**Введение**

Разрабатываемый сайт представляет собой новостной портал средств массовой информации.

В качестве цветовой палитры был выбран преобладающе белый цвет, дабы отразить некую организованность и строгость делового издания.

Новостные разделы представлены блоками, в каждом из которых присутствует картинка, заголовок новости и описание.

Все эти данные будут храниться в базе данных в виде таблиц. Картинки будут располагаться непосредственно на сервере, а в таблицах будет указан путь к файлам.

**Разработка информационной архитектуры**

Для распределения файлов по категориям в корневом каталоге сайта были созданы несколько папок. А именно:

* папка css для файлов стилей с расширением css;
* папка img для хранения изображений, необходимых для новостных статей. Все изображения представлены в формате jpg для удобства использования. Изображение логотипа хранится на стороннем ресурсе;
* папка js для хранения файлов с расширением js, в числе которых скриптовые сценарии выполнения, подключаемые к страницам, а также библиотеки JQuery, используемая по причине удобства использования, предоставляемых ею функций, в сравнении с написанием на «чистом» JavaScript.

Иерархия и архитектура файлов и папок представлена на рисунке 1.

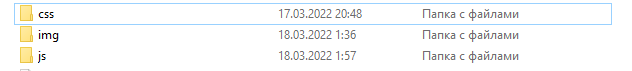


Рисунок 1 – Информационная архитектура сайта

**Разработка дизайна сайта**

Дизайн главной страницы сайта представлен на рисунке 2.

