

Full Stack Developer

UI Designer

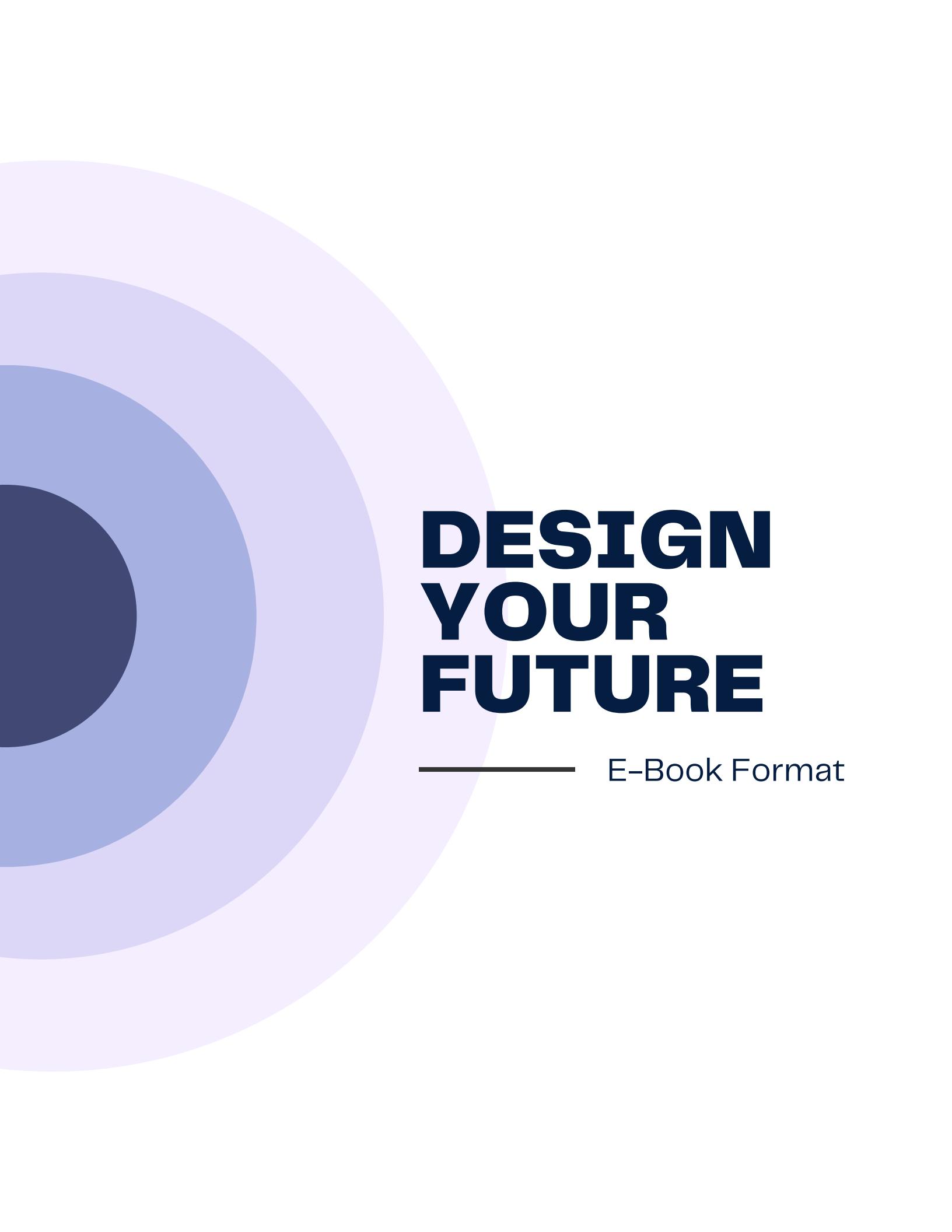


Master Path Towards How To Crack Full Stack

500+ Interview Question's

200+ Project Idea's

Om Patil



DESIGN YOUR FUTURE

E-Book Format

@Copyright 2023 Om Patil

All rights reserved

To be sold only on E-book format , World wide.
Based on Educational Purpose towards Programming.

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted.

In any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher.

PREFACE

This book is written to help you learn about Full-Stack Developer And UI Design. Proper and step by step Roadmap.

If you are an Totally beginner, then you will find that this book helps you from scratch to complex concepts in an easy to understand manner. If you don't know from where to start your Full-Stack Dev journey and stuck between these concept , this book will provide you proper roadmap in depth to start your coding immediately.

All topics in the book are carefully chosen to clear the all basic concepts to the mastery.

This Book have Covered Programming Part as well as designing part.

This book comes with an additional bonus project's on particular part. That means you can practice your skills. The project uses concepts covered in the book and gives you a chance to see how it all ties together.

And Also Provided quality of resources where you can learn and gain knowledge.

“If you are juggling in what to learn first and basically what to learn after that ,if you are seeking for a proper roadmap ” Then this book will clear your all concepts, you will learn what actual basic need and what to learn after basic to become a Full-Stack Web Developer In such a very simplest words.



INDEX

Roadmap Disclaimer	01
Front-End Dev Roadmap	02
Front-End Interview Questions	11
Front-End Perspective	27
Back-End Roadmap	30
Back-End Interview Questions	33
Back-End Perspective	41
HTML & CSS Projects	43

JavaScript Projects 44

ReactJS Projects 46

NodeJS + MongoDB Projects 48

Python Projects 50

Front-End Projects 52

Back-End Projects 54

Resources To Learn Full-Stack Dev 55

UI Roadmap 65

Resources To Learn UI 70

UI Perspective 84

DSA Interviews Question's

88

Conclusion Of Roadmap's

108

"Step Up Your Development Game: Our Full Roadmap Gives A Way To Achieve Your Dreams蝶"



Roadmap Disclaimer

All the Roadmap are beginner's friendly from scratch. This roadmap's will make your basic much stronger. once your basic will become strong , then you can easily understand further stuff very easily in short time. I hope this e-book helps you to start you journey.

This e-book is absolutely For beginner's. I tried to make roadmap as simple as so that beginner's will easily understand and start learning. mastering is not easy but becoming able to get job and also freelancing , freelancing is not hard as most people think.

There are lots of interview question in the roadmap, including very very basic to advance questions. Projects and questions might be repeated.

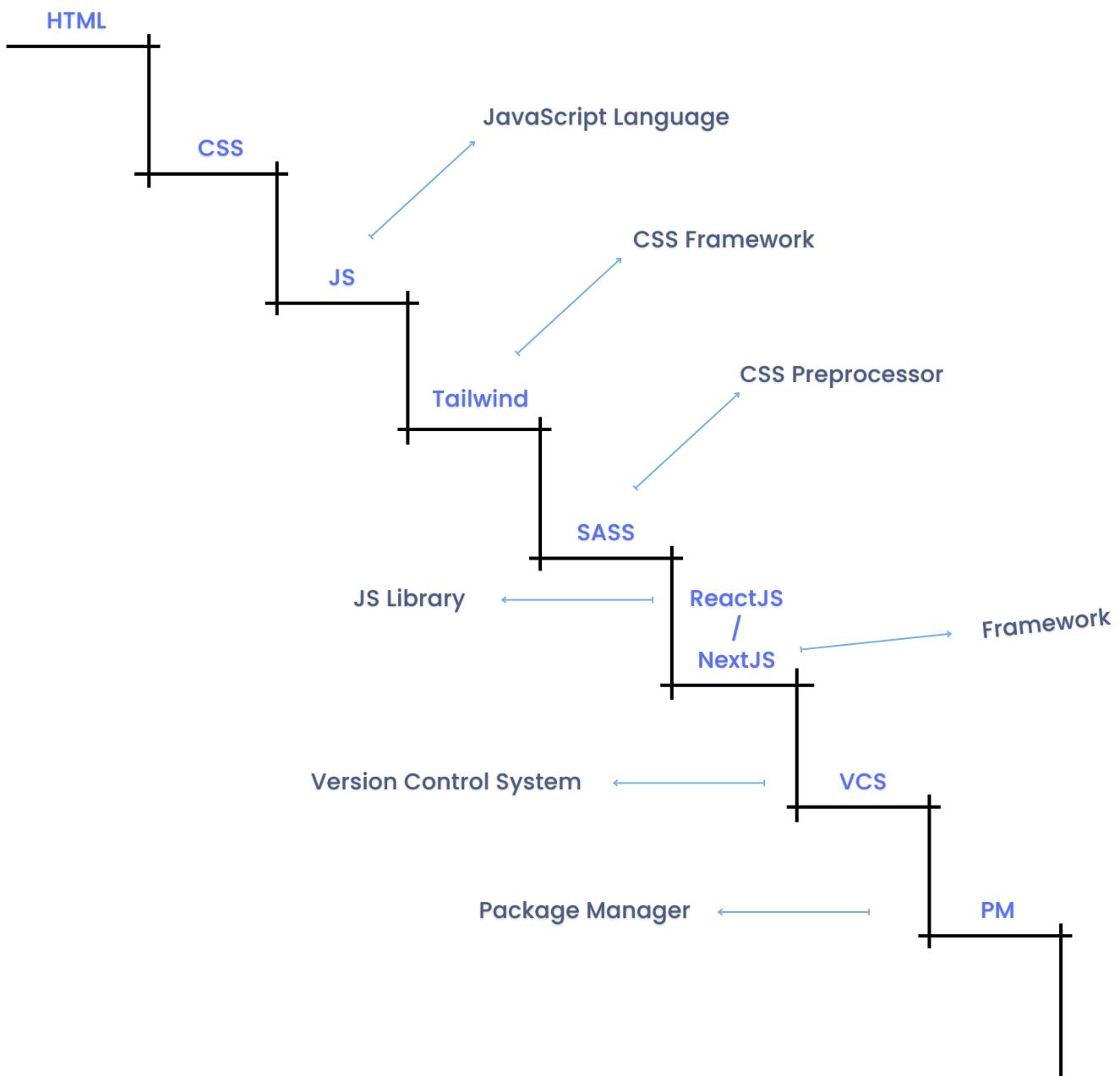
That's all and Yeah this is not only one roadmap which give a path towards Full-Stack Development. there are lots of roadmap where you can find good path. But in my perspective I found this roadmap found very simple and one of the best solution for beginner's.

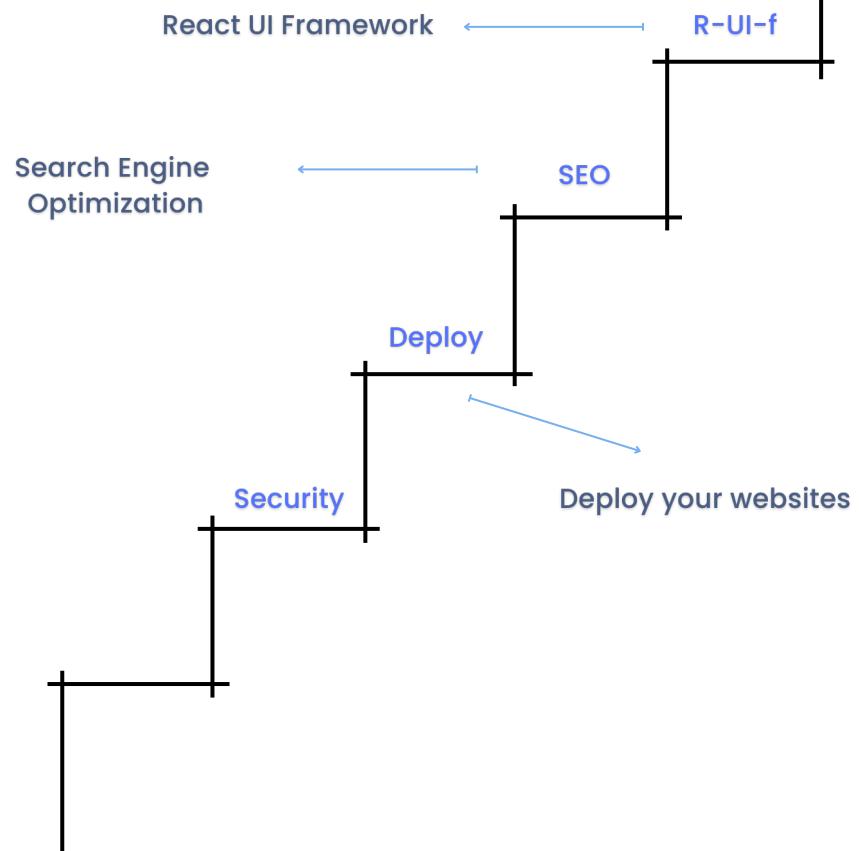
I Think it will not take much time to ready this E-book , hardly it will take 45 Min. there are so many resources and projects ideas are provides go through from that Once. it will helps you a lot in your journey.

This e-book is not teaching you anything. this is just path where you can start your learning process towards Full-Stack Development. this is not Experience Guidance.

FRONTEND

Front-End-Dev







HTML (Hyper Text Markup Language) is a markup language which make skeleton of website , Like home with only wall brick. Learn about

- HTML Features
- HTML Elements
- HTML Tags
- HTML Attributes
- HTML Table
- HTML Forms
- Metatags



CSS (Cascading Style Sheets) Basically It is used for style and design website , like once you make home with brick then you need to color it and design with different arts on wall Etc. Learn about

- CSS Selectors
- CSS Borders
- CSS Padding & Margin
- CSS Flexbox & Grid
- CSS Box Model
- CSS Display
- CSS Float

-
- CSS Positioning
 - CSS Media Queries
 - Specificity
 - CSS Animations
 - CSS Variables



JavaScript

JavaScript is a scripting or programming language which allows you to implement complex features on web pages , Ex : your home include fan how it run ? when you on the button right ! , there is a button which you have designed. if you On it nothing gonna happen because there is no functionality Of wires . so JavaScript helps to make it workable when you add wires. Learn about

- Expression
- Data Types
- Variables
- Math Operator
- Functions , Arrow Function
- Loops
- Scope
- Callbacks
- Template Literals

-
- DOM
 - Async / Await
 - Promises
 - Inheritance
 - Prototyping

CSS Framework

 Tailwind

 Bootstrap

 Tailwind

Tailwind is a CSS Framework. Tailwind is simple and easy to understand. it is a tool which make our work easy. It might take time to get friendly with utility classes name. but it is okay you can take refer of their documentation when you need. for beginner it very important to know about CSS before explore Tailwind CSS.

CSS Preprocessor

 SASS/SCSS

 {less} Less

 Stylus

★ SASS/SCSS

SCSS is a popular CSS Preprocessor which allows developer to write more efficient and maintainable CSS code. Learn about Variables , Nesting , Mixins , Function , Importing Etc. By using SCSS developers can save time and reduce the amount of code they need to write, while also improving the quality and consistency of their styles.

