**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ   
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и   
автоматизированных систем

**РГЗ**  
дисциплина: **Технологии Web-программирования**  
тема: «**Разработка Web-приложения»**

Выполнил: ст. группы ПВ-191  
Марченко Д.А.  
проверил: ст. пр. Картамышев С.В.

Белгород 2022

Оглавление

[1. «HTML. Разработка макетов и верстка шаблонов web-приложения с помощью языков HTML и CSS» 2](#_Toc120531473)

[2. Клиентское программирование. React.js 23](#_Toc120531474)

[3. Серверное программирование. Laravel 26](#_Toc120531475)

[4. Разработка и проектирование базы данных 30](#_Toc120531476)

[5. REST API 32](#_Toc120531477)

[6. Работа с Http запросами 35](#_Toc120531478)

# 1. «HTML. Разработка макетов и верстка шаблонов web-приложения с помощью языков HTML и CSS»

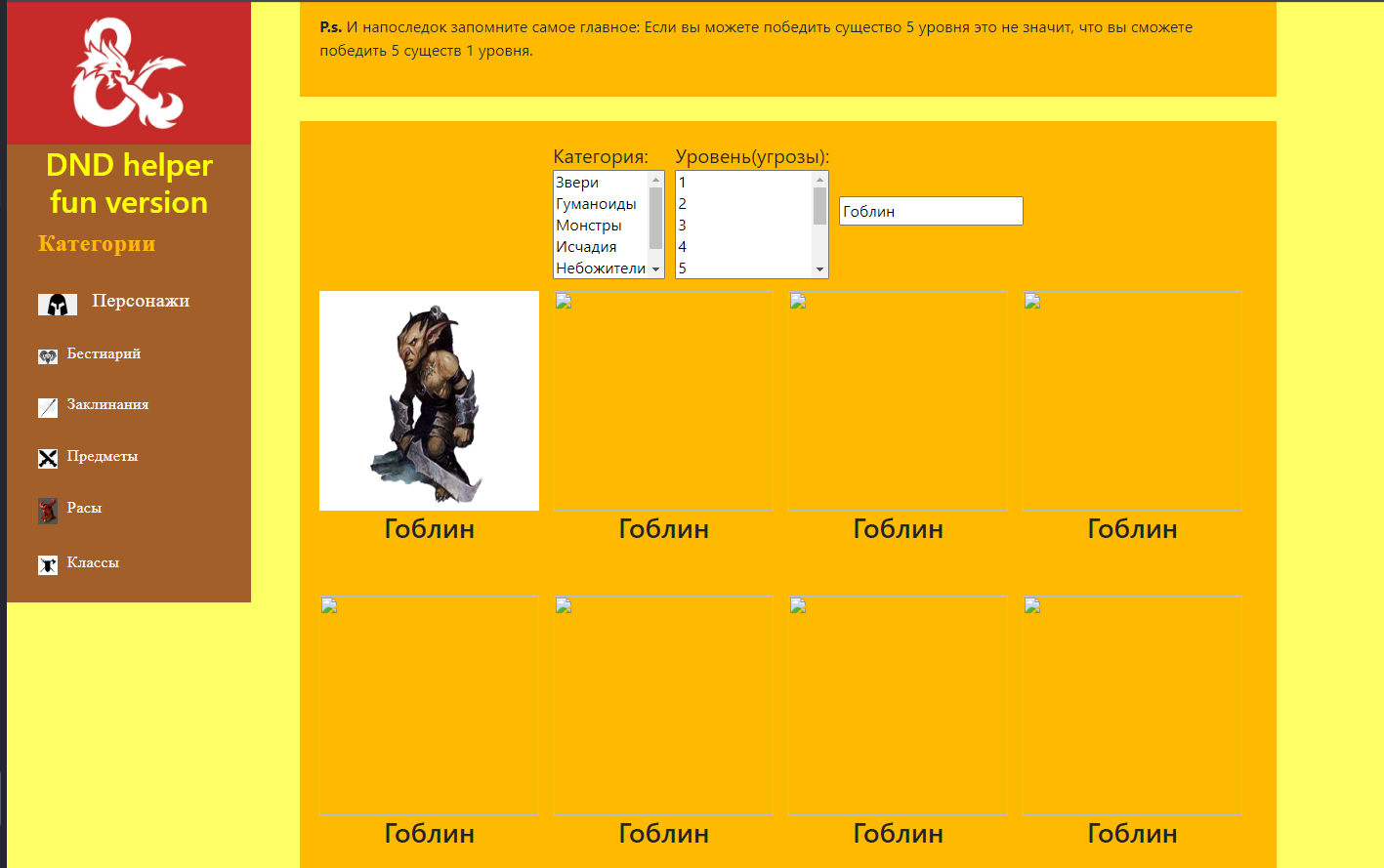
**Выполнение**

В ходе подготовки к работе мною был выбран предмет разработки (ДНД помошник (с шутками))

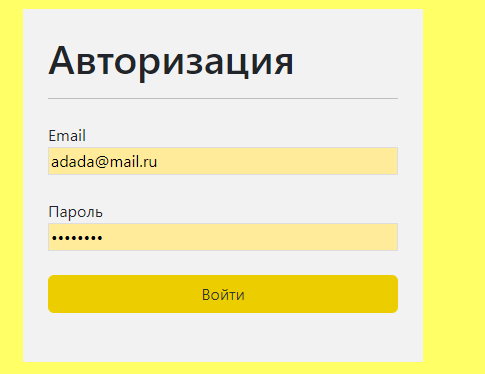
Персонаж



Бестиарий



Авторизация



Пример исходного кода

Файл DnDHeroes.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>DND helper</title>

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css">

    <!--<link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@600&family=Nunito: wght@300;400;600&display=swap" rel="stylesheet">-->

    <link rel="stylesheet" href="E:\Institute\Front\frontend\_course-1\style/style.css">

    <link rel="stylesheet" href="E:\Institute\Front\frontend\_course-1\style/authstyle.css">

    <script src="./js/heroes.js"></script>

    <script src = "./ts/firstTask.ts"></script>

</head>

<body>

    <div class="Body">

        <section class="Sidebar">

            <header>

                <section class="logoContainer">

                    <section class="logo">

                        <img src="img/logo.jpg" alt="logo">

                    </section>

                    <section class="logoText">

                        <h2>DND helper<br>fun version </h2>

                    </section>

                </section>

            </header>

            <form class="Category">

                <div>

                    <ul>

                        <lh>Категории</lh>

                            <li class="heroes"><img class="icon" style="width: 40px;" src="img\icon\hero.png"><a href="DnDHeroes.html" style="font-size: 20px;"> Персонажи </a></li>

                            <li class="monster"><img class="icon" src="img\icon\monster.png"><a href="Bestiary.html">Бестиарий</a></li>

                            <li class="magic"><img class="icon" src="img\icon\magic.jpg"><a href="#">Заклинания</a></li>

                            <li class="weapon"><img class="icon" src="img\icon\weapon.png"><a href="#">Предметы</a></li>

                            <li class="races"><img class="icon" src="img\icon\races.jpg"><a href="#">Расы</a></li>

                            <li class="classes"><img class="icon" src="img\icon\classes.jpg"><a href="#">Классы</a></li>

                    </ul>

                </div>

            </form>

        </section>

        <main class="Main">

            <div class="MainInfo">

                <h2>Персонажи</h2>

                <p>

                    Ага, и так мы добрались до самого весёлого. Хотите почуствовать себя Богом? Ну вообще перебьётесь, но вот персонажа создать можно. Стоит понимать,

                    что ваш персонаж - это не ваше отражение(<i>не ну если хотите, то пожалуйста, но это же скушно</i>). Реализуйте все свои отбитые фантазии

                    и проявите мастерство креатива. <br>

                    По поводу создания персонажа можно дать несколько советов:

                    <ol>

                        <li>Ваш персонаж-это новая личность. Так как играть с пелёнок долго и неинтересно то придётся это обдумывать самостоятельно. Потратьте немного времени

                            ,чтобы придумать историю из которой будет понятно мотивация персонажа и его будущий путь развития. Это не только упрощает работу, но и делает историю

                            куда интереснее и цельней. </li>

                        <li>Это фентези мир. К тому же ненастоящий. Так что не бойтесь экспериментов. Создайте максимально странного перса, от которого у ГМ-а глаза на потолок полезут.

