Описание создания

Как и при создании сайта, первым делом создается входной html файл «index.html». После прописи стандартной структуры html, в теге body создаются 2 input’а, и одна кнопка. Чтобы все эти элементы центрировать, подключается «style.css» и создается, оборачивающий все созданные до этого элементы, div (рис. ?.1).

Рис. ?.1 Итоговая разметка станицы входа

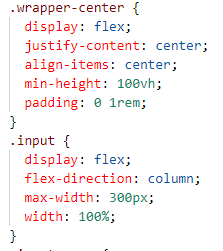
Для выравнивания элементов используется flex box (рис. ?.2).

Рис. ?.2 Центрирование элементов

Для использования упрощенного синтаксиса javaScript'а, используется популярная библиотека JQuery. Подключается она перед самим скриптом.

Скрипт для записи имени в локальное хранилище и перекидывания пользователя на следующую страницу, пишется в отдельном файле «input.js» и подключается перед закрывающим тегом body в html файле. (рис. ?.3).

Рис. ?.3 Скрипт для входа в программу

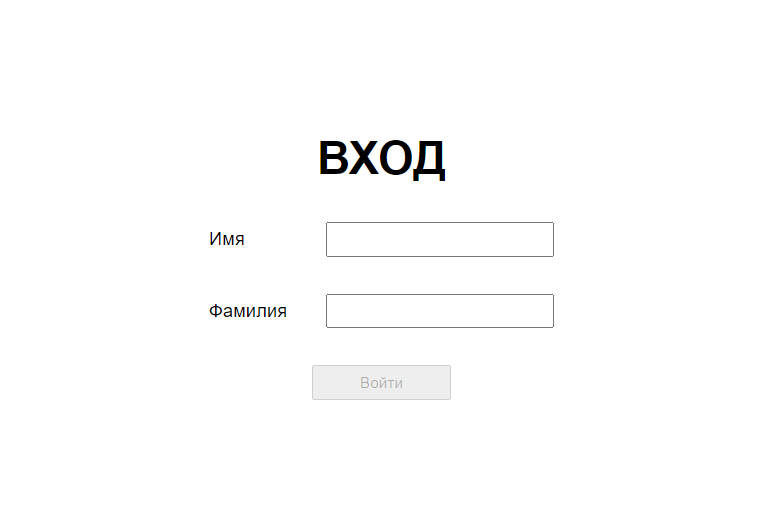
В результате получаем первую страницу (рис. ?.4).

Рис. ?.4 Страница входа

Следующая страница «pickTest.html», содержит в себе 4 кнопки, которые центрируются с помощью ранее описанных стилей, благодаря оборачиванию элементов тем же самым div’ом с классом «wrapper-center» и подключением «style.css» (рис. ?.5).

Рис. ?.5 Html разметка страницы с выбором теста

К каждому элементу кнопки привязано событие «onclick», которое позволяет по нажатию на кнопку передать нужные параметры с названием теста в функцию «setNameTest». Данная функция записывает название выбранного теста в локальное хранилище и переносит пользователя на страницу «test.html». Так как сама функция небольшая, то ее можно поместить прямо в тег script, не создавая при этом отдельный js файл.

В результате страница с выбором теста выглядит так (рис. ?.6).



Рис. ?.6 Страница с выбором теста

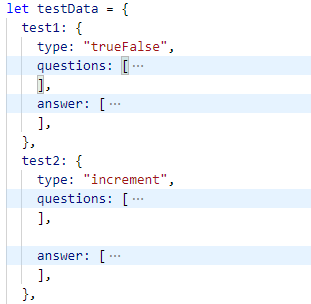
Сами же тесты содержатся в файле «testData.js» в папке module в виде объекта. Этот большой объект имеет 4 свойства и каждое из этих свойств является отдельным тестом (рис. ?.7).

Рис. ?.7 Страница с выбором теста

Каждый тест - это тоже объект, который содержит в себе свойства для явного указания типа теста, массив с вопросами и массив с ответами.

