Chapter– 13

YESMENTOR E-LEARNING APPLICATION

Ch.Id:-ASU/GRF/EB/RAPOET/2022/Ch-13
DOI: https://doi.org/10.52458/9789391842765.2022.eb.grf.asu.ch-13

[1]NISHANT JAIN
[1]Apeejay Stya University, Sohna, Gurugram

[2]Mr. JERRY JOY
[2]Industry Mentor

[3]Dr. GARIMA SHARMA
[3]Apeejay Stya University, Sohna, Gurugram

Yes Mentor, founded in March 2021. YesMentor Tech Education is a platform that focuses on providing quality education and takes their clients closer to their dream career.

Providing the utmost excellence for the overall preparation of exams with ease and excellence is their primary objective. In the current scenario, where aspirants are apprehensive about the proper resources to prepare and assess themselves. This is where Yes Mentor comes into action, to help them fulfil their aspirations, the platform will be a powerful resource-generating tool that will cut time and give them an advantage of weeks for a better learning experience. All their needs will be sufficed by this one platform, encouraging them to go deeper into the art of learning. (Website: https://yesmentor.in)

Yes Mentor was one of the Offline coaching institutes for the aspirant of civil services. During the lockdown they were facing the issues in teaching, hence they decide to create an Ed Tech platform for all the aspirants. The application should consist of the courses, Test series, Live Tests with analytics calculation feature, live lectures, recorded video lectures etc. So as per requirement we agreed upon creating an admin panel, user dashboard, and a website using react js and next js, which is powered by Nodejs backend.

Moreover, the payments for the content should be captured by the application only, hence we implement Razorpay for the payment system. Razorpay provides payment option via UPI, Net banking, Debit card, Credit card, and mostly all types of wallets.

FRONTEND

For creating admin panel and user dashboard we used React JS as a framework whereas for creating website we have used Next Js.

What is React-Js?

A JavaScript library for building user interfaces. (https://reactjs.org/).

React is a JavaScript library created by Facebook. It is a tool for building UI components.

What is Next-Js?

Next.js is an open-source web development framework built on top of Node.js (https://nextjs.org/) enabling React-based web applications functionalities such as server-side rendering and generating static websites. React documentation mentions Next.js among "Recommended Toolchains" advising it to developers as a solution when "Building a server-rendered website with Node.js". Where traditional React apps can only render their content in the client-side browser, Next.js extends this functionality to include applications rendered on the server-side.

Backend

We used Node Js as a backend service for creating API Exposed these apis so that it can be consumed by front end made in react and nextjs.

What is Nodejs?

As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications. In the following "hello world" example, many connections can be handled concurrently. Upon each connection, the call back is fired, but if there is no work to be done, Node.js will sleep.(https://nodejs.org/)

What is TypeScript?

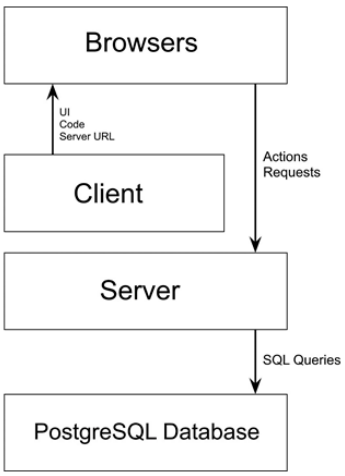
TypeScript is a strongly types programming language that build on JavaScript, giving users a better tooling at any scale. (https://www.typescriptlang.org/)

TypeScript adds additional syntax to JavaScript to support a tighter integration with your editor. Catch errors early in your editor.

TypeScript code converts to JavaScript, which runs anywhere JavaScript runs: In a browser, on Node.js or Deno and in your apps.

TypeScript understands JavaScript and uses type inference to give you great tooling without additional code.

Architecture of the project



(Figure: 3.3)

React-Js being the front-end framework it would communicate with the backend which is REST which is a wrapper over Express framework and Express framework internally uses ORM (Object Relational Model) to communicate with the database. And all this explains the above diagram.

