

# 1. JavaScript

■ **Debugging Help:** I expect to see 5 in the console, but it's showing undefined . What's wrong with my code?"

```
// I expect to see `5` in the console, but it's showing `undefined`. What's wrong with my code?"
function addNumbers(a, b) {
  let sum = a + b;
}
console.log(addNumbers(2, 3));
```

■ **Debugging Help:** Break Down What Happens Each Line of MyFun()

```
// Break Down What Happens Each Line of `MyFun()`
function MyFun(a, b) {
  let sum =a+b;
  let sub=a-b;
  let div=a/b;
  let mul=a*b;
  return sum+sub+div
}
console.log(MyFun(2, 3));
```

■ **Jest Unit Test :** Write Jest Unit Test Function And Execute Test

```
// Write Jest Unit Test Function And Execute Test
function addNumbers(a, b) {
  let sum = a + b;
}
module.exports = addNumbers;
```

■ **Code Explanation:** Explain the use of # here

```
// Explain the use of # here
class BankAccount {
  #balance = 0; // Private field

  deposit(amount) {
    this.#balance += amount;
  }

  withdraw(amount) {
    if (amount <= this.#balance) {
      this.#balance -= amount;
    } else {
      console.log("Insufficient balance");
    }
  }

  getBalance() {
    return this.#balance;
  }
}
```

```
const acc = new BankAccount();
acc.deposit(100);
console.log(acc.getBalance()); // Output: 100
console.log(acc.#balance); // Output: Error (private field is encapsulated)
```

■ **Learning Concepts:** How to create post request url = [www.example.com/api/login](http://www.example.com/api/login) postbody = {userName:"admin",password:"123"} using axios async/await.

■ **Best Practices & Clean Code:** Make this code clean with best practices

```
// Make this code clean with best practices
var d = new Date();
var days = ['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday'];
console.log(d.getDay());
```

■ **Code Review:** Review processItems() function code and give improvement

```
// Review processItems() function code and give improvement
let itemList=[{"id": 1, "name": "Item 1", "isActive": true, "isProcessed": false}, {"id": 2, "name": "Item 2", "isActive": true, "isProcessed": true}, {"id": 3, "name": "Item 3", "isActive": false, "isProcessed": false}, {"id": 4, "name": "Item 4", "isActive": true, "isProcessed": false}]

function processItems(itemList) {
  var result = [];
  for (var i = 0; i < itemList.length; i++) {
    var item = itemList[i];
    if (item.isActive && !item.isProcessed) {
      item.isProcessed = true;
      result.push(item);
    }
  }
  return result;
}
```

■ **Language conversion:** Convert this code in JavaScript to PHP

```
// Convert this code in JavaScript to PHP

class Father {
  //Properties
  first_name='Jhon'
  last_name='Dee'
  //method
  getName() {
    return (`The name of the person is ${this.first_name} ${this.last_name}`)
  }
}

class Son extends Father{
}

const SonObj = new Son();
console.log(SonObj.getName());
```

### Working With Real Life Problem :

- Convert Minutes into Seconds
- Write Regular Expression For Email Validation

#chatGPT

## 2. React

### ■ \*\*Working With Real Life Problem & Concepts Hacks:

- React file folder structure for my to do list project
- How to pass `JSON Object` using props
- Various DOM-related tasks example list with React `useRef`
- Write React Component & Make Table Using This JSON Array

```
// Write React Component & Make Table Using This JSON Array
[
  {"name": "Alice Smith", "age": 28, "city": "New York"},
  {"name": "Bob Johnson", "age": 35, "city": "Los Angeles"},
  {"name": "Carlos Gonzalez", "age": 42, "city": "Miami"},
  {"name": "Diana Chen", "age": 31, "city": "San Francisco"},
  {"name": "Ethan Watanabe", "age": 22, "city": "Seattle"}
]
```

- How to Call `async/await Axios Post Request` Inside `useEffect()`
- Features list of `JSX` in React component with simple example
- React `useState`, `useEffect`, `useRef` hooks method with example

### ■ Practice Project Ideas

- 5 Beginner Project Idea Using React `useState()` & `JSX`
- 5 Beginner Project Idea Using React & `Axios`

