# An-Najah National University

Faculty of Engineering and Information Technology

Department of Computer Engineering

Advanced Software Engineering

Course Project – RESTful API – Fall 2023

Dr. Amjad AbuHassan

EcoTrack: Environmental Monitoring and Reporting Platform

Done by :

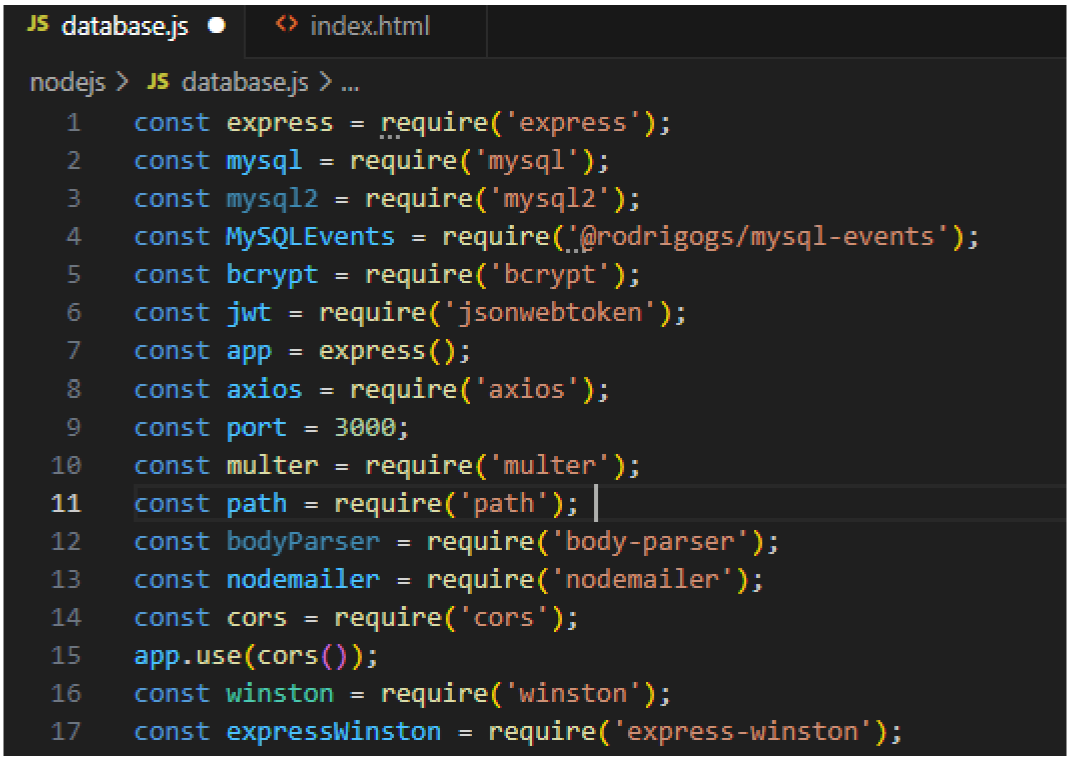
Fatima AbuReesh

Housnia Mashaqi

Aya Jabali

In this project we developed a robust backend API for an Environmental Monitoring and Reporting Platform. We use Node.js , Vs code , data base SQL , Postman , Xampp , wiki and GitHub to do it .

These are the libraries we installed and used and running in port 3000 :

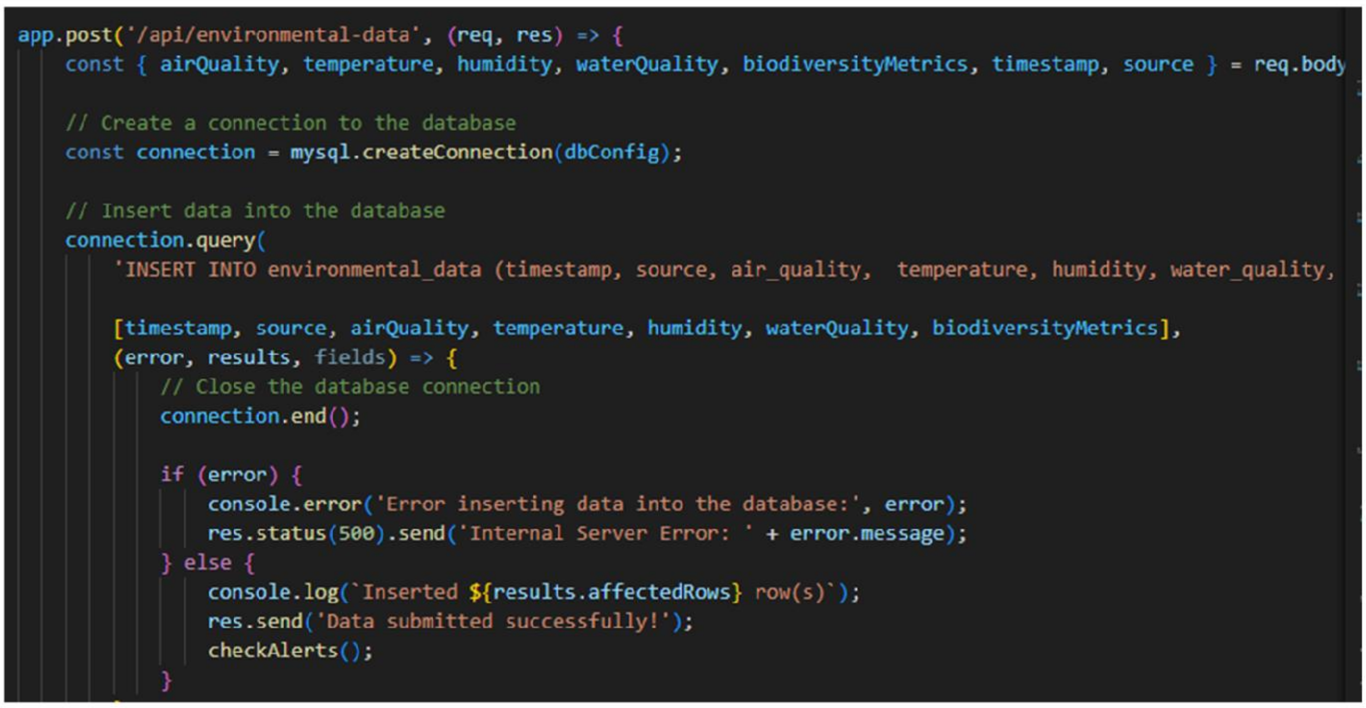


```
const express = require('express');
const mysql = require('mysql');
const mysql2 = require('mysql2');
const MySQLEvents = require('@rodrigogs/mysql-events');
const bcrypt = require('bcrypt');
const jwt = require('jsonwebtoken');
const app = express();
const axios = require('axios');
const port = 3000;
const multer = require('multer');
const path = require('path');
const bodyParser = require('body-parser');
const nodemailer = require('nodemailer');
const cors = require('cors');
app.use(cors());
const winston = require('winston');
const expressWinston = require('express-winston');
```

## Main Features:

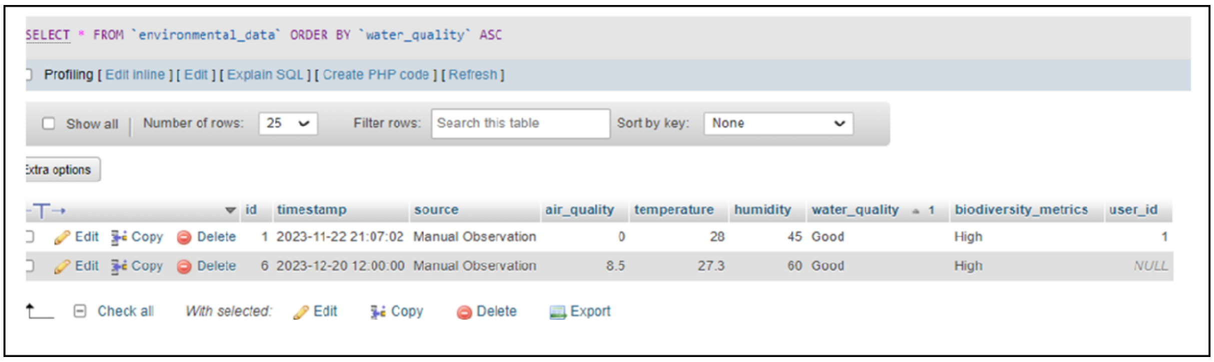
### 1. Data Collection

Users can submit environmental data from various sources, such as, manual observations, or data uploads. Data can include temperature, humidity, water quality, and more.And this code :



```
app.post('/api/environmental-data', (req, res) => {
    const { airQuality, temperature, humidity, waterQuality, biodiversityMetrics, timestamp, source } = req.body

    // Create a connection to the database
    const connection = mysql.createConnection(dbConfig);

    // Insert data into the database
    connection.query(
        'INSERT INTO environmental_data (timestamp, source, air_quality,  temperature, humidity, water_quality,

        [timestamp, source, airQuality, temperature, humidity, waterQuality, biodiversityMetrics],
        (error, results, fields) => {
            // Close the database connection
            connection.end();

            if (error) {
                console.error('Error inserting data into the database:', error);
                res.status(500).send('Internal Server Error: ' + error.message);
            } else {
                console.log(`Inserted ${results.affectedRows} row(s)`);
                res.send('Data submitted successfully!');
                checkAlerts();
            }
```

And the result in data base after running code :



```
SELECT * FROM `environmental_data` ORDER BY `water_quality` ASC
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

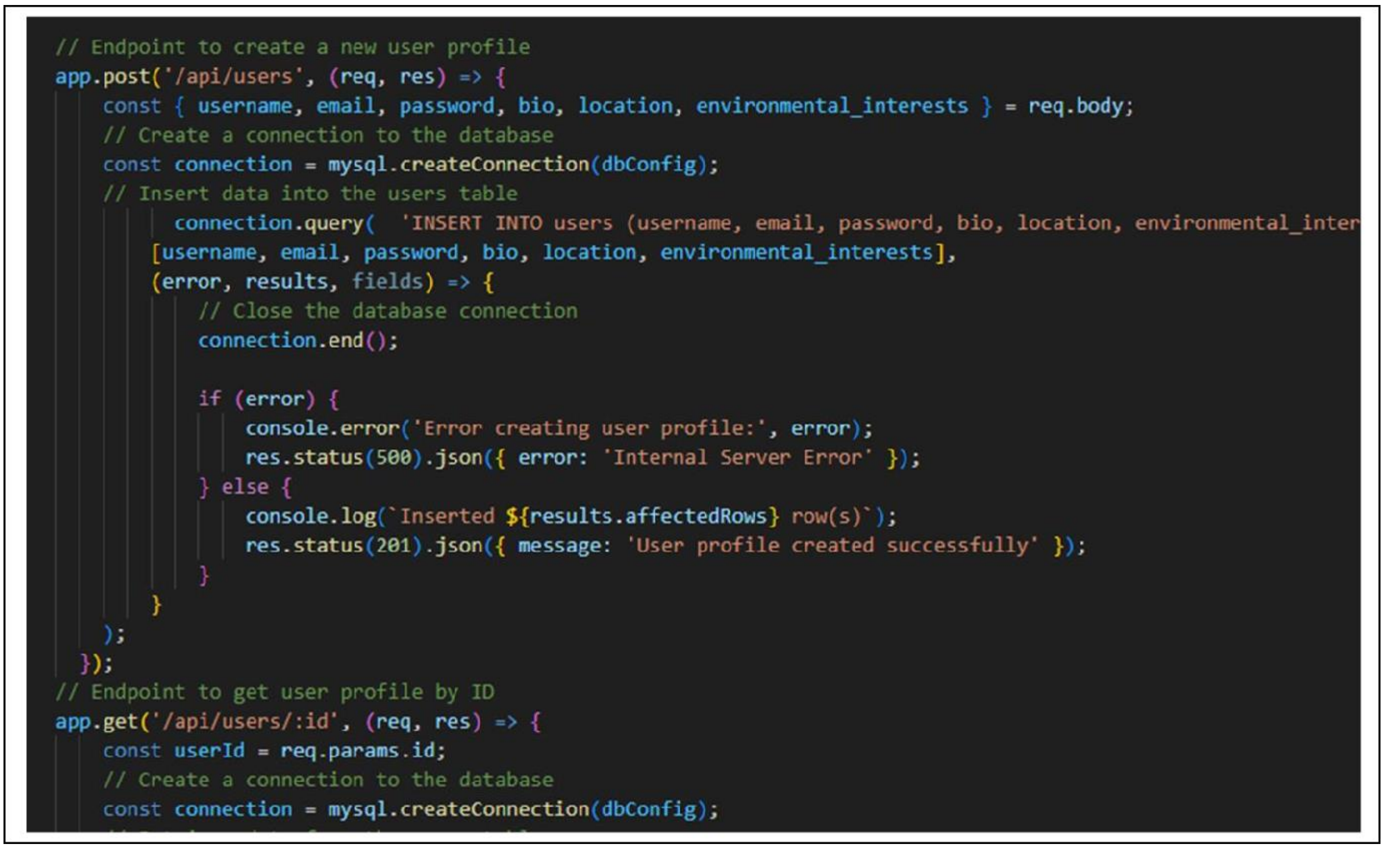
Extra options

| id | timestamp | source | air_quality | temperature | humidity | water_quality | biodiversity_metrics | user_id |
|---|---|---|---|---|---|---|---|---|
| 1 | 2023-11-22 21:07:02 | Manual Observation | 0 | 28 | 45 | Good | High | 1 |
| 6 | 2023-12-20 12:00:00 | Manual Observation | 8.5 | 27.3 | 60 | Good | High | NULL |

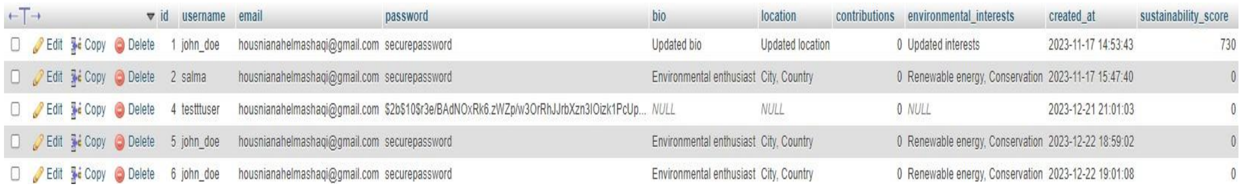
Check all With selected: Edit Copy Delete Export

## 2. User Profiles

Users can create and manage profiles to track their contributions and environmental interests. They can also connect with others who share similar concerns or locations.



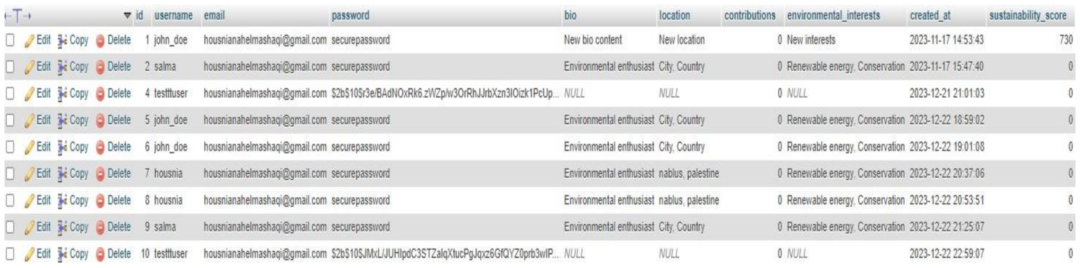
```
// Endpoint to create a new user profile
app.post('/api/users', (req, res) => {
    const { username, email, password, bio, location, environmental_interests } = req.body;
    // Create a connection to the database
    const connection = mysql.createConnection(dbConfig);
    // Insert data into the users table
        connection.query( 'INSERT INTO users (username, email, password, bio, location, environmental_inter
        [username, email, password, bio, location, environmental_interests],
        (error, results, fields) => {
            // Close the database connection
            connection.end();

