**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ ЦЕНТР ПРОФЕССИОНАЛЬНЫХ КВАЛИФИКАЦИЙ И СОДЕЙСТВИЯ ТРУДОУСТРОЙСТВУ «ПРОФЕССИОНАЛ»**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к итоговой аттестационной работе на тему

**«Разработка web-ресурса с использованием технологий**

**HTML, CSS, JavaScript, jQuery» (на примере web-ресурса:** **интернет-магазин)**

Слушателя Привезенцева Д. В. группы №: 361

программы профессиональной переподготовки

«Frontend разработка»

Москва, 2021

**План работы.**

В качестве итоговой аттестационной работы стояла задача сверстать полноценный веб-сайт интернет-магазина. Этапы выполнения:

* Разработка дизайна страницы
* Разработка функционала страницы
* HTML разметка страницы
* Применение CSS стилей
* Применение JavaScript к разным блокам и элементам страницы

**Основная часть.**

**Назначение веб-ресурса.**

Для разработки я выбрал интернет-магазин продуктов готового питания.

Должно быть реализовано модальное окно с формой обратной связи. «Табы» с примерами меню. Слайдер. Калькулятор потребности калорий. Карточки меню с ценой и описанием. Счетчик до конца акции.

**Этапы разработки.**

На первом этапе была сделана HTML разметка страницы. Далее стилизованны с помощью CSS все блоки и элементы на странице. Основной задачей была реализация функциональной части сайта, а Именно «Табы», модальное окно с формой обратной связи, рендер карточек меню, калькулятор потребности калорий и счетчик до конца акции.

**Пример кода (JS):**

window.addEventListener('DOMContentLoaded', () => {

*// TABS*

  const tabs = document.querySelectorAll('.tabheader\_\_item');

  const tabsContent = document.querySelectorAll('.tabcontent');

  const tabsParent = document.querySelector('.tabheader\_\_items');

  function hideTabContent() {

    tabsContent.forEach((*item*) => {

*item*.classList.add('hide');

*item*.classList.remove('show', 'fade');

    });

    tabs.forEach((*item*) => {

*item*.classList.remove('tabheader\_\_item\_active');

    });

  }

  function showTabContent(*i* = 0) {

    tabsContent[*i*].classList.add('show', 'fade');

    tabsContent[*i*].classList.remove('hide');

    tabs[*i*].classList.add('tabheader\_\_item\_active');

  }

  hideTabContent();

  showTabContent();

  tabsParent.addEventListener('click', (*event*) => {

    const target = *event*.target;

    if (target && target.classList.contains('tabheader\_\_item')) {

      tabs.forEach((*item*, *i*) => {

        if (target == *item*) {

          hideTabContent();

          showTabContent(*i*);

        }

      });

    }

  });

*// MODAL*

  const modalTrigger = document.querySelectorAll('[data-modal]');

  const modal = document.querySelector('.modal');

  const modalCloseBtn = document.querySelector('[data-close]');

  function openModal() {

    modal.classList.add('show');

    modal.classList.remove('hide');

    document.body.style.overflow = 'hidden';

  }

  function closeModal() {

    modal.classList.add('hide');

    modal.classList.remove('show');

    document.body.style.overflow = '';

  }

  modalTrigger.forEach((*btn*) => {

*btn*.addEventListener('click', openModal);

  });

  modalCloseBtn.addEventListener('click', closeModal);

  modal.addEventListener('click', (*e*) => {

    if (*e*.target === modal) {

      closeModal();

    }

  });

  document.addEventListener('keydown', (*e*) => {

    if (*e*.code === 'Escape' && modal.classList.contains('show')) {

      closeModal();

    }

  });

  function showModalByScroll() {

    if (

      window.pageYOffset + document.documentElement.clientHeight >=

      document.documentElement.scrollHeight - 1

    ) {

      openModal();

      window.removeEventListener('scroll', showModalByScroll);

    }

  }

  window.addEventListener('scroll', showModalByScroll);

*// TIMER*

  const deadline = '2022-06-30';

  function getTImeRemaining(*endtime*) {

    let days, hours, minutes, seconds;

    const t = Date.parse(*endtime*) - Date.parse(new Date());

    if (t <= 0) {

      days = 0;

      hours = 0;

      minutes = 0;

      seconds = 0;

    } else {

      days = Math.floor(t / (1000 \* 60 \* 60 \* 24));

      hours = Math.floor((t / (1000 \* 60 \* 60)) % 24);

      minutes = Math.floor((t / 1000 / 60) % 60);

      seconds = Math.floor((t / 1000) % 60);

    }

    return {

      total: t,

      days: days,

      hours: hours,

      minutes: minutes,

      seconds: seconds,

    };