METHODOLOGY

Since the deadlines were tight and the project was meant to be delivered at a fast pace within a month, so we decided to go with fast and short sprints.

Some of the tools and IDEs which we used during the project were:

- GitHub for Version Control and collaboration
- VS code for development

Monthly work breakdown

- **Month 1**
  - ✓ Rest Apis were created as per requirements, like basic CRUD, create read update and delete for news model.o https://expressjs.com/ was used a wrapper over Express for APIS and lot more
  - ✓ All the major requirement gathering were taken from designer
  - ✓ All requirements were given to designer and UI mocks were created for them

- **Month 2**
  - ✓ After the design mocks were given to engineers, they were converted into code, react was the framework, custom CSS was used and JSX was used to html elements
  - ✓ What is JSX? (https://reactjs.org/docs/introducing-jsx.html)
- **Month 3**
  - ✓ Rest Apis were tweaked as per feedback and changes requested
  - ✓ A separate model (in terms of ORM or table in terms of PostgreSQL was created) because client wanted to put local ads into his website, and he would take commission from those advertisers.
  - ✓ UI screens were also improved as the same was shared to client. 404 page not found, and search functionality was added.
- **Month 4**
  - ✓ Finishing up UI screens and minor bugs and put the app into production
  - ✓ Deployed the backend to AWS server (Amazon web services) https://aws.amazon.com/ and Linode Server (https://www.linode.com/)
  - ✓ Deployed frontend React app into Hostinger https://www.hostinger.in/
  - ✓ Domain names were purchased and admin panel, user dashboard and website were redirected to domain and sub-domains
  - ✓ See Live app at https://yesmentor.in/
- **Month 5**
  - ✓ React best practises like using API content which would reduce api calls was used.
  - ✓ It increased the UX of the web app.
  - ✓ Firebase Analytics was added to monitor the stats and events.
- **Month 6**
  - ✓ Admin panel was customized as client was finding it hard to operate it on.
  - ✓ Server-side rendering was experimented so that web pages preview could be added when shared across social media platform.

GitHub is a Git repository hosting service, but it adds many of its own features. While Git is a command line tool, **GitHub** provides a Web-based graphical interface. It also provides access control and several collaboration features, such as a wikis and basic task management tools for every project.

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

Design -Programming and Implementation Details

About React

Basics of React and why react?

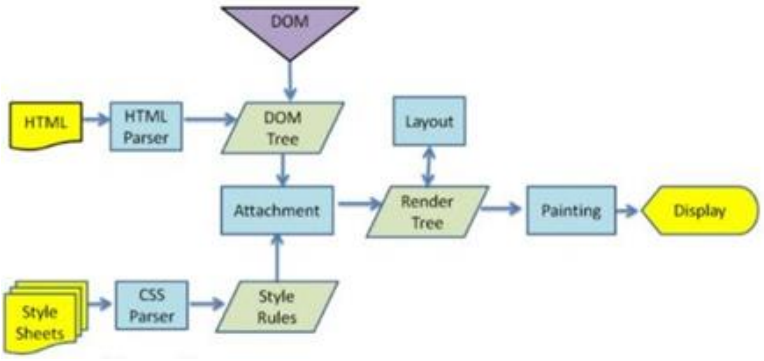
- **Declarative: React makes it** painless to create interactive UIs. Design simple views for each state in your application and react will efficiently update and render just the right components when your data changes. Declarative views make your code more predictable and easier to debug.
- **Component-Based:** Build encapsulated components that manage their own state, then compose them to make complex UIs. Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM.
- **Learn Once, Write Anywhere:** We don't make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code. React can also render on the server using Node and power mobile apps using React Native.

Virtual DOM - the game changer

ReactJS does not update the real DOM directly, but it updates the Virtual DOM. DOM stands for Document Object Model. As per w3.org DOM defines the logical structure of documents and the way a document is accessed and manipulated.

This causes a great performance benefit for ReactJS. Here, we will try to understand why updating the real DOM is slow and how updating Virtual DOM increase the performance?

Why updating Real DOM is slow?



