

# Project: Groen & Gewoon Doen

Website & Beheersysteem voor een Hovenier



<b>Weekplanning</b>	<b>3</b>
Introductie	4
Week 1 – Plan, ontwerp & structuur	8
Week 2 – Node.js setup & klant interview	9
Week 3 – Klantzijde: standaard pakketten & basisformulier	10
Week 4 – Klantzijde: custom offerte-tool	11
Week 5 – Klantzijde: orderoverzicht & akkoord/afkeuren	12
Week 6 – Admin page: orders beheren & inplannen	13
Week 7 – Afronden & presenteren	14
Beoordeling	15

# Weekplanning

Week	Fase / Thema	Focus van de week	Belangrijkste resultaten
1	Plan & Structuur	Projectstructuur en basisfront-end opzetten	Basis front-end, map- en bestandstructuur, Git commit
2	Node.js setup & Klantinterview	Node.js server opzetten en API routes maken	Werkende Node.js server, Git commit
3	Klantzijde: standaard pakketten	Formulier en pakketkeuze via API	Werkend formulier + pakketkeuze, Git commit
4	Klantzijde: custom offerte-tool	Dynamische offerte berekenen en opslaan	Werkende offertetool + API, Git commit
5	Klantzijde: orderoverzicht & akkoord	Orders bekijken en status aanpassen	Werkend klantoverzicht + akkoord/afkeur, Git commit
6	Beheerszijde: orders beheren & inplannen	Admin orders beheren en planning aanpassen	Werkende admin-orderspagina, Git commit
7	Pakketten & tarieven beheren + afronden	CRUD pakketten/tarieven, testen en presentatie	Werkende website met CRUD, testoverzicht, GitHub-repo, presentatiebestand

# Introductie

Dit project wordt uitgevoerd in teams van 2–3 studenten.

Het doel is een volledig werkende website te bouwen waarmee klanten tuin services kunnen bestellen en Henk zijn beheeromgeving kan beheren.

Tijdens het project werken jullie met:

- **Git** (versiebeheer)
- **Trello** (planning en taakverdeling)

Iedere student is verantwoordelijk voor het begrijpen en kunnen uitleggen van:

- JSON-data opslag en manipulatie
- JavaScript-functies en CRUD-logica
- Front-end en beheers zijde functionaliteit

## Leerdoelen

Na afronding van dit project kan de student:

- Een website ontwerpen en structureren voor klant- en beheers zijde
- Een JSON-configuratiebestand gebruiken om data op te slaan en te bewerken
- CRUD-functionaliteit implementeren in JavaScript
- Dynamische offertes berekenen en tonen
- Tussentijds testen en feedback verwerken
- Een project presenteren

## Technieken & Randvoorwaarden

- HTML / CSS / JavaScript
- Data-opslag in JSON-bestanden (`orders.json`, `packages.json`)
- Modulair werken via `functions.js`
- Responsive design (desktop + mobiel)

## Product: Wat ga je maken?

Een website voor Groen & Gewoon Doen waarin klanten:

- Standaard pakketten kunnen bestellen
- Custom opdrachten kunnen invoeren en een dynamische offerte krijgen
- Orders kunnen bekijken en akkoord/afkeuren

Henk kan via een beheersomgeving:

- Orders beheren en inplannen
- Pakketten en tarieven beheren

# Klantzijde (Gebruikers)



Groen & Gewoon Doen

Menu ▾



## Welkom!

Bestel uw tuinservice eenvoudig online!

### Standaard Pakketten

Kies een pakket

Bestel

### Custom Offerte

m<sup>2</sup> Gras

m<sup>2</sup> Tegels

Meters Heg

Opties

Opties

Bereken Offerte

### Mijn Orders

Bestel

Order ID	Pakket / Info	Offerte	Status	Datum
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]
[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]

Contact | Social Media | Disclaimer

# Beheerszijde (Admin)

☰ Admin - Groen & Gewoon Doen Logout

Orders Pakketten Tarieven

### Orders Overzicht

Order ID	Klant	Details	Offerte	Status	Actos
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	Inplannen
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	Aanvragen
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	Afgroond

### Pakketten Beheren

Naam	Prijs	Bewerk	Verwijder	Vervroegen
Nieuw Pakket				
Naam	<input type="text"/>			
Beschrijving	<input type="text"/>			
Prijs	<input type="text"/>			
<button>Toevoegen</button>				

### Tarieven Beheren (Optional)

Prijs per m <sup>2</sup> Gras	Prijs per n <sup>2</sup> Tegels	<input type="text"/>	<button>Opslaan</button>
-------------------------------	---------------------------------	----------------------	--------------------------

© Groen & Gewoon Doen – Admin Paneel

# Mobiele Weergave



# Week 1 – Plan, ontwerp & structuur

Uitleg: Je leert het project plannen en de basisfront-end opzetten.

