**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«ВЛАДИКАВКАЗСКИЙ КОЛЛЕДЖ ЭЛЕКТРОНИКИ»**

**ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

ПМ.01 «Разработка модулей программного обеспечения для компьютерных систем»

09.02.07 «Информационные системы»

Выполнил студент \_\_\_\_\_\_\_\_\_\_\_\_ *Гусов А.Х-Б.* Группа *ИС1-19*

подпись Ф.И.О.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_ *Такаев З.И.*

подпись Ф.И.О.

г. Владикавказ

2021 г.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc90589376)

[РАЗРАБОТКА ДИЗАЙН-МАКЕТА (PHOTOSHOP, FIGMA) 5](#_Toc90589377)

[ВЕРСТКА СТРАНИЦЫ (HTML, CSS) 8](#_Toc90589378)

[КЛИЕНТСКОЕ ПРОГРАММИРОВАНИЕ (JAVASCRIPT) 26](#_Toc90589379)

# ВВЕДЕНИЕ

Целью производственной практики было: расширение и закрепление полученных в колледже теоретических и практических знаний.

В этом проекте была поставлена задача, разработать одностраничный сайт для международной службы доставки.

Разрабатываемый одностраничный сайт должен удовлетворять следующим требованиям:

* Обладать гибким и удобным пользовательским интерфейсом;
* **Сайт должен корректно выглядеть в популярных браузерах и их различных версиях.**
* **Верстка должна быть валидной, без лишних нагромождений кода, с минимумом синтаксических ошибок.**
* **Верстка должна быть современной.**
* JavaScript код должен быть максимально читаемым и понятным
* Макет дизайна должен быть предоставлен в формате PSD
* Все слои одного логического элемента должны быть в одной папке

Разработка сайта «международная служба экспресс-доставки» предназначена для предоставления возможности оформить и отследить свою доставку.

**Объект исследования**: работа существующих служб доставки.

**Предмет исследования**: информационная система, предназначенная для работы служб доставки.

**На этапе постановки задачи были решены следующие вопросы:**

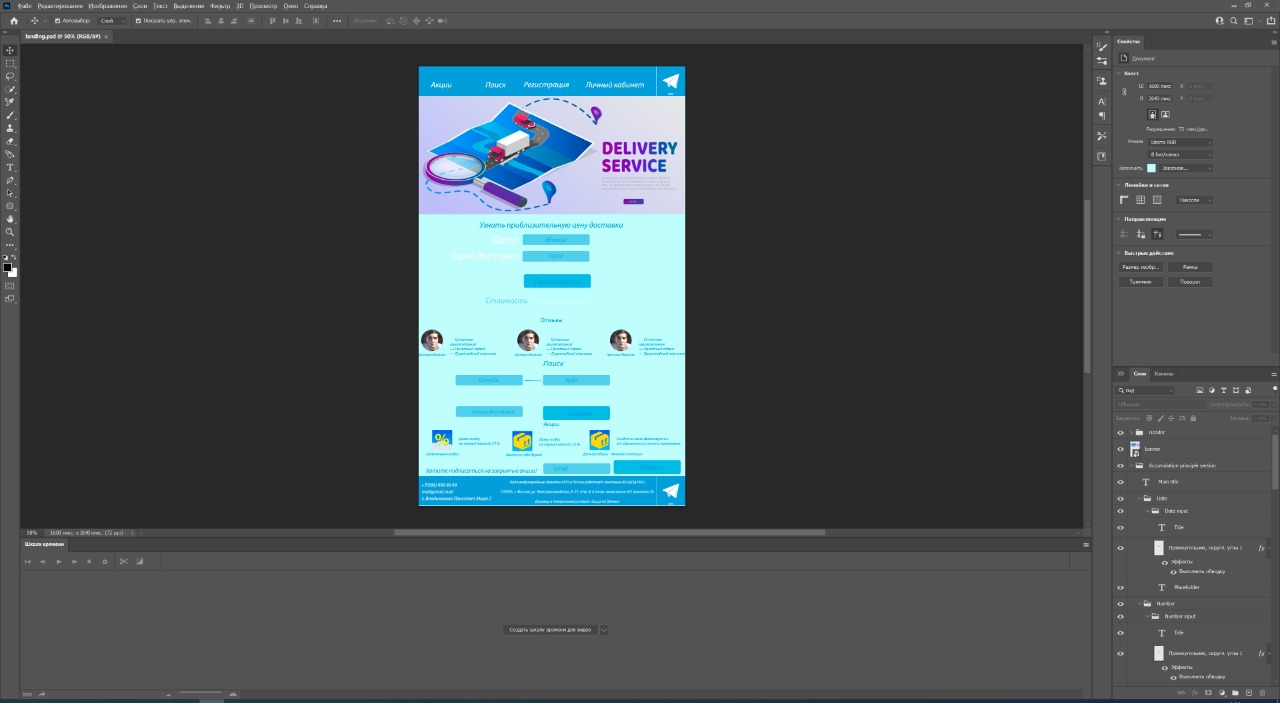
* Целевая аудитория, для которой рассчитано создание этого проекта;
* Основные цели и задачи проекта;
* Создана структура сайта.

