

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Фронт-энд разработка

Отчет

Лабораторная работа 1

Выполнил:

Рыбкин Михаил

Группа К33402

Проверил:

Добряков Д. И.

Санкт-Петербург

2022 г.

## **Задача**

Вёрстка сайта средствами HTML, CSS и Bootstrap

В рамках данной лабораторной работы Вам предложено выбрать один из нескольких вариантов. Выбранный вариант останется единым на весь курс и будет использоваться в последующих лабораторных работах.

По выбранному варианту необходимо будет выполнить вёрстку сайта средствами HTML, CSS и Bootstrap. Продумать и реализовать моменты, в которых необходим JS (например, открытие модальных окон).

## Ход работы

В качестве варианта для выполнения работы я выбрал систему для выдачи заданий преподавателями и сдачи студентами. Проект буду реализовывать на фреймворке VueJs с использованием компонент-библиотеки ElementPlus.

Для начала создам в файле `index.html` блок, внутри которого будут рендериться динамические компоненты:

*public/index.html*

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,initial-scale=1.0">
  <link rel="icon" href="<%= BASE_URL %>favicon.ico">
  <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Material+Icons|Material+Icons
+Outlined">
  <link
href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;7
00;900&display=swap"
rel="stylesheet">

  <title><%= htmlWebpackPlugin.options.title %></title>
</head>
<body>
<noscript>
  <strong>We're sorry but <%= htmlWebpackPlugin.options.title %> doesn't
work properly without JavaScript enabled.
  Please enable it to continue.</strong>
</noscript>
<div id="app"></div>
<!-- built files will be auto injected -->
</body>
</html>
```

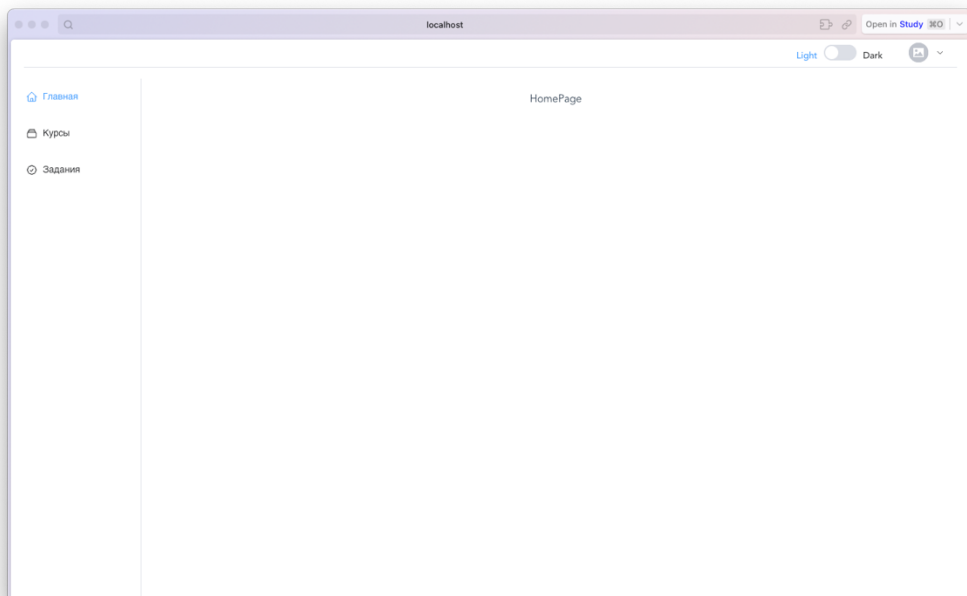
Также, в этом файле указываем ссылки на внешние шрифты, блок с предупреждением для браузеров, в которых выключен Javascript и meta-теги страницы.

Затем мы можем приступить к созданию стартового компонента. Разделим основную структуру страницы на сайдбар, хедер и основную рабочую зону:

*src/App.vue*

```
<template>
  <el-container v-loading="loading" class="wrapper">
    <el-header>
      <main-header/>
    </el-header>
    <el-container v-if="usersStore.isLoggedIn && !loading">
      <el-aside v-if="route.meta.sidebar" width="200px">
        <main-sidebar/>
      </el-aside>
      <el-main>
        <router-view/>
      </el-main>
    </el-container>
    <auth-modal :open-auth-modal="!usersStore.isLoggedIn && !loading"/>
  </el-container>
</template>
```

Выглядит такая раскладка так:



Элементы меню в сайдбаре:

*src/components/MainSidebar.vue*

```
<template>
  <el-menu
    :collapse="isCollapsed"
    :default-active="route.meta.sidebarIndex"
    :router="false"
    mode="vertical"
    style="height: 100%"
  >
    <el-menu-item
      v-for="route in routes"
      :key="route.path"
      :index="route.index"
      @click="redirectToPage(route.path)"
      :name="t(route.name)"
    >
      <el-icon>
        <component :is="route.icon"/>
      </el-icon>
      <template #title>
        {{ t(route.name) }}
      </template>
    </el-menu-item>

    <div class="flex-grow"/>
  </el-menu>
</template>
```

В конце, как можно заметить, использую `div` с классом `flex-grow` для выравнивания содержимого сайдбара ближе к его верхней границе. Для этого я применяю следующую таблицу стилей:

*src/components/MainSidebar.vue*

```
<style scoped>
.flex-grow {
  flex-grow: 1;
}
</style>
```

Элементы меню в хедере:

```
<template>
  <el-menu
    :ellipsis="false"
    mode="horizontal"
  >

    <div class="flex-grow"/>

    <el-menu-item>
      <template #title>
        <el-switch
          v-model="isDark"
          active-text="Dark"
          inactive-text="Light"
          size="large"
        />
      </template>
    </el-menu-item>
    <el-sub-menu index="account">
      <template #title>
        <el-avatar :size="30">
          
        </el-avatar>
      </template>

      <el-menu-item
        index="account-signout"
        @click="userStore.signOut()"
      >
        <template #title>
          {{ t('auth.signOutPrompt') }}
        </template>
      </el-menu-item>
    </el-sub-menu>

  </el-menu>
</template>
```

Ссылка на аватар пока статичная. Как можно заметить, здесь я применяю такой же блок с классом `flex-grow`, как и в сайдбаре. Соответственно, необходимая таблица стилей также используется.

После верстки структуры сайта можно приступить к созданию страниц и подключению к серверному API. Более подробно эти части проекта будут описаны в следующих ЛР.

## Вывод

По итогу выполнения лабораторной работы были получены базовые навыки верстки HTML-страниц.