Лабораторная работа №8  
Создание Формы регистрации

Одна из проблем с формами в предыдущих спецификациях HTML состоит в том, что поля формы — это всего лишь пустые ячейки для заполнения, не более того. На сервере, конечно же, требуется валидация данных, но для создания осмысленного интерфейса приходится дублировать эту проверку в браузере с помощью JavaScript. Учитывая, что формы используются практически на каждой веб-странице — формы поиска, регистрации, добавления комментария и т. д., — было бы здорово, если бы в браузеры была включена валидация тех типов данных, которые чаще всего вводятся пользователем и в случае неверного ввода появлялись сообщения об ошибках.

Как вы уже догадались, формы HTML5 позволяют делать именно это. Появились новые значения атрибута type элемента input.

ПРИМЕР ГОТОВОЙ ФОРМЫ: [ЛР Формы\forms.html](file:///\\Server301\длястудентов301\Предметы\ИТТ\ЛР%20Формы%20HTML%205\ЛР%20Формы\forms.html)

**1 Теория**

**Тип ввода данных email.**

Строка <input type=email> сообщает браузеру, что форма не должна быть отправлена, пока пользователь не ввел нечто, похожее на правильный адрес электронной почты, то есть браузер не проверяет, существует ли такой адрес, а только определяет правильность его формата. Пользователь может отправить форму с незаполненным полем, только если к этому полю не добавлен атрибут required (это верно для всех типов ввода данных). Примеры заполнения представлены на рисунках 10, 11.

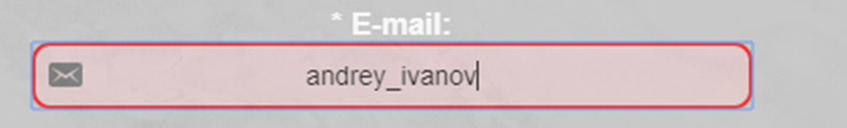


Рисунок 1 – Недопустимый e-mail

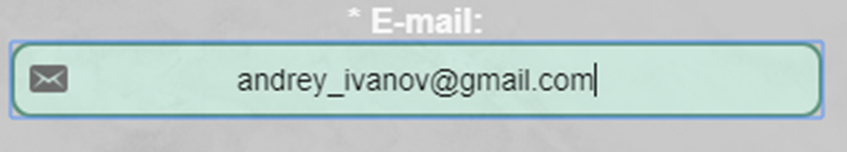


Рисунок 2 – Допустимый e-mail

**Тип ввода данных password.**

Строка <input type=password> заставляет браузер скрывать значения, вводимые пользователем – рисунок 12:

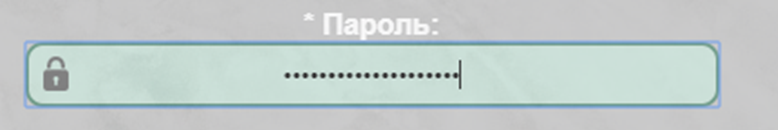


Рисунок 3 – Пример вводы пароля

**Тип ввода данных url.**

Строка <input type=url> заставляет браузер проверять, что пользователь ввел правильный URL-адрес (в формате http:// или https://) – рисунок 13:

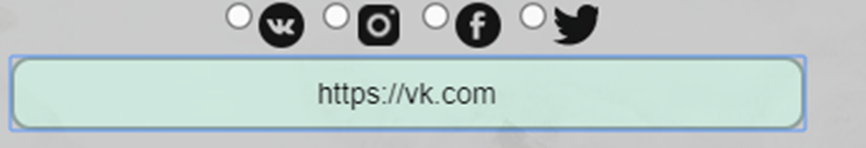


Рисунок 4 – Пример ввода допустимого url-адреса

**Тип ввода данных datetime-local.**

<input type=”datetime-local” создает поля ввода, позволяющие легко ввести дату и время, открывая виджеты календаря или другие; они ограничивают введенное время: часы – не больше 24, минуты – не больше 59 – рисунок 14:

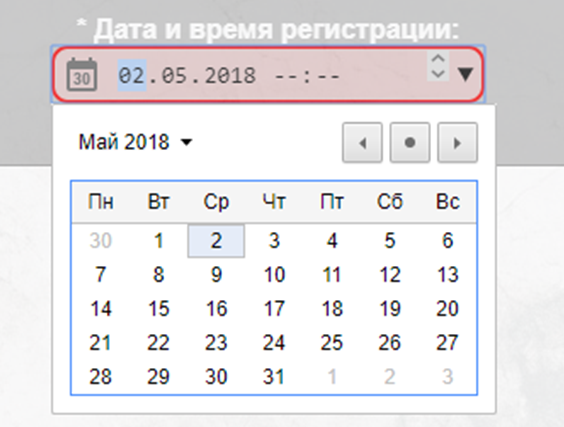


Рисунок 5 – Пример использования виджета «Календарь» в браузере «Opera»

**Тип ввода данных number.**

Как это ни удивительно, но тип ввода данных number выдает ошибку, если пользователь не ввел численные символы. Он не предназначен для ввода номеров телефона, так как их часто указывают с пробелами, скобками, значком плюса, дефисами и т. п.; используйте для телефонов <input type=tel>.

Он прекрасно работает с атрибутами min, max и step. В Opera и Chrome он выглядит как счетчик, значения которого можно ограничить сверху и снизу, а шаг задается с помощью атрибута step; при этом пользователь может ввести число с клавиатуры. Ввод буквенных символов в поля type=number в Opera и Chrome не приводит к ошибке валидации. Подобное случается из-за того, что браузер не занимается проверкой введенных данных — он даже не заменяет текущее значение поля введенными пользователем нецифровыми символами. К сожалению, пользовательский интерфейс предполагает, что вы ввели буквы и они были одобрены. Поэтому для ограничения ввода данных в некоторых браузерах необходимо использовать атрибут pattern, о котором будет говориться далее и действие которого показано на рисунках 15, 16.

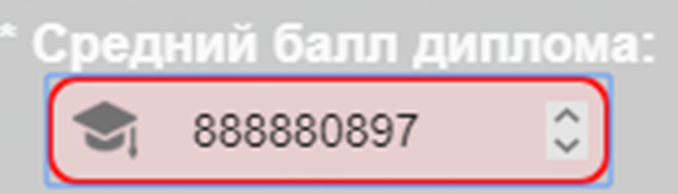


Рисунок 6 – Недопустимый ввод (ограничение по значению)

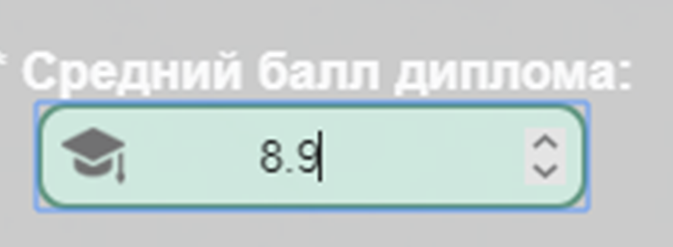


Рисунок 7 – Допустимый ввод

**Тип ввода данных tel.**

Тип tel используется для ввода телефонного номера. Особой проверки не производится; более того, не требуется вводить только цифры — многие номера часто пишутся с использованием других знаков, например +44 (0) 123 1234.