**Для достижения этой цели я определил следующие задачи:**

* Рассмотрим основные подходы к созданию дизайн макета для веб-страниц
* Рассмотрим этапы разработки клиентской части веб-сайта.
* Рассмотрим основные области использования языка JavaScript при создании HTML-страниц.

# РАЗРАБОТКА ДИЗАЙН-МАКЕТА (PHOTOSHOP, FIGMA)

При разработке дизайн-макета использовался Photoshop, размеры макета 1600 x 2640 px

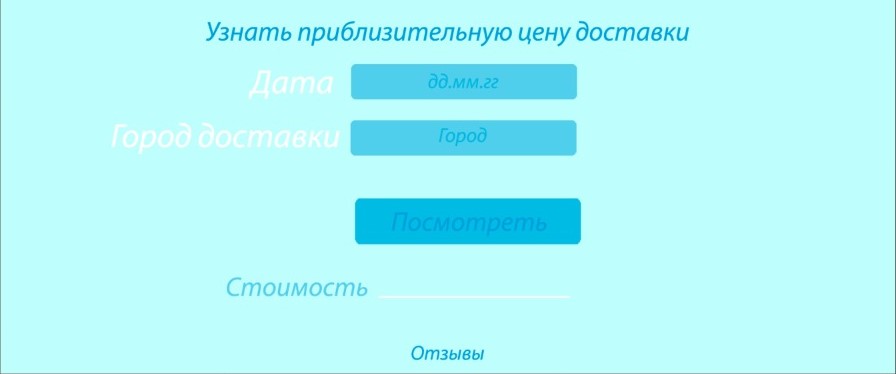


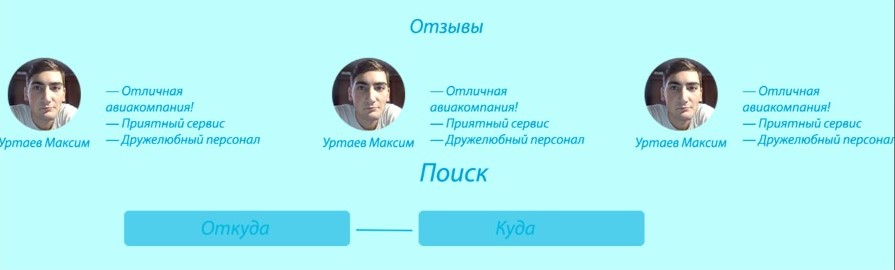
По всем правилам структуры web-сайта, было разделение на Header

