Минск 2024

**Содержание**

|  |  |  |
| --- | --- | --- |
| Введение | | 3 |
| 1. Постановка задачи | | 4 |
| 1.1. Обзор аналогичных решений | | 4 |
| 1.2. Техническое задание | | 5 |
| 1.3. Выбор средств реализации программного продукта | | 5 |
| 1.4. Выводы | | 5 |
| 2. Проектирование страниц веб-сайта | | 6 |
| 2.1. Выбор способа верстки | | 6 |
| 2.2. Выбор стилевого оформления | | 6 |
| 2.3. Выбор шрифтового оформления | | 6 |
| 2.4. Разработка логотипа | | 6 |
| 2.5. Разработка пользовательских элементов | | 7 |
| 2.6. Разработка спецэффектов | | 8 |
| 2.7. Выводы | | 8 |
| 3. Реализация структуры веб-сайта | | 9 |
| 3.1. Структура HTML-документа | | 9 |
| 3.2. Добавление таблиц стилей SCSS и CSS | | 10 |
| 3.3. Использование стандартов XML (SVG) | | 10 |
| 3.4. Управление элементами DOM | | 11 |
| 3.5. Выводы | | 11 |
| 4. Тестирование веб-сайта | | 12 |
| 4.1. Адаптивный дизайн веб-сайта | | 13 |
| 4.2. Кроссбраузерность веб-сайта | | 14 |
| 4.3. Выводы | | 14 |
| Заключение | | 15 |
| Список использованных литературных источников | | 16 |
| Приложение А Прототипы веб-страниц | | 17 |
| Приложение Б Макет структуры веб-сайта | | 20 |
| Приложение В Листинг НТML-документа | | 24 |
| Приложение Г Листинг SCSS и CSS | | 27 |
| Приложение Д Листинг XML-файлов | | 33 |
| Приложение Е Листинг JavaScript | | 35 |
|  |

**Введение**

В современной цифровой эпохе газетно-журнальные издания обрели новое измерение благодаря веб-сайтам. Они стали ключевым инструментом для распространения информации, предоставления актуальных новостей и удовлетворения потребностей читателей. Веб-сайты газетно-журнальных изданий играют существенную роль в обеспечении доступа к разнообразной информации и создании удобной платформы для чтения и взаимодействия с контентом.

Сайты газетно-журнальных изданий предоставляют уникальную возможность пользователям получить доступ к обширной гамме статей, репортажей, интервью и других материалов. Читатели могут легко найти интересующие их темы, вести поиск по ключевым словам, отслеживать последние новости и получать уведомления о важных событиях. Веб-сайты предлагают удобную навигацию, позволяющую читателям быстро перемещаться между разделами, журналами и статьями.

Одним из ключевых преимуществ веб-сайтов газетно-журнальных изданий является их мгновенная доступность. Читатели могут получать свежие новости и информацию в любое время, с любого устройства с подключением к Интернету. Благодаря адаптивному дизайну, веб-сайты корректно отображаются на разных устройствах, обеспечивая удобное чтение с мобильных телефонов или персональных компьютеров.

Задачи курсового проекта:

1. Создание макета веб-сайта и дизайна интерфейса.
2. Создание прототипов веб-страниц.
3. Разработка структуры веб-сайта.
4. Создание адаптивного дизайна для корректного отображения веб-сайта на разных устройствах с разной шириной экрана.
5. Тестирование веб-сайта в разных браузерах и на разных устройствах.

**1. Постановка задачи**

**1.1. Обзор аналогичных решений:**

* + 1. The New York Times ([https://www.nytimes.com](https://www.nytimes.com/))

На рисунке 1.1 представлен веб-сайт газетно-журнального издания The New York Times:

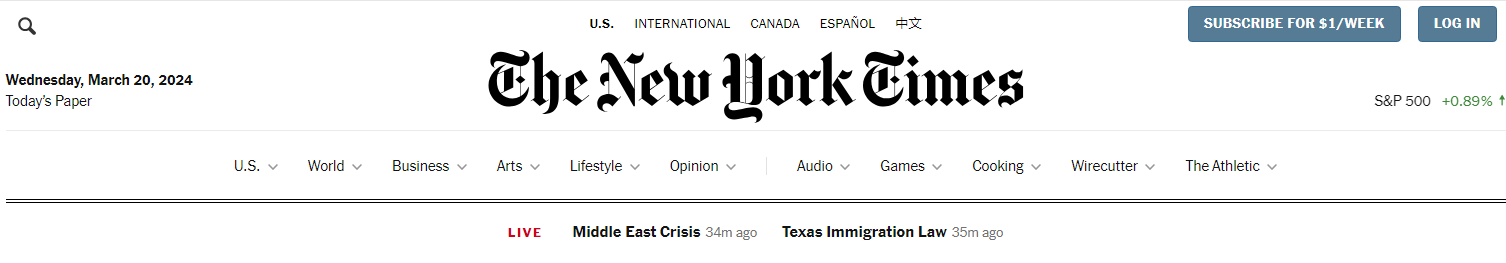


Рисунок 1.1 – Веб-сайт nytimes.com

**Преимущества**:

* + Один из ведущих газетно-журнальных изданий с широким спектром новостной информации.
  + Привлекательный и современный дизайн с акцентом на изображения и типографику.
  + Разделение контента на различные категории, такие как новости, мнения, бизнес, наука и технологии.
  + Система подписки для доступа к премиум-контенту и архиву статей.

**Недостатки**:

* Перегруженный интерфейс**:** Веб-сайт может показаться перегруженным и подавляющим из-за большого количества текста, изображений и рекламы.
* Сложная навигация**:** Структура меню может быть запутанной, что затрудняет поиск определенных разделов или статей.
* Медленная загрузка**:** Страницы могут загружаться медленно, особенно на мобильных устройствах.
  + 1. The Guardian ([https://www.theguardian.com](https://www.theguardian.com/))

На рисунке 1.2 представлен веб-сайт газетно-журнального издания The Guardian:



Рисунок 1.2 – Веб-сайт theguardian.com

**Преимущества:**

* + Простой и привлекательный дизайн с удобной навигацией по разделам и категориям.
  + Возможность подписки на ежедневные рассылки с самыми важными новостями.

**Недостатки:**

* Ограниченный бесплатный контент: Бесплатный контент на веб-сайте ограничен, что может разочаровать пользователей, которые не хотят платить за подписку.
* Не всегда точный: Некоторые статьи The Guardian подвергались критике за неточность или предвзятость.
* Отсутствие персонализации: Веб-сайт не предлагает значительной персонализации, что может затруднить пользователям поиск контента, соответствующего их интересам.

**1.2. Техническое задание:**

1.2.1. Главная страница:

- Разработка привлекательного и информативного макета главной страницы, включающего превью статей и новостей.

- Меню навигации для доступа к различным разделам.

1.2.2. Страница просмотра статьи/новости:

- Создание страницы для полного просмотра выбранной статьи или новости.

- Включение заголовка, текста статьи, фотографий и других медиа-элементов.

1.2.3. Адаптивный дизайн и кросс-браузерная совместимость:

- Разработка веб-сайта с адаптивным дизайном, который будет хорошо отображаться на различных устройствах, включая настольные компьютеры и мобильные устройства.

- Тестирование и обеспечение кросс-браузерной совместимости, чтобы сайт корректно работал в основных веб-браузерах, таких как Chrome, Yandex и Edge.

**1.3. Выбор средств реализации программного продукта:**

1. Для реализации программного продукта были выбраны языки: HTML, CSS/SСSS, XML.
2. Для управления элементами DOM и манипуляцией с данными будет использован JavaScript. Структура веб-сайта создана с помощью языка разметки HTML, дизайн веб-сайта оформлен с помощью CSS/SCSS. В XML формате хранятся данные.
3. Курсовой проект выполняется в редакторе кода Visual Studio Code. VS Code имеет большое количество расширений, которые значительно упрощают разработку, тестирование продукта.