                            Орк-пацифист: неплохо. гном-берсеркер: замечательно. огр-учёный: ваще огонь(<i>кста у него 2 головы значит в 2 раза умнее, так что это ещё и логично</i>)</li>

                    </ol>

                    <b>P.s.</b> И напоследок запомните самое главное: <u><b>НЕ создавайте законно-доброго человека воина</b></u>. Нет ничего шаблоннней подобного персонажа.

                    ГМ-ы подобного не любят, поэтому такие персонажи обычно съедаются в первой миссии каким нибудь вурдалаком, так что будьте осторожны

                </p>

            </div>

            <div class="MainList">

                <div class = "TopPage">

                    <div class = "Name">

                        <p style="font-size: 20px; margin: 0px;">Имя персонажа: </p>

                        <input type ="text" class = "NameField" value="Доктор Ливси">

                    </div>

                    <div class = "Race">

                        <p style="font-size: 20px; margin: 0px;">Раса: </p>

                        <div class = "NameField">

                            Человек

                        </div>

                    </div>

                    <div class="WorldView">

                        <p style="font-size: 20px; margin: 0px;">Мировозрение: </p>

                        <select id="selectvalue">

                            <option>Законно-добрый</option>

                            <option>Нейтрально-добрый</option>

                            <option>Хаотично-добрый</option>

                            <option>Законно-нейтральный</option>

                            <option>Нейтрально</option>

                            <option>Хаотично-нейтральный</option>

                            <option>Законно-злой</option>

                            <option>Нейтрально-злой</option>

                            <option>Хаотично-злой</option>

                        </select>

                    </div>

                    <div class = "Name">

                        <p style="font-size: 20px; margin: 0px;">Уровень персонажа: </p>

                        <input type ="text" class = "NameField" id ="level" value="5">

                    </div>

                </div>

                <div class="MainTable">

                    <div class = "ParametrTable">

                        <div class = "Bonus">

                            <div class = "Count">

                                3

                            </div>

                            <div class = "Name">

                                Бонус мастерства

                            </div>

                        </div>

                        <div class="Parametrs">

                            <div class = "Param" id="strength">

                                <p class ="Name" ><b>Сила</b></p>

                                <textarea>16</textarea>

                                <div class="Bonus">

                                    <p style="font-size:30px; text-align: center; margin: auto;">+3</p>

                                </div>

                            </div>

                            <div class = "Param" id="dexterity">

                                <p class ="Name" ><b>Ловкость</b></p>

                                <textarea>12</textarea>

                                <div class="Bonus">

                                    <p style="font-size:30px; text-align: center; margin: auto;">+1</p>

                                </div>

                            </div>

                            <div class = "Param" id="physique">

                                <p class ="Name" ><b>Телос<br>ложение</b></p>

                                <textarea>18</textarea>

                                <div class="Bonus" style="margin-top: -5px;">

                                    <p style="font-size:30px; text-align: center; margin: auto; ">+4</p>

                                </div>

                            </div>

                            <div class = "Param"  id="intelligence">

                                <p class ="Name"><b>Интеллект</b></p>

                                <textarea>14</textarea>

                                <div class="Bonus">

                                    <p style="font-size:30px; text-align: center; margin: auto; ">+2</p>

                                </div>

                            </div>

                            <div class = "Param" id="wisdom">

                                <p class ="Name" ><b>Мудрость</b></p>

                                <textarea>10</textarea>

                                <div class="Bonus">

                                    <p style="font-size:30px; text-align: center; margin: auto;">0</p>

                                </div>

                            </div>

                            <div class = "Param" id="charisma" style="margin-bottom: 0px;">

                                <p class ="Name" ><b>Харизма</b></p>

                                <textarea>20</textarea>

                                <div class="Bonus">

                                    <p style="font-size:30px; text-align: center; margin: auto;">+5</p>

                                </div>

                            </div>

                        </div>

                    </div>

                    <div class = "CenterLayout">

                        <p style="text-align:center; margin: 0;"><img src="https://regnum.ru/uploads/pictures/news/2015/09/01/1441099985\_остров-сокровищ-rejpg\_normal.jpg" ></p>

                        <div  style= "font-size: 16px;">

                            <textarea>Очень весёлый и интересный человек. Характер общительный. Гигачад</textarea>

                        </div>

                    </div>

                    <div class="RightLayout">

                        <div class="List">

                            <p>Классы</p>

                            <div>

                                <textarea placeholder="Введите описание класса"></textarea>

                            </div>

                        </div>

                        <div class="List">

                            <p>Предметы</p>

                            <div class = "Field">

                                <textarea placeholder="Введите описание предметов"></textarea>

                            </div>

                        </div>

                        <div class="List">

                            <p>Заклинания</p>

                            <div class = "Field">

                                <textarea placeholder="Введите описание заклинаний"></textarea>

                            </div>

                        </div>

                        <div class="m b-5">

                            <button type="submit" class="btn btn-green" onclick="checkParams()">Изменить</button>

                        </div>

                    </div>

                </div>

            </div>

        </main>

        <div class="Menu">

            <div>

                <p class="User">Хорошей игры Username</a>.</p>

                <img class="Usericon" src="https://w7.pngwing.com/pngs/944/770/png-transparent-computer-icons-login-google-account-internet-user.png">

                <button class="Menubutton"><img class="Usericon" src="img\icon\vector.png" onclick="window.location.href = 'DNDAuth.html';"></button>

            </div>

            <p>Помните вы сами решаете свою судьбу, но удача порой может...Впрочем на вас удача отдохнула</p>

        </div>

    </div>

</body>

Файл DnDAuth.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>DND helper</title>

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@600&family=Nunito: wght@300;400;600&display=swap" rel="stylesheet">

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css">

    <link rel="stylesheet" href="style/style.css">

    <link rel="stylesheet" href="style/authstyle.css">

    <script src="./js/func.js"></script>

</head>

<body>

    <div class="Body">

        <section class="Sidebar">

            <header>

                <section class="logoContainer">

                    <section class="logo">

                        <img src="img/logo.jpg" alt="logo">

                    </section>

                    <section class="logoText">

                        <h2>DND helper<br>fun version </h2>

                    </section>

                </section>

            </header>

            <form class="Category">

                <div>

                <ul>

                    <lh>Категории</lh>

                        <li class="heroes"><img class="icon" style="width: 40px;" src="img\icon\hero.png"><a href="DnDHeroes.html" style="font-size: 20px;"> Персонажи </a></li>

                        <li class="monster"><img class="icon" src="img\icon\monster.png"><a href="Bestiary.html">Бестиарий</a></li>

                        <li class="magic"><img class="icon" src="img\icon\magic.jpg"><a href="#">Заклинания</a></li>

                        <li class="weapon"><img class="icon" src="img\icon\weapon.png"><a href="#">Предметы</a></li>

                        <li class="races"><img class="icon" src="img\icon\races.jpg"><a href="#">Расы</a></li>

                        <li class="classes"><img class="icon" src="img\icon\classes.jpg"><a href="#">Классы</a></li>

                    </ul>

                </div>

            </form>

        </section>

        <div class="login-form">

            <h1>Авторизация</h1>

            <hr>

            <form class="mt-4" method="post" onsubmit="login(); event.preventDefault()" >

            <div class="mb-4">

              <div><label>Email</label></div>

              <div class = "input"><input type="email" class="w-100" name="email"></div>

              <div class="error"></div>

            </div>

            <div class="mb-4">

              <div><label>Пароль</label></div>

              <div class = "input"><input type="password" class="w-100" name="password" ></div>

              <div class="error"></div>

            </div>

            <div class="mb-5">

              <button type="submit" class="btn btn-yellow">Войти</button>

            </div>

          </form>

        </div>

    </div>

</body>

Файл Bestiary.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>DND helper</title>

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css">

    <link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@600&family=Nunito: wght@300;400;600&display=swap" rel="stylesheet">

    <link rel="stylesheet" href="E:\Institute\Front\frontend\_course-1\style/authstyle.css">

    <link rel="stylesheet" href="E:\Institute\Front\frontend\_course-1\style/product.css">

    <link rel="stylesheet" href="E:\Institute\Front\frontend\_course-1\style/style.css">

</head>

<body>

    <div class="Body">

        <section class="Sidebar">

            <header>

                <section class="logoContainer">

                    <section class="logo">

                        <img src="img/logo.jpg" alt="logo">

                    </section>

                    <section class="logoText">

                        <h2>DND helper<br>fun version </h2>

                    </section>

                </section>

            </header>

            <form class="Category">

                <div>

                <ul>

                    <lh>Категории</lh>

                        <li class="heroes"><img class="icon" style="width: 40px;" src="img\icon\hero.png"><a href="DnDHeroes.html" style="font-size: 20px;"> Персонажи </a></li>

                        <li class="monster"><img class="icon" src="img\icon\monster.png"><a href="#">Бестиарий</a></li>

                        <li class="magic"><img class="icon" src="img\icon\magic.jpg"><a href="#">Заклинания</a></li>

                        <li class="weapon"><img class="icon" src="img\icon\weapon.png"><a href="#">Предметы</a></li>

                        <li class="races"><img class="icon" src="img\icon\races.jpg"><a href="#">Расы</a></li>

                        <li class="classes"><img class="icon" src="img\icon\classes.jpg"><a href="#">Классы</a></li>

                </ul>

                </div>

            </form>

        </section>

        <main class="Main">

            <div class="MainInfo">

                <h2>Бестиарий</h2>

                <p>

                    Примечание: прежде чем продолжать, убедитесь что ваш персонаж умеет читать. Не портите себе отыгрыш игровыми условностями).