  }

  function getZero(*num*) {

    if (*num* >= 0 && *num* < 10) {

      return `0${*num*}`;

    } else {

      return *num*;

    }

  }

  function setClock(*selector*, *endtime*) {

    const timer = document.querySelector(*selector*);

    const days = timer.querySelector('#days');

    const hours = timer.querySelector('#hours');

    const minutes = timer.querySelector('#minutes');

    const seconds = timer.querySelector('#seconds');

    const timeInterval = setInterval(updateClock, 1000);

    updateClock();

    function updateClock() {

      const t = getTImeRemaining(*endtime*);

      days.innerHTML = getZero(t.days);

      hours.innerHTML = getZero(t.hours);

      minutes.innerHTML = getZero(t.minutes);

      seconds.innerHTML = getZero(t.seconds);

      if (t.total <= 0) {

        clearInterval(timeInterval);

      }

    }

  }

  setClock('.timer', deadline);

*// CARD MENU*

  class MenuCard {

    constructor(*src*, *alt*, *title*, *descr*, *price*, *parentSelector*, ...*classes*) {

      this.src = *src*;

      this.alt = *alt*;

      this.title = *title*;

      this.descr = *descr*;

      this.price = *price*;

      this.classes = *classes*;

      this.parent = document.querySelector(*parentSelector*);

    }

    render() {

      const element = document.createElement('div');

      if (this.classes.length === 0) {

        this.element = 'menu\_\_item';

        element.classList.add(this.element);

      } else {

        this.classes.forEach((*className*) => element.classList.add(*className*));

      }

      element.innerHTML = `

              <img src=${this.src} alt=${this.alt}>

              <h3 class="menu\_\_item-subtitle">${this.title}</h3>

              <div class="menu\_\_item-descr">${this.descr}</div>

              <div class="menu\_\_item-divider"></div>

              <div class="menu\_\_item-price">

                  <div class="menu\_\_item-cost">Цена:</div>

                  <div class="menu\_\_item-total"><span>${this.price}</span> RUB/день</div>

              </div>

            `;

      this.parent.append(element);

    }

  }

  new MenuCard(

    'img/tabs/vegy.jpg',

    'vegy',

    'Меню "Фитнес"',

    'Меню "Фитнес" - это новый подход к приготовлению блюд: больше свежих овощей и фруктов. Продукт активных и здоровых людей. Это абсолютно новый продукт с оптимальной ценой и высоким качеством!',

    300,

    '.menu .container'

  ).render();

  new MenuCard(

    'img/tabs/post.jpg',

    'post',

    'Меню "Постное"',

    'Меню “Постное” - это тщательный подбор ингредиентов: полное отсутствие продуктов животного происхождения, молоко из миндаля, овса, кокоса или гречки, правильное количество белков за счет тофу и импортных вегетарианских стейков.',

    250,

    '.menu .container'

  ).render();

  new MenuCard(

    'img/tabs/elite.jpg',

    'elite',

    'Меню “Премиум”',

    'В меню “Премиум” мы используем не только красивый дизайн упаковки, но и качественное исполнение блюд. Красная рыба, морепродукты, фрукты - ресторанное меню без похода в ресторан!',

    500,

    '.menu .container'

  ).render();

*// SLIDER*

  const slides = document.querySelectorAll('.offer\_\_slide');

  const prev = document.querySelector('.offer\_\_slider-prev');

  const next = document.querySelector('.offer\_\_slider-next');

  const total = document.querySelector('#total');

  const current = document.querySelector('#current');

  let slideIndex = 1;

  showSlides(slideIndex);

  if (slides.length < 10) {

    total.textContent = `0${slides.length}`;

  } else {

    total.textContent = slides.length;

  }

  function showSlides(*n*) {

    if (*n* > slides.length) {

      slideIndex = 1;

    }

    if (*n* < 1) {

      slideIndex = slides.length;

    }

    slides.forEach((*item*) => (*item*.style.display = 'none'));

    slides[slideIndex - 1].style.display = 'block';

    if (slides.length < 10) {

      current.textContent = `0${slideIndex}`;

    } else {

      current.textContent = slideIndex;

    }

  }

  function plusSlides(*n*) {

    showSlides((slideIndex += *n*));

  }

  prev.addEventListener('click', () => {

    plusSlides(-1);

  });

  next.addEventListener('click', () => {

    plusSlides(1);

  });

});

В репозитории можно ознакомиться с полной информацией этого проекта: [https://github.com/DimkaVeselov/food](#_top)

**Список литературы:**

* Электронный справочник <https://developer.mozilla.org/ru/>
* Электронный справочник <https://webref.ru/>
* Учебник по JavaScript <https://learn.javascript.ru/>
* Справочник <http://javascript.ru/manual>