

Як підготувати дизайн сайту до розробки

Робочий порядок: від бізнес-задачі та структури сторінок до перевірених макетів і передачі команді

У ЦЬОМУ ГАЙДІ

Дизайн сайту має зафіксувати не лише вигляд екранів, а й порядок інформації, дії відвідувача та поведінку інтерфейсу в типових і проблемних ситуаціях. Роботу варто будувати так, щоб замовник, дизайнер і розробник однаково розуміли: для чого існує кожна сторінка, що на ній показано та що має статися після дії користувача.

- 01 1. Сформулюйте задачу кожної сторінки**
- 02 2. Побудуйте карту сайту та маршрут відвідувача**
- 03 3. Перетворіть сценарії на правила інтерфейсу**
- 04 4. Застосуйте візуальну систему й адаптуйте зміст**
- 05 5. Проведіть приймання макетів і передайте їх у розробку**
- Q Часті питання**

01 · 1. СФОРМУЛЮЙТЕ ЗАДАЧУ КОЖНОЇ СТОРІНКИ

Починати з кольорів, референсів або головного екрана небезпечно: без задачі сторінки команда оформлює зміст, який ще не має порядку. Спершу потрібно визначити, кому сторінка потрібна, яку відповідь вона дає та до якої конкретної дії підводить людину.

Одна сторінка не повинна одночасно намагатися пояснити всі послуги, презентувати каталог, збирати звернення та бути довідкою. Визначте її головне призначення: представити послугу, показати товар, дати доступ до матеріалів, пояснити умови або привести до контакту.

Для кожної дії зафіксуйте наслідок натискання. Кнопка не є декором: її назва має відповідати тому, що побачить або отримає відвідувач далі. Так само форма має пояснювати, які дані потрібні та навіщо їх збирають.

На цьому етапі маркетинг задає пропозицію і зміст повідомлення, а дизайн перетворює їх на послідовність блоків, заголовків, форм і переходів. Якщо немає погодженого змісту, дизайнеру доводиться оформлювати порожнє місце.

01 Назвіть головну дію сторінки

Для кожної ключової сторінки запишіть одну дію, яку вона підтримує: перегляд послуги, вибір варіанта, перехід до каталогу, надсилання форми або звернення.

02 Визначте запитання відвідувача

Складіть перелік того, що людина має зрозуміти до дії: що їй пропонують, для кого це, що входить у пропозицію, де знайти потрібний розділ і що буде після звернення.

03 Підготуйте зміст для проектування

Зберіть опис продуктів або послуг, тексти, фото, бренд-матеріали, документи та дані, які сайт має показувати або збирати.

04 Зафіксуйте обмеження

Передайте команді перелік інтеграцій, доступних даних і технічних обмежень. Дизайнеру важливо не припускати, що сайт може показувати або робити те, чого система не підтримує.

ДЕ ВТРАЧАЮТЬ

Найбільше часу втрачають, коли вимогу формують як «зробіть красиво» або «як у цього сайту». Такі фрази не визначають ані зміст, ані сценарій, тому рішення доводиться неодноразово переробляти.

ЧЕКЛІСТ РОЗДІЛУ

- ☐ Для кожної ключової сторінки записано її призначення та одну головну дію.
Без цього блоки конкуруватимуть між собою, а відвідувачеві доведеться здогадуватися, що робити.
- ☐ Зібрано матеріали, які має показати сайт: тексти, фото, документи, дані й бренд-матеріали.
Без змісту макет перетворюється на набір умовних блоків, які не можна якісно перевірити.
- ☐ Для прикладів інших сайтів пояснено конкретну корисну рису.
Інакше приклад залишиться чужою картинкою, а не зрозумілою вимогою до меню, форми чи каталогу.

Карта сайту задає межі роботи: які сторінки існують, як вони пов'язані та де відвідувач шукатиме послугу, товар або відповідь. Вона не замінює макети, але не дає навігації перетворитися на випадковий список посилань.

Почніть із реальних розділів бізнесу. Меню має називати послуги, категорії каталогу або інформаційні розділи, які справді існують на сайті. Якщо назва пункту меню не відповідає змісту сторінки, людина не може передбачити, куди потрапить.

