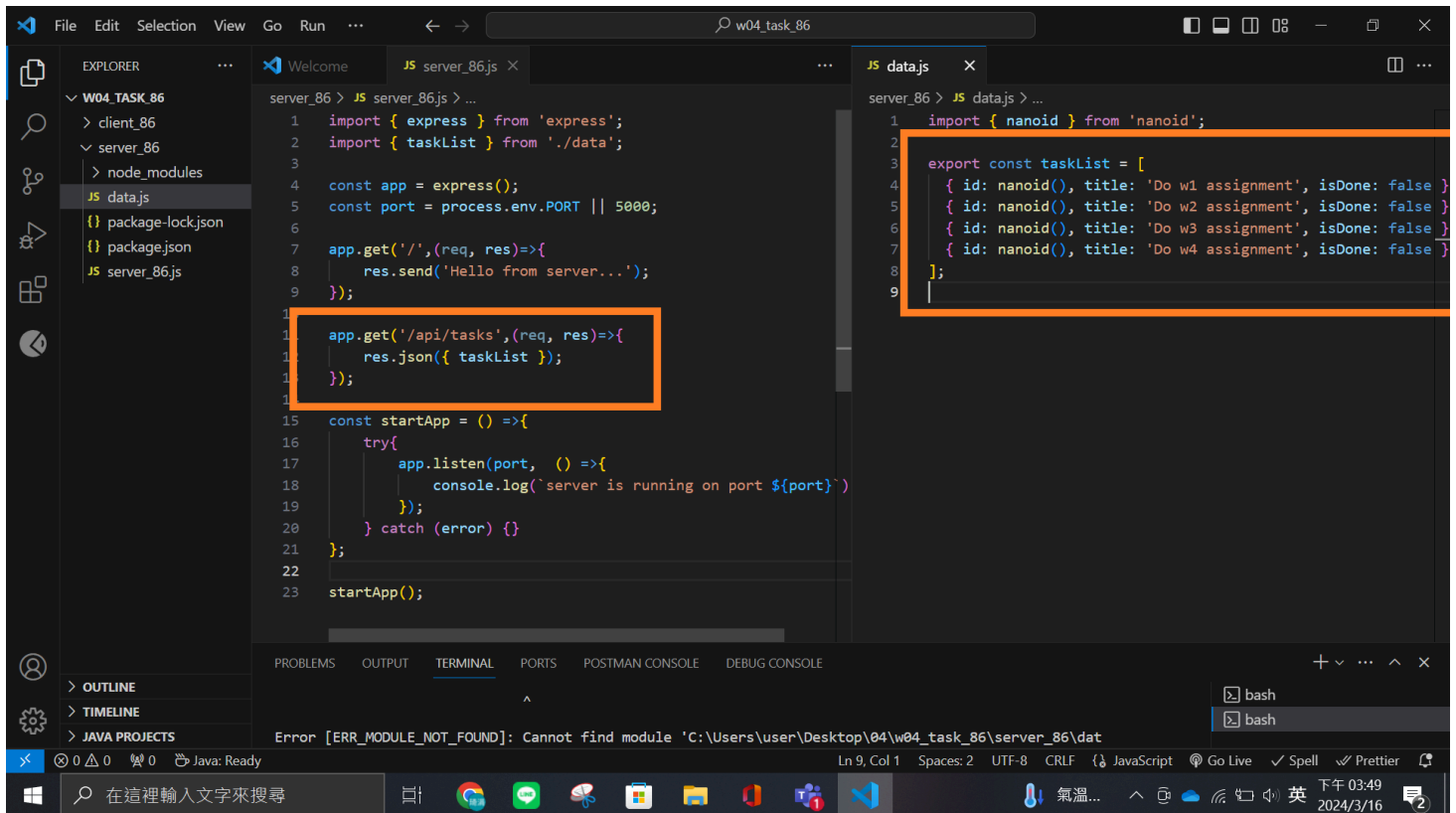


W4-P1: Server: implement GET /api/tasks



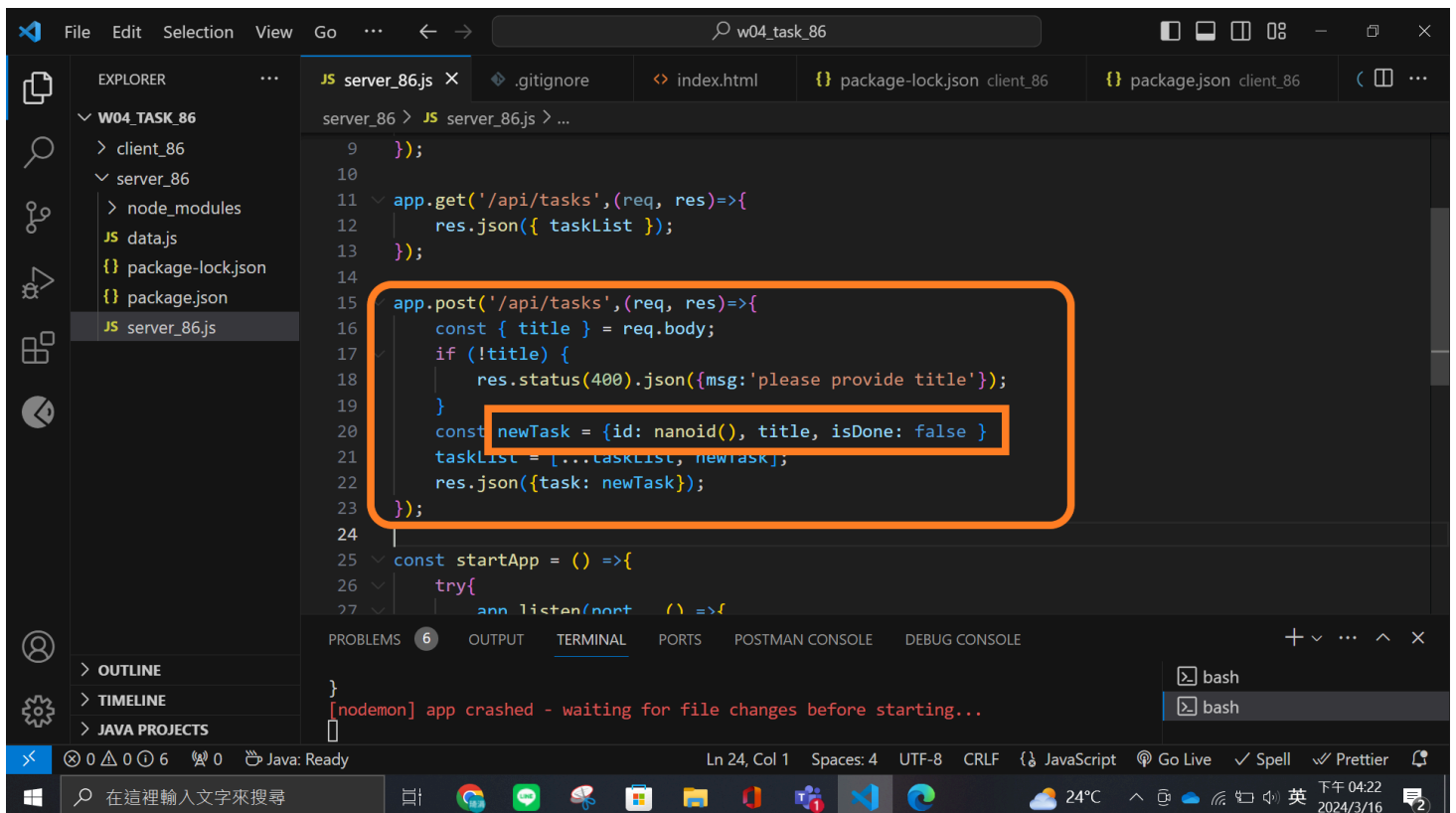
The screenshot shows the VS Code editor with two files open: `server_86.js` and `data.js`. The `server_86.js` file has a new route added for `GET /api/tasks`, which returns the `taskList` from `data.js`. The `data.js` file defines a `taskList` array with four tasks.

```
server_86 > JS server_86.js > ...
1 import { express } from 'express';
2 import { taskList } from './data';
3
4 const app = express();
5 const port = process.env.PORT || 5000;
6
7 app.get('/', (req, res) => {
8   res.send('Hello from server...');
9 });
10
11 app.get('/api/tasks', (req, res) => {
12   res.json({ taskList });
13 });
14
15 const startApp = () => {
16   try {
17     app.listen(port, () => {
18       console.log(`server is running on port ${port}`);
19     });
20   } catch (error) {}
21 }
22
23 startApp();
```

```
server_86 > JS data.js > ...
1 import { nanoid } from 'nanoid';
2
3 export const taskList = [
4   { id: nanoid(), title: 'Do w1 assignment', isDone: false },
5   { id: nanoid(), title: 'Do w2 assignment', isDone: false },
6   { id: nanoid(), title: 'Do w3 assignment', isDone: false },
7   { id: nanoid(), title: 'Do w4 assignment', isDone: false },
8 ];
9
```