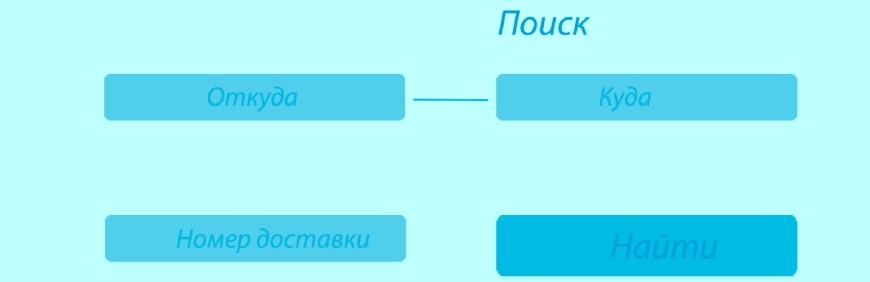


Баннер, который будет служить в будущем как слайдер



Так же была добавлена форма для приблизительной стоимости доставки

Блок с отзывами клиентов

Форма для поиска своего заказа на доставку

Блок акций

Блок для подписки на закрытые акции

Футер с полезной информацией для клиента

# ВЕРСТКА СТРАНИЦЫ (HTML, CSS)

Для web-разметки использовался язык-разметки HTML. Для стилизации внешнего вида web-приложения использовался язык стилей CSS

В теге <head> идет описание сайта, подключение стилей, метатеги, иконка сайте в браузерах и поисковиках

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="icon" href="icons/favicon.ico" type="image/x-icon">

<link rel="shortcut icon" href="icons/favicon.ico" type="image/x-icon">

<link rel="stylesheet" href="styles.css">

<title>DPD</title>

</head>

В теге <body> идут все блоки сайта

Тег <header> содержит в себе разметку шапки сайта

<header class="header">

<div class="header\_\_item\_wrapper">

<div class="header\_\_item">

<a href="#">Акции</a>

</div>

<div class="header\_\_item">

<a href="#">Поиск</a>

</div>

<div class="header\_\_item">

<a href="#">Регистрация</a></div>

<div class="header\_\_item">

<a href="#">Личный кабинет</a>

</div>

</div>

<a href="index.html" class="logo">

<img class="header\_\_logo" src="images/logo.png" alt="logo">

</a>

</header>

К нему применяются стили для фонового цвета, отступов, эффектов при наведении на ссылки

.header {

background-color: #379ed8;

font-size: 48px;

color: #fff;

display: flex;

align-items: center;

justify-content: space-between;

}

.header\_\_item\_wrapper {

display: flex;

align-items: center;

justify-content: space-evenly;

flex-wrap: wrap;

padding: 70px;

padding-bottom: 0;

padding-top: 0;

width: 100%;

}

.header\_\_item a {

color: #fefefe;

text-decoration: none;

font-size: 48px;

}

.header\_\_item a:hover {

opacity: 0.6;

}

.header\_\_item:not(:last-child) {

margin-right: 33px;

После идет блок баннера-слайдера, наполнение которого происходит средствами JavaScript

<section class="banner"></section>

Стили задают позиционирование элементов слайдера, размеры картинок и элементов

.slider {

height: 600px;

position: relative;

/\* margin-bottom: 50px; \*/

}

.slider\_\_inner {

display: flex;

height: 100%;

}

.slider\_\_item {

width: 100%;

display: none;

}

.slider\_\_item.active {

display: block;

}

.slider\_\_item img {

width: 100%;

height: 100%;

object-fit: cover;

}

.slider\_\_arrow {

width: 48px;

height: 48px;

padding: 5px;

background-color: transparent;

border: 1px solid #fff;

border-radius: 50%;

position: absolute;

top: calc(50% - 24px);

cursor: pointer;

}

.slider\_\_arrow--left {

left: 30px;

}

.slider\_\_arrow--right {

right: 30px;

}

.slider\_\_arrow--right img {

transform: rotate(180deg);

}

.slider\_\_arrow img{

width: 32px;

}

.slider\_\_pagination {

position: absolute;

left: 50%;

bottom: 30px;

transform: translateX(-50%);

display: flex;

list-style: none;

}

.slider\_\_pagination li {

margin: 0 5px;

width: 15px;

height: 15px;

border-radius: 50%;

background-color: #fff;

cursor: pointer;

}

.slider\_\_pagination li.active {

background-color: yellow;

}

}

Блок с формой для приблизительной цены доставки

<section class="principle">

<span class="principle\_\_title">Узнать приблизительную цену доставки</span>

<form action="post" class="principle\_\_form" onsubmit="return false;">

<div class="principle\_\_date principle\_\_block ">

<span class="principle\_\_date\_\_title principle\_\_input\_\_title">Дата</span>

<input class="principle\_\_input principle\_\_date\_input" type="date" name="principle\_\_date" placeholder="дд.мм.гг">

