

Recipe Generator Purpose

Introduction to the service

WHAT TO COOK FOR DINNER?

With the recipe generator, you no longer have to make a decision for a long time. With only a few added ingredients needed.

Audience

Portrait of the target audience: female/male age - over 20 years; income - average job - with a high probability sedentary, sedentary; needs: due to a busy schedule, there is no time to eat properly, there is no time to make a menu.

Constraints

The number of recipes from the selected ingredients.

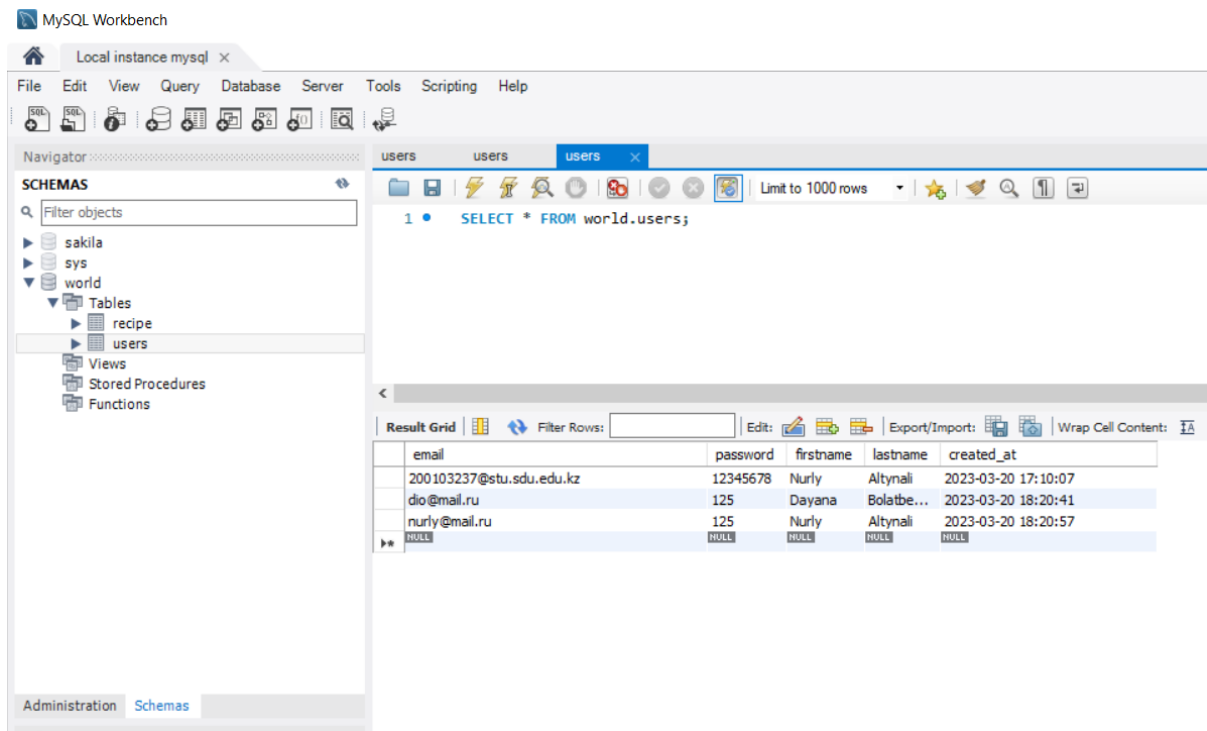
Report:

- Created a database and tables
- Connect to MySQL database
- Registration (data of users are to be sent to database)
- The list of recipes from the database is displayed;
- Authorization(data is checked from database)
- Search for recipes by ingredients
- Frontend pages

How to run:

You should have mysql db, creating tables with columns

Example:



The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays a tree view of databases: sakila, sys, and world. The 'world' database is expanded, showing tables (recipe, users), views, stored procedures, and functions. The 'users' table is selected. The main query editor shows the following SQL query:

```
1 • SELECT * FROM world.users;
```

The 'Result Grid' pane at the bottom displays the results of the query. The table has five columns: email, password, firstname, lastname, and created_at. The results are as follows:

email	password	firstname	lastname	created_at
200103237@stu.sdu.edu.kz	12345678	Nurly	Altynali	2023-03-20 17:10:07
dio@mail.ru	125	Dayana	Bolatbe...	2023-03-20 18:20:41
nurly@mail.ru	125	Nurly	Altynali	2023-03-20 18:20:57
NULL	NULL	NULL	NULL	NULL

In terminal - > go run main.go

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

[GIN-debug] POST /index/more --> main.handlerIndexMore (3 handlers)
PS D:\Go project\Golang> go run main.go
[GIN-debug] [WARNING] Creating an Engine instance with the Logger and Recovery middleware already attached.

