Team-61 (ReactRebels)

"Go-Fit": Health and Wellness Hub

Frontend:-

- React js
- HTML
- CSS

Backend:-

- Node js
- Express js

Database :-

-MongoDB(mongoose)

react js globally install

- npm create-react-app -g

reactjs app created

- D:\WPT mini Project Team@61>npx create-react-app go-fit

Download dependancies

- npm i react-bootsrap bootstrap axios react-router-dom react-router-bootstrap
- npm i axios

Creating Server

- create new folder
- download package.json by using *nmp i*
- next download *npm i express mongoose* for express and mongodb dependancies
- create index.js file(all apis methods define here by connecting server to database)
- and usermodule.js for server data, creating a new schema and stored in new object and that object lastily store in class and which class is import in index.js file
- to run file *node index.js*
- npm i bcryptjs
- npm i jwtwebtoken
- npm i

- npm express
- npm mongoose

Session Management:

- Session management involves maintaining user sessions during their interactions with a web application. Sessions are often managed using cookies or tokens to identify and authenticate users across multiple requests.

CORS (Cross-Origin Resource Sharing):

- CORS is a security feature implemented by web browsers that controls how web pages in one domain can request and interact with resources on another domain.

Hooks, State, Props (React):

- In React, hooks are functions that enable functional components to use state and lifecycle features. `useState` is a hook for managing state, and components receive data through props.

JWT Token (JSON Web Token):

- JWT is a compact, URL-safe means of representing claims between two parties. It is often used for authentication and information exchange between parties.

Navigate Function:

- The navigate function is commonly used in navigation libraries (e.g., React Router) to programmatically navigate between different views or pages in a single-page application.

Axios:

- Axios is a JavaScript library used for making HTTP requests. It works both in the browser and in Node.js environments.

Express (Node.js framework):

- Express is a web application framework for Node.js. It simplifies the process of building robust and scalable web applications and APIs.

Node.js:

- Node.js is a JavaScript runtime built on the V8 JavaScript engine. It enables server-side development using JavaScript.

MongoDB and Mongoose.js:

- MongoDB is a NoSQL database, and Mongoose.js is an ODM (Object Data Modeling) library for MongoDB and Node.js. Mongoose simplifies interaction with MongoDB by providing a schema-based solution.

List and Keys (React):

- In React, a list is often rendered using the `map` function. Keys are used to help React identify which items have changed, been added, or been removed.

Params (React Router):

- In React Router, params are used to capture dynamic segments of the URL. For example, in the route path "/users/:id", ":id" is a parameter capturing the user's ID.

Bcrypt:

- Bcrypt is a password-hashing function commonly used for securely storing passwords. It is designed to be slow and computationally intensive, making it resistant to brute-force attacks.

Schema and Collection (MongoDB):

- In MongoDB, a schema is a blueprint for the organization of data within a collection. A collection is a grouping of MongoDB documents.

APIs (Application Programming Interfaces):

- APIs define how software components should interact. They enable communication between different systems, allowing them to exchange data and functionality.

HTTP Methods:

- HTTP methods (GET, POST, PUT, DELETE, etc.) define the actions that can be performed on resources identified by URIs.

Ports (e.g., 3000) and IP Address (e.g., 127.0.0.1):

- Ports and IP addresses are used to specify network destinations for communication. Localhost (127.0.0.1) and port numbers (e.g., 3000) are often used for development servers.

.jsx (JavaScript XML) Extension:

- The .jsx file extension is commonly used for React components. JSX is a syntax extension for JavaScript recommended for use with React.

`.env`:

- The `.env` file is often used to store environment variables, including sensitive information, in a project.

It looks like your question includes various topics related to web development, including HTML, JavaScript, React, APIs, databases (MongoDB and MySQL), routing, state management, HTTP requests, and more. Let me provide brief explanations for each of these topics:

HTML - Linking Tag:

- The `<a>` tag is used to create hyperlinks. It is commonly used to link to other web pages or resources.

```
<a href="https://example.com">Visit Example.com</a>
```

JavaScript - `async` and `await`:

- `async` and `await` are used to work with asynchronous code in JavaScript. The `async` keyword is used to define asynchronous functions, and `await` is used to wait for a Promise to resolve before continuing.

```
async function fetchData() {
  const response = await fetch('https://api.example.com/data');
  const data = await response.json();
  console.log(data);
}
```

Events - `onchange`, `onsubmit`, `onclick`:

- These are event handlers in HTML and JavaScript.
  - `onchange`: Fires when the value of an input element changes.
  - `onsubmit`: Fires when a form is submitted.
  - `onclick`: Fires when an element is clicked.

Service API - Axios (HTTP Requests - GET, PUT):

- Axios is a JavaScript library for making HTTP requests. It supports various HTTP methods, including GET and PUT.

```
// Example GET request
axios.get('https://api.example.com/data')
  .then(response => console.log(response.data));

// Example PUT request
axios.put('https://api.example.com/data/123', { newData: 'updated value' })
  .then(response => console.log(response.data));
```

Validation of Signup:

- Signup validation typically involves checking that user-provided information meets certain criteria, such as valid email format, strong password, etc. Frontend and backend validation are both important.

MongoDB and MySQL Difference:

- MongoDB is a NoSQL database, while MySQL is a traditional relational database. MongoDB stores data in flexible, JSON-like documents, while MySQL uses tables with a fixed schema.

React - Browser Router, Routes, Route:

- In React, `BrowserRouter` provides the navigation context for `Route` components. `Route` components render UI based on the current URL, and multiple `Route` components are typically wrapped in a `Routes` component.

  ```
  import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';

  function App() {
    return (
      <Router>
        <Routes>
          <Route path="/" element={<Home />} />
          <Route path="/about" element={<About />} />
        </Routes>
      </Router>
    );
  }
  ```

Token:

- In the context of web development, a token is often used for authentication and authorization. Examples include JSON Web Tokens (JWTs) and access tokens.

Utils:

- "Utils" is a common term for utility functions or modules that provide reusable functionality across an application.

Local Storage:

- Local Storage is a web storage solution that allows data to be stored persistently in a web browser. It has a simple key-value pair interface.

  ```
  // Storing data
  localStorage.setItem('user', JSON.stringify({ id: 1, name: 'John' }));

  // Retrieving data
  const user = JSON.parse(localStorage.getItem('user'));
  ```

React - Component (Function and Class Component):

- In React, components are the building blocks of a user interface. There are two main types: Function Components (stateless) and Class Components (stateful).

  ```
  // Function Component
  function MyFunctionComponent() {
    return <div>Hello, Function Component!</div>;
  }
  ```

```
// Class Component
class MyClassComponent extends React.Component {
  render() {
    return <div>Hello, Class Component!</div>;
  }
}
```

HTTP Request Header:

- HTTP request headers contain additional information about the request, such as the browser type, accepted content types, and more.

Postman and Thunder Client:

- Postman and Thunder Client are tools for testing and debugging API requests. They allow you to send HTTP requests and view the responses.

Response & Request:

- In the context of HTTP, the response is what the server sends back after receiving a request. The request is what the client sends to the server.

`require` Keyword:

- In Node.js, `require` is used to import modules or files.

Ajax JSON:

- AJAX (Asynchronous JavaScript and XML) is a technique for making asynchronous requests to a server. JSON is commonly used as the data format for these requests and responses.

This is a broad overview of the topics you mentioned. If you have specific questions or need more in-depth information on any of these topics, feel free to ask!

Ajax is a client-side technology used for making asynchronous requests, while Express is a server-side framework for building web applications and APIs. They work together to enable dynamic and interactive web development.