Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Уфимский университет науки и технологий»

Факультет информатики и робототехники

Кафедра вычислительной математики и кибернетики

Отчет Лабораторной работе №1

«Вёрстка веб-страниц с использованием HTML/CSS на основе готового шаблона»

Выполнили:

студенты группы МО-321 Чащевой К.С. и Шемануев А.Е.

Проверил:

Юдинцев Б.С.

Уфа, 2023

**Задание**:

1. Выбрать готовый шаблон для оформления контентных страниц сайта.

2. Придумать тематическое содержание (контент) страниц сайта.

3. Разработать на основе выбранного шаблона 3-4 (в зависимости от количества человек в подгруппе) html-страниц, используя CSS-стили и HTML5- модель верстки.

4. На каждой странице в блоке уникального содержимого (теги ) реализовать различные стилизованные элементы: список, форма ввода, изображение с текстом, таблица с текстовым наполнением, предусмотреть блок ссылок на соцсети в «подвале».

5. Реализовать переходы на созданные страницы через навигационное меню (простыми тегами для ссылок).

**Теоретические сведения:**

**HTML** (от английского HyperText Markup Language) — это язык гипертекстовой разметки текста. Он нужен, чтобы размещать на веб-странице элементы: текст, картинки, таблицы и видео.

HTML состоит из тегов — команд, которые указывают браузеру, как отображать помещённый в них текст. Это и есть элементы веб-страницы. У каждого тега есть имя, которое заключается в угловые скобки < и >.

У каждого тега есть атрибуты. С их помощью можно передавать элементам веб-страницы дополнительные данные: размеры, уникальный id элемента, ссылки на изображения и так далее.

**CSS** — Cascading Style Sheets — это каскадные таблицы стилей. По сути — язык, который отвечает за описание внешнего вида HTML-документа. Подавляющее большинство современных веб-сайтов работают на основе связки HTML+CSS.

CSS используется для определения стилей (правил) оформления документов — включая дизайн, вёрстку и вариации макета для различных устройств и размеров экрана. У такого способа форматирования несколько достоинств:

* теги не дублируются;
* документ проще обслуживать;
* внешний вид всего сайта можно изменить централизованно, а не корректировать форматирование каждой странички.

Стили можно разметить внутри тега <HEAD> или использовать отдельный CSS-файл.

**Реализация HTML-кода.** Базовый код для html-страницы:

<!doctype html>

<html>

<head>

<meta charset="utf-8">

<title>Это заголовок страницы</title>

<link rel="stylesheet" href="css/styles.css">

</head>

<body>

Здесь будет реализация уникального контента страницы

</body>

</html>

Строка <link rel="stylesheet" href="css/styles.css"> в заголовке подключает css-файл с описанием стилей веб-страницы.

Базовый «каркас» **HTML5 документа**:

<body>

<header></header>

<nav></nav>

<main>

<aside></aside>

<article></article>

</main>

<footer></footer>

</body>

* <header> представляет собой группу вводного содержимого. Если

<header> является дочерним элементом <body>, то он определяет общий

заголовок для содержимого веб-страницы, т.е. здесь может быть логотип и /

или название сайта или компании. Если же <header> расположен как дочерний

элемент <article> или <section>, то он определяет конкретный заголовок для

отдельного раздела.

* <nav> содержит основные функции навигации для страницы.

Вторичные ссылки и т. д. не входят в навигацию.

* <main> предназначен для содержимого, уникального для этой

страницы. Используйте <main> только один раз на странице и размещайте

прямо внутри <body>. В идеале он не должен быть вложен в другие элементы.

* <article> содержит блок связанного содержимого (уникальный контент),

который имеет смысл сам по себе без остальной части страницы (например,

один пост в блоге, статья и т.д.). Каждый элемент <article> должен быть

выделен с помощью добавления заголовка (элементы <h1>-<h6>) в качестве

дочернего элемента:

<article>

<h2>Прогноз погоды</h2>

<p>Температура: +5 °C</p>

<p>Давление: 760 мм рт. ст.</p>

<p>Влажность: 2%</p>

</article>

* <section> подобен <article>, но больше подходит для

группирования одной части страницы, которая представляет собой одну часть

функциональности (например, мини-карту или набор заголовков статей и

сводок). Также считается хорошей практикой начинать каждый раздел

с заголовка. В зависимости от контекста вы можете разбить <article> на

несколько <section> или, наоборот, <section> на несколько <article>.

