Round 1

Q.1 What will be the status code when the user is unauthorized?

a. 400

b. 401

c. 402

d. 403

Ans. 401

Q.2 Authorization token has to be given in which of the below?

a. query

b. params

c. headers

d. body

Ans. headers

Q.3 If the user logged in, the API will give the below response back to client?

a. Authorization token

b. Body

c. Password

d. Username

Ans. Authorization token

Q.4 If the user logged in, the client has to send which of the below to every subsequent request for authorization?

a. Authorization token

b. Body

c. Password

d. Username

Ans. Authorization token

Q.5 Which of the below is implement in the API side to validate each request?

a. Middleware

b. FirstWare

c. LastWare

d. AuthChecker

Ans. Middleware

Q.6 Which of the below is used to generate Authorization token?

a. JWT

b. AWT

c. SWT

d. CWT

Ans. JWT

Q.7 JWT stands for?

a. Javascript Web Token

b. Java Web Token

c. Json Web Token

d. Json Web Translation

Ans. Json Web Token

Q.8 The process of login into the application is called?

a. Authorization

b. Authentication

c. None of the above

d. All the above

Ans. Authentication

Q.9 The process of checking the user access and perform operation is called?

a. Authorization

b. Authentication

c. None of the above

d. All the above

Ans. Authorization

Q.10 The Authorization token can be stored in the browser _____?

a. localstorage

b. cookie

c. session storage

d. None of the above

Ans. cookie

Q.11 Which of the below package is used for unit testing?

a. jest

b. test

c. npm

d. None of the above

Ans. jest

Q.12 The test report will be saved in which of the below path?

a. coverage/test-report/index.html

b. coverage/report/index.html

c. coverage/lcov-report/index.html

d. coverage/index.html

Ans. coverage/lcov-report/index.html

Q.13 Which of the below library is used to test the UI automation testing?

a. chaijs

b. jest

c. puppeteer

d. tape

Ans. puppeteer

Q.14 Which of the below is not a benefits of unit testing?

a. help to find and fix bugs quickly and easily

b. contribute to higher code quality

c. act as documentation

d. act as an issue fixer in production

Ans. act as an issue fixer in production

Q.15 Usually, developers write unit tests first, then write the software code. This approach is known as .

a. Quality-driven development (QDD)

b. bug-driven development (bDD)

c. test-driven development (TDD)

d. software-driven development (TDD)

Ans. test-driven development (TDD)

Q.16 _____ a specific test that checks if a unit of code is working properly. Tests are organized into test suites.

a. Test case

b. Test framework

c. Test runner

d. None of the above

Ans. Test case

Q.17 _____ runs test cases in the browser to see how the unit under test behaves

a. Test case

b. Test framework

c. Test runner

d. None of the above

Ans. Test runner

Q.18 _____ can import some JavaScript code, invoke it, and test if it works properly

a. Test case

b. Test framework

c. Test runner

d. None of the above

Ans. Test framework

Q.19 Which of the below is a code coverage

a. line coverage

b. statement coverage

c. branch coverage

d. All the above

Ans. All the above

Q.20 _____ is a JavaScript test runner that lets you access the DOM via jsdom

a. jest

b. mocha

c. chai

d. All the above

Ans. jest

Q.21 Which one of the following is Cloud Platform by Amazon?

a. Azure

b. AWS

c. Heroku

d. Netlify

Ans. AWS

Q.22 Which one of the following is not a Cloud Platform ?

a. Azure

b. TCS

c. AWS

d. Heroku

Ans. TCS

Q.23 What are the types of scalling?

a. 1

b. 2

c. 3

d. 4

Ans. 2

Q.24 The two types of horizontal scaling to use to ensure the best possible experience for the users of your application

a. manual

b. automatic

c. All the above

d. None of the above

Ans. All the above

Q.25 Which is the most popular deployment platform

a. AWS

b. Azure

c. Heroku

d. GCP

Ans. AWS

Q.26 Which of the below is used to build the code?

a. Jenkins

b. Vikings

c. V8

d. All the above

Ans. Jenkins

Q.27 What is the npm command to build the code?

a. npm run build

b. npm execute build

c. npm build

d. npm bundle

Ans. npm run build

Q.28 How do you supply a commit message to a commit?