            if (error) {
                console.error('Error creating user profile:', error);
                res.status(500).json({ error: 'Internal Server Error' });
            } else {
                console.log(`Inserted ${results.affectedRows} row(s)`);
                res.status(201).json({ message: 'User profile created successfully' });
            }
        }
    );
});
// Endpoint to get user profile by ID
app.get('/api/users/:id', (req, res) => {
    const userId = req.params.id;
    // Create a connection to the database
    const connection = mysql.createConnection(dbConfig);
```



| id | username | email | password | bio | location | contributions | environmental_interests | created_at | sustainability_score |
|---|---|---|---|---|---|---|---|---|---|
| 1 | john_doe | housnianahelmashaqi@gmail.com | securepassword | Updated bio | Updated location | 0 | Updated interests | 2023-11-17 14:53:43 | 730 |
| 2 | salma | housnianahelmashaqi@gmail.com | securepassword | Environmental enthusiast | City, Country | 0 | Renewable energy, Conservation | 2023-11-17 15:47:40 | 0 |
| 4 | testttuser | housnianahelmashaqi@gmail.com | $2b$10$r3e/BAdNOxRk6.zWZp/w3OrRhJJrbXzn3IOizk1PcUp... | NULL | NULL | 0 | NULL | 2023-12-21 21:01:03 | 0 |
| 5 | john_doe | housnianahelmashaqi@gmail.com | securepassword | Environmental enthusiast | City, Country | 0 | Renewable energy, Conservation | 2023-12-22 18:59:02 | 0 |
| 6 | john_doe | housnianahelmashaqi@gmail.com | securepassword | Environmental enthusiast | City, Country | 0 | Renewable energy, Conservation | 2023-12-22 19:01:08 | 0 |

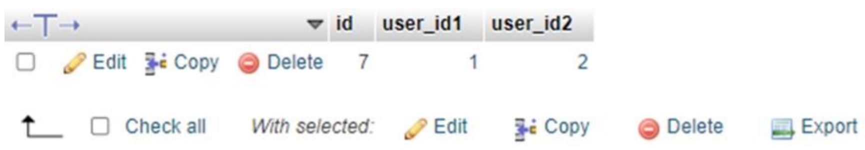
This screen is before add new user

And  this screen  after add users:



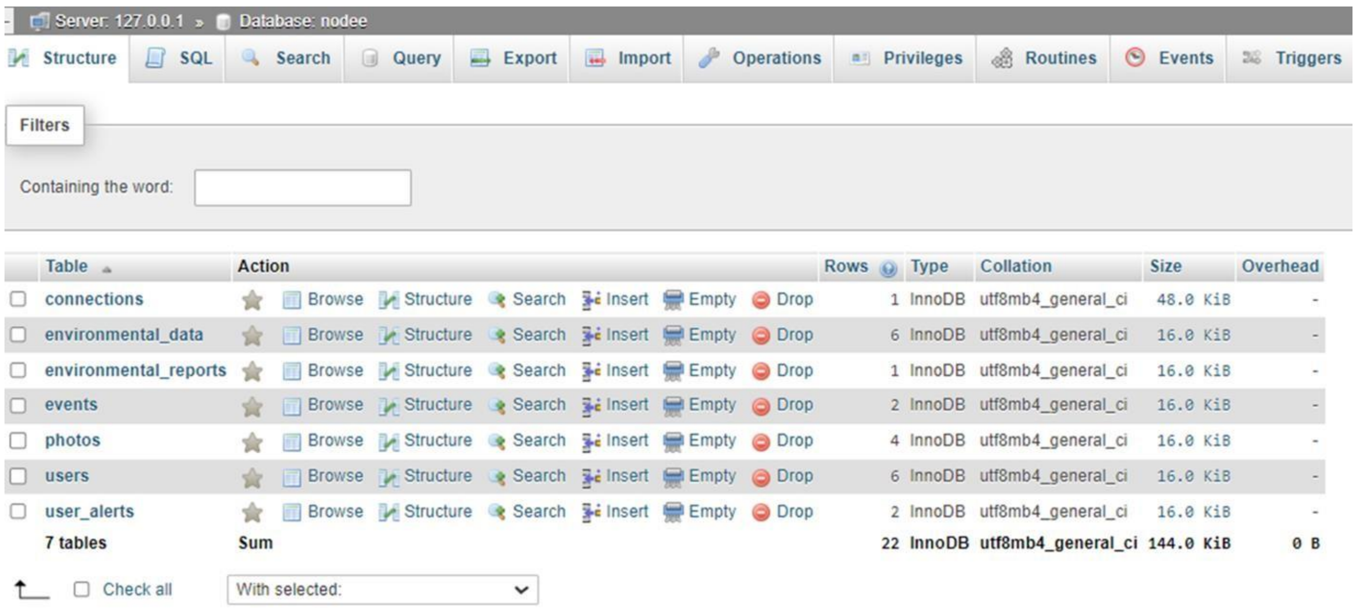
| | id | username | email | password | bio | location | contributions | environmental_interests | created_at | sustainability_score |
|---|---|---|---|---|---|---|---|---|---|---|
| Edit Copy Delete | 1 | john_doe | housnianahelmashaqi@gmail.com | securepassword | New bio content | New location | 0 | New interests | 2023-11-17 14:53:43 | 730 |
| Edit Copy Delete | 2 | salma | housnianahelmashaqi@gmail.com | securepassword | Environmental enthusiast | City, Country | 0 | Renewable energy, Conservation | 2023-11-17 15:47:40 | 0 |
| Edit Copy Delete | 4 | testttuser | housnianahelmashaqi@gmail.com | $2b$10$r3e/BAdNOxRk6.zWZp/w3OrRhJJrbXzn3lOizk1PcUp... | NULL | NULL | 0 | NULL | 2023-12-21 21:01:03 | 0 |
| Edit Copy Delete | 5 | john_doe | housnianahelmashaqi@gmail.com | securepassword | Environmental enthusiast | City, Country | 0 | Renewable energy, Conservation | 2023-12-22 18:59:02 | 0 |
| Edit Copy Delete | 6 | john_doe | housnianahelmashaqi@gmail.com | securepassword | Environmental enthusiast | City, Country | 0 | Renewable energy, Conservation | 2023-12-22 19:01:08 | 0 |
| Edit Copy Delete | 7 | housnia | housnianahelmashaqi@gmail.com | securepassword | Environmental enthusiast | nablus, palestine | 0 | Renewable energy, Conservation | 2023-12-22 20:37:06 | 0 |
| Edit Copy Delete | 8 | housnia | housnianahelmashaqi@gmail.com | securepassword | Environmental enthusiast | nablus, palestine | 0 | Renewable energy, Conservation | 2023-12-22 20:53:51 | 0 |
| Edit Copy Delete | 9 | salma | housnianahelmashaqi@gmail.com | securepassword | Environmental enthusiast | City, Country | 0 | Renewable energy, Conservation | 2023-12-22 21:25:07 | 0 |
| Edit Copy Delete | 10 | testttuser | housnianahelmashaqi@gmail.com | $2b$10$JMxL/JUHlpdC3STZalqXtucPgJqxz6GfQYZ0prb3wlP... | NULL | NULL | 0 | NULL | 2023-12-22 22:59:07 | 0 |

Table of connections :



| | id | user_id1 | user_id2 |
|---|---|---|---|
| Edit Copy Delete | 7 | 1 | 2 |

All tables in data base :

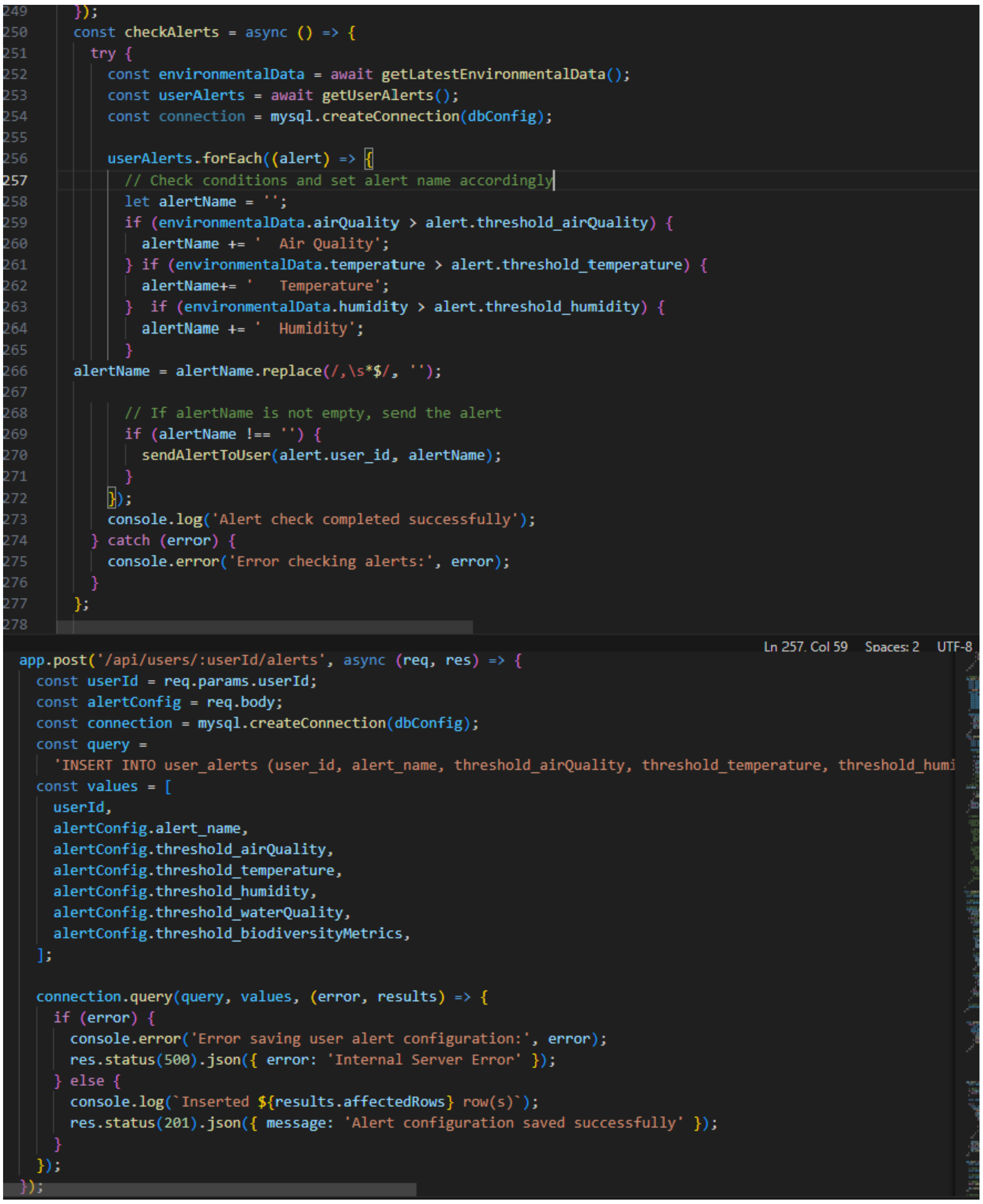


Server: 127.0.0.1 » Database: nodee

| Table | Action | Rows | Type | Collation | Size | Overhead |
|---|---|---|---|---|---|---|
| connections | Browse Structure Search Insert Empty Drop | 1 | InnoDB | utf8mb4_general_ci | 48.0 KiB | - |
| environmental_data | Browse Structure Search Insert Empty Drop | 6 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| environmental_reports | Browse Structure Search Insert Empty Drop | 1 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| events | Browse Structure Search Insert Empty Drop | 2 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| photos | Browse Structure Search Insert Empty Drop | 4 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| users | Browse Structure Search Insert Empty Drop | 6 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| user_alerts | Browse Structure Search Insert Empty Drop | 2 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| 7 tables | Sum | 22 | InnoDB | utf8mb4_general_ci | 144.0 KiB | 0 B |

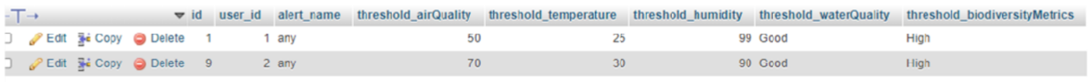
## 3. Environmental Alerts

Set up an alerting system that notifies users about significant changes or concerning trends in environmental data. Users can configure alert thresholds based on their interests, if we add data in environmental_data table check if values > threshold it sent Email :



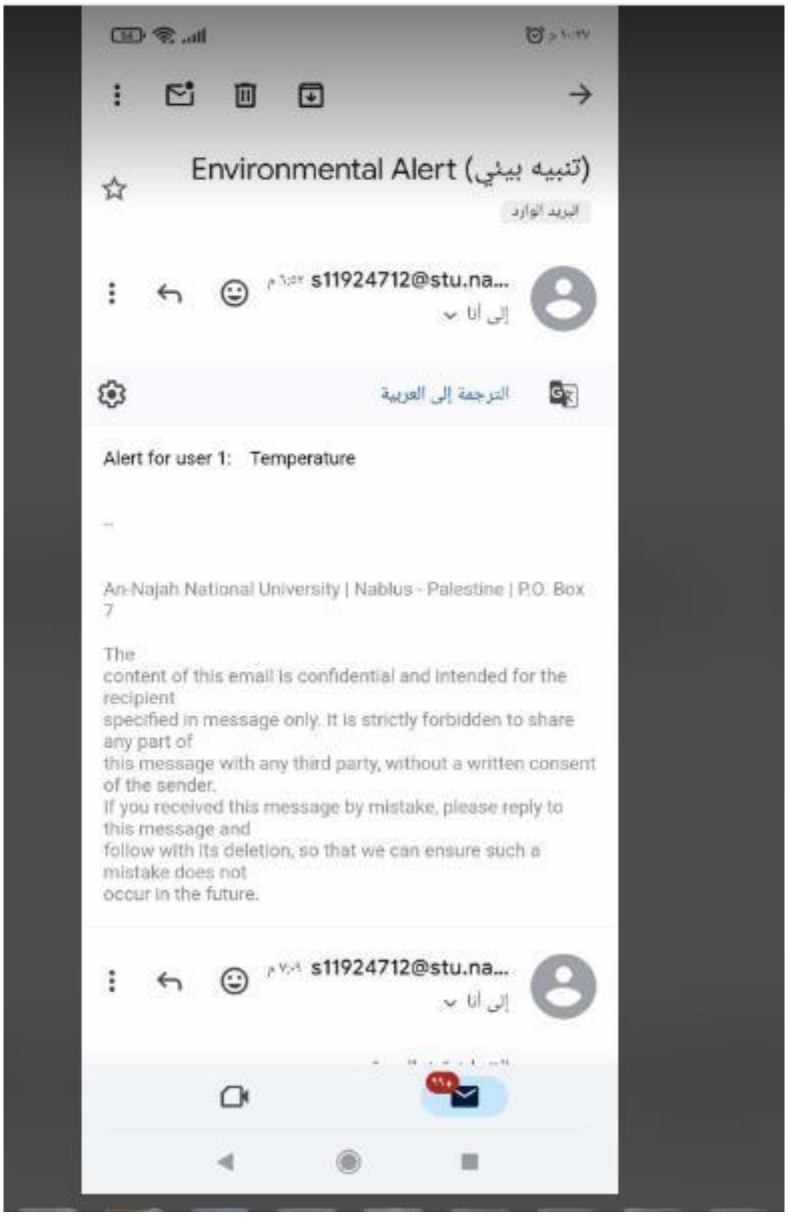
```
});
const checkAlerts = async () => {
  try {
    const environmentalData = await getLatestEnvironmentalData();
    const userAlerts = await getUserAlerts();
    const connection = mysql.createConnection(dbConfig);

