

DAY 2 ESD ASSIGNMENT

1.

Node JS - File Cleaner:

Read a file, remove all the extra spaces and write it back to the same file.

For example, if the file input was

'''

hello world my name is Raman

'''

After the program runs, the output should be

'''

hello world my name is Raman

'''

```
// Import the 'fs' module for file system operations
const fs = require('fs');
```

```
// Define the name of the file to clean up
const filename = 'input.txt';
```

```
// Read the file
fs.readFile(filename, 'utf8', (err, data) => {
  if (err) {
    console.error(`Error reading file: ${err}`);
    return;
  }
}
```

```
// Remove all extra spaces
const cleanedData = data.replace(/ +/g, '');
```

```
// Write the cleaned data back to the file
fs.writeFile(filename, cleanedData, err => {
  if (err) {
    console.error(`Error writing file: ${err}`);
  } else {
    console.log(`Successfully cleaned up ${filename}`);
  }
});
```

```
});
```

1. Express JS get put delete post -> Todo like kidneys

```
// Import the Express.js framework
const express = require('express');

// Create a new Express app
const app = express();

// Enable JSON parsing for request bodies
app.use(express.json());

// Sample Todo data (in a real app, you'd use a database)
let todos = [
  { id: 1, title: 'Buy milk', completed: false },
  { id: 2, title: 'Walk the dog', completed: true },
  { id: 3, title: 'Do laundry', completed: false },
];

// GET /todos - Get all todos
app.get('/todos', (req, res) => {
  res.json(todos);
});

// GET /todos/:id - Get a single todo by ID
app.get('/todos/:id', (req, res) => {
  const id = parseInt(req.params.id);
  const todo = todos.find(todo => todo.id === id);
  if (!todo) {
    res.status(404).json({ error: 'Todo not found' });
  } else {
    res.json(todo);
  }
});

// POST /todos - Create a new todo
app.post('/todos', (req, res) => {
  const newTodo = { id: todos.length + 1, title: req.body.title, completed: false };
  todos.push(newTodo);
  res.json(newTodo);
});

// PUT /todos/:id - Update a todo
app.put('/todos/:id', (req, res) => {
  const id = parseInt(req.params.id);
  const todo = todos.find(todo => todo.id === id);
```

```

    if (!todo) {
      res.status(404).json({ error: 'Todo not found' });
    } else {
      todo.title = req.body.title;
      todo.completed = req.body.completed;
      res.json(todo);
    }
  });

// DELETE /todos/:id - Delete a todo
app.delete('/todos/:id', (req, res) => {
  const id = parseInt(req.params.id);
  const index = todos.findIndex(todo => todo.id === id);
  if (index === -1) {
    res.status(404).json({ error: 'Todo not found' });
  } else {
    todos.splice(index, 1);
    res.json({ message: 'Todo deleted' });
  }
});

// Start the server
const port = 3000;
app.listen(port, () => {
  console.log(`Server started on port ${port}`);
});

```