Лабораторна Робота №5

Розгортання статичного сайту

Мета: ознайомитися з основними поняттями та синтаксисом JavaScript. Навчитися оголошувати змінні, працювати з базовими типами даних та операціями над ними. Засвоїти основні структури керування програмою (умовні оператори, цикли). Навчитися взаємодіяти з елементами HTML (DOM) за допомогою JavaScript. Зрозуміти основи обробки подій (Events) у браузері.

Хід роботи

1. Створити акаунт на сайті *github.com*
2. Створити порожній публічний репозиторій та завантажити в нього свій веб сайт(папка із кодом). Посилання на репозиторій та його скріншот додати до лабораторної роботи.
3. Створити акаунт на сайті *vercel.com* та згідно з інструкціями опублікувати власний сайт в мережі. Посилання на опублікований сайт та його скріншот додати до звіту лабораторної роботи.
4. Зробити будь-які зміни вихідного коду, що змінюють зовнішній вигляд сайту та закомітати їх, посилання на ваш коміт додати до лабораторної роботи. Переконатися, що після коміту опублікований сайт автоматично оновився.
5. (Опційно) Придбати власний домен та спробувати розібратися як підключити його до свого сайту.

Вказівки для виконання роботи

Для того, щоб зробити ваш сайт доступним у мережі, потрібен сервер. Проте, є безліч сервісів, які дозволяють опублікувати в мережі статичний сайт (html, css, javascript) безкоштовно. У цій роботі ми використаємо Vercel, оскільки він дозволяє підключити свій обліковий запис GitHub, та автоматично оновлювати сайт при змінах вихідного коду на GitHub.

Перед тим як приступати до виконання роботи, переконайтеся що вихідний код вашого сайту є доступний у одному з ваших репозиторіїв, а також, що ваш файл index.html знаходиться у кореневій папці репозиторію.

