

Node Js Task

There are two directories: '**task**' (containing problems 1, 2, and 3) and '**node-app**' (containing problems 4 and 5).

task (containing problems 1, 2, and 3)

Node_Modules: Run **npm install** inside the **tasks directory**

Problem 1: Asynchronous Operations

Create a function that takes in an array of URLs and downloads the contents from each URL using asynchronous methods. Once all downloads are complete, return an array with the downloaded contents in the same order as the URLs.

Directory_Path:

tasks → **src** → **asynchronousOperations.js**

How to run:

Navigate to the **tasks directory** and execute the command
npm run asynchronousOperations

Problem 2: Error Handling

Design a function that fetches data from an API endpoint. Implement proper error handling to handle various HTTP status codes and network failures. Log appropriate messages for each type of error encountered.

Directory_Path:

tasks → **src** → **errorHandling.js.js**

How to run

Navigate to the **tasks directory** and execute the command
npm run errorHandling

Problem 3: File System Operations

Develop a utility that reads a directory and lists all files with a specific extension (e.g., .txt). Implement this functionality using Node.js's File System module.

Directory_Path:

taks --> src --> fileSystemOperations.js

How to run:

Navigate to the **task directory** and execute the command

npm run fileSystemOperations

node-app (containing problems 4 and 5)

The technology stack used in the Node.js application includes:

- Express
- MySQL
- Prisma (for managing User and Post entities)

Installation:

Node_Modules: Run **npm install** inside the **node_app directory**

Migrations: **npx prisma migrate dev**

Problem 4: Database Interaction

Create a simple REST API endpoint using Express.js that interacts with a database (can be any database of your choice). Implement CRUD operations (Create, Read, Update, Delete) for a specific resource (e.g., users, posts).

I have attached a Postman collection for your testing. Please review it, and ensure that you run the migration.

I have created the following endpoints:

- Get all Posts
- Get Post by ID
- Create Post (Required authentication token - call login api and get authentication token)
- Delete Post (Required authentication token - call login api and get authentication token)
- Update Post"

Problem 5: Authentication

Implement an authentication middleware using JWT (JSON Web Tokens) in a Node.js application. Create endpoints that require authentication to access specific resources. Ensure proper token validation and handling of expired or invalid tokens.

I have attached a Postman collection for your testing. Please review it, and ensure that you run the migration.

I have created the following endpoints:

- Register User
- Login User

The **Create Post** and **Delete Post** endpoints require a token. These endpoints perform token verification and check the token's expiry date.