**Тип ввода данных color.**

<input type=color> позволяет пользователю ввести значение цвета, выбрав нужный вариант в палитре – рисунок 17:

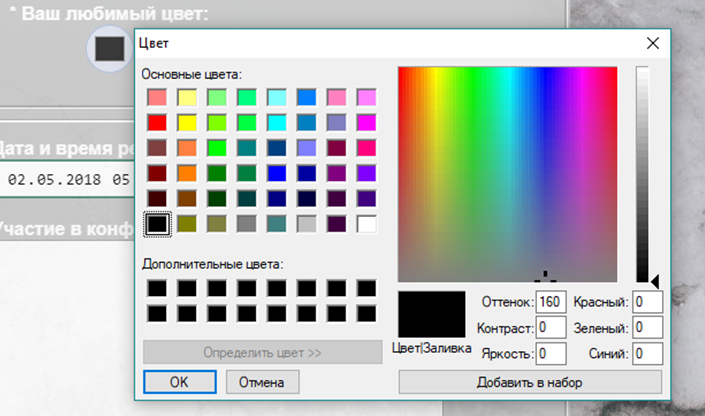


Рисунок 8 – Открытие цветовой палитры для типа поля «color»

**Тип ввода данных checkbox.**

Флажки используют, когда необходимо выбрать любое количество вариантов из предложенного списка (рисунок 18). Если требуется выбор лишь одного варианта, то для этого следует предпочесть переключатели (radio). Флажок создается следующим образом:

<input type="checkbox">.

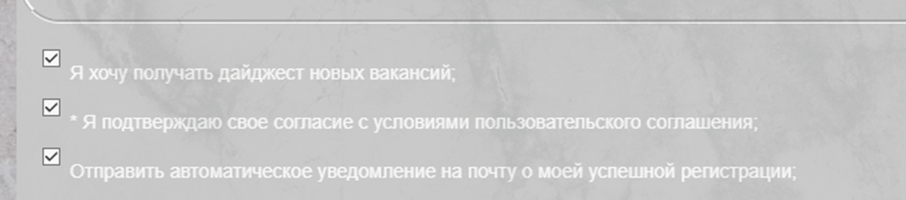


Рисунок 9 – Пример использования поля «флажок»

**Новые атрибуты.**

Помимо новых типов ввода данных, в HTML5 определен ряд новых атрибутов для элемента <input>, отвечающих за его поведение и ограничения: autocomplete, min, max, multiple, pattern и step.

**Атрибут autofocus.** Булев атрибут autofocus — способ задания установки фокуса на определенном элементе управления формы во время загрузки страницы. Раньше разработчику требовалось писать для этого JavaScript-код. Новый способ позволяет делать достаточно разумные вещи, например не устанавливать фокус на элементе, если пользователь в это время вводит данные в другом месте (проблема, типичная для старых JavaScript-скриптов, реализующих эту возможность).

На странице должно быть только одно такое поле ввода. Помните об удобстве использования и применяйте этот атрибут с осторожностью – рисунок 19:

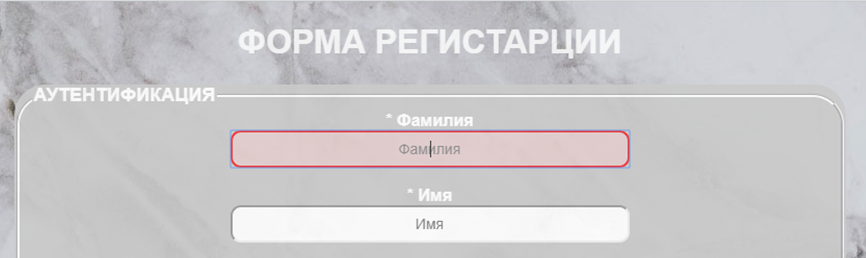


Рисунок 10 – Применение атрибута «autofocus»

**Атрибут placeholder.** Достаточно популярным у разработчиков является такой прием: изначально в поле ввода добавляется текст-подсказка, которая исчезает при установке фокуса на этом поле, а затем (при установке фокуса на другом элементе) снова появляется. Раньше для этого требовался JavaScript. Но теперь такое поведение можно задать с помощью атрибута placeholder – рисунки 20,21.

В спецификации говорится: «Для добавления более подробной подсказки или совета лучше использовать атрибут title».

Важно также помнить, что атрибут placeholder не заменяет элемент <label>.

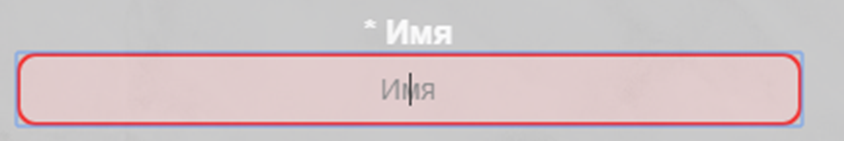


Рисунок 11 – Атрибут «placeholder» при фокусе курсора в поле ввода

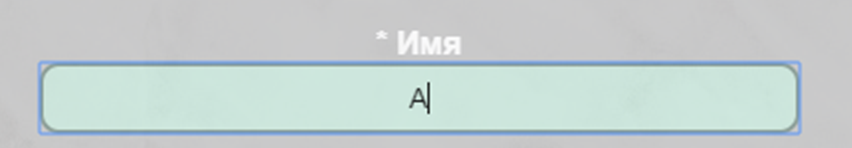


Рисунок 12 – Атрибут «placeholder» при вводе данных в поле

**Атрибут required.** Новый атрибут required можно использовать для <textarea> и большинства полей ввода (кроме случаев, когда атрибут type принимает значения hidden, image или задает кнопочный тип, такой как submit). Браузер не разрешит пользователю отправить форму, если обязательные поля будут пустыми, и вернет ошибку – рисунок 22:

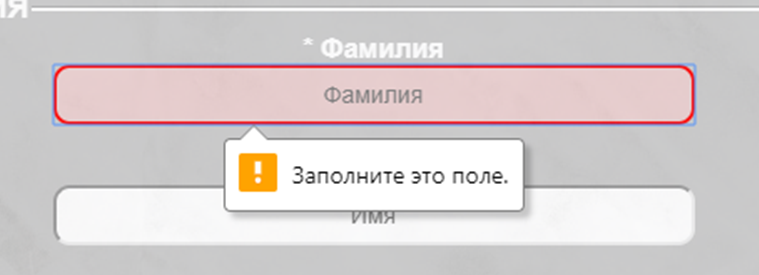


Рисунок 13 – Сообщение об ошибке в браузере «Opera»

**Атрибут pattern.** Некоторые типы ввода данных, о которых говорилось выше, — email, number, url и т. д. — на самом деле являются «встроенными» регулярными выражениями, так как браузер просто проверяет, что введенные данные выглядят так, как должны.

А что, если вы хотите использовать другой шаблон? Атрибут pattern позволяет вам самим создавать регулярные выражения, которым должны соответствовать вводимые данные. Так что если пользователь должен ввести одну цифру и три заглавных символа алфавита, регулярное выражение будет содержать одно число [0–9] и три буквы [A–Z]{3}, каждая из которых должна быть в верхнем регистре; тогда для элемента ввода будет использоваться такой код: <input pattern="[0-9][A-Z]{3}">.