* <aside> содержит контент, который не имеет прямого отношения к

основному содержимому, но может содержать дополнительную информацию,

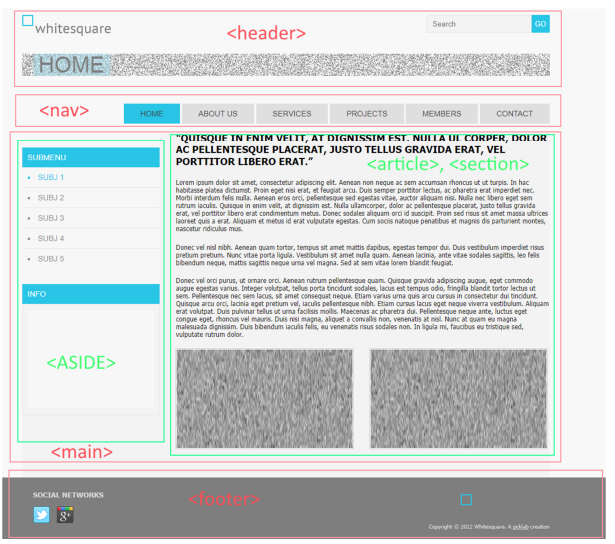
косвенно связанную с ним: словарь, биография автора, связанные ссылки и

т. д.).

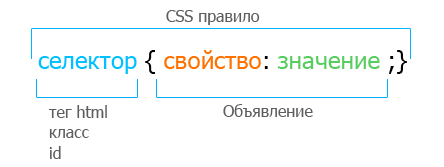
* <footer> представляет собой группу конечного контента для страницы,

т.н. «подвал». Обычно также содержит логотип компании и контактные

данные (ссылки на соцсети, почтовый адрес и т.п.)

Рис. 1 – Разметка базовых блоков на странице

У языка CSS относительно простой синтаксис. Сначала прописывается селектор — он выбирает конкретный элемент на странице. Потом, после фигурных скобок, указываются свойства со значениями — между ними ставится двоеточие. Сами свойства отделяются друг от друга точкой с запятой.



Например:

.my-class {

**background-color**: #999;

}

**БЭМ** (Блок, Элемент, Модификатор) — компонентный подход к вебразработке. В его основе лежит принцип разделения интерфейса на независимые блоки. Он позволяет легко и быстро разрабатывать интерфейсы любой сложности и повторно использовать существующий код, избегая «Copy-Paste».

В рамках данного подхода используется следующий шаблон для названия стилей:

имя-блока\_\_имя-элемента\_имя-модификатора\_значение-модификатора

**Ход работы:**

Изучив теоретический материал были написаны следующие страницы Html с использованием Css стиля:

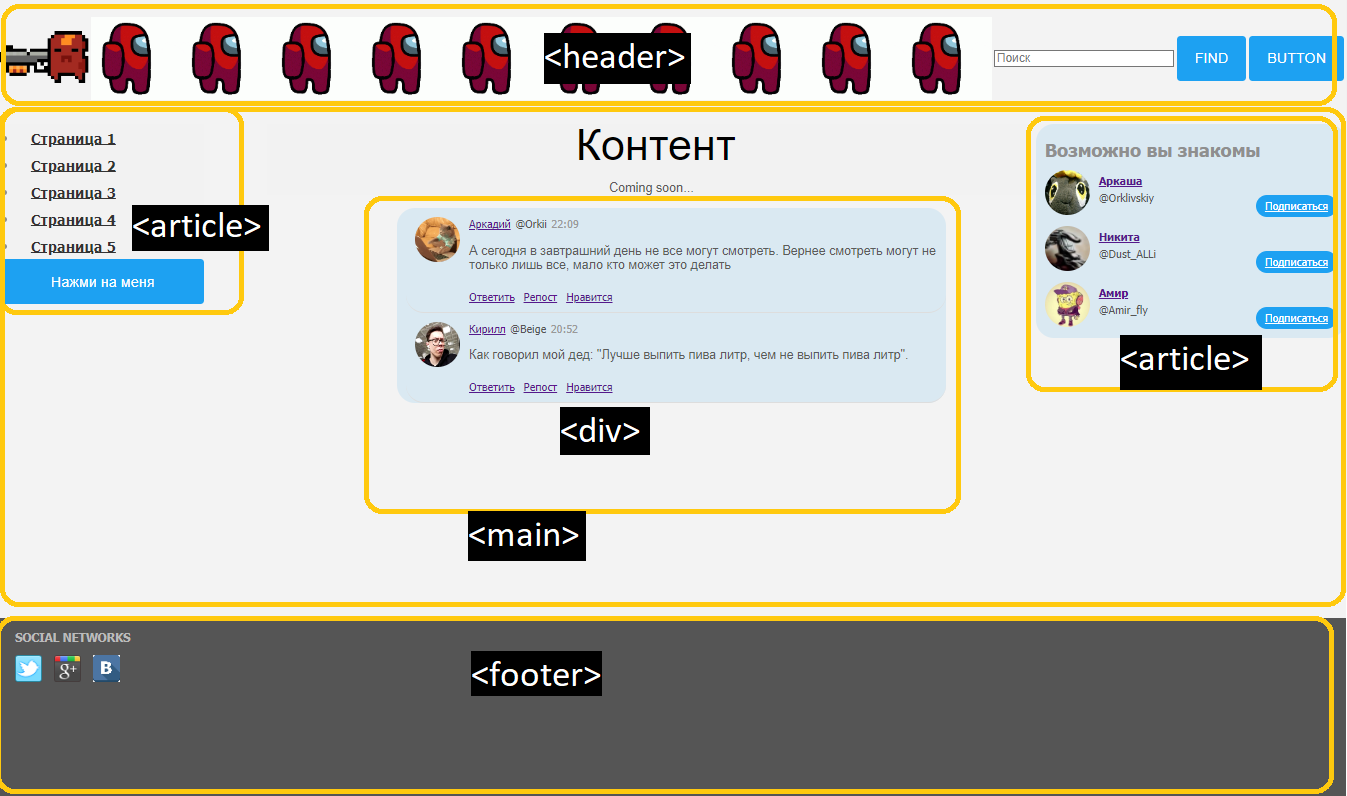
1 страница:

