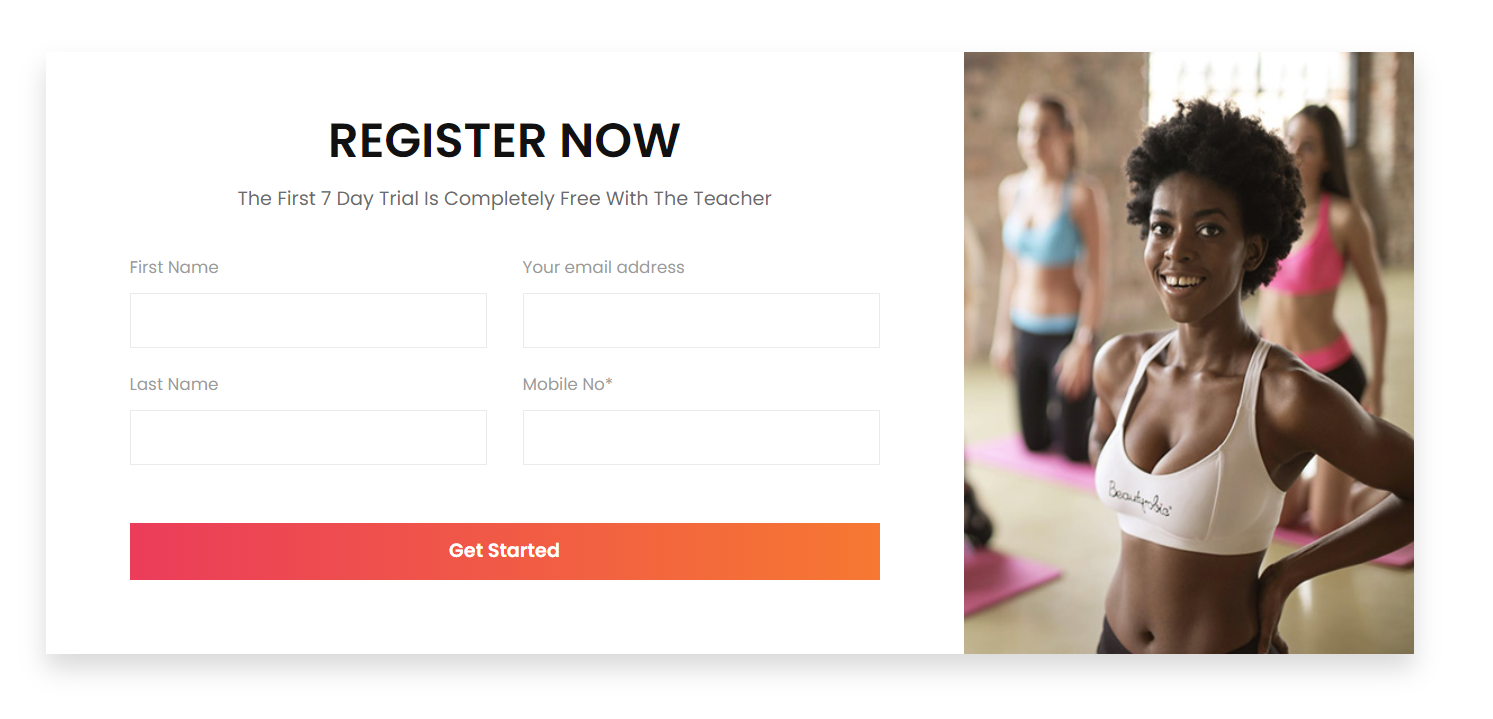
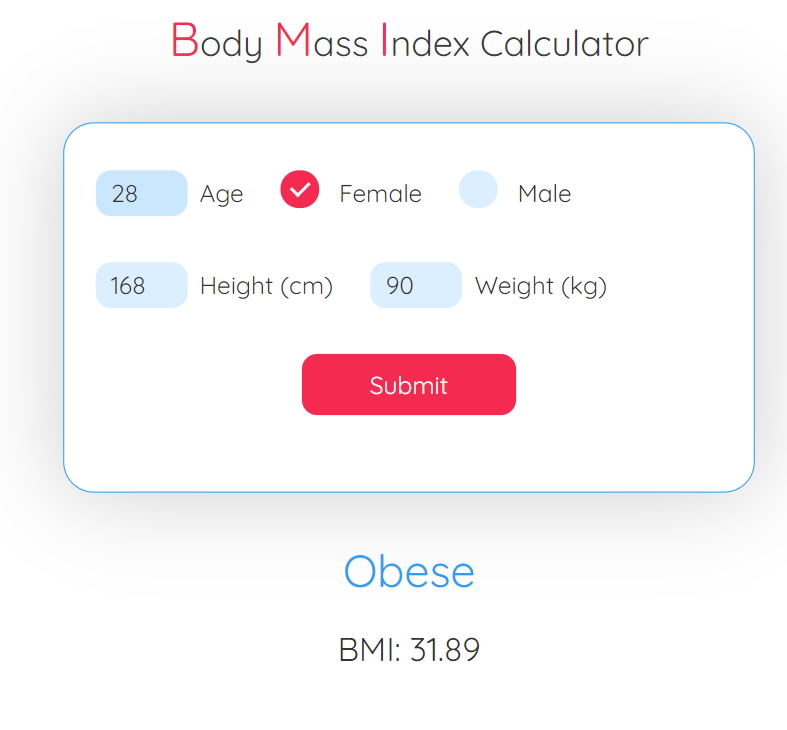
**T**he best diet and weight loss app.   
– Personal plan to lose weight or build muscles.  
– Calorie table with lots of foods.  
– Nutrition tracker and food diary for all meals.  
– Tracks your calories, carbs, proteins and fats.