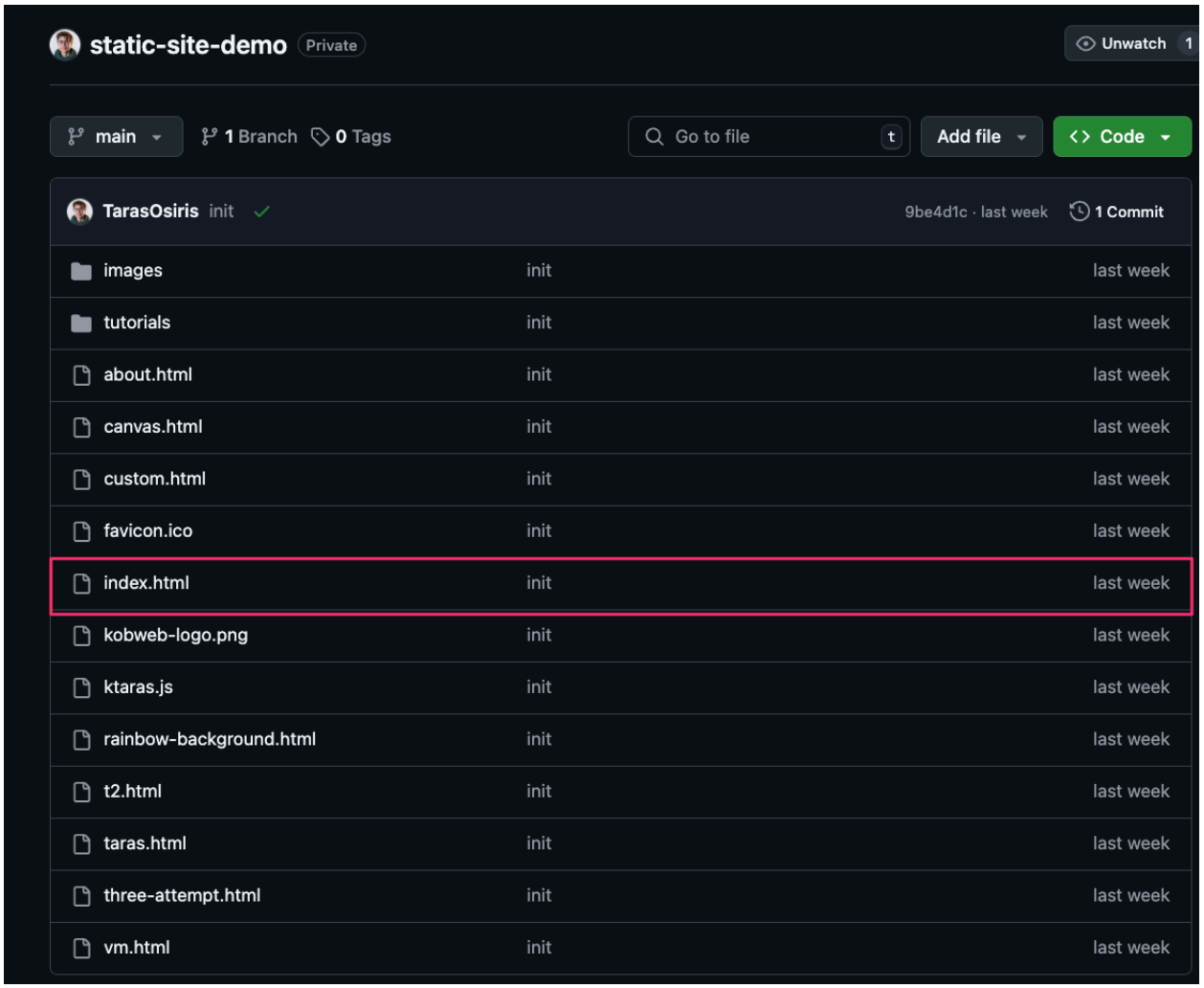

Рис 1. Приклад виихідного коду статичного сайту на GitHub.com

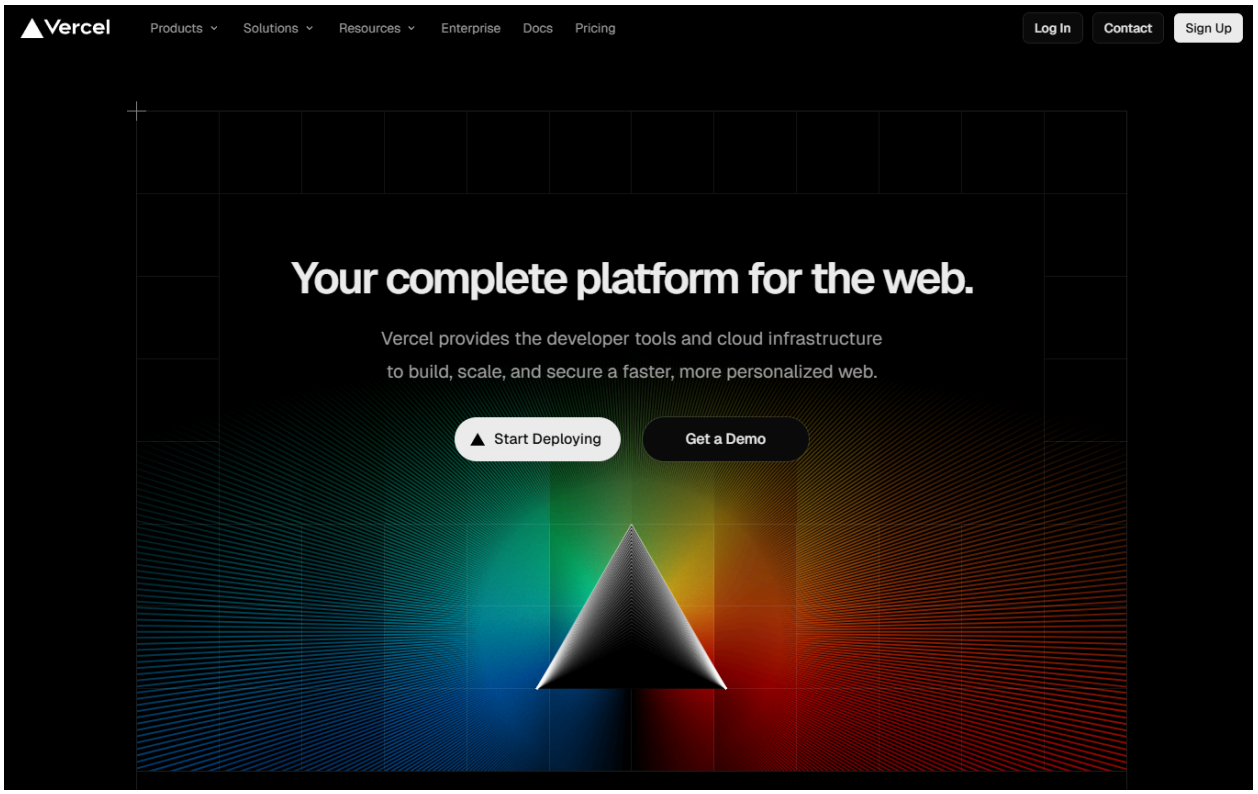

Рис 2. Головна сторінка vercel.com

Для початку, необхідно зайти на vercel.com та створити обліковий запис.