</div>

<div class="principle\_\_town principle\_\_block">

<span class="principle\_\_town\_\_title principle\_\_input\_\_title">Город</span>

<input class=" principle\_\_input principle\_\_town\_input" type="text" name="principle\_\_town" placeholder="Город">

</div>

<div class="principle\_\_block principle\_\_submit">

<input type="submit" value="Посмотреть" class="principle\_\_submit\_\_btn">

</div>

</form>

<div class="principle\_\_cost">

<div class="principle\_\_cost\_\_title">

<span>Стоимость</span>

</div>

<div class="principle\_\_cost\_\_line"></div>

</div>

</section>

С стилями для внешнего вида полей ввода, заголовков и кнопки

.principle {

display: flex;

flex-flow: column;

justify-content: center;

align-items: center;

}

.principle\_\_title {

font-size: 48px;

color: #379ed8;

margin-bottom: 33px;

display: inline-block;

}

.principle\_\_block {

display: flex;

justify-content: center;

align-items: center;

flex-wrap: wrap;

}

.principle\_\_date {

margin-bottom: 39px;

}

.principle\_\_town {

margin-bottom: 78px;

}

.principle\_\_input\_\_title {

color: #fefefe;

font-size: 40px;

margin-right: 33px;

}

.principle\_\_input {

border-radius: 10px 10px 10px 10px;

background: #84ceea;

height: 60px;

width: 400px;

border: none;

font-size: 26px;

text-align: center;

color: #379ed8;

cursor: pointer;

color: #379ed8;

font-size: 26px;

text-align: center;

}

.principle\_\_submit {

width: 400px;

height: 80px;

border-radius: 10px 10px 10px 10px;

background-color: #65bbe3;

margin: 0px 0px 60px auto;

}

.principle\_\_submit\_\_btn {

border-radius: 10px 10px 10px 10px;

width: 400px;

height: 80px;

background-color: #65bbe3;

font-size: 48px;

text-align: center;

color: #379ed8;

border: none;

cursor: pointer;

}

.principle\_\_submit\_\_btn:hover {

opacity: 0.6;

}

.principle\_\_cost {

margin-bottom: 86px;

display: flex;

align-items: flex-end;

justify-content: space-between;

}

.principle\_\_cost\_\_title {

font-size: 48px;

color: #84ceea;

margin-right: 18px;

}

.principle\_\_cost\_\_line {

width: 340px;

height: 3px;

border-bottom: 3px solid #fefefe;

}

Блок с отзывами клиентов

<section class="reviews">

<span class="reviews\_\_title">

Отзывы

</span>

<div class="reviews\_\_items\_\_wrapper">

<div class="reviews\_\_item reviews\_\_item\_\_first">

<div class="reviews\_\_item\_\_content">

<div class= " reviews\_\_item\_\_photo reviews\_\_item\_\_photo\_\_first"></div>

<span class="reviews\_\_item\_\_name">Бондаренко Руслан</span>

</div>

<ul class="reviews\_\_text">

<li class="reviews\_\_text\_\_item">Отличная

авиакомпания!</li>

<li class="reviews\_\_text\_\_item">Приятный сервис</li>

<li class="reviews\_\_text\_\_item">Дружелюбный персонал</li>

</ul>

</div>

<div class="reviews\_\_item reviews\_\_item\_\_second">

<div class="reviews\_\_item\_\_content">

<div class=" reviews\_\_item\_\_photo reviews\_\_item\_\_photo\_\_second"></div>

<span class="reviews\_\_item\_\_name">Уртаев Максим</span>

</div>

<ul class="reviews\_\_text">

<li class="reviews\_\_text\_\_item">Отличная

авиакомпания!</li>

<li class="reviews\_\_text\_\_item">Приятный сервис</li>

<li class="reviews\_\_text\_\_item">Дружелюбный персонал</li>

</ul>

</div>

<div class="reviews\_\_item reviews\_\_item\_\_third">

<div class="reviews\_\_item\_\_content">

<div class="reviews\_\_item\_\_photo reviews\_\_item\_\_photo\_\_third"></div>

<span class="reviews\_\_item\_\_name">Рубаев Давид</span>

</div>

<ul class="reviews\_\_text">

<li class="reviews\_\_text\_\_item">Отличная

авиакомпания!</li>

<li class="reviews\_\_text\_\_item">Приятный сервис</li>

<li class="reviews\_\_text\_\_item">Дружелюбный персонал</li>

</ul>

</div>

</div>

</section>

Стили используются для как для размера текста, цвета так же размеров и округления фото клиентов