1. Personne va s’inscrire, lors de son inscription elle va également rentrer les données afin qu’on calcule son BMI.
2. Les infos du BMI seront sauvegardés dans une DB.





1. Un calcul du métabolisme de base se fera également en fonction des données entrées (basal metabolic rate) 🡺 besoins énergétiques de l’organisme. Cad dépense d’énergie minimum quotidienne permettant à l’organisme de survivre.
2. Il y aura un planning dans lequel la personne pourra rajouter : (la personne pourra marquer l’heure exacte à laquelle elle mange et pourra le faire autant qu’elle veut)

- des aliments au fur et à mesure qu’elle mange

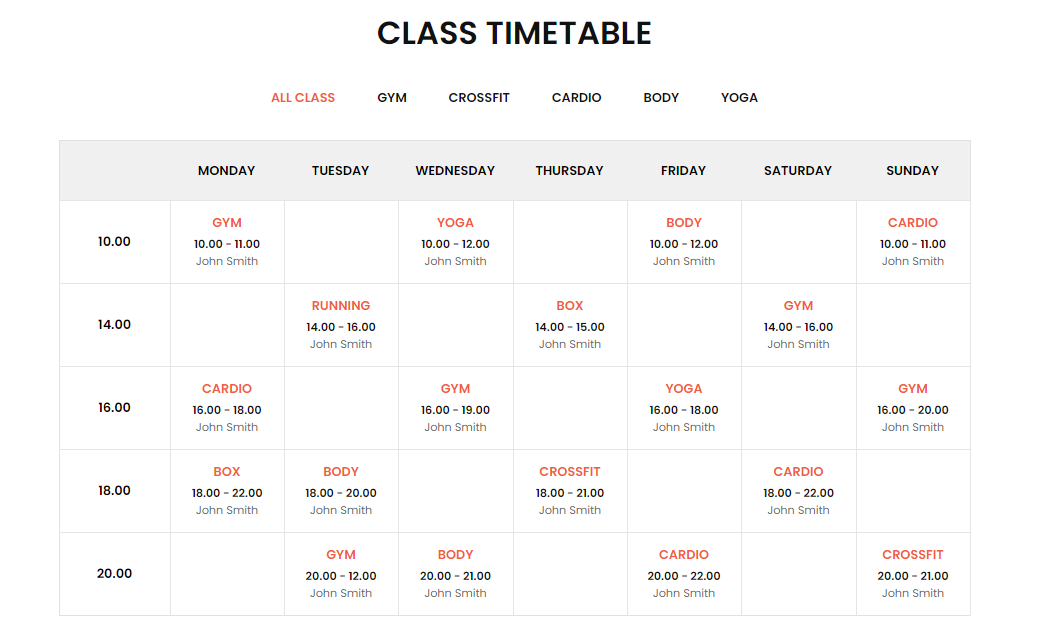
- eau qu’elle boit

