**API key**:

sk-proj-\_G0sJL7j3THFHoin0adsW4i6ZJZ-XU02BHqA\_ZyQS0M4VVbSqoKS86xBx-BvCvsrs9qJkg1TrbT3BlbkFJxYqGN5tz8Sn65nerOekjt1e8SIL9JE1oHKachttra5u4jgfUXhiWtEWZk8E\_g0o\_WNjud35poA

Below is:

* Text on screen when I click on the Render server webpage:
* The logs from Chrome (F12) outlining the issue
* The main scripts for the website

**Text on screen when I click on the Render server webpage:**

Cannot GET /

**Chrome logs**

Request URL:

https://sxr-backend.onrender.com/generate-prompts

Request Method:

POST

Status Code:

500 Internal Server Error

Remote Address:

216.24.57.4:443

Referrer Policy:

strict-origin-when-cross-origin

Response Headers

Request Headers

Console

What's new

AI assistance

Coverage

Highlights from the Chrome 131 update

**Debug CSS with Gemini**

Chat with Gemini right in the new AI assistance panel and get help with styling your page.

**Annotate performance findings**

Label items and time ranges, draw connections between items, save and share annotated traces in the Performance panel.

**Ignore listing improvements**

Stack trace now hides ignored frames and you can now set DevTools to ignore anonymous scripts.

**Get performance insights**

Experimental: Discover actionable insights right in the Performance panel, consolidated from the Lighthouse report and (soon to be deprecated) Performance insights panel.

Learn more Close

Network Log Row Name: generate-promptssxr-backend.onrender.com, Status: 500Internal Server Error, Type: fetch, Initiator: Script, Size: 314 B38 B, Time: 420 ms420 ms

Payload

1. {equation: "2+3"}
   1. equation: "2+3"

Response

{"error":"Failed to generate prompts"}

Render logs (for ‘sxr-backend’)

**==>** **Running 'node server.js'**

Dec 12 04:40:10 PMServer running on [http://localhost:3000](http://localhost:3000/)

Dec 12 04:47:51 PMError: Request failed with status code 404

Dec 12 04:47:51 PM at createError (/opt/render/project/src/node\_modules/axios/lib/core/createError.js:16:15)

Dec 12 04:47:51 PM at settle (/opt/render/project/src/node\_modules/axios/lib/core/settle.js:17:12)

Dec 12 04:47:51 PM at IncomingMessage.handleStreamEnd (/opt/render/project/src/node\_modules/axios/lib/adapters/http.js:322:11)

Dec 12 04:47:51 PM at IncomingMessage.emit (node:events:530:35)

Dec 12 04:47:51 PM at endReadableNT (node:internal/streams/readable:1698:12)

Dec 12 04:47:51 PM at process.processTicksAndRejections (node:internal/process/task\_queues:90:21) {

Dec 12 04:47:51 PM config: {

Dec 12 04:47:51 PM transitional: {

Dec 12 04:47:51 PM silentJSONParsing: true,

Dec 12 04:47:51 PM forcedJSONParsing: true,

Dec 12 04:47:51 PM clarifyTimeoutError: false

Dec 12 04:47:51 PM },

Dec 12 04:47:51 PM adapter: [Function: httpAdapter],

Dec 12 04:47:51 PM transformRequest: [ [Function: transformRequest] ],

Dec 12 04:47:51 PM transformResponse: [ [Function: transformResponse] ],

Dec 12 04:47:51 PM timeout: 0,

Dec 12 04:47:51 PM xsrfCookieName: 'XSRF-TOKEN',

Dec 12 04:47:51 PM xsrfHeaderName: 'X-XSRF-TOKEN',

Dec 12 04:47:51 PM maxContentLength: -1,

Dec 12 04:47:51 PM maxBodyLength: -1,

Dec 12 04:47:51 PM validateStatus: [Function: validateStatus],

Dec 12 04:47:51 PM headers: {

Dec 12 04:47:51 PM Accept: 'application/json, text/plain, \*/\*',

Dec 12 04:47:51 PM 'Content-Type': 'application/json',

Dec 12 04:47:51 PM 'User-Agent': 'OpenAI/NodeJS/3.3.0',

Dec 12 04:47:51 PM Authorization: 'Bearer sk-proj-iMDtoJ5b1Oy4o3N4Cm9D5tiEDI0LLR21JrhEtSIuwng94Q5T1B2\_yjlegC\_mjCbObQmdAe1yiOT3BlbkFJg-9ys3PE4dS-Vopb6LTSEuEeLfsE0hiwHz5h2jCqIG1So9evMGS2d3xR4WMlCwI9VSHVeV5A4A',

Dec 12 04:47:51 PM 'Content-Length': 469

Dec 12 04:47:51 PM },

Dec 12 04:47:51 PM method: 'post',

Dec 12 04:47:51 PM data: '{"model":"gpt-4","messages":[{"role":"system","content":"\\n You are a guide for Shepherd Math, a VR game that teaches middle school math to kids. \\n Your audience is adult instructors helping kids navigate math problems in VR headsets.\\n Ensure safeguarding priorities and focus on clarity, engagement, and the mechanics of the VR game.\\n "},{"role":"user","content":"Provide 3 helpful teaching prompts for the equation: 2+3"}],"max\_tokens":200}',

Dec 12 04:47:51 PM url: 'https://api.openai.com/v1/chat/completions'

Dec 12 04:47:51 PM },

Dec 12 04:47:51 PM request: <ref \*1> ClientRequest {

Dec 12 04:47:51 PM \_events: [Object: null prototype] {

Dec 12 04:47:51 PM abort: [Function (anonymous)],

Dec 12 04:47:51 PM aborted: [Function (anonymous)],

Dec 12 04:47:51 PM connect: [Function (anonymous)],

Dec 12 04:47:51 PM error: [Function (anonymous)],

Dec 12 04:47:51 PM socket: [Function (anonymous)],

Dec 12 04:47:51 PM timeout: [Function (anonymous)],

Dec 12 04:47:51 PM finish: [Function: requestOnFinish]