.reviews {

display: flex;

flex-direction: column;

flex-wrap: wrap;

align-items: center;

margin-bottom: 29px;

}

.reviews\_\_items\_\_wrapper {

display: flex;

flex-wrap: wrap;

align-items: center;

justify-content: space-between;

}

.reviews\_\_title {

font-size: 36px;

color: #379ed8;

display: inline-block;

}

.reviews\_\_item {

display: flex;

align-items: center;

}

.reviews\_\_item:not(:last-child) {

margin-right: 128px;

}

.reviews\_\_item\_\_photo {

width: 200px;

height: 200px;

border-radius: 50%;

-moz-border-radius: 50%;

-webkit-border-radius: 50%;

-khtml-border-radius: 50%;

background-repeat: no-repeat;

background-size: cover;

background-position: center;

}

.reviews\_\_item\_\_photo\_\_first {

background-image: url(reviews\_images/1.jpg);

}

.reviews\_\_item\_\_photo\_\_second {

background-image: url(reviews\_images/2.jpg);

}

.reviews\_\_item\_\_photo\_\_third {

background-image: url(reviews\_images/3.jpg);

}

.reviews\_\_item\_\_name {

display: inline-block;

font-size: 24px;

color: #379ed8;

text-align: center;

}

.reviews\_\_text {

list-style: none;

}

.reviews\_\_text\_\_item {

font-size: 24px;

color: #379ed8;

}

.reviews\_\_text\_\_item:before {

content: "—";

position: relative;

left: -5px;

}

.reviews\_\_item\_\_content {

display: flex;

justify-content: center;

flex-direction: column;

margin-right: 28px;

}

Блок с формой для поиска доставки

<section class="search">

<span class="search\_\_title">Поиск</span>

<div class="search\_\_form">

<div class="search\_\_path">

<input type="text" placeholder="Откуда" class="search\_\_path\_\_from search\_\_input">

<div class="search\_\_path\_\_line"></div>

<input type="text" placeholder="Куда" class="search\_\_path\_\_to search\_\_input">

</div>

<div class="search\_\_end">

<input type="text" placeholder="Номер доставки" class="search\_\_input">

<input type="submit" value="Найти" class="search\_\_btn">

</div>

</div>

</section>

Стили используются для размера текста, цвета, внешнего оформления полей ввода и кнопки

.search {

display: flex;

flex-direction: column;

align-items: center;

justify-content: space-between;

margin-bottom: 15px;

}

.search\_\_path {

display: flex;

align-items: center;

justify-content: center;

margin-bottom: 128px;

}

.search\_\_title {

font-size: 36px;

color: #379ed8;

display: inline-block;

margin-bottom: 52px;

}

.search\_\_input {

width: 400px;

height: 61px;

border: none;

border-radius: 10px 10px 10px 10px;

background-color: #84ceea;

font-size: 36px;

text-align: center;

font-size: 36px;

color: #5db6e1;

text-align: center;

}

.search\_\_path\_\_line {

width: 100px;

border-bottom: 3px solid #63b7e1;

margin: 0 12px 0 12px;

}

.search\_\_end {

display: flex;

align-items: center;

justify-content: center;

}

.search\_\_btn {

margin-left: 123px;

width: 400px;

height: 80px;

border-radius: 10px 10px 10px 10px;

background-color: #65bbe3;

font-size: 48px;

color: #379ed8;

border: none;

cursor: pointer;

}

.search\_\_btn:hover {

opacity: 0.6;

}

Блок с акциями

<section class="actions">

<span class="actions\_\_title">Акции</span>

<div class="actions\_\_wrapper">

<div class="action">

<div class="action\_\_content">

<div class="action\_\_image action\_\_image\_\_first"></div>

<span class="action\_\_title">Заоблачные скидки</span>

</div>

<span class="action\_\_text">

Даем скидку на<br>первый заказ до 25 %.

