

„Lista zakupów”



Jakub Karmański

rok 3 semestr 5 - zimowy
Wydział Matematyki Stosowanej
kierunek Informatyka

Spis treści

Założenia projektu	3
Sprawozdanie	3
Instrukcja obsługi:	3
Specyfikacja techniczna	14
Podsumowanie	20

Założenia projektu

Celem projektu jest stworzenie aplikacji pomagającej użytkownikowi w tworzeniu listy zakupów dla całej rodziny. Aplikacja zostanie napisana z wykorzystaniem: html, javascript, php przy użyciu programu visual studio code. W celu testowania działania aplikacji na przeglądarce oraz telefonie zostanie użyty serwer aplikacji xampp.

Celem programu będzie wyświetlanie aktualnych list zakupów użytkownika jak i możliwość edytowania/dodawania/usuwania list. Użytkownik, po zakończeniu zakupów może usunąć produkty z listy i tym samym dać jasny sygnał, że produkty zostały zakupione.

Sprawozdanie

Program "Lista zakupów" jest aplikacją webową, której celem jest przechowywanie/modyfikowanie list zakupów dla użytkownika.

Instrukcja obsługi:

Po wejściu na stronę główną pojawia się ekran logowania, na której wpisujemy swój login i hasło, lub jeżeli nie mamy jeszcze swojego loginu generujemy swój unikalny login.

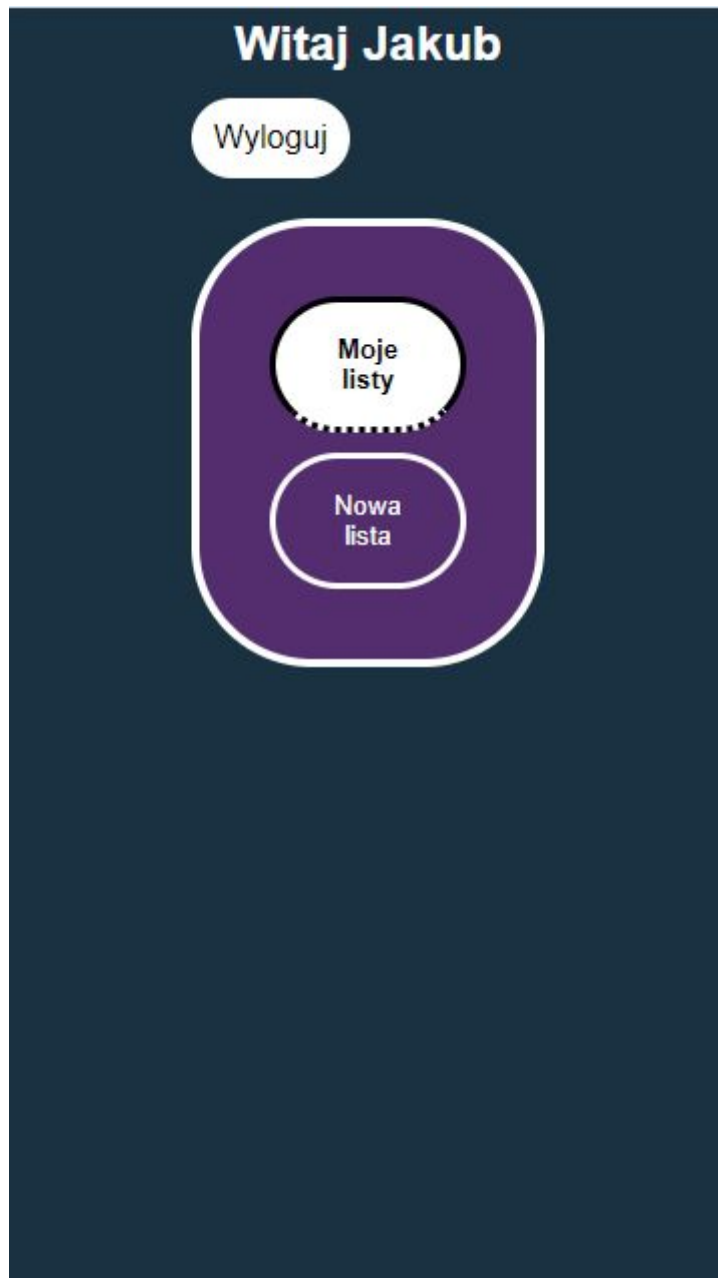
Witamy w liście zakupów!

Podaj swój unikalny login
użytkownika:

Dalej

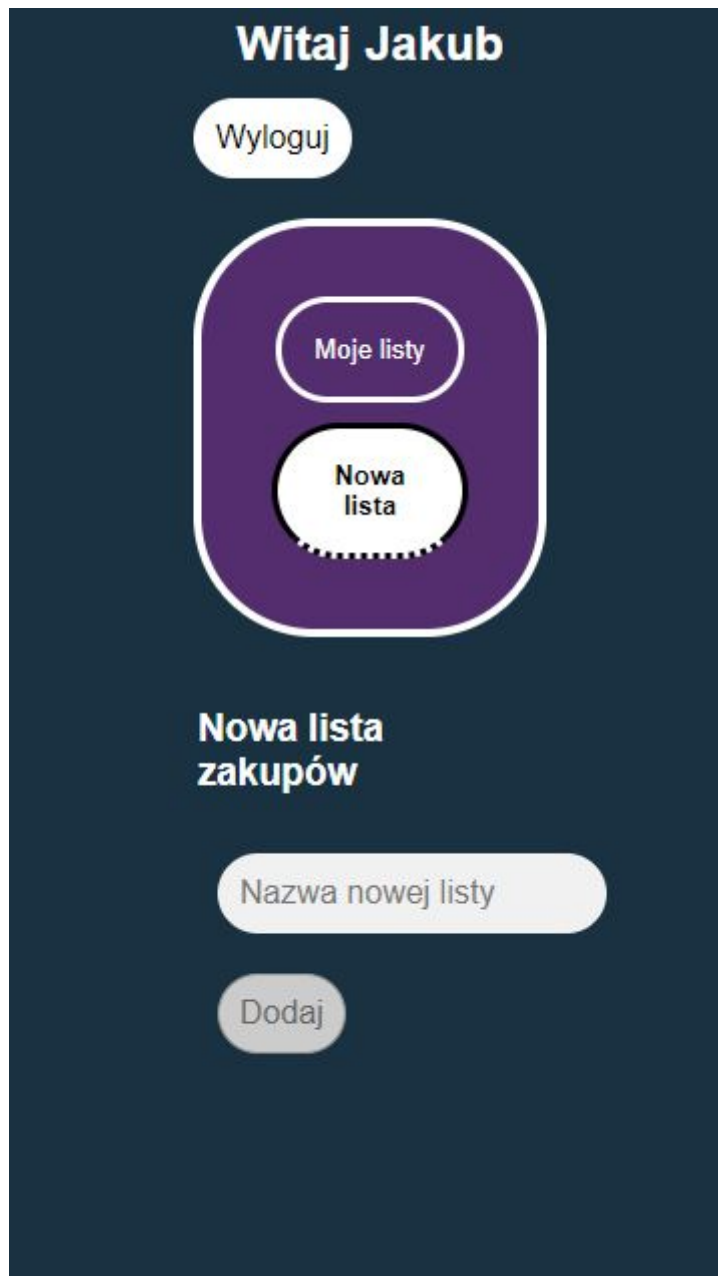
Wygeneruj unikalny login

Po wpisaniu swojego użytkownika jest on zapamiętywany w przeglądarce , zatem nie musimy logować się kolejny raz.



Na ekranie głównym widzimy przycisk do wylogowywania użytkownika oraz menu główne posiadające dwa przyciski:

1. Moje listy - do zarządzania listami użytkownika
2. Nowa lista - wykorzystywana w celu dodania nowej listy zakupów



W celu utworzenia nowej listy zakupów przechodzimy do zakładki nowa lista. Tutaj podajemy nazwę naszej listy i klikamy "dodaj" (przycisk zostanie odblokowany po podaniu nazwy)

Witaj Jakub

Wyloguj

Moje listy

Nowa
lista

**Nowa lista
zakupów**

Nowa lista

Dodaj

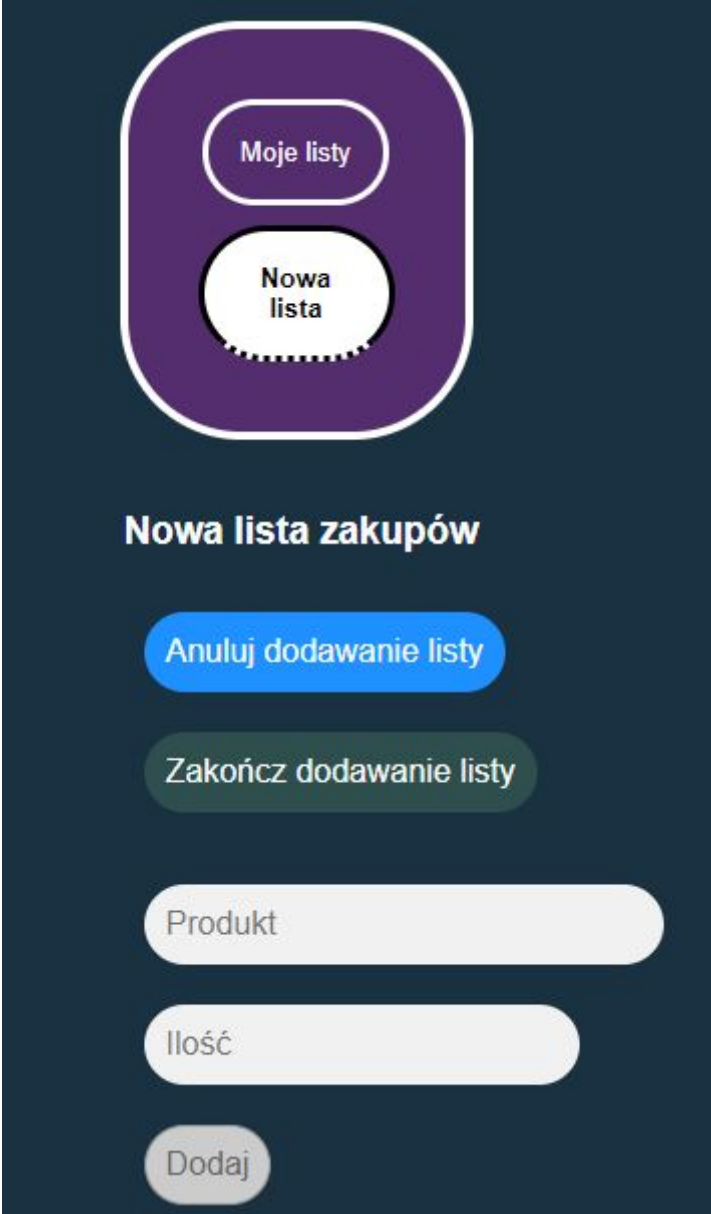
Podczas edycji nowej listy produktów mamy do wyboru 3 akcje:

1. Anuluj dodawanie listy - spowoduje anulowanie dodawania listy oraz usunięcie jej z listy użytkownika

2. Dodawanie nowych produktów - podajemy nazwę produktu oraz jego ilość , następnie po odblokowaniu się przycisku dodaj - dodajemy produkt do naszej listy

3. Zakończenie dodawania listy produktów - spowoduje wyjście z trybu dodawania listy i przejście do ekranu dodawania nowej listy.

(jeśli nie zostały wprowadzone żadne zakupy lista nie zapisuje się w listach użytkownika)



The screenshot shows a mobile application interface for creating a new shopping list. At the top, there is a purple rounded rectangle containing two buttons: 'Moje listy' (My lists) and 'Nowa lista' (New list). Below this, the title 'Nowa lista zakupów' (New shopping list) is displayed. Under the title, there are two buttons: 'Anuluj dodawanie listy' (Cancel adding list) in blue and 'Zakończ dodawanie listy' (Finish adding list) in green. Below these buttons are two input fields: 'Produkt' (Product) and 'Ilość' (Quantity). At the bottom, there is a 'Dodaj' (Add) button.

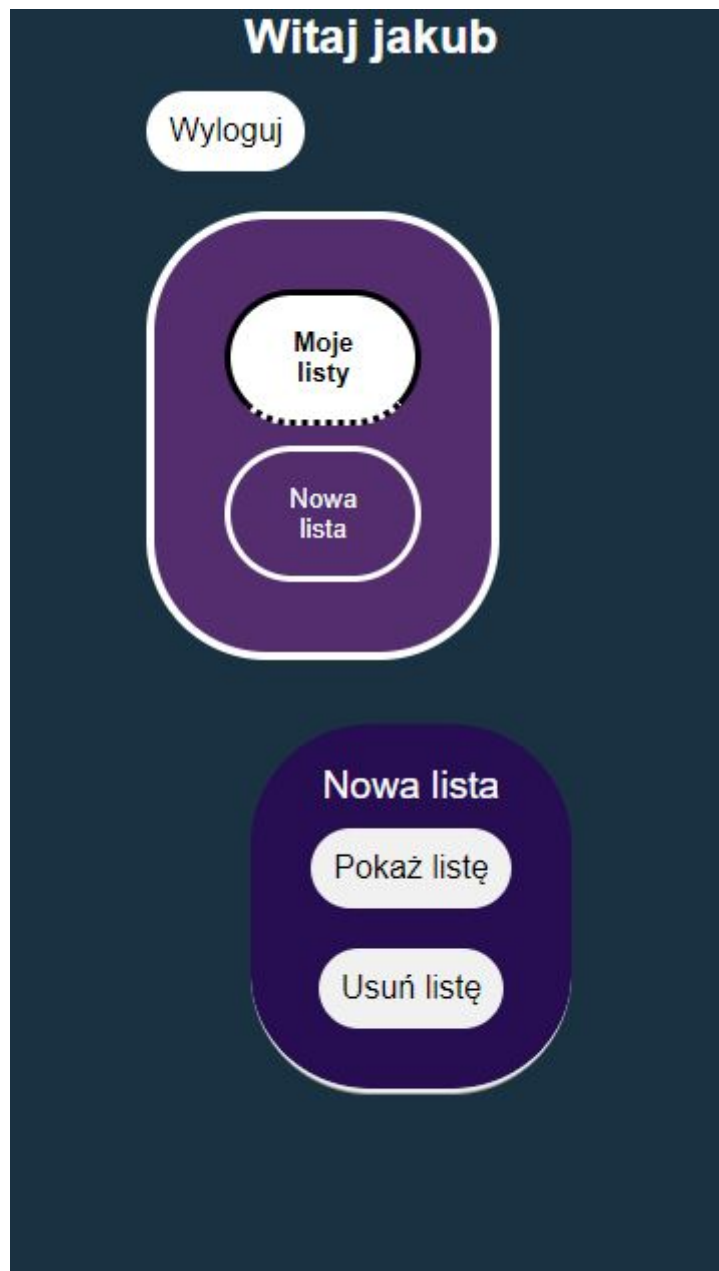
Każdy dodany przez nas produkt zapisywany jest przez aplikację w liście użytkownika, a wykorzystane produkty podpowiadane będą podczas chęci dodania ich ponownie. W tym celu wykorzystywany jest komponent z tzw. autocompletem, który podpowiada nam o jaki produkt może nam chodzić.

The image shows a mobile application interface for editing a product list. At the top, there are two existing items, each represented by a dark purple circle with white text. The first item is labeled 'bób 1' and has a white button with the text 'Usuń' (Delete) inside. The second item is labeled 'brokuł 1' and also has a white 'Usuń' button. Below these items are three input fields. The first field contains the letter 'b'. The second field is labeled 'Ilość' (Quantity). Below the input fields is a button labeled 'Dodaj' (Add). At the bottom of the screen, there is a list of suggestions, each in a dark purple rounded rectangle with white text: 'bakłażan', 'botwina', 'bób', and 'brokuł'.