Il y aura une liste d’aliments de base déjà présents dans la DB & la personne pourra y rajouter ses propres aliments.

5. Sauvegarde de la consommation quotidienne de la personne.

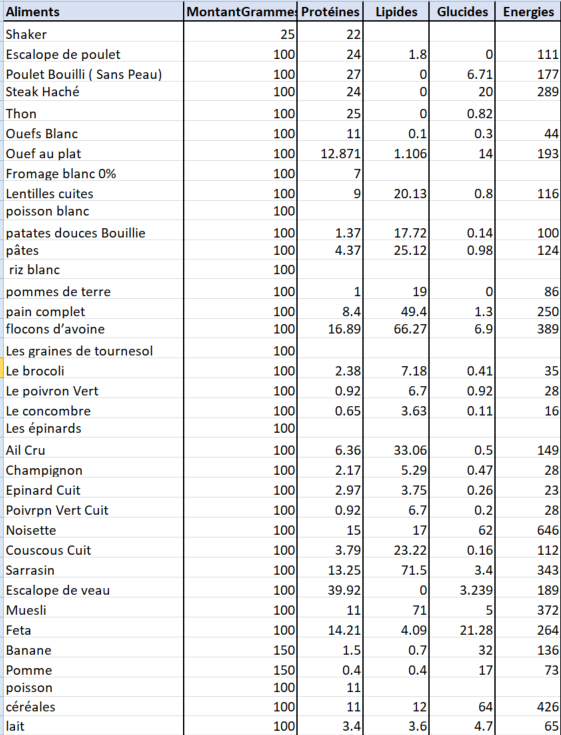
6. Objectifs fixés par la personne

EXTRA : si toutes les 2h j’dois manger, envoyer un rappel.



***DONNEES DE LA DB :***

Voici la liste d’aliments qui seront déjà présents dans la DB.

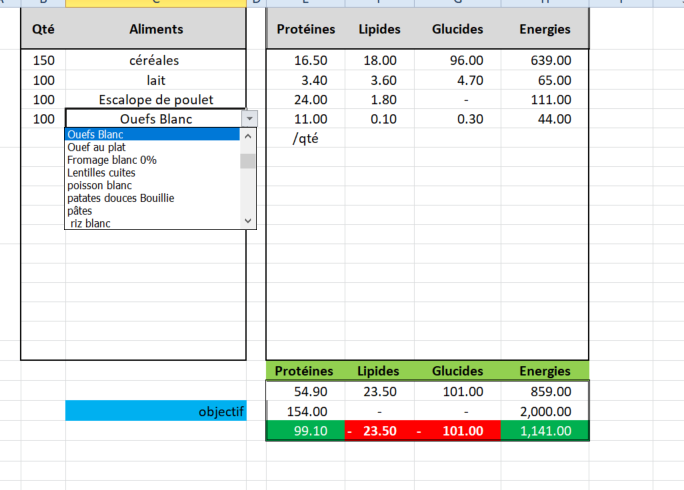


A gauche il y aura un form avec le choix des aliments pour la consommation quotidienne.

A droite, les protéines,lipides , glucides & l’énergie seront calculés en fonction du choix de la personne (grammes/aliment) 🡺 SI PAS DANS DB , alors la personne entrera toutes ces données pour l’aliment qu’elle rajoute.

