

Popis hlavních rysů programu:

Web aplikace je založená na relační databázi.

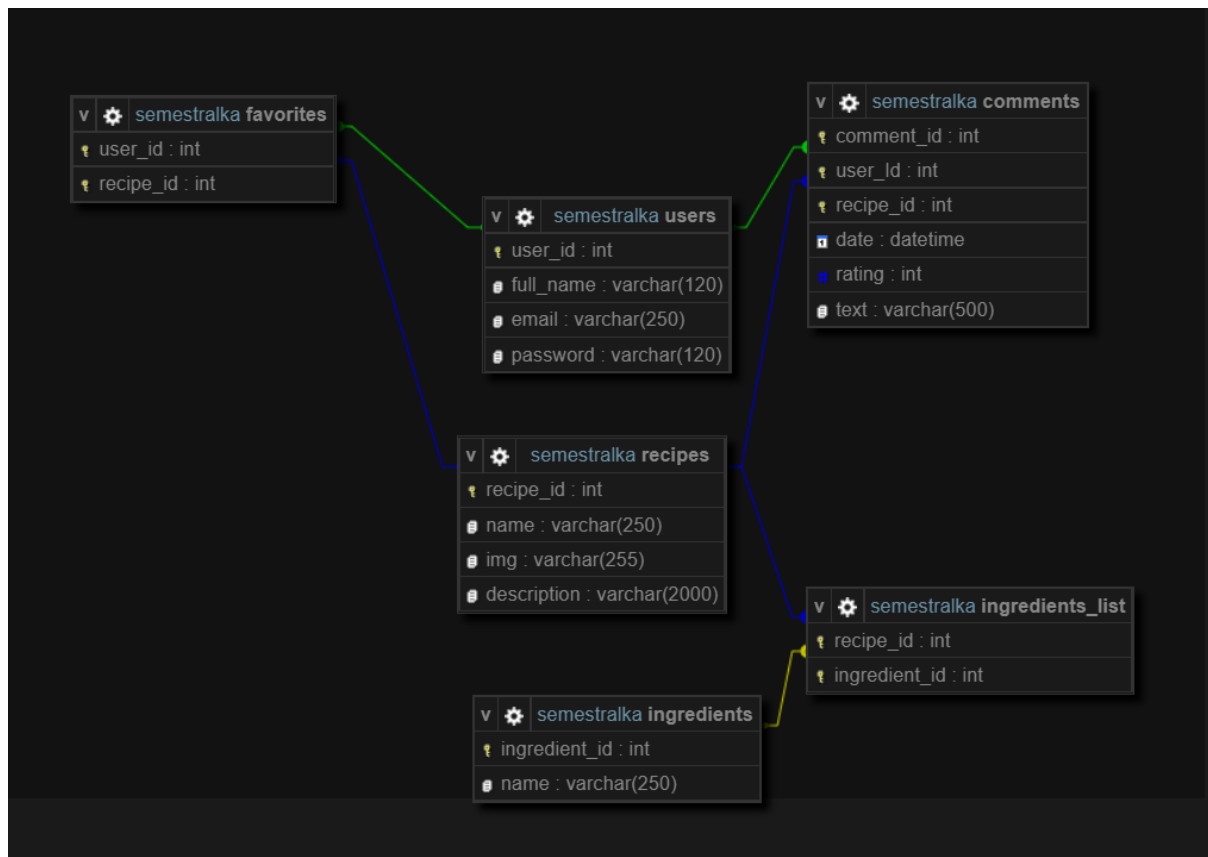
V db jsou celkem 6 tabulek.

Čtyři tabulky s entity 'users', 'recipes', 'comments', 'ingredients'.

V db jsou dvě spojovací tabulky, které zobrazují vztahy.

Tabulka 'ingredients_list' - vztah mezi ingrediencí a receptem, je-li on součástí tohoto receptu.

Tabulka 'favorites' - vztah mezi uživatelem a receptem, je-li recept je v oblíbených u uživatele.



