

Assignment Questions 10

💡 **Q.1** Explain Middlewares in NodeJS ?

💡 **Q.2** Why use Express Over NodeJS ?

💡 **Q.3** What are REST API ?

💡 **Q.4** What is the use of MongoDB?

💡 **Q.5** What is Mongoose and how does it relate to MongoDB?

💡 **Q.6** Build a Server Using Http Module NodeJS with api endpoints for Getting Products Data

‘/’ → send msg as “Welcome to Men & Women Dummy Data”

‘/men’ → send 10 products data of men

‘/women’ → send 10 products data of women

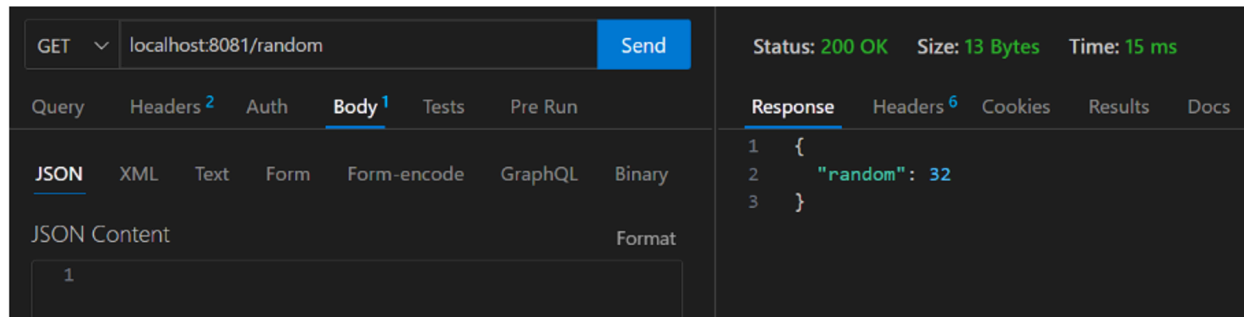
‘/other’ → send response as page not found

Note : You Can any Product Data

The screenshot shows a REST client interface. On the left, the 'Body' tab is selected, showing 'JSON' content. The request is a GET to 'localhost:8081/men'. On the right, the 'Response' tab is selected, showing a 200 OK status with a 1.11 KB size and 10 ms time. The response body is a JSON array with one object representing a product.

```
1  [
2    {
3      "id": "1",
4      "name": "Men's T-Shirt",
5      "brand": "ABC",
6      "price": 19.99,
7      "color": "Blue",
8      "size": "M",
9      "category": "Clothing"
10   },
11 ]
```

💡 **Q.7** Build a Server that Generate Random Number Using Express ‘/random’ → send random number in response {random:10}



💡 **Q.8** Create a Backend For Counter Web App with Api Endpoint for Managing Counter Using Express

- `'/'` → send counter data as `{counter:counter}`
- `/increment` → increment counter by 1 and send in response latest data as `{counter:counter}`
- `/decrement` → decrement counter by 1 and send in response latest data as `{counter:counter}`

Note: Using Express is Mandatory for this Question

💡 **Q.9** You are tasked with building the backend for a Todo web application. The application allows users to create, update, and delete tasks in their to-do list with Api Endpoint as mentioned below

`/` --> send all the todos which are created by user

`/add` --> to create todo create a middleware to validate user providing all the mandatory input (task), if user doesn't provide task as input send resp as validation failed

`/update/:id` --> to update todo with Todo Id, after updating send resp as `{msg:"todo update successfully",data:allTodos}`

`/delete/:id` --> to update todo with Todo id, after deleting send resp as `{msg:"todo deleted",data:allTodos}`

💡 **Q.10 Assignment:** Full Stack Blog Web Application (PW Blog) Development using **MERN Stack**

Description: You have been assigned the task of building a full stack blog web application using the MERN (MongoDB, Express.js, React.js, Node.js) stack. The application will allow users to create, read, update, and delete blog posts. Users can also register, log in. The backend will handle data storage and retrieval, while the frontend will provide a user-friendly interface for interacting with the application.

