

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное учреждение высшего
образования

Санкт-Петербургский национальный исследовательский университет информационных
технологий механики и оптики

Мегафакультет трансляционных информационных технологий

Факультет информационных технологий и программирования

Лабораторная работа №2

По дисциплине «Web-программирование»

Шаблонизация веб-страниц приложения

**Выполнил студент группы
М33122:**

Федотенко Николай Владимирович

Проверил:

Приискалов Роман Андреевич

САНКТ-ПЕТЕРБУРГ

2022

Цель работы:

Получение навыков работы с шаблонизаторами пользовательских представлений.

Краткие теоретические сведения:

- **Handlebars** – это бесплатная библиотека веб-шаблонов с открытым исходным кодом, позволяющая разработчикам создавать эффективные семантические шаблоны. Была выбрана в качестве шаблонизатора для выполнения данной работы.
- **Partials** – шаблоны, отвечающие за части веб-страниц и вызываемые впоследствии другими шаблонами. Для их использования в NestJS нужно явно зарегистрировать папку с данными шаблонами, как и папку со всеми шаблонами в целом (*views*).
- **Interceptor** – класс (“перехватчик”), аннотированный декоратором `@Injectable()` и реализующий интерфейс *NestInterceptor*. Впоследствии вызывается в контроллере посредством декоратора `@UseInterceptors()`.



Ход выполнения работы:

Данная лабораторная работа выполнена в операционной системе *macOS*.

Используемая IDE: *WebStorm 2021.3* (by JetBrains)

1. Установка шаблонизатора [Handlebars](#) (*hbs*):

```
npm install --save hbs
```

2. Определение шаблонизатора при инициализации приложения:

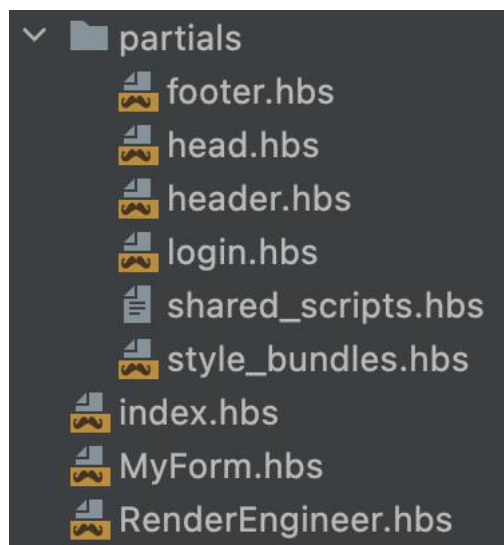
```
app.useStaticAssets(join(__dirname, '..', 'public'));
app.setBaseViewsDir(join(__dirname, '..', 'views'));
app.setViewEngine('hbs');

// eslint-disable-next-line @typescript-eslint/no-var-requires
const hbs = require('hbs');
hbs.registerPartials(join(__dirname, '..', 'views', 'partials'));
hbs.registerHelper('times', function (n, block) {
  let accum = '';
  for (let i = 1; i <= n; ++i) accum += block.fn(i);
  return accum;
});
```

Здесь *setBaseViewsDir* и *registerPartials* указывают путь к шаблонам и partials, а *registerHelper* создаёт вспомогательную функцию для шаблонизатора (инкремент цикл).

3. Выделение в отдельные представления

Общая архитектура шаблонов:



a. Общие стили (*style_bundles.hbs*):

```
<!-- Shared CSS -->
<link rel="stylesheet" href="css/outlines.css">
```

b. Общие скрипты (*shared_scripts.hbs*):

```
<!-- Shared Javascript -->
<script src="JS/LoadPageTime.js"></script>
<script src="JS/NavBar.js"></script>
```

с. Метаданные (*head.hbs*):

```
<!-- Базовые метайнформации -->
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Федотенко Николай М33122</title>
<meta name="description" content="Сайт-резюме Федотенко Николая
Владимировича">
<meta name="keywords" content="резюме, портфолио, ИТМО">
{{> style_bundles }}
{{> shared_scripts }}
```

d. Заголовок (*header.hbs*):

```
<div class="title">
  <h1>Николай Федотенко</h1>
</div>
{{> login }}
<div class="nav-bar">
  <nav>
    <a href="/"><span class="link">Разработчик C++</span></a>
    <a href="/RenderEngineer"><span class="link">Рендер
инженер</span></a>
    <a href="/MyForm"><span class="link">TODO List</span></a>
  </nav>
</div>
```

