МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ   
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Программной инженерии

Специальность 1–40 01 01 «Программное обеспечение информационных технологий»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КУРСОВОГО ПРОЕКТА**

по дисциплине «Компьютерные языки разметки»

ТемаВеб-сайт «Железнодорожный вокзал»

**Исполнитель**

студент 1 курса 6 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_ С. А. Розель

подпись, дата

**Руководитель**

старший преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.В. Ржеутская

должность, учен. степень, ученое звание подпись, дата

Допущен(а) к защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата, подпись

Курсовой проект защищен с оценкой

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.В. Ржеутская\_\_

подпись дата инициалы и фамилия

Оглавление

[1. Постановка задачи 4](#_Toc134574267)

[1.1. Обзор аналогичных решений 4](#_Toc134574268)

[1.2. Техническое задание 8](#_Toc134574269)

[1.3. Выбор средств реализации программного продукта 8](#_Toc134574270)

[1.4. Вывод 9](#_Toc134574271)

[2. Проектирование страниц веб-сайта 10](#_Toc134574272)

[2.1. Выбор способа верстки 10](#_Toc134574273)

[2.2. Выбор стилевого оформления 10](#_Toc134574274)

[2.3. Выбор шрифтового оформления 10](#_Toc134574275)

[2.4. Разработка логотипа 10](#_Toc134574276)

[2.5. Разработка пользовательских элементов 11](#_Toc134574277)

[2.6. Разработка спецэффектов 12](#_Toc134574278)

[2.7. Выводы 12](#_Toc134574280)

[3. Реализация структуры веб-сайта 13](#_Toc134574281)

[3.1. Структура HTML-документа 13](#_Toc134574282)

[3.2. Добавление таблиц стилей Sass и CSS 13](#_Toc134574283)

[3.3. Использование стандартов XML (SVG) 14](#_Toc134574284)

[3.4. Управление элементами DOM 14](#_Toc134574285)

[3.4. Выводы 16](#_Toc134574286)

[4. Тестирование веб-сайта 17](#_Toc134574287)

[4.1. Адаптивный дизайн веб-сайта 17](#_Toc134574288)

[4.2. Кроссбраузерность веб-сайта 18](#_Toc134574289)

[4.3. Руководство пользователя 18](#_Toc134574290)

[4.4. Выводы 19](#_Toc134574291)

[Заключение 20](#_Toc134574292)

[Список использованных источников 21](#_Toc134574293)

[Приложение А Прототип веб-старниц…………………………………………22](#_Toc134574294)

[Приложение Б Макет структуры веб-сайта 24](#_Toc134574296)

[Приложение В Листинг HTML-документа 26](#_Toc134574298)

[Приложение Г Листинг SCSS и CSS 27](#_Toc134574300)

[Приложение Д Листинг XML-файлов 31](#_Toc134574302)

[Приложение Е Листинг SVG 35](#_Toc134574304)

[Приложение Ж Листинг JavaScript 38](#_Toc134574306)

**Введениеу**

Возможность заказа мест на поезд и просмотр разных маршрутов намного облегчает перемещение по нашей земле. В наше время есть много сайтов на которых можно заказать билеты на поезд, также посмотреть расписание и пути, по которым ходят поезда. Если у железнодорожного вокзала есть сайт, это очень большой плюс для него.

Сайт — это то, что может выделить вокзал у пассажиров. Без веб-сайта

людям намного сложнее было заказывать билеты и смотреть расписание маршрутов, так как нужно было прийти на сам вокзал. В настоящее время можно дома зайти на сайт и выполнить данные действия онлайн.

Цель курсовой работы: разработать веб-сайт для железнодорожного вокзала с использованием HTML5, а также с применением Sass/CSS3.

Задачи курсовой работы:

* проанализировать существующие языки разметки, инструменты и библиотеки для создания веб-сайта;
* разработать макет и прототип сайта;
* разработать структуру веб-сайта;
* наполнить сайт информацией по теме;
* протестировать веб-сайт;
* разработать руководство пользователя.

Целевой аудиторией являются пользователи, заказывающее билеты на поезда.

1. Постановка задачи

1.1. Обзор аналогичных решений

Перед тем как приступать к разработке своего сайта можно изучить готовые решения и посмотреть, какие функции по мнению других разработчиков является наиболее популярным у посетителей сайта.

|  |
| --- |
| Дизайн аналога веб-сайта курсового проекта представлен представлен на рисунке 1.1. |

Рисунок 1.1 – Веб-сайт «https://www.rw.by/»

На сайте помимо контактов, бронирования билетов, списка маршрутов есть так же и новости. Данный сайт представляет разные услуги, связанные с поездами: поездки, перевозки, расписание маршрутов.

|  |
| --- |
| Дальше перейдем к бронированию мест, которое представлено на рисунке 1.2 |

Рисунок 1.2 – Форма бронирования мест

Данная форма представляет возможность выбрать помимо обычного откуда и куда, также в одну сторону или же туда и обратно, представляет вариант, что маршрут может быть составным. Так же можно забронировать место с ненумерованными местами и посмотреть расписание на каждой станции. Ниже находиться список самых популярных маршрутов и их цена. Так же имеется туристические маршруты.

У данного сайта есть только один недостаток, домашняя страница слишком переполнена информацией. Когда пользователь впервые зайдет, он может не сориентироваться.

|  |
| --- |
| На рисунке 1.3 представлен другой пример сайта для железной дороги |

Рисунок 1.3 – Веб-сайт «https://www.rzd.ru/»

На данном сайте сразу при входе есть форма для поиска нужного маршрута, что является очень большим плюсом, так как пользователю не нужно искать по странице данную форму или переходить на другую страницу.

Ниже, на рисунке 1.4, представлены разные услуги для пользователей. Можно авторизироваться и купить обычный или премиальный билет. Так же можно посмотреть какие скидки вокзал представляет для его посетителей. Все эти услуги удобно разбиты на квадраты, с помощью которых модно переместиться на следующую страницу для более подробного ознакомления с услугой. Так же можно посмотреть, какие услуги будут представлены при поездке. Они были разделены на: «До поездки», «Во время поездки», «После поездки». Также нажав на прямоугольник можно перейти на другую страницу и ознакомиться с представленными услугами более подробно.

|  |
| --- |
|  |

Рисунок 1.4 – Услуги представляющее вокзал

Данный сайт оформлен более компактно, форма для поиска маршрута сразу же на домашней странице, для покупки билета также нужно сделать минимум действий, всего перейти по ссылке, размещенной в первой плитке.

|  |
| --- |
| На рисунке 1.5 представлен ещё один, последний, пример железно дорожного вокзала |

Рисунок 1.5 – Веб-сайт «https://www.intercity.pl/ru/»

Дизайн сайта минималистичный, на главной странице находиться только самое нужное для пользователя, который зашел забронировать билет на поезд. Так же предлагается купить абонементный билет, если пользователь часто пользуется услугами данного железнодорожного вокзала.

|  |
| --- |
| На рисунке 1.6 отображены услуги, которые предоставляет данный железнодорожный вокзал. |

Рисунок 1.6 – Услуги веб-сайта «https://www.intercity.pl/ru/»

|  |
| --- |
| На рисунке 1.7 отображены «Часто разыскиваемые» |

Рисунок 1.7 – «Часто разыскиваемые»

На данных рисунках, представленных услуги и раздел в котором вынесено самое главное. Главный плюс этого сайта в том, что все самое главное вынесено на главную страницу. Пользователь может быстро найти все, что ему будет нужно.

