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

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность 1-40 05 01 «Информационные системы и технологии»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КУРСОВОГО ПРОЕКТА**

по дисциплине «Компьютерные языки разметки»

Тема «Онлайн-школа английского языка»

**Исполнитель**

студент 1 курса 2 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ П.А. Арцыхович

подпись, дата

**Руководитель**

ассистент \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Харланович

должность, учен. степень, ученое звание подпись, дата

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

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

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

дата, подпись

Курсовой проект защищен с оценкой

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Харланович

подпись дата инициалы и фамилия

Содержание

[Введение 3](#_Toc102946432)

[1. Постановка задачи 4](#_Toc102946433)

[1.1 Обзор аналогичных решений. 5](#_Toc102946434)

[1.2 Техническое задание 10](#_Toc102946435)

[1.3 Выбор средств реализации программного продукта 11](#_Toc102946436)

[1.4 Вывод 11](#_Toc102946436)

[2. Проектирование страниц веб-сайта. 12](#_Toc102946438)

[2.1 Создание прототипа. 12](#_Toc102946439)

[2.2 Выбор способа вёрстки. 12](#_Toc102946439)

[2.3 Выбор стилевого оформления. 12](#_Toc102946440)

[2.4 Выбор шрифтового оформления. 13](#_Toc102946441)

[2.5 Разработка логотипа. 13](#_Toc102946442)

[2.6 Разработка пользовательских элементов. 13](#_Toc102946443)

[2.7 Разработка картинок и спецэффектов. 15](#_Toc102946444)

[2.8 Вывод. 15](#_Toc102946445)

[3. Реализация структуры веб-сайта 16](#_Toc102946446)

[3.1. Структура HTML-документа 16](#_Toc102946447)

[3.2. Добавление таблиц стилей Sass и CSS 20](#_Toc102946448)

[3.3. Использование стандартов XML (SVG) 22](#_Toc102946449)

[3.4. Вывод 23](#_Toc102946450)

[4. Тестирование веб-сайта 24](#_Toc102946451)

[4.1. Адаптивный дизайн веб-сайта 25](#_Toc102946452)

[4.2. Кроссбраузерность веб-сайта 25](#_Toc102946453)

[4.3. Руководство пользователя 26](#_Toc102946454)

[4.4. Вывод 26](#_Toc102946455)

[Заключение 27](#_Toc102946456)

[Приложение А. Прототипы веб-сайта 28](#_Toc102946459)

[Приложение Б. Макеты структур веб-сайта 30](#_Toc102946460)

[Приложение В. Листинг HTML-документа 34](#_Toc102946461)

[Приложение Г. Листинг XML (SVG) 37](#_Toc102946462)

[Приложение Д. Листинг JavaScript 39](#_Toc102946463)

Введение

Наличие Интернета в нашей жизни, во многом, облегчает нам жизнь. В той или этой сфере мы часто обращаемся к этому феномену, чтобы получить достоверные знания о том, что нам интересно. В XXI веке овладеть любой информацией не составляет никакого труда. Мировым языком во всем мире на данный момент является именно Английский язык. Поехав в практически в любую страну, мы можем услышать Английскую речь. Но как нам ей овладеть?

Существует несколько способов изучения английского языка:

Первый из них это книги. Книги часто используются как дополнительный материал для более тесного знакомства. Для тех, кто изучает язык самостоятельно, [книги и самоучители](https://skyeng.ru/articles/a-vy-znali-chto-sushchestvuyut-poleznye-tehniki-chteniya/) могут стать превосходным средством освоения английского языка в приемлемом темпе, без ограничений и обязательств, которые присутствуют в других формах обучения.

Второй – это аудио-уроки. Этот метод постепенно устаревает, так как многие приходят к выводу, что используя один лишь этот метод, язык освоить сложно. Однако, хорошие онлайн-курсы английского языка нередко предлагают аудиоматериалы, чтобы дать более полное представление о произошедшем и диалектах.

Ну и наконец, по моему мнению, наиболее эффективный способ изучения английского языка — онлайн школы. Метод онлайн-обучения хорошо зарекомендовал себя как среди преподавателей, так и среди изучающих язык. Обладая гибкостью и удобством, он хорошо подойдет тем, кто не может поддерживать график занятий в классе, но умеет работать в группах и индивидуально. Однако, этот метод предполагает, что обучающийся имеет базовые навыки работы в интернете.

Имея представление о различных методах изучения английского, вы сможете выбрать наиболее подходящий для вас

Все вышеперечисленные факты отражают актуальность тематики курсовой работы.

Цель курсовой работы: разработать веб-сайт онлайн-школы с использованием HTML5 и XML, а также с применением Sass/CSS3.

Задачи:

— 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.

# 1 Постановка задачи

# 1.1 Обзор аналогичных решений

Курсовой проект представляет из себя веб-сайт онлайн-школы английского языка. В данной сфере огромное количество аналоговых продуктов, это связано с популярностью данной темы. В этом разделе будут приведены веб-сайты существующих онлайн-школ.

# 1.1.1 Сайт skyeng.

Сайт имеет приятную для глаз цветовую палитру. Сайт динамический, интерактивный. Дизайн представлен на рисунках ниже. Данный сайт написан на таких языках как JS, PHP, а также подключение React JS, в особенности на React JS написана форма для заполнения номера, которая представлена на рисунке 1.4. На рисунке 1.1 представлена шапка(header) данного сайта, на котором довольно удобно прорисовано меню, а также отличный логотип сайта. На рисунке 1.2 видны услуги, которые может приобрести пользователь.

На Рисунке 1.1 Представлен header, а также навигация сайта shyeng.

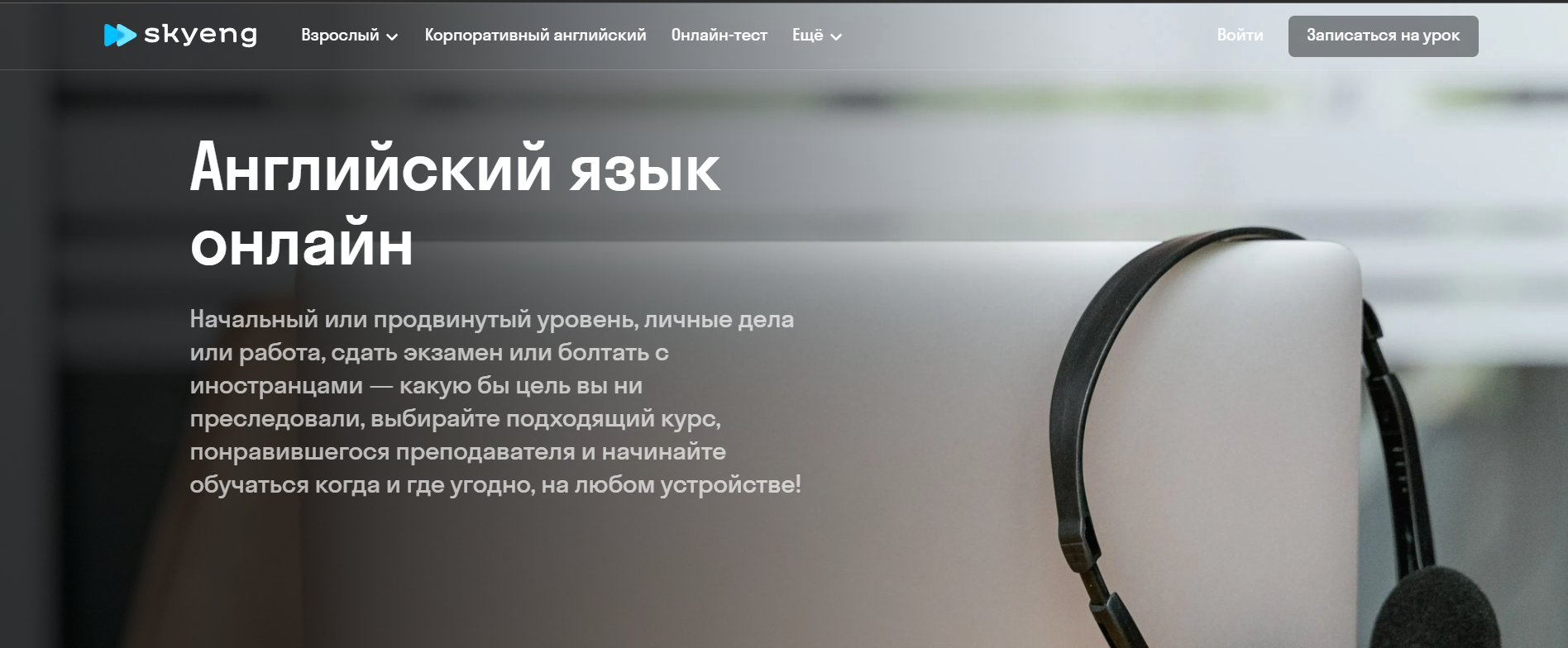


Рисунок 1.1 – Шапка сайта «skyeng»

Мобильная версия сайта также проработано с высоким качеством. На данном сайте хорошо проработана адаптивность. На первый взгляд бросается в глаза яркий фон цветов. На рисунке 1.3 представлены услуги на то, что предоставляет сайт. На рисунке 1.4 представлена форма заполнения личных данных, написанный на React JS. На рисунке 1.2 можно заметить список услуг, обычно данный элемент помещают в средней части сайта, для того, чтобы когда пользователь заинтересуется первой частью сайта, его палитрой цветов, сочетаемости элементов, была возможность ознакомиться с наиболее важной информацией, которая располагается обычно в средней части сайта, чтобы не терять уникальности.

На рисунке 1.2 представлен спектр услуг. Услуги обычно располагаются в средней части сайта, это связано с тем, что пользователь который заинтересуется сайта сразу смог бы приобрести услугу.

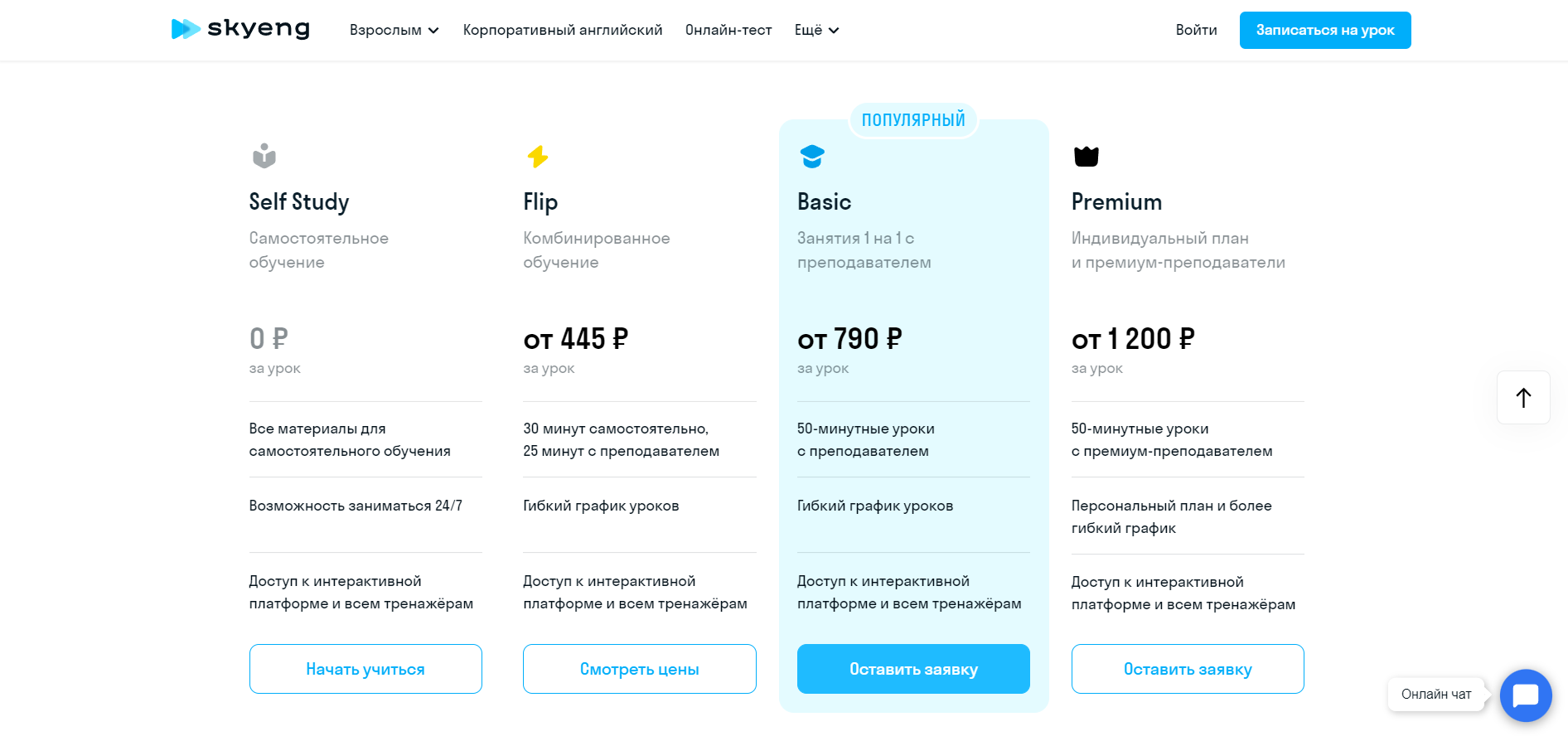


Рисунок 1.2 – Услуги «skyeng»

