiliging/rollen: alleen de superuser rol kan nieuwe toeslagtabellen toevoegen of bestaande aanpassen (dit voorkomt dat individuele organisaties de landelijke tabel wijzigen). Organisatiebeheerders kunnen wel het actieve jaar kiezen en hun gemeentelijke percentage instellen, maar niet de inhoud van de rijks-toeslagtabel zelf.

Logica toeslagberekening per kind

De rekentool berekent op basis van de ingevoerde gegevens van ouders de geschatte kinderopvangtoeslag per kind en de totale netto kosten. Hierbij wordt het **standaardmodel van de Belastingdienst** gevolgd [1]. De logica per calculatie is als volgt:

1. **Inkomen en toeslagpercentage bepalen:** Op basis van het door de ouder geselecteerde gezamenlijke **toetsingsinkomen** wordt de bijbehorende inkomensklasse opgezocht in de geladen toeslagtabel. Hieruit lezen we het toeslagpercentage voor het 1e kind en voor de 2e en volgende kinderen af. (NB: alle kinderen van dezelfde ouders vallen onder hetzelfde gezinsinkomen, dus één inkomensklasse is van toepassing op het hele gezin.)
2. **Identificatie eerste kind vs. volgende kinderen:** Als er meerdere kinderen in de berekening zijn meegenomen, moet één kind als “eerste kind” worden aangemerkt voor toeslagdoeleinden. Volgens de regelgeving geldt het kind met de **meeste opvanguren** als eerste kind. In de implementatie sorteren we de kinderen op afnemend aantal opvanguren; de eerste in die lijst krijgt het percentage voor 1e kind, alle andere kinderen krijgen het (meestal hogere) percentage voor 2e/v volgende kinderen. Bij gelijke uren kan arbitrair één kind als eerste worden genomen.
3. **Brutokosten per kind berekenen:** De brutokosten (zonder toeslag) voor ieder kind zijn normaliter al bepaald in de bestaande rekentool (op basis van uurtarief van de opvang en afgenomen uren). Voor de toeslagberekening is echter van belang: **het uurtarief tot het maximum**. Als de opvang een hoger tarief hanteert dan het door de overheid vergoede maximum, wordt voor de toeslagberekening het maximumtarief gebruikt [3]. Bijvoorbeeld, bij dagopvang met €11/u terwijl het maximale vergoede tarief €10,25/u is, rekenen we met €10,25 als basis voor toeslag. Is het opvangtarief lager dan het maximum, dan wordt uiteraard dat lagere tarief genomen (men krijgt toeslag over het daadwerkelijk betaalde tarief tot max.). Daarnaast geldt een maximum van **230 uur per maand per kind** dat in aanmerking komt voor

toeslag [1] . De tool kan het opgegeven aantal uren per maand per kind limiteren tot 230 indien nodig (als een ouder meer heeft ingevuld, worden voor de toeslagberekening toch maximaal 230 uur gerekend).

4. **Toeslag per kind berekenen:** Voor ieder kind passen we het toeslagpercentage toe op de berekeningsbasis (uren * vergoed tarief). Formule per kind: `toeslag_bedrag = min(uurtarief, maxUurtarief) * min(uren, 230) * (toeslagpercentage / 100)`. Dit resulteert in het **geschatte maandelijkse toeslagbedrag** voor dat kind. We doen deze berekening twee keer met het juiste percentage: eenmaal voor het eerste-kind (voor de toegewezen eerste kind) en voor elke volgende kind met het andere percentage.
5. **Netto kosten per kind en totaal:** De **nettokosten** voor elk kind zijn de brutokosten minus de toeslag voor dat kind. Brutokosten waren bijvoorbeeld `uren * uurtarief` (onafhankelijk van maximum), toeslag is zoals hierboven. Het systeem toont per kind desgewenst: brutokosten, toeslag, en nettobedrag. Ook wordt een totaal netto maandbedrag voor het hele gezin (alle kinderen samen) berekend door alle netto kosten op te tellen.
6. **Gemeentelijke toeslag (optioneel):** Indien de organisatie een gemeentelijke toeslagpercentage heeft geconfigureerd, wordt aanvullend **extra toeslag** berekend over de eigen bijdrage. In de praktijk betekent dit dat over dezelfde basis (uren * min(uurtarief, maxUurtarief)) nog eens X% wordt vergoed door de gemeente. Bijvoorbeeld bij een gemeentelijke toeslag van 4%: `gemeente_toeslag = min(uurtarief, maxUurtarief) * min(uren, 230) * (4%/100)`. Dit bedrag wordt opgeteld bij de landelijke toeslag. In de netto berekening wordt het dus afgetrokken van de brutokosten samen met de reguliere toeslag. **Let op:** nationale toeslag + gemeentelijke toeslag zal doorgaans niet boven 100% van de kosten uitkomen; in lage inkomensgevallen kan 96% landelijk + 4% gemeentelijk neerkomen op vrijwel volledige vergoeding [6] . De tool zou kunnen valideren dat de som van vergoedingen niet boven de (aangepaste) kosten uitkomt (eventuele afrondingsverschillen daargelaten).
7. **Ronding en validatie:** Alle berekende geldbedragen worden afgerond op centen (euro's). Bij het tonen van resultaten zorgen we dat de toeslagbedragen **indicatief** zijn – ouders krijgen een schatting. We nemen in de berekening geen harde beslissingen op basis van werkuren van ouders, aangezien sinds 2023 de koppeling met gewerkte uren is komen te vervallen [8] . Ouders hoeven dus geen werkuren meer in te voeren en de module gaat uit van volledige toeslagrechten tot 230 uur per maand.