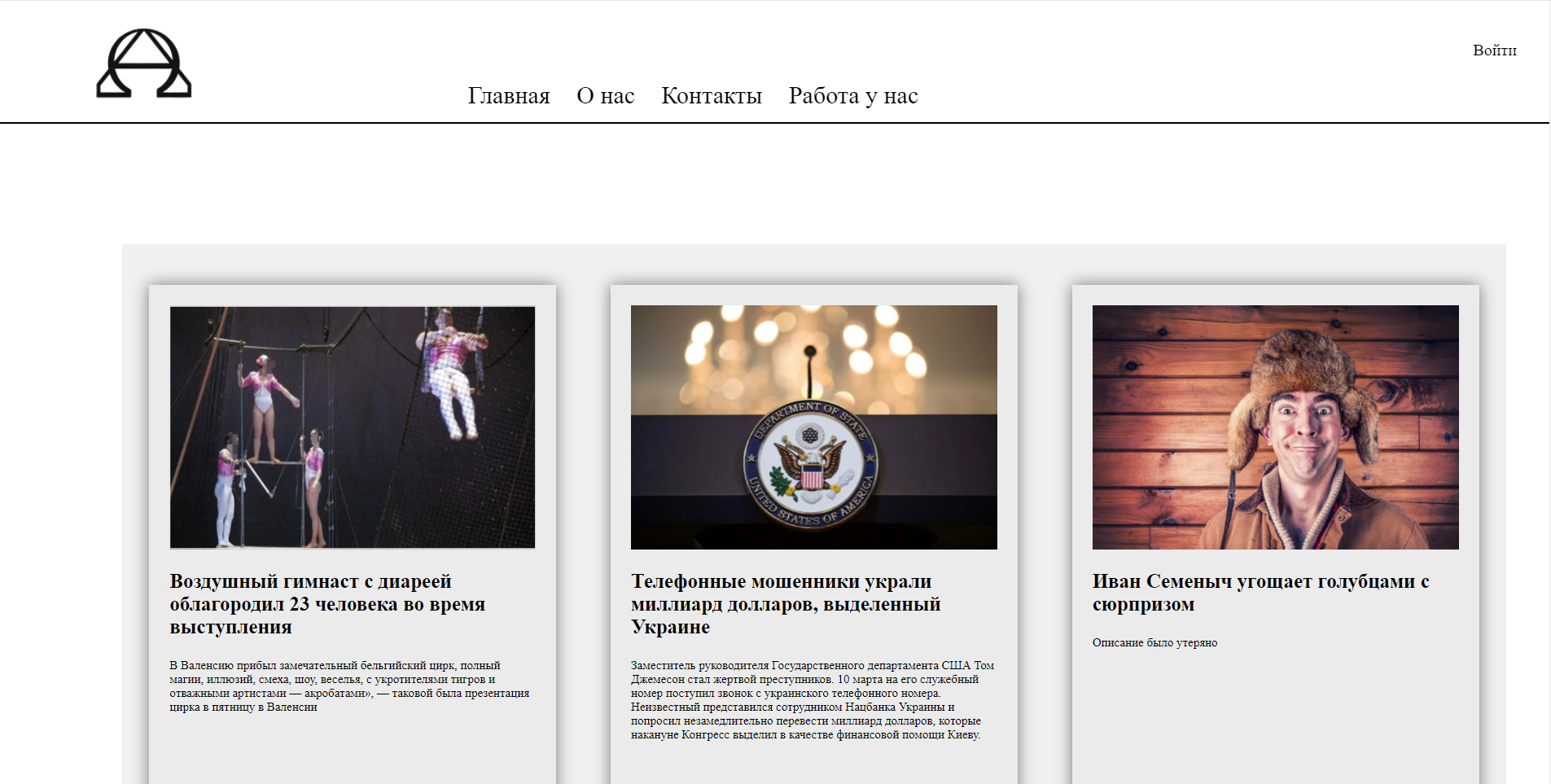


Рисунок 2 – Дизайн главной страницы сайта

Согласно представленному рисунку, следует отметить, что на сайте будет присутствовать так называемая «шапка сайта», на которой будет располагаться меню навигации, а также ссылка на окно авторизации и регистрации.

**Подготовка иллюстраций для WEB**

Для использования в новостных статьях были отобраны изображения из сети Интернет, в том числе с сайта издания «Панорама», откуда в качестве демонстрации работоспособности сайта были позаимствованы не только изображения, но и некоторые заголовки новостей, а также их содержание.

**Подготовка графических элементов: логотип, кнопки и фоновые картинки**

В качестве логотипа сайта было использовано изображение, совмещающее в себе греческие знаки альфа и омега черного цвета.

Оно было использовано в качестве ссылки на главную страницу сайта, а также для придания дизайну сайта фирменного стиля. Логотип «Альфа-Омега» представлен на рисунке 3.

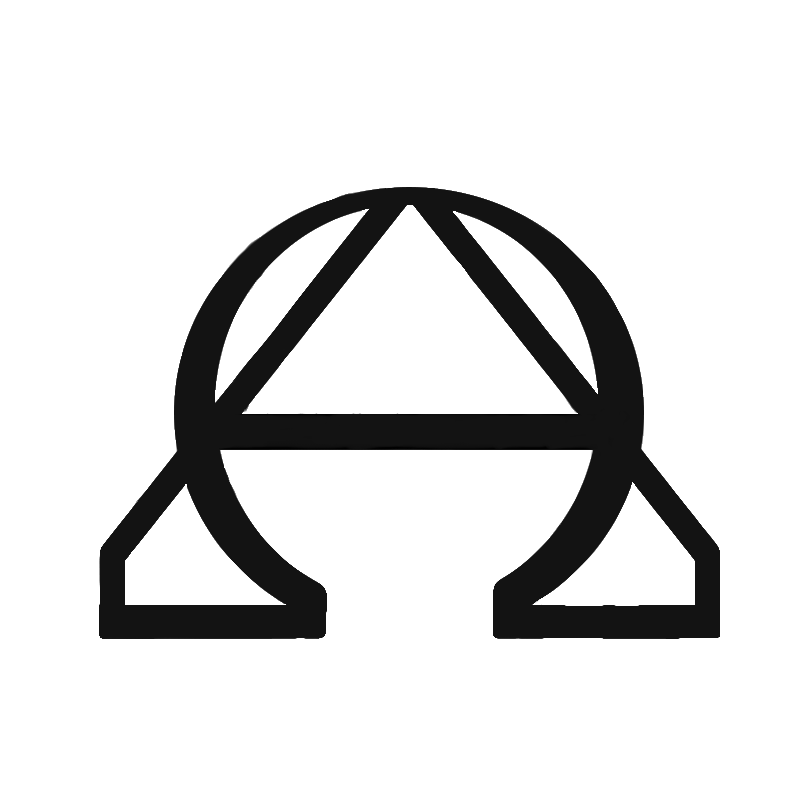


Рисунок 3 – Логотип «Альфа-Омега»

**Создание стилевого оформления с помощью каскадных таблиц стилей**

Абсолютно все редактирования объектов и элементов прописаны в файле style.css, который находится в каталоге css.