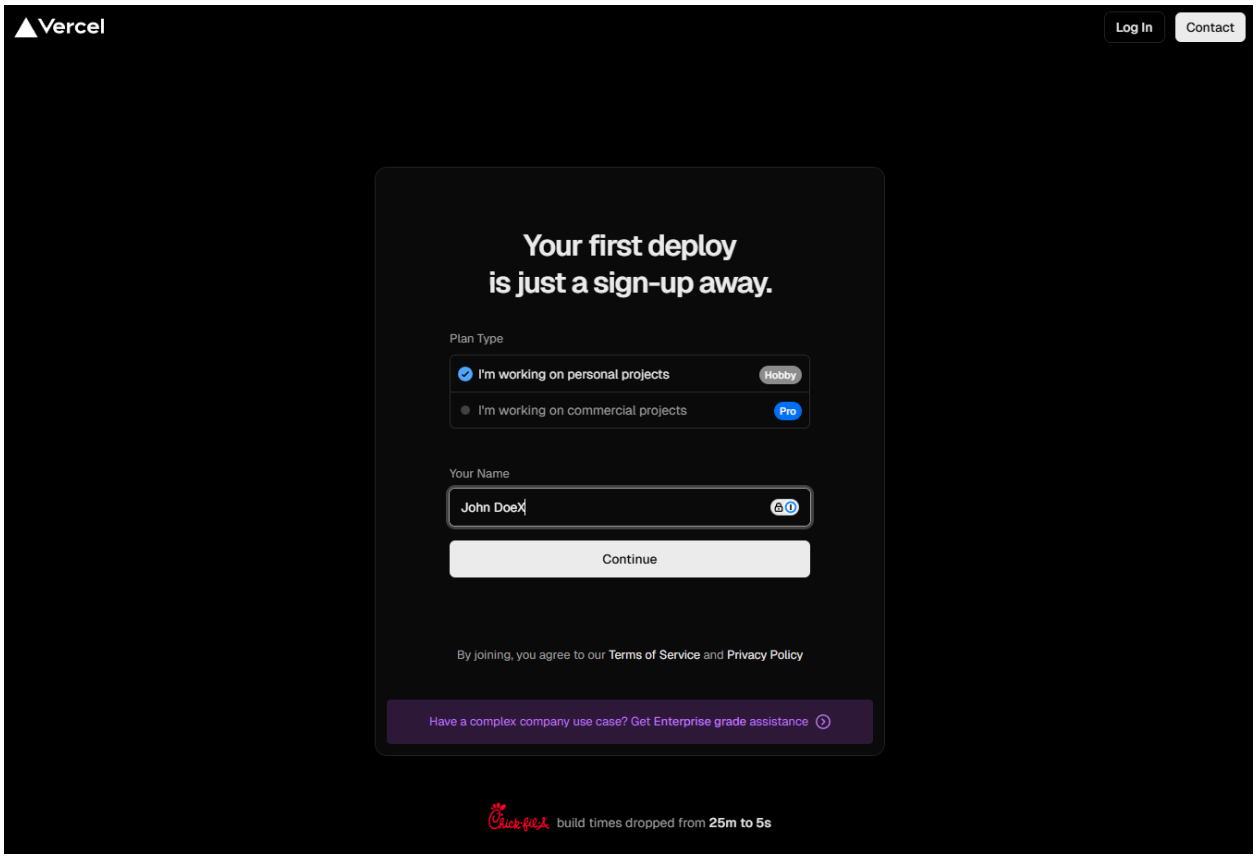

Рис 3. Реєстрація на сайті vercel.com

При реєстрації, ви можете одразу приєднати свій GitHub account, рекомендовано це одразу зробити, проте це можна зробити і пізніше в налаштуваннях.