JavaScript Framework / Library

 React

 Angular

 Vue

 Meteor

★ React

React is a powerful JavaScript library that you can use to build user interfaces for web applications. Learn about

- JSX
- Components
- State
- Props
- Lists and Keys
- Form Handling
- Data Handling
- Hooks
- Custom Hooks
- Context
- Portals
- Routing
- State Management
- Pattern

Version Control System



Git



Git

Git is a distributed version control system that allows you to track changes made to files in a project over time. Developers use this to collaborate on code and manage different versions of their codebase.

Basic Package Manager

 NPM

 Yarn

NPM

Npm stands for Node Package Manager. It is a package manager for the Node JavaScript platform . Npm is known as the world's largest software registry. Open-source developers all over the world use npm to publish and share their source code.

React UI Framework

 Material UI

 React Bootstrap

 Chakra UI

Material UI

Material UI (MUI) is an excellent React UI framework with multiple pre-built components and templates.

Search Engine Optimization

★ SEO

SEO (Search Engine Optimization) is used to optimize a website or online content to improve its visibility and ranking on the internet. The main goal of SEO is to increase organic traffic on your website.

Deploy Your Website

◆ Netlify

△ Vercel

🐱 GitHub Pages

★ Netlify

Netlify is a popular web development platform that offers a variety of services for building, deploying, and managing web application or websites. Netlify is a powerful and flexible platform that offers a range of tools and services to help developers build and deploy modern web applications quickly and easily.

🛡️ Security

Security is an essential aspect of website or web application development.



Front-End Interview Questions

HTML Questions

- Explain Different Versions of HTML ?
- In HTML5 , which tag or tags embed a webpage inside of a webpage ?
- Where do <header> and <footer> tags typically occur ?
- What is the best way to apply bold styling to text ?
- What is formatting in HTML ?
- What are entities in HTML ?
- When should you use the <aside> element ?
- What is meta tag in HTML ?
- What is the difference between <div> and Tag in HTML ?
- What is the use of the <canvas> tag in HTML5?
- What is the difference between an inline and block-level element ?
- What does the <label> element do ?

- What attribute applies a keyboard shortcut hint to the current element?
- Which tag is the root element of an HTML document?
- What is the use of the <nav> tag in HTML5?
- What is the use of the <textarea> tag in HTML ?
- What are the different types of form controls in HTML ?
- What is the difference between HTML and XHTML ?
- What is the purpose of HTML attributes ?
- To get a link to open in new window or tab , use the _attribute
- How many types of heading are available in HTML ?
- What is the use of marquee tag in HTML ?
- What is the use of <select> and <option> tag in HTML ?

CSS Question's

- What is CSS ?
- What are the selectors in CSS ?
- What is the syntax of CSS rule ?
- What is the difference between inline , internal and External CSS ?
- What is the difference between Pseudo-classes and Psuedo-elements ?
- Explain CSS grid
- What are three valid ways of adding CSS to an HTML page ?
- To change the color of an SVG using CSS, which property is used?
- How would you make the first letter of every paragraph on the page red?
- Which choice would give a block element rounded corners?
- What is the use of the !important keyword in CSS?
- What is the difference between padding and margin in CSS?
- What is padding and margin in CSS ?
- What is the difference between display: none and visibility: hidden in CSS?
- What is positioning in CSS ?
- What is the difference between Absolute and Relative Positioning in CSS ?

- What are the counters in CSS ?
- Which one would you prefer among px , em , or pt and why ?
- What is the difference between border-box and content-box in CSS ?
- What is the difference between float and clear in CSS?
- What is the purpose of the z-index property in CSS?
- What is the difference between font-size and font-weight in CSS?
- What is the ::placeholder pseudo-element used for?
- When would you use the @font-face method?
- What is an advantage of using inline CSS?
- Which property is used to create a drop shadow effect on an HTML element?
- What is CSS BEM ?
- What is flexbox in CSS ?

JavaScript Question's

- What is ECMAScript ?
- What is the difference between JavaScript and Java ?
- What is closure ?
- What is scope in JavaScript ?
- What is the difference between let , const and var ?
- What is callbacks ?
- Explain Implicit type coercion in JavaScript ?
- What is NaN property in JavaScript ?
- Explain passes by value and passes by reference ?
- What are the promises in JavaScript ?
- What is the difference between synchronous and asynchronous code ?
- What is JSON and how is it used in JavaScript ?
- What is a RESTful API ?
- Which property references the DOM object that dispatched an event ?
- What does the following expression evaluate to? `[] == [];`
- Which keyword is used to create an error ?

- Explain "This" Keyword ?
- What is the difference between map and forEach methods ?
- Explain call() ,apply() and bind() methods.
- What is the spread operator ?
- What is Destructuring ?
- What is the difference between null and undefined ?
- What is the difference between for and while loop ?
- what is cookie and How to create cookie in JavaScript ?
- What is module in JavaScript ?
- What is difference between JavaScript And TypeScript ?
- What is a prototype in JavaScript?
- What is arrow function ins JavaScript ?
- What is the HTTP verb to request the contents of an existing resource?
- What are async/await ?
- What is a higher-order function ?

Tailwind CSS Question's

- What is Tailwind CSS and why would you choose to use it ?
- How do you install and configure Tailwind CSS for your project ?
- How do you use Tailwind With Front-end framework like React or Vue.js?
- How do you download Tailwind CSS Typography ?
- How to download Tailwind CSS Table Plugin ?
- What is the syntax for applying styles with Tailwind CSS ?
- How do you use the @screen directive in Tailwind CSS ?
- What is the purpose of the @apply directive in Tailwind CSS ?
- How do you create and use custom plugin in Tailwind CSS?
- What is the utility-first approach in Tailwind CSS ?
- Customizing Breakpoints in Tailwind CSS
- How to set auto height on grid item using Tailwind ?
- How to add Dark mode in ReactJS using Tailwind CSS?
- How to Build a Card Component using Tailwind CSS ?
- How to set Transition duration in Tailwind CSS ?

SCSS Question's

- How is SCSS different from CSS ?
- What is the syntax for declaring variables in SCSS ?
- How do you use the @while directive in SCSS ?
- How do you use the @function directive in SCSS ?
- What selector Nesting in SASS is used for ?
- Why SASS is considered better than LESS ?
- Explain when can you use the %placeholders in SASS ?
- Explain how Mixins is useful ?
- What is the purpose of the & selector in SCSS?
- How do you use parent selector references in SCSS?
- What is the difference between @import and @use directives in SCSS?
- How do you install and configure SCSS for your project?
- How do you use the @if and @else directives in SCSS?
- How do you use the @for and @each directives in SCSS?
- What is SCSS and why would you choose to use it?

ReactJs Question's

- What is the difference between React and Angular
- What is JSX
- What is Virtual DOM in React ?
- What is the significance of keys in React?
- What is the difference between the ES6 and ES5 standards?
- What are forms in React?
- What is an arrow function and how is it used in React?
- What is the testing library most often associated with React
- What is the purpose of the & selector in SCSS?
- What can you use to handle code splitting?
- What is the children prop?
- To create a constant in JavaScript, which keyword do you use?
- What is sent to an Array.map() function?
- What is the purpose of refs in React?
- What is the difference between a state and props in React?
- What is Redux and how does it work with React?
- What is the purpose of React Router and how does it work?

- What is the significance of using CSS modules in React?
- What is the difference between Shallow Rendering and Mounting in React?
- How can you optimize the performance of a React application?
- What is the significance of using spread operator in React?
- Which function from React can you use to wrap Component imports to load them lazily?
- Which function is used to update state variables in a React class component?
- What is setCount?
- How would you generate the boilerplate code for a new app that you are building to collect underpants?
- What command can you use to generate a React project?
- React is an open-source project but is maintained by which company?
- What is the JavaScript syntax extension that is commonly used to create React elements?
- What is the purpose of PureComponent over Component in React?
- What is the purpose of using PropTypes in React?
- How do you handle events in React?
- What are React hooks and how do they work?

Git Question's

- What is Git and what is its purpose?
- What is the difference between Git and SVN?
- How can you check your current git version?
- `git rebase -i HEAD~10` what is the Use of this command in Git ?
- Why would you use a pre-receive hook in your remote repository?
- If you cloned an existing git repository, what would happen?
- Where are files stored before they are committed to the local repository?
- What are branches in Git and why are they useful?
- What is the difference between a merge and a rebase in Git?
- What is Git reset ?
- What is Git push and how do you use it?
- What is Git fetch and Git pull and what is the difference between them?
- What command would let you modify your previous commit?
- What command would you use to create a new git repository?
- How many ways are present in Git to integrate changes from one branch into another?

NPM Question's

- What is NPM and how does it work?
- What is the purpose of the npm registry?
- What is package.json file and what information does it contain?
- How do you install an npm package locally and globally?
- How do you update an npm package to its latest version?
- What is the difference between npm and Yarn?
- How do you publish a package to the npm registry?
- What is the difference between npm install and npm ci?
- What is the purpose of the node_modules folder?
- How do you remove a package using npm?
- What is the difference between npm start and npm run?
- What are the advantages of using npm for package management in Node.js applications?
- What is the difference between dependencies, devDependencies, and peerDependencies in package.json?

Material UI Question's

- What is Material UI and how does it work?
- What are the key components in Material UI?
- How do you customize the theme in Material UI?
- What is the purpose of the Grid component in Material UI?
- How do you implement responsive design using Material UI?
- What is the difference between a Button and a IconButton in Material UI?
- What is the purpose of the Typography component in Material UI?
- How do you implement a modal using the Dialog component in Material UI?
- What are some best practices for using Material UI in a project?
- How do you install Material UI in a project?

Netlify Question's

- What is Netlify platform and how does it work ?
- How do you deploy a website to Netlify?
- What are some of the key features of Netlify?
- How do you configure custom domains in Netlify?
- What are the advantages of using Netlify for website hosting?
- What are some best practices for using Netlify in a project?
- What is Netlify Identity and how is it used?
- What is the purpose of Netlify Analytics?
- How do you manage form submissions in Netlify?
- How do you set up continuous deployment with Netlify?
- What is Netlify Functions and how is it used?

SEO Question's

- What are the key components of on-page SEO?
- What is SEO and why is it important?
- How do you stay up-to-date with the latest SEO trends and updates?
- What are some common SEO mistakes to avoid?
- How do you optimize website content for local SEO?
- How do you track and analyze SEO performance?
- What is the importance of meta tags in SEO?
- How do you optimize images for SEO?
- What are some best practices for optimizing website structure for SEO?
- What is keyword research and how is it used for SEO?
- What is the impact of mobile-friendliness on SEO?

Security Question's

- What are some common website security threats and vulnerabilities?
- What is SSL and how does it enhance website security?
- What is SQL injection and how do you prevent it?
- How do you implement secure password storage on a website?
- What is two-factor authentication (2FA) and how does it enhance website security?
- How do you handle website security incidents and breaches?
- How do you protect a website from DDoS attacks?
- What is Cross-Site Scripting (xss) and how do you prevent it?
- How do you secure a website against malicious attacks and hacking attempts?
- What are some best practices for securing website APIs?
- What are some common website security compliance standards and regulations to be aware of?

Front End Perspective

As a beginner knowing which technology to learn is a problem that confuses many beginners, in this roadmap you will learn everything efficiently for free and in simple language.

If you are looking for Front-End Development as a career, you enter a field which is never going to end because the day internet will end, the need for web developers will also not end.

So Front-End is the part of the web where the user can interact with websites. website is nothing but bunch of files , buttons , images , videos comment boxes , forms and many more are part of Front-End.

basically the difference between Front-End and Back-End in simple language is if you are working or writing code for the browser then you are doing Front-End Development. and if you are working or writing code for server then you are doing back end development.

We will divide the roadmap into two parts which are basic and advanced. so we'll cover basic firstly. you need to learn HTML , CSS and JavaScript this is the base of Front-End. Learn HTML First Hardly it will take 7 to 10 days then come to CSS it have huge concepts , you don't need to learn everything just the things than can help you build a website.

Further you need to understand how to create responsive layouts and how media queries works how it helps to make website mobile and tab friendly.

Once you learned HTML and CSS , you should create at least 4 to 5 projects , which can be small, two-pages , three-pages Up to as many you want to create.

To become better at something , you have to do it yourself. Spend the minimum amount of time watching and more and more time doing.

After learning HTML and CSS , you need to learn how JavaScript works , how loops statement-conditional statements works , how switch cases work , how variables work , how functions are created , arrays and objects , and the most important DOM

Now while learning JavaScript you can do small tasks and projects , and when you have learning DOM well in JavaScript, you have learned events well , you know how to manage events , then you should make small projects of JavaScript.

You will realize that CSS is very time consuming because it's hard to debug and the files can become unmanageable very quickly . That's why you need a CSS framework. There are many options , there are 2 popular ones. the first one is Bootstrap which comes with pre-built components but you would not have complete control over what your website look like.

Second is Tailwind CSS which gives you good control over the look and feel of your website and gets rid of all the unused CSS automatically for you.

Now moving to the SASS which CSS preprocessor , that is used to write CSS fast and also reduces the time period of code. it is not hard not an complex concept.

Further we will move on Advanced JavaScript now this will take time at least 5 to 6 months , mastering the core language is essential for effectively using any framework . it allows developers to understand how the framework works, diagnose issues, customize the framework , optimize performance and create their solutions when necessary.

Create 4..8 projects by yourself best way to master is doing practice.

Once you complete Advanced JavaScript after completing the projects then finally move on framework/library of JavaScript. Now I recommend you to learn ReactJS. Due to its flexibility , performance , smaller learning curve and large community and resources. if you know JavaScript vey well then react will be easy for you.

However Next JS is framework which is best for server-side rendering. Using Next JS web Apps Built very Efficiently. where as react is a awesome library for Building UI. To learn Next JS you need to make foundation strong which means React JS.

However the choice between React JS and Angular JS definitely depends on the specific requirements of the project.

In Front-End you can clone any website , you don't have to focus on designing here, you have to see the design from somewhere and then you have to develop it.

Further you can learn about VCS(Version Control System) and NPM(Node Package Manager). VCS is a tool used to manage changes to source code over time, the most popular VCS used in web development is Git. And NPM is package manager for node.js application it provides a way to install , update and manage third party packages Etc.

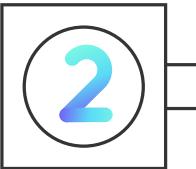
To make the development process more efficient and streamlined we use React UI Library.

Learn about SEO , which helps to optimize your website and its content to improve its visibility , improve user experience.

