

Практичний посібник зі створення вебсайту

Як організувати роботу від бізнес-завдання до підтримки без
втрати сценаріїв, доступів і даних

У ЦЬОМУ ГАЙДІ

Вебсайт створюють не з макета і не з коду. Робота починається з того, яку дію має виконати відвідувач, які дані для цього потрібні та хто відповідатиме за контент, доступи й зміни після публікації. Нижче — порядок роботи, за яким команда може погодити рішення, перевірити результат і передати сайт у керування.

- 01 1. Зафіксуйте завдання до початку дизайну**
- 02 2. Побудуйте шлях користувача до того, як оформлювати інтерфейс**
- 03 3. Перетворіть погоджені сценарії на технічну реалізацію**
- 04 4. Прийміть сайт через реальні сценарії, а не через перегляд екранів**
- 05 5. Передайте сайт у керування та організуйте зміни після запуску**
- Q Часті питання**

01 · 1. ЗАФІКСУЙТЕ ЗАВДАННЯ ДО ПОЧАТКУ ДИЗАЙНУ

Перший результат роботи — не ескіз головної сторінки, а погоджене розуміння того, навіщо потрібен сайт і як ним користуватимуться різні люди. Це прибирає припущення, через які дизайн, функції та інтеграції доводиться переробляти вже після початку розробки.

Сформулюйте бізнес-завдання через конкретну дію відвідувача: залишити заявку, обрати послугу, прочитати матеріал, увійти до кабінету, записатися на програму або зв'язатися з відповідальною людиною. Формулювання на кшталт «потрібен сучасний сайт» не дає команді підстав визначити сторінки, поля форми, ролі чи поведінку після надсилання даних.

Для кожної групи користувачів опишіть маршрут: звідки людина приходить, що хоче знайти, яку інформацію має побачити перед рішенням і якою дією завершується її шлях. Одразу визначте, що має статися після дії: кому надходять дані, яке повідомлення бачить користувач і що відбувається, якщо виконати дію неможливо.

На цьому ж кроці зафіксуйте межі запуску. Відокремте сторінки, функції та інтеграції, без яких сайт не може почати роботу, від змін, які стануть окремими задачами. Це допомагає не перетворювати додавання одного блоку на непогоджену зміну структури, доступів або логіки форми.

01 Сформулюйте основну дію сайту

Запишіть, яку цільову дію має виконати відвідувач і який результат потрібен бізнесу. Для кожної важливої дії вкажіть відповідального, який отримує звернення або ухвалює наступне рішення.

02 Опишіть маршрути аудиторій

Для кожної аудиторії зафіксуйте початкову сторінку, потрібну інформацію, точки вибору та кінцеву дію. Не обмежуйтеся основним сценарієм: додайте випадки, коли користувач не має доступу або вводить некоректні дані.

03 Зберіть перелік вимог

Внесіть до одного документа сторінки, мови, контент, форми, поля, повідомлення, ролі користувачів, правила доступу, інтеграції та спосіб редагування матеріалів.

04 Призначте власників рішень

Окремо вкажіть, хто погоджує структуру, передає контент, відповідає за технічні налаштування, перевіряє функції та ухвалює зміни після запуску.

ДЕ ВТРАЧАЮТЬ

Найчастіше час і бюджет втрачаються, коли команда починає з фрази «зробіть як у конкурентів». Схожий вигляд не визначає, які дані передаються, хто має доступ і як повинна працювати конкретна дія.

ЧЕКЛІСТ РОЗДІЛУ

- ☐ Опишіть кожну ключову дію відвідувача від входу на сайт до результату.
Без маршруту команда реалізує окремі екрани, але не цілісний сценарій.
- ☐ Погодьте перелік сторінок, форм, ролей, мов і інтеграцій до підготовки макетів.
Без цього вимоги з'являтимуться вже після затвердження дизайну або в розробці.
- ☐ Зафіксуйте, що входить у перший запуск, а що стане окремою зміною.
Без меж робіт складно приймати результат і контролювати зміни.

Структура, прототип і дизайн вирішують різні задачі, тому їх не варто змішувати. Спочатку потрібно визначити місце кожної сторінки та шлях до цільової дії, а вже потім погоджувати вигляд компонентів.

Почніть із карти сайту. Вона показує основні та службові сторінки, зв'язки між ними, рівні навігації й точки входу до важливих дій. Якщо сторінка не має зрозумілого місця у структурі або до неї немає логічного шляху, дизайн цього не виправить.