е. Подвал (*footer.hbs*):

```
<div class="inf-block" id="contact">
  <span class="block-title"><h2>Мои контакты</h2></span>
  <a href="https://github.com/kalkolay" target="_blank">GitHub</a>
  <a href="https://vk.com/kalkolay" target="_blank">VK</a>
  <a href="https://www.facebook.com/kalkolay"
target="_blank">Facebook</a>
  <a href="https://www.linkedin.com/in/nikolay-fedotenko-8472b215b/"
target="_blank">LinkedIn</a>
  <a href="https://join.skype.com/invite/n6dnsiBkd5mD"
target="_blank">Skype</a>
</div>

<div id="time">{{ time }}</div>
```

f. Информация о сессии (*login.hbs*):

```
{{unless signed_in}}  
<div class="login-wrapper">  
    <form id="login" action="/auth/login" method="post"  
target="_blank">  
        <label for="username">  
            <input placeholder="Логин" type="text" id="username"  
name="username">  
        </label>  
        <label for="password">  
            <input placeholder="Пароль" type="text" id="password"  
name="password">  
        </label>  
        <input type="submit" value="Войти">  
    </form>  
</div>  
{{else}}  
<div class="login-wrapper" style="margin-bottom: 8px">  
    Вы авторизованы под именем Николай&nbsp;
```

```

        <form id="logout" action="/logout" method="post" target="_blank">
            <button type='submit' name='parameter'>Выйти</button>
        </form>
    </div>
    {{/unless}}

```

g. Контейнер под основное содержание главной страницы (*index.hbs*):

```

<!DOCTYPE html>
<html lang="ru">
<head>
    {{> head }}
    <script src="JS/Network.js"></script>
    <script src="JS/SlidrWork.js"></script>
    <script type="text/javascript" src="JS/slidr/slidr.min.js"></script>
</head>
<body>
<div class="card">
    <!-- Header -->
    <header>
        {{> header }}
    </header>

    <main>

        <div class="inf-block" id="about">
            <span class="block-title"><h2>Главные факты обо мне</h2></span>
            <div class="facts" id="facts">
                <ol>
                    {{#each data.facts}}
                        <li>{{ this }}</li>
                    {{/each}}
                </ol>
            </div>
            <div class="image" id="image">
                
            </div>
        </div>
        <div class="inf-block" id="code-block">

            <span class="block-title"><h2>Пример моего кода</h2></span>
            <div id="code">
                <pre>
                    <code>
template <typename T>
static constexpr bool is_land_or_air_transport_v =
std::is_same<T, LandTransport>::value ||
std::is_same<T, AirTransport>::value;

template <typename T, class = std::enable_if_t<
    !is_land_or_air_transport_v<T> & & & >
inline T FindFastestTransport
(std::vector<T>& transports,
double distance) = delete;

template <typename T>
inline T FindFastestTransport
(std::vector<T>& transports,
double distance)
{
    T ans;

```

```

        double minTime = std::numeric_limits<double>::max();
        for (auto transport : transports)
        {
            if (transport.Update(distance) < minTime)
            {
                minTime = transport.Update(distance);
                ans = transport;
            }
        }
        return ans;
    }

</code>
</pre>
</div>
</div>
<div class="inf-block" id="stack-block">
    <span class="block-title"><h2>Мой стек</h2></span>
    <form>
        <button type="button" onclick="setPlaceholder(1 +
Math.floor(Math.random() * 4))">
            Получить новый Placeholder
        </button>
    </form>
    <div class="wrapper">
        <!-- Skills -->
        {{#each data.skills}}
            <div class="wrapper-child">
                <img class="skill-pic" src={{ this.link }} alt={{
this.name }}}>
            </div>
        {{/each}}
        <!-- Placeholders -->
        {{#times 5}}
            <div class="wrapper-child">
                
            </div>
        {{/times}}
    </div>

</div>
<div class="inf-block" id="projects">
    <span class="block-title"><h2>Мои проекты</h2></span>
    <div id="slidr-img">
        {{#each data.projects}}
            <img class="project-pic" data-slidr={{ this.id }} src={{
this.link }} alt={{ this.name }}/>
        {{/each}}
    </div>
</div>

</main>
<!-- Footer -->
<footer>
    {{> footer }}
</footer>
</div>
</body>
</html>

