

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования «Уральский федеральный университет имени первого

Президента России Б.Н. Ельцина» (УрФУ)

Институт радиоэлектроники и информационных технологий - РТФ

ОТЧЕТ ПО ПРОЕКТНОЙ РАБОТЕ

НА ТЕМУ: ОБУЧЕНИЕ ТЕХНИЧЕСКОМУ АНГЛИЙСКОМУ ЯЗЫКУ

ПО ДИСЦИПЛИНЕ: ПРОЕКТНЫЙ ПРАКТИКУМ

КОМАНДА: ППA1.ИРИТ-РТФ.ТАЯ П, АТ-03

Екатеринбург

2021

Содержание

[Календарный план проекта 3](#_Toc74824109)

[Аналитика 5](#_Toc74824110)

[Определение проблемы 5](#_Toc74824111)

[Выявление целевой аудитории 5](#_Toc74824112)

[Конкретизация проблемы 5](#_Toc74824113)

[Подходы к решению проблемы 5](#_Toc74824114)

[Определение платформы и стека для продукта 5](#_Toc74824115)

[Анализ аналогов 6](#_Toc74824116)

[SkyEng 6](#_Toc74824117)

[Lingualeo 7](#_Toc74824118)

[Memrise 8](#_Toc74824119)

[Duolingo 9](#_Toc74824120)

[English4it 10](#_Toc74824121)

[ComputerLanguage 11](#_Toc74824122)

[Требования к продукту 12](#_Toc74824123)

[Формулировка требований к MVP продукта 13](#_Toc74824124)

[Формулировка цели 13](#_Toc74824125)

[Проектирование 14](#_Toc74824126)

[Архитектура системы 14](#_Toc74824127)

[Сценарии использования системы 15](#_Toc74824128)

[Прототипы интерфейса 17](#_Toc74824129)

[Формирование контентной части 21](#_Toc74824130)

[Разработка 22](#_Toc74824131)

[Написание программного кода 22](#_Toc74824132)

[Оформление и внедрение контентной части 23](#_Toc74824133)

Календарный план проекта

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Название** | **Ответственный** | **Длительность** | **Дата начала** | **Временные рамки проекта** | | | |
| 1 неделя | 2 недели | 3 недели | 4 недели |
| ***Анализ*** | | | | | | | | |
| *1.1* | *Определение проблемы* | *Цветаев В.С.* | *2 дня* | *01.04.21* |  |  |  |  |
| *1.2* | *Выявление целевой аудитории* | *Цветаев В.С.* | *2 дня* | *01.04.21* |  |  |  |  |
| *1.3* | *Конкретизация проблемы* | *Цветаев В.С.* | *2 дня* | *01.04.21* |  |  |  |  |
| *1.4* | *Подходы к решению проблемы* | *Цветаев В.С.* | *2 дня* | *01.04.21* |  |  |  |  |
| *1.5* | *Анализ аналогов* | *Шинкарёв Е.Г.* | *4 дня* | *01.04.21* |  |  |  |  |
| *1.6* | *Определение платформы и стека для продукта* | *Друженьков Г.В.*  *Мухин К.В.* | *1,5 недели* | *03.04.21* |  |  |  |  |
| *1.7* | *Формулирование требований к продукту* | *Цветаев В.С.*  *Плотников Е.Д.* | *1,5 недели* | *08.04.21* |  |  |  |  |
| *1.8* | *Определение платформы и стека для MVP* | *Друженьков Г.В.*  *Мухин К.В.* | *Неделя* | *03.04.21* |  |  |  |  |
| *1.9* | *Формулирование требований к MVP продукта* | *Цветаев В.С.*  *Шинкарёв Е.Г.*  *Плотников Е.Д.* | *Неделя* | *03.04.21* |  |  |  |  |
| *1.10* | *Формулировка цели* | *Цветаев В.С.* | *4 дня* | *06.04.21* |  |  |  |  |
| *1.11* | *Определение задач* | *Цветаев В.С.* | *Неделя* | *08.04.21* |  |  |  |  |
| ***Проектирование*** | | | | | | | | |
| *2.1* | *Архитектура системы (компоненты, модули системы)* | *Все, кроме Плотникова Е.Д.* | *2 недели* | *16.04.21* |  |  |  |  |
| *2.2* | *Разработка сценариев использования системы* | *Мухин К.В.* | *2 недели* | *16.04.21* |  |  |  |  |
| *2.3* | *Прототипы интерфейсов* | *Друженьков Г.В.*  *Плотников Е.Д.* | *2 недели* | *16.04.21* |  |  |  |  |
| *2.4* | *Формирование контентной части* | *Плотников Е.Д.*  *Цветаев В.С.* | *2 недели* | *16.04.21* |  |  |  |  |
| ***Разработка*** | | | | | | | | |
| *3.1* | *Написание кода* | *Все, кроме Плотникова Е.Д.* | *4 недели* | *02.05.21* |  |  |  |  |
| *3.2* | *Тестирование приложения* | *4 недели* | *02.05.21* |  |  |  |  |
| *3.3* | *Оформление и внедрение контентной части* | *Плотников Е.Д.*  *Цветаев В.С.* | *4 недели* | *02.05.21* |  |  |  |  |
| ***Внедрение*** | | | | | | | | |
| *4.1* | *Написание отчета* | *Цветаев В.С.* | *2 недели* | *20.05.21* |  |  |  |  |
| *4.2* | *Оформление презентации* | *Цветаев В.С.*  *Плотников Е.Д.*  *Мухин К.В.* | *1,5 недели* | *20.05.21* |  |  |  |  |
|  | *Защита проекта* |  |  | *0.7.06 -15.06.21* |  |  |  |  |

Аналитика

## Определение проблемы

Большинство начинающих и профессиональных специалистов в IT-сфере плохо знают технический английский, либо не знают вообще, из-за чего возникает трудность в обучении и понимании изученного материала.

## Выявление целевой аудитории

Наша аудитория состоит из русскоязычных IT-специалистов различных направлений и уровней профессиональной подготовки, объединенных потребностью в изучении технического английского, и имеющих базовый (на уровне средней школы) уровень знания разговорного и письменного английского.

## Конкретизация проблемы

При обучении, либо решении какой-либо задачи, у специалиста может возникнуть потребность в изучении документации или получения помощи у профессионального сообщества, но при отсутствии определенного уровня знания технического английского (в нашем случае базового) это будет сделать невозможно.

## Подходы к решению проблемы

Реализовать сервис, предоставляющий в удобном и понятном формате актуальные темы из IT-сферы, которые позволят специалистам увеличить словарный запас, а также улучшить навыки аудирования и чтения. Итогом обучения будет являться лучшее понимание и восприятие англоязычных ресурсов.

