Veb sajt omogućava korisniku da kači recepte za krofne. On može da napravi svoju kategoriju, odnosno ukus, koje dodeljuje svakom receptu. Korisnik može da menja i briše postojeće recepte.

Link ka github-u : <https://github.com/anavucurevic/prvi-domaci-iteh>

# Index.php

Predstavlja početnu stranicu. Na njoj su izlistani svi recepti i omogućuje se korisniku da pretražuje date recepte i filtirira ih po kategorijama ili prema nazivu.

  function ispisi() {

            const kateg = $('#kateg').val();

            const sort = $('#sort').val();

            const imeFilter = $('#nazivFil').val();

            const niz = krofne.filter(element => {

                return (kateg == 0 || element.kategorija\_id == kateg) && element.ime.startsWith(imeFilter)

            })

            niz.sort((a, b) => {

                if (sort == 'ASC')

                    return (a.ime.toLowerCase() > b.ime.toLowerCase()) ? 1 : -1;

                return (a.ime.toLowerCase() > b.ime.toLowerCase()) ? -1 : 1;

            });

U index.php koristimo funkciju ispisi da nam prikaze rezultate za pozivane akcije.

U njoj smo definisali 4 konstante , prve tri se odnose na polja na početku stranice – sortiraj po nazivu, pretraži po nazivu i filtiriraj po kategoriji. Četvrta konstanta se odnosi na niz koji se sastoji od niza krofni i vraća se naziv krofne i id kategorije.

Unutar njega njega nalazi se i algoritam za sortiranje. On se prvo pita da li je sort === asc i kao povratnu vrednost vraća 1 ili -1. Ukoliko je objekat a veci od objekta b uzime se vrednost 1 što znači da se sortira b pre a, a u koliko se uzme vrednost -1 sortiramo prvo a pa onda b.

U slučaju da sort nije jednak asc automatski pse prelazi na drugi return koji vrši istu funkciju stim što se sada prvo vraća -1 pa 1 kako bi dobili abecedu unazad.

      let red = 0;

            let kolona = 0;

            $('#elementi').html(`<div id='row-${red}' class='row mt-2'></div>`)

            for (let krofna of niz) {

                if (kolona === 4 ) {

                    kolona = 0;

                    red++;

                    $('#elementi').append(`<div id='row-${red}' class='row mt-2'></div>`)

                }

                $(`#row-${red}`).append(

                    `

                        <div class='col-3 pt-2 bg-light'>

                            <img src='${krofna.slika}' width='100%' height='320' />

                            <h4 class='text-center'>${krofna.ime}</h4>

                            <h5 class='text-center'>${krofna.kategorija\_naziv}</h5>

                           <a href='./promena.php?id=${krofna.id}'> <button class='form-control btn-success mb-2 dugme-pretraga'>Vidi</button></a>

                        </div>

                    `

                )

            }

        }

Definisali smo dve promenljive red i kolona. For petlja radi tako što proverava da li je broj kolona jendak 4, ako jeste postavlja broj kolona na nulu i povecava red za 1.

# NapraviKorfnu.php

Ova stranica omogućava korisniku da kreira svoju krofnu odnosno svoj novi recept.

Sastoji se iz forme u koju korsnik unosi naziv, kalorije, kategoriju (select), sliku krofne i recept.

Kreiranje nove krofne je post zahtev.

Bitna nam je sledeća linija koda:

 <select id='kat' class="form-control" required name='kategorija\_id'>

                    </select>

Je se id=kat popunjava tako što se iz baze podataka iz tabele kategorija povlače sve dostupne kategorije. Popunjava se putem ajaxa, pošto smo ovde samo inicijalizovali ali nismo popunili.