Далі підготуйте прототипи. Вони мають показати порядок блоків, навігацію, переходи, поля форм і поведінку після натискання кнопок. Прототип перевіряє не естетику, а послідовність дій: що відвідувач бачить спочатку, де отримує деталі, як повертається назад і що відбувається у нестандартній ситуації.

Лише після погодження логіки переходьте до дизайну. У ньому визначаються типографіка, кольори, відступи, кнопки, поля, повідомлення та стани компонентів. Для кожної взаємодії мають бути передбачені щонайменше стан помилки, успішна дія, завантаження або недоступна функція.

01 Намалюйте карту сайту

Розташуйте сторінки за рівнями: основні розділи, сторінки деталей, контакти, матеріали, кабінет і службові сторінки. Позначте, з яких сторінок користувач переходить до заявки, входу або іншої цільової дії.

02 Перевірте навігацію для кожної аудиторії

Пройдіть маршрути, описані на попередньому етапі. Переконайтеся, що користувач може знайти потрібну інформацію без випадкових переходів між непов'язаними сторінками.

03 Створіть прототипи сценаріїв

Покажіть не лише звичайний шлях, а й усі ключові переходи, поля форм, підтвердження та повідомлення. Прототип має пояснювати, що бачить користувач після кожної дії.

04 Опишіть стани компонентів

Для кнопок, полів, повідомлень і закритих матеріалів визначте поведінку в разі помилки, успіху, завантаження та відсутності доступу.

05 Погодьте дизайн як систему елементів

Передавайте в розробку не окремі красиві екрани, а узгоджені правила для компонентів і їхніх станів. Це зменшує ризик, що одна й та сама дія виглядатиме або працюватиме по-різному на різних сторінках.

ДЕ ВТРАЧАЮТЬ

Вигляд форми не означає, що форма описана. Якщо в макеті немає валідації, повідомлення про помилку та підтвердження успішного надсилання, розробка буде змушена припускати поведінку самостійно.

ЧЕКЛІСТ РОЗДІЛУ

- ☐ Переконайтеся, що кожна сторінка має місце у карті сайту та зрозумілий зв'язок з іншими сторінками.

Без цього на сайті з'являються сторінки, до яких відвідувач не може логічно дістатися.

- ☐ Пройдіть у прототипі основний і недоступний сценарії для кожної важливої дії.

Без цього готовий інтерфейс може не пояснювати користувачу, чому дія не виконалася.

- ☐ Передайте правила станів кнопок, форм і повідомлень разом із дизайном.

Без правил різні частини інтерфейсу працюватимуть непослідовно.

Розробка має відтворити погоджені сценарії, а не просто зверстати макети. Для цього команда повинна окремо контролювати інтерфейс, правила роботи з даними, доступи та обмін із зовнішніми сервісами.

Фронтенд відповідає за те, що бачить відвідувач: сторінки, навігацію, форми, повідомлення та зміну станів інтерфейсу. Він реалізує структуру вебконтенту, стилізацію та динамічну реакцію на дії користувача.

Серверна частина працює з даними, бізнес-логікою, авторизацією, ролями та правами доступу. Саме тут мають перевірятися правила: хто може переглядати матеріал, кому доступна дія та які дані можна обробити. Не можна покладати такі рішення лише на видиму частину інтерфейсу.

Кожну інтеграцію опишіть як окремий сценарій. Назви сервісу недостатньо: потрібно вказати, які дані передаються, хто запускає передачу, де зберігаються налаштування, хто має до них доступ, який результат бачить користувач і як сайт поводить, якщо зовнішній сервіс не відповідає.

01 **Передайте в розробку узгоджений пакет**

До старту використовуйте погоджені вимоги, карту сайту, прототипи, дизайн, перелік контенту та сценарії інтеграцій. Якщо рішення немає в цьому пакеті, зафіксуйте його окремо до реалізації.

02 **Розділіть відповідальність фронтенду й серверної частини**

Визначте, які елементи відображає інтерфейс, а де перевіряються дані, авторизація, доступи та бізнес-правила. Це допомагає не змішувати візуальну поведінку з контролем прав.

03 **Опишіть контракт кожної форми**

Для форми зафіксуйте поля, обов'язковість, перевірку значень, повідомлення, одержувача даних і результат успішного або невдалого надсилання.

04 **Оформіть сценарії інтеграцій**

Для кожного зовнішнього сервісу запишіть джерело даних, момент передачі, відповідального за доступи та поведінку сайту у випадку помилки.

05 **Ведіть журнал рішень і змін**

Створіть робочу таблицю з полями: дата, зміна, причина, сторінки або сценарії, які зачіпає зміна, відповідальний, статус перевірки та рішення про публікацію.