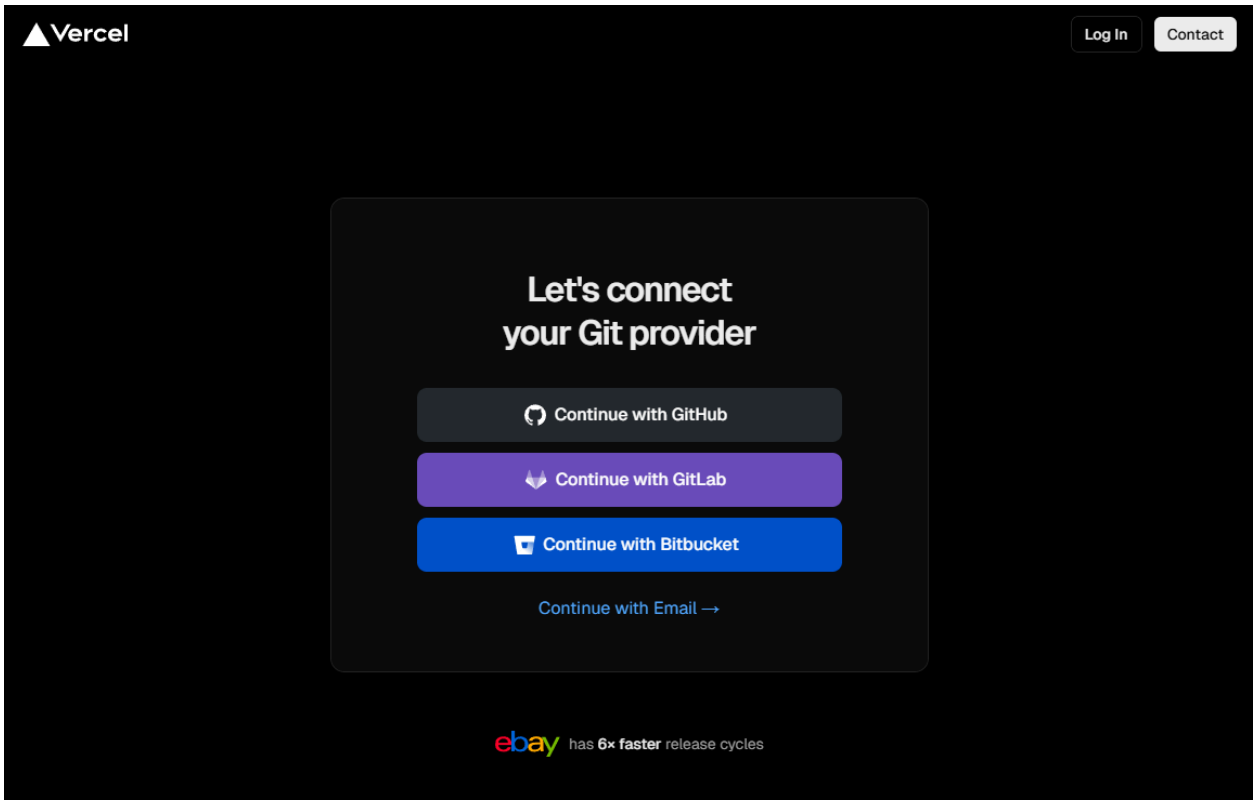

Рис 4. Модальне вікно для прикріплення облікового запису GitHub

Після цього, створюємо новий проект **Add New -> Project** та в наступному кроці обираємо репозиторій в якому знаходиться ваш сайт.

