

💋 Створюємо сайт на платформі Vercel

Ж Що нам знадобиться:

- Ø Обліковий запис на GitHub Для завантаження файлів у Vercel.
- 🗀 Система управління версіями GIT Для завантаження файлів на GitHub.
- ■ IDE Зручне середовище для написання коду.

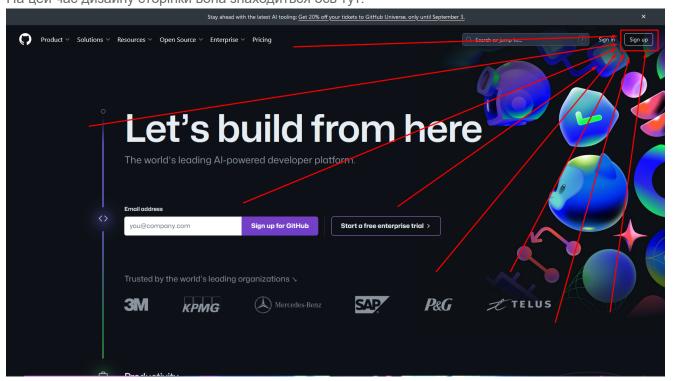
← Почнемо з GitHub:

GitHub — це популярний сервіс для зберігання коду, де можна працювати над проєктами разом з іншими.

Це ніби соціальна мережа для програмістів, де ти можеш показати свій код, подивитися на чужий і разом створювати щось круте. Але на разі зосередимо увагу на нашій задачі:

Створення облікового запису на GitHub:

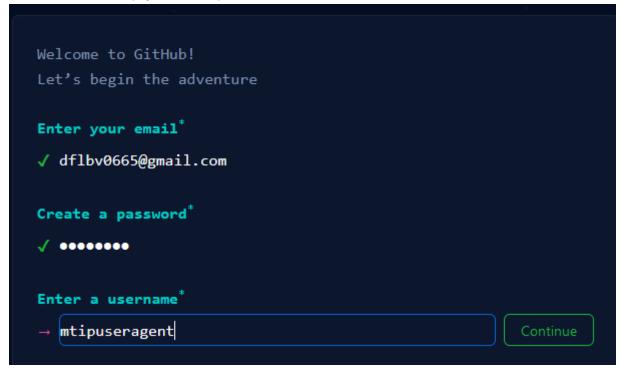
- 1. (1) Перейди по посиланню: github.com Здається, це найскладніше завдання! Далі просто потрібно зареєструватись.
- 2. **Знайди кнопку** Sign Up : На цей час дизайну сторінки вона знаходиться ось тут:



Натисни.

Заповни дані:

Далі все по стандарту: email, пароль, логін:



4. 🗑 Перевірка на людину:

Добре, зайти на сайт було не складно (кому-як), але далі найскладніше: ДОВЕДИ, що ти не робот.

5. Підтвердження пошти:

Також забув, сказати використовуй справжню почту, бо треба пройти процес підтвердження.

🥟 Вітаю! Ти створив свій аккаунт на GitHub

Далі можеш заповнити інформацію про себе. Але це не дуже важливо.

С Система управління версіями GIT:

Git — це інструмент, який допомагає відстежувати всі зміни в коді.

Уяви собі машину часу, яка дозволяє повернутися в минуле, якщо щось пішло не так, або поділитися своїм кодом з друзями, щоб разом працювати над проєктом.

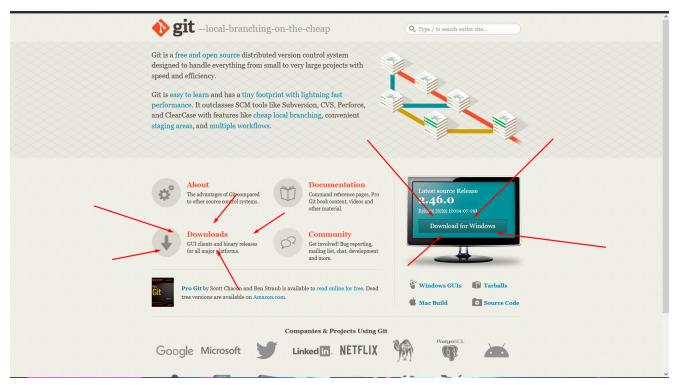
% Встановлення GIT:

1. () Переходимо до сайту: git-scm.com

Це офіційний сайт для завантаження GIT. Відкрий його у своєму браузері.

2. 📥 Обери свою операційну систему:

У верхній частині сторінки знайди кнопку, яка відповідає твоїй операційній системі (Windows, macOS, Linux):



Завантаження:

Натисни на кнопку завантаження. Після цього автоматично розпочнеться завантаження інсталяційного файлу GIT.