**1.4. Выводы:**

В данном разделе пояснительной записки были рассмотрены аналогичные решения, проведен анализ их преимуществ и недостатков, что сформировало представление о внешнем виде сайта. Были рассмотрены средства реализации веб-страницы и редактор Visual Studio Code. Были поставлены требования, которые должны быть соблюдены в конечном продукте.

**2. Проектирование страниц веб-сайта**

**2.1 Выбор способа верстки**

Для позиционирования элементов на странице применяется FlexBox-верстка (<https://developer.mozilla.org/ru/docs/Learn/CSS/CSSlayout/Flexbox>).

Flexbox позволяет легко создавать сложные макеты и выравнивания элементов на веб-странице. Flexbox позволяет равномерно или по заданному размеру распределить доступное пространство между элементами в контейнере.

**2.2 Выбор стилевого оформления**

Для сайта выбрано минималистическое стилевое оформление. Благодаря минималистичному стилю веб-сайтов, контент становится центральным элементом визуальной композиции, выделяясь и привлекая больше внимания.

Минималистичный стиль облегчает навигацию по веб-сайту, поскольку отсутствие лишней информации делает путь по сайту более прозрачным.

**2.3 Выбор шрифтового оформления**

Montserrat – это современный шрифт с чистыми линиями и хорошей читаемостью, который придаст веб-сайту газетно-журнального издания элегантный современный вид.

**2.4 Разработка логотипа**

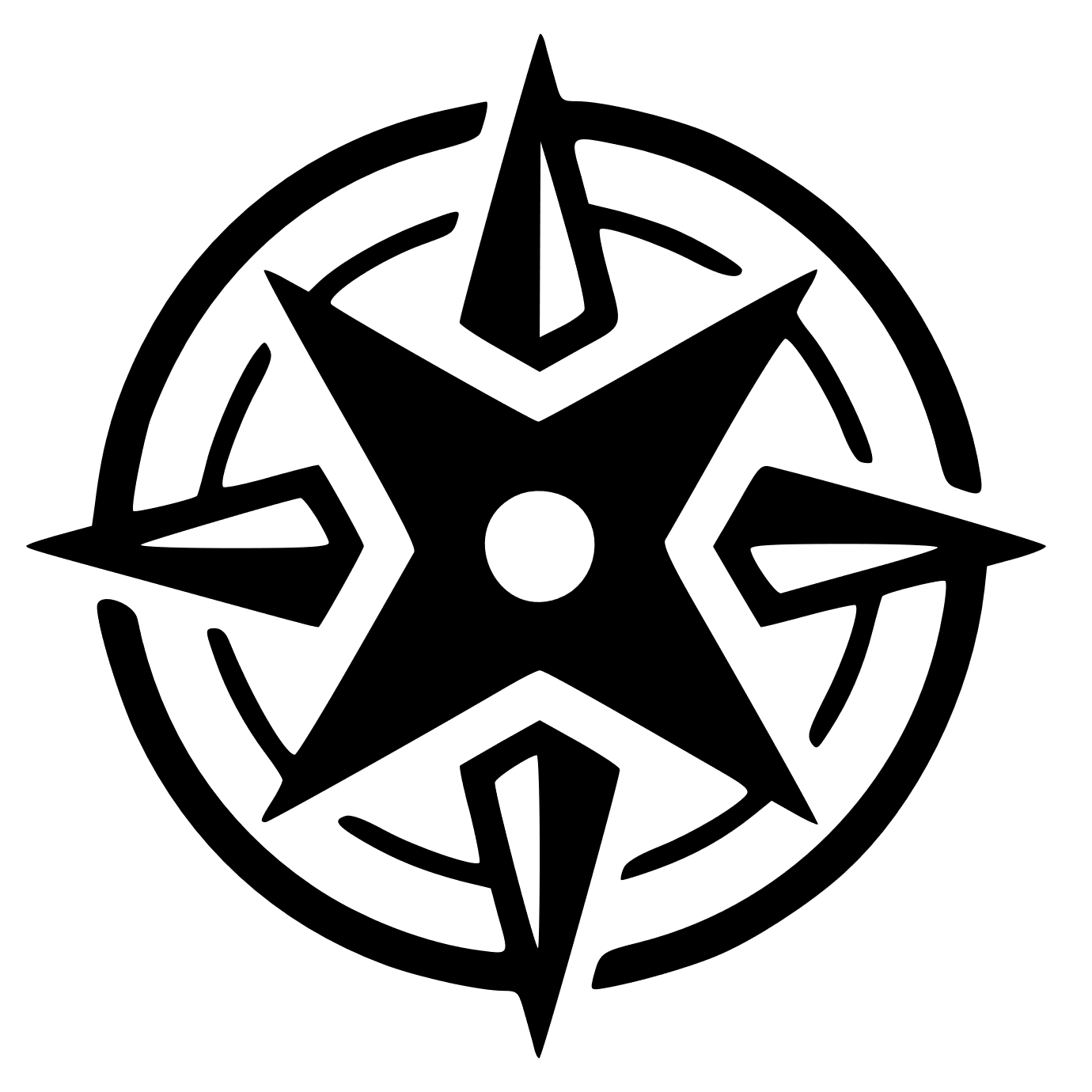
Логотип был разработан в векторном графическом редакторе Adobe Illustrator и будет использован на сайте в формате SVG. Как и сам сайт, логотип представлен в минималистичном стиле. Логотип представляет из себя иконку компаса, что символизирует о всенаправленности и объективности веб-сайта газетно-журнального издания.

Рисунок 2.1 – Логотип

**2.5 Разработка пользовательских элементов**

2.5.1 Статьи и новости: Предоставление пользователю актуальных статей и новостей, охватывающих различные темы, такие как политика, экономика, культура, спорт и другие. Каждая статья будет содержать заголовок, анонс, основной текст и, при необходимости, прикрепленные фотографии или видеоматериалы.

2.5.2 Форма регистрации: Реализация формы регистрации предоставит пользователю возможность зарегистрироваться или войти в существующий аккаунт на веб-сайте.

2.5.3 Адаптивный дизайн: Разработка веб-сайта с адаптивным дизайном, чтобы пользователи могли удобно просматривать контент на различных устройствах, включая компьютеры и смартфоны.



Рисунок 2.2 – Пример адаптивности веб-сайта: Выпадающее меню

**2.6 Разработка спецэффектов**

На веб-сайте будут реализованы следующие спецэффекты:

1. Переходы и анимации: Добавление плавных переходов между страницами или разделами сайта поможет создать более гармоничное впечатление.
2. Hover-эффекты: Применение стилей или анимаций при наведении курсора на элементы, таких как ссылки, изображения или заголовки, может помочь выделить их и привлечь внимание пользователя.
3. Иллюстрации и графические элементы: Использование

качественных иллюстраций или графических элементов может добавить эстетическое восприятие и уникальность к сайту газетно-журнального издания.

**2.7 Вывод**

Было определено расположение всех элементов на страницах. Так же была выбрана цветовая гамма сайта, используемое шрифтовое оформление, элементы пользовательского интерфейса и эффекты.

# 3. Реализация структуры веб-сайта

## **3.1. Структура HTML-документа**

Структура HTML-документа выполнена по стандарту HTML-5. В листинге 3.1 представлена декларация документа и основное содержимое тега head.

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <title>Main</title>      <link rel="stylesheet" href="/Css/Header.css">      <link rel="stylesheet" href="/Css/MainStyles.css">      <link rel="stylesheet" href="/Css/Footer.css">      <link rel="stylesheet" href="/Css/MediaRules.css">      <link rel="stylesheet" href="/Css/Animations.css">      <link rel="stylesheet" href="/Css/Links.css">      <link rel="icon" href="/Images/logo.ico"/>  </head>  </html> |

Листинг 3.1 – Декларация и тег head

Элемент <!DOCTYPE> предназначен для указания типа текущего документа.  Атрибут lang тега html устанавливает язык документа. Метатеги, расположенные в теге head устанавливают текущую кодировку, область просмотра и её начальное масштабирование. Тег title устанавливает название страницы. В тегах link происходит подключение каскадных таблиц стилей.