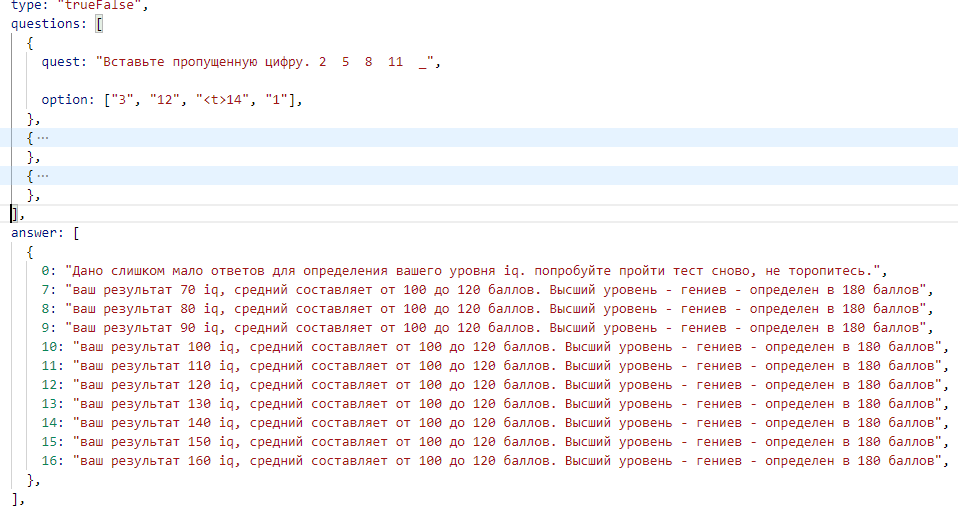
В программе предусмотрено 2 вида теста. Первый это стандартный с одним правильным ответом (рис. ?.8).

Рис. ?.8 Заполнение теста с типом «trueFalse»

Второй же накопительный, то есть если необходимо, допустим, определить архетип человека, очевидно нам не подойдет система с одним единственным правильным ответом. В этом случае используются ключи. При выборе варианта ответа будет увеличиваться значение определенного ключа. В конце из всех ключей будет выбран тот у которого значение больше всего (рис. ?.9).

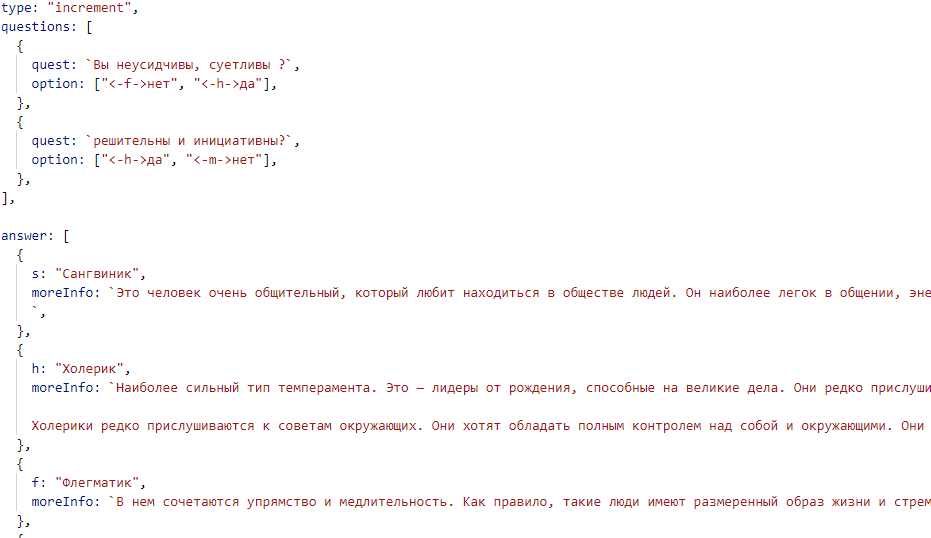


Рис. ?.9 Заполнение теста с типом «increment»

На основе данных из «testData.js» и записанным ранее данным в локальное хранилище, строится страница «test.html». Файл «test.html» имеет такую разметку (рис. ?.10).

Рис. ?.10 html разметка страницы с тестами

За отрисовку элементов теста отвечает следующая функция в файле «test.js» (рис. ?.11).

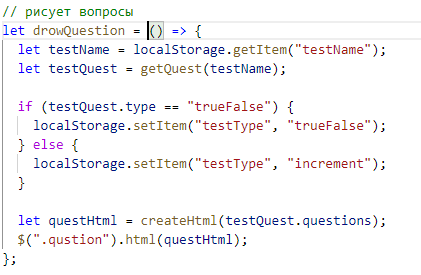


Рис. ?.11 Функция для отрисовки вопросов

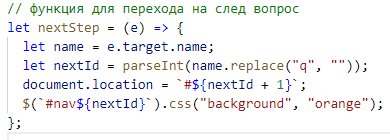
При выборе варианта ответа, программа автоматически скролит вниз до следующего вопроса. За это отвечает следующая функция (рис. ?.12)

Рис. ?.12 Функция автоматического скролла

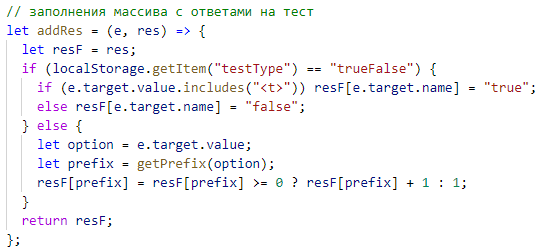
По нажатию на input записывается выбранный ответ в объект «res» c помощью следующей функции (рис. ?.13).

