Developing a server to interface with the model

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()); // To parse JSON request bodies

// Your API Key for Watson NLP
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: Analyze text using Watson NLP model
async function analyzeText(textData) {
    try {
        const token = await getAuthToken();

        // Prepare payload for NLP model
        const payload = {
            input_data: [{ text: textData }]
        };

        const response = await axios.post(
            'https://eu-de.ml.cloud.ibm.com/ml/v4/deployments/nlp-text-analysis/predictions?version=2021-05-01',
            payload,
            {
                headers: {
                    'Authorization': `Bearer ${token}`,
                    'Content-Type': 'application/json',
                },
            }
        );
        return response.data.predictions;
```

```
    } catch (error) {
        console.error('Error during text analysis:', error);
        throw new Error('Failed to analyze text');
    }
}

// Step 3: Endpoint for frontend to get NLP analysis results
app.post('/analyze', async (req, res) => {
    try {
        const textData = req.body.text;
        const analysisResults = await analyzeText(textData);
        res.json({ analysis: analysisResults });
    } catch (error) {
        res.status(500).json({ error: 'Error processing text analysis' });
    }
});

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