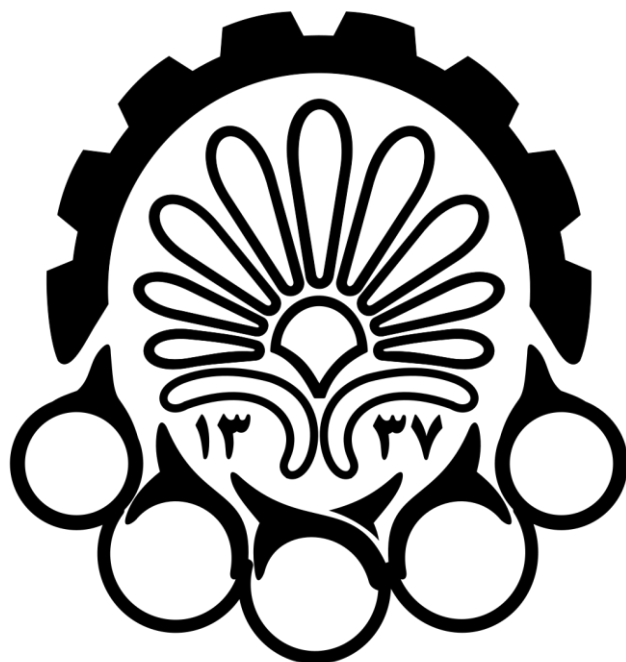


به نام خدا



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

تمرین سوم درس مهندسی اینترنت

پاییز ۱۴۰۰ – مهندس الوانی

سوال اول

درباره Virtual DOM در React تحقیق کنید و ذکر کنید که استفاده از این مفهوم، چگونه به سریع‌تر شدن React کمک می‌کند.

سوال دوم

در این تمرین قصد داریم یک شمارنده مشابه با تصویر زیر پیاده‌سازی کنیم:

Counter: 0

Reset

Start

Down Counting

این شمارنده به صورت زیر عمل می‌کند:

با فشردن دکمه start، شمارنده شروع به شمارش به سمت بالا می‌کند. مقدار این شمارنده در هر ثانیه یک واحد زیاد می‌شود (در حالت پیش‌فرض به صورت up counting عمل می‌کند).
با شروع شمارش، به جای دکمه start، دکمه pause نمایش داده می‌شود.
با فشردن دکمه pause شمارش متوقف خواهد شد و مجدداً دکمه start را خواهیم دید.
با فشردن دکمه down counting، شمارش شمارنده به صورت پایین رونده ادامه خواهد یافت و به جای این دکمه، up counting نمایش داده خواهد شد.
با فشردن دکمه reset، مقدار شمارنده صفر خواهد شد و به حالت اولیه باز خواهد گشت.
خلاصه‌ای از عملکرد این شمارنده را در تصاویر زیر می‌توانید مشاهده کنید.

در حالت اولیه:

Counter: 0

Reset

Start

Down Counting

پس از فشردن start

Counter: 19

Reset

Pause

Down Counting

پس از فشردن pause

Counter: 35

Reset

Start

Down Counting

پس از فشردن down counting و سپس start

Counter: 29

Reset

Pause

Up counting

و در نهایت، پس از فشردن reset

Counter: 0

Reset

Start

Down Counting

نکات پیاده‌سازی

1. می‌توانید به صورت دلخواه style دهی را انجام دهید اما حتما باید از روش css module استفاده کنید.
2. برای پیاده‌سازی این تمرین سعی کنید به جای تعریف دکمه‌های تکراری از یک کامپوننت reusable مانند Button استفاده کنید. همچنین می‌توانید کل شمارنده را در یک کامپوننت مجزا پیاده‌سازی کنید.
3. برای تعریف کامپوننت از هریک از روش‌های class component و یا functional component به دلخواه می‌توانید استفاده کنید.

سوال سوم

در این تمرین قصد داریم از سرویس [jsonPlaceholder](#) استفاده کنیم و دیتای آن را بر روی صفحه نمایش دهیم.

برای اینکار به دو کامپوننت User و UserList نیاز داریم. وظیفه کامپوننت UserList این است که در لحظه mount شدن، یک درخواست HTTP ارسال کند و سپس برای هر یک از userهای داخل نتیجه این درخواست، یک کامپوننت User رندر کند.

کامپوننت User باید propهای name, email, phone و website را داشته باشد و آن‌ها را نمایش دهد. همچنین این کامپوننت باید نوع propهای خود را با استفاده از [PropTypes](#) مشخص کند. برای propهای این کامپوننت مقدار default دلخواه نیز تعریف کنید.

در زمانی که UserList منتظر دریافت پاسخ از API است، باید حالت loading را نمایش دهد. برای نمایش loading می‌توانید از یک نوشته ساده و یا [loading gif](#) استفاده کنید. به محض رسیدن دیتا از سمت API، باید لیست Userها به جای loading نمایش داده شود.

نکات پیاده‌سازی

1. برای پیاده‌سازی UserList حتما از functional component استفاده کنید.
2. برای راحتی در ارسال درخواست HTTP، می‌توانید از پکیج‌های کمکی مانند [axios](#) کمک بگیرید.
3. برای گرفتن لیست کاربران می‌توانید از [این endpoint](#) استفاده کنید.
4. برای تعریف مقدار default یک prop از default props ها استفاده کنید.

سوال چهارم

در این تمرین قصد داریم یک ماشین حساب پیاده‌سازی کنیم. عملکرد این ماشین حساب مشابه عملکرد ماشین حساب windows است. پیاده‌سازی اولیه این برنامه را در [اینجا](#) می‌توانید مشاهده کنید.

| | | | |
|---|-----|---|---|
| 0 | | | |
| C | +/- | % | ÷ |
| 7 | 8 | 9 | x |
| 4 | 5 | 6 | - |
| 1 | 2 | 3 | + |
| 0 | | . | = |

در ابتدا باید کد اولیه را مطالعه کنید و سپس با استفاده از state ها و prop های مناسب، پیاده‌سازی را تکمیل کنید.

نکات پیاده‌سازی

1. نیاز به پیاده‌سازی کل دکمه‌های ماشین حساب windows نیست و صرفاً باید دکمه‌های مشخص شده را پیاده‌سازی کنید.
2. بعد از clone کردن پروژه، برای اجرا آن نیاز به زدن دستور npm i خواهید داشت (با این دستور تمام dependency ها بر روی سیستم شما نصب خواهد شد). سپس می‌توانید با دستور npm start برنامه را اجرا کنید.
3. شما می‌توانید style های اولیه را به صورت دلخواه تغییر دهید.

نحوه تحویل تمرین

پاسخ سوالات تئوری را در قالب یک فایل pdf به همراه کدهای خود، با نام «IE_HW3_StudentID.zip» در صفحه کورسز درس آپلود کنید.

در نهایت هر گونه سوال در مورد تمرین و بخش‌های آن را از طریق ایمیل ie.ce.aut@gmail.com مطرح فرمایید.

تندرست و موفق باشید