1.2. Техническое задание

Поставлена задача реализовать полноценный многостраничный веб-сайт на тему «Железнодорожный вокзал». На сайте должна быть описана разная информация, которая будет полезна пользователю, это контактные данные, представление услуг, расписание маршрутов.

Главной целью веб-сайта является предоставить пользователю удобный и быстрый функционал, с помощью которого он сможет быстро и легко найти нужный маршрут, забронировать билет на поезд и так далее.

Основными задачами сайта являются:

* предоставление навигационного меню;
* предоставление удобного расписания поездов;
* возможность бронирования билетов;
* предоставление создания личного аккаунта, для дальнейшей работы на сайте;

На главной странице меню будет располагаться сверху. В меню будут предоставляться ссылки на основную информацию, такую как: расписание, бронирование, вход или регистрация.

Веб-страница должна быть кроссбраузерной и адаптивной для мобильной версии (ширина до 425px), планшетного устройства (ширина до 1024px). Начиная с ширины в 1024px должна открываться версия для персонального компьютера.

1.3. Выбор средств реализации программного продукта

Курсовая работа выполняется в редакторе кода Visual Studio Code.

При создании веб-страниц были использованы языки: HTML, CSS, JS, XML.

HTML  — стандартизированный язык гипертекстовой разметки документов для просмотра [веб-страниц](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0) в [браузере](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80). Веб-браузеры получают HTML документ от сервера по протоколам [HTTP](https://ru.wikipedia.org/wiki/HTTP)/[HTTPS](https://ru.wikipedia.org/wiki/HTTPS) или открывают с локального диска, далее интерпретируют код в интерфейс, который будет отображаться на экране монитора.

CSS — это каскадная таблица стилей, которая определяет отображение HTML-документов. CSS может работать со всеми элементами HTML-документа.

JavaScript – это язык программирования, который используют разработчики для создания интерактивных веб-страниц. Функции JavaScript могут улучшить удобство взаимодействия пользователя с веб-сайтом: от обновления ленты новостей в социальных сетях и до отображения анимации и интерактивных карт.

 Node.js — программная платформа, превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами [ввода-вывода](https://ru.wikipedia.org/wiki/%D0%92%D0%B2%D0%BE%D0%B4-%D0%B2%D1%8B%D0%B2%D0%BE%D0%B4) через свой [API](https://ru.wikipedia.org/wiki/API), написанный на [C++](https://ru.wikipedia.org/wiki/C%2B%2B), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. Node.js применяется преимущественно на сервере, выполняя роль [веб-сервера](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80).

1.4. Вывод

В этом разделе рассмотрены недостатки и достоинства аналогичных решений на выбранную тему сайта для выбора лучших функций и их внедрения на своем сайте. Были определены основные технические задания по созданию веб-сайта. Также выбраны средства реализации программного продукта, а именно языки HTML, CSS, JS, XML.

2. Проектирование страниц веб-сайта

2.1. Выбор способа верстки

Требования проекта включают в себя кроссбраузерность и адаптивность. По этой причине была выбрана grid вёрстка.

Grid-модель позволяет создавать структуры, необходимые для обеспечения отзывчивости сайтов на различных устройствах. Это означает, что сайт будет одинаково хорошо смотреться на компьютере, телефоне и планшете.

2.2. Выбор стилевого оформления

Выбор цветовой гаммы— важный этап разработки проекта. Цвета должны сочетаться между собой. Цвета применялись как к текстовым, так и к блочным элементам в качестве фона. Далее представлены цвета, используемые на страницах.

Цветы были выбраны более теплые, как представлено на рисунке 2.1, у нас есть три основных цвета: два отценка оранжевого и серый для фона.

|  |
| --- |
|  |

Рисунок 2.1 – Цветовая гамма для сайта

2.3. Выбор шрифтового оформления

Основными шрифтами на сайте являются Roboto и IBM Plex Serif, которые прекрасно вписываются в концепт сайта и отлично дополняют дизайн.

Размерность была использована от 16 до 20, цвета текста #000000 или #FFFFFF в зависимости от того на каком фоне находится текст.

2.4. Разработка логотипа

Логотип — это главный знак сайта, при запоминании которого, люди будут ассистировать сайт с этим логотипом. Логотип — это не абстрактный набор символов или просто красивое изображение, это один из главных элементов сайта, который должен быть легко запоминаемым и узнаваемым.

|  |
| --- |
|  |

Рисунок 2.2 – Логотип веб-сайта

Логотип создан таким образом, чтобы выделить контуры вокзала, данное решение было принято с точки того, что логотип сайта у нас и есть вокзал и у пользователей будет ассоциация, что данный сайт сделан именно для минского вокзала и поиск этого сайта в интернете будет намного проще. Логотип, представленный на рисунке 2.2 был создан с помощью Adobe Illustrator.

2.5. Разработка пользовательских элементов

В проекте будут реализованы элементы пользовательского интерфейса такие как: навигационное меню, формы для входа и регистрации, форма для бронирования билетов, форма для поиска маршрутов.

|  |
| --- |
|  |

Рисунок 2.3 – Навигационное меню

На рисунке 2.3 представлен образ навигационного меню, нажав на логотип пользователь может вернуться к начальной странице. Элементы: расписание поездов, забронировать, войти переводят пользователя на соответствующее страницы.

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | |  |   а | |  | | --- | |  |   б |
| |  | | --- | |  |   в | |  | | --- | |  |   г |
| а – форма поиска, б – бронирование билетов, в – регистрация, г - вход | |

Рисунок 2.4 – формы для сайта

На рисунке 2.4 представлены четыре формы, с которыми пользователь может взаимодействовать.