Terminal output: Error [ERR_MODULE_NOT_FOUND]: Cannot find module 'C:\Users\user\Desktop\04\w04_task_86\server_86\data

W4-P2: Server: implement POST /api/tasks

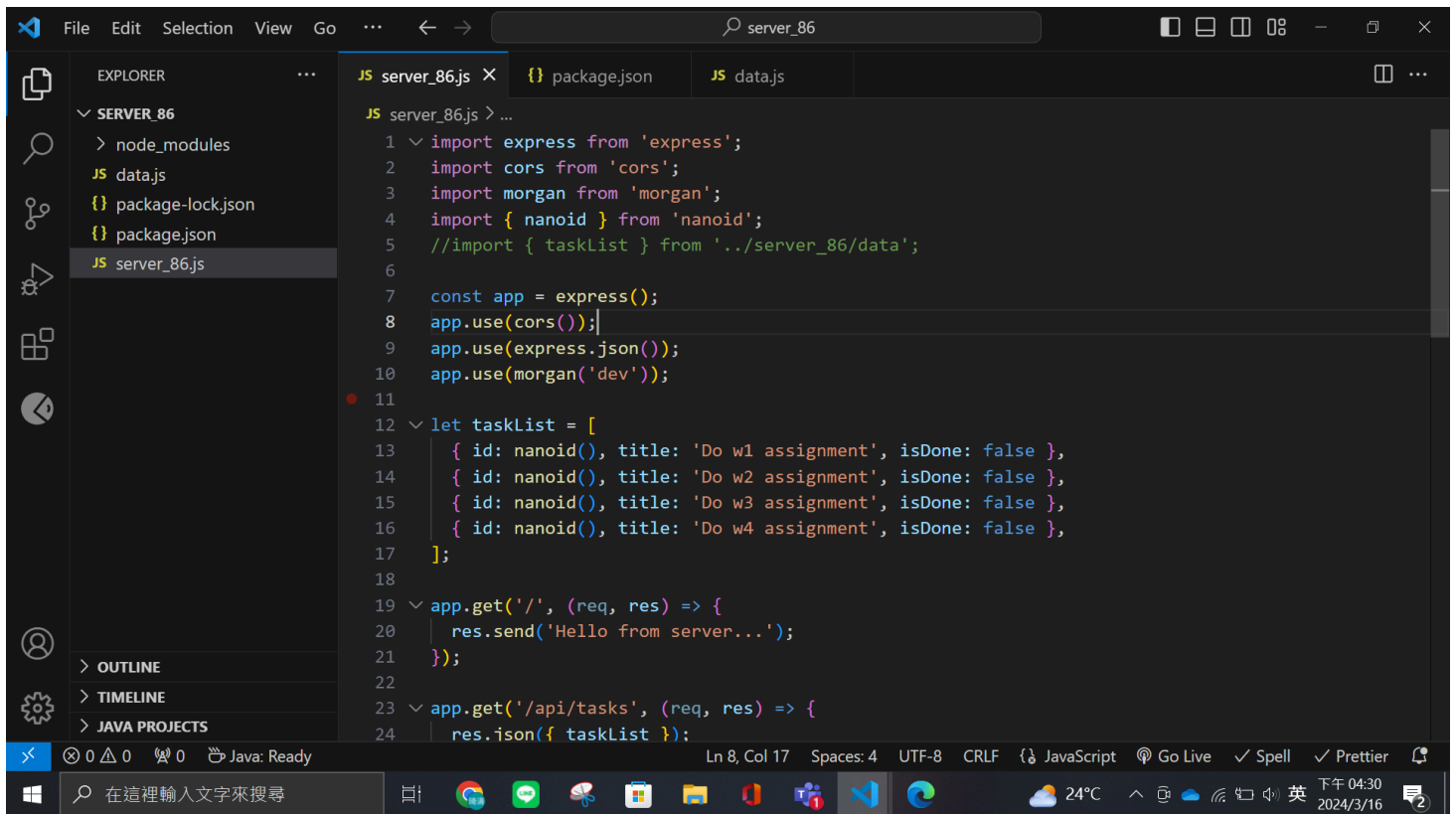


The screenshot shows the VS Code editor with the `server_86.js` file open. A new route is added for `POST /api/tasks`, which creates a new task and adds it to the `taskList` array. The `data.js` file defines a `taskList` array with four tasks.

```
server_86 > JS server_86.js > ...
9 });
10
11 app.get('/api/tasks', (req, res) => {
12   res.json({ taskList });
13 });
14
15 app.post('/api/tasks', (req, res) => {
16   const { title } = req.body;
17   if (!title) {
18     res.status(400).json({ msg: 'please provide title' });
19   }
20   const newTask = { id: nanoid(), title, isDone: false };
21   taskList = [...taskList, newTask];
22   res.json({ task: newTask });
23 });
24
25 const startApp = () => {
26   try {
27     app.listen(port, () => {
28       console.log(`server is running on port ${port}`);
29     });
30   } catch (error) {}
31 }
32
33 startApp();
```

Terminal output: [nodemon] app crashed - waiting for file changes before starting...

W4-P3: Server: implement PATCH /api/tasks/:id