На рисунке 1.3 в мобильной версии будет предоставлен список услуг предоставляемый сайта. Очень важно, чтобы данная информация была полезна пользователю. Именно тогда пользователь для себя выберет правильный для него курс.

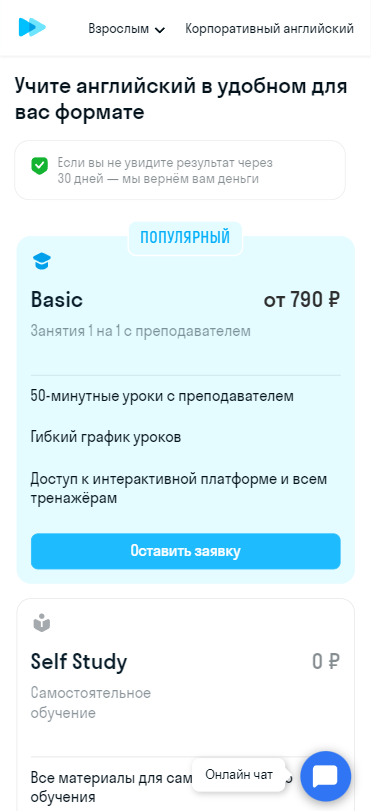


Рисунок 1.3 «Услуги в мобильной версии сайта»

На рисунке 1.4 представлена форма регистрации, на которой пользователь может оставить заявку на прохождение курсов. Такую форму обычно делают на библиотеке React — JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов. React разрабатывается и поддерживается Facebook, Instagram и сообществом отдельных разработчиков и корпораций. React может использоваться для разработки одностраничных и мобильных приложений. Цвета подобраны в наиболее яркой манере .На сайте преобладают оранжево-белые цвета.

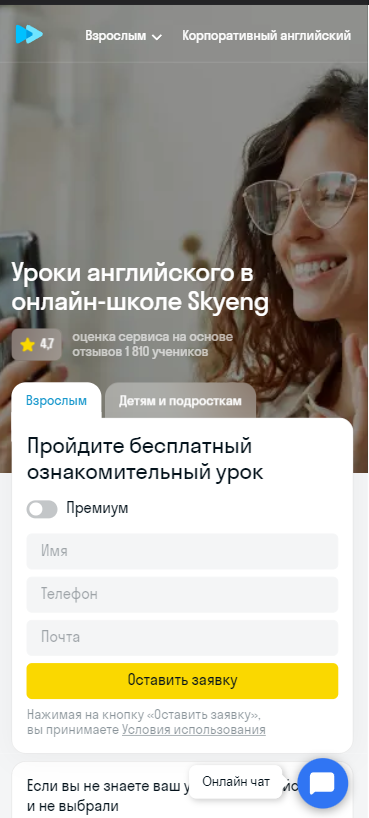


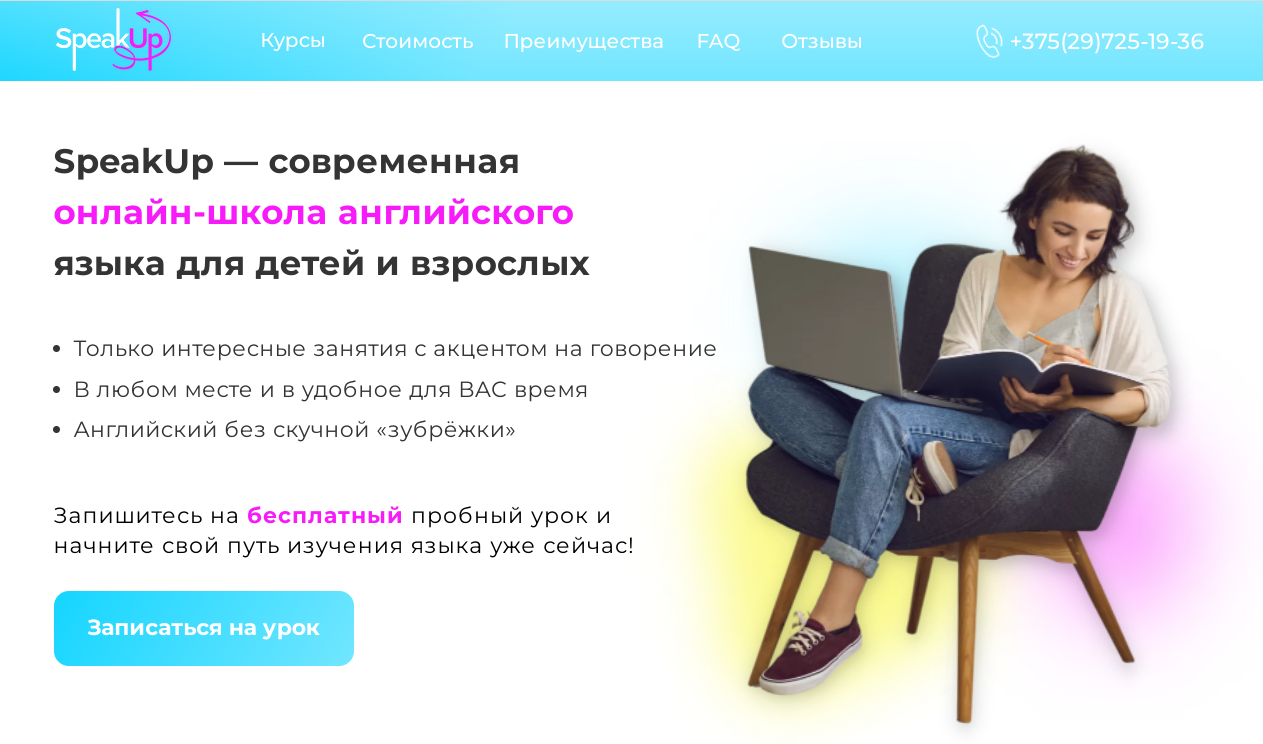
Рисунок 1.4 «Форма регистрации»

Сайт «skyeng» отлично демонстрирует сочетание цветов и отличный функционал.

# 1.1.2 Сайт speakup.by.

В данном экземпляре была подчеркнула внятность сайта, которые сразу дает понять, то, что будем изучено. Яркие фона цвета дают сразу понять, что данный экземпляр даст мягкость между пользователем и исполнителем. На рисунке 1.5 представлена меню, а также информация о сервисе, и кнопка записи на урок. На рисунке 1.6 предоставляется спектр услуг, который подойдет практически каждому пользователю интернета. На рисунке 1.7 дается информация о том, почему пользователь должен выбрать именно данный сайт. Футер сайта предоставлен на рисунке 1.8.

На рисунке 1.5 представлена шапка канала «SpeakUp», в которой в основном представлены голубой и белые тона. А также красивые подпункты

Рисунок 1.5 «Шапка канала SpeakUp»

На рисунке 1.6 представлен спектр услуг для сайта SpeakUp. Опять же, отличный дизайн придают уникальность данному образцу.

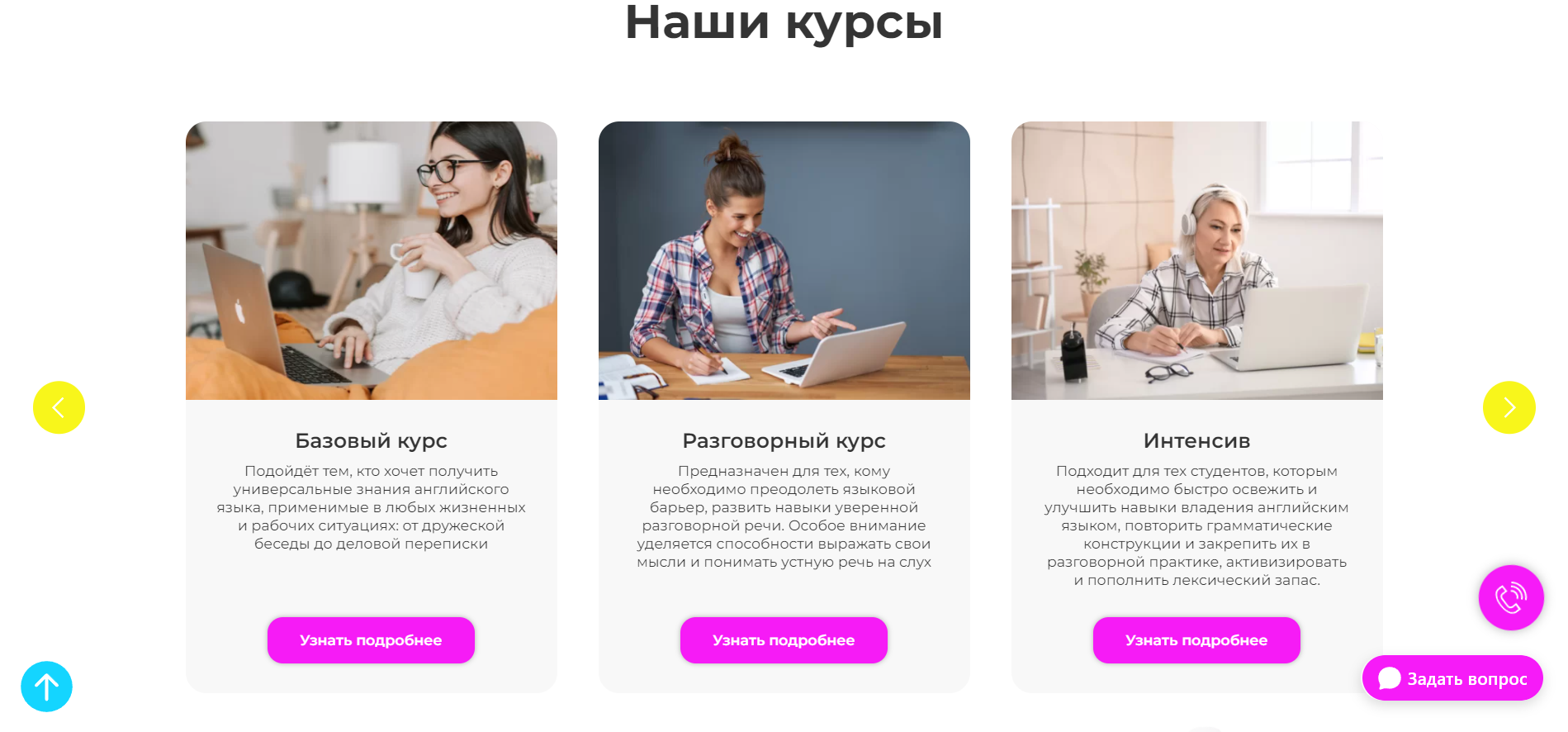


Рисунок 1.6 «Услуги сайта SpeakUp»

Тон данного сайта придает свою уникальность.

