دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

برنامه نویسی وب

نیم‌سال اول ۹۸-۹۹

تمرین دو | JavaScript | مهلت ارسال: ۱۱:۵۹ جمعه ۱۷ آبان

به موارد زیر توجه کنید:

- پاسخ تمرینات را در کوئرا بارگذاری کنید. توجه داشته باشید، تمرین‌ها به صورت دستی تصحیح خواهد شد.
- هم‌فکری و هم‌کاری در پاسخ به تمرینات اشکالی ندارد؛ ولی پاسخ ارسالی حتماً باید توسط خود شخص انجام شده‌باشد.
- در صورتی که تصحیح‌کنندگان قضاوت کنند که تقلبی انجام شده‌است؛ کل نمره‌ی این تمرین برای هر دو طرفِ تقلب‌گیرنده و تقلب‌دهنده برابر منفی صد درصد قرار داده خواهد شد.
- در طول ترم می‌توانید از سه روز تاخیر بدون کاهش نمره استفاده کنید. مثلا ۵ ساعت در تمرین اول و دو روز و ۱۹ ساعت در تمرین سوم. تاخیر بیش از این سه روز منجر به کاهش نمره خواهد شد. کاهش نمره به ازای هر روز ۱۰ درصد و به طور پلکانی خواهد بود، مگر در ساعت اول تاخیر که نمره به صورت خطی ۱۰ درصد کاهش می‌یابد.
- هرگونه سؤال مربوط به تمرین‌ها را با موضوع مناسب در کوئرای درس مطرح کنید.

سؤال ۱. تابع جمع (۳۰ نمره)

تابع fancySum را به گونه‌ای پیاده‌سازی کنید تا رابطه‌های زیر برقرار باشد:

```
fancySum(1) + fancySum(1) = 2

fancySum(1, 2) + fancySum(1) = 4

fancySum(1, 2, 3)(3) + fancySum(1) = 10

fancySum(1)(2)(3) + fancySum(1)(3)(2) = 12

fancySum(1)(2, 3) + fancySum(1, 2, 3) + fancySum(1)(2)(3) = 18

...
```

توجه کنید که

- تعداد آرگومان‌ها در هر پرانتز بزرگ‌تر مساوی یک و نامشخص است.
- تعداد پرانتزها جلوی هر تابع بزرگ‌تر مساوی یک و یا نامشخص است.
- تعداد جمع ـ تابع fancySum ـ بزرگ‌تر مساوی ۲ و نامشخص است.

سؤال ۲. دفترچه یادداشت (۳۰ نمره)

در این سوال از شما می‌خواهیم که به کمک زبان JavaScript (بدون استفاده از framework) یک اپلیکیشن وب برای نگهداری یادداشت‌ها بسازید. برای ساخت این اپلیکیشن ابتدا باید سروری را که در پایان تمرین آمده است روی سیستم عامل خود اجرا کنید. نحوه‌ی اجرا و کار با این سرور در انتهای مستند تمرین آمده است. در طول این سوال، هرگاه درخواست به سرور مطرح شده، منظور درخواست‌زدن به سروری است که روی سیستم خود اجرا کرده‌اید. تمام URLها در صورت تمرین ذکر شده‌اند. برای این درخواست‌ها باید از تابع fetch موجود در زبان JavaScript استفاده کنید.

کلاینت

در این سوال از شما توقع می‌رود موارد زیر را پیاده‌سازی کنید:

- نشان‌دادن لیست تمام یادداشت‌های ساخته‌شده تاکنون: برای این کار باید به localhost:5000/cards درخواست GET بزنید و اطلاعات تمام یادداشت‌هایی که تا کنون ساخته شده‌اند را از سرور بگیرید. اطلاعات به صورت json دریافت می‌شوند و هر یادداشت دارای یک id یک title و یک color می‌باشد. شما باید هر یادداشت را به صورت یک کارت در اپلیکیشن وب خود نشان دهید. این کارت به رنگی است که از اطلاعات یادداشت دریافت کرده‌اید و متن روی آن کارت هم از title دریافت شده می‌آید.
- نوشتن یک یادداشت جدید: اپلیکیشن شما باید این قابلیت را داشته باشد که بتوانید یادداشتی جدید با متن و رنگ به لیست یادداشت‌ها اضافه کنید. پس اینکه اطلاعات را از کاربر گرفتید باید در سمت کلاینت کارت جدید مربوطه را به صفحه اضافه کنید و همچنین با درخواست POST به localhost:5000/cards این اطلاعات گرفته‌شده را به سرور بفرستید. اطلاعات را باید به صورت json که متن گرفته‌شده در کلید "title" و رنگ گرفته‌شده در "color" می‌آیند و در body درخواست به سرور فرستاده می‌شوند. در صورت موفقیت‌آمیز بودن ،در پاسخ سرور id کارت ساخته‌شده به همراه اطلاعاتی مربوط به موفقیت‌آمیز بودن ساخت کارت می‌آید.