    <script>

    $(document).ready(function () {

        $.getJSON('./server/kategorija/returnAll.php', function (data) {

            console.log(data);

            if (!data.status) {

                alert(data.error);

                return;

            }

            for (let kateg of data.kolekcija) {

                $('#kat').append(`

                    <option  value='${kateg.id}'> ${kateg.naziv} </option>

                `)

            }

        })

    })

</script>

$(document).ready označava da se prvo inicijalizuju svi html elementi a tek kasnije se nastavlja sa izvršavanjem ove funkcije. Pomoću ajax poziva $.getJSON imamo putanju koja nas vodi do stranice returnAll.php koja će nam vratiti sve kategorije. Drugi parametar ajax poyiva je function(data) koja kada se uzvrši vraća nazad niz svih kategorija. Ona ima svoj property koji je status i odnosi se na to da li je uspešno ili nije i u tom slučaju javlja grešku- ako nema nijedna kategorija. Ako je uspešno prelazimo na for petlju

Za svaku kat koja postoji u kolekciji kategorija, koja je nama prosleđena putem parametra- putem funkcije kao povratna vrednost dodaje se u polje sa tagom „#kat” jednu po jednu kategoriju iz te kolekcije.

Klikom na dugme kreiraj šalje se post zahtev preko koga se odlazi na sledeću putanju server-krofna-kreiraj.php . Post zahtev u sebi sadrži property ime, property kalorije…

  <form action="./server/krofna/kreiraj.php" method="post" enctype="multipart/form-data" size='200'>

# returnAll.php - kategorija

Ova stranica komunicira sa brokerom i unutar nje se nalazi upit koji proseđujemo brokeru kako bi nam vratio sve kategorije iz tabele.

  echo json\_encode($broker->vratiKolekciju('select \* from kategorija'));

ona poziva funkciju vratiKolekciju koja je definisana unutar brokera i njoj proseđuje dati upit.

# kreiraj.php – krofna

Iz kreiranog objekta kojeg smo poslali putem post zahteva se izvlači jedna po jedna vrednost.

$rezultat=$broker->udc("insert into krofna(ime,kalorije,kategorija\_id,slika,recept) values ('".$ime."',".$kalorije.",".$kategorija\_id.",'".$lokacija."','".$recept."') ");

if($rezultat['status']){

    header("Location: ../../napraviKrofnu.php");

}else{

    header("Location: ../../napraviKrofnu.php?greska=".$rezultat['error']);

}

Preko funkcije udc, koju smo definisali u brokeru, ubacujemo gore pomenute vrednosti u bazu. Ukoliko smo uspesno izvršili ubacivanje vraća nas nazad na stranicu napraviKrofnu.php, a ako je došlo do greške izveštava nas o tome.

# Get\_image.php

<?php

if(isset($\_POST['get\_image']))

{

 $url=$\_POST['img\_url'];

 $data = file\_get\_contents($url);

 $new = 'slike/new\_image.jpg';

 file\_put\_contents($new, $data);

// echo "<img src='slike/new\_image.jpg'>";

}

?>

Klikom na dugme DODAJ SLIKU U SVOJ RACUNAR preko post zahteva odalzimo do stranice get\_image.php. Unutar nje u url stavljamo prosledjeni url slike. Preko file-get-contents pretvramo url u string i smestamo ovaj novi string zajedno sa putanjom slike u file-put-contents. Ovaj kod nam omogućava da sačuvamo sliku sa linka u našoj memoriji i korisitmo je prilikom pavljenja nove krofne.

# Get\_thubmail.php

<?php

if(isset($\_POST['get\_thumbnail']))

{

 $url=$\_POST['url'];

 $fetch=explode("v=", $url);

 $videoid=$fetch[1];

 echo '<img src="http://img.youtube.com/vi/'.$videoid.'/0.jpg" width="250"/>';

}

?>

Klikom na dugme PREUZMITE SLIKU SA YOUTUBE VIDEA preko post zahtev odlazimo do stranice get-thumbnail.php. Funkcija explode prima separator I url. Ona seče url pomoću datog separatora I smešta isečene stringove u niz fetch. U promenljivu videoed smeštamo iz niza fetch smeštam string sa indexom broj 1. I na ekranu pomoći echo prikazujemo datu sliku.

# Kategorije.php

Ova stranica daje korisniku mogućnost da sam kreira kateogoriju, tako što unese njen naziv, a nakon toga se uneta kategorija ispisuje. Unošenje kategorija predstavlja post zahtev.

 <table class='table table-light display'>

                    <thead>

                        <tr>

                            <th>ID</th>

                            <th>Naziv kategorije</th>

                        </tr>

                    </thead>

                    <tbody id='elementi'>

                    </tbody>

                </table>

Sadrži tabelu sa poljima id i naziv kategorije i popunjava se putem ajaxa. Kada se klikne na dugme napravi poziva se post zahtev.

