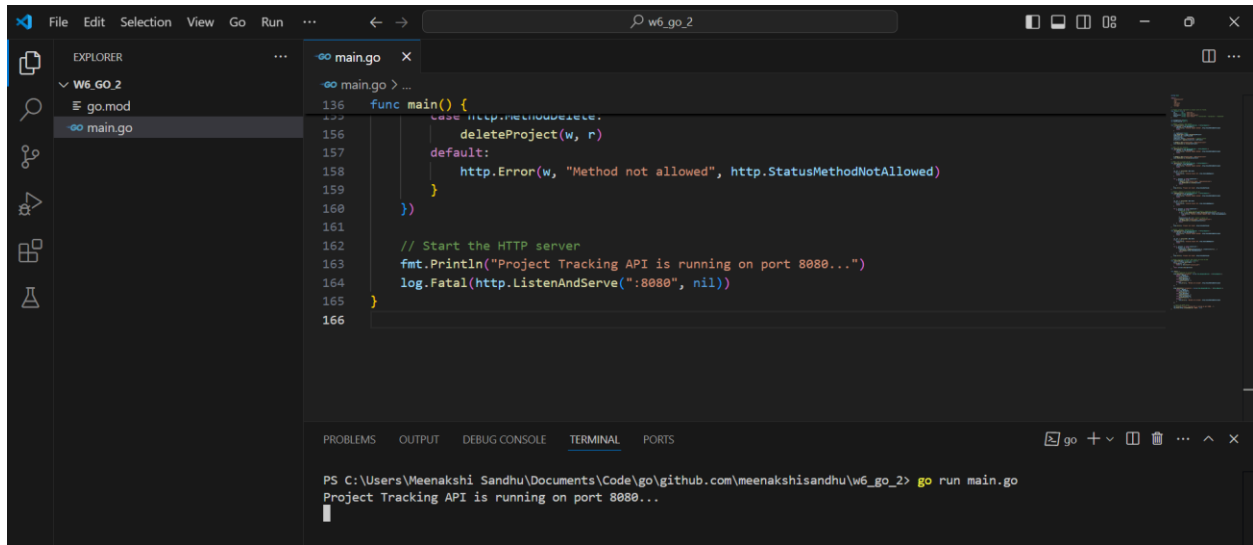


Server running

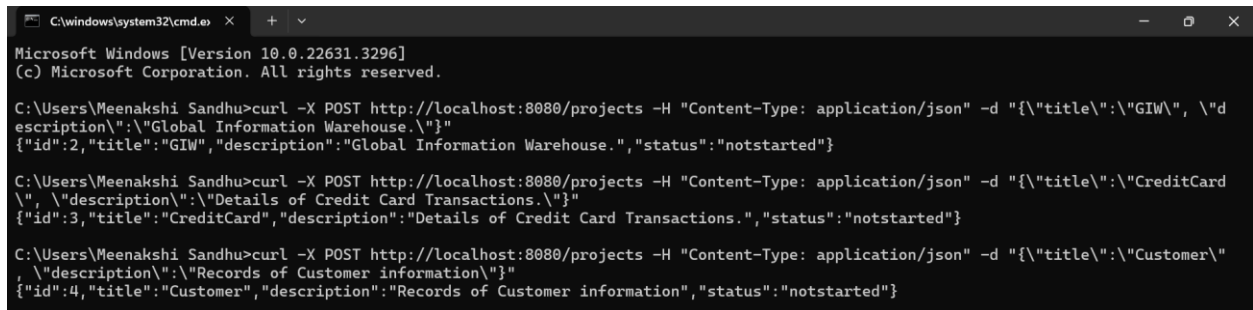


The screenshot shows the Visual Studio Code editor with a Go file named `main.go` open. The code defines a `main` function that sets up a simple HTTP server. It uses `http.HandleFunc` to register a `deleteProject` handler for the `/projects` endpoint. The `deleteProject` function takes a `http.ResponseWriter` and a `http.Request` as arguments and returns an `http.StatusMethodNotAllowed` error. The `main` function then starts the HTTP server on port 8080 using `log.Fatal(http.ListenAndServe(":8080", nil))`. The Explorer sidebar on the left shows the project structure with `W6_GO_2` as the root, containing `go.mod` and `main.go`. The Terminal at the bottom shows the command `go run main.go` being executed, and the output indicates that the Project Tracking API is running on port 8080.

```
136 func main() {
137     // Set up a simple HTTP server
138     http.HandleFunc("/projects", deleteProject)
139     deleteProject(w, r)
140     default:
141         http.Error(w, "Method not allowed", http.StatusMethodNotAllowed)
142     }
143 }
144
145 // Start the HTTP server
146 fmt.Println("Project Tracking API is running on port 8080...")
147 log.Fatal(http.ListenAndServe(":8080", nil))
148 }
```

PS C:\Users\Meenakshi Sandhu\Documents\Code\go\github.com\meenakshisandhu\w6_go_2> go run main.go
Project Tracking API is running on port 8080...

Create



The screenshot shows a Windows command prompt window with three `curl` commands being executed to create projects in the API. Each command uses the `-X POST` method to send a JSON payload to `http://localhost:8080/projects`. The first command creates a project with title "GIW" and description "Global Information Warehouse". The second command creates a project with title "CreditCard" and description "Details of Credit Card Transactions". The third command creates a project with title "Customer" and description "Records of Customer information". Each command outputs the JSON response from the server, which includes the project ID and status.

```
C:\Users\Meenakshi Sandhu>curl -X POST http://localhost:8080/projects -H "Content-Type: application/json" -d '{"title":"GIW", "description":"Global Information Warehouse."}'