На рисунках 23, 24, 25, 26 использовано значение атрибута pattern, при котором допустимо вводить только цифры и их должно быть 9 – <input type="tel" pattern="[0-9]{9}">:

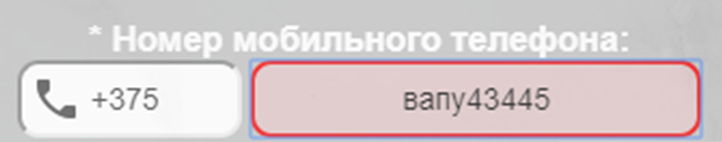


Рисунок 14 – Недопустимый ввод (буквы)

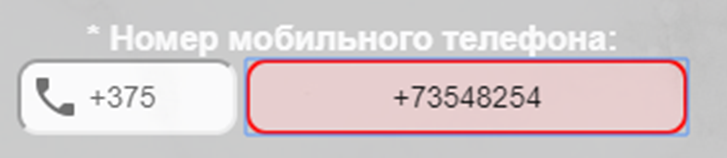


Рисунок 15 – Недопустимый ввод (знак «+»)

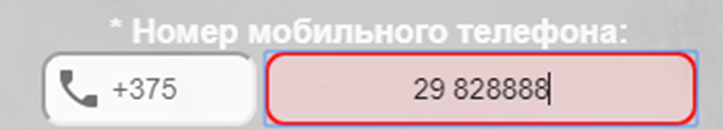


Рисунок 16 – Недопустимый ввод (пробел)

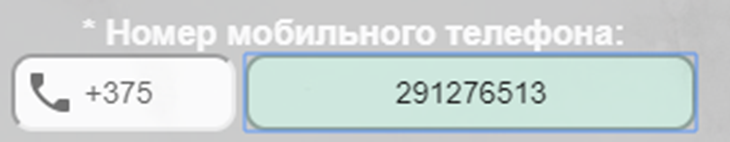


Рисунок 17 – Допустимый ввод (только цифры)

Видно, что мы не можем ввести в данное поле буквы, пробелы и любые знаки, кроме цифр. На рисунке 27 представлены примеры других регулярных выражений. Их можно комбинировать и получать необходимые ограничения.

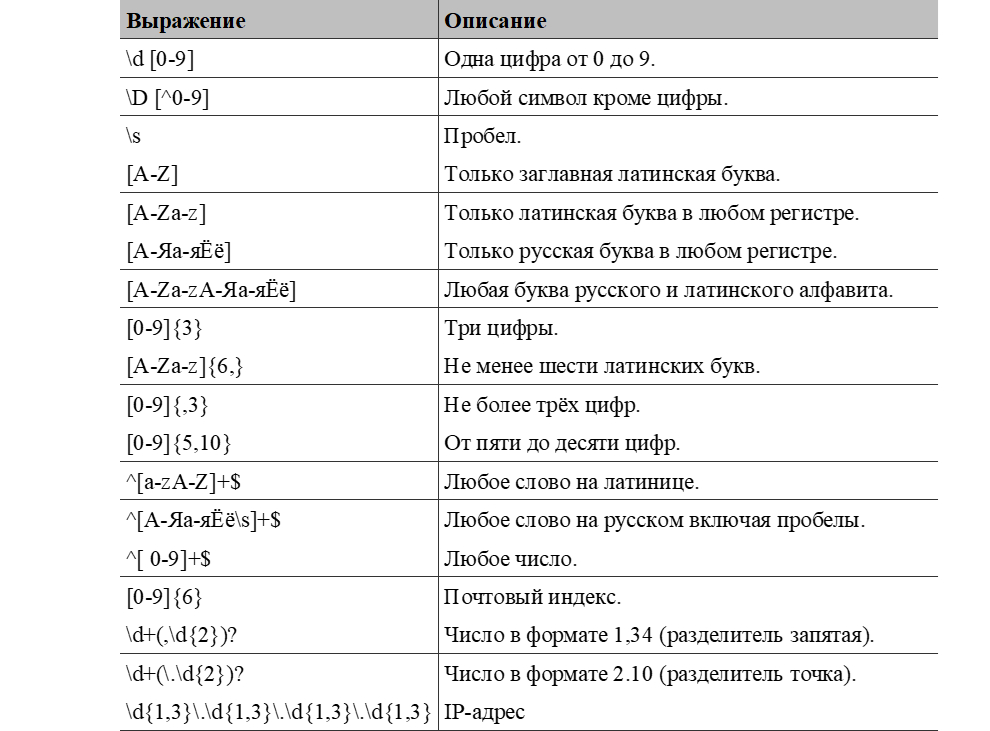


Рисунок 18 – Примеры регулярных выражений для атрибута «pattern»

Атрибуты min и max. Форма не будет отправлена, если значение меньше min или больше max. Их можно использовать с различными типами ввода данных — например, поле <input type=date min=2010-01-01 max=2010-12-31> примет только дату, относящуюся к 2010 году. Часто бывает полезно делать так, чтобы при создании HTML-кода сервер присваивал атрибуту min дату сегодняшнего дня, и тогда вводить можно было бы только последующие даты.

На рисунках 28, 29, 30, 31 видно, что, если введенное значение меньше 4 – поле не является верным (так как значение min=”4”) и нельзя ввести балл больше 10 (так как значение max=”10”).

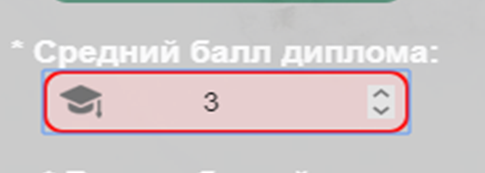


Рисунок 19 – Недопустимый ввод (значение меньше минимального)

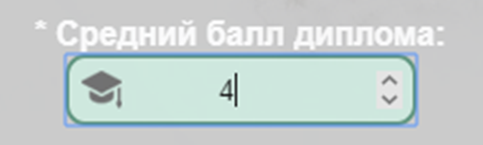


Рисунок 20 – Допустимый ввод (минимальное значение)

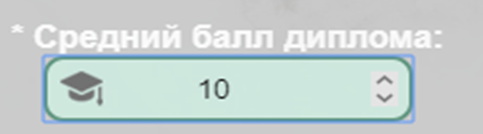


Рисунок 21 – Допустимый ввод (максимальное значение)

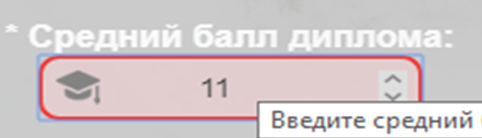


Рисунок 22 – Недопустимый ввод (значение больше максимального)

**Атрибут step.** Атрибут step отвечает за степень подробности вводимых данных. Если вы хотите, чтобы пользователь ввел число в процентах (между 0 и 100) с точностью до 5, можно использовать такой код: <input type=number mix=0 max=100 step=5>. Тогда счетчик будет увеличиваться с шагом 5.

На рисунках 32, 33 step=”0,1”, поэтому, нажимая на стрелочки к исходному значению будет добавляться одна десятая и значение с несколькими знаками

после запятой не будет являться верным.

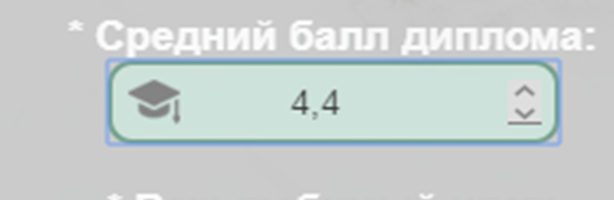


Рисунок 23 – Допустимое значение (один знак после запятой)

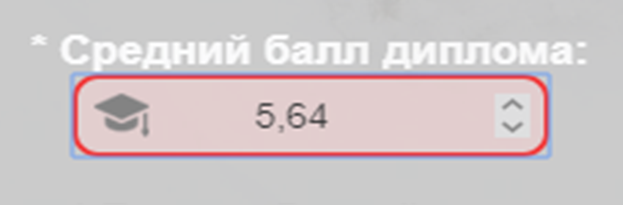


Рисунок 24 – Недопустимое значение (два знака после запятой)

**Атрибут title.** Данный атрибут позволяет добавлять подпись к полю, при наведении на него для уточнения, какие именно данные и в каком формате должен вводить пользователь – рисунок 34.

