|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**КУРСОВАЯ РАБОТА**

по дисциплине: Разработка клиентских частей интернет-ресурсов

по профилю:

направления профессиональной подготовки: 09.03.04 «Программная инженерия»

Тема: Клиентская часть интернет-ресурса «Киберспортивная организация»

Студент: Шутко Егор Дмитриевич

Группа: ИКБО-20-21

Работа представлена к защите\_\_\_\_\_\_\_\_\_\_\_(дата)\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

(подпись и ф.и.о. студента)

Руководитель: старший преподаватель, Синицын Анатолий Васильевич

Работа допущена к защите\_\_\_\_\_\_\_\_\_\_\_(дата)\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

(подпись и ф.и.о. рук-ля)

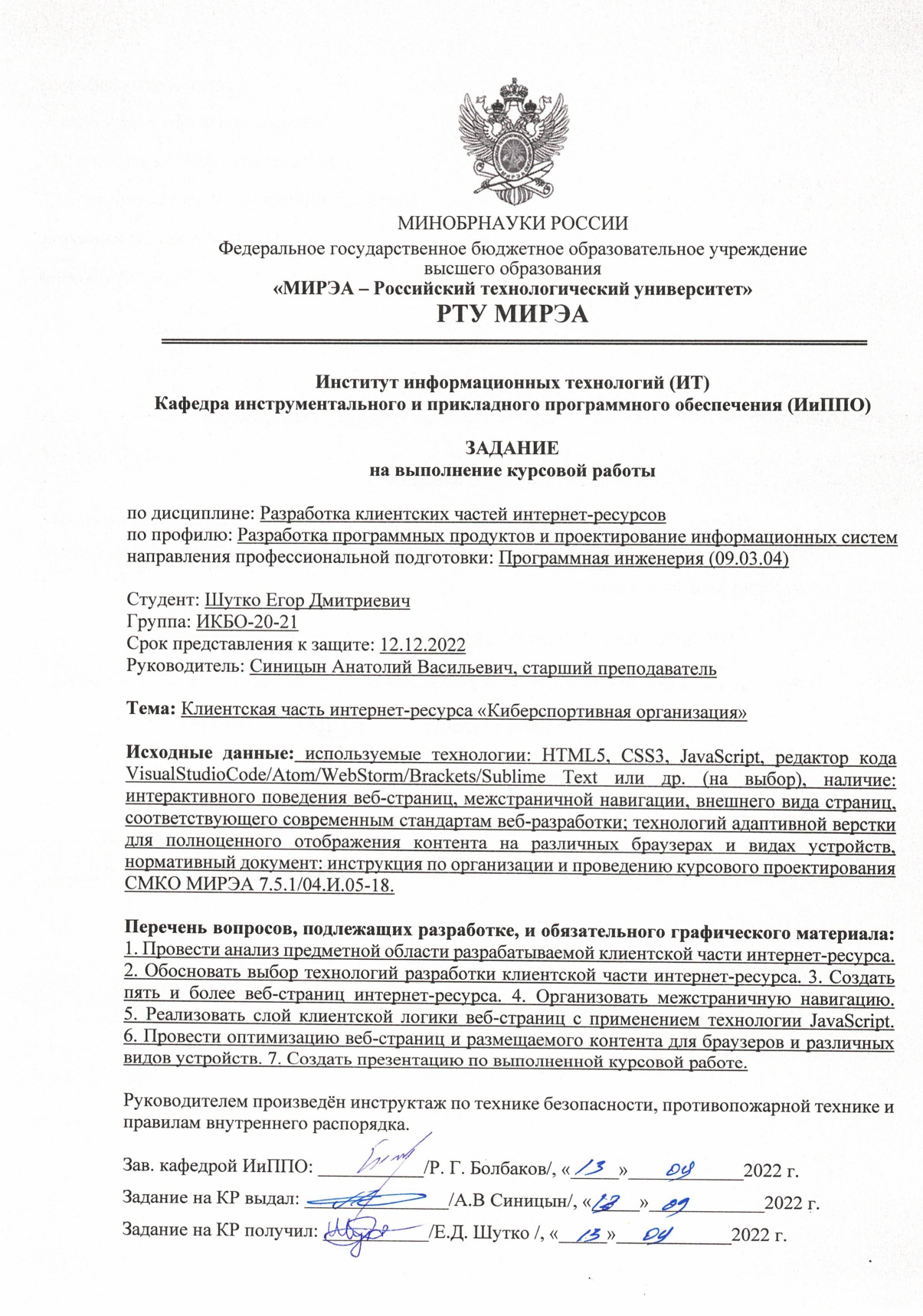
Оценка по итогам защиты: \_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /

(подписи, дата, ф.и.о., должность, звание, уч. степень двух преподавателей, принявших защиту)

М. РТУ МИРЭА. 2022 г.



**Оглавление**

[Введение 4](#_Toc121603098)

[1. Общие сведения 5](#_Toc121603099)

[2. описание логистической структуры 6](#_Toc121603100)

[2.1 Анализ предметной области разрабатываемой клиентской части интернет ресурса 6](#_Toc121603101)

[2.2 Выбор технологий разработки клиентской части интернет-ресурса 7](#_Toc121603102)

[2.3 Создание веб-страниц клиентской части интернет-ресурса с использованием технологий HTML5, CSS3 и JavaScript 8](#_Toc121603103)

[2.3.1 Каркас веб-страниц 8](#_Toc121603104)

[2.3.2 Основная страница «main» 9](#_Toc121603105)

[2.3.3 Страница «partners» 9](#_Toc121603106)

[2.3.4 Страница «roster» 10](#_Toc121603107)

[2.3.5 Страница «news» 11](#_Toc121603108)

[2.3.6 Страница «shop» 12](#_Toc121603109)

[2.3.7 Страница «media» 13](#_Toc121603110)

[2.4 Создание межстраничной навигации 15](#_Toc121603111)

[2.5 Реализация слоя клиентской логики веб-страницы с применением технологий JavaScript 15](#_Toc121603112)

[2.5.1 Реализация «каптчи» на странице «Партнеры» 15](#_Toc121603113)

[2.5.2 Создание карточек с товарами из массива на странице «Мерч» 18](#_Toc121603114)

[3. Оптимизация клиентской части интернет-ресурса 24](#_Toc121603115)

[3.1 Оптимизация клиентской части интернет-ресурса для основных браузеров 25](#_Toc121603116)