Updating a DOM is not slow, it is just like updating any JavaScript object; then what exactly makes updating real DOM slow?

Rendering engines which are responsible for displaying or rendering the webpage on the browser screen parses the HTML page to create DOM. It also parses the CSS and applies the CSS to the HTML, thus creating a render tree, this process is called as attachment. Layout process gives exact coordinates to each node of the render tree, where the node gets painted and displayed.

So, when we do, document.getElementById('elementId').innerHTML = "New Value" Following thing happens:

- The browser must parse the HTML
- It removes the child element of elementId
- Updates the DOM with the "New Value"
- Re-calculate the CSS for the parent and child
- Update the layout i.e., each elements exact coordinates on the screen
- Traverse the render tree and paint it on the browser display

Recalculating the CSS and changing layouts uses complex algorithms and they affect the performance. Thus, updating a real DOM does not involve just updating the DOM but, it involves a lot of other processes. Also, each of the above steps runs for each update of the real DOM i.e., if we update the real DOM 10 times each of the above step will repeat 10 times. Therefore, updating Real DOM is slow.

How does Virtual DOM solve this problem?

What is a virtual DOM?

Virtual DOM is an in-memory representation of real DOM. It is a lightweight JavaScript object which is a copy of Real DOM.

Updating virtual DOM in ReactJS is faster because ReactJS uses

- Efficient diff algorithm
- Batched update operations
- Efficient update of subtree only
- Uses observable instead of dirty checking to detect the change

ReactJS using diff algorithm compares both the Virtual DOM to find the minimum number of steps to update the Real DOM.

Finding the minimum number of modifications between two trees have complexity in the order of O (n^3). But react uses a heuristic approach with some assumptions which makes the problems to have complexity in the order of O(n).

React Context

Context provides a way to pass data through the component tree without having to pass props down manually at every level.

In a typical React application, data is passed top-down (parent to child) via props, but such usage can be cumbersome for certain types of props (e.g. locale preference, UI theme) that are required by many components within an application. Context provides a way to share values like these between components without having to explicitly pass a prop through every level of the tree.

When to Use Context

Context is designed to share data that can be considered "global" for a tree of React components, such as the current authenticated user, theme, or preferred language. For example, in the code below we manually thread through a "theme" prop to style the Button component:

Usage

const myContext = React.createContext(deafultValue);

Creates a Context object. When React renders a component that subscribes to this Context object it will read the current context value from the closest matching *Provider* above it in the tree.

The *defaultValue* argument is **only** used when a component does not have a matching Provider above it in the tree. This default value can be helpful for testing components in isolation without wrapping them. Note: passing *undefined* as a Provider value does not cause consuming components to use *defaultValue*.

Context. Provider

<MyContext.Provider value={}>

Every Context object comes with a Provider React component that allows consuming components to subscribe to context changes.

The Provider component accepts a value prop to be passed to consuming components that are descendants of this Provider. One Provider can be connected to many consumers. Providers can be nested to override values deeper within the tree.

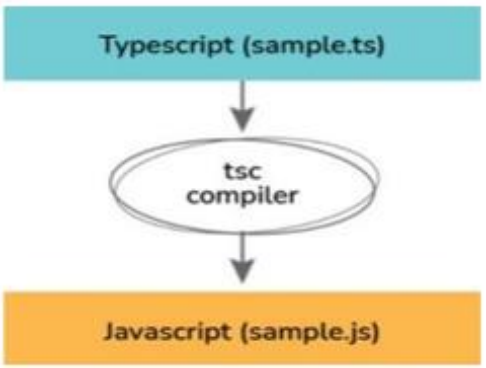
All consumers that are descendants of a Provider will re-render whenever the Provider's value prop changes. The propagation from Provider to its descendant consumers (including .contextType and useContext) is not subject to the shouldComponentUpdate method, so the consumer is updated even when an ancestor component skips an update.

