Practical 5

Problem 1: Emit an event "greet" and listen to it.

Code:

event_emitter_example.js | index.js U

Practical 5 > event_emitter_example.js > ...

```
// 1. Import the EventEmitter class from the 'events' module
const EventEmitter = require('events');

// 2. Create an instance of EventEmitter
const myEmitter = new EventEmitter();

// 3. Define the event name
const eventName = 'greet';

// 4. Set up a listener using the .on() method
// This function will execute whenever the 'greet' event is emitted.
myEmitter.on(eventName, (name) => {
  console.log(`Hello, ${name}! The '${eventName}' event was received.`);
});

// 5. Emit the 'greet' event using the .emit() method
console.log("Emitting the 'greet' event...");
myEmitter.emit(eventName, 'NodeJS'); // The second argument 'NodeJS' is passed to the listener function

console.log("Script finished.");
```

Output:

```
ravithakur@Ravis-MacBook-Air Node-JS-Practical-GF2023485999 % node "/Users/ravithakur/Documents/GitHub/Noc
ter_example.js"
Emitting the 'greet' event...
Hello, NodeJS! The 'greet' event was received.
Script finished.
ravithakur@Ravis-MacBook-Air Node-JS-Practical-GF2023485999 %
```

Problem 2: Connect NodeJS to MongoDB and insert a user record.

Code:

Practical 5 > JS index.js > ...

```
const mongoose = require('mongoose');
const dbURI = 'mongodb://127.0.0.1:27017/mydatabase';
const userSchema = new mongoose.Schema({
    name: {
        type: String,
        required: true
    },
    email: {
        type: String,
        required: true,
        unique: true
    },
    age: Number,
    createdAt: {
        type: Date,
        default: Date.now
    }
});
const User = mongoose.model('User', userSchema);
async function main() {
    try {
        await mongoose.connect(dbURI);
        console.log('✅ Successfully connected to MongoDB!');

        // Create a new user instance
        const newUser = new User({
            name: 'Ravi Kumar',
            email: 'ravi.thakur32212@gmail.com',
            age: 30
        });

        // Save the new user to the database
        const savedUser = await newUser.save();
        console.log('💾 User saved successfully:', savedUser);

    } catch (error) {
        console.error('❌ Error:', error.message);
    } finally {
        await mongoose.disconnect();
        console.log('🔌 Disconnected from MongoDB.');
    }
}

main();
```

Ln 1, Col 1 Spaces: 4 UTF-8 L

Output:

```
✅ Successfully connected to MongoDB!
💾 User saved successfully: {
  name: 'Ravi Thakur',
  email: 'ravi.thakur32212@gmail.com',
  age: 30,
  _id: new ObjectId("..."),
  createdAt: 2025-09-23T05:02:37.000Z,
  __v: 0
}
🔌 Disconnected from MongoDB.
```