

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**  
**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

Расчетно-графическое задание  
дисциплина: Технологии web-программирования  
тема: «Сайт, посвященный творчеству В. В. Набокова»

Выполнил: ст. гр. ПВ-42  
Богров Дмитрий Алексеевич  
Проверил: Картамышев С. В.

Белгород 2020

## Оглавление

Цель работы .....	3
Разработка макетов и верстка шаблонов web-приложения с помощью языков HTML и CSS .....	4
Клиентское программирование. Vue.js .....	15
Серверное программирование. Yii2 .....	23
Разработка и проектирование базы данных.....	27
REST API .....	34
Работа с HTTP запросами .....	37
Вывод.....	41

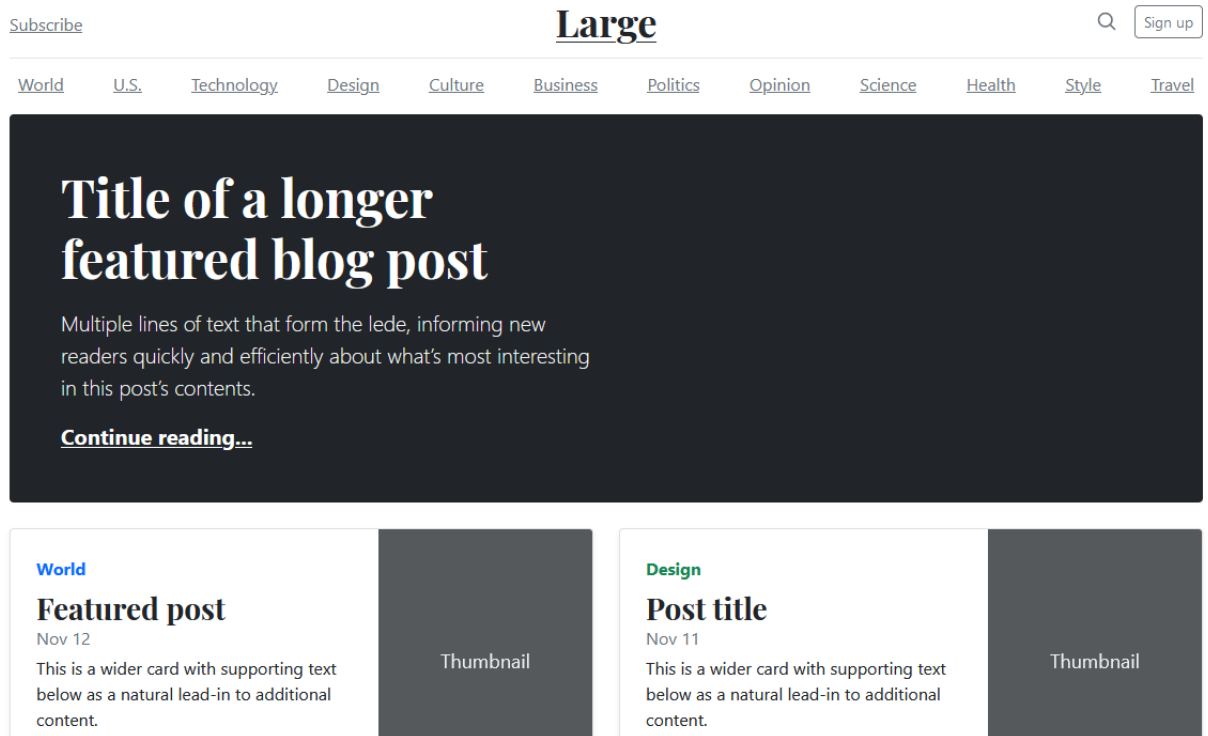
## **Цель работы**

Цель проекта: разработать веб-приложение по выбранной теме – сайт, посвященный творчеству В. В. Набокова (своего рода библиотека). Для реализации поставленной задачи необходимо разработать макеты и сверстать страницы, перенести верстку в компоненты фреймворка Vue.js, настроить бэкенд с docker и yii2, спроектировать и создать базу данных, реализовать REST API и связать бэкенд с фронтендом.

Функционал приложения прост. Пользователь может авторизоваться на сайте и читать материалы (для чтения авторизация необязательна). На главной странице, помимо навигационного бара (он есть на каждой странице), находится последняя добавленная на сайт статья. Кликая на навигационную панель, пользователь будет перемещаться по разделам сайта, которые в свою очередь делятся на категории. Каждая статья относится к определенной категории определенного раздела. Также пользователь может искать нужную статью через поиск.

# Разработка макетов и верстка шаблонов web-приложения с помощью языков HTML и CSS

В качестве темы проекта было решено разработать сайт, посвященный творчеству В. В. Набокова. Для реализации данной задумки прекрасно подходит шаблон из примеров Bootstrap:



Преобразовав этот шаблон, были созданы следующие страницы: главная, страница раздела (проза, поэзия и т.д.), страница статьи.

Скриншоты страниц:

[Войти](#)

# Владимир Набоков

[Проза](#)[Поэзия](#)[Интервью](#)[Письма](#)[Публицистика](#)[Энтомология](#)[Воспоминания](#)

## С чего начать

Краткий путеводитель по моему творчеству.

**Ознакомиться**

[Письма](#)

### О Бунине

1938

«Как он, Бунин, похож на старую, тощую черепаху...»

[Читать](#)

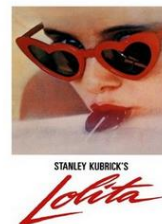
[Интервью](#)

### Премьера «Лолиты»

1962

Интервью перед премьерой экранизации Стэнли Кубрика.

[Читать](#)



## Последнее добавленное

[Войти](#)

# Владимир Набоков

[Проза](#)[Поэзия](#)[Интервью](#)[Письма](#)[Публицистика](#)[Энтомология](#)[Воспоминания](#)

## Публицистика

### Критические статьи

[Анкета о Прусте](#)

### Монографии

[Лекции по русской литературе](#)

[Лекции по зарубежной литературе](#)

[Лекции о драме](#)

### Эссе

[Кэмбридж](#)

[Что всякий должен знать?](#)

[Протест против вторжения в Финляндию](#)

[В начало](#)

[Войти](#)

# Владимир Набоков

[Проза](#)[Поэзия](#)[Интервью](#)[Письма](#)[Публицистика](#)[Энтомология](#)[Воспоминания](#)

## Анкета о Прусте

[Публицистика](#)

1930, "Числа", Париж, №1, с. 274

Редакция "Чисел" обратилась к ряду писателей с просьбой ответить на следующую анкету:

*Считаете ли Вы Пруста крупнейшим выразителем нашей эпохи?*

*Видите ли в современной жизни героев и атмосферу его эпохи?*

*Считаете ли, что особенности Прустовского мира, его метод наблюдения, его духовный опыт и его стиль должны оказать решающее влияние на мировую литературу ближайшего будущего, в частности на русскую?*

Мне кажется, что судить об этом невозможно: эпоха никогда не бывает "нашей". Мне неизвестно, в какую эпоху будущий историк нас ухlopает и какие найдет для нее приметы. К приметам, находимым современниками, я отношусь подозрительно.

Опять же, — мне трудно вообразить "современную" жизнь. Всякая страна живет по-своему, и всякий человек — по-своему. Но есть кое-что вечное. Изображение этого вечного только и ценно. Прустовские люди жили всегда и везде.

