**Министерство науки и высшего образования РФ**

**Государственное образовательное учреждение высшего профессионального образования**

**Белгородский Государственный Технологический Университет им В. Г. Шухова**

Кафедра программного обеспечения вычислительной техники и

автоматизированных систем.

Расчётно-графическое задание по дисциплине

«Технологии Web-программирования»

Выполнил:

студент группы ПВ-41

Донцов А.А.

Проверил: Картамышев С. В.

Белгород 2020 г.

**Оглавление**

|  |  |
| --- | --- |
| Цель работы……………………………………………………………………… | 3 |
| Выполнение……………………………………………………………………… | 3 |
| Глава 1. HTML страницы………………………………………………………. | 3 |
| Глава 2. Приложение React……………………………………………………. | 10 |
| Глава 3. Серверная часть. Приложение Spring……………………………… | 16 |
| Глава 4. База данных. Миграции и модели.…………………………………… | 21 |
| Глава 5. REST API………………………………………………………………. | 28 |
| Глава 6. Взаимодействие приложения с API………………………………… | 32 |
| Вывод……………………………………………………………………………… | 37 |

**Цель работы**

Целью данного проекта является изучение принципов работы протокола передачи гипертекста HTTP и языка CSS, ознакомление с основами разработки frontend-приложений, в частности, с использованием клиентских фреймворков (React), ознакомление с основами работы docker и фреймворком Spring и изучить основы взаимодействия web-приложения с базой данных и принципы работы отправки Ajax запросов.

**Выполнение**

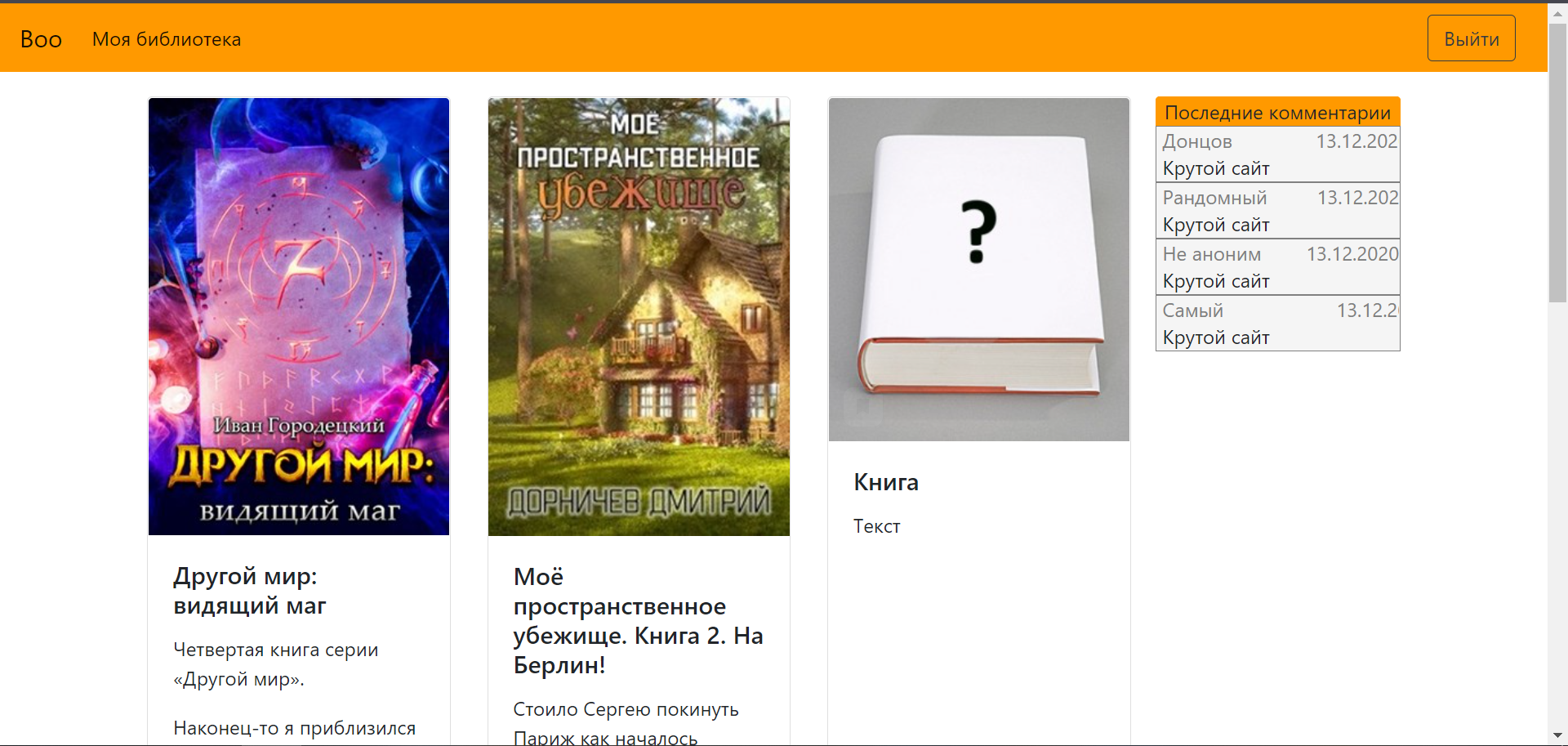
В качестве темы проекта был выбран сайт для чтения книг. По сути, данный сайт представляет собой набор книг, имеющих название, описание, содержание.

Также имеются пользователи, которые могут заходить на сайт (используя сервисы вк) для доступа к сайту.

**Глава 1. HTML страницы**

Сначала были спроектированы сами HTML страницы. В качестве основы и макета был выбран макет Album из примеров на сайте Bootstrap, карточки которого будут представлять собой вышеописанные изображения. Немного изменив данный шаблон, получаем следующий результат:





**Код страниц и css классов:**

**authorization.html**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Boo</title>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.7/css/bootstrap.min.css"/>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"/>

<link rel="stylesheet" href="static/css/authorization.css">

</head>

<body>

<div class="wrapper">

<div class="col-md-6">

<form class="form-horizontal">

<span class="heading">АВТОРИЗАЦИЯ</span>

<div class="form-group">

<input type="email" class="form-control" id="inputEmail" placeholder="E-mail">

<i class="fa fa-user"></i>

</div>

<div class="form-group help">

<input type="password" class="form-control" id="inputPassword" placeholder="Password">

<i class="fa fa-lock"></i>

<a href="#" class="fa fa-question-circle"></a>

</div>

<div class="form-group">

<div class="main-checkbox">

<input type="checkbox" value="none" id="checkbox1" name="check"/>

<label for="checkbox1"></label>

</div>

<span class="text">Запомнить</span>

<button type="submit" class="btn btn-default">ВХОД</button>

</div>

<a class="text-ref" href="#">Регистрация</a>

</form>

</div>

</div>

</body>

</html>

**content.html**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Boo</title>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.7/css/bootstrap.min.css"/>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"/>

<link rel="stylesheet" href="static/css/content.css">

</head>

<body>

<div class="content-text">

<h3>Глава 1</h3>

<p>Выйдя на улицу, натянула капюшон и закурила. Смена закончилась, можно выдохнуть и спокойно идти домой. Специально выжидала пару лишних часов и только после этого покидала клуб. Иначе чревато было наткнуться на ярых поклонников, считавших, что пределом моих мечтаний, является секс в подворотне с оборотнем или метаморфом.</p>

<p>А так они уставали ждать, переключали своё внимание на что-то другое и покидали свою вахту. В толстовке, джинсах и разбитых кроссовках, я вряд ли напоминала "Аврору", женщину-вамп, нимфоманку и дикую кошку. Заставлявшую своими танцами и полуобнажённым телом выть волков и свистеть демонов.</p>

<p>Логичный вопрос: кто я сама такая и зачем мне это? Не поверите, я не знаю, кто я. Очнулась в больнице с дикой болью в спине и дырой в голове. Не реальной, а в памяти. Нет, я понимала, как устроен мир, как говорить, есть и так далее. Но вот про себя не могла вспомнить ничего. Врачи разводили руками, исследовали меня, а потом махнули рукой и сказали, что со временем память вернётся.</p>

<p>Пока не особо. Мне очень помог молодой интерн, Иван — худощавый, улыбчивый, с шевелюрой светлых кудряшек на голове. Ухаживал за мной, веселил, и я настолько обнаглела, что теперь жила у него.</p>

<p>Мир… Ужасный наш мир. Людей осталось мало, кстати, я тоже человек. И с каждым годом становится всё меньше. Вокруг разврат, похоть и низменные инстинкты. Всё заполонили существа из других миров. Оборотни, демоны, вампиры, метаморфы и далее по списку. Они использовали людей, как пищу, рабов, энергетические батарейки. Светлые силы редкость тут. Они борются за души, но это место явно осталось за демонами.</p>

<p>Моя душа никого не интересовала, нужны были деньги и, увидев объявление, что требуются стриптизерши, со спокойной совестью пошла на пробы. Взяли меня сразу, тело жило своей жизнью, стоило услышать музыку. И сама того не желая, стала визитной карточкой заведения.</p>

<p>Единственным моим условием было — маска. Кружевная, шелковая, скрывающая лицо. Не хватало, чтобы днём меня узнал кто-то из посетителей. А видела таких часто, то в магазине, то просто на улице.</p>

<p>Ваня возмущался, отговаривал, но меня, как тянуло на всякую гадость. Нет. Никакого интима. Только танцы, но на грани фола. Завести, зажечь, чтобы в штанах тесно стало, а потом, плавно покачивая бедрами, уйти со сцены. Это был мой личный кайф.</p>

<p>Ходить по улицам одной было опасно, но четыре утра, это то время, когда все самые гадкие поступки уже совершены, все сыты и довольны.</p>

<p>Скоро рассвет, а это означает, что на улицах станет безлюдно, серо и дождливо. Солнца давно не видели эти улицы.</p>

<p>Вокруг грохочет музыка, пьяный смех, ругань, стоны. Город погряз в извращениях и преступлениях. А мне плевать. Надела наушники и, включив любимый трек, шлепая по лужам, шла домой.</p>

<p>Старый барак, да вся архитектура города напоминала трущобы, никто не занимался ничем, на улицах свалка, только клубы, да кабаки пестрили вывесками, тут были все деньги. А здесь деревянный пол, ступени скрипят, на втором этаже, вообще, дырка в полу, у соседей слышится мат и драки.</p>

<p>Вставила ключ в замочную скважину, но ее тут же отворил Ваня. На лице беспокойство, сколько ему не говори, чтобы не ждал и спал, бесполезно, ведь утром на дежурство в больницу!</p>

<p>— Всё в порядке? — с тревогой. Сняла капюшон и посмотрела на него. Беленький, кудрявый, с голубыми глазами, как ему удалось сохранить чистоту при всем том, что творится вокруг? Мысленно добавила ему крылья за спиной и усмехнулась.</p>