The screenshot shows the Visual Studio Code editor with the file `server_86.js` open. The Explorer sidebar on the left shows the project structure with `server_86.js` selected. The main editor area displays the following JavaScript code:

```
JS server_86.js > ...
1  import express from 'express';
2  import cors from 'cors';
3  import morgan from 'morgan';
4  import { nanoid } from 'nanoid';
5  //import { taskList } from '../server_86/data';
6
7  const app = express();
8  app.use(cors());
9  app.use(express.json());
10 app.use(morgan('dev'));
11
12 let taskList = [
13   { id: nanoid(), title: 'Do w1 assignment', isDone: false },
14   { id: nanoid(), title: 'Do w2 assignment', isDone: false },
15   { id: nanoid(), title: 'Do w3 assignment', isDone: false },
16   { id: nanoid(), title: 'Do w4 assignment', isDone: false },
17 ];
18
19 app.get('/', (req, res) => {
20   res.send('Hello from server...');
21 });
22
23 app.get('/api/tasks', (req, res) => {
24   res.json({ taskList });
```

The status bar at the bottom indicates the current cursor position is Line 8, Column 17, with 4 spaces. It also shows the file encoding as UTF-8 and line endings as CRLF. The system tray at the bottom right shows the date and time as 2024/3/16 at 04:30.