[3.2 Оптимизация клиентской части интернет-ресурса для стационарных и мобильных устройств 25](#_Toc121603117)

[Заключение 27](#_Toc121603118)

[Список источников 29](#_Toc121603119)

# Введение

В настоящее время интернет распространился во все области нашей жизни. Теперь почти у каждой организации среднего уровня есть свой сайт для введения бизнеса через Интернет, а также для комфортного взаимодействия с клиентами в сети. Трудно представить, что компания занимающаяся продажей чего-либо не используют интернет-ресурсы для получения дополнительной прибыли с интернет-аудитории. Интернет-ресурс позволяет бизнесу своевременно принимать и отправлять заказы, информировать клиентов об изменении режима работы, а также проверять наличие товара клиенту. Умелое использование современных веб-технологий несомненно даёт преимущество перед конкурентами. Отсутствие у организации собственного интернет-ресурса можно назвать критической точкой в развитии компании, при этом большинство малого бизнеса, в число которых и входит среднестатистический цветочный магазин, не понимают этого и из-за этого не могут развиваться и успешно конкурировать.

Целью данной курсовой работы является разработка интернет-ресурса “Цветочный магазин” с использованием технологий HTML5, CSS3, JavaScript.

Для упрощения разработки процесс был поделён на несколько частей:

* Изучение существующих интернет-ресурсов с данной тематикой. Выявления основных элементов интернет-ресурса. Выбор наиболее удобных технологий.
* Создание веб-страниц клиентской части интернет-ресурса. Создание межстраничной навигации. Оптимизация под различные устройства.
* Реализация слоя клиентской логики веб-страниц.

В результате интернет-ресурс должен обладать интерактивным поведением веб-страниц, удобной межстраничной навигацией, а также внешнем видом страниц соответствующим современным стандартам веб-разработки.

# Общие сведения

Ресурс является сайтом киберспортивной организации «Improve Esports».

Использование всей информации о данной организации было согласованно с ее CEO (Chief Executive Officer, Главный Исполнительный Директор).

Данный интерент-ресурс помогает пользователю ознакомится с основной информацией о данной организации, а так же прочитать последние новости или приобрести товары с дизайном команды, которые так же называются «мерчом».

Для разработки интернет-ресурса было решено воспользоваться редактором кода «Visual Stiduo Code» («VS Code»), разработанный Microsoft для Windows. Для верстки использовался HTML5, для создания дизайна страниц CSS3, а клиентская логика написана на JavaScript. Функционал и внешний вид были проверены в большинстве современных популярных браузерах: Google Chrome, Opera, Mozilla Firefox, Microsoft Edge. Использование такого программного обеспечения и языков программирования обуславливается их простотой, универсальностью и надежностью, что позволяет с уверенностью их применять в современных интернет ресурсах.

# описание логистической структуры

## 2.1 Анализ предметной области разрабатываемой клиентской части интернет ресурса

Для реализации данного интернет-ресурса, необходимо исследовать аналоги и проанализировать структуру похожих по тематике интернет-ресурсов.

Для анализа использовались сайты таких киберспортивных организаций, как: “Natus Vinsere”, “Gambit”, “Sentinels”. Первое, что бросается в глаза – это яркая цветовая гамма каждого сайта, выполненная в цветах команды. К примеру: сайт команды “Natus Vinsere” выполнен в черно-желтых оттенках, именно черный и желтый являются главными цветами команды и по совместительству единственными цветами логотипа. Из этого можно сделать вывод, что сайт должен иметь яркую цветовую гамму, составленную из основных оттенков логотипа.

Далее предметом анализа были основные вкладки интернет-ресурсов. После тщательного просмотра были выявлены критические пункты, которые должны быть реализованы:

* Вкладка с главной информацией и историей команды
* Вкладка со спонсорами и партнёрами
* Вкладка с составами по разным дисциплинам
* Вкладка с новостями и\или медиа, связанные с командой
* Вкладка с магазином по продаже «мерча»

Перечисленные выше пункты является обязательными для выполнения. После анализа был создан примерный макет интернет-ресурса, который содержит: основную страницу с самой главной информацией об организации, которая так же содержит ее историю, страница с перечислением всех спонсоров и партнеров команды, страница с полным составом по дисциплине, а также описание каждого из игроков с ссылками на сего социальные сети, страница с новостями, которые причастны к организации, страница с магазином товаров с дизайном команды для фанатов. Также необходимо реализовать меню сайта для удобной навигации и нижнюю часть, в которой указать необходимые контакты.

В результате меню содержит в себе 6 вкладок: главная, партнеры, состав, новости, мерч, медиа. Ниже идет основная часть страницы, наполненные которой зависит от выбранной страница. В самом низу находится так называемый «футер» (от англ. Footer – нижняя часть), в которой находятся контакты для связи.

Так же есть форма для отправки откликов на вакансии, которые открыты для набора в организацию.

## 2.2 Выбор технологий разработки клиентской части интернет-ресурса

Для данной работы были выбраны технологии, представленные в ходе прохождения учебной программы, среди них HTML5, CSS3 и JavaScript. Также был выбран текстовый редактор кода Visual Studio Code в качестве среды разработки и Microsoft Edge для отображения контента во время разработки.

## **Создание веб-страниц клиентской части интернет-ресурса с использованием технологий HTML5, CSS3 и JavaScript**

Все веб-страницы клиентской части интернет-ресурса выстраивались с помощью единого шаблона. Данный шаблон определяет структуру по которой будут выстроены страницы, а также основные цвета используемые на всей странице.

Каждая веб-страница выстроена по структуре «хедер - основная часть - футер». Каждый такой отдельный элемент помещён в собственный блок для удобства разработки.

## Каркас веб-страниц

Хедер, или «шапка» страницы, представляет собой верхнюю часть сайта с логотипом команды, а так же навигационном меню, с помощью которого пользователь может перемещаться по всем страницам интернет-ресурса (рис. 2.3.1.1).

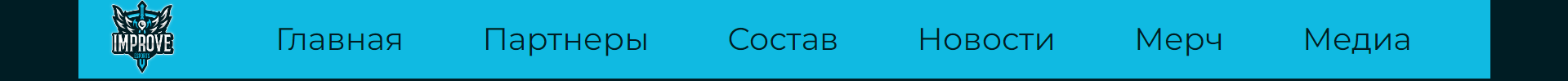


Рисунок 2.3.1.1 - Скриншот хэдера страницы.