Литературное влияние — темная и смутная вещь. Можно себе, например, представить двух писателей, А и В, совершенно разных, но находящихся оба под некоторым, очень субъективным, влиянием Пруста; это влияние читателю С незаметно, так как каждый из трех (А, В и С) воспринял Пруста по-своему. Бывает, что писатель влияет косвенно, через другого, или же происходит какая-нибудь сложная смесь влияний и т. д. Предвидеть что-нибудь в этом направлении нельзя.

## *Index.html:*

```
<!doctype html>
<html lang="ru">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
  <meta name="description" content="">
  <meta name="author" content="Mark Otto, Jacob Thornton, and Bootstrap
contributors">
  <meta name="generator" content="Jekyll v4.1.1">
  <title>Блог Владимира Набокова</title>

  <link rel="canonical"
href="https://getbootstrap.com/docs/4.5/examples/blog/">

  <!-- Bootstrap core CSS -->
  <link href="/docs/4.5/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZ0nIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous">
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.cs
s" integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZ0nIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous">

  <style>
    .bd-placeholder-img {
      font-size: 1.125rem;
      text-align: middle;
      -webkit-user-select: none;
      -moz-user-select: none;
      -ms-user-select: none;
      user-select: none;
    }

    @media (min-width: 768px) {
      .bd-placeholder-img-lg {
        font-size: 3.5rem;
      }
    }
  </style>
  <!-- Custom styles for this template -->
  <link
href="https://fonts.googleapis.com/css?family=Playfair+Display:700,900"
rel="stylesheet">
  <!-- Custom styles for this template -->
  <link href="css/style.css" rel="stylesheet">
</head>
<body>
<div class="container">
  <header class="blog-header py-3">
    <div class="row flex-nowrap justify-content-between align-items-
center">
      <div class="col-4 pt-1">
        <a class="btn btn-sm btn-outline-secondary"
```

```

href="#">Войти</a>
</div>
<div class="col-4 text-center">
  <a class="blog-header-logo text-dark"
href="index.html">Владимир Набоков</a>
</div>
<div class="col-4 d-flex justify-content-end align-items-center">
  <a class="text-muted" href="#" aria-label="Search">
    <svg xmlns="http://www.w3.org/2000/svg" width="20"
height="20" fill="none" stroke="currentColor" stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" class="mx-3" role="img" viewBox="0 0 24 24"
focusable="false"><title>Search</title><circle cx="10.5" cy="10.5"
r="7.5"/><path d="M21 21l-5.2-5.2"/></svg>
  </a>
</div>
</div>
</header>

<div class="nav-scroller py-1 mb-2">
  <nav class="nav d-flex justify-content-between">
    <a class="p-2 text-muted" href="#">Проза</a>
    <a class="p-2 text-muted" href="#">Поэзия</a>
    <a class="p-2 text-muted" href="interviews.html">Интервью</a>
    <a class="p-2 text-muted" href="#">Письма</a>
    <a class="p-2 text-muted" href="publicism.html">Публицистика</a>
    <a class="p-2 text-muted" href="#">Энтомология</a>
    <a class="p-2 text-muted" href="#">Воспоминания</a>
  </nav>
</div>

<div class="jumbotron p-4 p-md-5 text-white rounded bg-dark">
  <div class="col-md-6 px-0">
    <h1 class="display-4 font-italic">С чего начать</h1>
    <p class="lead my-3">Краткий путеводитель по моему
творчеству.</p>
    <p class="lead mb-0"><a href="#" class="text-white font-weight-
bold">Ознакомиться</a></p>
  </div>
</div>

<div class="row mb-2">
  <div class="col-md-6">
    <div class="row no-gutters border rounded overflow-hidden flex-
md-row mb-4 shadow-sm h-md-250 position-relative">
      <div class="col p-4 d-flex flex-column position-static">
        <strong class="d-inline-block mb-2 text-
primary">Письма</strong>
        <h3 class="mb-0">О Бунине</h3>
        <div class="mb-1 text-muted">1938</div>
        <p class="card-text mb-auto">«Как он, Бунин, похож на
старую, тощую черепаху...»
        </p>
        <a href="#" class="stretched-link">Читать</a>
      </div>
      <div class="col-auto d-none d-lg-block">
        
    </div>
</div>
</div>
<div class="col-md-6">
    <div class="row no-gutters border rounded overflow-hidden flex-
md-row mb-4 shadow-sm h-md-250 position-relative">
        <div class="col p-4 d-flex flex-column position-static">
            <strong class="d-inline-block mb-2 text-
success">Интервью</strong>
            <h3 class="mb-0">Премьера «Лолиты»</h3>
            <div class="mb-1 text-muted">1962</div>
            <p class="mb-auto">Интервью перед премьерой экранизации
Стэнли Кубрика.</p>
            <a href="interview-pered-premieroi-loliti.html"
class="stretched-link">Читать</a>
        </div>
        <div class="col-auto d-none d-lg-block">
            
        </div>
    </div>
</div>
</div>
</div>
</div>
<main role="main" class="container">
    <div class="row">
        <div class="col-md-8 blog-main">
            <h3 class="pb-4 mb-4 font-italic border-bottom">
                Последнее добавленное
            </h3>

            <div class="blog-post">
                <h2 class="blog-post-title"><a class="blog-post-link"
href="anketa-o-pruste.html">Анкета о Прусте</a></h2>
                <p class="blog-post-meta"> <a
href="publicism.html">Публицистика</a> 1930, "Числа", Париж, №1, с. 274</p>

                <p>Редакция "Чисел" обратилась к ряду писателей с просьбой
ответить на следующую анкету:</p>

                <p><em>Считаете ли Вы Пруста крупнейшим выразителем нашей
эпохи?</em></p>
                <p><em>Видите ли в современной жизни героев и атмосферу его
эпопеи?</em></p>
                <p><em>Считаете ли, что особенности Прустовского мира, его
метод наблюдения, его духовный опыт и его стиль должны оказать решающее
влияние на мировую литературу ближайшего будущего, в частности на
русскую?</em></p>

                <p>Мне кажется, что судить об этом невозможно: эпоха никогда
не бывает "нашей". Мне неизвестно, в какую эпоху будущий историк нас ухлопает
и какие найдет для нее приметы. К приметам, находимым современниками, я
отношусь подозрительно.</p>

```



<p>Опять же, – мне трудно вообразить "современную" жизнь. Всякая страна живет по-своему, и всякий человек – по-своему. Но есть кое-что вечное. Изображение этого вечного только и ценно. Прустовские люди жили всегда и везде.</p>

<p>Литературное влияние – темная и смутная вещь. Можно себе, например, представить двух писателей, А и В, совершенно разных, но находящихся оба под некоторым, очень субъективным, влиянием Пруста; это влияние читателю С незаметно, так как каждый из трех (А, В и С) воспринял Пруста по-своему. Бывает, что писатель влияет косвенно, через другого, или же происходит какая-нибудь сложная смесь влияний и т. д. Предвидеть что-нибудь в этом направлении нельзя.</p>