  <form class='mb-5' action="./server/kategorija/kreiraj.php" method="post">

                    <label>Naziv kategorije</label>

                    <input type="text" name='naziv' class='form-control'>

                    <label class='text-danger bg-light' <?php echo (!isset($\_GET['greska']))?'hidden':''; ?> ><?php echo $\_GET['greska']; ?></label>

                    <button type='submit' class='form-control btn btn-primary mt-2 dugme-pretraga'>Napravi</button>

Što nas opet vodi na stranicu kreiraj.php unutar foldera kategorija i ona dodaje prosleđene vrednosti u bazu tako što ih šalje brokeru zajedno sa odgovarajućim upitom.

Tabela se puni tako što mi učitavamo sve kategorije koje se nalaze u bazi.

  <script>

        $(document).ready(function () {

            $.getJSON('./server/kategorija/returnAll.php', function (data) {

                console.log(data);

                if (!data.status) {

                    alert(data.error);

                    return;

                }

                $('#elementi').html('');

                for (let kateg of data.kolekcija) {

                    $('#elementi').append(`

                        <tr>

                            <td>${kateg.id}</td>

                            <td>${kateg.naziv}</td>

                            <td>

                                <button class='form-control btn btn-danger' onClick=obrisi(${kateg.id}) >Obrisi</button>

                            </td>

                        </tr>

                    `)

                }

            })

        })

        function obrisi(id\_kategorije) {

            $.post('./server/kategorija/delete.php', { id: id\_kategorije }, function (data) {

                if (data.status=='false') {

                    alert(data.error)

                } else {

                    window.location.reload();

                }

            })

        }

    </script>

Ovde se sve kategorije ubacuju jedna po jedna u #elementi gde dodajemo id i naziv kategorije.

U svakom redu nalazi se i dugme obriši koje preko onClick poziva funkciju za brisanje . Funkcija za brisanje prima 1 parametar – id. Ona je takođe post zahtev i vodi do stranice delete.php unutar foldera kategorija

# Delete.php – kategorija

echo json\_encode($broker->udc('delete from kategorija where id='.$id));

Koristi funkciju udc iz brokera kojoj prosleđuje upit kojim briše iz tabele kategorija sve one kategorije čiji id odgovara prosleđenom.

# 4.Broker.php

Broker služi kao konekcija sa bazom.

    private function \_\_construct(){

        $this->mysqli = new mysqli("localhost","root","","krofne");

        $this->mysqli->set\_charset("utf8");

    }

    public static function getBroker(){

        if(!isset($broker)){

            $broker=new Broker();

        }

        return $broker;

    }

Napravili smo konstruktor unutar koga smo preko mysqli se povezali sa našom bazom – prosledili smo mu hostname- localhost, korisničko ime-root, lozinku – koja je prazna i naziv baze.

Funkcija getBroker inicijalizuje broker ako nije inicijalizovan u suprotnom nam vraća brokera.

Unutar njega sam definisala dve funkcije :

    public function vratiKolekciju($upit){

        $rezultat=$this->mysqli->query($upit);

        $response=[];

        if(!$rezultat){

            $response['status']=false;

            $response['error']=$this->mysqli->error;

        }

        else{

            $response['status']=true;

            while($red=$rezultat->fetch\_object()){

                $response['kolekcija'][]=$red;

            }

        }

        return $response;

    }

Ova funkcija izvršava prosleđeni upit, ona je funkcija za select. Ona inicijalzuje praznu kolekciju koju popunjavamo pomoću fetch\_object. On učitava jedan po jedan red iz baze i ubacuje ga u kolekciju rezultat. Na kraju dobijeni rezultat vraća kao povratnu vrednost.

    public function udc($upit){

        $rezultat=$this->mysqli->query($upit);

        $response=[];

        $response['status']=(!$rezultat)?false:true;

        if(!$rezultat){

            $response['error']=$this->mysqli->error;

        }

        return $response;

    }

Funkcija udc služi kao funkcija za update, delete i insert. Ona takođe izvršava prosleđeni uput i inicijalizuje praznu kolekciju. Kao povratnu vrednost vraća kolekciju response.