Первым делом, было необходимо убрать стандартные ограничения html файла, из-за которых элементы могли иметь ненужные отступы и границы. Для этого обратились к тегу body и изменили его свойства с помощью padding и margin, установив их значения равными нулю. Код представлен на рисунке 4.

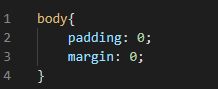


Рисунок 4 – Удаление отступов у body

После этого обратились к разделу header, который представляет собой шапку сайта. Установили для нее position: fixed, что позволяет ей оставаться перед пользователем на том же месте при прокрутке страницы вниз.

Задали значения ширины на 100% для того, чтобы она занимала все пространство и значение высоты на 150 пикселей.

Для дополнительной стилизации воспользовались свойством border-bottom, задав ему значение в два пикселя и черный цвет снизу элемента.

Чтобы шапка сайта не была прозрачным блоком при прокрутке выбрали белый цвет для ее фона.

Для того чтобы она не скрывалась за другими элементами и отображалась выше их всех, воспользовались свойством z-index, со значением 1, а также для создания последующей анимации установили для свойства transition время перехода в одну секунду.

Помимо этого, установили значения прозрачности и фоновый цвет серый для шапки сайта с классом bot, который понадобится в будущем для создания анимации.

Код представлен на рисунке 5. Итоговый результат представлен на рисунке 6.

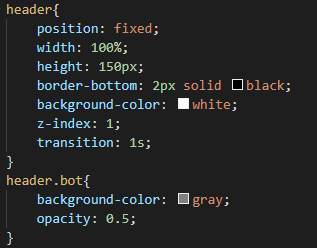


Рисунок 5 – Манипуляции над header

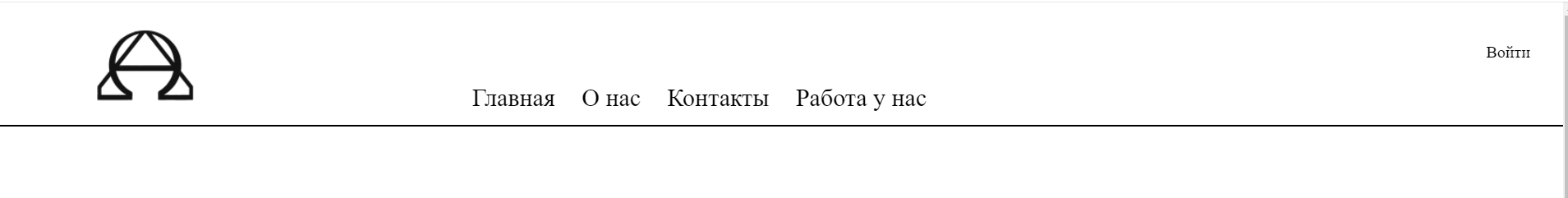


Рисунок 6 – Результат манипуляций

Для расположения элементов был создан центральный контейнер div с классом main, после чего к нему было применено несколько свойств:

* display: flex для ровного расположения контейнеров на главном блоке;
* z-index: 0 для того чтобы тело страницы не перекрывало шапку;
* position: absolute для контроля расположения блока относительно его изначальной позиции;
* background-color: rgb(240, 240, 240);
* width: 1700px;
* margin: 0 auto для корректного центрирования элемента;
* top: 300px для отступа в 300 пикселей от своей изначальной позиции;
* left: 150px аналогично для левого края;
* justify-content: space-around – специальная манипуляция над элементами flexbox, позволяющая задать для каждого из них равный отступ по краям, что в свою очередь располагает их максимально ровно.