Рис. 2 – страница 1

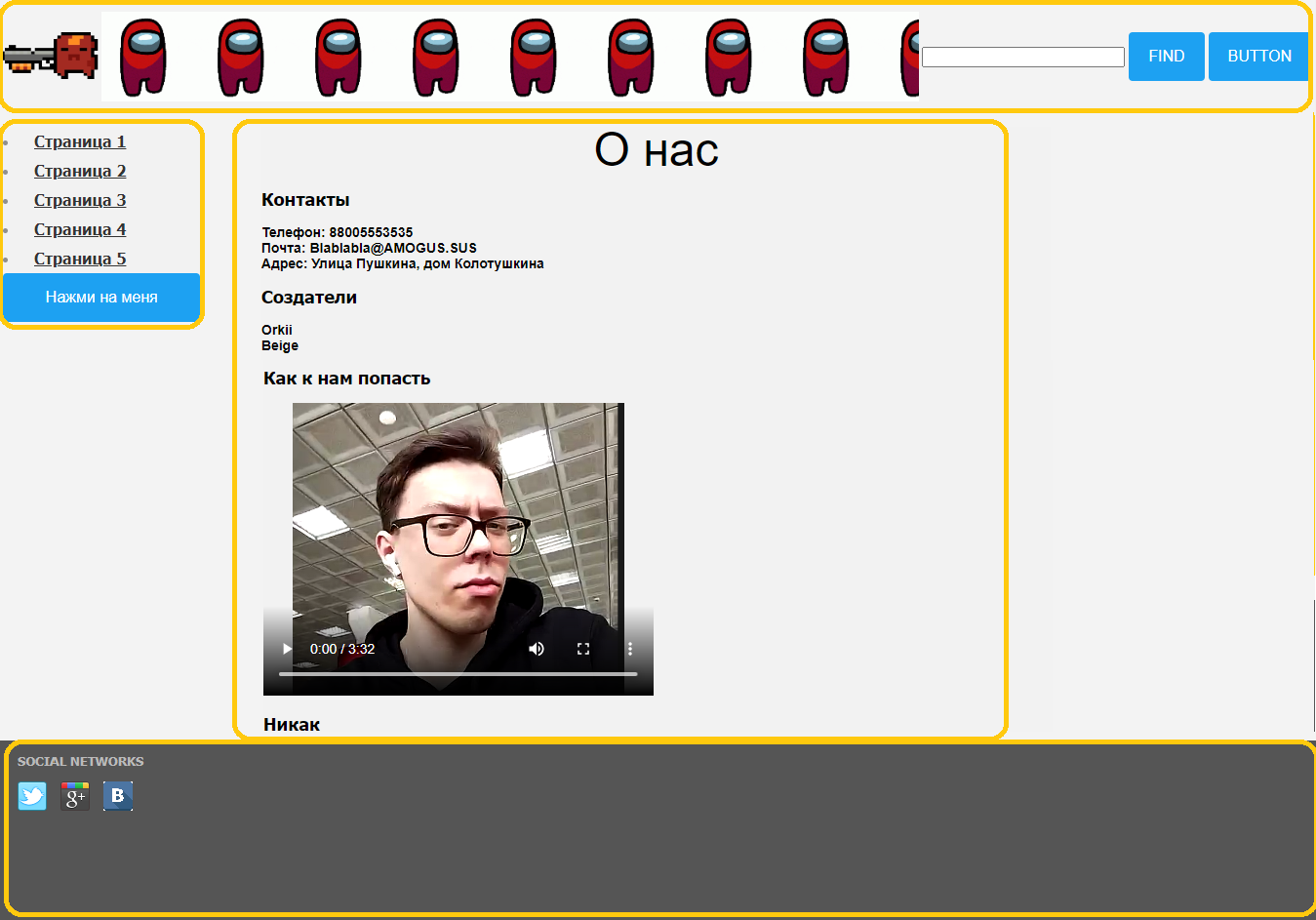