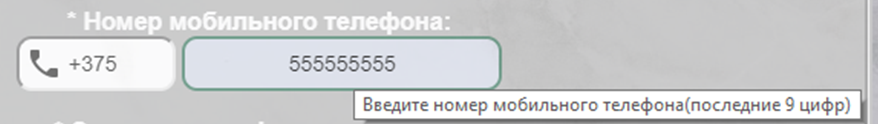


Рисунок 25 – Применение атрибута «title»

Элемент **<fieldset>** предназначен для группирования элементов формы. Такая группировка облегчает работу с формами, содержащими большое число данных. Например, один блок может быть предназначен для ввода текстовой информации, а другой — для флажков.

Браузеры для повышения наглядности отображают результат использования тега <fieldset> в виде рамки, ее цвет, ширину и остальные свойства можно изменить с помощью css.

Тег **<legend>** применяется для создания заголовка группы элементов формы, которая определяется с помощью тега <fieldset>. Группа элементов обозначается в браузере с помощью рамки, а текст, который располагается внутри контейнера <legend>, встраивается в эту рамку.

На рисунке 35 представлен блок «аутентификиция», а после него – блок

«регистрация в личном кабинете».

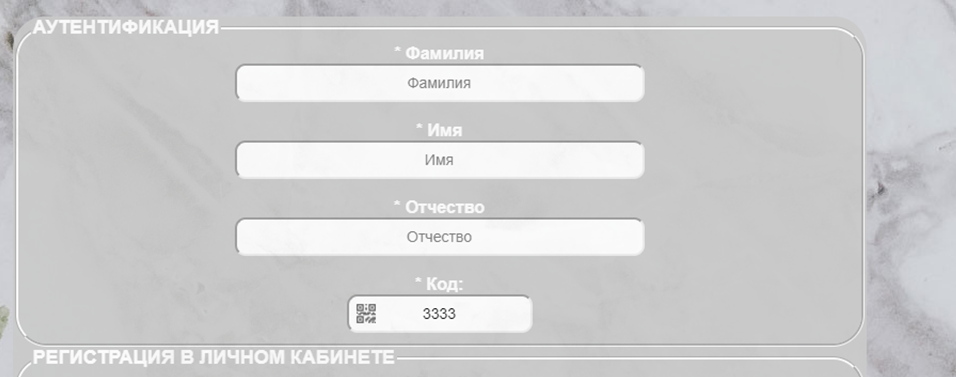


Рисунок 26 – Создание блоков (атрибут «fieldset») и подписей к ним (атрибут «legend»)

Общий синтаксис создания рамки и подписи к ней приведен ниже:

<fieldset>

<legend>Текст</legend>

</fieldset>

Ниже представлены сводные таблица 1 и таблица 2 тегов и атрибутов с их назначением.

Таблица 1 – Теги и их назанчение

|  |  |
| --- | --- |
| **Тег** | **Назначение** |
| <form> | Создание формы |
| <fieldset> | Рамка для блока формы |
| <legend> | Подпись к блоку |
| <input> | Поле формы |
| <label> | Имя поля формы |
| <radio> | Переключатель (выбирается один) |
| <select>  <option value="">Текст </option> (элемент списка)  …  …  </select> | Выбор из раскрывающегося списка |
| <textarea> | Текстовое поле |
| <textarea cols=”…” rows=”…”> | Многострочное текстовое поле |

Таблица 2 – Атрибуты и их назанчение

|  |  |
| --- | --- |
| **Атрибуты** | **Назначение** |
| type=email | Адрес электронной почты |
| type=password | Пароль (скрывает вводимые значения) |
| type=url | URL-адрес |
| type=”datetime-local” | Дата и время |
| type=number | Число |
| type=tel | Телефон |
| type=color | Выбор цвета |
| type=checkbox | Флажки (можно выбрать несколько) |
| type=«submit» | Кнопка отправки данных |
| min | Минимальное значение |
| max | Максимальное значение |
| step | Шаг |
| minlength | Минимальная длина (число знаков) |
| maxlength | Максимальная длина |
| pattern | Выражение, которому должны соответствовать вводимые данные |
| required | Обязательное поле |
| title | Подпись к полю (при наведении) |
| placeholder | Подпись поля (внутри) |
| autofocus | Фокус на элементе во время загрузки страницы |
| disabled | Защищенное от изменений поле |
| value | Значение по умолчанию |

**2 Создание полей формы**

Откроем блокнот/любой другой текстовый редактор/IDE.

Добавим основную структуру html документа и сохраним его с расширением \*.html.

Создадим заголовок формы и саму форму:

<h1>Форма регистрации</h1>

<form method="get" target="\_blank">

Создадим первый блок нашей формы регистрации:

1. Создадим рамку и подпись к ней. Создадим класс с именем «legend», чтобы потом применить ко всем подписям одинаковое оформление CSS:

<fieldset>

<legend class="legend">Аутентификация</legend>

</fieldset>

2. Создадим первые поля. Первые три поля будут иметь тип «text», добавим им идентификаторы id и подписи, чтобы поля имели названия и эти названия были связаны с конкретными полями, с помощью тега for элемента label (то есть label for=id поля).

3. Перенесем поле формы на строку под подписью формы (label) с помощью тега <br>.

4. Зададим возможные значения вводимых данных с помощью атрибута pattern="^[А-Яа-яЁёa-zA-Z]+$" (только кириллица и латинские буквы).

5. Добавим атрибут required, чтобы поле было обязательным для заполнения.

6. Добавим подсказывающий текст, который будет виден в поле формы и исчезнет при Добавим атрибут autofocus, чтобы при открытии страницы с формой, курсор уже находился в поле с фамилией.введении данных с помощью атрибута placeholder=”Фамилия”.

7. Добавим атрибут autofocus, чтобы при открытии страницы с формой, курсор уже находился в поле с фамилией.

8. Зададим максимальную длину символов с помощью атрибута maxlength=”18”.

9. Определим размер поля с помощью size="50".

Первые 3 поля готовы и выглядят примерно так(рисунок 36):