    userAlerts.forEach((alert) => {
      // Check conditions and set alert name accordingly
      let alertName = '';
      if (environmentalData.airQuality > alert.threshold_airQuality) {
        alertName += '  Air Quality';
      } if (environmentalData.temperature > alert.threshold_temperature) {
        alertName+= '   Temperature';
      }  if (environmentalData.humidity > alert.threshold_humidity) {
        alertName += '  Humidity';
      }
alertName = alertName.replace(/,\s*$/, '');

      // If alertName is not empty, send the alert
      if (alertName !== '') {
        sendAlertToUser(alert.user_id, alertName);
      }
    });
    console.log('Alert check completed successfully');
  } catch (error) {
    console.error('Error checking alerts:', error);
  }
};
```

```
app.post('/api/users/:userId/alerts', async (req, res) => {
  const userId = req.params.userId;
  const alertConfig = req.body;
  const connection = mysql.createConnection(dbConfig);
  const query =
    'INSERT INTO user_alerts (user_id, alert_name, threshold_airQuality, threshold_temperature, threshold_humi
  const values = [
    userId,
    alertConfig.alert_name,
    alertConfig.threshold_airQuality,
    alertConfig.threshold_temperature,
    alertConfig.threshold_humidity,
    alertConfig.threshold_waterQuality,
    alertConfig.threshold_biodiversityMetrics,
  ];

  connection.query(query, values, (error, results) => {
    if (error) {
      console.error('Error saving user alert configuration:', error);
      res.status(500).json({ error: 'Internal Server Error' });
    } else {
      console.log(`Inserted ${results.affectedRows} row(s)`);
      res.status(201).json({ message: 'Alert configuration saved successfully' });
    }
  });
});
```



| | id | user_id | alert_name | threshold_airQuality | threshold_temperature | threshold_humidity | threshold_waterQuality | threshold_biodiversityMetrics |
|---|---|---|---|---|---|---|---|---|
| Edit Copy Delete | 1 | 1 | any | 50 | 25 | 99 | Good | High |
| Edit Copy Delete | 9 | 2 | any | 70 | 30 | 90 | Good | High |

The email :



Environmental Alert (تنبيه بيئي)

البريد الوارد

s11924712@stu.na...

إلى أنا

الترجمة إلى العربية

Alert for user 1: Temperature

--

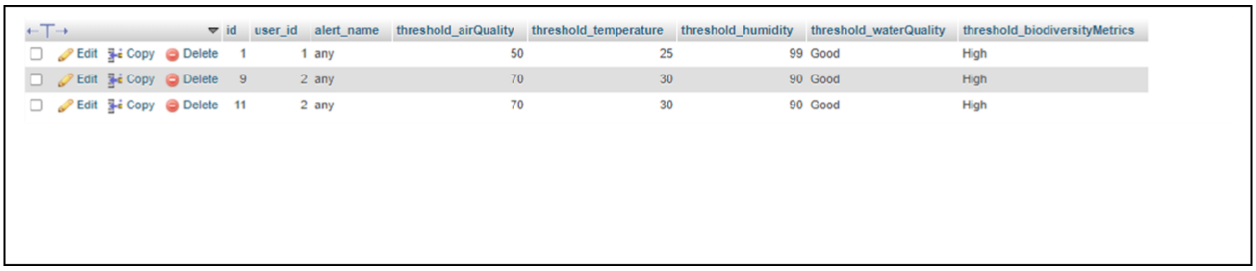
An-Najah National University | Nablus - Palestine | P.O. Box 7

The
content of this email is confidential and intended for the recipient
specified in message only. It is strictly forbidden to share any part of
this message with any third party, without a written consent of the sender.
If you received this message by mistake, please reply to this message and
follow with its deletion, so that we can ensure such a mistake does not
occur in the future.

s11924712@stu.na...

إلى أنا

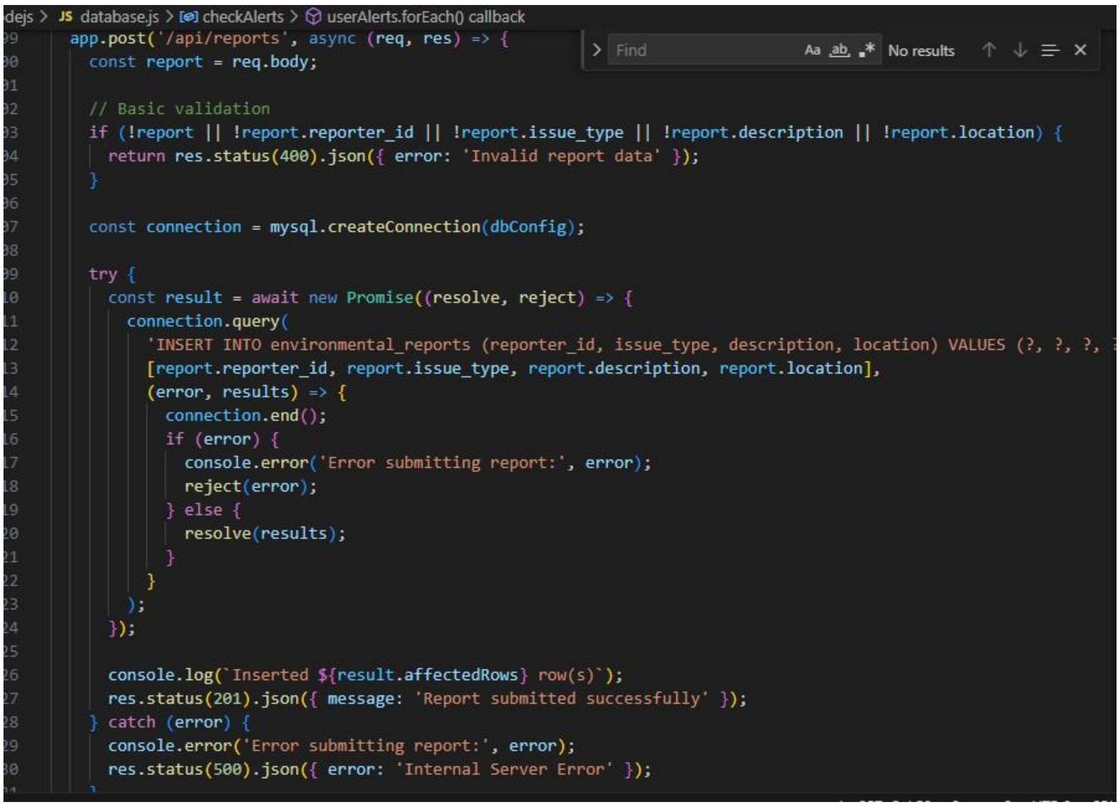
Table user_Alert after :



| | id | user_id | alert_name | threshold_airQuality | threshold_temperature | threshold_humidity | threshold_waterQuality | threshold_biodiversityMetrics |
|---|---|---|---|---|---|---|---|---|
| Edit Copy Delete | 1 | 1 | any | 50 | 25 | 99 | Good | High |
| Edit Copy Delete | 9 | 2 | any | 70 | 30 | 90 | Good | High |
| Edit Copy Delete | 11 | 2 | any | 70 | 30 | 90 | Good | High |

## 4. Community Reporting:

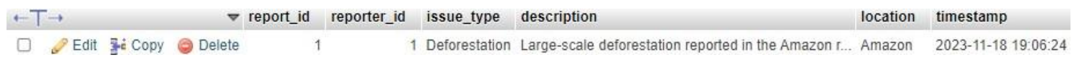
Allow users to report environmental issues, such as pollution, deforestation, or wildlife endangerment.



dejs > JS database.js > checkAlerts > userAlerts.forEach() callback

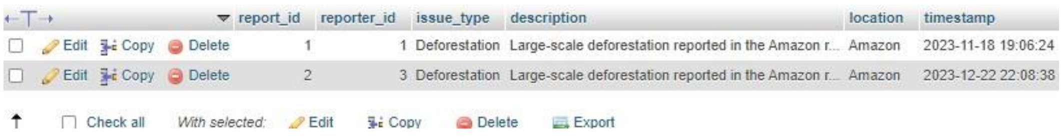
```
app.post('/api/reports', async (req, res) => {
  const report = req.body;

  // Basic validation
  if (!report || !report.reporter_id || !report.issue_type || !report.description || !report.location) {
    return res.status(400).json({ error: 'Invalid report data' });
  }

  const connection = mysql.createConnection(dbConfig);

  try {
    const result = await new Promise((resolve, reject) => {
      connection.query(
        'INSERT INTO environmental_reports (reporter_id, issue_type, description, location) VALUES (?, ?, ?,
        [report.reporter_id, report.issue_type, report.description, report.location],
        (error, results) => {
          connection.end();
          if (error) {
            console.error('Error submitting report:', error);
            reject(error);
          } else {
            resolve(results);
          }
        }
      );
    });

    console.log(`Inserted ${result.affectedRows} row(s)`);
    res.status(201).json({ message: 'Report submitted successfully' });
  } catch (error) {
    console.error('Error submitting report:', error);
    res.status(500).json({ error: 'Internal Server Error' });
```

## Report before :



| | report_id | reporter_id | issue_type | description | location | timestamp |
|---|---|---|---|---|---|---|
| Edit Copy Delete | 1 | 1 | Deforestation | Large-scale deforestation reported in the Amazon r... | Amazon | 2023-11-18 19:06:24 |

## Report after :

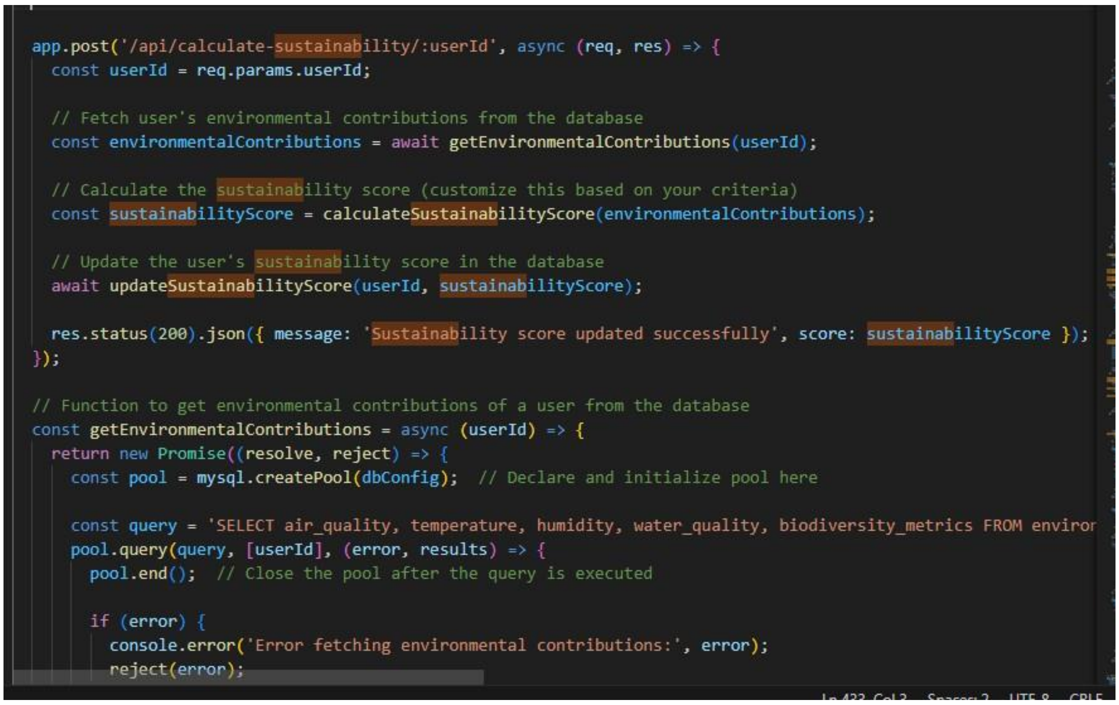


| | report_id | reporter_id | issue_type | description | location | timestamp |
|---|---|---|---|---|---|---|
| Edit Copy Delete | 1 | 1 | Deforestation | Large-scale deforestation reported in the Amazon r... | Amazon | 2023-11-18 19:06:24 |
| Edit Copy Delete | 2 | 3 | Deforestation | Large-scale deforestation reported in the Amazon r... | Amazon | 2023-12-22 22:08:38 |

Check all With selected: Edit Copy Delete Export

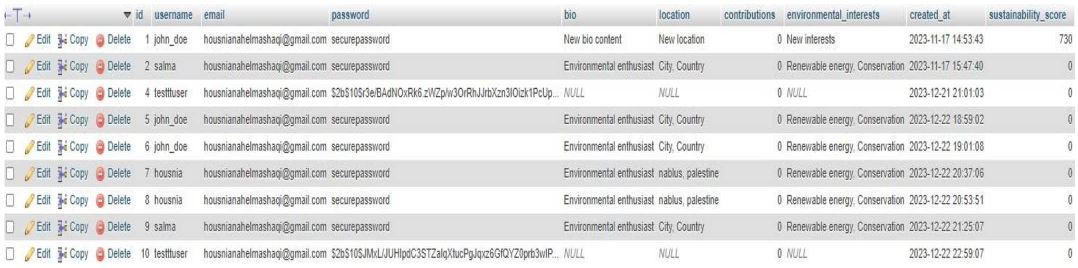
## 5. Sustainability Score

Develop a scoring system that assesses users' environmental contributions and sustainability efforts based on the data they provide and the actions they take.



