

## Лабораторна робота №3.

### Мережева функціональність JS

#### Мета роботи

Організація бекенда і фронтенда за допомогою JS та супутніх технологій.

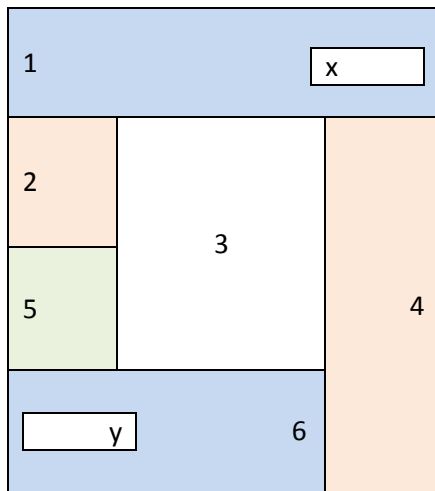
#### Тривалість роботи

4 години

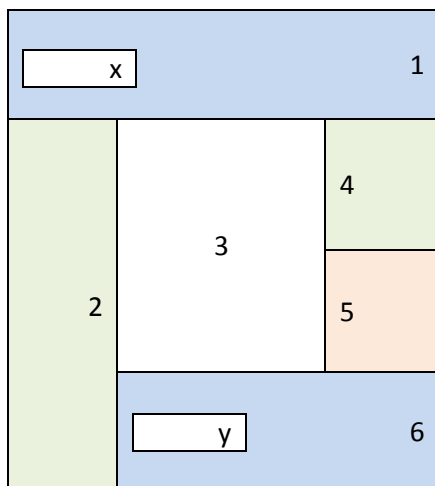
#### Завдання

1. Продумайте кілька способів реалізації завдання.
2. За допомогою HTML, CSS, мови програмування JS, та, при потребі, довільної серверної мови програмування (далі СМП) виконайте вказані маніпуляції з документами згідно з вашим варіантом:
  - a. В центральному блоці («3» або «4») однієї із ваших сторінок за допомогою JS (та при потребі CSS) автоматизуйте процес створення об'єкта (або набору об'єктів) згідно із варіантом завдання із довільним контентом. Кількість елементів, порядок розташування елементів і контент в елементах задає користувач сторінки.
  - b. За допомогою JS та обраної СМП організуйте збереження в асинхронному режимі технічних параметрів та контенту об'єкта (набору об'єктів) на сервері у вибраному вами форматі (файли, БД тощо).
  - c. В центральному блоці («3» або «4») іншої сторінки за допомогою JS (та при потребі обраної СМП) забезпечте підтягування технічних параметрів та контенту із сервера і формування необхідного для функціонування об'єкта (набору об'єктів) коду на HTML та CSS. Завантажений об'єкт (набір об'єктів) має працювати на чистому CSS, без використання JS та CSS-фреймворків типу Bootstrap.
  - d. При зміні технічних параметрів та/або контенту об'єкта (набору об'єктів) на першій сторінці оновлюйте об'єкт (набір об'єктів) на другій сторінці відповідно до змінених даних, логіка функціонування механізму оновлення на ваш розсуд.
3. Для запису стилів і скриптів використовуйте будь-які відомі вам способи.
4. Виконане завдання завантажте у Github і продемонструйте його роботу на будь-якому із доступних вам хостінгів або сервісів розгортання веб-додатків (Netlify, Vercel, Heroku, Firebase, Stormkit тощо).
5. Поясніть, в чому реалізований вами спосіб реалізації завдання кращий за інші з п.1.

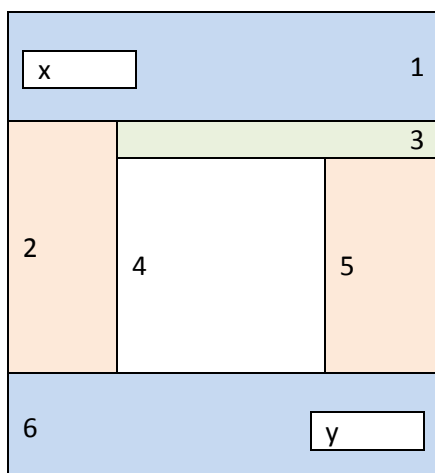
Варіант 1 – об'єкт «Accordion», приклад об'єкта за [посиланням](#).



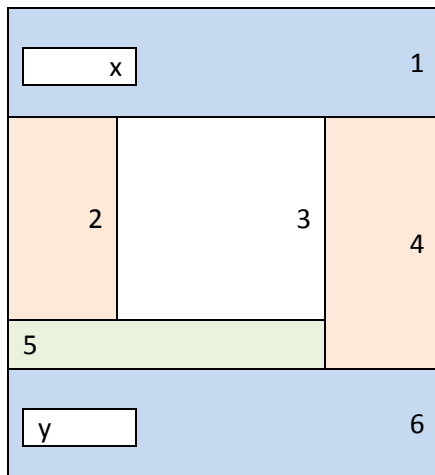
Варіант 2 – набір об'єктів «Collapse», приклад об'єкта за [посиланням](#).