4. 🗀 Встановлення:

Після завершення завантаження, відкрий інсталяційний файл і дотримуйся інструкцій на екрані. Під час встановлення залишай стандартні налаштування.



Q до речі: якщо ти просунутий користувач ПК, можеш встановити GIT за допомогою терміналу

(PowerShell ado cmd):

winget install --id Git.Git -e --source winget

↑ Для користувачів Linux:

Aгов хлопці, ви на Linux, самі знаєте як встановлювати пакети на вашому дистрибутиві. Для Debian: apt-get install git

Перевірка:

Після встановлення, відкрий командний рядок або термінал і введи команду:

git --version

Якщо все пройшло вдало, ти побачиш версію GIT на своєму екрані.

🦫 Вітаю! GIT успішно встановлений!

Тепер ти готовий використовувати GIT для управління версіями свого коду.

IDE та налаштування проєкт:

1. **Відкрий своє IDE та створи новий проєкт.**

Можливість використання GIT:

Якщо не хочеш все робити вручну, можеш використати GIT і ввести в терміналі команду:

git clone https://github.com/VadyKnyfy/skeleton_for_easily_vercel.git

Щ Це лише початок:

Тут я розповідаю про те, як створити найпростіший проєкт, тому подано лише 2% інформації. Сподіваюся, що це зацікавить вас, і ви продовжите дізнаватися більше про світ IT.

2. **Створіть у корені проєкту два каталоги:** public та api . Це будуть два основних каталоги: public для статичних файлів, усі файли будуть викликатися з цієї папки, арі — для серверних.

```
✓ ☐ readme C:\Users\vadyk\Desktop\readme☐ api☐ public
```

3. Створіть файл vercel.json:

Додайте до нього наступний код (магічні символи):

```
{
    "functions": {
        "api/**/*.php": {
            "runtime": "vercel-php@0.6.1"
      }
    },
    "routes": [
      {
        "src": "/api/(.*)",
        "dest": "/api/$1"
      }
    ]
}
```

% Functions:

• Functions використовуються для виконання серверних скриптів. Це дуже корисна річ! Більше інформації тут:

Vercel Functions.

Routes:

• **Routes** використовуються для перенаправлення запитів, щоб отримати файли з будь-якої вигаданої адреси на фактичну адресу файлу. Як це працює:

Уяви, що у нас є документ за адресою /public/src/dangers/dota2.html . Замість того, щоб використовувати таку URL-адресу: http://наш_сайт/src/dangers/dota2.html (де public не включено, бо Vercel шукає файли там за замовчуванням), ми можемо створити коротшу URL-адресу: http://наш_сайт/league_of_legends .

Щоб це зробити потрібно додати ось такий код(магічні символи) у масив routes :

```
{
  "src": "/league_of_legends",
  "dest": "/src/dangers/dota2.html"
}
```

- src : бажана адреса для переходу до документа.
- dest : фактична адреса до документа.

Зараз ми налаштували маршрути так, щоб усі документи з адресою, що починається з /арі/..., шукалися в директорії арі , яку ми створили раніше.

4. Створи файл package.json:

Корисний файл, але нами він буде використовуватись лише для того, щоб встановити версію, ядра, тож введи в нього, ось такий код(магічні символи):

```
{ "engines": { "node": "18.x" } }
```

5. Створимо файл, для перевірки: Перейди до каталогу public та створи файл index.html, та заповни його базовою структурою html, використовуючи магічні символи, або код(мій тестовий документ):