</div><!-- /.blog-post -->

</div><!-- /.blog-main -->

<aside class="col-md-4 blog-sidebar">

<div class="p-4 mb-3 bg-light rounded">

<h4 class="font-italic">0 сайте</h4>

<p class="mb-0"><em>Посвящается моей жене</em>.</p>

</div>

<div class="p-4">

<h4 class="font-italic">Архив</h4>

<ol class="list-unstyled mb-0">

<li><a href="#">1970–1977</a></li>

<li><a href="#">1960–1969</a></li>

<li><a href="#">1950–1959</a></li>

<li><a href="#">1940–1949</a></li>

<li><a href="#">1930–1939</a></li>

<li><a href="#">1920–1929</a></li>

</ol>

</div>

<div class="p-4">

<h4 class="font-italic">Ссылки</h4>

<ol class="list-unstyled">

<li><a href="https://github.com/damncorwell/bstu-web">GitHub</a></li>

</ol>

</div>

</aside><!-- /.blog-sidebar -->

</div><!-- /.row -->

</main><!-- /.container -->

<footer class="blog-footer">



<p>

<a href="#">В начало</a>

</p>

</footer>

</body>

</html>

## *Publicism.html:*

```
<!doctype html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="Mark Otto, Jacob Thornton, and Bootstrap
contributors">
    <meta name="generator" content="Jekyll v4.1.1">
    <title>Блог Владимира Набокова</title>

    <link rel="canonical"
href="https://getbootstrap.com/docs/4.5/examples/blog/">

    <!-- Bootstrap core CSS -->
    <link href="/docs/4.5/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZ0nIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.cs
s" integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZ0nIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous">

    <style>
      .bd-placeholder-img {
        font-size: 1.125rem;
        text-align: middle;
        -webkit-user-select: none;
        -moz-user-select: none;
        -ms-user-select: none;
        user-select: none;
      }

      @media (min-width: 768px) {
        .bd-placeholder-img-lg {
          font-size: 3.5rem;
        }
      }
    </style>
    <!-- Custom styles for this template -->
    <link
href="https://fonts.googleapis.com/css?family=Playfair+Display:700,900"
rel="stylesheet">
    <!-- Custom styles for this template -->
    <link href="css/style.css" rel="stylesheet">
  </head>

  <body>
    <div class="container">
      <header class="blog-header py-3">
        <div class="row flex-nowrap justify-content-between align-items-
```

```

center">
    <div class="col-4 pt-1">
        <a class="btn btn-sm btn-outline-secondary"
href="#">Войти</a>
    </div>
    <div class="col-4 text-center">
        <a class="blog-header-logo text-dark"
href="index.html">Владимир Набоков</a>
    </div>
    <div class="col-4 d-flex justify-content-end align-items-
center">
        <a class="text-muted" href="#" aria-label="Search">
            <svg xmlns="http://www.w3.org/2000/svg" width="20"
height="20" fill="none" stroke="currentColor" stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" class="mx-3" role="img" viewBox="0 0 24 24"
focusable="false"><title>Search</title><circle cx="10.5" cy="10.5"
r="7.5"/><path d="M21 21l-5.2-5.2"/></svg>
        </a>
    </div>
</div>
</header>

<div class="nav-scroller py-1 mb-2">
    <nav class="nav d-flex justify-content-between">
        <a class="p-2 text-muted" href="#">Проза</a>
        <a class="p-2 text-muted" href="#">Поэзия</a>
        <a class="p-2 text-muted" href="interviews.html">Интервью</a>
        <a class="p-2 text-muted" href="#">Письма</a>
        <a class="p-2 text-muted"
href="publicism.html">Публицистика</a>
        <a class="p-2 text-muted" href="#">Энтомология</a>
        <a class="p-2 text-muted" href="#">Воспоминания</a>
    </nav>
</div>
</div>
<main role="main" class="container">
    <h2 class="section-title">Публицистика</h2>
    <hr>
    <h3>Критические статьи</h3>
    <ol class="list-unstyled">
        <li><a href="anketa-o-pruste.html">Анкета о Прусте</a> </li>
    </ol>
    <h3>Монографии</h3>
    <ol class="list-unstyled">
        <li><a href="">Лекции по русской литературе</a></li>
        <li><a href="">Лекции по зарубежной литературе</a></li>
        <li><a href="">Лекции о драме</a></li>
    </ol>
    <h3>Эссе</h3>
    <ol class="list-unstyled">
        <li><a href="">Кэмбридж</a></li>
        <li><a href="">Что всякий должен знать?</a></li>
        <li><a href="">Протест против вторжения в Финляндию</a></li>
    </ol>
</main>
<footer class="blog-footer">
    

```

```

    <p>
      <a href="#">В начало</a>
    </p>
  </footer>
</body>

</html>

```

*Anketa-o-pruste.html:*

```

<!doctype html>
<html lang="ru">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1,
  shrink-to-fit=no">
  <meta name="description" content="">
  <meta name="author" content="Mark Otto, Jacob Thornton, and Bootstrap
  contributors">
  <meta name="generator" content="Jekyll v4.1.1">
  <title>Блог Владимира Набокова</title>

  <link rel="canonical"
  href="https://getbootstrap.com/docs/4.5/examples/blog/">

  <!-- Bootstrap core CSS -->
  <link href="/docs/4.5/dist/css/bootstrap.min.css" rel="stylesheet"
  integrity="sha384-
  JcKb8q3iqJ61gNV9KGb8thSsNjpsL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
  crossorigin="anonymous">
  <link rel="stylesheet"
  href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.cs
  s" integrity="sha384-
  JcKb8q3iqJ61gNV9KGb8thSsNjpsL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
  crossorigin="anonymous">

  <style>
    .bd-placeholder-img {
      font-size: 1.125rem;
      text-align: middle;
      -webkit-user-select: none;
      -moz-user-select: none;
      -ms-user-select: none;
      user-select: none;
    }

    @media (min-width: 768px) {
      .bd-placeholder-img-lg {
        font-size: 3.5rem;
      }
    }
  </style>
  <!-- Custom styles for this template -->
  <link
  href="https://fonts.googleapis.com/css?family=Playfair+Display:700,900"
  rel="stylesheet">
  <!-- Custom styles for this template -->

```

```

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

<body>
<div class="container">
    <header class="blog-header py-3">
        <div class="row flex-nowrap justify-content-between align-items-center">
            <div class="col-4 pt-1">
                <a class="btn btn-sm btn-outline-secondary"
href="#">Войти</a>
            </div>
            <div class="col-4 text-center">
                <a class="blog-header-logo text-dark"
href="index.html">Владимир Набоков</a>
            </div>
            <div class="col-4 d-flex justify-content-end align-items-center">
                <a class="text-muted" href="#" aria-label="Search">
                    <svg xmlns="http://www.w3.org/2000/svg" width="20"
height="20" fill="none" stroke="currentColor" stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" class="mx-3" role="img" viewBox="0 0 24 24"
focusable="false"><title>Search</title><circle cx="10.5" cy="10.5"
r="7.5"/><path d="M21 21l-5.2-5.2"/></svg>
                </a>
            </div>
        </div>
    </header>

    <div class="nav-scroller py-1 mb-2">
        <nav class="nav d-flex justify-content-between">
            <a class="p-2 text-muted" href="#">Проза</a>
            <a class="p-2 text-muted" href="#">Поэзия</a>
            <a class="p-2 text-muted" href="interviews.html">Интервью</a>
            <a class="p-2 text-muted" href="#">Письма</a>
            <a class="p-2 text-muted" href="publicism.html">Публицистика</a>
            <a class="p-2 text-muted" href="#">Энтомология</a>
            <a class="p-2 text-muted" href="#">Воспоминания</a>
        </nav>
    </div>
</div>
<main role="main" class="container">
    <div class="blog-post">
        <h2 class="post-title">Анкета о Прусте</h2>
        <p class="post-meta"> <a href="publicism.html">Публицистика</a>
<br/>1930, "Числа", Париж, №1, с. 274</p>
        <div class="post-text">
            <p>Редакция "Чисел" обратилась к ряду писателей с просьбой
ответить на следующую анкету:</p>

            <p><em>Считаете ли Вы Пруста крупнейшим выразителем нашей
эпохи?</em></p>
            <p><em>Видите ли в современной жизни героев и атмосферу его
эпопей?</em></p>
            <p><em>Считаете ли, что особенности Прустовского мира, его метод
наблюдения, его духовный опыт и его стиль должны оказать решающее влияние на
мировую литературу ближайшего будущего, в частности на русскую?</em></p>