2.6. Разработка спецэффектов

На сайте реализованы спецэффекты в виде анимаций, есть выплывающее уведомление, которые извещают пользователя о том, вошел ли он в аккаунт или нет.

|  |
| --- |
|  |

Рисунок 2.5 – Пример уведомления

2.7. Выводы

В данном разделе бал создан прототип веб-сайта, определены задачи по созданию содержания на веб-сайте. Выбрано единое цветовое и шрифтовое оформления сайта для удобства просмотра пользователем. Разработан дизайн и расположение на сайте пользовательских элементов, спецэффектов и анимации. Логотип выполнен под цветовую гамму и концепцию сайта.

3. Реализация структуры веб-сайта

3.1. Структура HTML-документа

Структура сайта – это система HTML-документов, которые связанны между собой ссылками, которые в свою очередь позволяют пользователю переходить по страницам нашего сайта.

Всю структуру сайта объединяет тег <html></html>. В нем лежат ещё два тега <head></head> и <body></body>

В теге <head></head> прописана служебная информация, метатеги, которые помогают поисковым системам обрабатывать информацию о сайте, название веб-сайта, язык сайта.

Содержимое тега <head> не отображается напрямую на веб-странице, за исключением тега <title> устанавливающего заголовок окна веб-страницы.

В теге <body></body> располагается все теги, которые нужны для верстки сайта и расположения информации на нем.

Структура документа состоит из семантических тегов. Листинг представлен в [приложении 3](#_Приложение_3_Листинг). Это теги, которые предназначены для того, чтобы компьютерные программы (поисковые системы, сборщики информации, речевые браузеры и т.д.) понимали, какой тип информации заложен в данных тегах.

Весь HTML, CSS, XML, Sass код был протестирован с помощью онлайн-сервиса validator.w3.org. Результат представлен на рисунке 3.1

|  |
| --- |
|  |

Рисунок 3.1 – Результат тестирования с помощью validator.w3.org

Ниже на листинге 3.1 представлена разметка навигационного меню, объединяет ее специальный семантический тег <nav></nav>. В нем находятся три ссылки на другие страницы сайта.

|  |
| --- |
| <nav class="menu-desktop">  <a href="/components/table\_train.html">Расписание поездов</a>  <a href="/components/booking.html">Забронировать</a>  <a href="/components/sign\_in.html" class="sign-in">Войти</a>  </nav> |

Листинг 3.1 – Навигационная панель

3.2. Добавление таблиц стилей Sass и CSS

В проекте использовалось в приоритете внешнее подключение таблиц стилей. Чтобы уменьшить повторение кода использовалась правило SCSS @mixin, пример использования приведен на листинге 3.2.

|  |
| --- |
| @mixin footer\_style {  width: 100%;  height: 50px;  background: #FCC12B;  p {  width: auto;  margin: 0 auto;  text-align: center;  font-family: 'IBM Plex Serif';  font-size: 15px;  }  } |

Листинг 3.2 – Использование миксина

В отдельные файлы были помещены медиа-запросы для двух основных страниц. К каждой странице стили были в отдельном файле, сделано это был с целью ускорить навигацию по сайту и читабельность (строковые стили ухудшают читабельность разметки сайта). В таблицах стилей используются селекторы по классу и селекторы тегов.

3.3. Использование стандартов XML (SVG)

Для создания логотипа сайта принято решение использовать изображение SVG-формата. Этот формат позволяет картинкам сохранять свое качество, несмотря на размер экрана.

XML в проекте использовался для хранения информации о пользователях, которые могут зарегистрироваться на сайте, и о маршрутах по которым ездят поезда и электрички.

## **3.4. Управление элементами DOM**

C помощью JavaScript реализовано меню бургер для мобильной версии (от 768px). Также на устройствах с тачскрином выпадающее подменю пункта контакты реализовано с помощью JavaScript. На листинге 3.3 представлен код бургер меню на JavaScript.

Так же с помощью JavaScript реализована вставка данных из XML с помощью fetch запроса и вставкой их в общий тег. На листинге 3.4 показан сам парс данных из XML.

|  |
| --- |
| let burgerMenu = document.querySelector('.burger-menu');  let burgerContainer = document.querySelector('.burger-container-off');  let line = document.querySelectorAll('.line');  let counter = 0;  burgerMenu.onclick = () => {  counter++;  if(counter % 2) {  burgerContainer.className = "burger-container-on";  line.forEach((el, i) => {  el.className = "line open-" +(i + 1);  })  } else if (!(counter % 2)) {  burgerContainer.className = "burger-container-off";  line.forEach((el, i) => {  el.className = "line close-" + (i + 1);  })  }  } |

Листинг 3.3 – Реализация бургер меню

|  |
| --- |
| let parser = (xmlString => {  const parser = new DOMParser();  const xmlDoc = parser.parseFromString(xmlString, 'text/xml');  const ways = xmlDoc.getElementsByTagName('way');  const times = xmlDoc.getElementsByTagName('time');  const train\_imgs = xmlDoc.getElementsByTagName('trainImg');  ways\_arr = table(ways, ways\_arr);  times\_arr = table(times, times\_arr);  train\_imgs\_arr = table(train\_imgs, train\_imgs\_arr);  for (let i = 0; i < ways\_arr.length; i++) {    document.querySelector('.table').insertAdjacentHTML("beforeend",  `<div class="rows">  <p class="time">${times\_arr[i]}</p>  <img class="trainImg" src="${train\_imgs\_arr[i]}">  <p class="way">${ways\_arr[i]}</p>  </div>`  );  }  }) |

Листинг 3.4 – Парсинг XML

|  |
| --- |
| async function parseXML() {    let a = await fetch('/xml/train\_schedule.xml');  let xmlString = await a.text();  parser(xmlString);  document.querySelector('.button\_booking').onclick = () => {  let str\_way, flag = true;  let date\_value = document.querySelector('.input\_date').value;  let where\_from = document.querySelector('.input\_where\_from').value;  let where\_to = document.querySelector('.input\_where\_to').value;  times\_arr.forEach((element, index) => {  str\_way = ways\_arr[index].split(' ');  if(element != date\_value || str\_way[2] != where\_to || str\_way[0] != where\_from) {    document.querySelectorAll('.rows')[index].style.display = 'none';  } else {  document.querySelectorAll('.rows')[index].style.display = 'grid';  flag = false;  }  })  if (flag) {  document.querySelector('.table').innerHTML = "<p class='error\_message'>Данного маршрута нет</p>";  }  };  } |

На листинге 3.5 показан fetch запрос и работа поиска маршутов.