Футер или же «подвал» страницы включает в себя контактную информацию для пользователя, с помощью которой можно связаться с генеральным директором команды по вопросам спонсорства или любым другим вопросам (рис. 2.3.1.2).



Рисунок 2.3.1.2 - Скриншот футера страницы.

Между хедером и футером находится основная часть сайта, именуемая как «body», тело сайта, где содержится вся основная информация.

## Основная страница «main»

Данная страница является основной на сайте, тк это та страница, которую

посещает пользователь, впервые попавший на сайт.

На этой страницы размещен девиз и слоган команды «Improve Esports», а так же информация для потенциальных работников команды, а так же спонсоров (рис 2.3.2.1).

Также данная страница содержит хедер и футер для быстрого доступа к информации и переключения между страницами.

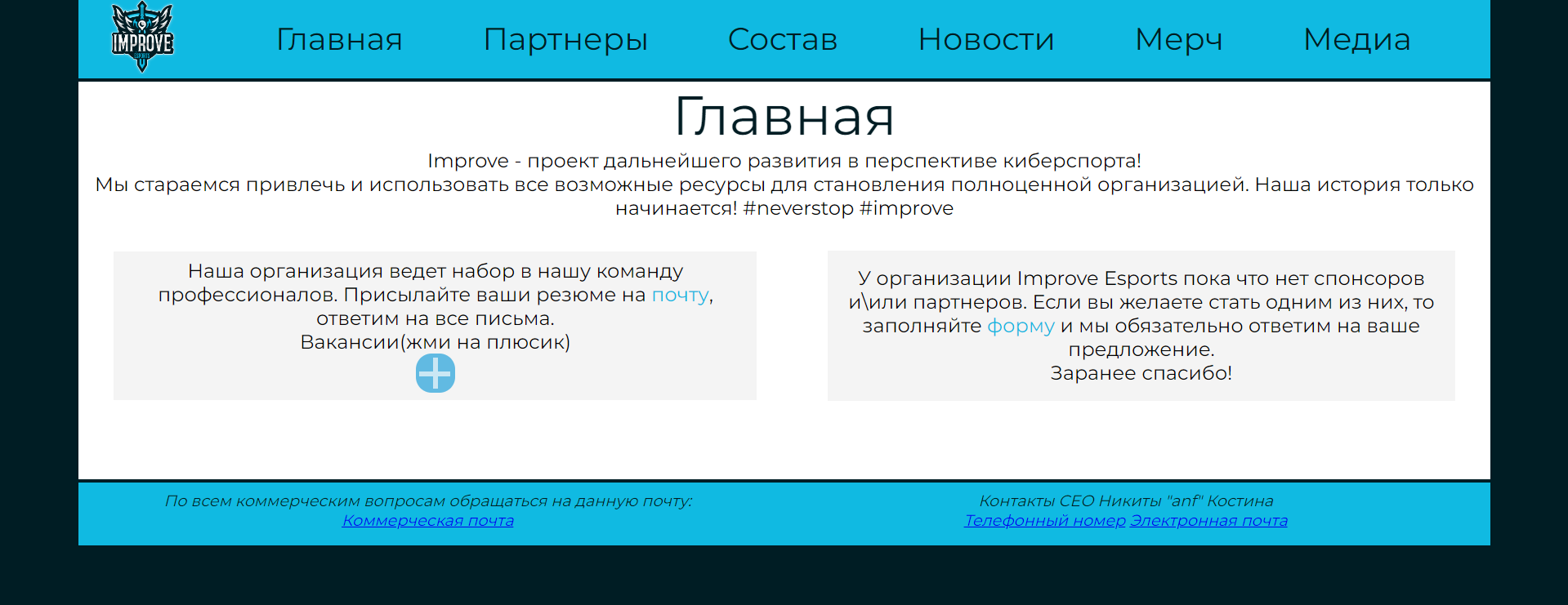


Рисунок 2.3.2.1 – Скриншот страницы «Главная».

## Страница «partners»

На странице «Партнеры» расположена форма, которая необходима для поиска новых спонсоров. Потенциальный партнер команды заполняет контактную информацию, а так же вкратце описывает преложение, и после прохождение каптчи может отправить данное предложение организации (рис 2.3.3.1).

Также данная страница содержит хедер и футер для быстрого доступа к информации и переключения между страницами.