a. git message "I'm coding"

b. git add "I'm coding"

c. git commit "I'm coding"

d. git commit -m "I'm coding"

Ans. git commit -m "I'm coding"

Q.29 What is the correct commit syntax for all changes with a message?

a. git message -am "I'm coding"

b. git add -a "I'm coding"

c. git commit -a "I'm coding"

d. git commit -am "I'm coding"

Ans. git commit -am "I'm coding"

Q.30 How to create a pull request in github?

a. By creating a Pull Request through the GitHub interface

b. git pull-request

c. git submit

d. git commit -am "Done with Lab"

Ans. By creating a Pull Request through the GitHub interface

Q.31 Which of the following route parameter formats are valid?

a. /category/!:food-:pizza

b. /category/:food-:pizza

c. /category/:food/pizza/:pizzaId

d. None of the above

Ans. /category/!:food-:pizza

Q.32 Which of the following route parameter query formats are valid?

a. /category?food=pizza,price=200

b. /category?food=pizza&price=200

c. /category&food=pizza&price=200

d. /category?food=pizza|price=200

Ans. /category?food=pizza&price=200

Q.33 Which of the below field is used to update the category?

a. name

b. id

c. type

d. None of the above

Ans. id

Q.34 What are the steps to structure My application?

a. Route listings

b. Route map

c. MVC style controllers

d. All the above

Ans. All the above

Q.35 What are the steps used for Error Handling in Express.js?

a. Create a Middleware

b. Install Error Handler Middleware

c. Both of these

d. None of the above

Ans. Both of these

Q.36 What function are arguments available to Express JS?

a. Request

b. Response

c. Next Function

d. All the above

Ans. All the above

Q.37 Which function tells what to do when a get request at the given route is called?

a. app.get(route, callback)

b. get(route, callback)

c. js.get(route, callback)

d. fun.get(route, callback)

Ans. app.get(route, callback)

Q.38 Which method requests that the server accept the data enclosed in the request as a new object/entity of the resource identified by the URI?

a. GET

b. POST

c. PUT

d. DELETE

Ans. POST

Q.39 Which method requests that the server accept the data enclosed in the request as a modification to existing object identified by the URI?

a. GET

b. POST

c. PUT

d. DELETE

Ans. PUT

Q.40 _____ is a middleware which parses cookies attached to the client request object.

a. cookie

b. req.cookies

c. cookie-parser

d. All the above

Ans. cookie-parser

Round 2

Question: 1

Question name: Construct the Rectangle

Problem Statement

A web developer needs to know how to design a web page's size. So, given a specific rectangular web page's area, your job by now is to design a rectangular web page, whose length L and width W satisfy the following requirements:

The area of the rectangular web page you designed must equal to the given target area.

The width W should not be larger than the length L, which means L >= W.

The difference between length L and width W should be as small as possible.

Return an array [L, W] where L and W are the length and width of the web page you designed in sequence.

Example 1:

Input: area = 4

Output: [2,2]

Explanation: The target area is 4, and all the possible ways to construct it are [1,4], [2,2], [4,1].

But according to requirement 2, [1,4] is illegal; according to requirement 3, [4,1] is not optimal compared to [2,2]. So the length L is 2, and the width W is 2.

Example 2:

Input: area = 37

Output: [37,1]

Example 3:

Input: area = 122122

Output: [427,286]

Constraints:

1 <= area <= 107

Template

```
var constructRectangle = function(area, w) {
    // code logic here
    return constructRectangle(area, --w);
};
```

```
const area = parseInt(readline());
let w = // code here
console.log(constructRectangle(area, w))
```

Solution

```
/**
 * @param {number} area
 * @return {number[]}
 */
var constructRectangle = function(area, w) {

    if (area % w == 0)
            return [area/w, w]
    return constructRectangle(area, --w);
};
```

```
const area = parseInt(readline());
let w = Math.floor(Math.sqrt(area))
console.log(constructRectangle(area, w))
```

Solution: https://www.ideone.com/GZin5m

Question: 2

Question name: Interleaving String

Problem Statement

Given strings `s1`, `s2`, and `s3`, find whether `s3` is formed by an **interleaving** of `s1` and `s2`.

