

Tailwind CSS

Asst.Prof.Drusawin Vongpramate

Department of Information Technology

Faculty of Science, BRU

Hint

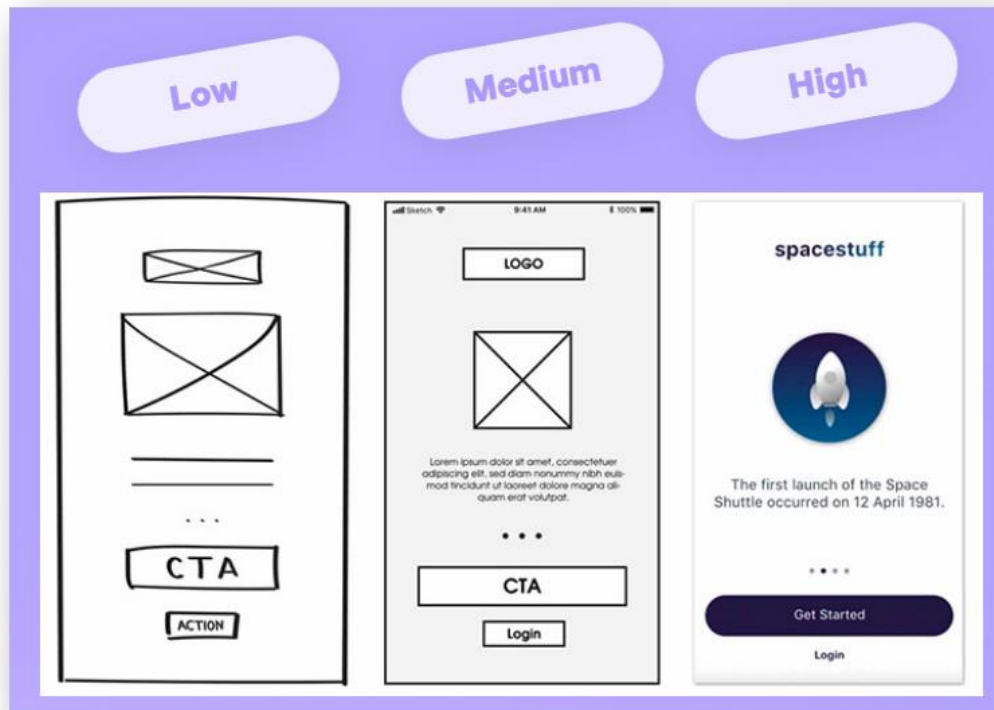


Main Topic



Sub Topic

Wireframe

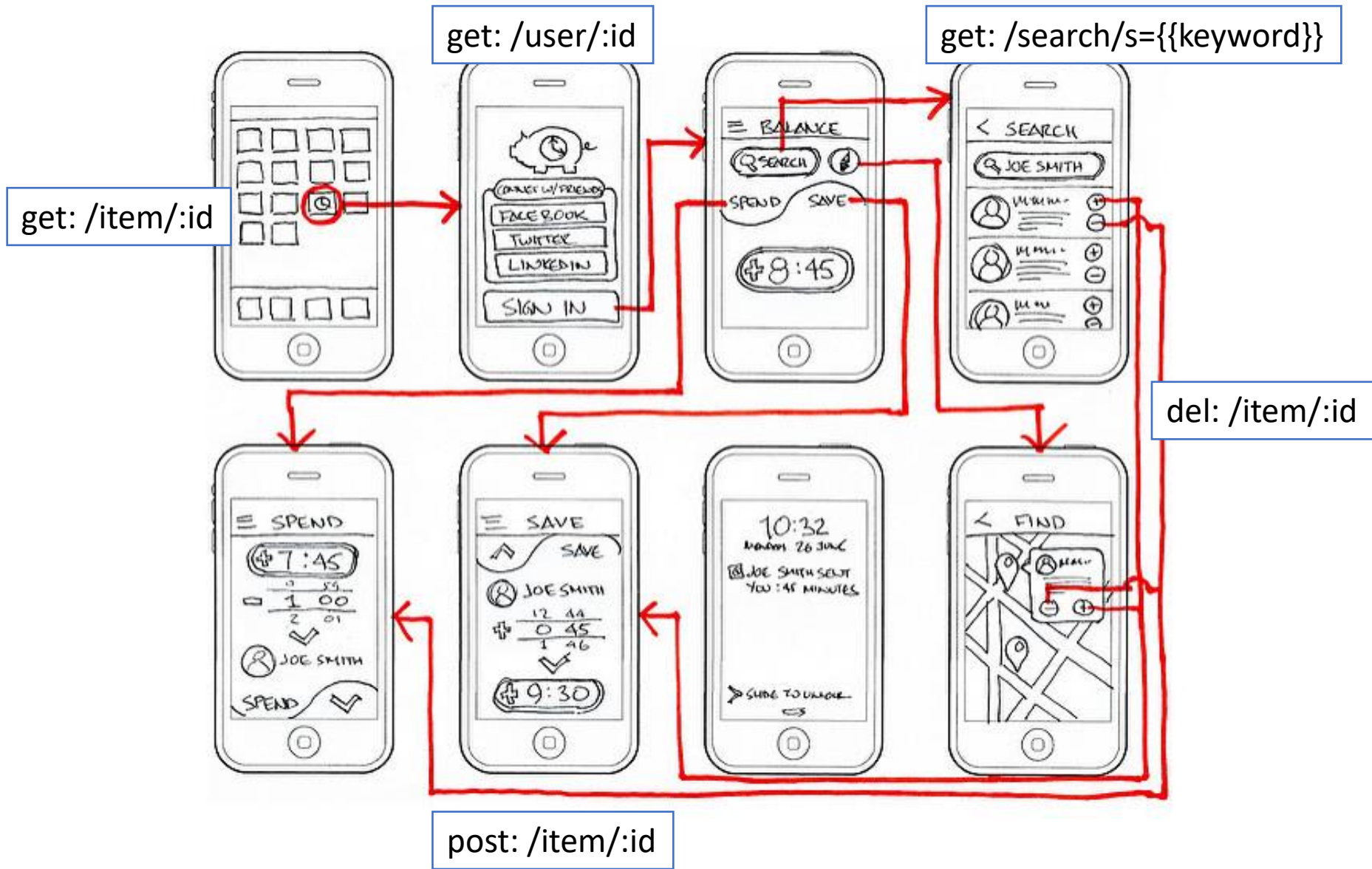


A low-fidelity wireframe is a basic sketch of a website or an app's interface. It's a simplified version of the final product, focusing on functionality rather than visual design.

Must included

- Basic Layout
- UI Elements
- Content Placement
- Navigation
- User Flow
- Annotations

Wireframe



Wireframe on Visily

The image displays the Visily wireframing tool interface. At the top, the navigation bar includes a home icon, the text "/ Welcome to Vi... / Getting Started", and icons for DESIGN, PROTOTYPE, and INSPECT. On the right, there are search, chat, play, download, and share icons.

The left sidebar contains several sections:

- Elements**: A search bar and a grid of basic elements including Section, Screen, Container, Rectangle, Overlay, Image, Oval, Triangle, Line, Heading, Text, and Link.
- Smart Components**: A grid of pre-designed UI components such as Button, Textbox, Dropdown, Search, Textarea, Checkbox, Radio, Menu, and Tag.

The central canvas, titled "Welcome to Visily!", features a central wireframe of a mobile application. It is annotated with callouts for various features:

- Top navigation bar**: A yellow callout box.
- Working modes**: A green callout box.
- Collaboration tools**: An orange callout box.
- UI Library**: A pink callout box.
- Visily AI**: A blue callout box.
- Right panel**: A purple callout box.
- Help and Resources**: A teal callout box.

At the bottom of the canvas, there are "Quick tips" for Move, Zoom, and Command search.