Next Learn about Deployment , Netlify is one of the best platform that allows you to deploy , manage and scale modern web projects . it provide wide range of features and benefits for you.

When you have gained good skills in Front-End , you can go for the Back-End.

BACK END



2



Back-End-Dev

01

Rest API

02

Node JS

03

Python

04

Mongo DB

05

AWS



Back End Dev Should Know

- ◆ Base Knowledge of Front End Technology
- ◆ Server Handling
- ◆ Knowledge of APIs
- ◆ DSA (Data Structures and Algorithms)
- ◆ Problem Solving
- ◆ Communication



Node JS

Node JS is Open source cross platform JavaScript Library for running web application outside the client's browser. learn about Modules , Console , Cluster , Global , Error Handling , Streaming , Buffer , Domain , DNS Debugger. Etc.

Python

Python is a interpreted programming language which is mostly used in various fields such as web development, scientific computing, data analysis, artificial intelligence, and more.

MongoDB

MongoDB is a document-oriented database which stores data in JSON-like documents with dynamic schema. learn about Environment , CRUD Operation ,Indexing , Replication , Aggregation and many more Etc.

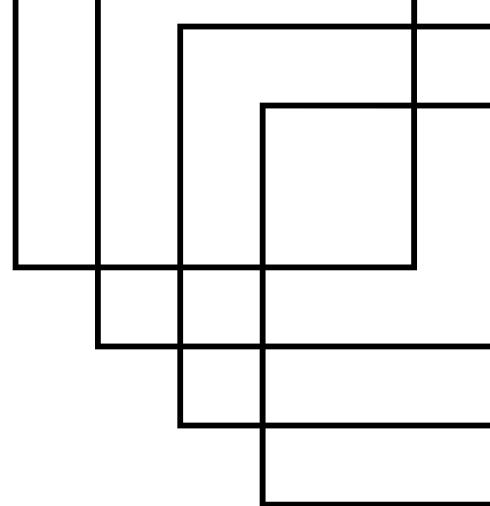
AWS

So AWS (Amazon Web Services) is a cloud computing platform which offers services and tools for building , deploying and managing application and services in cloud.

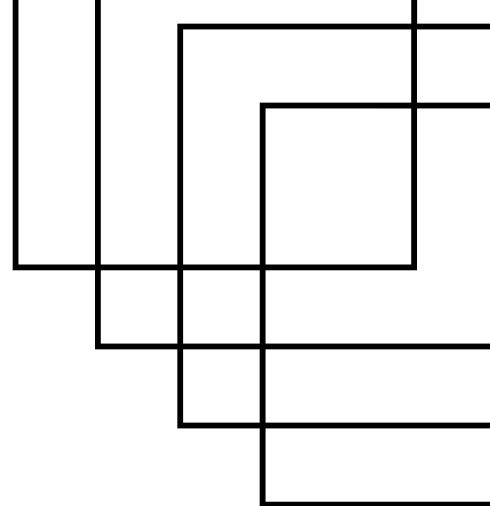
Back-End Interview Question

Node JS Question's

- How does Node.js handle concurrency?
- What is an event loop in Node.js?
- What is a callback function in Node.js?
- What is a module in Node.js?
- How do you create a server in Node.js?
- What is npm in Node.js?
- What is the purpose of package.json in Node.js?
- How do you check that a value is a date object in Node?
- What is the command to silence all process warnings?
- Which console method can be used to print the stack trace to the point of its execution?
- What is the purpose of the path module?
- How do you make an HTTP server object active and listen to requests on certain ports?
- What is the command to get a list of available commands for Node.js ?
- What is the difference between process.nextTick() and setImmediate() in Node.js?

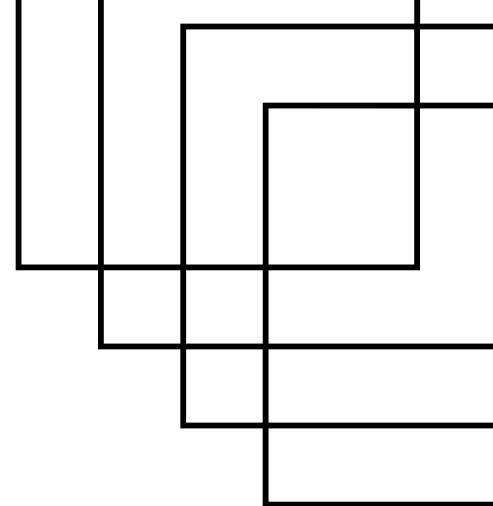


- What is the global object in Node.js?
- What is the purpose of the process object in Node.js?
- What is the purpose of the util module in Node.js?
- How do you read and write a file in Node.js?
- How do you create a child process in Node.js?
- What is the purpose of the cluster module in Node.js?
- What is the purpose of the net module in Node.js?
- What is the https module in Node.js?
- What does the .node file extension represent?
- What is the purpose of the crypto module in Node.js?
- Which core Node module has wrappers for OpenSSL methods?
- What are the arguments passed to the module wrapper function?
- Which library provides Node.js with the event loop?
- When a request event is received in the HTTP module, what is the type of the first argument passed to that event, usually named req?

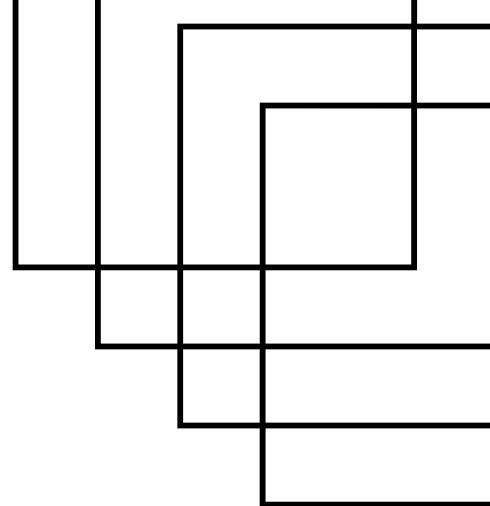


- Which line imports a promise-based version of the `readFile` method?
- What does REPL stand for?
- Which core module in Node can you use for testing?
- Which Node.js module should you use when you need to decode raw data into strings?
- What is the purpose of N-API?
- How can you convert path segments into a string using the platform-specific separator as a delimiter?
- Which global object acts like a bridge between a Node script and the host operating system?
- What is Node built on?

Python Questions

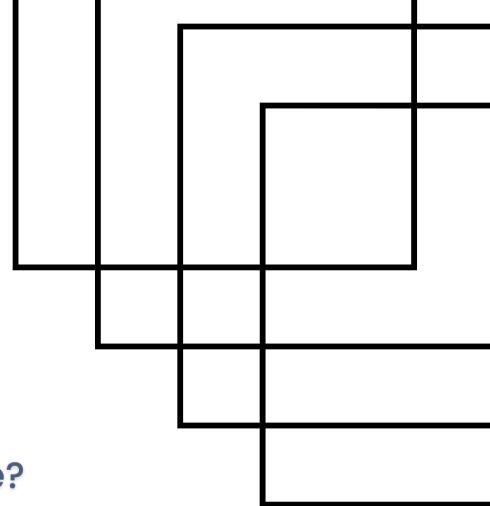


- What are the advantages of using Python?
- What is the difference between Python 2 and Python 3?
- What is PEP 8?
- What is an abstract class?
- What are attributes?
- What built-in list method would you use to remove items from a list?
- What is the runtime of accessing a value in a dictionary by using its key?
- What built-in Python data type is commonly used to represent a stack?
- What is a Python module?
- What is the difference between a list and a tuple?
- What is a generator in Python?
- What is a decorator in Python?
- What is a lambda function in Python?
- What is a closure in Python?
- What is a set in Python?
- What is the purpose of the range() function in Python?
- What data structure does a binary tree degenerate to if it isn't balanced properly?

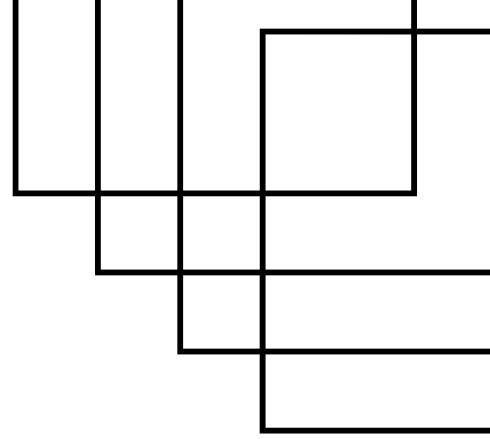


- What is the `zip()` function in Python?
- What is the purpose of the `enumerate()` function in Python?
- What is the `break` and `pass` statement in Python?
- What is the purpose of the `*args` syntax in Python?
- What is `init` method in Python ?
- What is `str` method in Python ?
- What does the built-in `map()` function do?
- Which collection type is used to associate values with unique keys?
- What happens when you use the built-in function `all()` on a list?
- What is the algorithmic paradigm of quick sort?
- What is runtime complexity of the list's built-in `.append()` method?
- What is key difference between a set and a list?
- What symbol(s) do you use to assess equality between two elements?
- What is meant by the phrase "space complexity"?
- What is the purpose of the `@classmethod` decorator in Python?
- What is the proper way to define a function?
- What is the correct syntax for creating a variable that is bound to a set?
- Why would you use mixin?

MongoDB Question's

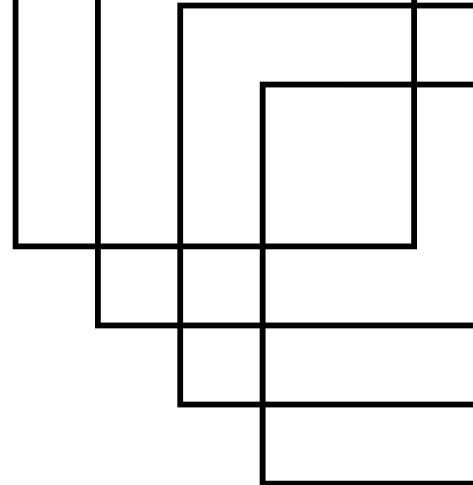


- How is MongoDB different from a relational database?
- What is a document in MongoDB?
- What is a database in MongoDB?
- What is a replica set in MongoDB?
- What is a primary node in a replica set?
- Which MongoDB shell command should you use to back up a database?
- Which shell query displays all citizens with an age greater than or equal to 21?
- What does a MongoDB collection consist of?
- Given an ObjectId in _id, how do you get the time it was created?
- To import a JSON array into Mongo, what flags are needed with MongoDBimport?
- Choose the shell command that connects to a MongoDB database.
- What is the recommended way to delete a user?
- What is the correct option to set up Kerberos when starting MongoDBd?
- How do you find documents with a matching item in an embedded array?
- What is a read preference in MongoDB?
- What is indexing in MongoDB?
- What is the purpose of the "distinct" command in MongoDB?



- What is the difference between "insert" and "insertOne" methods in MongoDB?
- What is the purpose of the "count" method in MongoDB?
- What is the purpose of the "aggregate" method in MongoDB?
- What is mapReduce method in MongoDB ?
- Which query bypasses the first 5 customers and returns the next 10?
- How do you create a text index ?
- Which shell query displays all citizens with an age greater than or equal to 21?
- What does a MongoDB collection consist of?
- Which command returns all of the documents in the customers collection?
- Which field is required of all MongoDB documents?
- Which MongoDB shell command deletes a single document?
- What is the recommended way to delete a user?
- To remove a database and all of its records from MongoDB, what operator should you use?
- From the MongoDB shell, how do you execute a JavaScript file named list.js?
- To cleanly shut down MongoDB, what command should you use from the MongoDB shell?
- What is the difference between "update" and "save" methods in MongoDB?
- To restrict the number of records coming back from a query, which command should you use?

AWS Question's

- 
- What are the benefits of using AWS?
 - What are the different services provided by AWS?
 - What is CloudFormation in AWS?
 - Which communication channel does SNS not support natively?
 - What is Route 53 in AWS?
 - What is SQS in AWS?
 - What is SNS in AWS ?
 - What is CloudWatch in AWS?
 - What is Autoscaling in AWS?
 - What is EFS in AWS?
 - What is Kinesis in AWS?
 - What is CloudTrail in AWS?
 - What is CodeBuild in AWS?
 - What is CodeDeploy in AWS?
 - What is CodePipeline in AWS?
 - What is indexing in MongoDB?
 - What option is best for Auto Scaling your EC2 instances for predictable traffic patterns?
 - Which big data store will let you store large streams of user activity data coming from both web and mobile applications?

Back end Perspective

Back-End development means in simple word you do code for server., which focuses on everything which we can't see on website. it is responsible for handling the logic and functionality of the web application. they use codes that helps browsers communicate with databases, store and delete data Etc.

At very first you need to what are protocols which HTTP and Rest APIs. If you can't write a simple get request in a notepad, you probably don't understand the HTTP protocol at least you need to know basic understanding about protocols.

Rest API provide a simple way to of building web services that can be used by different systems and applications.

Rest APIs uses HTTP protocol to transfer the data between client and server. Sometimes interviewer will ask about what are CRUD operation in Rest API.

Further A programming language is the third thing that comes to my mind when I think about creating a HTTP server.

The next step is to learn Linux and how the internet works. at beginner stage i would prefer Ubuntu.

You have to learn Linux commands and networking due to its Efficiency , Automation , Flexibility , Scalability Etc . as a as a back-end developer because you will be working with the HTTP protocol, the TCP protocol, and databases that have their own custom protocols.

Further Database is huge term used to store , manage , retrieve data used by the application. There are two types of databases on the top which are relational database and NoSql Database.

Relational Database includes MySQL , Oracle etc. and NoSQL includes MongoDB ,Firebase Etc. I recommend you go with MongoDB it has too fast performance and used to building scalable and flexible Back-End system if you are making for yourself it is best for you.

SQL you should learn to query from database tables. NoSQL doesn't use SQL, but if your application requires high performance and unstructured data, this might just be a great solution for you.

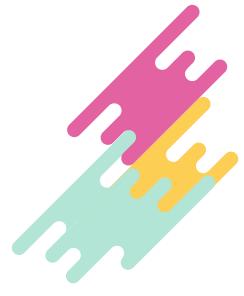
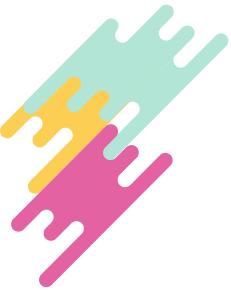
After that learn about Cloud Services if load comes on your computer then it will process slow, there many cloud provider like Google Cloud , AWS , Microsoft Azure , Personally I recommend you AWS which provide cost-effectiveness , flexibility , security Etc.

