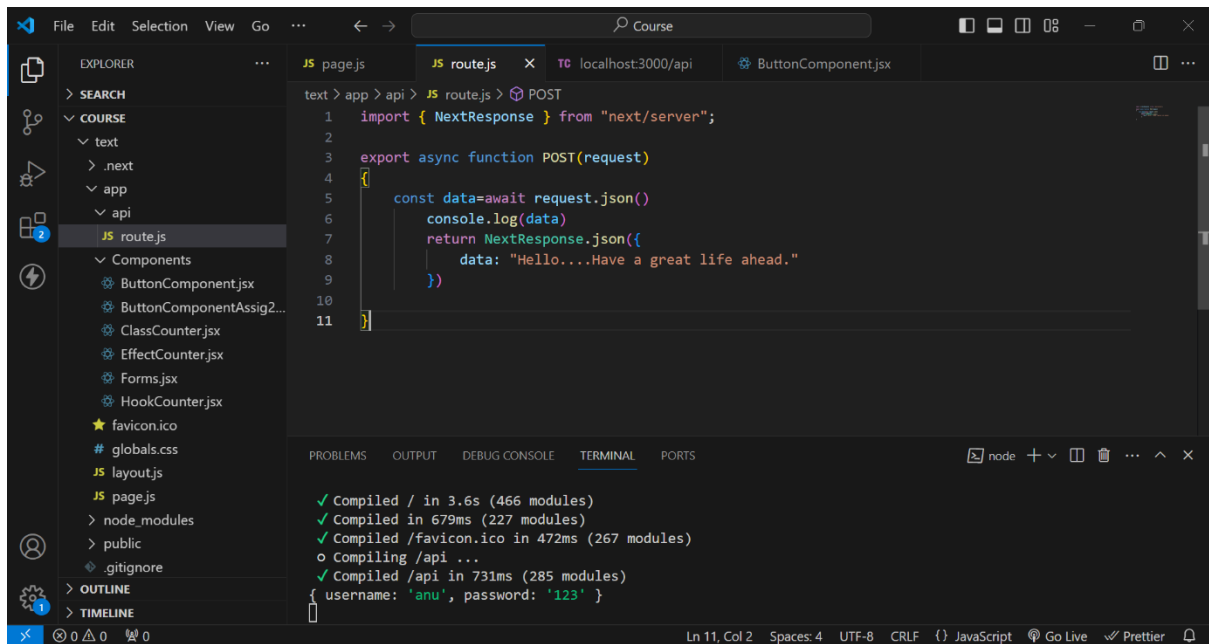


ASSIGNMENT 3- A simple RESTFUL API using React.js, DataBase Creation, and Creating Index

RestFul API code



The screenshot shows the VS Code editor with the `route.js` file open. The code defines a POST endpoint that returns a JSON response. The terminal at the bottom shows the output of the development server.

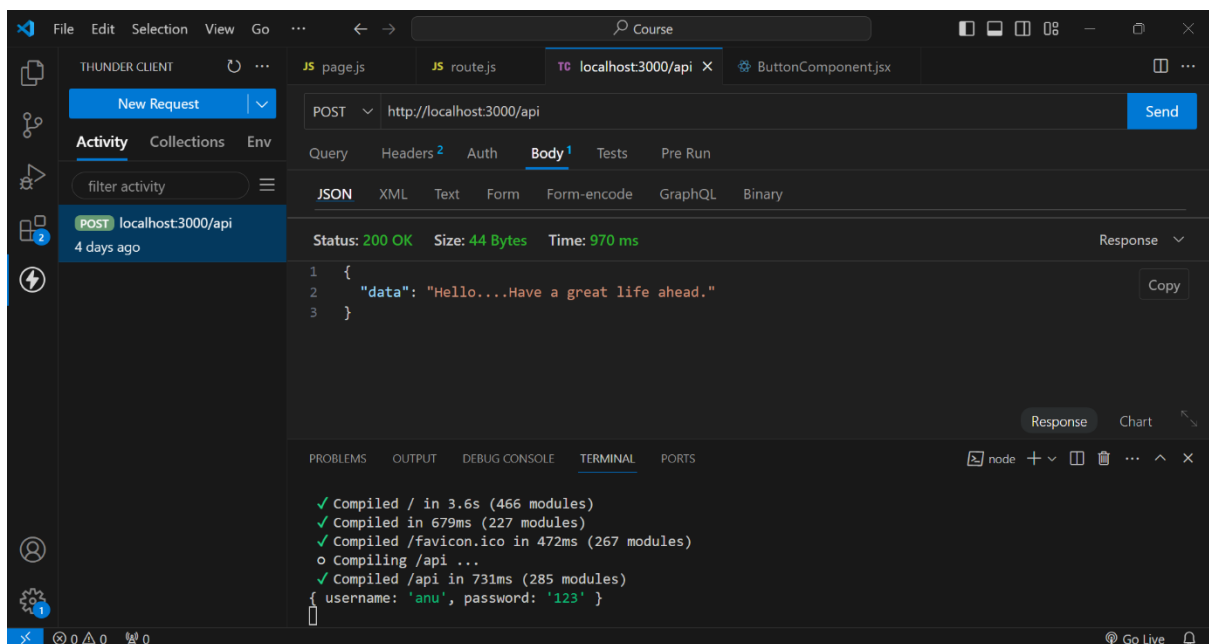
```
1 import { NextResponse } from "next/server";
2
3 export async function POST(request)
4 {
5   const data=await request.json()
6   console.log(data)
7   return NextResponse.json({
8     data: "Hello....Have a great life ahead."
9   })
10 }
11 }
```

