TODOS APP

TECHNICAL DOCUMENTATION AND AUDIT

Project description

The Todos App project is created using MVC architecture. MVC (Model-View-Controller) is a pattern that divides application in into three logical components: the model, the view, and the controller:

- *Model* updates the view after manipulation.
- *View* is a visual result of some manipulation, visible for a user.
- *Controller* is used/triggered by a user to manipulate the model.

Every new todo in our app is created and stored in the *model element*. When a new todo item is created, it is also stored with a unique id.
Methods that were used in the *model*:

- update - updates a model by giving it an ID, data to update, and a callback to fire when the update is complete.
- read - finds a model in storage.
- remove - removes a model from storage.
- removeAll - removes all data from the storage.
- getCount - returns numbers of active, completed, and total todos to be used in the view.

Let's have a look at the *view element*.
It has two simple entry points:

- bind(eventName, handler): takes a todo application event and registers the handler.
- render(command, parameterObject): renders the given command with the options.

And, finally, the *controller* element.

It takes *a model and a view* and acts as *the controller* between them. The *controller* loads and initiates *the view* with possible parameters: active or completed.
Also, all the actions on the todos list are designated to be triggered by the user's manipulations. Those are: addItem, editItem, deleteItem, toggleAll.

In this project, we can also find some additional elements that are necessary for the project's successful performance.

Those are:

- App - contains the global app. Within this file, a new Todo object is created. This object receives the todo list's name as its parameter, which defines the name for the local storage. For the properties model, template, and view the corresponding constructor functions are called. These will be explained later. The controller property of the Todo object calls the controller constructor function, which gets the Todo object's model and view passed in.
- Helpers - has a list of wrappers (event listeners and query selectors).
- Store - creates a new client-side Store object and will create an empty collection if no collection already exists. The local storage object is then turned into a string. The passed in callback calls with the argument this the name of the local storage object.
- Template - stores the template for every todo.

Bugs

During the work and code analysis, three bugs were found.

A typo

In the controller.js element (Line 95) 'Controller.prototype.*addd*Item = function (title) {...}. This line contained a typo in the 'adddItem'.

Duplicate id's conflict

In the store.js (Line 77) we see a function that stores a todo with a randomly assigned id. The id consists of 6 digits, but nonetheless it's not excluding a possible id conflict due to id duplication.

Therefore, a simple Boolean was added. By default 'checkIfUnique' is set as false. Then, we check whether a random id is not assigned already.

Incorrect display of todo's check

The check was not displayed after clicking on the area to complete a todo. In the index.css (lines 192 and 196) the HEX colors were changed to RGB. This made the check display correctly.

Tests

To prevent future breakings and bugs in the app, help developers that are going to work on this project in the future, some tests were written in addition to those that already exist.

The added tests are:

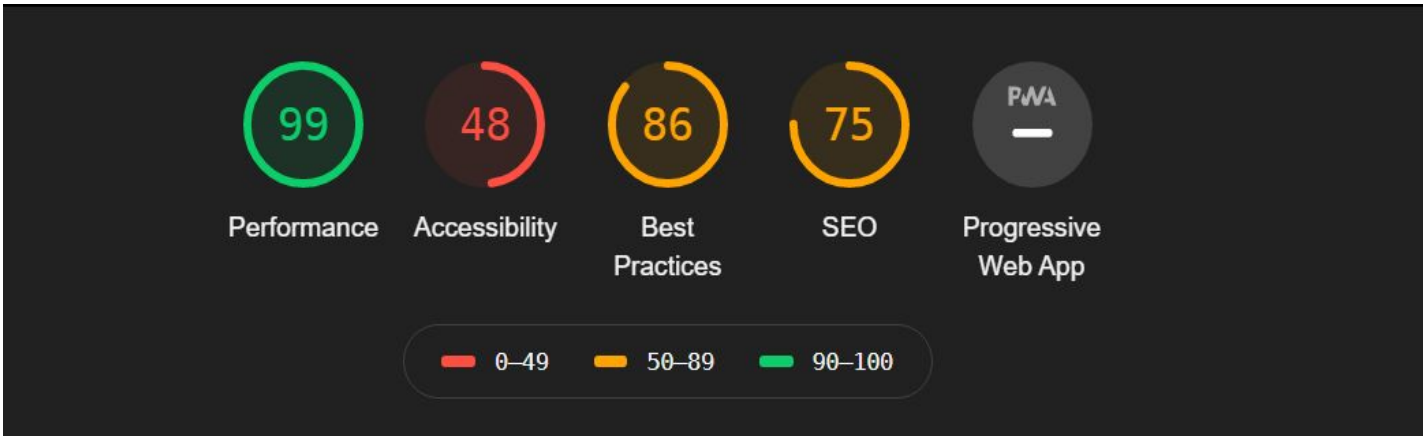
- Show entries on start-up
- Show active entries
- Show completed entries
- Highlight "All" filter by default
- Highlight "Active" filter when switching to the active view
- Toggle all todos to completed
- Update the view
- Add a new todo to the model
- Remove an entry from the model

In total 30 tests brought 0 failures.

