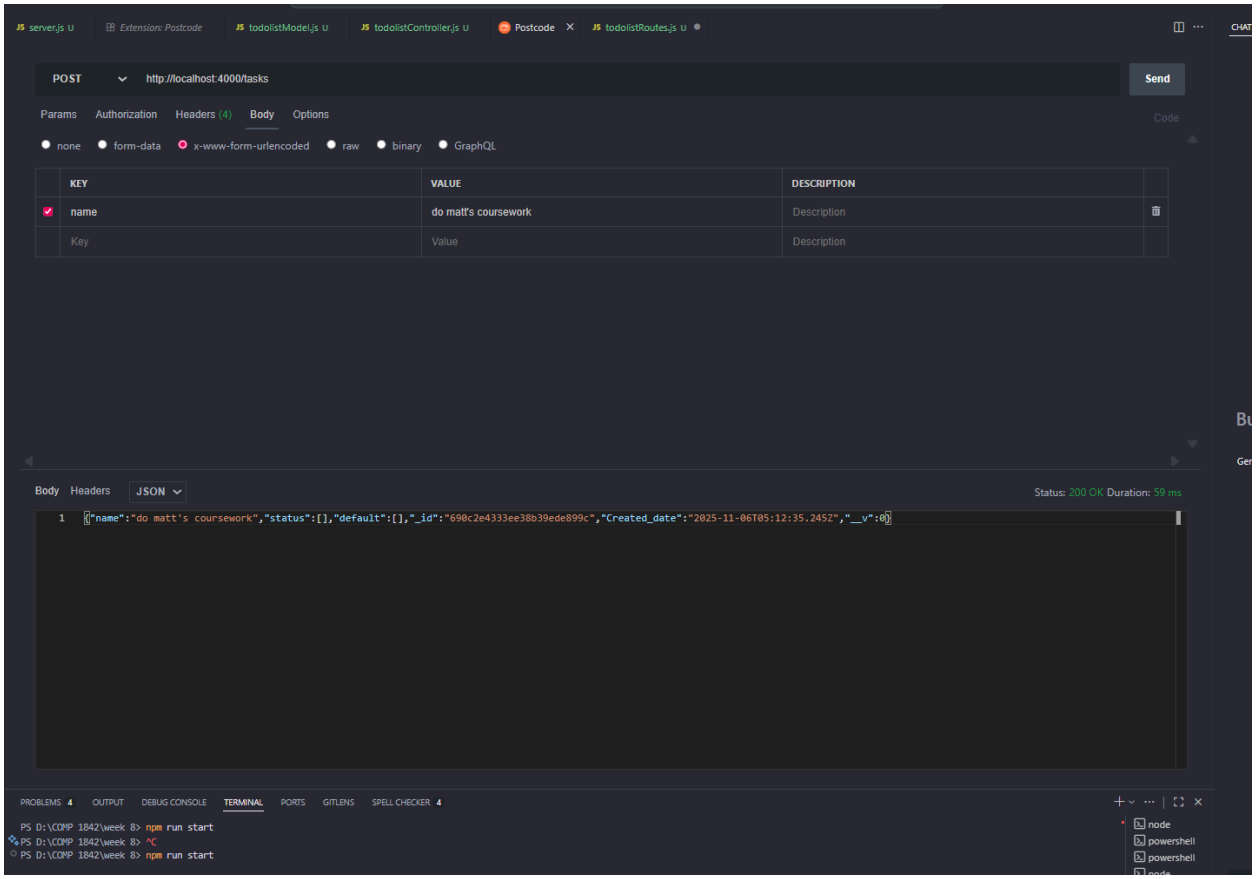
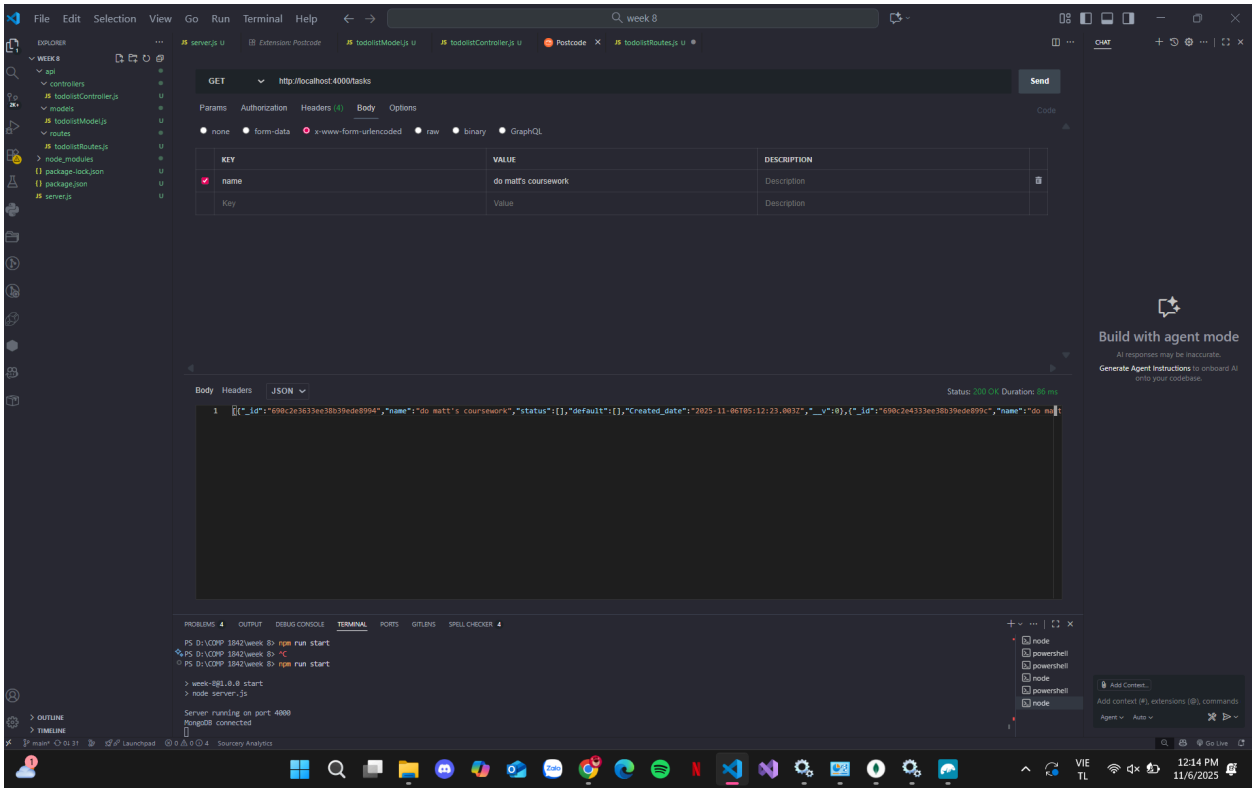
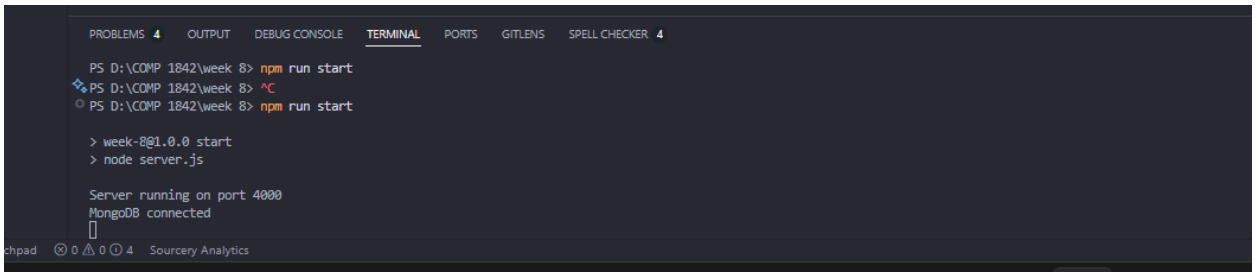
In this laboratory, I acquired the skills to construct and comprehend the architecture of a RESTful API project with Node.js and Express. The project was structured into three primary layers: models, controllers, and routes. The model delineates the structure of data and its interactions with the database. The controller encompasses the primary logic that manages HTTP requests, including the creation, retrieval, modification, and deletion of data. The route serves as a conduit linking API endpoints to their respective controller functions, guaranteeing the accurate processing of each request.