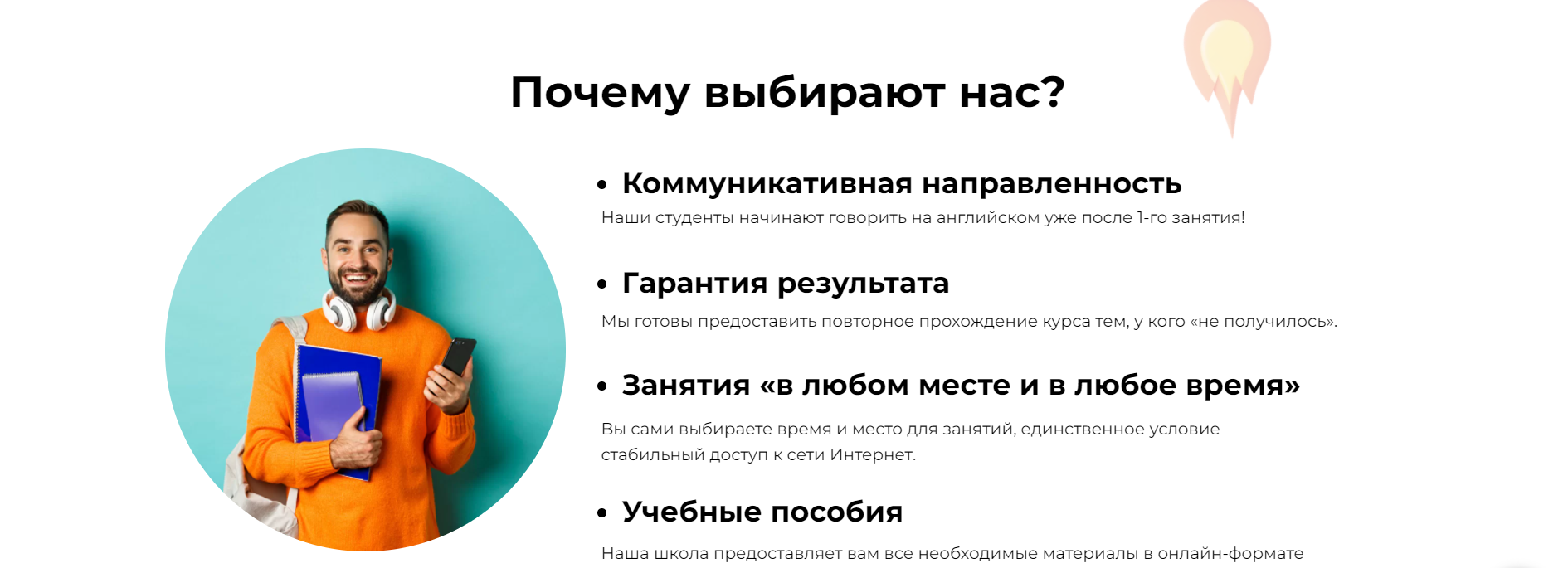
На рисунке 1.7 предоставлен раздел о нас. 

Рисунок 1.7 «О сайте SpeakUp»

На рисунке 1.8 представлена форма регистрации сайта “SpeakUp”. А также нижняя часть данного сайта. В большинстве преобладают голубая и белая палитра, которая придают сайту свою динамичность, яркостью, которая сразу бросается в глаза. Иконки, отображаемые в нижней части сайта сделаны гармонично, также проработаны панели звонков. А также с первых секунд в глаза бросается логотип данного сайта.

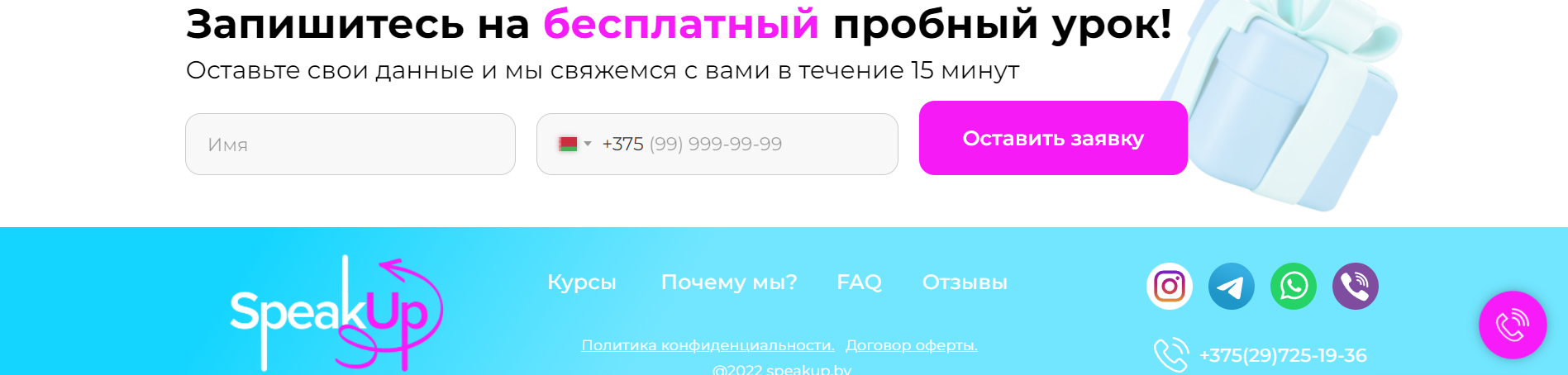


Рисунок 1.8 «Форма регистрации и нижняя часть сайта SpeakUp.

В мобильной версии, также бросается в глаза понятность уже с первой страницы. Наиболее структурированная информация, одно из главных качество сайта. Данный сайт явное представления структурированной информации. Пользователю с первого взгляда должно стать понятно о чем сайт. Обычно для заинтересованности пользователей, вводятся именно такие тона, которые большинству пользователей с первой секунды настраивают на приобретенные положительные эмоции во время серфинга данного сайта. На рисунке 1.9 предоставлена информация о формах обучения на данном сайте. Основная информация сайта, которая сразу бросается в глаза, его структура предоставлена на рисунке 1.10.

Формы обучения на данном сайте представлены ниже.

На рисунке 1.9 информация о формах обучения сайта “SpeakUp”. В основном на сайте преобладает именно белая палитра цветов, которая в свою очередь придают сайту свою динамичность, яркостью, которая сразу бросается в глаза. Интересный факт о белой палитре. Визуально белые (и светлые) предметы кажутся нам легче, чем равные им по весу черные (и темные). Солнце на самом деле белое, если смотреть из космоса. Атмосфера Земли искажает его цвет, поэтому люди видят его желтым. В политике белый цвет – цвет нейтралитета, традиционности и научного подхода.

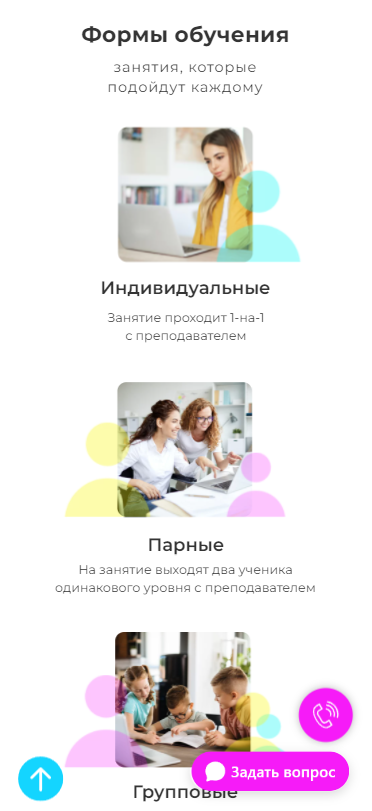


Рисунок 1.9 «Мобильная версия сайта SpeakUp»

Сайт «SpeakUp» отлично демонстрирует совмещение функционала и дизайна.

# 1.2 Техническое задание

В данном проекте предоставляется возможность создать одностраничный («лендинг») сайт онлайн-школы английского языка. Сайт должен содержать общую информацию о школе, такую как контактные данные, описание услуг, а также методы изучения языка. Основным контентом являются текстовые данные, описания товаров, фотографии товаров.

Главной задачей сайта является предоставление данных о услугах для привлечения внимания потенциальных клиентов. В перспективе сделать данный “лендинг” наиболее динамически быстрым, для того, чтобы пользователь имел возможность в короткие сроки получить наиболее важную для него информацию. Сайт должен быть интерактивным, содержать актуальную информацию. Данный сайт предназначен для желающих приобрести достоверные знания в сфере английского языка.

На главной странице меню будет располагаться сверху. В меню будут предоставляться ссылки на основную информацию (Skills (навыки и умения), About us(о нас), Pricing(цены) Contacts (контакты).

В мобильной версии (ширина 320px) страница должна адаптироваться под ширину устройства. Начиная с ширины в 1000px должна открываться версия для персонального компьютера.

Данный лендинг будет предоставлять достоверную информацию о изучении английского языка, работать он будет по принципу “якоря”, в шапке страницы будет предоставлены “якоря” на последующие части страницы. А именно: “Home”, “Slills”, “About us”, “Pricing”, “Contacts”. А также основной логотип “Britlex”, и флаг логотип в виде флага Великобритания.

Основные требования к исполнителю: создание дизайна, проведение анализа существующих решений, создание прототипа, адаптивность и кроссбраузерность сайта (корректное отображение в браузерах, таких как Google Chrome, Mozilla Firefox, Microsoft Edge, Opera.). В проекте используется гипертекстовый язык разметки HTML, CSS/Sass, графические элементы в формате SVG; данные хранятся в XML-формате.

Задачи курсового проекта:

— упростить поиск информации о онлайн-школе для пользователей;

— размещение основной информации на сайте;

— размещение информации о услугах;

— представление имиджа школы;

# 1.3 Выбор средств реализации программного продукта.

Для реализации проекта были выбраны языки: HTML, CSS/Sass, XML, (JavaScript). Структура сайта создана с помощью языка разметки HTML, дизайн сайта оформлен с помощью CSS/Sass. В XML формате хранятся данные сайта. JavaScript позволит добавить динамичности сайта, путем введения функций и событий. Курсовая работа выполняется в редакторе кода VS Code.

# 1.4 Вывод

Чтобы создать качественный веб-сайт, который с первого взгляда привлечет внимание пользователей, пришлось добыть качественные аналоги, проанализировать различные стили, сочетаемость цветов. Были поставлены конкретные задачи для создания интерактивного информационного ресурса. Кроме того, были рассмотрены средства реализации программного продукта, такие как языки разметки, а также редактор кода VS Code. Были проанализированы особенности языков и основные дополнительные средства редактора. Все вышеперечисленные пункты понадобятся для достижения поставленной задачи и для создания в дальнейшем качественного продукта.

# 2 Проектирование страниц веб-сайта

# 2.1 Создание прототипа.

Прототип сайта — это схема всех или нескольких страниц сайта в виде наброска, эскиза или html-документа, где отображены взаимодействия и структурные элементы будущего сайта: меню, кнопки, формы и другие. Полный прототип данного сайта будет представлен в [приложении А](#ПР_1). В основе картинок в прототипе будет выступать перечеркнутый прямоугольник. Прототип данного сайта составлялся на основе сайтов «skyeng» и «speakup». Стиль для создания прототипа был выбран «ретро-стиль» особенно которого заключается минимализм, а также черно-белые тона. В 21 веке выглядит очень стильно. Удобство данного прототипа состоит в том, что мы можем посмотреть основную часть сайта, без использования картинок и других составляющих

# 2.2 Выбор способа вёрстки.

В соответствие с требованиями к проекту, в частности адаптивность и кроссбраузерность сайта, было принято решение об использовании Sass — это метаязык на основе CSS-кода для увеличения уровня абстракции и упрощение файлов каскадных таблиц стилей. Язык Sass — отличается отсутствием фигурных скобок, а также адаптированный метаязык. Требует компиляцию и установку Sass легко приспособить к любому дизайну и пользовательскому интерфейсу.