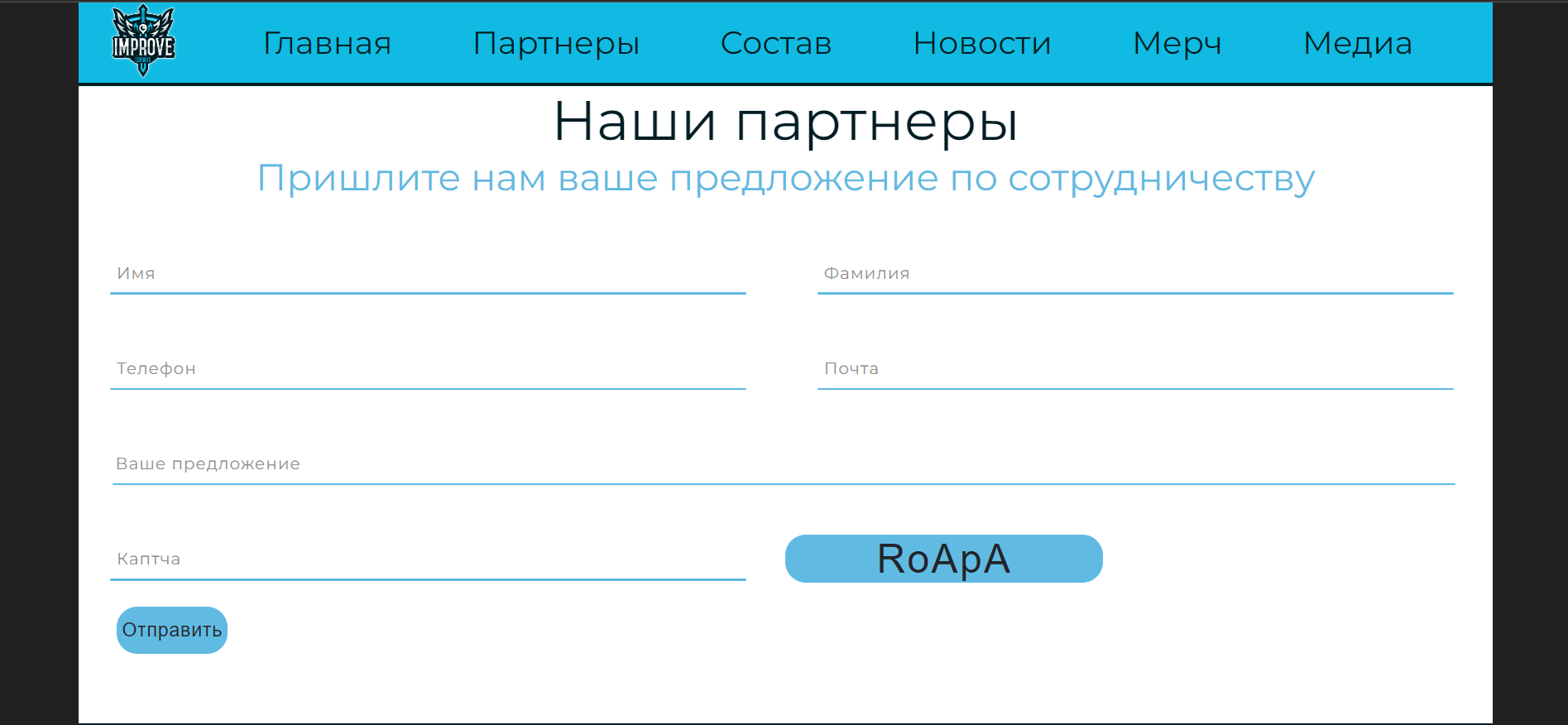


Рисунок 2.3.3.1 – Скриншот страницы «Партнеры».

## Страница «roster»

На данной странице расположена информация о текущем составе команды в виде карточек каждого игрока (рис 2.3.4.1), при нажатии на которые появляется секция с более подробной информацией об игроке, а так же ссылками на его социальные сети.

Также данная страница содержит хедер и футер для быстрого доступа к информации и переключения между страницами.



Рисунок 2.3.4.1 – Скриншот страницы «Состав» с карточками всех игроков.



Рисунок 2.3.4.2 – Скриншот страницы «Состав» с полной информацией об одном из игроков.

## Страница «news»

На данной странице находятся все свежие новости о команде и на сейчас там находится видео с анонсом нового состава команды (рис. 2.3.5.1 и рис. 2.3.5.2).

Также данная страница содержит хедер и футер для быстрого доступа к информации и переключения между страницами.

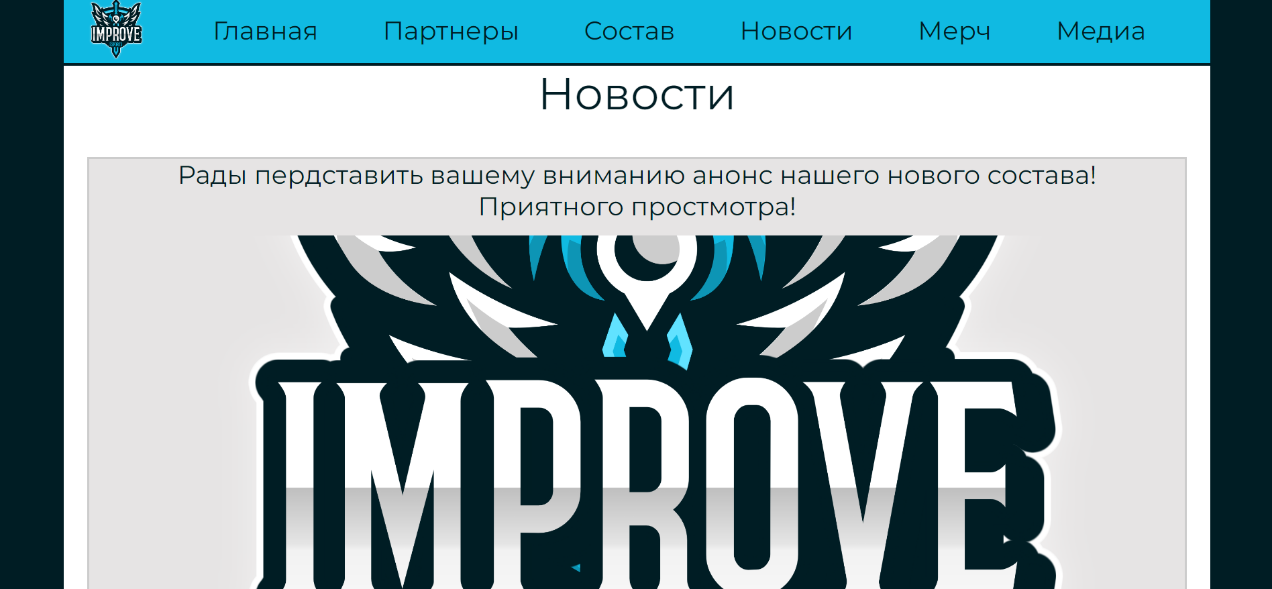


Рисунок 2.3.5.1 – Скриншот страницы «Новости» часть 1.



Рисунок 2.3.5.2 – Скриншот страницы «Новости» часть 2.

## Страница «shop»

Данная страница демонстрирует весь «мерч» команды, который сейчас находится в продаже. Страница состоит из блока с фильтрами по цветам, модели одежды, а также ценнику (рис. 2.3.6.1) и из блока с самими товарами (рис. 2.3.6.2). При наведении на карточку с товаром появляется более подробная информация об этой единице мерча, а именно ценна и состав.

Также данная страница содержит хедер и футер для быстрого доступа к информации и переключения между страницами.

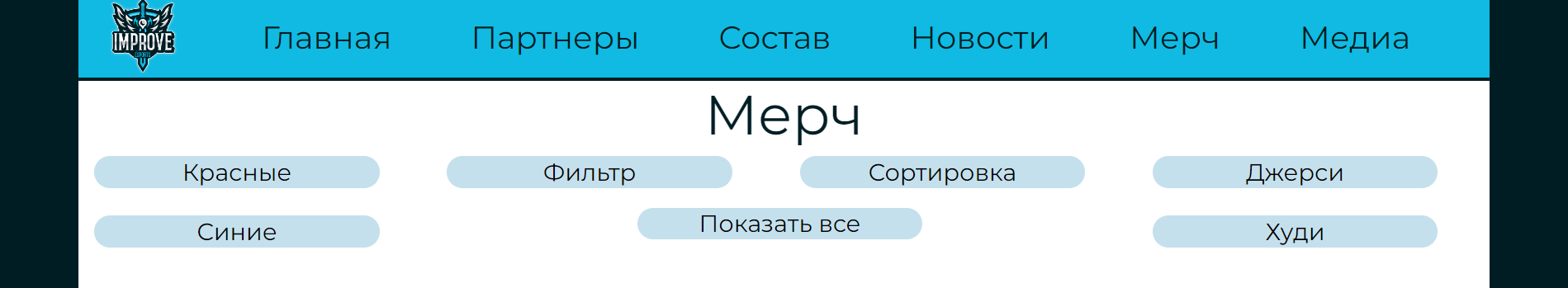


Рисунок 2.3.6.1 – Скриншот страницы «Мерч» с фильтрами для товаров.