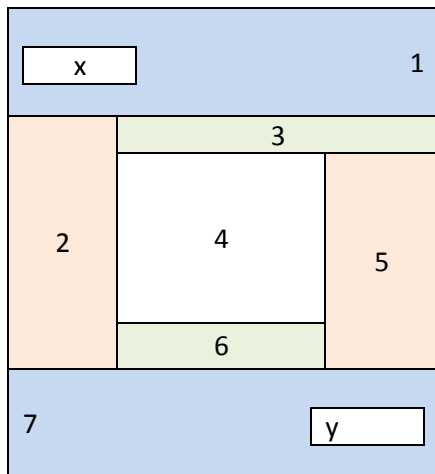
Варіант 3 – об'єкт «Slider», приклад об'єкта за [посиланням](#).



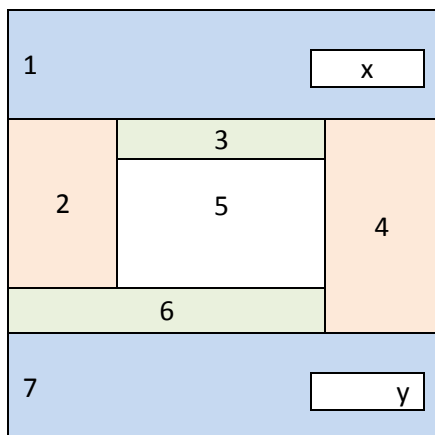
Варіант 4 – об'єкт «Tabs», приклад за [посиланням](#).



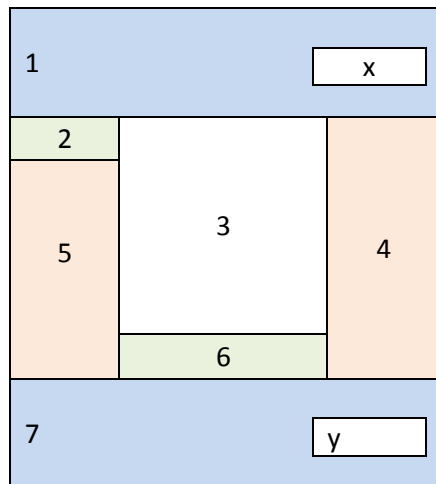
Варіант 5 – набір об'єктів «Dropdown», приклад за [посиланням](#).



Варіант 6 – об'єкт «Masonry», приклад за [посиланням](#).

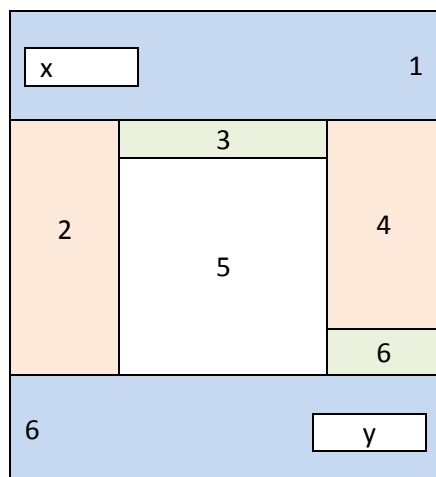


Варіант 7 – об'єкт «Glitch Text», приклад за [посиланням](#).

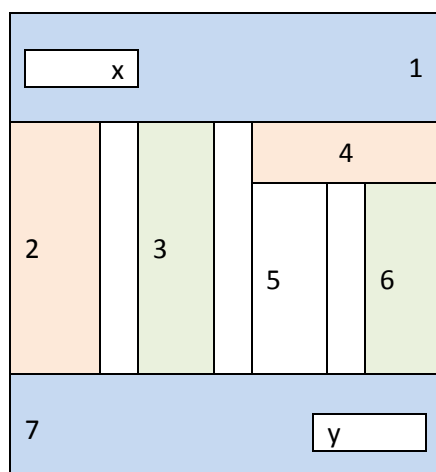


Змінювані параметри – відступи, кольори, тривалість анімації тощо – всі CSS-інструкції, що дозволяють змінити початкову візуалізацію тексту.

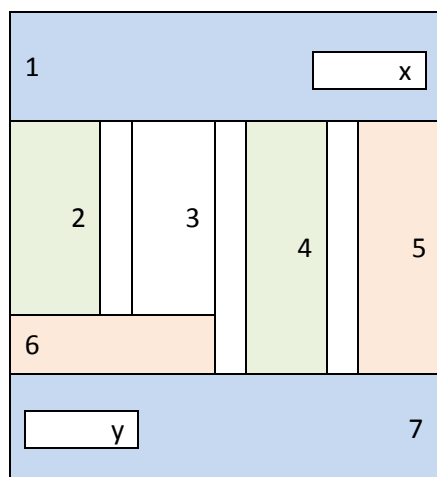
Варіант 8 – об'єкт «Notification Bar», приклад об'єкта за [посиланням](#).