Dec 12 04:47:51 PM },

Dec 12 04:47:51 PM \_eventsCount: 7,

Dec 12 04:47:51 PM \_maxListeners: undefined,

Dec 12 04:47:51 PM outputData: [],

Dec 12 04:47:51 PM outputSize: 0,

Dec 12 04:47:51 PM writable: true,

Dec 12 04:47:51 PM destroyed: true,

Dec 12 04:47:51 PM \_last: false,

Dec 12 04:47:51 PM chunkedEncoding: false,

Dec 12 04:47:51 PM shouldKeepAlive: true,

Dec 12 04:47:51 PM maxRequestsOnConnectionReached: false,

Dec 12 04:47:51 PM \_defaultKeepAlive: true,

Dec 12 04:47:51 PM useChunkedEncodingByDefault: true,

Dec 12 04:47:51 PM sendDate: false,

Dec 12 04:47:51 PM \_removedConnection: false,

Dec 12 04:47:51 PM \_removedContLen: false,

Dec 12 04:47:51 PM \_removedTE: false,

Dec 12 04:47:51 PM strictContentLength: false,

Dec 12 04:47:51 PM \_contentLength: 469,

Dec 12 04:47:51 PM \_hasBody: true,

Dec 12 04:47:51 PM \_trailer: '',

Dec 12 04:47:51 PM finished: true,

Dec 12 04:47:51 PM \_headerSent: true,

Dec 12 04:47:51 PM \_closed: true,

Dec 12 04:47:51 PM \_header: 'POST /v1/chat/completions HTTP/1.1\r\n' +

Dec 12 04:47:51 PM 'Accept: application/json, text/plain, \*/\*\r\n' +

Dec 12 04:47:51 PM 'Content-Type: application/json\r\n' +

Dec 12 04:47:51 PM 'User-Agent: OpenAI/NodeJS/3.3.0\r\n' +

Dec 12 04:47:51 PM 'Authorization: Bearer sk-proj-iMDtoJ5b1Oy4o3N4Cm9D5tiEDI0LLR21JrhEtSIuwng94Q5T1B2\_yjlegC\_mjCbObQmdAe1yiOT3BlbkFJg-9ys3PE4dS-Vopb6LTSEuEeLfsE0hiwHz5h2jCqIG1So9evMGS2d3xR4WMlCwI9VSHVeV5A4A\r\n' +