                    <b>Помните, чтобы отыграть дебила надо самому стать дебилом.</b><br>

                    Если вы начинающий искатель приключений, если вы отважный храбрец что ищет себе новые вызовы, если вы бомж, который не может себе купить всё что вам надо, то эта книга специально для вас.

                    Познайте окружающий мир, узнайте об интересных существах и о том, как их убить.

                    Бестиарий представляет собой очень богатый выбор. От обычных волчков и кабанчиков до буквально богов АДА. И пусть Бестиарий это книга для всех,

                    но стоит понимать, что ваши герои врядли найдут книгу в которой описаны слабые места местного Сатаны. Так что для лучшего отыгрыша следуйте этим рекомендациям.

                    <ol>

                        <li>Перед тем, как посмотреть какое то существо, подумайте о том, а может ли ваш персонаж знать что то о нём. Является ли он мега-професиональным охотником

                            книжным червём или просто достаточно наблюдательным. Возможно надо кинуть на характеристику, а то мало ли ответственный момент, а вы вдруг бац и всё забыли

                            Добавляет новых красок в ваше приключение (в основном коричневых конечно)

                        </li>

                        <li>

                            Постарайтесь не открывать Бестиарий во время боя. Согласитесь было бы страноо если бы во время боя с тиранозваром ваш герой достал и начал читать книгу о динозаврах.

                        </li>

                    </ol>

                    <b>P.s.</b> И напоследок запомните самое главное: Если вы можете победить существо 5 уровня это не значит, что вы сможете победить 5 существ 1 уровня.

                </p>

            </div>

            <div class="MainList">

                <div class = "TopPage">

                    <div class="category">

                        <p style="font-size: 20px; margin: 0px;">Категория: </p>

                        <select name="category" multiple="multiple" size="5">

                            <option value="1">Звери</option>

                            <option value="2">Гуманоиды</option>

                            <option value="3">Монстры</option>

                            <option value="4">Исчадия</option>

                            <option value="5">Небожители</option>

                            <option value="6">Феи</option>

                          </select>

                    </div>

                    <div class="category">

                        <p style="font-size: 20px; margin: 0px;">Уровень(угрозы): </p>

                        <select name="level" multiple="multiple" size="5" style="width: 100%;">

                            <option value="1">1</option>

                            <option value="2">2</option>

                            <option value="3">3</option>

                            <option value="4">4</option>

                            <option value="5">5</option>

                            <option value="6">6</option>

                            <option value="7">7</option>

                            <option value="8">8</option>

                            <option value="9">9</option>

                            <option value="10 ">10 </option>

                          </select>

                    </div>

                    <input type="text" class="search" value="Гоблин" >

                </div>

                <section class="products-list">

                    <div class="product">

                        <div>

                            <img src="https://i.pinimg.com/originals/ca/54/96/ca5496b0fb8d28148f1ebdbad180bfdb.png"/>

                        </div>

                        <h3>Гоблин обыкновенный</h3>

                    </div>

                    <div class="product">

                        <div>

                            <img src="https://i.pinimg.com/originals/03/20/da/0320da34b6876df378d6d26a9ed06277.jpg"/>

                        </div>

                        <h3>Гоблин обыкновенный</h3>

                    </div>

                    <div class="product">

                        <div>

                            <img src="https://i.pinimg.com/736x/5c/4e/ae/5c4eae4dbead0b4c873918dceaf1ac94.jpg"/>

                        </div>

                        <h3>Гоблин обыкновенный</h3>

                    </div>

                    <div class="product">

                        <div>

                            <img src="https://i.pinimg.com/736x/50/a8/25/50a8256f6532656f48795338f30e9295.jpg"/>

                        </div>

                        <h3>Гоблин обыкновенный</h3>

                    </div>

                    <div class="product">

                        <div>

                            <img src="https://i.pinimg.com/736x/1f/37/ee/1f37ee35c2a38cee7bc17dc9c5571b00--woodland-creatures-roma.jpg"/>

                        </div>

                        <h3>Гоблин обыкновенный</h3>

                    </div>

                    <div class="product">

                        <div>

                            <img src="https://i.pinimg.com/736x/10/72/78/107278e3f87b8b3ef8524480d3aa1d6a.jpg"/>

                        </div>

                        <h3>Гоблин обыкновенный</h3>

                    </div>

                    <div class="product">

                        <div>

                            <img src="https://i.pinimg.com/originals/84/99/68/849968b2244dd77e40c2966d03cf741d.jpg"/>

                        </div>

                        <h3>Гоблин обыкновенный</h3>

                    </div>

                    <div class="product">

                        <div>

                            <img src="https://i.pinimg.com/originals/73/91/2c/73912c2e4344bf41cd4fb92d7a4f1efb.jpg"/>

                        </div>

                        <h3>Гоблин обыкновенный</h3>

                    </div>

            </div>

        </main>

        <div class="Menu">

            <div>

                <p class="User">Хорошей игры Username</a>.</p>

                <img class="Usericon" src="https://w7.pngwing.com/pngs/944/770/png-transparent-computer-icons-login-google-account-internet-user.png">

                <button class="Menubutton"><img class="Usericon" src="img\icon\vector.png" onclick="window.location.href = 'DNDAuth.html';"></button>

            </div>

            <p>Помните вы сами решаете свою судьбу, но удача порой может...Впрочем на вас удача отдохнула</p>

        </div>

    </div>

</body>

Файл style.css

body{

    background-color: rgba(255,255,102);

    padding: 0;

    margin: 0;

}

.Body{

    display: flex;

    flex-direction: row;

}

.Sidebar {

    top:0;

    width: 250px;

    left: 0px;

    position:fixed;

    top:0;

    overflow: auto;

    display: flex;

    flex-direction: column;

    align-items: flex-start;

    background: rgba(162,95,42);

/\* Карточка \*/

/\*filter: drop-shadow(0px 10px 25px rgba(0, 0, 0, 0.25));\*/

}

.Sidebar .logoContainer img {

    max-width: 100%;

}

.logoText h2{

    color:rgba(255, 255, 0);

    text-align: center;

}

.Category {

    display: flex;

    flex-direction: column; align-items: flex-start; padding: 0px;

    position: static; width: 100%; height: 100%; left: auto;

    top: 0x;

}

.Category ul li {

    flex: none;

    order: 1;

    flex-grow: 0;

    margin: 30px 0px;

    list-style-type: none; font-family: Nunito; font-style: normal; font-weight: normal; font-size: 16px;

    line-height: 22px;

    letter-spacing: 0.025em;

/\* identical to box height \*/

}

.Category ul lh {

    font-family: Nunito; font-style: normal; font-weight: 600; font-size: 24px; line-height: 33px;

    letter-spacing: 0.03em;

    color: rgba(255,186,0);

}

.Category ul li a {

    color: #F9F9F9;

    text-decoration: none;

}

.Category li a:hover {

    text-decoration: underline;

}

.icon {

    width: 20px;

    padding-right: 10px;

    position: relative;

    top: 2px;

    background-image: none;

    line-height: 1;

    margin-right: 10px;

    padding: 0px;

}

.Main {

    width: 1000px;

    height: 1200px;

    margin-left: 300px;

    top: 10px;

    display: flex;

    flex-direction: column;

    align-items: flex-start;

}

.MainInfo {

    padding: 20px;

    width: 1000px;

    max-height: 45%;

    background: rgba(255,186,0);