<p>— Конечно, что со мной будет? Всем Аврору подавай, — разуваясь. — Чего опять не спишь?</p>

<p>— Как тут уснёшь? — сокрушенно. — Ты голодная?</p>

<p>Кивнула, и он заспешил на кухню, гремит тарелками и кастрюлями. Тепло на душе так стало. Почему только возится со мной?</p>

<p>— Вань, — крикнула, заходя в ванну, настолько маленькую, что задеваешь попой стену, он ответил протяжным: "А-а-а?", — девушку тебе надо найти хорошую, — громко и уже себе под нос. — А то, как мамка себя ведёшь.</p>

<p>— Эль, — недовольно за моей спиной. — Мы уже это обсуждали. Нет пока хороших.</p>

<p>Подняла лицо и, смотря на него через зеркало.</p>

<p>— Тогда более менее нормальную. Я не маленькая, — серьёзно. — Не надо меня оберегать.</p>

<p>— Позволь мне самому это решить, — закидывая полотенце себе на плечо. — Я тебе мешаю?</p>

<p>Сейчас он выглядел очень забавно, ещё и хмурится. Каждый раз разговор приходит вот к этой фразе.</p>

<p>— Нет. Я тебе всем обязана, — всегда отвечаю так.</p>

<p>Удалился. Ведь порчу ему настроение не потому, что плохая, просто, правда, хочу ему счастья.</p>

<p>— Чай или кофе? — раздается через пару минут. Я уже смываю с себя чужие похотливые взгляды и лосьон с блеском для тела.</p>

<p>— Чай, — выкрикиваю. Вытираюсь и надеваю белую простую ночнушку до колен.</p>

<p>На кухне накрыт стол, жареная картошка с яйцами и чай. Запах… В животе заурчало.</p>

<p>— М-м-м, — блаженно тяну, наполняя рот хрустящей картошечкой. — Я тебе говорила, что люблю?</p>

<p>— Раз триста, — подперев подбородок локтем и наблюдая за мной. И ведь не уйдет, пока посуду не помоет за мной.</p>

<p>— Триста первый. Ты лучший, — отпивая сладкий чай.</p>

<p>Подкрепившись, закуриваю и наблюдаю за тем, как мужчина моет тарелку.</p>

<p>— Когда ты бросишь эту гадость? — через плечо. — Это убивает тебя.</p>

<p>— Меня эта жизнь убивает, — легкомысленно отвечаю. — Посмотри в окно. Там кого-то насилуют или истязают, а ты про мою вредную привычку переживаешь.</p>

<p>Качает головой, но больше мы не развиваем эту тему. Квартира однокомнатная. Я сплю на диване, а он на кресле-кровати. Чёрт возьми, ему вставать через три часа, я же смогу спать сколько угодно. Но как с ним бороться?</p>

<p>Проваливаясь в сон, бормочу:</p>

<p>— Спокойной ночи, Ванюша.</p>

<p>— И тебе Эля, — зевок сквозь дрёму. Почему Эля? Решила выпендриться и придумала себе имя — Элионора. Мне нравится, настоящего-то не помню.</p>

<p>Опять этот сон. Ненавижу его. Бесполезный и выматывающий.</p>

<p>Обстановку не рассмотреть, сколько не старалась. Только свет яркий, белый. Аж глазам больно. Голоса. Они спорят, ругаются, а я стою посередине, они расположились кругом. Лиц не видно, да и голова моя опущена. Глупость, но волосы у меня пшеничного цвета, а ещё, знаю наверняка, что и глаза голубые. Почему глупость? Да потому, что это не я. Придя в себя, в больнице, и взяв зеркало, внимательно рассмотрела отражение. Волосы черные, глаза карие, большие и выразительные, ресницы длинные и пушистые. Отвлеклась. Стою и чувствую себя жутко виноватой, опустошенной и точно знаю, что последует наказание. Страшное, непоправимое.</p>

<p>— Ты понимаешь, — этот голос, скручивает вены, пробивает током, вздрагиваю. — За что?</p>

<p>Они всё решили, за такое не прощают…</p>

<p>— Да, — поднимаю лицо, от света начинают слезиться глаза.</p>

<p>— Жаль, — грустно и с теплом. — Я так в тебя верил…</p>

<p>А потом накрывает боль, она выворачивает тело и душу, дикая, ни с чем не сравнимая боль. В спине, в области лопаток. Где-то кричит раненый зверь или нет, это я. Меня бьёт агония, тело скручивает судорогой. Сознание мутится, теряю реальность, падаю и падаю в бездну.</p>

<p>— Эля! — сквозь вату. — Эля! — сквозь слёзы и мой крик. Открываю глаза. Испуганный Ваня. Опять. Опять этот кошмар, опять напугала его и не дала спать.</p>

<p>— Всё в порядке, — шепчу, всхлипывая, волосы прилипали ко лбу, меня саму хоть выжимай. А спина в области лопаток просто горит огнём. — Всё хорошо.</p>

<p>— Снова? — спрашивает, подходя к тумбочке и доставая обезболивающее.</p>

<p>— Да, — смысла врать нет. Эти шрамы, будь они неладны. Мало того, что спину не украшают, ещё и болят. Какой садист мне лопатки располосовал?</p>

<p>— Выпей, — протянул мне две таблетки и стакан. Проглотила и не заметила.</p>

<p>— Вань, — вышло жалобно. Но он понял, мне нужно его обнять, мне очень плохо сейчас. Взял своё одеяло и лег ко мне. Обняла его и так хорошо стало. — Не ходи завтра на работу? — с надеждой.</p>

</div>

</body>

</html>

**index.html**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="anonymous">

<link rel="stylesheet" href="static/css/app.css">

<title>Boo</title>

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-light" style="background-color: #f90">

<a class="navbar-brand" href="#">Boo</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav mr-auto">

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbarDropdownBook" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

Книги

</a>

<div class="dropdown-menu" aria-labelledby="navbarDropdownBook">

<a class="dropdown-item" href="#">Рекомендуемые</a>

<a class="dropdown-item" href="#">Новинки</a>

<div class="dropdown-divider"></div>

<a class="dropdown-item" href="#">Все книги</a>

</div>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Обсуждения<span class="sr-only">(current)</span></a>

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMeet" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

Сообщество

</a>

<div class="dropdown-menu" aria-labelledby="navbarDropdownMeet">

<a class="dropdown-item" href="#">Конкурсы</a>

<a class="dropdown-item" href="#">Иллюстрации</a>

<div class="dropdown-divider"></div>

<a class="dropdown-item" href="#">Авторы</a>

</div>

</li>

<li class="nav-item active">

<a class="nav-link" href="#">Моя библиотека<span class="sr-only">(current)</span></a>

</li>

</ul>

<form class="form-inline my-2 my-lg-0">

<input class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search">

<button class="btn btn-outline-dark my-2 my-sm-0" type="submit">Поиск</button>

</form>

</div>

</nav>

<div class="container">

<div class="row">

<div class="col-9">

<div class="card-deck" style="margin-top: 10px">

<div class="card">

<img class="card-img-top" src="image/1.jpg" alt="Волшебство не вызывает привыкания-3">

<div class="card-body">

<a href="#"> <h5 class="card-title">Волшебство не вызывает привыкания-3</h5></a>

<p class="card-text">Третья часть невероятных приключений Тимофея Бухлина.</p>

</div>

<div class="card-footer">

<small class="text-muted">Last updated 3 mins ago</small>

</div>

</div>

<div class="card">

<img class="card-img-top" src="image/2.jpg" alt="Card image cap">

<div class="card-body">

<a href="#"><h5 class="card-title">CyberMoscow77</h5></a>

<p class="card-text">Я командир спец роты. Уволен с волчьим билетом за отказ расстреливать гражданских.

Я лишился всего: честного имени, социального рейтинга, положения. Даже импланты вырезали с мясом.

Этот мир продажных корпораций и всевидящего ока государства...

Но я верну свое, даже если мне придется его разрушить.

</p>

</div>

<div class="card-footer">

<small class="text-muted">Last updated 6 mins ago</small>

</div>

</div>

<div class="card">

<img class="card-img-top" src="image/3.jpg" alt="Card image cap">

<div class="card-body">

<a href="#"><h5 class="card-title">Книга десятая - Без масок</h5></a>

<p class="card-text">Тоётоми побеждены, осталось разобраться со схроном Древних и кланом Хейг.</p>

</div>

<div class="card-footer">

<small class="text-muted">Last updated 10 mins ago</small>

</div>

</div>

</div>

<div class="card-deck" style="margin-top: 5px">

<div class="card">

<img class="card-img-top" src="image/4.jpg" alt="Card image cap">

<div class="card-body">

<a href="#"><h5 class="card-title">Набирая силу (Альфа-2)</h5></a>

<p class="card-text">Некоторые считают, что страшнее этого места на Севере ничего нет и быть не может. Гиблые края, шрам на теле планеты, обитель самых страшных созданий Рока. Даже сама Смерть здесь отметилась, оставив целые территории, кищащие созданиями, в которых уродливо переплелись силы разной природы.

Попасть в Чащобу сложно, выбраться еще сложнее. А если тебя в нее занесло не по твоей воле, то и вовсе невозможно. Если ты, конечно, не великий герой.

Нет, я не великий герой. Я всего лишь ноль. Нулевка, проживший одну жизнь и выживающий в другой.

А еще я не знаю слово "невозможно".</p>

</div>

<div class="card-footer">

<small class="text-muted">Last updated 15 mins ago</small>

</div>

</div>

<div class="card">

<img class="card-img-top" src="image/5.jpg" alt="Card image cap">

<div class="card-body">

<a href="#"><h5 class="card-title">Должник</h5></a>

<p class="card-text">“Поздравляем, дорогой друг (ввести ФИО -)!

Мы рады приветствовать тебя и спешим рассказать радостную весть!

Именно ты был выбран добровольцем, для ответственной миссии по колонизации планеты П.Е.К.1.0. Это дает тебе шанс, скорейшей выплаты образовавшейся задолженности, в тридцать миллионов космокойнов.

Ты всегда можешь отказаться от свалившейся на тебя удачи и покончить жизнь самоубийством.

Желаешь умереть?

Да. Нет.</p>

</div>

<div class="card-footer">

<small class="text-muted">Last updated 20 mins ago</small>

</div>

</div>

<div class="card">

<img class="card-img-top" src="image/6.jpg" alt="Card image cap">

<div class="card-body">

<a href="#"><h5 class="card-title">INTERITE (Опустошение) 1.1</h5></a>

<p class="card-text">"INTERITE" продолжает раскрывать мир Игры Богов, с их взаимными интригами и множеством миров. В этот раз всё началось с того, что одна из сущностей порядка, называемых здесь богами, решила создать новый инструмент в попытке превзойти более старших коллег. Инструмент получился своевольный, немного неустойчивый, но полезный. И имя ему Азраил - сущность, сплетённая из девяти человеческих душ, помещённая в смертное тело, способная к непрерывному развитию и эволюции в попытке найти свою идеальную суть.