При построении структуры тела документа были использованы семантические теги: footer, section.

На всех страницах присутствует навигационная панель. Навигационная панель представляет из себя секцию, которая содержит логотип, кнопку бургера для мобильного меню и список с ссылками для навигации по странице.

Шапка сайта является адаптированной под мобильные устройства. При достижении определенных размеров экрана браузера, лишние элементы скрываются и остается только логотип с названием компании и иконка бургера. Структура навигационного меню представлена в приложении В.

В теге footer находятся элементы навигации по сайту а также форма для подписки на новости. Навигационная панель и footer сайта будут одинаковыми для всех веб-страниц. Структура тега footer представлена в приложении В.

## **3.2. Добавление таблиц стилей SCSS и CSS**

Все CSS и SCSS файлы находятся в отдельной папке от html файлов сайта для удобства использования. Подключение стилей осуществлялось внешне с помощью тега link. Также, при разработке таблиц стилей использовался препроцессор SCSS. Особенности, реализованные в SCSS упрощают работу при создании таблиц стилей. В частности была использована вложенность. Пример использования SCSS при стилизации навигационного меню в теге header представлен в листинге 3.2.1.

|  |
| --- |
| ul.nav {          float: right;          list-style: none;          li {              display: inline;              margin-left: 59px;              cursor: pointer;              font-weight: 400;          }      } |

Листинг 3.2. Пример использования SCSS

В листинге 3.2.2 представлен сгенерированный препроцессором CSS-код.

|  |
| --- |
| header ul.nav {    float: right;    list-style: none;  }  header ul.nav li {    display: inline;    margin-left: 59px;    cursor: pointer;    font-weight: 400;  } |

Листинг 3.3 – Сгенерированный препроцессором CSS-код

## **3.3. Использование стандартов XML (SVG)**

В курсовом проекте масштабируемая векторная графика (SVG) использовалась в логотипe.

Данные для новостей хранятся в формате XML и помещаются на страницу с помощью JavaScript. Для примера в листинге 3.3 будет приведен один раздел.

|  |
| --- |
| <card>          <id>1</id>          <card-image>../Images/image1.svg</card-image>          <card-type>ILLUSTRATION</card-type>          <card-title>JAPAN HOUSE OPENS IN MOUNTAINSIDE TO <br/>FOSTER PEAK CREATIVITY.</card-title>          <card-author>BY RETA TORPHY</card-author>      </card> |

Листинг 3.4 – Хранение данных новостей в XML

## **3.4. Управление элементами DOM**

Функционал на страницах был реализован с помощью JavaScript. Пример представлен в листинге Г. На всех страницах в нижнем колонтитуле при нажатии на кнопку загружается файл с введенными данными в формате .json и .xml. В листинге 3.4 приведен пример скачивания данных в XML формате.

|  |
| --- |
| let xmlLink = document.createElement("a");      xmlLink.href = xmlUrl;      let timestamp = new Date().getTime();      xmlLink.download = "data\_email\_" + timestamp + ".xml"; |

Листинг 3.5 – Пример скачивания данных в XML

## **3.5. Вывод.**

На данном этапе была успешно завершена реализация основной структуры веб-сайта, включая главную страницу и остальные страницы, заданные в начальном этапе проекта. Все элементы и блоки сайта были тщательно интегрированы в соответствии с их расположением и предназначением, что было определено еще на этапе создания прототипа.

Также важной частью данного этапа является верстка страницы, где были определены блоки контента, их расположение и структура, а также использование CSS для стилизации и оформления веб-страниц.

Благодаря успешной реализации структуры сайта, была создана полноценная и функционирующая веб-платформа, которая полностью соответствует всем требованиям, установленным на этапе создания прототипа. Веб-сайт готов к запуску и готов к использованию пользователями.

# 4. Тестирование веб-сайта

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

Для того, чтобы веб-сайт мог корректно отображаться на любых устройствах, был использован адаптивный дизайн. Была создана такая версия сайта, которая будет оптимально отображаться на экранах различных устройств, будь то мобильный телефон или настольный компьютер.

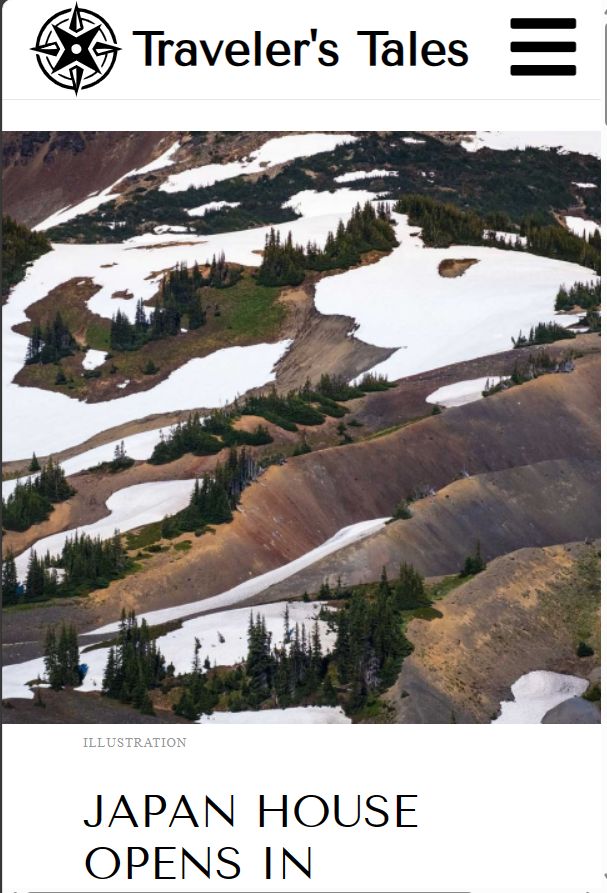
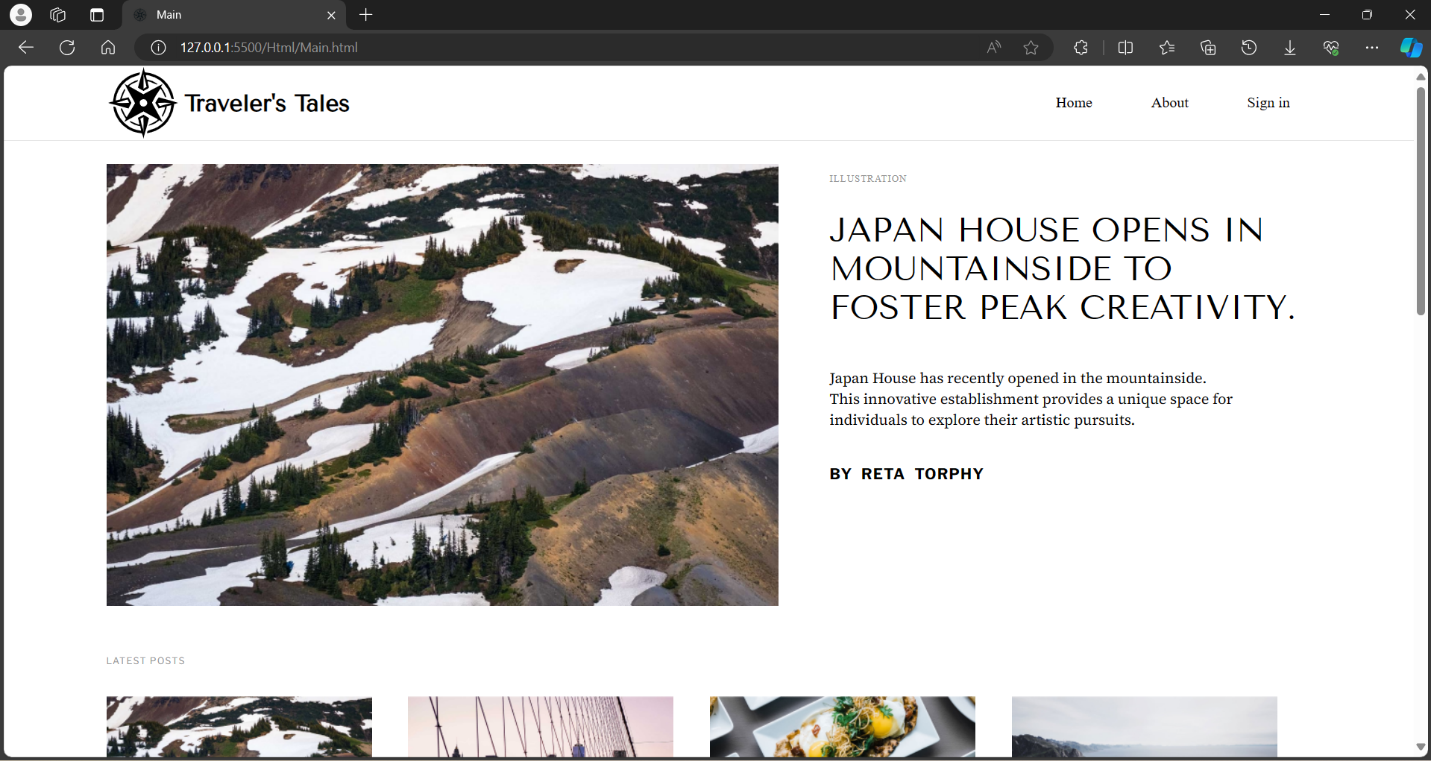
Для реализации адаптивности были применены медиа-запросы, которые определяют стили для страницы на основе размера окна браузера. В данном случае, были использованы медиа-запросы для размера экрана 1000px. На рисунке 4.1 представлен пример того, как главная страница сайта адаптируется для устройства шириной меньше 1000px. Благодаря адаптивному дизайну, веб-сайт будет удобным и доступным для пользователей, независимо от того, какое устройство они используют для просмотра