## Определение платформы и стека для продукта

Сервис будет представлять собой web-приложение на платформе ASP.NET MVC с использованием различных библиотек (EntityFramework, Identity и др.).

Развертывание и совместная разработка будут проходить с использованием Microsoft Azure, Azure DevOps, GitHub.

Frontend-составляющая проекта будет разрабатываться с использованием Gulp, который обеспечивает сборку HTML, SASS & SCSS, JS, подключаемые фреймворки и библиотеки.

## Анализ аналогов

[SkyEng](https://skyeng.ru/)

Стартовый экран сайта представлен на рисунке 1.

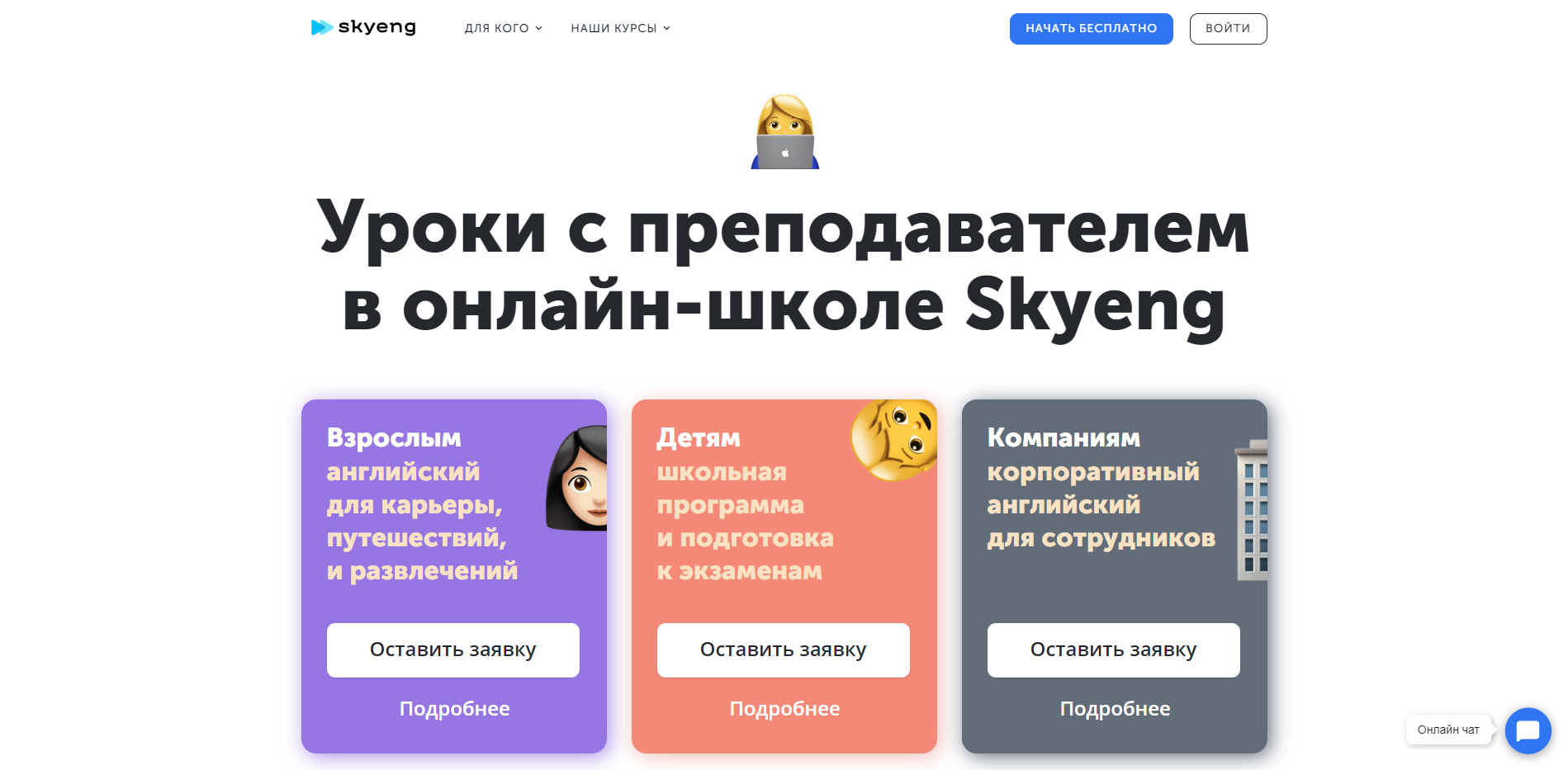
******

Рисунок 1 — Внешний вид сайта "SkyEng"

Российская онлайн-школа по изучению английского языка. Есть возможность заниматься самостоятельно или с репетитором.

Имеются видео-лекции и упражнения с автоматической проверкой, словарь и статистика по всем занятиям. Программа разбита на курсы, которые можно покупать по отдельности, вводное занятие бесплатно.

[Lingualeo](https://lingualeo.com/ru)

Стартовый экран сайта представлен на рисунке 2.

******

Рисунок 2 — Внешний вид сайта "Lingualeo"

Также является российской образовательной платформой, но в отличие от SkyEng позволяет изучать множество языков. Процесс обучения проходит в игровой форме, нет возможности заниматься с репетитором. Есть как платные, так и бесплатные курсы. Среди платных есть курс “английский для IT-бизнеса”. Имеется подписка, которая открывает доступ к расширенным функциям.

[Memrise](https://www.memrise.com/ru/)

Стартовый экран сайта представлен на рисунке 3.

