Smarter payout project set up

**🧱 1. Initial Project Setup (React App)**

bash

CopyEdit

npx create-react-app smarterpayouts

cd smarterpayouts

* This scaffolds a React project with essential config and structure.

**🧩 2. Project Structure Setup**

Created folders and files:

bash

CopyEdit

/src

/components

Navbar.jsx

Footer.jsx

FABSpeedDial.jsx

/pages

Index.jsx

About.jsx

Contact.jsx

Articles.jsx

GetAQuote.jsx

GetYourCash.jsx

ReviewOffer.jsx

CourtApproval.jsx

FAQs.jsx

Privacy.jsx

Terms.jsx

PricingCalculator.jsx

/Blog

StructuredSettlements.jsx

ShouldYouSell.jsx

HowFastPayout.jsx

**📦 3. Dependencies Installed**

bash

CopyEdit

npm install react-router-dom

Used for routing between pages.

**🧭 4. Routing Setup (App.js)**

Used react-router-dom to define routes:

jsx

CopyEdit

import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';

// Imported all page components

**💻 5. Page Building**

Each page used components like <Navbar />, <Footer />, <FABSpeedDial />. Examples include:

* Hero sections
* Responsive grids
* Accordions (FAQs)
* Cards (Articles)
* Contact forms
* Stats counters

**⚙️ 6. Calculator Logic**

Front-end in PricingCalculator.jsx used fetch:

jsx

CopyEdit

await fetch('/api/calculate', { method: 'POST', body: JSON.stringify({ futureValue, discountRate }) })

Back-end logic via Vercel function:

js

CopyEdit

// api/calculate.js

export default function handler(req, res) {

const { futureValue, discountRate } = req.body;

const presentValue = futureValue / (1 + discountRate);

res.status(200).json({ presentValue });

}

**☁️ 7. Vercel Serverless Setup**

* /api/calculate.js under the root or /api directory
* Auto-deployed with Vercel (no Express needed)

**🌐 8. Vercel Deployment**

Steps:

1. Created GitHub repo.
2. Pushed code via terminal:

bash

CopyEdit

git init

git add .

git commit -m "Initial commit"

git remote add origin https://github.com/YOUR\_USERNAME/smarterpayouts.git

git push -u origin main

1. Connected GitHub to Vercel.
2. Deployed the project (automatic detection).

**📈 9. Analytics & SEO Setup (Optional)**

* Google Analytics script added in public/index.html
* React Helmet planned for SEO/meta tags
* Sitemap to be submitted via Google Search Console

**✅ 10. Next Steps / Enhancements**

* Improve calculator logic (multi-period discounting)
* Add database or form submission backend (e.g., with Supabase or Firebase)
* Implement cookie consent
* Enable Analytics + structured data (schema)