DOCUMENTACIÓN

EME

2024

DOCUMENTACION WEB

A CONTINUACIÓN, SE DEJA POR ESCRITO EL FUNCIONAMIENTO DE LA PÁGINA WEB DE LA ESTACIÓN METEOROLÓGICA ESTÁTICA DE LA INSTITUCIÓN ETec

MARTÍN GIMENEZ - ETec

El proyecto consiste en una aplicación web de una estación meteorológica que consta de tres archivos principales: `index.html`, `style.css` y `app.js`. A continuación se detalla el funcionamiento y los principios básicos de cada uno.

Index.html

El archivo `index.html` es la página principal de la aplicación. A continuación se describe su contenido y funcionalidad:

- **Doctype y etiquetas HTML**: El documento comienza con la declaración *'<!DOCTYPE html>'* que es una instrucción para el navegador web sobre la versión de HTML que el documento está usando. Luego, el documento HTML se envuelve con la etiqueta *<html>* y se especifica el idioma del documento como español con el atributo *lang="ES"*.
- **Head**: Dentro de la etiqueta *<head>*, se definen varios metadatos para el documento HTML. Esto incluye la codificación de caracteres, la configuración de la vista del puerto para dispositivos móviles, el título de la página y las referencias a los archivos de estilo CSS y a los iconos de la página.
- **Body**: El cuerpo del documento HTML contiene el contenido principal de la página web. Se divide en varias secciones:
    - **Sección de Datos Básicos**: Esta sección muestra los datos meteorológicos básicos como la temperatura, la humedad, la presión atmosférica y la última actualización. Los datos se actualizan dinámicamente utilizando JavaScript.
    - **Tabla de Datos**: Esta tabla muestra una lista de lecturas meteorológicas recientes. Los datos se insertan dinámicamente en la tabla utilizando JavaScript.
    - **Gráfico**: Esta sección contiene un elemento de canvas para un gráfico para visualizar los datos meteorológicos y botones para ajustar la escala del gráfico.
    - **Pie de página**: El pie de página contiene notas y descriptores sobre el proyecto, así como enlaces a la página de GitHub del proyecto y a una función para exportar los datos a Excel.
- **JavaScript**: Al final del cuerpo del documento, hay un bloque de código JavaScript que utiliza la API Fetch para obtener los últimos datos meteorológicos de un servidor local en http://localhost:3000/ultimos-datos. Los datos obtenidos se utilizan para actualizar el contenido de la página y para insertar filas en la tabla de datos.

STYLE.CSS

Brevemente explicando el funcionamiento de este archivo es el encargado de brindarle los estilos de la página web a index.html

APP.JS

Introducción

El archivo `app.js` es un script de Node.js que utiliza el marco de trabajo Express.js para crear un servidor web. Este servidor se conecta a una base de datos MySQL y proporciona una interfaz de API para interactuar con los datos. También utiliza el middleware CORS para permitir solicitudes de origen cruzado.

Importación de módulos

El script comienza importando los módulos necesarios. `express` es un marco de trabajo para crear servidores web en Node.js, `mysql` es un módulo para interactuar con bases de datos MySQL y `cors` es un middleware para habilitar CORS (Cross-Origin Resource Sharing).

Creación de la aplicación Express

Aquí se crea una nueva aplicación Express.

Configuración de la base de datos

Se crea una conexión a la base de datos MySQL utilizando el método `createConnection` del módulo `mysql`. Los detalles de la conexión se proporcionan como un objeto.

Conexión a la base de datos

```
db.connect((err) => {
  if (err) {
    throw err;
  }
  console.log(
'Conexión a la base de datos estableci
da'
);
});
```

Se establece la conexión a la base de datos. Si hay un error durante la conexión, se lanzará una excepción.

Configuración de CORS

```
app.use(cors());
```

Se configura el middleware CORS en la aplicación Express para permitir solicitudes de origen cruzado.

Rutas de la API

```
app.get('/ultimos-datos', (req, res)
=> {
  const sql =
'SELECT * FROM estacion_test ORDER BY
`Dato N°` DESC LIMIT 6'
;
  db.query(sql, (err, result) => {
    if (err) {
      throw err;
    }
    res.json(result);
  });
});
```

Se define una ruta de API para obtener los últimos datos de la base de datos. La consulta SQL selecciona los últimos 6 registros de la tabla `estacion_test`.

Servir archivos estáticos

```
app.get('/', (req, res) => {
  res.sendFile(__dirname +
'/index.html');
});
```

Se define una ruta para servir el archivo `index.html` cuando se accede a la raíz del servidor.

Inicio del servidor

```
const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(
`Servidor iniciado en http://localhos
t:
${PORT}`);
});
```

Finalmente, el servidor se inicia en el puerto especificado por la variable de entorno `PORT` o, si no está definida, en el puerto 3000.

CONCLUSIÓN

La documentación proporcionada detalla de manera clara y concisa el funcionamiento y los principios básicos del proyecto de la estación meteorológica. Desde la estructura de los archivos principales hasta los detalles de implementación en app.js, cada componente está explicado con suficiente detalle para comprender su propósito y funcionamiento.

La inclusión de imágenes ilustrativas mejora aún más la comprensión visual de la estructura del proyecto y la relación entre sus componentes. Esto facilita tanto el entendimiento inicial del proyecto como su mantenimiento continuo y la realización de contribuciones por parte de otros desarrolladores.

En resumen, la documentación proporcionada sirve como una guía completa y accesible para cualquier persona interesada en comprender, utilizar o contribuir al proyecto de la estación meteorológica.