```

<p>Мне кажется, что судить об этом невозможно: эпоха никогда не бывает "нашей". Мне неизвестно, в какую эпоху будущий историк нас ухлопает и какие найдет для нее приметы. К приметам, находимым современниками, я отношусь подозрительно.</p>

<p>Опять же, – мне трудно вообразить "современную" жизнь. Всякая страна живет по-своему, и всякий человек – по-своему. Но есть кое-что вечное. Изображение этого вечного только и ценно. Прустовские люди жили всегда и везде.</p>

<p>Литературное влияние – темная и смутная вещь. Можно себе, например, представить двух писателей, А и В, совершенно разных, но находящихся оба под некоторым, очень субъективным, влиянием Пруста; это влияние читателю С незаметно, так как каждый из трех (А, В и С) воспринял Пруста по-своему. Бывает, что писатель влияет косвенно, через другого, или же происходит какая-нибудь сложная смесь влияний и т. д. Предвидеть что-нибудь в этом направлении нельзя.</p>

</div>

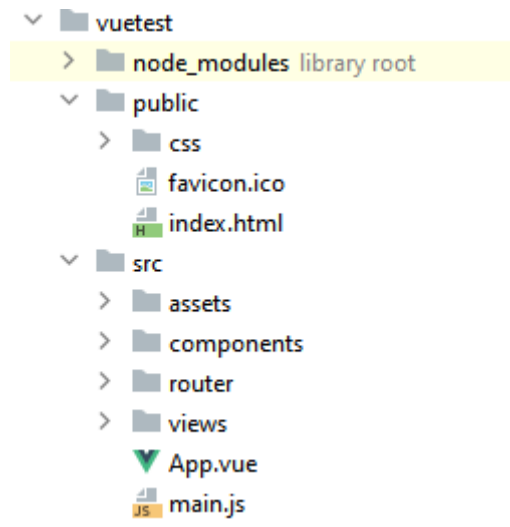
```
</div><!-- /.blog-post -->
</main>
<footer class="blog-footer">
  
  <p>
    <a href="#">В начало</a>
  </p>
</footer>
</body>

</html>
```

## Клиентское программирование. Vue.js

На данном этапе мы познакомились с основами разработки на языке JavaScript, с основами разработки frontend приложения, в частности с использованием клиентских фреймворков (Vue.js). Научились переносить вёрстку в отдельные компоненты.

Перед началом работы необходимо было развернуть в проекте базовое приложение на фреймворке Vue.js – для этого использовался консольная команда **npm install vue@cli**. В результате последующих требуемых действий для развертки приложения была получена следующая структура приложения:



После этого можно приступить к переносу верстки, выполненной в предыдущей лабораторной работе.

Выделим отдельные блоки в компоненты.

*Aside.vue:*

```
<template>
  <aside class="col-md-4 blog-sidebar">
    <div class="p-4 mb-3 bg-light rounded">
      <h4 class="font-italic">0 сайте</h4>
      <p class="mb-0"><em>Посвящается моей жене</em>.</p>
    </div>

    <div class="p-4">
      <h4 class="font-italic">Архив</h4>

      <div class="p-4">
        <h4 class="font-italic">Ссылки</h4>
        <ol class="list-unstyled">
          <li><a href="https://github.com/damncorwell/nabokov-
blog">GitHub</a></li>
        </ol>
      </div>
    </aside><!-- /.blog-sidebar -->
  </template>
```

```

<script>
export default {
  name: "Aside"
}
</script>

```

*Cards.vue:*

```

<template>
  <div class="row mb-2">
    <div class="col-md-6">
      <div class="row no-gutters border rounded overflow-hidden flex-md-row
mb-4 shadow-sm h-md-250 position-relative">
        <div class="col p-4 d-flex flex-column position-static">
          <strong class="d-inline-block mb-2 text-primary">Письма</strong>
          <h3 class="mb-0">О Буныне</h3>
          <div class="mb-1 text-muted">1938</div>
          <p class="card-text mb-auto">«Как он, Бунин, похож на старую, тощую
черепаху...»
          </p>
          <a href="#" class="stretched-link">Читать</a>
        </div>
        <div class="col-auto d-none d-lg-block">
          
        </div>
      </div>
    </div>
    <div class="col-md-6">
      <div class="row no-gutters border rounded overflow-hidden flex-md-row
mb-4 shadow-sm h-md-250 position-relative">
        <div class="col p-4 d-flex flex-column position-static">
          <strong class="d-inline-block mb-2 text-success">Интервью</strong>
          <h3 class="mb-0">Премьера «Лолиты»</h3>
          <div class="mb-1 text-muted">1962</div>
          <p class="mb-auto">Интервью перед премьерой экранизации Стэнли
Кубрика.</p>
          <router-link v-bind:to="{name: 'Interview pered premieroi
Loliti'}">Читать</router-link>
        </div>
        <div class="col-auto d-none d-lg-block">
          
        </div>
      </div>
    </div>
  </div>
</template>

<script>
export default {
  name: "Cards"

```



```

}
</script>

```

*Jumbotron.vue:*

```

<template>
  <div class="jumbotron p-4 p-md-5 text-white rounded bg-dark">
    <div class="col-md-6 px-0">
      <h1 class="display-4 font-italic">С чего начать</h1>
      <p class="lead my-3">Краткий путеводитель по творчеству.</p>
      <p class="lead mb-0"><a href="#" class="text-white font-weight-
bold">Ознакомиться</a></p>
    </div>
  </div>
</template>

<script>
export default {
  name: "Jumbotron"
}
</script>

```

*LastAdded.vue:*

```

<template>
  <div class="col-md-8 blog-main">
    <h3 class="pb-4 mb-4 font-italic border-bottom">
      Последнее добавленное
    </h3>

    <div class="blog-post">
      <h2 class="blog-post-title">
        <router-link class="blog-post-link"
          v-bind:to="{name: 'Post', params: {url: post.url}}">
          Анкета о Прусте
        </router-link>
      </h2>
      <div class="blog-post-meta">
        <router-link v-if="post.section === 'publicism'"
          v-bind:to="{name: 'Publicism'}">Публицистика
        </router-link>
        {{post.meta}}
      </div>
      <div class="post-text">
        {{post.text}}
      </div>
    </div><!-- /.blog-post -->

  </div><!-- /.blog-main -->
</template>

<script>
import PostList from "@components/PostsList"

export default {
  name: "LastAdded",
  components: {PostList},
  data() {
    return {

```

```

    post: {
      title: 'Анкета о Прусте',
      url: 'anketa-o-pruste',
      section: 'publicism',
      category: 'critical articles',
      meta: 1930,
      text: 'Текст поста "Анкета о Прусте"'
    }
  }
}
}
</script>