Листинг 3.5 – Реализация fetch запроса

3.4. Выводы

На данном этапе была реализована структура на HTML, а также созданы таблицы стилей Sass/CSS. Было продемонстрировано использование стандартов XML и SVG, а также языка программирования JS. Было определено стилевое оформление веб-сайта, созданы анимации, подзагрузка информации, а также веб-сайт был подготовлен к следующему этапу – к тестированию.

# **4. Тестирование веб-сайта**

## **4.1. Адаптивный дизайн веб-сайта**

Для создания адаптивного дизайна использовались технологии, такие как медиа-запросы, CSS Grid и изменяемые изображения. Большинство значений в свойствах прописывались с адаптивными единицами измерения. Так же предусматривалось то, чтобы пользователь мог удобно находиться на сайте. Пример адаптивности главной страницы на планшетах представлен на рисунке 4.1.

|  |  |
| --- | --- |
| а | б |

а – вид на планшете, б – вид на мобильных устройствах

Рисунок 4.1 – Адаптивность сайта

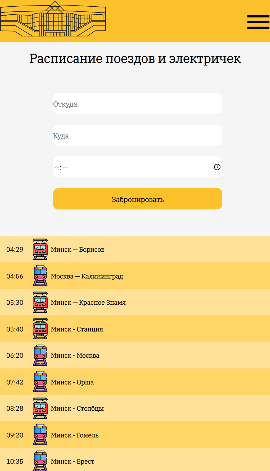
На рисунке 4.2 представлен вид на мобильных устройствах страницы с расписание поездов

Рисунок 4.2 – Вид на мобильных устройствах страницы с расписанием

## **4.2. Кроссбраузерность веб-сайта**

Для создания кроссбраузерности сайта использовались префиксы в CSS-свойствах.

Список префиксов:

* -webkit- - для браузеров на основе движка WebKit, таких как Chrome и Safari;
* -moz- - для браузера Firefox;
* -ms- - для браузера Microsoft Edge;
* -o- - для браузера Opera.

Примеры отображения веб-сайта в браузерах Opera, Microsoft Edge, Chrome представлены на рисунке 4.3.

|  |  |
| --- | --- |
| а | б |
| в | г |

а – Chrome, б - Microsoft Edge, в – Firefox, в - Opera

Рисунок 4.3 – Вид сайта в разных браузерах

4.3. Руководство пользователя

Дизайн сайта создавался таким образом, чтобы пользователь, зайдя на него мог понять, куда нужно нажимать дальше, в зависимости от того что он хочет сделать.

Если спуститься в самый низ сайта можно получить информацию о вокзале.

Если перейти во вкладку «расписание поездов» можно найти свой маршрут, пример поиска представлен на рисунке 4.4.

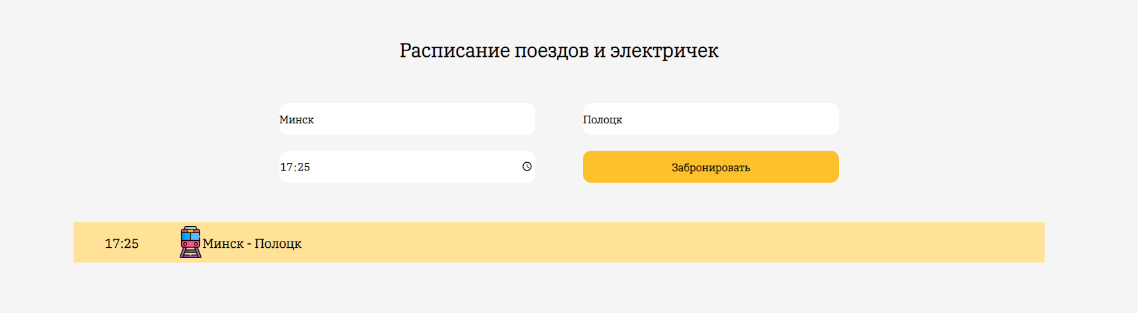


Рисунок 4.4 – Поиск маршрута

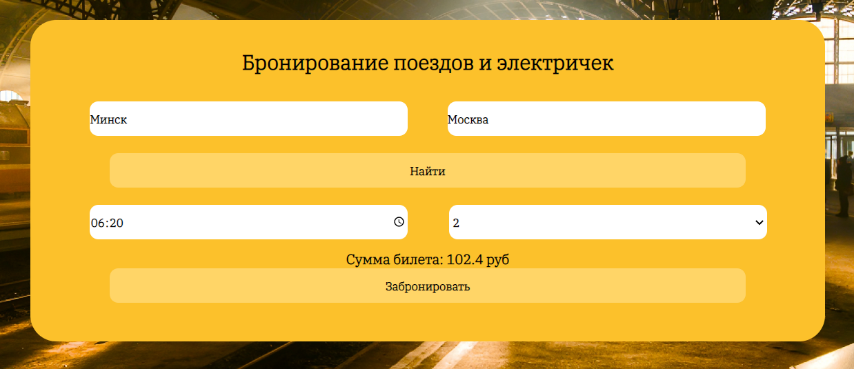
В форме «Забронировать» можно забронировать билет на поезд, пример заполения формы представлен на рисунке 4.5.

Рисунок 4.5 – Бронирование билета

В форме «зарегистрироваться» можно создать аккаунт, если пользователь впервые на сайте, а в форме «войти» можно войти в уже созданный аккаунт. Примеры, каким образом это работает изображено на рисунке 4.6.

|  |  |
| --- | --- |
| а | б |

а – пример регистрации, б – пример входа на сайт

Рисунок 4.6 – Заполнение форм

4.4. Выводы

В данном разделе мною было рассмотрено: поведение сайта на разных устройствах, а также браузерах. Так же создано руководство пользователя, где кратко расписано, как пользоваться веб-сайтом.

Заключение

В результате работы у нас получился сайт железнодорожного вокзала. Сайт работает быстро и стабильно. Так же сайт со временем можно дорабатывать сделав, допустим, страницу профиля пользователя.

Было проведенно сравнение аналогов, опираясь на которые были взяты только плюсы этих аналогов.

Для удобного написания кода веб-страницы была использована среда разработки Visual Studia Code, в которой были установлены плагины для облегченного пользования.

Были прочтены документации по использованию технологий, которые будут использоваться на этапе написания сайта.

Реализация важных элементов страницы либо же скриптов были представлены в приложениях, которые дают общую информацию о проекте.

Если подытожить, то за этот курсовой были использованы как теоретические знания, так и практические знания разработки веб-сайтов, для выполнения всех поставленных задач.