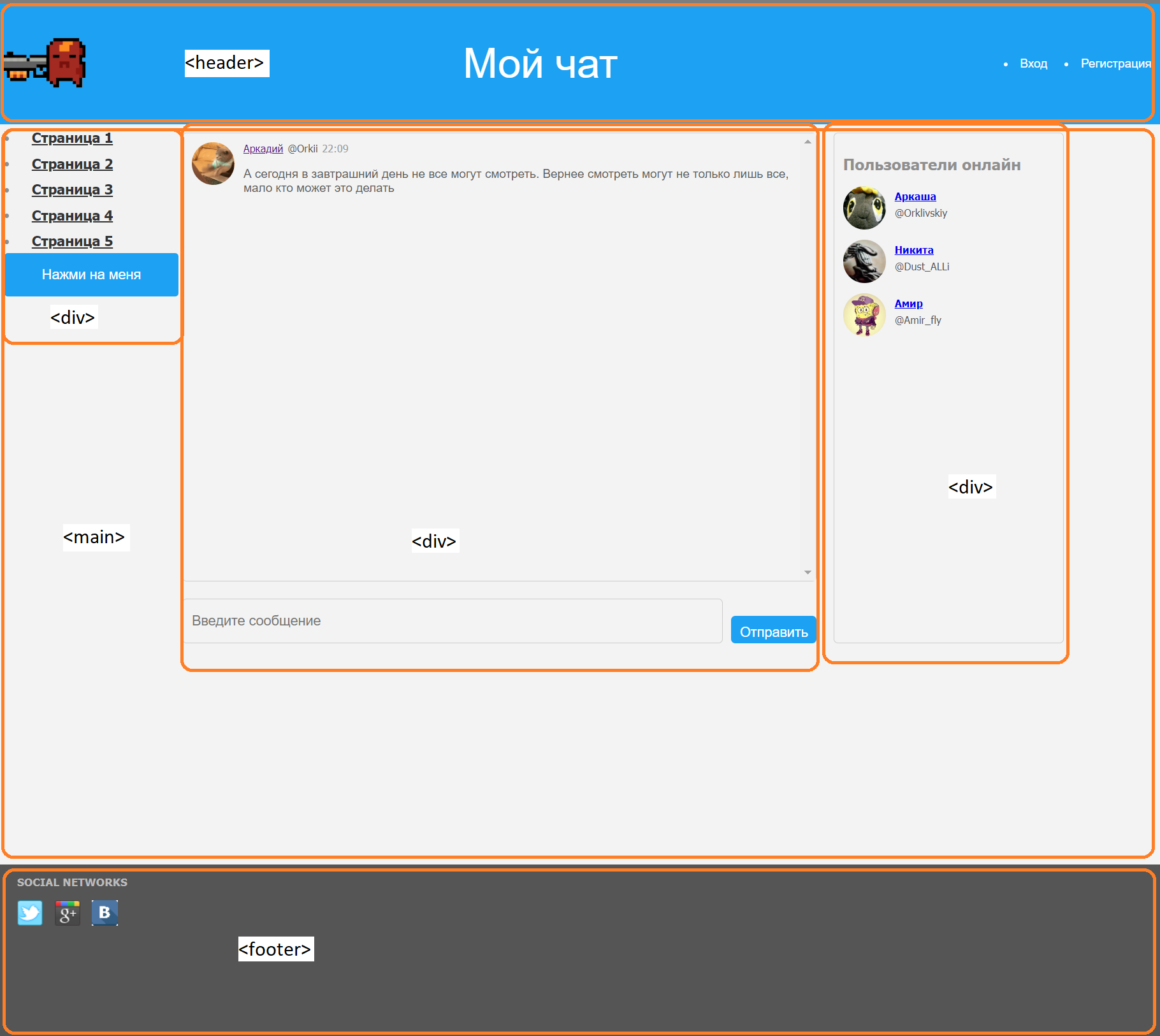
Рис. 3 – страница 2