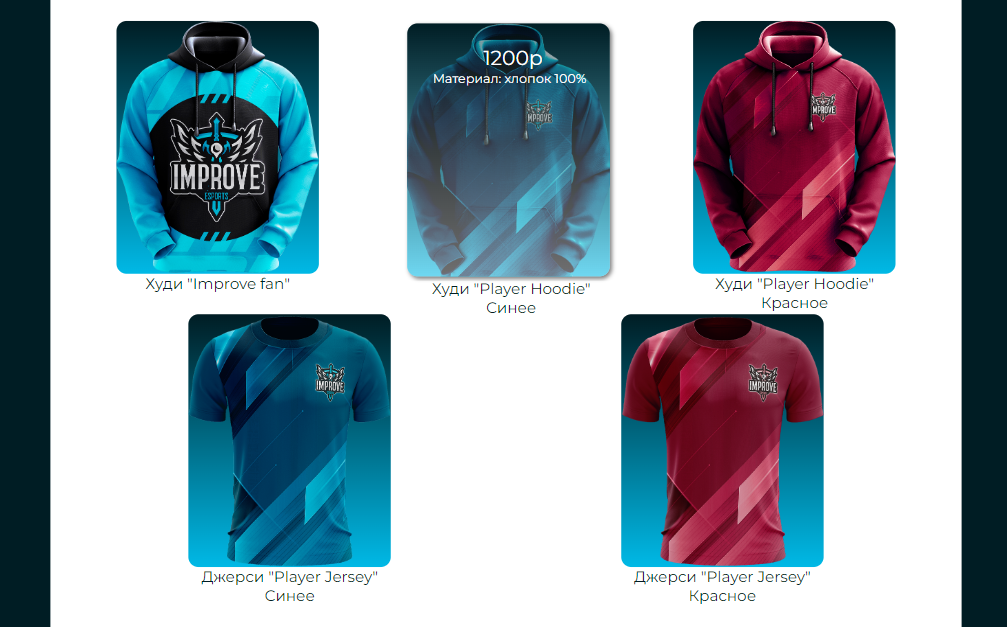


Рисунок 2.3.6.2 – Скриншот страницы «Мерч» с товарами.

## Страница «media»

Последняя страница сайта является галерей с фотографиями игроков разных команд по дисциплинам «Valorant» и «CS:GO», а также арен для проведения турниров по дисциплине «Valorant» (рис. 2.3.7.1 и рис. 2.3.7.2).

Также данная страница содержит хедер и футер для быстрого доступа к информации и переключения между страницами.

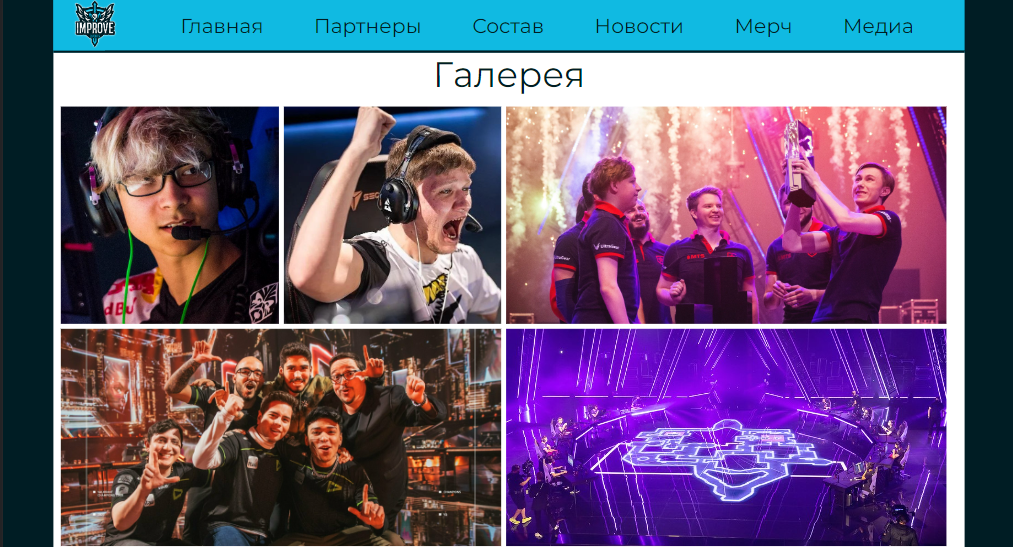


Рисунок 2.3.7.1 – Скриншот страницы «Медиа» часть 1.

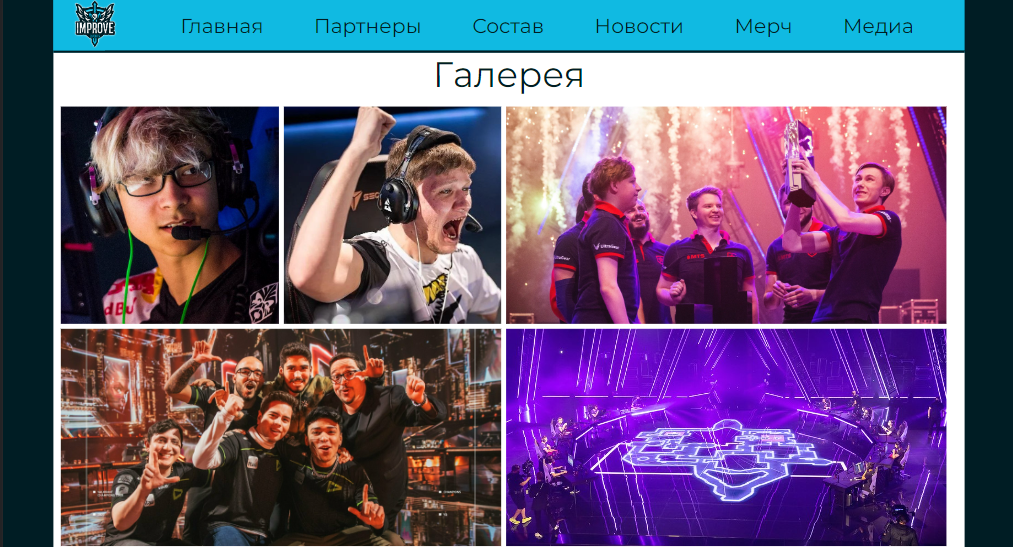


Рисунок 2.3.7.1 – Скриншот страницы «Медиа» часть 2.