Рисунок 4.1 - Адаптивность

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

Для обеспечения кроссбраузерности веб-сайта были соблюдены стандарты веб-разработки в HTML, CSS и JavaScript. Соблюдение стандартов помогает веб-сайту работать более надежно на разных браузерах. Примеры корректного отображения веб-сайта в браузерах Yandex Browser, Microsoft Edge и Google Chrome представлены на соответствующих рисунках 4.3, 4.4 и 4.5.

Благодаря использованию техники кроссбраузерности удалось обеспечить корректное отображение веб-сайта на различных браузерах и устройствах, что позволяет максимально расширить аудиторию пользователей и повысить качество пользовательского опыта.

Рисунок 4.2 – Веб-сайт в браузере Microsoft Edge

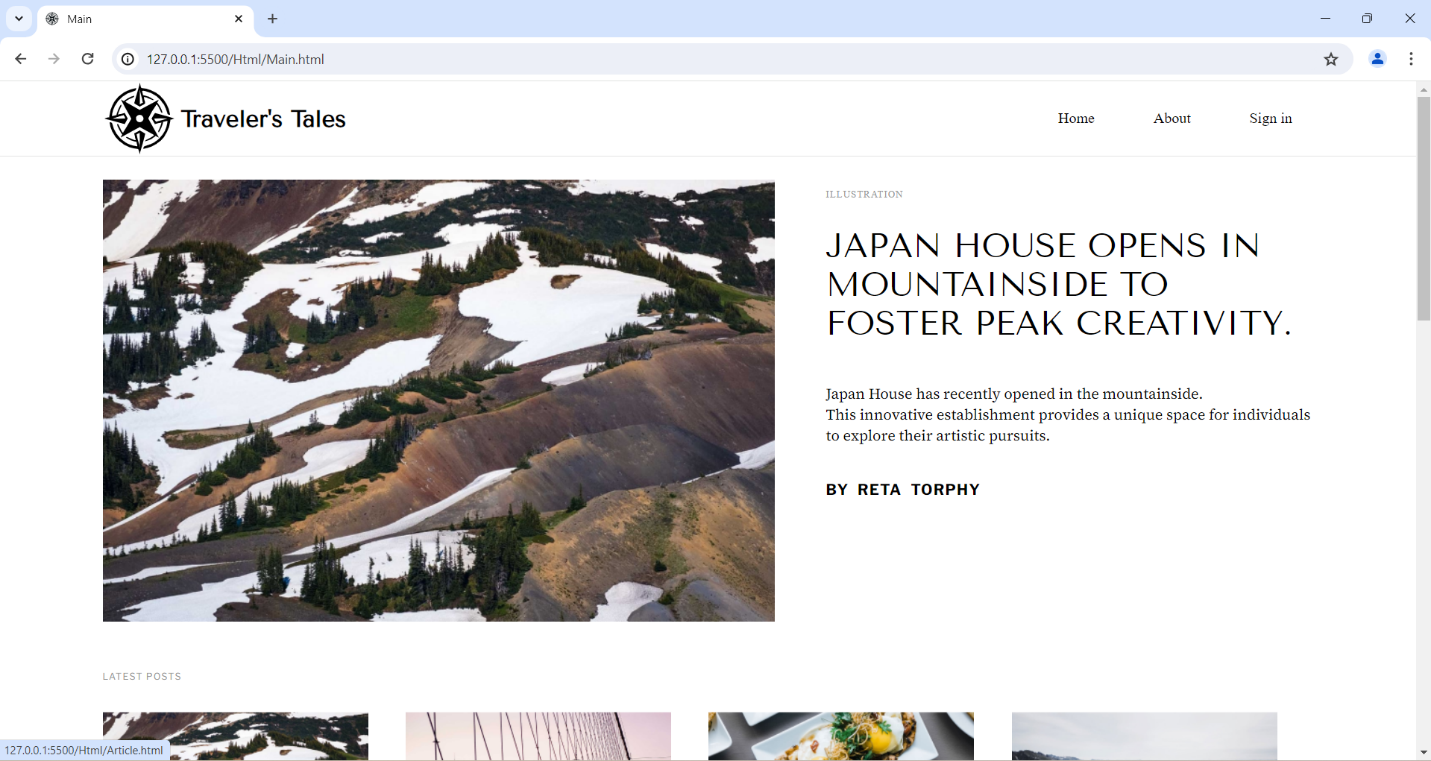


Рисунок 4.3 – Веб-сайт в браузере Google Chrome

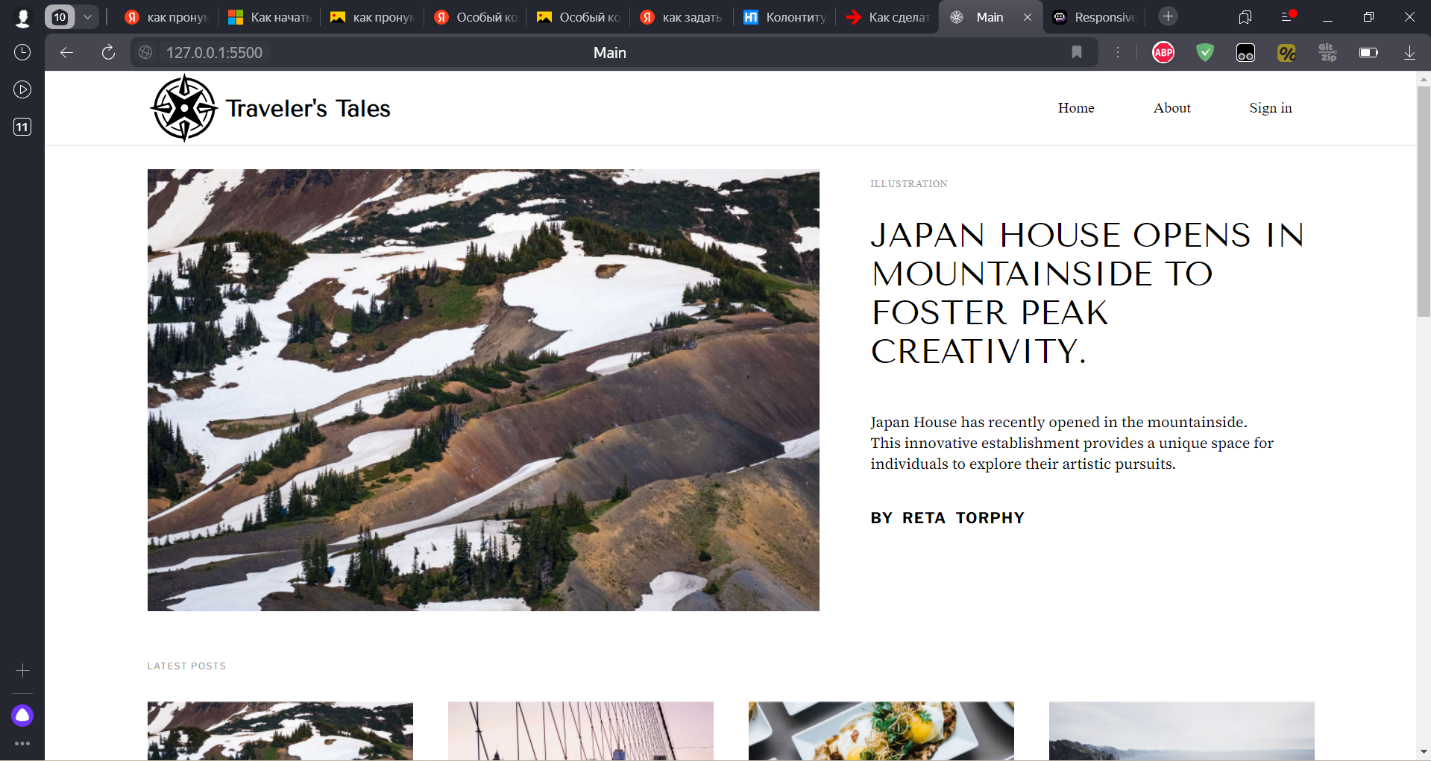


Рисунок 4.4 – Веб-сайт в браузере Yandex Browser

**4.3. Руководство пользователя**

Главная страница является точкой входа на веб-сайт. Вверху страницы располагается горизонтальное меню в десктопной версии или выпадающее меню-«бургер» на мобильной версии. В десктопной версии меню содержит ссылки на все остальные страницы, а так же логотип компании.

Страницы регистрации и входа содержат поля для, соответственно, регистрации и входа на страницу.

Страница новости содержит описание новости и фотографии к ней.

Страница «ABOUT» содержит информацию о газетно-журнальном издании Traveler’s Tales и его истории.

Также, все страницы содержать форму подписки на новости, в которую необходимо ввести email и нажать кнопку “SEND”.

## **4.4. Вывод.**

Было проведено тестирование веб-сайта в различных браузерах и на различных устройствах. Так же мы разработали краткое руководство пользователя для адаптации по сайту и взаимодействия с его ресурсами.

