ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
Факультет информационных технологий и программирования

Лабораторная работа №2

«Шаблонизация веб-страниц приложения»
по дисциплине «Web программирование»

Выполнил: студент группы M33122 Белоногов Е.В.

Санкт-Петербург
2022

Был выбран шаблонизатор “Handlebars”. Определение используемого в приложении шаблонизатора и пути к partial шаблонам:

main.ts

src > main.ts > bootstrap

```ts
import { NestFactory } from '@nestjs/core';
import { NestExpressApplication } from '@nestjs/platform-express';
import { join } from 'path';
import { AppModule } from './app.module';

async function bootstrap() {
  const app = await NestFactory.create<NestExpressApplication>(
    AppModule,
  );

  const hbs = require('hbs');

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

  hbs.registerPartials(join(__dirname, '..', 'views/partials'))

  await app.listen(process.env.PORT == undefined ? 3000 : process.env.PORT);
}
bootstrap();
```

Далее были выделены повторяющиеся блоки:

1) Меню

main.ts | menu.hbs

views > partials > menu.hbs > nav.sidebar > div.text.d-flex.p-2 > div#nav-icon3.pushmenu.opened

```hbs
<div id="nav-icon3" class="pushmenu" data-title="MENU">
    <span></span>
    <span></span>
    <span></span>
    <span></span>
  </div>
  <nav class="sidebar">
    <div class="text d-flex p-2">
      <h4>МЕНЮ САЙТА</h4>
      <div id="nav-icon3" class="pushmenu opened">
        <span></span>
        <span></span>
        <span></span>
        <span></span>
      </div>
    </div>
    <div class="menu-main-menu-container">
      <ul id="menu-main-menu">
        <li><a href="index.html">Главная</a></li>
        <li><a href="page.html">Галерея фотографий</a></li>
        <li><a href="plants.html">Таблица</a></li>
        <li><a href="dhtml.html">Страница с DHTML</a></li>
        <li><a href="promise.html">Страница с Promise</a></li>
        <li><a href="libusing.html">Библиотека CodeMirror</a></li>
      </ul>

      {{> auth }}

    </div>
  </div>
    </div>
  </nav>
  <div class="hidden-overley"></div>
```

2) Header

main.ts | header.hbs

views > partials > header.hbs > script

```hbs
<title>{{ title }}</title>
<meta charset="utf-8"/>
<meta name="author" content="Belonogov Evgeny"/>
<meta name="description" content={{ metaDescription }}/>
<link rel="stylesheet" type="text/css" href="/styles/main.css">
<script src="/scripts/script.js"></script>

```

3) Footer

main.ts | footer.hbs M

views > partials > footer.hbs > footer > p#loadingtime

```hbs
<footer>
  <p><a title="Go to vk.com" href="https://www.vk.com/evgeny_belonogov">Ссылка на VK</a></p>
  <div class ="advertisement">
    <marquee direction="right" scrolldelay="500">Здесь могла быть ваша реклама</marquee>
    <p>РЕКЛАМА</p>
  </div>

  <p id="loadingtime">{{ serverResponseTime }}</p>
  <script>
    const d = new Date();
    const day = new Array("Воскресенье","Понедельник","Вторник","Среда","Четверг","Пятница","Суббота");
    const month = new Array("января","февраля","марта","апреля","мая","июня","июля","августа","сентября","октября","ноября","декабря");

    document.write("<p align=\"center\"><time>" + day[d.getDay()] + " " + d.getDate() + " " + month[d.getMonth()]
    + " " + d.getFullYear() + " г." + "</time></p>");
  </script>
</footer>
```

4) Статус аутентификации

main.ts | auth.hbs

views > partials > auth.hbs > div.authorization

```hbs
<div class="authorization">
    <p>Статус авторизации:</p>
        {{#if authStatus}}
            <p>авторизован как {{ user }} (молодец)</p>
            <a href="/logout.html">Выйти</a>
        {{else}}
            <p>не авторизован</p>
            <div class="authLink">
                <a href="/login.html">Авторизоваться</a>
            </div>
        {{/if}}
</div>
```

Изменения в AppController для обработки Get запросов и рендеринга запрашиваемых страниц:

TS main.ts | TS app.controller.ts M | TS app.service.ts M

src > TS app.controller.ts > ...

```ts
import { Controller, Get, Redirect, Render, UseInterceptors } from '@nestjs/common';
import { AppService, ServerResponseTimeInterceptor } from './app.service';


@Controller()
@UseInterceptors(ServerResponseTimeInterceptor)
export class AppController {
  constructor(private readonly appService: AppService) {}

  // @Get()
  // getHello(): string {
  //   return this.appService.getHello();
  // }

  @Get('/')
  @Redirect('index.html')
  getRoot() {}

  @Get('/index.html')
  @Render('index')
  getIndex() {
    return {
      title: 'Portfolio Belonogov Evgeny M33122',
      etaDescription: 'This page can be used as a portfolio',
      authStatus: false,
      serverResponseTime: String(this.appService.getServerResponseTime())
    };
  }
```