```javascript
app.post('/api/calculate-sustainability/:userId', async (req, res) => {
  const userId = req.params.userId;

  // Fetch user's environmental contributions from the database
  const environmentalContributions = await getEnvironmentalContributions(userId);

  // Calculate the sustainability score (customize this based on your criteria)
  const sustainabilityScore = calculateSustainabilityScore(environmentalContributions);

  // Update the user's sustainability score in the database
  await updateSustainabilityScore(userId, sustainabilityScore);

  res.status(200).json({ message: 'Sustainability score updated successfully', score: sustainabilityScore });
});

// Function to get environmental contributions of a user from the database
const getEnvironmentalContributions = async (userId) => {
  return new Promise((resolve, reject) => {
    const pool = mysql.createPool(dbConfig);  // Declare and initialize pool here

    const query = 'SELECT air_quality, temperature, humidity, water_quality, biodiversity_metrics FROM enviror
    pool.query(query, [userId], (error, results) => {
      pool.end();  // Close the pool after the query is executed

      if (error) {
        console.error('Error fetching environmental contributions:', error);
        reject(error);
```

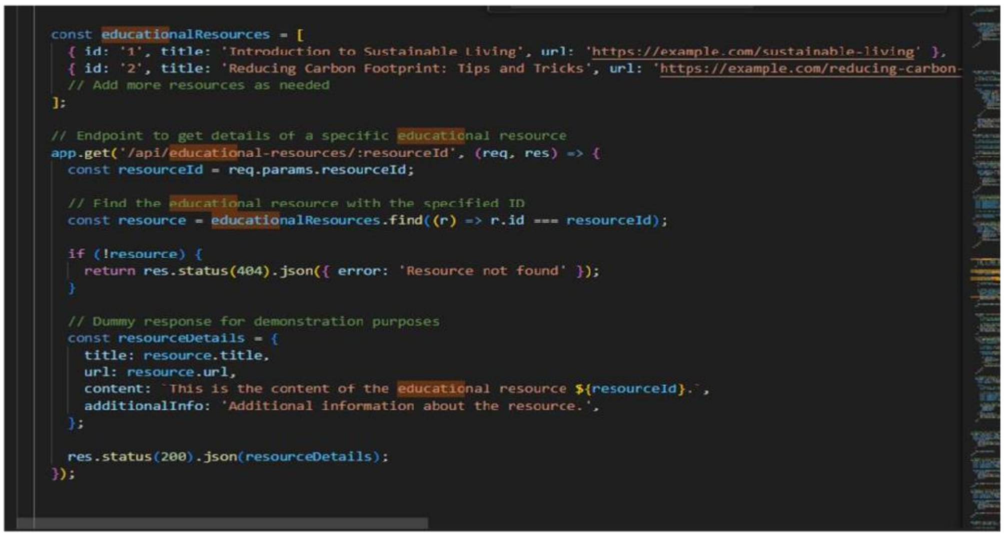
The result in users table :



| | id | username | email | password | bio | location | contributions | environmental_interests | created_at | sustainability_score |
|---|---|---|---|---|---|---|---|---|---|---|
| Edit Copy Delete | 1 | john_doe | housnianahelmashaqi@gmail.com | securepassword | New bio content | New location | 0 | New interests | 2023-11-17 14:53:43 | 730 |
| Edit Copy Delete | 2 | salma | housnianahelmashaqi@gmail.com | securepassword | Environmental enthusiast | City, Country | 0 | Renewable energy, Conservation | 2023-11-17 15:47:40 | 0 |
| Edit Copy Delete | 4 | testttuser | housnianahelmashaqi@gmail.com | $2b$10$r3e/BAdNOxRk6.zWZp/w3OrRhJJrbXzn3lOizk1PcUp... | NULL | NULL | 0 | NULL | 2023-12-21 21:01:03 | 0 |
| Edit Copy Delete | 5 | john_doe | housnianahelmashaqi@gmail.com | securepassword | Environmental enthusiast | City, Country | 0 | Renewable energy, Conservation | 2023-12-22 18:59:02 | 0 |
| Edit Copy Delete | 6 | john_doe | housnianahelmashaqi@gmail.com | securepassword | Environmental enthusiast | City, Country | 0 | Renewable energy, Conservation | 2023-12-22 19:01:08 | 0 |
| Edit Copy Delete | 7 | housnia | housnianahelmashaqi@gmail.com | securepassword | Environmental enthusiast | nablus, palestine | 0 | Renewable energy, Conservation | 2023-12-22 20:37:06 | 0 |
| Edit Copy Delete | 8 | housnia | housnianahelmashaqi@gmail.com | securepassword | Environmental enthusiast | nablus, palestine | 0 | Renewable energy, Conservation | 2023-12-22 20:53:51 | 0 |
| Edit Copy Delete | 9 | salma | housnianahelmashaqi@gmail.com | securepassword | Environmental enthusiast | City, Country | 0 | Renewable energy, Conservation | 2023-12-22 21:25:07 | 0 |
| Edit Copy Delete | 10 | testttuser | housnianahelmashaqi@gmail.com | $2b$10$JMxL/JUHlpdC3STZalqXtucPgJqxz6GfQYZ0prb3wlP... | NULL | NULL | 0 | NULL | 2023-12-22 22:59:07 | 0 |

## 6. Educational Resources

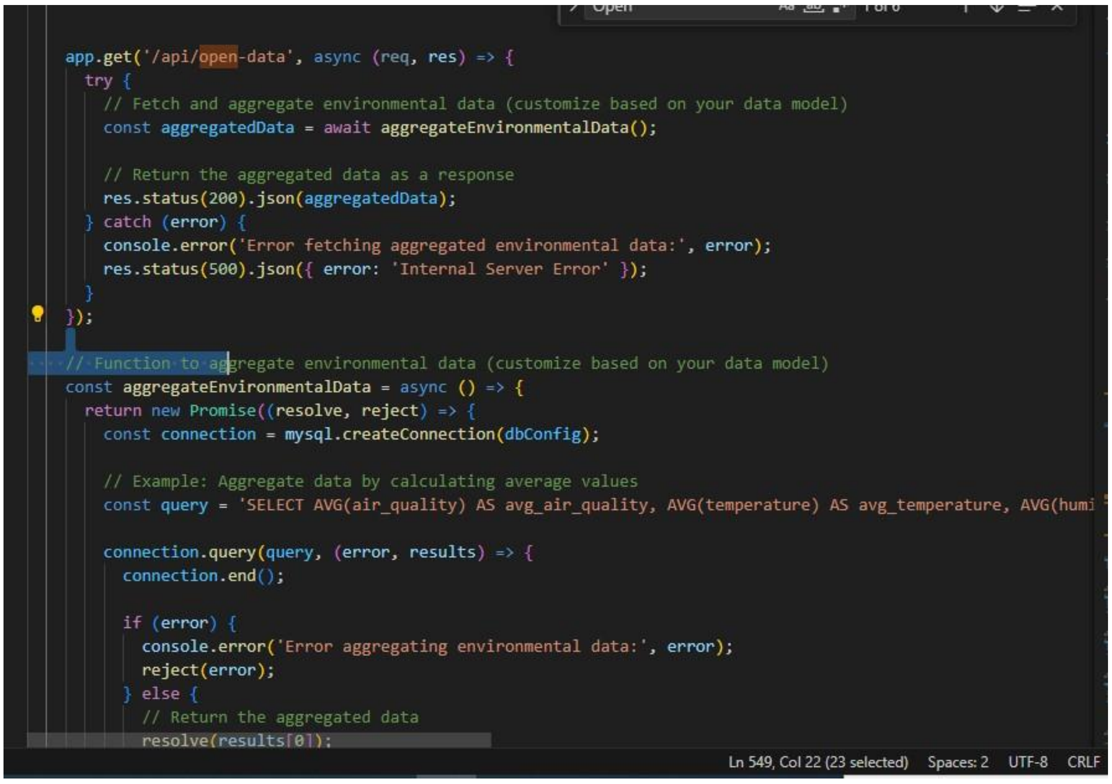
Offer educational resources, articles, and guides on environmental topics to raise awareness and educate users on sustainable practices.



```javascript
const educationalResources = [
  { id: '1', title: 'Introduction to Sustainable Living', url: 'https://example.com/sustainable-living' },
  { id: '2', title: 'Reducing Carbon Footprint: Tips and Tricks', url: 'https://example.com/reducing-carbon-
  // Add more resources as needed
];

// Endpoint to get details of a specific educational resource
app.get('/api/educational-resources/:resourceId', (req, res) => {
  const resourceId = req.params.resourceId;

  // Find the educational resource with the specified ID
  const resource = educationalResources.find((r) => r.id === resourceId);

  if (!resource) {
    return res.status(404).json({ error: 'Resource not found' });
  }

  // Dummy response for demonstration purposes
  const resourceDetails = {
    title: resource.title,
    url: resource.url,
    content: `This is the content of the educational resource ${resourceId}.`,
    additionalInfo: 'Additional information about the resource.',
  };

  res.status(200).json(resourceDetails);
});
```

## 7. Open Data Access

Provide APIs for researchers, scientists, and organizations to access the aggregated environmental data for research and analysis.

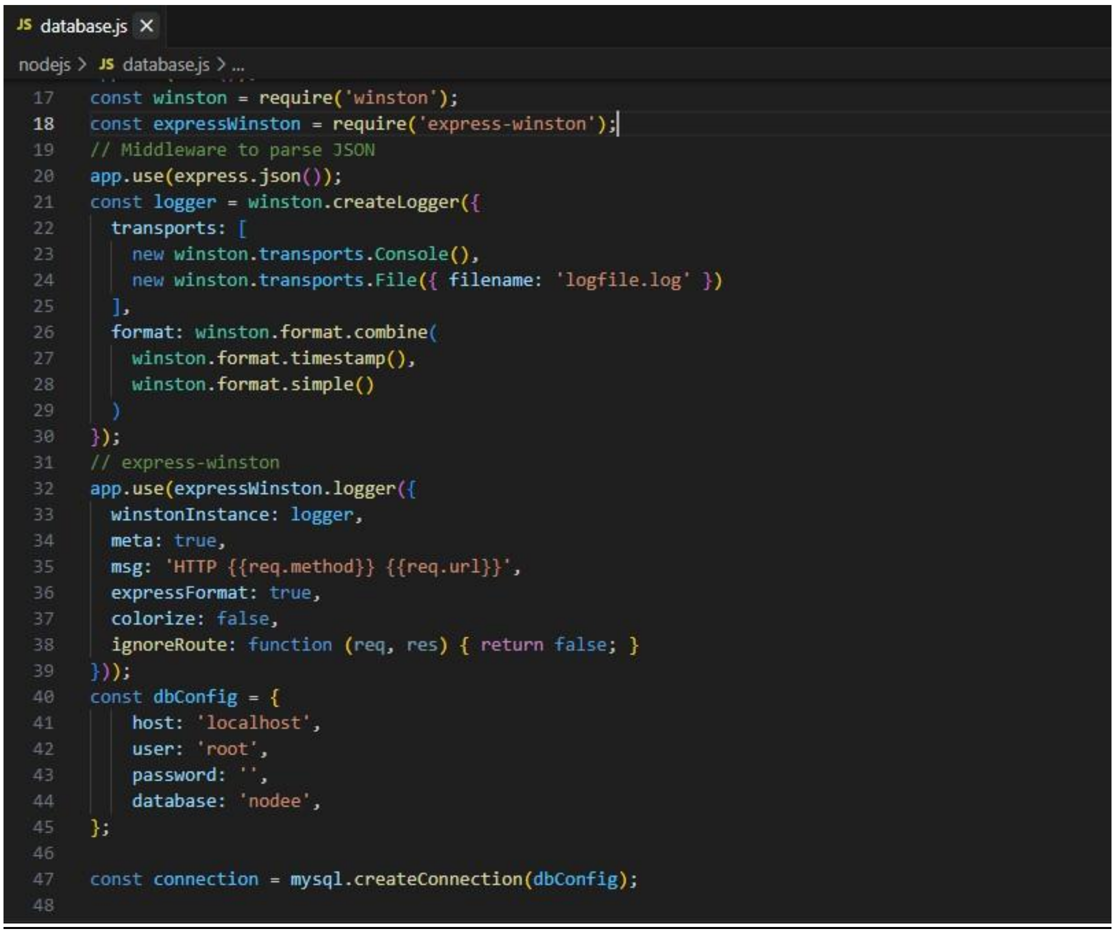