Для создания отдельных блоков страницы, такие как горизонтальное меню, будет использована Flexbox-верстка. С помощью этой технологии можно очень просто и гибко расставить элементы в контейнере, распределить доступное пространство между ними, и выровнять их тем или иным способом даже если они не имеют конкретных размеров. CSS Flexbox поддерживается всеми используемые на сегодняшний момент современными браузерами (с использованием префиксов: IE10+, Edge12+, Firefox 2+, Chrome 4+, Safari 3.1+, Opera 12.1+, iOS Safari 3.2, Opera mini, Android 2.1+, Blackberry 7+).

Исходя из выбора способа верстки были расположены все основные элементы сайта и разработаны макеты страниц. Макеты смотреть в приложении 1.

# 2.3 Выбор стилевого оформления.

Для реализации стилевого оформления сайта было выбрано минималистическое направление с элементами ретро стилей в веб-дизайне. Такой стиль позволяет презентовать качественный продукт(онлайн-школы), а также привлечь внимание к нестандартного рода фотографиям.

Сайт не загромождается лишними деталями, большим количеством спецэффектов. Акцент на сайте ориентируется на основной контент и фотографии товаров.

Причем сами фотографии выглядят очень стильно и оригинально.

Большинство изображений выдержаны в одном стиле, не перегружены мелкими элементами.

Цветовые решения не яркие, довольно темные. Основные цвета преобладающие в проекте: rgb(186, 181, 181) и rgb(5, 4, 4).

# 2.4 Выбор шрифтового оформления.

Основным шрифтом был выбран Poppins. Шрифт будет подключен с помощью библиотеки Google Fonts.

Шрифт Poppins является вариативным шрифтом и относится к категориям гротески, декоративные. Гарнитура содержит 20 файлов. Поддерживает 57 языков. Имеет лицензию SIL OFL 1.1. Разработка шрифта Poppins велась Jonny Pinhorn и Ninad Kale в студии типографики Indian Type Foundry.

# 2.5 Разработка логотипа.

Разработка логотипа осуществлялась в графическом редакторе Figma.X. ссылку на данный сайт можно увидеть в списке литературы [1] Дизайн довольно прост. Логотип представляет из себя название онлайн-школы английского языка (Briltex). Казалось бы, довольно просто, но часто именно минимализм залог хорошего логотипа. В данном случае составленный дизайн обозначает: Brit – британец. А lex – лексика.



Рисунок 2.1 «Логотип»

Логотип построен в ретро стиле (преобладают черно-белые цвета). Максимально простой, но простой не значит плохой. Простота придает свой стиль данному логотипу.

# 2.6 Разработка пользовательских элементов.

В проекте будут представлены такие элементы пользовательского интерфейса, как флаг Великобритании и svg лампочка для улучшения декорации на сайте.

Элементы флага и лампочки представлены на рисунке 2.2, рисунке 2.3.

Данный экземпляр был добавлен с целью прохождения теста, который будет доступен при нажатии на флаг. Лампочка была придумана с целью упрощения задачи пользователей изменять палитру цветов с черно-белого, на бело-черный.



Рисунок 2.2 « Флаг Великобритании»

Данный экземпляр создан для того, чтобы при нажатии на него, изменялся тон страницы(темная и светлая тема)



Рисунок 2.3 «Лампочка смены цвета»

На сайте также будет представлен svg значок валюты “Евро” представленный на рисунке 2.4



Рисунок 2.4 «Рисунок Евро»

На рисунке 2.5 представлен список доступных услуг, которые предоставляет исполняемое лицо. В каждой услуге элемент представленный на рисунке 2.4. А также картинки нарисованные с помощью векторов.

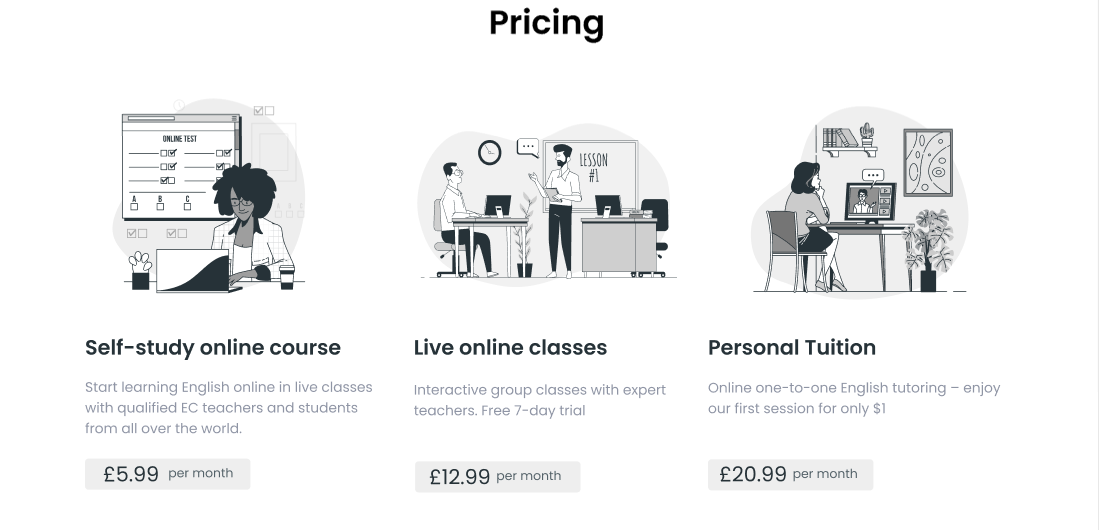


Рисунок 2.5 «Услуги на сайте»

Минимализм придает данному сайта свою «вишенку на торте». Минимализм (англ. minimalism от лат. minimus — наименьший) — стиль в дизайне, характеризующийся лаконичностью выразительных средств, простотой, точностью и ясностью композиции. Отвергая классические приёмы творчества и традиционные художественные материалы, минималисты используют промышленные и природные материалы простых геометрических форм, нейтральных цветов (чёрный, серый) и малых объёмов.

# 2.7 Разработка картинок и спецэффектов.

На странице веб-сайта будут разработаны динамические эффекты и анимации, которые позволят придать веб-сайту интерактивность и сделают его более привлекательным для пользователя. Для создания данных спецэффектов, будет использована библиотека AOS. И применен скрипт AOS.init().

На сайте будут разработаны некоторые картинки, было запланировано сделать картинки в строго ретро-стиле, а именно черно-белый стиль, который придаст эпичности и минимизации сайту. Такого типа картинки будут присутствовать на сайте на всех этапах реализации.

Примеры представлены на рисунке 2.6

На рисунке 2.6 представлен список навыков, которые будут затронуты в ходе обучения.

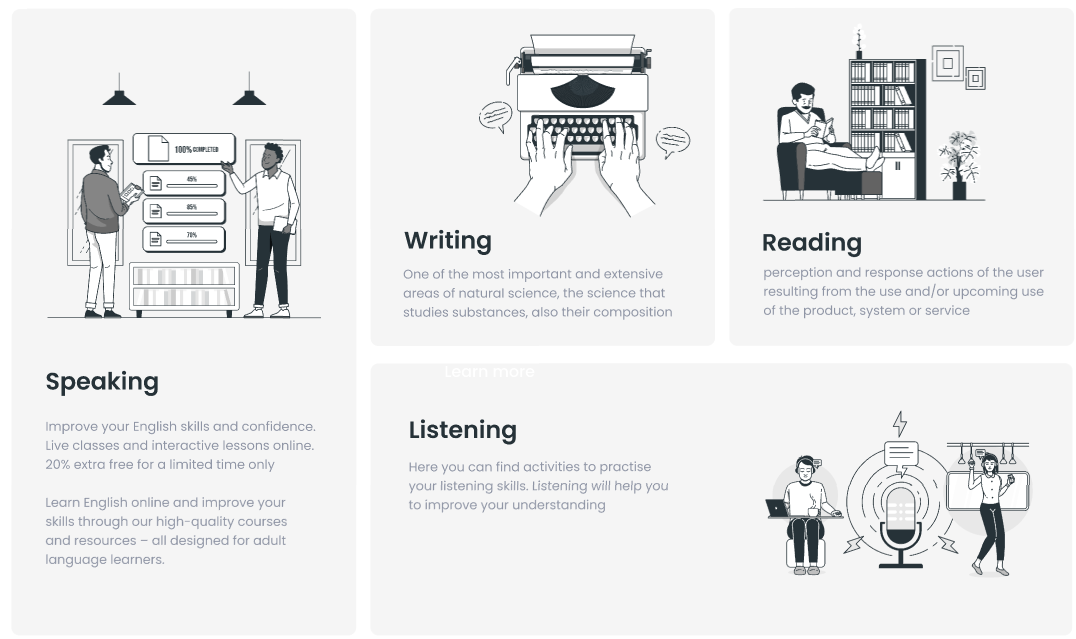


Рисунок 2.6 «Список приобретаемых навыков»

Картинки были выбраны в одном стиле, чтобы подчеркнуть минимализм.

# 2.8 Вывод.

На данном этапе был создан макет сайта для облегчения последующей верстки. Было определено стилевое оформление сайта, то есть основная цветовая гамма в ретро стиле , шрифтовое оформление, элементы пользовательского интерфейса придающие динамики страницам. Данный стиль отлично сочетается с данной тематикой, так как английский язык пришел к нам из древности, и картинки часто были черно-белые. Именно такая стилистика подчеркнет данную атмосферу. Натолкнёт пользователя на изучение центрального языка мира.

# 3 Реализация структуры веб-сайта

# 3.1. Структура HTML-документа

HTML-документ будет определять базовую структуру для будущего веб-сайта. Код выполнен по стандарту HTML5.

HTML (Hypertext Markup Language) - это код, который используется для структурирования и отображения веб-страницы и её контента. Например, контент может быть структурирован внутри множества параграфов, маркированных списков или с использованием изображений и таблиц данных.

Структура тела документа состоит из семантических тегов header, section, main, и footer. Листинг представлен в [приложении 3](#ПР_3).

Семантические теги — 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. Политика спецификации HTML5 заключается в том, чтобы каждый HTML-тег имел свою семантику.

HTML-элемент <header> представляет собой вводный контент, обычно группу вводных или навигационных средств. Он может содержать другие элементы-заголовки, а также логотип, форму поиска, имя автора и другие элементы.

Тег section задаёт раздел документа, может применяться для блока новостей, контактной информации, глав текста, вкладок в диалоговом окне и др. Раздел обычно содержит заголовок. Допускается вкладывать один элемент.

HTML-элемент <main> предназначен для основного контента (содержимого) <body> документа (страницы). Основной контент состоит из контента, который непосредственно относится к главной теме документа или её развивает.

HTML-элемент <footer> представляет собой нижний колонтитул (футер, подвал) для своего ближайшего секционного контента или секционного корня. Футер обычно содержит информацию об авторе раздела, информацию об авторском праве или ссылки на связанные документы.

В теге header находится логотип, флаг Великобритании с тестом по мировому языку, а также горизонтальная навигационная панель. Она содержит элементы перехода на все основные разделы сайта. Навигационное меню было реализовано при помощи ссылок, переход на отдельные части страницы выполнен с помощью JavaScript. Листинг представлен в [приложении 3](#ПР_3), а также в листинг 3.1

В тегах section находятся отдельные элементы сайта, такие как: Skills, About Us, Pricing, Contact Us. Листинг представлен в [приложении 3](#ПР_3), а также в листинг 3.3.