Азраилу поставлена весьма простая задача - окончить войну между двумя цивилизациями, технологической и магической, причём с мирным итогом. От этого будет зависеть также и судьба самого мира, ибо в противном случае судьями экзамен не будет засчитан - у богов ведь тоже есть свои правила. Впрочем как и конкуренты, и даже враги - такие же как они. Ведь никто и не обещал что будет легко!

Азраилу придётся нажить очень много врагов, и ещё больше своих врагов уничтожить!</p>

</div>

<div class="card-footer">

<small class="text-muted">Last updated 25 mins ago</small>

</div>

</div>

</div>

</div>

<div class="col">

<div class="panel">

<div class="panel-heading">

Последние комментарии

</div>

<div>

<a class="panel-ref" href="#">

<div class="panel-content">

<header class="panel-author">

Донцов Александр

</header>

<div class="panel-content-text">

Классная книга

</div>

</div>

</a>

<a class="panel-ref" href="#">

<div class="panel-content">

<header class="panel-author">

Виктор Цой

</header>

<div class="panel-content-text">

Скучная

</div>

</div>

</a>

<a class="panel-ref" href="#">

<div class="panel-content">

<header class="panel-author">

Иван Рудской

</header>

<div class="panel-content-text">

Слишком много пафоса

</div>

</div>

</a>

</div>

</div>

</div>

</div>

</div>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx" crossorigin="anonymous"></script>

</body>

</html>

**App.css**

.panel {

margin-top: 14px;

}

.panel-heading {

background-color: #ff9900;

border-top-left-radius: 3px;

border-top-right-radius: 3px;

display: flex;

justify-content: center;

align-items: center;

}

.panel-content {

display: block;

background-color: #ffdcb6;

border-bottom: 1px solid #717171;

font-size: 16px;

line-height: 15px;

}

.panel-content-text {

margin-top: 3px;

color: black;

}

.panel-content-text:hover {

margin-top: 3px;

color: red;

}

.panel-ref:hover{

text-decoration: none;

color: red;

}

.panel-author {

color: #717171;

}

**Authorization.css**

body {

background-color: black;

height: 100vh;

}

/\* Form Style \*/

.form-horizontal {

background: #fff;

padding-bottom: 40px;

border-radius: 15px;

text-align: center;

}

.form-horizontal .heading {

display: block;

font-size: 35px;

font-weight: 700;

padding: 35px 0;

border-bottom: 1px solid #f0f0f0;

margin-bottom: 30px;

}

.form-horizontal .form-group {

padding: 0 40px;

margin: 0 0 25px 0;

position: relative;

}

.form-horizontal .form-control {

background: #f0f0f0;

border: none;

border-radius: 20px;

box-shadow: none;

padding: 0 20px 0 45px;

height: 40px;

transition: all 0.3s ease 0s;

}

.form-horizontal .form-control:focus {

background: #e0e0e0;

box-shadow: none;

outline: 0 none;

}

.form-horizontal .form-group i {

position: absolute;

top: 12px;

left: 60px;

font-size: 17px;

color: #c8c8c8;

transition: all 0.5s ease 0s;

}

.form-horizontal .form-control:focus + i {

color: #ff9900;

}

.form-horizontal .fa-question-circle {

display: inline-block;

position: absolute;

top: 12px;

right: 60px;

font-size: 20px;

color: #808080;

transition: all 0.5s ease 0s;

}

.form-horizontal .fa-question-circle:hover {

color: #000;

}

.form-horizontal .main-checkbox {

float: left;

width: 20px;

height: 20px;

background: #ff9900;

border-radius: 50%;

position: relative;

margin: 5px 0 0 5px;

border: 1px solid #ff9900;

}

.form-horizontal .main-checkbox label {

width: 20px;

height: 20px;

position: absolute;

top: 0;

left: 0;

cursor: pointer;

}

.form-horizontal .main-checkbox label:after {

content: "";

width: 10px;

height: 5px;

position: absolute;

top: 5px;

left: 4px;

border: 3px solid #fff;

border-top: none;

border-right: none;

background: transparent;

opacity: 0;

-webkit-transform: rotate(-45deg);

transform: rotate(-45deg);

}

.form-horizontal .main-checkbox input[type=checkbox] {

visibility: hidden;

}

.form-horizontal .main-checkbox input[type=checkbox]:checked + label:after {

opacity: 1;

}

.form-horizontal .text {

float: left;

margin-left: 7px;

line-height: 20px;

padding-top: 5px;

text-transform: capitalize;

}

.form-horizontal .btn {

float: right;

font-size: 14px;

color: #fff;

background: #ff9900;

border-radius: 30px;

padding: 10px 25px;

border: none;

text-transform: capitalize;

transition: all 0.5s ease 0s;

}

@media only screen and (max-width: 479px) {

.form-horizontal .form-group {

padding: 0 25px;

}

.form-horizontal .form-group i {

left: 45px;

}

.form-horizontal .btn {

padding: 10px 20px;

}

}

.wrapper {

display:flex;

justify-content: center;

align-items: center;

height: inherit;

}

.text-ref {

color: gray;

}

.text-ref:hover {

color: #ff9900;

}

**Content.css**

.content-text {

margin: 0 200px;

}

**Глава 2. Приложение React**

Далее проект был перенесён на фреймворк React. Если рассматривать код HTML-файлов, то можно спокойно выделить корточки эмоций в один компонент, так как все карточки одинаковые, и передавать этим компонентам в качестве пропсов данные. В результате получилась следующая структура:

**Index.tsx**

import React from 'react';  
import ReactDOM from 'react-dom';  
import {Router} from 'react-router-dom'  
import history from './history'  
import 'bootstrap/dist/css/bootstrap.min.css';  
  
import App from "./app/App";  
  
const Index = () => {  
 return (  
 <Router history={history}>  
 <App/>  
 </Router>  
 )  
}  
  
ReactDOM.render(  
 <Index/>,  
 document.getElementById('root')  
);

**App.tsx**

import React from 'react';

import Routes from "../routes/routes";

const App = () => {

return (

<div>

<Routes />

</div>

);

}

export default App;

**mainPage.tsx**

import React from 'react'  
import Books from '../../components/Books/Books'  
import Comments from '../../components/Comments/Comments'

import s from './style.module.scss'

import Header from "../../components/header/Header";

const MainPage = () => {

return (

<div>

<Header/>

<div className={`${s.main} col-lg-10 col-xl-10 offset-lg-1 offset-xl-1`}>

<div>

<Books/>

</div>

<Comments/>

</div>

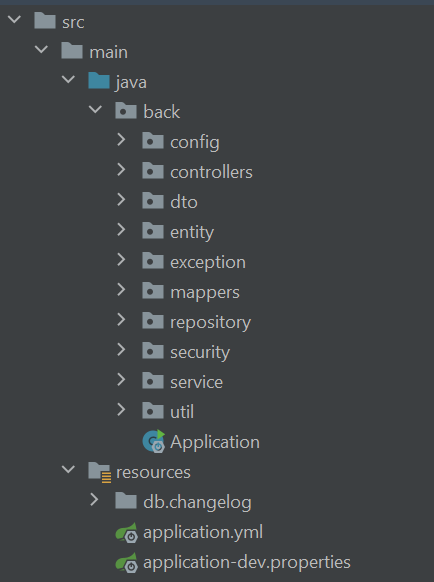
</div>

)

}

export default MainPage

**Глава 3. Серверная часть. Приложение Spring**



Изначально я написал контроллеры, для принятия запросов

BookController

package back.controllers;

import back.dto.book.BookDto;

import back.dto.book.ChapterBookDto;

import back.dto.book.ShortBookDto;

import back.service.BookService;

import lombok.RequiredArgsConstructor;

import org.springframework.core.io.Resource;

import org.springframework.http.CacheControl;

import org.springframework.http.HttpStatus;

import org.springframework.http.MediaType;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.\*;

import org.springframework.web.multipart.MultipartFile;

import javax.validation.Valid;

import java.util.List;

import java.util.concurrent.TimeUnit;

@RestController

@RequiredArgsConstructor

@RequestMapping("api/v1/book")

public class BookController {

private final BookService bookService;

@PostMapping

public ResponseEntity<Long> addBook(@RequestBody @Valid BookDto book, @RequestParam(required = false)MultipartFile logo) {

return ResponseEntity.ok().body(bookService.createBook(book, logo));

}

@PostMapping("/chapter")

public ResponseEntity<Long> addChapterBook(@RequestBody ChapterBookDto chapter) {

return ResponseEntity.ok().body(bookService.addPageBook(chapter));

}

@GetMapping("/chapter/{id}/{chapter}")

public ResponseEntity<ChapterBookDto> getBookChapter(@PathVariable Long id, @PathVariable("chapter") Long chapter) {

return ResponseEntity.ok().body(bookService.getBookChapter(id, chapter));

}

@DeleteMapping("/{id}")

public HttpStatus deleteBook(@PathVariable Long id) {

bookService.deleteBook(id);

return HttpStatus.OK;

}

@DeleteMapping("/chapter")

public HttpStatus deleteChapterBook(@RequestParam Long bookId, @RequestParam Long chapterId) {

bookService.deleteChapterBook(bookId, chapterId);

return HttpStatus.OK;

}

@PutMapping

public ResponseEntity<Long> updateBook(@RequestBody BookDto book) {

return ResponseEntity.ok().body(bookService.updateBook(book));

}

@GetMapping("/all")

public ResponseEntity<List<ShortBookDto>> getAllBook() {

return ResponseEntity.ok().body(bookService.getAllBook());

}

@GetMapping(value = "/logo/{bookId}",

produces = MediaType.IMAGE\_JPEG\_VALUE)

public ResponseEntity<Resource> download(@PathVariable(value = "bookId") Long id) {

Resource resource = bookService.downloadPhoto(id);

return ResponseEntity.ok()

.cacheControl(CacheControl.maxAge(60, TimeUnit.SECONDS).cachePublic())

.body(resource);

}

}

CommentsController

package back.controllers;

import back.entity.Comment;

import back.service.CommentService;

import lombok.RequiredArgsConstructor;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import java.util.List;

@RestController

@RequestMapping("api/v1/comments")

@RequiredArgsConstructor

public class CommentsController {

private final CommentService commentService;

@GetMapping("/last")

public ResponseEntity<List<Comment>> getLastComments(){

return ResponseEntity.ok(commentService.getLastComments());