Код представлен на рисунке 7.

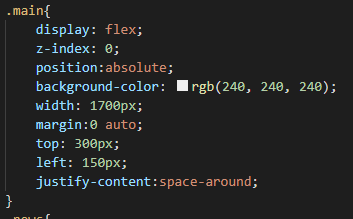


Рисунок 7 – Манипуляции над контейнером main

Каждый блок с новостью имеет класс с наименованием news. Обратившись к классу с помощью css-файла, задали для каждого из них следующие параметры:

* margin-top:50px;
* width: 500px;
* height: 700px;
* background-color: rgb(235, 235, 235);
* box-shadow: 0 0 20px rgba(0,0,0, 0.5).

В каждом из новостных блоков для позиционирования были проведены следующие манипуляции.

Над изображениями:

* width:450px;
* height:300px;
* margin-left:25px;
* margin-top: 25px.

Над заголовками новостных статей:

* white-space: pre-wrap для того чтобы текст переносился в случае если ему не хватает места, а не выходил за границы отведенного ему блока;
* font-size: 25px для изменения размера шрифта;
* font-weight: bold для того чтобы сделать шрифт жирным;
* margin-left: 25px;
* margin-right:25px.

Над описанием новостей:

* margin-left: 25px;
* margin-right: 25px;
* white-space: pre-wrap;
* font-size: 15px.

Код программы представлен на рисунках 8 и 9.

Результат работы над блоками представлен на рисунке 10.

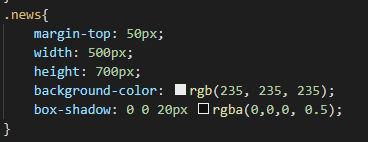


Рисунок 8 – Код манипуляций на классом news

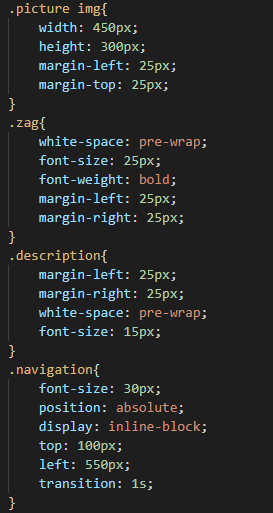


Рисунок 9 – Код манипуляций над классами zag, description, navigation и изображениями контейнера класса picture

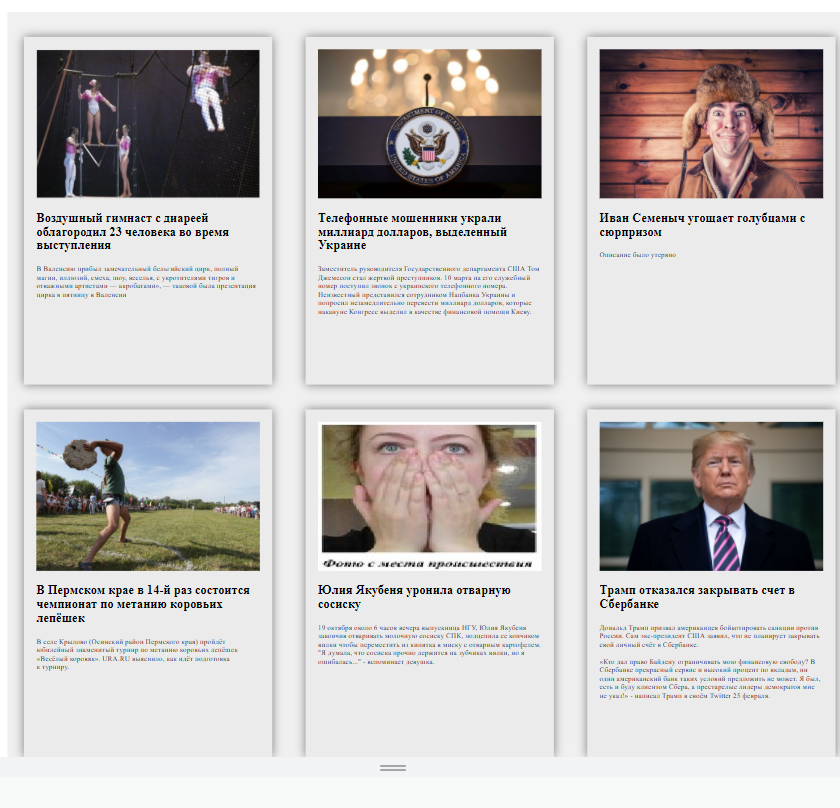


Рисунок 10 – Результат манипуляций над контейнерами

Помимо этого, было создано меню навигации, для перехода по страницам сайта, а также простая форма авторизации и регистрации.