{"id":2,"title":"GIW","description":"Global Information Warehouse.", "status":"notstarted"}

C:\Users\Meenakshi Sandhu>curl -X POST http://localhost:8080/projects -H "Content-Type: application/json" -d '{"title":"CreditCard", "description":"Details of Credit Card Transactions."}'
{"id":3,"title":"CreditCard","description":"Details of Credit Card Transactions.", "status":"notstarted"}

C:\Users\Meenakshi Sandhu>curl -X POST http://localhost:8080/projects -H "Content-Type: application/json" -d '{"title":"Customer", "description":"Records of Customer information."}'
{"id":4,"title":"Customer","description":"Records of Customer information", "status":"notstarted"}
```

Read All

```
localhost:8080/projects x +
localhost:8080/projects
Pretty-print
[
  {
    "id": 1,
    "title": "",
    "description": "",
    "status": "notstarted"
  },
  {
    "id": 2,
    "title": "GIW",
    "description": "Global Information Warehouse.",
    "status": "notstarted"
  },
  {
    "id": 3,
    "title": "CreditCard",
    "description": "Details of Credit Card Transactions.",
    "status": "notstarted"
  },
  {
    "id": 4,
    "title": "Customer",
    "description": "Records of Customer information",
    "status": "notstarted"
  }
]
```

Read based on ID

```
localhost:8080/projects/4 x +
localhost:8080/projects/4
Pretty-print
{
  "id": 4,
  "title": "Customer",
  "description": "Records of Customer information",
  "status": "notstarted"
}
```

Update

```
localhost:8080/projects/2 x +
localhost:8080/projects/2
Pretty-print
{
  "id": 2,
  "title": "GIW",
  "description": "Global Information Warehouse.",
  "status": "Inprogress"
}
```

Delete

```
C:\windows\system32\cmd.exe X + v
C:\Users\Meenakshi Sandhu>curl -X DELETE http://localhost:8080/projects/1
```

```
localhost:8080/projects x +
localhost:8080/projects ☆ | m :
Pretty-print ☒
[
  {
    "id": 2,
    "title": "GIW",
    "description": "Global Information Warehouse.",
    "status": "Inprogress"
  },
  {
    "id": 3,
    "title": "CreditCard",
    "description": "Details of Credit Card Transactions.",
    "status": "notstarted"
  },
  {
    "id": 4,
    "title": "Customer",
    "description": "Records of Customer information",
    "status": "notstarted"
  }
]
```