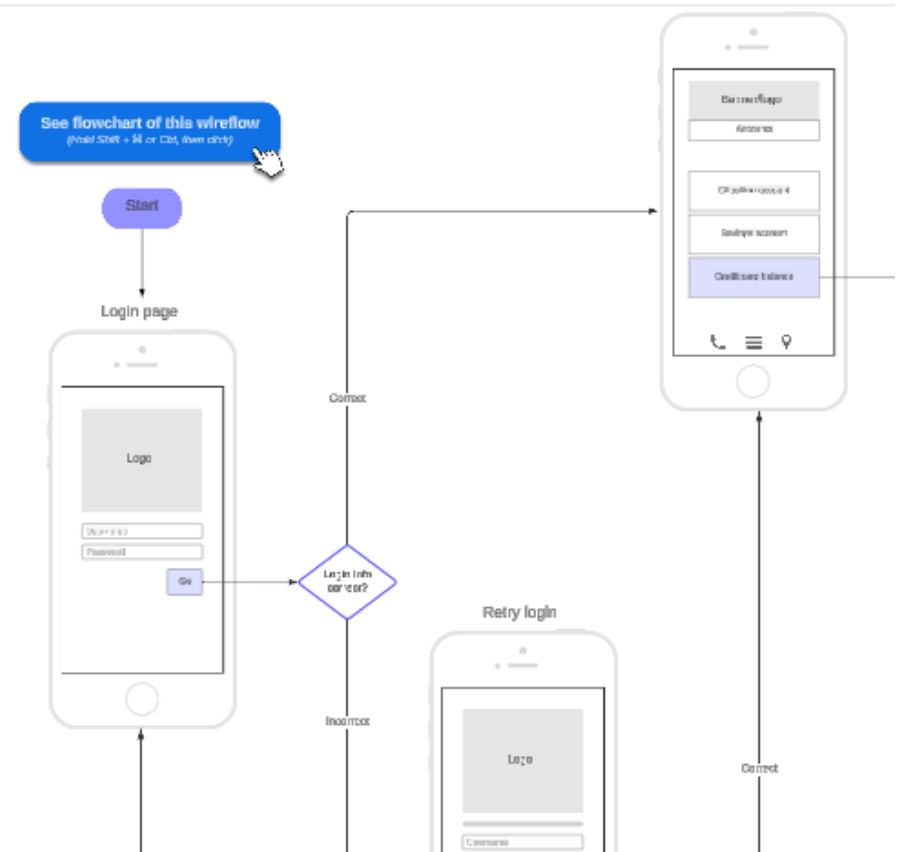
The right sidebar shows a wireframe of a "Homepage 2" with sections for "Website heading goes here", "Product Features", and "Our Team".

Wireframe on



<https://t.ly/H1gPS>

Bank app wireflow example



Tailwind CSS

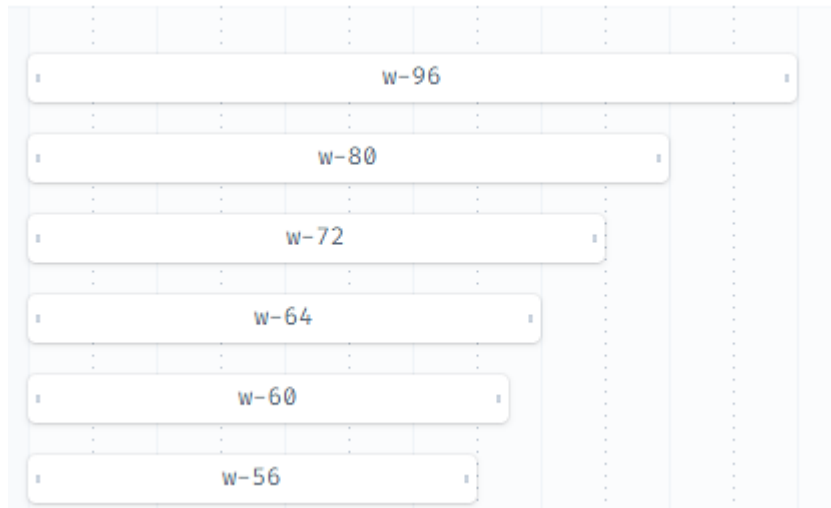
Rapidly build modern websites without ever leaving your HTML.

