Assignment 5

Part 1: Simple CRUD Operations Using Express.js (5 Grades)

- **Task 1.1:** Create an Express.js route that adds a new user to your users stored in a JSON file. The user data should include **"id, name, age, and email"** The id should be generated based on the current number of users in the JSON file, and ensure the email is unique by checking that no other user in the JSON file has the same email**. *(1 Grade) Note: The JSON file will contain an array of users.***
  - **URL:** POST /addUser

input	output
{	{
"name": "User 1",	"message": "User added successfully."
"age": 27,	}
"email": "user@email.com"	Or
}	{
	"message": "Email already exists."
	}

- **Task 1.2:** updates an existing user's name, age, or email by their ID. The user ID should be retrieved from the **params**. **(1 Grade)** ***Note:*** **Remember to update the corresponding values in the JSON file**
  - **URL**: PATCH /updateUser/:id

Input	output
{	{
"name": "Updated name",	"message": "User updated successfully."
"age": 28	}
}	Or
	{
	"message": " User id not found."
	}

Assignment 5

- **Task 1.3:** Delete a User by ID. The user id should be retrieved from either the **request body or optional params. (1 Grade)** ***Note:*** **Remember to delete from the file**
  - **Input URL:** DELETE "/deleteUser/0" or "/deleteUser"
  - **Input body (in case you didn't pass the id in the params):** { "id":0 }
  - **Output: {** "message": "User deleted successfully."} or { "message": "User id not found"}
- **Task 1.4**: Get a user by their name. The name will be provided as a **query parameter**. **(1 Grade)**
  - **URL:** GET /getUserByName
  - **Input:** GET /getUserByName?name=User 1
  - **Output**

```
{
  "id":0
  "name": "User 1",
   "age": 27,
   "email":
  "user@email.com"
}
```

OR

```
{
  "message":"User name not found"
}
```

- **Task 1.5:** Get User by ID. **(1 Grade)**
  - **URL:** GET /getUserById/:id
  - **Input:** GET /getUserById/0
  - **Output:**

```
{
  "id":0
  "name": "User 1",
   "age": 27,
   "email":
  "user@email.com"
}
```

OR

```
{
  "message":"User name not found"
}
```

Assignment 5

Part 2: Q&A on Node.js Internals (5 Grades)

Task 2.1: What is the Node.js Event Loop? (1 Grade)

Task 2.2: What is the Role of the V8 Engine? (1 Grade)

Task 2.3: What is the Node.js Thread Pool and How to Set the Thread Pool Size? (1 Grades)

Task 2.4: What is the purpose of the libuv library in Node.js? *(1 Grade)*

Task 2.5: Explain how Node.js handles asynchronous I/O operations. *(1 Grade)*

⚠️ Important Notes about postman

1. **Name the endpoint with a meaningful name like 'Add User', not dummy names.**
2. **Save your changes on each request(ctrl+s).**
3. **Include the Postman collection link (export your Postman collection) in the email with your assignment link**

Bonus (2 Grades)

How to deliver the bonus?

1- Solve the problem Majority Element on **LeetCode**

2- Inside your assignment folder, create a **SEPARATE FILE** and name it “bonus.js”

3- Copy the code that you have submitted on the website inside ”bonus.js” file