

Создание сайта-визитки на языке JavaScript

Халиманенков Андрей Сергеевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассматриваются разработка сайта визитки для веб-разработчика на языке JavaScript, в котором используется работа с RESTAPI при помощи запросов к локальной базе данных для получения портфолио и дальнейшей генерации страницы выполненных работ.

Ключевые слова: визитная карточка, портфолио, JavaScript, HTML, CSS, RESTAPI, веб-приложение.

Creating a visit card site in JavaScript

Khalimanenkov Andrey Sergeevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses the development of a business card website for a web developer in JavaScript, which uses RESTAPI using queries to a local database to obtain a portfolio and further generate a page of completed site projects.

Keywords: visit card, food delivery, portfolio, HTML, CSS, RESTAPI, web application.

Создание личного сайта визитки — это очень важная часть резюме для каждого веб-разработчика, т.к. это напрямую демонстрирует навыки специалиста. Другими словами, сайт-визитка является «лицом» разработчика и очень важно, чтобы этот сайт выглядел привлекательно и работал без сбоев. В этой статье будет рассмотрено создание такого сайта для начинающего разработчика на чистом JavaScript [5], HTML [6] и CSS [7] без использования серьёзных библиотек для frontend разработки, таких как React, Vue или Angular.

Э. Сафаров и В.С. Исмагилова в своей статье рассмотрели сайт-визитку компании как один из инструментов привлечения покупателей. Авторы статьи дают определение данного понятия, анализируют структуру сайтов-визиток. Также в статье уделяется внимание оптимальной структуре сайта, даются рекомендации по его общему ведению [1]. Н. А. Борсук, Т. Д. Карпова и А. В. Егорова рассмотрели актуальность разработки интернет приложений для различных областей деятельности человека, приведена классификация сайтов и произведен сравнительный анализ средств

разработки сайтов-визиток средствами конструктора Wix и средствами языка гипертекстовой разметки HTML [2]. И. И. Каширская рассмотрела создание сайта-визитки учителя [3]. А. А. Бондарев и др. выполнили комплексный анализ проблематики, связанной с созданием, внедрением, поддержкой эксплуатации, обеспечением эффективности использования и информационной безопасности сайтов - визиток. Типичные цели создания и использования сайтов - визиток проанализированы с позиций их владельцев: отдельных физических лиц, их групп, небольших организаций [4].

На сайте находятся данные о разработчике, а именно:

- имя;
- фото;
- кнопка «заказать сайт»;
- список достоинств;
- галерея работ.

Слева находится неподвижный блок, в котором находятся:

- телефон;
- электронная почта;
- ссылки на социальные сети.

Это продемонстрировано на рисунках 1 и 2.