If you want to become a developer, you're going to have to make something really crappy at first and fail so hard, but that feeling of oh I finally got it is an addictive feeling that you need to make yourself earn.

If you want to build a strong backend for your web application, you have to build some projects. When you have gained some good skill inside the backend, and you have gained skills in the frontend, you can make more projects.

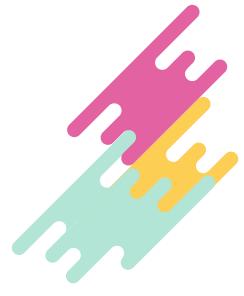
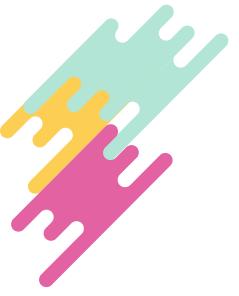
You will have to give some time for learning anything because it will not happen in 10 to 15 days . you have to give at least 5 to 6 month to learn all this stuffs.

Do practice as many you can or there is no other way to Master it.



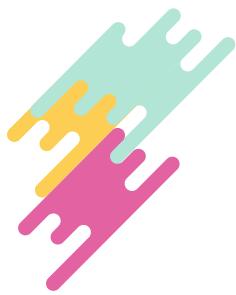
HTML & CSS Project Ideas

- ★ Personal Portfolio Website
- ★ Recipe Blog
- ★ E-commerce Website
- ★ Online Resume/CV
- ★ Music Player Website
- ★ Photography Portfolio
- ★ Online Store For Handmade Crafts
- ★ Fashion Blog
- ★ News Website
- ★ Sports Team Website
- ★ Fitness Website With Workout Plans And Recipes
- ★ Travel Blog
- ★ Charity Website
- ★ Educational Website For A Specific Topic
- ★ Event Website For Wedding , Conferences Or Festivals
- ★ Social Media Website With User Profiles And Feeds
- ★ Online Quiz Website
- ★ Video Tutorials Website
- ★ Job Board Website
- ★ Health & Wellness Website With Information on Meditation , Yoga.

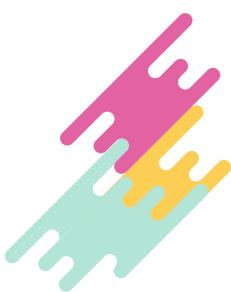


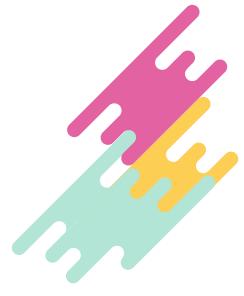
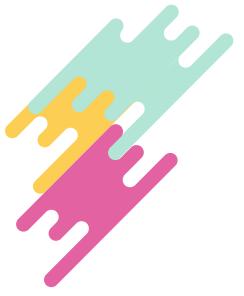
JavaScript Projects

- ★ To-Do List Application
- ★ Calculator
- ★ Countdown Timer
- ★ Quiz Game
- ★ Weather App
- ★ Currency Converter
- ★ Random Quote Generator
- ★ Calendar Application
- ★ Digital Clock
- ★ Browser Extension
- ★ Pomodoro Timer
- ★ Password Generator
- ★ Memory Game
- ★ Hangman Game
- ★ BMI Calculator
- ★ Typing Speed Test
- ★ Tic Tac Toe Game
- ★ Rock Paper Scissors Game
- ★ Image Slider/Carousel
- ★ Image Gallery



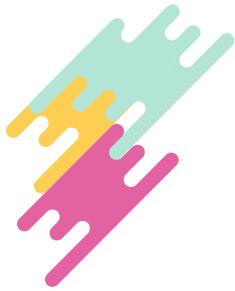
- ★ Quiz Application With Timer & Scoring
- ★ Expense Tracker
- ★ Text Editor
- ★ Suduko Game
- ★ Snake Game
- ★ Chat Application
- ★ Drawing Application
- ★ Music Player With Playlist and Controls
- ★ Memory Card Matching Game
- ★ Calculator With History & Memory Functions



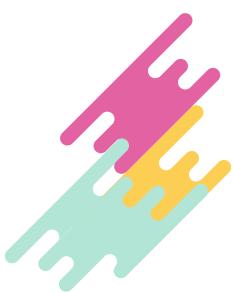


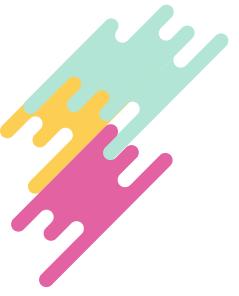
ReactJS Projects

- ★ To-Do List Application
- ★ Weather App with real-time data
- ★ News Aggregator Website
- ★ Recipe Search Engine
- ★ Expense Tracker With Budgeting Features
- ★ Social Media Dashboard
- ★ E-commerce Website With Shopping Cart And Payment Gateway
- ★ Movie Search Engine with Trailers And Rating
- ★ Multi-page Website With Routing And Navigation
- ★ Chat Application With Real-Times Updates
- ★ Dashboard For Displaying Financial Data
- ★ Music Streaming Website With Playlist And Controls
- ★ Online Resume Builder
- ★ Quiz Resume Builder
- ★ Quiz Application With Timer And Scoring
- ★ Task Management Application
- ★ Job board website with filters and search functionality
- ★ Forum Website for discussion and Q&A
- ★ Flight Booking website with real-time updates
- ★ Social Media clone with posts, likes , and comments

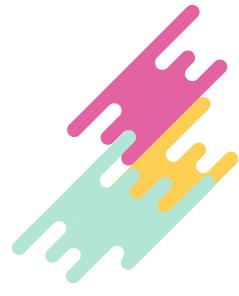


- ★ Blog with CMS for managing articles
- ★ Travel planner with map and itinerary builder
- ★ Event Management application with registration and ticketing
- ★ Recipe book with search and filter functionality
- ★ Hotel booking website with availability and booking functionality
- ★ Real estate website with property listing and filters
- ★ Fitness tracker with exercise and meal planning
- ★ Online education platform with courses and quizzes
- ★ Digital portfolio website
- ★ Multi user chat application with authentication and authorization
- ★ Appointment booking system for scheduling and reminders

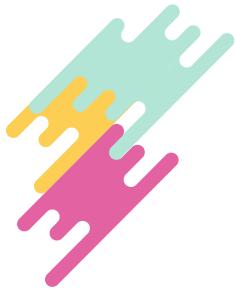




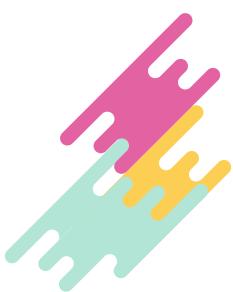
NodeJs + MongoDB Projects

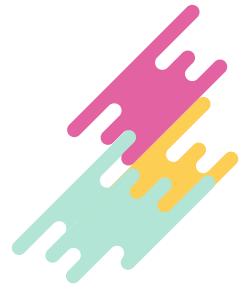
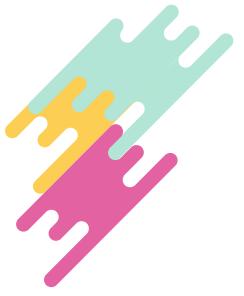


- ★ Real-time chat application with message history and search functionality
- ★ Social media platform with posts , likes and comments
- ★ Blogging Platform with user authentication and authorization
- ★ E-commerce website with shopping cart and payment gateway
- ★ Digital marketplace for buying and selling products
- ★ Online ticket booking system for events or movies
- ★ Food Ordering and delivery system
- ★ User Feed back and review system with rating and comment features
- ★ Forum website with user filters and search functionality
- ★ Online Education platform with courses and quizzes
- ★ Online appointment booking system for healthcare professionals
- ★ Digital wallet application for managing transactions
- ★ Event management platform with registration and ticketing
- ★ CRM system for managing customer data
- ★ Task management application with scheduling and deadline reminders
- ★ Online reservation system for hotels or rental properties
- ★ Employee management system with payroll and attendance tracking
- ★ Automated email marketing system with analytics
- ★ E-learning platform for teachers and students
- ★ Freelance job board with bidding and payment system



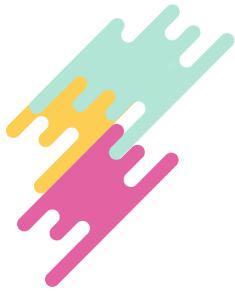
- ★ Social network analysis tool for analyzing relationships
- ★ Stock trading simulation game with real-time data
- ★ Audio and video streaming website
- ★ Document management system with version control
- ★ Fitness tracking application with exercise and meal planning
- ★ Geo-location based services platform
- ★ IoT dashboard for managing devices and data
- ★ Real-time analytics dashboard for business metrics
- ★ AI chatbot for customer support or personal assistance.



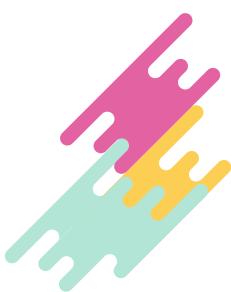


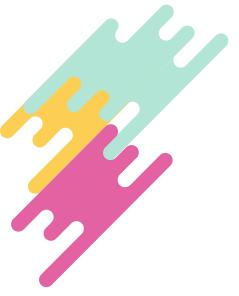
Python Projects

- ★ Web scraper for collecting data from websites
- ★ Data analysis and visualization tool
- ★ Twitter sentiment analysis tool
- ★ Email automation system
- ★ Web-based chatbot for customer support
- ★ GUI calculator with history and memory
- ★ Image processing and manipulation tool
- ★ Quiz application with timer and scoring
- ★ Password generator with strength meter
- ★ Text summarization tool
- ★ Automated web testing tool
- ★ CLI-based to-do list application
- ★ Automated invoice and billing system
- ★ Sudoku game with difficulty levels
- ★ Stock trading simulation game with real-time data
- ★ Automated attendance system for classrooms or meetings
- ★ Online store with shopping cart and payment gateway
- ★ Forum Website for discussion and Q&A
- ★ Content management system for blogs or websites
- ★ Weather forecasting tool with real-time data

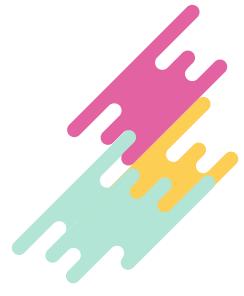


- ★ Digital clock with different time zones
- ★ URL shortener with analytics
- ★ Language translator with multiple language support
- ★ Voice recognition and speech-to-text tool
- ★ Automated email sender for newsletters or marketing campaigns
- ★ Music player with playlist and controls
- ★ Data encryption and decryption tool
- ★ Social media sentiment analysis tool
- ★ Automated image captioning tool
- ★ SMS reminder system for events or appointments
- ★ Scientific calculator with advanced features.

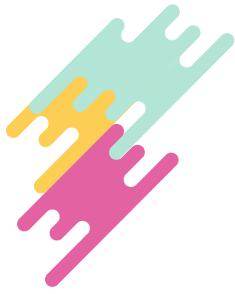




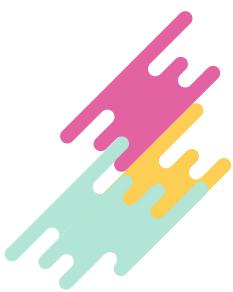
Front-End Projects

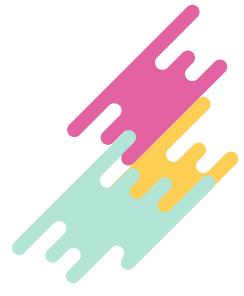
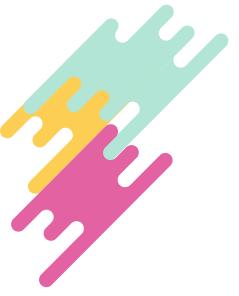


- ★ Online portfolio website with animations and interactions
- ★ Online quiz application with timer and scoring
- ★ Interactive map with markers and info windows
- ★ Real-time dashboard for analytics and data visualization
- ★ Photo gallery with slideshow and filters
- ★ Real-time stock market data visualization tool
- ★ Event registration and management platform
- ★ Multi-step form with validation and progress bar
- ★ Real-time chatbot for customer support
- ★ News website with filtering and search functionality
- ★ Digital resume or CV website with animations and interactions
- ★ Interactive timeline with markers and events
- ★ Flight booking website with real-time updates
- ★ Multi-user chat application with authentication and authorization
- ★ Social network analysis tool for analyzing relationships
- ★ File uploader and downloader with authentication and authorization
- ★ Email verification system with security features
- ★ Interactive world map with data visualization
- ★ Interactive calculator with animations and custom themes
- ★ Animated infographic with data visualization



- ★ Real-time multiplayer game with chat functionality
- ★ Real-time news ticker with animations and interactions
- ★ Video player with playback controls and captions
- ★ Animated progress bar with custom themes
- ★ Interactive quiz game with leaderboard and sound effects
- ★ Interactive 3D product viewer with zoom and rotation
- ★ Interactive slider with custom themes and animations
- ★ Customized photo editor with filters and effects





Back-End Projects

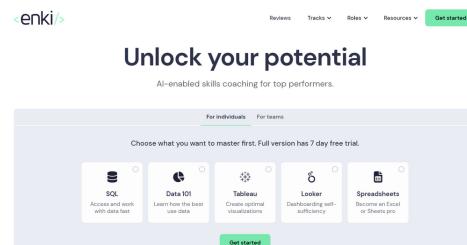
- ★ Content Management System (**CMS**) with user authentication and authorization
- ★ E-commerce platform with payment gateway integration and inventory management
- ★ Customer relationship management (**CRM**) platform with sales tracking and analytics
- ★ Cloud storage and file sharing platform with user permissions
- ★ Photo gallery with slideshow and filters
- ★ Online fundraising and donation platform with payment gateway integration
- ★ Online appointment scheduling system for salons and spas
- ★ Real-time cryptocurrency exchange with transaction tracking and reporting
- ★ Real-time bidding platform for auctioning off domain names
- ★ Online subscription platform for digital content
- ★ Online donation platform for non-profit organizations
- ★ Online property management system for landlords and tenants
- ★ Online legal services platform with document management and user permissions
- ★ Online photo printing and delivery platform with payment gateway integration
- ★ Online market research platform with reporting and analytics.
- ★ Online fundraising and donation platform with payment gateway integration

