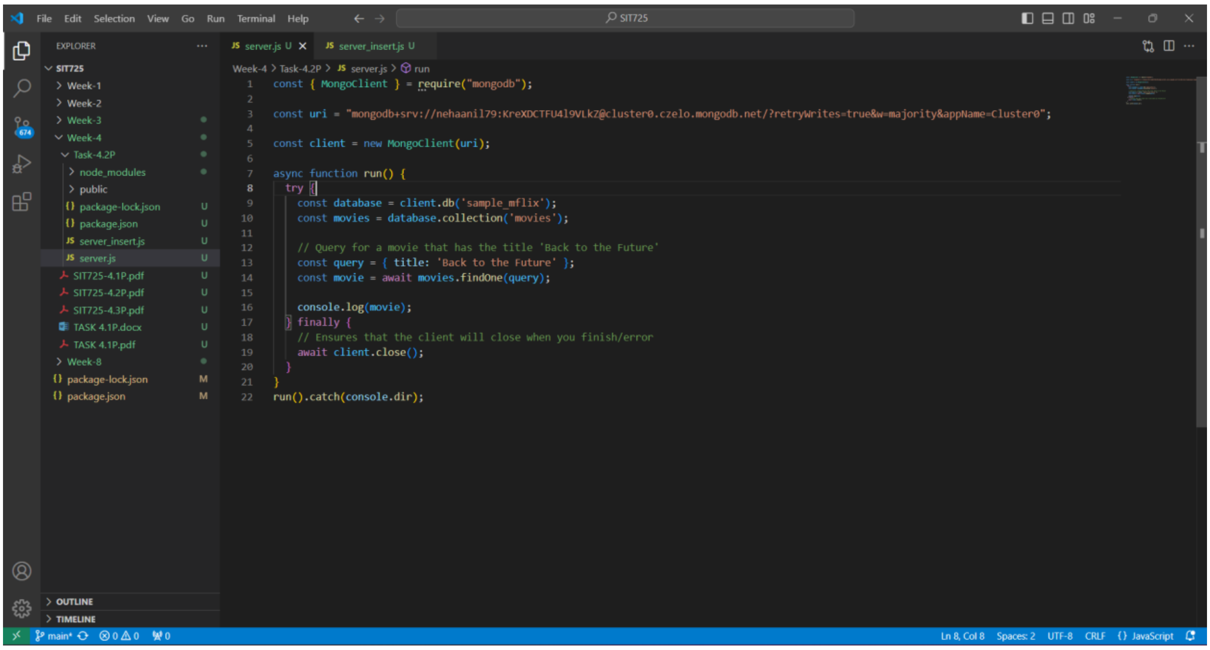
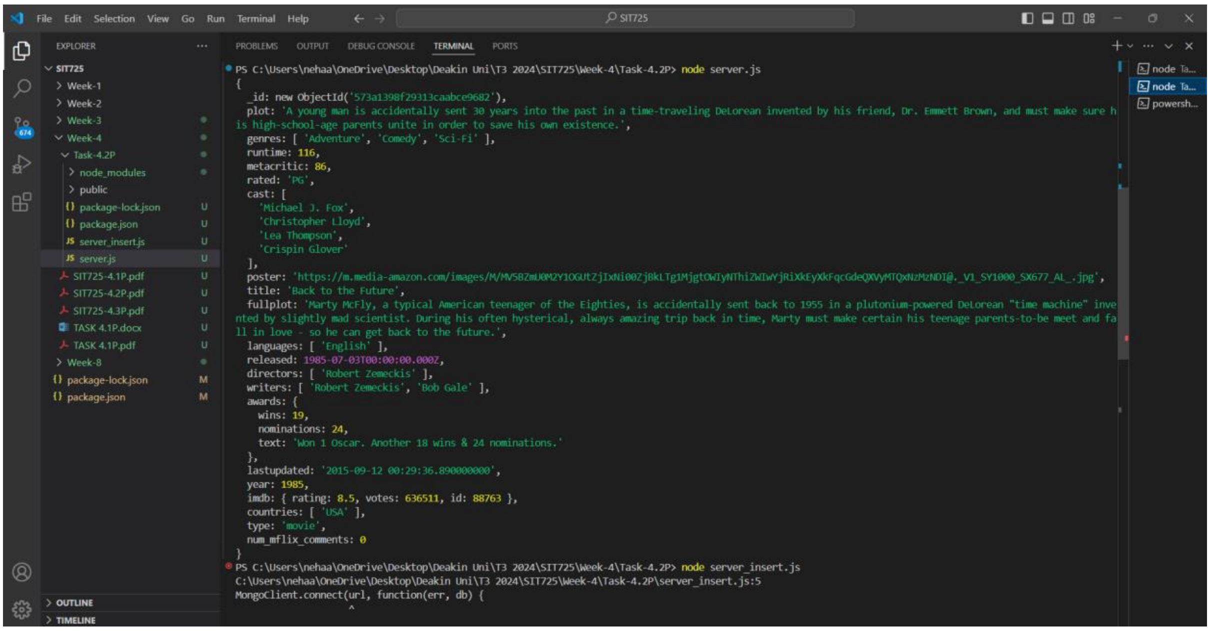
TASK 4.2P

