

Имена: Владимир Пеков

Дата: 02.07.2017г.

e-mail: vladdey@gmail.com

GitHub: <https://github.com/vlpekov/Recipes>

Предмет: JAVA WEB

Рецептурник

сайт за рецепти

1. Условие

Рецептите ще имат следните характеристики:

- време за приготвяне;
- степен на трудност;
- вид (местна, вегетарианска, веган)
- брой порции и др.

Ще могат да се показват рецепти по различни признаци. Пример:

Покажи ми рецепти: бързи, средна степен на трудност, вегетариански;

За използването на сайта не е необходимо задължително вписване на потребител и регистрация, но при този вариант ще има ограничени функционалности.

Потребители: Потребителите могат да публикуват свои рецепти, както и да коментират/оценяват рецепти на други потребители.

Регистрация на потребители. Паролите се записват в hash-нат вид (Salted Password Hashing) в базата данни.

При влизане на потребител в сайта паролата се записва отново в hash-нат вид в cookie, както и името на потребителя. Истинската парола няма да се запазват никъде. След излизане (log out) cookie-тата се заличават. При извършване на действие изискващо вписан потребител (публикуване на рецепта, коментиране и т.н.), всеки път ще се прави проверка на записаните кукита, дали името на потребителя и съответната му парола в кукита съответстват, на тази в базата данни. Ако съответстват - операцията се извършва. В противен случай се изтриват кукитата и се изисква наново log in.

2. Въведение

Web-приложението е реализирано с технологиите Java Servlets и Java Server Pages (JSP) изпълняващи се на сървър за Web-приложения Tomcat. За подобрене работата на сайта са използвани JavaScript и JQuery. За визуализация и позициониране на съдържание е използван CSS.

3. Теория

Алгоритъм е написан на език за програмиране Java EE, HTML, CSS, JavaScript, SQL заявки към MySQL база данни, реализиран с помощта на среда за разработка

EclipseEE – специализирана среда за разработване на Java EE софтуер и уеб приложения, която включва интегрирана среда за разработка (IDE) и плъгин система.

За разработването на проекта е използван Apache Tomcat уеб сървър за Java уеб приложения. Той представлява контейнер за Java уеб приложенията, който осигурява среда, в която да се изпълняват те.

Необходимите данни и информационни ресурси необходими за работа на сайта се съхраняват в релационна база данни. За целта е използван MySQL - SQL система за управление на бази данни.

За автоматизирано сглобяване (build) на проекта се използва приложение Gradle , чиято цел е да създаде war (Web application Archive).

4. Използвани технологии

При стартиране на сайта се показва начална страница, от която потребителя позволява на потребителя да ползва съдържанието на сайта. При изискването на даден ресурс се изпращат get или post заявки към java servlet-и, където заявките биват обработвани и връщат резултат на jsp страници, и след изпълнението им се генерира HTML, чрез който съдържанието се визуализира от браузера. Текстовата и графична информация на рецептите се записват и извличат от базата данни. Връзката с базата данни се осъществява посредством SQL заявки.

5. Инсталация и настройки

Страницата се стартира на сървър. За разработването и демонстрирането на проекта се използва локален сървър, благодарение на Apache Tomcat. След стартирането на сървъра, страницата е достъпна от браузера на локален адрес: <http://localhost:8080/Recipes/>

Необходимо е да има инсталиран и на MySQL. Базата данни ще се генерира автоматично, като единственото необходимо нещо е, потребителят да смени името и паролата (и ако е необходимо други настройки) класа *SetupDB.class*, намиращ се в *com.database.utilities*.

6. Кратко ръководство на потребителя

Страницата е пригодена да се използва, както от регистрирани така и о не регистрирани потребителите. Като нерегистрираните имат ограничени възможности. За разлика от нерегистрираните потребители, регистрираните могат да добавят рецепти, като в бъдеще е предвидено да се развият множество нови модули, като потребителите да могат да сготвят рецепти на друг потребител (ще се отбелязва като сготвена), ще публикуват снимки, ще пишат забележки (по количествата, операциите и т.н.), възможности за вариации и др. Така ще бъде изградена рейтингова система на потребителите.