}

}

После написал для контроллеров логику, а точнее сервисы

BookService

package back.service.impl;

import back.config.MultipartConfig;

import back.dto.book.BookDto;

import back.dto.book.ChapterBookDto;

import back.dto.book.ShortBookDto;

import back.entity.Book;

import back.entity.PageBook;

import back.exception.BadRequestException;

import back.exception.ResourceNotFoundException;

import back.mappers.BookMapper;

import back.mappers.PageBookMapper;

import back.repository.BookRepository;

import back.repository.PageBookRepository;

import back.service.BookService;

import lombok.RequiredArgsConstructor;

import lombok.SneakyThrows;

import lombok.extern.slf4j.Slf4j;

import org.springframework.core.io.ByteArrayResource;

import org.springframework.core.io.Resource;

import org.springframework.stereotype.Service;

import org.springframework.util.MimeTypeUtils;

import org.springframework.web.multipart.MultipartFile;

import java.io.File;

import java.nio.file.Files;

import java.nio.file.NoSuchFileException;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.util.List;

import java.util.UUID;

@Service

@RequiredArgsConstructor

@Slf4j

public class BookServiceImpl implements BookService {

private final MultipartConfig multipartConfig;

private final BookMapper bookMapper;

private final PageBookMapper pageBookMapper;

private final BookRepository bookRepository;

private final PageBookRepository pageBookRepository;

@Override

public Long createBook(BookDto book, MultipartFile logo) {

Book entity = bookMapper.toEntity(book);

entity.setPathLogo(downloadLogo(logo));

return bookRepository.save(entity).getId();

}

@Override

public Long addPageBook(ChapterBookDto chapter) {

if(pageBookRepository.findByBook\_IdAndNumPage(chapter.getBookId(),chapter.getNumChapter()).isPresent())

throw new BadRequestException("This chapter is exist");

PageBook entity = pageBookMapper.toEntity(chapter);

return pageBookRepository.save(entity).getId();

}

@Override

public List<ShortBookDto> getAllBook() {

List<Book> books = bookRepository.findAll();

return bookMapper.toShortBookDtos(books);

}

@Override

public ChapterBookDto getBookChapter(Long id, Long chapter) {

PageBook pageBook = pageBookRepository.findByBook\_IdAndNumPage(id, chapter).orElseThrow(() -> new ResourceNotFoundException("PageBook not found by book id = " + id + "chapter id " + chapter));

return pageBookMapper.toDto(pageBook);

}

@Override

public void deleteBook(Long id) {

Book book = bookRepository.findById(id).orElseThrow(() -> new ResourceNotFoundException("Book", "id", id));

for(PageBook page: book.getPages()){

pageBookRepository.delete(page);

}

bookRepository.delete(book);

}

@Override

public void deleteChapterBook(Long bookId, Long chapterId) {

PageBook pageBook = pageBookRepository.findByBook\_IdAndNumPage(bookId, chapterId).orElseThrow(() -> new ResourceNotFoundException("PageBook not found by book id = " + bookId + "chapter id " + chapterId));

pageBookRepository.delete(pageBook);

}

@Override

public Long updateBook(BookDto dto) {

return bookRepository.save(bookMapper.toEntity(dto)).getId();

}

@Override

public Long updatePageBook(ChapterBookDto dto) {

return pageBookRepository.save(pageBookMapper.toEntity(dto)).getId();

}

@SneakyThrows

@Override

public Resource downloadPhoto(Long bookId) {

log.info("download id: [{}] ", bookId);

Book book = bookRepository.findById(bookId).orElseThrow(() -> new ResourceNotFoundException("Photo logo", "bookId", bookId));

File file = new File(multipartConfig.getLocation() + book.getPathLogo());

String mimeType = Files.probeContentType(file.toPath());

if (mimeType == null) {

throw new ResourceNotFoundException("Photo", "id", bookId.toString() + " is broken mimeType = null");

}

Path path = Paths.get(file.getAbsolutePath());

try {

return new ByteArrayResource(Files.readAllBytes(path));

} catch (NoSuchFileException ex){

throw new ResourceNotFoundException("Photo", "bookId", bookId);

}

}

@SneakyThrows

private String downloadLogo(MultipartFile logo) {

if(logo == null)

return null;

if (logo.isEmpty()) {

throw new BadRequestException("Can not upload file Image. Cause file is empty");

}

String originalFilename = logo.getOriginalFilename();

if (originalFilename == null || originalFilename.isEmpty()) {

throw new BadRequestException("Can not upload file Image. Cause no name file");

}

String mimeType = logo.getContentType();

if (mimeType == null || !mimeType.startsWith("image/")) {

throw new BadRequestException("Can not upload file Image. Cause type is unsupported");

}

String fileExtension = "." + MimeTypeUtils.parseMimeType(mimeType).getSubtype();

String newNamePhotoUUID = UUID.randomUUID().toString();

String folderUuid = newNamePhotoUUID.substring(0, 2) + File.separator + newNamePhotoUUID.substring(2, 4);

String uuidNamePhoto = File.separator + newNamePhotoUUID + fileExtension;

String uri = folderUuid + uuidNamePhoto;

String directory = multipartConfig.getLocation() + folderUuid;

File file = fileWithDirectoryAssurance(directory, uuidNamePhoto);

logo.transferTo(file);

return uri;

}

private File fileWithDirectoryAssurance(String directory, String filename) {

File dir = new File(directory);

if (!dir.exists()) {

//noinspection ResultOfMethodCallIgnored

dir.mkdirs();

}

return new File(directory + filename);

}

}

CommentsService

package back.service.impl;

import back.entity.Comment;

import back.repository.CommentRepository;

import back.service.CommentService;

import lombok.RequiredArgsConstructor;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

@RequiredArgsConstructor

public class CommentServiceImpl implements CommentService {

private final CommentRepository commentRepository;

@Override

public List<Comment> getLastComments() {

return commentRepository.findAll();

}

}

Затем я занялся БД, написал миграции используя Liquibase

<databaseChangeLog xmlns="http://www.liquibase.org/xml/ns/dbchangelog"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog

http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-3.4.xsd">

<property name="blob\_type" value="bytea" dbms="postgresql"/>

<property name="blob\_type" value="blob" dbms="h2"/>

<changeSet id="01" author="Dontsov">

<createTable tableName="users">

<column name="id" autoIncrement="true" type="BIGINT" >

<constraints nullable="false" primaryKey="true"/>

</column>

<column name="name" type="varchar(255)">

<constraints nullable="false"/>

</column>

<column name="image\_url" type="varchar(512)"/>

<column name="email" type="varchar(100)">

<constraints unique="true" nullable="false"/>

</column>

<column name="email\_verified" type="BOOLEAN">

<constraints nullable="false"/>

</column>

<column name="password" type="varchar(512)"/>

<column name="provider" type="varchar(30)">

<constraints nullable="false"/>

</column>

<column name="provider\_id" type="varchar(60)">

<constraints nullable="false"/>

</column>

</createTable>

<createTable tableName="roles">

<column name="id" remarks="ID" type="BIGINT" autoIncrement="true">

<constraints primaryKey="true" primaryKeyName="role\_pkey"/>

</column>

<column name="name" type="varchar(255)">

<constraints unique="true" uniqueConstraintName="role\_name\_uk"/>

</column>

</createTable>

<createTable tableName="user2role">

<column name="user\_id" type="BIGINT">

<constraints nullable="false" primaryKey="true" primaryKeyName="user2role\_pkey"

references="users(id)" foreignKeyName="user2role\_user\_fk"

/>

</column>

<column name="role\_id" type="BIGINT">

<constraints nullable="false" primaryKey="true" primaryKeyName="user2role\_pkey"

references="roles(id)" foreignKeyName="user2role\_role\_fk"/>

</column>

</createTable>

<createTable tableName="books">

<column name="id" remarks="ID" type="BIGINT" autoIncrement="true">

<constraints primaryKey="true" primaryKeyName="book\_pkey"/>

</column>

<column name="time" type="varchar(30)"/>

<column name="name" type="varchar(255)">

<constraints nullable="false"/>

</column>

<column name="path\_logo" type="varchar(255)"/>

<column name="description" type="varchar(2048)"/>

<column name="author" type="varchar(512)"/>

</createTable>

<createTable tableName="genres">

<column name="id" remarks="ID" type="BIGINT" autoIncrement="true">

<constraints primaryKey="true" primaryKeyName="genre\_pkey"/>

</column>

<column name="name" type="varchar(255)">

<constraints unique="true" uniqueConstraintName="genres\_name\_uk"/>

</column>

</createTable>

<createTable tableName="book2genre">

<column name="book\_id" type="BIGINT">

<constraints nullable="false" primaryKey="true" primaryKeyName="book2genre\_pkey"

references="users(id)" foreignKeyName="book2genre\_book\_fk"

/>

</column>

<column name="genre\_id" type="BIGINT">

<constraints nullable="false" primaryKey="true" primaryKeyName="book2genre\_pkey"

references="roles(id)" foreignKeyName="book2genre\_genre\_fk"/>

</column>

</createTable>

<createTable tableName="page\_books">

<column name="id" remarks="ID" type="BIGINT" autoIncrement="true">

<constraints primaryKey="true" primaryKeyName="page\_book\_pkey"/>

</column>

<column name="text" type="TEXT">

<constraints nullable="false"/>

</column>

<column name="time" type="varchar(30)"/>

<column name="num\_page" type="INTEGER">

<constraints nullable="false"/>

</column>

<column name="book\_id" type="BIGINT">

<constraints nullable="false" references="books(id)" foreignKeyName="page\_books\_fk"/>

</column>

</createTable>

</changeSet>

<changeSet id="02" author="Dontsov">

<insert tableName="roles">

<column name="name">ROLE\_USER</column>

<column name="id">1</column>

</insert>

<insert tableName="roles">

<column name="name">ROLE\_ADMIN</column>

<column name="id">2</column>

</insert>

</changeSet>

<changeSet id="03" author="Dontsov">

<createTable tableName="comments">

<column name="id" type="BIGINT">

<constraints nullable="false" primaryKey="true"/>

</column>

<column name="author" type="varchar(100)"/>

<column name="time" type="varchar(100)"/>

<column name="text" type="varchar(512)"/>

</createTable>

</changeSet>

</databaseChangeLog>

Затем заполнил несколько книг в бд, также миграциями

<databaseChangeLog xmlns="http://www.liquibase.org/xml/ns/dbchangelog"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog

http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-3.4.xsd">

<changeSet id="01" author="Dontsov">

<insert tableName="comments">

<column name="id">1</column>

<column name="author">Донцов Александр</column>

<column name="text">Крутой сайт</column>

<column name="time">13.12.2020</column>

</insert>

<insert tableName="comments">

<column name="id">2</column>

<column name="author">Рандомный парень</column>

<column name="text">Крутой сайт</column>

<column name="time">13.12.2020</column>

</insert>

<insert tableName="comments">

<column name="id">3</column>

<column name="author">Не аноним</column>

<column name="text">Крутой сайт</column>

<column name="time">13.12.2020</column>

</insert>

<insert tableName="comments">

<column name="id">4</column>

<column name="author">Самый популярный человек</column>

<column name="text">Крутой сайт</column>

<column name="time">13.12.2020</column>

</insert>

</changeSet>

<changeSet id="02" author="Dontsov">

<insert tableName="books">

<column name="id">1</column>

<column name="name">Другой мир: видящий маг</column>

<column name="description">Четвертая книга серии «Другой мир».\n

Наконец-то я приблизился к тому, в каком направлении двигаться. Ведь в мои руки попала книга древних,

дающая ответы на многие мучившие меня вопросы. Но как говорится, аппетит приходит во время еды. Чем

больше я узнаю, тем сильнее понимаю, как много еще мне недоступно и что только предстоит узнать. Я всего

лишь делаю первые робкие шаги на пути к обретению могущества древних магов. Но для тех, кто меня

окружает, даже мои нынешние успехи кажутся феноменальными. И стоит быть крайне осторожным, чтобы не

заперли где-нибудь, как подопытного кролика, и не начали исследовать. А тут еще интриги, плетущиеся в

моем окружении, и в которых тоже не мешало бы разобраться! Голова идет кругом от попыток все успеть и

ничего не упустить. Но не зря говорят, что трудности лишь закаляют.