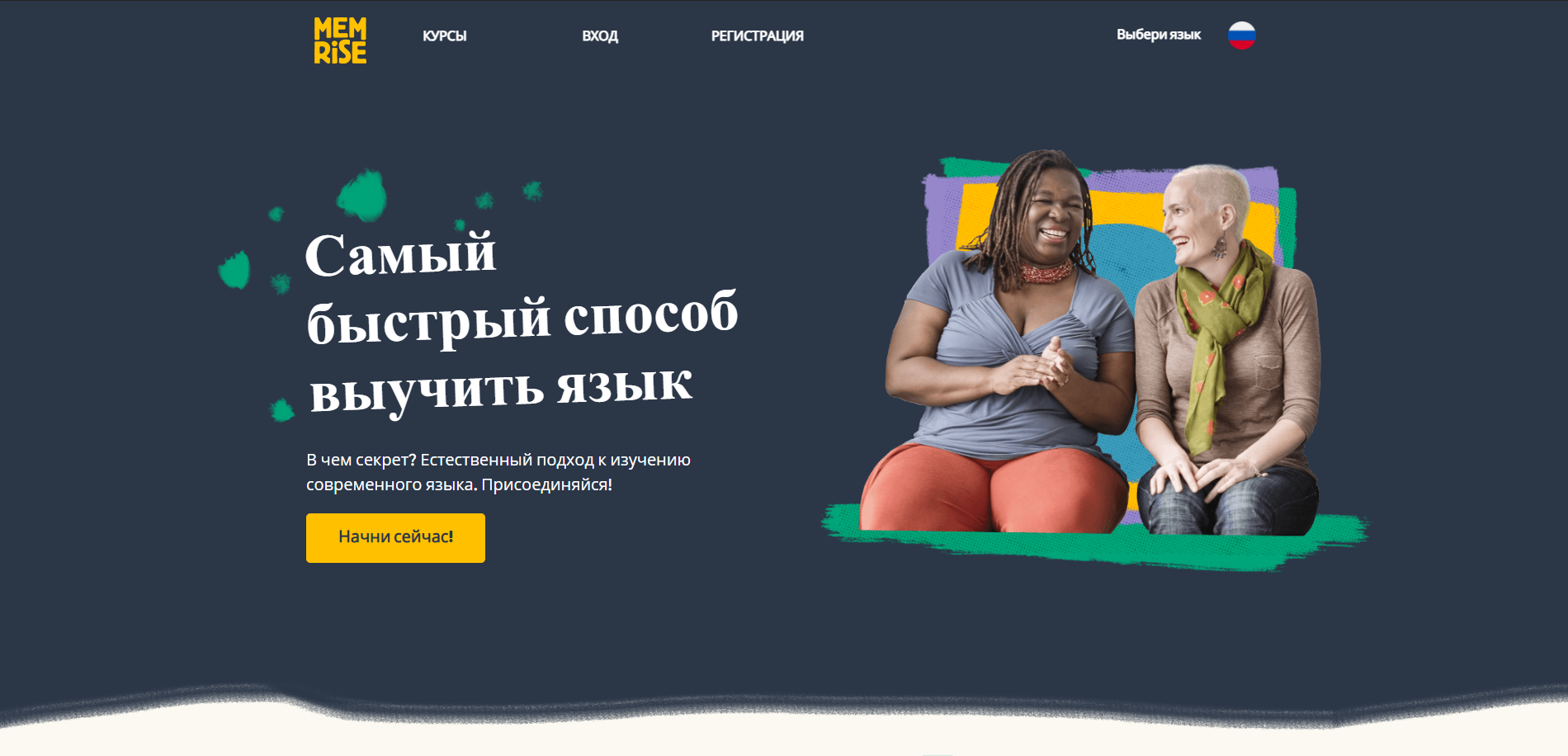
**

Рисунок 3 — Внешний вид сайта "Memrise"

Сервис для заучивания слов. Нет упражнений и теории, вместо них блоки слов на одну тематику. Основан на системе интервального повторения. Имеется подписка, открывающая больше функций. Можно создать свой курс.

[Duolingo](https://www.duolingo.com/)

Стартовый экран сайта представлен на рисунке 4.

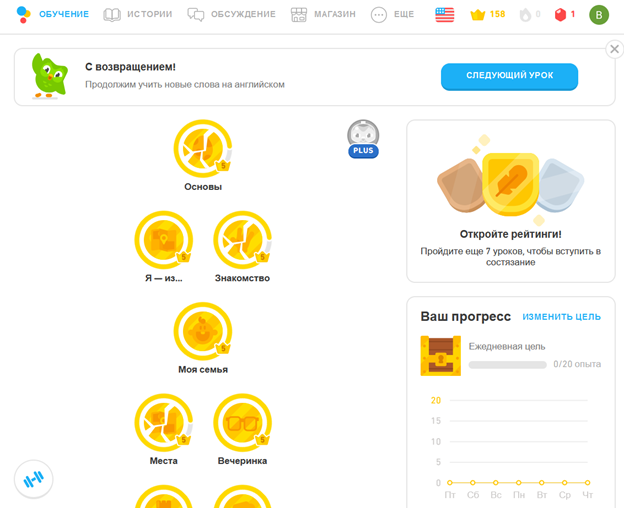
**

Рисунок 4 — Внешний вид сайта "Duolingo"

Сервис для изучения языков и [краудсорсинговых](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B0%D1%83%D0%B4%D1%81%D0%BE%D1%80%D1%81%D0%B8%D0%BD%D0%B3) переводов. Теория представлена в тестовом виде, в упражнениях есть игровая и соревновательная составляющая. Фокус смещен на разговорный английский. Можно купить платную подписку, которая убирает рекламу. Все основные функции бесплатны.

[English4it](https://english4it.online/)

Стартовый экран сайта представлен на рисунке 5.