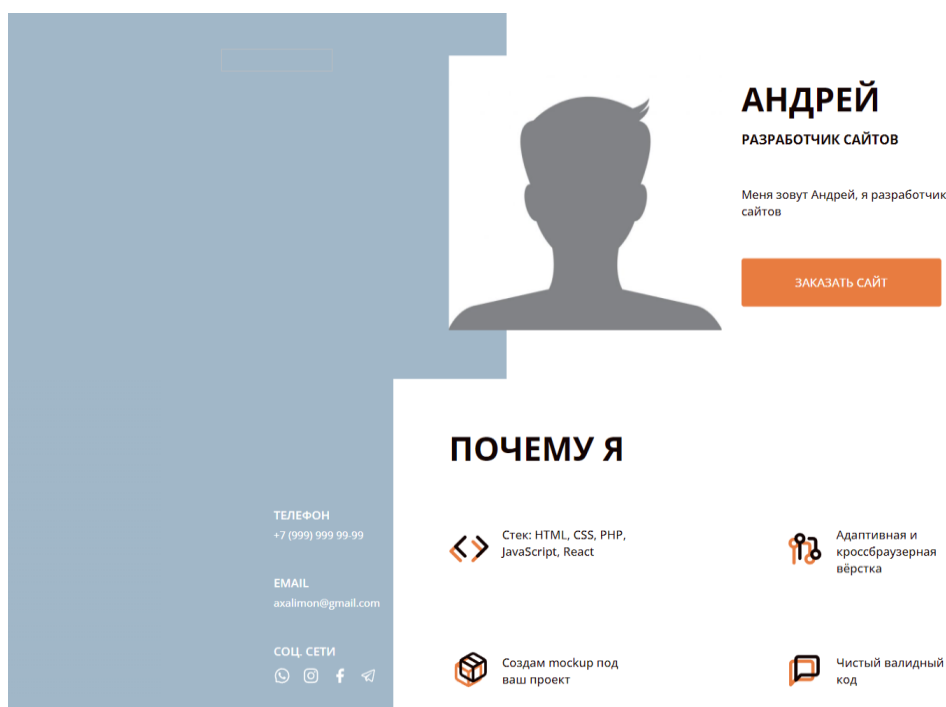


Рисунок 1 – Сайт-визитка

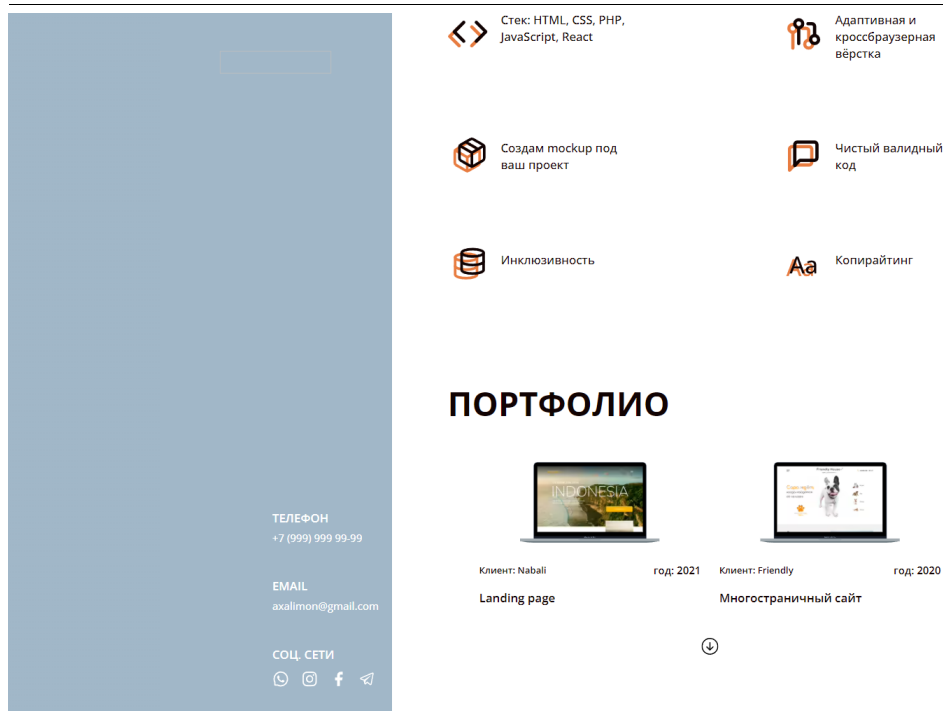


Рисунок 2 – Сайт-визитка

Область с портфолио генерируется из базы данных. В ней хранится JSON файл со всеми сайтами разработчика. Это значит, что для добавления работ не нужно каждый раз изменять исходный код сайта, а требуется лишь добавить краткую информацию в базу данных, которая выполнена в виде локального JSON файла. После чего новые сайты из портфолио будут загружаться по требованию пользователя.

Каждый сайт имеет свой объект и следующие его ключи:

- изображение предпросмотра;
- полное изображение всего сайта;
- клиент, заказывавший услугу;
- год создания;
- тип сайта.