- ویرایش یک یادداشت: کارت‌های نمایش‌داده‌شده باید گزینه‌ای برای تغییر متن و رنگ یادداشت داشته باشند. تغییرات اعمال شده باید به کاربر نشان داده شوند، و توسط درخواست PUT به localhost:5000/cards/:id به سرور فرستاده شوند. body درخواست مانند قسمت قبل است و در URL، به جای id: باید id ای که برای کارت از سرور دریافت کرده‌اید را جاگذاری کنید.
- حذف یک یادداشت: کارت‌های نمایش‌داده‌شده باید گزینه ای برای حذف آن یادداشت داشته باشند و با استفاده از آن، باید کارت از لیست کارت‌ها حذف شود و با درخواست DELETE به localhost:5000/cards/:id به سرور اطلاع داده شود که آن را از دیتابیس حذف کند. باز هم به جای id: باید id ای که برای کارت از سرور دریافت کرده‌اید را جاگذاری کنید.
- پین‌کردن یک یادداشت به بالای لیست یادداشت‌ها: برای این کار باید کارت یادداشت گزینه برای پین‌کردن آن کارت به بالای لیست داشته باشد که با این کار، اگر کارتی قبلا پین شده است با این کارت جابه‌جا شود و اگر کارتی قبل پین نشده است، این کارت در بالای لیست پین شود. هم‌چنین روی هر کارت پین‌شده با علامتی مشخص کنید که آن کارت پین شده است. لازم نیست این کار را به سرور اطلاع دهید؛ اما باید سمت کلاینت این رویداد را ذخیره کنید. برای این کار باید از local storage استفاده کنید. بار بعدی که این اپلیکیشن تحت وب اجرا می‌شود باید کارت پین‌شده به درستی نشان داده شود.

برای الهام‌گرفتن در انجام این تمرین می‌توانید از این نمونه بازدید کنید که همین کار یادداشت‌برداری و نمایش به صورت کارت را به طور پیشرفته‌تر و با جزئیات بیشتر انجام می‌دهد.

راهنمای اجرای سرور

پیش از این که سرور را اجرا کنید لازم است با موارد زیر آشنایی داشته باشید:

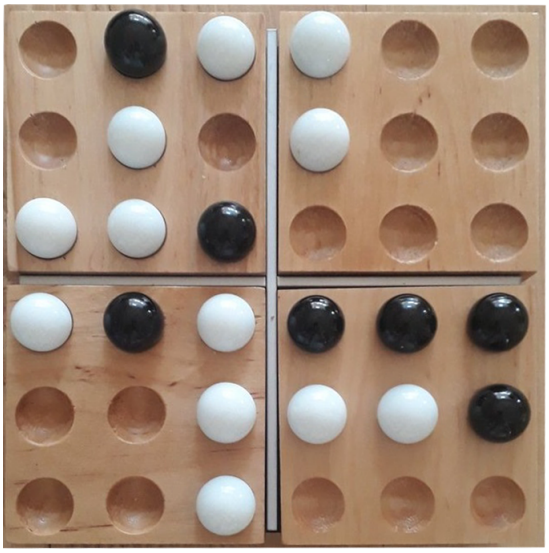
- Node.js: این سرور توسط یک framework جاوااسکریپت به نام Node نوشته شده است که framework ای بسیار قدرتمند برای backend پروژه‌های نرم‌افزاری است. برای اجرای سرور نیاز دارید که Node را روی سیستم عامل خود نصب کنید. برای این کار به سایت رسمی node مراجعه کرده و فایل نصبی متناسب سیستم عامل خود را دانلود کنید.
- NPM: در Node مانند بقیه زبان‌ها از کتابخانه‌هایی استفاده می‌شود که اصطلاحا module نامیده می‌شوند. NPM یک package manager قدرتمند برای این framework است که با استفاده از آن می‌توانید به راحتی از ماژول‌های آماده استفاده کنید و هم‌چنین پروژه‌ی خود را مدیریت و اجرا کنید. لازم به ذکر است که npm همراه با node نصب می‌شود.
- MongoDB: یک دیتابیس NoSQL بسیار مشهور است که سروری که در اختیار شما قرار می‌گیرد از آن برای ذخیره یادداشت‌ها (کارت‌ها) استفاده می‌کند. پس لازم است که این دیتابیس را روی سیستم خود نصب کنید تا سروری که روی کامپیوتر شما اجرا می‌شود بتواند از آن استفاده کند. درایور MongoDB را از سایت رسمی آن برای دستگاه خود دانلود و نصب کنید.