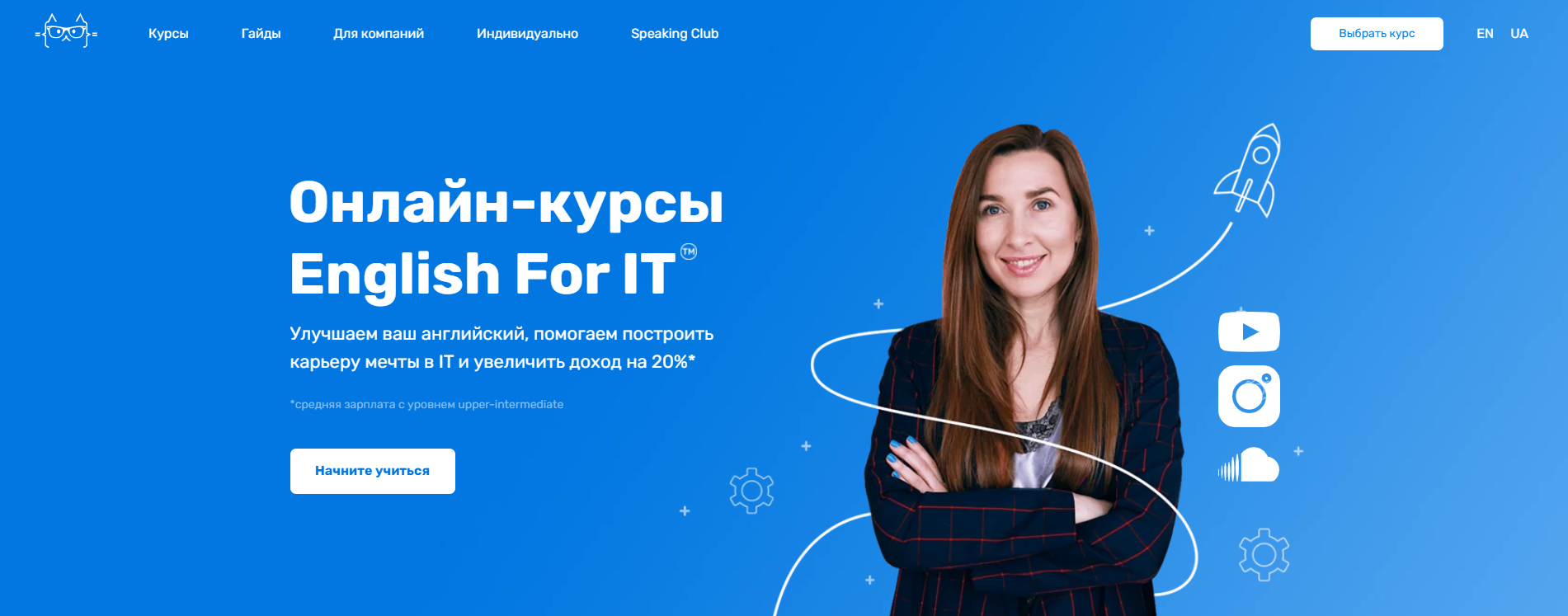


Рисунок 5 — Внешний вид сайта "English4it"

Онлайн-школа для изучения английского. Предоставляют полноценные (по их словам) курсы с репетиторами и заданиями с ручной проверкой. Нет бесплатного контента, но иногда проводят открытые вебинары.

[ComputerLanguage](https://www.computerlanguage.com/)

Стартовый экран сайта представлен на рисунке 6.

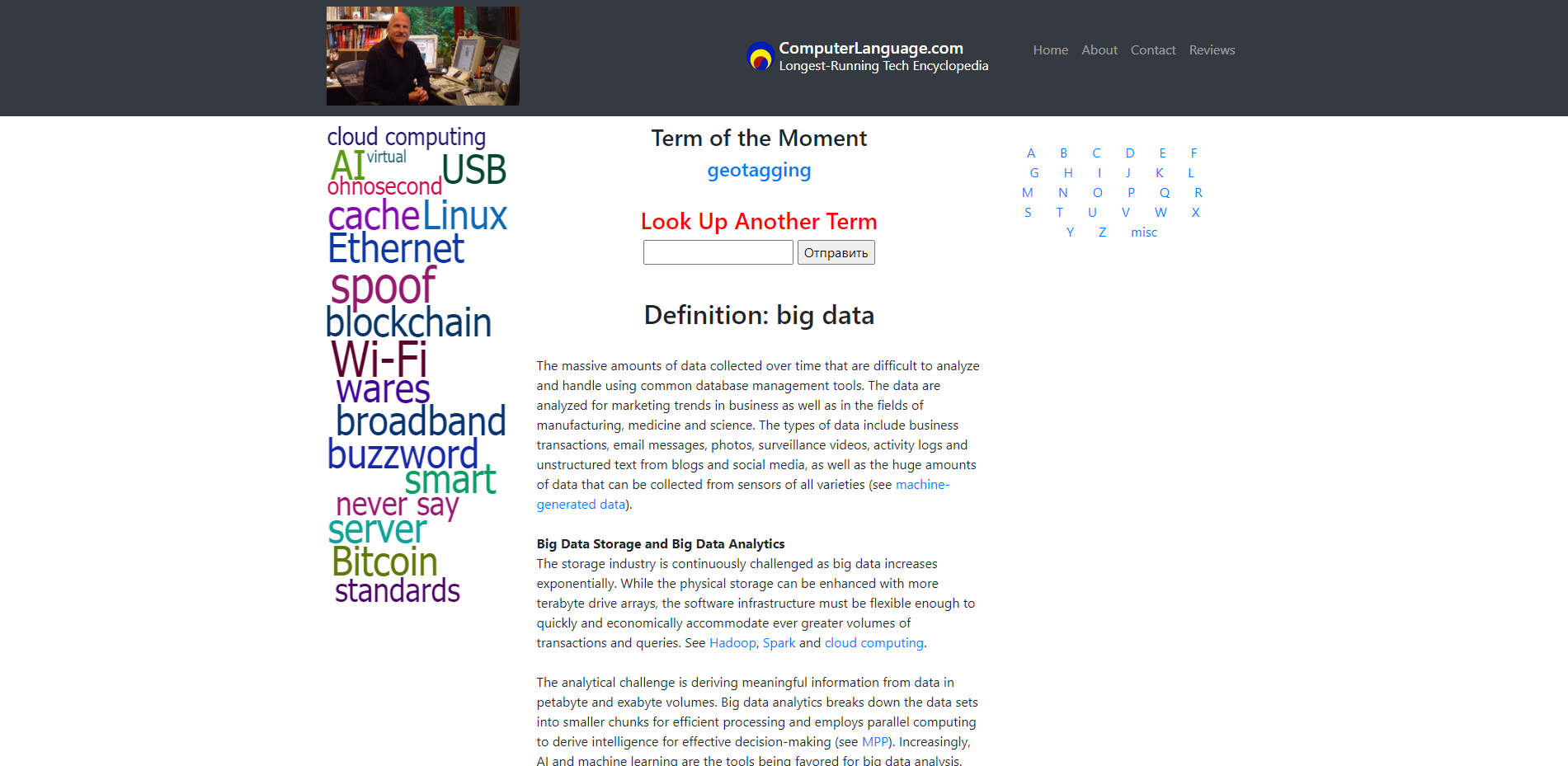


Рисунок 6 — Внешний вид сайта "ComputerLanguage"

Толковый словарь IT-терминов. Доступен только на английском языке, полностью бесплатен. Позволяет получить определение термина и пример использования, всё написано на простом английском.

## Требования к продукту

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Код** | **Требование** | **Источник** | **Приоритет**  **<Высокий, средний, низкий>** | **Примечание**  **Взаимосвязи с другими требованиями** |
| Спецификация обучающего сервиса | | | | |
| Т1\_1 | Сервис представляет собой сайт, который направлен на изучение технического английского на различные темы из IT-сферы | Команда проекта | Высокий | Т2 |
| Т1\_2 | Сервис должен быть удобным настолько, чтобы можно было сразу разобраться в его возможностях | Команда проекта | Средний |  |
| Контентная часть | | | | |
| Т2\_1 | Сервис базируется на техническом английском языке, подразумевается, что пользователь заранее знает базовый английский язык на уровне общего образования | Команда проекта | Высокий | Т1\_1 |
| Т2\_1 | Изучаемые пользователем темы строго соответствуют своей тематике | Команда проекта | Высокий | Т2\_1 |
| Т2\_3 | Темы должны быть полностью раскрытыми и интересными, с наличием новых подборок слов и практик на аудирование и чтение | Команда проекта | Высокий |  |
| Т2\_4 | Наличие словаря, автоматически пополняемого словами, которые изучил пользователь | Команда проекта | Средний | Т2\_3 |
| Т2\_5 | Накопление статистики для каждого пользователя | Команда проекта | Средний |  |
| [Требования будут добавляться] | | | | |

## Формулировка требований к MVP продукта

Аналогично требованиям к продукту, но будет реализована только одна доступная тема для изучения.

