

## Загальні вимоги

- Адаптивна верстка: Інтерфейс має коректно відображатися на всіх пристроях – від смартфонів до великих екранів.
- Кросбраузерність: Підтримка останніх версій сучасних браузерів (Chrome, Firefox, Safari, Edge).
- Валідність коду: Перед відправкою завдання перевірити HTML на <https://validator.w3.org/> або за допомогою інструментів ESLint/Stylelint.
- Чистий та структурований код: Зрозумілі назви класів (BEM або аналогічна методологія), коментарі в ключових місцях коду. Не повинно бути великої кількості вкладень і довгих назв класів.
- Використання інструменту автоматизації задач для веб-розробки (gulp, vite).

## Функціональні вимоги

- Header і Footer:
  - Logo повинно бути посиланням на головну сторінку сайту.
  - Посилання в меню повинні мати hover ефект.
  - На мобільних пристроях меню повинне переходити в “бургер-меню”.
- Блок 1:
  - Повинен займати всю висоту екрану.
  - Кнопка SCROLL DOWN повинна бути якорним посиланням до наступної секції.
- Блок 2
  - Блоки повинні відображатися на різних екранах і з різною кількістю тексту правильно, не перекривати і не ламати один одного.
- Блок 3
  - Реалізувати інтерактивний слайдер з можливістю ручного переключення та (опційно) автопрокруткою. Рекомендовані інструменти: Swiper.js або аналогічні бібліотеки.
  - Кнопки повинні мати hover ефект
  - Перехід на сторінку проекту можливий через зображення або назву.
- Блок 4
  - Реалізувати функціонал акордеону (за допомогою бібліотеки, наприклад bootstrap)
- Блок 5
  - Реалізувати функціонал контактної форми і відправку за допомогою Ајах
  - Поля форми повинні мати стани та валідацію.
  - Після відправки форма повинна обробляти відповідь сервера.

## Технології

### HTML5

- **Валідність коду:**
  - Використовувати стандартний і правильний синтаксис.
  - Перевіряти код через валідатори, такі як W3C Validator.
- **Семантична розмітка:**
  - використовувати семантичні HTML5-теги (<header>, <main>, <section>, <article>, <footer>, <nav>).
- **Атрибути:** вказувати всі обов'язкові атрибути (alt для всіх зображень <img>).

### CSS

- **Адаптивність:** використовувати медіа-запити для різних пристроїв, застосовувати відносні одиниці вимірювання (% , em, rem, vh, vw) замість фіксованих (px).
- **Зовнішні стилі:** використовувати зовнішні CSS-файли, а не вбудовані стилі, завантажувати стилі через <link> у <head>.
- **Рекомендовано** використовувати препроцесори SCSS/SASS.

### JavaScript

- **Читабельність та чистота коду:**
  - використовувати зрозумілі назви змінних, функцій та класів (наприклад, calculateTotalPrice, а не calcTP).
  - Дотримуватись єдиного стилю написання коду (camelCase для змінних та функцій, PascalCase для класів).
- **Оптимізація продуктивності:**
  - Уникати дублювання коду, створюючи багаторазово використововані функції.
- **Кросбраузерна сумісність:**
  - Перевіряти код у різних браузерах або застосовувати поліфіли для підтримки застарілих версій.
  - Писати код відповідно до стандартів ECMAScript, які підтримуються цільовими браузерами.

### Додаткові вимоги

- Оптимізація продуктивності (використання сучасних форматів зображень та технології lazy loading).
- Використання сучасних підходів до верстки (Flexbox, Grid).
- Дотримання Accessibility (WCAG) стандартів для доступності контенту.

### Великим плюсом буде інтеграція з CMS Wordpress

- Інтеграція верстки з wordpress повинна бути виконана на чистій темі з використанням ACF.
- Всі тексти на сайті повинні редагуватися з адмін панелі.
- Назви полів повинні бути логічними (наприклад, Hero title, а не Field 1).

- Проекти повинні бути створені за допомогою custom post.
- Відправку форми можна реалізувати за допомогою плагіна або коду.  
Листи повинні відправлятися, а відповідь сервера оброблятися.
- Меню в header і footer повинно бути реалізоване через wp\_nav\_menu.
- Оптимізація продуктивності: оптимізувати сайт за допомогою плагінів (наприклад WP Rocket)