</column>

<column name="time">14.12.2020</column>

<column name="path\_logo">insert\f80f75719c8049e880556ac85d152bf4.jpg</column>

<column name="author">Иван Городецкий</column>

</insert>

<insert tableName="page\_books">

<column name="id">1</column>

<column name="text">«Основы аурного зрения и работы с перворунами». Название многообещающее, если судить из

того, что я успел узнать о магии древних. Честно говоря, даже немного страшно было открывать книгу и

начинать читать. Что если я окажусь просто не способен понять то, что там написано? Это ведь как

школьнику дать учебник для вузов! Тогда разочарование, боюсь, будет слишком сильным. Но глаза боятся,

руки делают. Тем более что слово «основы» давало надежду на то, что рассчитана книга все-таки не на

профессионалов. А на тех, кто только начал постигать эту дисциплину.\n

Опасения оказались все же не беспочвенны. В книге оказалось много терминов древнего языка, который был

понятен мне не настолько хорошо, как хотелось бы. До многого приходилось доходить чисто интуитивно. Но

худо-бедно удалось понять, что переходу на аурное зрение можно научиться с помощью специальных техник. В

книге, которая оказалась учебником для студентов времен древних магов, приводилось их сразу пять.

И я, естественно, попытался применить их. Вот только меня ждал большущий облом! Несмотря на то, что

упражнения были в чем-то похожи на медитации, и я думал, что труда они не составят, не тут-то было. И

вроде все правильно делаю – сосредотачиваюсь на источнике и пытаюсь увидеть его глубину и взаимосвязь с

внешним миром. Но никакого аурного зрения при этом не возникает.\n

Вздохнув, я решил отложить это на время и начал листать учебник дальше. Из того, что понял. Аурное

зрение – особый пласт реальности, который позволяет видеть все энергетические потоки мира в виде того,

что древние назвали перворунами. Весь мир, по их утверждению, состоит из них. И маг с помощью своего

эльма может влиять на них. Если, конечно, научится видеть этот пласт реальности и будет понимать, что

делает. При этом чем тоньше эльм, тем более действенными могут быть манипуляции с перворунами. Последнее

меня особенно порадовало и заставило воспрянуть духом. Как оказалось, то, что я считал недостатком, по

меркам древних магов, наоборот, являлось достоинством. Еще бы научиться как-то им пользоваться! Я тяжко

вздохнул.\n

По сути, реальность можно сравнить со слоеным тортом. Перворуны на последующих пластах реальности

трансформируются в то, с чем все более сложно работать. Нужны более изощренные методы и что-то вроде

костылей, которые помогают манипулировать с энергией на более грубых пластах. Для сравнения, есть, к

примеру, перворуна, обозначающая огонь. Тут ее называли «фера». Если посмотреть на нее в истинном

зрении, доступном видящим магам, то легкая и незамысловатая закорючка превращается в целое кружево. И

вот это самое кружево и нужно нарисовать, если не владеешь аурным зрением и не способен создать

непосредственно перворуну.\n

Я, разумеется, попробовал сплести перворуны на том плане, что был мне доступен. Ничего не получилось.

Они просто разрушались при каждой попытке. Работать как надо «фера» будет лишь при ее прорисовке в

аурном плане или в виде плетения на энергетическом. И никак иначе.

Перворун в книге приводилось всего тридцать, но при этом они могли сочетаться между собой и сливаться,

благодаря чему получались более сложные плетения. Также имелись еще насечки, которые при рисовке на

перворунах или их комбинациях могли оказывать существенное влияние. Усиливать мощь или уменьшать, к

примеру, менять цвет и многое другое. Целый особый язык, чем-то напоминающий даже язык программирования

в нашем мире. Впрочем, весьма отдаленно.\n

Заинтересовало еще упоминание о куполе прорицания. Это область в аурном зрении, которая доступна при его

развитии, позволяющая создать вокруг мага что-то вроде экспериментального пространства. На нем можно

увидеть, как то или иное сочетание перворун будет проявляться не только в энергетическом, но и

физическом плане. Какой эффект будет оказывать плетение. Очень удобно, кстати! Можно обойтись без

рискованных экспериментов, а сразу понять, удачное ли получилось плетение. Что в него можно добавить

или, наоборот, устранить.\n

Ух, если я все это освою, как артефактору, мне точно не будет равных! Даже не потребуется больше помощь

Гриана Борда. Но губу пока лучше не раскатывать. Все ведь не так просто.

Но перворуны я начал изучать со всем возможным рвением, как и их наиболее распространенные комбинации.

Подозреваю, древние маги знали их куда больше, но в учебник для начинающих внесли лишь самые

распространенные. Впрочем, мне и этого пока более чем достаточно. Площадка для экспериментов просто

гигантская! От перспектив дух захватывало.\n

Что особенно порадовало, так это некоторые примеры, имеющиеся в книгах. Соответствие обычных плетений

перворунам и их комбинациям. Причем имелись и такие плетения, которые в этом мире считались утерянными.

Стук в дверь заставил оторваться от увлекательного чтива. Я с досадой посмотрел на часы, висящие на

стене. Уже начало двенадцатого ночи. Кто мог заявиться ко мне в такое время?

Книгу пришлось спрятать подальше от любопытных глаз. Пусть увидеть, что именно я читаю, и не смогут, но

уже по внешнему виду книги станет понятно, что она старинная и дорогая. Для простолюдина, пусть и

проявляющего успехи в торговле, роскошь немалая. Даже если просто сболтнут в разговоре о том, что у меня

есть такое сокровище, это может навлечь неприятности. Кто-то пожелает отнять или даже просто

конфисковать в пользу короны. Мрачно усмехнулся, вспомнив о последней беседе с королем. Да уж, тот своей

выгоды не упустит, это точно! Обдерет как липку – только прояви неосторожность.\n

Моргана, стоящая за дверями, видимо, что-то такое прочла по моему лицу, поскольку неуверенно спросила:

– Я не вовремя?\n

– Нет, просто учебу надо было наверстывать, – я посторонился, пропуская ее внутрь.\n

Все-таки мои предположения о том, что она может проявить инициативу, оказались пророческими. Может, и

стоило бы выпроводить поскорее, сказав, что нужно еще многое выучить или просто отдохнуть. Но я подумал:

какого черта? И так мозги уже вскипают. А видимых успехов в аурном зрении я сегодня точно не достигну.

Так почему бы не переключиться на более приятное занятие, которое поможет отбросить все проблемы и

расслабиться?\n

Прильнув ко мне своим горячим упругим телом, Моргана жарко зашептала на ухо:\n

– Честно говоря, я безумно соскучилась!\n

Ее язычок дразняще скользнул в ухо, вызывая по телу легкую дрожь. А потом она, не тратя больше времени

на прелюдии, начала быстро снимать с меня и себя одежду. Даже помогать особо не пришлось.\n

Усмехнувшись, я с удовольствием позволил ей делать все, что заблагорассудится. И вскоре посторонние

мысли попросту улетучились. Моргана, как всегда, была безумно страстной и напористой, что мне даже

нравилось. Но всю ночь кувыркаться с ней не было возможности. Все-таки не мешало бы поспать хотя бы пару

часиков. Умница Моргана чутко уловила мое настроение и не стала слишком уж усердствовать.\n

Поцеловав меня напоследок особенно чувственно, быстро оделась и покинула мою комнату. Ну до чего же

идеальная любовница, что ни говори! Даже не ожидал, что с темной эльфийкой окажется так просто и приятно

иметь дело. Вот только до конца расслабляться с ней все же не стоит. У Морганы есть куча скелетов в

шкафу. И неизвестно, какие у нее дальнейшие планы. Пока она не знает о моем происхождении, девушка

вполне может позволить себе просто развлечься. Но кто знает, как отреагирует на то, что я сын тирра? Не

пожелает ли использовать это обстоятельство в свою пользу? Ну да ладно, сейчас об этом можно не думать.

И я позволил себе погрузиться в долгожданный и так необходимый мне сон.\n

</column>

<column name="num\_page">0</column>

<column name="book\_id">1</column>

</insert>

<insert tableName="page\_books">

<column name="id">2</column>

<column name="text">Эйра Айнтерела я утром отыскал в тренировочном зале в самой Академии. Именно там он

назначил мне встречу для предстоящего первого занятия. Честно говоря, предпочел бы позаниматься на

свежем воздухе. Все-таки последние теплые деньки хочется использовать по максимуму. Скоро наступит

полноценная осень со своими извечными спутниками: дождями и холодом. Но оспаривать пожелание нового

наставника по таким пустякам было бы глупо.\n

Удивило то, что Айнтерел, в отличие от моей прежней наставницы Морганы Сатари, не занимался разминкой в

ожидании ученика. Он сидел в позе лотоса и медитировал.\n

Некоторое время я переминался с ноги на ногу, не решаясь прервать его концентрацию. Уже раздумывал, не

начать ли пока разминаться самому, пока он не соизволит заметить, но в этот момент в меня понеслось

что-то. Не ожидавший этого от неподвижно сидящего и безмятежно-расслабленного эльфа, я вовремя не

отреагировал. И мне в лицо довольно болезненно шмякнулся небольшой водный шарик, расплескавшись

брызгами. Какого лешего?!\n

И что самое странное, Айнтерел сидел с закрытыми глазами и видеть меня, по идее, не мог. От нового

шарика я все же уклонился, хотя и опять даже не уловил движения или какой-то реакции от эльфа. Но уже

следующий, вылетевший всего через мгновение, угодил мне в ухо. Ну, гад эльфийский!\n

Мстительно улыбнувшись, я направил в сторону Айнтерела ментально-универсальный щуп. И, к собственному

удивлению, не уловил ничего. Абсолютно! Обычно, даже несмотря на защитную брошь, я мог улавливать

отголоски эмоций и намерений. В голове же эльфа будто царила полная пустота. Попытался ударить

ментально, и мой заряд энергии бессильно растекся в пустоте. Новый шарик угодил в лоб, и я в сердцах

скрипнул зубами.\n

А еще по спине пробежал неприятный холодок, когда понял, что если бы выступал против такого противника в

поединке, у меня не было бы ни шанса. Как он вообще это делает?!\n

Светлый эльф оказался на ногах так стремительно, что я даже вздрогнул. С губ его сорвался короткий и

четкий приказ, произнесенный совершенно лишенным эмоций тоном:\n

– Бери тренировочный меч и нападай! Твоя задача – хотя бы раз задеть меня.\n

Я честно попытался это сделать, но уровень противника был несоизмеримо выше. И что самое поразительное –

эльф по-прежнему действовал с закрытыми глазами. На первый или второй уровень транса я не переходил,

прекрасно понимая, что тогда он тоже это сделает, и я только быстрее вымотаюсь. Безуспешно пытался

достать противника, используя все доступные мне навыки. Но это было тем же самым, что поймать ветер.