Requirements:

1. **User Authentication:** Implement user authentication and authorization mechanisms to ensure secure access to the application. Users should be able to register, log in, and log out.
2. **Blog Post Management:** Create endpoints to manage blog posts, including the following functionalities:
 - Display a list of blog posts
 - View detailed information for each blog post
 - Create new blog posts
 - Update existing blog posts
 - Delete blog posts
3. **Note:** User Should be able to delete his own blog Only
4. **Database Design:** Design the database schema to store blog posts, user details. Use MongoDB to store and retrieve data efficiently.
5. **API Design:** Define a RESTful API to expose the backend functionalities. Ensure the API endpoints are well-structured, follow best practices, and provide appropriate error handling.
6. **Frontend Development:** Create an intuitive user interface using React.js to interact with the backend API. Implement features such as blog post listing, User Page where he can see his blog post, and blog creation and editing, Login & SignUP Page
7. **Deployment:** Deploy the application to a hosting environment (e.g., Vercel, Netlify) and ensure it is accessible to users.

Deliverables:

- A fully functional full stack blog web application built using the MERN stack.
- Source code for the backend and frontend components, properly organized and commented for clarity and maintainability.
- Documentation ([readme.md](#)) describing the API endpoints, frontend components, and their usage.
- Any additional instructions or dependencies required to run the application.

Please note that the design and aesthetics of the application's user interface will be handled separately. Your focus should be solely on building the backend and frontend components necessary for the blog functionalities.

Api Endpoints 📌

Register

Method: POST
Endpoint: /user/register
Payload:
{ "name": "user_name", "email": "user_email", "password": "user_password" }

LOGIN

Method: POST
Endpoint: /user/login
Payload:
{ "email": "user_email", "password": "user_password" }

Add Blog

Method: POST
Endpoint: /blog/add
Payload:
{ "title": "blog_title", "desc": "blog_description", "img": "blog_img_url", "content": "blog_content" } headers:
{ "authorization": bearer: \${token} }

Get all Blog

Method: Get
Endpoint: /blog/
Response: All blog Data

delete Blog By Id

Method: DELETE
Endpoint: /blog/id
headers:
{ "authorization": bearer: \${token} }
Note: The blog will be deleted if You are the author of the blog

Update blog by Id

Method: PATCH
Endpoint: /blog/id
payload: whatever you want to update
headers:
{ "authorization": bearer: \${token} }
Note: The blog will be updated if You are the author of the blog

