REPORT ON BACKEND PROJECT

Name = Sayali Balasaeb Badar

Mob : 8767983178

Introduction:

The Task Handler is simple RESTFUL API which is build in Node js. Basically this API is used to manage collections of task by providing CRUD (Create, Read, Update, Delete) operations. The tasks are store in an array. This report includes approach, algorithm choices, API documentation and testing the API.

Approach:

☐ **Initialize the Project**: Use 'npm init' to create a package.json file for managing project dependencies.

☐ **Install Dependencies**: Install Express.js and body-parser for handling HTTP requests and parsing JSON bodies respectively.

API Design:

The API includes five endpoints to support CRUD operations:

1. GET /tasks: Retrieve a list of all tasks.
2. GET /tasks/:id: Retrieve a specific task by ID.
3. POST /tasks: Create a new task.
4. PUT /tasks/:id: Update an existing task by ID.
5. DELETE /tasks/:id: Delete a task by ID.

API Documentation:

Base URL:

http://localhost:3000

End Points:

A. Get/tasks:

1. URL: '/tasks/

2. Method Used: Get

3. Explanation: Used to retrieve list of all

4. Response: (200 OK)

```
 {

       "id": 1,

       "title": "To develop a Backend Project",

       "description": "This task is for developer"

 }
```

B. Get/tasks/:id :

1. URL: '/tasks/:id'

2. Method Used: Get

3. Explanation: Used to retrieve specific task by it's id.

4. Request:

        'http://localhost/tasks/2'

5. Response: (200 OK)

```
{

       "id":2,

       "title": "Design a webpage",

       "description": "This task is only"

}
```

6. Error: (If record not found : 404)

```
{
        "error": "Task not found"
}
```

C. Post:

1. URL: '/tasks'

2. Method Used: post

3. Explanation: Used to create new Task.

4. Request:

```
{
        "title": "Enter new task",
        "description": "Enter  a new task description"
}
```

5. Response: (200 OK)
```
{
        "id":3,
         "title": "Enter new task",
         "description": "Enter  a new task description"
}
```

6. Error: (If record not found : 404)
```
{
  "error": "Title and description are required"
}
```

D. Put:

1. URL: '/tasks/:id'

2. Method Used: put

3. Explanation: Used to update an existing record by id.

4. Request:

{

        "id":2,

        "title": "Update this task",

        "description": "This is updated description"

}

5. Response: (200 OK)

{

        "id":2,

        "title": "Analyst",

        "description": "This is for Data Analyst"

}

6. Error: (If record not found: 404)

{

  "error" : "Title and description are required"

}

E. Delete:

1. URL: '/tasks/:id'

2. Method Used: delete

3. Explanation: Used to update an existing record by id.

4. Request: (204 No Content)

'http://localhost:3000/tasks/1'

5. Error: (If record not found: 404)

{

  "error" : "Task not found"

}

Testing Tool:

- Postman: Postman is an utility tool which is used for API testing and helps in documentation. In which we have to pass localhost link with proper port number('http://localhost:3000') . According to end points select the end points which you want to perform and test the API.

Conclusion:

This is Task Handler will allow user to do CRUD (Create, Read, Update and Delete) operations. It is a simple RESTful API which is build in Express Js. It includes many features like validation, error handling and API documentation. Postman is utility tool which is used for API documentation and testing of API's.