

Лабораторна робота №2

Тема:

Документ та об'єкт сторінки. Робота із подіями.

Налаштування середовища розробки

- 1) Створюємо директорію **lab2** у якому знаходитимуться вихідні коди лабораторної роботи.
- 2) Запускаємо команду **'npm init'**:
 - a. *package name: (code)* > lab1
 - b. *version* (1.0.0)
 - c. *description:* > Lab2
 - d. *entry point:* > server.js
 - e. *test command:*
 - f. *git repository:*
 - g. *keywords:*
 - h. *author:* > {yourname}
 - i. *license: (ISC)* > MIT
- 3) Запускаємо команду у терміналі (або Git Bash для Windows) **'git init'**.
- 4) Створюємо файл **.gitignore** у якому прописуємо стрічку **node_modules/**.
- 5) Виконуємо команду **./node_modules/.bin/eslint --init'**:
 - a. *How would you like to use ESLint?* > To check syntax, find problems, and enforce code style
 - b. *What type of modules does your project use?* > JavaScript modules (import/export)
 - c. *Which framework does your project use?* > None of these
 - d. *Where does your code run?* > <a> (both)
 - e. *How would you like to define a style for your project?* > Use a popular style guide
 - f. *Which style guide do you want to follow?* > Standard
 - g. *What format do you want your config file to be in?* > JSON
- 6) Встановлюємо Express.js виконавши команду **'npm install express --save'**.
- 7) Встановлюємо шаблонізатор Pug (Jade) **'npm install pug --save'**.
- 8) Виконуємо команду **'git status'**.
- 9) Виконуємо команду **'git add .'**.
- 10) Виконуємо команду **'git commit -m "Initial commit"'**.

Рекомендації щодо виконання

Структуру проекту простіше буде організувати за прикладом розміщеним тут:

<https://github.com/vitaliipolovyi/web-part2/tree/master/Labs/L2/Example>

Shop

Name Address

Show entries

Search:

| ID | Name | Address |
|----|-------|---------|
| 1 | Sho1 | Lviv |
| 2 | SHoP2 | Kiyv |

Showing 1 to 2 of 2 entries

Previous Next

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 напямку її вимагають Bootstrap, DataTables ...

Використані джерела

1. <https://habr.com/ru/post/434686/>
2. <https://expressjs.com>
3. <https://pugjs.org/api/getting-started.html>

Варіанти робіт

Варіант 1

1. Реалізувати з допомогою класів такі інформаційні об'єкти:

- a. Пасажир
 - i. Ім'я та Прізвище
 - ii. Номер паспорта (унікальне значення)
- b. Квиток
 - i. Номер квитка (унікальне значення)
 - ii. Ціна
- c. Потяг
 - i. Назва
 - ii. Маршрут (сутність або стрічка)
 - iii. Номер (унікальне значення)
- d. Продані квитки
 - i. Пасажир
 - ii. Потяг
 - iii. Квиток
 - iv. Дата

- 1. Не можна продати той же квиток на один потяг на конкретну дату

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
- Усі колекції записувати у `localStorage`

Варіант 2

1. Реалізувати з допомогою класів такі інформаційні об'єкти:

- a. Порт
 - i. Назва
 - ii. Країна
 - iii. Номер ЗКПО (унікальне поле)
 - iv. Адреса
- b. Корабель
 - i. Назва
 - ii. Бортовий номер (унікальне поле)
 - iii. Країна
 - iv. Тоннажність
- c. Пристань
 - i. Порт
 - ii. Номер (унікальне поле)
 - iii. Місткість (максимальна кількість кораблів на пристані)
- d. Кораблі на пристані
 - i. Корабель
 - ii. Пристань

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
- Усі колекції записувати у `localStorage`

Варіант 3

1. Реалізувати з допомогою класів такі 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)
- 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

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 проекти заданого замовника
- l. Звіт про проекти заданого виконавця
- m. Звіт по проектах на задану дату (хто і який проект виконував)
- Усі колекції записувати у `localStorage`

Варіант 4

1. Реалізувати з допомогою класів такі інформаційні об'єкти:
 - a. Космічна станція
 - i. Номер (унікальне значення)
 - ii. Місткість (кількість доставлених вантажів не повинна перевищувати це значення)
 - iii. Потреба (потреба станції у вантажах)
 - b. Планета
 - i. Назва (унікальне значення)
 - ii. Місткість складів (кількість доставлених вантажів не повинна перевищувати це значення)
 - 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

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
- Усі колекції записувати у `localStorage`

Варіант 5

1. Реалізувати з допомогою прототипів або класів такі інформаційні об'єкти:

- a. Магазин
 - i. Назва
 - ii. Адреса
- b. Товар
 - i. Код (унікальне значення)
 - ii. Назва
 - iii. Країна виробник
- c. Склад
 - i. Номер (унікальне значення)
 - ii. Магазин
 - iii. Місткість (кількість одиниць товарів не повинна перевищувати кількість товарів на складі)
- d. Товари на складі
 - i. Склад
 - ii. Товар

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. Звіт по складах у яких кількість одиниць товару менша від 20% від місткості
- r. Звіт одиниці товару на заданому складі
- s. Усі колекції записувати у `localStorage`