JSON файл выглядит следующим образом:

```
[{
  "preview": "img/db/indotravel-min",
  "image": "img/db/indotravel",
  "client": "Nabali",
  "year": 2021,
  "type": "Landing page"
},
{
  "preview": "img/db/friendly_house-min",
  "image": "img/db/friendly_house",
  "client": "Friendly",
  "year": 2020,
  "type": "Многостраничный сайт"
},
{
```

```
    "preview": "img/db/dom-min",
    "image": "img/db/dom",
    "client": "Дом",
    "year": 2020,
    "type": "Корпоративный сайт"
  },
  {
    "preview": "img/db/iphone-min",
    "image": "img/db/iphone",
    "client": "Full Apple",
    "year": 2021,
    "type": "Landing-page"
  },
  {
    "preview": "img/db/go-min",
    "image": "img/db/go",
    "client": "Game Over",
    "year": 2019,
    "type": "Landing-page"
  },
  {
    "preview": "img/db/cinema-min",
    "image": "img/db/cinema",
    "client": "Cinema",
    "year": 2020,
    "type": "Многостраничный сайт"
  },
  {
    "preview": "img/db/mth-min",
    "image": "img/db/mth",
    "client": "МТН",
    "year": 2019,
    "type": "Корпоративный сайт"
  },
  {
    "preview": "img/db/artdeco-min",
    "image": "img/db/artdeco",
    "client": "ArtDeco",
    "year": 2019,
    "type": "Корпоративный сайт"
  }
]
```

Область с портфолио имеет кнопку со стрелкой вниз, при клике на которую вызывается скрипт, добавляющий ещё два сайта из портфолио на главную страницу визитки. И так, пока сайты в портфолио не закончатся, после чего стрелка исчезает (рис. 3 и 4). Сайты добавляются по 2 штуки. Но это можно настроить в исходном JS коде, изменив переменную COUNT_CARD. При этом для добавления из массива всех работ используется функция slice, которая гарантирует, что если в портфолио

останется меньше работ, чем указано в COUNT_CARD, то скрипт не выдаст ошибку, а корректно загрузит остатки.

ПОРТФОЛИО



Клиент: Nabali

год: 2021

Landing page



Клиент: Friendly

год: 2020

Многостраничный сайт



Клиент: Дом

год: 2020

Корпоративный сайт



Клиент: Full Apple

год: 2021

Landing-page



Рисунок 3 – Загрузка двух работ из портфолио

ПОРТФОЛИО



Клиент: Nabali

год: 2021

Landing page



Клиент: Friendly

год: 2020

Многостраничный сайт



Клиент: Дом

год: 2020

Корпоративный сайт



Клиент: Full Apple

год: 2021

Landing-page



Клиент: Game Over

год: 2019

Landing-page



Клиент: Cinema

год: 2020

Многостраничный сайт



Клиент: MTH

год: 2019

Корпоративный сайт



Клиент: ArtDeco

год: 2019

Корпоративный сайт

Рисунок 4 – Все загруженные работы из портфолио

При нажатии на любую из карточек в портфолио – откроется модальное окно с полным изображением сайта, которую можно промотать и

просмотреть (рис. 5). Открытие происходит плавно из-за медленного прибавления непрозрачности для картинке. Также сохраняется место горизонтальной промотки сайта. Это значит, что после закрытия полного изображения пользователя не вернёт в самое начало сайта, а он останется на том месте, где был до открытия картинки.