}

.Main h2 {

    color:rgba(255, 255, 0);

    flex: none;

    order: 0;

    align-self: stretch; flex-grow: 0; margin: 0px 10px;

/\* Акцент \*/

/\* Inside Auto Layout \*/

}

.MainList {

    top: 20px;

    margin-top: 25px;

    padding: 20px;

    background: rgba(255,186,0);

    width: 100%;

    }

.TopPage {

    display: flex;

    flex-direction: row;

    justify-content: center;

    align-items: center;

    width: 100%;

}

.TopPage div{

    margin-right: 10px;

    margin-left: 0px;

}

.TopPage.Name{

     height: 60px;

     width: 20%;

     margin: 0px;

}

.NameField{

    height: 30px;

    width: 95%;

    background-color: white;

    align-items: flex-end;

    border: 3px black solid;

    font-size: 25px;

    text-align:center;

    margin: 0px;

    line-height: 1;

}

#selectvalue{

    font-size: 25px;

    border: 3px black solid;

}

.RightLayout textarea{

    height: 100%;

    width: 100%;

    resize: none;

}

.CenterLayout textarea{

    height: 100%;

    width: 100%;

    resize: none;

}

.MainTable{

    display: flex;

    flex-direction: row;

    margin-top: 10px;

    height: 900px;

}

.ParametrTable{

    display: flex;

    flex-direction: column;

    width: 20%;

    background: rgba(64,58,58);

}

.Bonus{

    display: flex;

    flex-direction: row;

    width: 100%;

    height: 50px;

}

.Bonus .Count{

    width: 20%;

    height: 100%;

    background-color: white;

    border: 4px black solid;

    font-size: 40px;

    text-align: center;

    line-height: 1;

}

.Bonus .Name{

    margin-top: 2px;

    width: 80%;

    height: 100%;

    background-color: white;

    border: 1px black solid;

    text-align: justify;

    font-size: 24px;

    line-height: 1;

}

.Parametrs{

    margin-top: 10px;

    display: flex;

    flex-direction: column;

    margin-left: auto;

    margin-right: auto;

    width: 50%;

}

.Parametrs .Param{

    height: 100px;

    width: 100%;

    background: white;

    text-align: center;

    margin-bottom: 40px;

}

.Parametrs .Param .Name{

    font-size:19px;

    margin: 0px;

    padding:0px;

    line-height: 1;

}

.Parametrs .Param textarea{

    width: 100%;

    height: 50%;

    resize: none;

    font-size:35px;

    padding: 0px;

    text-align: center;

    line-height: 1;

}

.Parametrs .Bonus {

    /\*background-color: black;\*/

    margin-top:0px ;

    height: 50%;

    width: 50%;

    display: flex;

    margin-left: auto;

    margin-right: auto;

    border: 5px solid black;

    box-sizing: border-box;

    border-radius: 200px;

    background: white;

    text-align: center;

}

.CenterLayout{

    margin-left: 3px;

    width: 50%;

}

.CenterLayout img{

    max-width: 100%;

    max-height: 50%;

    -o-object-fit: cover;

    object-fit: cover;

    object-position: center;

    border: 2px solid black;

}

.CenterLayout div{

    height: 540px;

    background: white;

    border: 2px solid black;

    margin: 0;

}

.RightLayout{

    display: flex;

    flex-direction: column;

    margin-left: 10px;

    height: 100%;

    width: 25%;

}

.RightLayout .List p{

    margin: 0px;

    font-size: 25px;

}

.RightLayout .List div{

    overflow: auto;

    width: 100%;

    height: 250px;

    background: white;

    border: 2px solid black;

}

.Menu{

    position: absolute;

    display: flex;

    flex-direction: column ;

    align-items: flex-end;

    width: 300px;

    left: 1400px;

}

.Menu div{

    display: flex;

    flex-direction: row ;

    align-items: center;

}

.Usericon{

    width: 42px;

    height: 42px;

    top: 0px;

    border-radius: 50%;

}

.Menubutton {

    position: relative;

    padding: 0px;

    width: 40px;

    height: 40px;

    /\* Smooth shadow \*/

    }

.Menubutton img{

    max-width: 100%;

    max-width: 100%;

}

.btn-green {

    width: 100%;

    background-color: #7cfc00;

    color: #333;

    transition: 2s;

  }

# 2. Клиентское программирование. React.js

В ходе выполнения работы были изучены основы разработки frontend-приложений на языке TypeScript, развернуто базовое приложение на фреймворке React.js, в него перенесены страницы из 1ой лабораторной работы, разделены на компоненты.

Пример кода

Главная страница где всё прогружается

import React from 'react';

import {Header} from './components/App/Header'

import { Footer } from './components/App/Footer';

import {Hero} from './Views/Hero';

import { Bestiary } from './Views/Bestiary';

import {Login} from './Views/Login';

import {Monster} from './Views/Bestiary/Monster';

import { Registration } from './Views/Registration';

import { useState } from 'react';

import './App.css';

import {Routes, Route, BrowserRouter} from 'react-router-dom'

const App = () => {

  const [userName, setUserName] = useState<string|null>("Пользователь");

  const rememberMe = localStorage.getItem('rememberMe') === 'true';

  const updateName = () => {

    if (rememberMe)

    {

      setUserName(localStorage.getItem('userName'));

    }

  }

  const setName = (name:string) => {

    setUserName(name);

  }

  React.useEffect(updateName);

  return (

      <BrowserRouter>

      <body>

        <div className="Body">

          <Header/>

          <Routes>

            <Route path="/hero" element = {<Hero/>}/>

            <Route path='/' element = {<Bestiary/>} />

            <Route path = '/login' element = {<Login setNickName={setName}/>}/>

            <Route path='/autorisation' element = {<Registration setNickName={setName}/>}/>

            <Route path='/monster/:id' element = {<Monster />}/>

          </Routes>

          <Footer nickName = {userName}/>

        </div>

      </body>

      </BrowserRouter>

  );

}

export default App;

Боковая панель

import React from 'react';

import {Link} from 'react-router-dom';

import './Header.css';

const Header = () => {

  return <section className="Sidebar">

        <header>

            <section className="logoContainer">

                <img src="/img\logo.jpg" alt="logo"/>

            </section>

            <section className="logoText">

                <h2>DND helper <br/> fun version </h2>

            </section>

        </header>

        <form className="Category">

            <div>

                <ul>

                    <li className="heroes"><img className ="icon" style={{width: "40px"}} src="/img/icon/hero.png"/> <Link to='/hero' style={{font: "20px"}}> Персонажи </Link> </li>

                    <li className="monster"><img className="icon" src="/img\icon\monster.png"/><Link to='/'>Бестиарий</Link></li>

                    <li className="magic"><img className="icon" src="/img\icon\magic.jpg"/><a href="#">Заклинания</a></li>

                    <li className="weapon"><img className="icon" src="/img\icon\weapon.png"/><a href="#">Предметы</a></li>

                    <li className="races"><img className="icon" src="/img\icon\races.jpg"/><a href="#">Расы</a></li>

                    <li className="classes"><img className="icon" src="/img\icon\classes.jpg"/><a href="#">Классы</a></li>

                </ul>

            </div>

        </form>

    </section>

}

export default Header;

**Мини отображение монстра**

import React from 'react';

import {Link} from 'react-router-dom'

import './Product.css';

type Props = {

    img: string,

    name: string

  }

const Product:React.FC<Props> = ({img, name}) =>  {

  return (

  <div className ="product">

  <div>

      <img src={img}/>

  </div>

  <h3>{ name }</h3>

</div>

)

}

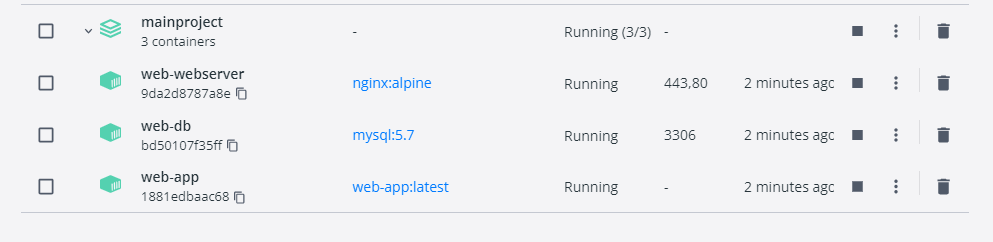
export default Product;

# 3. Серверное программирование. Laravel

Выполнение запроса в postman



Запущенные docker контейнеры



docker-compose.uml

version: '3'

services:

  app:

    build:

      context: .

      dockerfile: docker/Dockerfile

    image: web-app

    container\_name: web-app

    tty: true

    volumes:

      - ./:/var/www

      - ./php/local.ini:/usr/local/etc/php/conf.d/local.ini

    networks:

      - app-network

  webserver:

    image: nginx:alpine

    container\_name: web-webserver

    tty: true

    ports:

      - "80:80"

      - "443:443"

    volumes:

      - ./:/var/www

      - ./nginx/conf.d/:/etc/nginx/conf.d/

    networks:

      - app-network

  db:

    image: mysql:5.7

    container\_name: web-db

    tty: true

    ports:

      - "3306:3306"

    environment:

      MYSQL\_ROOT\_PASSWORD: 'superpass321!'