## Формулировка цели

До 4 семестра запустить удобный сервис по изучению технического английского языка, направленный на русскоязычных специалистов, всех уровней профессиональной подготовки и желающих повысить свои навыки владения английским для лучшего понимания зарубежных IT-ресурсов.

Проектирование

## Архитектура системы

Приложение разрабатывается на ASP.NET Core с использованием схемы MVC, фреймворков Entity и Bootstrap.

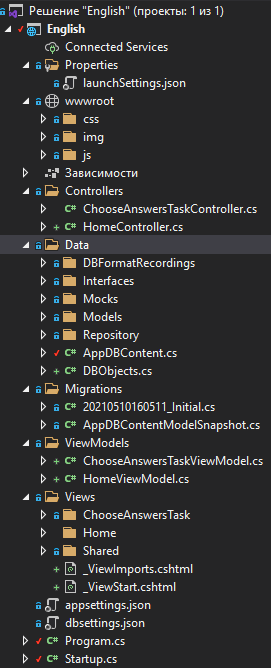
1. **wwwroot** – хранение статических файлов

Рисунок 7 — Архитектура системы

*css* – таблицы стилей

*img* – изображения

*js* – скрипты

1. **Controllers** – контроллеры, обрабатывающие запросы на сервер
2. **Data** – логика и данные приложения
   * + - *DBFormatRecordings* – классы, на основе которых создаются таблицы в бд
       - *Interfaces* – интерфейсы взаимодействия логики с данными
       - *Mocks* – тестовое хранилище данных, реализует интерфейсы получения данных
       - *Models* – классы, отвечающие за логику
       - *Repository* – реализует интерфейсы получения данных, которые берутся из бд
       - *AppDBContent*.cs – таблицы в бд
       - *DBObjects*.cs – инициализация бд
3. **Migrations** – код для создания бд
4. **ViewModels** – модели для передачи данных из контроллера в View
5. **View** – шаблоны html с внедренной razor – разметкой
   * + - *Shared* – общие шаблоны
       - *ViewImports* – зависимости, позволяет не писать using something
       - *ViewStarts* – основной шаблон
6. **appsetting.json** – [поставщик конфигурации JSON](https://docs.microsoft.com/ru-ru/aspnet/core/fundamentals/configuration/?view=aspnetcore-3.1#file-configuration-provider)
7. **dbsetting.json** – поставщик конфигурации для бд
8. **Program.cs** – запуск приложения
9. **Startup.cs** – конфигурация приложения, добавление сервисов

## Сценарии использования системы

Создание и расширение User-ways (Рисунок 8) позволяет визуализировать требуемый функционал, чтобы в дальнейшем облегчить процесс создания и развития продукта, а также выделить приоритетные задачи и направления разработки. Для создания MVP мы выделили две основные роли и минимальный набор функций.

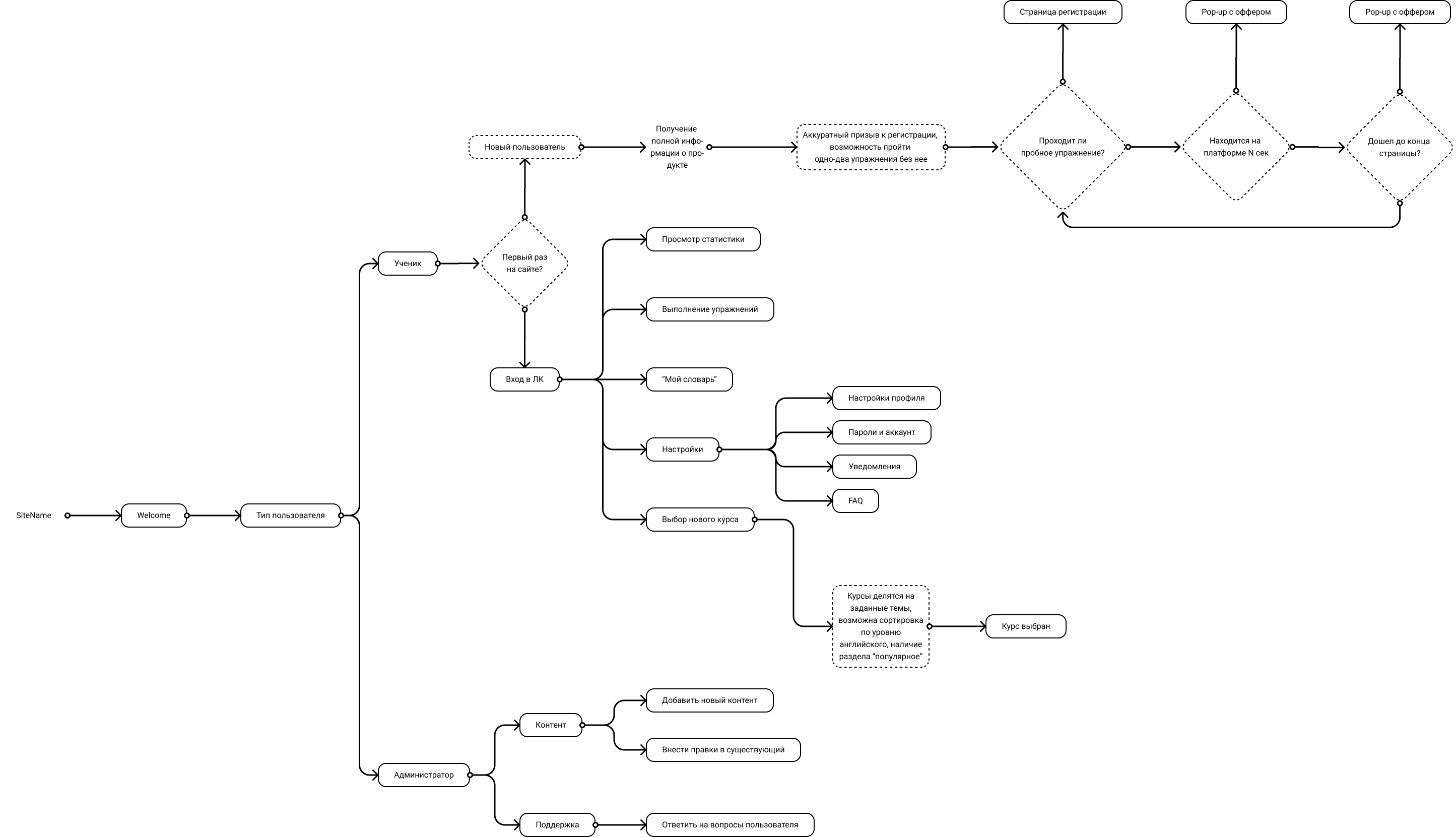


Рисунок 8 — User-ways

## Прототипы интерфейса

На рисунках представлены основные окна создаваемой системы:

* Рисунок 9 — Окно регистрации;
* Рисунок 10 — Окно авторизации;
* Рисунок 11 — Главная страница для неавторизованного пользователя;
* Рисунок 12 — Личный кабинет пользователя со статистикой и настройками.

Окна для выбора тем (курсов) в рамках MVP не используются, поскольку реализована будет только одна тема.

По мере внедрения контента, планируется дорабатывать интерфейс системы, чтобы потом сделать его максимально удобным и интуитивно понятным.

Впоследствии будут изменены цвета с пастельных на более контрастные, а также заменен шрифт на более универсальный (с поддержкой Кириллицы и Латиницы одновременно).

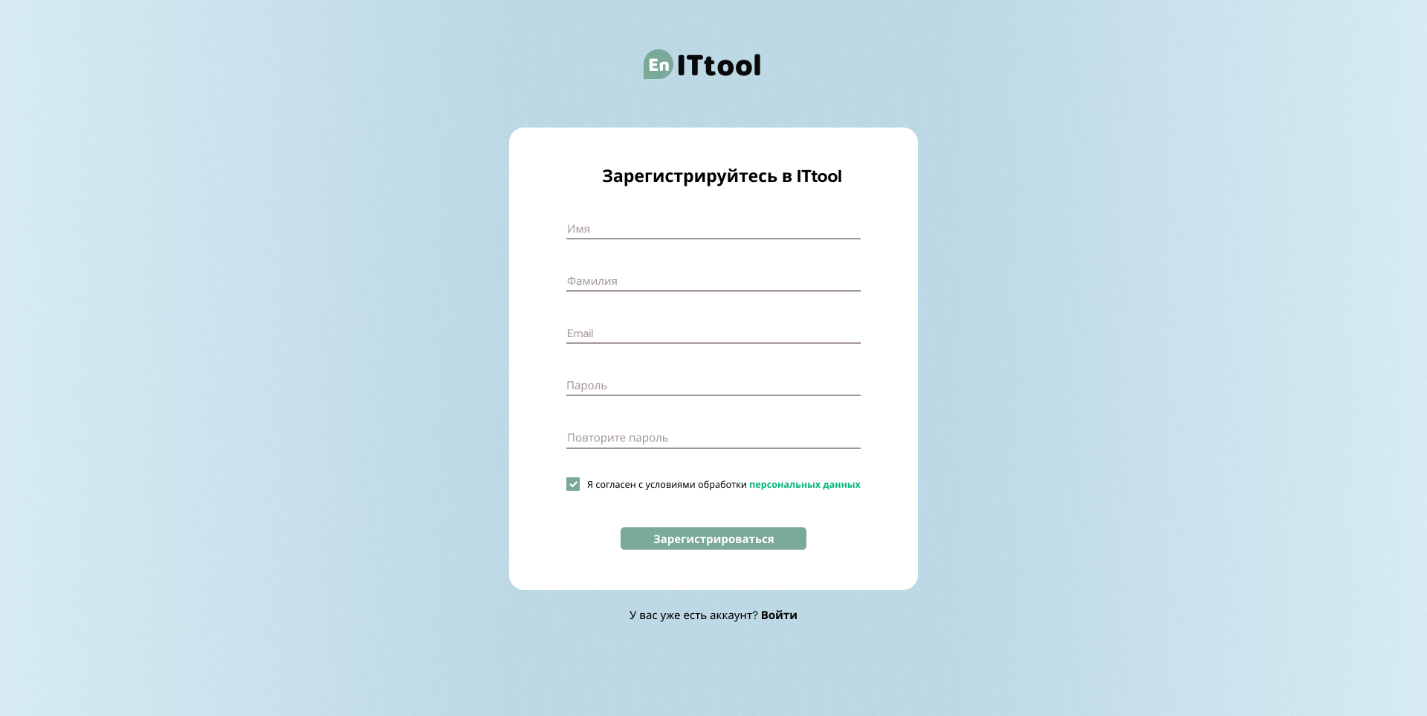


Рисунок 9 — Окно регистрации



Рисунок 10 — Окно авторизации

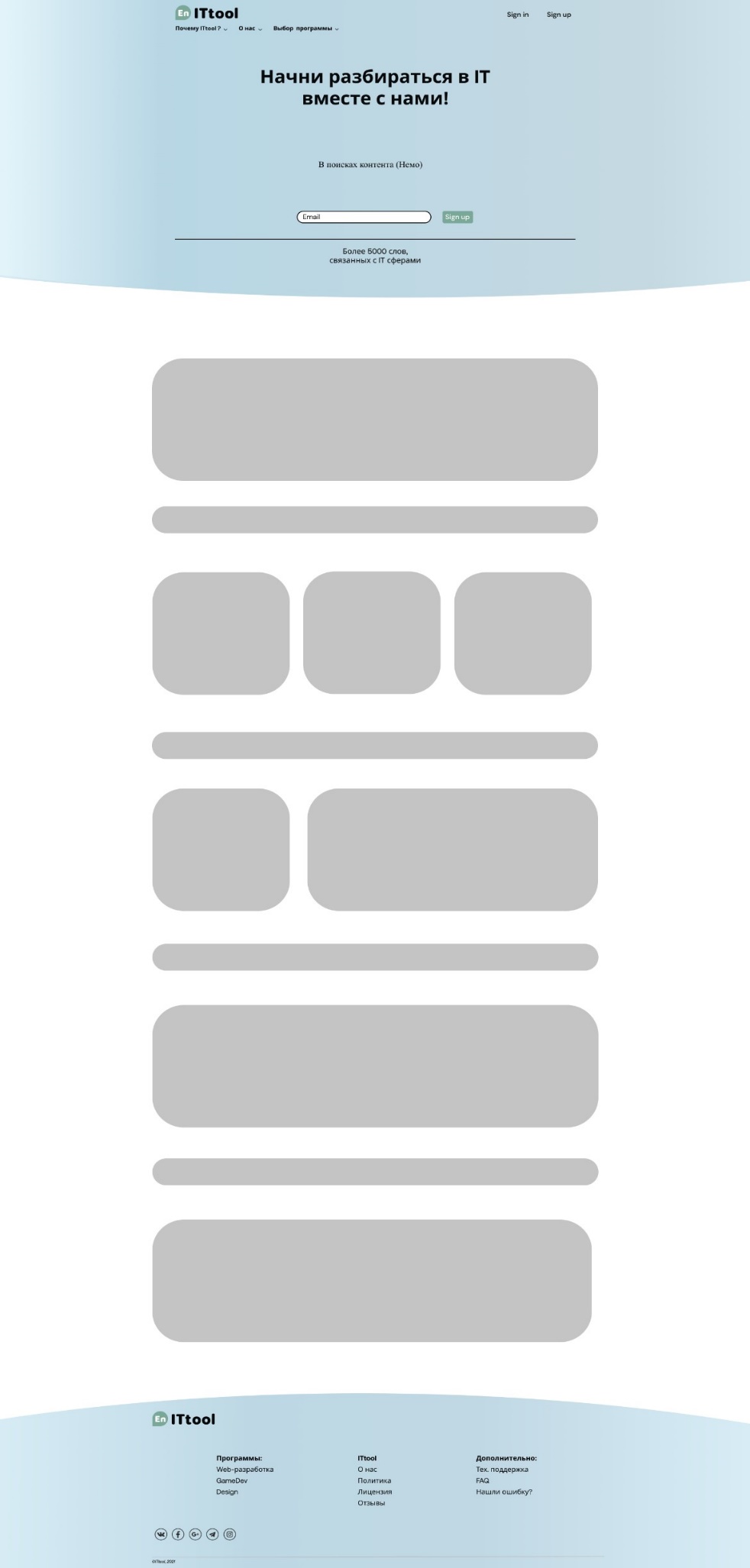
