{

"README.md": "# JustDuckIt Store - Complete Setup Guide\n\nThis project contains the complete code for your e-commerce store, featuring a \"Buy Now\" checkout flow using Stripe Embedded Checkout. The frontend is managed by Vite, and the backend is a secure Node.js server using Express.\n\n## Final File Structure\n\nYour project must be organized in this exact structure for it to work correctly:\n\n```\njustduckit-store/\n ├── .env\n ├── package.json\n ├── server.js\n ├── vite.config.js\n └── public/\n ├── index.html\n ├── index.js\n ├── checkout.html\n ├── checkout.js\n ├── done.html\n ├── done.js\n └── app.css\n```\n\n## Step-by-Step Setup Instructions\n\nFollow these steps exactly to get your store online for development.\n\n### 1. Unzip and Organize Files\n\n- Unzip this package into a new folder named `justduckit-store`.\n- Ensure all the files and the `public` folder are in the correct places as shown in the structure above.\n\n### 2. Get Your Stripe API Keys\n\n- Go to your [Stripe Dashboard](https://dashboard.stripe.com/test/apikeys).\n- Find your **Publishable key** (starts with `pk_test_...`).\n- Reveal and copy your **Secret key** (starts with `sk_test_...`).\n\n### 3. Configure Your Project Secrets\n\n- **Backend Key:** In the main `justduckit-store` folder, create a new file named `.env`.\n- Open the `.env` file and add your secret key:\n ```\n STRIPE_SECRET_KEY=sk_test_...your_actual_secret_key...\n ```\n- **Frontend Key:** Open `public/checkout.js`. Find the line `const stripe = await loadStripe('pk_test_...');` and replace the placeholder with your **Publishable key**.\n\n### 4. Create Products in Stripe\n\nThis is the most important step for making your products available for purchase.\n\n- Go to the [Products section](https://dashboard.stripe.com/test/products) in your Stripe Dashboard.\n- For **each** of your 6 items, click `+ Add product`.\n- Enter the product's name (e.g., \"Premium Decentralized Tee\") and price.\n- After saving, click on the product. Under the \"Pricing\" section, you will find the **Price ID** (it starts with `price_...`).\n- Copy this Price ID.\n- Open `public/index.js` and paste the ID into the corresponding product in the `products` array.\n\n**You must do this for all 6 products.**\n\n### 5. Install and Run the Application\n\n- Open your computer's terminal or command prompt.\n- Navigate into the `justduckit-store` folder.\n- **Install dependencies:** Run the command `npm install`.\n- **Run the project:** Run the command `npm run dev`.\n\nThis will start both your backend and frontend servers. A browser window should open automatically to `http://localhost:3000`. Your store is now live and ready for testing!\n",

".env": "STRIPE_SECRET_KEY=sk_test_...REPLACE_WITH_YOUR_SECRET_KEY...",

"package.json": "{\n \"name\": \"stripe-sample\",\n \"version\": \"0.1.0\",\n \"dependencies\": {\n \"@stripe/stripe-js\": \"^3.0.0\",\n \"cors\": \"^2.8.5\",\n \"dotenv\": \"^16.4.5\",\n \"express\": \"^4.19.2\",\n \"stripe\": \"^14.0.0\",\n \"vite\": \"^5.2.11\"\n },\n \"devDependencies\": {\n \"concurrently\": \"8.2.2\"\n },\n \"scripts\": {\n \"dev\": \"concurrently \\\"vite --port 3000 --open\\\" \\\"node server.js\\\"\",\n \"build\": \"vite build\"\n }\n}",

