Figma

- What is Figma ?

  Figma is a collaborative web-based design tool that has revolutionized the way designers and teams create, share , and collaborate on digital projects.

- Key Features of Figma:

  Web-Based Platform : accessible from any device online .

  Real-Time Collaboration : Multiple users can work on a design simultaneously .

  Vector Editing Tools : Ability to create high-fidelity designs, UI Components, and illustrations,

  Prototyping and Interactions: Linking frames, defining interaction such as transistions, animations, overlays and more.

  Plugins and Integrations: It’s Functionality can be extended through plugins and integartions which allows users to customize their workflows to better suit their needs.

  Design Systems and Libraries: Teams can create shared libs of components, styles and assets, ensuring consistency accros projects.

Front-End dev with *Figma*

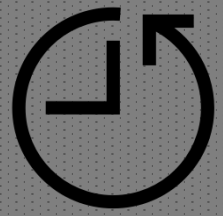
- Figma has Dev Mode that gives developers everything they need to navigate design files and transform designs into code !
- With Dev Mode, Designers and developers stay on the same page, making sure important details aren’t lost in the handoff process
- Dev Mode allows for 3rd party plugins to turn Figma mockups into code ; one of the best plugins is Builder.io tool which converts it to HTML/CSS with different tecs. such ,React/TailwindCSS and more.
- Also Dev Mode offers a VS Code extension to manage and mocking up the Figma design with it’s files and directories .

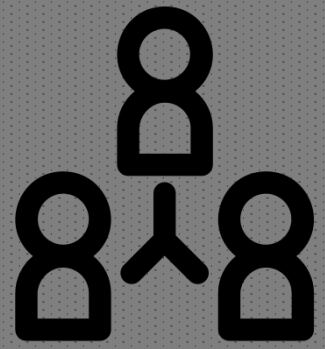
Front-end , Designer and Figma

Figma makes desiging and coding much easier , especially for a team of designers and developers .

Figma offers all what designer will need to design with clarity and simplicity tools for animations, styling ,transitions and more.

Figma structs the design into pages , layers and elements like a nested tree which provides for developers more details to focus on and clarity in order to safe time and collaborating effectively with **live** changes of designers.





Figma Design Conversion to Code

1. Manual Conversion :
   - Writing React Components from scratch to replicate the layout, styles, animations and interactions.
   - This type of conversion gives the full control over code structure and implementation , in addition the flexibility to customize it ;
   - But it’s time-consuming especially for inticate designs with many elements and requires knowledge of design principles and coding practices

2. Using Figma Plugins (Automatic):
   - Like Figmotion or Reactifier automate the conversion process by generating React Components directly.
   - These plugins analyze Figma layers, styles, animations, and interactions translating them into ready-to-use React Code.

   Most used extension : Builder.io AI tool.

Github

Github is a developer platform that allows developers to create, store, manage and share their code.

It uses Git software, providing the distributed version control of Git plus access control, bug tracking, software feature requests, task management and continuous integration .

It’s the biggest place for developers to learning and sharing open-source code.

Also it used along with many IDEs such as VS code, Figma, Netbeans IDE and many there, nearly with all platforms.

It’s used also by teams to organize, sharing, managing versions of codes so easily .

How to use Github

First of all a Repo creation is the first step, after account authentication (register or login) , head to create New repository in the header .

Then, Enter your project’s name in the Repository name, optionally add a Description for this repo.

You have 2 ways to upload ur project/folder ; using CLI or just drag and drop.

CLI commands explanation:

-init: initializing git files in your project for cloning-pushing-polling..etc

-add: adding the new files or last modified files to the uploading list.

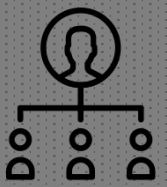
-commit -m(message) :explaning what has changed and ensuring the addition.

-branch -M ‘main’(name) : create-new/convert-to branch that named ‘main’ .

-remote add ‘origin’ ‘url’ : add a repo url with the name ‘origin’.

-push -u ‘origin’ ‘main’(branch) : pushing the latest commited files to the main branch with the origin(repo-url).

Github Collaboration:



The repo has Access section which contains contributers whom sharing codes with this repo.

Owner/Admin of a repo has to add the contributers with specific permission of Actions.

The workflow operations for contributers :

Fork - Edit Code/Files(add-commit) - Pull

Fork : copying the project.

Editing: uploading the new to it’s own branch .

Pull : request the owner to merge it with the base project.

Owner Workflow :

Then the owner/admin for this repo applies the desired changes to the original code base (Merge Pull Requests) at the Pull requests section.

Github offers a history section in order to ReChange it to previous version of committed code , which gives the ability to get back to any point of editions.