```

*Post.vue:*

```

<template>
  <div class="blog-post">
    <h2 class="post-title">{{ post.title }}</h2>
    <p class="post-meta"> </p>
    <div class="post-text">{{ post.text }}</div>

  </div><!-- /.blog-post -->
</template>

<script>
export default {
  name: "Post",
  props: ['post']
}
</script>

```

*PostsList.vue:*

```

<template>
  <div>
    <span>
      <router-link v-bind:to="{name: 'Post', params: {url: post.url}}">
        {{post.title}} ({{post.meta}})
      </router-link>
    </span>
  </div>

</template>

<script>
import Post from "@components/Post"
export default {
  name: "PostsList",
  components: {Post},
  props: ['post'],
  data() {
    return {
      posts: [
        {
          title: 'Анкета о Прусте',
          url: 'anketa-o-pruste',
          section: 'publicism',
          category: 'critical articles',

```

```

    meta: 1930,
    text: 'Текст поста "Анкета о Прусте"'
  },
  {
    title: 'Лекции о драме',
    url: 'lekcii-o-drame',
    section: 'publicism',
    category: 'monograph',
    meta: null,
    text: ''
  },
  {
    title: 'Что каждый должен знать?',
    url: 'cto-kazhdiy-dolzhen-znat',
    section: 'publicism',
    category: 'essay',
    meta: 1931,
    text: 'Текст поста "Что каждый должен знать?"'
  },
  {
    title: 'Интервью перед премьерой фильма «Лолита»',
    url: 'interview-pered-premieroi-loliti',
    section: 'interview',
    category: 'interview ru',
    meta: 1962,
    text: 'Текст поста "Интервью перед премьерой фильма «Лолита»"'
  }
]
}
}
}
}
</script>

```

Страницы в проектах Vue принято называть views. Перечислим их:

*Home.vue:*

```

<template>
  <div>
    <div class="home">
      <Jumbotron></Jumbotron>
      <Cards></Cards>
    </div>
    <div class="row">
      <LastAdded></LastAdded>
      <Aside></Aside>
    </div>
  </div>
</template>

<script>
import Product from "@components/products/Product";
import Jumbotron from "@components/Jumbotron";
import Cards from "@components/Cards";
import LastAdded from "@components/LastAdded";
import Aside from "@components/Aside";
export default {
  name: 'Home',

```

```

    components: {Aside, LastAdded, Cards, Jumbotron},
  }
</script>

```

*Interviews.vue:*

```

<template>
  <div>
    <h2 class="section-title">Интервью</h2>
    <hr>
    <h3>Интервью на русском и в переводе на русский</h3>
    <ol class="list-unstyled">
      <li v-for="post in categoryRu">
        <PostsList v-bind:post="post"></PostsList>
      </li>
    </ol>
    <h3>Интервью на английском</h3>
    <ol class="list-unstyled">
      <li v-for="post in categoryEn">
        <PostsList v-bind:post="post"></PostsList>
      </li>
    </ol>
  </div>
</template>

<script>
import PostsList from "@/components/PostsList";
export default {
  name: "Interviews",
  components: {PostsList},
  data() {
    return {
      list: PostsList
    }
  },
  computed: {
    categoryRu: function () {
      return this.list.data().posts.filter(post => post.category ===
'interview ru')
    },
    categoryEn: function () {
      return this.list.data().posts.filter(post => post.category ===
'interview en')
    }
  }
}
</script>

```

*Post.vue:*

```

<template>
  <div class="blog-post">
    <h2 class="post-title">{{post.title}}</h2>
    <p class="post-meta">
      <router-link v-if="post.section === 'publicism'"
        v-bind:to="{name: 'Publicism'}">Публицистика <br>
    </router-link>
  </div>

```

```

        <router-link class="text-success"
                    v-if="post.section === 'interview'"
                    v-bind:to="{name: 'Interviews'}">Интервью <br>
        </router-link>
        {{post.meta}}
    </p>
    <div class="post-text">
        {{post.text}}
    </div>
</div>
</template>

<script>
export default {
  name: "Post",
  data() {
    return {
      post: {}
    }
  },
  created() {
    const url =this.$route.params.url
    // send ajax request
    this.post = {
      title: 'Анкета о Прусте',
      url: url,
      section: 'publicism',
      category: 'critical articles',
      meta: 1930,
      text: 'Текст поста "Анкета о Прусте"'
    }
  }
}
</script>

```

*Publicism.vue:*

```

<template>
  <div>
    <h2 class="section-title">Публицистика</h2>
    <hr>
    <h3>Критические статьи</h3>
    <ol class="list-unstyled">
      <li v-for="post in categoryCritArticles">
        <PostsList v-bind:post="post"></PostsList>
      </li>
    </ol>
    <h3>Монографии</h3>
    <ol class="list-unstyled">
      <li v-for="post in categoryMonograph">
        <PostsList v-bind:post="post"></PostsList>
      </li>
    </ol>
    <h3>Эссе</h3>
    <ol class="list-unstyled">
      <li v-for="post in categoryEssay">
        <PostsList v-bind:post="post"></PostsList>
      </li>
    </ol>
  </div>
</template>

```

```

        </li>
    </ol>
</div>
</template>

<script>
import PostsList from "@components/PostsList";
export default {
  name: "Publicism",
  components: {PostsList},
  data() {
    return {
      list: PostsList
    }
  },
  computed: {
    categoryCritArticles: function() {
      return this.list.data().posts.filter(post => post.category ===
'critical articles')
    },
    categoryMonograph: function() {
      return this.list.data().posts.filter(post => post.category ===
'monograph')
    },
    categoryEssay: function() {
      return this.list.data().posts.filter(post => post.category === 'essay')
    }
  }
}
</script>

```

## Серверное программирование. Yii2

На данном этапе мы ознакомились с основами backend разработки web-приложений, научились писать скрипты на языке PHP. Познакомились с основами работы docker. Познакомились с фреймворком Yii2 и научились разворачивать проект, производить его настройку. Научились работать с API в приложении Postman.

Базовое приложение Yii2 разворачивается с помощью php-композера командой `composer create-project --prefer-dist yiisoft/yii2-app-basic basic`. В результате выполнения команды были получены следующие директории:



Перед запуском сервера необходимо настроить файл конфигурации **docker-compose.yml**:

```
version: '3.1'
services:
  php:
    image: yiisoftware/yii2-php:7.3-apache
    volumes:
      - ~/.composer-docker/cache:/root/.composer/cache:delegated
      - ./:/app:delegated
    ports:
      - '1199:80'

  mysql:
    image: mysql
    command: --default-authentication-plugin=mysql_native_password --
character-set-server=utf8 --collation-server=utf8_general_ci
    environment:
      - MYSQL_ROOT_PASSWORD=verysecret
      - MYSQL_DATABASE=bstu
      - MYSQL_USER=bstu
      - MYSQL_PASSWORD=bstu321!
```

Для запуска сервера используется команда **docker-compose up -d**.

После проделанных операций можно перейти к добавлению контроллеров со статическими данными (перед этим создав контроллер для разрешения cors-запросов).