```
app.get('/api/open-data', async (req, res) => {
  try {
    // Fetch and aggregate environmental data (customize based on your data model)
    const aggregatedData = await aggregateEnvironmentalData();

    // Return the aggregated data as a response
    res.status(200).json(aggregatedData);
  } catch (error) {
    console.error('Error fetching aggregated environmental data:', error);
    res.status(500).json({ error: 'Internal Server Error' });
  }
});

// Function to aggregate environmental data (customize based on your data model)
const aggregateEnvironmentalData = async () => {
  return new Promise((resolve, reject) => {
    const connection = mysql.createConnection(dbConfig);

    // Example: Aggregate data by calculating average values
    const query = 'SELECT AVG(air_quality) AS avg_air_quality, AVG(temperature) AS avg_temperature, AVG(humi

    connection.query(query, (error, results) => {
      connection.end();

      if (error) {
        console.error('Error aggregating environmental data:', error);
        reject(error);
      } else {
        // Return the aggregated data
        resolve(results[0]);
```

## External API Integration:

### 1. Logging file feature (error log file):

is a file that contains a detailed record of events and errors that may occur during the execution of an application or system. This file serves as a diagnostic tool to identify and examine errors and issues that may arise during program execution, monitoring and recording of errors effectively and providing valuable information to enhance the quality of the program and identify issues efficiently.

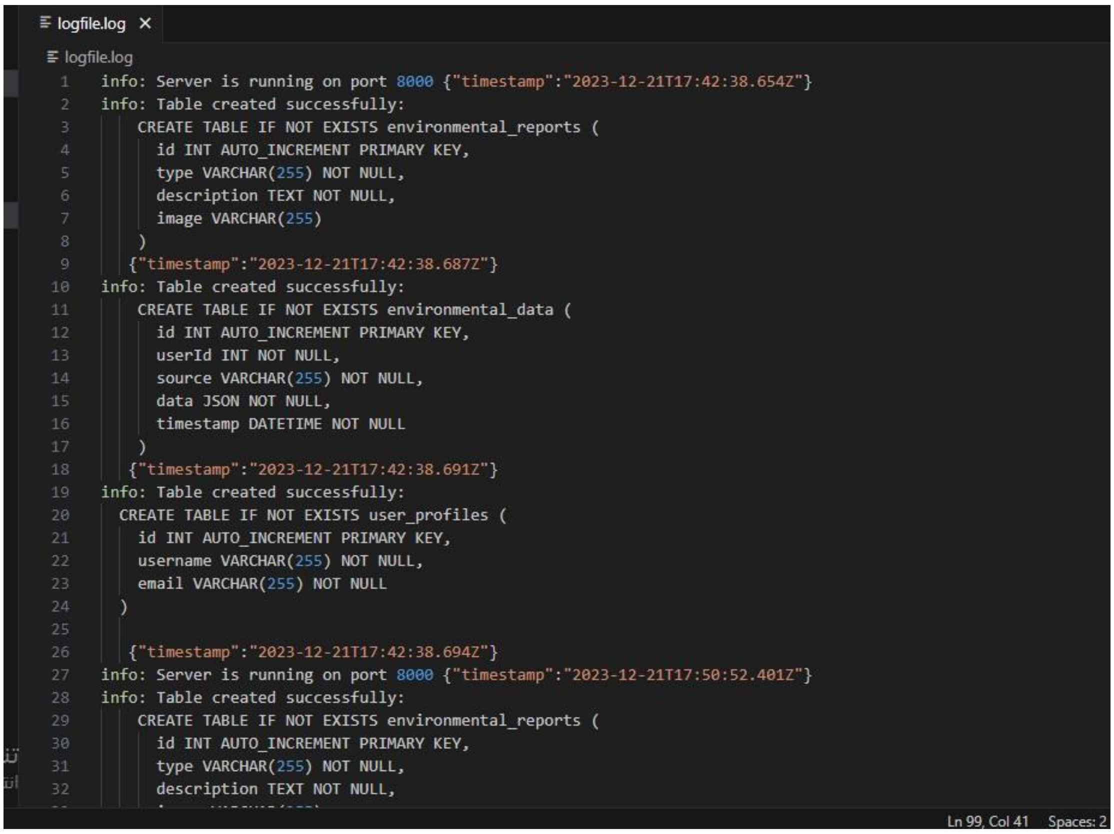


database.js

nodejs > database.js > ...

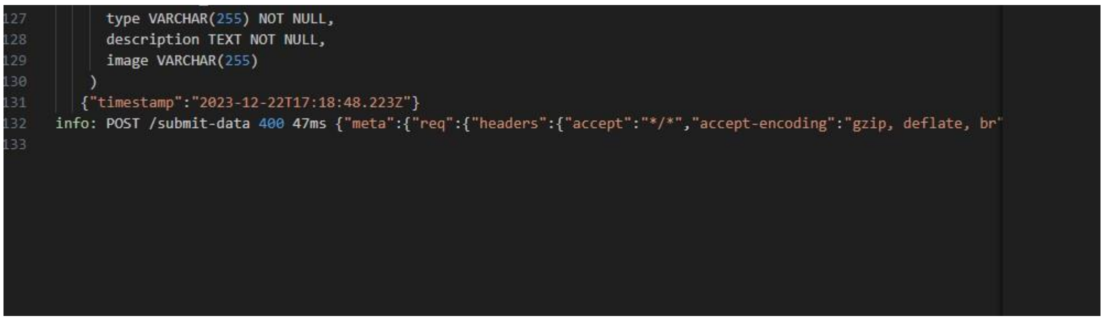
```javascript
const winston = require('winston');
const expressWinston = require('express-winston');
// Middleware to parse JSON
app.use(express.json());
const logger = winston.createLogger({
  transports: [
    new winston.transports.Console(),
    new winston.transports.File({ filename: 'logfile.log' })
  ],
  format: winston.format.combine(
    winston.format.timestamp(),
    winston.format.simple()
  )
});
// express-winston
app.use(expressWinston.logger({
  winstonInstance: logger,
  meta: true,
  msg: 'HTTP {{req.method}} {{req.url}}',
  expressFormat: true,
  colorize: false,
  ignoreRoute: function (req, res) { return false; }
}));
const dbConfig = {
    host: 'localhost',
    user: 'root',
    password: '',
    database: 'nodee',
};

const connection = mysql.createConnection(dbConfig);

```

After that the app create a logfile.log :



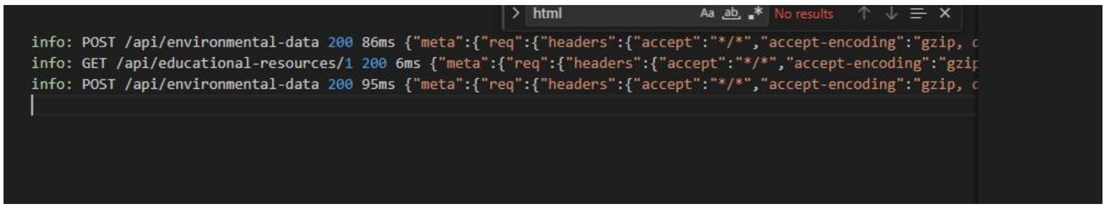
```
info: Server is running on port 8000 {"timestamp":"2023-12-21T17:42:38.654Z"}
info: Table created successfully:
    CREATE TABLE IF NOT EXISTS environmental_reports (
      id INT AUTO_INCREMENT PRIMARY KEY,
      type VARCHAR(255) NOT NULL,
      description TEXT NOT NULL,
      image VARCHAR(255)
    )
   {"timestamp":"2023-12-21T17:42:38.687Z"}
info: Table created successfully:
    CREATE TABLE IF NOT EXISTS environmental_data (
      id INT AUTO_INCREMENT PRIMARY KEY,
      userId INT NOT NULL,
      source VARCHAR(255) NOT NULL,
      data JSON NOT NULL,
      timestamp DATETIME NOT NULL
    )
   {"timestamp":"2023-12-21T17:42:38.691Z"}
info: Table created successfully:
  CREATE TABLE IF NOT EXISTS user_profiles (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL
  )

   {"timestamp":"2023-12-21T17:42:38.694Z"}
info: Server is running on port 8000 {"timestamp":"2023-12-21T17:50:52.401Z"}
info: Table created successfully:
    CREATE TABLE IF NOT EXISTS environmental_reports (
      id INT AUTO_INCREMENT PRIMARY KEY,
      type VARCHAR(255) NOT NULL,
      description TEXT NOT NULL,
```

if they are an error (as example in postman in body I sent Incomplete information) and the result was :



```
      type VARCHAR(255) NOT NULL,
      description TEXT NOT NULL,
      image VARCHAR(255)
    )
   {"timestamp":"2023-12-22T17:18:48.223Z"}
info: POST /submit-data 400 47ms {"meta":{"req":{"headers":{"accept":"*/*","accept-encoding":"gzip, deflate, br"
```

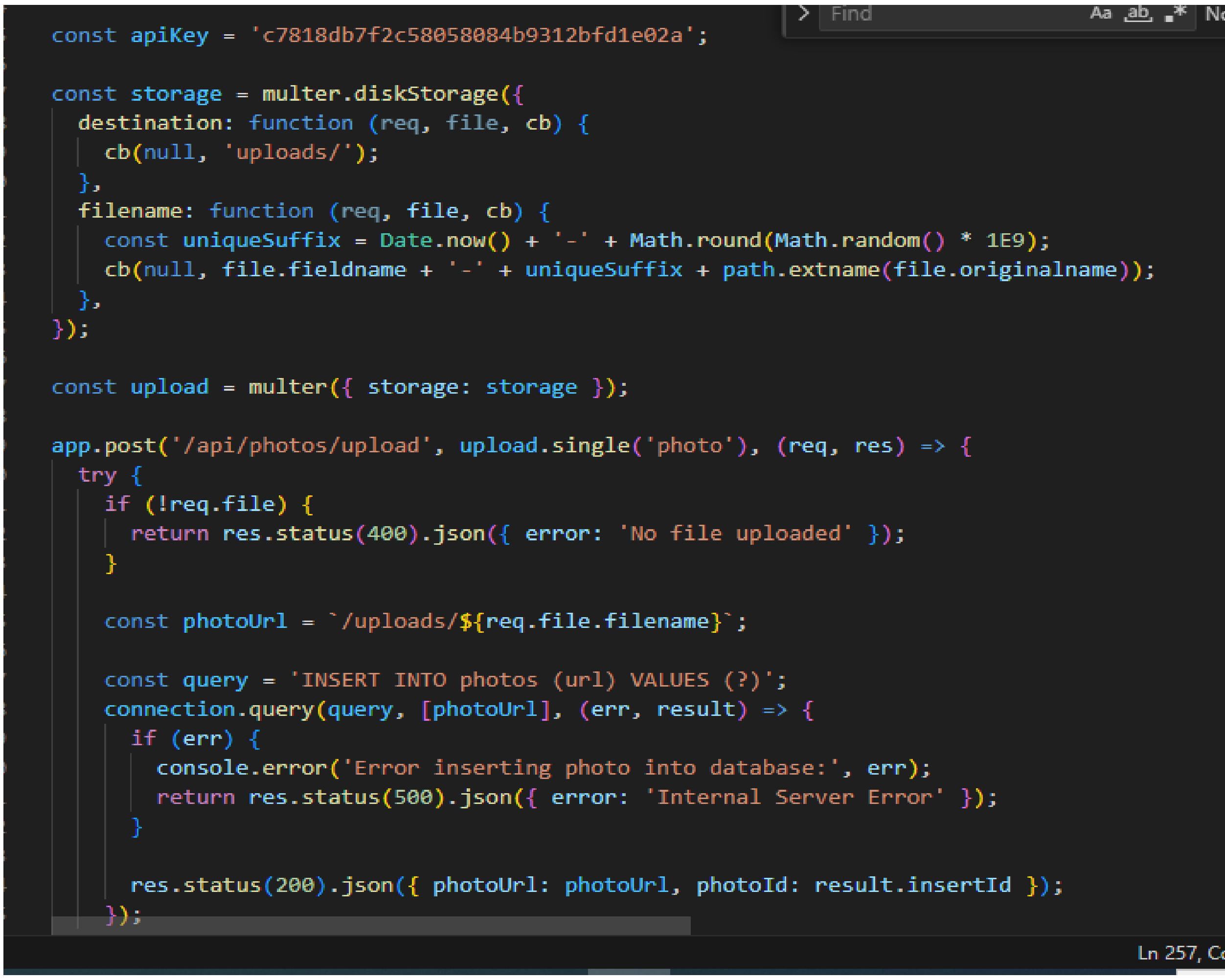
If  no errors the result was :