```

h. Контейнер под основное содержание второй страницы (*RenderEngineer.hbs*):

```
<!DOCTYPE html>
<html lang="ru">
<head>
  {{> head}}
  <link rel="stylesheet" href="css/Animate.css">
</head>
<body>
<div class="card">
<!-- Header -->
<header>
  {{> header }}
</header>

  <main>

    <div class="inf-block" id="study">
      В процессе переквалификации
      <div class="wrapper">
        {{#times 11}}
          <div class="line line{{ this }}">
            <span class="circle circle-top"></span>
            <div class="dotted">
              <span class="dot dot-top"></span>
              <span class="dot dot-middle-top"></span>
              <span class="dot dot-middle-bottom"></span>
              <span class="dot dot-bottom"></span>
            </div>
            <span class="circle circle-bottom"></span>
          </div>
        {{/times}}
      </div>
    </div>

  </main>

</div>
</body>
</html>
```

i. Контейнер под основное содержание третьей страницы (*MyForm.hbs*):

```
<!DOCTYPE html>
<html lang="ru">
<head>
  {{> head }}
  <link rel="stylesheet" href="css/Form.css">
  <script src="JS/FormLoader.js"></script>
</head>
<body>
<div class="card">
  <!-- Header -->
  <header>
    {{> header }}
  </header>

  <main>

    <div class="form-block" id="form">
      <span class="form-title"><h2>Список заданий на
неделю</h2></span>

      <form id="settings">
```

```

<h2>Настройки</h2>

<div class="settings-wrapper">
<div class="form-item">
  <label>Максимальное число заданий
    <select name="tasks-number">
      {{#times 5}}
        <option value={{ this }}>{{ this
}}</option>
      {{/times}}
    </select>
  </label>
</div>
</div>

<div class="form-item">
  <label>Количество дней
    <select name="days-number">
      <option value="5">5</option>
      <option value="6">6</option>
      <option value="7">7</option>
    </select>
  </label>
</div>

  <input type="submit" value="Создать">
</form>
<div class="task-list"></div>

<template id="day-template">
  <div class="day">
    <div class="day-name"></div>
    <form>
      <label>
        <input placeholder="Введите задание"
type="text" name="text">
      </label>
    </form>
    <div class="day-tasks"></div>
  </div>
</template>
<template id="task-template">
  <div class="task-data">
    <div class="task-data__index"></div>
    <div class="task-data__content"></div>
  </div>
</template>
</div>

</main>

<!-- Footer -->
<footer>
  {{> footer }}
</footer>
</div>
</body>
</html>

```

Повторяющиеся элементы шаблонизируются с помощью вспомогательного метода.