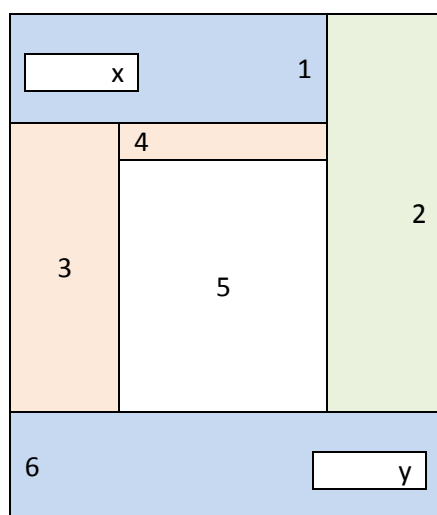
Варіант 9 – набір об'єктів «Accordion», приклад об'єкта за [посиланням](#).



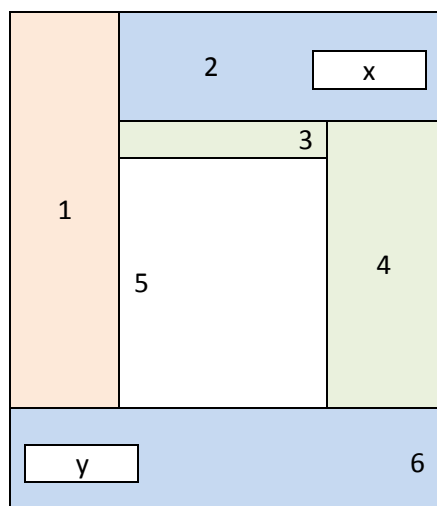
Варіант 10 – набір об'єктів «Collapse», приклад об'єкта за [посиланням](#).



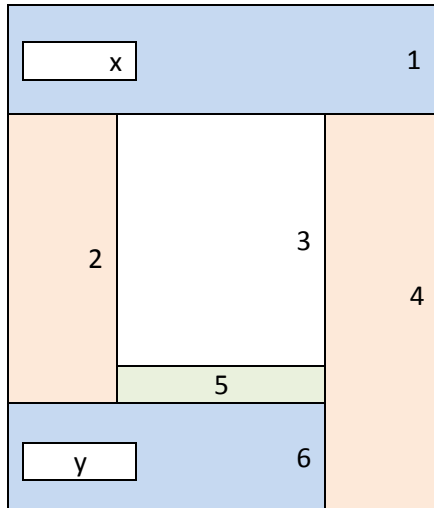
Варіант 11 – об'єкт «Slider», приклад об'єкта за [посиланням](#).



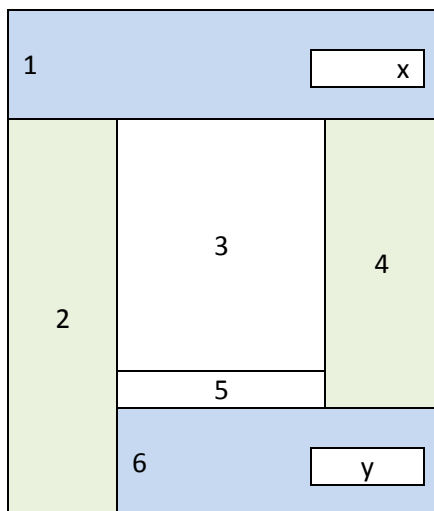
Варіант 12 – об'єкт «Tabs», приклад за [посиланням](#).



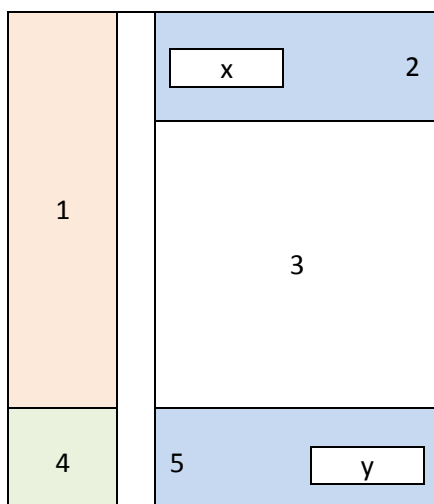
Варіант 13 – набір об'єктів «Dropdown», приклад за [посиланням](#).



Варіант 14 – об'єкт «Masonry», приклад за [посиланням](#).



Варіант 15 – об'єкт «Glitch Text», приклад за [посиланням](#).



Змінювані параметри – відступи, кольори, тривалість анімації тощо – всі CSS-інструкції, що дозволяють змінити початкову візуалізацію тексту.

Варіант 16 – об'єкт «Notification Bar», приклад об'єкта за [посиланням](#).