Айнтерел каждый раз безошибочно понимал, где я нахожусь, пусть и не видел этого. А в том, что он и не

думает подглядывать, я был точно уверен.\n

Наконец, я настолько вымотался, что прекратил новые попытки и остановился отдышаться. Светлый эльф

открыл глаза и недовольно покачал головой.\n

– Плохо! Ты не используешь все свои возможности по максимуму. А туда же – уже посчитал, что можешь

перейти на третий уровень транса. До того, как ты будешь реально готов, еще очень и очень долго,

мальчик.\n

Хотел было возразить на такое обращение, но мысленно усмехнулся. Для эльфа я и правда могу казаться еще

ребенком. Сколько ему, интересно? По ушастым вообще же не поймешь.\n

– Отличие нашей методики от темных в том, что мы делаем акцент не на максимальном усовершенствовании

физического тела. Пусть это и тоже важно. Но без идеального восприятия чувства пространства ты не

сможешь в полной мере познать все, что тебе доступно. Темные, конечно, учатся и этому, но не так, как

мы. Они к этому приходят уже после достаточного развития тела. А мы с этого начинаем учить. Пока ученик

не достигнет совершенства в подобном, не переходим к силовым тренировкам. Впрочем, в твоем случае мне

придется сделать исключение. У вас, людей, нет столько времени, сколько у нас, чтобы потратить его

достаточно для самосовершенствования. Так что тренировки будут проходить в двух аспектах. Но все-таки

советую тебе сконцентрироваться на том, что у тебя хромает больше. Восприятие и чуткость к тому, что

происходит вокруг тебя. Впрочем, одно с другим взаимосвязано. Ты увидишь, как легче тебе будет

заниматься и физическими упражнениям, когда научишься входить в это особое состояние. Мы называем его

«сархар». Полная чистота в голове и четкое понимание того, что происходит вокруг даже без применения

органов чувств.\n

– Не совсем понял, если честно.\n

Эльф поморщился. Ему явно не хотелось раскрывать мне больше положенного, но раз уж взялся учить, то

собирался подойти к делу ответственно.\n

– Знаешь, как обучают у нас особо нерадивых учеников, кто не может долгое время освоить «сархар» с

помощью медитативных практик?\n

Я отрицательно замотал головой.\n

– Их помещают в абсолютно темное помещение и заставляют ориентироваться с помощью других органов чувств.

Сначала им легче всего это сделать с помощью слуха. Но через какое-то время их лишают и его. Для такого

у нас есть особые артефакты. Остаются обоняние и осязание. Постепенно лишают и их. После вдыхания особых

снадобий нюх пропадает на несколько дней. А насчет осязания… Есть и особые паралитические экстракты.\n

Я невольно передернул плечами. М-да, не хотелось бы мне оказаться на месте их нерадивых учеников!

– И вот когда ученик становится способен воспринимать мир без помощи физического тела, он и понимает в

полной мере, чтотакое «сархар».\n

– Эм...м…\n

Я был озадачен. Как это возможно вообще? Воспринимать мир без помощи органов чувств? Или, скорее,

раскрыть в себе то чувство, в которое многие даже и не верят. Интуицию или что-то вроде того.\n

– Именно с помощью «сархара» вы и знали, где я нахожусь? – осторожно спросил я.\n

– Так и есть. Умение ощущать чужое присутствие таким образом может быть весьма полезно и когда речь о

магах, владеющих «темной завесой». Постигшие «сархар» все равно ощутят чужое присутствие и смогут

действовать более эффективно. А также поймут, если в помещении находятся посторонние. Разумеется, если

противник тоже не будет в состоянии «сархара». Все мастера-воины светлых эльфов обучаются такому. Насчет

дроу, не знаю, но думаю, и у них есть нечто подобное. А еще в состоянии «сархара» ни один, даже самый

сильный менталист не сможет ничего сделать против тебя.\n

М-да, враждовать с эльфами действительно было бы весьма неразумно! Остается порадоваться, что, как и у

нас, у них далеко не все способны стать воинами-мастерами.\n

– Вы научите меня этому? – осторожно спросил.\n

– По крайней мере, попытаюсь, – хмыкнул эйр Айнтерел. – Для мага войти в состояние «сархара» обычно

полегче, чем для обычного человека. Так что будем надеяться, что радикальных мер не потребуется.

Невольно сглотнув, я кивнул.\n

– Что нужно делать?\n

....

</column>

<column name="num\_page">1</column>

<column name="book\_id">1</column>

</insert>

<insert tableName="books">

<column name="id">2</column>

<column name="name">Моё пространственное убежище. Книга 2. На Берлин!</column>

<column name="description">Стоило Сергею покинуть Париж как началось вторжение в убежище! Главному герою

предстоит осознать, что всё что было раньше это цветочки, а ягодки... да сама центральная Европа, та ещё

ягодка...

Берлин он вроде так близко, но в тоже время так далеко...

</column>

<column name="time">14.12.2020</column>

<column name="path\_logo">insert\46bb25735ce543f8a3ebd9b618ea2669.jpg</column>

<column name="author">Дорничев Дмитрий</column>

</insert>

<insert tableName="page\_books">

<column name="id">3</column>

<column name="text">Некоторый текст

</column>

<column name="num\_page">0</column>

<column name="book\_id">2</column>

</insert>

<insert tableName="books">

<column name="id">3</column>

<column name="name">Книга</column>

<column name="description">Текст</column>

<column name="time">14.12.2020</column>

<column name="path\_logo">insert\unknown-book-263x300.jpg</column>

<column name="author">Книга</column>

</insert>

<insert tableName="page\_books">

<column name="id">4</column>

<column name="text">Некоторый текст

</column>

<column name="num\_page">0</column>

<column name="book\_id">3</column>

</insert>

</changeSet>

</databaseChangeLog>

Для тестирования запросов написал интеграционные тесты  
package back.controllers;

import lombok.SneakyThrows;

import org.junit.Test;

import org.springframework.http.MediaType;

import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.\*;

import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.status;

public class BookControllerTest extends ApiBaseTest {

@SneakyThrows

@Test

public void addBook() {

mockMvc.perform(post("/api/v1/book").content(getBookDto(null)).contentType(MediaType.APPLICATION\_JSON)).andExpect(status().isOk());

}

@SneakyThrows

@Test

public void addChapterBook() {

mockMvc.perform(post("/api/v1/book/chapter").contentType(MediaType.APPLICATION\_JSON).content(getChapterBookDto())).andExpect(status().isOk());

}

@SneakyThrows

@Test

public void getBookChapter() {

mockMvc.perform(get("/api/v1/book/chapter/1/0")).andExpect(status().isOk());

}

@SneakyThrows

@Test

public void deleteBook() {

mockMvc.perform(delete("/api/v1/book/3")).andExpect(status().isOk());

}

@SneakyThrows

@Test

public void deleteChapterBook() {

mockMvc.perform(delete("/api/v1/book/chapter").param("bookId", "2").param("chapterId", "0")).andExpect(status().isOk());

}

@SneakyThrows

@Test

public void updateBook() {

mockMvc.perform(put("/api/v1/book").content(getBookDto(2L)).contentType(MediaType.APPLICATION\_JSON)).andExpect(status().isOk());

}

@SneakyThrows

@Test

public void getAllBook() {

mockMvc.perform(get("/api/v1/book/all")).andExpect(status().isOk());

}

@SneakyThrows

@Test

public void download() {

mockMvc.perform(get("/api/v1/book/logo/1")).andExpect(status().isOk());

}

}

Для авторизации через ВК, была использована библиотека Spring Security, за счет нее происходит регистрация пользователя в бд.

Инициализация данных приложения для работы с ВК

public class CustomOAuth2Provider {

private CustomOAuth2Provider(){

}

public static ClientRegistration getVk() {

ClientRegistration.Builder builder = ClientRegistration.withRegistrationId("vk");

builder.clientAuthenticationMethod(ClientAuthenticationMethod.POST);

builder.scope("email");//доступ

builder.authorizationUri("https://oauth.vk.com/authorize?v=5.95");

builder.tokenUri("https://oauth.vk.com/access\_token");

builder.userInfoUri("https://api.vk.com/method/users.get?{user\_id}&v=5.95&fields=photo\_id,verified,sex,bdate,city,country,photo\_max,home\_town,has\_photo&display=popup&lang=ru&access\_token={access\_token}");

builder.clientName("vkontakte");

builder.authorizationGrantType(AuthorizationGrantType.AUTHORIZATION\_CODE);//-------

builder.redirectUri("{baseUrl}/oauth2/callback/{registrationId}");//адрес куда возвращается код пользователя

builder.clientId("7004922");

builder.clientSecret("4DQyjH3uP8VphRMfGOyg");

builder.userNameAttributeName("user\_id");

return builder.build();

}

}

Регистрация пользователя

package back.security.oauth2;

import back.entity.Role;

import back.entity.User;

import back.exception.OAuth2AuthenticationProcessingException;

import back.repository.UserRepository;

import back.security.UserPrincipal;

import back.security.oauth2.user.OAuth2UserInfo;

import back.security.oauth2.user.OAuth2UserInfoFactory;

import lombok.RequiredArgsConstructor;

import lombok.SneakyThrows;

import org.springframework.http.HttpEntity;

import org.springframework.http.HttpMethod;

import org.springframework.http.ResponseEntity;

import org.springframework.security.authentication.InternalAuthenticationServiceException;

import org.springframework.security.core.GrantedAuthority;

import org.springframework.security.oauth2.client.userinfo.DefaultOAuth2UserService;

import org.springframework.security.oauth2.client.userinfo.OAuth2UserRequest;

import org.springframework.security.oauth2.core.OAuth2AuthenticationException;

import org.springframework.security.oauth2.core.user.DefaultOAuth2User;

import org.springframework.security.oauth2.core.user.OAuth2User;

import org.springframework.security.oauth2.core.user.OAuth2UserAuthority;

import org.springframework.stereotype.Service;

import org.springframework.util.LinkedMultiValueMap;

import org.springframework.util.MultiValueMap;

import org.springframework.web.client.HttpClientErrorException;

import org.springframework.web.client.RestTemplate;

import javax.naming.AuthenticationException;

import javax.security.auth.message.AuthException;

import java.util.\*;