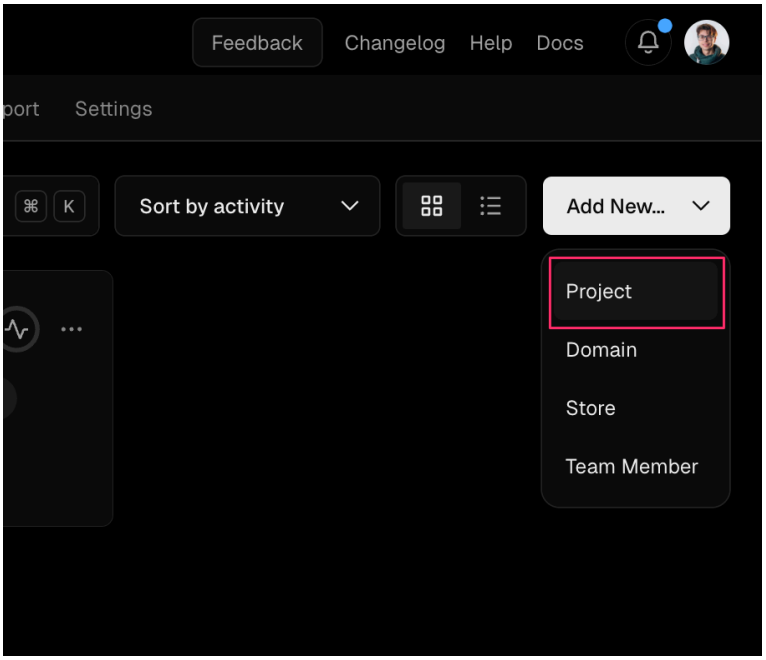

Рис 5. Приклад створення нового проекту

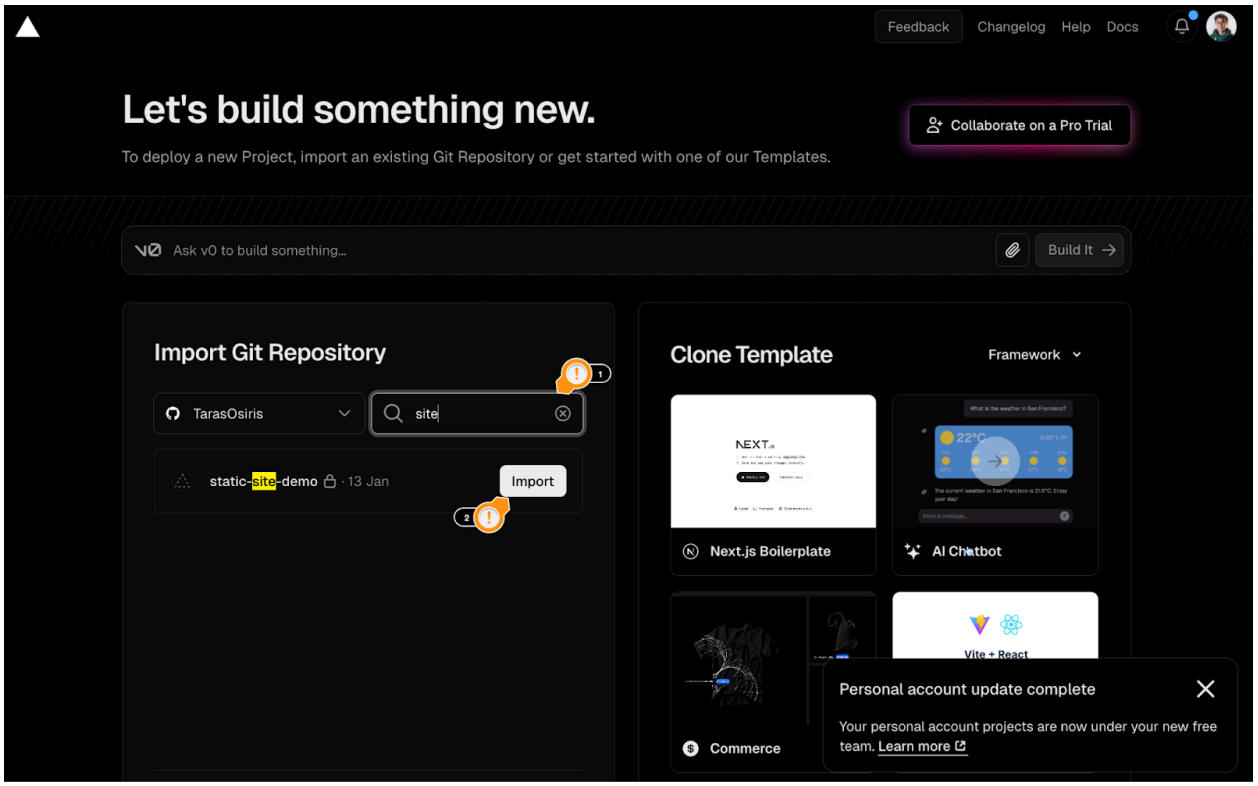

Рис 6. Вибір репозиторію звідки буде братися вихідний код

Далі налаштовуємо параметри розгортання. В розділі **Framework Preset** залишаємо значення **Other** оскільки ми не використовуємо ніякі фреймворки. Root directory вказуємо теку, в якій знаходиться файл index.html, у нашому випадку це корінь папки, тому це значення залишаємо без змін. Після вибору налаштувань, натискаємо кнопку **Deploy**.