*ApiController.php:*

```
<?php

namespace app\modules\v1\controllers;

use yii\filters\auth\CompositeAuth;
use yii\filters\ContentNegotiator;
use yii\filters\Cors;
use yii\filters\RateLimiter;
use yii\filters\VerbFilter;
use yii\rest\Controller;
use yii\web\Response;

class ApiController extends Controller
{
    public function behaviors()
    {
        return [
            'contentNegotiator' => [
                'class' => ContentNegotiator::className(),
                'formats' => [
                    'application/json' => Response::FORMAT_JSON,
                ],
            ],
        ];
    }
}
```



```

],
    'verbFilter' => [
        'class' => VerbFilter::className(),
        'actions' => $this->verbs(),
    ],
    'authenticator' => [
        'class' => CompositeAuth::className(),
    ],
    'rateLimiter' => [
        'class' => RateLimiter::className(),
    ],
    'corsFilter' => [
        'class' => Cors::class
    ]
];
}
}

```

*PageController.php:*

```

<?php

namespace app\modules\v1\controllers;

class PageController extends ApiController
{
    public function actionPublicism ()
    {
        return [
            'id' => 5,
            'name' => 'Публицистика',
            'text' => 'Текст страницы "Публицистика"'
        ];
    }

    public function actionInterviews ()
    {
        return [
            'id' => 3,
            'name' => 'Интервью',
            'text' => 'Текст страницы "Интервью"'
        ];
    }
}

```

# Скриншоты работы контроллеров в Postman:

Untitled Request

BUILD

POST

http://localhost:1199/api/v1/page/publicism

Send

Save

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

Co

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Ed
	Key	Value	Description		

Body

Cookies

Headers (10)

Test Results

⌚

Status: 200 OK

Time: 50 ms

Size: 456 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

{

"id": 5,

"name": "Публицистика",

"text": "Текст страницы \"Публицистика\""

}

POST

http://localhost:1199/api/v1/page/interviews

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Query Params

	KEY	VALUE	DESCRIPTION	..
	Key	Value	Description	

Body

Cookies

Headers (10)

Test Results

⌚

Status: 200 OK

Time: 19 ms

Size: 438 B

Save

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

{

"id": 3,

"name": "Интервью",

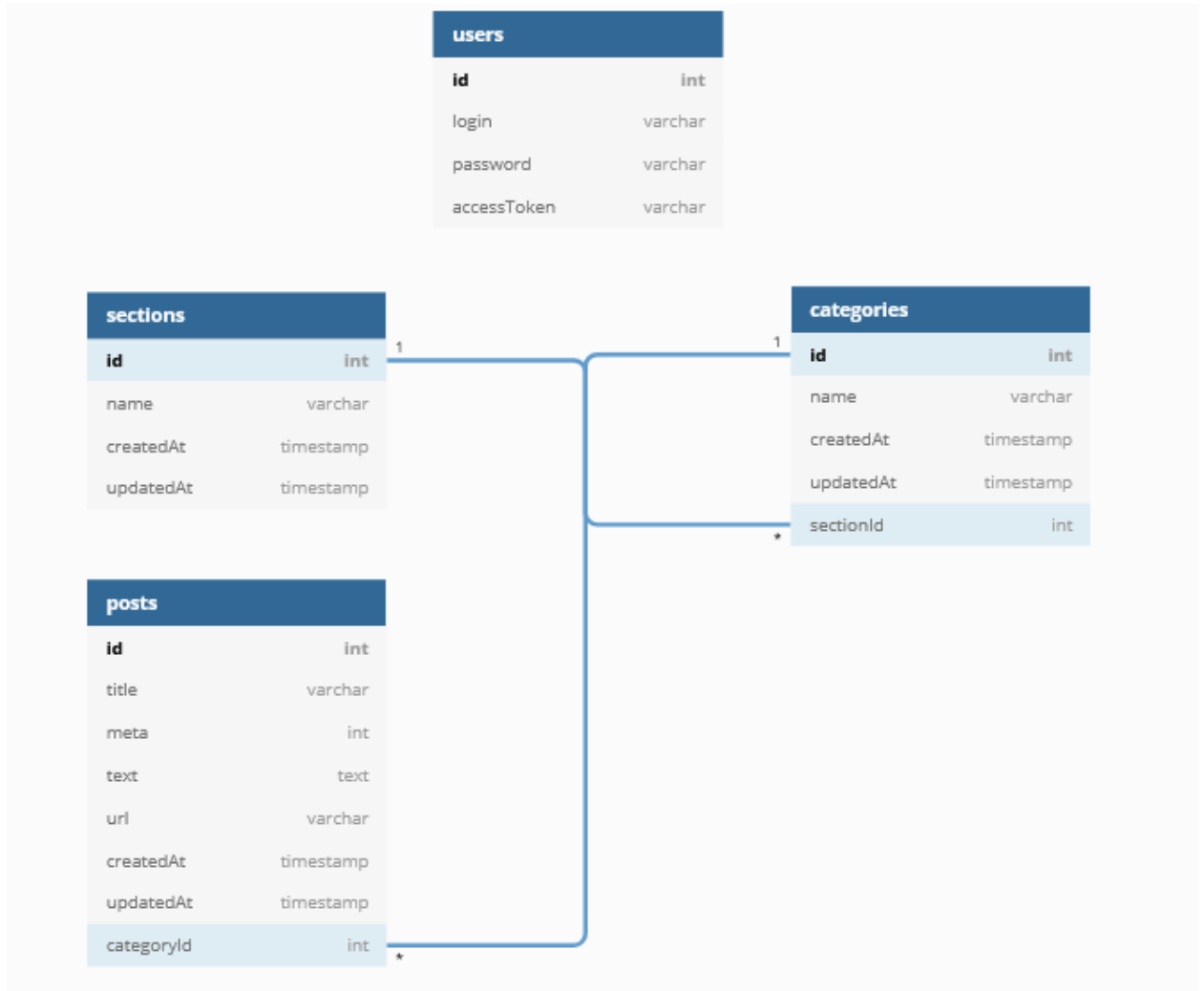
"text": "Текст страницы \"Интервью\""

}

## Разработка и проектирование базы данных

На данном этапе были изучены основы взаимодействия web-приложения с базой данных, спроектирована база данных для хранения информации приложения (страницы, пользователи и т.п.).

Схема базы данных:



Для создания таблиц в базе данных использовались миграции.

*Create\_posts\_table.php:*

```
<?php
use yii\db\Migration;

/**
 * Handles the creation of table `{{%posts}}`.
 */
class m201120_154205_create_posts_table extends Migration
{
    /**
     * {@inheritdoc}
     */
    public function safeUp()
```

```

    {
        $this->createTable('{{%posts}}', [
            'id' => $this->primaryKey(),
            'title' => $this->string(128)->notNull()->comment('Заголовок'),
            'meta' => $this->integer()->comment('Дата публикации'),
            'text' => $this->text()->comment('Текст статьи'),
            'url' => $this->string()->notNull()->comment('URL'),
            'createdAt' => $this->dateTime()->notNull()->comment('Дата
создания'),
            'updatedAt' => $this->dateTime()->comment('Дата изменения')
        ]);
    }

    /**
     * {@inheritdoc}
     */
    public function safeDown()
    {
        $this->dropTable('{{%posts}}');
    }
}

```

*Create\_categories\_table.php:*

```
<?php
```

```
use yii\db\Migration;
```

```

/**
 * Handles the creation of table `{{%categories}}`.
 */
class m201129_080808_create_categories_table extends Migration
{
    /**
     * {@inheritdoc}
     */
    public function safeUp()
    {
        $this->createTable('{{%categories}}', [
            'id' => $this->primaryKey(),
            'name' => $this->string(64)->notNull()->comment('Наименование'),
            'createdAt' => $this->dateTime()->notNull()->comment('Дата
создания'),
            'updatedAt' => $this->dateTime()->comment('Дата изменения')
        ]);

        $this->addColumn('{{%posts}}', 'categoryID',
            $this->integer()->comment('Категория'));
        $this->addForeignKey('fk_posts_categoryId', '{{%posts}}',
            'categoryID',
            '{{%categories}}', 'id', 'SET NULL');
    }

    /**
     * {@inheritdoc}
     */
    public function safeDown()
    {
        $this->dropForeignKey('fk_posts_categoryId', '{{%posts}}');
    }
}