Impact op de frontend

Voor de gebruikerskant (ouders die de rekentool invullen) zijn enkele uitbreidingen nodig om de toeslagmodule te ondersteunen:

- **Extra invoerveld: Inkomen selectie.** In de rekenmodule wordt een nieuwe vraag toegevoegd waarin de ouder de **gezamenlijke inkomensrange** per jaar kiest. Dit realiseren we met een dropdown/select-veld dat de inkomensklassen toont. De opties in deze dropdown worden dynamisch gevuld op basis van de actieve toeslagtabel van de organisatie. Elke optie geeft een bereik weer, bijv. *"€ 0 – € 22.346"* of *"€ 22.347 – € 23.834"* etc., zodat ouders hun bruto jaarinkomen in de juiste categorie kunnen plaatsen. (We kunnen eventueel ter info ook het bijbehorende toeslagpercentage in de label tonen, bijvoorbeeld "*€ 0 – €22.346 (96% toeslag)*", om inzicht te geven.) Door gebruik van een selectievak voorkomen we invoerfouten en sluit het aan op de discrete tabel van de Belastingdienst.
- **Eventuele overige invoervelden:** Aangezien sinds 2023 werkuren niet meer nodig zijn voor de berekening, kan een eerder veld "aantal gewerkte uren" vervallen of verborgen worden [8] . De ouder hoeft alleen nog per kind de opvangvorm en aantal uren (per week/maand) op te geven, en dus éénmaal het inkomensveld voor het hele gezin.

- **Resultatenweergave: Bruto, toeslag, netto.** Na het invullen van alle gegevens en het activeren van de berekening ziet de gebruiker een uitgebreide resultatensectie. Hierbij wordt per kind en in totaal weergegeven:
- **Brutokosten:** het volledige bedrag aan opvangkosten zonder toeslag (zoals reeds gebeurde in het oorspronkelijke rekentoolresultaat).
- **Kinderopvangtoeslag:** het berekende toeslagbedrag van de Belastingdienst voor dat scenario. Dit kan per kind worden getoond (bij meerdere kinderen) en/of als totaal. Bijvoorbeeld: "Verwachte kinderopvangtoeslag: € X per maand".
- **Gemeentelijke toeslag:** indien van toepassing (als de organisatie deze functie aan heeft staan), tonen we óók de extra gemeentelijke bijdrage. Dit kan onder de landelijke toeslag vermeld worden, bijvoorbeeld als "Gemeentelijke toeslag: € Y extra" of geïntegreerd: "Inclusief gemeentelijke toeslag van 4%". Als de organisatie geen gemeentelijke regeling gebruikt, wordt dit niet getoond aan de gebruiker.
- **Nettokosten:** het bedrag dat de ouder uiteindelijk zelf betaalt (= bruto – toeslagen). Dit is het meest prominente resultaat. De nettokosten kunnen vetgedrukt of visueel benadrukt worden, zodat de gebruiker direct ziet welk bedrag hij/zij **eigenlijk betaalt** voor de opvang.
- **Visualisatie:** Ter verduidelijking zou de frontend een eenvoudige grafiek of staafdiagram kunnen tonen waarin het aandeel overheidstoeslag vs. eigen bijdrage is weergegeven. Bijvoorbeeld een stapelbalk die €100% kosten opdeelt in X% toeslag (rijk), Y% toeslag (gemeente) en restpercentage eigen bijdrage. Dit helpt ouders te begrijpen hoeveel procent ze zelf betalen. Deze visualisatie is optioneel, maar tekstueel is de opsplitsing in bruto/toeslag/netto al voorzien.
- **UX en validatie:** Het selecteren van inkomensrange via dropdown is eenvoudig; we kunnen een default ("Selecteer uw inkomenscategorie") tonen om de gebruiker te dwingen een keuze te maken. Eventueel kunnen we tooltips of info-iconen toevoegen bij het inkomensveld en toeslagresultaat, om uit te leggen dat het een schatting betreft op basis van huidig toeslagjaar en dat de definitieve toeslagaanvraag via de Belastingdienst loopt. Als een gebruiker geen inkomen selecteert, zal de module geen toeslag berekenen en kan een foutmelding of reminder verschijnen ("Selecteer uw gezinsinkomen om toeslag te berekenen").