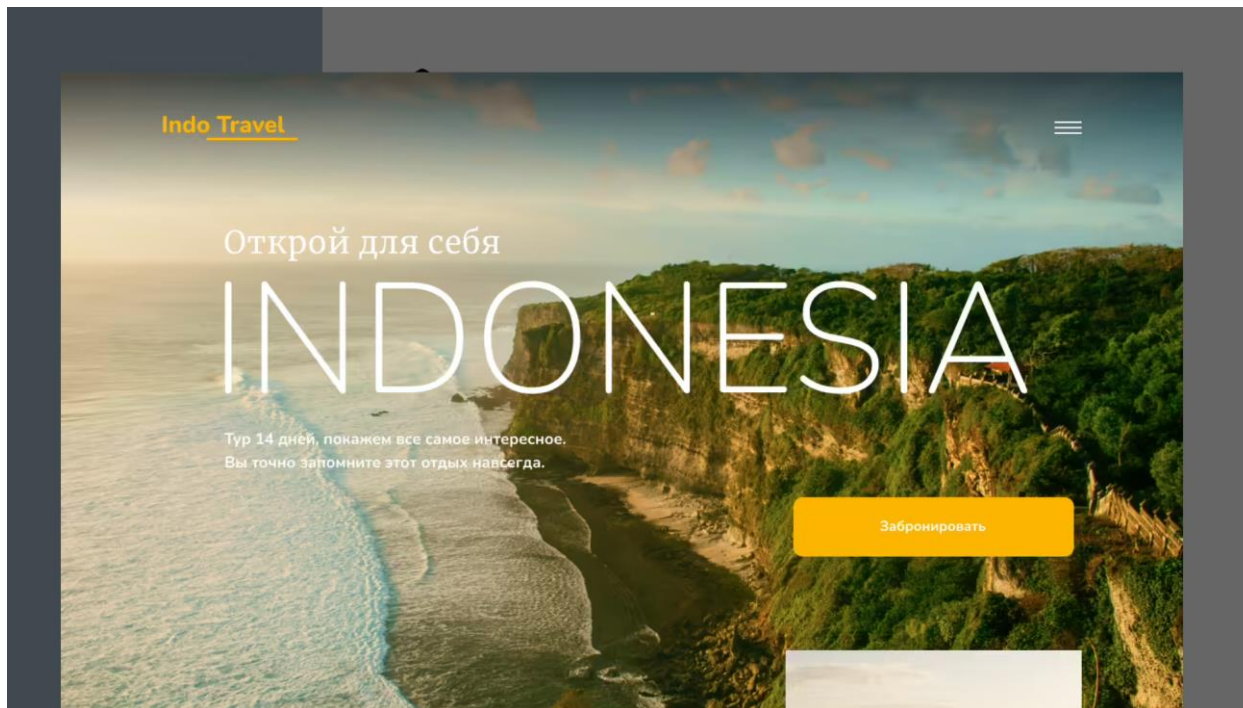


Рисунок 5 – Открытое модальное окно

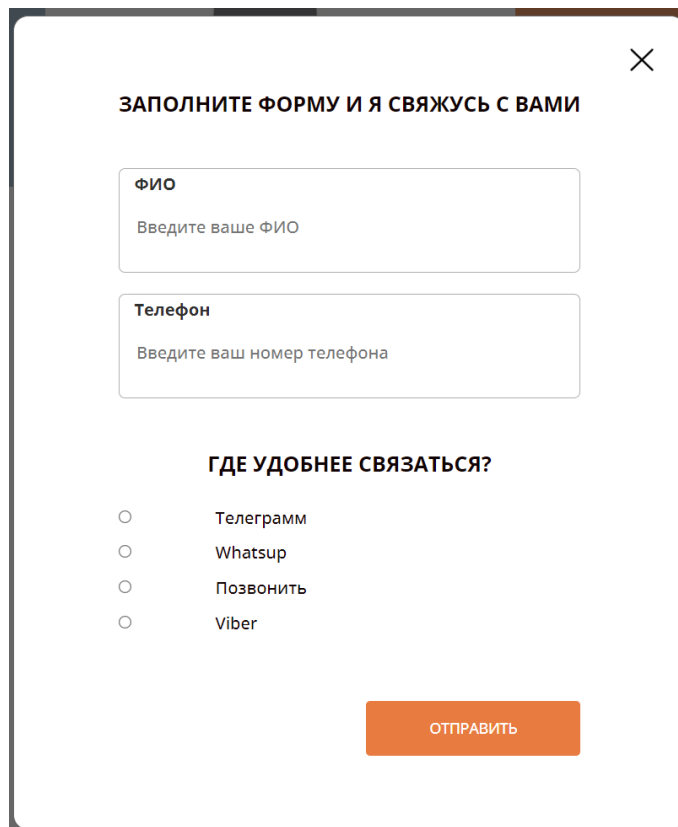
При нажатии на кнопку «Заказать сайт» (рис. 6) открывается модальное окно с формой, в которой нужно заполнить:

- ФИО;
- Телефон;
- Предпочтительный способ связи.

Это окно закрывается по клику «мимо» формы. То есть по полупрозрачной области вокруг формы. Проверяется условием:

```
modal.addEventListener('click', (e) => {  
  if (e.target === modal) {  
    closeModal();  
  }  
});
```

Такой же принцип закрытия работает и у полных изображений из портфолио.



ЗАПОЛНИТЕ ФОРМУ И Я СВЯЖУСЬ С ВАМИ

ФИО
Введите ваше ФИО

Телефон
Введите ваш номер телефона

ГДЕ УДОБНЕЕ СВЯЗАТЬСЯ?