Регистриран потребител може да се впише от началния екран. Чрез формата за вписване намираща се в горния десен ъгъл на страницата. Потребителите, които нямат регистрация могат да направят това избирайки линка „Регистрация“ под формата за вписване на регистрирани потребители.



Фиг.1 Форма за вписване на потребителите.

След вписване в горния десен ъгъл се показва снимка на потребителят с приветствие съдържащо неговото потребителско име. Потребителското име е линк, който води до разглеждане на профила на потребителя. Появява се също и навигационно меню, което към момента съдържат само три бутона, но за сметка на това само „Публикувай рецепта“ води до нова функционалност. „Моите рецепти“ отваря потребителския профил, на който отдолу има списък с рецептите публикувани от потребителя, а „Сготви рецепта“ е празен линк. (Фиг.2)

Фиг. 2 Обекти, който стават видими при вписване на потребител.

В основната част на страницата с показан списък с рецептите, по 12 рецепти на страница. Избирането на, рецепта води до отваряне на пълната информация за рецептата. Над списъка има поле за търсене и бързо сортиране на рецептите (фиг.3)



по признаците: „местни“ (рецепти съдържащи месо), вегетариански (рецепти без месо), веган (рецепти не съдържащи никакви животински деривати).

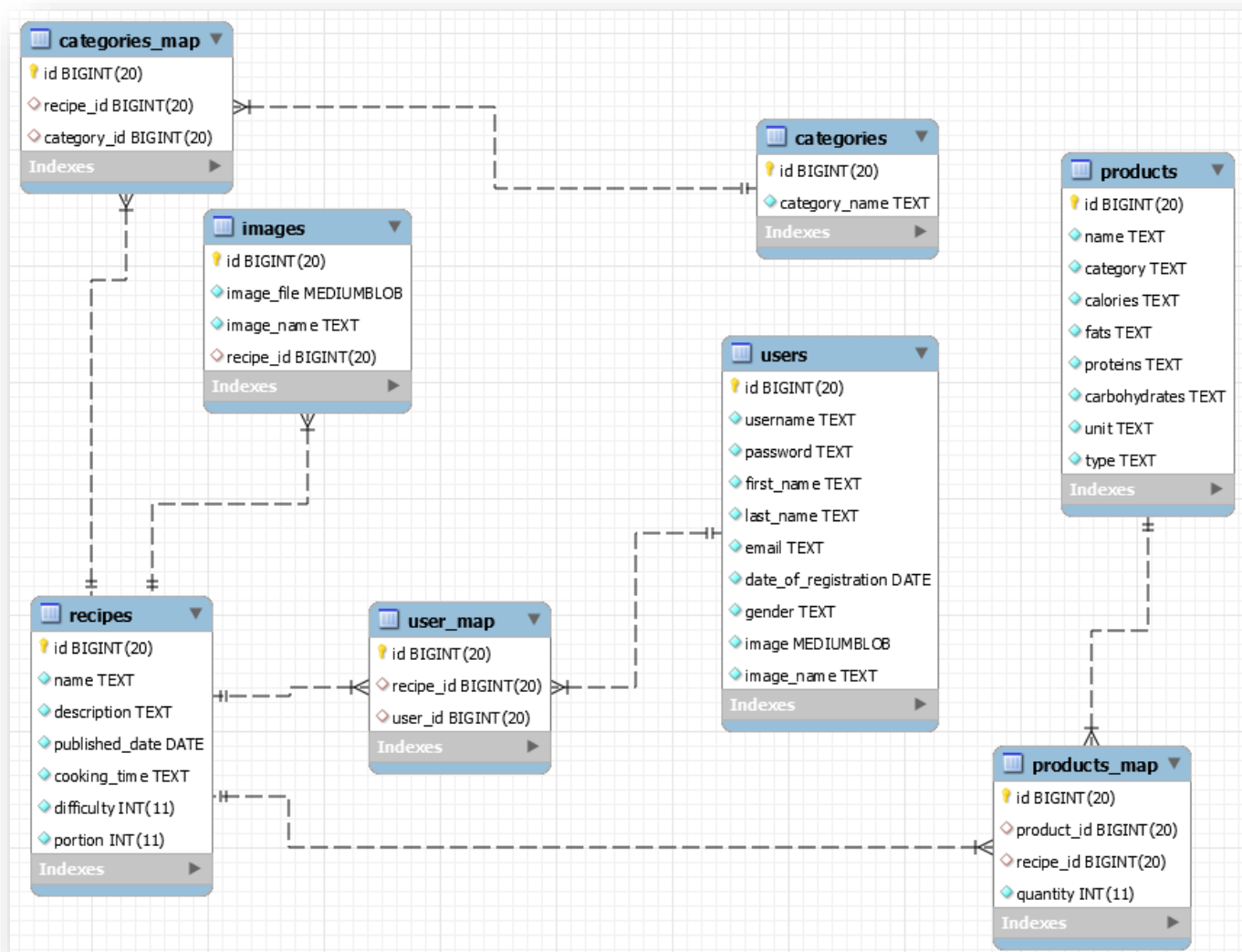
Покажи рецепти:
 ☒ всички
 ☐ месни
 ☐ вегетариански
 ☐ веган