An **interleaving** of two strings `s` and `t` is a configuration where `s` and `t` are divided into `n` and `m` **non-empty** substrings respectively, such that:

- $s = s_1 + s_2 + ... + s_n$

- $t = t_1 + t_2 + ... + t_m$
- $|n - m| <= 1$
- The **interleaving** is $s_1 + t_1 + s_2 + t_2 + s_3 + t_3 + ...$ or $t_1 + s_1 + t_2 + s_2 + t_3 + s_3 + ...$

Note: `a + b` is the concatenation of strings `a` and `b`.

Example 1:

Input: `s1 = "aabcc", s2 = "dbbca", s3 = "aadbbcbcac"`

Output: `true`

Explanation: One way to obtain s3 is:

Split s1 into s1 = "aa" + "bc" + "c", and s2 into s2 = "dbbc" + "a".

Interleaving the two splits, we get "aa" + "dbbc" + "bc" + "a" + "c" = "aadbbcbcac".

Since s3 can be obtained by interleaving s1 and s2, we return true.

Example 2:

Input: `s1 = "aabcc", s2 = "dbbca", s3 = "aadbbbaccc"`

Output: `false`

Explanation: Notice how it is impossible to interleave s2 with any other string to obtain s3.

Example 3:

Input: `s1 = "", s2 = "", s3 = ""`

Output: `true`

Constraints:

- `0 <= s1.length, s2.length <= 100`
- `0 <= s3.length <= 200`
- `s1`, `s2`, and `s3` consist of lowercase English letters.

Template:

```
const isInterleave = function(s1, s2, s3) {
        const dp = new Map();
        const solve = (a = 0, b = 0, c = 0) => {
        // Implementation here
        return takeS1 || takeS2;
        }
        return solve();
};

const [s1, s2, s3] = readline().split(' ');
console.log(isInterleave(s1, s2, s3));
```

Solution:

```
const isInterleave = function(s1, s2, s3) {
        const dp = new Map();
        const solve = (a = 0, b = 0, c = 0) => {
        if(c == s3.length) return a == s1.length && b == s2.length;
        const key = [a, b, c].join(':');

    if(dp.has(key)) {
       return dp.get(key);
        }
```

```
            let takeS1 = false, takeS2 = false;
            if(s1[a] == s3[c]) takeS1 = solve(a + 1, b, c + 1);
            if(s2[b] == s3[c]) takeS2 = solve(a, b + 1, c + 1);

        dp.set(key, takeS1 || takeS2);
            return takeS1 || takeS2;
            }
            return solve();
};

const [s1, s2, s3] = readline().split(' ');
console.log(isInterleave(s1, s2, s3));
```

IDEOne link - https://www.ideone.com/CHozli

Round 3

Create a TODO API with the below functionality

Below are the API endpoints for the TODO Application

1) SignIn and SignUp API with below parameters
    a. id
    b. Email
    c. Password
    d. Created At

2) If user signed in successfully then return the Authorization token (JWT Token)