## Создание межстраничной навигации

Как говорилось раннее, вся межстраничная навигация находится в «хедере» страницы, а так же с каждой страницы интернет-ресурса можно попасть на любую другую, следовательно схема навигации выглядит так (рис 2.4).

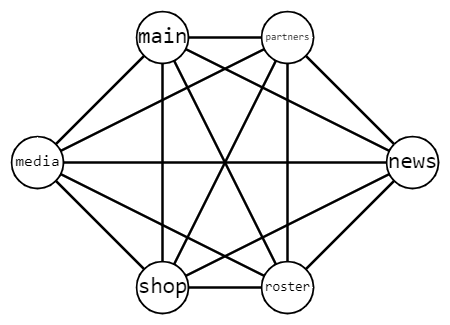


Рисунок 2.4 – Структура межстраничной навигации.

## Реализация слоя клиентской логики веб-страницы с применением технологий JavaScript

## Реализация «каптчи» на странице «Партнеры»

Для отправки предложения по партнерству на странице «Партнеры» существует каптча, чтобы избежать спама. Для ее реализации было создано поле для ввода каптчи, а также кнопка (листинг 2.5.1.1), надпись которой изменяется с помощью скрипта на языке JavaScript, который находится в файле captcha.js (листинг 2.5.1.2).

|  |
| --- |
| <div class="item\_1">  <div class="inputbox">  <input required="required" type="text" name="captcha" id="captcha">  <span>Каптча</span>  <i></i>  </div>  </div>  <div class="inputbox">  <button type="button" name="captcha\_btn" class="btn\_submit btn\_captcha"></button>  </div> |

Листинг 2.5.1.1 - Листинг кода для поля ввода каптчи и кнопки для нее же

|  |
| --- |
| var captcha\_field = document.getElementById('captcha')  var captcha\_btn = document.querySelector('.btn\_captcha')  function generateCaptchaAlphabet(length) {  var result = '';  var characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz';  var charactersLength = characters.length;    for (var i = 0; i < length; i++) {  result += characters.charAt(Math.floor(Math.random() \* charactersLength));  }  return result;  }  var secondCaptchaArray  function generateCaptchaNumeric() {  let array =[];  let first\_num = Math.floor(Math.random() \* (100) + 1);  let second\_num = Math.floor(Math.random() \* (100) + 1);  array.push(first\_num, second\_num, first\_num + second\_num)  return array;  }  function isEmpty(obj) {  return Object.getOwnPropertyNames(obj).length === 1;  }  captcha\_btn.innerText = generateCaptchaAlphabet(5)  captcha\_btn.addEventListener("click", () => {  if (isEmpty(captcha\_field.value)) {  alert("Поле для ввода пустое, потворите попытку")  return  }  if (isNaN(captcha\_btn.innerText.replace(' + ', ''))) {  if (captcha\_field.value === captcha\_btn.innerText) {  alert("Введено верно. Добро пожаловать!")  }  else {  secondCaptchaArray = generateCaptchaNumeric()  captcha\_btn.innerText = secondCaptchaArray[0] + " + " + secondCaptchaArray[1]  }  } else {  if (isNaN(captcha\_field.value)) {  alert("Это не число")  } else {  if (Number(captcha\_field.value) === secondCaptchaArray[2]) {  alert("Введено верно. Добро пожаловать!")  }  else {  secondCaptchaArray = generateCaptchaNumeric()  captcha\_btn.innerText = secondCaptchaArray[0] + " + " + secondCaptchaArray[1]  }  }  }  }) |

Листинг 2.5.1.2 - Листинг файла captcha.js

Изначально, при открытии страницы, каптча состоит из 5 символов английского алфавита нижнего и верхнего регистров, пользователь должен ввести в поле с надписью: «Каптча» данные 5 символов, после чего нажать на кнопку с этими символами. Если каптча введена верно, то появится уведомление, сообщающее об этом, если же пользователь ошибся, то появится вторая версия каптчи, которая будет выглядит как сумма двух чисел: число 1 + число 2. Пользователю необходимо будет ввести итог данной суммы, при правильном вводе будет выведено сообщение об этом. Если же пользователь ничего не ввел и нажал на кнопку каптчи, то появится сообщение о том, что поле ввода пустое. Для создания символьной каптчи и каптчи с суммой были реализованы специальные генераторы, а именно функция generateCaptchaAlphabet() и generateCaptchaNumeric().

## Создание карточек с товарами из массива на странице «Мерч»

Для реализации работы фильтров на странице «Мерч» использовались кнопки в HTML-файле (листинг 2.5.2.1), а так же код на языке JavaScript, который находится в файле array.js (листинг 2.5.2.2.).

Для создания карточек на странице «Мерч» использовались заранее созданные классы в CSS-файле, которые отвечали за их внешний вид, а так же массив с необходимыми данными в файле со скриптом на языке JS cards.js (листинг 2.5.2.3).

|  |
| --- |
| <div class="btn\_container">  <button class ="solid\_btn" id = "redOnly"><p>Красные</p></button>  <button class ="solid\_btn" id = "filter\_array"><p>Фильтр</p></button>  <button class ="solid\_btn" id = "sort\_array"><p>Сортировка</p></button>  <button class ="solid\_btn" id = "jerseyOnly"><p>Джерси</p></button>  <button class ="solid\_btn" id = "blueOnly"><p>Синие</p></button>  <button class ="solid\_btn" id = "showAll"><p>Показать все</p></button>  <button class ="solid\_btn" id = "hoodieOnly"><p>Худи</p></button>  </div> |