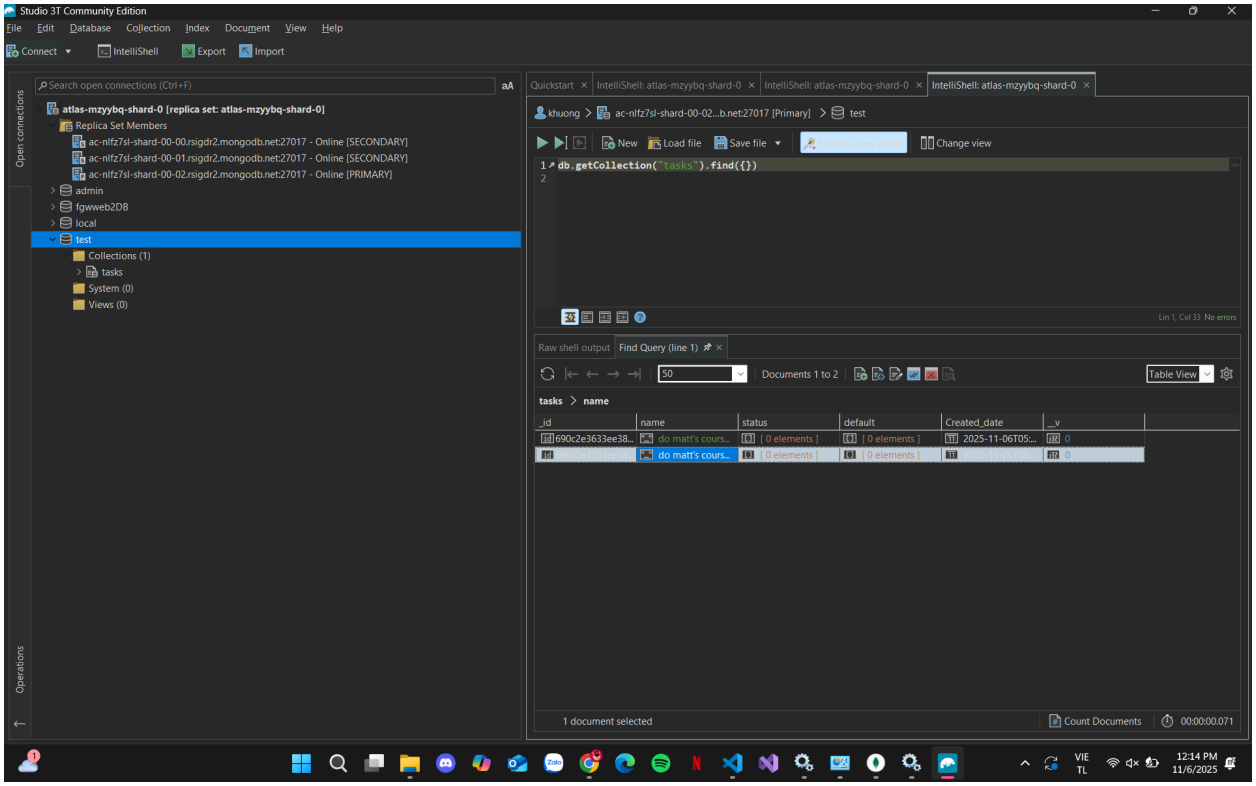
I practiced API testing with Postman, doing several HTTP calls like GET, POST, PUT, and DELETE. For instance, executing POST /users yields a 201 (Created) response upon the successful addition of a new record. The GET /users endpoint returns a 200 (OK) status with a list of users, whereas the GET /users/{id} endpoint returns a 200 (OK) status if the user exists, or a 404 (Not Found) status if the ID is invalid. PUT /users/{id} yields a 200 (OK) or 204 (No Content) status upon a successful update, whereas DELETE /users/{id} returns a 200 (OK) status if the record is successfully deleted.

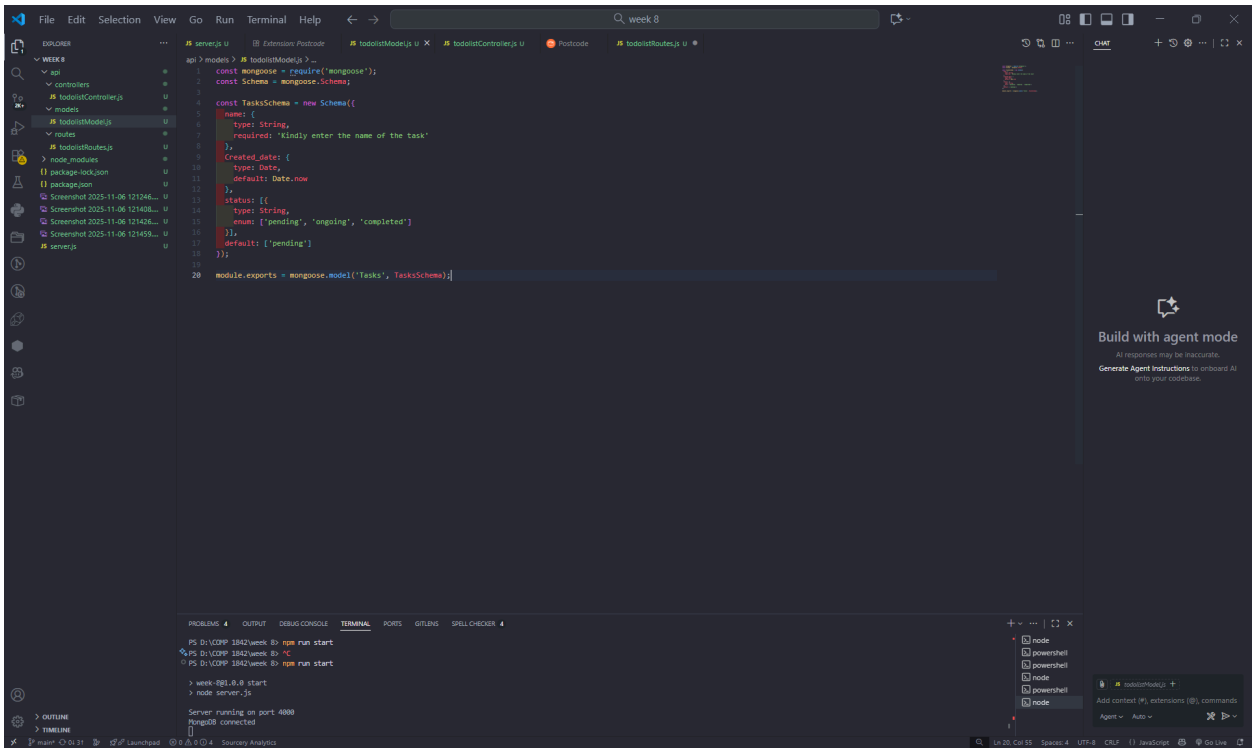
During this process, I acquired a comprehensive understanding of the correlation between HTTP methods and CRUD processes, as well as the interpretation of prevalent status codes including 200, 201, 400, 404, and 500. This lab facilitated the development of a robust foundation in the design, organization, and testing of RESTful APIs.




File Edit Selection View Go Run Terminal Help
week 8
GET
http://localhost:4000/tasks
Send
Params Authorization Headers (4) Body Options
none form-data x-www-form-urlencoded raw binary GraphQL
KEY VALUE DESCRIPTION
name
do matt's coursework
Description
Body Headers JSON
Status: 200 OK Duration: 86 ms
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS SPELL CHECKER 4
PS D:\COMP 1842\week 8> npm run start
PS D:\COMP 1842\week 8> ^C
PS D:\COMP 1842\week 8> npm run start
> week-8@1.0.0 start
> node server.js
Server running on port 4000
MongoDB connected
Build with agent mode
Generate Agent Instructions
12:14 PM
11/6/2025


PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS SPELL CHECKER 4
PS D:\COMP 1842\week 8> npm run start
PS D:\COMP 1842\week 8> ^C
