

Міністерство освіти і науки України Національний технічний університет України
«Київський політехнічний інститут ім. І. Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

РОЗРАХУНКОВА ГРАФІЧНА РОБОТА
з курсу «Інтеграційні програмні системи»

Виконали студенти 4-го курсу
групи ІО-34
Богдан Олександр
Герасименко Гліб
Сивобородько Сергій
Устимчик Назарій

Команда: «nonames»

Київ 2016

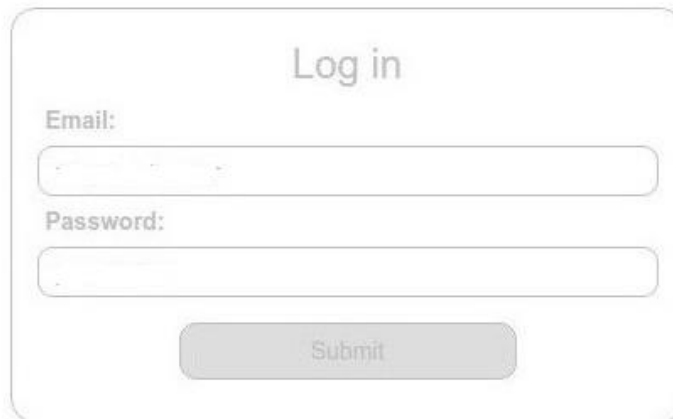
1. Опис проекту

Simple Mail Sender – це система з простим інтерфейсом, яка призначена для відправки повідомлень на електронну пошту.

Система надає наступні можливості користувачу:

- Реєстрація;
- Авторизація;
- Відправка повідомлень;

При переході на головну сторінку, користувачу буде запропоновано авторизуватися в системі.



The login form is titled "Log in" and is enclosed in a rounded rectangle. It contains two input fields: "Email:" and "Password:". Below these fields is a "Submit" button.

Мал. 1 – форма авторизації.

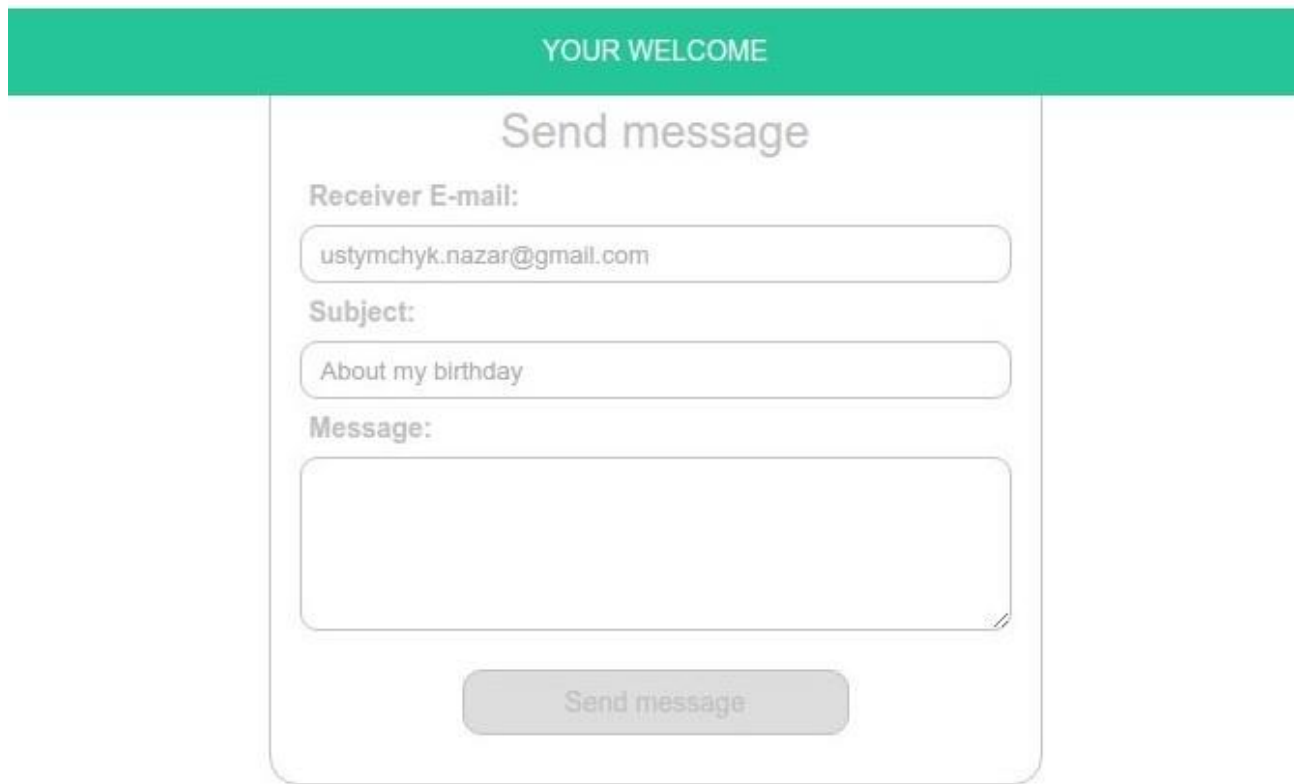
Якщо користувач зареєстрований в системі і авторизація пройшла успішно, то юзера буде перенаправлено на сторінку відправки електронної пошти. У випадку, якщо користувача немає в системі, його буде перенаправлено на сторінку реєстрації.