```

```

        $this->dropColumn('{{%posts}}', 'categoryID');
        $this->dropTable('{{%categories}}');
    }
}

```

*Create\_sections\_table.php:*

```

<?php

use yii\db\Migration;

/**
 * Handles the creation of table '{{%sections}}'.
 */
class m201129_081554_create_sections_table extends Migration
{
    /**
     * {@inheritdoc}
     */
    public function safeUp()
    {
        $this->createTable('{{%sections}}', [
            'id' => $this->primaryKey(),
            'name' => $this->string(64)->notNull()->comment('Наименование'),
            'createdAt' => $this->dateTime()->notNull()->comment('Дата
создания'),
            'updatedAt' => $this->dateTime()->comment('Дата изменения')
        ]);

        $this->addColumn('{{%categories}}', 'sectionId',
            $this->integer()->comment('Раздел'));
        $this->addForeignKey('fk_categories_sectionId', '{{%categories}}',
            'sectionId',
            '{{%sections}}', 'id', 'SET NULL');
    }

    /**
     * {@inheritdoc}
     */
    public function safeDown()
    {
        $this->dropTable('{{%sections}}');
    }
}

```

*Create\_users\_table.php:*

```

<?php

use yii\db\Migration;

/**
 * Handles the creation of table '{{%users}}'.
 */
class m201220_183408_create_users_table extends Migration
{
    /**
     * {@inheritdoc}
     */
    public function safeUp()
    {

```

```

{
    $this->createTable('{{%users}}', [
        'id' => $this->primaryKey(),
        'login' => $this->string(512)->notNull()->comment('Логин'),
        'password' => $this->string(512)->notNull()->comment('Пароль'),
        'accessToken' => $this->string(512)->comment('Токен
пользователя')
    ]);
}

/**
 * {@inheritdoc}
 */
public function safeDown()
{
    $this->dropTable('{{%users}}');
}
}

```

В результате база данных стала выглядеть следующим образом:

**Adminer 4.7.7 4.7.8**  
 DB: bstu  
 SQL-запрос    Импорт  
 Экспорт    Создать таблицу  
 выбрать **categories**  
 выбрать migration  
 выбрать posts  
 выбрать sections  
 выбрать users

### Таблица: categories

[Выбрать](#)
[Показать структуру](#)
[Изменить таблицу](#)
[Новая запись](#)

поле	Тип	Комментарий
<b>id</b>	int	Автоматическое приращение
<b>name</b>	varchar(64)	Наименование
<b>createdAt</b>	datetime	Дата создания
<b>updatedAt</b>	datetime NULL	Дата изменения
<b>sectionId</b>	int NULL	Раздел

#### Индексы

<b>PRIMARY</b>	id
<b>INDEX</b>	sectionId

[Изменить индексы](#)  

#### Внешние ключи

Источник	Цель	При стирании	При обновлении	
<b>sectionId</b>	sections(id)	SET NULL	RESTRICT	<a href="#">Изменить</a>

[Добавить внешний ключ](#)

Adminer 4.7.7 4.7.8

DB: bstu

SQL-запрос

Импорт

Экспорт

Создать таблицу

выбрать categories

выбрать migration

выбрать **posts**

выбрать sections

выбрать users

Таблица: posts

Выбрать

Показать структуру

Изменить таблицу

Новая запись

поле	Тип	Комментарий
<b>id</b>	int	Автоматическое приращение
<b>title</b>	varchar(128)	Заголовок
<b>meta</b>	int NULL	Дата публикации
<b>text</b>	text NULL	Текст статьи
<b>url</b>	varchar(255)	URL
<b>createdAt</b>	datetime	Дата создания
<b>updatedAt</b>	datetime NULL	Дата изменения
<b>categoryID</b>	int NULL	Категория

#### Индексы

<b>PRIMARY</b>	<i>id</i>
<b>INDEX</b>	<i>categoryID</i>

[Изменить индексы](#)

#### Внешние ключи

Источник	Цель	При стирании	При обновлении	
<b>categoryID</b>	categories( <i>id</i> )	SET NULL	RESTRICT	<a href="#">Изменить</a>

[Добавить внешний ключ](#)

Adminer 4.7.7 4.7.8

DB: bstu

SQL-запрос

Импорт

Экспорт

Создать таблицу

выбрать categories

выбрать migration

выбрать posts

выбрать **sections**

выбрать users

Таблица: sections

Выбрать

Показать структуру

Изменить таблицу

Новая запись

поле	Тип	Комментарий
<b>id</b>	int	Автоматическое приращение
<b>name</b>	varchar(64)	Наименование
<b>createdAt</b>	datetime	Дата создания
<b>updatedAt</b>	datetime NULL	Дата изменения

#### Индексы

<b>PRIMARY</b>	<i>id</i>
----------------	-----------

[Изменить индексы](#)

Adminer 4.7.7 4.7.8

DB: bstu

SQL-запрос

Импорт

Экспорт

Создать таблицу

выбрать categories

выбрать migration

выбрать posts

выбрать sections

выбрать **users**

Таблица: users

Выбрать

Показать структуру

Изменить таблицу

Новая запись

поле	Тип	Комментарий
<b>id</b>	int	Автоматическое приращение
<b>login</b>	varchar(512)	Логин
<b>password</b>	varchar(512)	Пароль
<b>accessToken</b>	varchar(512) NULL	Токен пользователя

#### Индексы

<b>PRIMARY</b>	<i>id</i>
----------------	-----------

[Изменить индексы](#)

С помощью генератора Yii/gii были сгенерированы модели для таблиц базы данных. Например, модель Категории:

```
<?php
```

```
namespace app\modules\v1\models;
```

```
use Yii;
```

```

/**
 * This is the model class for table "categories".
 *
 * @property int $id
 * @property string $name Наименование
 * @property string $createdAt Дата создания
 * @property string|null $updatedAt Дата изменения
 *
 * @property Post[] $posts
 * @property Section $section
 */
class Category extends BaseModel
{
    /**
     * {@inheritdoc}
     */
    public static function tableName()
    {
        return 'categories';
    }

    /**
     * {@inheritdoc}
     */
    public function rules()
    {
        return [
            [['name'], 'required'],
            [['createdAt', 'updatedAt'], 'safe'],
            [['name'], 'string', 'max' => 64],
        ];
    }

    /**
     * {@inheritdoc}
     */
    public function attributeLabels()
    {
        return [
            'id' => 'ID',
            'name' => 'Наименование',
            'createdAt' => 'Дата создания',
            'updatedAt' => 'Дата изменения',
        ];
    }

    /**
     * {@inheritdoc}
     */
    public function toArray(array $fields = [], array $expand = [],
$recursive = true)
    {
        return [
            'id' => $this->id,
            'name' => $this->name,
            'createdAt' => $this->createdAt,
            'section' => $this->section
        ];
    }
}

```



```
    ];  
}  
  
public function getSection()  
{  
    return $this->hasOne(Section::class, ['id' => 'sectionId']);  
}  
  
/**  
 * @return \yii\db\ActiveQuery  
 */  
public function getPosts()  
{  
    return $this->hasMany(Post::class, ['categoryId' => 'id']);  
}  
}
```

## REST API

На данном этапе были изучены основы разработки API для web-приложений, а также разработан REST API для своего проекта.