A utility-first CSS framework packed with classes like flex, pt-4, text-center and rotate-90 that can be composed to build any design, directly in your markup.



tailwindcss

API design system



font-sans

The quick brown fox jumps over the lazy dog.

shadow

shadow-lg

shadow-2xl



VS code Extension



The image shows the VS Code extension marketplace page for Tailwind CSS IntelliSense. The extension is by Tailwind Labs, with a version of v0.12.5. It has 7,054,054 installations and a 5-star rating from 97 reviews. The page includes buttons for 'Disable', 'Uninstall', and 'Switch to Pre-Release Version', along with an 'Auto Update' checkbox. Below the extension information, there are tabs for 'DETAILS', 'FEATURES', and 'CHANGELOG'. The main content area shows a code editor with HTML code and a dropdown menu for selecting background colors, including 'bg-transparent' and various shades of 'bg-teal'.

Tailwind CSS IntelliSense v0.12.5

Tailwind Labs tailwindcss.com | 7,054,054 | ★★★★★ (97)

Intelligent Tailwind CSS tooling for VS Code

Disable | Uninstall | Switch to Pre-Release Version | Auto Update

DETAILS | FEATURES | CHANGELOG

```
1 <div class="w-full flex items-center justify-between block p-6 space-x-6">
2   <div class="flex-1 truncate">
3     <div class="flex items-center space-x-3">
4       <h3 class="text-gray-900 text-sm font-medium truncate">Jane Cooper</h3>
5       <span class="text-teal-600 bg-t" >Admin</span>
6     </div>
7     <p class="mt-1 text-gray-500">
8   </div>
9   
10 </div>
```

- bg-transparent
- bg-teal-50
- bg-teal-100
- bg-teal-200
- bg-teal-300
- bg-teal-400

Create Next project

> npx create-next-app@latest

```
✓ Would you like to use TypeScript? ... No / Yes
✓ Would you like to use ESLint? ... No / Yes
✓ Would you like to use Tailwind CSS? ... No / Yes
✓ Would you like to use `src/` directory? ... No / Yes
✓ Would you like to use App Router? (recommended) ... No / Yes
✓ Would you like to customize the default import alias (@/*)? ... No / Yes
```

Installation

Install tailwindcss via npm, and create your tailwind.config.js file.

> npx tailwindcss init

Configure your template paths in tailwind.config.js

```
import type { Config } from "tailwindcss";

const config: Config = {
  content: [
    "./pages/**/*.{js,ts,jsx,tsx,mdx}",
    "./components/**/*.{js,ts,jsx,tsx,mdx}",
    "./app/**/*.{js,ts,jsx,tsx,mdx}",
  ],
}
```

Tailwind CSS directives

Add the Tailwind CSS directives that Tailwind will use to inject its generated styles to a Global Stylesheet in your application, for example:

```
app > globals.css > {} @media (prefers-color-scheme: dark)
1  @tailwind base;
2  @tailwind components;
3  @tailwind utilities;
4
```

Ignore css rule

```
.vscode > {} settings.json > ...
...
1  {
2  |   "css.lint.unknownAtRules": "ignore"
3  | }
4
```

Statically Typed Links

Next.js can statically type links to prevent typos and other errors when using `next/link`, improving type safety when navigating between pages. edit `next.config.mjs`

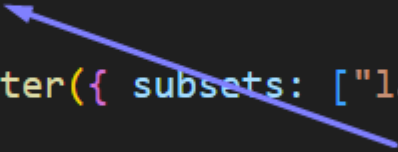
```
next.config.mjs > ...
1  /** @type {import('next').NextConfig} */
2  const nextConfig = {
3    experimental: {
4      typedRoutes: true,
5    },
6  };
7
8  export default nextConfig;
9
```

```
import type { Route } from 'next';
import Link from 'next/link'
// TypeScript errors if href is not a valid
route<Link href="/about" />
```

Importing Styles

Inside the root layout (app/layout.tsx), import the globals.css stylesheet to apply the styles to every route in your application.

```
app > layout.tsx > RootLayout
1  import type { Metadata } from "next";
2  import { Inter } from "next/font/google";
3  import "./globals.css";
4
5  const inter = Inter({ subsets: ["latin"] });
6
7  export const metadata: Metadata = {
8    title: "Create Next App",
9    description: "Generated by create next app",
```



