**1  Create the React + TypeScript project (Vite)**

bash

CopyEdit

# 1‑A  Generate the project

npm create vite@latest teacher-maternity-calc -- --template react-ts

cd teacher-maternity-calc

# 1‑B  Install dependencies

npm install

You now have a skeleton:

pgsql

CopyEdit

teacher-maternity-calc/

├─ package.json

├─ vite.config.ts

├─ tsconfig.json

├─ index.html

└─ src/

├─ main.tsx

└─ App.tsx

**2  Add Tailwind + shadcn/ui (matches our component styling)**

bash

CopyEdit

# 2‑A  Tailwind

npm install -D tailwindcss postcss autoprefixer

npx tailwindcss init -p

# 2‑B  Tell Tailwind which files to scan

# tailwind.config.cjs

module.exports = {

content: ['./index.html', './src/\*\*/\*.{js,ts,jsx,tsx}'],

theme: { extend: {} },

plugins: [],

};

# 2‑C  Create src/index.css

echo -e "@tailwind base;\n@tailwind components;\n@tailwind utilities;" > src/index.css

Add at the **top** of src/main.tsx:

tsx

CopyEdit

import './index.css';

**shadcn/ui primitives (card, button, input, table)**

bash

CopyEdit

npx shadcn-ui@latest init # accept defaults

npx shadcn-ui@latest add card input button table

**3  Paste the calculator component**

1. Create src/components/MaternityCalculator.tsx.
2. Paste the final *phase‑level* code (the version that no longer errors).
3. Fix imports (use "~/components/ui/..." or relative ../ui/...).

Then replace src/App.tsx:

tsx

CopyEdit

import MaternityCalculator from './components/MaternityCalculator';

export default function App() {

return (

<div className="min-h-screen flex items-center justify-center bg-gray-100">

<MaternityCalculator />

</div>

);

}

Run locally:

bash

CopyEdit

npm run dev # http://localhost:5173

You should see the full calculator UI.

**4  Add a “Buy Plan” trigger that talks to the parent page**

In **MaternityCalculator.tsx** (anywhere inside the component, for example after you show the totals):

tsx

CopyEdit

<button

className="mt-4 px-4 py-2 bg-blue-600 text-white rounded"

onClick={() => {

// post a message to the parent (Shopify page)

window.parent.postMessage({ action: 'BUY\_PLAN' }, '\*');

}}

>

Buy Plan (£24.95)

</button>

*(Keep your existing “Calculate” button separate.)*

**5  Commit to Git and push to GitHub**

bash

CopyEdit

echo -e "node\_modules\ndist\n.env\*\n" > .gitignore

git init

git add .

git commit -m "Initial React calculator"

git branch -M main

git remote add origin https://github.com/<your-user>/teacher-maternity-calc.git

git push -u origin main

**6  Deploy to Vercel**

1. **Sign in** at vercel.com with GitHub.
2. **Add New Project** → import teacher‑maternity‑calc.
3. Build settings auto‑detected (Vite → npm run build, output dist).
4. Click **Deploy**.
5. In ≈ 1 min you get a URL like https://teacher-maternity-calc.vercel.app. Rename it in **Settings → Domains** (e.g., https://calc-tmh.vercel.app).

**7  Shopify setup**

**7‑A  Create the product**

* Products → **Add product** → “Personalised Maternity Plan PDF” → £24.95 → **Digital download** → Save.
* Grab the **Variant ID** (copy the long number at the end of the URL after /variants/).

**7‑B  Generate Buy Button**

* Sales Channels → **Buy Button** → Create → Product → select product.
* Style → *Minimal*, **Checkout** action.
* Click **Generate Code**. Copy the <script> block.

**8  Create the Shopify landing page**

Online Store → **Pages** → Add page → Title “Calculator” → switch to HTML editor, paste:

html

CopyEdit

<h1 style="text-align:center">Teacher Maternity Calculator</h1>

<iframe id="tmh-frame"

src="https://calc-tmh.vercel.app"

style="width:100%;max-width:1000px;height:1400px;border:none;margin:0 auto;display:block">

</iframe>

<!-- Shopify Buy-Button SDK -->

<script src="https://sdks.shopifycdn.com/buy-button/latest/buy-button-storefront.min.js"></script>

<!-- Paste the code Shopify generated BELOW this line -->

<!-- -------------- BEGIN SHOPIFY SNIPPET -------------- -->

<!-- (Replace <YOUR\_DOMAIN>, <TOKEN>, <VARIANT\_ID>) -->

<script>

var client = ShopifyBuy.buildClient({

domain: 'YOUR\_DOMAIN.myshopify.com',

storefrontAccessToken: '<TOKEN>',

});

ShopifyBuy.UI.onReady(client).then(function (ui) {

window.buyButton = ui.createComponent('product', {

id: 'gid://shopify/ProductVariant/<VARIANT\_ID>',

node: document.body.appendChild(document.createElement('div')),

options: {

product: {

iframe: false,

contents: { img: false, title: false, price: false },

text: { button: 'Buy Plan (£24.95)' },

templates: { button: '<button class=\"shopify-buy\_\_btn\">{{data.buttonText}}</button>' },

},

cart: { startOpen: false },

},

});

});

</script>

<!-- -------------- END SHOPIFY SNIPPET -------------- -->

<script>

/\* Listen for messages from iframe \*/

window.addEventListener('message', (e) => {

if (e.data?.action === 'BUY\_PLAN' && window.buyButton) {

// simulate a click on the hidden buy-button to open checkout

document.querySelector('.shopify-buy\_\_btn').click();

}

});

</script>

Save. View **/pages/calculator** → calculator loads in iframe → click “Buy Plan” → Shopify checkout opens.

**9  (Important) Fulfil PDF automatically**

Because static pages can’t run code:

1. **Add a Vercel serverless function** api/fulfil.ts (see previous reply).
2. Shopify Admin → Settings → Notifications → **Webhooks** → *Order Paid* → URL https://calc-tmh.vercel.app/api/fulfil.
3. The function regenerates the PDF (reuse calculation logic) and emails it via SendGrid or Resend.

*(Need exact code? Ask next and I’ll paste a working function.)*

**10  Polish**

| **Enhancement** | **Quick pointer** |
| --- | --- |
| **Iframe auto‑height** | In calculator, window.parent.postMessage({action:'HEIGHT', h: document.body.scrollHeight}, '\*'); ↔ parent sets iframe.style.height. |
| **SEO content** | Put a normal landing copy *above* the iframe so Google sees your keywords. |
| **GA events** | Fire gtag('event','calc\_complete', { value: gain }) in parent when receiving CALC\_DONE postMessage. |
| **Custom domain** | Vercel Settings → Domains → add calc.yourdomain.com → CNAME record in DNS. |