ДЕ ВТРАЧАЮТЬ

Інтеграція часто виглядає завершеною, поки не виникає помилка зовнішнього сервісу. Якщо цей випадок не описаний заздалегідь, користувач може не отримати результату, а команда — не зрозуміти, де зупинилися дані.

ЧЕКЛІСТ РОЗДІЛУ

- ☐ Перевірте, що правила доступу реалізуються на серверній частині, а не лише приховані в інтерфейсі.
Без цього закритий матеріал або дія можуть бути доступні не тим користувачам.
- ☐ Опишіть для кожної інтеграції дані, відповідального, успішний результат і помилку.
Без цього зовнішня залежність стане причиною неочевидних збоїв після запуску.
- ☐ Фіксуйте зміни в журналі до їх реалізації.
Без історії змін складно зрозуміти, чому змінилася поведінка сторінки або сценарію.

04 • 4. ПРИЙМІТЬ САЙТ ЧЕРЕЗ РЕАЛЬНІ СЦЕНАРІЇ, А НЕ ЧЕРЕЗ ПЕРЕГЛЯД ЕКРАНІВ

Сайт готовий до публікації тоді, коли погоджені дії працюють у реальному середовищі, а не тоді, коли сторінки виглядають завершеними. Приймання потрібно організувати як послідовну перевірку маршрутів, даних, доступів, медіа та зовнішніх зв'язків.

Основою перевірки є сценарії користувача. Пройдіть весь шлях: відкрийте потрібну сторінку, знайдіть інформацію, заповніть форму або увійдіть до кабінету, отримайте результат і перевірте, чи дані дійшли до потрібного сервісу або відповідальної людини.

Окремо тестуйте негативні сценарії. Залиште обов'язкове поле порожнім, введіть некоректне значення, спробуйте відкрити закритий матеріал без доступу, перевірте сторінку без контенту та ситуацію, коли інтеграція не передає дані. У кожному випадку сайт має дати зрозумілий результат, а не мовчазно зупинитися.

Перевірка має охопити навігацію, внутрішні посилання, форми, повідомлення, ролі, авторизацію, медіа, відображення на різних екранах, продуктивність та інтеграції. Результати фіксуйте в журналі перевірки, а не збирайте в особистих повідомленнях.

01 Складіть перелік сценаріїв приймання

Для кожної ключової дії запишіть передумову, кроки, очікуваний результат, фактичний результат, відповідального та статус. Використовуйте окремі рядки для успішного й помилкового сценаріїв.

02 Перевірте маршрути й контент

Відкрийте сторінки через меню та внутрішні посилання, звірте тексти, зображення, назви сторінок, контакти й матеріали. Переконайтеся, що немає маршрутів на сторінки без потрібного контенту.

03 Протестуйте форми як користувач

Перевірте обов'язкові поля, некоректні значення, успішне надсилання та повідомлення про результат. Після цього підтвердьте, що дані надійшли до потрібного сервісу.

04 Перевірте доступи та ролі

Увійдіть під різними ролями або без авторизації. Переконайтеся, що закриті сторінки, матеріали та дії доступні лише тим, кому вони призначені.

05 Перевірте інтеграції в обох станах

Пройдіть сценарій успішної передачі даних і сценарій помилки зовнішнього сервісу. Зафіксуйте, що бачить користувач і як команда дізнається про проблему.

06 Проведіть фінальне приймання перед відкриттям доступу

Повторіть ключові маршрути на середовищі, де сайт буде доступний відвідувачам.
Публікуйте результат лише після закриття критичних помилок у журналі.

ДЕ ВТРАЧАЮТЬ

Перевірка лише візуального вигляду не знаходить зламані форми, неправильні права доступу та помилки передачі даних. Після публікації такі проблеми бачать відвідувачі, а не команда.

ЧЕКЛІСТ РОЗДІЛУ

- ☐ Ведіть журнал перевірки з очікуваним і фактичним результатом кожного сценарію.
Без журналу помилки губляться, дублюються або залишаються без відповідального.
- ☐ Перевірте успішний і помилковий сценарії для форм та інтеграцій.
Без негативних сценаріїв сайт може не пояснювати користувачу, що сталося.
- ☐ Перевірте закриті матеріали без авторизації та під різними ролями.
Без цього можна відкрити доступ до матеріалів або функцій не тим людям.
- ☐ Пройдіть ключові дії на різних екранах перед запуском.
Без цього інтерфейс може бути незручним або недоступним для частини відвідувачів.