Після карти сторінок опишіть маршрут для ключових сценаріїв: що людина бачить спочатку, куди переходить далі, яку інформацію отримує перед дією та як повертається до потрібного кроку. Особливо уважно перевіряйте каталог, пошук, фільтри, картки товарів або послуг і форми.

Структура має продовжувати логіку дії. Посилання між сторінками не повинні вести у довільне місце: кожен перехід має або давати відповідь, або наближати відвідувача до наступного кроку.

01 Складіть перелік сторінок

Випишіть усі необхідні сторінки та розділи: головну, послуги, каталог, картки, контакти, матеріали й інші потрібні частини сайту.

02 Зв'яжіть сторінки переходами

Для кожного ключового переходу визначте початкову точку, цільову сторінку та причину переходу: переглянути варіанти, уточнити умови, знайти відповідь або звернутися.

03 Опишіть ключові сценарії

Зафіксуйте послідовність дій для відвідувача: вхід на сторінку, ознайомлення з пропозицією, вибір потрібного розділу, заповнення форми або інша цільова дія.

04 Визначте пріоритети змісту

Позначте, що людина має побачити одразу, що має пояснювати рішення, а що доречно показати після основної інформації.

ДЕ ВТРАЧАЮТЬ

Каталог без логіки фільтрів або послуги, змішані в одному списку, змушують людину переглядати зайве. Проблему не вирішить новий колір кнопки: спершу потрібно відновити структуру вибору.

ЧЕКЛІСТ РОЗДІЛУ

☐ Меню відповідає реальним розділам, послугам або категоріям сайту.

Без відповідності між меню та змістом відвідувач не знатиме, де шукати потрібне.

- Для кожного ключового сценарію визначено наступний крок після перегляду сторінки.

Без маршруту сторінки не підтримуватимуть дію, а лише зберігатимуть інформацію.

- Порядок блоків погоджено до роботи над візуальним оформленням.

Інакше команда витрачатиме час на обговорення вигляду замість усунення структурних проблем.

Макет має описувати поведінку сайту, а не лише показувати один статичний екран. Людина взаємодіє з кнопками, полями, меню, пошуком і фільтрами; для кожного з цих елементів потрібні зрозумілі правила до, під час і після дії.

Розділіть роботу на компоненти та сценарії. Компоненти — це кнопки, поля, перемикачі, підказки, повідомлення про помилки й порожні стани. Сценарій показує, що відвідувач бачить до дії, як інтерфейс реагує після неї та що допомагає повернутися до роботи, якщо виникла проблема.

Однакові елементи мають працювати однаково. Кнопка з тим самим підписом повинна вести до тієї самої дії, поля форм — мати спільні правила підказок і помилок, а картки каталогу — послідовно показувати назву, властивості та доступні переходи.

Окремо спроектуйте ситуації, які часто залишають на етап розробки: неправильне введення даних, відправлення форми, недоступна дія, пошук без результатів, порожній каталог або недоступний товар. Якщо їх не узгодити, розробник буде змушений самотійно визначати поведінку інтерфейсу.

01 Складіть список повторюваних компонентів

Випишіть кнопки, поля, меню, картки, перемикачі, підказки та повідомлення, які повторюються на різних сторінках.

02 Опишіть призначення кожного елемента

Для компонента зафіксуйте, яку дію він запускає, які дані збирає або показує та який результат повинен отримати відвідувач.

03 Покажіть стани взаємодії

Для кнопок, полів і форм підготуйте звичайний стан, помилку, недоступну дію та результат відправлення. Для пошуку й каталогу покажіть також порожні результати.

04 Узгодьте назви дій

Перевірте, щоб однакові дії мали однакові підписи на всіх екранах і щоб назва кнопки не обіцяла результату, якого не буде.

05 Ведіть журнал рішень

Створіть таблицю з полями: сторінка або компонент, дія відвідувача, очікуваний результат, стан помилки, відповідальний і статус погодження. Вносьте туди кожне рішення, яке може по-різному трактуватися в дизайні та розробці.