Фиг. 3 Поле за търсене и показване на рецептите по признак.

Публикуването на нова рецепта става като от навигацията се избере „Публикувай рецепта“. Отваря се страница с полета за попълване, като условие е всички ад са попълнени. Изборът на продукти към списъка с продукти става след избор на продукт и въвеждане на количеството, и натискане на бутона “Добави”.

7. Описание на програмния код

База данни. Към настоящия момент базата данни се състои от 8 таблици (фиг.4), като *recipes*, *images*, *users*, *categories*, *products* са основните таблици, съдържащи информация за основните модели, а останалите три: *user_map*, *categories_map*,



products_map, служат за определяне на връзките между самите модели. Схемата на архитектурата на базата е показана на фиг.

Фиг. 4 Архитектура на БД

Кодът на сайта е разпределен в пакети по: класове за опериране с БД, сървлети, интерфейси, JavaBeans, филтри, модели, jsp файлове, css, JS.

com.database.utilities	
ConnectorDB.java	- сингълтън клас за връзка с БД
CreateDB.java	- създава БД, ако не съществува
CreateTables.java	- създава таблиците в БД
FillTables.java	- попълва таблиците в БД
LoginCheck.java	- сравнява записаната hash-ната парола с тази от Cookies
SetupDB.java	- сингълтън клас настройване и връзка с БД
UserManager.java	- взима паролата на конкретен потребител

Фиг.5 Класове за работа с БД

com.javabeans	
UserRecipesList.java	- създава списък с рецепти на потребител
RecipesListGenerator.java	- създава списък с рецепти
FormRecipeSearchNames.java	- имена на таговете recipesList и searchList.jsp
IndexTagsNames.java	- имена на таговете в index.jsp
FormRegistration.java	- имена на таговете в register.jsp
RecipeTagsNames.java	- имена на таговете в recipeView.jsp
UserProfileTags.java	- имена на таговете в userProfile.jsp
CookiesManager.java	- работа с Cookies
FormNewRecipe.java	- имена на таговете в newRecipe.jsp

Фиг.6 JavaBeans

com.vso.models	
Recipe.java	- модел за рецепта
Product.java	- модел за продукт
User.java	- модел за потребител
Password.java	- модел за парола

Фиг.7 Модели

com.vso.controllers	
AutocompleteProducts.java	- връща списък с всички продукти към newRecipe.jsp
GetProfilePicture.java	- взима снимката на потребителя от БД
GetRecipe.java	- взима рецепта от БД за recipeView.jsp
GetRecipePicture.java	- взима снимката на рецептата от БД
GetUserProfile.java	- извличане на потребител от БД за userProfile.jsp
Login.java	- вписване на потребител
Logout.java	- отписване на потребител
RecipeSave.java	- запис на нова рецепта в БД
RegistrationValidation.java	- извършване на регистрация на нов потребител
SearchRecipes.java	- резултати от търсене на рецепта по име
ShowByCategory.java	- резултати на рецептите по категория
UploadImage.java	- качване на изображение в БД