Stappenplan:

- Map- en bestandstructuur maken:

/projectmap

  /data

    orders.json

    packages.json

  /js

    functions.js

    main.js

    admin.js

  index.html

  admin.html

  style.css

  server.js        <- Node.js server

  package.json     <- Node project

  README.md

- HTML-skeletten maken voor index.html en admin.html
- GitHub-repo aanmaken en eerste commit

Hulp/Links:

- HTML: <https://www.w3schools.com/html/>
- CSS: <https://www.w3schools.com/css/>
- Git: <https://www.w3schools.com/git/>
- Node.js basis: <https://nodejs.org/en/docs/guides/getting-started-guide/>

Op te leveren:

- Basis front-end in Git
- Map- en bestandstructuur

## Week 2 – Node.js setup & klant interview

Uitleg: Node.js server opzetten en API-structuur maken voor orders en pakketten.

Stappenplan:

- Node.js project initialiseren: `npm init -y`
- Express installeren: `npm install express body-parser cors`
- Basis server (`server.js`) maken: routes voor `/orders` en `/packages` (GET, POST, PUT, DELETE)
- Docent-interview: ontwerp en functionaliteiten checken
- Feedback verwerken in ontwerp

Hulp/Links:

- Node.js Express: <https://expressjs.com/en/starter/hello-world.html>
- JSON-bestanden lezen/schrijven in Node.js: <https://nodejs.dev/en/learn/reading-files-with-nodejs/>
- JavaScript fetch API: [https://www.w3schools.com/js/js\\_api\\_fetch.asp](https://www.w3schools.com/js/js_api_fetch.asp)

Op te leveren:

- Werkende Node.js server met basis API
- Git commit: "Node.js server setup + feedback verwerkt"

## Week 3 – Klantzijde: standaard pakketten & basisformulier

Uitleg: Klant kan een standaard pakket bestellen; data wordt via API naar Node.js gestuurd.  
Stappenplan:

- `packages.json` vullen met testpakketten
- `loadPackages()` functie maakt GET-request naar `/packages`
- Formulier op index.html: naam, adres, datum/tijd, pakketkeuze
- `createOrder(orderData)` maakt POST-request naar `/orders`

Hulp/Links:

- HTML forms: [https://www.w3schools.com/html/html\\_forms.asp](https://www.w3schools.com/html/html_forms.asp)
- JavaScript DOM manipulatie: [https://www.w3schools.com/js/js\\_htmlDom.asp](https://www.w3schools.com/js/js_htmlDom.asp)
- Fetch POST request: [https://www.w3schools.com/js/js\\_api\\_fetch.asp](https://www.w3schools.com/js/js_api_fetch.asp)

Op te leveren:

- Werkend formulier + pakketkeuze via Node.js API
- Git commit: "add order form & loadPackages API"

## Week 4 – Klantzijde: custom offerte-tool

Uitleg: Dynamische offerte berekenen en custom order opslaan via Node.js.

Stappenplan:

- `rates.json` aanmaken
- `loadRates()` en `calculateQuote()` functies
- Inputvelden: m<sup>2</sup> gras/tegels, meters heg, extra opties
- Offerteprijs tonen op de pagina
- POST naar `/orders` via Node.js

Hulp/Links:

- JSON manipulatie in JavaScript: [https://www.w3schools.com/js/js\\_json\\_intro.asp](https://www.w3schools.com/js/js_json_intro.asp)
- JavaScript math: [https://www.w3schools.com/js/js\\_math.asp](https://www.w3schools.com/js/js_math.asp)
- Event listeners: [https://www.w3schools.com/js/js\\_htmldom\\_eventlistener.asp](https://www.w3schools.com/js/js_htmldom_eventlistener.asp)

Op te leveren:

- Werkende custom offertetool + API-integratie
- Git commit: "custom order + calculateQuote API"

## Week 5 – Klantzijde: orderoverzicht & akkoord/afkeuren

Uitleg: Klant kan orders bekijken, status aanpassen (akkoord/niet akkoord).