```
info: POST /api/environmental-data 200 86ms {"meta":{"req":{"headers":{"accept":"*/*","accept-encoding":"gzip, c
info: GET /api/educational-resources/1 200 6ms {"meta":{"req":{"headers":{"accept":"*/*","accept-encoding":"gzip
info: POST /api/environmental-data 200 95ms {"meta":{"req":{"headers":{"accept":"*/*","accept-encoding":"gzip, c
```

## 2. Adding files (photos) feature:

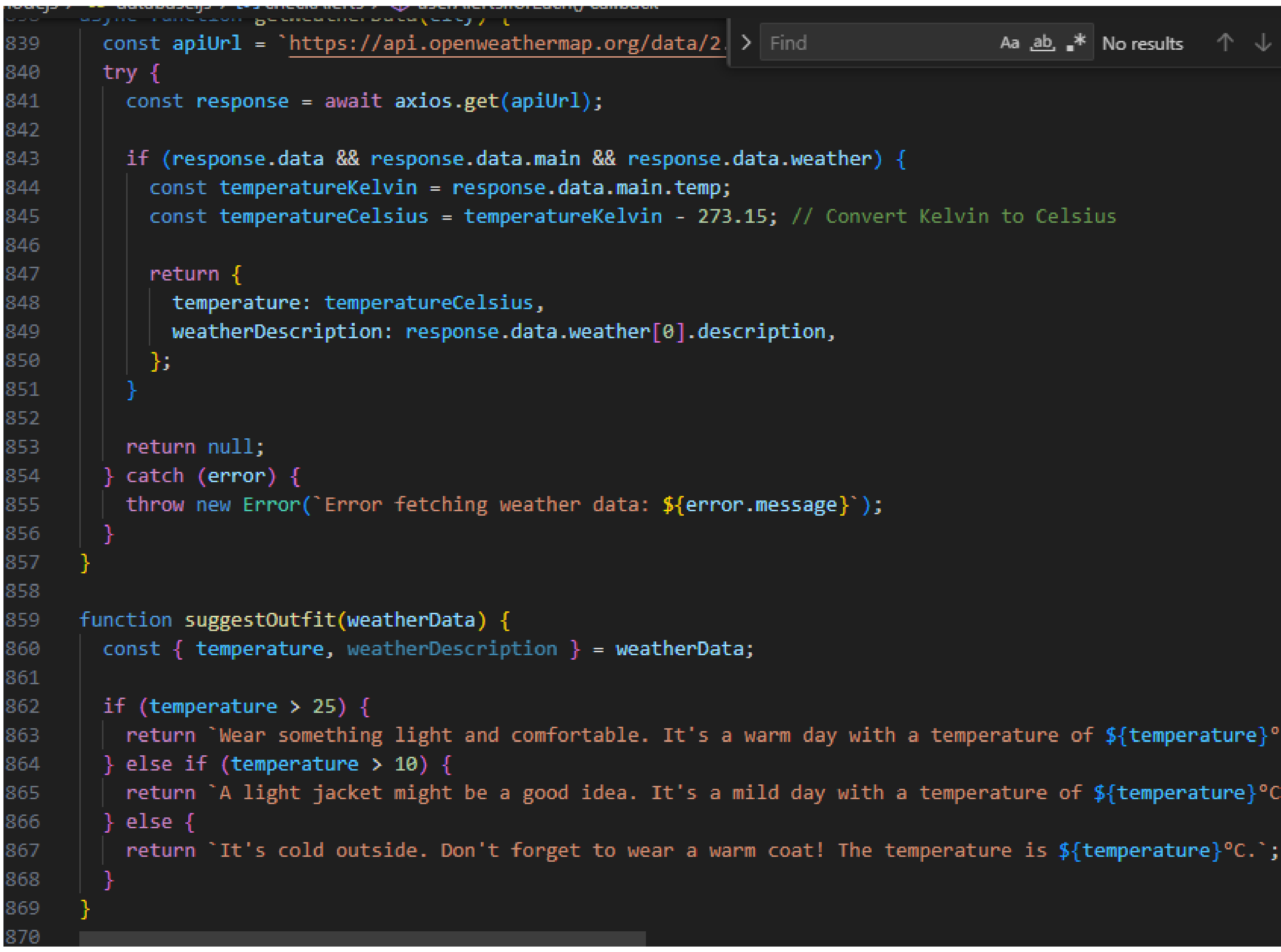
This feature, in short, creates a path for managing file uploads, saves the files that are uploaded to a server directory, and saves the file URLs in a database table called "photos." It offers suitable error handling for many situations, such as database failures and missing files.



```
const apiKey = 'c7818db7f2c58058084b9312bfd1e02a';

const storage = multer.diskStorage({
  destination: function (req, file, cb) {
    cb(null, 'uploads/');
  },
  filename: function (req, file, cb) {
    const uniqueSuffix = Date.now() + '-' + Math.round(Math.random() * 1E9);
    cb(null, file.fieldname + '-' + uniqueSuffix + path.extname(file.originalname));
  },
});

const upload = multer({ storage: storage });

app.post('/api/photos/upload', upload.single('photo'), (req, res) => {
  try {
    if (!req.file) {
      return res.status(400).json({ error: 'No file uploaded' });
    }

    const photoUrl = `/uploads/${req.file.filename}`;

    const query = 'INSERT INTO photos (url) VALUES (?)';
    connection.query(query, [photoUrl], (err, result) => {
      if (err) {
        console.error('Error inserting photo into database:', err);
        return res.status(500).json({ error: 'Internal Server Error' });
      }

      res.status(200).json({ photoUrl: photoUrl, photoId: result.insertId });
    });
```

## 3. Dress suggestion from weather condition feature:

these features can be combined to get a city's weather information and offer clothing recommendations depending on the temperature and weather description. Please be aware, though, that in order for the getWeatherData function to function as intended, you must define the apiKey variable and ensure that the Axios library is correctly imported and configured in your code.



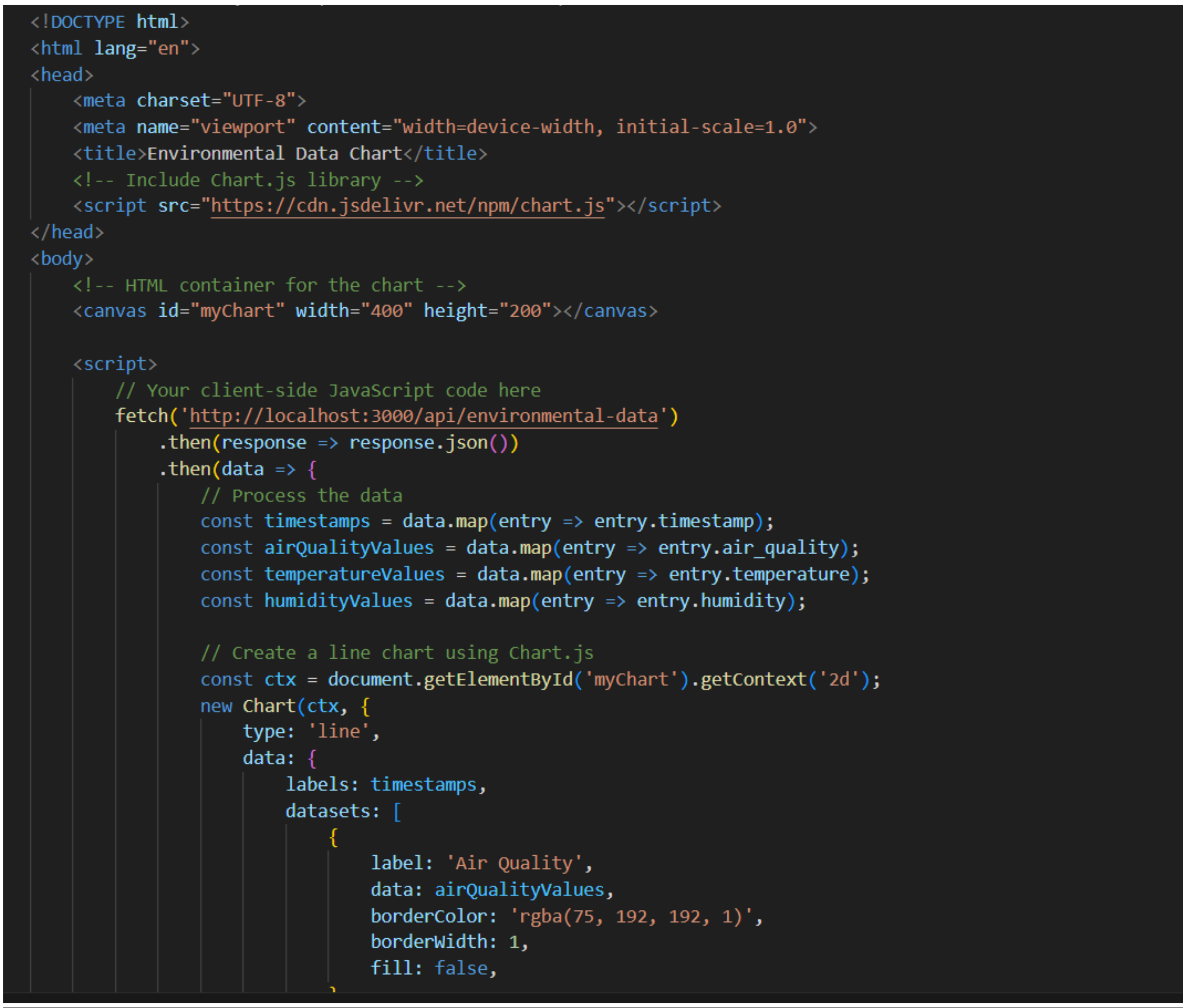
```
  const apiUrl = `https://api.openweathermap.org/data/2.
  try {
    const response = await axios.get(apiUrl);

    if (response.data && response.data.main && response.data.weather) {
      const temperatureKelvin = response.data.main.temp;
      const temperatureCelsius = temperatureKelvin - 273.15; // Convert Kelvin to Celsius

      return {
        temperature: temperatureCelsius,
        weatherDescription: response.data.weather[0].description,
      };
    }

    return null;
  } catch (error) {
    throw new Error(`Error fetching weather data: ${error.message}`);
  }
}

function suggestOutfit(weatherData) {
  const { temperature, weatherDescription } = weatherData;

  if (temperature > 25) {
    return `Wear something light and comfortable. It's a warm day with a temperature of ${temperature}°
  } else if (temperature > 10) {
    return `A light jacket might be a good idea. It's a mild day with a temperature of ${temperature}°C
  } else {
    return `It's cold outside. Don't forget to wear a warm coat! The temperature is ${temperature}°C.`;
  }
}
```

Find | Aa | ab | .* | No results

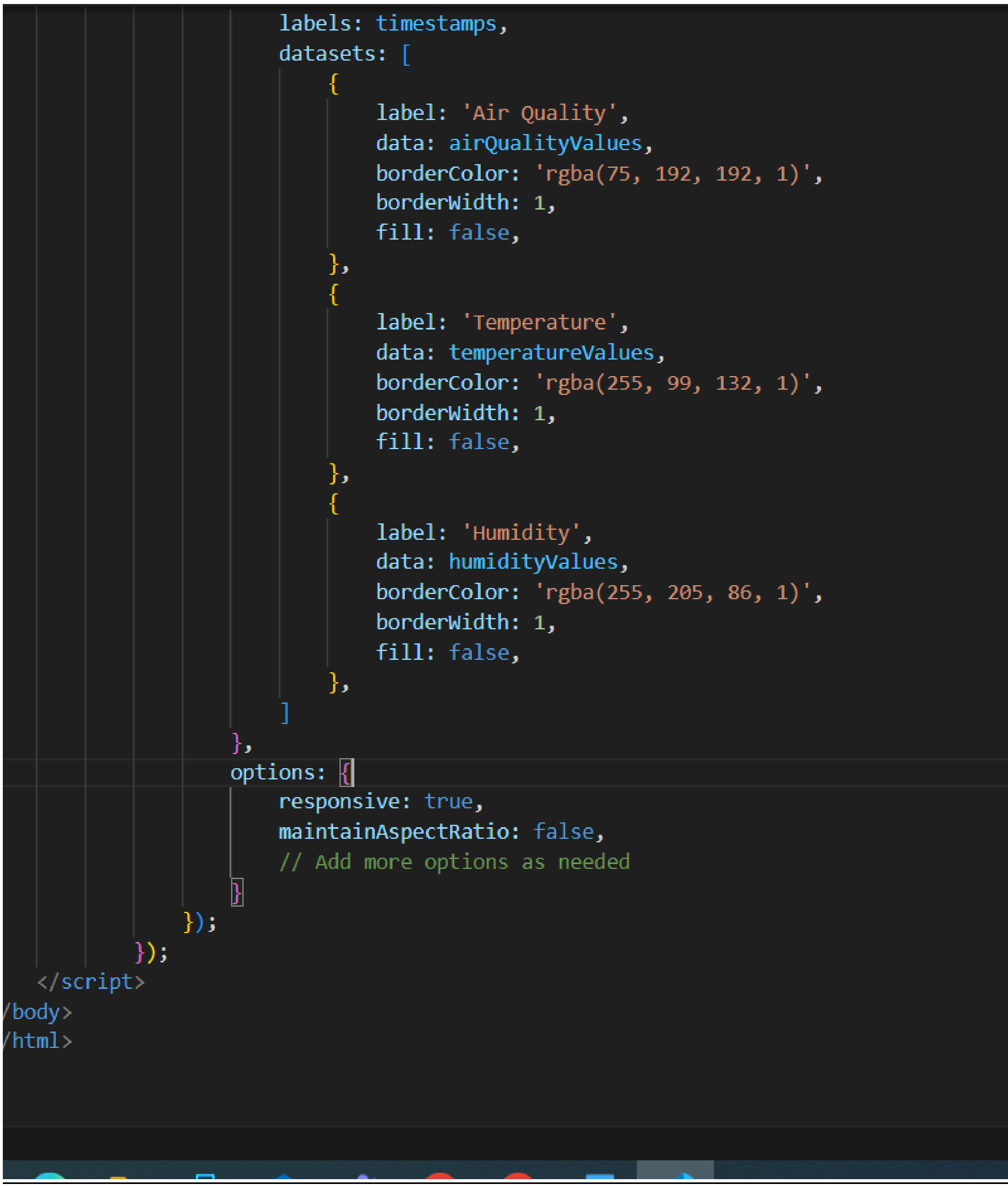
The functions getWeatherData and suggestOutfit, which are defined in this code, can be used to obtain weather information for a certain city and recommend a suitable outfit based on the temperature and weather details.

## 3. Environmental data chart feature:

This code essentially fetches environmental data from a specified API endpoint, processes the data, and dynamically generates a line chart using the Chart.js library. The chart displays air quality, temperature, and humidity values over time. The chart is rendered within an HTML canvas element on the web page. The resulting web page provides a visual representation of the environmental data for analysis and interpretation.

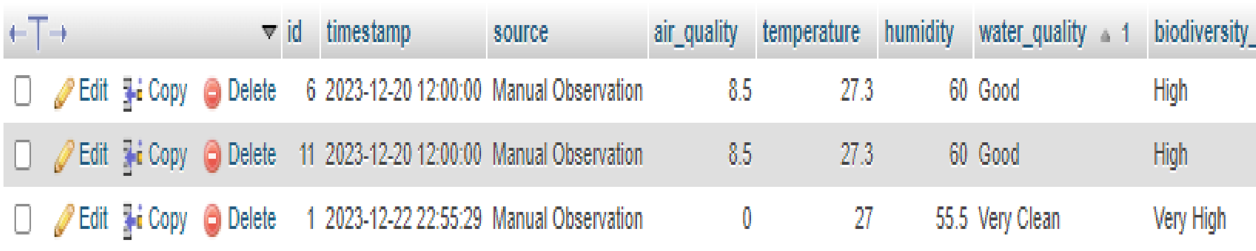


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Environmental Data Chart</title>
    <!-- Include Chart.js library -->
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
<body>
    <!-- HTML container for the chart -->
    <canvas id="myChart" width="400" height="200"></canvas>

    <script>
        // Your client-side JavaScript code here
        fetch('http://localhost:3000/api/environmental-data')
            .then(response => response.json())
            .then(data => {
                // Process the data
                const timestamps = data.map(entry => entry.timestamp);
                const airQualityValues = data.map(entry => entry.air_quality);
                const temperatureValues = data.map(entry => entry.temperature);
                const humidityValues = data.map(entry => entry.humidity);

                // Create a line chart using Chart.js
                const ctx = document.getElementById('myChart').getContext('2d');
                new Chart(ctx, {
                    type: 'line',
                    data: {
                        labels: timestamps,
                        datasets: [
                            {
                                label: 'Air Quality',
                                data: airQualityValues,
                                borderColor: 'rgba(75, 192, 192, 1)',
                                borderWidth: 1,
                                fill: false,
```