В теге main находиться главная часть страницы, которая загружается пользователю при открытии сайта, на которой присутствует слайдер флагов с формой регистрации (кнопка Get Started), а также тест на знания английского языка, который располагается под флагом. Листинг представлен в [приложении 3](#ПР_3).

Для верстки была выбрана flexbox верстка. CSS Flexbox предназначен для создания гибких макетов. С помощью этой технологии можно очень просто и гибко расставить элементы в контейнере, распределить доступное пространство между ними, и выровнять их тем или иным способом даже если они не имеют конкретных размеров.

В верхней части сайта «header» присутствует тег <div class="wrapper"> данный тег основообразующий тег, для длины и ширины обертки данной части страницы. В данном теге присутсвует такие теги как <div class="Britlex">, <div class="nav">, <div class="flag">, а также <div class="burger"> для создания «бургер-меню в мобильной версии». В <div class="Britlex"> располагается логотип сайта, в котором заключен тег <a href="kurs.html">, который будет перезагружать страницу при нажатии на логотип. ». В <div class="nav"> располагается навигационное меню, которое состоит из представляет собой маркированный список, при нажатии на элементы которого пользователя будет переносить на ту часть страницы, которая ему интересно. Сами же элементы будут подгружаться от XML. Простейшая структура начала XML: <?xml version="1.0" encoding="UTF-8"?> <?xml version="1.0" encoding="UTF-8"?> <root>. Элементы будут помещаться в класс «none» и приобретать имена «Home, Skills, About us, Pricing. Contacts». Что касается флага, то здесь будет тег <a href="#popup">, который включает в себя картинку, при нажатии на которую будет вылетать модальное окно. Модальные (всплывающие) окна – это очень популярный элемент интерфейса современных сайтов. Оно может использоваться для вывода различного контента веб-страниц такого, например, как формы (обратной связи, регистрации, авторизации), блоки рекламной информации, изображения, уведомления и др. Модальные окна реализованы таким образом, что основополагающим тегом будет являться <div id="popup" class="popup">, в котором будет присутствовать ссылка <a href="##" class="popup\_area"> при нажатии на которую и должно появляться модальное окно. Тег <div class="popup\_content "> отвечающий за содержимое модального окна( в данном случае тест по английскому языку) , а также ссылка<a href="#" class="popup\_close"> X </a> для закрытия модального окна. Реализация теста по английскому языку происходило с помощью тега <form> для тестовой части создавались вопросы с помощью маркированного списка. Выбор правильного ответа реализовывался с помощью тега <input type=”radio” name=”1”> c типом radio для выбора радиокнопок, а также одинаковым именем для того, чтобы выбор радиокнопок не повторялся на других элементах.

|  |
| --- |
| <header>  <div class="wrapper">  <div class="Britlex">...  </div>  <div class="nav">...  </div>  <div class="flag">...  </div>  <div class="burger">  <div class="container" id="burger">...  </div>  </header> |

Листинг 3.1 – Структура тега header

Основной блок сайта открывает тег <main class="main">, в котором задан тег <div class="wrapper"> для установки размеров страницы. Тег <div class="flex-count"> отвечающий за то, что все происходящее далее будет выполняться во flexbox контейнере. Тег <div class="Learn"> отвечает за общий блок, в котором будет присутствовать все на данной части страницы. Тег <div class="Text\_Main"> будет отвечать за изображение текста, а также svg картинки вставленной в данный блок. Теги <div class="undertext"> и <button class="button\_1"> отвечают за кнопку, а также подтекст присутствующий в основном блоке, который следует после тега <header>.

|  |
| --- |
| <!--Основной блок сайта-->  <main class="main">  <div class="wrapper">  <div class="flex-count">  <div class="Learn">...  </div>  <div class="Country ">...  </div>  </div>  </div>  </main> |

Листинг 3.2 – Структура тега main

Следующую часть сайта будет представлять секция, которая задана тегом <section class="section\_skils">. В ней присутствуют такие основополагающие теги как: <div class="wrapper">, <div class="Wrapper\_Skills"> (для всех навыков, которые будут отображаться в данной секции), <div data-aos="fade-right" class="Speaking"> (тег благодаря которому будет отображаться блок «Speaking»). Data-aos — атрибут в библиотеки для создания того, какие анимации мы будем видеть при «scroll» страницы. Тег <div data-aos="fade-left" class="flex-column"> со значением класса «flex-column» для того, чтобы последующие элементы отображались строго в столбик. Тег <div class="flex\_one"> отвечает за создание первого «flex» элемента. Тег <div class="Reading"> для создания блока отвечающего за раздел «Чтения» в нашей части страницы. Тег <div class="flex\_last"> отвечает за создание последнего flex элемента на страницы. В основе данных тегов лежат картинки и подтекст, располагающийся под картинками.

Следующую часть сайта представляет тег <div class="about\_us"> с оберткой сделанной под размеры <div data-aos="flip-left" class="wrapper">. Тег <div class="block\_text\_about\_us"> отвечает за текст в блоке «About Us». Тег <div class="about\_static"> будет отвечать за статичную часть сайта, а именно подписи под основным текстом. <div data-aos="flip-right" class="about\_img"> создан для вставления картинки в данную часть блока.

Секция <section data-aos="zoom-in" class="section"> будет отвечать за часть сайта, в которой будут определены цены на услуги, который предоставляет сайт. Тег <div class="Price\_wrap"> отвечает за обертку, которая предоставляет блоки в которых представлены целых. Общая структура каждого из них выглядит: <div class="block\_one"> и структура тех услуг, которые представлены на странице.

Секция <section class="contact\_section"> отвечает за часть сайта, в которой реализованы контакты, для связи с пользователем. Тег <div class="Contact\_Us"> отвечает за отображение всех данных на странице контактов, а именно текста, и кнопку с помощью которой пользователя перенаправит на форму регистрации. Тег реализованный для кнопки: <div class="input\_plus\_but">.

|  |
| --- |
| <section class="section\_skils">  <div class="wrapper">  <div class="Wrapper\_Skills">  <div data-aos="fade-right" class="Speaking">...  </div>  <div data-aos="fade-left" class="flex-column">...  </div>  </div>  </div>  </section>  <section data-aos="zoom-in" class="section">  <div class="wrapper">  <div class="Price\_wrap">...  <div class="block\_one">...  </div>  <div class="block\_one">...  </div>  <div class="block\_one">...  </div>  </div>  </div>  </section> |

Листинг 3.3 – Структура тега section

Нижняя часть сайта представлена тегом <footer class="foot"> обертка для всех flex элементов будет представлена тегом <div class="flex-footer">. В самом низу сайта будет располагаться логотип, при нажатии на который будет происходить переход на начальную часть страницы, а также политика конфиденциальности. Которая, опять же, будет реализована с помощью модальных окон. Структура одного из них выглядит следующим образом:<a class="text\_footer" href="#popup\_1">.

|  |
| --- |
| <!--Нижняя часть сайта-->  <footer class="foot">  <div class="wrapper">  <div class="flex-footer">  <div class="Britlex">  Brit<span class="gray">lex</span>  </div>  <div class="text\_footer">  <p> Terms and Conditions </p>  <p>  • Privacy Policy </p>  <p>  • Cookie Policy</p>  </div>  </div>  </div>  </footer> |

Листинг 3.4 – Структура тега footer

Таким образом была показана реализация структуры веб-сайта, а также демонстрация структуры в листинге.

# 3.2 Добавление таблиц стилей Sass и CSS

Добавление таблиц стилей Sass(Scss) и Css позволило упростить создание стилистического оформления страницы. Компиляция осуществлялась благодаря open-source приложению Preprose. Преимуществами данного приложения являются компиляция файлов sass, less, scss и т.д. Приложение не зависит от того, установлены ли в системе соответствующие программы и библиотеки (sass, compass и т.п). Обновление страниц в браузере в реальном времени и конечно http-server. При активированной опции автокомпиляции после каждого сохранения файла программа выводит информацию о проведенной обработке файлов.

CSS — формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Также может применяться к любым XML-документам, например, к SVG или XUL.

При верстке было выбрано использовать CSS3. Основное различие между CSS и CSS3 заключается в том, что в CSS3 есть модули. CSS является базовой версией и не поддерживает адаптивный дизайн. CSS3, с другой стороны, является последней версией и поддерживает адаптивный дизайн. CSS нельзя разбить на модули, но CSS3 можно разбить на модули.

Sass — модуль, включенный в Haml. Sass — это метаязык на основе CSS, предназначенный для увеличения уровня абстракции CSS-кода и упрощения файлов каскадных таблиц стилей. Язык Sass имеет два синтаксиса: sass — отличается отсутствием фигурных скобок, в нём вложенные элементы реализованы с помощью отступов. Ссылку на источники информации о Sass можно увидеть в списке литературы [2]

Главными преимуществами Sass(Scss) перед Css это прежде всего автоматическая кроссбраузерность при компиляции Sass. Наличие вложенности в отличие от Css. Например, Sass облегчает применение свойств благодаря миксинам. Они позволяют один раз создать набор правил, чтобы потом использовать их многократно или смешивать с другими правилами. Переменные, управлять которыми намного проще в отличие от Css. А также импорт, Sass делает это без HTTP-запросов. Конкретно преимущества рассмотрим в листинге 3.5.

Что касаемо адаптивности, которую сайт приобрел в ходи использвания медиа запросов представлено в листинге 3.6.

Рассмотрим преимущества на явном примере вложенности, которая присутствует в CSS и Scss. В Scss мы можем заменять родительский элемент знаком &. Также как и в объявлении ссылки на родителя, оператор & в SassScript относится к текущему родительскому селектору. Это список, разделенный запятыми, состоящий из списков, разделенных пробелами. Например:

|  |  |
| --- | --- |
| Css | Sass(Scss) |
| .main\_slider {  margin-top: 20px;  }  .get\_started .main\_slider .containers {  margin: 0 auto;  width: 1000px;  max-width: calc(100% - 30px);  position: relative;  height: 450px;  overflow: hidden;  }  .get\_started .main\_slider .slides {  display: -webkit-box;  display: -ms-flexbox;  display: flex;  height: 100%;  }  .get\_started .main\_slider .slide {  min-width: 100%;  position: relative;  }  .get\_started .main\_slider .slide-controls {  position: absolute;  top: 50%;  left: 0;  -webkit-transform: translateY(-50%);  transform: translateY(-50%);  width: 100%;  display: -webkit-box;  display: -ms-flexbox;  display: flex;  -webkit-box-pack: justify;  -ms-flex-pack: justify;  justify-content: space-between;  -webkit-box-align: center;  -ms-flex-align: center;  align-items: center;  }  вложенность | &.main\_slider {  margin-top: 20px;  .containers {  margin: 0 auto;  width: 1000px;  max-width: calc(100% - 30px);  position: relative;  height: 450px;  overflow: hidden;  }  .slides {  display: flex;  height: 100%;  }  .slide {  min-width: 100%;  position: relative;  }  .slide img {  object-fit: cover;  width: 100%;  height: 100%;  }  .slide-controls {  position: absolute;  top: 50%;  left: 0;  transform: translateY(-50%);  width: 100%;  display: flex;  justify-content: space-between;  align-items: center;  }    вложенность |

Листинг 3.5 – Сравнение CSS с Scss

Для тонкой настройки элементов во время уменьшения экрана были использованы медиа запросы.

|  |
| --- |
| @media (max-width: 1062px) {  .wrapper {  max-width: 800px;  }  .main .wrapper .Country {  width: 100%;  }  .main .wrapper .Country .img\_main {  width: 100%;  display: table;  margin: 0 auto;  }  .main .wrapper .flex-count {  display: -webkit-box;  display: -ms-flexbox;  display: flex;  width: 100%;  -webkit-box-pack: center;  -ms-flex-pack: center;  justify-content: center;  -webkit-box-orient: vertical;  -webkit-box-direction: reverse;  -ms-flex-direction: column-reverse;  flex-direction: column-reverse;  }  .Learn .Text\_Main {  text-align: center;  -webkit-box-orient: vertical;  -webkit-box-direction: normal;  -ms-flex-direction: column;  flex-direction: column;  }  .Learn .Text\_Main img {  width: 30%;  margin: 0 auto;  max-width: 30%;  }  .Learn .undertext {  font-size: 35px;  }  .Learn .button\_1 {  display: block;  text-align: center;  width: 100%;  }  .Skil,  .Text\_price {  margin-top: 0px;  }  } |

Листинг 3.6 – Листинг медиа запросов

В данной структуре используется внешнее подключение таблиц стилей через тег link, так как это самый мощный и удобный способ определения стилей и правил для веб-сайта. Стили хранятся в отдельном файле, который может быть использован для любых веб-страниц. В таблицах стилей используются селекторы по классу, по идентификатору, универсальный селектор, а также селекторы потомков, псевдоклассы и псевдоэлементы.