The registration form is titled "Registration" and is enclosed in a rounded rectangle. It features an orange banner at the top with the text "YOU ARE NOT REGISTERED, PLEASE REGISTERED NOW". The form contains several input fields: "First Name:" (with the value "Nazar"), "Last Name:" (with the value "Ustymchyk"), "Email:" (with the value "ustymchyk.nazar@gmail.com"), "Username:" (with the value "ustymchyk.nazar"), and "Password:" (with the value "placeholder"). A "Submit" button is located at the bottom of the form.

Мал. 2 – форма реєстрації

На сторінці відправки повідомлень, користувач заповнює поля «Адрес отримувача», «Тема листа», вводить текст повідомлення та натискає на кнопку «Відправити повідомлення». Після вдалої відправки листа, користувач отримує відповідне сповіщення в браузері.



The image shows a web form titled "Send message" within a teal header that says "YOUR WELCOME". The form has three input fields: "Receiver E-mail:" with the value "ustymchyk.nazar@gmail.com", "Subject:" with the value "About my birthday", and "Message:" which is empty. Below the fields is a "Send message" button.

Мал. 3 – форма відправки повідомлення

2. Система автоматичної збірки

Travis CI - розподілений веб-сервіс для побудови та тестування програмного забезпечення, що використовує GitHub як хостинг вихідного коду.

Веб-сервіс підтримує збірку проектів на багатьох мовах, включно з C, C++, JavaScript, Java, PHP, Python і Ruby. На Travis виконуються наступні задачі:

- Білдінг контейнера Vagga і запуск тестів на ньому
- Білдінг контейнера Дослук і запуск тестів на ньому
- При успішному проходженні тестів, відбувається автоматичне розгортання проекту на сервері

Існують 4 види тестів, які виконуються на сервері CI:

- Статичний аналіз коду
- Перевірка коду без залежностей
- З використанням моків
- Інтеграційні тести

3. Перелік та опис задач, які виконуються на сервері безперервної інтеграції

Для кожного нового коміта виконуються наступні дії:

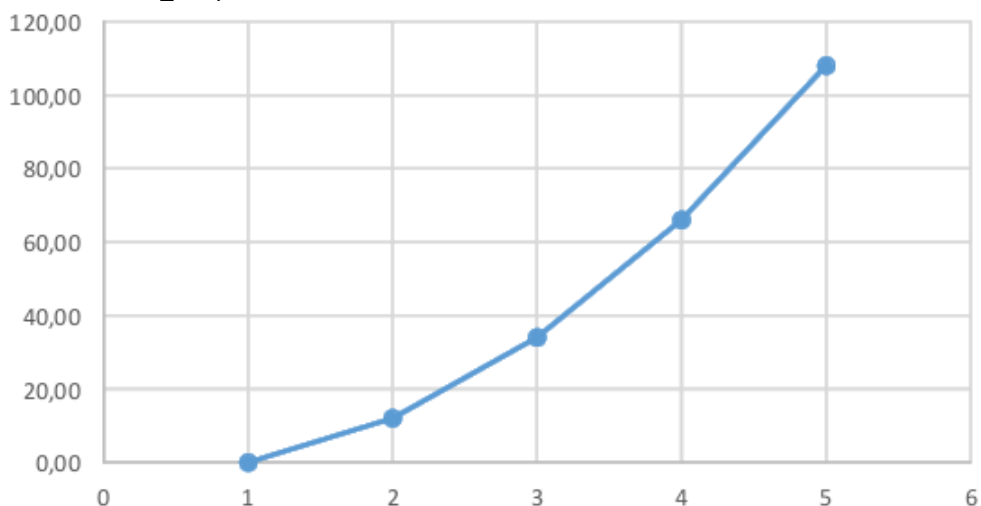
- 1) Збірка проекту
- 2) Запуск автоматичних тестів. Якщо хоча б один із тестів не проходить, збірка помічається як неуспішна.
- 3) Створення контейнера Docker. Виконується створення та запуск образу контейнера із додатком. Якщо хоча б одна з цих операцій не виконується успішно, збірка також помічається як неуспішна.

4. Експоненційна витримка

У даному проекті механізм експоненційної витримки був реалізований не серверній частині, задля забезпечення повторного підключення до бази даних, у разі виникнення збою в останній. В разі отримання статусу 200 ОК, джерело ігнорується до наступного запуску.

Для отримання часових даних був використаний наступний код:

```
def retry(ExceptionToCheck, tries=4, delay=3, backoff=2, logger=None):
    def deco_retry(f):
        @wraps(f)
        def f_retry(*args, **kwargs):
            mtries, mdelay = tries, delay
            while mtries > 1:
                try:
                    return f(*args, **kwargs)
                except ExceptionToCheck as e:
                    msg = "%s, Retrying in %d seconds..." % (str(e), mdelay)
                    if logger:
                        logger.warning(msg)
                    else:
                        print(msg)
                    time.sleep(mdelay)
                    mtries -= 1
                    mdelay *= backoff
            return f(*args, **kwargs)
        return f_retry # true decorator
    return deco_retry
```



Мал. 4 – графік експоненційної витримки