# **Список использованных источников**

1. Документация по Figma [Электронный ресурс]. – Режим доступа: <https://help.figma.com> – Дата доступа 01.03.2023.
2. HTML5, CSS3 и JavaScript. Исчерпывающее руководство. [Электронный ресурс]. Режим доступа – <https://webbooks.com.ua/?p=1048> – Дата доступа 05.03.2022.
3. Новая большая книга CSS [Печатный ресурс]. Дата доступа 07.03.2023
4. Документация по Sass/SCSS [Электронный ресурс]. – Режим доступа: <https://sass-scss.ru/guide/> – Дата доступа 10.03.2023.
5. Документация по JavaScript/HTML [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org> – Дата доступа 10.03.2023.
6. Введение в кросс-браузерное тестирование [Электронный ресурс]. – Режим доступа: [https://developer.mozilla.org/ru/docs/Learn/Tools\_and\_testing/  
   Cross\_browser\_testing/Introduction](https://developer.mozilla.org/ru/docs/Learn/Tools_and_testing/Cross_browser_testing/Introduction). – Дата обращения: 01.05.2023.

**Приложение А**

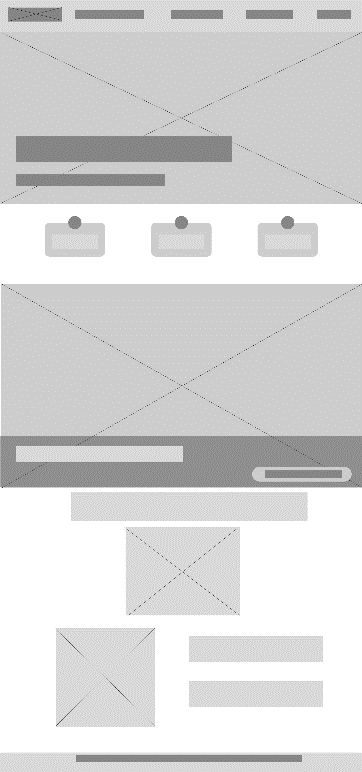
**** **Прототипы веб-страниц**

Рисунок 1 – Главная страница

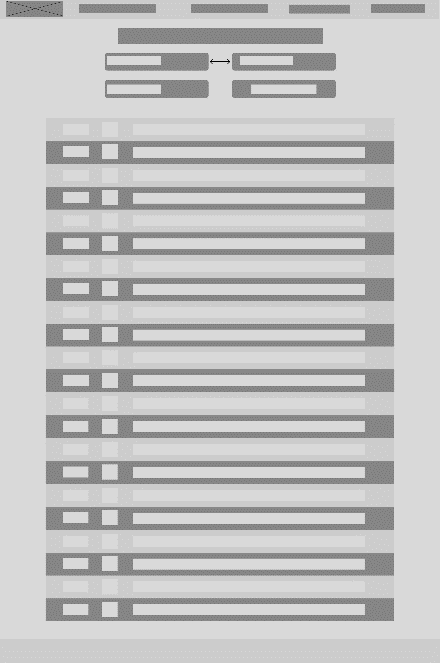


Рисунок 2 – Страница «Расписание поездов»

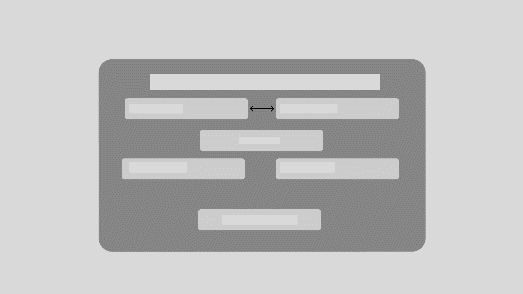


Рисунок 3 – Форма для бронирования билета

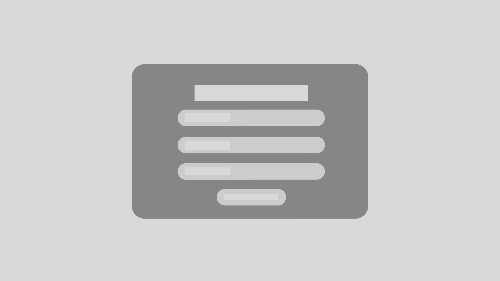


Рисунок 4 – Форма для входа в аккаунт

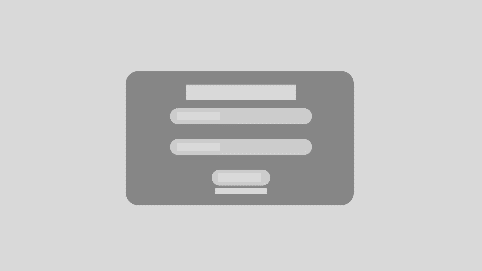


Рисунок 5 – Форма для регистрации

**Приложение Б**

**Макеты страниц**

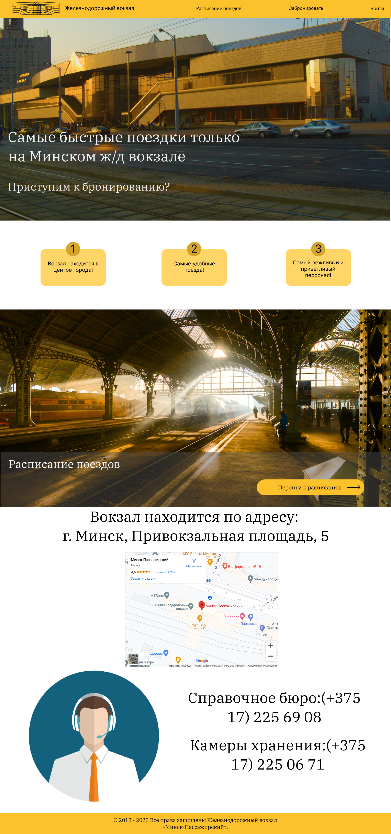


Рисунок 1 – Главная страница



Рисунок 2 – Страница «Расписание поездов»

