

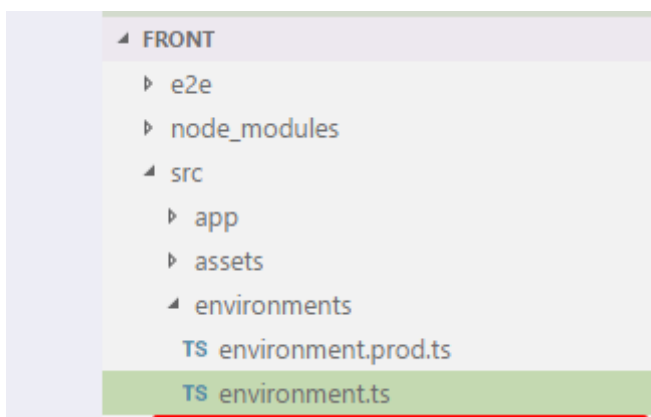
Лабораторна робота №4

Тема:

Написання веб-застосунку використовуючи Angular 7 + Node.js, Express.js та MongoDB.

Налаштування середовища розробки

- 1) Створюємо директорию **lab4** у якій знаходитимуться вихідні коди лабораторної роботи.
- 2) У директорії **lab4** створюємо директорії **front** та **back** для вихідних кодів фронтенду та бекенду відповідно.
- 3) Клонувати репозиторій
<https://github.com/vitaliipolovyi/web-part2/tree/master>
та у директорию **lab4** скопіювати код із **Labs/L4** у відповідні директорії фронтенду та бекенду.
- 4) Виконати команду **'npm i'** у директоріях **front** та **back**
- 5) Використовувати базу даних створену у «Лабораторній роботі №3».
- 6) У директорії **back** виконати команду **'npm run devstart'**. Результатом буде створення сервера для бекенду за посиланням <http://localhost:8080>
- 7) У директорії **front** виконати команду **'ng serve'**. Результатом буде створення сервера для фронтенду за посиланням <http://localhost:4200>
- 8) У випадку зміни порту 8080 у посиланні бекенду потрібно змінити у файлі **environment.ts** значення налаштування **apiUrl**.



```
1 // This file can be replaced during build by using the `fileReplacements` array.
2 // `ng build --prod` replaces `environment.ts` with `environment.prod.ts`.
3 // The list of file replacements can be found in `angular.json`.
4
5 export const environment = {
6   production: false,
7   apiUrl: 'http://localhost:8080'
8 };
9
10 /*
11  * For easier debugging in development mode, you can import the following file
12  * to ignore zone related error stack frames such as `zone.run`, `zoneDelegate.invokeTask`.
13  *
14  * This import should be commented out in production mode because it will have a negative impact
15  * on performance if an error is thrown.
16  */
17 // import 'zone.js/dist/zone-error'; // Included with Angular CLI.
18
```

Рекомендації щодо виконання

Структуру проекту простіше буде організувати за прикладом розміщеним тут:

<https://github.com/vitaliipolovyi/web-part2/tree/master/Labs/L4>

Тобто, **першочерговим завданням є розгорнути приклад та розібратися у ньому**. І далі уже або на його основі або виконуючи завдання в іншому ключі приступати до виконання персонального варіанту.

Кожна описана сутність повинна мати форму додавання/редагування/видалення та зручний список для перегляду усі записів.

Кожна описана сутність не обмежена вказаними полями та за потреби може бути змінена/розширена.

Звіти повинні бути реалізовані у вигляді таблиць.

Використання бібліотеки jQuery дозволене тільки у випадках роботи із компонентами, які напряду її вимагають Bootstrap, DataTables ...