# 3.3. Использование стандартов XML (SVG)

В качестве хранилища данных (горизонтальная панель навигации) использовался формат XML. XML – расширяемый язык разметки. Используется для хранения и передачи данных. В качестве формата расширения картинок, был выбран формат SVG. SVG – язык разметки масштабируемой векторной графики, предназначен для описания двумерной векторной и смешанной графики в формате XML.

Пример подключения XML можно найти в источнике списке литературы [3]

SVG — язык разметки масштабируемой векторной графики, созданный Консорциумом Всемирной паутины и входящий в подмножество расширяемого языка разметки XML, предназначен для описания двумерной векторной и смешанной векторно/растровой графики в формате XML.

Пример подключения xml:

|  |
| --- |
| var xmlhttp = new XMLHttpRequest();  var path = " data.xml"; xmlhttp.open("GET", path, false); xmlhttp.send(); var xmlDoc = xmlhttp.responseXML; var  x = Array.from(xmlDoc.getElementsByTagName("ul")); let  wrapper = document.querySelector(".navbut"); wrapper.innerHTML = ""; x.forEach(item => {  var div = document.createElement("div");  div.innerHTML = `  <li class="marg">  <a class=${item.querySelector("class").textContent}>  ${item.querySelector("name").textContent}  </a>  </li>`  wrapper.innerHTML += div.innerHTML;  }) |

Листинг 3.6 – Подключение XML

# 3.4 Вывод

На данном этапе была реализована структура на HTML, а также созданы таблицы стилей CSS/Sass. Было продемонстрировано использование стандарта XML (SVG), а также языка программирования JS(JavaScript — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений.). Было определено стилевое оформление веб-сайта, созданы анимации, а также веб-сайт был подготовлен к следующему этапу – к тестированию.

# 4 Тестирование веб-сайта

# 4.1 Адаптивный дизайн веб-сайта

На данный момент количество пользователей, которые заходят в Интернет с мобильных устройств очень велико, и с каждым днем их становится все больше. А потому адаптация веб-сайта под маленькие экраны – актуальная задача и для курсового проекта.

При создании сайта, была поставлена задача: создать сайт, который будет сам подстраиваться под разные типы экранов. Главными условиями адаптивности является нефиксированный размер элементов страницы.

# 4.2 Кроссбраузерность веб-сайта

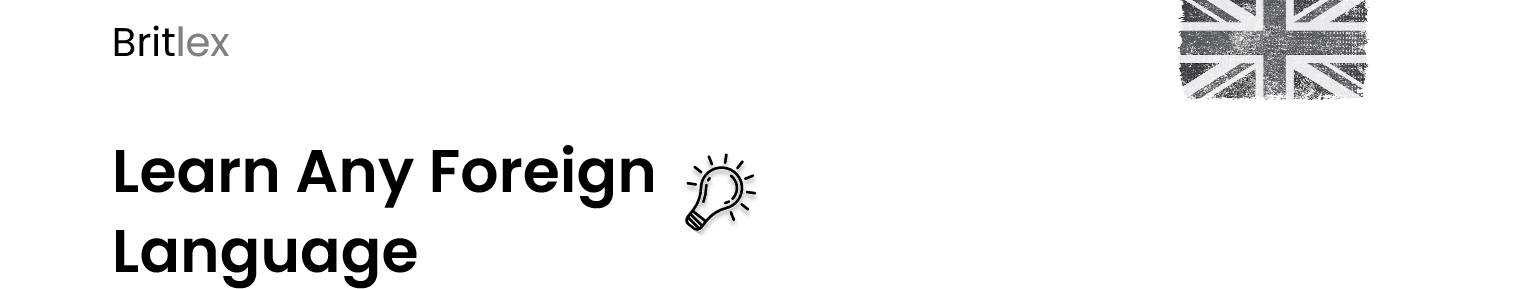
Кроссбраузерность – это способность веб-ресурса отображаться одинаково и работать во всех популярных браузерах, без перебоев в функционировании и ошибок в верстке, а также с одинаково корректной читабельностью контента.

В ходе тестирования были проверены все ссылки(включая модальные окна), проверены формы регистрации пользователя. Была выполнена проверка контента. Также в ходе тестирования была проверена совместимость с браузерами.(Google Chrome, Opera, Firefox, Microsoft Edge). В браузере Internet Explorer тестирование давала некоторые сбои. А именно проблема с запуском с открытого сервера(live server), а также анимации сделанные непосредственно через JavaScript ввели себя некорректно.

У каждого браузера есть свои встроенные, экспериментальные или нестандартные свойства и для того, чтобы они корректно работали ввелись вендорне префиксы. Они были автоматически прописаны после компиляции файла с расширением sass.

После написания основной структуры страницы на HTML и внешнего стилевого оформления появился вопрос о тестировании. Веб-сайт был открыт при помощи различных браузеров.

Также в ходе тестов были проверены конкретные браузеры, к примеру браузер Internet Explorer некорректно отображал навигационное меню, и кнопки лампочки и регистрации. При нажатии на лампочку, браузер ничего не выполнял.



# 4.3. Руководство пользователя

Данный веб-сайт предоставляет информацию потенциальному клиенту о услугах в изучении иностранных языков. После нажатия на кнопку ”Get Started”, пользователю предлагается возможность выбрать( с помощью удобного слайдера) язык, который ему интересен, а затем заполнить форму регистрации, указав там свой логин и пароль соответственно.

Для обеспечения надежности, а также удобства данного веб-сайта выполнено:

—разработан удобный динамический пользовательский интерфейс, без лишних страниц, но со всем необходимым функционалом;

—веб-страница включает в себя “якоря” для того, чтобы пользователь мог удобно и быстро перейти на любую часть страницы.

Вверху сайта, вы можете увидеть навигационную панель. Она содержит основные тематические разделы веб-сайта: “Home”, “Skills”, “About Us”, “Pricing” “Contacts”. В ней отображаются основные вкладки на сайте. Кликнув ЛКМ по вкладке панели, пользователя перенаправит на данную часть сайта. В случае использования веб-ресурса в мобильной версии навигационное меню открывается по клику на “Бургер-меню”;

Во вкладке “Skills” показаны навыки, которые будут приобретены пользователем в ходе курса.

Во вкладке “About Us” пользователь может получить краткую информация о услугах, предоставляемых сайтом.

Во вкладке “Pricing” пользователь может узнать цены на услуги, которые предоставляет сайт.

Во вкладке “Contacts” пользователь может оставить заявку, на участие в данных курсах.

# 4.4. Вывод

Цель тестирования Главной целью такой тщательной проверки сайта является грамотная настройка всех показателей, одной или всех страниц сайта, оперативное обнаружение и устранение всех дефектных мест на сайте, а также его адаптация под различные устройства.

Динамика сайта на разных устройствах – наиболее важная часть в написании сайта, поэтому, на данном этапе было проанализировано поведение веб-сайта при использовании с разных устройств, а также веб-ресурс был протестирован в большом количестве браузеров. По итогу выполнения данного этапа для веб-сайта был разработан адаптивный дизайн и кроссбраузерность. Работа, проделанная с сайтом, позволяет пользователям ориентироваться по сайту. Лишних элементов нет, что упрощает пользование сайтом. При пользовании сайтом при помощи мобильного устройства некоторые элементы меняют свое положение, для более удобного пользования.

Заключение

Целью курсового проекта была разработка веб-сайта онлайн школы для изучения иностранных языков. Который предоставлял бы информацию о услугах и был бы наиболее удобен в использовании человеку из любой точки мира. Цель состояла в том, чтобы обеспечить гибкость, динамичность, и что самое главное, понятность данному “лендингу.”

Для реализации макета веб-страницы были исследованы сайты схожей тематикой и исследованы недостатки веб-сайтов. Показана и предложено иные реализации блоков веб-страницы.

Visual Studio Code стала отличной IDE для комфортного и ускоренного написания кода для веб-страницы. А именно выбор необходимых плагинов и цветовой темы.

По итогам реализации веб-сайта, было принято решения о тестировании веб-сайта. В результате тестирования была обнаружена неспособность веб-браузера(Internet Explorer) работать с некоторыми свойствами.

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

Для достижения поставленной цели для курсового проекта функционально были реализованы следующие задачи:

— анализ аналогичных решений;

— выбор способа верстки;

— выбор стилевого оформления;

— разработка логотипа;

— разработка пользовательских элементов;

— разработка картинок и спецэффектов;

— добавление веб-сайту адаптивности;

— кроссбраузерность веб-сайта;

— создание руководства пользователя;

Список используемых литературных источников представлен в пункте 6. Также дополнительная информация находится в приложениях 1-6.

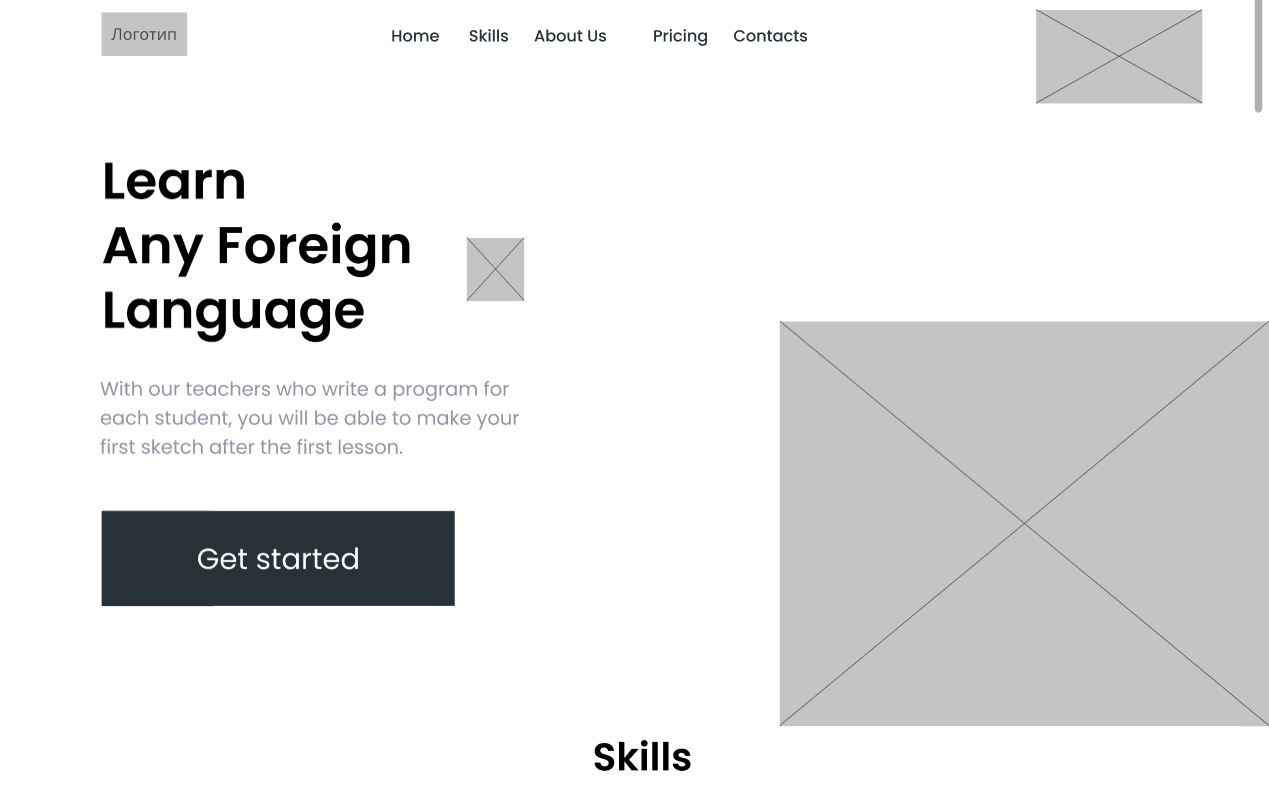
Результатом курсового проекта стал готовый к работе веб-сайт, в котором размещена вся информация о курсах, и языках изучаемых на платформе. В дальнейшем разработанный веб-сайт можно будет использовать в реальной жизни, как конкурент современным онлайн школам. Так же возможно расширение списка предоставляемых услуг. Данный ресурс стабилен в работе и не требует высоких знаний для работы с ним.