Samengevat, de frontend krijgt één extra invoerveld en de resultaatsweergave wordt uitgebreid met de berekende toeslag en nettokosten. Deze wijzigingen moeten aansluiten bij de bestaande stijl van de rekentool, bijvoorbeeld in dezelfde resultatenkaart of tabelvorm als brutokosten nu al worden getoond.

Database- en API-structuur aanpassingen

Om de bovenstaande functionaliteit te ondersteunen, zijn de volgende aanpassingen in de database en API nodig:

- **Database uitbreidingen:**
- Een nieuwe tabel/model **ToeslagTabel** met velden: `jaar` (integer, primary key of uniek) en `data` (JSON type) waarin de toeslagtabel opgeslagen is. Alternatief kan gekozen worden voor een gerelateerd model **ToeslagPercentage** met velden (jaar, inkomens_min, inkomens_max, pct_eerste_kind, pct_volgende_kind) voor elke tabelrij, en een apart model **MaxUurprijs** (jaar, opvangtype, bedrag). Echter, het opslaan van de ruwe JSON per jaar is voor dit doel efficiënt en sluit aan bij de input (minder relationele complexiteit).
- Uitbreiding van het **Organisatie** model: voeg een veld toe `actief_toeslagjaar` (ForeignKey naar ToeslagTabel, of integer jaar en dan impliciet koppelen) om de keuze van toeslagjaar per organisatie te registreren. Daarnaast een veld `gemeente_toeslag_percentage` (Decimal/numeric) om een eventueel gemeentelijk toeslagpercentage op te slaan. Dit veld kan null of 0.0 zijn als er geen lokale toeslag van toepassing is.

