



স্বাগতম
Asadullah Ahmed



৪৪ ক্লাস জয়েনিং

আমার কোর্স

৬৫ রেকর্ডিং

৬৫ রিসোর্স

৬৫ Skillable

৬৫ ট্যালেন্টপুল



প্রোফাইল

← পেছনে যাই



লাইভ টেস্ট

গাইডলাইন

এক্সাম চলছে



সময়কাল

রাত ১১:০০ - রাত
১১:৪৫



টোটাল
মার্ক
১০

Total
Time

৩৭:৩১ min

Question

Live Test

A) Creating a Simple String Response with Express.js

In this test, you will create a basic Express.js application that responds with a simple string message when you access a specific URL. You'll set up a route to handle the request and send a plain text response.

Part 1: Setup the project on your computer

1. Create a new directory for your Express.js project.
2. Navigate to your project directory in your terminal.
3. Initialize your project by running `npm init` and follow the prompts to create a package.json file.

সমস্যায় পড়েছি? !



স্বাগতম

Asadullah Ahmed



৪৪ ক্লাস জয়েনিং

আমার কোর্স

৬৫ রেকর্ডিং

৬৫ রিসোর্স

Skillable

৬৫ ট্যালেন্টপুল



প্রোফাইল



4. Install the `express` module by running `npm install express` in your project directory.

Part 2: Express Application

5. Create an `app.js` file in your project directory.

6. Write a Node.js program that does the following:

- Import the `express` module and create an instance of the Express application.
- Define a route handler that listens for GET requests on a specific URL path (e.g., "/greet").
- Inside the route handler, send a plain text response with a greeting message (e.g., "Hello, Express!") using the `res.send()` method.

Part 3: Running the Application

7. Start your Express.js application using the `node` command. For example:

```
node app.js
```

8. Open a web browser to make a GET request to the specified URL path. For example:

- If you've defined the route as "/greet", access `http://localhost:3000/greet` in your web browser.

B) Handling Different HTTP Response Status Codes in Express.js

In this test, you will create an Express.js application that handles different HTTP response status codes for various routes. You'll set up routes with specific status codes and customize the responses accordingly.

Part 1: Setup the project on your computer

1. Create a new directory for your Express.js project.

সমস্যায় পড়েছি?



স্বাগতম

Asadullah Ahmed



৪৪ ক্লাস জয়েনিং

আমার কোর্স

৫৫ রেকর্ডিং

৫৫ রিসোর্স

Skillable

৫৫ ট্যালেন্টপুল



প্রোফাইল



2. Navigate to your project directory in your terminal.
3. Initialize your project by running ``npm init`` and follow the prompts to create a ``package.json`` file.
4. Install the ``express`` module by running ``npm install express`` in your project directory.

Part 2: Express Application

5. Create an ``app.js`` file in your project directory.
6. Write a Node.js program that does the following:
 - Import the ``express`` module and create an instance of the Express application.
 - Define a route handler that listens for GET requests on a specific URL path (e.g., ``/success``).
 - Inside the route handler, send a response with a custom message using the ``res.send()`` method and set the HTTP status code to indicate success (e.g., 200 OK).
 - Define another route handler for a different URL path (e.g., ``/notfound``).
 - Inside this route handler, send a response with a message indicating that the resource was not found and set the HTTP status code to indicate a not found error (e.g., 404 Not Found).

Part 3: Running the Application

7. Start your Express.js application using the ``node`` command. For example:
``node app.js``

C) Setting Cookies in Express.js

In this test, you will create an Express.js application that sets cookies with various values and options. You'll use the built-in `res.cookie()` method to accomplish this task.

Part 1: Setup the project on your computer

সমস্যায় পড়েছি?



স্বাগতম

Asadullah Ahmed



৪৪ ক্লাস জয়েনিং

আমার কোর্স

৫৫ রেকর্ডিং

৬৬ রিসোর্স

Skillable

৭৭ ট্যালেন্টপুল



প্রোফাইল



1. Create a new directory for your Express.js project.
2. Navigate to your project directory in your terminal.
3. Initialize your project by running `npm init` and follow the prompts to create a `package.json` file.
4. Install the `express` module by running `npm install express` in your project directory.

Part 2: Express Application

5. Create an `app.js` file in your project directory.

6. Write a Node.js program that does the following:

- Import the `express` module and create an instance of the Express application.
- Define a route handler for a GET request to the root URL path (`/`).
- Inside the route handler, use the `res.cookie()` method to set the following cookies:
 - A cookie named "username" with the value "JohnDoe."
 - A cookie named "language" with the value "en".

Part 3: Running the Application

7. Start your Express.js application using the `node` command. For example:

```
node app.js
```

8. Open a web browser and access `http://localhost:3000`. You can check the browser's developer tools to inspect the cookies.

D) Clearing Cookies in Express.js

সমস্যায় পড়েছি?



স্বাগতম

Asadullah Ahmed



৪৪ ক্লাস জয়েনিং

আমার কোর্স

৬৫ রেকর্ডিং

৬৫ রিসোর্স

Skillable

৬৫ ট্যালেন্টপুল



৬৫ প্রোফাইল



In this test, you will create an Express.js application that allows users to clear cookies. You'll set up a route that, when accessed, clears specific cookies and provides feedback to the user.

Part 1: Setup the project on your computer

1. Create a new directory for your Express.js project.
2. Navigate to your project directory in your terminal.
3. Initialize your project by running `npm init` and follow the prompts to create a `package.json` file.
4. Install the `express` module by running `npm install express` in your project directory.

Part 2: Express Application

5. Create an `app.js` file in your project directory.
6. Write a Node.js program that does the following:

- Import the `express` module and create an instance of the Express application.
- Define a route handler for a GET request to a specific URL path (e.g., "/clearcookies") that clears specific cookies.
- Inside the route handler, use the `res.clearCookie()` method to clear specific cookies by name.
- Send a response to the user indicating that the cookies have been cleared.

Part 3: Running the Application

7. Start your Express.js application using the `node` command. For example:
`node app.js`
8. Open a web browser and access the URL for the route you defined (e.g., `http://localhost:3000/clearcookies`) to clear the cookies.

সমস্যায় পড়েছি?



স্বাগতম

Asadullah Ahmed



৪৪ ক্লাস জয়েনিং

আমার কোর্স

৬৫ রেকর্ডিং

৬৫ রিসোর্স

iS Skillable

৬৫ ট্যালেন্টপুল



৬৫ প্রোফাইল

Submission Instruction :

Create a number pagination and submit the code here. Submit only app.js codes. Paste the generated codes from your computer here respectively.

Examples:**1. Your code.....****2. Your code.....****Write your answer here...**

Write here...

SUBMIT ANSWER

সমস্যায় পড়েছি? !



স্বাগতম

Asadullah Ahmed



৪৪ ক্লাস জয়েনিং



আমার কোর্স



রেকর্ডিং



রিসোর্স



Skillable



ট্যালেন্টপুল



প্রোফাইল

সমস্যায় পড়েছি?