Публікація відкриває доступ до сайту, але не завершує роботу з ним. Щоб сайт залишався робочим, команді потрібен порядок оновлення контенту, контролю доступів, перевірки інтеграцій і погодження змін.

До передачі визначте ролі: хто редагує тексти й зображення, хто додає матеріали, хто може змінювати структуру, хто переглядає заявки, а хто керує доменом, поштою та налаштуваннями зовнішніх сервісів. Власність і доступи не повинні залежати від однієї людини або зберігатися лише в особистих повідомленнях.

Зберіть в одному зрозумілому місці вихідні макети, файли, інструкції, дані для відновлення доступу та правила подання змін. Для кожної зміни використовуйте однаковий порядок: опис задачі, оцінка того, які сторінки, доступи або інтеграції вона зачіпає, погодження, реалізація, перевірка та фіксація результату.

Після запуску регулярно перевіряйте форми, доступи, актуальність матеріалів і зв'язок із зовнішніми сервісами. Підтримка не означає постійно змінювати сайт; вона створює контрольований спосіб робити потрібні зміни без втрати даних, функцій і відповідальності.

01 Розподіліть ролі керування

Призначте окремих відповідальних за контент, технічні рішення, заявки, доступи та погодження змін. Зафіксуйте межі кожної ролі, щоб редактор не мав випадково змінювати технічні налаштування.

02 Передайте доступи та матеріали

Зберіть доступи до домену, пошти, сайту й зовнішніх сервісів, а також вихідні макети, файли та інструкції. Визначте порядок відновлення доступу, якщо хтось втрачає дані для входу.

03 Запровадьте запит на зміну

Кожна зміна повинна містити мету, опис, відповідального, сторінки або сценарії, яких вона стосується, і результат перевірки. Не починайте зміни лише з усного прохання або повідомлення без фіксації.

04 Перевіряйте критичні сценарії після оновлень

Після змін у формах, доступах, структурі або інтеграціях повторно проходите пов'язані сценарії. Це допомагає знайти наслідки правки до того, як їх помітять користувачі.

05 Підтримуйте актуальність контенту й доступів

Регулярно переглядайте контакти, матеріали, ролі та зовнішні зв'язки. Видаляйте або коригуйте доступи, які більше не відповідають відповідальності людей.

ДЕ ВТРАЧАЮТЬ

Коли доступи до домену, пошти або зовнішніх сервісів прив'язані до однієї людини, будь-яка її недоступність може зупинити оновлення, відновлення входу або роботу інтеграцій.

ЧЕКЛІСТ РОЗДІЛУ

- ☐ **Зафіксуйте власників контенту, доступів, технічних рішень і погодження змін.**
Без відповідальних важливі оновлення залишаються без виконання або дублюються.
- ☐ **Зберігайте доступи, вихідні матеріали та інструкції в узгодженому місці.**
Без цього бізнес стає залежним від особистих повідомлень і пам'яті окремої людини.
- ☐ **Перевіряйте форми, ролі й інтеграції після змін у пов'язаних сценаріях.**
Без повторної перевірки невелика правка може зламати робочу дію.

ЧАСТІ ПИТАННЯ

Чи можна почати створення сайту з дизайну?

Дизайн варто починати після фіксації цілей, вимог і структури. Він визначає вигляд інтерфейсу, але не може самостійно вирішити, які сторінки потрібні, кому доступний матеріал або що має статися після надсилання форми.

Який мінімальний набір потрібно погодити до розробки?

Потрібно погодити бізнес-завдання, маршрути аудиторій, перелік сторінок, контент, форми, ролі, доступи, мови, інтеграції та спосіб редагування матеріалів після публікації. Також важливо визначити, що входить у запуск, а що стане окремою задачею.

Як правильно перевіряти форму перед запуском?

Пройдіть її як користувач: залиште обов'язкове поле порожнім, введіть некоректне значення, надішліть правильні дані та перевірте повідомлення про результат. Після успішного надсилання переконайтеся, що дані дійшли до потрібного сервісу або відповідальної людини.

Хто має володіти доступами до сайту та пов'язаних сервісів?

Доступи мають бути розподілені між відповідальними ролями, а порядок їх відновлення — зафіксований. Важливо, щоб домен, пошта, сайт та інтеграції не залишалися доступними лише одній людині.

Чи потрібно тестувати сайт після кожної зміни?

Потрібно повторно перевіряти сценарії, яких стосується зміна. Особливо це важливо після правок у формах, доступах, навігації, структурі контенту та інтеграціях.

ЩО ДАЛІ

Обговорити з Apricode



apri-code.com/kontakty/