

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій**

ЗВІТ

До індивідуального проєкту №1

**на тему «Розробка інформаційного WEB-застосунку для медичної клініки
"MedEmoc"»**

з дисципліни «WEB-орієнтовані технології. Frontend розробки»

Виконав(ла): студент(ка) групи ІА-33
Піголь Тетяна Олександрівна

Перевірів(ла): Проскура Світлана Леонідівна

Київ – 2025 року

ВСТУП

Стрімкий розвиток інформаційних технологій та глобальна цифровізація суспільства докорінно змінюють підходи до надання послуг у різних сферах, зокрема й у медицині. Наявність власного вебресурсу для медичного закладу сьогодні є не просто перевагою, а необхідною умовою для ефективної взаємодії з пацієнтами, забезпечення прозорості роботи та підвищення конкурентоспроможності.

Сучасний користувач очікує швидкого та зручного доступу до інформації про лікарів, послуги та графік роботи, а також можливості записатися на прийом у режимі онлайн без необхідності здійснювати телефонні дзвінки чи відвідувати заклад особисто.

1 АНАЛІЗ ТА МОДЕЛЮВАННЯ СИСТЕМИ

Тема WEB-застосунку: Розробка вебсайту для приватної медичної клініки «MedEmos».

Актуальність теми: У сучасному світі цифровізація медичних послуг є критично важливою. Пацієнти потребують швидкого доступу до інформації про лікарів, графік роботи та переліку послуг без необхідності фізичного відвідування закладу чи телефонних дзвінків. Розробка власного вебресурсу для клініки дозволяє автоматизувати процес інформування та спростити процедуру запису на прийом, що підвищує лояльність клієнтів та ефективність роботи персоналу.

Мета роботи: Створення адаптивного вебзастосунку, який надає користувачам вичерпну інформацію про клініку «MedEmos», її спеціалістів, дозволяє переглядати графік роботи та здійснювати запис на прийом з можливістю збереження талона.

Бізнес-логіка роботи: Користувач заходить на головну сторінку, де ознайомлюється із загальною інформацією та графіком роботи. Він може перейти на сторінку "Наші лікарі", щоб переглянути детальну інформацію про спеціалістів через модальні вікна. Після вибору послуги користувач переходить до форми

запису, вводить персональні дані, обирає дату та час (система автоматично фільтрує минулі дати та генерує часові слоти). Після підтвердження формується файл із деталями запису, який завантажується на пристрій користувача.

Функціональні вимоги:

1. Перегляд інформації про клініку та графік роботи.
2. Перегляд списку лікарів з фотографіями та спеціалізацією.
3. Перегляд детальної інформації про лікаря у спливаючому вікні.
4. Валідація форми запису (обов'язкові поля, коректність дати).
5. Генерація доступних часових слотів для запису.
6. Формування та завантаження текстового файлу з даними про запис.

Нефункціональні вимоги:

1. Адаптивність інтерфейсу.
2. Інтуїтивно зрозумілий UI/UX.
3. Швидкодія завантаження сторінок.
4. Кросбраузерність.

Use-case діаграма:



Стек-технологій: HTML5, CSS3, JavaScript (ES6+).

2 РОЗРОБЛЕННЯ АДАПТИВНОГО ІНТЕРФЕЙСУ WEB-ЗАСТОСУНКУ.

Структурні компоненти інтерфейсу:

1. Шапка (header): Містить логотип клініки, назву "MedEmoc" та навігаційне меню (Головна, Послуги, Зв'язатися з нами).

Код:

```
<header>
  <div id="logo-section">
    
    <span class="clinic-name">Клініка "MedEmoc"</span>
  </div>
  <nav>
    <a href="main-L2.html" class="nav-link">Головна</a>
    <a href="#" class="nav-link">Послуги</a>
    <a href="mailto:pigol.tanya@lil.kpi.ua" class="nav-link" id="contact-us">Зв'язатися з нами</a>
  </nav>
</header>
```

Вигляд:

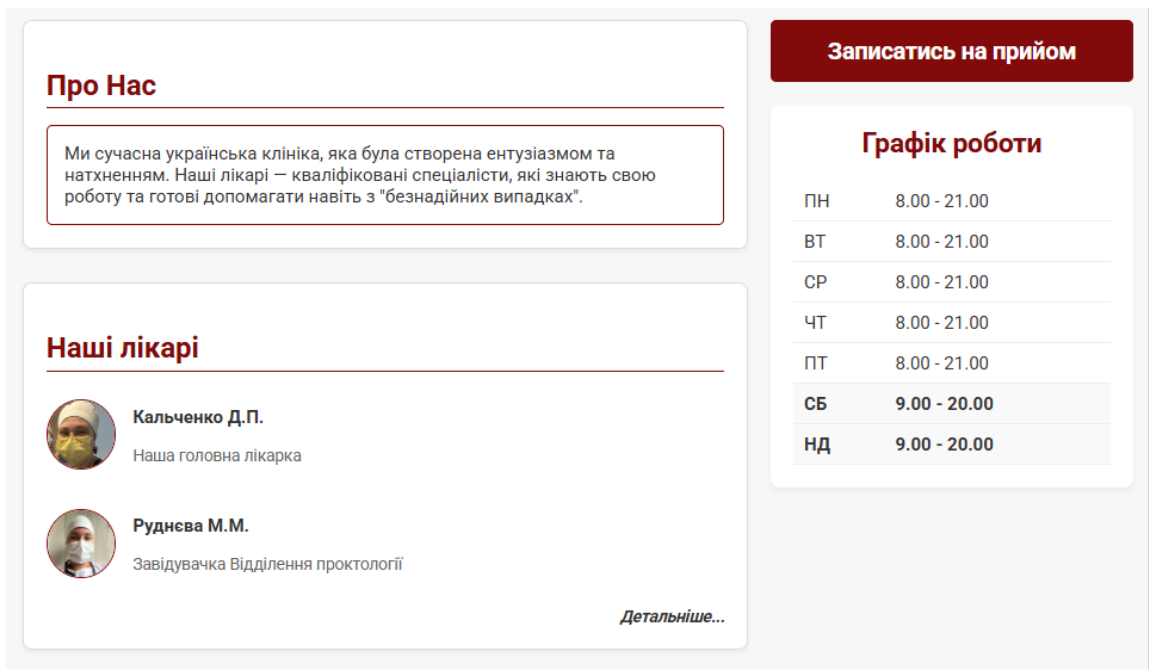


2. Основний блок (main): Головна сторінка розділена на дві колонки (Grid/Flex). Ліва частина містить інформацію про клініку та картки лікарів. Права частина містить кнопку запису та таблицю графіку роботи.

Код:

```
<main>
  <section class="left-column">
    <div class="info-block">...</div>
    <div class="info-block">...</div>
  </section>
  <aside class="right-column">...</aside>
</main>
```

Вигляд:

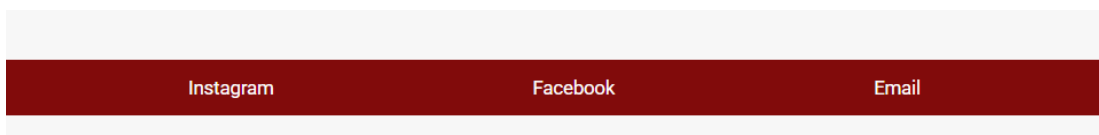


3. Нижній колонтитул (footer): Містить посилання на соціальні мережі Instagram, Facebook та Email.

Код:

```
<footer>
  <a href="https://www.instagram.com/">Instagram</a>
  <a href="https://www.facebook.com/">Facebook</a>
  <a href="mailto:pigol.tanya@lll.kpi.ua">Email</a>
</footer>
```

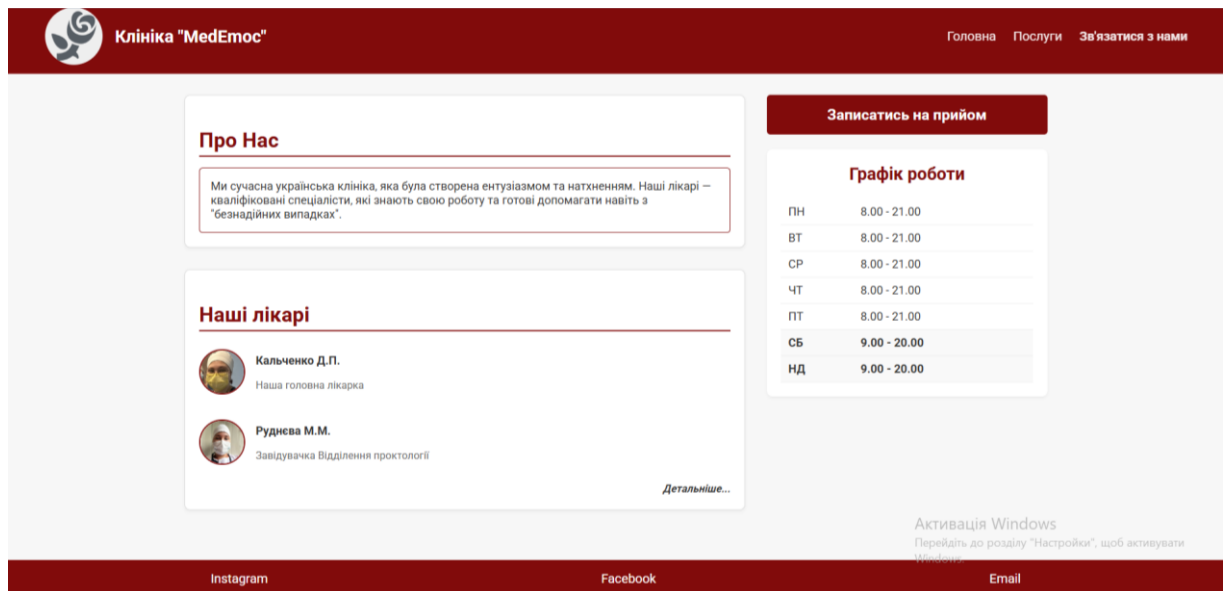
Вигляд:



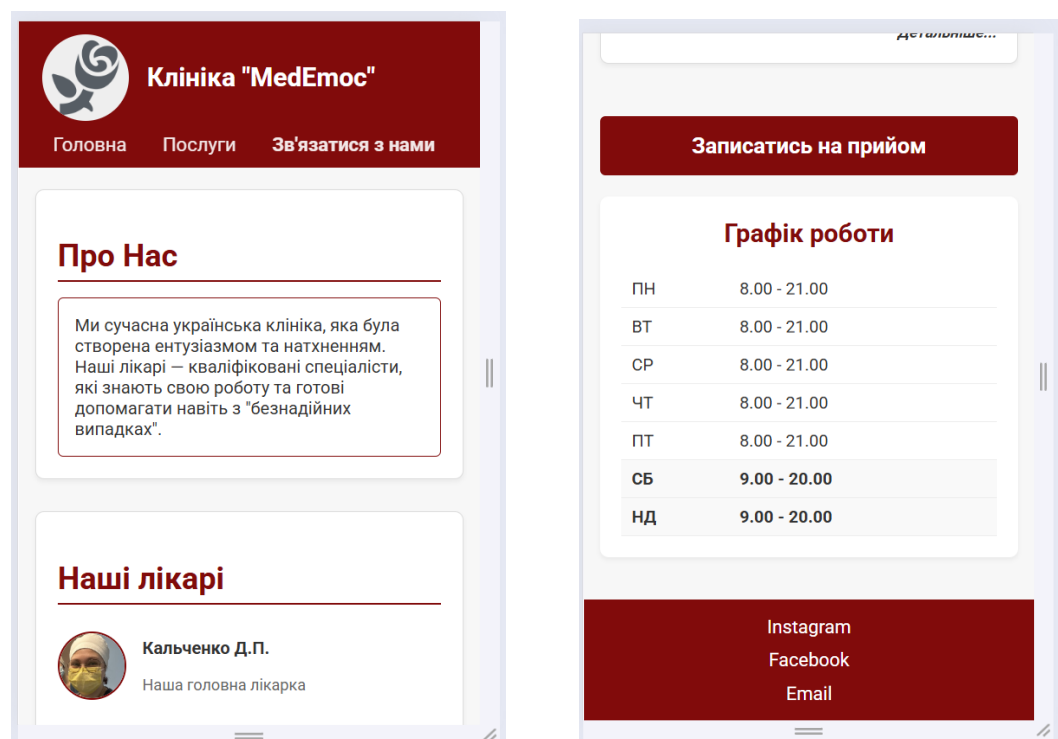
Функціонально-стилістичні вимоги:

Для реалізації адаптивності використано медіа-запити (@media).

- На великих екранах контент розміщується у дві колонки (left-column та right-column).