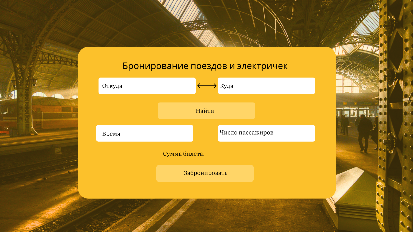


Рисунок 3 – Форма для бронирования билетов

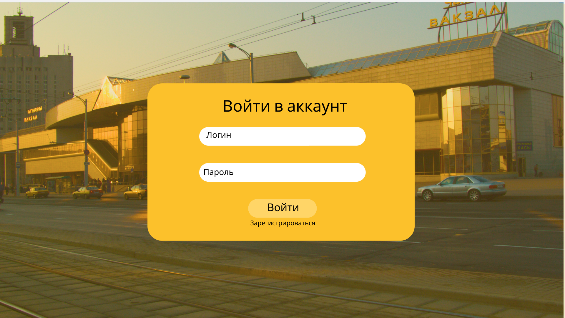


Рисунок 4 – Форма для входа в аккаунт

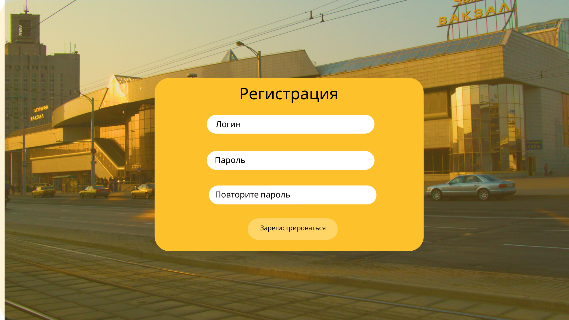


Рисунок 5 – Форма для регистрации

**Приложение В**

**Листинг HTML-документа**

|  |
| --- |
| <div class="menu">  <a href="/components/home.html"><img src="/img/вокзал.svg" alt="Иконка вокзала"></a>  <h3>Железнодорожный вокзал</h3>  <div class="menu-desktop">  <a href="/components/table\_train.html">Расписание поездов</a>  <a href="/components/booking.html">Забронировать</a>  <a href="/components/sign\_in.html" class="sign-in">Войти</a>  </div>  <div class="burger-menu">  <div class="line"></div>  <div class="line"></div>  <div class="line"></div>  </div>  </div>  <div class="burger-container-off">  <a href="/components/table\_train.html">Расписание поездов</a>  <a href="/components/booking.html">Забронировать</a>  <a href="/components/sign\_in.html" class="sign-in">Войти</a>  </div> |

Листинг 1 – Реализация верхнего колонтитула

|  |
| --- |
| <div class="info">  <p class="adress">Вокзал находится по адресу:  г. Минск, Привокзальная площадь, 5</p>  <iframe  src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d1089.646614760959!2d27.549525129411048!3d53.89059920122846!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x46dbcfdc32da8473%3A0x1cf1974b543e26c8!2z0JzQuNC90YHQui3Qn9Cw0YHRgdCw0LbQuNGA0YHQutC40Lk!5e0!3m2!1sru!2sby!4v1683651821649!5m2!1sru!2sby"  width="500" style="border:0;" allowfullscreen="" loading="lazy"  referrerpolicy="no-referrer-when-downgrade">  </iframe>  <img src="/img/support.png" alt="">  <p class="number support">Справочное бюро:(+375 17) 225 69 08</p>  <p class="number camera">Камеры хранения:(+375 17) 225 06 71</p>  </div> |

Листинг 2 – Реализация блока с дополнительной информацией

**Приложение Г**

**Листинг SCSS и CSS**

|  |
| --- |
| header {  @include header\_style();  margin-bottom: 10px;  } |

Листинг 3 – Реализация верхнего колонтитула SCSS

|  |
| --- |
| @mixin header\_style {  .menu {  width: 100%;  height: 80px;  background-color: #FCC12B;  font-family: Roboto;  line-height: 37px;  display: grid;  grid-template-columns: 270px auto auto 55px;  align-items: center;  img {  width: 200px;  height: 70px;  margin-left: 40px;  }  h3 {  justify-self: start;  user-select: none;  }  .sign-in {  margin-right: 40px;  }  }  .menu-desktop {  display: grid;  grid-template-columns: repeat(3, 1fr);  a {  width: auto;  height: auto;  justify-self: center;  user-select: none;  text-decoration: none;  color: #000;  font-family: 'Roboto';  font-weight: bold;  cursor: pointer;  }  }  .img-header {  width: 100%;  height: 760px;  background-image: url("/img/вокзал\_фон.png");  background-position: center center;  background-size: cover;  background-position: center;  p {  position: relative;  font-family: 'IBM Plex Serif';  font-style: normal;  font-weight: 400;  color: #fff;  }  .text-header {  width: 78%;  left: 40px;  top: 50%;  font-size: 4.5vw;  }  .booking {  width: 60%;  left: 40px;  top: 55%;  font-size: 2vw;  }  }  } |

Листинг 4 – реализация миксина для верхнего колонтитула SCSS

|  |
| --- |
| header {  margin-bottom: 10px;  }  header .menu {  width: 100%;  height: 80px;  background-color: #FCC12B;  font-family: Roboto;  line-height: 37px;  display: grid;  grid-template-columns: 270px auto auto 55px;  align-items: center;  }  header .menu img {  width: 200px;  height: 70px;  margin-left: 40px;  }  header .menu h3 {  justify-self: start;  -webkit-user-select: none;  -moz-user-select: none;  user-select: none;  }  header .menu .sign-in {  margin-right: 40px;  }  header .menu-desktop {  display: grid;  grid-template-columns: repeat(3, 1fr);  }  header .menu-desktop a {  width: auto;  height: auto;  justify-self: center;  -webkit-user-select: none;  -moz-user-select: none;  user-select: none;  text-decoration: none;  color: #000;  font-family: "Roboto";  font-weight: bold;  cursor: pointer;  }  header .img-header {  width: 100%;  height: 760px;  background-image: url("/img/вокзал\_фон.png");  background-position: center center;  background-size: cover;  background-position: center;  }  header .img-header p {  position: relative;  font-family: "IBM Plex Serif";  font-style: normal;  font-weight: 400;  color: #fff;  }  header .img-header .text-header {  width: 78%;  left: 40px;  top: 50%;  font-size: 4.5vw;  }  header .img-header .booking {  width: 60%;  left: 40px;  top: 55%;  font-size: 2vw;  } |