Фиг.8 Сървлети

com.vso.interfaces	
Cookies.java	- имена и параметри на Cookies
DatabaseNames.java	- име на БД и таблиците в нея
FormNamesNewRecipe.java	- имена на тагове в форма за нова рецепта
FormNamesRegistration.java	- имена на тагове в форма за регистрация на потребител
FormSearchRecipe.java	- имена на тагове в форма за търсене на рецепта
IndexTags.java	- имена на таговете в index.jsp
RecipeTags.java	- имена на таговете в recipeView.jsp
TableCategories.java	- имена на колони в таблица от БД
TableCategoriesMapNames.java	- имена на колони в таблица от БД
TableImages.java	- имена на колони в таблица от БД
TableProductsMapNames.java	- имена на колони в таблица от БД
TableProductsNames.java	- имена на колони в таблица от БД
TableRecipesNames.java	- имена на колони в таблица от БД
TablesColumnNames.java	- наследява всички останали интерфейси с имена таблици
TableUserMap.java	- имена на колони в таблица от БД
TableUsersNames.java	- имена на колони в таблица от БД
UserProfileTagsNames.java	- имена на таговете в userProfile.jsp

Фиг.9 Интерфейси

com.vso.filters	
CharsetFilter.java	- филтър за безпроблемна кирилица

Фиг. 10 Филтър за кирилицата

JSP файлове	
register.jsp	- страница за регистрация на потребител
userProfile.jsp	- страница за разглеждане на потребителски профил
recipeView.jsp	- страница за преглед на рецепта
newRecipe.jsp	- страница за нова рецепта
login.jsp	- страница за вписване на потребител
userRecipesList.jsp	- списък с рецептите на потребител
recipesList.jsp	- списък с рецепти
searchList.jsp	- списък с резултати от търсене
profilePicture.jsp	- показване на потребителска снимка и име

Фиг. 11 JSP файлове

JS и JQuery	
jquery-1.11.2.js	използван за Autocomplete в newRecipe.jsp
jquery-ui.js	използван за Autocomplete в newRecipe.jsp
jquery-ui.min.js	използван за Autocomplete в newRecipe.jsp
newInput.js	използван за добавяне в списък с продукти в newRecipe.jsp

Фиг.12 JS JQuery файлове

8. Приноси на курсиста, ограничения и възможности за бъдещо разширение.

Сайтът е на базово ниво от където може да бъде развит значително, като например:

- да могат да се избират рецепти по определени основни продукти;
- да могат да се избират рецепти, които не съдържат дадени продукти;
- рецептите ще може да се преизчисли за различен брой порции т.е. ще се променят количествата на продуктите.
- да могат да се конвертират мерните единици на продуктите (*например 1 чаша брашно – в 130г. брашно*)
- потребителите да могат да сготвят рецепти на друг потребител (ще се отбелязва като сготвена), ще публикуват снимки, ще пишат забележки (по количествата, операциите и т.н.), възможности за вариации и др.
- рейтингова система на потребителите.

9. Използвани източници:

1. [The Java™ Tutorials](#)
2. [Stack Overflow](#)
3. [Java Singleton Design Pattern Best Practices with Examples](#)
4. [Login application using jsp servlet and mysql database](#)
5. ["Remember Me" in Spring Security Example](#)
6. [JSP Cookies Handling](#)
7. [Servlets - Cookies Handling](#)
8. [Upload & Retrieve Image JSP,Servlet,DataBase](#)
9. [Stack Overflow: How to upload files to server using JSP/Servlet?](#)
10. [Stack Overflow: How to upload an image and save it in database?](#)
11. [Show and hide multiple div using javascript](#)
12. [Stack Overflow: JSP tricks to make templating easier?](#)
13. [Stack Overflow: How can i hash a password in java](#)
14. [JSP - Include Directive](#)
15. [Page layout css no tables frames](#)
16. [Center and crop images with a single line of CSS](#)