La personne se fixera un OBJECTIF de calories à atteindre & le calcul se fera en soustrayant ce qui a été fixé comme objectif de ce qui a été consommé.

Côté gauche : manque DATE , l’heure



1. Formulaire où personne va choisir une date, aliment(liste déployable de la DB),
2. Champ du formulaire de ce qu’il a mangé , l’heure à laquelle il a mangé, un autre champ du formulaire : montant
3. Diagramme de classe : utilisateur 🡪1 planning (1-1), un planning a des infos qu’on rentre, une entrée 🡺 l’id généré automatiquement, nom(string) de ce qu’il mange, quantité, le datetime quand il mange (entity type), planning lié one to many entries(entité entry)
4. Entité :

* Planning/user (1-1)
* Planning a plein d’entrées(une autre entité nommée ENTRIES) one to many
* Planning il y aura 4 objectifs (id , protéines, lipides, glucides, énergies). Un planning a des entry(one to many). Un planning a un seul objectif
* Entry many to many avec les aliments(many to one vers detailentry), one to many de détailentry to aliment
* entry (date)/planning (many-1)
* entry/aliments (many to many) (entrydétail one to many, many to one detail entry vers aliments)

<?php

namespace App\Controller;

use App\Entity\User;

use App\Form\RegistrationFormType;

use Doctrine\Persistence\ObjectManager;

use Symfony\Component\HttpFoundation\Request;

use Symfony\Component\HttpFoundation\Response;

use Symfony\Component\Routing\Annotation\Route;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;

use Symfony\Component\Security\Core\Encoder\UserPasswordEncoderInterface;

class SecurityController extends AbstractController

{

    /\*\*

     \* @Route("/inscription", name="security\_registration")

     \*/

    public function registration(Request $request, ObjectManager $manager){

        $user = new User();

        $form = $this->createForm(RegistrationFormType::class, $user);

        $form->handleRequest($request);

        if($form->isSubmitted() && $form->isValid())

        {

            $manager->persist($user);

            $manager->flush();

        }

        return $this->render('security/registration.html.twig', [

            'form' => $form->createView()

        ]);

    }

}

 {% extends 'base.html.twig' %}

{# {% block body %}  #}

    {% block content %}

    <section class="register-section spad" >

        <div class="container">

            <div class="row">

                <div class="col-lg-8">

                    <div class="register-text">

                        <div class="section-title" id="register">

                            <h2>Register Now</h2>

                            <p>The First 7 Day Trial Is Completely Free With The Teacher</p>

                        </div>

                        {{ form\_start(form) }}

                        {{ form\_row(form.email, { 'label': 'Email adress', 'attr': {'placeholder': 'Email adress'}}) }}

                        {{ form\_row(form.password, { 'label': 'Password', 'attr': {'placeholder': 'Password'}}) }}

                        {{ form\_row(form.confirm\_password, { 'label': 'Confirm password', 'attr': {'placeholder': 'Confirm password'}}) }}

                        {{ form\_row(form.name, { 'label': 'Name', 'attr': {'placeholder': 'Name'}}) }}

                        {{ form\_row(form.surname, { 'label': 'Surname', 'attr': {'placeholder': 'Surname'}}) }}

                        {{ form\_row(form.sexe, { 'label': 'Sexe', 'attr': {'placeholder': 'Sexe'}}) }}

                        {{ form\_row(form.height, { 'label': 'Height', 'attr': {'placeholder': 'Height'}}) }}

                        {{ form\_row(form.weight, { 'label': 'Weight', 'attr': {'placeholder': 'Weight'}}) }}

                        <button type="submit" class="register-btn">Get Started</button>

                        {{ form\_end(form) }}

                    </div>

                </div>

                <div class="col-lg-4">

                    <div class="register-pic">

                        <img src="assets/img/register-pic.jpg" alt="">

                    </div>

                </div>

            </div>

        </div>

    </section>

    <!-- Register Section End -->

    {% endblock %}

{# {% endblock %}  #}

{{ '&'|escape }}