

Node server code

```
const express = require('express');
const axios = require('axios');
const cors = require('cors');
const app = express();
const port = process.env.PORT || 3000;

// Enable CORS to allow cross-origin requests
app.use(cors());
app.use(express.json());

// Your API Key for Watson
const API_KEY = ""; // Replace with your actual API key

// Step 1: Fetch token for IBM Watson authentication
async function getAuthToken() {
  try {
    const response = await
    axios.post('https://iam.cloud.ibm.com/identity/token', null, {
      params: {
        apikey: API_KEY,
        grant_type: 'urn:ibm:params:oauth:grant-type:apikey',
      },
    });
    return response.data.access_token;
  } catch (error) {
    console.error('Error getting auth token:', error);
    throw new Error('Failed to retrieve token');
  }
}

// Step 2: Cleanse healthcare data using Watson's NLP model
async function cleanseHealthcareData(healthcareRecord) {
  try {
    const token = await getAuthToken();
    const payload = {
      text: healthcareRecord.text, // The healthcare record text to
      cleanse
    };

    const response = await axios.post(
      'https://<region>.nlp.cloud.ibm.com/your-nlp-
      endpoint/predictions', // Replace with actual Watson NLP endpoint
      payload,
      {
        headers: {
          'Authorization': `Bearer ${token}`,
          'Content-Type': 'application/json',
        },
      },
    );

    return response.data.cleanedData; // Cleaned data from the NLP
  }
}
```

```

model
  } catch (error) {
    console.error('Error during data cleansing:', error);
    throw new Error('Failed to cleanse data');
  }
}

// Step 3: Endpoint to send healthcare record data for cleansing
app.post('/cleanse', async (req, res) => {
  try {
    const healthcareRecord = req.body.healthcareRecord; // Expecting
    a healthcare record text in the request body
    const cleanedData = await
    cleanseHealthcareData(healthcareRecord);
    res.json({
      cleanedData,
    });
  } catch (error) {
    res.status(500).json({
      error: 'Error processing data cleansing',
    });
  }
});

// Start the server
app.listen(port, () => {
  console.log(Server running on port ${port});
});

```