Використані джерела

1. <https://expressjs.com>
2. <https://mongodb.com>
3. https://developer.mozilla.org/uk/docs/Learn/Server-side/Express_Nodejs
4. <https://angular.io>

Варіанти робіт

Варіант 1

1. Реалізувати з допомогою класів такі інформаційні об'єкти:
 - a. Пасажир
 - i. Ім'я
 - ii. Прізвище
 - iii. Номер паспорта (унікальне значення)
 - b. Потяг
 - i. Назва
 - ii. Маршрут (сутність або стрічка формату "Станція1:Станція2")
 - iii. Номер (унікальне значення)
 - iv. Кількість місць
 - c. Квитки
 - i. Номер квитка (унікальне значення)
 - ii. Ціна
 - iii. Пасажир
 - iv. Потяг
 - v. Номер місця
 - vi. Квиток
 - vii. Дата
- Потрібно враховувати, що в один день не можна продати більше квитків на потяг чим вказано у полі [Потяг].[Кількість місць].
- Не можна продати той же квиток на один потяг на конкретну дату (враховуючи номер місця).

Реалізувати такі можливості системи:

- a. Додавання нового пасажир в колекцію
- b. Редагування пасажир в колекції
- c. Видалення пасажир з колекції
- d. Пошук одного пасажир в колекції
- e. Додавання потяга в колекцію
- f. Редагування потяга в колекції
- g. Видалення пасажир з колекції
- h. Пошук одного потяга в колекції
- i. Покупка пасажиром квитка на потяг
- j. Зміна квитка із одного потяга на інший
- k. Скасування покупки квитка
- l. Звіт найпопулярніші маршрути
- m. Звіт найприбутковіші маршрути
- n. Звіт продані квитки на заданий потяг
- o. Звіт маршрути без проданих квитків

Варіант 2

1. Реалізувати інформаційні об'єкти:

- a. Порт
 - i. Назва
 - ii. Країна
 - iii. Номер ЄДРПОУ (унікальне поле)
 - iv. Адреса
 - b. Корабель
 - i. Назва
 - ii. Бортовий номер (унікальне поле)
 - iii. Країна
 - iv. Тоннажність
 - v. Осадка (у метрах)
 - c. Пристань
 - i. Порт
 - ii. Номер (унікальне поле)
 - iii. Місткість (максимальна тоннажність кораблів на пристані)
 - iv. Мінімальна осадка корабля
 - d. Кораблі на пристані
 - i. Корабель
 - ii. Пристань
- Сумарна тоннажність кораблів на пристані не може перевищувати [Пристань].[Місткість].
 - Корабель не може причалити на пристань якщо [Пристань].[Мінімальна осадка корабля] < [Корабель].[Осадка]

Реалізувати такі можливості системи:

- a. Додавання нового порту в колекцію
- b. Редагування порту в колекції
- c. Видалення порту з колекції
- d. Пошук одного порту в колекції
- e. Додавання корабля в колекцію
- f. Редагування корабля в колекції
- g. Видалення корабля з колекції
- h. Пошук одного корабля в колекції
- i. Додавання пристані до порту
- j. Видалення пристані із порту
- k. Прибуття/Відбуття корабля від пристані
- l. Звіт кораблів усіх кораблів на заданій пристані
- m. Звіт про розміщення кораблів на пристанях заданого порту

Варіант 3

1. Реалізувати інформаційні об'єкти:

- a. Замовник
 - i. Назва
 - ii. Код (унікальне поле)
 - iii. Бюджет
 - b. Проект
 - i. Назва
 - ii. Номер (унікальне поле)
 - iii. Короткий опис
 - iv. Власник (Замовник)
 - v. Мінімальний досвід виконавця
 - c. Виконавець
 - i. Назва
 - ii. Код (унікальне поле)
 - iii. Досвід
 - iv. Кількість працівників
 - d. Проекти на виконанні
 - i. Проект
 - ii. Виконавець
 - iii. Час початку
 - iv. Час завершення
- Виконавець не може виконувати більше ніж $\text{ceil}([\text{Виконавець}].[Кількість працівників] / 5)$ проектів одночасно.
 - Виконавець не може виконувати проект, якщо $[\text{Виконавець}].[Досвід] < [\text{Проект}].[Мінімальний досвід виконавця]$