Листинг 2.5.2.1 - Листинг кода с блоком кнопок для фильтров

|  |
| --- |
| var btn\_filter = document.getElementById("filter\_array")  var btn\_sort = document.getElementById("sort\_array")  var btn\_red = document.getElementById("redOnly")  var btn\_blue = document.getElementById("blueOnly")  var btn\_jers = document.getElementById("jerseyOnly")  var btn\_hood = document.getElementById("hoodieOnly")  var btn\_all = document.getElementById("showAll")  var menu = document.querySelector(".productsMenu")  var sortCounter = 1  btn\_filter.addEventListener("click", () => {  let a = prompt("Выдать товары по цене от:")  let b = prompt("Выдать товары по цене до:")  let res\_array = filterArray(items, a, b)  array\_to\_Cards(res\_array)  })  btn\_sort.addEventListener("click", () =>{  if (sortCounter === 1){  clearCardsTable()  let res\_array = items.sort(compareItemsHigher)  array\_to\_Cards(res\_array)  sortCounter = 0  }  else{  clearCardsTable()  let res\_array = items.sort(compareItemsLower)  array\_to\_Cards(res\_array)  sortCounter = 1  }  })  btn\_red.addEventListener("click", () =>{  array\_to\_Cards(items)  let containers = document.querySelectorAll('.productBlock')  containers.forEach(item => {  if (item.classList.contains("blue")){  menu.removeChild(item)  }  })  })  btn\_blue.addEventListener("click", () =>{  array\_to\_Cards(items)  let containers = document.querySelectorAll('.productBlock')  containers.forEach(item => {  if (item.classList.contains("red")){  menu.removeChild(item)  }  })  })  btn\_all.addEventListener("click", ()=> {  array\_to\_Cards(items);  })  btn\_jers.addEventListener("click", () =>{  array\_to\_Cards(items)  let containers = document.querySelectorAll('.productBlock')  containers.forEach(item => {  if (item.classList.contains("hoodie")){  menu.removeChild(item)  }  })  })  btn\_hood.addEventListener("click", () =>{  array\_to\_Cards(items)  let containers = document.querySelectorAll('.productBlock')  containers.forEach(item => {  if (item.classList.contains("jersey")){  menu.removeChild(item)  }  })  })  function filterArray(list\_of\_cards, from, to){  return list\_of\_cards.filter(item => item.price > from && item.price < to )  }  function swap(item1, item2){  let itemtmp = item2  item2 = item1  item1 = itemtmp  }  function compareItemsHigher(itemFirst, itemSecond){  if (itemFirst.price > itemSecond.price) return 1  if (itemFirst.price === itemSecond.price) return 0  if (itemFirst.price < itemSecond.price) return -1  }  function compareItemsLower(itemFirst, itemSecond){  if (itemFirst.price > itemSecond.price) return -1  if (itemFirst.price === itemSecond.price) return 0  if (itemFirst.price < itemSecond.price) return 1  } |

Листинг 2.5.2.2 - Листинг файла array.js

|  |
| --- |
| var main\_div = document.querySelector('main')  class Item\_card {  material = "aboba"  constructor(material, title, price, background, color, type){  this.material = material;  this.title = title;  this.price = price;  this.background = background;  this.color = color;  this.type = type;  }  get material() {  return this.material;  }  }  var items = []  items.push(new Item\_card("хлопок 100%", 'Худи "Improve fan"', 1300, "photo/productsPhoto/hoodi1.png", "blue", "hoodie"))  items.push(new Item\_card("хлопок 100%", 'Худи "Player Hoodie" Синее', 1200, "photo/productsPhoto/hoodiePlayer.png", "blue", "hoodie"))  items.push(new Item\_card("хлопок 100%", 'Худи "Player Hoodie" Красное', 1100, "photo/productsPhoto/hoodiePlayerRed.png", "red", "hoodie"))  items.push(new Item\_card("хлопок 100%", 'Джерси "Player Jersey" Синее', 1400, "photo/productsPhoto/jerseyPlayer.png", "blue", "jersey"))  items.push(new Item\_card("хлопок 100%", 'Джерси "Player Jersey" Красное', 1500, "photo/productsPhoto/jerseyPlayerRed.png", "red", "jersey"))  array\_to\_Cards(items)  function array\_to\_Cards(items) {  clearCardsTable()  for(i = 0; i < items.length; i++){  itemToCard(items[i])  }  }  function itemToCard(item){  let tmp\_block = document.createElement("div")  tmp\_block.classList.add("productBlock")  tmp\_block.classList.add(item.color)  tmp\_block.classList.add(item.type)  let tmp\_info = document.createElement("div")  tmp\_info.classList.add("productInfo")  tmp\_block.appendChild(tmp\_info)  let tmp\_price = document.createElement("p")  tmp\_price.classList.add("price")  tmp\_price.innerHTML = item.price + 'р'  let tmp\_text = document.createElement("p")  tmp\_text.innerHTML ='Материал: ' + item.material  let tmp\_title = document.createElement("h1")  tmp\_title.innerHTML = item.title    tmp\_info.innerHTML = '<img src="'+item.background+'">'  tmp\_info.appendChild(tmp\_price)  tmp\_info.appendChild(tmp\_text)  tmp\_block.appendChild(tmp\_title)  menu.appendChild(tmp\_block)    }  function clearCardsTable(){  let containers = document.querySelectorAll('.productBlock')  containers.forEach(item => {  menu.removeChild(item)  })  } |

Листинг 2.5.2.3 - Листинг файла cards.js

Все карточки товаров загружаются из массива «items», который находится в файле cards.js. Загружаются они с помощью метода itemToCard(), который создает контейнер с помощью данных из массива объектов класса Item\_card. Кнопки фильтров реализованы с помощью файла array.js, где для каждой кнопки добавлен слушатель действий и они реагируют на клик, полностью очищая страницу от всех карт, а затем выводят только необходимые. Есть всего 7 кнопок фильтров, которые выводят только красные предметы, только синие предметы, только худи, только джерси, и также добавлено 2 кнопки для сортировки: сортировка (первый клик сортирует по возрастанию, второй – по убыванию, и так и зациклено) и фильтр (пользователь вводит цену от и цену до, по которой должны выводится товары).

# Оптимизация клиентской части интернет-ресурса

Для проведения адаптации веб-страниц было решено использовать медиазапросы, помогающие задать размеры экрана, под который пишется тот или иной стиль.

## Оптимизация клиентской части интернет-ресурса для основных браузеров

При тестировании веб-сайта в различных браузерах (Microsoft Edge,

Mozilla Firefox, Yandex Browser, Google Chrome, Safari) не было выявлено

проблем с отображением контента. Данное явление связано с тем, что в курсовой

работе были использованы технологии, которые поддерживаются 96%

браузеров.