Рис. ?.13 Функция записи результата

Чтобы завершить тест, необходимо нажать на кнопку «завершить тест» выполняется следующая функция (рис. ?.14).

Рис. ?.14 Функция проверки результата

Сама страница выглядит так (рис. ?.15).

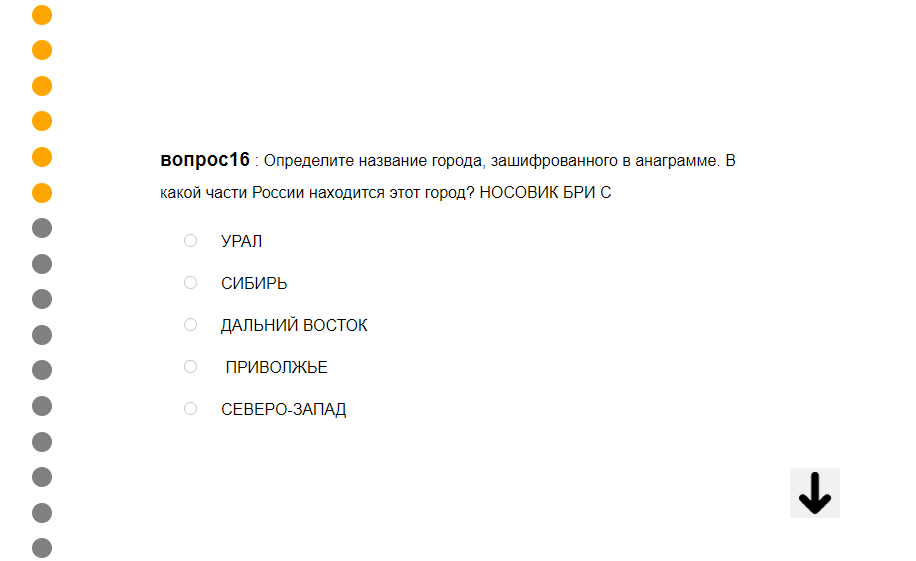


Рис. ?.15 Страница с тестами

Содержимое страницы с результатом рендериться с помощью «result.js».

Рис. ?.15 содержимое «result.js»

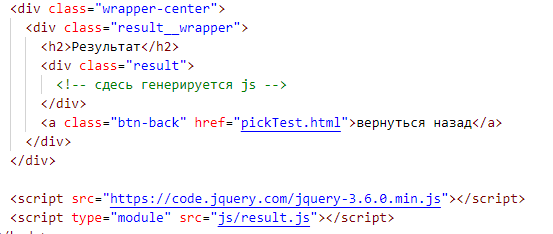
Разметка страницы «result.html» выглядит так (рис. ?.16).

Рис. ?.16 Разметка страницы «result.html»

На данном этапе проект уже можно заливать на хостинг, и он будет работать как сайт. Но чтобы проект открывался с exe файла как полноценная настольное приложение, нужна библиотека Electron.js.

Существует уже готовая среда electron-forge, где уже прописаны основные настройки этой библиотеки.

Чтобы воспользоваться этой библиотекой, необходимо установить node.js, git и сам electron-forge глобально.

Можно инициализировать проект electron-forge как это показано на официальном сайте библиотеки. Необходимо перенести ранее созданную верстку в папку «src». Если протестировать работу программы сейчас, то можно заметить, что тесты не будут выводиться там, где должны. Все это из-за того, что electron плохо дружит с модульностью js файлов. Поэтому необходимо подключить «testData.js» напрямую к «test.html». А еще из-за того что верстка открывается не в браузере, необходимо прописать «<script>if (typeof module === "object") {window.module = module; module = undefined;}</script>» во всех файлах html перед подключением скриптов, чтобы библиотека JQuery работала (рис. ?.17).



Рис. ?.17 Подключение скриптов для Electron

В конечном итоге остается лишь прописать «npm run package», для финальной сборки проекта.