About Backend Stack

- **Node Js:** Node.js is a virtual machine that uses JavaScript as its scripting language and runs Chrome's V8 JavaScript engine. Basically, Node.js is based on an event driven architecture where I/O runs asynchronously making it lightweight and efficient. It is being used in developing desktop applications as well with a popular framework called electron as it provides API to access OS-level features such as file system, network, etc.
- **TypeScript:** TypeScript is an open-source language developed by Anders Hejlsberg at Microsoft. It's a statically typed superset of JavaScript that compiles to plain JavaScript. It runs on any browser, host, and operating system. That means all valid JavaScript code is also TypeScript code. It offers advanced features such as IntelliSense, code completion, safe refactoring's, etc.
- **Why TypeScript?**

As JavaScript projects grow, they become difficult to maintain. There are a few reasons for this. First, JavaScript was never designed to build large-scale applications. Its original purpose was to provide small scripting functionality for a web page.

Until recently, it didn't provide tools and constructs for structuring large projects, such as classes, modules, and interfaces. Also, JavaScript is dynamically typed. It doesn't support features such as IntelliSense.



TypeScript files use a .ts extension, in contrast to the .js extension used by the JavaScript files. Since TypeScript is a superset of JavaScript, all valid JavaScript code is a valid TypeScript code, and renaming a .js file to .ts won't change anything.

- **PostgreSQL**
- PostgreSQL is an object-relational database management system (ORDBMS) first introduced in the department of Computer Science, University of California. It is an open-source database management system, and its functions are written in C

language. It is rich in features, highly extensible, and super easy to learn.

- PostgreSQL can run on all operating systems. It is ACID compliant (Atomicity, Consistency, Isolation, and Durability), is SQL compliant, and supports JSON and some other NoSQL features like native XML support. It additionally offers Multi-Version Concurrency Control (MVCC), which means that several users can concurrently work on a database. You will face no difficulties in using PostgreSQL as it has a great and nice community that is willing to help you whenever you need it.
- PostgreSQL supports advanced data types such as arrays, hstore, and user-defined data types. It is highly customizable since you can customize it by developing plugins to make the DBMS fit your requirements. Overall, it provides great performance, functionalities, and security and at the same time, it is also user-friendly.

Why to use PostgreSQL?

- PostgreSQL is an open-source and free object-relational database management system.
- Users can create custom functions made with programming languages like C/C++ or Java.
- MVCC allows many concurrent users to work on one system.
- It is feature-rich, scalable, and supports modern applications like XML, JSON, etc.
- Supports foreign keys for efficient storage of data.
- Stored procedures/functions for complex operations.
- PostgreSQL has synchronous replication where data is simultaneously replicated from source storage (master) to target storage (slave). It makes the replication process easy.

Modules Descriptions

- **Trello:** Trello is the visual tool that empowers our team to manage any type of project, workflow, or task tracking. Project manager assigns the task with the deadline to every developer here in Trello. Developers gets the notification via mail. Developers shifts the card to the different columns as par the status of the task.

  For example, if the task in in progress, then the card should be shifted to the work in progress column and if it is under testing it should be shifted to the testing column. This way the project manager and developers can track the work.
- **Admin Panel (build with React js):** This is the screenshot for the admin panel.

This panel is only accessible to the admins (https://admin.yesmentor.in). Using this admin panel, admins can track the activity of the students, Create new courses, Assignments, test series for practice and the live test series.

Using this panel, an admin can manage the users and staff members. An admin can add, delete, update users and staff members from the admin panel.

Admin can also track the accounts section of the company from this panel, they can see the fee details of the users, salary details of the users, they can send notification or mail for reminder, if there are some dues left.

We have a blogs page in our website, in which users can read the blogs, like the blogs and can share their comments on it.

- **User Dashboard (build with React js)**

This is the screenshot of the user's Dashboard (https://user.yesmentor.in).

This dashboard is only accessible to those users who are registered in the database. So, for accessing the content from the user's dashboard a user must sign up using his valid email id or phone number. Using this dashboard, a user can get access of all the courses, Test series, video lectures, live lectures etc. A user can also book a free counselling session using this dashboard.