Рис. 4 – страница 3

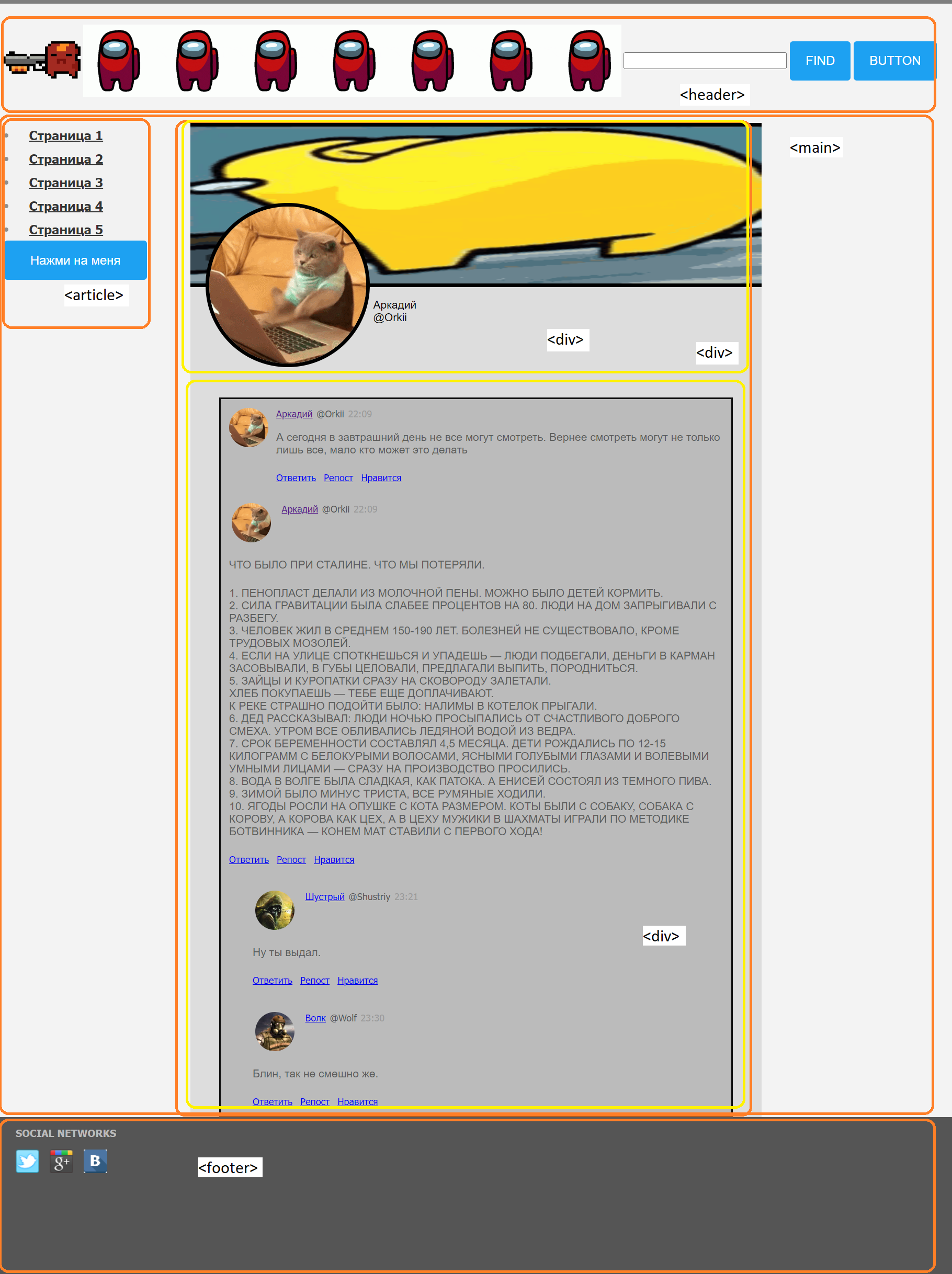
-

Рис. 5 – страница 4

|  |  |
| --- | --- |
| <header> | header {  padding: 20px 0;  width:100%;  height:100px;  } |
| <main> | main {  background:#f3f3f3;  min-height:1000px;  } |
| <footer> | footer{  left: 0;  bottom: 0;  width: 100%;  height: 80px;  } |
| <div class="navigation"> | .navigation {  width: 100%;  margin-left: 5px;  background-color: #f2f2f2;  } |
| <container> | .container {  max-width: 600px;  margin: 0 auto;  background-color: #DAE9F2;  border-radius: 20px;  padding-left: 10px;  } |
| <div class="news"> | .news {  margin-left:20%;  margin-right:20%;  background-color:#f2f2f2;  } |
| <div class="chat-window"> | .chat-window{  flex: 3;  margin-right: 20px;  } |
| <div class="user-list"> | .user-list{  flex: 1;  border: 1px solid #ccc;  border-radius: 5px;  padding: 10px;  } |
| <div class="userPage"> | .userPage {  margin-left:20%;  margin-right:20%;  margin-top:1%;  background-color:#ddd;  min-height: 1000px;  } |
| <table class="userPageTweetTable"> | .userPageTweetTable{  background-color:#111;  margin-left:5%;  margin-right:5%;  min-width:90%;  } |

**Вывод:**

В процессе выполнения лабораторной работы были изучены основные принципы и инструменты для создания и вёрстки веб-страниц на языках HTML и CSS. Был освоен базовый синтаксис и структура HTML и CSS, а также основные способы создания и размещения элементов на странице.