      MYSQL\_DATABASE: 'web'

      MYSQL\_USER: 'gendalin'

      MYSQL\_PASSWORD: 'vcnbntkb123'

    networks:

      - app-network

networks:

  app-network:

    driver: bridge

nginx.conf

server {

    listen 80;

    index index.html index.php;

    error\_log  /var/log/nginx/error.log;

    access\_log /var/log/nginx/access.log;

    root /var/www/public;

    client\_max\_body\_size 128M;

    location / {

        try\_files $uri $uri/ /index.html;

    }

    location /api {

        try\_files $uri $uri/ /index.php?$query\_string;

        gzip\_static on;

    }

    location ~ \.php$ {

        try\_files $uri =404;

        fastcgi\_split\_path\_info ^(.+\.php)(/.+)$;

        fastcgi\_pass app:9000;

        fastcgi\_index index.php;

        include fastcgi\_params;

        fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

        fastcgi\_param PATH\_INFO $fastcgi\_path\_info;

    }

}

Docker file

FROM php:7.4-fpm

# Copy composer.lock and composer.json

# COPY composer.json /var/www/

# Set working directory

WORKDIR /var/www

# Install dependencies

RUN apt-get update && apt-get install -y \

    build-essential \

    libpng-dev \

    libjpeg62-turbo-dev \

    libfreetype6-dev \

    locales \

    libzip-dev \

    zip \

    jpegoptim optipng pngquant gifsicle \

    vim \

    unzip \

    git \

    curl

# Clear cache

RUN apt-get clean && rm -rf /var/lib/apt/lists/\*

# Install extensions

RUN docker-php-ext-install pdo\_mysql zip exif pcntl

RUN docker-php-ext-configure gd --with-freetype=/usr/include/ --with-jpeg=/usr/include/

RUN docker-php-ext-install gd

# Install composer

RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --filename=composer

# Add user for laravel application

RUN groupadd -g 1000 www

RUN useradd -u 1000 -ms /bin/bash -g www www

# Copy existing application directory contents

COPY . /var/www

# Copy existing application directory permissions

COPY --chown=www:www . /var/www

# Change current user to www

USER www

# Expose port 9000 and start php-fpm server

EXPOSE 9000

CMD ["php-fpm"]

# 4. Разработка и проектирование базы данных

Таблица из базы данных



Содержимое файла с моделью Monster.php

<?php

namespace App\Models\Monster;

use Illuminate\Database\Eloquent\Model;

class Monster extends Model

{

    protected $table = 'monster';

    protected $fillable = [

        'id',

        'name',

        'heat',

        'armor',

        'streight',

        'dexterity',

        'physique',

        'intelligence',

        'wisdom',

        'charisma',

        'description',

        'image',

        'category\_id',

        'level\_id'

    ];

}

Содержимое файла с моделью User.php

<?php

namespace App\Models;

use Illuminate\Contracts\Auth\MustVerifyEmail;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Foundation\Auth\User as Authenticatable;

use Illuminate\Notifications\Notifiable;

use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable

{

    use HasApiTokens, HasFactory, Notifiable;

    /\*\*

     \* The attributes that are mass assignable.

     \*

     \* @var array<int, string>

     \*/

    protected $fillable = [

        'name',

        'email',

        'password',

    ];

    /\*\*

     \* The attributes that should be hidden for serialization.

     \*

     \* @var array<int, string>

     \*/

    protected $hidden = [

        'password',

        'remember\_token',

    ];

    /\*\*

     \* The attributes that should be cast.