Рендер шаблонизированных страниц (*app.controller.ts*):

```
import {
  Controller,
  Get,
  Render,
  Request,
  Post,
  UseGuards,
  UseInterceptors,
} from '@nestjs/common';

@Controller()
export class AppController {
  constructor(private authService: AuthService) {}

  @Get()
  @Render('index')
  root() {
    return {
      signed_in: true,
      data: {
        facts: [
          'Живу в Санкт-Петербурге',
          'Портирую игры в General Arcade',
          'Студент ИТМО',
          'Обладаю глубокими познаниями в C++ и графике',
          'Участник и призёр многих хакатонов',
          'Автор статьи на <a href="https://habr.com/ru/post/440388/"
target="_blank">Хабре</a>',
        ],
        skills: [
          {
            name: 'C++',
            link:
'https://camo.githubusercontent.com/341abe5ac8494136a07b71a4ee902b1d6432e31ad08f35775b91810810ebc953/68747470733a2f2f69736f6370702e6f72672f6173736574732f696d616765732f6370705f6c6f676f2e706e67',
          },
          {
            name: 'Boost',
            link:
'https://camo.githubusercontent.com/6c9e25acf9131ff7f5ca89c5cc606bad483c1733b10eb76343aea1eef4fc0cd9/68747470733a2f2f746865626f6f73746370706c69627261726965732e636f6d2f7374617469632f6d61696e2f696d672f626f6f73742d6c6f676f2e737667',
          },
          {
            name: 'CMake',
            link:
'https://camo.githubusercontent.com/c18886afaa90feblad8177ed1d8f51b46304d86895b38aaf8cbf487dd567e468/68747470733a2f2f636d616b652e6f72672f77702d636f6e74656e742f75706c6f6164732f323031382f31312f636d616b655f6c6f676f5f736c696465722e706e67',
          },
          {
            name: 'OpenCL',
            link:
'https://camo.githubusercontent.com/5c4cb3688d035f32aaab3e4e320a116a47984710b88aa787467d0f4938ac2dae/68747470733a2f2f6873746f2e6f72672f67657470726f2f6861622722f706f73745f696d616765732f3136642f6161642f6139392f31366461616461393961623932316561353331316336343138653435396531382e706e67',
          },
          {
            name: 'CUDA',
```

```
        link:
'https://camo.githubusercontent.com/ef12567304a64d11a16be9eb99f957eefabcb2e
3e5578f89049ec10685c25e/68747470733a2f2f6e76776f726c642e72752f66696c65732f6e6
577732f6e76696469612d637564612d33322d72656c656173652d63616e6469646174652d322f
6e76696469612d637564612e6a7067',
    },
    {
        name: 'Python',
        link:
'https://camo.githubusercontent.com/888e388801f947dec7c3d843942c277af25fe2b1a
ed1821542c4e711f210312a/68747470733a2f2f75706c6f61642e77696b696d656469612e6f7
2672f77696b6970656469612f636f6d6d6f6e732f7468756d622f632f63332f507974686f6e2d
6c6f676f2d6e6f746578742e7376672f37363870782d507974686f6e2d6c6f676f2d6e6f74657
8742e7376672e706e67',
    },
    {
        name: 'Tensorflow',
        link:
'https://camo.githubusercontent.com/b37a6f7520a19b76cdb4d21dcba8efb6bf0fb77c1
2710a6ba24105f0ef46fb2e/68747470733a2f2f75706c6f61642e77696b696d656469612e6f7
2672f77696b6970656469612f636f6d6d6f6e732f7468756d622f322f32642f54656e736f7266
6c6f775f6c6f676f2e7376672f3132303070782d54656e736f72666c6f775f6c6f676f2e73766
72e706e67',
    },
    {
        name: 'PyTorch',
        link:
'https://camo.githubusercontent.com/3482591d996028fd207a726ac63ce563239e00990
260e677e8c451c005d00888/68747470733a2f2f75706c6f61642e77696b696d656469612e6f7