Реалізувати такі можливості системи:

- a. Додавання нового замовника в колекцію
- b. Редагування замовника в колекції
- c. Видалення замовника з колекції
- d. Пошук одного замовника в колекції
- e. Додавання проекту в колекцію
- f. Редагування проекту в колекції
- g. Видалення проекту з колекції
- h. Додавання виконавця в колекцію
- i. Редагування виконавця в колекції
- j. Видалення виконавця з колекції
- k. Звіт про виконані проекти заданого замовника
- l. Звіт про виконані проекти заданого виконавця
- m. Звіт по проектах на задану дату (хто і який проект виконував)

Варіант 4

1. Реалізувати інформаційні об'єкти:
 - a. Космічна станція
 - i. Номер (унікальне значення)
 - ii. Місткість
 - iii. Потреба (потреба станції у вантажах)
 - b. Планета
 - i. Назва (унікальне значення)
 - ii. Місткість складів
 - iii. Маса
 - c. Станції на орбітах
 - i. Космічна станція
 - ii. Планета
 - d. Вантаж
 - i. Код (унікальне поле)
 - ii. Назва
 - iii. Вага
 - e. Доставлені вантажі на станцію
 - i. Космічна станція
 - ii. Вантаж
 - f. Доставлені вантажі на планету
 - i. Планета
 - ii. Вантаж
- Сумарна кількість доставлених станцію вантажів не повинна перевищувати $[Космічна\ станція].[Місткість]$ та $[Космічна\ станція].[Потреба] * 1.2$.
- Сумарна кількість доставлених на планету вантажів не повинна перевищувати $[Планета].[Місткість]$

Реалізувати такі можливості системи:

- a. Додавання нової станції колекцію
- b. Редагування станції в колекції
- c. Видалення станції з колекції
- d. Пошук однієї станції в колекції
- e. Додавання планети в колекцію
- f. Редагування планети в колекції
- g. Видалення планети з колекції
- h. Пошук планети в колекції
- i. Додавання вантажу в колекцію
- j. Редагування вантажу в колекції
- k. Видалення вантажу з колекції
- l. Пошук вантажу в колекції
- m. Доставка вантажу на станцію
- n. Доставка вантажу на планету
- o. Перевезення вантажу із станції на станцію
- p. Перевезення вантажу із планети на станцію
- q. Звіт станції у на які доставлено вантажів менше 30% від потреби

Варіант 5

1. Реалізувати інформаційні об'єкти:

- a. Магазин
 - i. Назва
 - ii. Адреса
 - iii. Місткість
 - b. Товар
 - i. Код (унікальне значення)
 - ii. Назва
 - iii. Країна виробник
 - c. Склад
 - i. Номер (унікальне значення)
 - ii. Магазин
 - iii. Місткість
 - d. Товари на складі
 - i. Склад
 - ii. Товар
 - e. Товари в магазині
 - i. Магазин
 - ii. Товар
- Сумарна кількість товарів на складі не може перевищувати [Склад].[Місткість].
 - Сумарна кількість товарів в магазині не може перевищувати [Магазин].[Місткість].

Реалізувати такі можливості системи:

- a. Додавання нового магазину в колекцію
- b. Редагування магазину в колекції
- c. Видалення магазину з колекції
- d. Пошук одного магазину в колекції
- e. Додавання товару в колекцію
- f. Редагування товару в колекції
- g. Видалення товару з колекції
- h. Пошук одного товару в колекції
- i. Додавання складу в колекцію
- j. Редагування складу в колекції
- k. Видалення складу з колекції
- l. Пошук складу в колекції
- m. Доставка товару на склад
- n. Видалення товару на склад
- o. Трансфер товару із складу на склад тільки якщо вони належать одному магазину
- p. Відвантаження товару в магазин із складу
- q. Повернення товару із магазину на склад
- r. Звіт по складах у яких кількість одиниць товару менша від 20% від місткості
- s. Звіт одиниці товару на заданому складі