Dec 12 04:47:51 PM 'Content-Length: 469\r\n' +

Dec 12 04:47:51 PM 'Host: api.openai.com\r\n' +

Dec 12 04:47:51 PM 'Connection: keep-alive\r\n' +

Dec 12 04:47:51 PM '\r\n',

Dec 12 04:47:51 PM \_keepAliveTimeout: 0,

Dec 12 04:47:51 PM \_onPendingData: [Function: nop],

Dec 12 04:47:51 PM agent: Agent {

Dec 12 04:47:51 PM \_events: [Object: null prototype],

Dec 12 04:47:51 PM \_eventsCount: 2,

Dec 12 04:47:51 PM \_maxListeners: undefined,

Dec 12 04:47:51 PM defaultPort: 443,

Dec 12 04:47:51 PM protocol: 'https:',

Dec 12 04:47:51 PM options: [Object: null prototype],

Dec 12 04:47:51 PM requests: [Object: null prototype] {},

Dec 12 04:47:51 PM sockets: [Object: null prototype] {},

Dec 12 04:47:51 PM freeSockets: [Object: null prototype],

Dec 12 04:47:51 PM keepAliveMsecs: 1000,

Dec 12 04:47:51 PM keepAlive: true,

Dec 12 04:47:51 PM maxSockets: Infinity,

Dec 12 04:47:51 PM maxFreeSockets: 256,

Dec 12 04:47:51 PM scheduling: 'lifo',

Dec 12 04:47:51 PM maxTotalSockets: Infinity,

Dec 12 04:47:51 PM totalSocketCount: 1,

Dec 12 04:47:51 PM maxCachedSessions: 100,

Dec 12 04:47:51 PM \_sessionCache: [Object],

Dec 12 04:47:51 PM [Symbol(shapeMode)]: false,

Dec 12 04:47:51 PM [Symbol(kCapture)]: false

Dec 12 04:47:51 PM },

Dec 12 04:47:51 PM socketPath: undefined,

Dec 12 04:47:51 PM method: 'POST',

Dec 12 04:47:51 PM maxHeaderSize: undefined,

Dec 12 04:47:51 PM insecureHTTPParser: undefined,

Dec 12 04:47:51 PM joinDuplicateHeaders: undefined,

Dec 12 04:47:51 PM path: '/v1/chat/completions',

Dec 12 04:47:51 PM \_ended: true,

Dec 12 04:47:51 PM res: IncomingMessage {

Dec 12 04:47:51 PM \_events: [Object],

Dec 12 04:47:51 PM \_readableState: [ReadableState],

Dec 12 04:47:51 PM \_maxListeners: undefined,

Dec 12 04:47:51 PM socket: null,

Dec 12 04:47:51 PM httpVersionMajor: 1,

Dec 12 04:47:51 PM httpVersionMinor: 1,

Dec 12 04:47:51 PM httpVersion: '1.1',

Dec 12 04:47:51 PM complete: true,

Dec 12 04:47:51 PM rawHeaders: [Array],

Dec 12 04:47:51 PM rawTrailers: [],

Dec 12 04:47:51 PM joinDuplicateHeaders: undefined,

Dec 12 04:47:51 PM aborted: false,

Dec 12 04:47:51 PM upgrade: false,

Dec 12 04:47:51 PM url: '',

Dec 12 04:47:51 PM method: null,

Dec 12 04:47:51 PM statusCode: 404,

Dec 12 04:47:51 PM statusMessage: 'Not Found',

Dec 12 04:47:51 PM client: [TLSSocket],

Dec 12 04:47:51 PM \_consuming: false,

Dec 12 04:47:51 PM \_dumped: false,

Dec 12 04:47:51 PM req: [Circular \*1],

Dec 12 04:47:51 PM \_eventsCount: 4,

Dec 12 04:47:51 PM responseUrl: 'https://api.openai.com/v1/chat/completions',

Dec 12 04:47:51 PM redirects: [],

Dec 12 04:47:51 PM [Symbol(shapeMode)]: true,

Dec 12 04:47:51 PM [Symbol(kCapture)]: false,

Dec 12 04:47:51 PM [Symbol(kHeaders)]: [Object],

Dec 12 04:47:51 PM [Symbol(kHeadersCount)]: 28,

Dec 12 04:47:51 PM [Symbol(kTrailers)]: null,

Dec 12 04:47:51 PM [Symbol(kTrailersCount)]: 0

Dec 12 04:47:51 PM },

Dec 12 04:47:51 PM aborted: false,

Dec 12 04:47:51 PM timeoutCb: null,

Dec 12 04:47:51 PM upgradeOrConnect: false,

Dec 12 04:47:51 PM parser: null,

Dec 12 04:47:51 PM maxHeadersCount: null,

Dec 12 04:47:51 PM reusedSocket: false,

Dec 12 04:47:51 PM host: 'api.openai.com',

Dec 12 04:47:51 PM protocol: 'https:',

Dec 12 04:47:51 PM \_redirectable: Writable {

Dec 12 04:47:51 PM \_events: [Object],

Dec 12 04:47:51 PM \_writableState: [WritableState],

Dec 12 04:47:51 PM \_maxListeners: undefined,

Dec 12 04:47:51 PM \_options: [Object],

Dec 12 04:47:51 PM \_ended: true,

Dec 12 04:47:51 PM \_ending: true,

Dec 12 04:47:51 PM \_redirectCount: 0,

Dec 12 04:47:51 PM \_redirects: [],

Dec 12 04:47:51 PM \_requestBodyLength: 469,

Dec 12 04:47:51 PM \_requestBodyBuffers: [],

Dec 12 04:47:51 PM \_eventsCount: 3,

Dec 12 04:47:51 PM \_onNativeResponse: [Function (anonymous)],

Dec 12 04:47:51 PM \_currentRequest: [Circular \*1],

Dec 12 04:47:51 PM \_currentUrl: 'https://api.openai.com/v1/chat/completions',

Dec 12 04:47:51 PM [Symbol(shapeMode)]: true,

Dec 12 04:47:51 PM [Symbol(kCapture)]: false

Dec 12 04:47:51 PM },

Dec 12 04:47:51 PM [Symbol(shapeMode)]: false,

Dec 12 04:47:51 PM [Symbol(kCapture)]: false,

Dec 12 04:47:51 PM [Symbol(kBytesWritten)]: 0,

Dec 12 04:47:51 PM [Symbol(kNeedDrain)]: false,

Dec 12 04:47:51 PM [Symbol(corked)]: 0,

Dec 12 04:47:51 PM [Symbol(kChunkedBuffer)]: [],

Dec 12 04:47:51 PM [Symbol(kChunkedLength)]: 0,

Dec 12 04:47:51 PM [Symbol(kSocket)]: TLSSocket {

Dec 12 04:47:51 PM \_tlsOptions: [Object],

Dec 12 04:47:51 PM \_secureEstablished: true,

Dec 12 04:47:51 PM \_securePending: false,

Dec 12 04:47:51 PM \_newSessionPending: false,

Dec 12 04:47:51 PM \_controlReleased: true,

Dec 12 04:47:51 PM secureConnecting: false,

Dec 12 04:47:51 PM \_SNICallback: null,

Dec 12 04:47:51 PM servername: 'api.openai.com',

Dec 12 04:47:51 PM alpnProtocol: false,

Dec 12 04:47:51 PM authorized: true,

Dec 12 04:47:51 PM authorizationError: null,

Dec 12 04:47:51 PM encrypted: true,

Dec 12 04:47:51 PM \_events: [Object: null prototype],

Dec 12 04:47:51 PM \_eventsCount: 9,