Resources For Full Stack Development

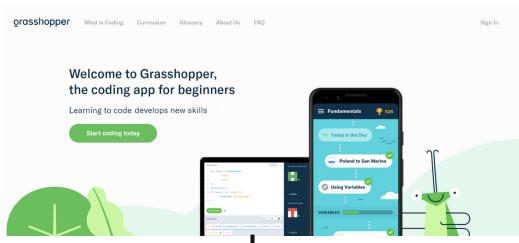
Apps to learn coding



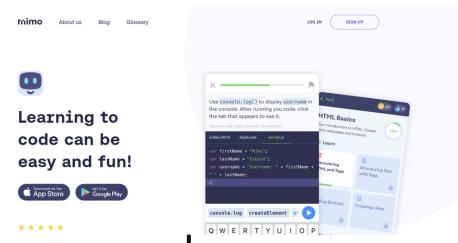
Sololearn



Enki

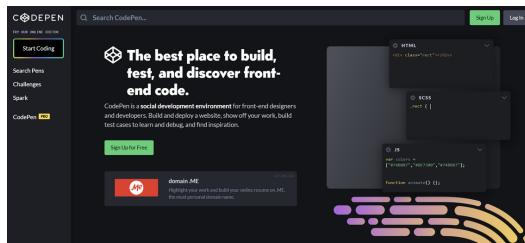


Grasshopper

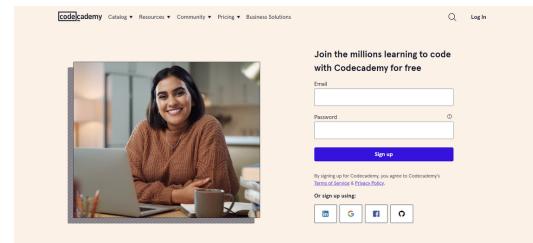


Mimo

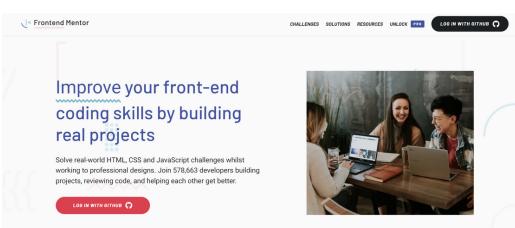
Websites to practice Front End



CodePen



Codecademy

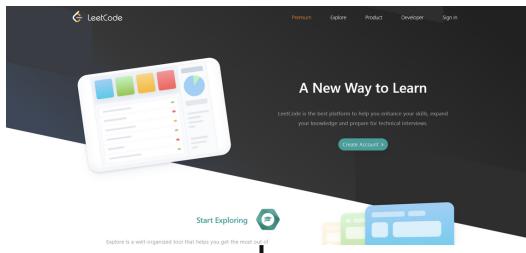


Frontend Mentor

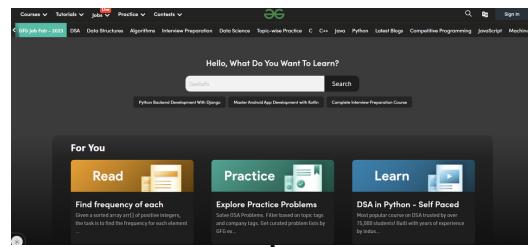


CSS Zen Garden

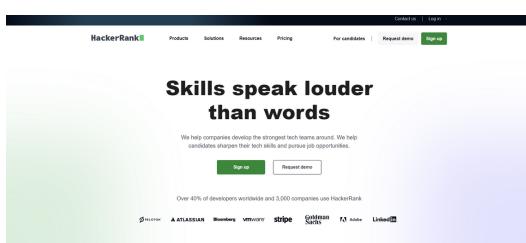
Websites to practice Coding



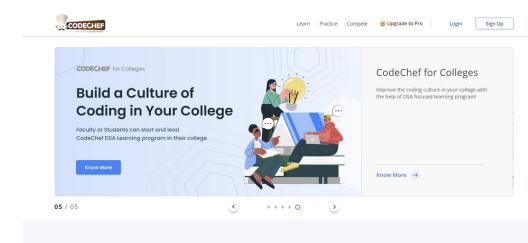
LeetCode



GFG

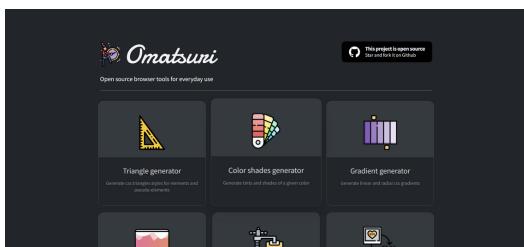


Hackerrank

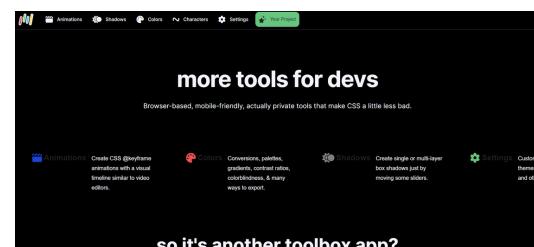


Codechef

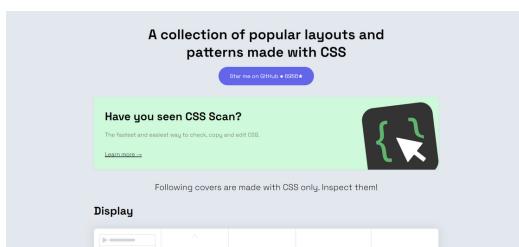
Free Tools For Developers



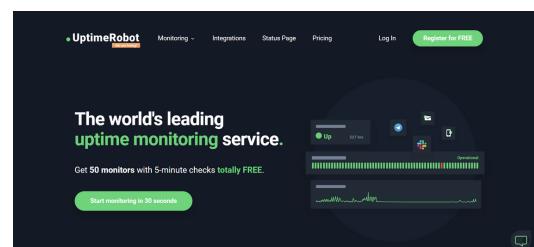
Omatsuri



Keyframes

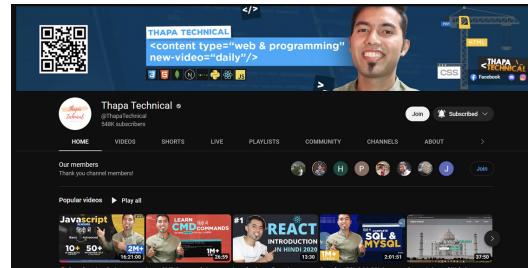
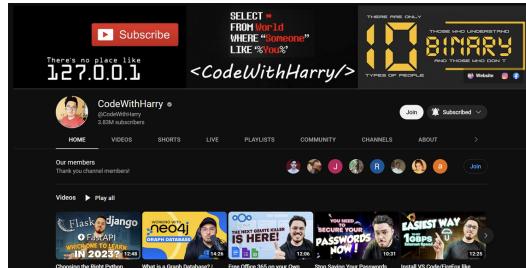


CSS Layout.io



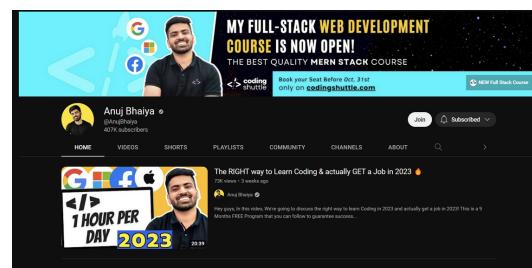
Uptime Robot

Best YouTube channels



Codewithharry

Thapa Technical



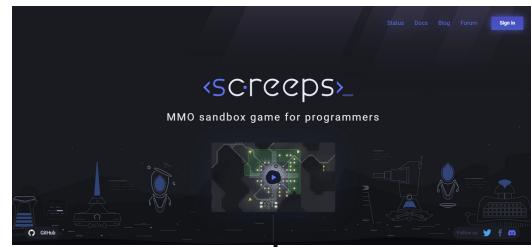
Apna college

Anuj Bhaiya

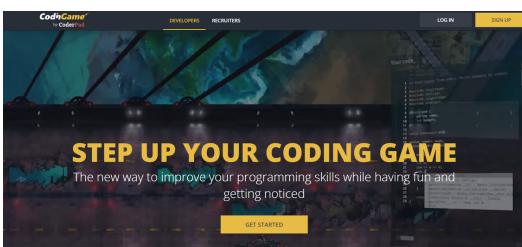
Games to learn Coding



Code Combat



Screeeps

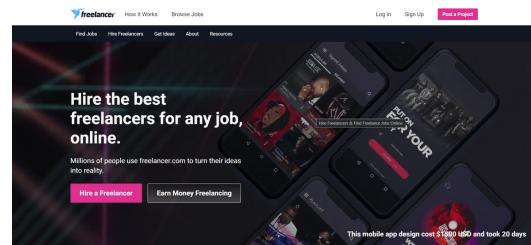
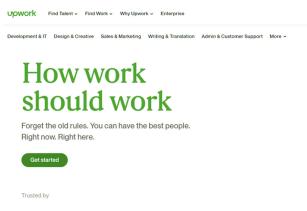


CodinGame



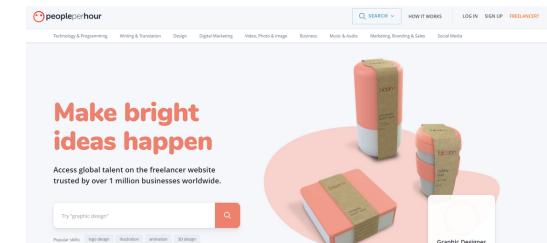
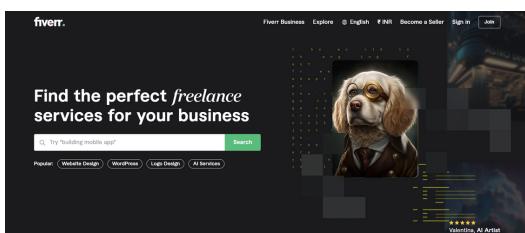
Checkio

Best Freelancing Sites



Upwork

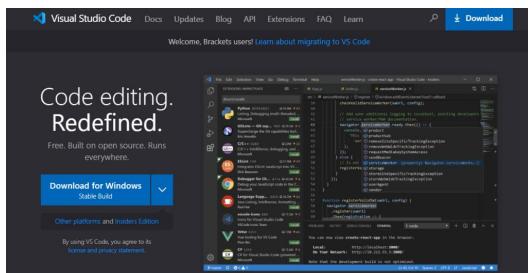
Freelancer



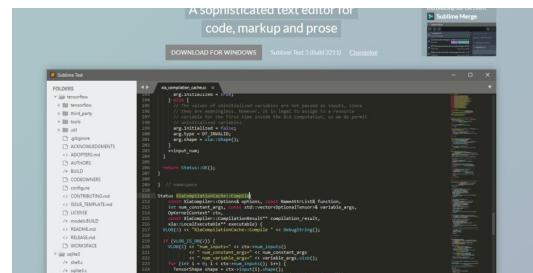
Fiverr

PeoplePerHour

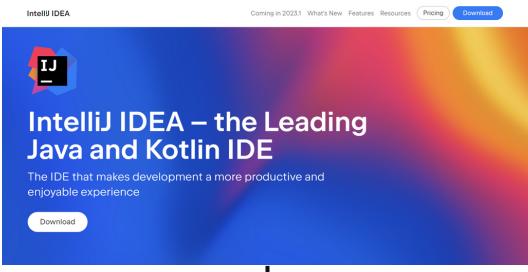
Best Code Editor



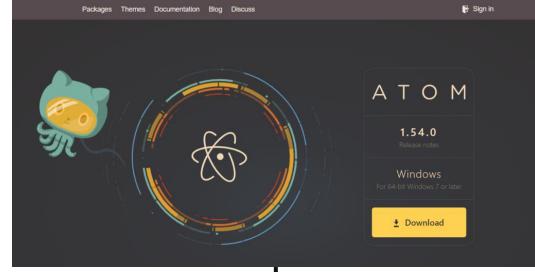
VScode



Sublime

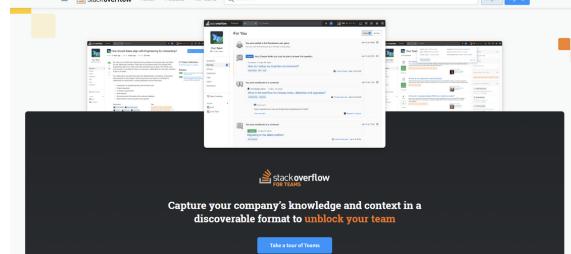
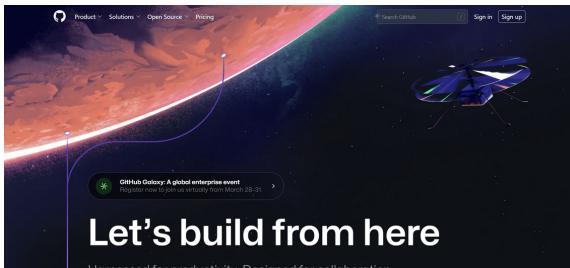


Intelij



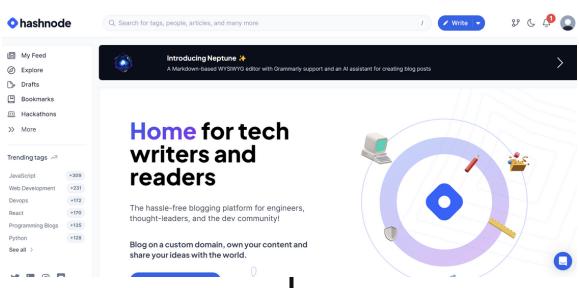
Atom

Developer Communities



Github

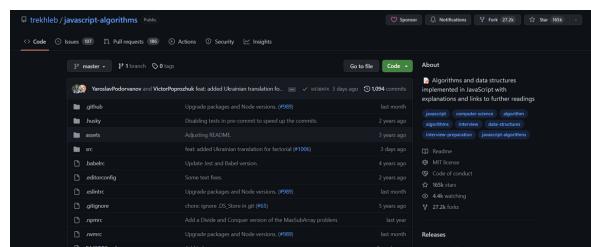
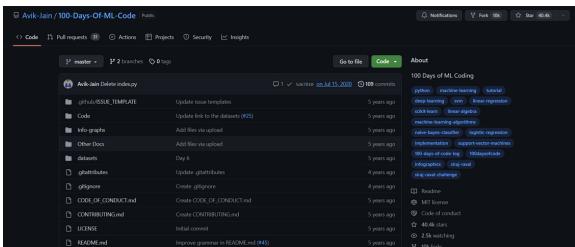
StackOverflow