Измененные свойства других, менее важных элементов представлены на рисунках 11 и 12.

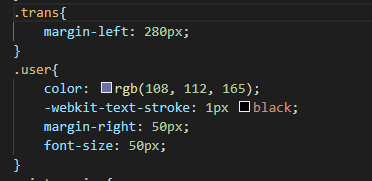


Рисунок 11 – Стилизация элементов

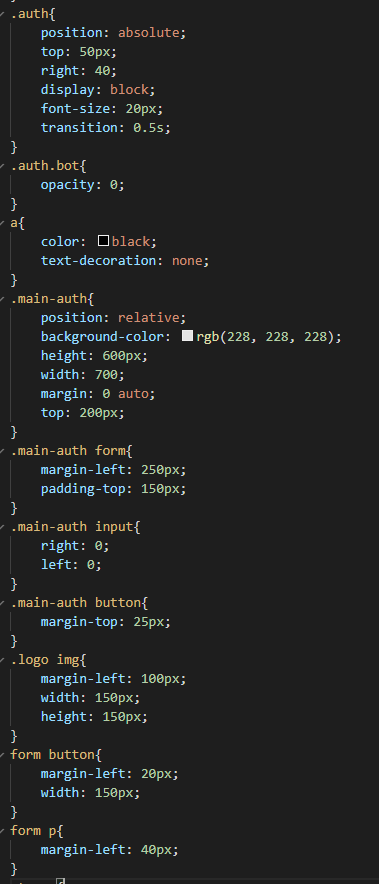


Рисунок 12 – Стилизация элементов

**Язык сценариев JavaScript**

Для повышения уровня привлекательности сайта и его наполнения различными элементами, с помощью библиотеки JQuery была создана анимация. Ее код представлен на рисунке 13.

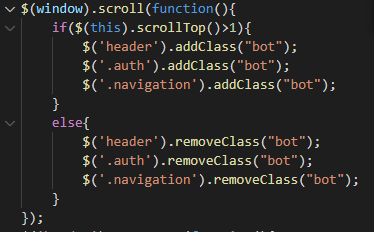


Рисунок 13 – Код анимации, реализованной с помощью библиотеки JQuery

Суть данного кода заключается в следующем. При прокрутке страницы пользователем вниз срабатывает обработчик события, которые задействует функцию. Первым в этой функции является условие на проверку, находится ли в данный момент пользователь в верху страницы или нет за счет проверки высоты расположения скролла с помощью scrollTop()>1.

Если это условие верно, то классы header, auth, navigation получают дополнительный класс bot. Именно для этого в css файле и было необходимо наличие атрибута transition, для плавного перехода между состояниями. В случае если условие не верно, то класс bot напротив исчезает. Все это создано для того, чтобы шапка сайта становилась прозрачной, а некоторые элементы с нее и вовсе исчезали, чтобы не мешать просмотру контента.

Также, в случае если пользователь захочет воспользоваться шапкой сайта в каком-либо из разделов сайта, он может просто навестись на шапку, и она станет более явной. За это отвечает регистратор события mouseover. Код представлен на рисунке 14.

Демонстрация анимации представлена на рисунках 15,16 и 17.