Using Classes

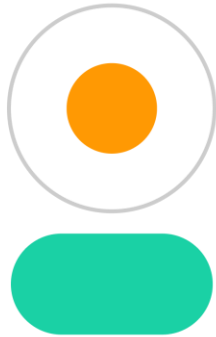
After installing Tailwind CSS and adding the global styles, you can use Tailwind's utility classes in your application.

```
1 export default function Home() {
2   return (
3     <>
4     |   <h1 className="text-3xl font-bold underline">Hello, Next.js!</h1>
5     |   </>
6     | );
7   }
8
```

DaisyUI

The most popular component library for Tailwind CSS

daisyUI adds component class names to Tailwind CSS so you can make beautiful websites faster than ever.



daisyUI

Pure Tailwind vs DaisyUI

```
// Styling a simple button
```

```
<button class="bg-indigo-600 px-4 py-3 text-center text-sm  
font-semibold inline-block text-white cursor-pointer  
uppercase transition duration-200 ease-in-out rounded-md  
hover:bg-indigo-700 focus-visible:outline-none focus-  
visible:ring-2 focus-visible:ring-indigo-600 focus-  
visible:ring-offset-2 active:scale-95">
```

```
  Tailwind Button  
</button>
```

```
// Result:
```



TAILWIND BUTTON



```
// Styling a simple button
```

```
<button class="btn btn-primary">  
  daisyUI Button  
</button>
```

```
// Result:
```



daisyUI Button

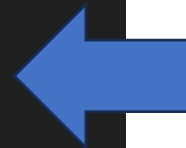


Install & Config DaisyUI

> npm i -D daisyui@latest

tailwind.config.ts

```
tailwind.config.ts > ...
3  const config: Config = {
9    theme: {
10     extend: {
11       backgroundImage: {
12         "gradient-radial": "radial-gradient(circle, #f47320 1px, transparent 1px)",
13         "gradient-conic": "conic-gradient(from 180deg at 50% 50%, #f47320 0deg, #f47320 360deg)",
14       },
15     },
16   },
17 },
18 plugins: [
19   require('daisyui'),
20 ],
21 };
22 export default config;
23
```



Add this line

Copy & Custom

daisyUI 4.12.10

- modal
- Swap
- Theme Controller
- Data display
 - Accordion
 - Avatar
 - Badge
 - Card**
 - Carousel
 - Chat bubble
 - Collapse
 - Countdown
 - Diff
 - Kbd
 - Stat
 - Table
 - Timeline
- Navigation
 - Breadcrumbs
 - Bottom navigation
 - Link
 - Menu
 - Navbar
 - Pagination
 - Steps
 - Tab
- Feedback
 - Alert
 - Loading
 - Progress
 - Radial progress

Search... (ctrl K)

card-normal	Responsive	Applies default paddings
card-compact	Responsive	Applies smaller padding
card-side	Responsive	The image in <figure> will be on to the side

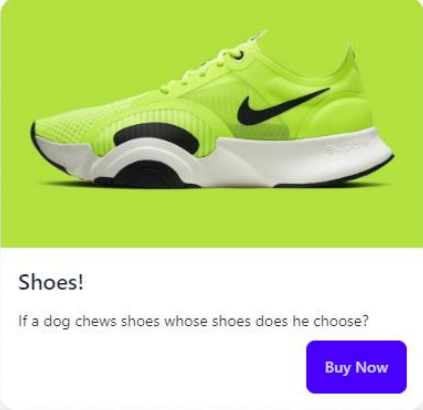
Card

Preview HTML JSX

```
<div className="card bg-base-100 w-96 shadow-xl">
  <figure>
    
    </figure>
  <div className="card-body">
    <h2 className="card-title">Shoes!</h2>
    <p>If a dog chews shoes whose shoes does he choose?</p>
    <div className="card-actions justify-end">
      <button className="btn btn-primary">Buy Now</button>
    </div>
  </div>
</div>
```

Compact card (less padding for `card-body`)

