

# DISASTER RECOVERY PROJECT ON IBM VIRTUAL CLOUD SERVER

## OVERVIEW:

This documentation provides an overview of the Disaster Recovery project, which involves deploying a To-Do List web application on IBM Cloud Virtual Servers. The project includes server-side code development and a disaster recovery plan.

## SERVER SCRIPT:

### Description:

The server script is responsible for handling HTTP requests, serving static files, and managing To-Do list data.

**File:** server.js

### Dependencies:

**Express.js:** A web application framework for Node.js.

**body-parser:** Middleware for parsing request bodies. **url:**

Node.js module for working with URLs.

**path:** Node.js module for working with file and directory paths.

### Initialization:

The server is initialized using Express.js on a specified port (3000).

### Middleware:

The server uses the body-parser middleware for parsing form data in POST requests.

Static files (HTML, CSS, JavaScript) are served using express.static middleware from the "public" directory.

```
1 import express from "express";
2 import bodyParser from "body-parser";
3 import { fileURLToPath } from "url";
4 import path from "path";
5
6 const __filename = fileURLToPath(import.meta.url);
7 const __dirname = path.dirname(__filename);
8
9 const app = express();
10 const port = 3000; // Middleware
11 app.use(bodyParser.urlencoded({ extended: true }));
12 app.use(express.static(path.join(__dirname, "public")));
13
14 // Data storage
15 const todolist = [];
16
17 // Routes
18 app.get("/", (req, res) => {
19   res.render("index.ejs", { data: todolist });
20 });
21
22 app.get("/add", (req, res) => {
23   res.sendFile(path.join(__dirname, "public", "add.html"));
24 });
25
26 app.post("/adder", (req, res) => {
27   const currentDate = new Date();
28   const year = currentDate.getFullYear();
29   const month = currentDate.getMonth() + 1;
30   const day = currentDate.getDate();
31   const date = `${day}:${month}:${year}`;
32   const amOrPm = currentDate.getHours() >= 12 ? "PM" : "AM";
33   const hours = currentDate.getHours() % 12;
34   const minutes = currentDate.getMinutes();
```

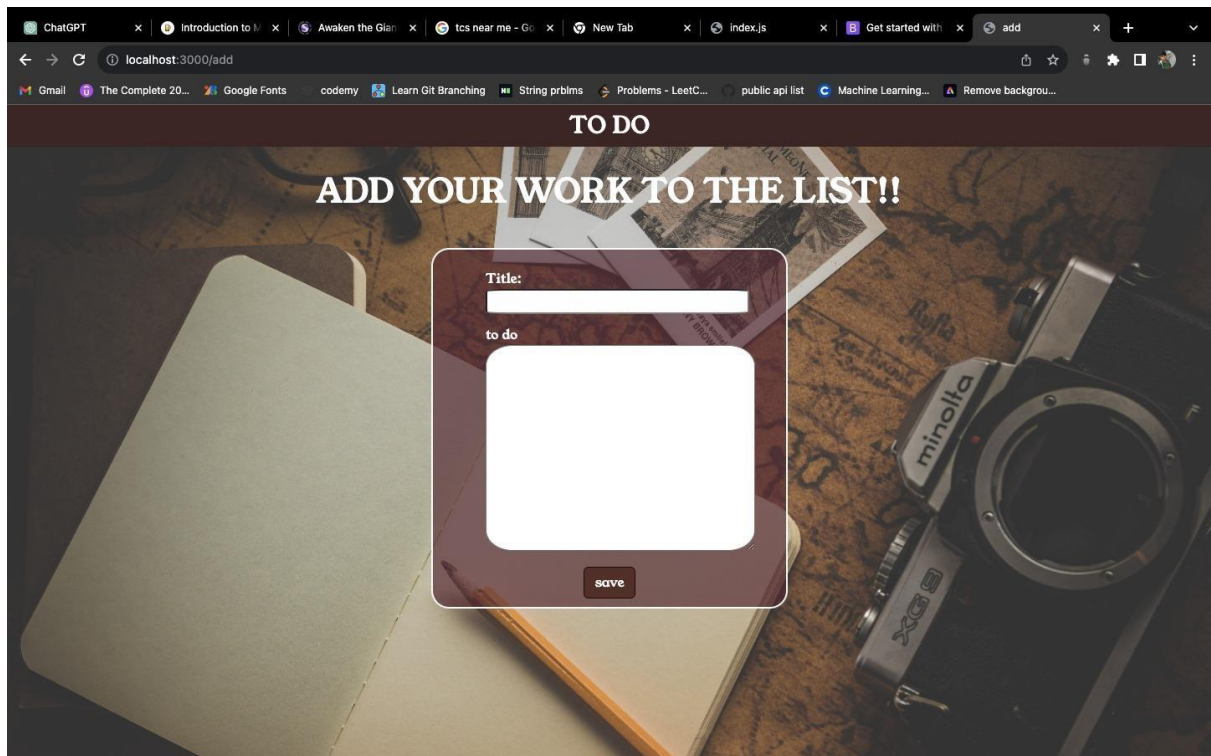
## Data Storage:

To-Do list data is currently stored in memory as an array named todolist.

## Routes:

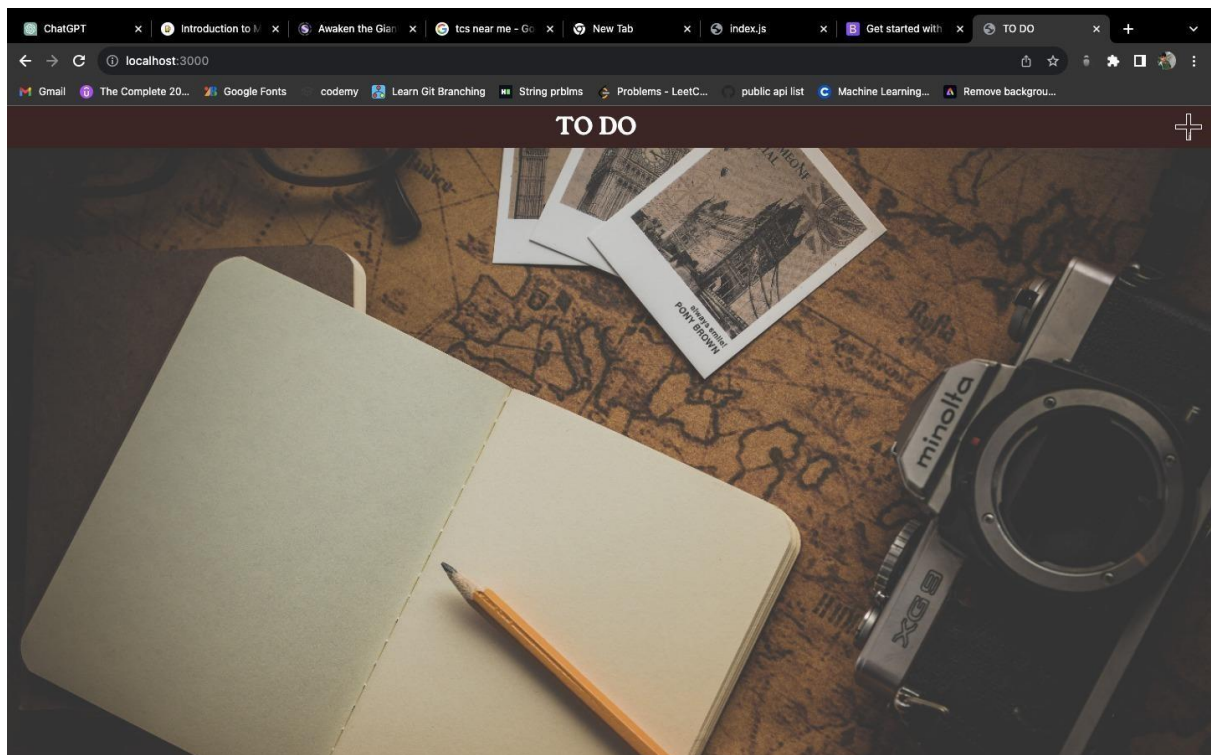
**GET /:** Renders the To-Do list on the homepage.