     \*

     \* @var array<string, string>

     \*/

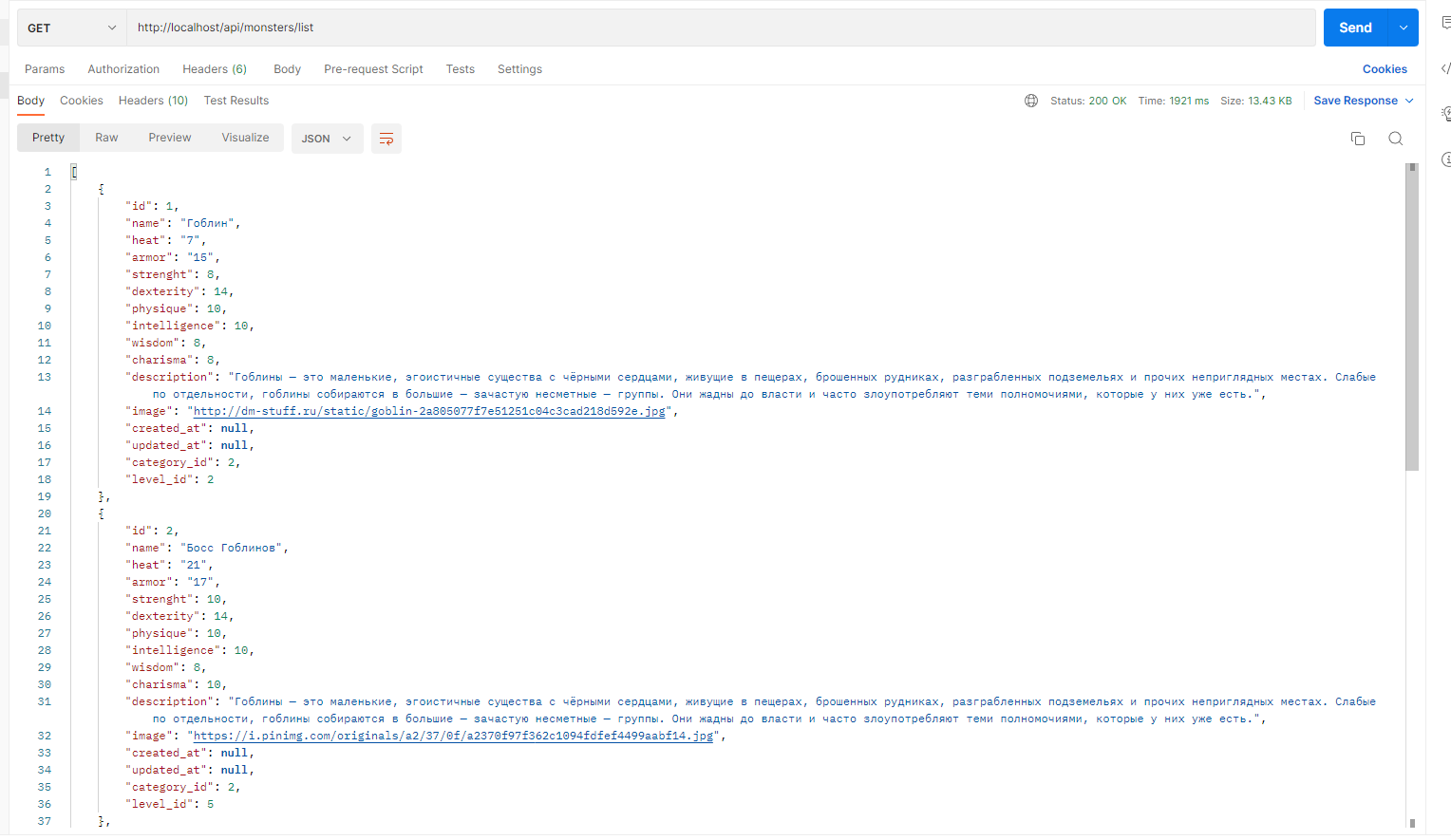
    protected $casts = [

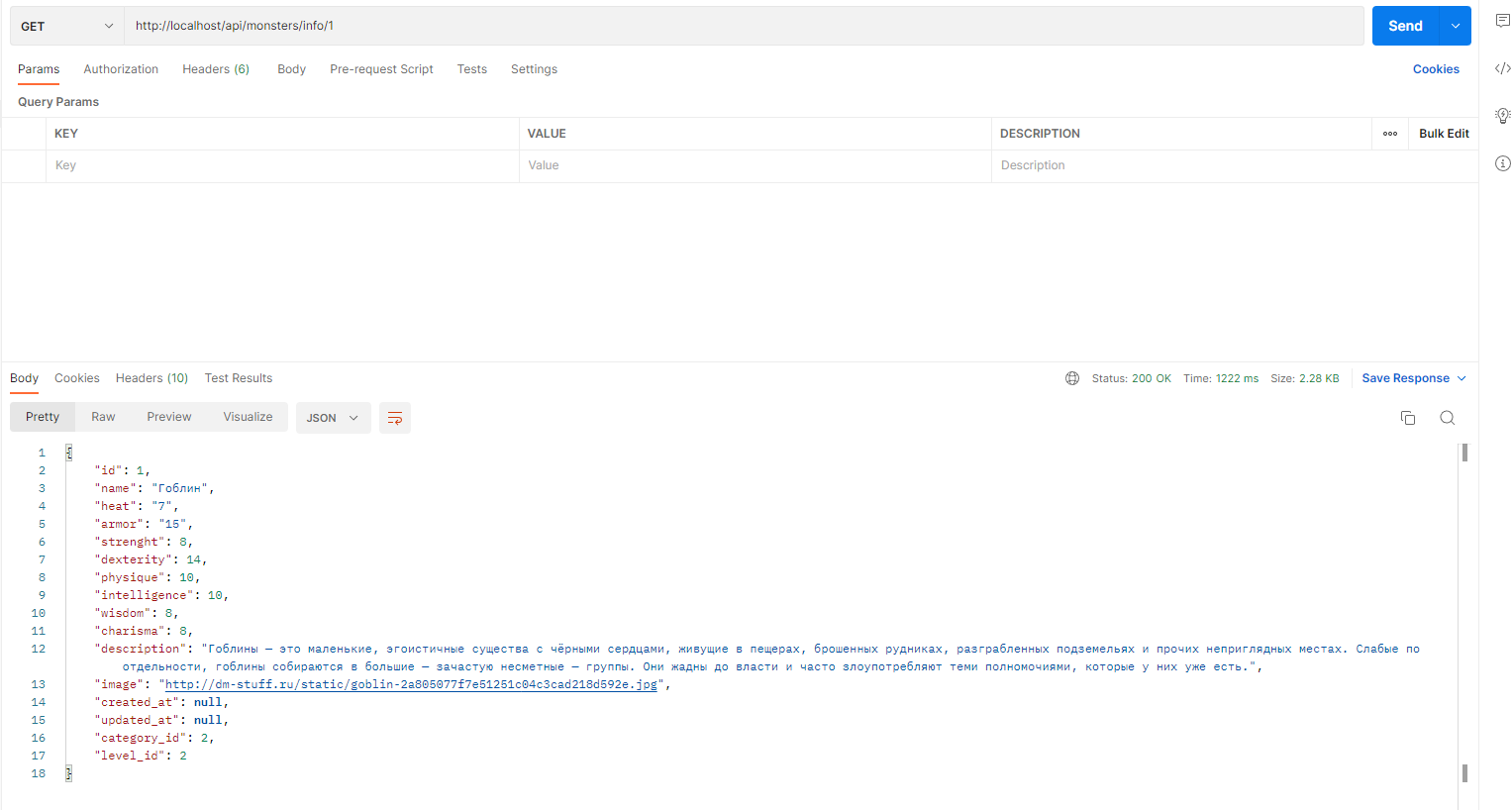
        'email\_verified\_at' => 'datetime',

    ];

}

Демонстрация запроса из Postman

****



# 5. REST API

MonsterController.php

<?php

namespace App\Http\Controllers\Monster;

use Illuminate\Http\Request;

use App\Http\Controllers\Controller;

use App\Models\Monster\Monster;

class MonsterController extends Controller

{

    public function list(){

        return Monster::query() -> get();

    }

    public function info($id){

        return Monster::query()->where('id','=',$id)->first();

    }

}

Код файла UserController.php

<?php

namespace App\Http\Controllers\User;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\User;

use Illuminate\Support\Facades\Hash;

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\DB;

use Symfony\Component\HttpKernel\Exception\NotFoundHttpException;

class UserController extends Controller

{

    public function login(Request $request)

    {

        $email = $request->get('email');

        $password = $request->get('password');

        // проверка авторизованного пользователя

        if (Auth::attempt(['email' => $email, 'password' => $password])) {

            $user = Auth::user();

            return $user;

        }

        throw new NotFoundHttpException('Неверный логин и пароль');

    }

    public function addNewUser(Request $request)

    {

        $name = $request->get('name');

        $email = $request->get('email');

        $password = $request->get('password');

        // проверка авторизованного пользователя

        if (Auth::attempt(['email' => $email])) {

            throw new NotFoundHttpException("aboba");

        }

        else{

            User::create([

                'name' => $name,

                'email' => $email,

                'password' => Hash::make($password),

            ]);

            if (Auth::attempt(['email' => $email, 'password' => $password])) {

                $user = Auth::user();

                return $user;

            }

            else{

                $user = Auth::user();

                return $user;

            }

        }

    }

}

Содержимое файла Api.php

<?php

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\Monster\MonsterController;

use App\Http\Controllers\Category\CategoryController;

use App\Http\Controllers\Level\LevelController;

use App\Http\Controllers\User\UserController;

use App\Http\Controllers\Hero\HeroController;

Route::prefix('monsters')->group(function(){

    Route::get('list', [MonsterController::class, 'list']);

    Route::get('info/{id}', [MonsterController::class, 'info']);

});

Route::prefix('category')->group(function(){

    Route::get('list', [CategoryController::class, 'list']);

    Route::get('info/{id}', [CategoryController::class,'info']);

});

Route::prefix('level')->group(function(){

    Route::get('list', [LevelController::class, 'list']);

    Route::get('info/{id}', [LevelController::class,'info']);

});

Route::prefix('user')->group(function () {

    Route::post('login', [UserController::class, 'login']);

    Route::post('registration', [UserController::class, 'addNewUser']);

});

Route::prefix('hero')->group(function () {

    Route::get('get/{user\_id}', [HeroController::class, 'get']);

    Route::post('update', [HeroController::class, 'updateUser']);

});

# 6. Работа с Http запросами

В качестве библиотеки для работы с ajax запросами был выбран axios.

Файл Hero.tsx

import {useNavigate} from 'react-router-dom'

import axios from 'axios';

import './Hero.css';

import { useState } from 'react';

import React from 'react';

import 'bootstrap/dist/css/bootstrap.min.css';

type HeroParam ={

    name: string,

    level: number,

    strenght: number,

    dexterity: number,

    physique: number,

    intelligence: number,

    wisdom: number,

    charisma: number,

    description: string,

    image: string

}

const Hero = () => {

    const navigate = useNavigate();

    const checkUser = () => {

        if (localStorage.getItem('rememberMe') !== 'true' || localStorage.getItem('userId') === null){

            alert("Просматривать героя может только пользователь");

            navigate('/login');

        }

    }

    React.useEffect(checkUser,[]);

    class Params {

        strength: number;

        dexterity: number;

        physique: number;

        intelligence: number;

        wisdom: number;

        charisma: number;

        constructor(strength: number, dexterity: number, physique: number, intelligence: number, wisdom: number, charisma: number){

            this.strength = strength;

            this.dexterity =dexterity;

            this.physique = physique;

            this.intelligence = intelligence;

            this.wisdom =wisdom;

            this.charisma = charisma;

        }

       }

       let [name, setName] = useState("");

       let [level, setLevel] = useState(1);

       let [levelBonus, setLevelBonus] = useState(getLevelBonus(level));

       let [params, setParams] = useState(new Params(10,10,10,10,10,10));

       let [paramBonus, setParamBonus] = useState (['0', '0', '0', '0', '0']);

       let [description, setDescription] = useState("");

       let [image, setImage] = useState("https://www.clipartmax.com/png/middle/474-4749661\_question-mark-clipart-lime-green-question-mark-clipart-lime-green.png");

    let HeroSave:HeroParam = {

        name: name,

        level: level,

        strenght: params.strength,

        dexterity: params.dexterity,

        physique: params.physique,

        intelligence: params.intelligence,

        wisdom: params.wisdom,

        charisma: params.charisma,

        description: description,

        image: image

    };

    const getHero = () => {

        axios.get<HeroParam>("http://localhost/api/hero/get/" + localStorage.getItem("userId")).then(({data}) => {

            if (data !== null){

                setName(data.name);

                setLevel(data.level);

                setLevelBonus(getLevelBonus(data.level));

                setParams(new Params(data.strenght,data.dexterity,data.physique,data.intelligence,data.wisdom,data.charisma));

                setParamBonus([getParamBonus(data.strenght), getParamBonus(data.dexterity), getParamBonus(data.physique),