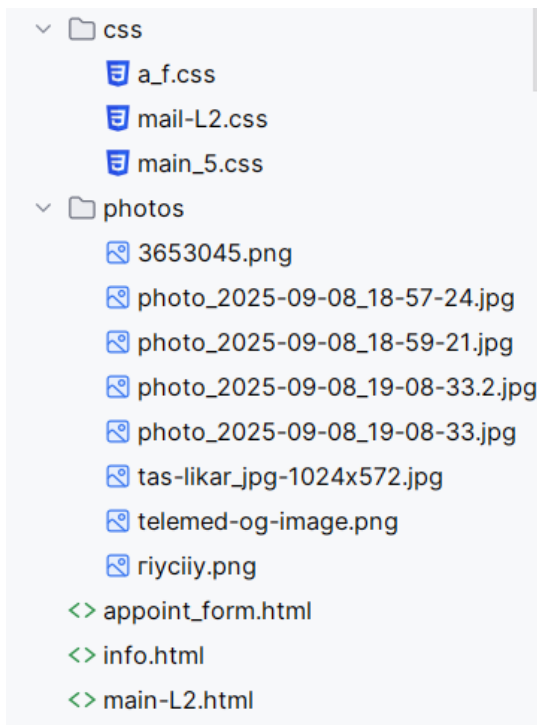


- На мобільних пристроях блоки шикуються вертикально в одну колонку для зручного скролінгу.



- Навігація адаптується під ширину екрана.

Файлова структура веб-застосунку:



Розміщення результатів роботи в GitHub.

css	adding styles
photos	adding photos
README.md	Initial commit
appoint_form.html	adding main files
info.html	adding main files
main-L2.html	adding main files

Посилання на GitHub: https://github.com/iatan33/IND_PROJECT_1

Посилання на живу сторінку: https://iatan33.github.io/IND_PROJECT_1/main-L2.html

3 ПРОГРАМНА ЛОГІКА (JAVASCRIPT)

У цьому розділі описано функціональність вебзастосунку, реалізована за допомогою JavaScript, а також логіка взаємодії користувача з інтерфейсом, обробка подій та структура бізнес-логіки.

3.1. Опис функціоналу

У проєкті реалізовано такі основні функціональні можливості:

1. Модальне вікно: Використано для відображення детальної інформації про лікаря без перезавантаження сторінки. Дані беруться з data- атрибутів HTML- елементів відповідаючи на подію click на блок лікаря.

Скрипт зчитує data-details та data-schedule, підставляє їх у модальне вікно і змінює його стиль display на block.

Код:

```
<div id="doctorModal" class="modal">
  <div class="modal-content">
    <span class="close-button"></span>

    <h2 id="modalDoctorName">Ім'я лікаря</h2>

    <p id="modalDoctorInfo" style="..."></p>

    <hr> <h3>Детальна інформація:</h3>
    <p id="modalFullDetails"></p>

    <p><strong>Графік роботи:</strong> <span id="modalSchedule"></span></p>
  </div>
</div>

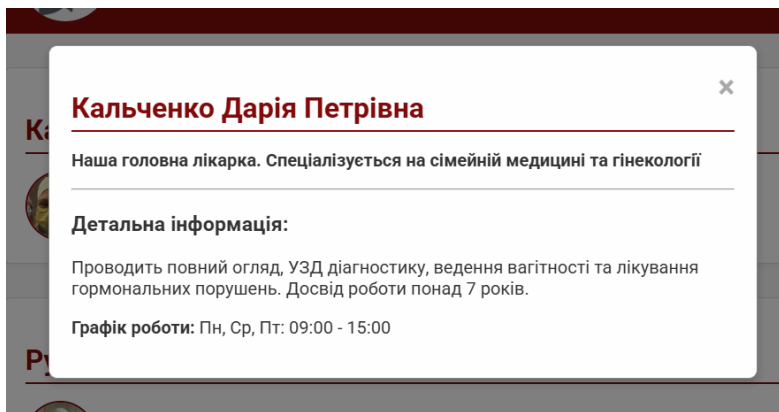
container.addEventListener('click', (event) => {
  const doctorBlock = event.target.closest('.info-block');
  if (!doctorBlock) return;

  const name = doctorBlock.querySelector('h2').innerText;
  const shortDescription = doctorBlock.querySelector('.doctor-text p').innerText;
  const fullDetails = doctorBlock.dataset.details || "Інформація уточнюється";
  const schedule = doctorBlock.dataset.schedule || "За записом";

  modalName.innerText = name;
  modalInfo.innerText = shortDescription;
  modalFullDetails.innerText = fullDetails;
  modalSchedule.innerText = schedule;

  modal.style.display = 'block';
});
```

Вигляд:



2. Генерація дати та часу в appoint_form.html: Функція `getTodayDate()` отримує поточну дату і встановлює атрибут `min` для поля вибору дати, щоб не можна було обрати минулий день. Використано цикли `for` для динамічного створення опцій (`<option>`) вибору часу з інтервалом 5 хвилин.