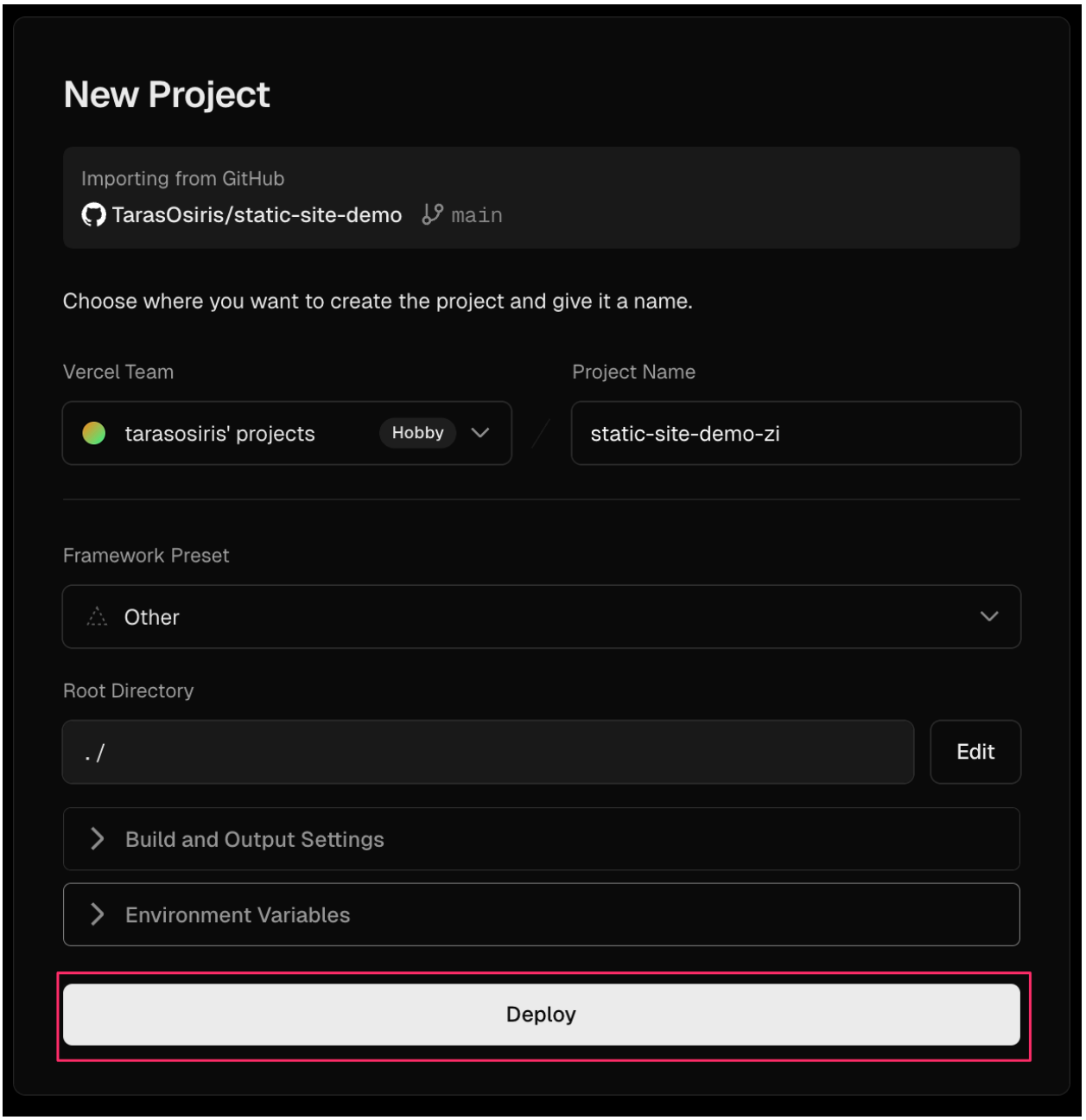

Рис 7. Розгортання проекту

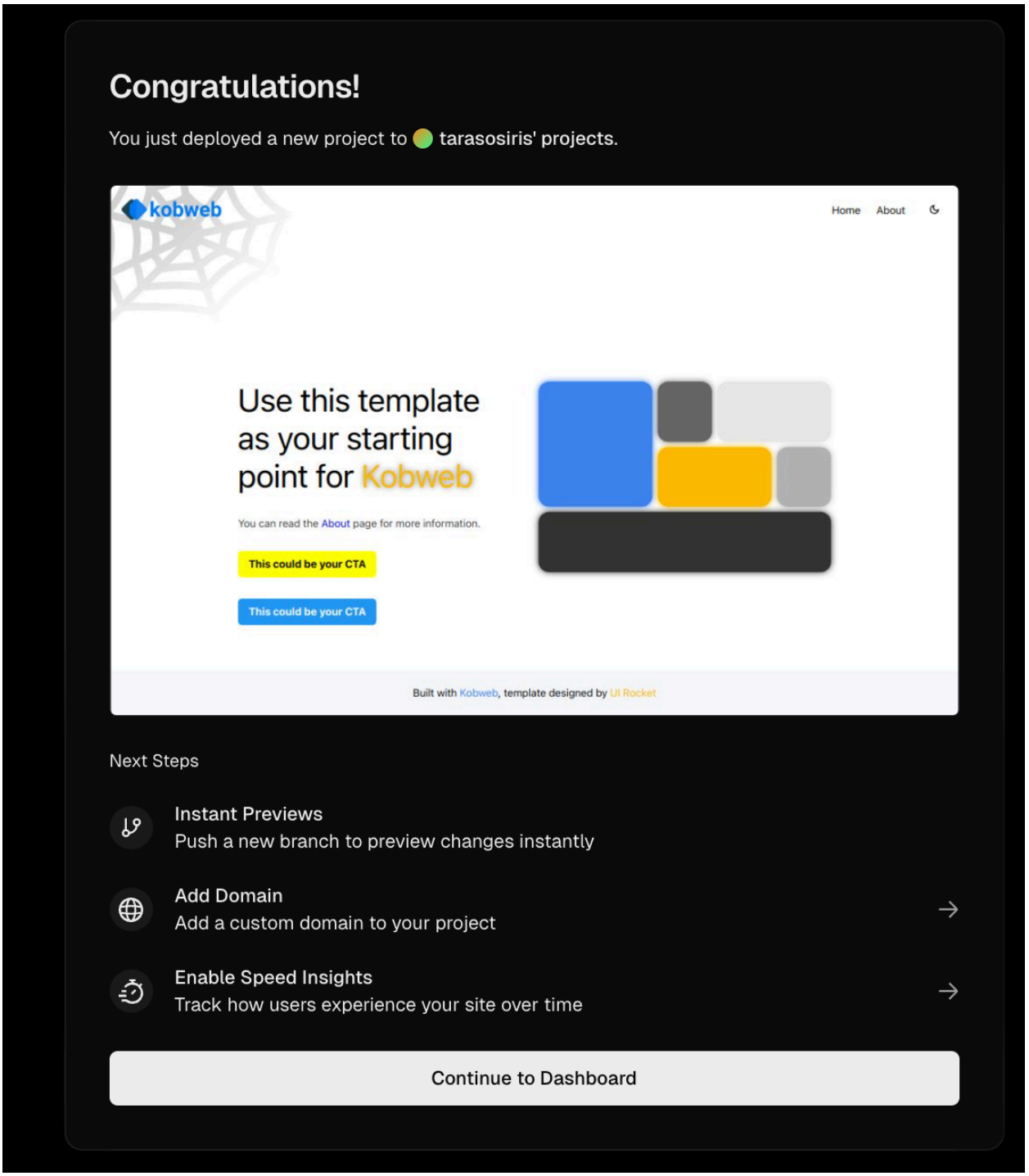

Рис 8. Повідомлення про успішне розгортання сайту

Готово! Тепер можемо перейти назад до панелі керування (**Continue to dashboard**), щоб дізнатися домен сайту, через який він тепер доступний у мережі. Також, коли ви робите зміни в репозиторії, vercel автоматично буде сповіщений про це і ваш сайт буде автоматично оновлений (спробуйте дізнатися як саме працює така інтеграція). Як додаткове завдання, якщо у вас є власний домен, ви можете спробувати підключити його до вашого нового сайту.