Hashnode

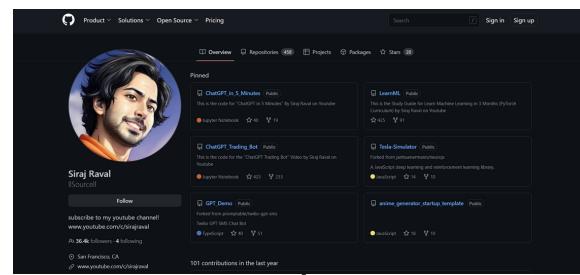
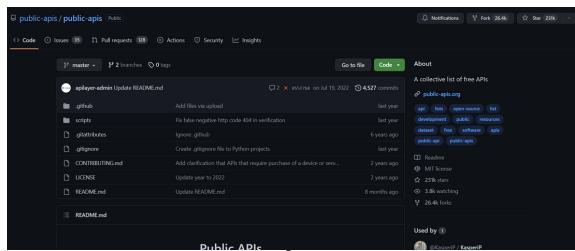
Hackernoon

GitHub Repository For Devs



100 Days Of ML

JavaScript Algo



Public APIs

Siraj Raval

UI
DESIGN



UI Design

Design Theory ,
What Is Actually
UI/UX Design &
Difference



Action
1

Start Learning About
Design Principles



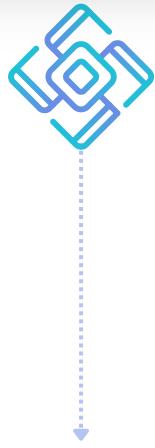
Action
2

Digital Psychology &
Methodologies



Action
3

Explore and Gain Knowledge of Trend and Style



Action
4

Learn essential Tools And Practice UI Design



Action
5

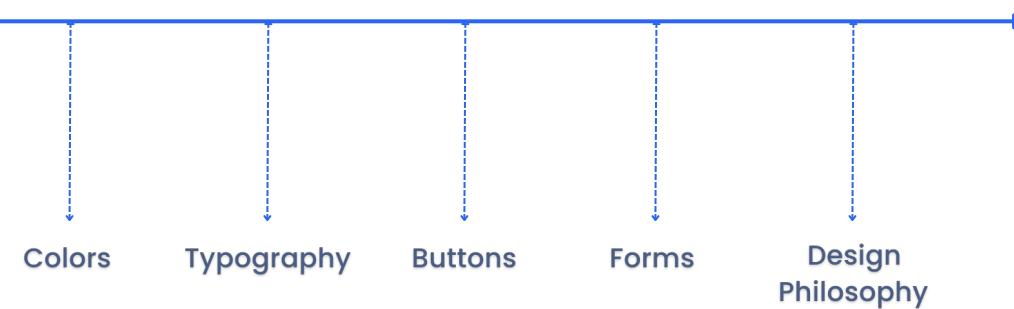
Gain Basic Knowledge of Technical Skills



Action
6

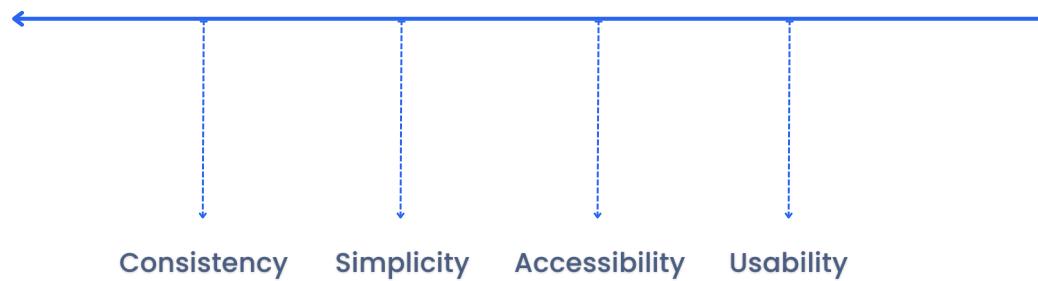
Action 1

If you are starting to learn UI/UX then you need to understand firstly about What is UI/UX design and the difference between them. In short UI Design Focuses on Visual Aspect of product In simple Language which we see from our eyes And UX Design focuses on the User experience Interaction with product.



Action 2

Learn about principles of design , design principles are important in designing Due to they helps to create creative interfaces which are more intuitive for users. Further learn about Law of UX





Action 3

Digital psychology and methodologies helps designers to create design with both functional and user-friendly. by understanding user behavior and designer perspective designer can make a Awesome positive emotional experience for the user.



Action 4

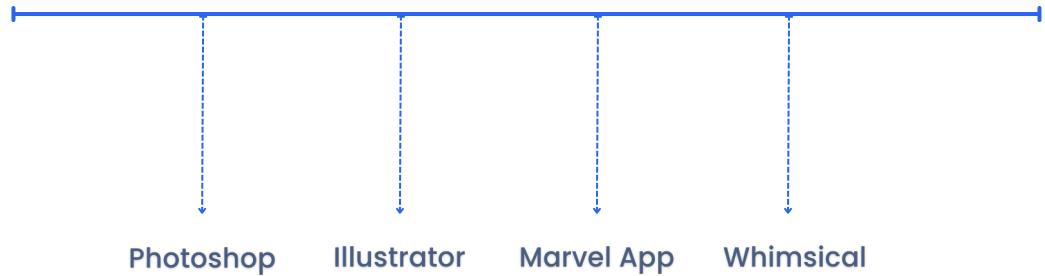
Keep exploring and engaging with trend and styles it is very useful to create modern , engaging , effective interfaces. many trends are there like 3D , Neumorphism , Dark mode , Skewmorphism and Material designs.



Action 5

Now its time to make UI by yourself . in my perspective for the beginner's Figma is the best platform . it is free and also have much features which helps you to make amazing UI. it supports plugin also. there are many tools apart from this Like sketch and Adobe XD.





Action 6

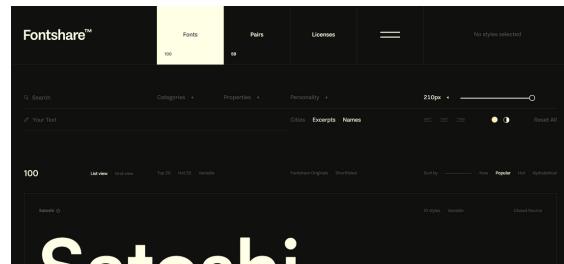
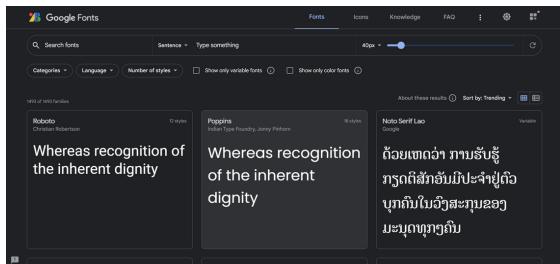


Designer should have a basic understanding of the screen sizes, pixel densities etc. Work with developers who are closely at first and technical skills helps to understand their actual need or concern towards project. you don't need to expertise coding but you should have the basic Knowledge of HTML and CSS.

Here are some quality resources

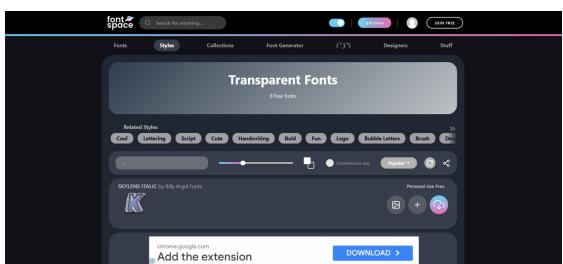
Free Fonts

This are Awesome website which offers a collection of high-quality font. that can be used for personal and commercial projects.



Google Fonts

Fontshare

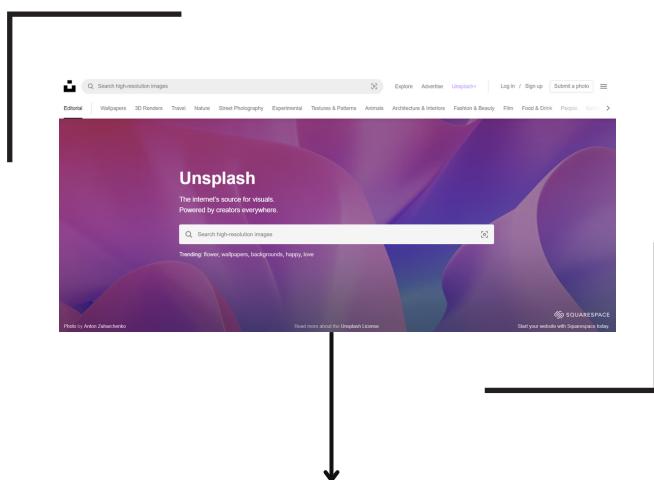


Fontspace

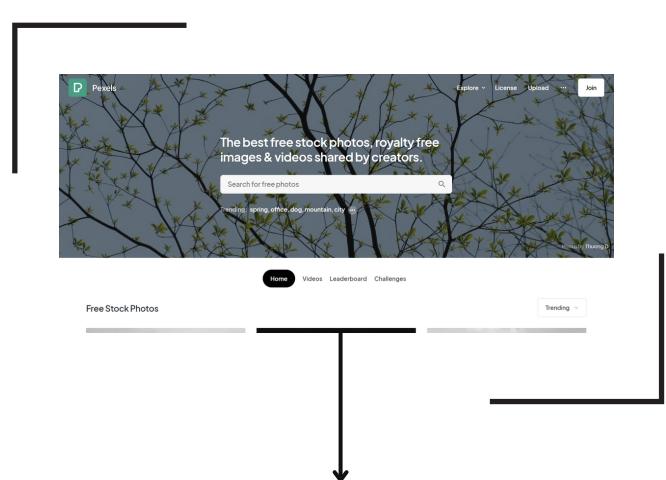
Fontsquirrel

Free Images

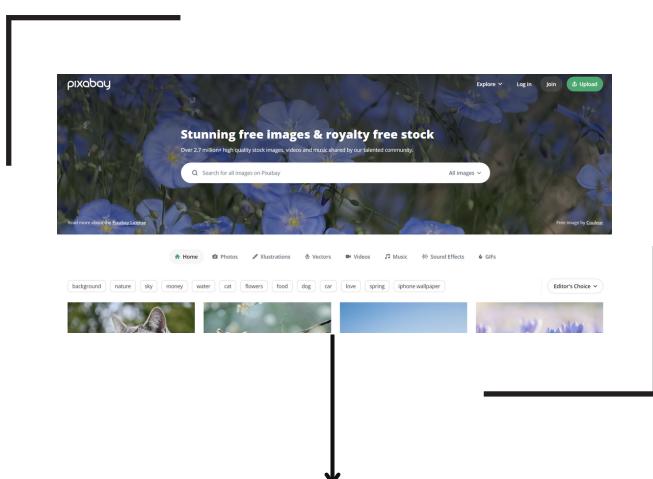
Over Million free high-resolution stock images share by creative or professional photographers.



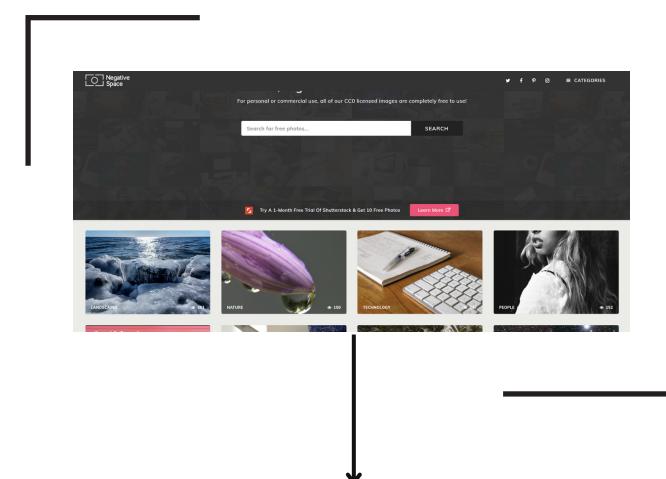
Unsplash



Pexels



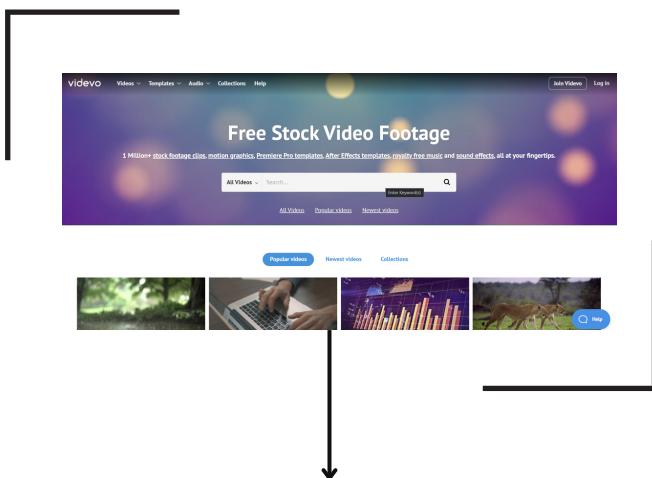
Pixabay



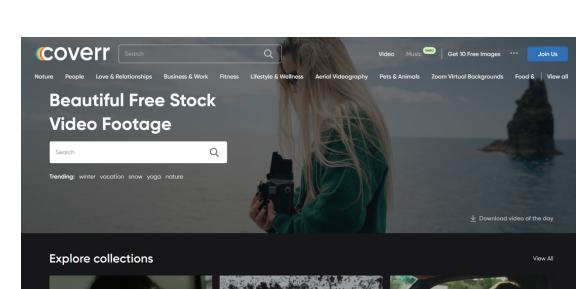
Negative space

Free Videos

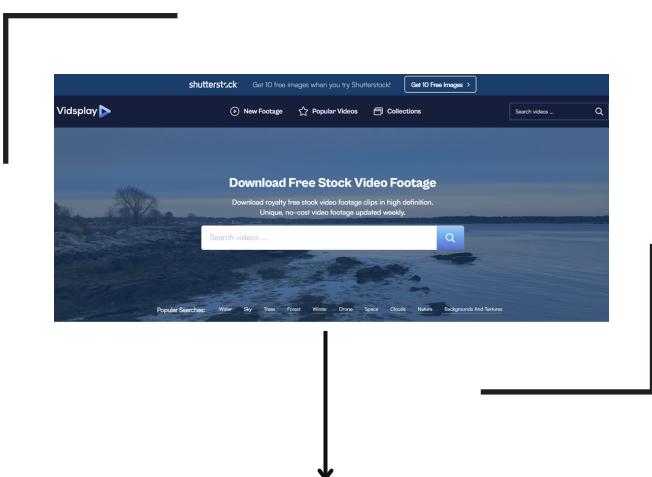
Free high-quality HD Videos , Clips , Arts and Beautiful footage.