Stappenplan:

- "Mijn orders"-pagina: GET `/orders?customerId=xxx`
- `renderCustomerOrders(orderList)` functie
- Knoppen PUT `/orders/:id/status`

Hulp/Links:

- JavaScript foreach: [https://www.w3schools.com/js/js\\_loop\\_for.asp](https://www.w3schools.com/js/js_loop_for.asp)
- DOM update: [https://www.w3schools.com/js/js\\_htmldom\\_nodes.asp](https://www.w3schools.com/js/js_htmldom_nodes.asp)
- Fetch PUT request: [https://www.w3schools.com/js/js\\_api\\_fetch.asp](https://www.w3schools.com/js/js_api_fetch.asp)

Op te leveren:

- Werkend klantoverzicht + akkoord/afkeur via Node.js
- Git commit: "customer orders overview + approval API"

## Week 6 – Admin page: orders beheren & inplannen

Uitleg: Henk kan pakketten en tarieven beheren; project afronden en presenteren.

Stappenplan:

- CRUD pakketten: GET/POST/PUT/DELETE [/packages](#)
- Tarievenbeheer (optioneel): GET/PUT [/rates](#)
- Testplan uitvoeren: formulier, custom offerte, admin-acties, status updates, akkoord/afkeuren
- Bugs oplossen, code opschonen, design afronden
- Presentatie voorbereiden (3–5 min demo)

Hulp/Links:

- Node.js CRUD: [https://www.w3schools.com/nodejs/nodejs\\_routing.asp](https://www.w3schools.com/nodejs/nodejs_routing.asp)
- JSON file opslaan/aanpassen: <https://nodejs.dev/en/learn/reading-files-with-nodejs/>
- Git workflow: [https://www.w3schools.com/git/git\\_workflow.asp](https://www.w3schools.com/git/git_workflow.asp)

Op te leveren:

- Werkende website met CRUD via Node.js
- Testoverzicht
- Complete GitHub-repo
- Presentatiebestand

## Week 7 – Afronden & presenteren

**Uitleg:** Henk kan pakketten/tarieven beheren; systeem testen en afronden.

**Stappenplan:**

- Admin: lijst pakketten (`packages.json`) met toevoegen, bewerken, verwijderen (`addPackage()`, `editPackage()`, `deletePackage()`)
- Optioneel: tarievenbeheer (`rates.json`)
- Testplan uitvoeren: formulier, custom offerte, admin-acties, status updates, akkoord/afkeuren
- Bugs oplossen, code opruimen, design afronden
- Presentatie voorbereiden: klant- en beheerszijde, demo 3–5 min

**Hulp/Links:**

- JS modules & functions: [https://www.w3schools.com/js/js\\_modules.asp](https://www.w3schools.com/js/js_modules.asp)

**Op te leveren:**

- Werkende website + complete GitHub-repo
- Testoverzicht
- Presentatiebestand (PowerPoint / PDF / video)

## Beoordeling

Onderdeel	V (Voldoende)	RV (Ruim Voldoende)	G (Goed)	Gewicht
Structuur & Front-end	Basis mappenstructuur aanwezig, homepage functioneel	Extra pagina's aanwezig, bestandsindeling overzichtelijk	Responsief design, interactieve elementen, nette layout	15%
JSON & JavaScript kennis	Packages en orders correct in JSON, basisfuncties werken	Modulair gebruik van functions.js, foutafhandeling	Geavanceerde functies, dynamische offerte volledig geïntegreerd	20%
CRUD-functionaliteit	Orders toevoegen, bekijken, bewerken, verwijderen werkt	Alle acties correct geïntegreerd, visuele feedback	CRUD volledig geïntegreerd met klant- en adminzijde, logs/historie	25%
Tussentijdse controle	Project loopt volgens planning	Regelmatige updates, kleine verbeteringen doorgevoerd	Volledige teamreflectie zichtbaar, project klaar voor presentatie	20%
Presentatie	Demo werkende applicatie	Demo duidelijk, uitleg werking	Professionele presentatie, technisch en functioneel, vragen beantwoorden	20%

## Bronnen & Hulplinks

- JSON in JS: [https://www.w3schools.com/js/js\\_json\\_intro.asp](https://www.w3schools.com/js/js_json_intro.asp)
- DOM manipulatie: [https://www.w3schools.com/js/js\\_htmldom.asp](https://www.w3schools.com/js/js_htmldom.asp)
- JS Events: [https://www.w3schools.com/js/js\\_events.asp](https://www.w3schools.com/js/js_events.asp)
- Forms: [https://www.w3schools.com/html/html\\_forms.asp](https://www.w3schools.com/html/html_forms.asp)
- CSS Layouts: [https://www.w3schools.com/css/css\\_layout.asp](https://www.w3schools.com/css/css_layout.asp)