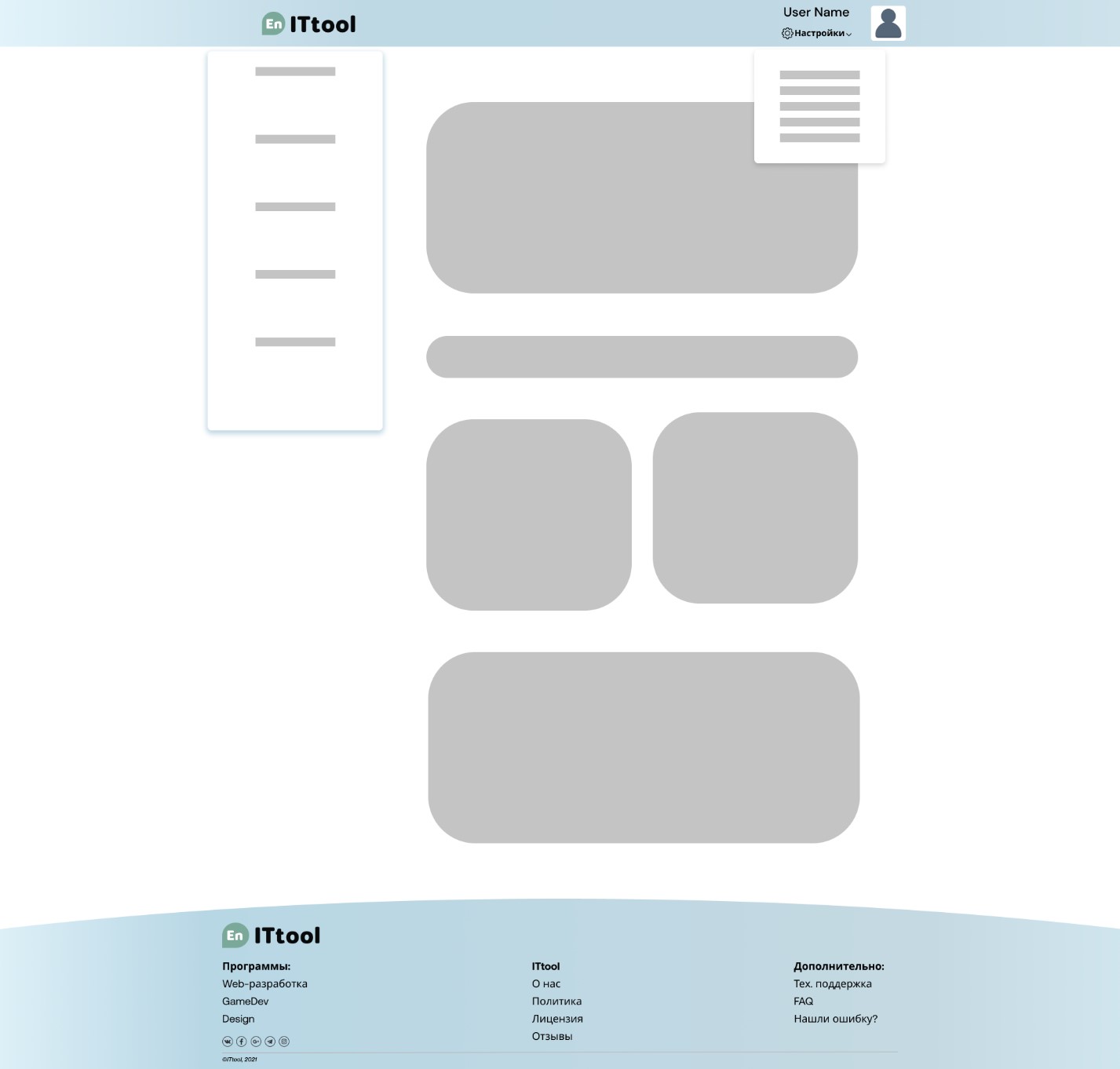


Рисунок 11 — Главная страница

****

*Рисунок 12 — Личный кабинет*

## Формирование контентной части

В качестве нашего первого курса мы выбрали HTML. Нам показалось, что это будет наилучшим вариантом, потому что он интересует множество людей и является хорошим началом для погружения в мир IT.

1. Введение

Мы кратко объясняем, что такое HTML, для чего он нужен и как используется. Рассказываем про такие термины, как тег, гипертекст, разметка и метаданные.

1. Структура HTML

Показываем строение файла HTML: <head>, <body>, <title>, <meta>.

1. Основные элементы и их названия.

Рассказываем об основных тегах и их назначении. Каждый тег иллюстрируем примером. Оставляем полное название тега.

1. Атрибуты.

