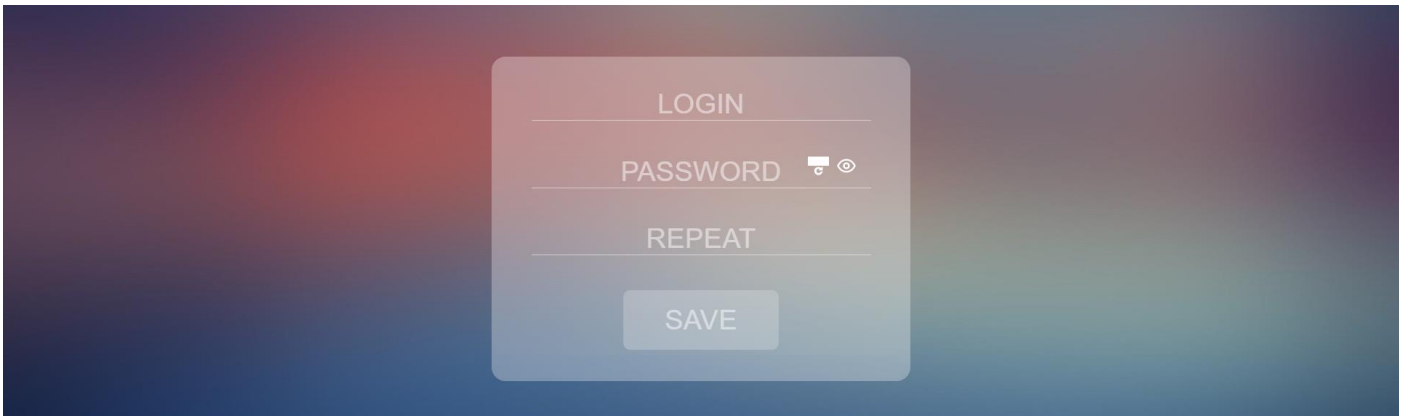
Форма регистрации/авторизации



- создать **HTML** файл с **формой**, в которой **поля** для ввода **логина**, **пароля**, **повтора** пароля, **кнопка** для сохранения данных в **локальном хранилище**.

- учитывайте, что в поле для **пароля** необходимо при помощи **абсолютного** позиционирования создать **иконки** **сгенерировать** пароль и **показать/скрыть** пароль. Для поля **повтор** пароля также создать иконку **показать/скрыть** пароль. Иконки лучше сделать при помощи тега, для создания **ссылок**. Также необходимо создать **общий контейнер** для поля и ссылок, для которого задать **относительное позиционирование**.

- все **проверки по заполнению** полей и **контролю символов** необходимо делать **самостоятельно при помощи JS**, поэтому использовать специализированный тип поля для логина (например **email**) или атрибут для обязательного заполнения поля **required запрещено**.

- для **CSS** можно использовать **классы**, а для **JS** - **идентификаторы**.

- добавьте **подсказку** пользователю, какую информацию необходимо ввести в поле.

```
<form>
    <input type="text" id="login" placeholder="login">
    <div class="block">
        <input type="password" id="password" placeholder="password">
        <a href="#" id="control"></a>
        <a href="#" id="generate"></a>
    </div>
    <input type="password" id="repeat" placeholder="repeat">
    <input type="button" id="save" value="save">
</form>
```

Саманчук Елена Дмитриевна
samancuk_e@itstep.academy

- в стилях настройте **универсальный селектор**, тег **body**, **форму**, **поля** для ввода, **иконки**, **кнопку**, **шрифты**, **отображение при изменении размера** экрана устройства.
- иконки можно найти на сайте **Icons8**, шрифты - **Google Fonts**.

```css
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {
    width: 100vw;
    height: 100vh;
    overflow: hidden;
    display: flex;
    align-items: center;
    justify-content: center;
    background-image: url(https://celes.club/uploads/posts/2022-05/1653348956_1-celes-club-p-fon-dlya-avtorizatsii-krasivie-1.jpg);
    background-repeat: no-repeat;
    background-size: cover;
    background-position: center;
}

form {
    width: 30vw;
    height: 50vh;
    border: none;
    border-radius: 1vw;
    background: rgba(255, 255, 255, 0.3);
    backdrop-filter: blur(2vw);
    padding: 1vw;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: space-around;
    font-family: sans-serif;
}
```

```css
.block {
    position: relative;
}

#control {
    position: absolute;
    top: 0;
    right: 1vw;
    display: block;
    width: 1.5vw;
    height: 1.5vw;
    background-image: url(open.png);
    background-size: cover;
    filter: invert(1);
}

#control.view {
    background-image: url(close.png);
    background-size: cover;
}

#generate {
    position: absolute;
    top: 0;
    right: 3vw;
    display: block;
    width: 1.5vw;
    height: 1.5vw;
    background-image: url(generate.png);
    background-size: cover;
    filter: brightness(100);
    -webkit-filter: brightness(100);
}
```