# Заключение

В рамках выполнения курсового проекта был создан веб-сайт газетно-журнального издания, который отвечает всем первоначальным требованиям. Для успешной реализации проекта был проведен анализ существующих веб-сайтов с аналогичной тематикой и выбраны наиболее эффективные решения.

Кроме того, были выполнены работы по выбору метода верстки, стилевому оформлению и разработке логотипа, пользовательских элементов и спецэффектов. Также были добавлены адаптивность и кроссбраузерность для корректного отображения веб-сайта на различных устройствах и браузерах.

Все работы были сопровождены тестированием, а результатом проекта стал веб-сайт, который предоставляет пользователям всю необходимую информацию. Кроме того, было разработано руководство пользователя, которое поможет пользователям быстро освоить веб-сайт и получить полезную информацию.

**Список использованных литературных источников**

1. HTML [Электронный ресурс]. – Режим доступа http:/htmlbook.ru/html. – Дата доступа: 12.04.2024.

2. Документация Scss [Электронный ресурс]. – Режим доступа: https://sass-scss.ru/documentation. – Дата доступа: 17.04.2024.

3. Руководство по графическом редактору Figma [Электронный ресурс]. – Режим доступа: https://vc.ru/design/184941-figma-vse-chto-vam-nuzhno-znat. Дата доступа: 23.03.2024.

4. Flexbox [Электронный ресурс]. – Режим доступа: https://developer.mozilla.org/ru/docs/Learn/CSS/CSSlayout/Flexbox. – Дата доступа: 20.04.2024

5. JavaScript DOM [Электронный ресурс]. – Режим доступа: https://developer.mozilla.org/enUS/docs/Web/API/Document\_Object\_Model. Дата доступа: 25.04.2024 .

6. XML [Электронный ресурс]. – Режим доступа: https://doka-guide.vercel.app/tools/xml/https://doka-guide.vercel.app/tools/xml/. Дата доступа: 22.04.2024

# Приложение А

# Прототипы веб-страниц

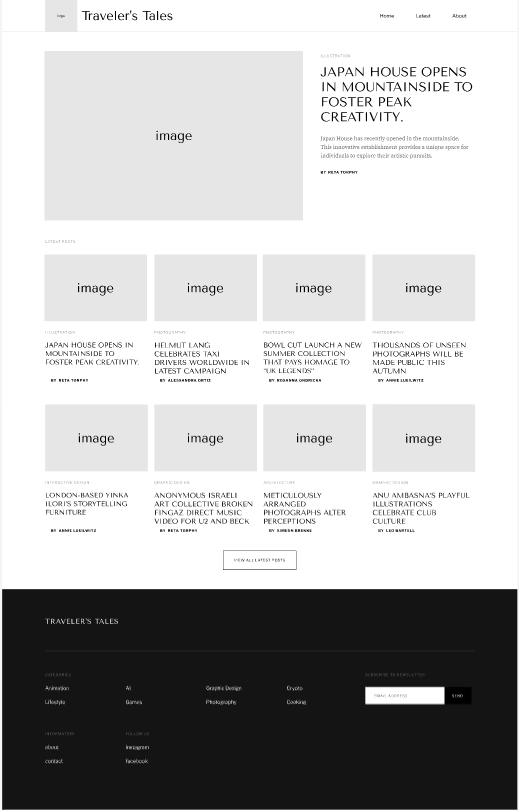


Рисунок 1 – Главная страница

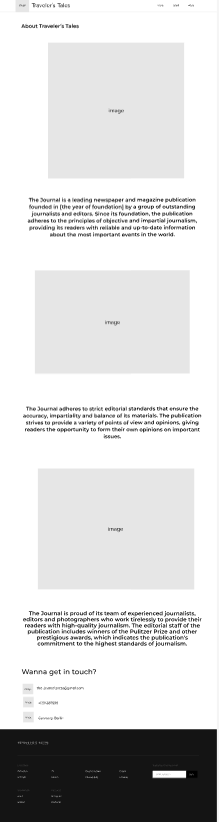


Рисунок 2 – О нас

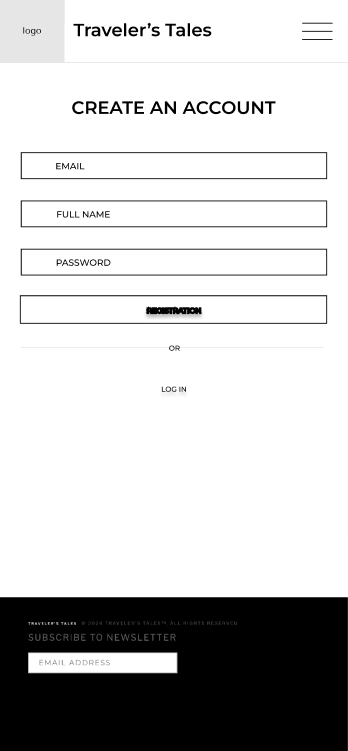
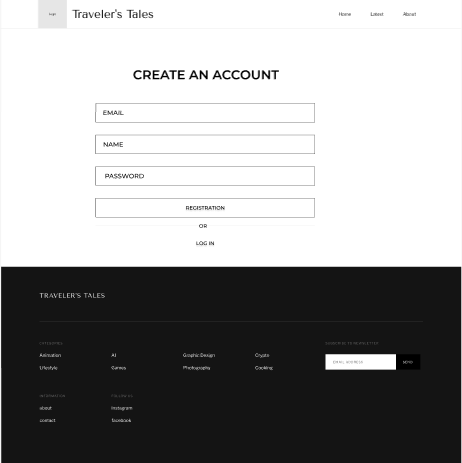