Hledání receptů je založeno na checkboxech s metodou GET. Checkboxy s ingrediencí jsou generovány serverem na základě query

```
$sql = 'SELECT * FROM ingredients';
```

```
$stm = $db->query($sql);
```

```
$ingredients = $stm->fetchAll();
```

Pak používám foreach \$ingredients a tvořím z toho checkboxy.

Dále uživatel může si vybrat ingredience, ve values kterých je id ingredience z db. Taký lze vybrat sorting buď žádný nebo po počtu komentářů. Po submit do serveru se odesílá informace o typu sortingu a ingredience metodou GET. Pokud byl zvolen typ sortingu, query se změní, pokud ne, zobrazí jen všichni odpovídající výsledky z databáze. Pak s pomocí joinů

```
$sql = 'SELECT * FROM recipes ';
```

```
$sql .= $ingredients_query;
```

```
$sql .= 'LIMIT :limit OFFSET :offset';
```

\$ingredients_query může být i prázdná, pokud uživatel nezmačkne ani jeden checkbox. Pak ukážeme všechny recepty, které jsme dostali z databázi, pomocí foreach a vytvoříme pro každý unikátní link.

```
a href="<?="/recipe.php?recipe_id=' . $recipe['recipe_id'] ;?>">
```

Pokud nic se nenajde, nebo hodnoty byli nesmyslné, vznikne nadpis. 'Oops, we didn't find anything'.

Přidání receptu. Popsání receptu a název jsou omezené shora a zdola. Hodnoty id ingredience jsou validované tak, aby neprošli id které nejsou v tabulce ingredients. Pokud uživatel vložil nějaký soubor začíná se validace: velikost je omezená, typ souboru je omezen jen na jpeg, png, a jpg. Pokud validace souboru byla úspěšná, on bude nahrán na server do složky uploads. Cesta do obrázku bude uložena do proměnné a pak předaná do fce postNewRecipe.

Pokud validace všech insertů byla úspěšná, spustí se funkce postNewRecipe, která pomocí INSERT přidá nový recept to tabulky recipes, pak pokud uživatel vybral nějaké ingredience, které patří tomuto receptu, udělá INSERT ingredience a id nového receptu do tabulky ingredient_list. Takovým způsobem bude lze najít tento recept pomocí ingredience. Pokud uživatel nezadal žádný obrázek, nový recept bude mít obrázek "emptyImage".

Přidání komentáře. Při přidání komentářů automatické se přidává aktuální datum. Hodnoty ratingu jsou omezené jen na 1 až 5, text komentáře je omezen shora a zdola. Id uživatele, který píše komentář se bere z proměnné SESSION, id receptu z aktuálního adresu web stránky \$_GET['recipe_id']?.

Zabezpečení proti XSS

Díky použití prepared statements, moje web apka je zachráněna před sql injections. Taky kde bylo vhodné jsem používala metodu htmlspecialchars. Po každém submitu formy zadávám redirect, aby zajistit odolnost proti dvojímu odeslání stejného formuláře.

AJAX

Ajax používám dvakrát:

Pro přidání receptu do záložek:

Spuštění JS scriptu to kliknutí (onclick=JSscript()) na ikonku. Pak v té funkci je ajax kód, url ve kterém ukazuje na php script, a odesílá do serveru metodou POST user id a id receptu. Pak tento php script buď přidává do tabulky favorites nový řádek nebo zmaže ho. Pak pokud to bylo úspěšně ikonka srdíčka se změní díky tomu, že se přidá nebo se odebere class "is favorite". Díky tomu, že já na to používám ajax, ikonka srdíčka se mění okamžitě po kliknutí a není potřeba obnovit stránku.

Pro změnu stylu web stránky:

Spuštění JS scriptu to kliknutí (onclick=JSscript()) na ikonku. Php script buď přidá cookie, která má v sebe informaci, že uživatel kliknul na ikonku dark mode (měsíc), nebo jestli ona už byla přidána tak ji smaže. V headeru mám dva linky na dva různé stylesheets, jestli

cookies dark mode nebyla nastavena, tak přidá běžný stylesheet, jestli byla, tak přidá darkmode.css.