Audit

The Todos App is created in an unnecessarily complicated way. To estimate the performance of the app, I took the competitor's website Todolistme.net and compared these two applications. For audit, I used Lighthouse inside of DevTools.

Todos App results



Performance

Accessibility

Contrast — These are opportunities to improve the legibility of your content.

- ▲ Background and foreground colors do not have a sufficient contrast ratio.

Names and labels — These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

- ▲ Form elements do not have associated labels

Best Practices

- ▲ Does not use HTTP/2 for all of its resources — 12 requests not served via HTTP/2
- ▲ Browser errors were logged to the console

SEO

Mobile Friendly — Make sure your pages are mobile friendly so users don't have to pinch or zoom in order to read the content pages. Learn more.

- ▲ Does not have a `<meta name="viewport">` tag with `width` or `initial-scale` No `<meta name="viewport">` tag found

Content Best Practices — Format your HTML in a way that enables crawlers to better understand your app's content.

- ▲ Document does not have a meta description

Progressive Web App

Fast and reliable

- ● Page load is fast enough on mobile networks
- ▲ Current page does not respond with a 200 when offline
- ▲ `start_url` does not respond with a 200 when offline No usable web app manifest found on page.

Installable

- Uses HTTPS
- Does not register a service worker that controls page and `start_url`
- Web app manifest does not meet the installability requirements Failures: No manifest was fetched.

PWA Optimized

- Does not redirect HTTP traffic to HTTPS
- Is not configured for a custom splash screen Failures: No manifest was fetched.
- Does not set a theme color for the address bar.
  Failures: No manifest was fetched, No `<meta name="theme-color">` tag found.
- Content is sized correctly for the viewport
- Does not have a `<meta name="viewport">` tag with `width` or `initial-scale` No `<meta name="viewport">` tag found
- Contains some content when JavaScript is not available
- Does not provide a valid `apple-touch-icon`

Todolistme.net results

Performance	Accessibility	Best Practices	SEO	Progressive Web App
59	38	71	78	PWA —

0–49 50–89 90–100

Performance

First Contentful Paint	1.6 s	First Meaningful Paint	1.6 s
Speed Index	4.1 s	First CPU Idle	9.2 s
Time to Interactive	10.1 s	Max Potential First Input Delay	900 ms

Accessibility

Contrast — These are opportunities to improve the legibility of your content.

- Background and foreground colors do not have a sufficient contrast ratio.

Best practices — These items highlight common accessibility best practices.

- `[id]` attributes on the page are not unique

Best Practices

- Does not use HTTPS — 41 insecure requests found
- Does not use HTTP/2 for all of its resources — 27 requests not served via HTTP/2
- Includes front-end JavaScript libraries with known security vulnerabilities — 2 vulnerabilities detected
- Browser errors were logged to the console

SEO

- Does not have a `<meta name="viewport">` tag with `width` or `initial-scale` No `<meta name="viewport">` tag found

Content Best Practices — Format your HTML in a way that enables crawlers to better understand your app's content.

- Image elements do not have `[alt]` attributes

Progressive Web App

Fast and reliable

Page load is not fast enough on mobile networks

Your page loads too slowly and is not interactive within 10 seconds. Look at the opportunities and diagnostics in the "Performance" section to learn how to improve.

— Interactive at 10.1 s

Current page does not respond with a 200 when offline

`start_url` does not respond with a 200 when offline No usable web app manifest found on page.

Installable

Does not use HTTPS — 41 insecure requests found

Does not register a service worker that controls page and `start_url`

Web app manifest does not meet the installability requirements Failures: No manifest was fetched.

PWA Optimized

Does not redirect HTTP traffic to HTTPS

Is not configured for a custom splash screen Failures: No manifest was fetched.

Does not set a theme color for the address bar.

Failures: No manifest was fetched, No `<meta name="theme-color">` tag found.

Content is sized correctly for the viewport

Does not have a `<meta name="viewport">` tag with `width` or `initial-scale` No `<meta name="viewport">` tag found

Contains some content when JavaScript is not available

Does not provide a valid `apple-touch-icon`

Audit Conclusion

As we see, Todos App is much faster than Todolistme.net. Todos App is more user-friendly, it is light and fast, not cluttered with unnecessary features or images.

Todos App can be significantly improved. First, I could improve code to make the app look better on mobile devices since more and more users switch to mobile phones and tablets.

Second, it is important to make this app more accessible. A good idea would be to change color scheme and add a night mode for this page.

From the structural logic, the app should be refactored and minified if possible. This will improve general performance and help to escape the callback hell.

On the other hand, the competitor's website Todolistme.net is over-engineered from the UI perspective. It is playing the role of a todo list, but also somewhat like a calendar/organizer as well. The color scheme is poorly chosen, and that we can also see as a worse result in the accessibility segment. This webpage is also using advertisement, which makes it look less clean and slows down performance.