Рисунок 3 – Страница регистрации

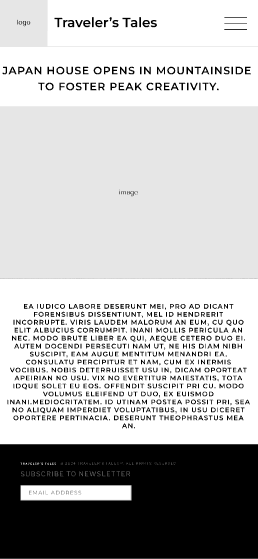
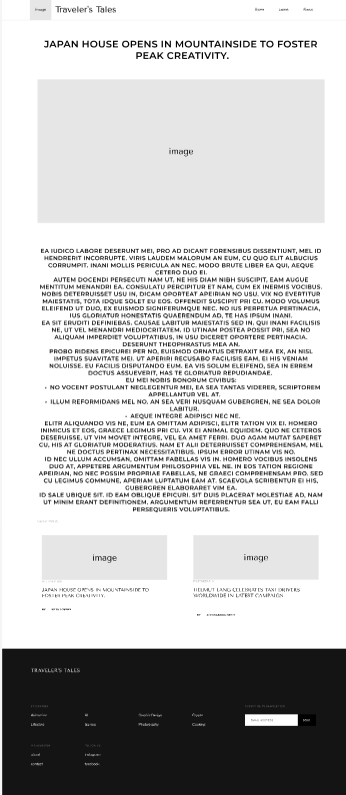


Рисунок 4 – Страница новости

# Приложение Б

# Макеты страниц

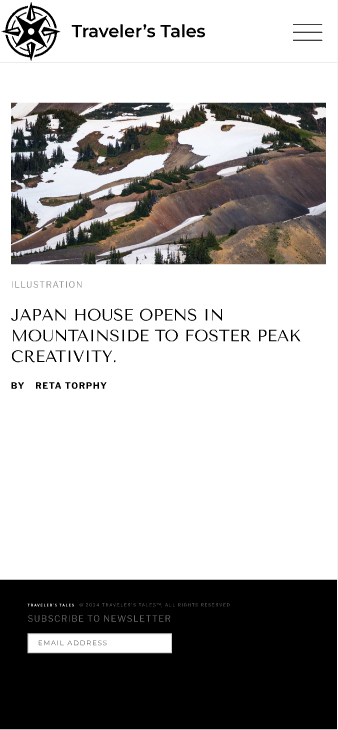
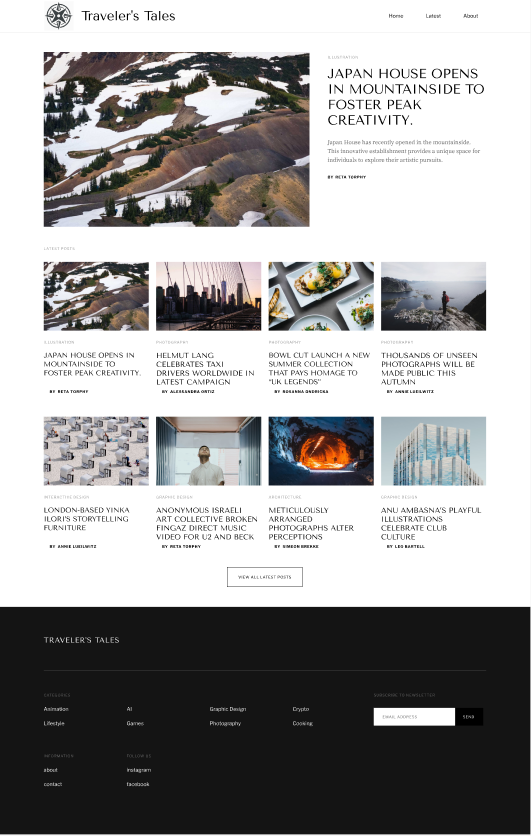


Рисунок 1 – Главная страница



Рисунок 2 – О нас

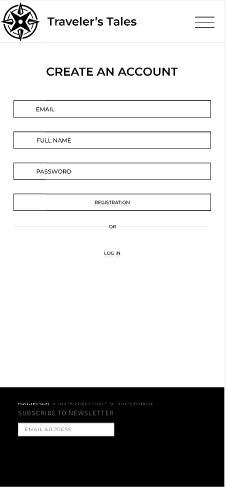
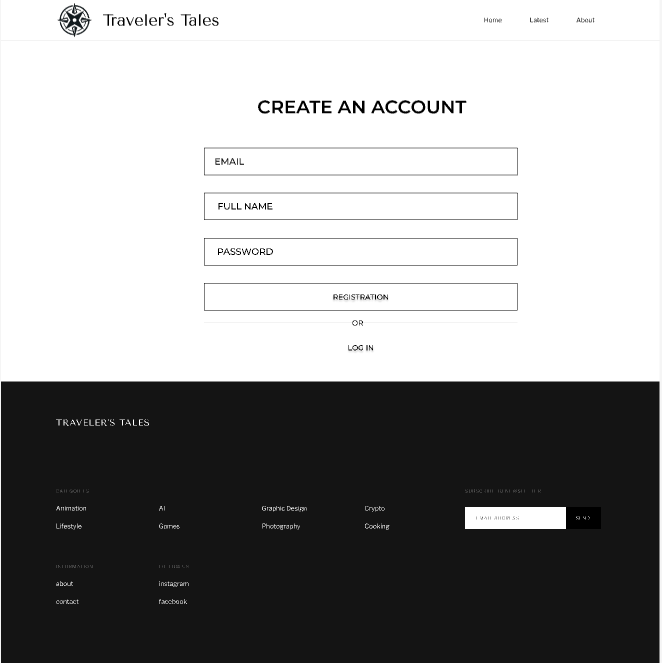


Рисунок 3 – Страница регистрации

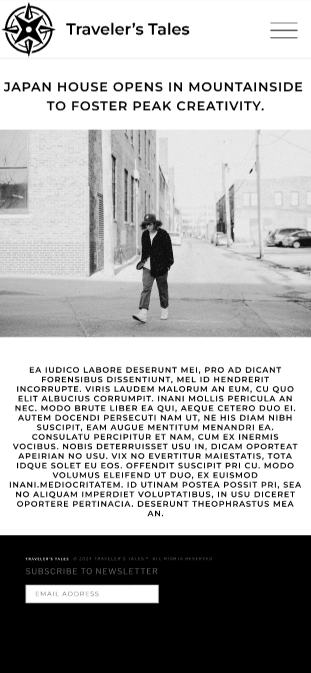
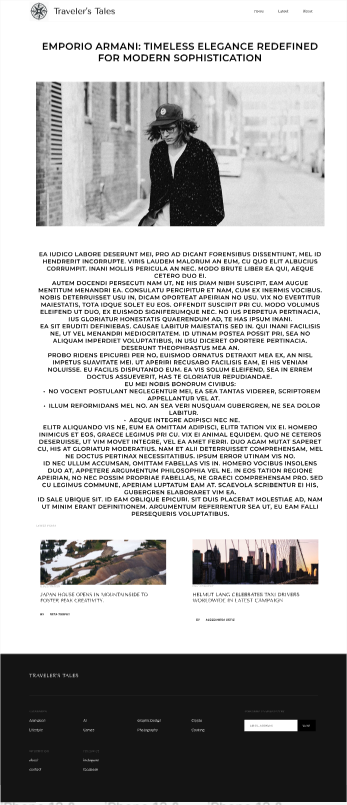