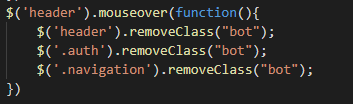


Рисунок 14 – Код обработки наведения мыши на шапку сайта

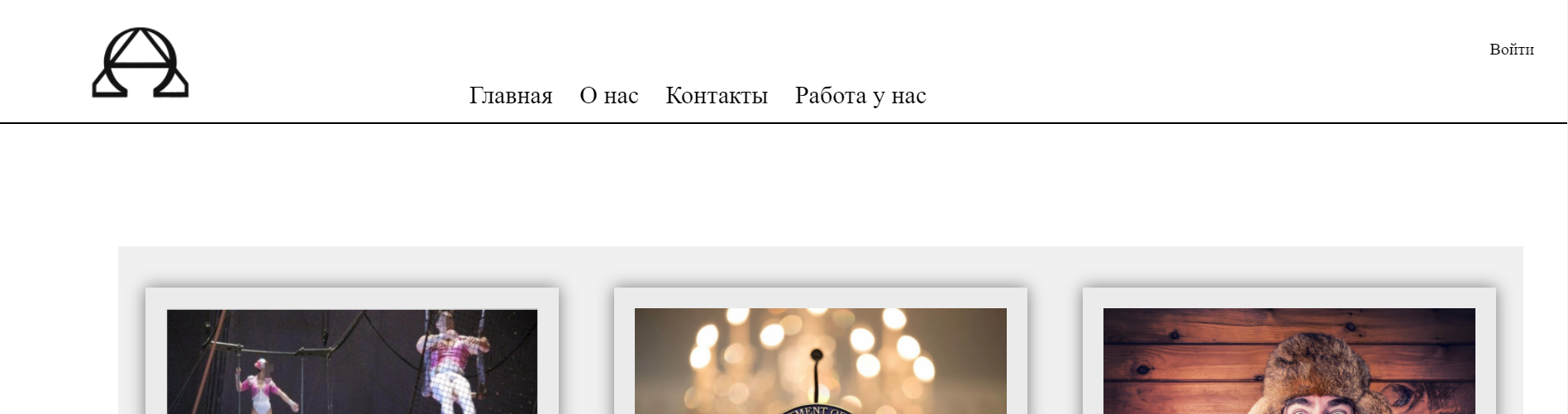


Рисунок 15 – Обычное состояние шапки сайта

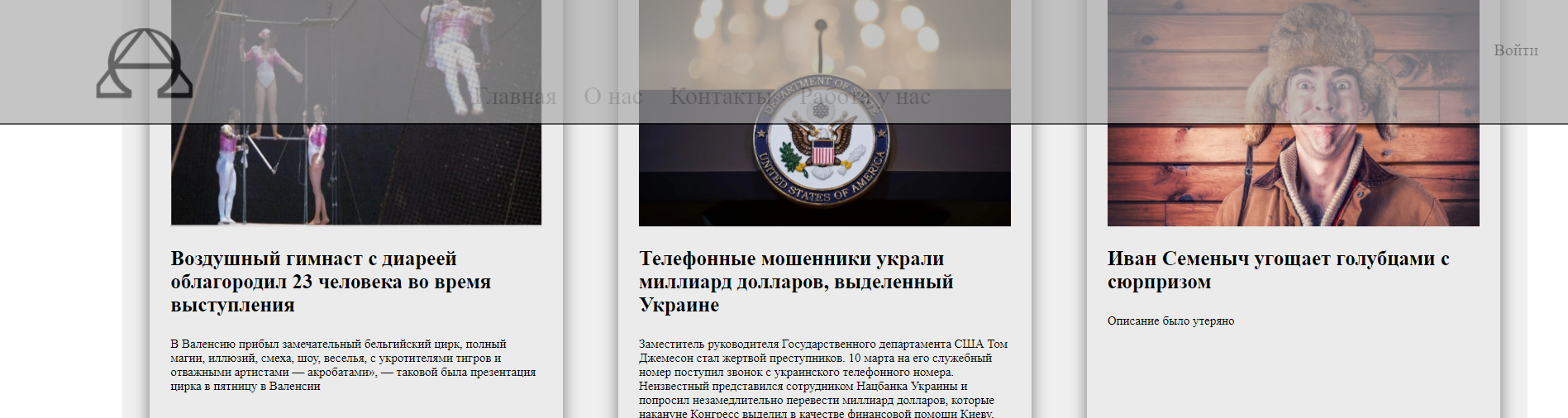


Рисунок 16 – Промежуточное состояние шапки сайта

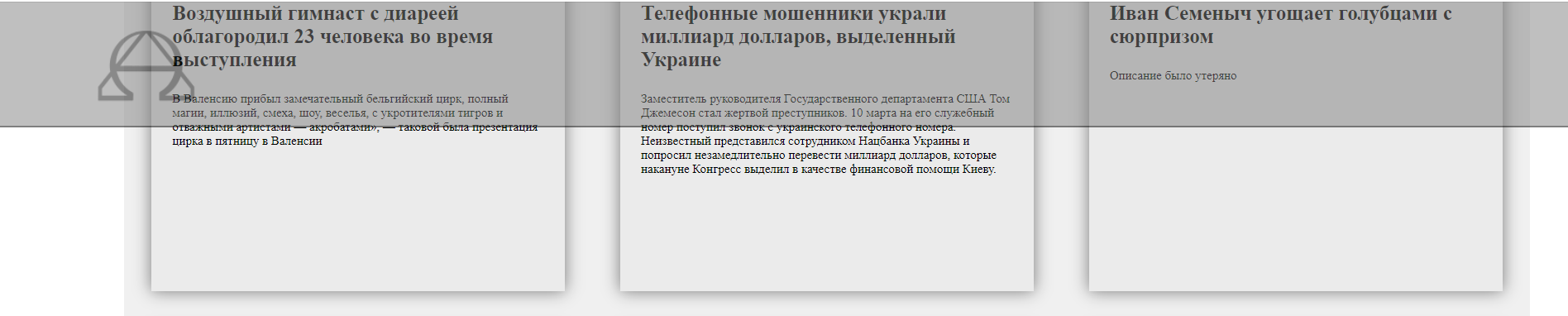


Рисунок 17 – Конечное состояние шапки сайта