2672f77696b6970656469612f636f6d6d6f6e732f7468756d622f312f31302f5079546f726368
5f6c6f676f5f69636f6e2e7376672f36333570782d5079546f7263685f6c6f676f5f69636f6e2
e7376672e706e67',
    },
    {
        name: 'OpenCV',
        link:
'https://camo.githubusercontent.com/b922b6a2486cbbae8b031fc1336b74ae50b3c568d
c16a2c8691b4a0f242b5296/68747470733a2f2f7777772e706e676974656d2e636f6d2f70696
d67732f6d2f3135322d313532373137335f6f70656e63762d6c6f676f2d706e672d7472616e73
706172656e742d706e672e706e67',
    },
    {
        name: 'DirectX',
        link:
'https://camo.githubusercontent.com/d418e24f1d117b519f839f5a809652d883affbf55
5308faaf73d3a47a9fc318c/68747470733a2f2f7777772e6e6f74696369617333642e636f6d2
f696d6167656e65732f6e6f7469636961732f3230313930382f447831316c6f676f2e6a7067',
    },
    {
        name: 'C#',
        link:
'https://camo.githubusercontent.com/e5f1cbf59a8752f8a31ba28ea3b788daf4c188a84
870865acfc16c5567bfd5ce/68747470733a2f2f7365656b6c6f676f2e636f6d2f696d6167657
32f432f632d73686172702d632d6c6f676f2d303246313737313442412d7365656b6c6f676f2e
636f6d2e706e67',
    },
    {
        name: 'Java',
        link:
'https://camo.githubusercontent.com/3220bf0ac0ab62fd72ebe46f2317e16a9daf3f90b
7c066bcd6589f0181c113f9/68747470733a2f2f63646e2e776f726c64766563746f726c6f676
f2e636f6d2f6c6f676f732f6a6176612e737667',
    },
    }
```

```

        name: 'Git',
        link:
'https://camo.githubusercontent.com/b8ee9fd2e9b26a7265ece6dbc6f5c7449928b84f4
5a08fe5852d6a8dfd915fb3/68747470733a2f2f6769742d73636d2e636f6d2f696d616765732
f6c6f676f732f646f776e6c6f6164732f4769742d49636f6e2d31373838432e706e67',
    },
    {
        name: 'Lua',
        link:
'https://camo.githubusercontent.com/ac23621aa950bb432fda8bf60d9c75a4701dbdea3
4f6201d809aced5529c75b8/68747470733a2f2f75706c6f61642e77696b696d656469612e6f7
2672f77696b6970656469612f636f6d6d6f6e732f7468756d622f632f63662f4c75612d4c6f67
6f2e7376672f3132303070782d4c75612d4c6f676f2e7376672e706e67',
    },
    {
        name: 'ActionScript',
        link:
'https://camo.githubusercontent.com/72068492a769f0083ee9768ff52ddeceab503db65
f656a4d11b11e53b0cf6882/68747470733a2f2f692e70696e696d672e636f6d2f6f726967696
e616c732f61642f32352f31352f61643235313565396437663739343536343231363866333638
663964626435372e6a7067',
    },
],
projects: [
    {
        id: 'one',
        name: 'Witcher 3',
        link:
'https://ixbt.online/gametech/games/2021/01/31/CEgdlW9GHLsXlYnUXse3tNBWvu8EWi
AMVvoUqCfI.jpg',
    },
    {
        id: 'two',
        name: 'Darkest Dungeon',
        link: 'https://fs-prod-cdn.nintendo-
europe.com/media/images/10_share_images/games_15/nintendo_switch_download_software_1/H2x1_NSwitchDS_DarkestDungeon_image1600w.jpg',
    },
    {
        id: 'three',
        name: 'Risen',
        link: 'https://goodtorr.ru/wp-content/uploads/2018/06/risen-
e1529751408214.jpg',
    },
    {
        id: 'four',
        name: 'MudRunner',
        link: 'https://i.ytimg.com/vi/v0866rNdZ8E/maxresdefault.jpg',
    },
    {
        id: 'five',
        name: 'Warhammer 40k',
        link: 'https://i.playground.ru/p/wCC1IHm7Bb6HYW81Pl0BOw.jpeg',
    },
],
],
};
}

@Get('RenderEngineer')
@Render('RenderEngineer')
renderEngineerPage() {
    return;
}