## Оптимизация клиентской части интернет-ресурса для стационарных и мобильных устройств

Для правильного отображения на различных устройствах в вертикальном

и горизонтальном положениях были использованы медиа-запросы. Именно через данную технологию были перестроены многие элементы и свойства, как,

например, размер изображений и шрифта.

Результат представлен на (рис 3.2.1, рис 3.2.2, рис 3.2.3).



Рисунок 3.2.1 – Отображение ресурса на экране ноутбука расширением 1920х1080 пикселей.

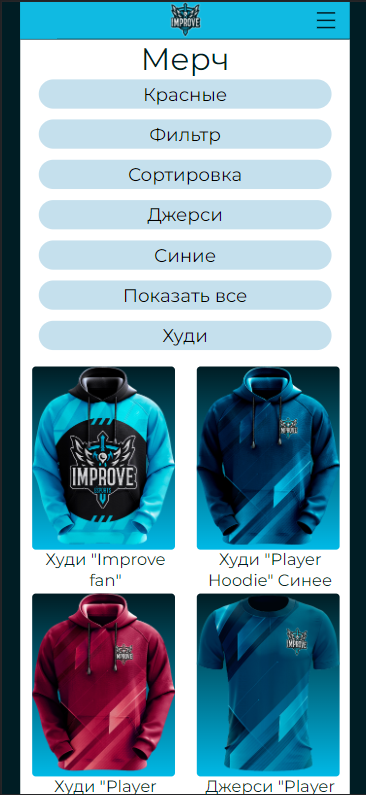


Рисунок 3.2.2 – Отображение ресурса на экране телефона IPhone 12 pro (вертикальная ориентация).

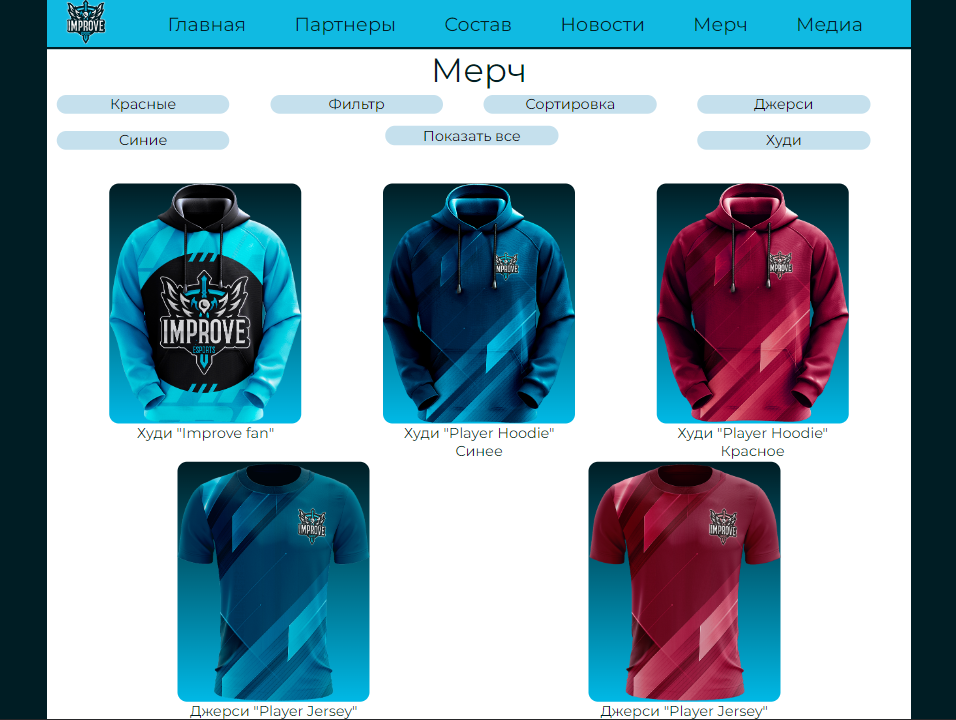


Рисунок 3.2.3 – Отображение ресурса на экране планшета IPad mini (горизонтальная ориентация).

# ЗАключение

В ходе выполнения курсовой работы было создано 6 HTML-страниц на тему «Киберспортивная организация», 6 CSS-файлов со стилями для каждой из страниц, а также 1 CSS-файл со стилем «футера», «хедера» и основным для всех страниц стилем «мейна» и 3 файла с кодом на языке JavaScript для более удобной работы с интернет-ресурсом и добавления новых возможностей в него. Все данный файлы были созданы с применением технологий HTML5, CSS3 и JavaScript. Так же в ходе создания проекта была изучена новая среда разработки Visual Studio Code. Были изучены и применены технологии, необходимые для адаптивной верстки.

Был проведен анализ предметной области разрабатываемого интернет-ресурса. Все созданные веб-страницы содержат подобранный в ходе разработки текстовый и визуальный контент. Также они имеют одинаковую стилизацию основной структуры, соответствующую современным стандартам веб-разработки. Благодаря проработанной идеи размещения контента на веб-страницах, весь предоставленный пользователю материал легкодоступен и понятен. Все страницы были адаптированы под различные размеры устройств и браузеры.

Все пункты, поставленные в задании на курсовую работу, были выполнены. Исходя из этого, выполнение курсовой работы можно считать успешным.

Исходный код интернет-ресурса доступен по ссылке - <https://github.com/at0mchik/Cursach>.

Доменный доступ к интернет-ресурсу для просмотра на любых устройствах и в любых браузерах доступен по ссылке - <https://at0mchik.github.io/>.

# Список источников

* Справочник по HTML [Электронный ресурс]. – URL: [**http://htmlbook.ru/html**](http://htmlbook.ru/html) (дата обращения 06.11.2022).
* Справочник по CSS [Электронный ресурс]. - URL: [**https://html5css.ru/cssref/default.php**](https://html5css.ru/cssref/default.php) (дата обращения 06.11.2022)
* NAVI: Официальный сайт клуба Natus Vinsere [Электронный ресурс]. - URL: <https://navi.gg/>(дата обращения 06.11.2022)
* Gambit Esports – В киберспорте с 2013 года [Электронный ресурс]. - URL: <https://gambit.gg/>(дата обращения 06.11.2022)
* SENTINELS [Электронный ресурс]. - URL: <https://sentinels.gg/>(дата обращения 06.11.2022)