حالا باید سرور را اجرا کنید. برای انجام این کار ابتدا سرور را از این لینک دریافت کنید. به پوشه اصلی پروژه بروید و با دستور npm install ماژول‌های مورد نیاز برنامه را نصب کنید تا آماده‌ی اجرای پروژه شوید. پیش از اجرای سرور باید MongoDB را هم روی سیستم خود اجرا کنید تا سرور بتواند به آن دسترسی داشته باشد. این کار در هر سیستم عامل به یک شکل انجام می‌گیرد. برای انجام این کار می‌توانید از این لینک استفاده کنید. مطمئن شوید که MongoDB روی پورت ۲۷۰۱۷ بر localhost در حال اجرا باشد (حالت پیش‌فرض برای MongoDB همین است).
پس از آن که ماژول‌های لازم را نصب، و هم‌چنین MongoDB را نصب و اجرا کردید نوبت به اجرای سرور می رسد. این کار با npm به راحتی انجام می گیرد. با ترمینال یا کنسول به پوشه‌ی اصلی پروژه بروید و دستور npm run server

را اجرا کنید. با این کار سرور روی localhost:5000 اجرا می‌شود و می‌توانید به آدرس‌هایی که در بخش‌های تمرین به آن‌ها اشاره شد درخواست بزنید. برای بررسی صحت اجراشدن پروژه می‌توانید با استفاده از دستور fetch در JavaScript به سرور درخواست‌های آزمایشی بزنید و جواب بگیرید، و یا از نرم افزار POSTMAN استفاده کنید.

سؤال ۳. Pentago (۴۰ نمره و تا ۱۰ نمره‌ی امتیازی)

بازی Pentago یک بازی دونفره است. بازی روی یک صفحه‌ی ۶ در ۶ انجام می‌شود که از ۴ زیرصفحه به ابعاد ۳ در ۳ تشکیل شده است. در هر نوبت، بازیکن یک مهره از رنگ مربوط به خود (سیاه یا سفید) در یکی از خانه‌های خالی صفحه قرار می‌دهد و سپس یکی از زیرصفحه‌ها را ۹۰ درجه می‌چرخاند که این چرخش می‌تواند به صورت ساعت‌گرد و یا پاد ساعت‌گرد انجام شود. بازیکنی که بتواند ۵ مهره از رنگ خود را در یک ردیف (به صورت عمودی، افقی و یا قطری) قرار دهد برنده‌ی بازی است. اگر همه‌ی خانه‌های صفحه پر شوند و هیچ یک از دو بازیکن برنده نشود بازی مساوی می‌شود. می‌توانید از این‌جا درباره‌ی بازی بیش‌تر بخوانید.



بازی Pentago را با JavaScript همراه با HTML و CSS با رعایت تغییرات و نکات زیر پیاده‌سازی کنید:

- در ابتدا ابعاد هر زیر صفحه و تعداد مهره‌های یک‌رنگ پشت سر هم که برای برنده شدن لازم است با استفاده از prompt() از کاربر گرفته می‌شود.
- هر بازیکن به اندازه‌ی ۱۰ ثانیه فرصت دارد تا حرکت خود را انجام دهد وگرنه مهره‌ی او به صورت تصادفی در یکی از خانه‌های خالی قرار می‌گیرد و یکی از زیرصفحه‌ها به صورت تصادفی می‌چرخد.
- هر بازیکن با کلیک بر روی یکی از خانه‌های خالی صفحه می‌تواند مهره‌ی خود را در آن خانه قرار دهد، سپس باید کاربر بتواند زیرصفحه‌ی دلخواه خود را در جهت دلخواه ۹۰ درجه بچرخاند که نحوه‌ی پیاده‌سازی آن به طراحی دلخواه شما بستگی دارد.
- چرخش زیرصفحه‌ها باید با یک انیمیشن همراه باشد.
- در پایان اگر بازیکنی برنده شد باید مهره‌هایی که باعث برنده شدن او شده‌اند به نحوی از سایر مهره‌ها متمایز گردند که می‌تواند با تغییر رنگ آن‌ها و یا چشمک زدن آن‌ها با یک انیمیشن همراه شود. پیاده‌سازی انیمیشن نمره‌ی امتیازی دارد.
- پس از پایان بازی در صورت برد یک بازیکن، برد او با پیغامی نمایش داده می‌شود و در صورت تساوی نیز تساوی با پیغامی به اطلاع بازیکنان می‌رسد.