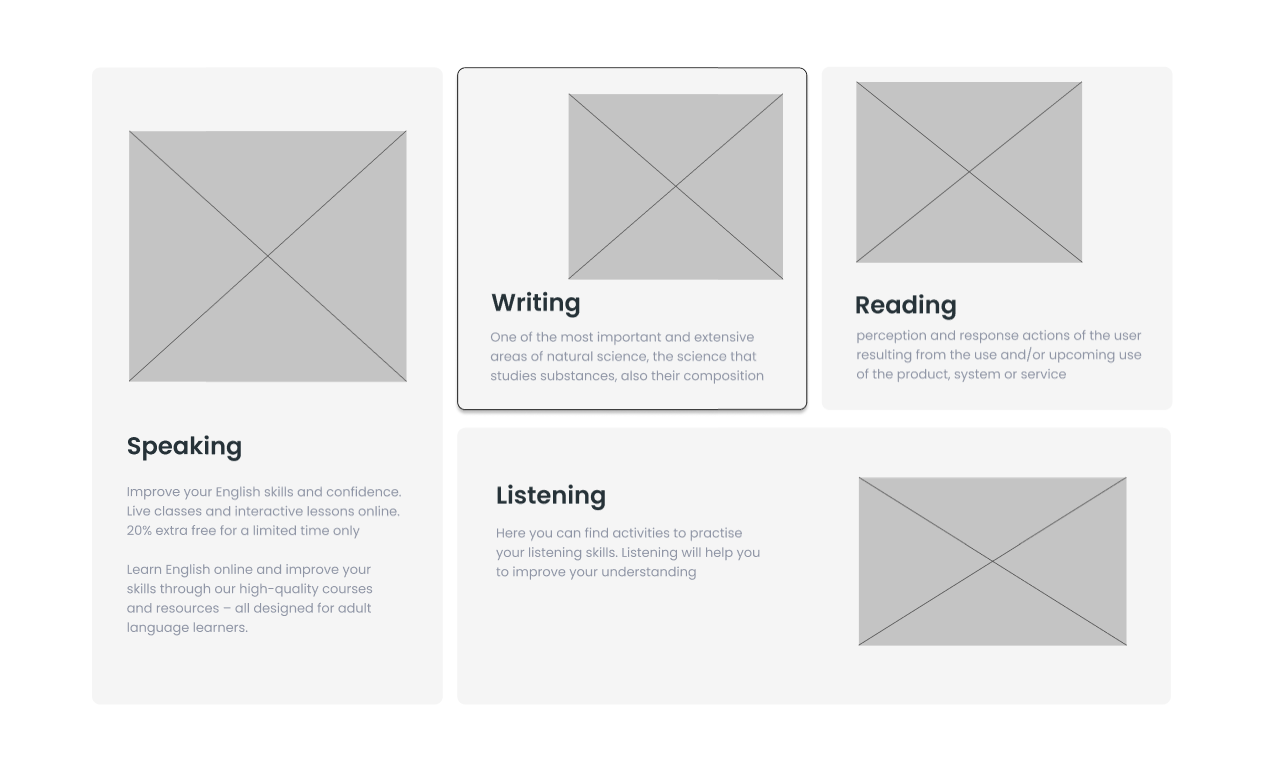
Таким образом, из выше сказанного, можно сделать вывод, что в курсовом проекте раскрывается все теоретические и практические аспекты темы и достигнуты поставленные цели и задачи роботы.

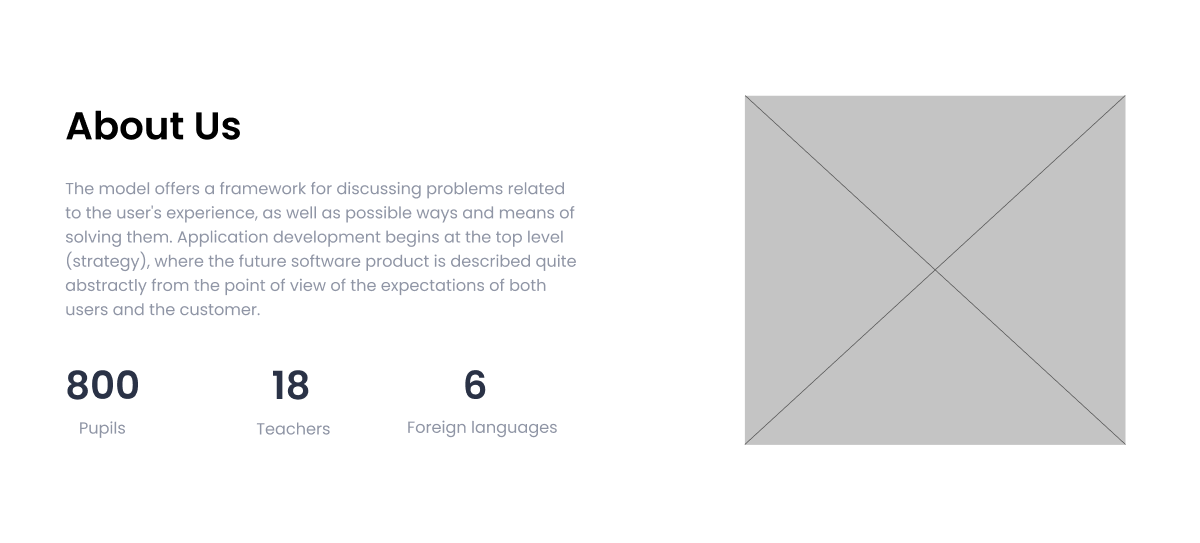
# Список использованных источников

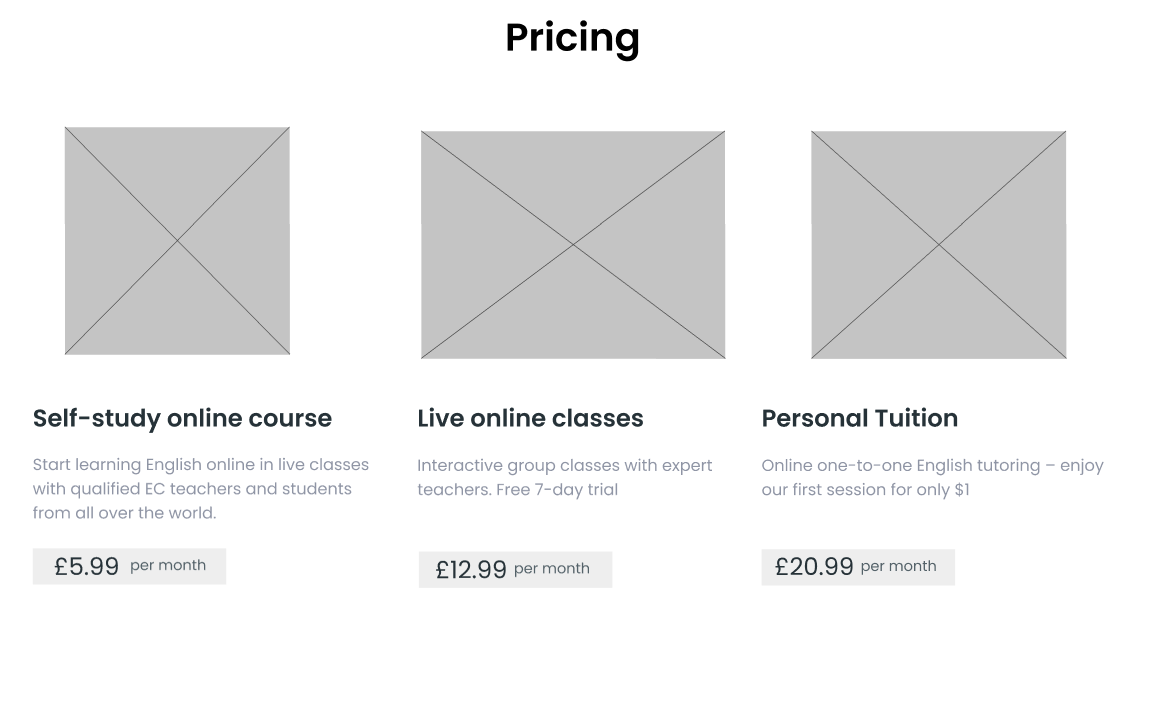
1. Документация по Figma [Электронный ресурс]. - Режим доступа: <https://help.figma.com>. – Дата обращения 01.02.2022.
2. Документация по Sass/Scss [Электронный ресурс]. - Режим доступа: <https://sass-scss.ru/guide/>. – Дата обращения 01.02.2022.
3. Документация по JavaScript [Электронный ресурс]. - Режим доступа: <https://learn.javascript.ru/>. – Дата обращения 01.02.2022.