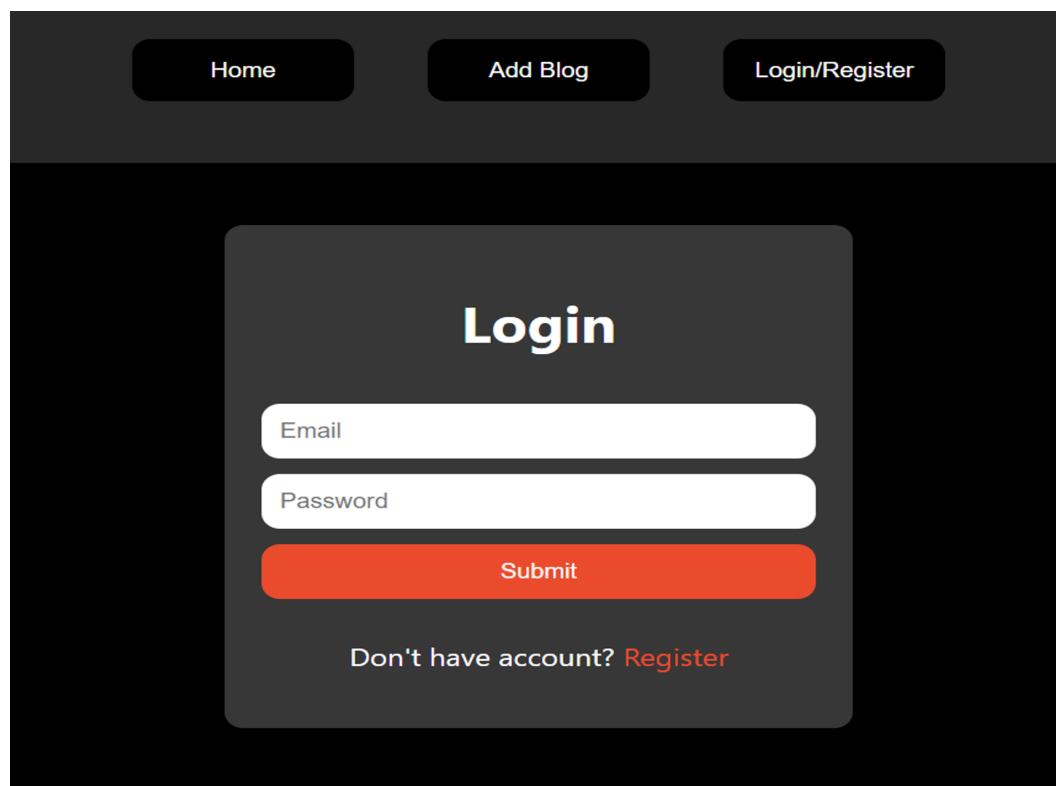
Get Specific Blog By Id

Method: GET
Endpoint: /blog/blogID
Response: blog Data

Get blogs of Specific User

Method: GET
Endpoint: /myblog/authorId
headers:
{ "authorization": bearer: \${token} }

1. Login Page



The image shows a login page with a dark background. At the top, there is a navigation bar with three buttons: "Home", "Add Blog", and "Login/Register". Below the navigation bar, there is a central login form. The form has a title "Login" in white text. Below the title, there are two input fields: "Email" and "Password". Below the input fields, there is a red "Submit" button. At the bottom of the form, there is a link that says "Don't have account? Register".

2. Register Page

[Home](#)
[Add Blog](#)
[Login/Register](#)

Register

Submit

Don't have account? [Login](#)

3. Home Page

[Home](#)
[Add Blog](#)
[MyBlog](#)
[LogOut](#)
Welcome

Step By Step Guide to Buy Courses on PW Skills

As the world becomes increasingly digital, programming and coding skills are in high demand. Whether you are a student looking to start a career in IT or a professional looking to upskill, this guide provides a step-by-step process for buying courses on PW Skills.

Read

Full Stack Web Development Course and Certification Programs

Typically, course designers design a Web Development Course to teach individuals the necessary skills and knowledge to develop web applications. This course covers a wide range of topics, including HTML, CSS, JavaScript, and more.

Read

The Web Development Handbook: A Step-by-Step Guide to Success

Web development refers to all of the labor that goes into creating a website. Everything from HTML and code to scripting, network settings, and more. This handbook provides a comprehensive guide to web development, from the basics to advanced techniques.

Read

4. Add Blog

HomeAdd BlogMyBlogLogOutWelcome

Write Your Blog

Enter Title of Your Story

Enter Url of Blog Image

Type Short Description for your Blog

Type Your Blog

Publish Blog

5. My Blog Page

HomeAdd BlogMyBlogLogOutWelcome


Top 20

Website Development

Language Trends

In 2023

SALARY SCOPE



Top 20 Website Development
Language Trends in 2023 With
salary Scope

A solid online presence is vital for
organizations and individuals in
today's digital world. The current
field of website building is evolving

Read

Edit

Delete

6.Edit Blog/Update Blog



Top 20 Website Development Language Trends in 2023 With salary Scope

A solid online presence is vital for organizations and individuals in today's digital world. The current field of website building is evolving due to

Read

Edit

Delete

Top 20 Website Development

A solid online presence is vi

<https://pwskills.com/blog/wp>

A solid online presence is vi

Update

Close