Телеграмм

Whatsup

Позвонить

Viber

ОТПРАВИТЬ

Рисунок 6 – форма для заказа сайта

Полный исходный JS код сайта:

```
const disableScroll = () => {
  document.body.scrollTop = window.scrollY;
  const widthSCroll = window.innerWidth - document.body.offsetWidth;
  if (window.innerWidth >= 992) {
    document.querySelector('.page__header').style.left = `calc(50% - 50vw - ${Math.floor(widthSCroll / 2)}px)`;
  }
  if (window.innerWidth >= 1440) {
    document.querySelector('.page__header').style.left = `calc(50% - ${720 + Math.floor(widthSCroll / 2)}px)`;
  }

  document.body.style.cssText = `
    overflow: hidden;
    position: fixed;
    top: -${document.body.scrollTop}px;
    left: 0;
    height: 100vh;
    width: 100vw;
    padding-right: ${widthSCroll}px;
  `;
}

const enableScroll = () => {
  document.body.style.cssText = 'position: relative';
  window.scroll({top: document.body.scrollTop});
  document.querySelector('.page__header').style.left = ``;
}
```



```
{
  const $presentOrderBtn = document.querySelector('.present__order-btn');
  const $modalClose = document.querySelector('.modal__close');
  const $pageOverlayModal = document.querySelector('.page__overlay_modal');

  const handlerModal = (openTrigger, closeTrigger, modal, modalClass,
animSpeed) => {
    let opacity = 0;
    speed = {
      fast: 0.05,
      medium: 0.025,
      slow: 0.02,
    }

    const openModal = () => {
      disableScroll();
      modal.style.opacity = opacity;
      modal.classList.add(modalClass);

      const anim = () => {
        opacity += speed[animSpeed];
        modal.style.opacity = opacity;
        if (opacity < 1) requestAnimationFrame(anim);
        else {
          opacity = 1;
          modal.style.opacity = opacity;
        }
      };
      requestAnimationFrame(anim);
    };

    const closeModal = () => {
      const anim = () => {
        opacity -= speed[animSpeed];
        modal.style.opacity = opacity;
        if (opacity > 0) {
          requestAnimationFrame(anim);
        } else {
          modal.classList.remove(modalClass);
          enableScroll();
          opacity = 0;
          modal.style.opacity = opacity;
        }
      };
      requestAnimationFrame(anim);
    };

    openTrigger.addEventListener('click', openModal);
    closeTrigger.addEventListener('click', closeModal);
    modal.addEventListener('click', (e) => {
      if (e.target === modal) {
        closeModal();
      }
    });
  };

  handlerModal($presentOrderBtn, $modalClose, $pageOverlayModal,
'page__overlay_modal_open', 'fast');
}
```

```
{
  const $headerContacts = document.querySelector('.header__contacts');
  const $burgerButton = document.querySelector('.header__contacts-burger');

  const handlerBurger = (openButton, menu, openSelector) => {
    openButton.addEventListener('click', () => {
      if (menu.classList.contains(openSelector)) {
        menu.style.height = '';
        menu.classList.remove(openSelector);
      } else {
        menu.style.height = menu.scrollHeight + 'px';
        menu.classList.add(openSelector);
      }
    })
  }

  handlerBurger($burgerButton, $headerContacts, "header__contacts_open");
}

{
  const $portfoliolist = document.querySelector('.portfolio__list');
  const $pageOverlay = document.createElement('div');
  $pageOverlay.classList.add('page__overlay');

  let opacity = 0;
  speed = {
    fast: 0.05,
    medium: 0.025,
    slow: 0.02,
  }

  const openModal = (animSpeed) => {
    disableScroll();
    $pageOverlay.style.opacity = opacity;

    const anim = () => {
      opacity += speed[animSpeed];
      $pageOverlay.style.opacity = opacity;
      if (opacity < 1) requestAnimationFrame(anim);
    };
    requestAnimationFrame(anim);
  };

  const closeModal = (animSpeed) => {
    const anim = () => {
      opacity -= speed[animSpeed];
      $pageOverlay.style.opacity = opacity;
      if (opacity > 0) {
        requestAnimationFrame(anim);
      } else {
        $pageOverlay.textContent = '';
        $pageOverlay.remove();
        enableScroll();
      }
    };
    requestAnimationFrame(anim);
  };
};
```

```
$portfolioList.addEventListener('click', (e) => {
  const $card = e.target.closest('.card')

  if ($card) {
    document.body.append($pageOverlay);
    const title = $card.querySelector('.card__client');

    const $picture = document.createElement('picture');
    $picture.style.cssText = `
      position: absolute;
      top: 5%;
      left: 50%;
      transform: translateX(-50%);
      width: 90%;
      max-width: 1440px;
    `;

    $picture.innerHTML = `
      <source srcset="${$card.dataset.fullImage}.avif"
type="image/avif">
      <source srcset="${$card.dataset.fullImage}.webp"
type="image/webp">
      
    `;
    $pageOverlay.append($picture);
    openModal('fast');
  }
});

$pageOverlay.addEventListener('click', (e) => {
  closeModal('fast');
});

{
  const COUNT_CARD = 2;
  const $portfolioList = document.querySelector('.portfolio__list');
  const $portfolioAdd = document.querySelector('.portfolio__add');

  const getData = () => {
    return fetch('db.json')
      .then((response) => {
        if (response.ok) {
          return response.json();
        } else {
          throw `Ошибка: ${response.status}`;
        }
      })
  };

  const createStore = async () => {
    const data = await getData();
    return {
      data,
      counter: 0,
      count: COUNT_CARD,
      get length() {
        return this.data.length;
      },
    },
  },
}
```

```

        get cardData() {
            const renderData = this.data.slice(this.counter,
this.counter + this.count);
            this.counter += renderData.length;
            return renderData;
        }
    };

    const renderCard = data => {
        const cards = data.map((item) => {

            const {image, preview, client, year, type} = item;

            const $li = document.createElement('li');
            $li.classList.add('portfolio__item');
            $li.innerHTML = `
                <article class="card" tabindex="0" role="button" aria-
label="открыть макет" data-full-image="${image}">
                    <picture class="card__picture">
                        <source srcset="${preview}.avif"
type="image/avif">
                        <source srcset="${preview}.webp"
type="image/webp">
                        
                    </picture>
                    <p class="card__data">
                        <span class="card__client">Клиент:
${client}</span>
                        <time class="card__date" datetime="${year}">год:
${year}</time>
                    </p>
                    <h3 class="card__title">${type}</h3>
                </article>
            `;
            return $li;
        })
        $portfolioList.append(...cards);
    }

    const initPortfolio = async () => {
        const store = await createStore();
        renderCard(store.cardData);
        $portfolioAdd.addEventListener('click', () => {
            renderCard(store.cardData);
            if (store.length === store.counter) {
                $portfolioAdd.remove();
            }
        })
    }

    initPortfolio();
}