Dec 12 04:47:51 PM connecting: false,

Dec 12 04:47:51 PM \_hadError: false,

Dec 12 04:47:51 PM \_parent: null,

Dec 12 04:47:51 PM \_host: 'api.openai.com',

Dec 12 04:47:51 PM \_closeAfterHandlingError: false,

Dec 12 04:47:51 PM \_readableState: [ReadableState],

Dec 12 04:47:51 PM \_writableState: [WritableState],

Dec 12 04:47:51 PM allowHalfOpen: false,

Dec 12 04:47:51 PM \_maxListeners: undefined,

Dec 12 04:47:51 PM \_sockname: null,

Dec 12 04:47:51 PM \_pendingData: null,

Dec 12 04:47:51 PM \_pendingEncoding: '',

Dec 12 04:47:51 PM server: undefined,

Dec 12 04:47:51 PM \_server: null,

Dec 12 04:47:51 PM ssl: [TLSWrap],

Dec 12 04:47:51 PM \_requestCert: true,

Dec 12 04:47:51 PM \_rejectUnauthorized: true,

Dec 12 04:47:51 PM timeout: 5000,

Dec 12 04:47:51 PM parser: null,

Dec 12 04:47:51 PM \_httpMessage: null,

Dec 12 04:47:51 PM autoSelectFamilyAttemptedAddresses: [Array],

Dec 12 04:47:51 PM [Symbol(alpncallback)]: null,

Dec 12 04:47:51 PM [Symbol(res)]: [TLSWrap],

Dec 12 04:47:51 PM [Symbol(verified)]: true,

Dec 12 04:47:51 PM [Symbol(pendingSession)]: null,

Dec 12 04:47:51 PM [Symbol(async\_id\_symbol)]: -1,

Dec 12 04:47:51 PM [Symbol(kHandle)]: [TLSWrap],

Dec 12 04:47:51 PM [Symbol(lastWriteQueueSize)]: 0,

Dec 12 04:47:51 PM [Symbol(timeout)]: Timeout {

Dec 12 04:47:51 PM \_idleTimeout: 5000,

Dec 12 04:47:51 PM \_idlePrev: [TimersList],

Dec 12 04:47:51 PM \_idleNext: [TimersList],

Dec 12 04:47:51 PM \_idleStart: 462228,

Dec 12 04:47:51 PM \_onTimeout: [Function: bound ],

Dec 12 04:47:51 PM \_timerArgs: undefined,

Dec 12 04:47:51 PM \_repeat: null,

Dec 12 04:47:51 PM \_destroyed: false,

Dec 12 04:47:51 PM [Symbol(refed)]: false,

Dec 12 04:47:51 PM [Symbol(kHasPrimitive)]: false,

Dec 12 04:47:51 PM [Symbol(asyncId)]: 924,

Dec 12 04:47:51 PM [Symbol(triggerId)]: 922,

Dec 12 04:47:51 PM [Symbol(kAsyncContextFrame)]: undefined

Dec 12 04:47:51 PM },

Dec 12 04:47:51 PM [Symbol(kBuffer)]: null,

Dec 12 04:47:51 PM [Symbol(kBufferCb)]: null,

Dec 12 04:47:51 PM [Symbol(kBufferGen)]: null,

Dec 12 04:47:51 PM [Symbol(shapeMode)]: true,

Dec 12 04:47:51 PM [Symbol(kCapture)]: false,

Dec 12 04:47:51 PM [Symbol(kSetNoDelay)]: false,

Dec 12 04:47:51 PM [Symbol(kSetKeepAlive)]: true,

Dec 12 04:47:51 PM [Symbol(kSetKeepAliveInitialDelay)]: 1,

Dec 12 04:47:51 PM [Symbol(kBytesRead)]: 0,

Dec 12 04:47:51 PM [Symbol(kBytesWritten)]: 0,

Dec 12 04:47:51 PM [Symbol(connect-options)]: [Object]

Dec 12 04:47:51 PM },

Dec 12 04:47:51 PM [Symbol(kOutHeaders)]: [Object: null prototype] {

Dec 12 04:47:51 PM accept: [Array],

Dec 12 04:47:51 PM 'content-type': [Array],

Dec 12 04:47:51 PM 'user-agent': [Array],

Dec 12 04:47:51 PM authorization: [Array],

Dec 12 04:47:51 PM 'content-length': [Array],

Dec 12 04:47:51 PM host: [Array]

Dec 12 04:47:51 PM },

Dec 12 04:47:51 PM [Symbol(errored)]: null,

Dec 12 04:47:51 PM [Symbol(kHighWaterMark)]: 65536,

Dec 12 04:47:51 PM [Symbol(kRejectNonStandardBodyWrites)]: false,

Dec 12 04:47:51 PM [Symbol(kUniqueHeaders)]: null

Dec 12 04:47:51 PM },

Dec 12 04:47:51 PM response: {

Dec 12 04:47:51 PM status: 404,

Dec 12 04:47:51 PM statusText: 'Not Found',

Dec 12 04:47:51 PM headers: {

Dec 12 04:47:51 PM date: 'Thu, 12 Dec 2024 21:47:51 GMT',

Dec 12 04:47:51 PM 'content-type': 'application/json; charset=utf-8',

Dec 12 04:47:51 PM 'content-length': '211',

Dec 12 04:47:51 PM connection: 'keep-alive',

Dec 12 04:47:51 PM vary: 'Origin',

Dec 12 04:47:51 PM 'x-request-id': 'req\_1372412d066628c5e4b3bfebecca7e2e',

Dec 12 04:47:51 PM 'strict-transport-security': 'max-age=31536000; includeSubDomains; preload',

Dec 12 04:47:51 PM 'cf-cache-status': 'DYNAMIC',

Dec 12 04:47:51 PM 'set-cookie': [Array],

Dec 12 04:47:51 PM 'x-content-type-options': 'nosniff',

Dec 12 04:47:51 PM server: 'cloudflare',

Dec 12 04:47:51 PM 'cf-ray': '8f10ea50be81e5f0-IAD',

Dec 12 04:47:51 PM 'alt-svc': 'h3=":443"; ma=86400'

Dec 12 04:47:51 PM },

Dec 12 04:47:51 PM config: {

Dec 12 04:47:51 PM transitional: [Object],

Dec 12 04:47:51 PM adapter: [Function: httpAdapter],

Dec 12 04:47:51 PM transformRequest: [Array],

Dec 12 04:47:51 PM transformResponse: [Array],

Dec 12 04:47:51 PM timeout: 0,

Dec 12 04:47:51 PM xsrfCookieName: 'XSRF-TOKEN',

Dec 12 04:47:51 PM xsrfHeaderName: 'X-XSRF-TOKEN',

Dec 12 04:47:51 PM maxContentLength: -1,

Dec 12 04:47:51 PM maxBodyLength: -1,

Dec 12 04:47:51 PM validateStatus: [Function: validateStatus],

Dec 12 04:47:51 PM headers: [Object],

Dec 12 04:47:51 PM method: 'post',

Dec 12 04:47:51 PM data: '{"model":"gpt-4","messages":[{"role":"system","content":"\\n You are a guide for Shepherd Math, a VR game that teaches middle school math to kids. \\n Your audience is adult instructors helping kids navigate math problems in VR headsets.\\n Ensure safeguarding priorities and focus on clarity, engagement, and the mechanics of the VR game.\\n "},{"role":"user","content":"Provide 3 helpful teaching prompts for the equation: 2+3"}],"max\_tokens":200}',

Dec 12 04:47:51 PM url: 'https://api.openai.com/v1/chat/completions'