Рисунок 4 – Страница новости

# Приложение В

# Листинг HTML-документа

|  |
| --- |
| <header>  <div class="top">  <div class="container">  <span class="logo">  <a href="Main.html" class="logo-link">  <img src="/Images/logo.svg" height="80" width="80" alt="logo">  <p class="logo-text">Traveler's Tales</p>  </a>  </span>  <ul class="nav">  <li><a href="Main.html">Home</a></li>  <li><a href="About.html">About</a></li>  <li><a href="Registration.html">Sign in</a></li>  </ul>  </div>  </div>  <div class="container1">  <span class="logo">  <a href="Main.html" class="logo-link">  <img src="/Images/logo.svg" height="80" width="80" alt="logo">  <p class="logo-text">Traveler's Tales</p>  </a>  </span>  <h1>  <a href="#menu"><i class="fas fa-bars"></i></a>  </h1>  <div class="popover" id="menu">  <div class='content'>  <a href="#" class="close"></a>  <div class='nav'>  <ul class='nav-list'>  <div class='nav-list-item'>  <li><a href="Main.html">Home</a></li>  </div>  <div class='nav-list-item'>  <li><a href="About.html">About</a></li>  </div>  <div class='nav-list-item'>  <li><a href="Registration.html">Sign in</a></li>  </div>  </ul>  </div>  </div>  </div>  </div>  </header> |

Листинг 1 – Header

|  |
| --- |
| <footer>  <div class="bottom">  <div class="container-text">  <h1 class="footer-text">  <a class="main-footer-link" href="Main.html">TRAVELER's TALES</a>  </h1>  </div>  <div class="sections">  <div class="categories">  <div class="topics-container">  <h3 class="first-topic">CATEGORIES</h3>  <h3 class="second-topic">SUBSCRIBE TO NEWSLETTER</h3>  </div>  <ul>  <li><a href="#">Animation</a></li>  <li><a href="#">Lifestyle</a></li>  </ul>  <ul>  <li><a href="#">AI</a></li>  <li><a href="#">Games</a></li>  </ul>  <ul>  <li><a href="#">Graphic Design</a></li>  <li><a href="#">Photography</a></li>  </ul>  <ul style="margin-right: 80px">  <li><a href="#">Crypto</a></li>  <li><a href="#">Cooking</a></li>  </ul>  <div class="send-email">  <input type="email" placeholder="EMAIL ADDRESS" id="email">  <button id="send">SEND</button>  </div>  </div>  <div class="information">  <div class="info-container">  <h3 class="first-info">INFORMATION</h3>  <h3 class="second-info">FOLLOW US</h3>  </div>  <ul>  <li><a href="About.html">about</a></li>  <li><a href="#">contact</a></li>  </ul>  <ul class="second-ul">  <li><a href="https://www.instagram.com/">instagram</a></li>  <li><a href="https://www.facebook.com/?locale=ru\_RU">facebook</a></li>  </ul>  </div>  </div>  </div>  <div class="second-bottom">  <div class="container-text">  <h1 class="footer-text">  <a class="main-footer-link" href="Main.html">TRAVELER's TALES</a>  </h1>  </div>  <div class="send-email2">  <span>SUBSCRIBE TO NEWSLETTER</span>  <input type="email" placeholder="EMAIL ADDRESS" id="email1">  <button id="send1">SEND</button>  </div>  </div>  </footer> |

Листинг 2 – Footer

# Приложение Г

# Листинг SCSS и CSS

|  |
| --- |
| \* {  margin: 0;  padding: 0;  user-select: none;  }  header {  .logo-text {  font-family: $font-family\_1;  font-weight: 600;  font-size: 25px;  height: inherit;  vertical-align: middle;  }  border-bottom-color: $border-bottom-color\_1;  border-bottom-width: 1px;  border-bottom-style: solid;  ul.nav {  float: right;  list-style: none;  li {  display: inline;  margin-left: 59px;  cursor: pointer;  font-weight: 400;  }  }  }  img {  pointer-events: none;  display: inline-block;  vertical-align: middle;  }  .logo-text {  display: inline-block;  vertical-align: middle;  }  .nav {  margin-top: 30px;  li {  &:nth-of-type(3) {  margin-right: 23px;  }  }  } |

# Листинг 3 – Навигационное меню SCSS

|  |
| --- |
| \* {  margin: 0;  padding: 0;  -webkit-user-select: none;  -moz-user-select: none;  user-select: none;  }  header {  border-bottom-color: #E6E6E6;  border-bottom-width: 1px;  border-bottom-style: solid;  }  header .logo-text {  font-family: "Tenor Sans", sans-serif;  font-weight: 600;  font-size: 25px;  height: inherit;  vertical-align: middle;  }  header ul.nav {  float: right;  list-style: none;  }  header ul.nav li {  display: inline;  margin-left: 59px;  cursor: pointer;  font-weight: 400;  }  img {  pointer-events: none;  display: inline-block;  vertical-align: middle;  }  .logo-text {  display: inline-block;  vertical-align: middle;  }  .nav {  margin-top: 30px;  }  .nav li:nth-of-type(3) {  margin-right: 23px;  } |

Листинг 4 – Навигационное меню CSS

|  |
| --- |
| footer {  margin-top: 54px;  height: 500px;  color: $color\_1;  background-color: $background-color\_1;  ul {  li {  font-weight: 400;  line-height: 120%;  }  }  h3 {  font-weight: 400;  line-height: 120%;  }  }  .container-text {  height: 120px;  width: 100%;  border-bottom-style: solid;  border-bottom-color: $border-bottom-color\_1;  border-bottom-width: 1px;  }  .footer-text {  position: absolute;  margin-top: 50px;  font-family: $font-family\_1;  font-size: 20px;  font-weight: 400;  line-height: 120%;  letter-spacing: 1px;  }  .categories {  margin-top: 50px;  .topics-container {  display: flex;  justify-content: space-between;  margin-bottom: 30px;  }  ul {  display: inline-block;  margin-right: 14.5%;  margin-bottom: 50px;  li {  list-style-type: none;  margin-bottom: 30px;  font-family: $font-family\_2;  font-size: 14px;  }  }  }  .topics-container {  .second-topic {  margin-right: 164px;  }  h3 {  color: $color\_2;  font-size: 10px;  font-family: $font-family\_2;  }  }  .send-email {  button {  width: 80px;  height: 54px;  text-align: center;  color: $color\_1;  background-color: $background-color\_2;  margin-left: -5px;  border: none;  outline: none;  }  input {  padding: 0 0 0 10px;  font-family: $font-family\_2;  outline: none;  height: 50px;  width: 220px;  }  }  .information {  .info-container {  display: flex;  margin-bottom: 30px;  }  .first-info {  margin-right: 14.4%;  }  ul {  display: inline-block;  margin-right: 14.4%;  li {  list-style-type: none;  margin-bottom: 30px;  font-family: $font-family\_2;  font-size: 14px;  }  }  .second-ul {  margin-left: 18px;  }  }  .info-container {  h3 {  color: $color\_2;  font-size: 10px;  font-family: $font-family\_2;  }  } |