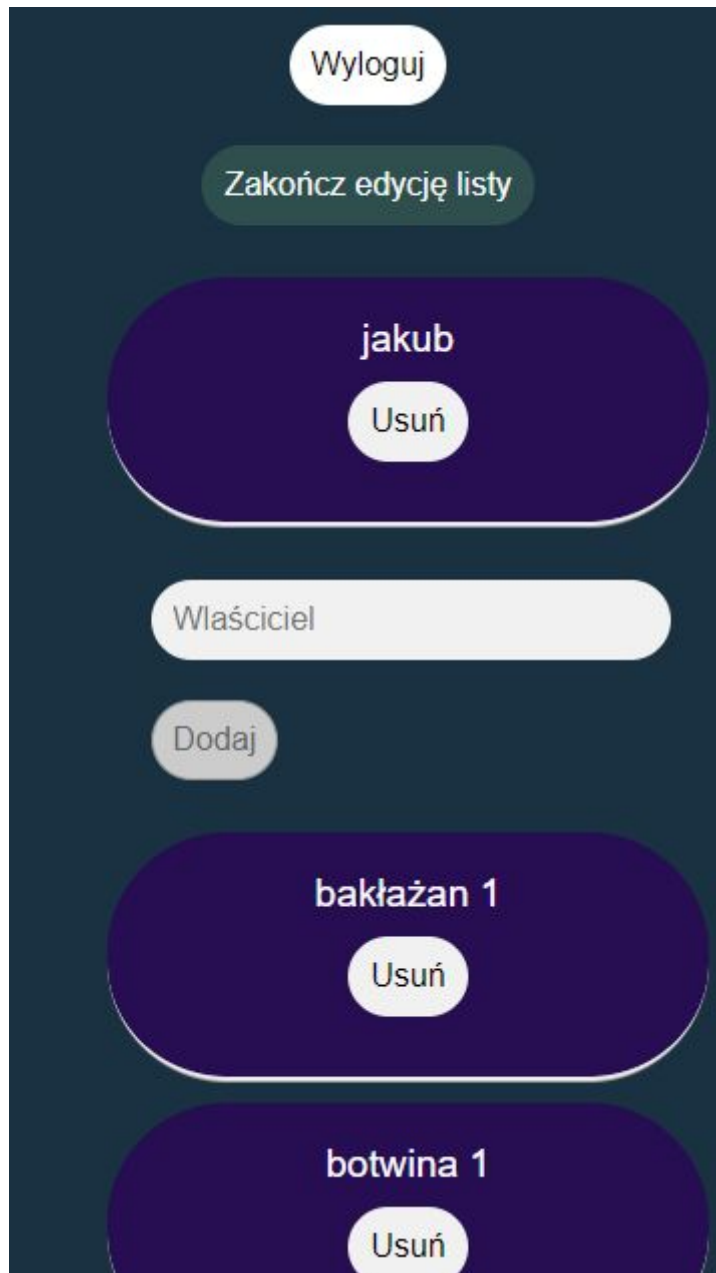
W widoku edycji istnieje również funkcjonalność służąca do usuwania produktu z naszej listy, co może być przydatne w momencie pomyłki.

Wracając do widoku głównego widzimy nowo dodane listy produktów i możemy wybrać z dwóch opcji:

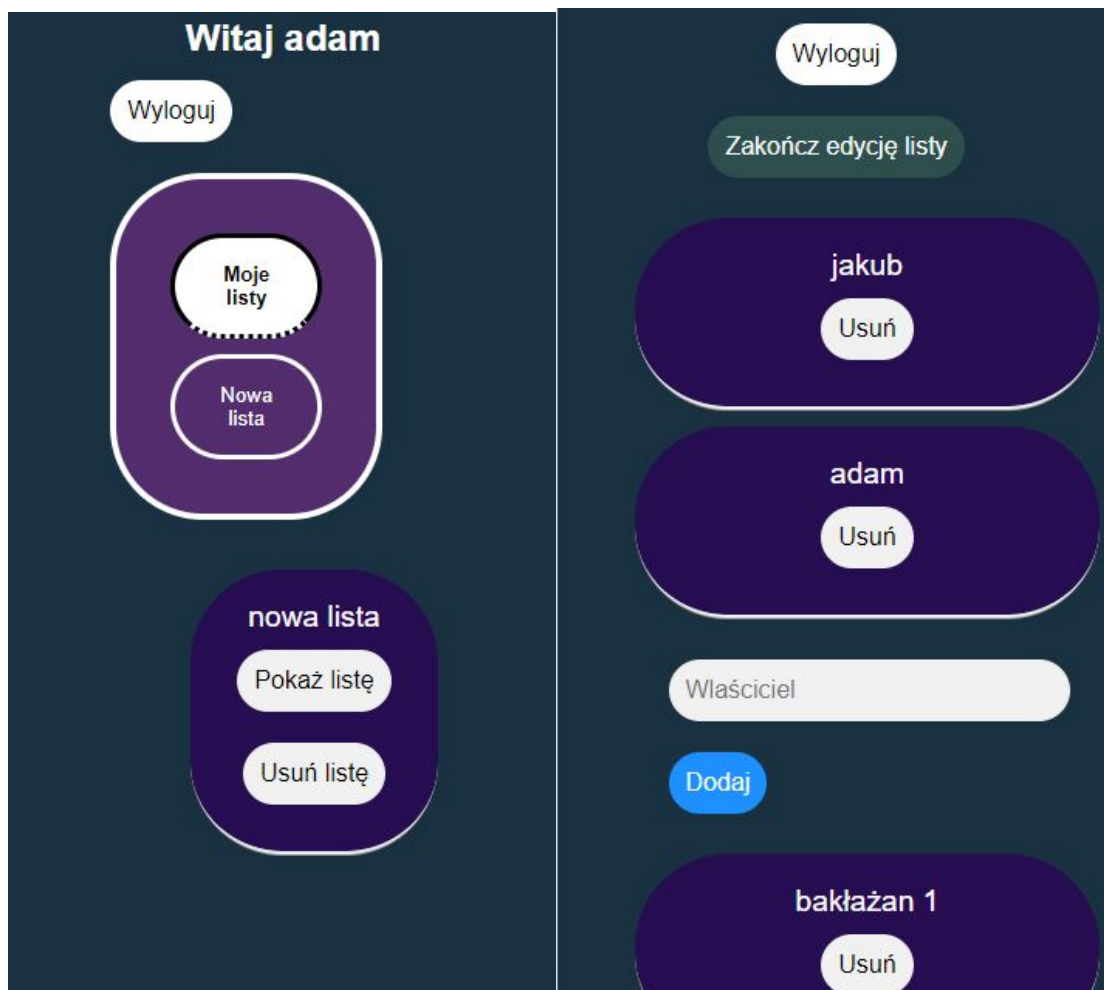
1. Usunięcia listy produktów
2. Pokazania oraz edycji listy