Dec 12 04:47:51 PM },

Dec 12 04:47:51 PM request: <ref \*1> ClientRequest {

Dec 12 04:47:51 PM \_events: [Object: null prototype],

Dec 12 04:47:51 PM \_eventsCount: 7,

Dec 12 04:47:51 PM \_maxListeners: undefined,

Dec 12 04:47:51 PM outputData: [],

Dec 12 04:47:51 PM outputSize: 0,

Dec 12 04:47:51 PM writable: true,

Dec 12 04:47:51 PM destroyed: true,

Dec 12 04:47:51 PM \_last: false,

Dec 12 04:47:51 PM chunkedEncoding: false,

Dec 12 04:47:51 PM shouldKeepAlive: true,

Dec 12 04:47:51 PM maxRequestsOnConnectionReached: false,

Dec 12 04:47:51 PM \_defaultKeepAlive: true,

Dec 12 04:47:51 PM useChunkedEncodingByDefault: true,

Dec 12 04:47:51 PM sendDate: false,

Dec 12 04:47:51 PM \_removedConnection: false,

Dec 12 04:47:51 PM \_removedContLen: false,

Dec 12 04:47:51 PM \_removedTE: false,

Dec 12 04:47:51 PM strictContentLength: false,

Dec 12 04:47:51 PM \_contentLength: 469,

Dec 12 04:47:51 PM \_hasBody: true,

Dec 12 04:47:51 PM \_trailer: '',

Dec 12 04:47:51 PM finished: true,

Dec 12 04:47:51 PM \_headerSent: true,

Dec 12 04:47:51 PM \_closed: true,

Dec 12 04:47:51 PM \_header: 'POST /v1/chat/completions HTTP/1.1\r\n' +

Dec 12 04:47:51 PM 'Accept: application/json, text/plain, \*/\*\r\n' +

Dec 12 04:47:51 PM 'Content-Type: application/json\r\n' +

Dec 12 04:47:51 PM 'User-Agent: OpenAI/NodeJS/3.3.0\r\n' +

Dec 12 04:47:51 PM 'Authorization: Bearer sk-proj-iMDtoJ5b1Oy4o3N4Cm9D5tiEDI0LLR21JrhEtSIuwng94Q5T1B2\_yjlegC\_mjCbObQmdAe1yiOT3BlbkFJg-9ys3PE4dS-Vopb6LTSEuEeLfsE0hiwHz5h2jCqIG1So9evMGS2d3xR4WMlCwI9VSHVeV5A4A\r\n' +

Dec 12 04:47:51 PM 'Content-Length: 469\r\n' +

Dec 12 04:47:51 PM 'Host: api.openai.com\r\n' +

Dec 12 04:47:51 PM 'Connection: keep-alive\r\n' +

Dec 12 04:47:51 PM '\r\n',

Dec 12 04:47:51 PM \_keepAliveTimeout: 0,

Dec 12 04:47:51 PM \_onPendingData: [Function: nop],

Dec 12 04:47:51 PM agent: [Agent],

Dec 12 04:47:51 PM socketPath: undefined,

Dec 12 04:47:51 PM method: 'POST',

Dec 12 04:47:51 PM maxHeaderSize: undefined,

Dec 12 04:47:51 PM insecureHTTPParser: undefined,

Dec 12 04:47:51 PM joinDuplicateHeaders: undefined,

Dec 12 04:47:51 PM path: '/v1/chat/completions',

Dec 12 04:47:51 PM \_ended: true,

Dec 12 04:47:51 PM res: [IncomingMessage],

Dec 12 04:47:51 PM aborted: false,

Dec 12 04:47:51 PM timeoutCb: null,

Dec 12 04:47:51 PM upgradeOrConnect: false,

Dec 12 04:47:51 PM parser: null,

Dec 12 04:47:51 PM maxHeadersCount: null,

Dec 12 04:47:51 PM reusedSocket: false,

Dec 12 04:47:51 PM host: 'api.openai.com',

Dec 12 04:47:51 PM protocol: 'https:',

Dec 12 04:47:51 PM \_redirectable: [Writable],

Dec 12 04:47:51 PM [Symbol(shapeMode)]: false,

Dec 12 04:47:51 PM [Symbol(kCapture)]: false,

Dec 12 04:47:51 PM [Symbol(kBytesWritten)]: 0,

Dec 12 04:47:51 PM [Symbol(kNeedDrain)]: false,

Dec 12 04:47:51 PM [Symbol(corked)]: 0,

Dec 12 04:47:51 PM [Symbol(kChunkedBuffer)]: [],

Dec 12 04:47:51 PM [Symbol(kChunkedLength)]: 0,

Dec 12 04:47:51 PM [Symbol(kSocket)]: [TLSSocket],

Dec 12 04:47:51 PM [Symbol(kOutHeaders)]: [Object: null prototype],

Dec 12 04:47:51 PM [Symbol(errored)]: null,

Dec 12 04:47:51 PM [Symbol(kHighWaterMark)]: 65536,

Dec 12 04:47:51 PM [Symbol(kRejectNonStandardBodyWrites)]: false,

Dec 12 04:47:51 PM [Symbol(kUniqueHeaders)]: null

Dec 12 04:47:51 PM },

Dec 12 04:47:51 PM data: { error: [Object] }

Dec 12 04:47:51 PM },

Dec 12 04:47:51 PM isAxiosError: true,

Dec 12 04:47:51 PM toJSON: [Function: toJSON]

Dec 12 04:47:51 PM}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>ShepherdXR</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<!-- Section 1 -->

<div class="whitespace"></div>

<header>

<img src="assets/logo.png" alt="ShepherdXR Logo" class="logo">

</header>

<div class="whitespace"></div>

<section class="mission">

<img src="assets/classroom1.jpg" alt="Classroom" class="classroom-image">

<div class="whitespace"></div>

<h1 class="purple-header">ShepherdXR</h1>

<div class="whitespace"></div>

<p class="mission-text">

Mission: Zoooooooooooom! At ShepherdXR, we’re on a mission to discover and empower the unexpected Einsteins of tomorrow.

As an ed-tech non-profit, we’re tackling the crisis of lost potential: when middle schoolers hit that giant wall

between arithmetic and algebra — often derailing their math journey for life.

Our flagship product, Shepherd Math, transforms the subject from a dreaded chore into a thrilling adventure

in Extended Reality (XR/VR).