</span>

</div>

<div class="action">

<div class="action\_\_content">

<div class="action\_\_image action\_\_image\_\_second"></div>

<span class="action\_\_title">Много на себя берем!</span>

</div>

<span class="action\_\_text">

Даем скидку <br>на первый заказ до 25 %.

</span>

</div>

<div class="action">

<div class="action\_\_content">

<div class="action\_\_image action\_\_image\_\_third"></div>

<span class="action\_\_title">Дальше едешь – меньше платишь</span>

</div>

<span class="action\_\_text">

Скидка на заказ формируется<br>от удаленности пункта назначения.

</span>

</div>

</div>

</section>

Стили для оформления текста акции и картинок

.actions {

display: flex;

flex-direction: column;

align-items: center;

margin-bottom: 27px;

}

.action {

display: flex;

align-items: center;

justify-content: space-between;

flex-wrap: wrap;

}

.actions\_\_title {

font-size: 36px;

color: #379ed8;

}

.actions\_\_wrapper {

display: flex;

align-items: center;

justify-content: space-evenly;

flex-wrap: wrap;

}

.action\_\_content {

display: flex;

flex-direction: column;

align-items: center;

justify-content: space-between;

flex-wrap: wrap;

}

.action\_\_title {

font-size: 24px;

color: #379ed8;

}

.action\_\_image {

width: 120px;

height: 120px;

background-repeat: no-repeat;

background-size: cover;

background-position: center;

}

.action\_\_image\_\_first {

background-image: url(actions\_images/action1.png)

}

.action\_\_image\_\_second {

background-image: url(actions\_images/action2.png)

}

.action\_\_image\_\_third {

background-image: url(actions\_images/action3.png)

}

.action\_\_text {

font-size: 24px;

color: #379ed8;

}

# Блок с формой для подписки на закрытые акции

<section class="closedAction">

<span class="closedAction\_\_title">Хотите подписаться на закрытые акции?</span>

<form action="post" method="post" onsubmit="return false;" class="closedActionForm">

<input type="text" placeholder="Email" class="closedActionForm\_\_input">

<input type="submit" value="Найти" class="closedActionForm\_\_btn">

</form>

</section>

Стили задают внешний вид тексу, полю ввода почты и кнопки

.closedAction {

display: flex;

align-items: center;

justify-content: space-around;

flex-wrap: wrap;

}

.closedAction\_\_title {

font-size: 36px;

color: #379ed8;

}

.closedActionForm {

display: flex;

align-items: center;

flex-wrap: wrap;

justify-content: space-between;

margin-bottom: 13px;

}

.closedActionForm\_\_input {

width: 400px;

height: 61px;

background-color: #65bbe3;

border-radius: 10px 10px 10px 10px;

border: none;

font-size: 36px;

color: #379ed8;

text-align: center;

margin-right: 22px;

}

.closedActionForm\_\_btn {

height: 80px;

width: 400px;

border-radius: 10px 10px 10px 10px;

border: none;

background-color: #65bbe3;

font-size: 48px;

color: #379ed8;

text-align: center;

cursor: pointer;

}

.closedActionForm\_\_btn:hover {

opacity: 0.6;

}

Футер сайта

<footer class="footer">

<div class="footer\_\_info">

<span class="footer\_\_info\_\_text">+7(999) 999 99 99</span>

<span class="footer\_\_info\_\_text">mail@mail.mail</span>

<span class="footer\_\_info\_\_text">г. Владикавказ Проспект Мира 2</span>

</div>

<span class="footer\_\_text">

Под международным брендом DPD в России работает компания АО «ДПД РУС» <br>107023, г. Москва, ул. Электрозаводская, д. 27, стр. 8, 4 этаж, помещение ХVI, комната<br>34Договор и Генеральные условия • Защита данных