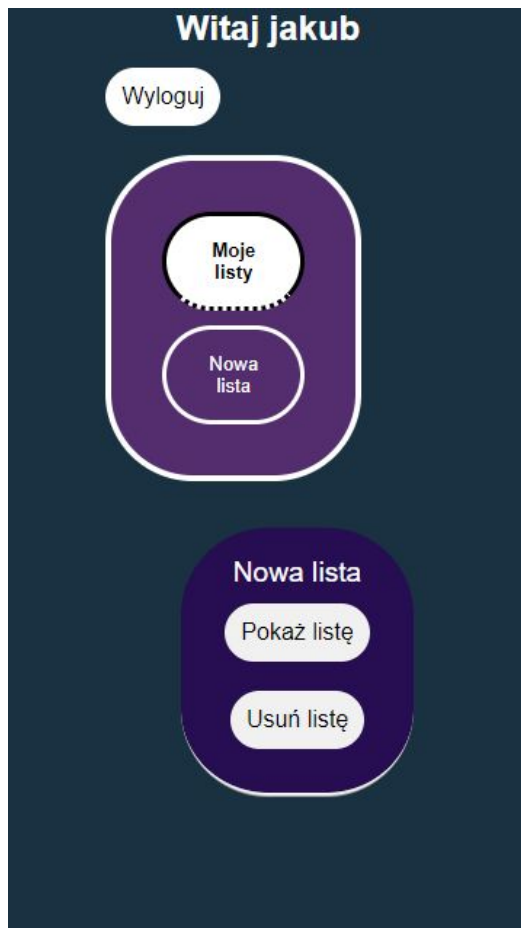
Wchodząc w tryb edycji w nowym oknie ukazuje się widok podobny do widoku dodawania nowej listy , lecz poszerzony o możliwość dodawania nowych użytkowników do listy



Dodając nowego użytkownika do listy będzie on miał taki sam dostęp do aplikacji jak my, a lista pokaże się w jego liście:



Będzie miał on kontrolę nad listą



Dodatkowo aplikacje jest odświeżana co interwał 10 sekund , tak aby użytkownik widział wszelkie modyfikacje innych użytkowników bez konieczności odświeżania strony.

Specyfikacja techniczna

Aplikacja składa się z 7 plików:

1.index.html - odpowiada za widok główny (ekran logowania)

```
<div class="content">

  <body onload="checklocalStorage()">
    <header>
      <h2>Witamy w liście zakupów!</h2>
    </header>
    <main>
      Podaj swój unikalny login użytkownika:
      <input type="text" id="id" name="id" required onkeyup="stoppedTyping()"><br>
      <input type="button" value="Dalej" onclick="setlocalStorage()" id="start_button" disabled></input>
      <input type="button" value="Wygeneruj unikalny login" onclick="generate()"></input>
    </main>
    <footer>
    </footer>
  </body>
</div>

<!DOCTYPE html>
<html lang="pl">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" type="text/css" href="stylesMainPage.css">
</head>

<script>
function setlocalStorage() {
  if (document.getElementById('id').value != '') {
    localStorage.setItem('ID', document.getElementById('id').value);
    window.location.href = "page1.html?id=" + document.getElementById('id').value;
  }
}

function checklocalStorage() {
  if (localStorage.getItem("ID")) {
    window.location.href = "page1.html";
  }
}

var items = Array("Aardvark", "Albatross", "Alligator", "Alpaca", "Ant", "Anteater", "Antelope", "Ape",
  "Armadillo", "Donkey", "Baboon", "Badger", "Barracuda", "Bat", "Bear", "Beaver", "Bee", "Bison", "Boar",
  "Buffalo", "Butterfly", "Camel", "Capybara", "Caribou", "Cassowary", "Cat", "Caterpillar", "Cattle", "Chamois",
  "Cheetah", "Chicken", "Chimpanzee", "Chinchilla", "Chough", "Clam", "Cobra", "Cockroach", "Cod", "Cormorant", "Coyote",
  "Crab", "Crane", "Crocodile", "Crow", "Curlew", "Deer", "Dinosaur", "Dog", "Dogfish", "Dolphin", "Dotterel", "Dove",
  "Dragonfly", "Duck", "Dugong", "Dunlin", "Eagle", "Echidna", "Eel", "Eland", "Elephant", "Elk", "Emu", "Falcon", "Ferret",
  "Finch", "Fish", "Flamingo", "Fly", "Fox", "Frog", "Gaur", "Gazelle", "Gerbil", "Giraffe", "Gnat", "GNU", "Goat", "Goldfinch", "Goldfish",
  "Goose", "Gorilla", "Goshawk", "Grasshopper", "Grouse", "Guanaco", "Gull", "Hamster", "Hare", "Hawk", "Hedgehog", "Heron", "Herring",
  "Hippopotamus", "Hornet", "Horse", "Human", "Hummingbird", "Hyena", "Ibex", "Ibis", "Jackal", "Jaguar", "Jay", "Jellyfish", "Kangaroo",
  "Kingfisher", "Koala", "Kookabura", "Kouprey", "Kudu", "Lapwing", "Lark", "Lemur", "Leopard", "Lion", "Llama", "Lobster", "Locust", "Loris",
  "Louse", "Lyrebird", "Magpie", "Mallard", "Manatee", "Mandrill", "Mantis", "Marten", "Meerkat", "Mink", "Mole", "Mongoose", "Monkey", "Moose",
  "Mosquito", "Mouse", "Mule", "Narwhal", "Newt", "Nightingale", "Octopus", "Okapi", "Opossum", "Oryx", "Ostrich", "Otter", "Owl", "Oyster", "Panther",
  "Partridge", "Peafowl", "Pelican", "Penguin", "Pheasant", "Pig", "Pigeon", "Pony", "Porcupine", "Porpoise", "Quetzal", "Rabbit", "Raccoon",
  "Rail", "Ram", "Rat", "Raven", "Shark", "Sheep", "Snail", "Snake", "Spider", "Squirrel", "Starling", "Stingray", "Stinkbug", "Stork", "Swallow",
  "Swan", "Tapir", "Tarsier", "Termite", "Tiger", "Toad", "Trout", "Turkey", "Turtle", "Viper", "Vulture", "Wallaby", "Walrus", "Wasp", "Weasel",
  "Whale", "Wildcat", "Wolf", "Wolverine", "Wombat", "Woodcock", "Woodpecker", "Worm", "Zebra");

function generate() {
  document.getElementById("id").value = items[Math.floor(Math.random() * items.length)] + (Math.random().toString(10).substr(2, 5)).toString();
  document.getElementById('start_button').disabled = false;
}
}
```

```

function stoppedTyping() {
    if (document.getElementById("id").value.length > 0) {
        document.getElementById('start_button').disabled = false;
    } else {
        document.getElementById('start_button').disabled = true;
    }
}
</script>

```

2.lista.html

```

<div class="content">
    <body onload="start()">
        <main>
            <div>
                <input type="submit" value="Wyloguj" name="wyloguj" id="wyloguj" onclick="wyloguj()">
            </div>
            <div class="anulujDodawanieListy" id="anulujDodawanieListy">
                <input type="submit" value="Zakończ edycję listy" name="zakoncClick" id="zakoncClick" onclick="zakoncClick()">
            </div>
            <div class="panelWlascieli1" id="panelWlascieli1">
                <div class="listaWlascieli1">
                    <ul id="wlasciele"></ul>
                </div>
                <div class="autocomplete">
                    <div id="dodawanieProduktu">
                        <input id="myInputWlasciciel" type="text" name="wlasciciel" onpaste="return false" required onkeyup="stoppedTypingWlasciciel()" placeholder="Właściciel">
                    </div>
                    <input type="submit" value="Dodaj" name="dodajWlascicielaClick" id="dodajWlascicielaClick" disabled onclick="dodajWlascicielaDoListy()">
                </div>
            </div>
            <div class="panelProduktow" id="panelProduktow">
                <div class="listaProduktow">
                    <ul id="produkty"></ul>
                </div>
                <div class="autocomplete">
                    <div id="dodawanieProduktu">
                        <input id="myInput" type="text" name="produkt" onpaste="return false" required onkeyup="stoppedTyping()" placeholder="Produkt">
                        <input id="ilosc" type="number" name="ilosc" required oninput="stoppedTyping()" placeholder="Ilość">
                    </div>
                    <input type="submit" value="Dodaj" name="dodajClick" id="dodajClick" disabled onclick="dodajDoListy()">
                </div>
            </div>
        </main>
    </body>
</div>