"server.js": "require(\"dotenv\").config();\nconst stripe = require(\"stripe\")(process.env.STRIPE_SECRET_KEY);\nconst express = require(\"express\");\nconst cors = require('cors'); // Added for development\nconst app = express();\nconst router = express.Router();\n\n// --- Middleware ---\napp.use(cors({ origin: 'http://localhost:3000' })); \napp.use(express.static(\"public\"));\napp.use(express.urlencoded({ extended: true }));\napp.use(express.json());\n\n// --- API Routes ---\nrouter.post(\"/create-checkout-session\", async (req, res) => {\n const { priceId } = req.body;\n\n if (!priceId) {\n return res.status(400).send({ error: 'A priceId is required to create a session.' });\n }\n\n try {\n const session = await stripe.checkout.sessions.create({\n line_items: [\n {\n price: priceId,\n quantity: 1,\n },\n],\n mode: \"payment\",\n ui_mode: \"embedded\",\n return_url: `${process.env.DOMAIN || 'http://localhost:3000'}/done.html?session_id={CHECKOUT_SESSION_ID}`,\n });\n\n res.send({ clientSecret:

session.client_secret });\n } catch (e) {\n   res.status(500).json({ error: e.message });\n }\n});\n\nrouter.get(\"/session-status\", async (req, res) => {\n const { session_id } = req.query;\n try {\n   const session = await stripe.checkout.sessions.retrieve(session_id);\n   res.send({ status: session.status });\n } catch (e) {\n   res.status(500).json({ error: e.message });\n }\n});\n\napp.use(\"/api\", router);\n\napp.listen(4242, () => console.log(\"Server running on http://localhost:4242\"));",

"vite.config.js": "import { defineConfig } from \"vite\";\nimport fs from \"fs\";\nimport path from \"path\";\n\n// Convert a URL to an html pathname with the parameters\nfunction htmlFallbackPlugin() {\n return {\n   name: \"html-fallback\",\n   configureServer(server) {\n   return () => {\n   server.middlewares.use((req, res, next) => {\n   const url = new URL(req.url, `http://${req.headers.host}`);\n   const pathname = url.pathname;\n\n   if (pathname.match(/^\\/\\w+$/)) {\n   const htmlPath = path.join(\n   process.cwd(),\n   \"public\",\n   pathname + \".html\"\n   );\n   if (fs.existsSync(htmlPath)) {\n   url.pathname += \".html\";\n   req.url = url.pathname + url.search;\n   }\n   }\n   next();\n   });\n   };\n   },\n   resolveId(id) {\n   const [pathname] = id.split(\"?\");\n   if (pathname.match(/^\\/\\w+$/)) {\n   const htmlPath = path.join(process.cwd(), \"public\", pathname + \".html\");\n   if (fs.existsSync(htmlPath)) {\n   return htmlPath;\n   }\n   }\n   return null;\n   },\n };\n}\n\nexport default defineConfig({\n root: \"public\",\n envDir: \"./..\",\n plugins: [htmlFallbackPlugin()],\n server: {\n   // Proxy API requests to server\n   proxy: {\n   \"/api\": \"http://localhost:4242\",\n   },\n },\n});",

"public/app.css": "body {\n background: #242d60;\n font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto',\n 'Helvetica Neue', 'Ubuntu', sans-serif;\n height: 100vh;\n margin: 0;\n -webkit-font-smoothing: antialiased;\n -moz-osx-font-smoothing: grayscale;\n}\n\n.App {\n display: flex;\n justify-content: center;\n align-items: center;\n width: 100%;\n min-height: 100vh;\n padding: 20px 0;\n}\n\n.container {\n display: flex;\n flex-direction: column;\n width: 400px;\n gap: 24px;\n}\n\n.logo {\n color: white;\n font-weight: 500;\n font-size: 24px;\n color: #e0e3f0;\n}\n\n.product {\n border-radius: 6px;\n overflow: hidden;\n}\n\n.product-info {\n display: flex;\n flex: 1;\n padding: 12px;\n background: #ffffff;\n}\n\n.description {\n display: flex;\n flex-direction: column;\n justify-content: center;\n}\n\nimg {\n border-radius: 6px;\n width: 54px;\n height: 57px;\n margin-right: 10px\n}\n\n.button {\n height: 42px;\n background: #556cd6;\n color: white;\n width: 100%;\n border-radius: 6px;\n font-size: 14px;\n border: 0;\n font-weight: 500;\n cursor: pointer;\n transition: all 0.2s ease;\n text-align: center;\n display: flex;\n justify-content: center;\n align-items: center;\n text-decoration: none;\n font-family: inherit\n}\n\n.button:hover {\n opacity: 0.8;\n}\n\n.product .button {\n height: 36px;\n border-radius: 0;\n}\n\np {\n font-style: normal;\n font-weight: 500;\n font-size: 14px;\n line-height: 20px;\n letter-spacing: -0.154px;\n color: #242d60;\n height: 100%;\n width: 100%;\n margin: 0;\n display: flex;\n align-items: center;\n justify-content: center;\n box-sizing: border-box;\n}\n\nh3,\nh5 {\n font-style: normal;\n font-weight: 500;\n font-size: 14px;\n line-height: 20px;\n letter-spacing: -0.154px;\n color: #242d60;\n margin: 0;\n}\n\nh5 {\n opacity: 0.5;\n}\n\n.message {\n color: #fff;\n font-size: 16px;\n}\n\n.checkout {\n width: 100%;\n align-self: flex-start;\n background-color: white;\n margin: 40px;\n padding: 20px;\n border-radius: 16px;\n}\n\n.hidden {\n   display: none;\n}",