Саманчук Елена Дмитриевна
samancuk_e@itstep.academy

- не забывайте про размещение при помощи **flex**, настройку **эффектов**.

```css
input {
    border: none;
    border-bottom: 0.1vw solid rgba(255, 255, 255, 0.5);
    outline: none;
    background: none;
    text-transform: uppercase;
    font-size: 2vw;
    text-align: center;
    color: rgba(255, 255, 255, 0.5);
}

input::placeholder {
    color: rgba(255, 255, 255, 0.5);
}

input[type='button'] {
    border: none;
    border-radius: 0.5vw;
    padding: 1vw 3vw;
    background: rgba(228, 228, 228, 0.3);
    text-transform: uppercase;
    color: rgba(255, 255, 255, 0.5);
    font-size: 2vw;
}

input[type='button']:hover {
    cursor: pointer;
    background: rgba(255, 255, 255, 0.3);
}

input[type='button']:active {
    transform: scale(0.95);
}
```

- для проверки количества символов в поле можно использовать метод **length**. Для **проверки сложности пароля** лучше всего использовать простые **регулярные выражения**. Для создания регулярных выражений описываются **символы для проверки**, далее вызывается функция **test**, которой необходимо передать **строку для проверки**. Символ **^** в самом начале регулярного выражения означает поиск символов, **отличающихся** от перечисленных. Также необходимо **сравнить значения** в полях для пароля и повтора пароля.

```js
save.addEventListener('click', () => {
    if (password.value.length < 7 || password.value.length > 15) {
        alert('Пароль должен быть от 7 до 15 символов');
    } else if (!(/[A-Z]/.test(password.value))) {
        alert('Пароль должен содержать хотя бы одну букву в верхнем регистре');
    } else if (!(/[a-z]/.test(password.value))) {
        alert('Пароль должен содержать хотя бы одну букву в нижнем регистре');
    } else if (!(/[0-9]/.test(password.value))) {
        alert('Пароль должен содержать хотя бы одну цифру');
    } else if (!(/[!@#$%^&*]/.test(password.value))) {
        alert('Пароль должен содержать хотя бы один спецсимвол !@#$%^&');
    } else if ((/[^!@#$%^&*A-Z-a-z-0-9]/.test(password.value))) {
        alert('Пароль содержит недопустимые символы. Разрешены английские буквы в верхнем и нижнем регистре, цифры и спецсимволы !@#$%^&');
    } else if (password.value != repeat.value) {
        alert('Повтор пароля не совпадает');
    } else {
        localStorage.clear();
        localStorage.setItem(login.value, password.value);
        window.location.href = 'login.html';
    }
});
```

Саманчук Елена Дмитриевна
samancuk_e@itstep.academy

- логин и пароль можно сохранить в **локальном хранилище** для передачи информации на следующую **страницу для авторизации**. Информация будет храниться в виде пары **{'ключ':'значение'}**, в нашем случае **{'логин':'пароль'}**.

- для **отображения пароля** можно динамически добавлять класс **view**.

```
control.addEventListener('click', () => {
    if (password.getAttribute('type') == 'password') {
        control.classList.add('view');
        password.setAttribute('type', 'text');
    } else {
        control.classList.remove('view');
        password.setAttribute('type', 'password');
    }
});
```

- для **генерации надежного пароля** можно создать строку из подходящих символов, задать длину пароля и, используя функцию **random()**, сгенерировать и вывести пользователю пароль.

```
generate.addEventListener('click', () => {
    let chars = "0123456789abcdefghijklmnopqrstuvwxyz!@#$%^&*()ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    let passwordLength = 10;
    let pass = "";

    for (let i = 0; i <= passwordLength; i++) {
        let randomNumber = Math.floor(Math.random() * chars.length);
        pass += chars.slice(randomNumber, randomNumber + 1);
    }

    password.value = pass;
    console.log(pass);
});
```

- **проверку логина реализовать самостоятельно**.

- для **авторизации** необходимо реализовать подобную форму, только без поля для повтора пароля.

```
<form>
    <input type="text" id="login" placeholder="login">
    <input type="password" id="password" placeholder="password">
    <input type="button" id="enter" value="enter">
</form>
```

Саманчук Елена Дмитриевна
samancuk_e@itstep.academy

- для **проверки данных пользователя** необходимо получить информацию из **локального хранилища**.

```
enter.addEventListener('click', () => {
    if (login.value != localStorage.key(0)) {
        alert('Пользователь не найден');
        console.log(localStorage.key(0));
    } else if (password.value != localStorage.getItem(login.value)) {
        alert('Неверный пароль');
        console.log(localStorage.getItem(login.value));
    } else {
        alert('Success');
    }
});
```

- используя тип данных **словарь** создать визуальный **англо-русский и русско-английский словарь.**

Саманчук Елена Дмитриевна
samancuk_e@itstep.academy