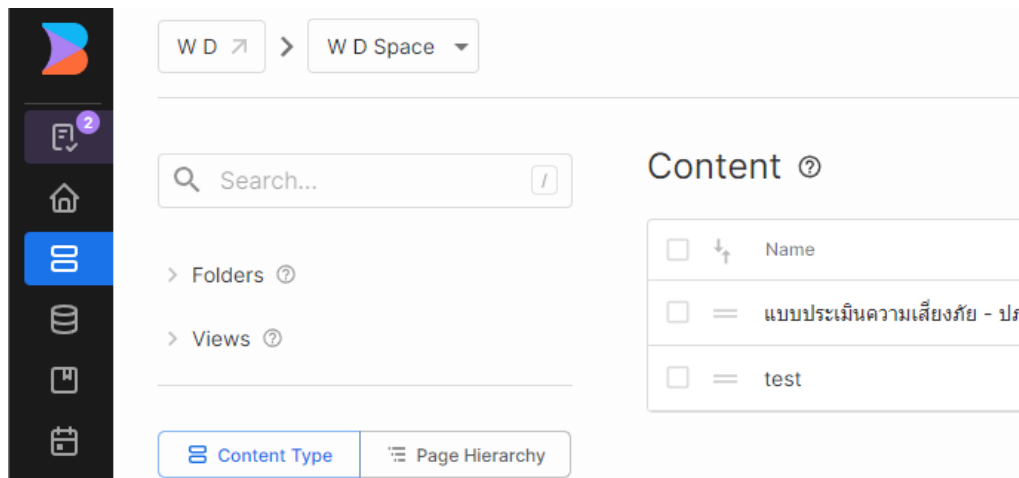
Preview HTML JSX





Take Ideas to Production in Seconds, Not Sprints

Accelerate your digital teams with AI-powered design-to-code, visual editing, and enterprise CMS, all in our Visual Development Platform.

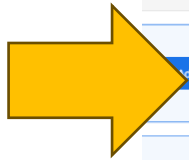




2. Get Code



1. Create Layout



The screenshot shows the Builder.io interface in a preview mode. At the top, there are navigation tabs: EDIT, A/B TESTS, INSIGHTS, VERSIONS, and HISTORY. On the right side, there are icons for navigation and a 'Publish' button. Below the navigation, there is a 'Get code' button. The main area shows a mobile layout editor with several 'Add Block' buttons. A blue banner at the top of the editor says 'a preview using the fallback editor. Learn how to integrate Builder into your codebase'. On the right side, there is a panel for editing page fields, including 'Page url' (set to '/test') and 'Title' (set to 'test').

3. Select Option & Generate by AI

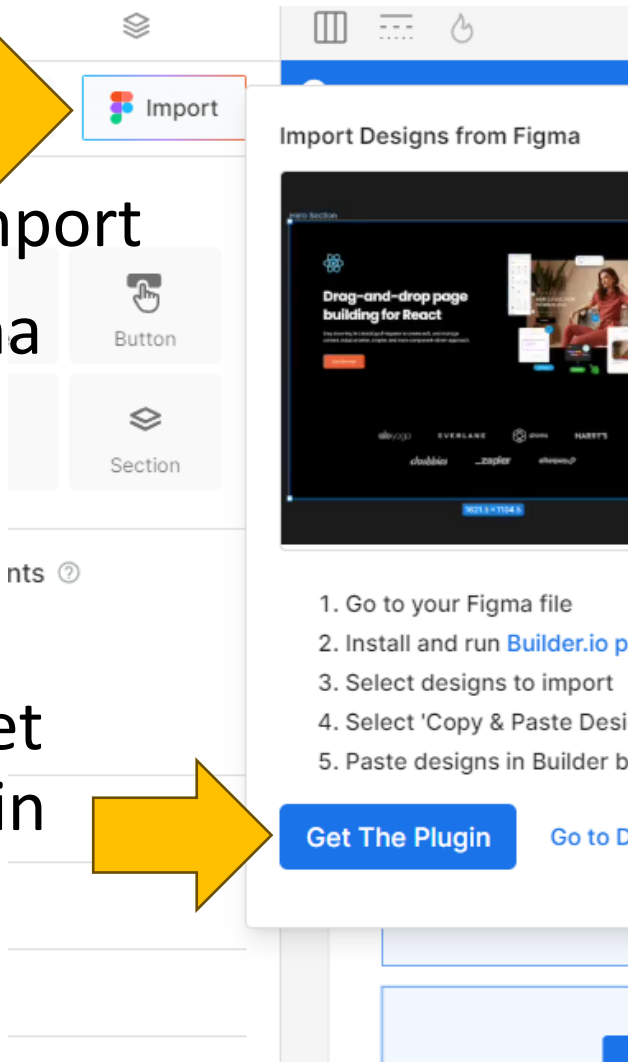
The screenshot shows the Builder.io interface with the 'Code Settings' panel open on the left. The 'Code Settings' panel has the following options: Framework (React), Styling (Tailwind), Language (TypeScript), and Testing Framework (None). There is a '+ Custom Instructions' button below these settings. In the center, there is a list of components: Layout.tsx, Row.tsx, Column.tsx, and MyComponent.tsx. On the right, there is a code editor showing the generated code for the 'Layout.tsx' component. The code is as follows:

```
1 import React from 'react';
2 import Column from './Column';
3 import Row from './Row';
4
5 interface LayoutProps {
6   children: React.ReactNode;
7 }
8
9 const Layout: React.FC<LayoutProps> = ({ children }) => {
10   return (
11     <main className="box-border flex relative flex-col shrink-0 mt-5">
12       {children}
13     </main>
14   );
15 };
16
17 export default Layout;
```