                                 getParamBonus(data.intelligence), getParamBonus(data.wisdom), getParamBonus(data.charisma)]);

                setDescription(data.description);

                setImage(data.image);

                HeroSave ={

                    name: name,

                    level: level,

                    strenght: params.strength,

                    dexterity: params.dexterity,

                    physique: params.physique,

                    intelligence: params.intelligence,

                    wisdom: params.wisdom,

                    charisma: params.charisma,

                    description: description,

                    image: image

                };

            }

        }

    )}

    React.useEffect(getHero,[]);

    function getLevelBonus(level: number):string{

        return (Math.floor((level-1)/4) + 2).toString();

    }

    function getParamBonus(num:number):string{

        let bonusString = "";

        let buff = Math.floor(num/2) - 5;

        if (buff>0){

            bonusString += "+";

        }

        bonusString += buff.toString();

        return bonusString;

    }

    function checkParams():void{

        if (isValid()){

            HeroSave ={

                name: name,

                level: level,

                strenght: params.strength,

                dexterity: params.dexterity,

                physique: params.physique,

                intelligence: params.intelligence,

                wisdom: params.wisdom,

                charisma: params.charisma,

                description: description,

                image: image

            };

            axios.post("http://localhost/api/hero/update/", {Hero: HeroSave, userID: localStorage.getItem("userId")});

            setLevelBonus(getLevelBonus(level));

            setParamBonus([getParamBonus(params['strength']), getParamBonus(params['dexterity']), getParamBonus(params['physique']),

                             getParamBonus(params['intelligence']), getParamBonus(params['wisdom']), getParamBonus(params['charisma'])]);

        }

        else{

            setName(HeroSave.name);

            setLevel(HeroSave.level);

            setParams(new Params(HeroSave.strenght,HeroSave.dexterity,HeroSave.physique,HeroSave.intelligence,HeroSave.wisdom,HeroSave.charisma));

            setDescription(HeroSave.description);

        }

    }

    function isValid():boolean{

        let isValid:boolean = true;

        if(name === ''){

            alert("Окей, если ты не хочешь называть своего персонажа, то я назову его косипошой");

            setName('Косипоша');

        }

        if (!checkLevel(level))

            isValid = false;

        for(let character in params){

            let num:number = params[character as keyof Params];

            if(num < 1 || num > 20){

                isValid = false;

                alert("Характеристика установлена неверно. Допускаемые параметры: [1...20]");

                break;

            }

        }

        if (description === ""){

            isValid = false;

            alert("Можно хоть немного описание, позязя?");

        }

        return isValid;

    }

    function checkLevel(level:number){

        if (level >= 1 && level <= 20){

            return true;

        }

        else{

            if(isNaN(level))

                alert("И как ты представляешь себе такой уровень? Пиши цифрами");

            if(level < 1)

                alert("Я понимаю, что развитие назад это тоже развитие, но уровень не может быть меньше 1");

            if(level >20)

                alert("Быть Богом это круто, но к сожалению вы обычный сверхчеловек так что ваш максимальный уровень 20");

            return false;

        }

    }

    return (

        <main className="Main">

<div className="MainInfo">

    <h2>Персонажи</h2>

    <p>

        Ага, и так мы добрались до самого весёлого. Хотите почуствовать себя Богом? Ну вообще перебьётесь, но вот персонажа создать можно. Стоит понимать,

        что ваш персонаж - это не ваше отражение(<i>не ну если хотите, то пожалуйста, но это же скушно</i>). Реализуйте все свои отбитые фантазии

        и проявите мастерство креатива. <br/>

        По поводу создания персонажа можно дать несколько советов:

        <ol>

            <li>Ваш персонаж-это новая личность. Так как играть с пелёнок долго и неинтересно то придётся это обдумывать самостоятельно. Потратьте немного времени

                ,чтобы придумать историю из которой будет понятно мотивация персонажа и его будущий путь развития. Это не только упрощает работу, но и делает историю

                куда интереснее и цельней. </li>

            <li>Это фентези мир. К тому же ненастоящий. Так что не бойтесь экспериментов. Создайте максимально странного перса, от которого у ГМ-а глаза на потолок полезут.

                Орк-пацифист: неплохо. гном-берсеркер: замечательно. огр-учёный: ваще огонь(<i>кста у него 2 головы значит в 2 раза умнее, так что это ещё и логично</i>)</li>

        </ol>

        <b>P.s.</b> И напоследок запомните самое главное: <u><b>НЕ создавайте законно-доброго человека воина</b></u>. Нет ничего шаблоннней подобного персонажа.

        ГМ-ы такого не любят, поэтому такие персонажи обычно съедаются в первой миссии каким нибудь вурдалаком, так что будьте осторожны

    </p>

</div>

<div className="MainList">

    <div className = "TopPage">

        <div className = "Name">

            <p style={{fontSize: "20px", margin: "0px"}}>Имя персонажа: </p>

            <input type ="text" className = "NameField" value = {name}

            onChange={event => setName(event.target.value)}

            />

        </div>

        <div className = "Race">

            <p style={{font: "20px", margin: "0px"}}>Раса: </p>

            <div className = "NameField">

                Человек

            </div>

        </div>

        <div className = "Name">

            <p style={{font: "20px", margin: "0px"}}>Уровень персонажа: </p>

            <input type ="text" className = "NameField" id ="level"  value={level}

                onChange={event => setLevel(Number(event.target.value))} />

        </div>

    </div>

    <div className="MainTable">

        <div className = "ParametrTable">

            <div className = "BonusHero">

                <div className = "Count">

                    {levelBonus.toString()}

                </div>

                <div className = "Name">

                    Бонус мастерства

                </div>

            </div>

            <div className="Parametrs">

                <div className = "Param" id="strength">

                    <p className ="NameP" ><b>Сила</b></p>

                    <textarea value={params['strength']}

                        onChange={event => setParams(new Params( Number(event.target.value), params['dexterity'], params['physique'],

                                    params['intelligence'], params['wisdom'], params['charisma']))}

                    ></textarea>

                    <div className="Bonus">

                        <p>{ paramBonus[0] }</p>

                    </div>

                </div>

                <div className = "Param" id="dexterity">

                    <p className ="NameP" ><b>Ловкость</b></p>

                    <textarea value={params['dexterity']}

                        onChange={event => setParams(new Params( params['strength'] , Number(event.target.value),  params['physique'],

                                    params['intelligence'], params['wisdom'], params['charisma']))}

                    ></textarea>

                    <div className="Bonus">

                        <p>{ paramBonus[1]}</p>

                    </div>

                </div>

                <div className = "Param" id="physique">

                    <p className ="NameP"  style={{height:"40px"}}><b>Телос<br/>ложение</b></p>

                    <textarea value={params['physique']}

                        onChange={event => setParams(new Params( params['strength'], params['dexterity'], Number(event.target.value),

                                    params['intelligence'], params['wisdom'], params['charisma']))}

                    ></textarea>

                    <div className="Bonus" style={{marginTop: "-5px"}}>

                        <p>{ paramBonus[2]}</p>

                    </div>

                </div>

                <div className = "Param"  id="intelligence">

                    <p className ="NameP"><b>Интеллект</b></p>

                    <textarea value={params['intelligence']}

                        onChange={event => setParams(new Params( params['strength'], params['dexterity'], params['physique'],

                            Number(event.target.value), params['wisdom'], params['charisma']))}

                    ></textarea>

                    <div className="Bonus">

                        <p>{ paramBonus[3]}</p>

                    </div>

                </div>

                <div className = "Param" id="wisdom">

                    <p className ="NameP" ><b>Мудрость</b></p>

                    <textarea value={params['wisdom']}

                        onChange={event => setParams(new Params( params['strength'], params['dexterity'], params['physique'],