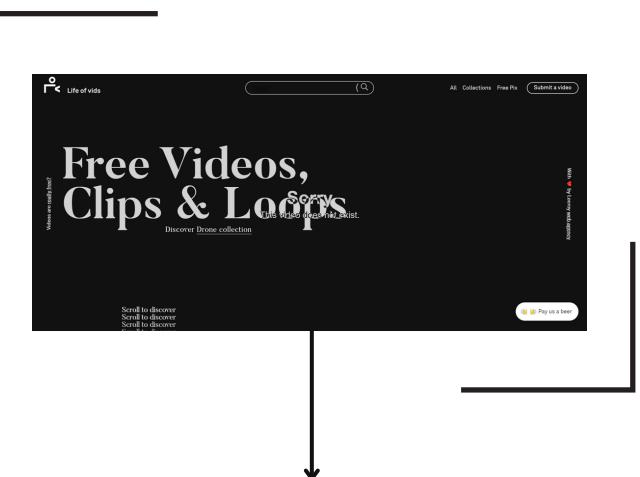
Videovo



Coverr



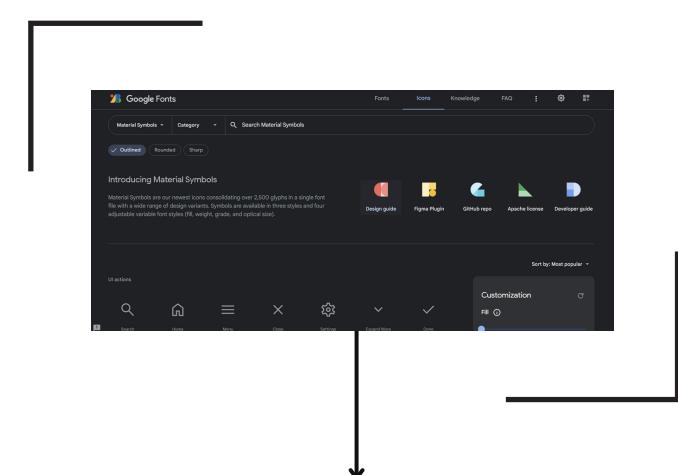
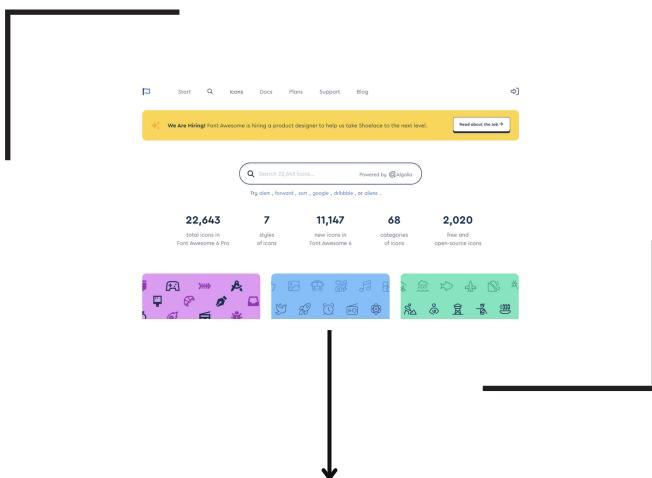
Vidsplay



Life of Vids

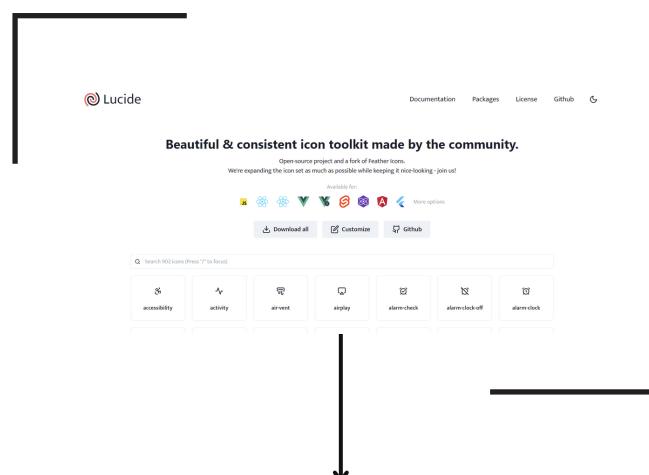
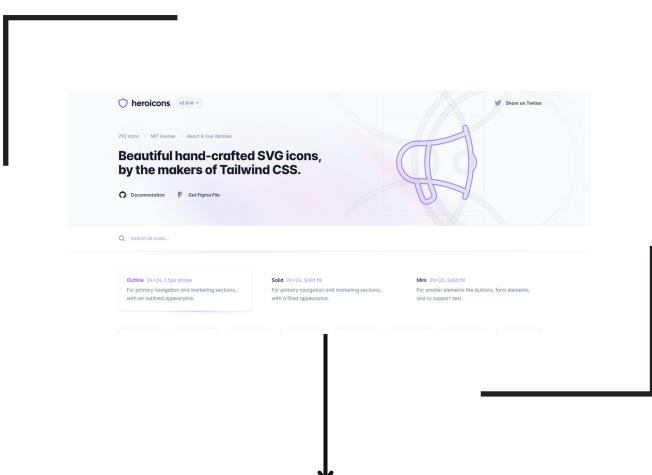
Free Icons

Amazing icons , SVG icons , Vector icons which you can use in your projects.



Fontawesome

Heroicons

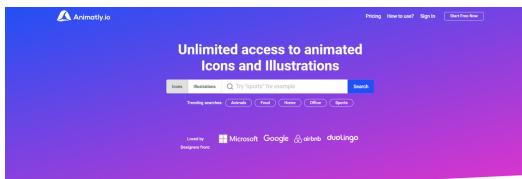


Lucide

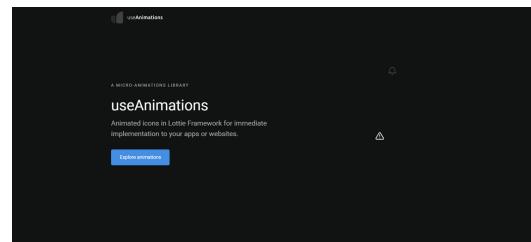
material icons

Free Animated Icons

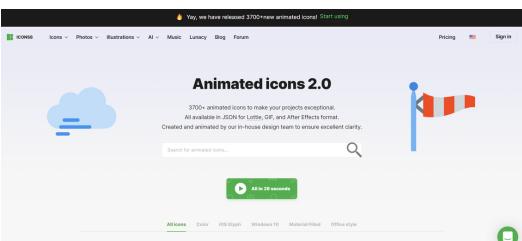
Beautiful Animated icons and beautifully designed and free and simple animated SVG Icon editor also Interactive web icons Etc.



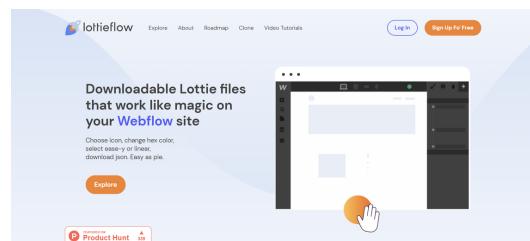
Animatly



UseAnimation



Icons 2.0



Lottieflow

Free Design & Prototyping Tools

Very powerful design and prototype tools which collaborate on rich user experiences.



Nothing great is
made alone.

Figma connects everyone in the design process
so teams can deliver better products, faster.

Get started

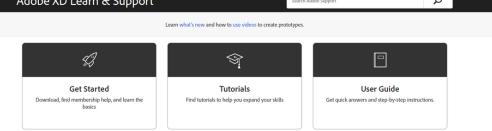
Design

Figma



Adobe XD Learn & Support

Learn what's new and how to use Adobe XD to create prototypes.



Troubleshooting & help

Common issues

Cloud documents

How do I file create, open, save, or data loss issues with XD cloud documents?

How do I fix publishing issues with design specs and prototype links?

Adobe XD



Sketch is where
great *design* happens.

A Mac app for designers to create, team up, prototype, and more.

A web app for everyone else to browse,
give feedback, inspect, and handoff - in
any browser.

A complete design platform, made by a
sustainable indie company since 2010.

Sketch



Pixso Products Resource Community Pricing Download Business Service Support English

User Interface/Experience Design
- Collaborative and Creative

Enhance digitized teamwork. Prototype, design, program and deliver using the one-stop tool Pixso.

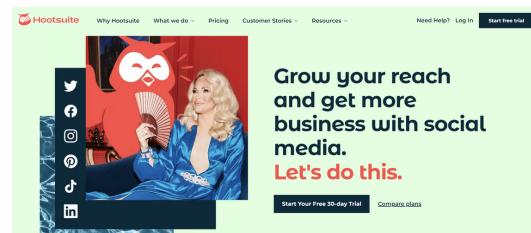
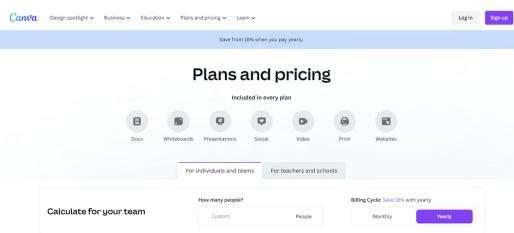
Get Free Account →



Pixso

Free Social Media Tools

Best free software , design or create anything in few minutes with beautiful templates. presentation , documents , videos , branded content and many more.



Canvas

Hootsuite

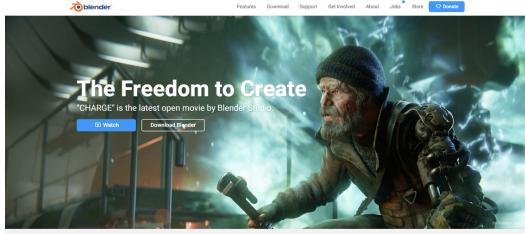


Visme

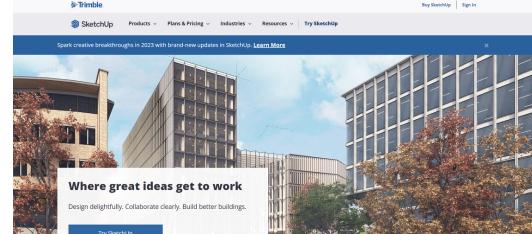
Glorify

3D Design Tools

Free easiest 3D Goopiest software's of your dreams , high-quality real-time 3D experience.



Blender



SketchUp



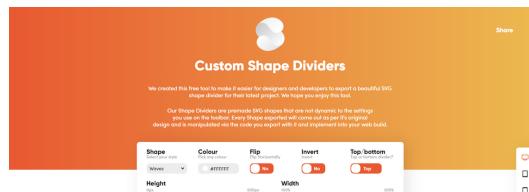
FreeCAD



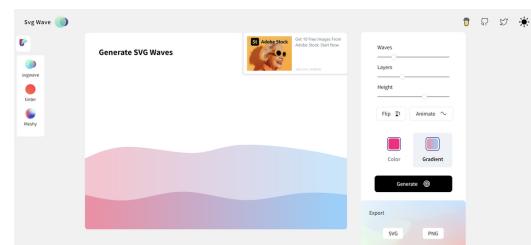
Womp

Free Shapes & Pattern Tools

Best tools for amazing SVG shape divider which create random , unique good looking organic SVG shapes. and pattern generators.



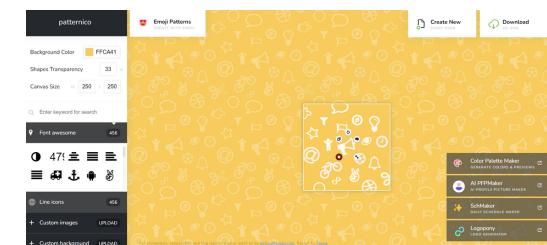
Shape Divider



SVG Waves



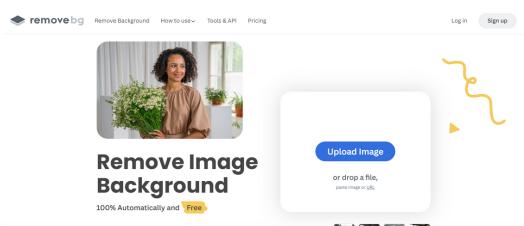
BGJar



Patternico

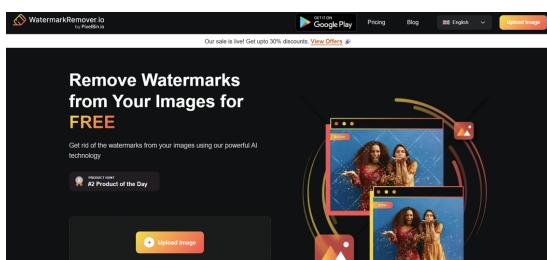
Free Background Remover

Remove background from images and download in high-quality resolution for free.



Remove.bg

Erase.bg

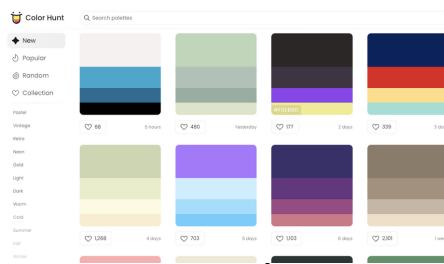


Watermark R

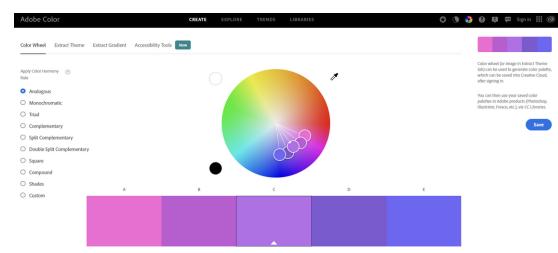
Unfakepng

Free Colors Apps

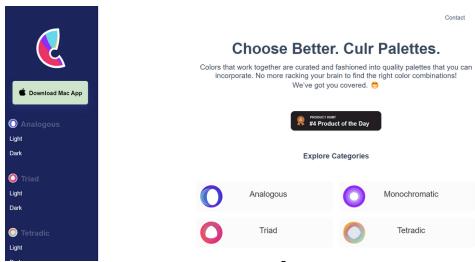
Here are 5 Open source color tools to create color themes and find colors simple and Eye catching.



