1) Установленные пакеты приложений

   "axios": "^0.19.2",

    "classnames": "^2.2.6",

    "eslint-config-airbnb": "^18.2.0",

    "eslint-config-prettier": "^6.11.0",

    "eslint-plugin-flowtype": "^5.1.3",

    "eslint-plugin-import": "^2.21.2",

    "eslint-plugin-jsx-a11y": "^6.3.1",

    "eslint-plugin-react": "^7.20.0",

    "gh-pages": "^3.1.0",

    "husky": "^4.2.5",

    "lint-staged": "^10.2.11",

    "node-sass": "^4.14.1",

    "prettier": "^2.0.5",

    "prop-types": "^15.7.2",

    "react": "^16.13.1",

    "react-dom": "^16.13.1",

    "react-scripts": "3.4.1",

    "react-scroll": "^1.7.16",

    "react-tooltip": "^4.2.7",

    "styled-components": "^5.1.1"

"eslint-plugin-prettier": "^3.1.4"

Axios для запросов на бэкенд

Classnames для записи классов

Eslint и husky для стилизации и нормализации кода

React tooltip – для тултипов

React scroll – для скролла вниз по элементам

Styled components – для стилизации компонентов

2) Приложение написано на Vanilla React.js с использованием классовых компонентов.

3) Приложение разделено соответственно на три главных раздела header, main, footer.

4) Основная логика приложения заключена в main, как классовый компонент, который и собирает, и передает информацию другим компонентам вниз по дереву. Он же и обрабатывает ошибки.

5) Логика работы с бэкэндом заключена в папке service, где и выполняются запросы на аксиос, токен передается в LocalStorage. Есть повторяющийся код, особенно захардкоденные запросы, но их можно в дальнейшем при рефакторинге убрать.

6) Для удобства и пере использования компонентов – была выведена папка shared, которая на данный момент использует модальку, которая срабатывает и на успех и на «фейл».

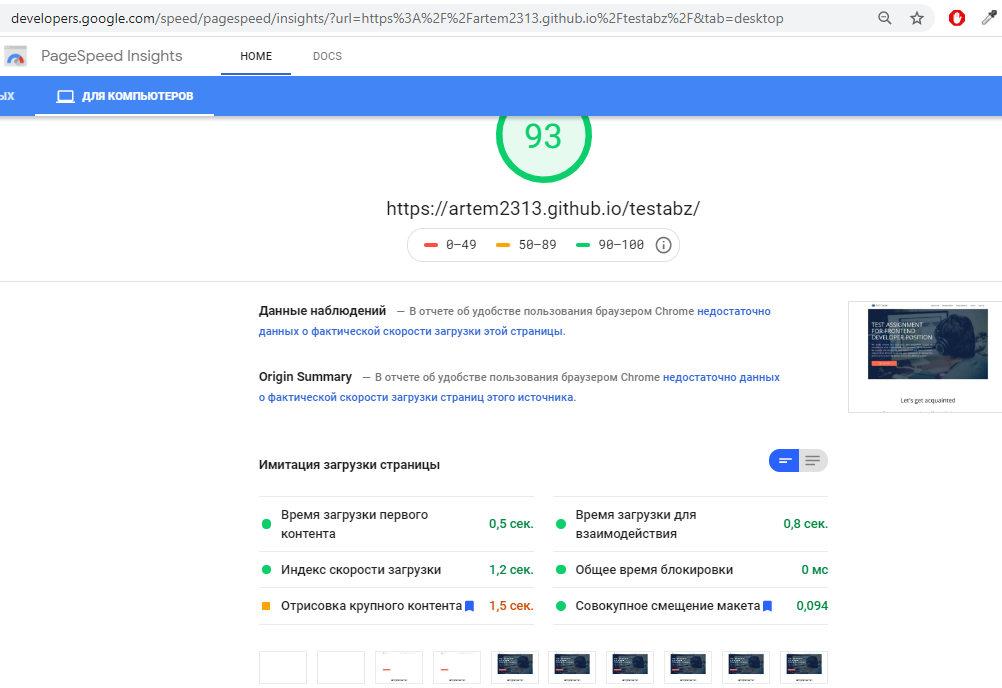
7) Касательно логики верстки – была использована адаптивно-резиновая верстка с использованием в основном SASS и styled-component. Для общих стилизованных компонентов и фонтов была выведена папка style, которая и заключает в себя стили других элементов разбросанных в папках по компонентам.