"public/index.html": "<!DOCTYPE html>\n<html lang=\"en\">\n <head>\n   <meta charset=\"UTF-8\" />\n   <link rel=\"icon\" type=\"image/svg+xml\" href=\"/favicon.ico\" />\n   <link rel=\"stylesheet\" href=\"./app.css\">\n   <meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0\" />\n   <title>JustDuckit</title>\n   <script type=\"module\" src=\"./index.js\"></script>\n </head>\n <body>\n   <div class=\"App\">\n   <div class=\"container\">\n   <div class=\"logo\">JustDuckit \n   \n   \n   \n </body>\n</html>",

"public/index.js": "// This is the new index.js file for a \"Buy Now\" flow.\n\n// --- DATA & STATE ---\n// TODO: You must create these products in your Stripe Dashboard and get their Price IDs.\nconst products = [\n { name: 'Premium Decentralized Tee', price: 39.00, image: 'https://i.imgur.com/U9e6xWi.png', priceId: 'price_1P...' },\n { name: 'Classic Decentralized Tee', price: 24.50, image: 'https://i.imgur.com/FYS3LsC.png', priceId: 'price_1P...' },\n { name: 'Quack Head Hat', price: 21.95, image: 'https://i.imgur.com/PtDFXyR.png', priceId: 'price_1P...' },\n { name: 'Decentralized Sanctuary Hat', price: 21.95, image: 'https://i.imgur.com/z2kDC2D.png', priceId: 'price_1P...' },\n { name: 'Decent Quack Socks', price: 13.05, image: 'https://i.imgur.com/mbfFxEp.png', priceId: 'price_1P...' },\n { name: 'Duck Pfp Canvas', price: 27.00, image: 'https://i.imgur.com/afjkjvx.png', priceId: 'price_1P...' }\n];\n\n// --- FUNCTIONS ---\n\n// Function to render all products on the page\nfunction renderProducts(container) {\n products.forEach(product => {\n const productDiv = document.createElement('div');\n productDiv.className = 'product';\n productDiv.innerHTML = `\n <div class=\"product-info\">\n <img src=\"${product.image}\" alt=\"${product.name}\" />\n <div class=\"description\">\n <h3>${product.name}</h3>\n <h5>$${product.price.toFixed(2)}</h5>\n </div>\n </div>\n <button class=\"button\" onclick=\"buyNow('${product.priceId}')\">Buy Now</button>\n `;\n container.appendChild(productDiv);\n });\n}\n\n// Function to redirect to checkout with a specific product's priceId\nwindow.buyNow = function(priceId) {\n if (!priceId || priceId === 'price_1P...') {\n alert('This product is not configured for sale yet. Please set the Price ID in index.js.');\n return;\n }\n // Redirect to the checkout page, passing the priceId in the URL\n window.location.href = `/checkout.html?priceId=${priceId}`;\n}\n\n// --- INITIALIZATION ---\nwindow.addEventListener('DOMContentLoaded', () => {\n const container = document.querySelector('.container');\n\n // Create and append product list container\n const productList = document.createElement('div');\n productList.id = 'product-list';\n productList.className = 'container';\n productList.style.gap = '12px';\n container.appendChild(productList);\n \n // Render initial content\n renderProducts(productList);\n});",