Листинг 5 – Footer SCSS

|  |
| --- |
| footer {  margin-top: 54px;  height: 500px;  color: white;  background-color: #141414;  }  footer ul li {  font-weight: 400;  line-height: 120%;  }  footer h3 {  font-weight: 400;  line-height: 120%;  }  .container-text {  height: 120px;  width: 100%;  border-bottom-style: solid;  border-bottom-color: #4D4D4D;  border-bottom-width: 1px;  }  .footer-text {  position: absolute;  margin-top: 50px;  font-family: "Tenor Sans", sans-serif;  font-size: 20px;  font-weight: 400;  line-height: 120%;  letter-spacing: 1px;  }  .categories {  margin-top: 50px;  }  .categories .topics-container {  display: flex;  justify-content: space-between;  margin-bottom: 30px;  }  .categories ul {  display: inline-block;  margin-right: 14.5%;  margin-bottom: 50px;  }  .categories ul li {  list-style-type: none;  margin-bottom: 30px;  font-family: "Libre Franklin", sans-serif;  font-size: 14px;  }  .topics-container .second-topic {  margin-right: 164px;  }  .topics-container h3 {  color: #666666;  font-size: 10px;  font-family: "Libre Franklin", sans-serif;  }  .send-email button {  width: 80px;  height: 54px;  text-align: center;  color: white;  background-color: black;  margin-left: -5px;  border: none;  outline: none;  }  .send-email input {  padding: 0 0 0 10px;  font-family: "Libre Franklin", sans-serif;  outline: none;  height: 50px;  width: 220px;  }  .information .info-container {  display: flex;  margin-bottom: 30px;  }  .information .first-info {  margin-right: 14.4%;  }  .information ul {  display: inline-block;  margin-right: 14.4%;  }  .information ul li {  list-style-type: none;  margin-bottom: 30px;  font-family: "Libre Franklin", sans-serif;  font-size: 14px;  }  .information .second-ul {  margin-left: 18px;  }  .info-container h3 {  color: #666666;  font-size: 10px;  font-family: "Libre Franklin", sans-serif;  } |

Листинг 6 – Footer CSS

# Приложение Д

# Листинг XML-файлов

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" ?>  <cards>  <card>  <id>1</id>  <card-image>../Images/image1.svg</card-image>  <card-type>ILLUSTRATION</card-type>  <card-title>JAPAN HOUSE OPENS IN MOUNTAINSIDE TO <br/>FOSTER PEAK CREATIVITY.</card-title>  <card-author>BY RETA TORPHY</card-author>  </card>  <card>  <id>2</id>  <card-image>../Images/image2.svg</card-image>  <card-type>PHOTOGRAPHY</card-type>  <card-title>HELMUT LANG <br/>CELEBRATES TAXI <br/>DRIVERS WORLDWIDE IN LATEST CAMPAIGN</card-title>  <card-author>BY ALESSANDRA ORTIZ</card-author>  </card>  <card>  <id>3</id>  <card-image>../Images/image3.svg</card-image>  <card-type>PHOTOGRAPHY</card-type>  <card-title>BOWL CUT LAUNCH A NEW SUMMER COLLECTION THAT PAYS HOMAGE TO <br/>“UK LEGENDS”</card-title>  <card-author>BY ROSANNA ONDRICKA</card-author>  </card>  <card>  <id>4</id>  <card-image>../Images/image4.svg</card-image>  <card-type>PHOTOGRAPHY</card-type>  <card-title>THOUSANDS OF UNSEEN PHOTOGRAPHS WILL BE MADE PUBLIC THIS AUTUMN</card-title>  <card-author>BY ANNIE LUEILWITZ</card-author>  </card>  <card>  <id>5</id>  <card-image>../Images/image5.svg</card-image>  <card-type>INTERACTIVE DESIGN</card-type>  <card-title>LONDON-BASED YINKA <br/>ILORI’S STORYTELLING <br/>FURNITURE</card-title>  <card-author>BY ANNIE LUEILWITZ</card-author>  </card>  <card>  <id>6</id>  <card-image>../Images/image6.svg</card-image>  <card-type>GRAPHIC DESIGN</card-type>  <card-title>ANONYMOUS ISRAELI <br/>ART COLLECTIVE BROKEN <br/>FINGAZ DIRECT MUSIC <br/>VIDEO FOR U2 AND BECK</card-title>  <card-author>BY RETA TORPHY</card-author>  </card>  <card>  <id>7</id>  <card-image>../Images/image7.svg</card-image>  <card-type>ARCHITECTURE</card-type>  <card-title>METICULOUSLY <br/>ARRANGED <br/>PHOTOGRAPHS ALTER <br/>PERCEPTIONS</card-title>  <card-author>BY SIMEON BREKKE</card-author>  </card>  <card>  <id>8</id>  <card-image>../Images/image8.svg</card-image>  <card-type>GRAPHIC DESIGN</card-type>  <card-title>ANU AMBASNA’S PLAYFUL <br/>ILLUSTRATIONS <br/>CELEBRATE CLUB <br/>CULTURE</card-title>  <card-author>BY LEO BARTELL</card-author>  </card>  </cards> |

Листинг 7 – Хранение данных в формате XML

# Приложение Е

# Листинг JavaScript

|  |
| --- |
| fetch("/Data/cards.xml")  .then(response => response.text())  .then(xmlString => {  let parser = new DOMParser();  let xmlDoc = parser.parseFromString(xmlString, "text/xml");  let cards = xmlDoc.getElementsByTagName("card");  let firstCardsContainer = document.querySelector(".first-cards-container");  let secondCardsContainer = document.querySelector(".second-cards-container");  for (let i = 0; i < cards.length; i++) {  let card = cards[i];  let id = card.getElementsByTagName("id")[0].textContent;  let image = card.getElementsByTagName("card-image")[0].textContent;  let type = card.getElementsByTagName("card-type")[0].textContent;  let title = card.getElementsByTagName("card-title")[0].textContent;  let author = card.getElementsByTagName("card-author")[0].textContent;  let cardContainer = document.createElement("div");  let imgElement = document.createElement("img");  let typeElement = document.createElement("h5");  let titleElement = document.createElement("h2");  let authorElement = document.createElement("strong");  imgElement.src = image;  imgElement.alt = "picture-" + id;  typeElement.textContent = type;  titleElement.innerHTML = title;  authorElement.textContent = author;  cardContainer.appendChild(imgElement);  cardContainer.appendChild(typeElement);  cardContainer.appendChild(titleElement);  cardContainer.appendChild(authorElement);  if (i < 4) {  firstCardsContainer.appendChild(cardContainer);  } else {  cardContainer.style.marginRight = "0";  secondCardsContainer.appendChild(cardContainer);  }  }  })  .catch(error => {  console.log("Ошибка при загрузке XML файла:", error);  }); |

Листинг 9 - Получение данных из XML

|  |
| --- |
| document.getElementById("registration").addEventListener("click", function() {  let email = document.getElementById("mail").value;  let name = document.getElementById("name").value;  let password = document.getElementById("password").value;  let userData = {  email: email,  name: name,  password: password  };  let jsonData = JSON.stringify(userData, null, 2);  let jsonBlob = new Blob([jsonData], { type: "application/json" });  let jsonUrl = URL.createObjectURL(jsonBlob);  let xmlData = convertToXml(userData);  let xmlBlob = new Blob([xmlData], { type: "application/xml" });  let xmlUrl = URL.createObjectURL(xmlBlob);  let jsonLink = document.createElement("a");  jsonLink.href = jsonUrl;  let jsonTimestamp = new Date().getTime();  jsonLink.download = "data\_registration\_" + jsonTimestamp + ".json";  let xmlLink = document.createElement("a");  xmlLink.href = xmlUrl;  let xmlTimestamp = new Date().getTime();  xmlLink.download = "data\_registration\_" + xmlTimestamp + ".xml";  jsonLink.click();  xmlLink.click();  URL.revokeObjectURL(jsonUrl);  URL.revokeObjectURL(xmlUrl);  });  function convertToXml(obj) {  let xml = '';  for (let prop in obj) {  if (obj.hasOwnProperty(prop)) {  xml += "<" + prop + ">" + obj[prop] + "</" + prop + ">";  }  }  return "<root>" + xml + "</root>";  } |

Листинг 10 – Загрузка данных из формы регистрации