3) Create TODO with below fields

    a. id

    b. Todo title

    c. Is Completed

    d. Created At

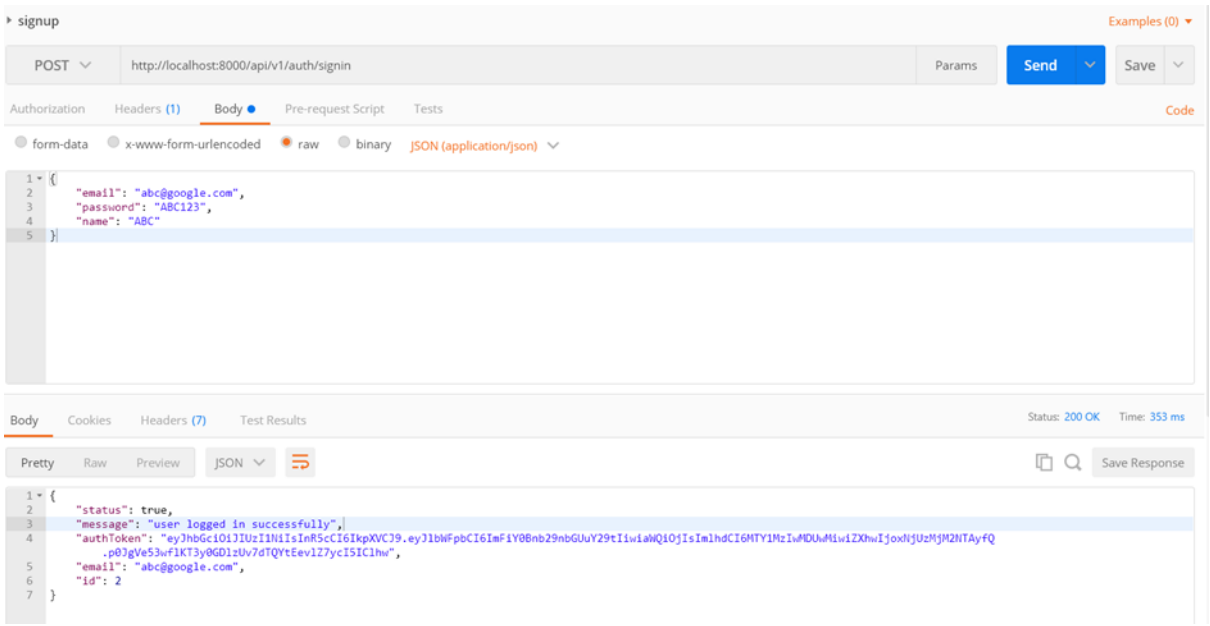
4) Every request an Authorization Bearer token has to send in the header for Authorization.

Round 4

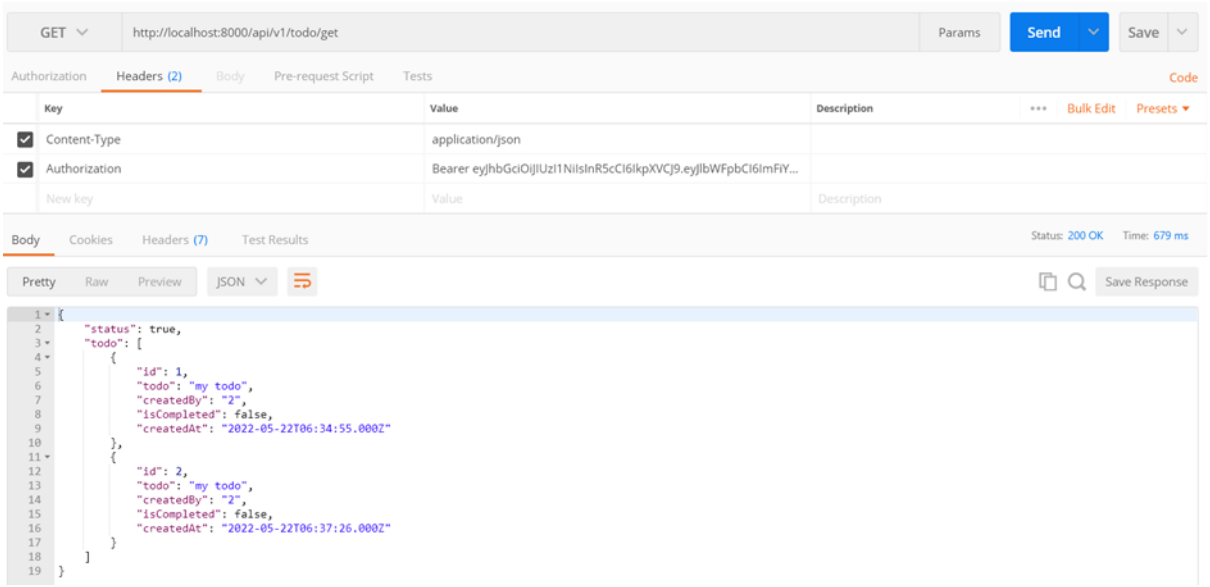
5) Update Todo with todo id and fields which need to update and it will allow only update the fields which the user had created.

6) Get Todo list which the particular user has created and apply filter in the request by using query string

Signin Postman:



Get TODO Postman:



Solution:

https://github.com/Saravananslb/todo-api