```

```

@Get ('MyForm')
@Render ('MyForm')
myFormPage () {
  return;
}
}

```

4. Подготовка двух состояний о сессии пользователя:

В прошлом пункте показано, как данная подготовка (авторизован и не авторизован) реализована в partial'e *login.hbs* с помощью директивы ***unless*** (контроллером в нужную страницу передаётся переменная *signed_in*, которая меняет состояния).

5. Подсчёт времени загрузки страницы сервером с помощью Interceptor'a:

Был реализован собственный Interceptor, который впоследствии используется в контроллере благодаря декоратору *@UseInterceptors()*.

```

timer.interceptor.ts
1  import {
2    CallHandler,
3    ExecutionContext,
4    Injectable,
5    NestInterceptor,
6  } from '@nestjs/common';
7  import { map, Observable } from 'rxjs';
8
9  @Injectable()
10 export class TimerInterceptor implements NestInterceptor {
11   intercept(
12     context: ExecutionContext,
13     next: CallHandler,
14   ): Observable<{ time: string }> | Promise<Observable<{ time: string }>> {
15     const startTime = Date.now();
16     return next
17       .handle()
18       .pipe(map( project: () => ({ time: `+ ${Date.now() - startTime} ms (сервер)` })));
19   }
20 }

```

Вызов в контроллере для одной из страниц:

```

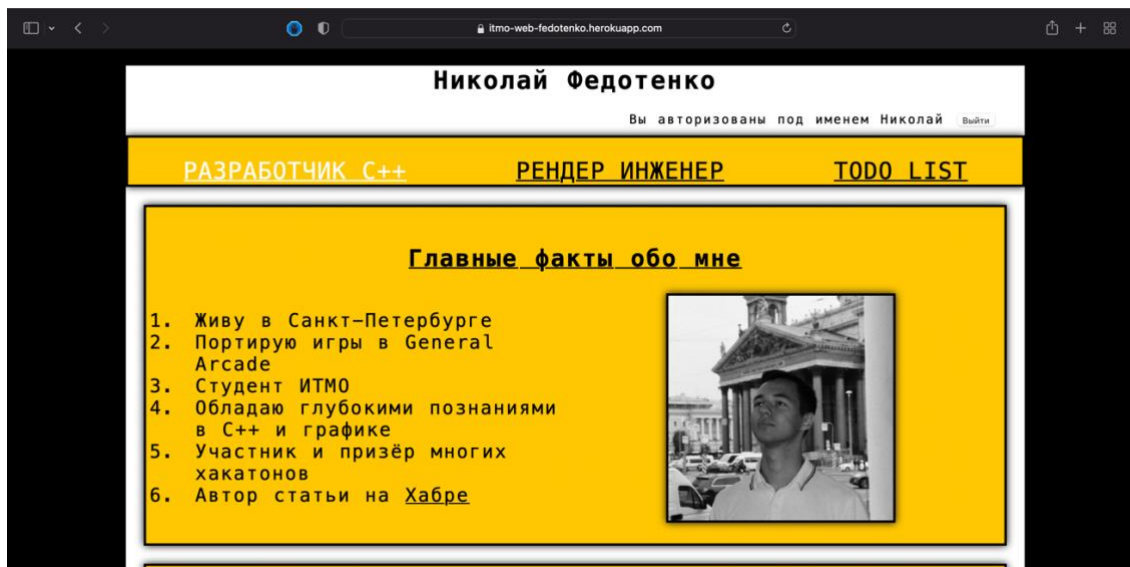
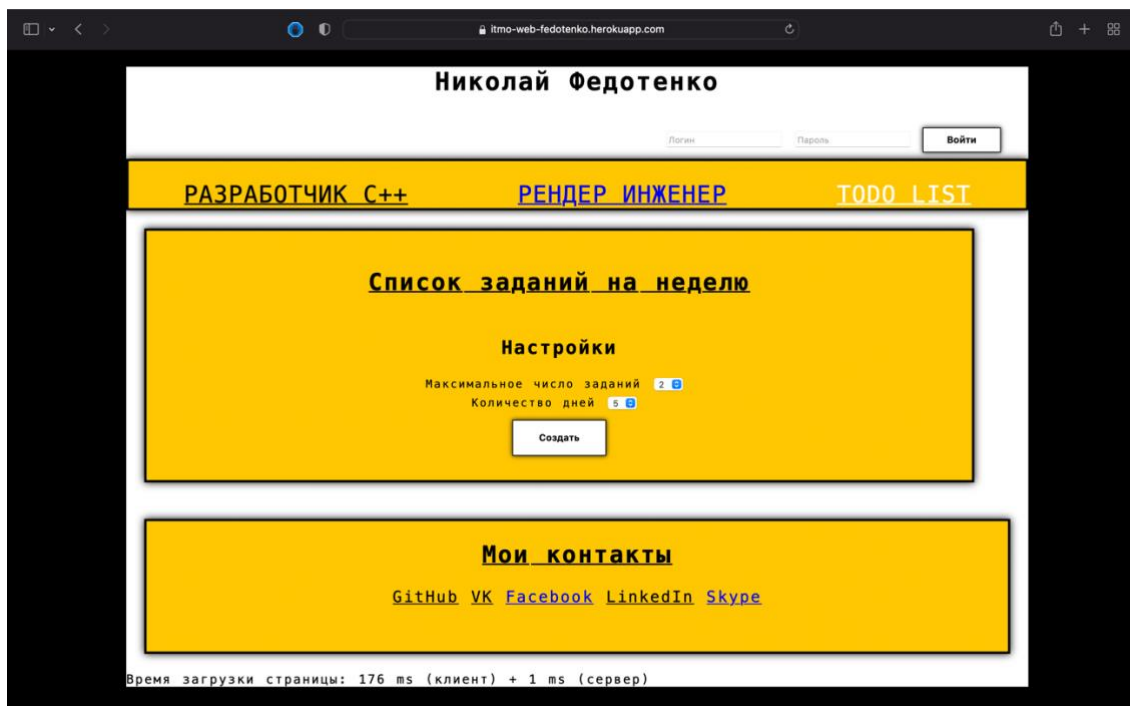
import { TimerInterceptor } from '../timer.interceptor';
...
@Get ('MyForm')
@Render ('MyForm')
@UseInterceptors (TimerInterceptor)
myFormPage () {
  return;
}
...

```

Также, был модифицирован JavaScript файл, подсчитывающий время загрузки клиента:

```
LoadPageTime.js
1  (() => {
2    window.onload = () => {
3      const timer = Math.round(performance.now());
4      const currentText = document.getElementById('time').innerHTML;
5      document.getElementById(
6        elementId: 'time',
7      ).innerHTML = `Время загрузки страницы: ${timer} ms (клиент) ${currentText}`;
8    };
9  })();
```

6. Проверка результата (<https://itmo-web-fedotenko.herokuapp.com/>):



Вывод:

Я получил навыки работы с шаблонизатором пользовательских представлений [Handlebars](#), научился выделять отдельные части (*partials*) веб страниц. Также, я реализовал два состояния для первичной работы с сессиями и научился работать с перехватчиками.