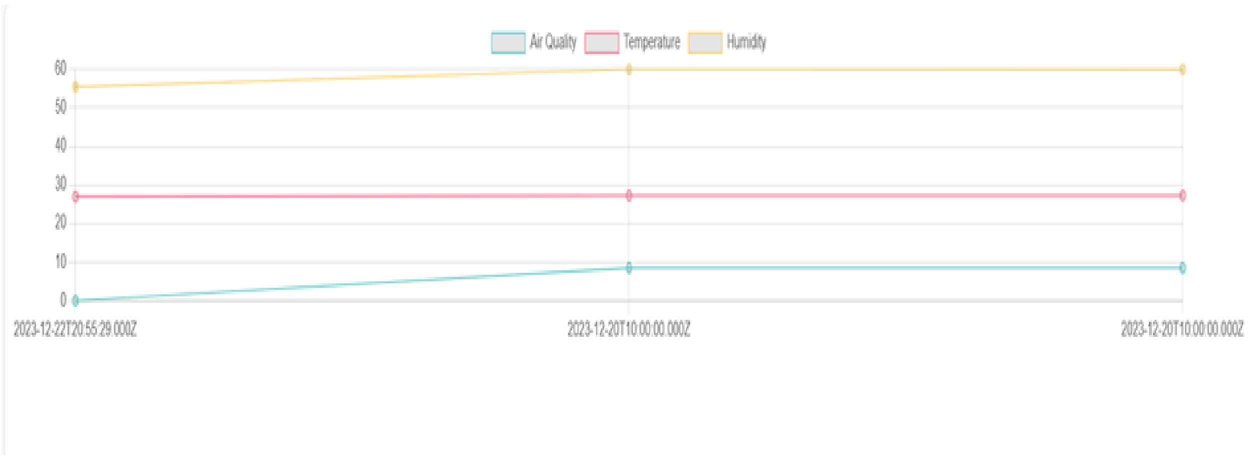
```
                labels: timestamps,
                datasets: [
                    {
                        label: 'Air Quality',
                        data: airQualityValues,
                        borderColor: 'rgba(75, 192, 192, 1)',
                        borderWidth: 1,
                        fill: false,
                    },
                    {
                        label: 'Temperature',
                        data: temperatureValues,
                        borderColor: 'rgba(255, 99, 132, 1)',
                        borderWidth: 1,
                        fill: false,
                    },
                    {
                        label: 'Humidity',
                        data: humidityValues,
                        borderColor: 'rgba(255, 205, 86, 1)',
                        borderWidth: 1,
                        fill: false,
                    },
                ]
            },
            options: {
                responsive: true,
                maintainAspectRatio: false,
                // Add more options as needed
            }
        });
    });
  </script>
/body>
/html>
```

Environmental data chart in database:

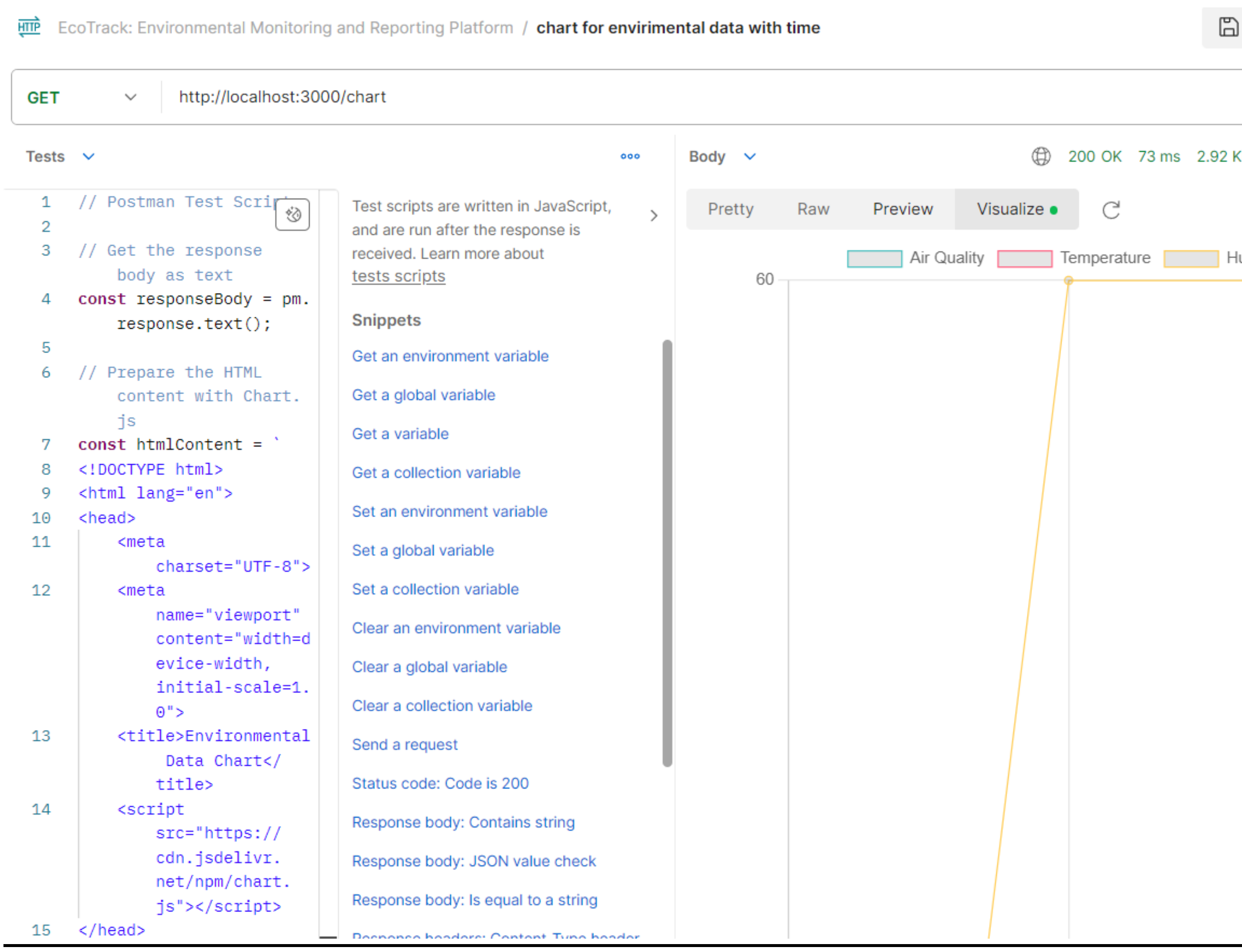


| | | | | id | timestamp | source | air_quality | temperature | humidity | water_quality | biodiversity_ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ☐ | Edit | Copy | Delete | 6 | 2023-12-20 12:00:00 | Manual Observation | 8.5 | 27.3 | 60 | Good | High |
| ☐ | Edit | Copy | Delete | 11 | 2023-12-20 12:00:00 | Manual Observation | 8.5 | 27.3 | 60 | Good | High |
| ☐ | Edit | Copy | Delete | 1 | 2023-12-22 22:55:29 | Manual Observation | 0 | 27 | 55.5 | Very Clean | Very High |

<u>Environmental data chart in web page:</u>



<u>Environmental data chart in postman:</u>



EcoTrack: Environmental Monitoring and Reporting Platform / **chart for envirimental data with time**

GET http://localhost:3000/chart

```
// Postman Test Scri

// Get the response
    body as text
const responseBody = pm.
    response.text();

// Prepare the HTML
    content with Chart.
    js
const htmlContent = `
<!DOCTYPE html>
<html lang="en">
<head>
    <meta
        charset="UTF-8">
    <meta
        name="viewport"
        content="width=d
        evice-width,
        initial-scale=1.
        0">
    <title>Environmental
         Data Chart</
        title>
    <script
        src="https://
        cdn.jsdelivr.
        net/npm/chart.
        js"></script>
</head>
```

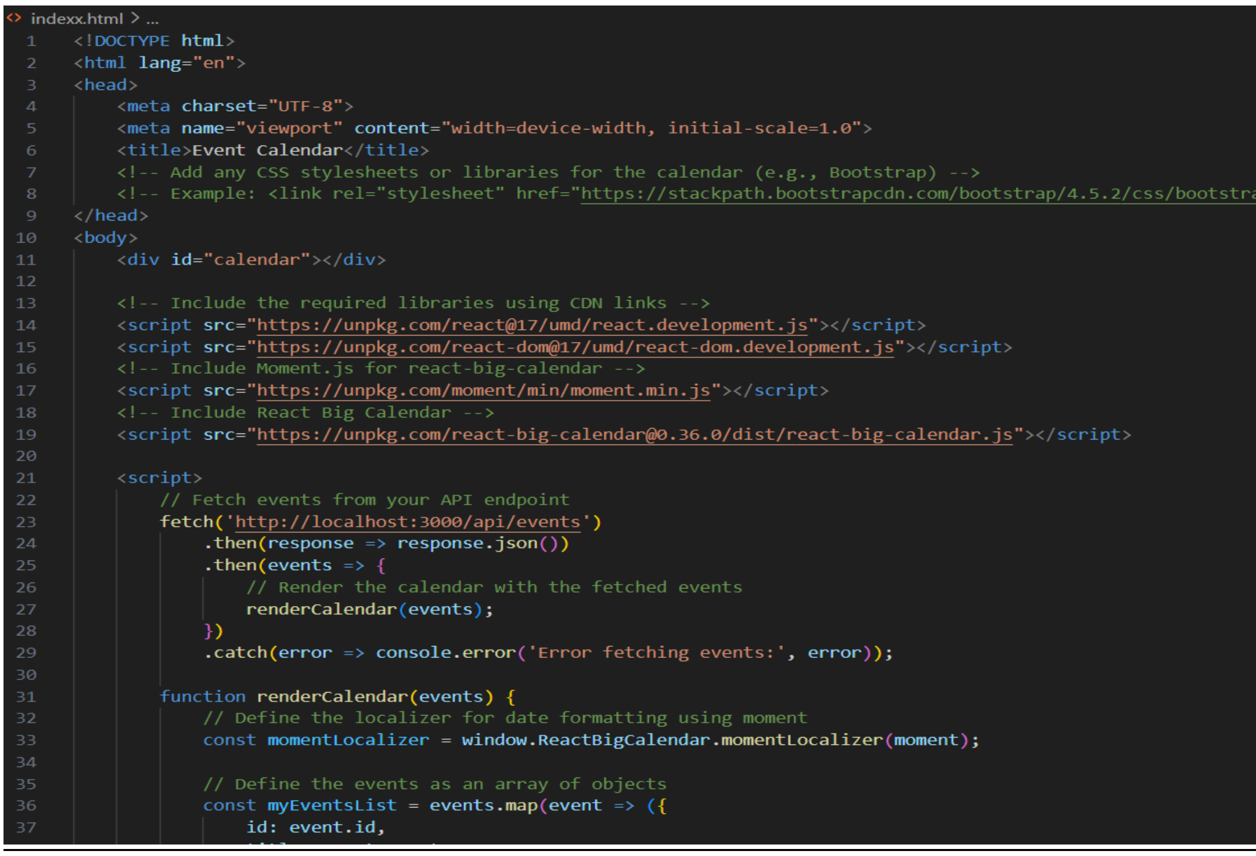
Test scripts are written in JavaScript, and are run after the response is received. Learn more about tests scripts

**Snippets**

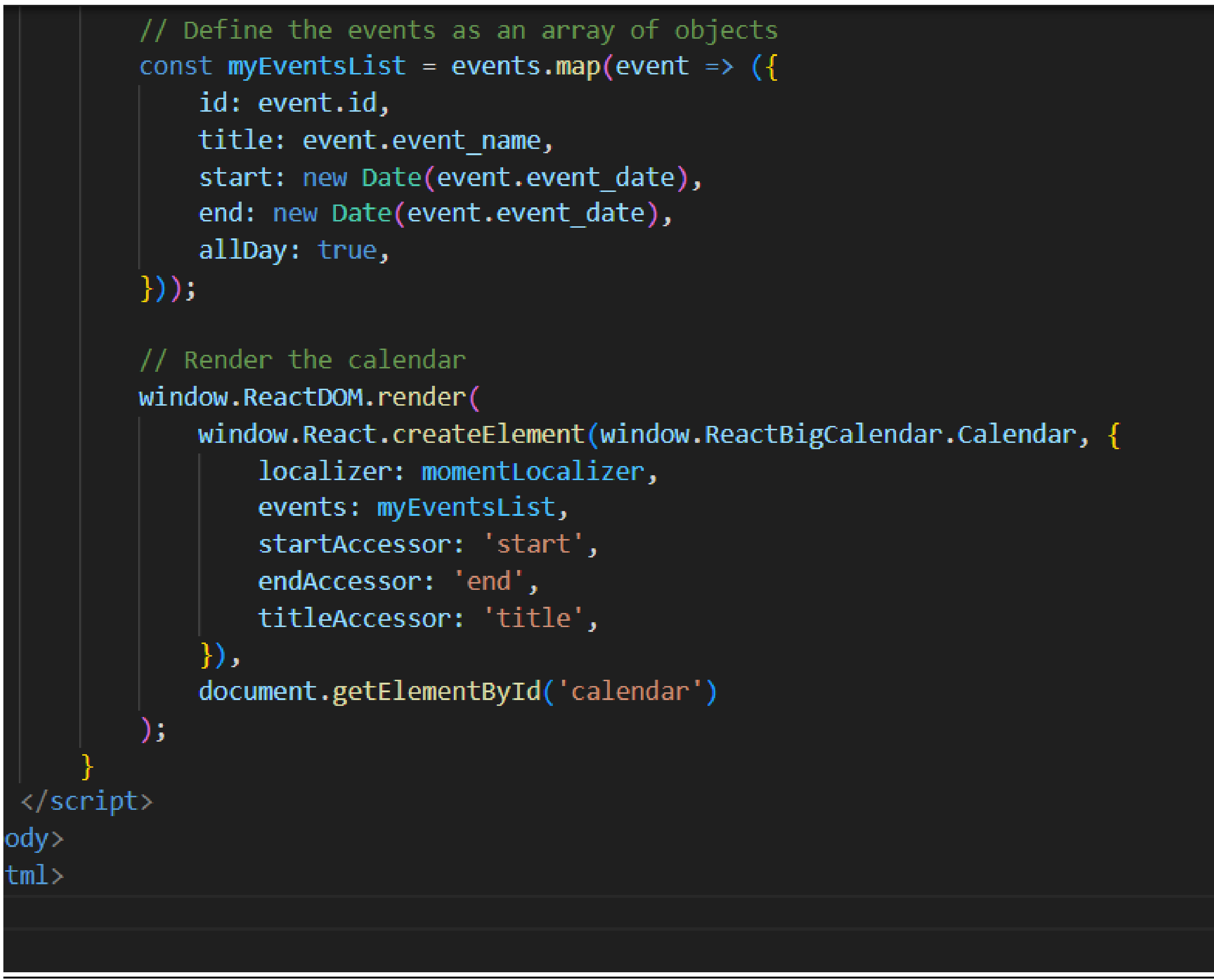
Get an environment variable
Get a global variable
Get a variable
Get a collection variable
Set an environment variable
Set a global variable
Set a collection variable
Clear an environment variable
Clear a global variable
Clear a collection variable
Send a request
Status code: Code is 200
Response body: Contains string
Response body: JSON value check
Response body: Is equal to a string