Code for finding a document:



```javascript
const { MongoClient } = require("mongodb");

const uri = "mongodb+srv://nehaanil79:KreXDCTFU4l9VLkZ@cluster0.czelo.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0";

const client = new MongoClient(uri);

async function run() {
  try {
    const database = client.db('sample_mflix');
    const movies = database.collection('movies');

    // Query for a movie that has the title 'Back to the Future'
    const query = { title: 'Back to the Future' };
    const movie = await movies.findOne(query);

    console.log(movie);
  } finally {
    // Ensures that the client will close when you finish/error
    await client.close();
  }
}
run().catch(console.dir);
```

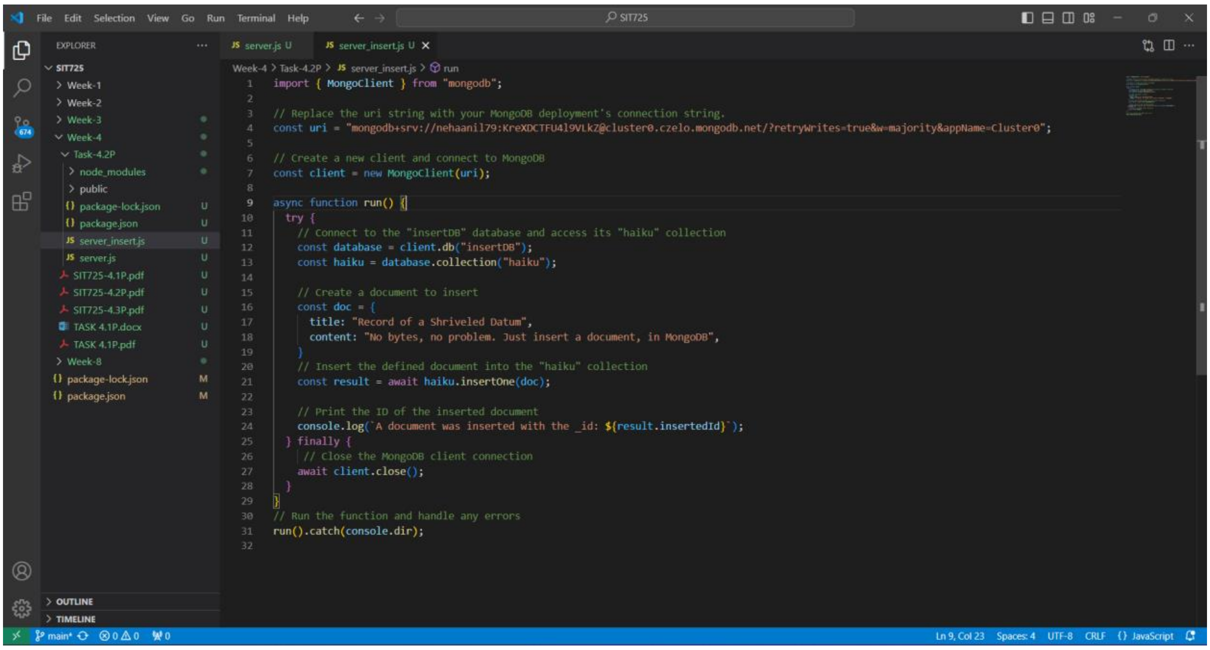
Output:



File Edit Selection View Go Run Terminal Help
SIT725
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
node Ta...
node Ta...
powersh...

