

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМ. І. СІКОРСЬКОГО»  
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ  
КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

## **Розрахунково-графічна робота**

з інтеграційних програмних систем

Виконали:  
студенти IV курсу  
групи ІО-42 ФІОТ

Жабокрицький Ігор  
Кохан Олена  
Махров Антон  
Чайковський Олександр

**Київ – 2017**

### **Короткий опис проекту**

---

В рамках лабораторної роботи було написано невеликий сервер для чата. Під словом «чат» можна усвідомлювати усю систему миттєвого обміну повідомленнями. Від форумів та

іншого веб — спілкування чати розрізняються тим, що обмін повідомленнями відбувається в режимі теперішнього часу. Проект відкритий і, доступний як git-репозиторій на GitHub за посиланням: <https://github.com/dream-team-kpi/rgr> . Для розробки було використано nodejs з використанням мов розмітки HTML та CSS. Вибір саме таких технологій пов'язаний з можливістю простого та ефективного способу візуалізації даних, а також з можливістю запуску програми моделювання на будь-якій операційній системі з підтримкою сучасних браузерів. В якості бази даних було використано MongoDB, оскільки це швидка і масштабована система, що не потребує опису таблиць, підтримує зберігання документів в JSON-подібному форматі, має досить гнучку мову для формування запитів. Встановлення даного проекту просте; конфігурації не потрібно.

Було налаштовано автоматичну збірку проекту за допомогою інтеграції GitHub та Travis. У результаті, для кожного нового коміта, який з'являється на github репозиторії, наш проект збирається на сервері Travis'a і запускаються автоматичні тести. У програмному коді проекту реалізовано механізм експоненціальної витримки.

## Вимоги до ПЗ

---

Для запуску проекту на вашому комп'ютері має бути встановлене таке програмне забезпечення:

- Git. Можна завантажити з <https://git-scm.com/downloads>.
- node.js та npm. Можна завантажити з <https://nodejs.org/download>
- Docker. Можна завантажити з <https://docs.docker.com/engine/installation/>
- mongoDB. Можна завантажити з <https://www.mongodb.com/download-center>

## Установка

---

Склонуйте репозиторій за допомогою команди:

```
git clone https://github.com/dream-team-kpi/rgr.git
```

Переконайтеся, що демон mongodb запущений і працює.

Перейдіть в папку з проектом:

```
cd ./rgr
```

Встановіть залежності проекту:

```
npm install
```

Запустіть сервер:

```
npm start
```

Перейдіть в браузер і введіть в адресну строку :

```
http://localhost:3000/
```

## Запуск за допомогою Docker

---

Переконайтеся, що Docker запущений і працює, а демон mongodb не запущений.

Виконайте наступні команди в корені директорії проекту:

```
docker-compose pull
docker-compose build
docker-compose up
```

Перейдіть в браузер і введіть в адресну строку :

`http://localhost:3000/`

## Запуск тестів

---

Переконайтеся, що демон mongodb запущений і працює.

Запустіть сервер у фоновому режимі:

```
npm start &
```

Виконайте тести:

```
npm test
```

Всі тести повинні бути пройдені.

## Опис системи збірки, що використовується у проекті

---

В якості системи збірки для нашого проекту ми використали Travis CI - розповсюджений веб-сервіс для збірки та тестування програмного забезпечення, що використовує GitHub як хостинг вихідного коду.

## Перелік та опис задач, які виконуються на сервері безперервної інтеграції

---

На сервері безперервної інтеграції виконуються такі задачі:

- Збірка проекту
- Встановлення залежностей проекту
- Запуск тестів:
  - Should successfully authorize new user
  - Should fail to authorize new user
  - Should broadcast user connection to already connected user
  - Should broadcast user disconnection to already connected user
  - Should broadcast user message to already connected user

## Графік експоненціальної витримки

---

Графік ілюструє вибрані інтервали для повтору спроб при експоненціальній витримці. На осі X - номер спроби, на осі Y - вибраний інтервал. Ми отримали щось схоже на експоненціальну криву .