# Приложение А. Прототипы веб-сайта

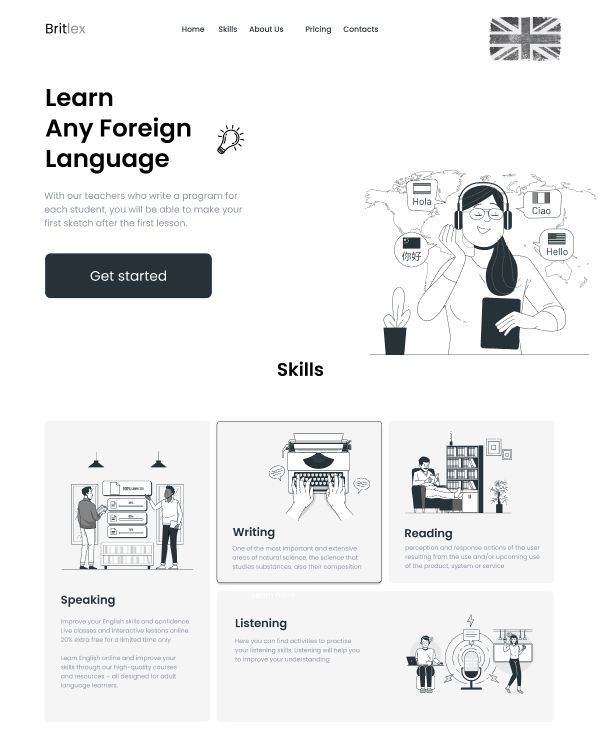
****

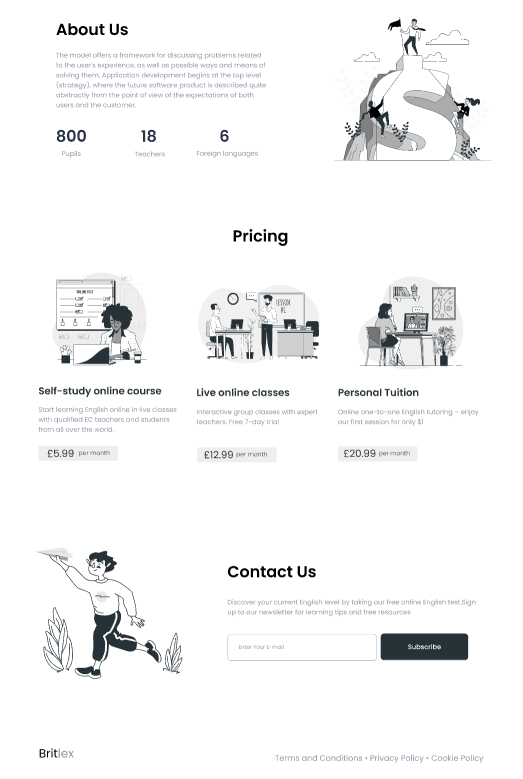
****

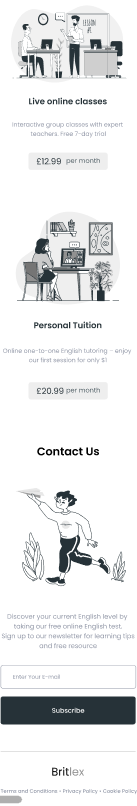
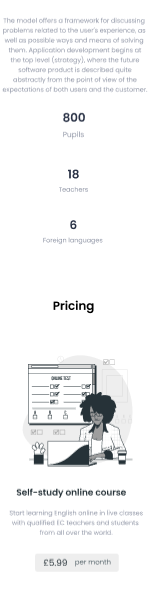
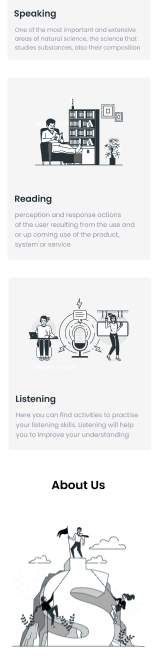
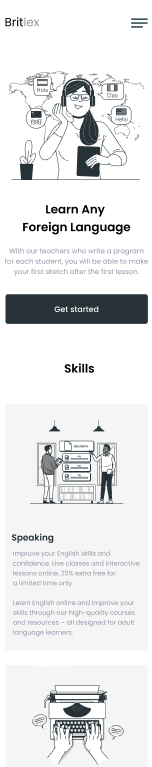


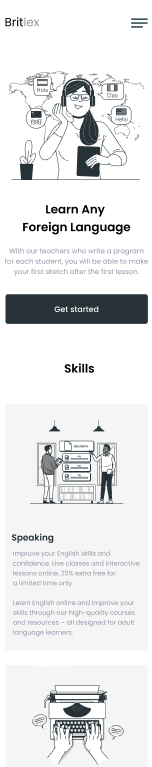
****

# Приложение Б. Макеты структур веб-сайта









# Приложение В. Листинг HTML-документа

|  |
| --- |
| <header>  <div class="wrapper">  <div class="Britlex">...  </div>  <div class="nav">...  </div>  <div class="flag">...  </div>  <div class="burger">  <div class="container" id="burger">...  </div>  </div>  </div>  </header> |

Структура тега header

|  |
| --- |
| <!--Отдельная секция-->  <div class="about\_us">...  </div>  <div class="Text\_price">  Pricing  </div>  <!--Отдельная секция-->  <section data-aos="zoom-in" class="section">...  </section>  <!--Отдельная секция-->  <section class="contact\_section">...  </section>  <!--Нижняя часть сайта-->  <footer class="foot">...  </footer>  </div> |

Структура html

|  |
| --- |
| <section class="section\_skils">  <div class="wrapper">  <div class="Wrapper\_Skills">  <div data-aos="fade-right" class="Speaking">...  </div>  <div data-aos="fade-left" class="flex-column">...  </div>  </div>  </div>  </section>  <section data-aos="zoom-in" class="section">  <div class="wrapper">  <div class="Price\_wrap">...  <div class="block\_one">...  </div>  <div class="block\_one">...  </div>  <div class="block\_one">...  </div>  </div>  </div>  </section> |

Структура тегов section

|  |
| --- |
| <!--Основной блок сайта-->  <main class="main">  <div class="wrapper">  <div class="flex-count">  <div class="Learn">...  </div>  <div class="Country ">...  </div>  </div>  </div>  </main> |

Структура тега main

|  |
| --- |
| <!--Нижняя часть сайта-->  <footer class="foot">  <div class="wrapper">  <div class="flex-footer">  <div class="Britlex">  Brit<span class="gray">lex</span>  </div>  <div class="text\_footer">  <p> Terms and Conditions </p>  <p>  • Privacy Policy </p>  <p>  • Cookie Policy</p>  </div>  </div>  </div>  </footer> |

Структура тега footer

# Приложение Г. Листинг XML (SVG)

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <root>  <ul>  <class>  none  </class>  <name> Home</name>  </ul>  <ul>  <class>  skills  </class>  <name> Skills  </name>  </ul>  <ul>  <class>  about\_us  </class>  <name> About us  </name>  </ul>  <ul>  <class>  priising  </class>  <name> Pricing  </name>  </ul>    <ul>  <class>  contacts\_btn  </class>  <name> Contacts  </name>  </ul>  </root> |

|  |
| --- |
| <svg width="120" height="120" viewBox="0 0 120 120" fill="none"  xmlns="http://www.w3.org/2000/svg" class="light">  <g clip-path="url(#clip0\_43\_17167)" filter="url(#filter0\_d\_43\_17167)">  <path>...  </clipPath>  </defs>  </svg> |

# Приложение Д. Листинг JavaScript

|  |
| --- |
| let burger = document.querySelector("#burger");  let nav = document.querySelector(".nav")  burger.addEventListener("click", () => {  if (!nav.classList.contains("active")) {  nav.classList.add("active");  document.body.style.overflow = "hidden"  burger.classList.add("change")  }  else {  nav.classList.remove("active");  burger.classList.remove("change")  document.body.style.overflow = "visible"  }  }) |

Burger-меню

|  |
| --- |
| let light = document.querySelector(".light");  let count = 0;  light.onclick = () => {  count++;  if (count % 2 == 0) {  applyTheme("style/Kurs");  } else {  applyTheme("style/Night");  }  };  function applyTheme(themeName) {  document  .querySelector('[title="theme"]')  .setAttribute("href", `${themeName}.css`); // Помещаем путь к файлу темы в пустой link в head  } |

Изменение цвета страницы(Темная тема)

|  |
| --- |
| let skills = document.querySelector(".skills")  let Skil = document.querySelector(".Skil")  skills.addEventListener("click", () => {  nav.classList.remove("active");  burger.classList.remove("change")  document.body.style.overflow = "visible"  Skil.scrollIntoView({  behavior: "smooth",  block: "start"  })  })  let about\_us = document.querySelector(".about\_us")  let title\_about = document.querySelector(".title\_about")  about\_us.addEventListener("click", () => {  nav.classList.remove("active");  burger.classList.remove("change")  document.body.style.overflow = "visible"  title\_about.scrollIntoView({  behavior: "smooth",  block: "start"  })  })  let priising = document.querySelector(".priising");  let Text\_price = document.querySelector(".Text\_price")  priising.addEventListener("click", () => {  nav.classList.remove("active");  burger.classList.remove("change")  document.body.style.overflow = "visible"  Text\_price.scrollIntoView({  behavior: "smooth",  block: "start"  })  })  let contacts\_btn = document.querySelector(".contacts\_btn")  let Text\_contact = document.querySelector(".Text\_contact")  contacts\_btn.addEventListener("click", () => {  nav.classList.remove("active");  burger.classList.remove("change")  document.body.style.overflow = "visible"  Text\_contact.scrollIntoView({  behavior: "smooth",  block: "start"  })  }) |

Плавные переходы (якоря)

|  |
| --- |
| let form\_register = document.querySelector(".forms")  let close\_register = document.querySelector(".close\_register")  let buy\_course = document.querySelectorAll(".buy\_course")  close\_register.addEventListener("click", () => {  form\_register.classList.remove("active")  })  buy\_course.forEach(item => {  item.addEventListener("click", () => {  get\_started.classList.remove("active")  form\_register.classList.add("active")  })  })  let close\_login = document.querySelector(".close\_login")  let Subscribe\_btn = document.querySelector(".Subscribe\_btn");  let formsLogin = document.querySelector(".formsLogin")  Subscribe\_btn.addEventListener("click", () => {  formsLogin.classList.add("active")  })  close\_login.addEventListener("click", () => {  formsLogin.classList.remove("active")  })  // Слайдер  const slideContainer = document.querySelector(".containers"); // контейнер слайдов  const slide = document.querySelector(".slides"); // получаем контейнер слайдов  const nextBtn = document.getElementById("next-btn"); // кнопка вперед  const prevBtn = document.getElementById("prev-btn"); // кнопка назад  const interval = 3000; // интервал между слайдами  let slides = document.querySelectorAll(".slide"); // получаем все слайды  let index = 1; // индекс текущего слайда  let slideId;  const firstClone = slides[0].cloneNode(true); // клонируем первый слайд и добавляем его в конец контейнера слайдов  const lastClone = slides[slides.length - 1].cloneNode(true); // клонируем последний слайд и добавляем его в начало контейнера слайдов  firstClone.id = "first-clone"; // добавляем id клонированному слайду  lastClone.id = "last-clone"; // добавляем клонированный слайд в конец контейнера слайдов  slide.append(firstClone); // добавляем клонированный слайд в конец контейнера слайдов  slide.prepend(lastClone); // добавляем клонированный слайд в начало контейнера слайдов  var slideWidth; // ширина слайда  setInterval(() => {  slideWidth = slides[0].clientWidth;  }, 0);  slide.style.transform = `translateX(${-slideWidth \* index}px)`; // перемещаем слайд на нужную позицию  const startSlide = () => {  // запускаем слайдер  slideId = setInterval(() => {  moveToNextSlide(); //  }, interval);  };  const getSlides = () => document.querySelectorAll(".slide"); // получаем все слайды  slide.addEventListener("transitionend", () => {  // при завершении анимации  slides = getSlides(); // получаем все слайды  if (slides[index].id === firstClone.id) {  // если переместились на первый слайд  slide.style.transition = "none"; // отключаем анимацию  index = 1; // возвращаем индекс на первый слайд  slide.style.transform = `translateX(${-slideWidth \* index}px)`; // перемещаем слайд на первый слайд  }  if (slides[index].id === lastClone.id) {  // если переместились на последний слайд  slide.style.transition = "none"; // отключаем анимацию  index = slides.length - 2; // возвращаем индекс на последний слайд  slide.style.transform = `translateX(${-slideWidth \* index}px)`; // перемещаем слайд на последний слайд  }  });  const moveToNextSlide = () => {  slides = getSlides(); // получаем все слайды  if (index >= slides.length - 1) return; // если переместились на последний слайд  index++; // увеличиваем индекс  slide.style.transition = ".7s ease-out";  slide.style.transform = `translateX(${-slideWidth \* index}px)`;  };  const moveToPreviousSlide = () => {  if (index <= 0) return; // если переместились на первый слайд  index--; // уменьшаем индекс  slide.style.transition = ".7s ease-out";  slide.style.transform = `translateX(${-slideWidth \* index}px)`;  };  slideContainer.addEventListener("mouseenter", () => {  // при наведении на контейнер слайдов  clearInterval(slideId); // останавливаем слайдер  });  slideContainer.addEventListener("mouseleave", startSlide); // при отведении курсора от контейнера слайдов  nextBtn.addEventListener("click", moveToNextSlide); // при нажатии на кнопку "следующий слайд"  prevBtn.addEventListener("click", moveToPreviousSlide); // при нажатии на кнопку "предыдущий слайд"  startSlide(); // запускаем слайдер  // Это бесконечный слайдер с таймером и кнопками переключения слайдов |

Форма регистрации, Слайдер