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>Найпростіший Vercel проект</title>
    <body>
        <h1>
            Усе працює
        </h1>
        >
            Сторінка створена з допомогою магії
        </body>
 </html>
```

Перейди до каталогу арі та створи файл іndex.php , заповни його цим:

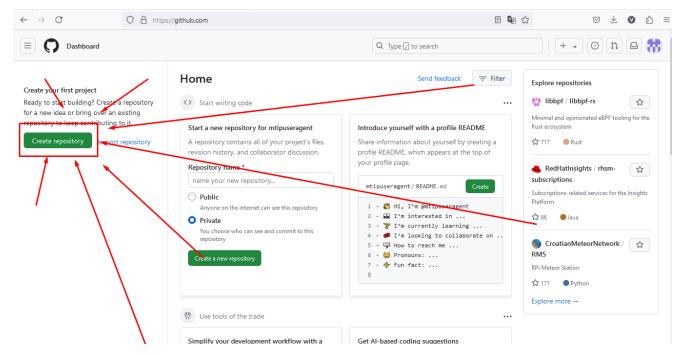
```
<?php
phpinfo();
?>
```

🏂 Вітаю! Проєкт успішно створено!

Далі нас очікує завантаження, його на GITHUB, та довгождана інтеграція з Vercel

Створення нового репозиторію на GITHUB та завантаження проєкт:

- 1. Перейди на сторінку GitHub
- 2. Якщо в тебе ще немає жодного репозиторію, шукай кнопку Create repository, як показано на рисунку:



Натискай її.

3. Далі потрібно заповнити форму. Обов'язково введи лише назву репозиторію, усе інше можеш залишити так, як є:

Create a new repository

| create a new repository |
|---|
| A repository contains all project files, including the revision history. Already have a project repository elsewhere? |
| Import a repository. |
| Required fields are marked with an asterisk (*). |
| Owner * Popository name * |
| # mtipuseragent ▼ / |
| Great repository names are short and memorable. Need inspiration? How about glowing-potato ? |
| Description (optional) |
| |
| |
| O Public Anyone on the internet can see this repository. You choose who can commit. |
| |
| O Private You choose who can see and commit to this repository. |
| Initialize this repository with: |
| ☐ Add a README file |
| This is where you can write a long description for your project. <u>Learn more about READMEs.</u> |
| Add .gitignore |
| .gitignore template: None 🔻 |
| Choose which files not to track from a list of templates. <u>Learn more about ignoring files.</u> |
| Choose a license |
| License: None 🔻 |
| A license tells others what they can and can't do with your code. <u>Learn more about licenses.</u> |
| (i) You are creating a public repository in your personal account. |
| Communication . |

Потім натисни на кнопку Create repository .

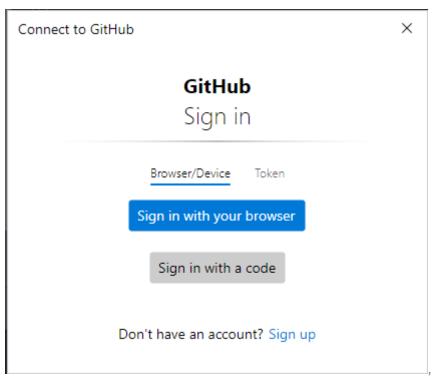
Репозиторій створено

4. Далі скопіюй вміст поля з написом …or create a new repository on the command line :

...or create a new repository on the command line

```
echo "# test_vercel_repo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/mtipuseragent/test_vercel_repo.git
git push -u origin main
```

Після того як скопіював, встав скопійований текст у термінал. **Переконайся, що в терміналі** відкритий корінь твого проєкту. Очікуй, на появлення вікна для авторизації Git на GitHub:



або авторизація пройде у

терміналі де потрібно ввести дані для входу у твій профіль GitHub

5. Тепер твій проєкт зареєстрований у системі GIT, але ще треба завантажити файли. Введи у терміналі наступні команди:

```
    git add --all
    git commit -m "init"
    git push -u origin main
```

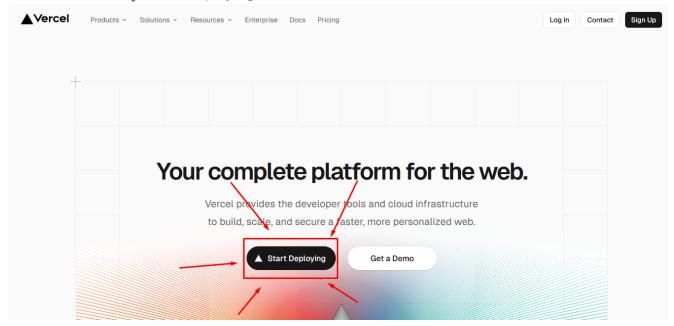
Перейди на сторінку GitHub свого репозиторію та онови сторінку, якщо ти бачиш файли, які ти створив у себе на ПК...

🏂 Вітаю! Проєкт успішно завантажений на GitHub!

Ø Розгортання проєкту на Vercel:

1. (1) Перейди на сторінку Vercel

2. ♦ Натисни кнопку Start Deploying:

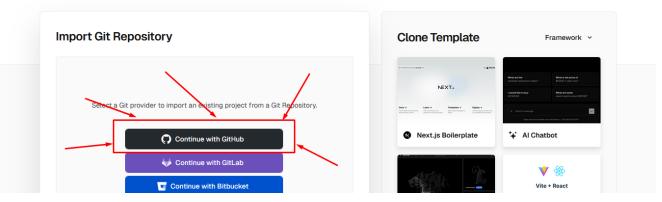


3. ♦ Натискай кнопку Continue with GitHub:

Contact Login Up

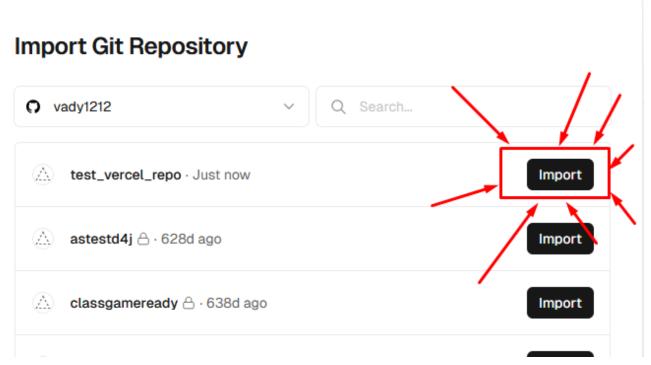
Let's build something new.

To deploy a new Project, import an existing Git Repository or get started with one of our Templates.

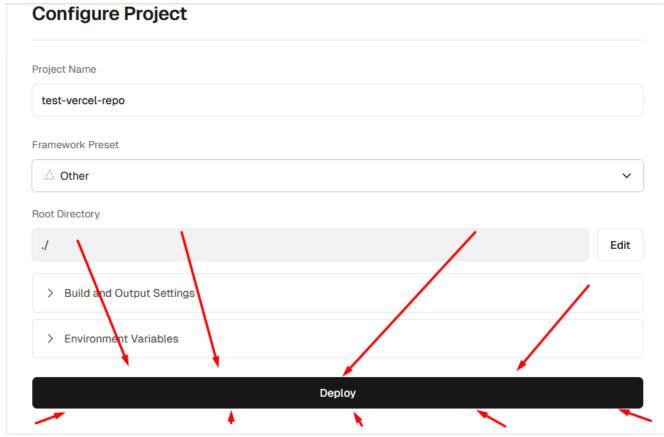


4. ◆ У вікні, що з'явиться, вибери профіль GitHub зі створеним тобою проєктом та натисни кнопку Authorize Vercel .

5. ♦ У вікні, що з'явиться, знайди свій проєкт та натисни на кнопку Імрогт :

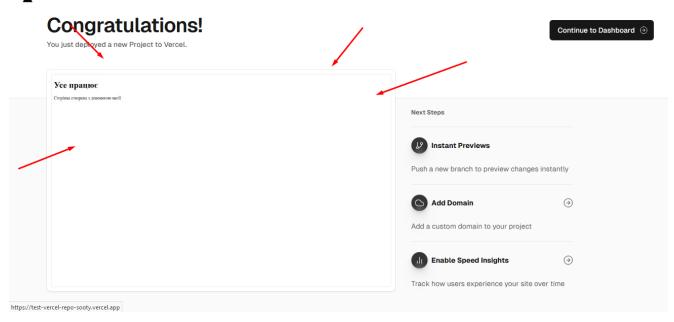


6. **У вікні конфігурації натисни кнопку** Deploy:



7. ◆ Почнеться процес розгортання проєкту. Коли він закінчиться, тебе автоматично переадресує на сторінку твого сайту!

8. • Натисни на це вікно:



і ти перейдеш на свій проєкт, розміщений на платформі Vercel.



🥟 Вітаю! Проєкт успішно розміщено на платформі Vercel

Робота з файлами та оновлення вмісту сторінки

На разі все не так складно. Розберімо, як на наш скелет накинути статичні файли. Ми розробимо базову структуру проєкту, напишемо дві HTML-сторінки та налаштуємо маршрутизатор для них.

- Створення базової структури у каталозі риblіс :
- Структура папок:



У каталозі src/views будемо зберігати каталоги з файлами сторінок. Тепер створимо два файли: src/views/page1/template.html Ta src/views/page1/template.css .

Створення сторінок

.../page1/template.html:

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Мій перший проект на Vercel</title>
  <link rel="stylesheet" href="/src/views/page1/template.css">
</head>
<body>
<header>
  <h1>Мій перший проєкт на Vercel</h1>
</header>
<main>
  <button onclick="window.location.href='/src/views/page2/template.html';">Перейти на іншу сторінку
</main>
</body>
</html>
```