ДЕ ВТРАЧАЮТЬ

Дорого обходиться припущення, що розробник «і так зрозуміє», що робити після помилки або відправлення форми. Саме в таких неописаних станах одна й та сама дія починає працювати по-різному на різних сторінках.

ЧЕКЛІСТ РОЗДІЛУ

- ☐ Для форм показано, які дані потрібні, навіщо вони потрібні та як виправити помилку.
Без цього відвідувач може не завершити дію, а команда не матиме узгодженого правила для повідомлень.
- ☐ Для пошуку, каталогу та доступності товарів або варіантів передбачено порожні стани.
Без цього проблемні ситуації з'являться вже в кодї без погодженого рішення.
- ☐ Повторювані компоненти мають однакові назви, вигляд і правила використання.
Без спільних правил інтерфейс виглядатиме зібраним із різних сайтів, а поведінку доведеться щоразу вгадувати.

Візуальна система потрібна не для прикрашання макета, а щоб зробити ролі елементів передбачуваними: заголовки читалися як заголовки, кнопки — як дії, а однакові блоки не змінювали значення через випадкове оформлення. Після цього слід окремо перевірити, чи зберігається доступність змісту на малих екранах.

Зафіксуйте правила для типографіки, кольорів, відступів, іконок, карток і зображень. Вони мають підтримувати ієрархію сторінки: показувати, що головне, що пояснює пропозицію, а що є додатковою інформацією.

Зображення повинні мати роль у сценарії: показувати продукт, процес або контекст послуги. Зображення без зв'язку зі змістом займає місце, але не допомагає відвідувачеві зрозуміти пропозицію.

Адаптивність не зводиться до зменшення десктопного екрана. На телефоні той самий зміст має лишатися доступним, але порядок блоків, меню та спосіб роботи з формою можуть вимагати окремого рішення.

01 **Задайте правила візуальної ієрархії**

Визначте стилі заголовків, основного тексту, посилань, кнопок і допоміжних повідомлень так, щоб їхня роль не залежала від випадкового кольору чи відступу.

02 **Уніфікуйте повторювані блоки**

Перевірте, що картки, кнопки, поля та заголовки використовують ті самі правила на всіх сторінках, де виконують однакову функцію.

03 **Перевірте зміст на мобільних екранах**

Окремо покажіть меню, порядок блоків, текст, кнопки та поля на малому екрані. Переконайтеся, що їх можна прочитати, натиснути й заповнити без зайвих дій.

04 **Перегляньте роль кожного зображення**

Залиште зображення, які допомагають пояснити продукт, процес або послугу. Для кожного іншого визначте, чи не займає воно місце важливішого змісту.

ДЕ ВТРАЧАЮТЬ

Зменшена копія десктопної сторінки на телефоні часто приховує проблему пріоритетів: текст стає важко читати, кнопки — складно натиснути, а меню не підтримує швидкий перехід до потрібного розділу.

ЧЕКЛІСТ РОЗДІЛУ

- ☐ **Визначено спільні правила для шрифтів, кольорів, відступів, іконок і зображень.**
Без системи сторінки можуть виглядати неузгоджено, а однакові елементи втратять впізнаваність.
- ☐ **Мобільні екрани мають окремо перевірені рішення для навігації, блоків і форм.**
Без цього важливий зміст або дія можуть стати недоступними на малому екрані.
- ☐ **Кожне зображення підтримує зміст сторінки, продукт або контекст послуги.**
Без перевірки візуальні матеріали займатимуть місце, не допомагаючи людині прийняти рішення.

05 · 5. ПРОВЕДІТЬ ПРИЙМАННЯ МАКЕТІВ І ПЕРЕДАЙТЕ ЇХ У РОЗРОБКУ

Передача макетів — це не момент, коли всі декоративні деталі доведені до межі. Макети можна передавати, коли вони не залишають ключові рішення на розсуд того, хто пише код: зрозумілі сторінки, переходи, компоненти, адаптація та поведінка у важливих станах.