#chatGPT

### 3. MongoDB Database

#### ■ Demo Data Generation JSON | SQL | Mongo DB:

- Write a JSON array with 5 JSON object name, age, city use demo real data
- Write 5 SQL INSERT Query with name, age, city use demo real data
- Write 5 Mongo DB INSERT Query with name, age, city use demo real data

#### ■ JSON Validation & Correct Format

```
//Find Error Inside This JSON
```

```
let itemList=[{"id": 1, "name": "Item 1", "isActive": true, "isProcessed": false, {"id": 2, "name": "Item 2", "isActive": true, "isProcessed": true}, {"id": 3, "name": "Item 3", "isActive": false, "isProcessed": false}, {"id": 4, "name": "Item 4", "isActive": true, "isProcessed": false}]
```

#### ■ Design a database schema

- Create database schema model using mongoose
- Create database schema model using Prisma

```
// Create database schema model using mongoose
```

```
{
  "name": "Rain",
  "designation": "Sub Asst Engineer",
  "salary": 30000,
  "city": "Rangpur"
}
```

#### ■ Query Assistant

- Write insert, read, update, delete `mongodb` query for this object
- Write insert, read, update, delete `mongodb` mongoose for this object
- Write insert, read, update, delete `mongodb` aggregation query for this object
- Write insert, read, update, delete `mongodb` prisma query for this object

```
{
  "name": "Rain",
  "designation": "Sub Asst Engineer",
  "salary": 30000,
  "city": "Rangpur"
}
```

#### ■ Complex Query Assistant

- Write MongoDB aggregation query to select product join with brand & category

```
Brand Collection
```

```
{
  "_id": {
    "$oid": "64f8751a502e1b80556da142"
  },
  "brandName": "Apple",
  "brandImg": "https://photo.teamrabbil.com/images/2023/08/15/Phones---Tablets-01-3944.png"
}
```

Category Collection

```
{
  "_id": {
    "$oid": "64f875ed502e1b80556da14d"
  },
  "categoryName": "Laptop",
  "categoryImg": "https://photo.teamrabbil.com/images/2023/08/15/Phones---Tablets-01-3944.png"
}
```

Product Collection

```
{
  "_id": {
    "$oid": "64f87d60502e1b80556da15a"
  },
  "title": "Laptop",
  "shortDes": "Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,",
  "price": "1000",
  "discount": true,
  "discountPrice": "890",
  "image": "https://photo.teamrabbil.com/images/2023/08/15/Phones---Tablets-01-3944.png",
  "star": "4",
  "stock": true,
  "remark": "new",
  "categoryID": {
    "$oid": "64f875ed502e1b80556da14d"
  },
  "brandID": {
    "$oid": "64f8751a502e1b80556da142"
  }
}
```

Write MongoDB aggregation query to select product join with brand & category

#chatGPT

## 4 Software Engineering

I want to develop a inventory management system to manage my bakery shop

- Customer
- Product supplier
- Product category, brand
- Product list, price, discount
- Purchase from supplier
- Sales to customer
- Product stock
- Necessary business report

### ■ Requirement analysis

Prepare software requirement analysis features

### ■ Technology Selection

Suggest Framework, Database Based On JavaScript Ecosystem

Give Rating Score For MongoDB+Express+React Combination

### ■ Database Schema Design

Prepare Database Schema Using MongoDB Mongoose By Embedded Data Model

Prepare Database Schema Using MongoDB Mongoose By Normalized Data Model

Prepare Database Schema Using MySQL Prisma By Normalized Relational Data Schema

### ■ ERD Diagram

Can You Give Me ERD Diagram

Generate Graphical Diagram

### ■ Rest API List to Develop

Give me Rest API List to Develop

### ■ Generate Front End Page & Component List

Give Front End Page & Component List to Develop

### ■ Create Back-End Rest API Project File Folder Structure

Create Back-End Rest API Project File Folder Structure Use Express JS

### ■ Create Front End File Folder Structure

Create Front-End Project File Folder Structure Use React JS

### ■ Generate Front End UI Using DALL-E

Generate Front End UI Using DALL-E

#chatGPT