In user dashboard, there are some special courses also. Those courses are named as Pro courses. These pro courses are for longer period (around 2-3 years). Fee for the pro courses is high then the normal courses, hence for make it available and affordable to everyone we have introduce a dynamic fee payment method using which a user can pay the instalment of his choice and the content will be unlocked accordingly. This is the screenshot for the video player in pro courses. Here users can play there recorded and live lectures. On the right side of the window all the content of the course is mentioned, if the video is watched by then the checkbox will get automatically filled.

Below this video player there are three options:

1. **Description**- it shows the description of the course.
2. **Notes**- it shows all the list of pdf notes (Topic wise).
3. **Assignments**- it shows the list of chapter wise assignments.

These are the screenshots of the Notes and Assignments. We have achieved this pdf view in react user "react-pdfjs" library.

- **Website (build with Next js)**

Now comes the website of the company (https://yesmentor.in). This website is built with Nextjs. Next js is a react framework. Next js is very known for the search engine optimization. Since we have the blogs and podcast facility on the website, we

must take care of the search engine optimization. Hence, we used Nextjs for building website. This website is a showcase of all the products and services provides by the Yes Mentor Pvt. Ltd. Website has a separate blogs section; this blogs section is known as issues of the week. The blogs which are posted here are related to the current affairs. Users can search the blogs by name and, they can also sort the blogs by date or by language. These are the screenshots from the mentor talks. Mentor talks are the Recorded videos in which an admin talks about a particular topic (for e.g.: Preamble). These videos are uploaded by the content team on every Wednesday.

Users can also like and share their view about the topics in the comment sections.

- **Postman Collections and Documentation of the apis**

This is the window of the postman. This is divided into three sections. First section shows the list of all the apis. Second sections show the testing area where we can test the API. The third sections show the documentation of the API with is updated by the developer. In the documentation window I have updated the endpoint of the API, and all the responses which includes (success response + error response) so that the frontend developer can get an idea of handling the response.

- **Deployment Of backend**

In Yes Mentor we have deployed two servers one for the production and second for the development. The production server was deployed on the AWS EC2 instance. Whereas the Backend server is deployed in the Linode instance.

- **Deployment of frontend application**

We have deployed the frontend applications in Hostinger as well as Linode server.

For react applications we have used Hostinger but for deploying Nextjs we need a server which can render html pages server side, hence we have used Linode server for it.

(Hostinger Screenshots) For deploying React application on Hostinger a tool is required known as FileZilla. We can upload the build folder directly in the subdomain folder shown in the FileZilla.

- **Load-Testing for Backend Server**

For testing the server load we have used a tool known as loader.io (https://loader.io). Loader.io helps to send bulk request simultaneously and provides the report for the success responses and failed responses. This way we can test the load capacity of the server.

CONCLUSION

This project has turned out to be a valuable experience as a software developer. The theory of how the development should be carried out and the things that can go wrong in the development process were experienced first time. Many little challenges were encountered and the understanding of why managing, and Web development is not an easy task grew during this project. This internship offered me opportunities to learn and develop in many areas and I gained a lot of experience in many technologies which were new for me such as, TypeScript, Postgres, Docker, Linode, AWS, Next.js etc.

The valuable experience has transformed me as an individual & taught me how to handle the workload in industry & how to make documentations and prototypes before beginning a project, what are the things you must keep in mind before taking up a project, such as target audiences, user-friendly, attractiveness etc.

Finally, this internship allowed me, autonomously, to acquire skills such as:

- Core Backend development
- Database management
- Scalability of the application
- Software Design
- Testing and debugging.
- Frontend web designing.
- Next.js Framework.
- Responsive design skills.
- Search engine optimization

BIBLOGRAPHY

- *https://nodejs.org/en/about/*
- *https://www.typescriptlang.org/*
- *https://reactjs.org/*
- *https://nextjs.org/#:~:text=Next.js%20gives%20you%20the%20best%2 0developer*
- *https://www.postgresql.org/#:~:text=PostgreSQL%20is%20a%20power ful*
- *https://trello.com/en/tour#:~:text=What%20is%20Trello%3F-,Trello%20is%20the,-visual%20tool%20that*