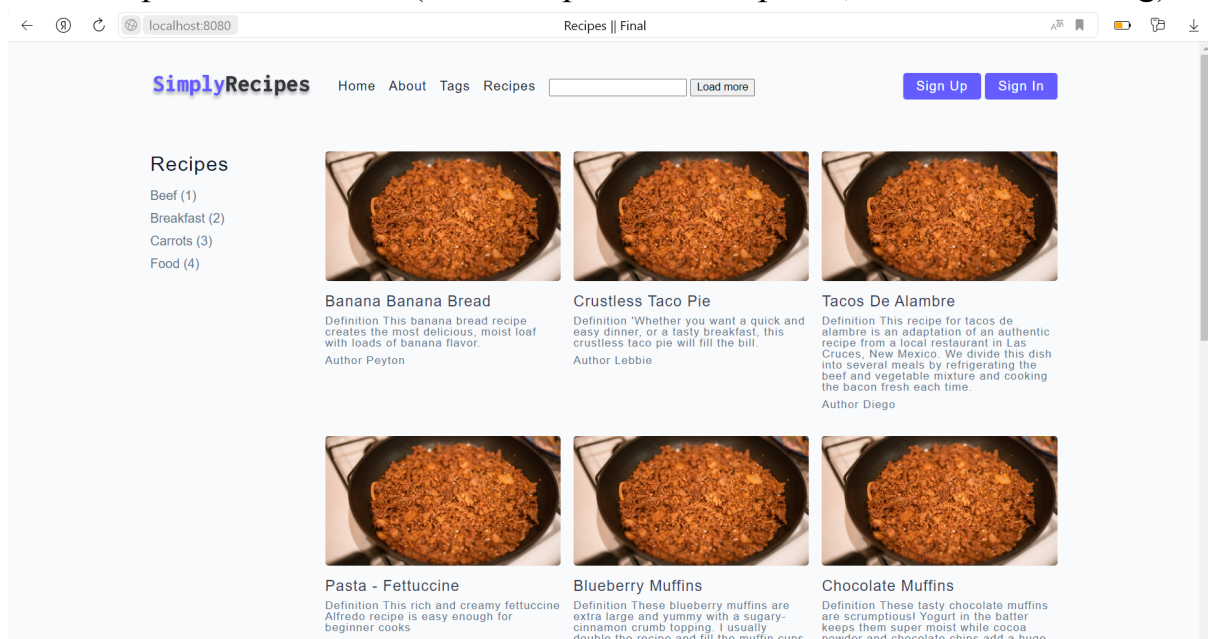
[GIN-debug] [WARNING] Running in "debug" mode. Switch to "release" mode in production.
- using env: export GIN_MODE=release
- using code: gin.SetMode(gin.ReleaseMode)

[GIN-debug] GET /assets/*filepath --> github.com/gin-gonic/gin.(*RouterGroup).createStaticHandler.
func1 (3 handlers)
[GIN-debug] HEAD /assets/*filepath --> github.com/gin-gonic/gin.(*RouterGroup).createStaticHandler.
func1 (3 handlers)
[GIN-debug] Loaded HTML Templates (8):
- registration.html
-
- 400.html
- authorization.html
- index.html
- layout.html
- loggedin.html
- profile.html

