```
const express = require('express');
const axios = require('axios');
const _ = require('lodash');

const app = express();
const port = process.env.PORT || 3000;

// Memoize function to cache results
const memoizedGetBlogStats = _.memoize(async () => {
  try {
    // Make the API request using Axios
    const response = await axios.get('https://intent-kit-16.hasura.app/api/rest/blogs', {
      headers: {
        'x-hasura-admin-secret': '32qR4KmXOIpsGPQKMqEJHGJS27G5s7HdSKO3gdtQd2kv5e852SiYwW
NfxkZOBuQ6',
      },
    });

    const blogData = response.data;

    // Calculate the total number of blogs fetched
    const totalBlogs = blogData.length;

    // Find the blog with the longest title
    const longestBlog = _.maxBy(blogData, 'title.length');

    // Determine the number of blogs with titles containing the word "privacy"
    const privacyBlogs = _.filter(blogData, (blog) =>
      blog.title.toLowerCase().includes('privacy')
    );
    const numPrivacyBlogs = privacyBlogs.length;

    // Create an array of unique blog titles
    const uniqueTitles = _.uniqBy(blogData, 'title');

    return {
      totalBlogs,
      longestBlog: longestBlog ? longestBlog.title : null,
      numPrivacyBlogs,
      uniqueTitles: uniqueTitles.map((blog) => blog.title),
    };
  } catch (error) {
    throw new Error('Error fetching and analyzing blog data');
  }
}, { maxAge: 60000 }); // Cache for 1 minute (adjust as needed)

// Middleware for /api/blog-stats route
app.get('/api/blog-stats', async (req, res) => {
  try {
    const stats = await memoizedGetBlogStats();
    res.json(stats);
  } catch (error) {
    console.error(error);
    res.status(500).json({ error: 'Internal Server Error' });
  }
});

// Blog search endpoint
app.get('/api/blog-search', (req, res) => {
  const query = req.query.query || '';

  if (!query) {
    return res.status(400).json({ error: 'Query parameter is required' });
  }

  try {
    // Implement the search functionality (case-insensitive)
    memoizedGetBlogStats.clear(); // Clear the cache when a search is performed
    const searchResults = memoizedGetBlogStats().uniqueTitles.filter((title) =>
      title.toLowerCase().includes(query.toLowerCase())
    );

    res.json(searchResults);
  } catch (error) {
    console.error(error);
    res.status(500).json({ error: 'Internal Server Error' });
  }
});

// Error handling middleware
app.use((err, req, res, next) => {
  console.error(err.stack);
  res.status(500).json({ error: 'Internal Server Error' });
});

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