Builder.io & Figma

1. Import Figma

2. Get Plugin



Import

Import Designs from Figma

Drag-and-drop page building for React

Button

Section

nts ?

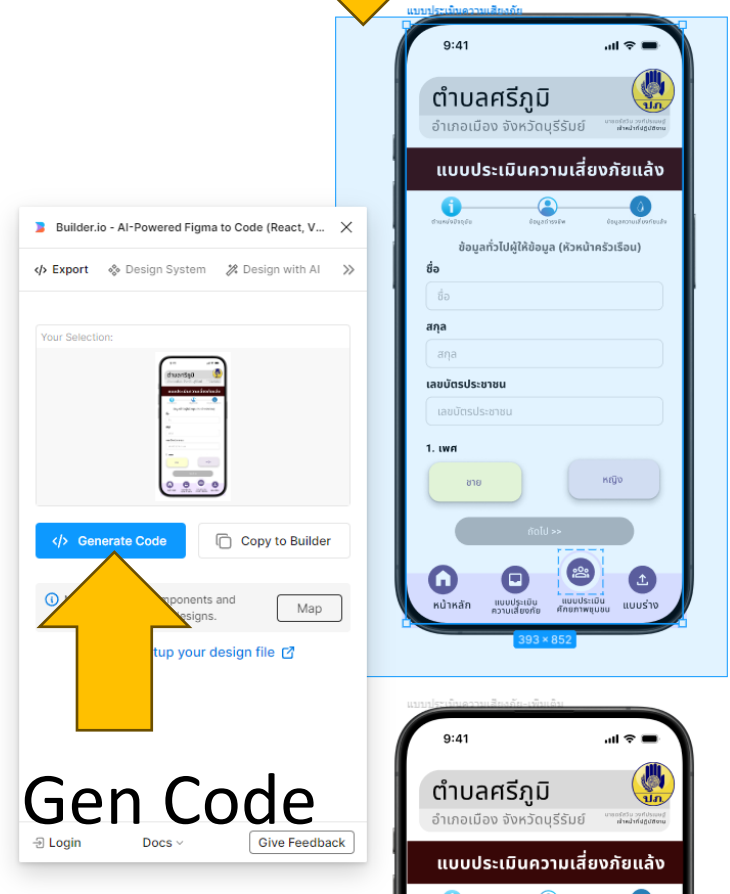
1. Go to your Figma file
2. Install and run [Builder.io p](#)
3. Select designs to import
4. Select 'Copy & Paste Desi
5. Paste designs in Builder b

Get The Plugin

Go to D

3. Select Area

4. Gen Code



Builder.io - AI-Powered Figma to Code (React, V... X

</> Export Design System Design with AI >>

Your Selection:

Generate Code

Copy to Builder

Components and designs. Map

Setup your design file

Login Docs Give Feedback

แบบประเมินความเสี่ยงภัยแล้ง

ต้าบลศรีภูมิ

อำเภอเมือง จังหวัดบุรีรัมย์

ข้อมูลทั่วไปผู้ให้ข้อมูล (หัวหน้าครัวเรือน)

