Practical 4

**Project:** **File Organizer Tool**  
  
**Objective**: Build a CLI tool to organize files into folders based on their type (e.g., images, documents, videos).

**Tasks**:

* Accept a directory path as an input from the user.
* Use the fs module to read all files in the directory.
* Move files into folders like Images, Documents, and Others based on their extensions.
* Log the operations performed into a summary.txt file.

**Building a Simple RESTful API with Core HTTP Module**

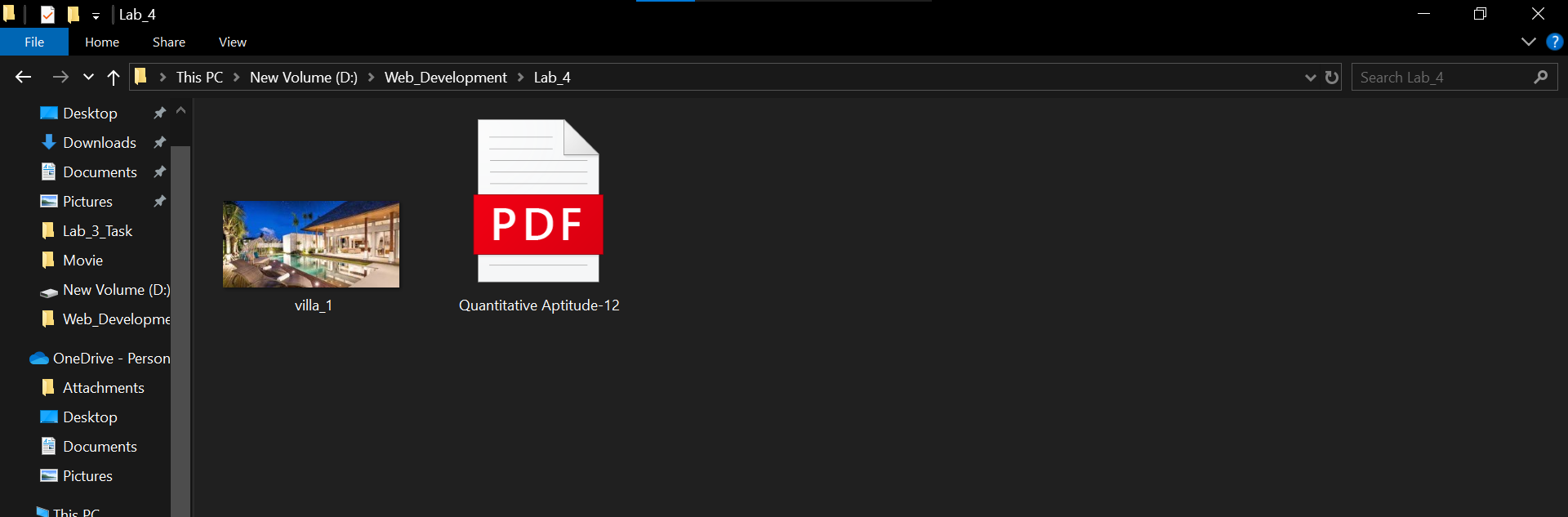
**Screenshots of Code:**

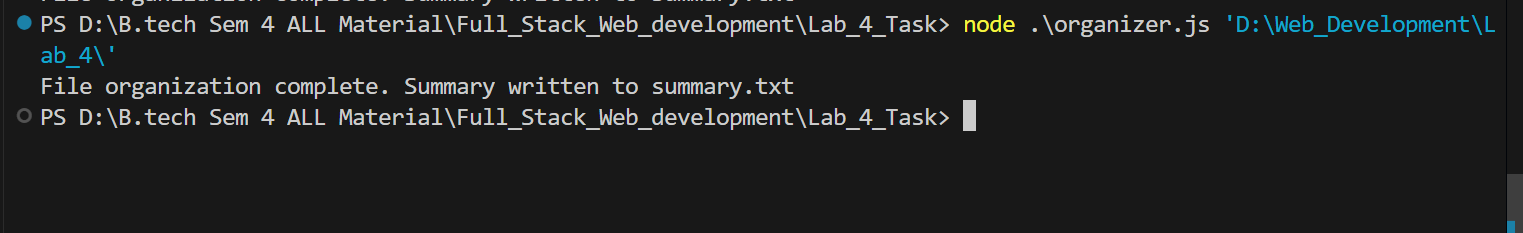




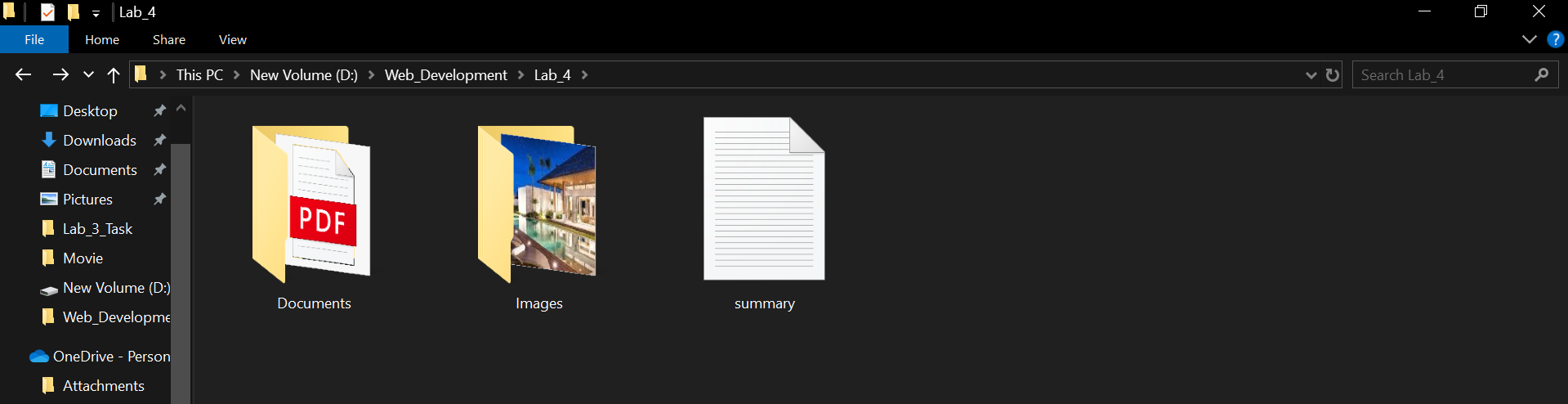
**Screenshots of Output:**

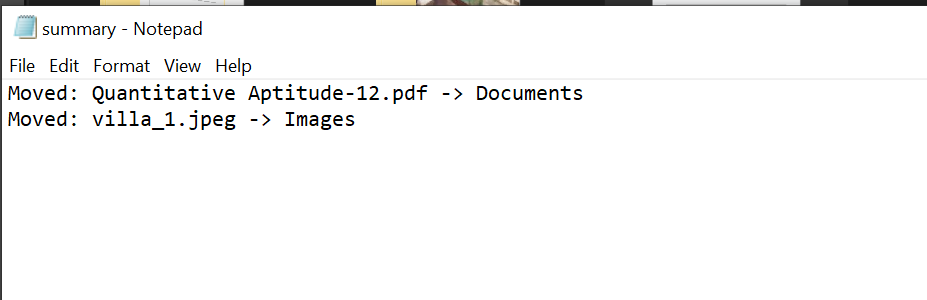
**When code is not running then Document and image format**