- Eventueel een boolean `gemeente_toeslag_actief` kan toegevoegd worden om het expliciet aan/uit te zetten, maar dit is redundant als we ervan uitgaan dat `gemeente_toeslag_percentage` van 0 betekent "uit". Desgewenst kan voor duidelijkheid toch een aparte kolom gebruikt worden.
- Migraties: Bestaande data migreren met default values (bijv. actief_toeslagjaar default op huidig jaar bij alle organisaties, zodat het gedrag niet verandert voor zij die de tool nu gebruiken; gemeentelijke toeslag default op 0).
- **API aanpassingen:**
- Als de rekentool een API endpoint heeft voor het berekenen van kosten (bijvoorbeeld een POST endpoint `/api/calculate-costs/` waar de frontend de ingevulde data naartoe stuurt), dan moet het **request payload** uitgebreid worden met het inkomensgegeven. Dit kan ofwel het geselecteerde **inkomensbereik ID**/waarde zijn, of direct het **gezamenlijk inkomen** als getal. Gezien we in de frontend een dropdown hebben, waarschijnlijk sturen we een identificerende waarde voor de gekozen range. Simpel is: stuur de ondergrens of een code van de range, of het midden van de range. De backend kan ook zelf de juiste range afleiden als het bruto inkomen als getal wordt gestuurd. We kiezen één aanpak en documenteren deze in de API docs.
- De **response payload** van de berekening wordt uitgebreid met de nieuwe outputs: toeslag per kind, toeslag totaal, nettokosten per kind en netto totaal. Als er al een gestructureerde response bestond (bijv. per kind een object met kosten), voegen we daar velden `estimated_allowance` en `net_cost` toe. En op topniveau eventueel `total_allowance` en `total_net_cost`.
- Nieuwe API endpoint(s) voor de toeslagtabel: we zullen mogelijk een endpoint aanbieden zodat de frontend de beschikbare inkomensranges kan ophalen voor de actieve toeslagjaar. Bijvoorbeeld een GET endpoint `/api/allowance-table/{year}/` dat de lijst van inkomensklassen retourneert. Dit kan de frontend gebruiken om de dropdown te vullen en labels (in plaats van de frontend hardcoded waarden te laten bevatten). Alleen geautoriseerde users (zoals organisatiebeheerders) of misschien publiek als de rekenpagina publiek is, kunnen deze data ophalen. Aangezien de toeslagtabel geen privacy-gevoelige info bevat, zou dit eventueel openbaar mogen zijn.
- Beheer endpoints: voor superusers een endpoint of admin GUI om de JSON upload uit te voeren. In een Django-achtig framework zou dit via admin interface kunnen; anders een protected POST endpoint `/api/allowance-table/upload/` om een JSON te uploaden. Deze details hangen af van implementatiekeuzes (GUI vs API upload).
- **Interne logica** (service layer): De berekeningslogica voor toeslag komt in een aparte module of service (`allowance_service.calculate_allowance(...)`) die door de bestaande kostencalculatie wordt aangeroepen. Dit houdt de code georganiseerd. Deze service zal:
- Op basis van organisatie-ID de juiste toeslagtabel (jaar) ophalen uit de database.
- Het inkomensniveau van de request matchen naar een percentage (door de JSON te parsen of query in de percentages tabel).
- De toeslag bedragen berekenen per kind en totaal (zoals in de logica sectie beschreven).
- Resultaat teruggeven aan de controller/endpoint die het verwerkt in de response.
- **Backward compatibility:** Als sommige organisaties (of de tool in algemeen) de toeslagmodule (nog) niet gebruiken, kan het inkomensveld optioneel zijn. De API kan zodanig werken dat als geen inkomen meegegeven wordt, er van 0% toeslag (of een bepaalde safe default) wordt uitgegaan. Echter, gezien de uitbreiding waarschijnlijk door iedereen gebruikt gaat worden, is het logisch om het inkomensveld verplicht te maken zodra de module live gaat.

Testen van de toeslagmodule

Om te zorgen dat de nieuwe toeslagmodule correct en robuust functioneert, wordt een uitgebreide testaanpak gevolgd:

- **Unit tests (logica):** Schrijf unit tests voor de kernberekeningen. Bijvoorbeeld:
- Test de functie die op basis van inkomen het juiste toeslagpercentage ophaalt: voer een inkomen in net onder, op, en boven de grens van een inkomensklasse en controleer dat het percentage klopt volgens de toeslagtabel.
- Test de logica die bepaalt welk kind als eerste kind wordt aangemerkt. Bijv: twee kinderen, één met 40 uur en één met 20 uur opvang per maand – de eerste moet het eerste-kind-percentage krijgen. Ook een test waar beide kinderen gelijke uren hebben om te zien dat er toch precies één als eerste wordt gemarkeerd.
- Test de berekening van toeslagbedrag: scenario's waarbij uurtarief onder, gelijk aan en boven het maximum liggen. Controleer dat bij hoog tarief de toeslag berekend is met het max.uurtarief en niet het echte tarief boven de cap.
- Test het toepassen van de 230-uur grens: voer bijv. 250 uur in, controleer dat toeslagberekening maar over 230 uur gaat.
- Test de gemeentelijke toeslagberekening: bijvoorbeeld met toeslag 96% landelijk + 4% gemeente, controleer dat netto eigen bijdrage ~0% wordt bij een laag inkomen scenario [6]. Of test een middeninkomen waar landelijk bv. 76%, gemeentelijk 4% = 80% totaal vergoeding, zodat netto 20% van kosten overblijft.
- Rekenvoorbeeld vergelijking: Gebruik een bekend rekenvoorbeeld (bijv. een testcasus gepubliceerd door de Belastingdienst) en voer die data in onze module; verifieer dat de output overeenkomt met de verwachte toeslag volgens dat voorbeeld [9] [10].
- **Integratie/API tests:** Deze tests draaien de volledige keten (mogelijk met dummy frontend calls of via API endpoints):
- Test het **uploaden** van een toeslagtabel JSON via de admin/endpoint: stuur een geldig JSON en controleer dat het in de database komt en dat een organisatie hem kan selecteren. Test ook foutgevallen: ongeldig formaat JSON moet een nette foutmelding geven.
- Test de **berekening endpoint** end-to-end: simuleer een API-call met bijvoorbeeld 2 kinderen, hun uren en opvangsoorten, en een gekozen inkomensrange. Controleer of de response de juiste velden bevat (bruto, toeslag, netto) en of de bedragen correct berekend zijn. Doe dit voor een paar verschillende inkomensranges (laag, midden, hoog) om te zien of percentages variëren zoals bedoeld.
- Test de werking van **actief toeslagjaar**: zet voor een organisatie het toeslagjaar op een ander jaar (bijv. 2023 vs 2024 data) en voer met dezelfde input twee berekeningen uit. De verwachting is dat de toeslagresultaten verschillen overeenkomstig de wijzigingen in percentages tussen die jaren. Verifieer dit verschil handmatig aan de hand van bekende tabelwijzigingen tussen jaren (bijv. iets strenger in 2024 voor bepaalde inkomens).
- Test de **gemeentelijke toeslag opt-in**: voor een organisatie met 0% gemeentelijke toeslag vs een met 4%. Voer identieke berekeningen uit en check dat in de tweede case de netto kosten lager zijn precies met dat gemeentelijke deel. Als er een toggle is, test het aan- en uitzetten.
- **UI/gebruikerstesten:** Voer handmatige of geautomatiseerde frontend tests uit:
- Controleer dat de dropdown met inkomensranges correct gevuld wordt (komt overeen met de JSON data van dat jaar). Bijvoorbeeld, na selectie van jaar 2024 in admin, verwacht je in de UI opties startend met "€ 0 – € 22.346 …" etc., conform de toeslagtabel 2024.
- Vul het formulier op de frontend met verschillende gegevens en check visueel of de resultaten kloppen. Dit omvat ook edge cases zoals: geen inkomen geselecteerd (moet nette foutmelding geven), 0 uur opvang (toeslag zou dan ook €0 moeten zijn), extreem hoog inkomen (valt in laatste categorie "en hoger", toeslagpercentages minimaal).