ชื่อ

สกุล

เลขบัตรประชาชน

เลขบัตรประชาชน

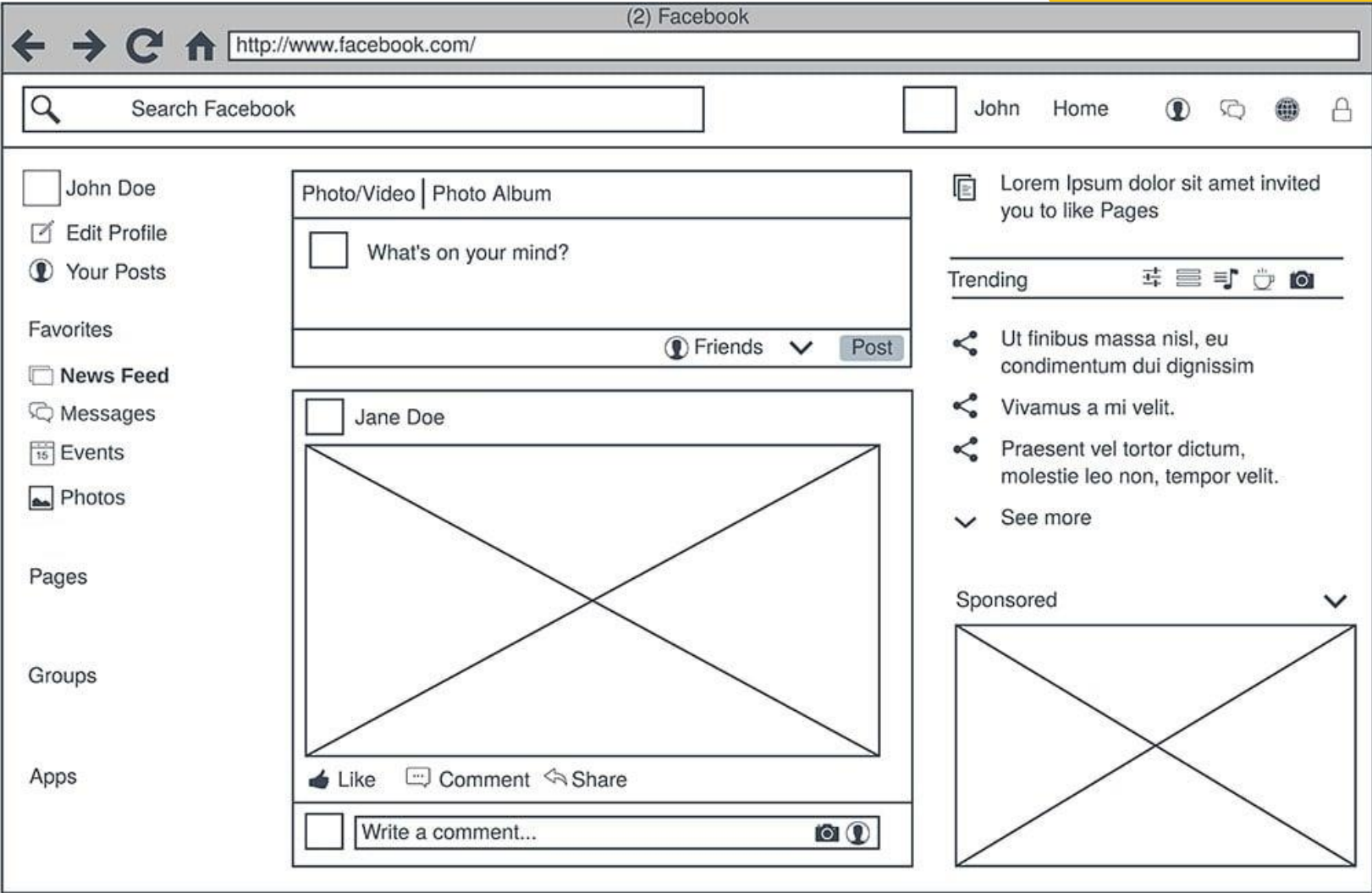
1. เพศ

ชาย หญิง

393 x 852

Create UI

Let's Try



Q & A