TS main.ts | TS app.controller.ts M | TS app.service.ts M

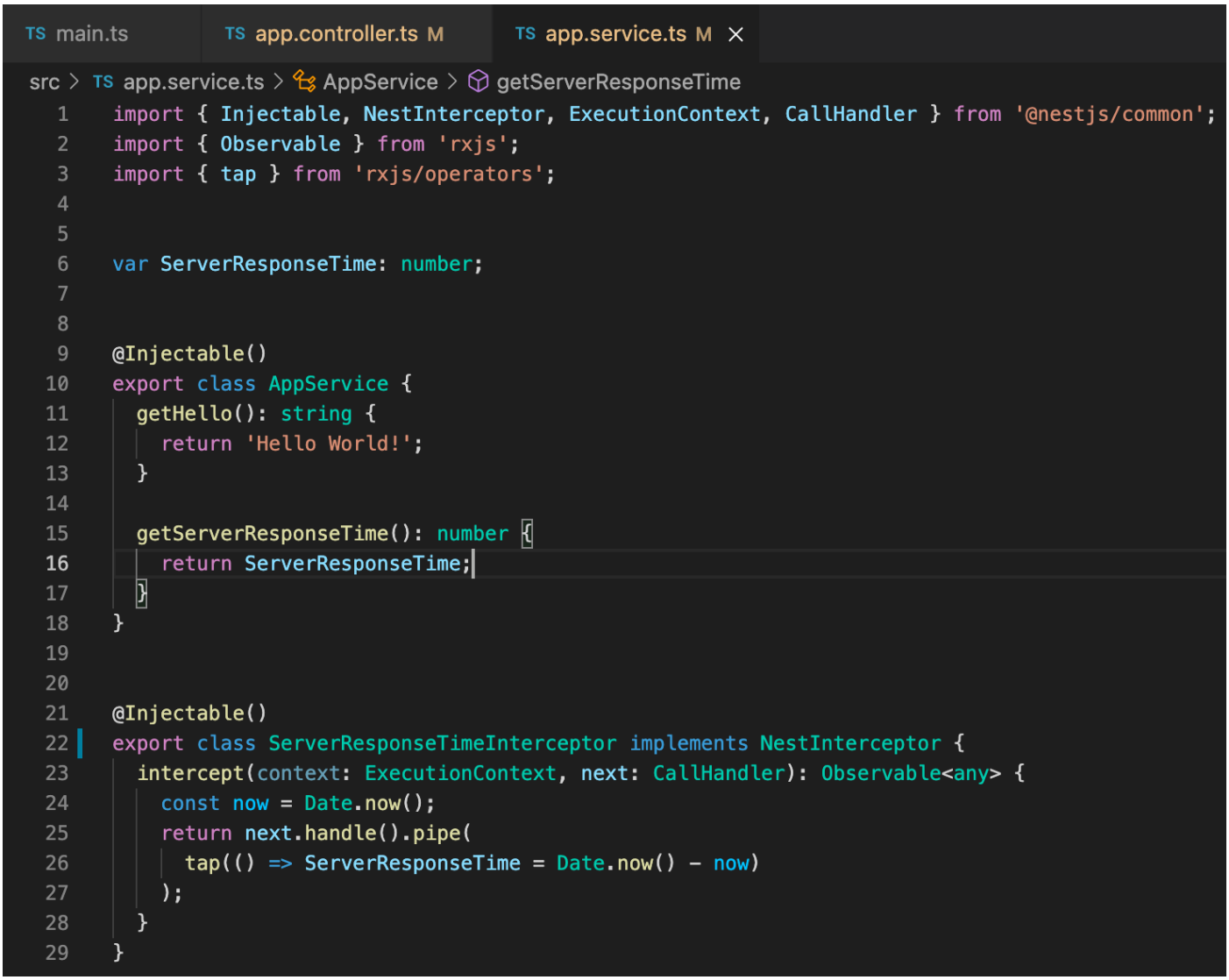
src > TS app.controller.ts > ...

```ts
  @Get('page.html')
  @Render('page')
  getPage() {
    return {
      title: 'Photo gallery',
      metaDescription: 'This page contains photo',
      serverResponseTime: String(this.appService.getServerResponseTime())
    };
  }

  @Get('plants.html')
  @Render('plants')
  getPlants() {
    return {
      title: 'Plants info',
      metaDescription: 'This page contains information about plants',
      serverResponseTime: String(this.appService.getServerResponseTime())
    };
  }

  @Get('dhtml.html')
  @Render('dhtml')
  getDhtml() {
    return {
      title: 'DHTML',
      metaDescription: 'This page contains dynamic html (DHTML)',
      serverResponseTime: String(this.appService.getServerResponseTime())
    };
  }
```

Для вычисления времени обработки запроса сервером имплементирован класс типа “Interceptor”:



Ссылка на приложение:
https://evgeny-backend-itmo.herokuapp.com/