Terminal Output:

```
✓ Compiled / in 3.6s (466 modules)
✓ Compiled in 679ms (227 modules)
✓ Compiled /favicon.ico in 472ms (267 modules)
○ Compiling /api ...
✓ Compiled /api in 731ms (285 modules)
{ username: 'anu', password: '123' }
```

Rest Output on ThunderClient

This is the output when we create an API using REST.



The screenshot shows the ThunderClient interface with a POST request to `http://localhost:3000/api`. The response is a JSON object with a message.

Request: POST `http://localhost:3000/api`

Response:

```
1 {
2   "data": "Hello....Have a great life ahead."
3 }
```

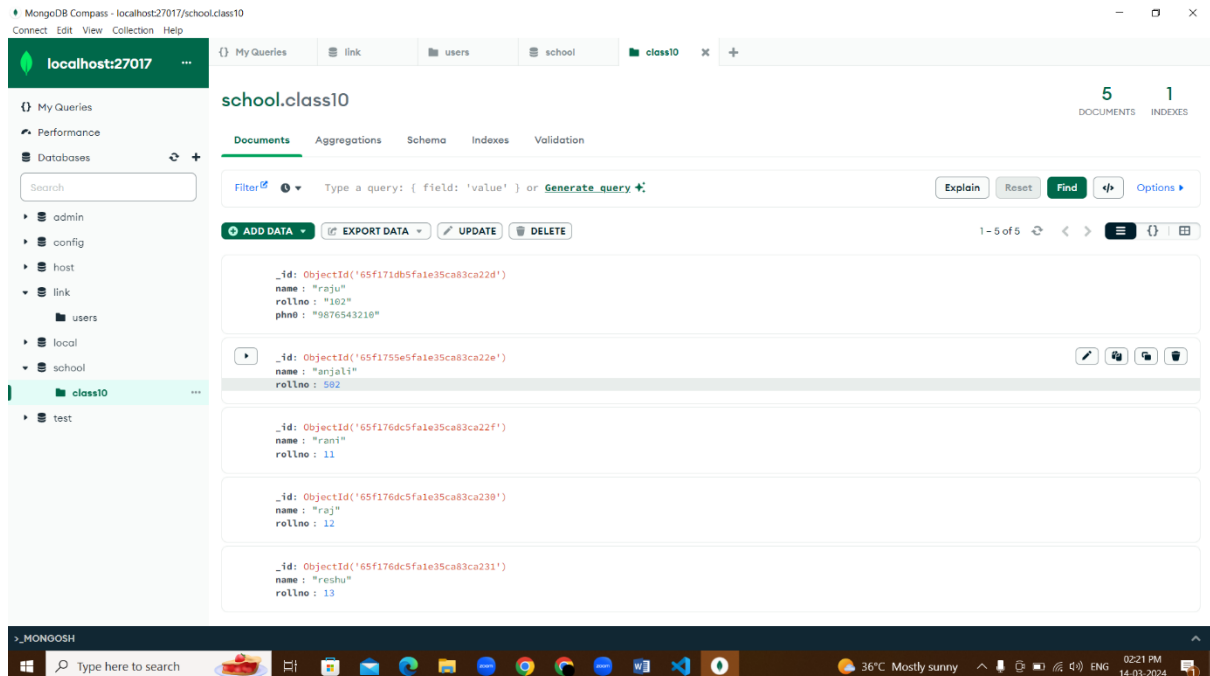
Terminal Output (repeated from the previous screenshot):

```
✓ Compiled / in 3.6s (466 modules)
✓ Compiled in 679ms (227 modules)
✓ Compiled /favicon.ico in 472ms (267 modules)
○ Compiling /api ...
✓ Compiled /api in 731ms (285 modules)
{ username: 'anu', password: '123' }
```

DataBase Creation

1st we created a database named school

In that test we created a collection named class10.



Indexing Creation