# 4. The calendar displays events with their titles and dates feature:

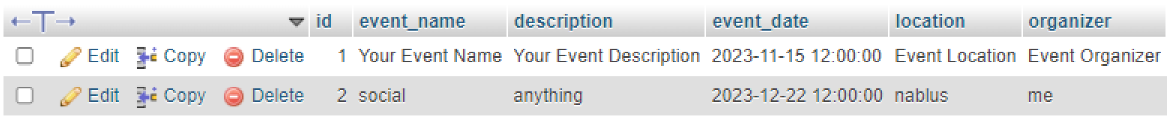
This code creates a dynamic event calendar web page by fetching event data from a specified API endpoint, formatting the data, and rendering it using React Big Calendar. The calendar displays events with their titles and dates, allowing users to visualize and interact with scheduled activities. The use of React and Moment.js enhances the modularity and date formatting capabilities of the calendar component. The resulting web page serves as a user-friendly interface for managing and viewing events in a calendar format.



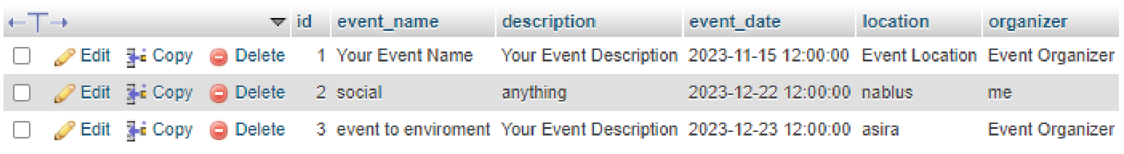
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Event Calendar</title>
    <!-- Add any CSS stylesheets or libraries for the calendar (e.g., Bootstrap) -->
    <!-- Example: <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstra
</head>
<body>
    <div id="calendar"></div>

    <!-- Include the required libraries using CDN links -->
    <script src="https://unpkg.com/react@17/umd/react.development.js"></script>
    <script src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
    <!-- Include Moment.js for react-big-calendar -->
    <script src="https://unpkg.com/moment/min/moment.min.js"></script>
    <!-- Include React Big Calendar -->
    <script src="https://unpkg.com/react-big-calendar@0.36.0/dist/react-big-calendar.js"></script>

    <script>
        // Fetch events from your API endpoint
        fetch('http://localhost:3000/api/events')
            .then(response => response.json())
            .then(events => {
                // Render the calendar with the fetched events
                renderCalendar(events);
            })
            .catch(error => console.error('Error fetching events:', error));

        function renderCalendar(events) {
            // Define the localizer for date formatting using moment
            const momentLocalizer = window.ReactBigCalendar.momentLocalizer(moment);

            // Define the events as an array of objects
            const myEventsList = events.map(event => ({
                id: event.id,
```



```
        // Define the events as an array of objects
        const myEventsList = events.map(event => ({
            id: event.id,
            title: event.event_name,
            start: new Date(event.event_date),
            end: new Date(event.event_date),
            allDay: true,
        }));

        // Render the calendar
        window.ReactDOM.render(
            window.React.createElement(window.ReactBigCalendar.Calendar, {
                localizer: momentLocalizer,
                events: myEventsList,
                startAccessor: 'start',
                endAccessor: 'end',
                titleAccessor: 'title',
            }),
            document.getElementById('calendar')
        );
    }
 </script>
ody>
tml>
```

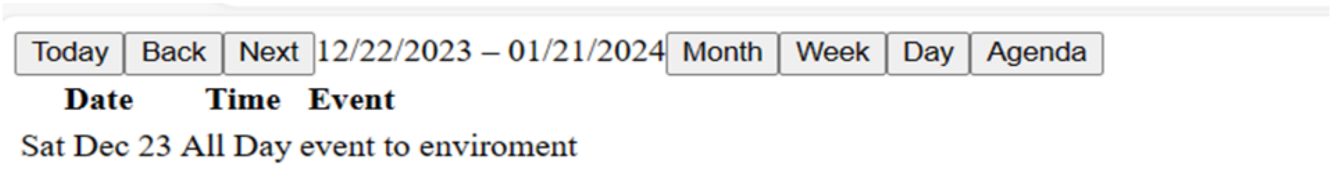


| | id | event_name | description | event_date | location | organizer |
|---|---|---|---|---|---|---|
| Edit Copy Delete | 1 | Your Event Name | Your Event Description | 2023-11-15 12:00:00 | Event Location | Event Organizer |
| Edit Copy Delete | 2 | social | anything | 2023-12-22 12:00:00 | nablus | me |



| | id | event_name | description | event_date | location | organizer |
|---|---|---|---|---|---|---|
| Edit Copy Delete | 1 | Your Event Name | Your Event Description | 2023-11-15 12:00:00 | Event Location | Event Organizer |
| Edit Copy Delete | 2 | social | anything | 2023-12-22 12:00:00 | nablus | me |
| Edit Copy Delete | 3 | event to enviroment | Your Event Description | 2023-12-23 12:00:00 | asira | Event Organizer |

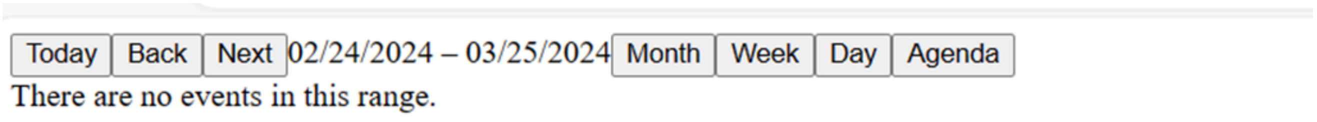
It show events only dated more recently than today in the range:



Today Back Next 12/22/2023 – 01/21/2024 Month Week Day Agenda

| Date | Time | Event |
|---|---|---|
| Sat Dec 23 | All Day | event to enviroment |

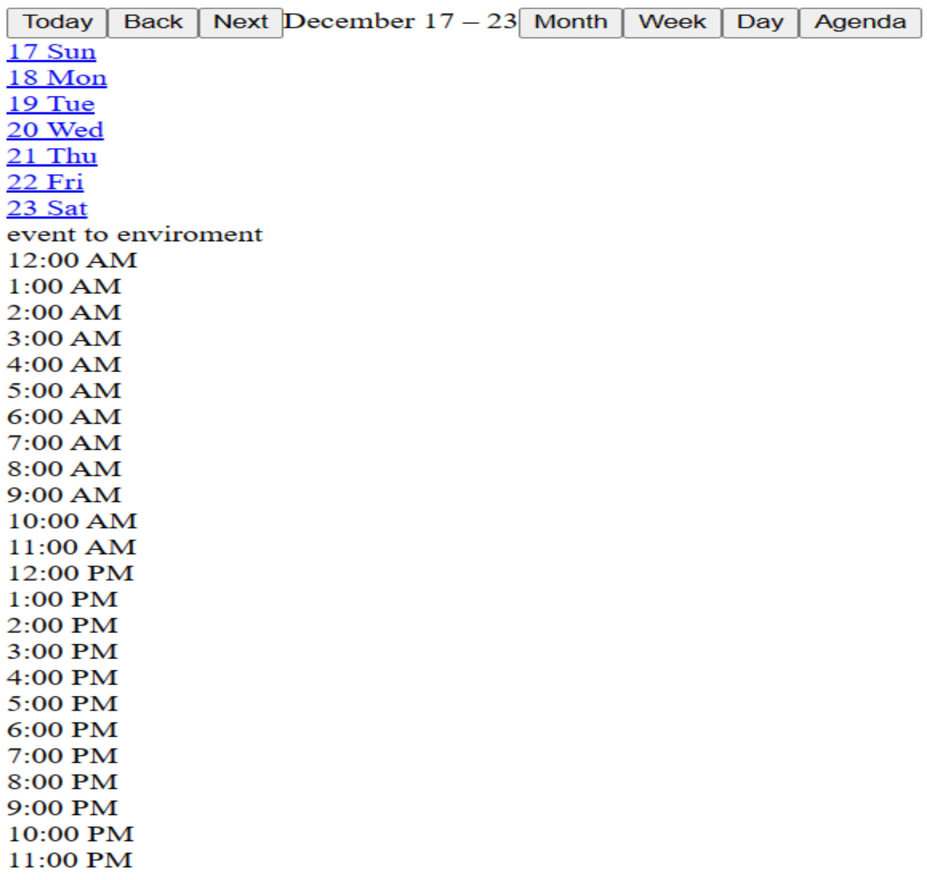
No events in this range:



Today Back Next 02/24/2024 – 03/25/2024 Month Week Day Agenda

There are no events in this range.

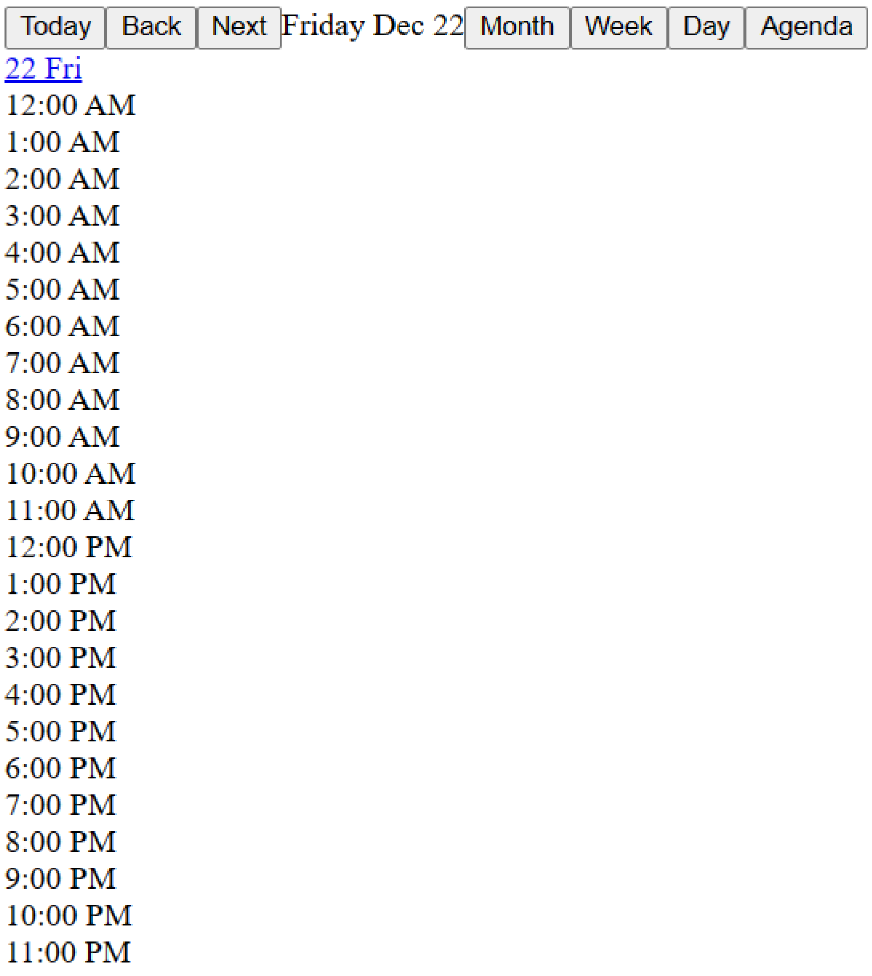
I have event in 23:



Today Back Next December 17 – 23 Month Week Day Agenda

17 Sun
18 Mon
19 Tue
20 Wed
21 Thu
22 Fri
23 Sat
event to enviroment
12:00 AM
1:00 AM
2:00 AM
3:00 AM
4:00 AM
5:00 AM
6:00 AM
7:00 AM
8:00 AM
9:00 AM
10:00 AM
11:00 AM
12:00 PM
1:00 PM
2:00 PM
3:00 PM
4:00 PM
5:00 PM
6:00 PM
7:00 PM
8:00 PM
9:00 PM
10:00 PM
11:00 PM

Button Day:



Friday Dec 22

22 Fri

12:00 AM
1:00 AM
2:00 AM
3:00 AM
4:00 AM
5:00 AM
6:00 AM
7:00 AM
8:00 AM
9:00 AM
10:00 AM
11:00 AM
12:00 PM
1:00 PM
2:00 PM
3:00 PM
4:00 PM
5:00 PM
6:00 PM
7:00 PM
8:00 PM
9:00 PM
10:00 PM
11:00 PM

Button week:

Today Back Next December 2023 Month Week Day Agenda

Sun
Mon
Tue
Wed
Thu
Fri
Sat
26
27
28
29
30
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
event to enviroment