Листинг 5 – Реализация верхнего колонтитула CSS

**Приложение Д**

**Листинг XML-файлов**

|  |
| --- |
| <container>  <train>  <time>04:29</time>  <trainImg>../img/electric-train.png</trainImg>  <way>Минск — Борисов</way>  <price>10</price>  </train>  <train>  <time>04:56</time>  <trainImg>../img/train.png</trainImg>  <way>Москва — Калининград</way>  <price>134</price>  </train>  <train>  <time>05:30</time>  <trainImg>../img/electric-train.png</trainImg>  <way>Минск — Красное Знамя</way>  <price>16</price>  </train>  <train>  <time>05:40</time>  <trainImg>../img/electric-train.png</trainImg>  <way>Минск - Городея</way>  <price>6</price>  </train>  <train>  <time>06:20</time>  <trainImg>../img/train.png</trainImg>  <way>Минск - Москва</way>  <price>100</price>  </train>  <train>  <time>07:42</time>  <trainImg>../img/train.png</trainImg>  <way>Минск - Орша</way>  <price>8</price>  </train>  <train>  <time>08:28</time>  <trainImg>../img/electric-train.png</trainImg>  <way>Минск - Столбцы</way>  <price>8</price>  </train>  <train>  <time>09:20</time>  <trainImg>../img/train.png</trainImg>  <way>Минск - Гомель</way>  <price>15</price>  </train>  <train>  <time>10:35</time>  <trainImg>../img/train.png</trainImg>  <way>Минск - Брест</way>  <price>16</price>  </train>  <train>  <time>11:25</time>  <trainImg>../img/train.png</trainImg>  <way>Минск - Орша</way>  <price>12</price>  </train>  <train>  <time>12:58</time>  <trainImg>../img/electric-train.png</trainImg>  <way>Минск — Руденск</way>  <price>11</price>  </train>  <train>  <time>14:01</time>  <trainImg>../img/train.png</trainImg>  <way>Минск — Калинковичи</way>  <price>9</price>  </train>  <train>  <time>15:35</time>  <trainImg>../img/train.png</trainImg>  <way>Минск - Могилёв</way>  <price>15</price>  </train>  <train>  <time>16:13</time>  <trainImg>../img/train.png</trainImg>  <way>Минск - Орша</way>  <price>9</price>  </train>  <train>  <time>17:25</time>  <trainImg>../img/train.png</trainImg>  <way>Минск - Полоцк</way>  <price>12</price>  </train>  <train>  <time>18:14</time>  <trainImg>../img/electric-train.png</trainImg>  <way>Минск — Гудогай</way>  <price>6</price>  </train>  <train>  <time>19:25</time>  <trainImg>../img/electric-train.png</trainImg>  <way>Минск — Молодечно</way>  <price>3</price>  </train>  <train>  <time>20:41</time>  <trainImg>../img/train.png</trainImg>  <way>Минск - Орша</way>  <price>11</price>  </train>  <train>  <time>21:34</time>  <trainImg>../img/electric-train.png</trainImg>  <way>Минск - Полоцк</way>  <price>12</price>  </train>  <train>  <time>22:49</time>  <trainImg>../img/train.png</trainImg>  <way>Минск - Калинковичи</way>  <price>12</price>  </train>  <train>  <time>23:23</time>  <trainImg>../img/train.png</trainImg>  <way>Минск - Витебск</way>  <price>17</price>  </train>  <train>  <time>00:01</time>  <trainImg>../img/train.png</trainImg>  <way>Минск - Могилёв</way>  <price>20</price>  </train>  </container> |