</p>

<div class="whitespace"></div>

<div class="slideshow">

<img src="assets/classroom1.jpg" alt="Classroom 1" class="slide" id="slide">

</div>

<div class="whitespace"></div>

</section>

<!-- Shepherd Guide Link -->

<section class="guide-link">

<div class="whitespace"></div>

<h2>For Instructors</h2>

<p>Visit the <a href="shepherd-guide.html">Shepherd Guide</a> to input math equations and get rendering and helpful prompts!</p>

<div class="whitespace"></div>

</section>

<!-- Contact Us Section -->

<footer class="contact">

<h2>Contact Us</h2>

<p>Email: support@shepherdxr.com</p>

<p>Phone: +1 (800) 555-1234</p>

</footer>

<script src="script.js"></script>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Shepherd Guide</title>

<script src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/3.2.2/es5/tex-mml-chtml.js" async></script>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<header>

<img src="assets/logo.png" alt="ShepherdXR Logo" class="logo">

</header>

<div class="whitespace"></div>

<section class="guide-section">

<h1 class="purple-header">Shepherd Guide</h1>

<p>Input your LaTeX math equation below and get a rendering along with helpful prompts for instructors.</p>

<!-- Input Field and Buttons -->

<div class="input-section">

<textarea id="latexInput" placeholder="Enter your equation in LaTeX format here"></textarea>

<button onclick="renderEquation()">Render</button>

<button onclick="generatePrompts()">Get Helpful Prompts</button>

</div>

<!-- Rendered Equation -->

<div class="render-section">

<h2>Rendered Equation</h2>

<div id="equation"></div>

</div>

<!-- Helpful Prompts -->

<div class="prompts-section">

<h2>Helpful Prompts</h2>

<div id="prompts"></div>

</div>

</section>

<!-- Back to Home -->

<footer>

<p><a href="index.html">Back to Home</a></p>

</footer>

<script>

function renderEquation() {

const latexInput = document.getElementById("latexInput").value;

const equationDiv = document.getElementById("equation");

equationDiv.innerHTML = `$$${latexInput}$$`;

MathJax.typeset();

}

async function generatePrompts() {

const equation = document.getElementById("latexInput").value;

const promptsDiv = document.getElementById("prompts");

promptsDiv.innerHTML = "Generating prompts...";

try {

const response = await fetch('https://sxr-backend.onrender.com/generate-prompts', {

method: 'POST',

headers: { 'Content-Type': 'application/json' },

body: JSON.stringify({ equation }),

});

const data = await response.json();

promptsDiv.innerHTML = data.prompts

.map((prompt, index) => `<p><strong>Prompt ${index + 1}:</strong> ${prompt}</p>`)

.join('');

} catch (error) {

promptsDiv.innerHTML = "Error generating prompts. Please try again.";

console.error(error);

}

}

</script>

</body>

</html>

/\* Root variables \*/

**:root** {

--purple: #6a1b9a;

--white: #ffffff;

--whiteseparation: 2rem;

--font-sans: 'Arial', sans-serif;

--font-serif: 'Georgia', serif;

}

/\* Reset styles \*/

**\*** {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: var(--font-sans);

color: #333;

background-color: var(--white);

line-height: 1.6;

margin: 0;

}

/\* Logo styling \*/

header **.logo** {

display: block;

max-width: 200px;

margin: 0 auto;

}

/\* White space \*/

**.whitespace** {

height: var(--whiteseparation);

}

/\* Mission section \*/

**.mission** {

text-align: center;

padding: 0 2rem;

}

**.mission** **.classroom-image** {

max-width: 100%;

height: auto;

border-radius: 8px;

}

**.purple-header** {

font-family: var(--font-serif);

font-size: 2.5rem;

color: var(--purple);

text-transform: uppercase;

}

**.mission-text** {

font-family: var(--font-sans);

font-size: 1.2rem;

color: #555;

max-width: 800px;

margin: 0 auto;

padding: 0 1rem;

}

/\* Slideshow \*/

**.slideshow** {

position: relative;

max-width: 100%;

margin: 0 auto;

height: auto;

overflow: hidden;

}

**.slide** {

max-width: 100%;

height: auto;

border-radius: 8px;

transition: opacity 1s ease-in-out;

}

/\* Contact Us Section \*/

**.contact** {

background-color: var(--purple);

color: var(--white);

text-align: center;

padding: 2rem 1rem;

}

**.contact** h2 {

font-size: 1.8rem;

margin-bottom: 1rem;

}

**.contact** p {

font-size: 1.1rem;

margin: 0.5rem 0;

}

/\* Shepherd Guide output box \*/

**.output-box** {

margin-top: 1rem;

padding: 1rem;

border: 1px solid #ddd;

border-radius: 8px;

background-color: #f9f9f9;

min-height: 50px;

text-align: center;

}

document.addEventListener("DOMContentLoaded", () => {

// Slideshow logic

const slides = ["assets/classroom1.jpg", "assets/classroom2.jpg", "assets/classroom3.jpg"];

const slideElement = document.getElementById("slide");

let currentSlide = 0;

function changeSlide() {

currentSlide = (currentSlide + 1) % slides.length;

slideElement.src = slides[currentSlide];

}

// Change the slide every 3 seconds

setInterval(changeSlide, 3000);

});

const express = require('express');

const bodyParser = require('body-parser');

const cors = require('cors');

const { Configuration, OpenAIApi } = require('openai');

require('dotenv').config();

const app = express();

const port = 3000;

// Middleware

app.use(cors());

app.use(bodyParser.json());

// OpenAI setup

const configuration = new Configuration({

apiKey: process.env.OPENAI\_API\_KEY,

});

const openai = new OpenAIApi(configuration);

// Endpoint to generate prompts

app.post('/generate-prompts', async (req, res) => {

const { equation } = req.body;

if (!equation) return res.status(400).send({ error: "Equation is required" });

const systemMessage = `

You are a guide for Shepherd Math, a VR game that teaches middle school math to kids.

Your audience is adult instructors helping kids navigate math problems in VR headsets.

Ensure safeguarding priorities and focus on clarity, engagement, and the mechanics of the VR game.