@Service

@RequiredArgsConstructor

public class CustomOAuth2UserService extends DefaultOAuth2UserService {

private final UserRepository userRepository;

@SneakyThrows

@Override

public OAuth2User loadUser(OAuth2UserRequest oAuth2UserRequest) throws OAuth2AuthenticationException {

OAuth2User oAuth2User;

if(oAuth2UserRequest.getClientRegistration().getRegistrationId().equals("vk"))

oAuth2User = loadVkUser(oAuth2UserRequest);

else

oAuth2User = super.loadUser(oAuth2UserRequest);

try {

return processOAuth2User(oAuth2UserRequest, oAuth2User);

} catch (AuthenticationException ex){

throw new AuthException(ex.getMessage());

} catch (Exception ex) {

throw new InternalAuthenticationServiceException(ex.getMessage(), ex.getCause());

}

}

private OAuth2User processOAuth2User(OAuth2UserRequest oAuth2UserRequest, OAuth2User oAuth2User) throws AuthenticationException {

OAuth2UserInfo oAuth2UserInfo = OAuth2UserInfoFactory.getOAuth2UserInfo(oAuth2UserRequest.getClientRegistration().getRegistrationId(), oAuth2User.getAttributes());

User socialUser;

Optional<User> userOptional = userRepository.findByProviderId(oAuth2UserInfo.getAttributes().get("user\_id").toString());

if(userOptional.isPresent()) {

socialUser = userOptional.get();

if (!socialUser.getProvider().equals(oAuth2UserRequest.getClientRegistration().getRegistrationId()))

throw new OAuth2AuthenticationProcessingException("Looks like you're signed up with " +

socialUser.getProvider() + " account. Please use your " + socialUser.getProvider() +

" account to login.");

socialUser = updateExistingUser(socialUser, oAuth2UserInfo);

} else

socialUser = registerNewUser(oAuth2UserRequest, oAuth2UserInfo);

return UserPrincipal.create(socialUser, oAuth2User.getAttributes());

}

private User registerNewUser(OAuth2UserRequest oAuth2UserRequest, OAuth2UserInfo oAuth2UserInfo) {

User user = new User();

user.setProvider(oAuth2UserRequest.getClientRegistration().getRegistrationId());

user.setProviderId(oAuth2UserInfo.getAttributes().get("user\_id").toString());

user.setName(oAuth2UserInfo.getAttributes().get("first\_name").toString() + " " + oAuth2UserInfo.getAttributes().get("last\_name").toString());

user.setEmail(oAuth2UserInfo.getEmail());

user.setRoles(Collections.singletonList(new Role(Role.ROLE\_USER)));

user.setImageUrl(oAuth2UserInfo.getAttributes().get("photo\_max").toString());

return userRepository.save(user);

}

private User updateExistingUser(User existingUser, OAuth2UserInfo oAuth2UserInfo) {

existingUser.setName(oAuth2UserInfo.getAttributes().get("first\_name").toString() + " " + oAuth2UserInfo.getAttributes().get("last\_name").toString());

existingUser.setEmail(oAuth2UserInfo.getEmail());

existingUser.setImageUrl(oAuth2UserInfo.getAttributes().get("photo\_max").toString());

return userRepository.save(existingUser);

}

private OAuth2User loadVkUser(OAuth2UserRequest oAuth2UserRequest) {

RestTemplate template = new RestTemplate();

MultiValueMap<String, String> headers = new LinkedMultiValueMap<>();

headers.add("Content-Type", "application/json");

headers.add("Authorization", oAuth2UserRequest.getAccessToken().getTokenType().getValue() + " " + oAuth2UserRequest.getAccessToken().getTokenValue());

HttpEntity<?> httpRequest = new HttpEntity<>(headers);

String uri = oAuth2UserRequest.getClientRegistration().getProviderDetails().getUserInfoEndpoint().getUri();

String userNameAttributeName = oAuth2UserRequest.getClientRegistration().getProviderDetails().getUserInfoEndpoint().getUserNameAttributeName();

uri = uri.replace("{user\_id}", userNameAttributeName + "=" + oAuth2UserRequest.getAdditionalParameters().get(userNameAttributeName));

uri = uri.replace("{access\_token}", oAuth2UserRequest.getAccessToken().getTokenValue());

try {

ResponseEntity<Map> entity = template.exchange(uri, HttpMethod.GET, httpRequest, Map.class);

Map<?, ?> body = entity.getBody();

ArrayList<?> valueList = (ArrayList<?>) body.get("response");

Map<String, Object> userAttributes = (Map<String, Object>) valueList.get(0);

userAttributes.put(userNameAttributeName, oAuth2UserRequest.getAdditionalParameters().get(userNameAttributeName));

userAttributes.put("email", oAuth2UserRequest.getAdditionalParameters().get("email"));

Set<GrantedAuthority> authorities = Collections.singleton(new OAuth2UserAuthority(userAttributes));

return new DefaultOAuth2User(authorities, userAttributes, userNameAttributeName);

} catch (HttpClientErrorException ex) {

ex.printStackTrace();

throw new OAuth2AuthenticationProcessingException(ex.getMessage());

}

}

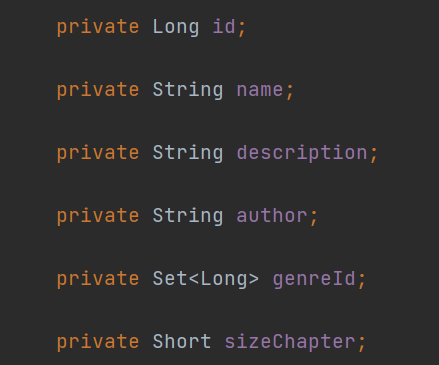
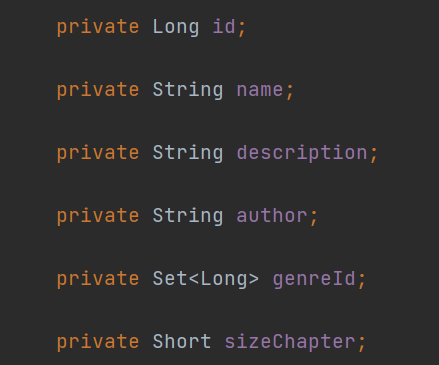
}

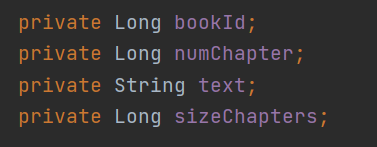
**Глава 5. REST API**

Для взаимодействия с базой данных было реализовано следующее REST API:

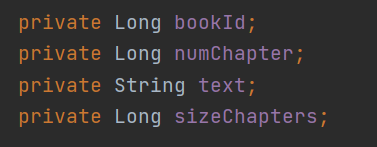
Работа с книгами.

/api/v1/book

1. Создание книги POST  
   Отправляется дто с информацией о книге
2. Удаление книги DELETE  
   Передается id
3. Обновление книги PUT  
   
4. Получение всех книг GET
5. Получение главы книги GET  
   Передается id книги и номер главы
6. Создание главы POST



1. Обновление главы PUT



Работа с авторизацией

Редирект на /authorize/vk, он перенаправит на страницу авторизации вк, который в последствии вернет токен авторизации.

**Глава 6. Взаимодействие приложения с API**

Вся логика взаимодействия с REST API была реализована в ранее созданном приложении React:

Страница сохранения токена на фронте

import React from 'react'

import history from "../../history";

const AuthPage = () => {

React.useEffect(() => {

if (localStorage.getItem('token') == null)

localStorage.setItem('token', window.location.href.split('/')[4])

history.push("/main")

}, [])

return (

<div> </div>

)

}

export default AuthPage;

Страница входа на сайт

import React from 'react';

import history from "../../history";

import './style.css';

const LoginPage = () => {

function handleClick() {

console.log("Нажал");

window.location.href = "http://localhost:8090/oauth2/authorize/vk";

}

return (

<div className="wrapper">

<div className="col-md-6">

<form className="form-horizontal">

<span className="heading">АВТОРИЗАЦИЯ ВК</span>

<div className="main-btn form-group">

<button onClick={handleClick} type="button" className="btn btn-default">ВХОД</button>

</div>

</form>

</div>

</div>)

}

export default LoginPage;

Главная страница

import React from 'react'

import Books from '../../components/Books/Books'

import Comments from '../../components/Comments/Comments'

import s from './style.module.scss'

import Header from "../../components/header/Header";

import {authService} from "../../services/auth-service/auth-service";

const MainPage = () => {

React.useEffect(() => {

if (localStorage.getItem('token') == null)

localStorage.setItem('token', window.location.href.split('/')[4])

}, [])

React.useEffect(() => {

authService.checkTokenEvent()

}, [])

return (

<div>

<Header/>

<div className={`${s.main} col-lg-10 col-xl-10 offset-lg-1 offset-xl-1`}>

<div>

<Books/>

</div>

<Comments/>

</div>x

</div>

)

}

export default MainPage

Страница для чтения

import React from 'react'

import Header from "../../components/header/Header";

import {authService} from "../../services/auth-service/auth-service";

import {useStore} from "effector-react";

import {readService} from "../../services/read-service/read-service";

import a from "./style.module.scss"

const ReadPage = () => {

const read = useStore(readService.$read)

React.useEffect(() => {

authService.checkTokenEvent()

}, [])

React.useEffect(() => {

readService.getBookChapter()

}, [])

var num = (read == null) ? 0:read.numChapter;

const size = (read == null) ? 0:read.sizeChapters;

const url1 = `${(num - 1)}`;

const url2 = `${num}`;

const url3 = `${num+1}`;

const paginations = <ul className={a.pagination}>

{num-1 >= 0 ? <li className={a.pagination}><a href={url1}>{num}</a></li>:''}

<li className={a.pagination}><a href={url2}>{num+1}</a></li>

{num+1 >= size ? '':<li className={a.pagination}><a href={url3}>{num+2}</a></li>}