- Usability test: laat bij voorkeur een paar eindgebruikers (ouders) de nieuwe velden en output bekijken om te zien of het begrijpelijk is. Krijgen ze correct inzicht in wat hun netto kosten worden en hoe de toeslag is opgebouwd? Hun feedback kan nog leiden tot aanpassingen in labeling of uitleg in de interface.
- **Performance test:** Hoewel de berekeningen niet zwaar zijn, kan een performance test bevestigen dat het opzoeken in de JSON tabel en berekenen voor meerdere kinderen vlot verloopt, ook bij gelijktijdig gebruik door veel users. Dit is met name relevant als de toeslagtabel als JSON bij elke request opnieuw geparsed wordt. Indien nodig kan caching toegepast worden (bijv. de toeslagtabel éénmaal in memory houden per proces).

Door deze mix van testen verzekeren we dat de toeslagmodule betrouwbaar werkt. Daarnaast is het aan te raden de resultaten te vergelijken met de officiële proefberekening van de Belastingdienst voor een paar gevallen, als extra validatie [11]. Bij afwijkingen moet de berekeningslogica of invoerinterpretatie herzien worden totdat we consistente uitkomsten hebben. Zodra alle tests slagen, kan de module met vertrouwen live gezet worden als onderdeel van de kinderopvang rekentool.

---

[1] [3] [5] [9] [10] [11] Bedragen kinderopvangtoeslag 2023 | Kinderopvangtoeslag | Rijksoverheid.nl
https://www.rijksoverheid.nl/onderwerpen/kinderopvangtoeslag/bedragen-kinderopvangtoeslag-2023

[2] [4] Bedragen kinderopvangtoeslag 2024 | Kinderopvangtoeslag | Rijksoverheid.nl
https://www.rijksoverheid.nl/onderwerpen/kinderopvangtoeslag/bedragen-kinderopvangtoeslag-2024

[6] Vergoeding eigen bijdrage kinderopvang - Gemeente Amsterdam
https://www.amsterdam.nl/werk-en-inkomen/regelingen-bij-laag-inkomen-pak-je-kans/regelingen-alfabet/vergoeding-eigen-bijdrage-kinderopvang/

[7] Vergoeding kinderopvang - Den Haag
https://www.denhaag.nl/nl/geld-en-schulden/vergoeding-kinderopvang/

[8] Kinderopvangtoeslag vanaf 2023 niet meer afhankelijk van gewerkte uren - Kinderopvangtotaal
https://www.kinderopvangtotaal.nl/kinderopvangtoeslag-vanaf-2023-niet-meer-afhankelijk-van-gewerkte-uren/