Проведіть приймання за сценаріями, а не за загальним враженням. Людина, яка перевіряє макет, повинна мати змогу без пояснень сказати, що пропонує сторінка, де знайти потрібне, яку дію можна зробити та що станеться після неї.

Важливо відділити відповідальність. Бізнес або замовник підтверджує зміст, пропозицію, матеріали та пріоритети. Дизайнер фіксує структуру, компоненти, стани й адаптацію. Розробник перевіряє, як рішення реалізуються з урахуванням даних, інтеграцій і технічних обмежень.

Макет не замінює технічне завдання: він не описує всі інтеграції та правила роботи даних. Але він суттєво зменшує простір для різних трактувань, бо всі учасники бачать одну структуру й один сценарій.

01 **Перевірте перший екран**

Переконайтеся, що без прокручування зрозуміло, що пропонує сторінка, для кого вона та яку головну дію можна виконати.

02 **Пройдіть усі ключові сценарії**

Перевірте меню, переходи, картки, фільтри, форми й повідомлення так, ніби ви відвідувач. Для кожної дії визначте очікуваний результат.

03 **Зіставте дизайн з журналом рішень**

Перевірте, чи всі зафіксовані дії, стани помилок, порожні результати та відповідальні матеріали показані в макетах або явно позначені для реалізації.

04 **Позначте матеріали замовника**

Окремо виділіть тексти, фото, документи й дані, які має надати замовник. Не залишайте такі блоки без власника та статусу.

05 **Передайте пакет розробнику**

Передайте ключові сторінки, правила компонентів, мобільні екрани, переходи, стани форм і обмеження, що впливають на роботу інтерфейсу.

ДЕ ВТРАЧАЮТЬ

Якщо передати лише красиві екрани без переходів, станів і правил компонентів, ключові рішення будуть прийняті вже під час кодування. Це створює різне трактування задачі та повертає команду до погоджень у найдорожчий момент.

ЧЕКЛІСТ РОЗДІЛУ

- ☐ У макетах показані всі ключові сторінки та зрозумілі переходи між ними.
Без цього розробник не бачить повної логіки сайту й мусить припускати, що має бути між екранами.
- ☐ Для кнопок, полів і форм показані звичайний стан, помилка, недоступна дія та результат відправлення.
Без критеріїв приймання ці стани реалізують непослідовно або залишать без рішення.
- ☐ Погоджено, хто надає зміст, хто відповідає за дизайн-рішення і хто перевіряє технічні обмеження.
Без розподілу ролей матеріали та відповіді затримуються, а команда не розуміє, хто має закрити питання.
- ☐ Макет перевірено на десктопному й мобільному сценаріях до початку розробки.
Без цього проблеми навігації, читання та взаємодії виявляються вже після реалізації.

Чи достатньо показати розробнику макет головної сторінки?

Ні. Для розробки потрібні ключові сторінки, переходи між ними, повторювані компоненти, адаптивні рішення та стани інтерфейсу. Інакше ключові правила доведеться вигадувати під час кодування.

Чим макет відрізняється від технічного завдання?

Макет показує структуру, інтерфейс, компоненти та сценарії взаємодії. Технічне завдання описує, зокрема, інтеграції, дані й правила функціональності; макет не замінює його.

Як правильно використовувати приклади інших сайтів?

Не просіть просто зробити «як у них». Поясніть, що саме є корисним: структура меню, порядок блоків, спосіб порівняти варіанти, логіка форми або подача ціни.

Чи потрібен окремий дизайн, якщо використовується готовий шаблон?

Готовий шаблон уже містить дизайнерські рішення, але вони не обов'язково відповідають вашій структурі, бренду та сценаріям відвідувача. Все одно потрібно визначити пріоритети сторінок, зміст і цільові дії.

Хто має вирішувати, що відбувається після натискання кнопки?

Дизайн має зафіксувати поведінку кнопки в межах сценарію, а розробка реалізує її з урахуванням даних та інтеграцій. Якщо поведінка не описана, розробник буде змушений трактувати вимогу самостійно.

ЩО ДАЛІ

Обговорити з Apricode



apri-code.com/kontakty/