```

```

<script>
    var products = [];

    function start() {
        if (localStorage.getItem("lista")) {
            odczytajListeProduktow(localStorage.getItem("lista").toString().toLowerCase());
            try {
                pobierzSlownikProduktow();
            } catch (error) {}
        } else {
            window.close();
        }
    }

    function odczytajListeProduktow(nazwa) {
        const toSend = {
            opcja: 0,
            id: nazwa
        }
        const jsonString = JSON.stringify(toSend);
        var req = new XMLHttpRequest();
        req.open('POST', 'listaZakupow.php', false);
        req.onreadystatechange = function(aEvt) {
            if (req.readyState == 4) {
                if (req.status == 200) {
                    myObj = JSON.parse(req.responseText);
                    tworzenieListy(nazwa, myObj);
                }
            } else {
                console.log("Błąd podczas ładowania strony\n");
            }
        };
        req.send(jsonString);
    }

    function tworzenieListy(nazwa, myObj) {
        myObj[0].produkty.forEach(produkt => {
            dodajProduktDoWidokuIstniejacy(produkt.nazwa, produkt.ilosc);
        });
        myObj[0].wlascciele.forEach(wlasciciel => {
            dodajWlascicielaDoWidokuIstniejacy(wlasciciel.nazwa);
        });
    }
}

```


3.page1.html

```
<body onload="start()">
  <div id="content">
    <div id="content-wrap">
      <header>
        <h2 id="title">Witaj w liście zakupów</h2>
      </header>
      <main>
        <div>
          <input type="submit" value="Wyloguj" name="wyloguj" id="wyloguj" onclick="wyloguj()">
        </div>
        <!-- Tab links -->
        <div class="tab">
          <button class="tablinks" id="listy_moje" onclick="openTab(event, 'ListyMoje')">Moje listy </button>
          <button class="tablinks" id="nowa_lista" onclick="openTab(event, 'NowaLista')">Nowa lista</button>
        </div>

        <!-- Tab content -->
        <div id="ListyMoje" class="tabcontent">
          <ul id="mojeListy"></ul>
        </div>

        <div id="NowaLista" class="tabcontent">
          <h3>Nowa lista zakupów</h3>
          <div class="nazwalisty" id="nazwalistyPanel">
            <input id="nazwalisty" type="text" name="nazwalisty" required onkeyup="stoppedTypingNazwalisty()" placeholder="Nazwa nowej listy">
            <input type="submit" value="Dodaj" name="dodajListeClick" id="dodajListeClick" disabled onclick="dodajNazweListy()">
          </div>

          <div class="anulujDodawanieListy" id="anulujDodawanieListy">
            <input type="submit" value="Anuluj dodawanie listy" name="anulujDodawanieListy" id="anulujDodawanie" onclick="anulujDodawanieListy()">
            <input type="submit" value="Zakończ dodawanie listy" name="zakonczClick" id="zakonczClick" onclick="zakonczClick()">
          </div>

          <div class="panelProduktow" id="panelProduktow">
            <div class="listaProduktow">
              <ul id="produkty"></ul>
            </div>
            <div class="autocomplete">
              <div id="dodawanieProduktu">
                <input id="myInput" type="text" name="produkt" onpaste="return false" required onkeyup="stoppedTyping()" placeholder="Produkt">
                <input id="ilosc" type="number" name="ilosc" required oninput="stoppedTyping()" placeholder="Ilość">
              </div>
              <input type="submit" value="Dodaj" name="dodajClick" id="dodajClick" disabled onclick="dodajDoListy()">
            </div>
          </div>
        </div>
      </main>
    </div>
  </div>
</body>
```

4.listaZakupow.php

```
$rawdata = file_get_contents("php://input");
$daneJSON = json_decode($rawdata, true);
$ok = true;

if ($daneJSON == null) {
    $wynik = array('rez' => false, 'kod' => 1, 'bo' => 'zły format');
    $ok = false;
}

if ($ok) {
    if (isset($daneJSON['id'])) {
        if (isset($daneJSON['opcja'])) {
            $co = (int) round($daneJSON['opcja'] * 1);
            switch ($co) {
                case 0:
                    $wynik = odczytajListe($daneJSON);
                    break;
                case 1:
                    $wynik = pobierzUzytkownika($daneJSON);
                    break;
                case 2:
                    $wynik = usunPlikListy($daneJSON);
                    break;
                case 3:
                    $wynik = zapiszListyUzytkownika($daneJSON);
                    break;
                case 4:
                    $wynik = pobierzSownikProduktow($daneJSON);
                    break;
                case 5:
                    $wynik = zapiszProduktyNaListe($daneJSON);
                    break;
                case 6:
                    $wynik = zapiszListyProduktow($daneJSON);
                    break;
                case 7:
                    $wynik = usunProduktZListy($daneJSON);
                    break;
                default:
                    $wynik = array('rez' => false, 'kod' => 3, 'bo' => 'nieznane polecenie');
            }
        } else {
            $wynik = array('rez' => false, 'kod' => 1, 'bo' => 'brak polecenia');
        }
    } else {