Объясняем, что такое атрибут. Рассказываем про основные атрибуты, их использование и расположение в коде.

Разработка

## Написание программного кода

Создано и развернуто работающее web-приложение, клиентская часть которого состоит из 7 страниц. На серверной части существуют работающие базы данных MySQL и набор функций API, поддерживающие работу клиента

## Оформление и внедрение контентной части

HTML

**Chapter 1. Introduction. What the heck is HTML?**

*Wordlist:*

Hypertext is text which contains links to other texts. It allows a user to click a link and be redirected to a new page referenced by that link.

Tags are keywords which defines how web browser will format and display the content. Some elements come in pairs that indicate when some display effect is to begin and when it is to end (e.g., <title> … </title>).

Markup is a set of tags assigned to elements of a text to indicate their relation to the rest of the text or dictate how they should be displayed.

Metadata describes other data. It provides information about a certain item's content. A text document's metadata may contain information about how long the document is, who the author is, when the document was written, and a short summary of the document.

*Main:*

HyperText Markup Language (HTML) is the basic scripting language used by web browsers to render pages on the world wide web. It is the set of markup symbols or codes inserted into a file intended for display on the Internet. The markup tells web browsers how to display a web page's words and images.

At its core, HTML is a series of short codes typed into a text-file. These are the tags that power HTML’s capabilities. The text is saved as an HTML file and viewed through a web browser. The browser reads the file and translates the text into a visible form, as directed by the codes the author used to write what becomes the visible rendering. Writing HTML requires tags to be used correctly to create the author’s vision.

The tags are what separate normal text from HTML code. Tags are the words between what are known as angle-brackets, which allow graphics, images, and tables to appear on the webpage. Different tags perform different functions. The most basic tags apply formatting to text. As web interfaces need to become more dynamic, Cascading Style Sheets (CSS) and JavaScript applications may be used. CSS tells a web page the color of background, the fonts, and other visual decorations. JavaScript adds interactive behavior to web pages such as zooming, different animations and sliding through images.

**Chapter 2. Basic construction of an HTML page.**

Before you can start adding content to your document, there's a basic structure you need to set up in your file. This structure isn't only required for your document to be compliant but will also allow you to provide useful information about your document. The basic structure of any HTML document consists of the following sections or elements:

<!DOCTYPE> tag is the very first thing that every compliant web document should have. It's purpose is to inform the browser the type of document it's about to process.

[<html>](https://html.com/tags/html/) <!DOCTYPE> — This tag signals that from here on we are going to write in HTML code.

[<head>](https://html.com/tags/head/) — Tag that surrounds important content that is invisible to the user, but is important to the browser. Elements within this tag contain metadata about the page and links to stylesheets, scripts, etc.

[<body>](https://html.com/tags/body/) — This is where the **content of the page** goes.



Inside the <head> tag, there is one tag that is always included: <title>, but there are others that are just as important:

[<title>](https://html.com/tags/title/)— Tag tells the browser what to display as the page title at the top and tells search engines what the title of your site is.

<title>HTML Glossary</title>

[<meta>](https://html.com/tags/meta/)— This is where information about the document is stored: character encoding, name (page context), description.

<meta charset="UTF-8">  
  <meta name="description" content="Free Web tutorials">  
  <meta name="keywords" content="HTML, CSS, JavaScript">  
  <meta name="author" content="John Doe">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

**Chapter 3. Tags.**

HTML contains lots of predefined tag. Some of them are described below:

**Heading tag:** It is used to define the heading of html document. HTML headings are defined with the <h1> to <h6> tags. <h1> defines the most important heading. <h6> defines the least important heading.

<h1>Heading 1</h1>  
<h2>Heading 2</h2>  
<h3>Heading 3</h3>  
<h4>Heading 4</h4>  
<h5>Heading 5</h5>  
<h6>Heading 6</h6>

**Paragraph tag:** It is used to define paragraph content in html document. A paragraph always starts on a new line, and browsers automatically add some white space (a margin) before and after a paragraph.

<p> Statements... </p>

**Emphasis tag: Marks up emphasized text in a document.**

<p>You <em>have</em> to hurry up!</p>

**Italic tag:** It is used to write the content in italic format.

<i> Statements... </i>

**Anchor tag:** It is used to link one page to another page.

<a href="..."> Statements... </a>

**Unordered list** starts with the <ul> tag. Each list item starts with the <li> tag. The list items will be marked with bullets (small black circles) by default.

<ul>  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ul>

**Ordered list** starts with the <ol> tag. Each list item starts with the <li> tag.

The list items will be marked with numbers by default.

<ol>  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>

**Blockquote tag** specifies a section that is quoted from another source.

<blockquote cite="http://www.worldwildlife.org/who/index.html">  
For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.  
</blockquote>

**Horizontal rule tag** defines a thematic break in an HTML page (e. g. a shift of topic).

<p>HTML is the standard markup language for creating Web pages. HTML describes the structure of a Web page, and consists of a series of elements. HTML elements tell the browser how to display the content.</p>  
  
<hr>  
  
<p>CSS is a language that describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work, because it can control the layout of multiple web pages all at once.</p>

**Image tag:** It is used to add image element in html document.

<img src="img\_girl.jpg" alt="Girl in a jacket" width="500" height="600">

**Division tag** defines a division or a section in an HTML document. It is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript. Any sort of content can be put inside the <div> tag!

<div class="myDiv">  
  <h2>This is a heading in a div element</h2>  
  <p>This is some text in a div element.</p>  
</div>

**Chapter 4. Attributes.**

Attributes contain **additional pieces of information**. Attributes take the form of an opening tag and additional info is **placed inside**.

The <img> tag embeds an image into your HTML. Always found with the ‘src’ attribute, which tells the browser where to find the image. Note that the <img/> tag is self-closing.

<img src='mylocalimage.jpg'/>

An HTML element can have an id attribute to identify it. id elements should always be unique to that single element, and each element should never have more than one id.

<div id="my-box">This is my box! Put your text in some other box.</div>

Links tell the browser where to go using an href attribute, which stores a URL.

<a href="http://google.com">Google it!</a>

An element that is an immediate descendent of another element or nested within another element is called a child.

<ul id="parent">  
  <li id="child">I'm a child of parent!</li>  
</ul>

The style attribute is used to add styles to an element, such as color, font, size, and more.

<p style="color:red;">This is a red paragraph.</p>

The title attribute defines some extra information about an element. The value of the title attribute will be displayed as a tooltip when you mouse over the element.

<p title="I'm a tooltip">This is a paragraph.</p>

The HTML standard does not require quotes around attribute values. However, we recommend quotes in HTML.