****

****

**When code is running then Document and image format**

****

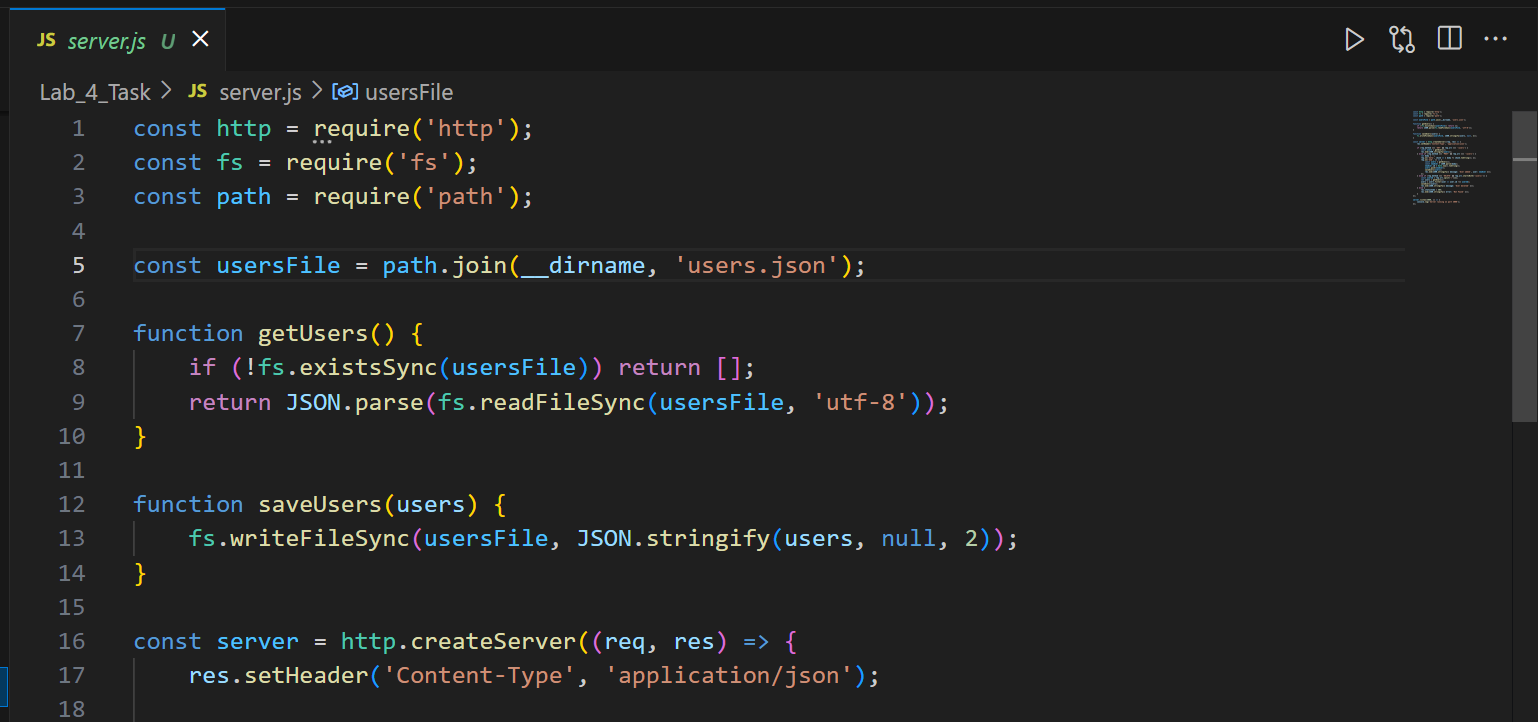
****

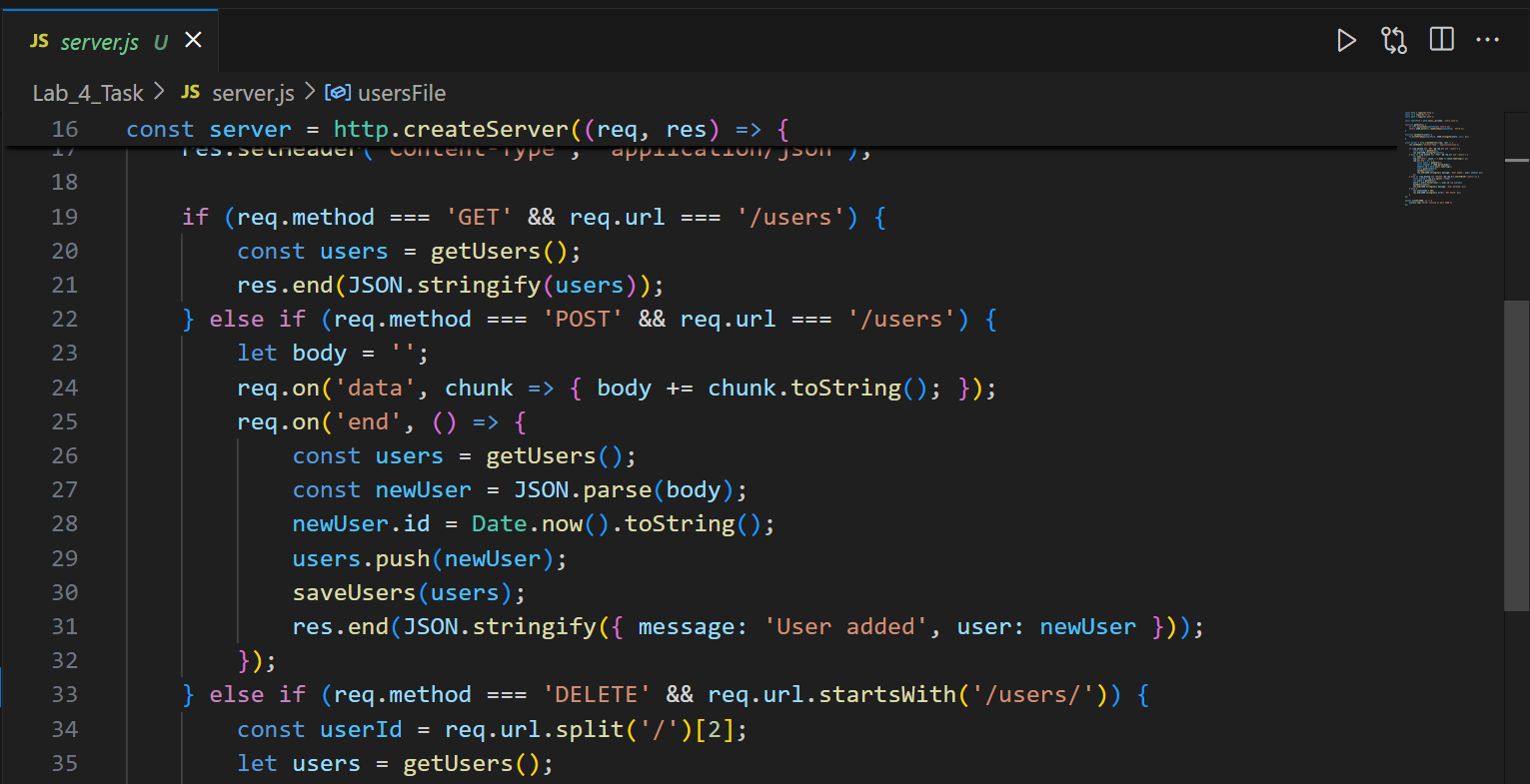
**And summary.txt file created**

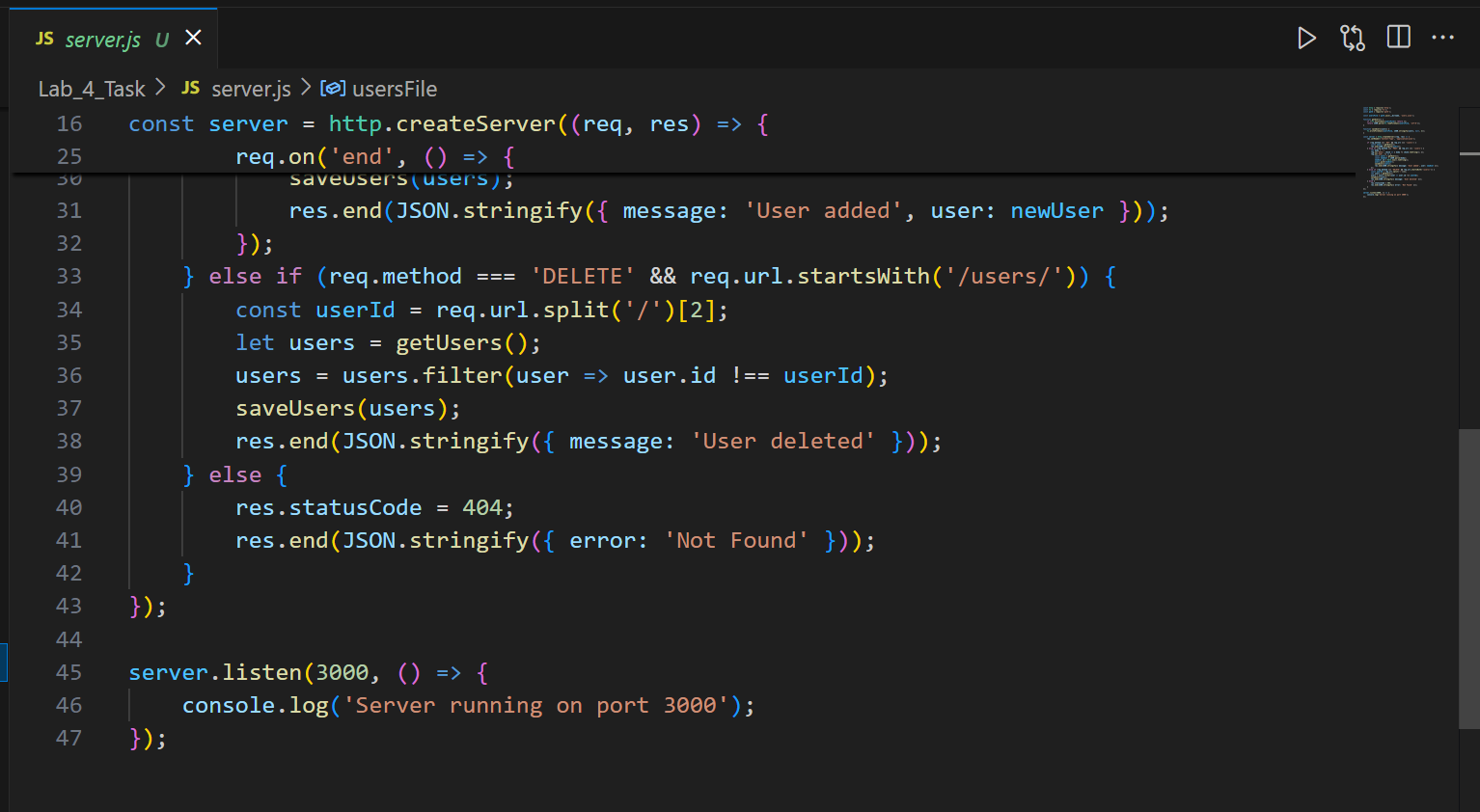
**Project:** User Management System  
  
**Objective:** Create a RESTful API to manage user data without using Express.js.  
  
Tasks:

* Implement the following endpoints using the http module:
* GET /users: Return a list of all users stored in a JSON file.