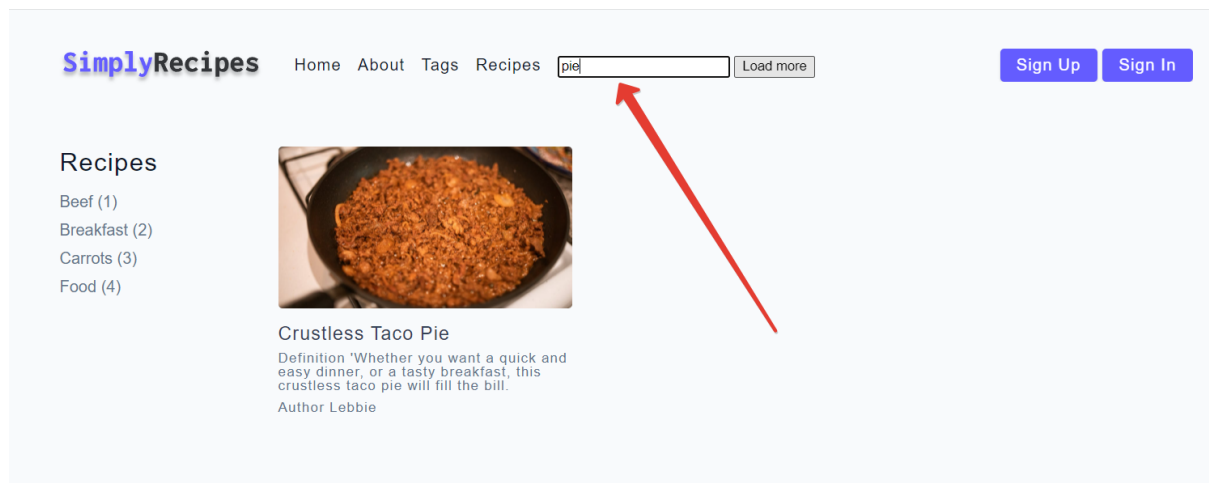
[GIN-debug] GET / --> main.handlerIndex (3 handlers)
[GIN-debug] GET /index --> main.handlerIndex (3 handlers)
[GIN-debug] POST /index/more --> main.handlerIndexMore (3 handlers)
[GIN-debug] GET /index/more --> main.handleSearch (3 handlers)
[GIN-debug] GET /registration --> main.handlerRegistration (3 handlers)
[GIN-debug] GET /authorization --> main.handlerAuthorization (3 handlers)
[GIN-debug] POST /user/reg --> main.handlerUserRegistration (3 handlers)
[GIN-debug] POST /user/auth --> main.handlerUserAuthorization (3 handlers)
[GIN-debug] GET /user/profile --> main.profileHandler (4 handlers)
[GIN-debug] [WARNING] You trusted all proxies, this is NOT safe. We recommend you to set a value.
Please check https://pkg.go.dev/github.com/gin-gonic/gin#readme-don-t-trust-all-proxies for details.
[GIN-debug] Listening and serving HTTP on :8080
[GIN] 2023/03/20 - 23:09:46 | 200 | 32.6805ms | ::1 | GET | "/"
```

Main page

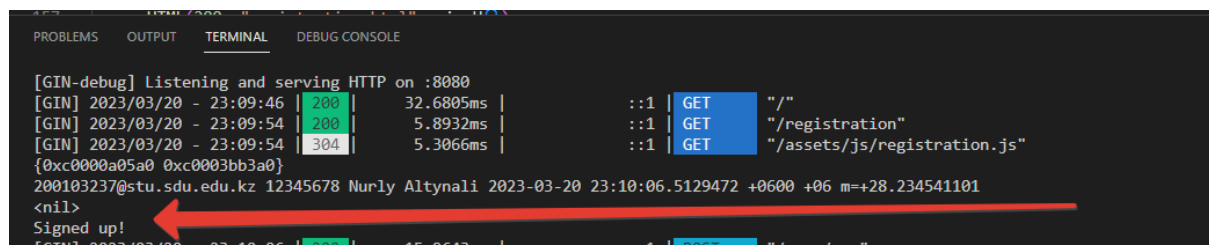
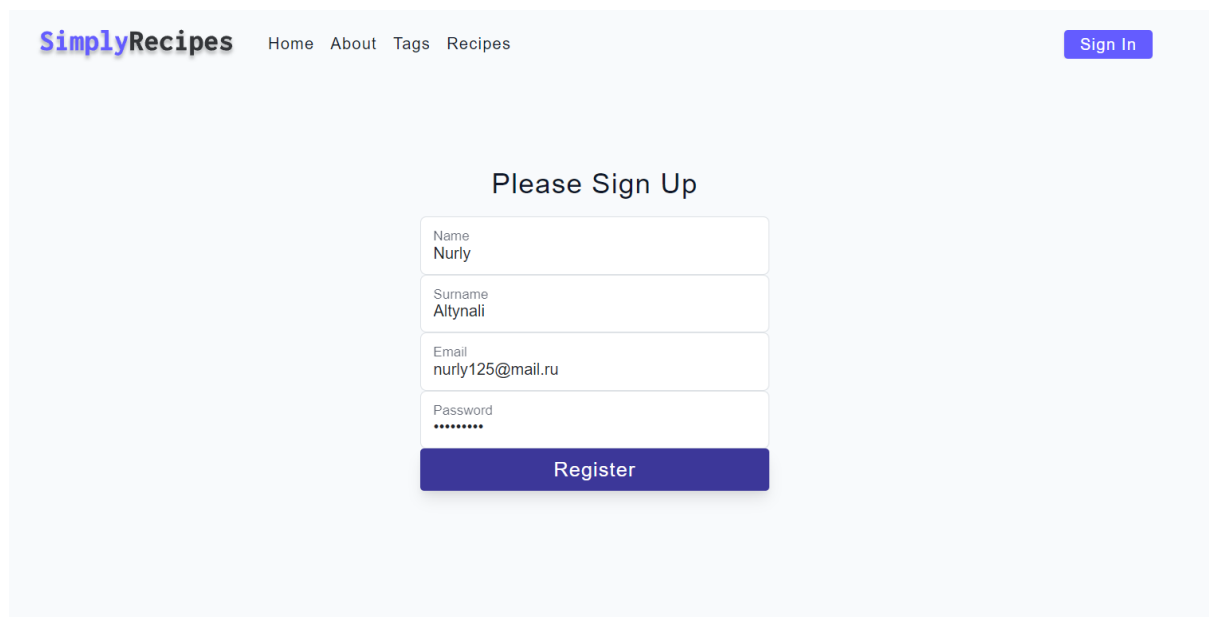
Here recipes from database(we will update it with photo, now db has not img)



You can search the recipes by names:



You can sign up:



We see the result in terminal

You can sign in:

SimplyRecipes

[Home](#) [About](#) [Tags](#) [Recipes](#)

Sign Up

Please Log In

Email
nurl125@mail.ru

Password
••••••••

9

Sign In

[illegible]