**Written BluePrint/Plan Basics only**

To create a web server that works with npm (Node Package Manager) for a calorie tracker, you can use a server-side language, such as Node.js, and a web framework, such as Express.

Here are the steps to create a web server with npm and Node.js for a calorie tracker:

1. Install Node.js on your computer by following the instructions on the Node.js website. This will allow you to run JavaScript code on the server-side.
2. Install Express, a web framework for Node.js, by running the following command in a terminal or command prompt: npm install express. This will enable you to create a web server and handle HTTP requests and responses with Node.js.
3. Create a file named server.js and add the following code to set up a web server with Express:

const express = require('express')

const app = express()

app.listen(3000, () => {

console.log('Server listening on port 3000')

})

1. Run the web server by running the following command in a terminal or command prompt: node server.js. This will start the web server and listen for incoming requests on port 3000.
2. Test the web server by opening a web browser and navigating to http://localhost:3000. You should see a message that says "Server listening on port 3000".
3. Add routes to the web server to handle different types of requests, such as GET requests to retrieve data and POST requests to add data. For example, you can add a route to handle a GET request to retrieve a list of meals that the user has entered:

app.get('/meals', (req, res) => {

// retrieve list of meals from database

const meals = [

{ id: 1, name: 'Banana', calories: 150 },

{ id: 2, name: 'Yogurt', calories: 300 },

{ id: 3, name: 'Chicken Breast', calories: 300 }

]

// send the list of meals as a JSON response

res.json(meals)

})

1. Create a user interface for the calorie tracker using HTML and JavaScript, and use the web server to retrieve and save data. For example, you can use AJAX to make GET and POST requests to the web server to retrieve and add data, respectively.

By using npm and Node.js