• .../page1/template.css:

```
/* Загальні стилі */
body {
font-family: 'Arial', sans-serif;
margin: 0;
padding: 0;
background-color: #f0f8ff;
display: flex;
flex-direction: column;
align-items: center;
justify-content: center;
height: 100vh;
}
/* Стилі для хедера */
header {
position: absolute;
top: 20px;
width: 100%;
text-align: center;
}
header h1 {
font-size: 2rem;
color: #333;
margin: 0;
}
```

```
/* Стилі для кнопки */
main button {
font-size: 1.5rem;
padding: 10px 20px;
background-color: #007bff;
color: white;
border: none;
border-radius: 5px;
cursor: pointer;
transition: background-color 0.3s ease;
}
main button:hover {
background-color: #0056b3;
}
```

Як бачимо, на сторінці є кнопка, що буде відсилати нас до src/views/page2/template.html , тож створимо ще два файла:


```
<!DOCTYPE html>
<html lang="uk">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Друга сторінка</title>
  <link rel="stylesheet" href="/src/views/page2/template.css">
</head>
<body>
   <header>
       <h1>Мій перший проєкт на Vercel</h1>
   </header>
    <main>
        <р>Це друга сторінка</р>
       <button onclick="window.location.href='/src/views/page1/template.html';">Повернутися назад</but
</body>
</html>
```

• .../page2/template.css:

```
/* Загальні стилі */
```

```
body {
font-family: 'Arial', sans-serif;
margin: 0;
padding: 0;
background-color: #f0f8ff;
display: flex;
flex-direction: column;
align-items: center;
justify-content: center;
height: 100vh;
/* Стилі для хедера */
header {
position: absolute;
top: 20px;
width: 100%;
text-align: center;
header h1 {
font-size: 2rem;
color: #333;
margin: 0;
}
/* Стилі для головного блоку */
main p {
font-size: 1.5rem;
color: #333;
margin-bottom: 20px;
/* Cтилі для кнопки */
main button {
font-size: 1.5rem;
padding: 10px 20px;
background-color: #007bff;
color: white;
border: none;
border-radius: 5px;
cursor: pointer;
transition: background-color 0.3s ease;
main button:hover {
background-color: #0056b3;
```

Коли ми створили всі ці файли, спробуємо оновити їх на платформі Vercel. Для цього потрібно оновити файли на GitHub. У терміналі (*He забувайте перевірити, що в терміналі відкрита саме коренева директорія проєкту*), введіть такі команди:

```
git add --all
git commit -m "update"
git push origin main
```

Після цього оновлення завантажиться на GitHub, а Vercel автоматично почне розгортати нові версії, тож треба трохи почекати.

Перейдіть до своєї сторінки у браузері. Як бачите, на сайті з'явиться стартова сторінка: index, а щоб перейти на створену сторінку, впишіть в URL: https://[ваш домен]/src/views/page1/template.html:

```
Мій перший проект на Vercel

Перейти на іншу сторінку
```

Сторінка створена, але перехід на неї не дуже зручний. Тож налаштуємо маршрутизатор, щоб при переході на https://[ваш домен]/раде1 відображалася сторінка раде1, а при переході на https://[ваш домен]/раде2 відображалася сторінка раде2.

Змінюємо файл vercel.json:

I змінимо посилання на сторінках:

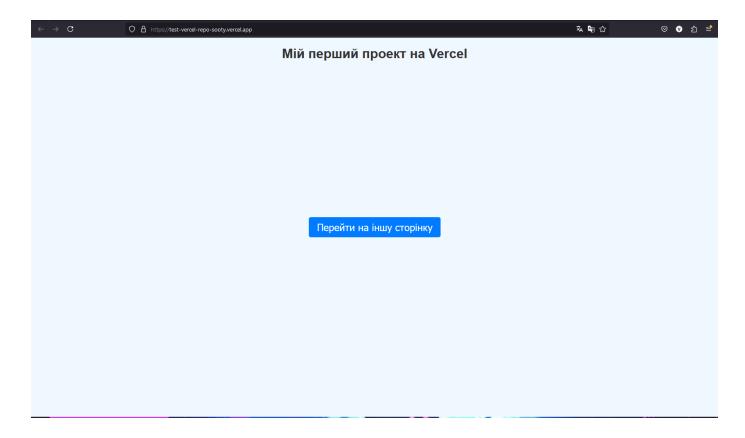
.../page2/template.html

I нарешті, щоб усе працювало правильно, видаляємо файл public/index.html .

Знову оновлюємо наш проєкт:

```
git add --all
git commit -m "update"
git push origin main
```

Після того як Vercel обробить оновлення, перейдіть на ваш сайт.



Як бачимо, тепер нам не потрібно вводити довгі URL-адреси.

Також натисніть на кнопку для перевірки сторінки 2. Тепер для переходу до неї достатньо ввести https://[ваш домен]/page2 :