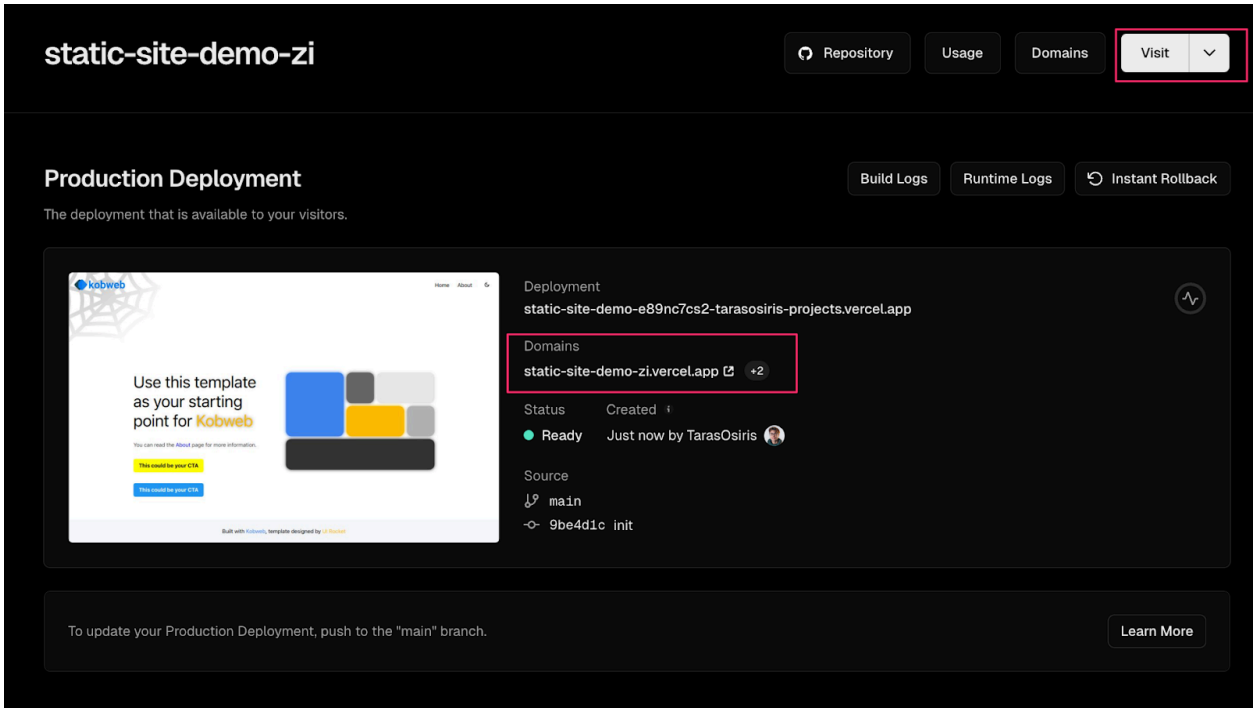

Рис 9. Готовий розгорнутий статичний сайт в мережі

Питання для самоконтролю

1. Які існують способи оголошення змінних у JavaScript та чим вони відрізняються?
2. Які типи даних підтримує JavaScript? Як відрізнити прості (primitive) та складні (reference) типи даних?
3. Поясніть, як працює оператор порівняння == та ===, в чому різниця та коли доцільно використовувати кожен із них?
4. Які основні структури керування використовують у JavaScript?
5. Чим відрізняється *Function Declaration* від *Function Expression* та *Arrow Function*? Наведіть приклади оголошення кожного типу функцій.
6. Які стандартні методи обробки масивів ви знаєте? Коротко опишіть призначення кожного.
7. Яким чином у JavaScript створюються та змінюються об’єкти? Як звернутися до властивостей та методів об’єкта?
8. Як отримати доступ до елементів HTML (DOM) з JavaScript та змінити їх властивості чи вміст?
9. Що таке *події (events)* у JavaScript? Як додавати слухачі подій (event listeners) та які бувають основні події?
10. Чим корисний метод *console.log* під час розробки та відлагодження JavaScript-коду і які інші інструменти налагодження ви знаєте?