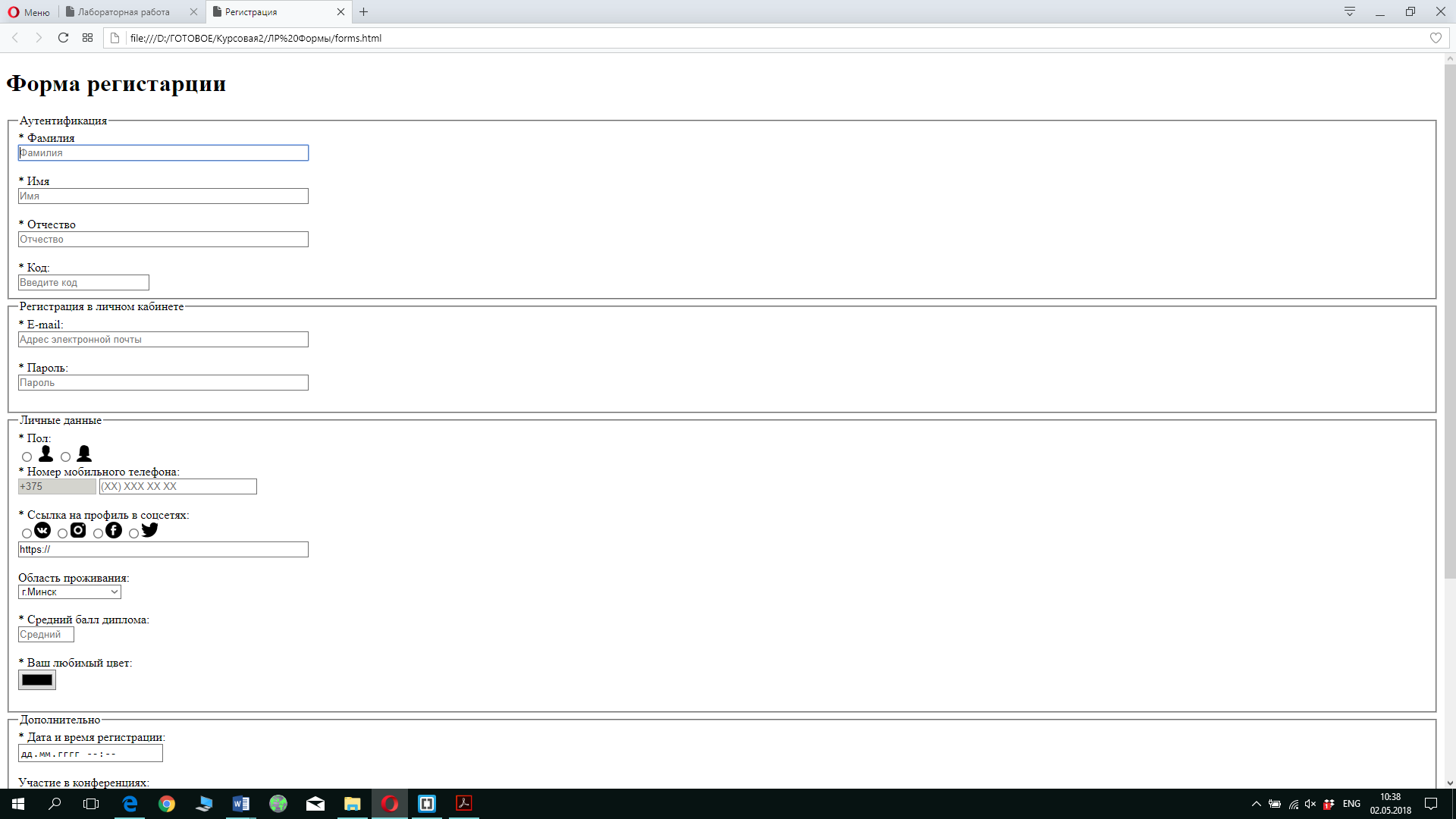


Рисунок 27 – Создание текстового поля

*<label for="surname">\* Фамилия</label><br>*

*<input type="text" id="surname" name="surname" pattern=**"^[А-Яа-яЁёa-zA-Z]+$" required placeholder="Фамилия" autofocus size="50" title="Введите Вашу фамилию" maxlength="18><br><br>*

*<label for="name">\* Имя</label><br>*

*<input type="text" id="name" name="name" pattern="^[А-Яа-яЁёa-zA-Z]+$" required placeholder="Имя" size="50" title="Введите Ваше имя" maxlength="18"><br><br>*

*<label for="patronymic">\* Отчество</label><br>*

*<input type="text" id="patronymic" name="patronymic" pattern="^[А-Яа-яЁёa-zA-Z]+$" required placeholder="Отчество" size="50" title="Введите Ваше отчество" maxlength="18"><br><br>*

10.Следующее поле должно содержать одну цифру и 3 заглавные буквы латинского алфавита. Зададим это с помощью атрибута pattern="[0-9][A-Z]{3}". Тип поля можно не задавать.

11.Количество символов у нас должно быть ограничено 4. Добавим атрибут maxlength="4". Данный атрибут задает количество элементов, которые могут быть введены. Его можно использовать с атрибутами min, max (но не для поля number).

*<label for="code">\* Код:</label><br>*

*<input pattern="[0-9][A-Z]{3}" name="code" id="code" required maxlength="4" title="Введите код(одна цифра от 1 до 9 и три заглавные буквы латинского алфавита от A до Z)" placeholder="Введите код">*

Первый блок готов (рисунок 37):

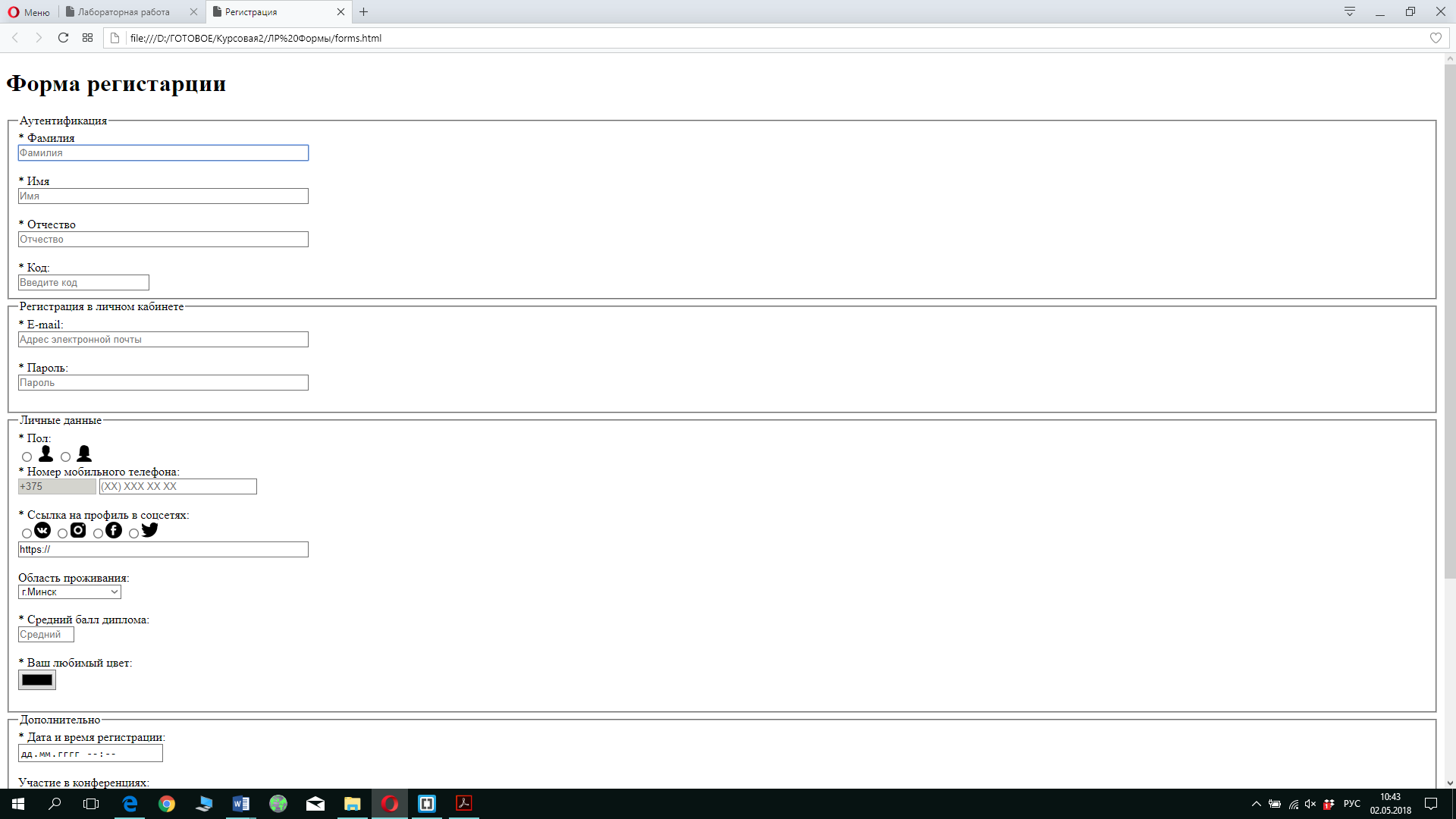


Рисунок 28 – Создание первого блока

По аналогии сделаем второй блок (рисунок 38):