Базовый URL: localhost:1199/api/v1

1. Получение последней добавленной статьи

URL: /post/lastadded

Тип запроса: GET

Параметры: нет

2. Получение конкретной статьи

URL: /post/info

Тип запроса: GET

Параметры:

Название	Описание	Тип	Обязательный?
url	URL нужной статьи	String	Да

3. Получение всех статей сайта

URL: /post/list

Тип запроса: GET

Параметры:

Название	Описание	Тип	Обязательный?
id	id нужной категории	Int	Нет

4. Авторизация пользователя

URL: /user/login

Тип запроса: POST

Параметры:

Название	Описание	Тип	Обязательный?
username	Логин пользователя	String	Да
password	Пароль пользователя	String	Да

5. Регистрация пользователя

URL: /user/signup

Тип запроса: POST

Параметры:

Название	Описание	Тип	Обязательный?
username	Логин пользователя	String	Да
password	Пароль пользователя	String	Да

Adminer 4.7.7 4.7.8

DB:

[SQL-запрос](#) [Импорт](#)  
[Экспорт](#) [Создать таблицу](#)

[выбрать categories](#)  
[выбрать migration](#)  
[выбрать posts](#)  
[выбрать sections](#)  
**[выбрать users](#)**

Выбрать: users

[Выбрать](#) [Показать структуру](#) [Изменить таблицу](#) [Новая запись](#)

[Выбрать](#)

[Поиск](#)

[Сортировать](#)

Лимит

Длина текста

Действие [Выбрать](#)

SELECT \* FROM `users` LIMIT 50 (0.000 s) [Редактировать](#)

<input type="checkbox"/> Изменить	id	login	password	accessToken
<input type="checkbox"/> редактировать	1	user@ya.ru	123	11c38e4b0ff1b10bfd19918ea9e9d832

POST  [Send](#)

[Params](#) [Authorization](#) [Headers \(8\)](#) **[Body](#)** [Pre-request Script](#) [Tests](#) [Settings](#)

☒ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL [JSON](#)

```
1 {
2   "username": "user@ya.ru",
3   "password": 123
4 }
```

[Body](#) [Cookies](#) [Headers \(10\)](#) [Test Results](#) [Status: 200 OK](#) [Time: 115 ms](#) [Size: 443 B](#) [Save](#)

[Pretty](#) [Raw](#) [Preview](#) [Visualize](#) [JSON](#)

```
1 {
2   "id": 1,
3   "login": "user@ya.ru",
4   "password": "123",
5   "accessToken": "11c38e4b0ff1b10bfd19918ea9e9d832"
6 }
```

POST

http://localhost:1199/api/v1/user/login

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON

```
1 {
2   "username": "uwer@ya.ru",
3   "password": 123
4 }
```

Body

Cookies

Headers (10)

Test Results

Status: 401 Unauthorized

Time: 18 ms

Size: 456 B

Save

PrettyRawPreviewVisualizeJSON

```
1 {
2   "statusText": "Имя пользователя или пароль введены неверно."
3 }
```

## Работа с HTTP запросами

На данном этапе были изучены принципы работы отправки Ajax запросов. Изучена работа отправки синхронных и асинхронных запросов в Vue JS. Получены навыки работы с промисами.

Вход и выход пользователя:

```
signIn: function (e) {
  e.preventDefault()

  const submissionData = {
    username: this.username,
    password: this.password
  }

  this.$http.post('/user/login', submissionData)
    .then((response) => {
      localStorage.setItem('userData', JSON.stringify(response.data))
      location.href = '/'
    })
    .catch((error) => alert(error.response.data.statusText))

  logout() {
    localStorage.removeItem('userData')
    this.user = null
    location.href = '/'
  }
}
```

Поиск статей по сайту (перед этим получив список всех статей сайта):

```
searchList: function () {
  return this.list.filter(post => {return post.title.toLowerCase()
    .includes(this.search.toLowerCase())})
},
created() {
  this.$http.get('/post/list')
    .then((response) => this.list = response.data)
}
```

Получение последней добавленной статьи:

```
created() {
  this.$http.get('/post/lastadded')
    .then((response) => this.post = response.data)
}
```

Получение конкретной статьи:

```
created() {
  const url = this.$route.params.url
  this.$http.get('/post/info', {params: {url: url}})
    .then((response) => this.post = response.data)
}
```

Общий вид сайта:

[Войти](#)


Владимир Набоков

[Проза](#)[Поэзия](#)[Интервью](#)[Письма](#)[Публицистика](#)[Энтомология](#)[Воспоминания](#)

# С чего начать

Краткий путеводитель по творчеству.

Ознакомиться




[Письма](#)

## О Бунине

1938

«Как он, Бунин, похож на старую, тощую черепаху...»

[Читать](#)




[Интервью](#)

## Премьера «Лолиты»

1962

Интервью перед премьерой экранизации Стэнли Кубрика.

[Читать](#)



STANLEY KUBRICK'S  
*Lolita*

[Войти](#)

Владимир Набоков

[Проза](#)[Поэзия](#)[Интервью](#)[Письма](#)[Публицистика](#)[Энтомология](#)[Воспоминания](#)


## Авторизация

Email

Пароль

[Войти](#)

Еще не зарегистрированы? [Регистрация](#)

[В начало](#)

[Профиль](#)


Владимир Набоков

[Проза](#)[Поэзия](#)[Интервью](#)[Письма](#)[Публицистика](#)[Энтомология](#)[Воспоминания](#)

## Профиль

Привет! Это твой профиль.

[Выйти](#)

[В начало](#)

Поэзия

Поэмы

Бледный огонь (1962)

Стихи

Билет (1927)

И. А. Бунину (1920)



В начало

И. А. Бунину

Поэзия  
1920

Как воды гор, твой голос горд и чист.  
Алмазный стих наполнен райским медом.  
Ты любишь мир и юный месяц, лист,  
желтеющий над смуглым сочным плодом.

Ты любишь змей, тяжелых злых узлов  
лиловый лоск на дне сухой ложбины.  
Ты любишь нежный шелест голубиный  
вокруг лазурных, влажных куполов.

Твой стих роскошный и скупой, холодный  
и жгучий стих один горит, один  
над маревом губительных годин,  
и весь в цветах твой жертвенник свободный.

Он каплет в ночь росой ледяной  
и янтарями благовоний знойных,  
и нагота твоих созвучий стройных  
сияет, мила как блеск звезд, шаров и звездной

Профиль

# Владимир Набоков



Проза

Поэзия

Интервью

Письма

Публицистика

Энтомология

Воспоминания

## Поиск

пруст

Анкета о Прусте (1930)

[В начало](#)



## **Вывод**

В результате выполнения работы нам удалось разработать, можно сказать, полноценное веб-приложение. Сначала мы сверстали страницы с помощью HTML, CSS и шаблона Bootstrap. Затем использовали Vue.js для написания клиентской части будущего приложения. Развернув серверную часть приложения с помощью docker и yii2 и создав базу данных MySQL, мы написали REST API для запросов. После заполнения базы данных мы соединили клиентскую и серверную часть, используя для запросов к API библиотеку Axios.