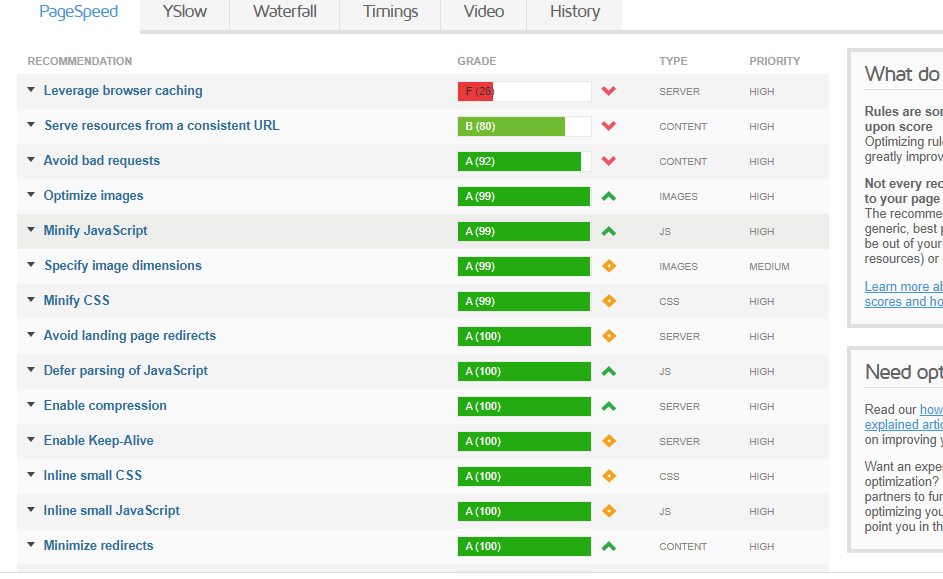
8) Касательно верстки в плане пиксель перфект, в основном придерживался оригинала psd, но, вполне возможно, что финальный результат будет немного отличаться, но, в целом, верстка не плывет на разных экранах и девайсах. Использовались variables и normalize для общей стилизации и изменений.

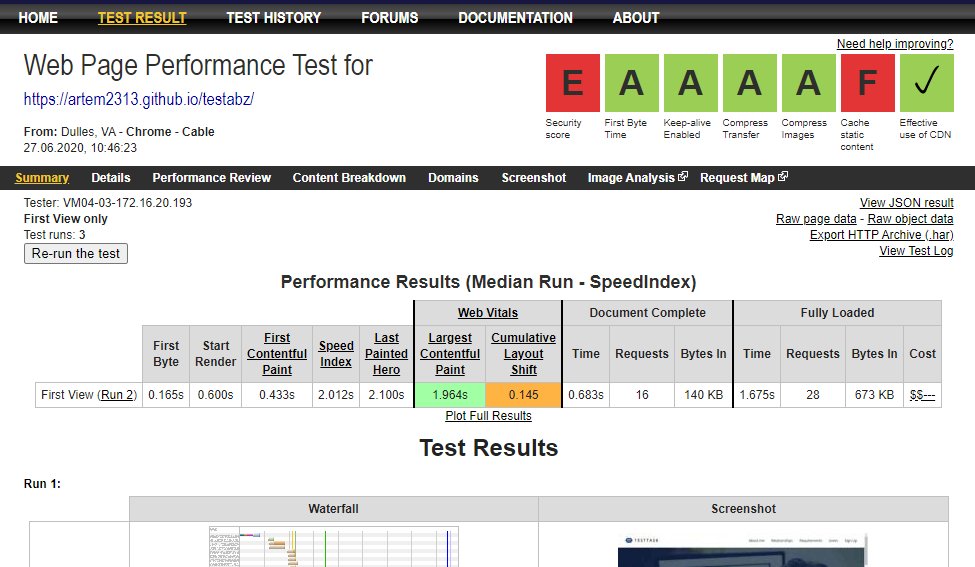
9) Для валидации на стороне пользователя – был создан метод validate, который и не допускает отправки на сервер, если не будут выполнены условия отправки (имя не меньше 6 символов, имейл содержит @, фото только определенного размера и т.д.).

10) Для удобства использования, сайт выложен на гитхаб pages.

11) Speed of the website







Как и гугл метрика, так и gtmetrix показывают хорошие результаты загрузки страницы, в целом, заниженные оценка только одна, и это из-за кеширования, которое я не включал в сайт. Опять же, есть две основные ошибки – это кеширование статичных элементов (не предусматривал и секъюрность кода – из-за гитхаба)

12) Для разбивки сайта на chunks, используется Lazy и Suspense.

13) Для проверки того, что приходит в компоненты – использовался PropTypes.

Итог: Возможны некоторые ошибки и повторяющийся компоненты или код, но при рефакторинге его можно убрать, а повторяющийся компоненты записать в shared (кнопки, формы, контейнеры и т.д. маневра для пространства много) и в дальнейшем переиспользовать.

Приложение также можно усложнить для возможного дальнейшего расширения, к примеру, использовать Redux для менеджмента и Redux Saga для асинхронки.