PS D:\COMP 1842\week 8> npm run start
> week-8@1.0.0 start
> node server.js
Server running on port 4000
MongoDB connected
Sourcery Analytics


Studio 3T Community Edition
File Edit Database Collection Index Document View Help
Connect IntelliShell Export Import
Search open connections (Ctrl+F)
atlas-mzyybq-shard-0 [replica set: atlas-mzyybq-shard-0]
Replica Set Members
ac-nlfz7sl-shard-00-00.rsigdr2.mongodb.net:27017 - Online [SECONDARY]
ac-nlfz7sl-shard-00-01.rsigdr2.mongodb.net:27017 - Online [SECONDARY]
ac-nlfz7sl-shard-00-02.rsigdr2.mongodb.net:27017 - Online [PRIMARY]
admin
fgwweb2DB
local
test
Collections (1)
tasks
System (0)
Views (0)
Quickstart
IntelliShell: atlas-mzyybq-shard-0
khuong > ac-nlfz7sl-shard-00-02....b.net:27017 [Primary] > test
New Load file Save file Change view
db.getCollection("tasks").find({})
Raw shell output
Find Query (line 1)
Documents 1 to 2
Table View
tasks > name
_id name status default Created_date _v
690c2e3633ee38... do matt's cours... [0 elements] [0 elements] 2025-11-06T05:... 0
do matt's cours... [0 elements] [0 elements] 0
1 document selected
Count Documents
00:00:00.071
12:14 PM
11/6/2025


File Edit Selection View Go Run Terminal Help
week 8
const mongoose = require('mongoose');
const Schema = mongoose.Schema;
const TasksSchema = new Schema({
name: {
type: String,
required: 'Kindly enter the name of the task'
},
Created_date: {
type: Date,
default: Date.now
},
status: [{
type: String,
enum: ['pending', 'ongoing', 'completed']
}],
default: ['pending']
});
module.exports = mongoose.model('Tasks', TasksSchema);
Build with agent mode
PS D:\COMP 1842\week 8> npm run start
PS D:\COMP 1842\week 8> ^C
PS D:\COMP 1842\week 8> npm run start
> week-8@1.0.0 start
> node server.js
Server running on port 4000
MongoDB connected