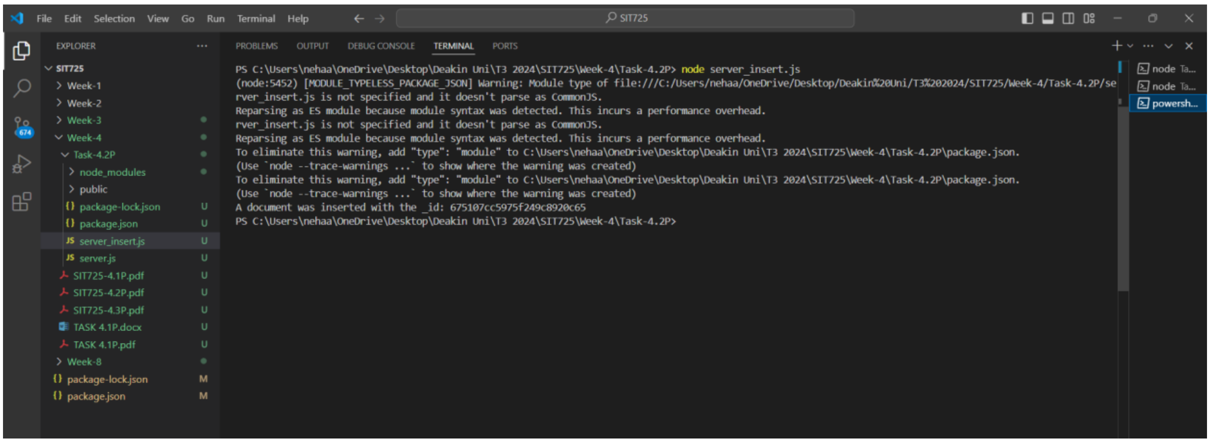
```
PS C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-4\Task-4.2P> node server.js
{
  _id: new ObjectId('573a1398f29313caabce9682'),
  plot: 'A young man is accidentally sent 30 years into the past in a time-traveling DeLorean invented by his friend, Dr. Emmett Brown, and must make sure h
is high-school-age parents unite in order to save his own existence.',
  genres: [ 'Adventure', 'Comedy', 'Sci-Fi' ],
  runtime: 116,
  metacritic: 86,
  rated: 'PG',
  cast: [
    'Michael J. Fox',
    'Christopher Lloyd',
    'Lea Thompson',
    'Crispin Glover'
  ],
  poster: 'https://m.media-amazon.com/images/M/MV5BZmU0M2Y1OGUtZjIxNi00ZjBkLTg1MjgtOWIyNThiZWIwYjRiXkEyXkFqcGdeQXVyMTQxNzMzNDI@._V1_SY1000_SX677_AL_.jpg',
  title: 'Back to the Future',
  fullplot: 'Marty McFly, a typical American teenager of the Eighties, is accidentally sent back to 1955 in a plutonium-powered DeLorean "time machine" inve
nted by slightly mad scientist. During his often hysterical, always amazing trip back in time, Marty must make certain his teenage parents-to-be meet and fa
ll in love - so he can get back to the future.',
  languages: [ 'English' ],
  released: 1985-07-03T00:00:00.000Z,
  directors: [ 'Robert Zemeckis' ],
  writers: [ 'Robert Zemeckis', 'Bob Gale' ],
  awards: {
    wins: 19,
    nominations: 24,
    text: 'Won 1 Oscar. Another 18 wins & 24 nominations.'
  },
  lastupdated: '2015-09-12 00:29:36.890000000',
  year: 1985,
  imdb: { rating: 8.5, votes: 636511, id: 88763 },
  countries: [ 'USA' ],
  type: 'movie',
  num_mflix_comments: 0
}
PS C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-4\Task-4.2P> node server_insert.js
C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-4\Task-4.2P\server_insert.js:5
MongoClient.connect(url, function(err, db) {
                         ^
```

Code for insert a document:



```
import { MongoClient } from "mongodb";

// Replace the uri string with your MongoDB deployment's connection string.
const uri = "mongodb+srv://nehaanil79:KreXDCTFU4l9VLkZ@cluster0.czelo.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0";

// Create a new client and connect to MongoDB
const client = new MongoClient(uri);

async function run() {
  try {
    // Connect to the "insertDB" database and access its "haiku" collection
    const database = client.db("insertDB");
    const haiku = database.collection("haiku");

    // Create a document to insert
    const doc = {
      title: "Record of a Shriveled Datum",
      content: "No bytes, no problem. Just insert a document, in MongoDB",
    }
    // Insert the defined document into the "haiku" collection
    const result = await haiku.insertOne(doc);

    // Print the ID of the inserted document
    console.log(`A document was inserted with the _id: ${result.insertedId}`);
  } finally {
     // Close the MongoDB client connection
    await client.close();
  }
}
// Run the function and handle any errors
run().catch(console.dir);
```

Output:


File Edit Selection View Go Run Terminal Help
SIT725
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
node Ta...
node Ta...
powersh...

```
PS C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-4\Task-4.2P> node server_insert.js
(node:5452) [MODULE_TYPELESS_PACKAGE_JSON] Warning: Module type of file:///C:/Users/nehaa/OneDrive/Desktop/Deakin%20Uni/T3%202024/SIT725/Week-4/Task-4.2P/se
rver_insert.js is not specified and it doesn't parse as CommonJS.
Reparsing as ES module because module syntax was detected. This incurs a performance overhead.
rver_insert.js is not specified and it doesn't parse as CommonJS.
Reparsing as ES module because module syntax was detected. This incurs a performance overhead.
To eliminate this warning, add "type": "module" to C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-4\Task-4.2P\package.json.
(Use `node --trace-warnings ...` to show where the warning was created)
To eliminate this warning, add "type": "module" to C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-4\Task-4.2P\package.json.
(Use `node --trace-warnings ...` to show where the warning was created)
A document was inserted with the _id: 675107cc5975f249c8920c65
PS C:\Users\nehaa\OneDrive\Desktop\Deakin Uni\T3 2024\SIT725\Week-4\Task-4.2P>
```

My Repo Link : https://github.com/nehaanil12345/SIT725.git