**GET /add:** Serves the HTML form for adding new tasks.

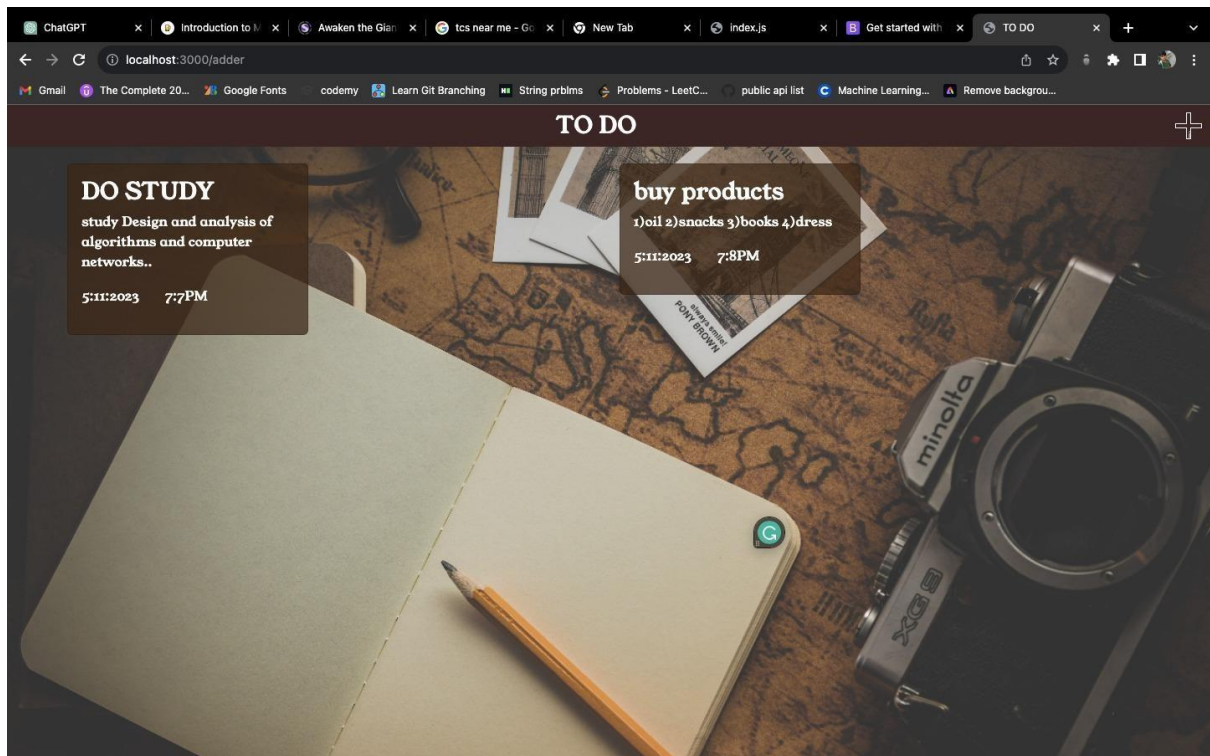


**POST /adder:** Processes the form submission, adds a new task to the To-Do list, and re-renders the To-Do list.

**BEFORE ADDING THE TASK:**



## RENDERING THE TO DO LIST AFTER PROCESSING THE FORM SUBMISSION:

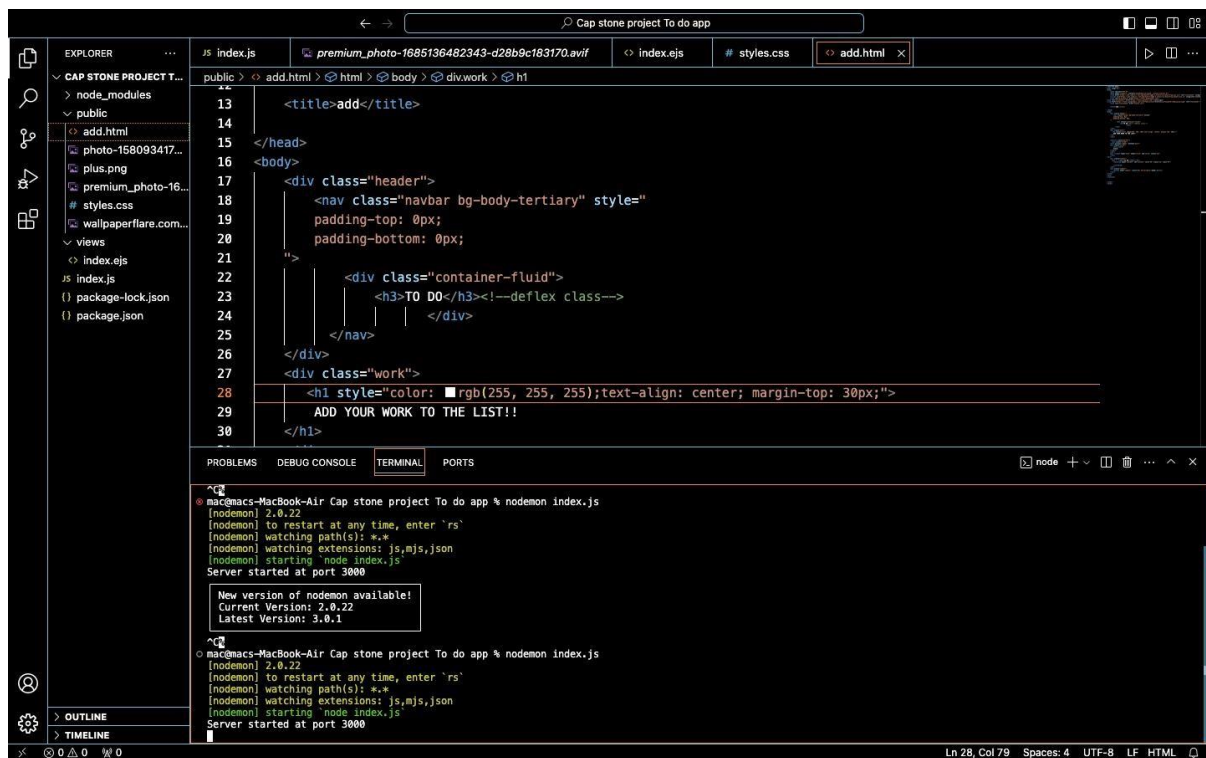


### Date and Time Handling:

The server script captures the current date and time for each added task.

### Starting the Server:

The server is started with a message indicating the port it's running on.



## Disaster Recovery Plan:

### Description:

The disaster recovery plan outlines measures for ensuring the high availability and reliability of the To-Do List application on IBM Cloud Virtual Servers.

### Components:

- 1.Backup and Recovery:** Describe the backup and data recovery procedures for the application.
- 2.Failover and Load Balancing:** Explain the failover mechanisms and load balancing configuration.
- 3.Monitoring and Alerts:** Detail the monitoring and alerting systems in place.
- 4.Testing and Validation:** Describe the testing and validation processes for disaster recovery.
- 5.Documentation Updates:** Explain how the disaster recovery plan and documentation will be updated.
- 6.Contact Information:** Include contact information for team members responsible for disaster recovery execution.

### Future Development:

The current project is at a development stage and does not yet include features such as user authentication or database integration. Future development steps may include:

Integrating a MongoDB database for persistent data storage.

Implementing user authentication to allow multiple users to manage their own tasks.

Enhancing the user interface and adding features like task editing and completion status.

Ensuring security measures to protect user data and application resources.

**Conclusion:**

This documentation provides an initial overview of the To-Do list application's server script and disaster recovery plan. Future updates and enhancements will be made to expand the functionality of the application and the disaster recovery capabilities.