Код:

```
if (timeSelect) {
  for (let hour = startTime; hour <= endTime; hour++) {
    for (let minute = 0; minute < 60; minute += stepMinutes) {
      if (hour === endTime && minute > 0) {
        break;
      }

      const option = createTimeOption(hour, minute);
      timeSelect.appendChild(option);
    }
  }
}
```

3.3. Валідація форм та бізнес-логіка збереження

Реалізовано перевірку заповнення полів через атрибут `required` та обробку події `submit`.

Ключова бізнес-логіка: Формування талона запису.

Оскільки проєкт не використовує серверну базу даних, реалізовано збереження даних у текстовий файл на стороні клієнта за допомогою об'єкта `Blob`.

Код:

Збереження файлу:

```
form.addEventListener('submit', function(event) {
    event.preventDefault();
```

Збір даних:

```
const surname = document.getElementById('surname').value;
const name = document.getElementById('name').value;
const date = document.getElementById('date').value;
const time = document.getElementById('time').value;

const serviceSelect = document.getElementById('service');
const serviceText = serviceSelect.options[serviceSelect.selectedIndex].text;
```

Формування структури файлу:

```
const fileContent =
    `=== НОВИЙ ЗАПИС ДО ЛІКАРЯ ===
Прізвище: ${surname}
Ім'я: ${name}
Дата: ${date}
Час: ${time}
Послуга: ${serviceText}
Дата створення заявки: ${new Date().toLocaleString('uk-UA')}
=====`;
```

Створення та завантаження файлу:

```
const blob = new Blob([fileContent], { type: "text/plain;charset=utf-8" });
const link = document.createElement("a");

link.href = URL.createObjectURL(blob);
link.download = `Zapys_${surname}_${date}.txt`;
link.click();
```

ВИСНОВОК

У ході виконання індивідуального проєкту було розроблено Frontend-частину вебзастосунку для медичної клініки «MedEmos». Головною метою роботи було створення зручного та адаптивного інтерфейсу, що дозволяє користувачам отримувати інформацію про лікарів та записуватися на прийом онлайн.

У результаті роботи створено повноцінну структуру вебсайту, яка складається з головної сторінки, сторінки з інформацією про спеціалістів та форми запису. Було використано технологій CSS Grid та Flexbox у поєднанні з медіа-запитами для забезпечення коректного відображення контенту на пристроях із різною роздільною здатністю (десктопи, планшети, смартфони).

За допомогою мови програмування JavaScript реалізовано динамічні елементи інтерфейсу, зокрема модальні вікна для перегляду детальної інформації про лікарів без перезавантаження сторінки.

Розроблено логіку валідації введених даних (перевірка дати, часу, обов'язкових полів). Реалізовано клієнтську емуляцію збереження даних: після успішного заповнення форми генерується та завантажується текстовий файл із деталями запису.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Оформлення текстових документів у навчальному процесі. Стандарт організації (кафедри) СОУ ІСТ 01-22. Для студентів кафедри інформаційних систем та технологій.

2. Conventional Commits URL:
<https://www.conventionalcommits.org/en/v1.0.0/#commit-message-with-description-and-breaking-change-footer>

3. HTML Підручник. W3Schools українською. URL:
<https://w3schoolsua.github.io/html/index.html>

4. HTML. Мова для створення веб-сторінок. W3Schools українською. URL:
<https://w3schoolsua.github.io/index.html#gsc.tab=0>

5. HTML Довідник тегів. URL: <https://w3schoolsua.github.io/tags/index.html>

6. CSS Підручник. W3Schools українською. URL:
<https://w3schoolsua.github.io/css/index.html#gsc.tab=0>

7. HTML. Language for creating web pages. W3Schools (English). URL:
https://w3schoolsua.github.io/index_en.html#gsc.tab=0

8. HTML Tutorial. W3Schools (English). URL:
https://w3schoolsua.github.io/html/index_en.html#gsc.tab=0

9. Підручники HTML та CSS. HTMLBook. URL:
https://htmlbook.at.ua/news/tutorial_html/1-0-1

10. Learn to Code. With the world's largest web developer site. W3Schools. URL:
<https://www.w3schools.com/>

11. Український веб-довідник. URL: <https://css.in.ua/>

12. Довідник по HTML тегам. URL: <https://css.in.ua/html/tags>

13. Довідник HTML тегів. HTML-CSS.co.ua. URL: <https://html-css.co.ua/dovidnuk-html-tags/>

14. HTML конструювання. HTMLBook. URL: <http://htmlbook.in.ua/>;
<http://htmlbook.in.ua/tegs-html/>; <http://htmlbook.in.ua/pr01-html/>

15. CSS-TRICKS. Guides. Articles. URL: <https://css-tricks.com/>