</ul>;

return (

<div>

<Header/>

<div style={{margin: '20px 10% 0 10%'}}>

{read != null ? read.text.split('\\n').map(a => <p>{a}</p>):''}

</div>

{paginations}

</div>

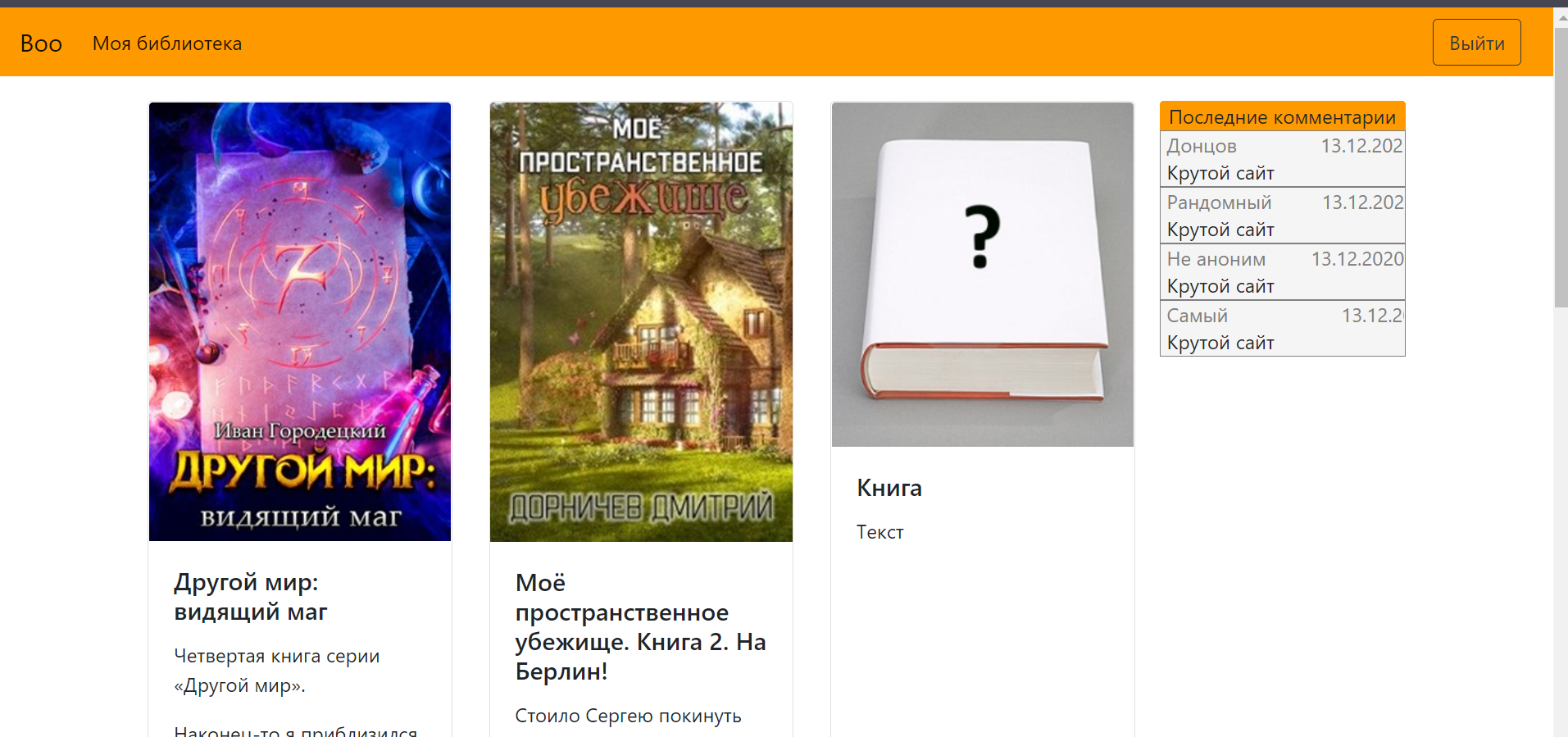
)

}

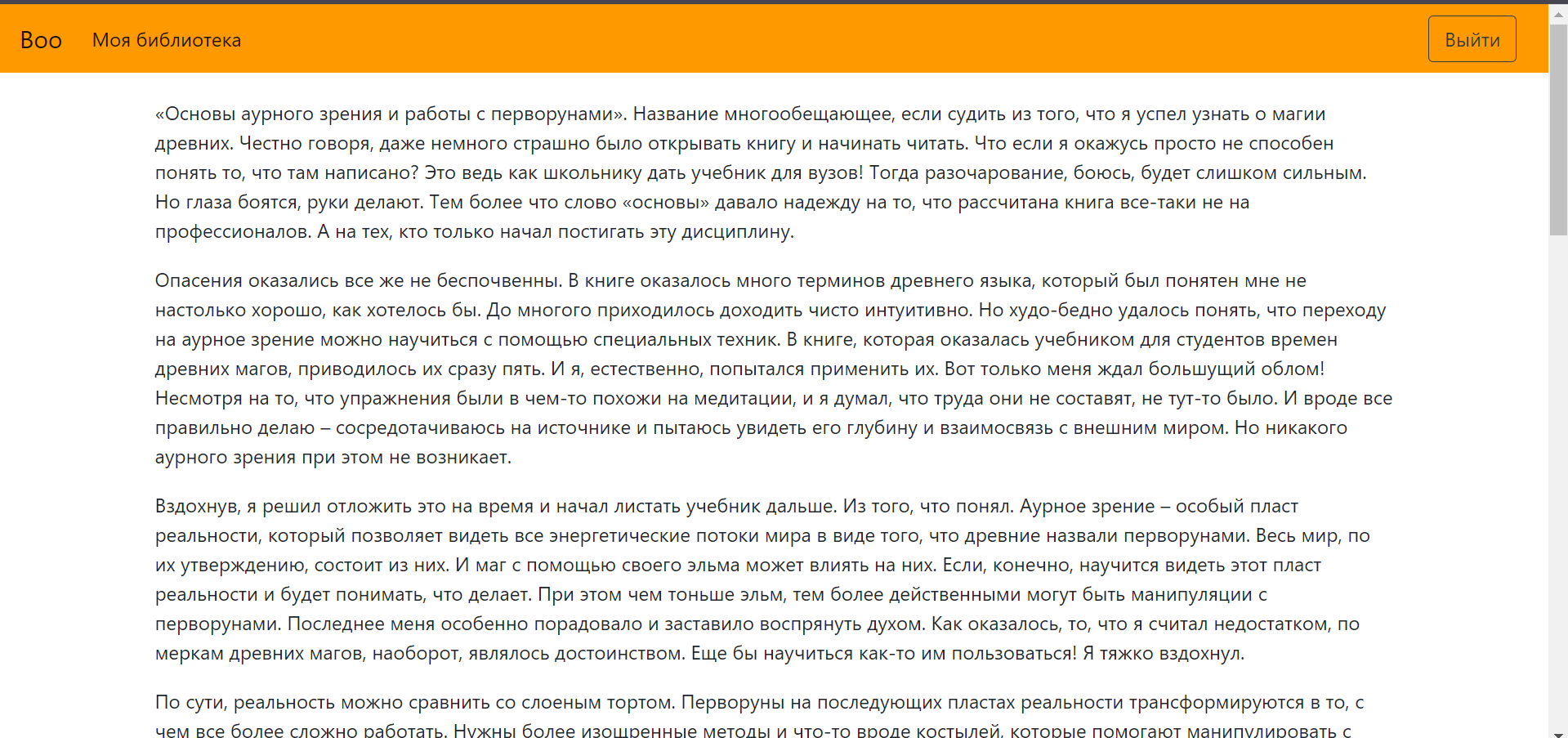
export default ReadPage

В итоге, получился следующий результат:

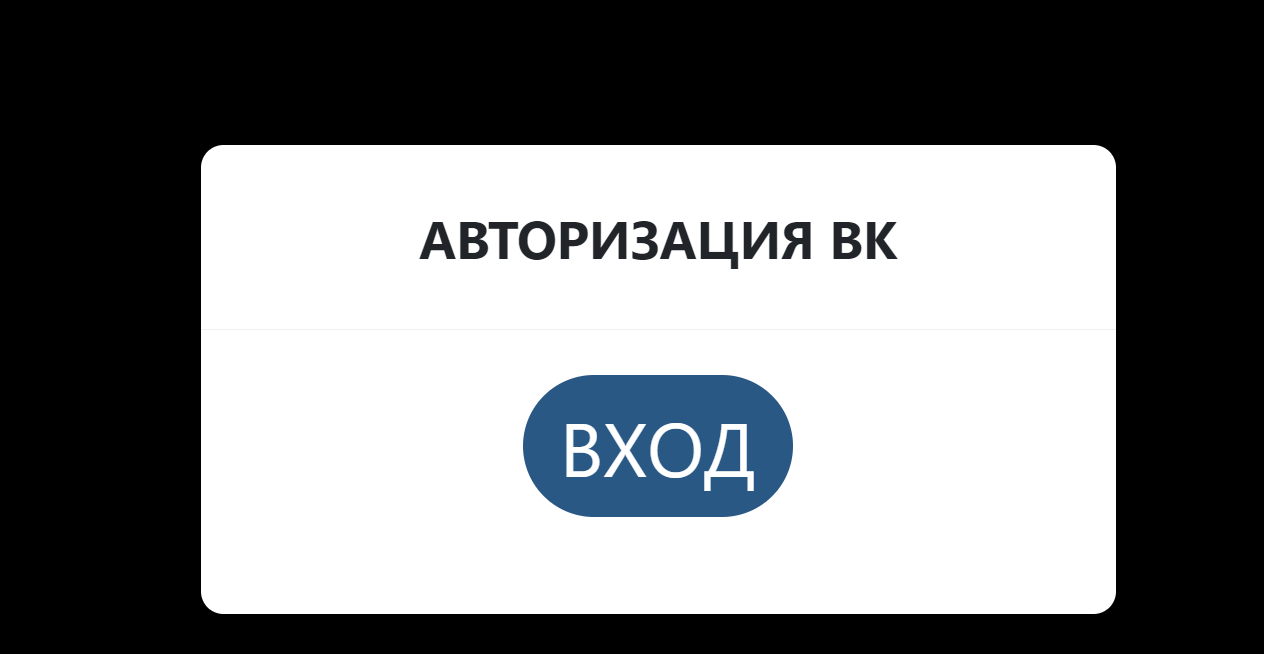
Главная страница



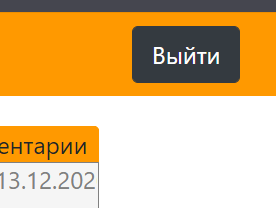
Чтение книги



Вход



Выход



**Вывод**

В результате выполнения данной работы были изучены основы работы протокола передачи гипертекста HTTP и языка CSS и основы разработки frontend-приложений, в частности, с использованием клиентских фреймворков (React). Также были рассмотрены основы работы docker и фреймворком Spring и Изучены основы взаимодействия web-приложения с базой данных и принципы работы отправки Ajax запросов.