* POST /users: Accept new user data in the request body and add it to the JSON file.
* DELETE /users/:id: Remove a user by their ID from the JSON file.
* Use the fs module to store and retrieve user data persistently.
* Test the API using Postman or curl.

**Screenshots of Code:**

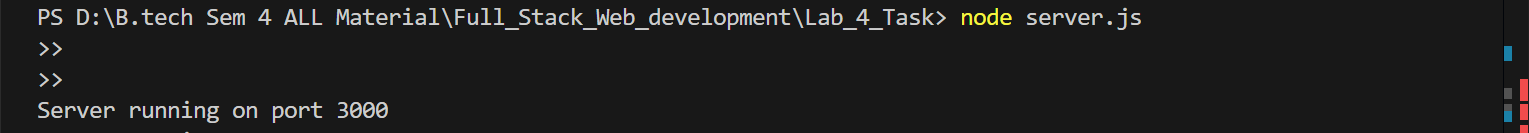


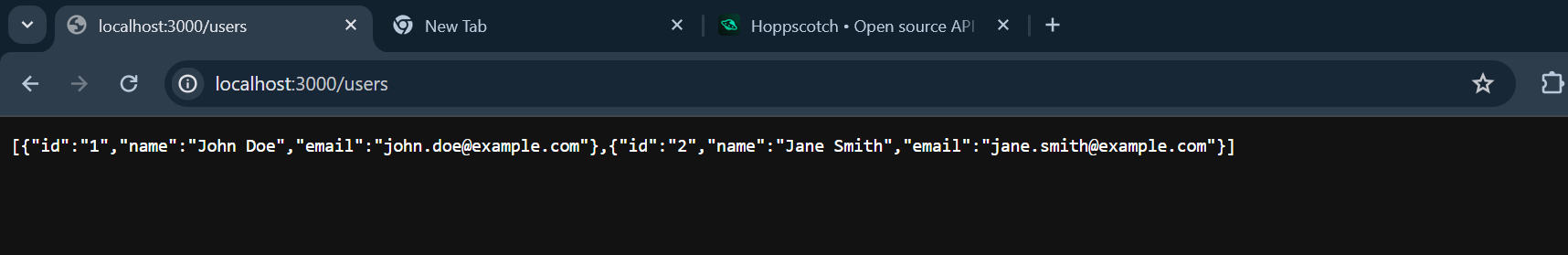






**Screenshots of Output:**





**Conclusion**:

The **User Management System** successfully implements a lightweight RESTful API using Node.js without external frameworks like Express.js. By utilizing the http and fs modules, it efficiently handles user data storage, retrieval, and deletion through a JSON file. This project demonstrates core concepts of API development, including request handling and persistent data management. It highlights how basic Node.js functionality can be leveraged to build functional web services. Future improvements could include authentication, data validation, and database integration for enhanced scalability and security.