

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

Розрахункова графічна робота
З дисципліни «Інтеграційні програмні системи»

Виконали:
Студенти 4 курсу ФІОТ
Групи ІО-51
Ярошенко Олександр
Сташенко Марія
Суліменко Микита

Київ 2018

1. Короткий опис проекту

Даний проект представляє собою платформу для бронювання робочого місця в закладі (кафе, коворкінгу).

Цей проект наразі має функціонал для відображення місць, доступних для бронювання та форму для оформлення броні. Після оформлення броні менеджер зв'язується з замовником і підтверджує бронювання.

Сторінка зі списком місць:



Форма бронювання місця:

The screenshot shows the 'Book Place' form within the app. It includes input fields for 'Your Name' and 'Your contacts'. The 'Booking Start' and 'Booking Finish' sections feature date pickers (yyyy-mm-dd) and time selectors (HH and MM) with up/down arrows. A 'Rating' section shows a row of seven stars. A 'Book' button is located at the bottom left of the form area.

2. Система збірки, що використовується у проекті

В даному проекті використовується .net компілятор, який переводить програмний код c# в інструкції та метадані IL, які вже виконує Common Language Runtime. IDE Visual Studio за замовчуванням має всі необхідні інструменти для збірки та публікації проектів для .net core. Так, у нашому випадку, весь програмний код за допомогою системи збірки перетворюється на єдиний файл .dll.

Для збірки Angular проекту використовується система збірки webpack, яка теж за замовчуванням наявна в Angular CLI і спрощує збірку проекту. Збірка проекту з папки /dist береться для публікації на сервері і містить лише мініфіковані версії скриптів та стилів.

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

Завантаження залежностей та необхідних nuget пакетів для проекту

- dotnet restore

Створення бази даних для тестування

- psql -c "CREATE DATABASE RGRDatabase;" -U postgres

Перехід в директорию проекту RGRProject

- cd RGRProject

Запуск міграцій БД

- dotnet ef database update

Перехід в кореневу директорию

- cd ..

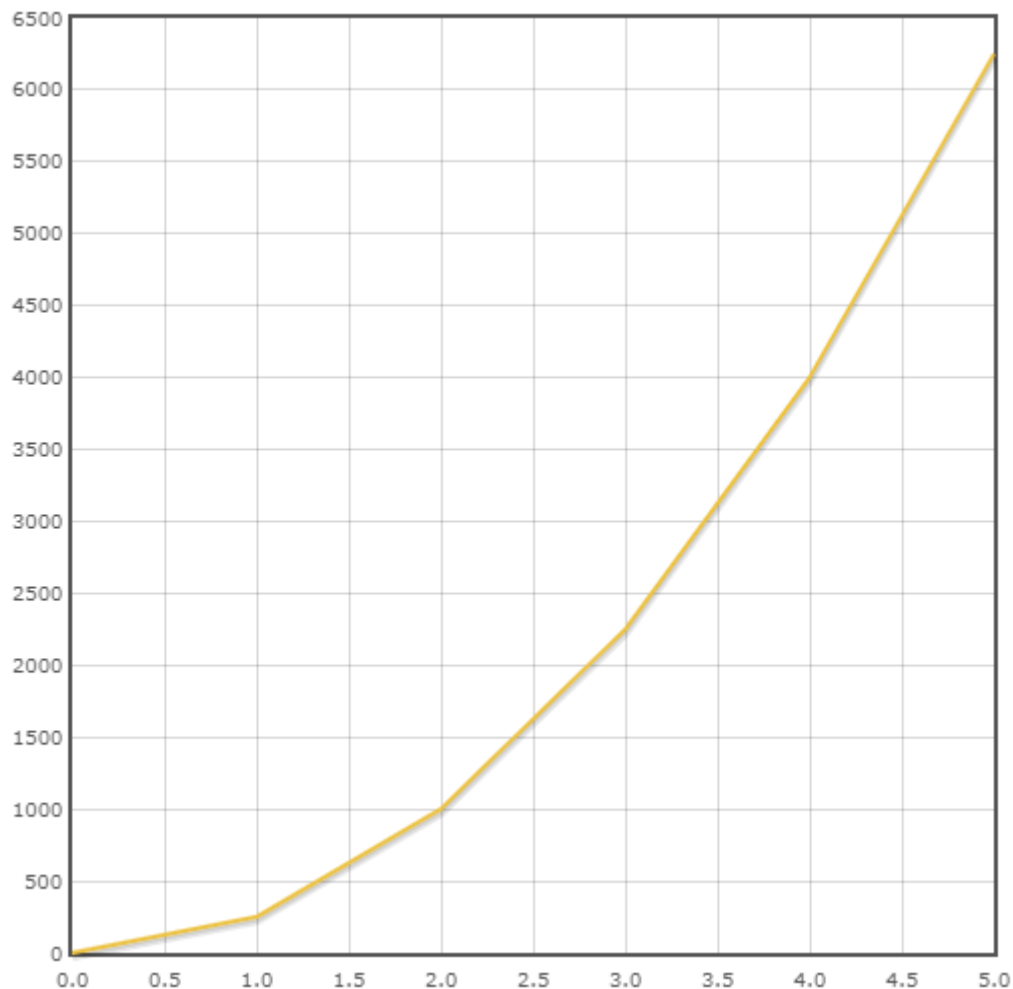
Запуск інтеграційних тестів

- dotnet test ./XUnitTests

4. Графік, який ілюструє вибрані інтервали для повтору спроб при експоненціальній витримці.

По вісі x – номер спроби

По вісі y – час у мс



Випадок з помилкою Сервера

- Сервер спочатку функціонує правильно.
- Потім на сервері стається помилка.
- Клієнт виконує спроби повтору.
- Коли Сервер стає доступним, то система функціонує знову правильно.

Код, що відповідає за експоненційну витримку:

```
backoff(maxTries, ms) {  
  return pipe(  
    retryWhen(attempts => range(1, maxTries))  
      .pipe(  
        zip(attempts, (i) => i),  
        map(i => i * i),  
        mergeMap(i => timer(i * ms))  
      )  
    )  
  )  
  )  
};
```

Вкладка Network у випадку запиту GetPlaces і помилки з сервером