Листинг 6 – Хранение данных о маршрутах

**Приложение Е**

**Листинг SVG**

|  |
| --- |
| <svg id="Слой\_1" data-name="Слой 1"  xmlns="http://www.w3.org/2000/svg" viewBox="0 0 737.01 255.12">  <defs>  <style>.cls-1,.cls-2{fill:none;stroke:#1d1e1c;}.cls-1{stroke-width:4px;}.cls-2{stroke-width:3px;}</style>  </defs>  <path class="cls-1" d="M409,34.47c11.44-4.28,324.73,14,324.73,14V101.5"/>  <polyline class="cls-1" points="3.2 139.25 120.03 139.25 242.46 209.64 273.57 209.64 273.57 247.9 225.62 247.9 104.21 174.44 3.2 173.93"/>  <polyline class="cls-1" points="273.57 247.9 492.42 247.9 492.42 209.64 519.46 209.64 638.83 139.25 733.71 139.25"/>  <polyline class="cls-1" points="733.71 173.42 656.17 173.42 537.82 247.9 492.42 247.9"/>  <polyline class="cls-1" points="537.82 247.9 607.71 247.9 662.81 210.66 733.71 210.66 733.71 101.5 716.88 101.5"/>  <polyline class="cls-1" points="462.83 247.9 462.83 210.66 302.65 210.66 302.65 247.9"/>  <path class="cls-1" d="M638.83,139.25V101.5s63.77-5.62,78,0v37.75"/>  <path class="cls-1" d="M120,139.25V101.5s-40.3-7.15-78.56,3.06v34.69"/>  <line class="cls-1" x1="3.2" y1="104.56" x2="41.46" y2="104.56"/>  <polyline class="cls-1" points="120.03 101.5 215.42 104.56 215.42 194.85"/>  <polyline class="cls-1" points="638.83 101.5 609.75 102.44 544.46 104.56 547.01 194.85"/>  <path class="cls-1" d="M215.42,104.56s38.77-26,109.68-13.27l57.64-54.58,55.1,54.58H325.1"/>  <path class="cls-1" d="M437.84,210.66V91.29s33.52-11.75,91.71,7.77q7.2,2.41,14.91,5.5"/>  <path class="cls-1" d="M733.71,87.21s-254-13.77-286.18-13.77L382.74,7.12l-26.08,26.7L318,73.44S20.55,79.56,3.2,87.21V211.68l95.4,2,56.62,34.18h70.4"/>  <path class="cls-1" d="M353.84,36.71S87.38,37.22,3.2,50.48V87.21"/>  <line class="cls-2" x1="519.46" y1="209.64" x2="519.46" y2="96.17"/>  <line class="cls-2" x1="242.46" y1="209.64" x2="242.46" y2="93.48"/>  <line class="cls-1" x1="325.1" y1="91.29" x2="325.1" y2="210.66"/>  <polygon class="cls-2" points="347.19 43.51 418.29 43.51 382.74 70.63 347.19 43.51"/>  <line class="cls-1" x1="317.96" y1="73.44" x2="447.53" y2="73.44"/>  <line class="cls-2" x1="623.49" y1="101.99" x2="623.49" y2="148.17"/>  <line class="cls-2" x1="611.03" y1="155.64" x2="611.03" y2="102.44"/>  <line class="cls-2" x1="599.33" y1="162.54" x2="599.33" y2="102.78"/>  <line class="cls-2" x1="586.07" y1="170.36" x2="586.07" y2="103.21"/>  <line class="cls-2" x1="571.35" y1="179.04" x2="571.35" y2="103.21"/>  <line class="cls-2" x1="559.32" y1="186.14" x2="559.32" y2="104.07"/>  <line class="cls-2" x1="462.83" y1="210.66" x2="462.83" y2="87.83"/>  <line class="cls-2" x1="492.42" y1="209.64" x2="492.42" y2="89.88"/>  <line class="cls-2" x1="534.45" y1="200.8" x2="534.45" y2="100.76"/>  <line class="cls-2" x1="302.65" y1="210.66" x2="302.65" y2="88.4"/>  <line class="cls-2" x1="273.57" y1="209.64" x2="273.57" y2="88.26"/>  <line class="cls-2" x1="258.27" y1="209.64" x2="258.27" y2="90.05"/>  <line class="cls-2" x1="227.08" y1="200.8" x2="227.08" y2="98.71"/>  <line class="cls-2" x1="288.37" y1="247.9" x2="288.37" y2="87.78"/>  <line class="cls-2" x1="476.1" y1="247.9" x2="476.1" y2="88.26"/>  <line class="cls-2" x1="201.58" y1="186.14" x2="201.58" y2="104.07"/>  <line class="cls-2" x1="189.24" y1="179.04" x2="189.24" y2="104.07"/>  <line class="cls-2" x1="174.13" y1="170.36" x2="174.13" y2="104.07"/>  <line class="cls-2" x1="160.54" y1="162.54" x2="160.54" y2="103.21"/>  <line class="cls-2" x1="148.54" y1="155.64" x2="148.54" y2="102.44"/>  <line class="cls-2" x1="135.55" y1="148.17" x2="135.55" y2="101.99"/>  <path class="cls-2" d="M638.83,129H544.46c-66.57-22.45-106.62-12.24-106.62-12.24H325.1"/>  <path class="cls-2" d="M604,159.78H546l-1.56-.77c-66.57-22.45-106.62-12.24-106.62-12.24H325.1"/>  <path class="cls-2" d="M552.74,190h-5.87l-2.41-1c-66.57-22.45-106.62-12.24-106.62-12.24H325.1"/>  <path class="cls-2" d="M325.1,206.84H437.84s48.14-4,68.87,2.8v-117"/>  <path class="cls-2" d="M325.1,116.61s-46.55-9.12-106.62,12.24"/>  <polyline class="cls-2" points="219.41 128.52 215.42 129.94 120.03 129.94"/>  <path class="cls-2" d="M325.1,146.58s-46.55-9.12-106.62,12.24"/>  <polyline class="cls-2" points="219.41 158.49 215.42 159.91 157.08 159.91"/>  <path class="cls-2" d="M325.1,176.55s-46.55-9.12-106.62,12.24"/>  <polyline class="cls-2" points="219.41 188.46 215.42 189.88 208.08 189.88"/>  <path class="cls-2" d="M323.82,206.84s-46.61-3.4-65.8,1.93"/>  <line class="cls-2" x1="426.61" y1="206.84" x2="426.61" y2="91.29"/>  <line class="cls-2" x1="408.98" y1="206.84" x2="408.98" y2="91.29"/>  <line class="cls-2" x1="389.89" y1="206.84" x2="389.89" y2="91.29"/>  <line class="cls-2" x1="372.67" y1="206.84" x2="372.67" y2="91.29"/>  <line class="cls-2" x1="354.88" y1="206.84" x2="354.88" y2="91.29"/>  <line class="cls-2" x1="338.62" y1="207.27" x2="338.62" y2="91.29"/>  <polyline class="cls-1" points="447.53 247.9 447.53 221.63 317.96 221.63 317.96 247.9"/>  <polyline class="cls-2" points="273.57 218.06 242.46 218.06 120.03 148.17 3.2 148.17"/>  <polyline class="cls-2" points="492.42 218.06 519.46 218.06 638.83 148.17 733.71 148.17"/>  <line class="cls-2" x1="422.79" y1="247.9" x2="422.79" y2="221.63"/>  <line class="cls-2" x1="437.84" y1="247.9" x2="437.84" y2="221.63"/>  <line class="cls-2" x1="412.59" y1="247.9" x2="412.59" y2="221.63"/>  <line class="cls-2" x1="393.46" y1="247.9" x2="393.46" y2="221.63"/>  <line class="cls-2" x1="372.67" y1="247.9" x2="372.67" y2="221.63"/>  <line class="cls-2" x1="353.84" y1="247.9" x2="353.84" y2="221.63"/>  <line class="cls-2" x1="340.91" y1="247.9" x2="340.91" y2="221.63"/>  <line class="cls-2" x1="326.63" y1="247.9" x2="326.63" y2="221.63"/>  </svg> |

Листинг 7 - Логотип

**Приложение Ж**

**Листинг JavaScript**

|  |
| --- |
| let burgerMenu = document.querySelector('.burger-menu');  let burgerContainer = document.querySelector('.burger-container-off');  let line = document.querySelectorAll('.line');  let counter = 0;  burgerMenu.onclick = () => {  counter++;  if(counter % 2) {  burgerContainer.className = "burger-container-on";    line.forEach((el, i) => {  el.className = "line open-" +(i + 1);  })  } else if (!(counter % 2)) {  burgerContainer.className = "burger-container-off";  line.forEach((el, i) => {  el.className = "line close-" + (i + 1);  })  }  } |

Листинг 8 – Реализация бургер меню

|  |
| --- |
| const fs = require("fs");  const express = require('express');  const app = express();  const PORT = 3000;  app.use(express.urlencoded({ extended: false }));  app.listen(PORT);  app.get('/', (req, res) => {  res.sendFile(\_\_dirname + "/public/components/home.html");  })  app.get('/sign\_in', (req, res) => {  res.sendFile(\_\_dirname + "/public/components/sign\_in.html");  })  app.get('/log\_in', (req, res) => {  res.sendFile(\_\_dirname + "/public/components/log\_in.html");  })  app.get('/table\_train', (req, res) => {  res.sendFile(\_\_dirname + "/public/components/table\_train.html");  })  app.post('/log\_in', (req, res) => {  let login\_user = req.body.login\_user;  let password\_user = req.body.password\_user;  let second\_password\_user = req.body.second\_password\_user;  if (login\_user == "" || password\_user == "" || second\_password\_user == "") {  return res.redirect("/home");  } else {  if (second\_password\_user == password\_user) {  let data = fs.readFileSync(\_\_dirname + "/public/xml/users.xml", "utf8");  let str\_to\_arr = data.split('\n');  str\_to\_arr.pop();  let dataUser =  `${str\_to\_arr.join('\n')}  <user>  <login>${login\_user}</login>  <password>${password\_user}</password>  </user>  </users>`  fs.writeFileSync(\_\_dirname + "/public/xml/users.xml", dataUser, (error) => {  if (error) throw error; // если возникла ошибка  console.log("Асинхронная запись файла завершена.");  });  return res.redirect("/sign\_in");  } else {  return res.redirect("/log\_in");  }}  })  app.use(express.static('public')) |

Листинг 9 – Обработка и запись данных о новом пользователе