</span>

<a href="index.html" class="logo">

<img class="footer\_\_logo" src="images/logo.png" alt="logo">

</a>

</footer>

Стили задают внешний вид тексту и логотипу

.footer {

background-color: #379ed8;

display: flex;

align-items: center;

justify-content: space-between;

flex-wrap: wrap;

}

.footer\_\_info {

display: flex;

flex-direction: column;

align-items: flex-start;

justify-content: center;

flex-wrap: wrap;

margin-left: 15px;

}

.footer\_\_info\_\_text {

font-size: 30px;

color: #fefefe;

}

.footer\_\_text {

font-size: 24px;

color: #fefefe;

}

.footer\_\_logo {

width: 100%;

height: 100%;}

# КЛИЕНТСКОЕ ПРОГРАММИРОВАНИЕ (JAVASCRIPT)

Заключительный этап разработки - логика сайта. На этом этапе я создал слайдер. Для реализации данной идеи использовался язык веб-разработки «JavaScript». Перед нами стоит следующая задача: Получить блок слайдера, который изначально пустой и заполнить его картинками из папки, так же добавить пагинацию и стрелки переключения. Вкратце логику написанного кода можно описать так:

Пишем функцию которая создает слайдер. В ней мы создаем элемент DOM-дерева, сам слайдер и добавляем его в блок с классом 'banner'. Создаем блок обертку для элементов слайдера с классом 'slider\_\_inner'. Далее с помощью цикла for создаем 4 элемента слайдера, даем им класс‘'slider\_\_item' и картинки. Далее функция для активного слайда. После идет функция для смены слайда, которая так же добавляет автосмену слайдов. После идут две функции для следующего и предыдущего слайда, которые циклично работают. После функция для создания стрелок и точек для переключения

const banner = document.querySelector('.banner')

function createSlider (id = 'slider') {

const slides = []

const slider = document.createElement('div')

slider.classList.add('slider')

slider.id = id

banner.append(slider)

const sliderInner = document.createElement('div')

sliderInner.classList.add('slider\_\_inner')

slider.append(sliderInner)

for (let i = 1; i < 4; i++) {

const sliderItem = document.createElement('div')

sliderItem.classList.add('slider\_\_item')

const img = document.createElement('img')

img.src = 'banner\_images/' + i + '.jpg'

img.alt = i

sliderItem.append(img)

sliderInner.append(sliderItem)

slides.push(sliderItem)

}

const activeSlide = () => {

slides.forEach((slide, index) => {

if (currentSlide === index) {

slide.classList.add('active')

} else {

slide.classList.remove('active')

}

})

}

const switchSlide = () => {

activeSlide()

activeDot()

setInterval(next, 5000)

}

const next = () => {

if (currentSlide === slides.length - 1) {

currentSlide = 0

} else {

currentSlide++

}

switchSlide()

}

const prev = () => {

if (currentSlide === 0) {

currentSlide = slides.length - 1

} else {

currentSlide--

}

switchSlide()

switchSlide()

}

function createArrow(parent, previos = true) {

const arrow = document.createElement('div')

const arrowImg = document.createElement('img')

arrowImg.src = 'icons/arrow.svg'

arrowImg.alt = 'arrow'

arrow.append(arrowImg)

if (previos) {

newClass = 'slider\_\_arrow--left'

arrow.addEventListener('click', prev)

}

else {

arrow.addEventListener('click', next)

newClass = 'slider\_\_arrow--right'

}

arrow.classList.add('slider\_\_arrow')

arrow.classList.add(newClass)

parent.append(arrow)

}

function createPagination () {

const pagination = document.createElement('ul')

pagination.classList.add('slider\_\_pagination')

for (let i = 0; i < slides.length; i++) {

const dot = document.createElement('li')

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

currentSlide = i

switchSlide()

})

pagination.append(dot)

}

return pagination

}

createArrow(slider, true)

createArrow(slider, false)

const pagination = createPagination()

slider.append(pagination)

let currentSlide = 0

const dots = pagination.childNodes

const activeDot = () => {

dots.forEach((dot, index) => {

if (currentSlide === index) {

dot.classList.add('active')

} else {

dot.classList.remove('active')

}

})

}

activeSlide()

activeDot()

}

createSlider ('slider')