*<fieldset>*

*<legend class="legend">Регистрация в личном кабинете</legend>*

*<label for="email">\* E-mail:</label><br>*

*<input type="email" id="email" name="email" required placeholder="Адрес электронной почты" size="50" title="Введите Ваш адрес электронной почты"><br><br>*

*<label for="password">\* Пароль:</label><br>*

*<input type="password" id="password" name="password" required placeholder="Пароль" size="50" minlength="4" maxlength="20" title="Введите пароль (от 4 до 20 знаков)"><br><br>*

*</fieldset>*

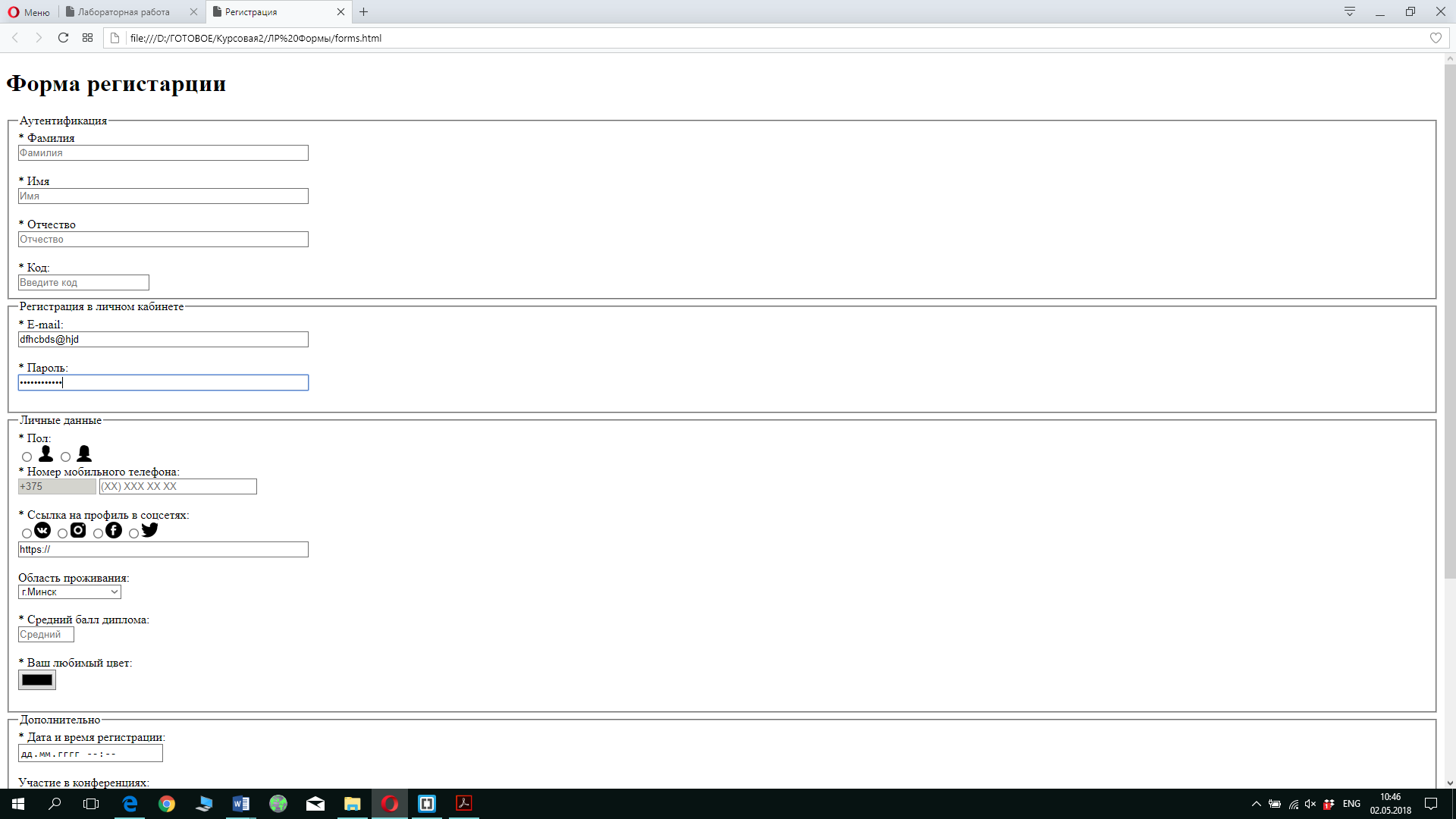


Рисунок 29 – Создание второго блока («e-mail» и «password»)

Третий блок содержит тип поля переключатель (radio). Для того, чтобы выбрать пол, создадим два переключателя и зададим им одно имя – name (так браузер будет понимать, что они связаны, и выбирать только одно значение из двух). Вместо слов для названия поля можно выбрать картинки: <img src="image/woman.png" title="Женский" alt="Ж">.

*<fieldset>*

*<legend class="legend">Личные данные</legend>*

*<label for="sex">\* Пол:</label><br>*

*<label><input type="radio" name="sex" class="sex" required title="Мужской"> <img src="image/man.png" title="Мужской" alt="М"> </label>*

*<label><input type="radio" name="sex" class="sex" required title="Женский"> <img src="image/woman.png" title="Женский" alt="Ж"></label><br>*

Поле номер мобильного телефона разобьем на 2 поля. Первое – с кодом – это поле не будет изменяться(атрибут disabled). Значение будет постоянным " value="+375".

Второе поле будет содержать только цифры и их должно быть 9. Зададим это с помощью атрибута pattern="[0-9]{9}"

*<label for="tel">\* Номер мобильного телефона:</label><br>*

*<input type="tel" id="tel\_code" name="tel\_code" value="+375" disabled size="10">*

*<input type="tel" id="tel" name="tel" required pattern="[0-9]{9}" title="Введите номер мобильного телефона(последние 9 цифр)" size="25" placeholder="(XX) XXX XX XX" maxlength="9"><br><br>*

Поле для ссылки требует адрес, введенные в формате http:// или https://.

Зададим сразу ему значение по умолчанию с помощью value="https://" (так как соцсети имеют защищенное соединение)

*<input type="url" id="url" name="url"* *value="https://" required placeholder="Введите ссылку в формате https://" size="50" title='Введите ссылку на Ваш профиль в социальных сетях в формате "https://"'><br><br>*

Для того, чтобы выбрать значение из раскрывающегося списка, используем тег select. Внутри тега select тег option и пустой атрибут value. Если переключатели мы связывали между собой с помощью одинаковых имен(name), то значения раскрывающегося списка связываются тем, что находятся внутри родительского элемента select.

*<label for="city">Область проживания:</label><br>*

*<select name="city" id="city" required title="\* Выберите область Вашего проживания">*

*<option value="">Брестская обл.</option>*

*<option value="">Витебская обл.</option>*

*<option value="">Гомельская обл.</option>*

*<option value="">Гродненкская обл.</option>*

*<option value="">Минская обл.</option>*

*<option value="" selected г.Минск</option>*

*<option value="">Могилевская обл.</option>*

*</select><br><br>*

Поле со средним баллом диплома может содержать только значения от 4 до 10. Зададим это с помощью min="4" max="10".Шаг будет равен 0,1 step="0.1".