"public/checkout.html": "<!DOCTYPE html>\n<html lang=\"en\">\n <head>\n <meta charset=\"utf-8\" />\n <link rel=\"icon\" type=\"image/svg+xml\" href=\"/favicon.ico\" />\n <link rel=\"stylesheet\" href=\"./app.css\">\n <meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0\" />\n <title>JustDuckit - Checkout</title>\n <script type=\"module\" src=\"./checkout.js\" defer></script>\n </head>\n <body>\n <div class=\"App\">\n <div class=\"container\">\n <div id=\"checkout\">\n \n \n <div id=\"return-message\" class=\"message\"> \n \n \n </body>\n</html>",

"public/checkout.js": "import { loadStripe } from \"@stripe/stripe-js\";\n\n// TODO: Replace with your actual Stripe publishable key\nconst stripe = await loadStripe('pk_test_...REPLACE_WITH_YOUR_PUBLISHABLE_KEY...');\n\nasync function initialize() {\n const messageContainer = document.getElementById('return-message');\n\n const fetchClientSecret = async () => {\n // Get priceId from the URL\n const urlParams = new URLSearchParams(window.location.search);\n const priceId = urlParams.get('priceId');\n\n if (!priceId) {\n if (messageContainer) messageContainer.textContent = \"No product selected. Please go back and choose a product.\";\n return null;\n }\n\n // Fetch the client secret from your server\n const response = await fetch(\"/api/create-checkout-session\", {\n method: \"POST\",\n headers: { \"Content-Type\": \"application/json\" },\n body: JSON.stringify({ priceId }), // Pass the single priceId\n });\n\n if (!response.ok) {\n const { error } = await response.json();\n throw new Error(`Server error: ${error || 'Failed to create session.'}`);\n }\n\n const { clientSecret } = await response.json();\n return clientSecret;\n };\n\n try {\n const clientSecret = await fetchClientSecret();\n if (clientSecret) {\n const checkout = await stripe.initEmbeddedCheckout({\n clientSecret,\n });\n // Mount Checkout\n checkout.mount('#checkout');\n }\n } catch (error) {\n console.error(error);\n if (messageContainer) messageContainer.textContent = error.message;\n }\n}\n\ninitialize();",

"public/done.html": "<!DOCTYPE html>\n<html>\n <head>\n   <meta charset=\"UTF-8\" />\n   <link rel=\"icon\" type=\"image/svg+xml\" href=\"/favicon.ico\" />\n   <link rel=\"stylesheet\" href=\"./app.css\">\n   <meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0\" />\n   <title>JustDuckIt - Thank You</title>\n   <script type=\"module\" src=\"./done.js\"></script>\n </head>\n\n <body>\n   <div class=\"App\">\n   <div class=\"container\">\n   <p id=\"success\" class=\"message hidden\">Your purchase was successful</p>\n   Back to products\n   \n   \n </body>\n</html>",

"public/done.js": "// This is the new done.js file.\n\ninitialize();\n\nasync function initialize() {\n const queryString = window.location.search;\n const urlParams = new URLSearchParams(queryString);\n const sessionId = urlParams.get('session_id');\n\n // If there's no session ID, we can't check the status.\n if (!sessionId) {\n   // Redirect to home or show an error.\n   window.location.replace('http://localhost:3000/');\n   return;\n }\n\n const response = await fetch(`/api/session-status?session_id=${sessionId}`);\n const session = await response.json();\n\n if (session.status == 'open') {\n   // This means the payment was not completed. Redirect back to the homepage.\n   window.location.replace('http://localhost:3000/');\n } else if (session.status == 'complete') {\n   // The payment was successful. Show the success message.\n   document.getElementById('success').classList.remove('hidden');\n }\n}"

}