

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики й обчислювальної техніки
Кафедра обчислювальної техніки

Розрахунково-графічна робота
з предмету «Інтеграція програмних систем»
за темою: «Система контролю харчування кота»

Виконали:
Студенти 4 курсу
ФІОТ, група ІО-43
Книш Владислав
Цимбал Денис
Міщенко Людмила

Перевірів:
Мазур Р. Ф.

Київ, 2017

Короткий опис проекту

У даній роботі розроблено програму – систему контролю харчування кота. Створена програма дозволяє слідкувати за правильним харчуванням домашнього улюбленця, збалансовано та вчасно його годувати.

За допомогою цієї програми користувач може ввести дані про свого кота, а саме його стать, вік, вагу, спосіб життя (активний, пасивний), смакові пріоритети. Кожного дня можна вводити дані про сьогоднішню активність тварини та її настрій. Також користувач має змогу ввести список продуктів чи кормів, які в нього є зараз вдома .

Система збирає, оброблює та аналізує дані. На основі чого пропонує раціон харчування домашньої тварини на сьогоднішній день та імовірні меню на подальші декілька днів. Іще програма нагадує хазяїну які саме корисні речовини давно не вживав кіт, для того, щоб користувач неодмінно придбав їх і почастував домашнього улюбленця.

Опис збірки Webpack

У проекті використовується збірщик Webpack для збірки модулів Vue. Він приймає цілий ряд файлів, розглядаючи кожен як модуль, виявляючи залежності між ними та об'єднуючи їх у статичні, готові до розгортання.

Опис безперервної інтеграції

Для безперервної інтеграції створено тести для перевірки роботи навігації на стороні клієнта та відображення інформації.

```
Vue app
  ✓ should render correct contents
PhantomJS 2.1.1 (Linux 0.0.0): Executed 1 of 1 SUCCESS (0.037 secs / 0.012 secs)
TOTAL: 1 SUCCESS
```

Експоненціальна витримка

Експоненціальна витримка створена в компоненті App та використано для запиту при реєстрації на сервері. На графіку відображено залежність від номеру невдалого запиту та інтервалу затримки

```
function(router,url,data,options,max,delay,callback) {  
  let t = this;  
  router.post(url,data,options).then(response =>{  
    callback(response);  
  },err =>{  
    if (max > 0) {  
      console.log('Problem with connect');  
      setTimeout(function() {  
        t.exponentialBackoff(router,url,data,options, --max, delay * 2, callback);  
      }, delay);  
    }else {  
      console.err('we give up');  
    }  
  })  
}
```