Хотелось бы еще задать максимальное количество вводимых символов, но именно для типа поля number это будет работать не во всех браузерах.

*<label for="diploma">\* Средний балл диплома:</label><br>*

*<input type="number" id="diploma" name="diploma" min="4" max="10" step="0.1" required title="Введите средний балл Вашего диплома" placeholder="Средний балл"><br><br>*

Поле с цветом представляет собой самое простое поле, для которого можно указать только id, name и title.

*<label for="color">\* Ваш любимый цвет:</label><br>*

*<input type="color" id="color" name="color" title="Выберите Ваш любимый цвет:"><br><br>*

*</fieldset>*

Результат создания третьего блока (рисунок 39):

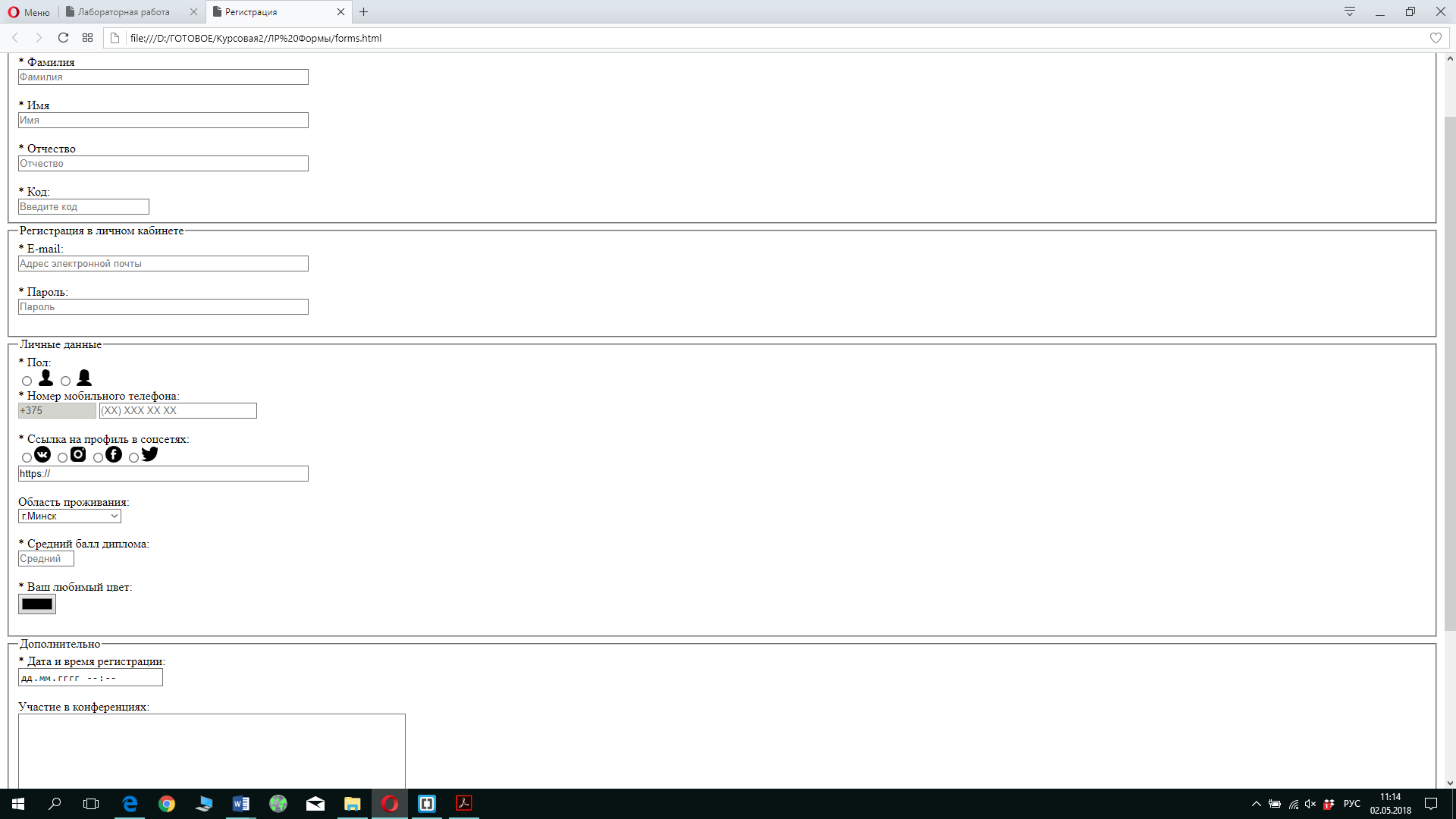


Рисунок 30 – Создание третьего блока

Четвертый блок (рисунок 39):

Дата и время. Вы уже должны понимать как задается это поле, в нем нет ничего сложного.

<fieldset>

<legend class="legend">Дополнительно</legend>

<label for="datetime">\* Дата и время регистрации:</label><br>

<input type="datetime-local" required id="datetime" name="datetime" title="Введите дату и время регистрации"><br><br>

Далее создадим текстовое поле. Но не обычное, а многострочное. Оно задается с помощью отдельного тега textarea. Ему сразу можно задать количество строк и столбцов cols="70" rows="10". Именно такого размера оно и будет отображаться на веб-странице.

<label for="conferention">Участие в конференциях:</label><br>

<textarea name="conferention" id="conferention" cols="70" rows="10" title="Введите названия конференций, в которых Вы принимали участие, и темы Ваших выступлений"></textarea><br><br>

Тип поля файл открывает диалоговое окно с обзором документов, находящихся у вас на компьютере.

<label for="letter">\* Мотивационное письмо:</label><br>

<input type="file" id="letter" name="letter" required title="Прикрепите краткое мотивационное письмо"><br><br>

</fieldset>

Последний блок содержит флажки. Отличие флажков от переключателей в том, что флажки поддерживают множественный выбор значений, а переключатель только одно значение из множества. Однако связываются они так же, как и переключатели – с помощью одинаковых имен (name).

<fieldset id="rules">

<label><input type="checkbox" name="rules" class="rules" checked> Я хочу получать дайджест новых вакансий;</label><br>

<label><input type="checkbox" name="rules" class="rules" required> \* Я подтверждаю свое согласие с условиями пользовательского соглашения;</label><br>

<label><input type="checkbox" name="rules" class="rules" checked> Отправить автоматическое уведомление на почту о моей успешной регистрации;</label><br><br>

</fieldset>

Завершающий элемент на любой форме – кнопка отправки данных:

<label><input type="submit" id="submit" value="Отправить" title="Отправить"><br></label>

Так же ее можно создать с помощью отдельного тега <button>.

Итак, важно помнить, что каждое поле формы должно обязательно содержать 2 атрибута:

* Имя – name – для передачи значения на сервер;
* Идентификатор – id – для оформления полей с помощью css и связывания отдельных элементов, относящихся к одному полю.

Также необходимо добавлять тег label к каждому полю для того, чтобы можно было легко определить, какие данные вводить.