                                    params['intelligence'], Number(event.target.value), params['charisma']))}

                    ></textarea>

                    <div className="Bonus">

                        <p>{ paramBonus[4]}</p>

                    </div>

                </div>

                <div className = "Param" id="charisma" style = {{marginBottom: "0px"}}>

                    <p className ="NameP" ><b>Харизма</b></p>

                    <textarea value={params['charisma']}

                        onChange={event => setParams(new Params( params['strength'], params['dexterity'], params['physique'],

                                    params['intelligence'], params['wisdom'], Number(event.target.value)))}

                    ></textarea>

                    <div className="Bonus">

                        <p>{ paramBonus[5]}</p>

                    </div>

                </div>

            </div>

        </div>

        <div className = "CenterLayout">

            <img src= {image} />

            <div  style= {{ font: "16px" , width: '100%', maxHeight: '6%'}}>

                <textarea style = {{ whiteSpace: 'nowrap'}} value={image} placeholder="Введите новую ссылку на изображение. Если хотите чтобы оно не менялось, то оставьте пустым" onChange = {

                    event => setImage(event.target.value)

                }

                ></textarea>

            </div>

            <div className = "heroDescripton">

                <textarea value={description} onChange = {

                    event => setDescription(event.target.value) }></textarea>

            </div>

        </div>

        <div className="RightLayout">

            <div className="List">

                <p>Классы</p>

                <div>

                    <textarea placeholder="Введите описание класса"></textarea>

                </div>

            </div>

            <div className="List">

                <p>Предметы</p>

                <div className = "Field">

                    <textarea placeholder="Введите описание предметов"></textarea>

                </div>

            </div>

            <div className="List">

                <p>Заклинания</p>

                <div className = "Field">

                    <textarea placeholder="Введите описание заклинаний"></textarea>

                </div>

            </div>

            <div className="mb-5">

                <button type="submit" className="btn-green" onClick={checkParams}>Изменить</button>

            </div>

        </div>

    </div>

</div>

</main>)

}

export default Hero;

Файл Bestiary.tsx

import './Bestiary.css';

import axios from 'axios';

import React from 'react';

import {Link, useNavigate, useSearchParams, useLocation} from 'react-router-dom';

import Select from 'react-select';

import { useState } from 'react';

import {ProductView} from './Product';

type Category = {

    id: number,

    name: string

  }

type SelectItem = {

    value: string,

    label: string

}

type MonsterItem = {

    id:number,

    name: string,

    image: string,

    category\_id: number,

    level\_id: number

}

type Level = {

    id:number,

    value:number

}

const Bestiary = () => {

    const [monsterItems, setMonsterItems] = useState<MonsterItem[]>([]);

    const [optionsCategory, setCategory] = useState<SelectItem[]>([])

    const [optionsLevel, setLevels] = useState<SelectItem[]>([]);

    const getCategory = () => {

        axios.get<Category[]>('http://localhost/api/category/list').then(({data})=> {

            const category:SelectItem[] = []

            for (let item of data){

                category.push({value: item.id.toString(), label: item.name});

            }

            setCategory(category);

          })

      }

      const getLevel = () => {

        axios.get<Level[]>('http://localhost/api/level/list').then(({data})=> {

            const levels:SelectItem[] = []

            for (let item of data){

                levels.push({value: item.id.toString(), label: item.value.toString()});

            }

            setLevels(levels);

          })

      }

    const getMonsters = () => {

        axios.get<MonsterItem []>('http://localhost/api/monsters/list').then(({data})=> {

            setMonsterItems(data);

        })

    }

    React.useEffect(getCategory,[]);

    React.useEffect(getMonsters,[]);

    React.useEffect(getLevel,[]);

    const navigate = useNavigate();

    const location = useLocation();

    let [searchParams, setSearchParams] = useSearchParams();

    const [searchName, setSearcName] = useState(searchParams.get('name')|| '');

    const [selectedCategory,setSelectedCategory] = useState(searchParams.getAll('categorys'||[]));

    const [selectedLevel,setSelectedLevel] = useState(searchParams.getAll('level'||[]));

    const getSearch = (e:any) => {

        const searchParam = {

            name: searchName,

            categorys: selectedCategory,

            levels: selectedLevel

        }

        setSearchParams(searchParam);

    }

    return (

        <main className="Main">

            <div className="MainInfo">

                <h2>Бестиарий</h2>

                <p>

                    Примечание: прежде чем продолжать, убедитесь что ваш персонаж умеет читать. Не портите себе отыгрыш игровыми условностями).

                    <b>Помните, чтобы отыграть дебила надо самому стать дебилом.</b><br/>

                    Если вы начинающий искатель приключений, если вы отважный храбрец что ищет себе новые вызовы, если вы бомж, который не может себе купить всё что вам надо, то эта книга специально для вас.

                    Познайте окружающий мир, узнайте об интересных существах и о том, как их убить.

                    Бестиарий представляет собой очень богатый выбор. От обычных волчков и кабанчиков до буквально богов АДА. И пусть Бестиарий это книга для всех,

                    но стоит понимать, что ваши герои врядли найдут книгу в которой описаны слабые места местного Сатаны. Так что для лучшего отыгрыша следуйте этим рекомендациям.

                    <ol>

                        <li>Перед тем, как посмотреть какое то существо, подумайте о том, а может ли ваш персонаж знать что то о нём. Является ли он мега-професиональным охотником

                            книжным червём или просто достаточно наблюдательным. Возможно надо кинуть на характеристику, а то мало ли ответственный момент, а вы вдруг бац и всё забыли

                            Добавляет новых красок в ваше приключение (в основном коричневых конечно)

                        </li>

                        <li>

                            Постарайтесь не открывать Бестиарий во время боя. Согласитесь было бы страноо если бы во время боя с тиранозваром ваш герой достал и начал читать книгу о динозаврах.

                        </li>

                    </ol>

                    <b>P.s.</b> И напоследок запомните самое главное: Если вы можете победить существо 5 уровня это не значит, что вы сможете победить 5 существ 1 уровня.

                </p>

            </div>

            <div className="MainList">

                <div className = "TopPage">

                    <div className="category" style = {{width: '300px'}}>

                        <p style={{fontSize: "20px", margin: "0px"}}>Категория: </p>

                        <Select

                            options={optionsCategory}

                            value = {optionsCategory.filter(item=> selectedCategory.includes(item.value)===true)}

                            isMulti

                            placeholder = "Выбрать категорию"

                            onChange={

                                event =>{

                                    const valueArr= event.map(item => item.value);

                                    setSelectedCategory(valueArr);

                                    setSearchParams({name: searchName, categorys: valueArr, levels: selectedLevel});

                                }

                            }

                        />

                    </div>

                    <div className="category" style = {{width: '300px'}}>

                        <p style={{fontSize: "20px", margin: "0px"}}>Уровень(угрозы): </p>

                        <Select

                            options={optionsLevel}

                            value = {optionsLevel.filter(item=> selectedLevel.includes(item.value)===true)}

                            isMulti

                            placeholder = "Выбрать уровень"

                            onChange={

                                event =>{

                                    const valueArr= event.map(item => item.value);

                                    setSelectedLevel(valueArr);

                                    setSearchParams({name: searchName, categorys: selectedCategory, levels: valueArr});

                                }

                            }

                        />

                    </div>

                    <input type="search" className="search" value= {searchName} onChange={

                        event => {

                            event.preventDefault();

                            setSearcName(event.target.value);

                            const name = event.target.value;

                            if (name!==""){

                                setSearchParams({name: event.target.value, categorys: selectedCategory, levels: selectedLevel});

                            }

                            else{

                                setSearchParams({categorys: selectedCategory, levels: selectedLevel});

                            }

                        }

                    } />

                </div>

                <section className="products-list">

                    {monsterItems.filter(

                        item => {

                            const name = searchParams.get('name')|| '';

                            const categorys = searchParams.getAll('categorys')||[];

                            const levels = searchParams.getAll('levels')||[];

                            if (!item.name.includes(name!!)){

                                return false;

                            }

                            if (categorys.length !== 0 && !(categorys.includes(item.category\_id.toString()))){

                                return false;

                            }

                            if (levels.length !== 0 && !(levels.includes(item.level\_id.toString()))){

                                return false;

                            }

                            return true;

                        }

                    ).map(item =>

                        <Link to={'/monster/' + item.id.toString()}><ProductView img = {item.image} name = {item.name}/> </Link>

                    )}

                </section>

            </div>

        </main>

    )

}

export default Bestiary;