```

```
function zad0($daneJSON)
{
    if (isset($daneJSON['id'])) {
        $id = $daneJSON['id'];
        if (isset($daneJSON['wzor']) && isset($daneJSON['wybor']) && isset($daneJSON['kolor']) && isset($daneJSON['data'])) {
            if (file_exists("zapisplikow5583678dgd786geg/$id")) {
                $plik = json_decode(file_get_contents("zapisplikow5583678dgd786geg/$id"), true);
                if ($plik == null) $plik = array();
            } else {
                $plik = array();
            }

            $post_data = array('wzor' => $daneJSON['wzor'], 'wybor' => $daneJSON['wybor'], 'kolor' => $daneJSON['kolor'], 'data' => $daneJSON['data']);

            file_put_contents("zapisplikow5583678dgd786geg/$id",
                json_encode($post_data, JSON_PRETTY_PRINT + JSON_UNESCAPED_UNICODE + JSON_UNESCAPED_SLASHES)
            );
            return array('rez' => true, 'kod' => 101, 'bo' => 'ok');
        } else {
            return array('rez' => false, 'kod' => 5, 'bo' => 'Brak poprawnych danych!');
        }
    } else {
        return array('rez' => false, 'kod' => 2, 'bo' => 'brak id');
    }
}

function zapiszListyUzytkownika($daneJSON)
{
    $id=$daneJSON['id'];
    if(file_exists("ListyUzytkownikow/$id.txt"))
    {
        $post_data = array('listy' => $daneJSON['lista']);
        file_put_contents("ListyUzytkownikow/$id.txt",json_encode($post_data, JSON_PRETTY_PRINT + JSON_UNESCAPED_UNICODE + JSON_UNESCAPED_SLASHES));
        return array('rez' => true, 'kod' => 101, 'bo' => 'ok');
    }
    else
    {
        fopen("ListyUzytkownikow/$id.txt", "w");
        $post_data = array('listy' => $daneJSON['lista']);
        file_put_contents("ListyUzytkownikow/$id.txt",json_encode($post_data, JSON_PRETTY_PRINT + JSON_UNESCAPED_UNICODE + JSON_UNESCAPED_SLASHES));
        return array('rez' => true, 'kod' => 101, 'bo' => 'ok');
    }
}
}
```

```

function zapiszListyProduktow($daneJSON)
{
    $id=$daneJSON['id'];
    if(file_exists("Produkty/$id.txt"))
    {
        $post_data = array('lista' => $daneJSON['lista']);
        file_put_contents("Produkty/$id.txt",json_encode($post_data, JSON_PRETTY_PRINT + JSON_UNESCAPED_UNICODE + JSON_UNESCAPED_SLASHES));
        return array('rez' => true, 'kod' => 101, 'bo' => 'ok');
    }
    else
    {
        return array('rez'=>false, 'kod'=>5, 'bo'=>'Brak danych dla podanego pliku');
    }
}

function zapiszProduktyNaListe($daneJSON)
{
    $id=$daneJSON['id'];
    if(file_exists("Listy/$id.txt"))
    {
        $post_data = array('wlasosci' => $daneJSON['wlasosci'], 'produkty' => $daneJSON['produkty']);
        file_put_contents("Listy/$id.txt",json_encode($post_data, JSON_PRETTY_PRINT + JSON_UNESCAPED_UNICODE + JSON_UNESCAPED_SLASHES));
        return array('rez' => true, 'kod' => 101, 'bo' => 'ok');
    }
    else
    {
        fopen("Listy/$id.txt", "w");
        $post_data = array('wlasosci' => $daneJSON['wlasosci'], 'produkty' => $daneJSON['produkty']);
        file_put_contents("Listy/$id.txt",json_encode($post_data, JSON_PRETTY_PRINT + JSON_UNESCAPED_UNICODE + JSON_UNESCAPED_SLASHES));
        return array('rez' => true, 'kod' => 101, 'bo' => 'ok');
    }
}

function pobierzSlownikProduktow($daneJSON)
{
    $id=$daneJSON['id'];
    if(file_exists("Produkty/$id.txt"))
    {
        $plik=json_decode(file_get_contents("Produkty/$id.txt"),true);
        return array($plik);
    }
    else
    {
        return array('rez'=>false, 'kod'=>5, 'bo'=>'Brak danych');
    }
}

```

```

function pobierzUzytkownika( $daneJSON )
{
    $id=$daneJSON['id'];
    if(file_exists("ListyUzytkownikow/$id.txt"))
    {
        $plik=json_decode(file_get_contents("ListyUzytkownikow/$id.txt"),true);
        return array($plik);
    }
    else
    {
        fopen("ListyUzytkownikow/$id.txt", "w");
        $post_data = array('listy' => '');
        file_put_contents("ListyUzytkownikow/$id.txt",json_encode($post_data, JSON_PRETTY_PRINT + JSON_UNESCAPED_UNICODE + JSON_UNESCAPED_SLASHES));
        $plik=json_decode(file_get_contents("ListyUzytkownikow/$id.txt"),true);
        return array($plik);
    }
}

function odczytajListe( $daneJSON )
{
    $id=$daneJSON['id'];
    if(file_exists("Listy/$id.txt"))
    {
        $plik=json_decode(file_get_contents("Listy/$id.txt"),true);
        return array($plik);
    }
    else
    {
        return array('rez'=>false, 'kod'=>5, 'bo'=>'brak danych dla id tego');
    }
}

function odczytajPlik( $daneJSON )
{
    $id=$daneJSON['id'];
    if(file_exists("$id"))
    {
        $plik=json_decode(file_get_contents("$id"),true);
        return array($plik);
    }
    else
    {
        return array('rez'=>false, 'kod'=>5, 'bo'=>'brak danych dla id tego');
    }
}

```

5.stylesMainPage.css

6.stylesListPage.css

7.stylesPageOne.css

Podsumowanie

Założenia projektu zostały osiągnięte, program po testowaniu działa poprawnie.