Последние блоки (рисунок 40):

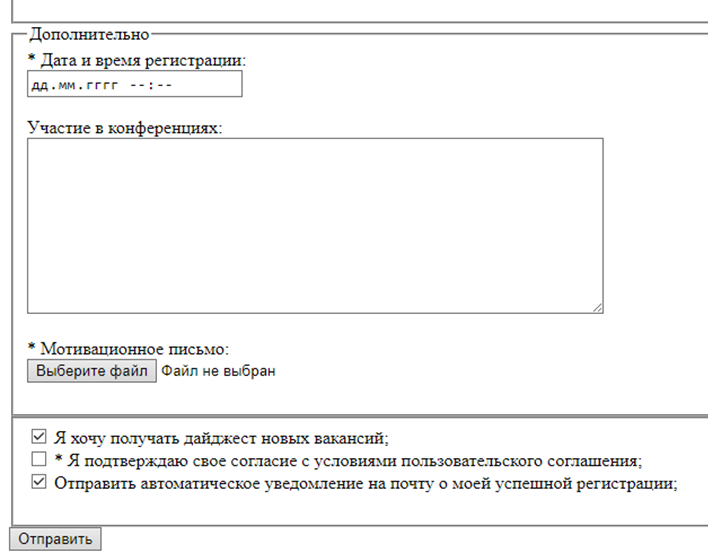


Рисунок 31 – Создание последнего блока

3) Оформление формы

Почему форму регистрации, которую мы сделали нельзя назвать привлекательной? Ответ прост – в ней нет CSS.

Разберем основные моменты.

Для начала откроем наш html документ и вставим ссылку на этот файл css, для этого в head добавляем: <link rel=”stylesheet” href=”name.css”>

Для форм важно, чтобы пользователь сразу видел, что именно он ввел неверно. То есть необходимо цветовое выделение правильно заполненных полей зеленым, неправильно заполненных – красным.

В этом нам помогут псевдоклассы.

Зададим для поля(input), заполнение которого является обязательным(required) и которое было заполнено правильно(valid), рамку зеленого цвета:

input:required:valid {

border: solid #558f78 2px;

}

Для поля(input), в котором находится курсор(focus) и которое было заполнено правильно(valid), рамку и фон зеленого цвета:

input:focus:valid {

border: solid #889d96 2px;

background-color: #ceeadf;

}

Для поля(input), в котором находится курсор(focus) и которое было заполнено неправильно( invalid), рамку и фон красного цвета:

input:focus:invalid {

background-color: #eacece;

border: red solid 2px;

}

Для поля(input), на которое наведен курсор мыши(hover), зададим цвет фона(background-color), прозрачность(opacity) и цвет рамки(border-color):

input:hover {

background-color: #e5ebf7;

opacity: 0.8;

border-color: #a7bbe4;

}

Для всего тела нашей страницы зададим цвет фона:

body {

background-color: blue;

}

Зададим оформление для заголовка:

h1 {

text-align: center;

color: white;

text-transform: uppercase;

font-family: Arial;

opacity: 0.8;

}

Для самой формы, которая содержит различные поля ввода данных, зададим выравнивание текста(text-align),цвет фона(background), шрифт(font-family), цвет шрифта(color), ширину блока с формой(width), выравнивание формы (margin), размер шрифта(font-size).

form {

text-align: center;

background: #cacaca;

font-family: Arial;

color: white;

width:800px;

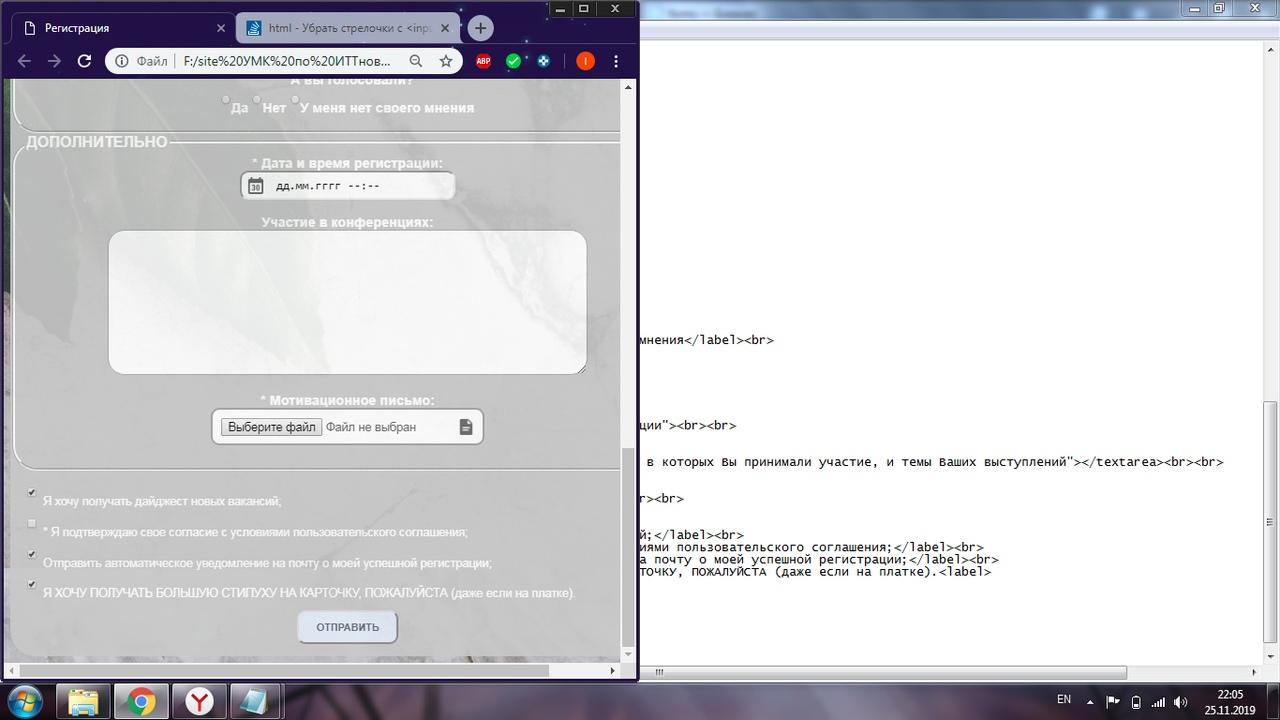
margin: 0 auto;

font-size: 16px;

}

**Контрольные задания**

1. Добавить новый блок формы с различными видами полей, используя:
   1. Поле со списком (минимум 3 значения)
   2. Переключатели
   3. Текстовое поле (добавить ограничения по минимальному и максимальному количеству символов)
   4. Многострочное текстовое поле
   5. Добавить поле «Возраст» (type=number, минимальное значение – 10, максимальное – 25, шаг – 1)
   6. Добавить текстовое поле «Номер группы», с помощью внешней таблицы стилей задать значение по умолчанию (№ своей группы) и защитить поле от изменений.
2. Добавить поле с флажком (допускаются авторские изменения).



1. Сделать пароль видимым и изменить выражение, которому должны соответствовать данные поля «Пароль».
2. Сделать поле электронного адреса необязательным.
3. Изменить выражение, которому должны соответствовать данные поля «Код» (одна цифра от 1 до 9 и две прописные буквы латинского алфавита.
4. Изменить тип рамки для созданного блока (точечный).
5. Изменить прозрачность формы, цвет границы блоков формы.
6. Наименования блоков формы передвинуть в правый угол.
7. Изменить фон для блока «Регистрация в личном кабинете».

**Контрольные вопросы**

1. Каков принцип работы атрибута required ?
2. Какой тег необходимо использовать для создания заголовка группы элементов формы, которая определяется с помощью тега <fieldset> ?
3. Можно ли вводить телефонный номер в поле с типом number ?
4. Объясните принцип работы атрибута pattern ?
5. Для чего нужны атрибуты min и max ?
6. Какой тип данных нужно поставить вместо троеточия <input type=…>, чтобы заставить браузер скрывать значения, вводимые пользователем ?
7. Какой атрибут отвечает за степень подробности вводимых данных ?