```

Таким образом был разработан сайт-визитка, который помогает при устройстве на работу каждому веб-разработчику. Использовались AJAX запросы к локальному JSON файлу, а также обработчики событий `addEventListener`, изменение непрозрачности для модальных окон с полными изображениями работ из портфолио и сохранение вертикальной прокрутки пользователя, чтобы он не отправлялся в начало сайта и не отматывал каждый раз до портфолио снова. Также реализована форма по заказу сайта.

Библиографический список

1. Сафарова Э., Исмагилова В.С. Сайт-визитка компании как необходимый инструмент привлечения покупателей // Символ науки. 2016. №8-1.
2. Борсук Н. А., Карпова Т. Д., Егорова А. В. Сравнительный анализ средств разработки сайтов-визиток // Символ науки. 2017. №3.
3. Каширская И. И. Создание сайта-визитки учителя // Информатика: проблемы, методология, технологии. 2015. С. 88-92.
4. Бондарев А. А. и др. Разработка, введение в эксплуатацию и использование "сайтов-визиток": анализ целей, возможных подходов, трудоемкости, структуры рисков // Прикаспийский журнал: управление и высокие технологии. 2020. №. 1 (49). С. 61-76.
5. JavaScript URL: <https://ru.wikipedia.org/wiki/JavaScript> (дата обращения: 03.01.2022).
6. HTML URL: <https://ru.wikipedia.org/wiki/HTML> (дата обращения: 03.01.2022).
7. CSS URL: <https://ru.wikipedia.org/wiki/CSS> (дата обращения: 03.01.2022).