`;

try {

const response = await openai.createChatCompletion({

model: 'gpt-4',

messages: [

{ role: 'system', content: systemMessage },

{ role: 'user', content: `Provide 3 helpful teaching prompts for the equation: ${equation}` },

],

max\_tokens: 200,

});

const prompts = response.data.choices[0].message.content.split('\n').filter(line => line.trim());

res.send({ prompts });

} catch (error) {

console.error(error);

res.status(500).send({ error: "Failed to generate prompts" });

}

});

// Start server

app.listen(port, () => {

console.log(`Server running on http://localhost:${port}`);

});

{

"name": "shepherd-guide-backend",

"version": "1.0.0",

"main": "server.js",

"scripts": {

"start": "node server.js"

},

"dependencies": {

"express": "^4.17.1",

"body-parser": "^1.19.0",

"cors": "^2.8.5",

"dotenv": "^16.0.3",

"openai": "^3.2.1"

}

}

OPENAI\_API\_KEY=sk-proj-iMDtoJ5b1Oy4o3N4Cm9D5tiEDI0LLR21JrhEtSIuwng94Q5T1B2\_yjlegC\_mjCbObQmdAe1yiOT3BlbkFJg-9ys3PE4dS-Vopb6LTSEuEeLfsE0hiwHz5h2jCqIG1So9evMGS2d3xR4WMlCwI9VSHVeV5A4A

Ok a few notes to help with the troubleshooting:

* For some reason, it seems that when I update the code on my desktop, then commit and push to github… it’s not appearing on the website link via GitHub Pages (e.g. see the ‘zoooooom’ I added to the text below in my updated scripts
* Does the backend hyperlink look right? I copied it from Render. However, I’m not sure if I was meant to add the ‘/generate-prompts’ bit?
* When I open the backend link (https://sxr-backend.onrender.com) in chrome, all I get is a blank white page with the message: Cannot GET /
* Also here is the log from Render that I got when I first set it up
* All logs
* Search
* Dec 12, 1:44 PM - 1:47 PM
* EST
* Dec 12 01:45:01 PM**==>** **Cloning from** [**https://github.com/femishep222/sxr-backend**](https://github.com/femishep222/sxr-backend)
* Dec 12 01:45:01 PM**==>** **Checking out commit 5f1145a675ea70f861cef926a1f1f8cce5dae6c8 in branch main**
* Dec 12 01:45:03 PM**==>** **Using Node.js version 22.11.0 (default)**
* Dec 12 01:45:03 PM**==>** **Docs on specifying a Node.js version:** [**https://render.com/docs/node-version**](https://render.com/docs/node-version)
* Dec 12 01:45:05 PM**==>** **Using Bun version 1.1.0 (default)**
* Dec 12 01:45:05 PM**==>** **Docs on specifying a bun version:** [**https://render.com/docs/bun-version**](https://render.com/docs/bun-version)
* Dec 12 01:45:05 PM**==>** **Running build command 'yarn install'...**
* Dec 12 01:45:05 PMyarn install v1.22.22
* Dec 12 01:45:05 PMwarning package.json: No license field
* Dec 12 01:45:05 PMinfo No lockfile found.
* Dec 12 01:45:05 PMwarning shepherd-guide-backend@1.0.0: No license field
* Dec 12 01:45:05 PM[1/4] Resolving packages...
* Dec 12 01:45:06 PM[2/4] Fetching packages...
* Dec 12 01:45:06 PM[3/4] Linking dependencies...
* Dec 12 01:45:06 PM[4/4] Building fresh packages...
* Dec 12 01:45:06 PMsuccess Saved lockfile.
* Dec 12 01:45:06 PMDone in 1.68s.
* Dec 12 01:45:08 PM**==>** **Uploading build...**
* Dec 12 01:45:16 PM**==>** **Build uploaded in 8s**
* Dec 12 01:45:16 PM**==>** **Build successful 🎉**
* Dec 12 01:45:18 PM**==> Deploying...**
* Dec 12 01:45:37 PM**==>** **Running 'node server.js'**
* Dec 12 01:45:38 PMServer running on [http://localhost:3000](http://localhost:3000/)
* Dec 12 01:46:15 PM**==> New primary port detected: 3000. Restarting deploy to update network configuration...**
* Dec 12 01:46:16 PM**==> Docs on specifying a port:** [**https://render.com/docs/web-services#port-binding**](https://render.com/docs/web-services#port-binding)
* Dec 12 01:46:34 PM**==>** **Running 'node server.js'**
* Dec 12 01:46:35 PMServer running on [http://localhost:3000](http://localhost:3000/)
* Dec 12 01:46:36 PM**==> Your service is live 🎉**
* Dec 12 01:47:18 PM**==> Detected a new open port HTTP:3000**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>ShepherdXR</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<!-- Section 1 -->

<div class="whitespace"></div>

<header>

<img src="assets/logo.png" alt="ShepherdXR Logo" class="logo">

</header>

<div class="whitespace"></div>

<section class="mission">

<img src="assets/classroom1.jpg" alt="Classroom" class="classroom-image">

<div class="whitespace"></div>

<h1 class="purple-header">ShepherdXR</h1>

<div class="whitespace"></div>

<p class="mission-text">

Mission: Zoooooooooooom! At ShepherdXR, we’re on a mission to discover and empower the unexpected Einsteins of tomorrow.

As an ed-tech non-profit, we’re tackling the crisis of lost potential: when middle schoolers hit that giant wall

between arithmetic and algebra — often derailing their math journey for life.

Our flagship product, Shepherd Math, transforms the subject from a dreaded chore into a thrilling adventure

in Extended Reality (XR/VR).

</p>

<div class="whitespace"></div>

<div class="slideshow">

<img src="assets/classroom1.jpg" alt="Classroom 1" class="slide" id="slide">

</div>

<div class="whitespace"></div>

</section>

<!-- Shepherd Guide Link -->

<section class="guide-link">

<div class="whitespace"></div>

<h2>For Instructors</h2>

<p>Visit the <a href="shepherd-guide.html">Shepherd Guide</a> to input math equations and get rendering and helpful prompts!</p>

<div class="whitespace"></div>

</section>

<!-- Contact Us Section -->

<footer class="contact">

<h2>Contact Us</h2>

<p>Email: support@shepherdxr.com</p>

<p>Phone: +1 (800) 555-1234</p>

</footer>

<script src="script.js"></script>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Shepherd Guide</title>

<script src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/3.2.2/es5/tex-mml-chtml.js" async></script>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<header>

<img src="assets/logo.png" alt="ShepherdXR Logo" class="logo">

</header>

<div class="whitespace"></div>

<section class="guide-section">

<h1 class="purple-header">Shepherd Guide</h1>

<p>Input your LaTeX math equation below and get a rendering along with helpful prompts for instructors.</p>

<!-- Input Field and Buttons -->

<div class="input-section">

<textarea id="latexInput" placeholder="Enter your equation in LaTeX format here"></textarea>