**Разработка форм взаимодействия пользователя с веб-приложением**

Для отправки данных на сервер было разработано несколько форм. Форма авторизации и регистрации. Форма авторизации представлена на рисунке 18. Код формы представлен на рисунке 19.

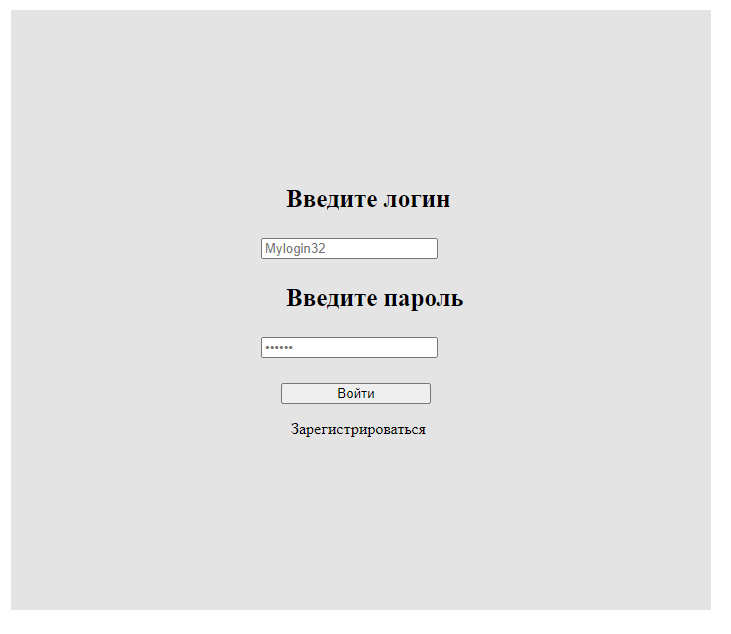


Рисунок 18 – Форма авторизации

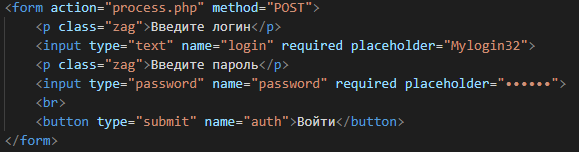


Рисунок 19 – Код формы авторизации