<button onclick="renderEquation()">Render</button>

<button onclick="generatePrompts()">Get Helpful Prompts</button>

</div>

<!-- Rendered Equation -->

<div class="render-section">

<h2>Rendered Equation</h2>

<div id="equation"></div>

</div>

<!-- Helpful Prompts -->

<div class="prompts-section">

<h2>Helpful Prompts</h2>

<div id="prompts"></div>

</div>

</section>

<!-- Back to Home -->

<footer>

<p><a href="index.html">Back to Home</a></p>

</footer>

<script>

function renderEquation() {

const latexInput = document.getElementById("latexInput").value;

const equationDiv = document.getElementById("equation");

equationDiv.innerHTML = `$$${latexInput}$$`;

MathJax.typeset();

}

async function generatePrompts() {

const equation = document.getElementById("latexInput").value;

const promptsDiv = document.getElementById("prompts");

promptsDiv.innerHTML = "Generating prompts...";

try {

const response = await fetch('https://sxr-backend.onrender.com', {

method: 'POST',

headers: { 'Content-Type': 'application/json' },

body: JSON.stringify({ equation }),

});

const data = await response.json();

promptsDiv.innerHTML = data.prompts

.map((prompt, index) => `<p><strong>Prompt ${index + 1}:</strong> ${prompt}</p>`)

.join('');

} catch (error) {

promptsDiv.innerHTML = "Error generating prompts. Please try again.";

console.error(error);

}

}

</script>

</body>

</html>

document.addEventListener("DOMContentLoaded", () => {

// Slideshow logic

const slides = ["assets/classroom1.jpg", "assets/classroom2.jpg", "assets/classroom3.jpg"];

const slideElement = document.getElementById("slide");

let currentSlide = 0;

function changeSlide() {

currentSlide = (currentSlide + 1) % slides.length;

slideElement.src = slides[currentSlide];

}

// Change the slide every 3 seconds

setInterval(changeSlide, 3000);

});

/\* Root variables \*/

**:root** {

--purple: #6a1b9a;

--white: #ffffff;

--whiteseparation: 2rem;

--font-sans: 'Arial', sans-serif;

--font-serif: 'Georgia', serif;

}

/\* Reset styles \*/

**\*** {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: var(--font-sans);

color: #333;

background-color: var(--white);

line-height: 1.6;

margin: 0;

}

/\* Logo styling \*/

header **.logo** {

display: block;

max-width: 200px;

margin: 0 auto;

}

/\* White space \*/

**.whitespace** {

height: var(--whiteseparation);

}

/\* Mission section \*/

**.mission** {

text-align: center;

padding: 0 2rem;

}

**.mission** **.classroom-image** {

max-width: 100%;

height: auto;

border-radius: 8px;

}

**.purple-header** {

font-family: var(--font-serif);

font-size: 2.5rem;

color: var(--purple);

text-transform: uppercase;

}

**.mission-text** {

font-family: var(--font-sans);

font-size: 1.2rem;

color: #555;

max-width: 800px;

margin: 0 auto;

padding: 0 1rem;

}

/\* Slideshow \*/

**.slideshow** {

position: relative;

max-width: 100%;

margin: 0 auto;

height: auto;

overflow: hidden;

}

**.slide** {

max-width: 100%;

height: auto;

border-radius: 8px;

transition: opacity 1s ease-in-out;

}

/\* Contact Us Section \*/

**.contact** {

background-color: var(--purple);

color: var(--white);

text-align: center;

padding: 2rem 1rem;

}

**.contact** h2 {

font-size: 1.8rem;

margin-bottom: 1rem;

}

**.contact** p {

font-size: 1.1rem;

margin: 0.5rem 0;

}

/\* Shepherd Guide output box \*/

**.output-box** {

margin-top: 1rem;

padding: 1rem;

border: 1px solid #ddd;

border-radius: 8px;

background-color: #f9f9f9;

min-height: 50px;

text-align: center;

}

Lovely. Let’s do this. Is it possible to execute this using GitHub pages - both as the host for the website and also for the server stuff you mentioned above? If so, please provide step by step instructions for getting it all set up!

Note: I already have a ChatGPT Plus account. Also, I already have a Github account too… and already have the GitHub desktop app installed. But I haven’t yet added my website as a repository.

it worked beautifully. Praise God. Ok, now, I’d like to explore the possibility of linking that shepherd-guide’ web page to a gpt (ideally a custom GPT that links to my chatgpt subscription and uses a model that doesn’t have limited requests like the 4o model?) that will do the following:

* The custom gpt will have context information saved in its configuration that basically explains how my maths VR game works and also my overall goals and points around child safeguarding and the audience being adult instructors who are helping kids work through the maths problems on their VR headsets (given they can’t necessarily see what’s on the screen directly!)
* A teacher can then input an example equation into the input field of the shepherd-guide web page, then click the ‘Render’ button
* That web page will then provide the 2D rendering (as it already successfully does with the code you provided!) and…
* Further below on the web page, it will also spit out 3 ‘helpful prompts’ which are generated by the custom gpt based on its knowledge of the games workings, and the wider priorities, applied to the specific equation in question

Any ideas on a simple way to achieve this (if it’s indeed possible through relatively simple means)?!

Ok, let’s start from scratch with a new structure:

* Firstly: all white background for the landing page!
* Section 1 (going sequentially from top to bottom):
  + white space (use the variable ‘whiteseperation’ to define this vertical distance moving forward)
  + Shepherdxr logo (logo file)
  + White space
  + classroom1 image
  + white space
  + ShepherXR (as a header) in purple
  + White space
  + Mission: **At ShepherdXR, we’re on a mission to discover and empower the unexpected Einsteins of tomorrow.**

As an ed-tech non-profit, we’re tackling the crisis of lost potential: when middle schoolers hit that giant wall between arithmetic and algebra — often derailing their math journey for life.

Our flagship product, *Shepherd Math*, transforms the subject from a dreaded chore into a thrilling adventure in Extended Reality (XR/VR).

* + White space
  + Slideshow of clasroom1, classroom2, classroom3 images (jpeg files)
  + White space
  + Contact us section at bottom with white text and purple background

Use standard / sensible spacings and formatting. Make it stylish and attractive (e.g. fonts) relative to the audience.

Please share the index, styles and script files for me to copy-paste!