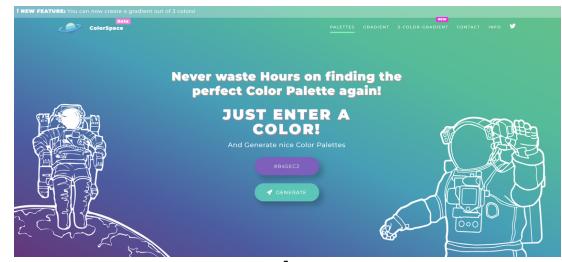
Color Hunt



Adobe Color



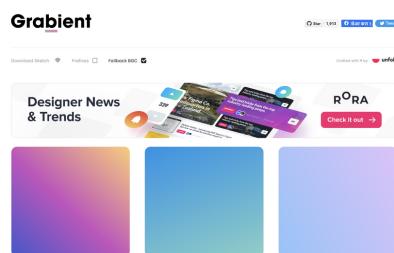
Curls



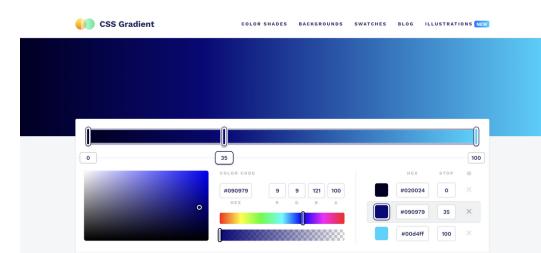
Colorspace

Free Gradient Websites

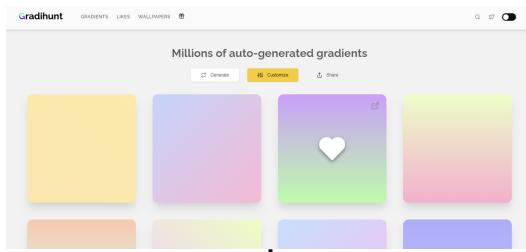
Beautiful color gradient combination websites .



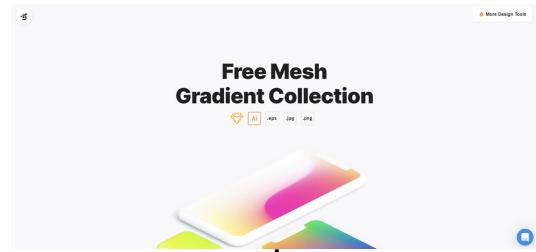
Grabient



CSS Gradient



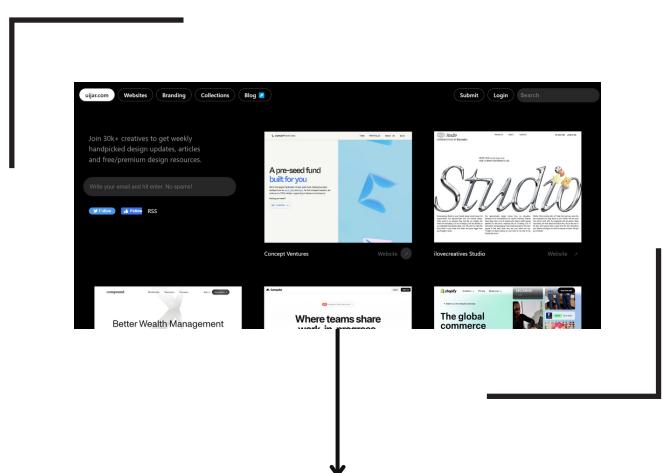
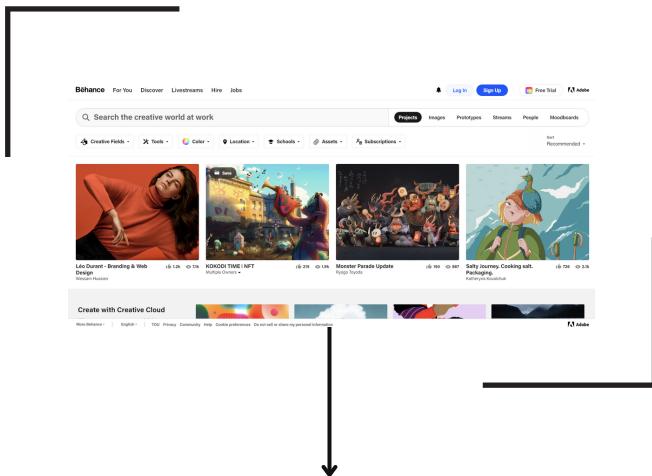
Gradihunt



Mesh Gradients

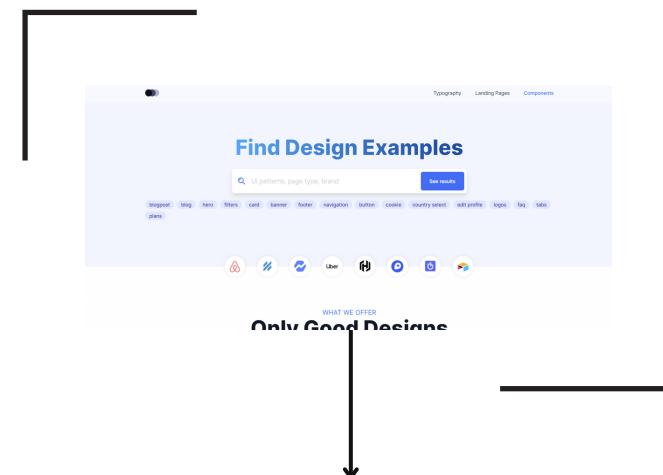
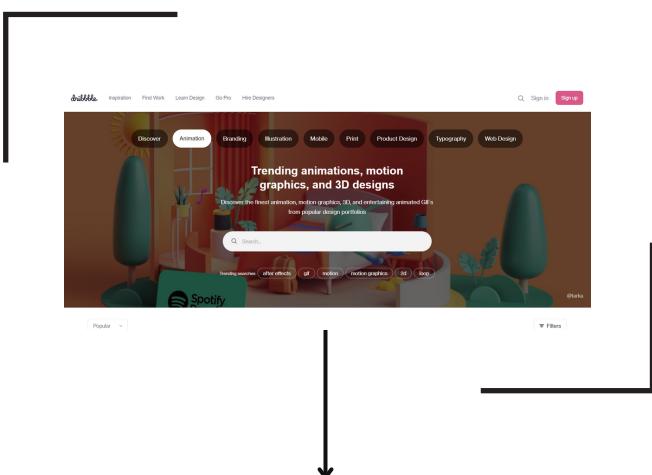
Design Inspiration Websites

This are the platform where community share their design project . screenshots of their project. you can also show your design or project here.



Behance

UI Jar

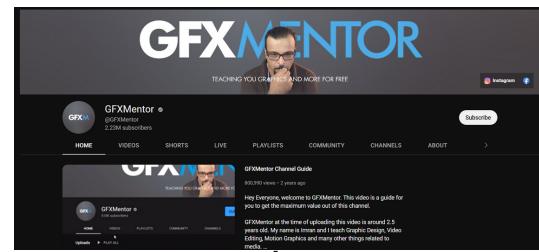
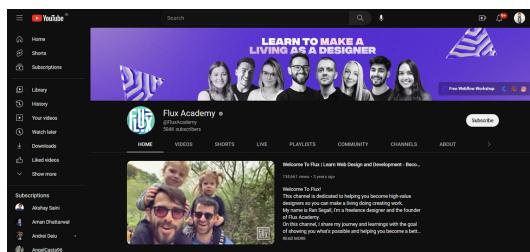


Dribbble.com

lookup.design

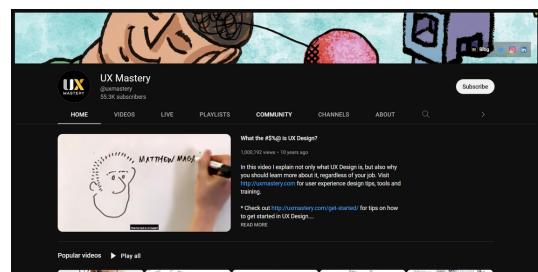
YouTube Channels

Here are some YouTube channel where you can start your learning process. which includes resources for UI/UX design thinking . UI design tutorials, and also about software like Figma adobe XD etc.



Flux

GFX Mentor



Design Course

UX Mastery

UI Perspective

Now a Days UI/UX design is becoming more popular but most of people don't know how to start learning or from where to start so this roadmap helps you to become a UI Designer with easiest path.

I have the roadmap very very simple to help people learn design users interfaces.

Firstly you need to understand what is UI/UX design and what's the difference between UI and UX.

To become a UI designer , you need to learn the fundamentals and design principles which are very important. because it helps to create effective and engaging designs that meet the needs of both the users and businesses.

Further you can visit website called Law of UX , which provides a set of principles and guidelines for designers to create user-friendly product. it describe how humans perceive attractive.

Keep in mind Folks do not try to by hard definitions and terms. only focus on understanding of it. There is lots of theory and you might be get confuse. don't worry you will going to focus only on how it looks and classic simple.

By understanding how user think and feel when interacting with digital product. UI designer are able to create awesome interfaces for that you need to understand Digital psychology and methodologies.

Explore more and get familiar with new trend and styles like dark mode , Neumorphism , Animations , Bold Typography , 3D design , Illustration. UI design trends and style are constantly evolving.

Once you have learned principles and psychologies , its time to learn a design tools. there many tools but at beginner stage I recommend you Figma. which is free , versatile tool allowing you to design , prototype and even collaborate with other designers in real time.

So now you can practice by solving a daily design challenges or re-designing existing websites and apps.

Behance is platform where you can publish your designs like show case and seek feedback.

you don't need to know how to code to create a portfolio . Behance allows create your professional looking portfolio in free.

Gain basic theory knowledge of technical skills like HTML and CSS. if your company expects you to do work outside of UI design this will help, however it might affect your growth as a product designer.

As a product designer you will also need to learn how to present your work. keep practicing at home until it becomes natural to you.

Knowledge without action is useless , so give time every day to learn something about UI design and execute the things that you have learned at least once a week.

And also you can learn from Social media platform like Instagram and YouTube and try to make sense of the design thought process.

you should always keep learning , even if you are not a beginner, to improve your design skills.

I have provided much quality of resources. like free images , icons and illustration and many more thing.

UI Project Ideas

- ★ News Feed
- ★ Sign Up Page
- ★ User profile
- ★ Shopping Cart
- ★ Progress bar
- ★ Chat Screen
- ★ Confirmation
- ★ Search Bar
- ★ Subscription Card
- ★ Movie card
- ★ Error Page 404
- ★ Job listing
- ★ Music app
- ★ Food app
- ★ FAQ App
- ★ Yoga App
- ★ Blog Post
- ★ Drop down menu
- ★ calculator
- ★ Weather App



- ★ E-commerce for clothing store
- ★ Landing page for travel company
- ★ UI for meditation and mindfulness app
- ★ Web-based dashboard for data analytic company
- ★ UI for a video edition software
- ★ Mobile app for language learning platform
- ★ Landing page for health and wellness website
- ★ UI for a financial planning app
- ★ UI for an online interior design platform
- ★ UI for a digital marketing dashboard
- ★ UI for grocery delivery app
- ★ UI for a fashion e-commerce website
- ★ Landing page for pet adoption website
- ★ Mobile app for mental health support platform
- ★ Landing page for furniture e-commerce website

Data Structure and Algorithm Questions

Array

- ❖ Reverse the array.
- ❖ Find the maximum and minimum elements in an array.
- ❖ Maximum Subarray.
- ❖ Minimise the maximum difference between heights [***]
- ❖ Search in Rotated Sorted Array.
- ❖ Repeat and Missing Number Array.
- ❖ Find the Union and Intersection of two sorted Arrays.
- ❖ Kth-Largest Element in an Array.
- ❖ Chocolate distribution problem
- ❖ Find Duplicate in an array of N+1 Integers.
- ❖ Next Permutation
- ❖ Kadane's Algo [***]
- ❖ Find the Minimum in Rotated sorted array.
- ❖ Median of 2 sorted arrays of equal size

- ❖ Subarray Sum Divisible K
- ❖ Trapping Rain water problem
- ❖ Minimum swaps required bring elements less equal K together.
- ❖ Minimum no. of operations required to make an array palindrome.
- ❖ Merge Overlapping Intervals
- ❖ Container with most water
- ❖ Best time to buy and sell stock
- ❖ Mo's Algo
- ❖ Space Optimization Using Bit Manipulation

String

- ❖ Reverse a string
- ❖ Check whether a string is palindrome or not
- ❖ Remove Consecutive Characters
- ❖ Valid parentheses
- ❖ Print all the Duplicates in the input String
- ❖ Print all the permutations of the given string
- ❖ Word Wrap Problem [***]
- ❖ Write a program to find the longest palindrome in a string
- ❖ Edit Distance [***]
- ❖ balanced parenthesis problem [***]
- ❖ KMP Algo
- ❖ Word break problem
- ❖ Split the binary string into two substring with equal 0's and 1's
- ❖ Count all Palindrome subsequence in a given string
- ❖ Wildcard String Matching
- ❖ Print all the Duplicates in the input string
- ❖ count of number given string in 2D character array.

- ✧ Minimum window substring
- ✧ Smallest window in a string containing all the characters of another string.
- ✧ Valid Anagram
- ✧ Recursively remove all adjacent duplicates
- ✧ Longest common Prefix
- ✧ Check if two given strings are isomorphic to each other.
- ✧ Minimum number of swaps for bracket balancing.
- ✧ Rabin karp Algo
- ✧ Find the First repeated word in string
- ✧ Program to generate all possible Valid IP addresses from given string

Searching & Sorting

- ❖ Find first and last positions of an elements in a sorted array.
- ❖ Search in a rotated sorted array
- ❖ Find the repeating and missing
- ❖ Find common elements three sorted arrays
- ❖ Searching in an array where adjacent differ by at most k
- ❖ Maximum & minimum of an array using minimum number of comparisons
- ❖ Find majority elements
- ❖ Inversion of array
- ❖ Radix sort
- ❖ Product array puzzle
- ❖ Count triplet with sum smaller than a given value.
- ❖ Product of array except itself
- ❖ minimum no. of swaps required to sort the array
- ❖ Rasta and Kheshtak
- ❖ Median of two sorted array with different size
- ❖ check if Reversing a sub array make the array sorted

- ✧ Kth smallest number
- ✧ Book Allocation problem
- ✧ Print Subarrays with 0 sum
- ✧ Job Scheduling Algo
- ✧ Implement Merge-sort in-place
- ✧ Counting Sort
- ✧ Kth-Element of two of two sorted array
- ✧ Sort array according to count of set bits
- ✧ Aggressive Cows

Linked List

- ❖ Remove duplicated in a un-sorted Linked List
- ❖ Merge Two sorted List
- ❖ Write a program to reverse the Linked List
- ❖ Reverse a linked list in group of given size [***]
- ❖ Remove nth node from end of list
- ❖ Detect and remove loop in a Linked List
- ❖ Write a function to get the intersection point of two Linked List
- ❖ Linked list in zig-zag fashion
- ❖ Reverse a doubly Linked List
- ❖ Write a program to move the last element to front in a Linked List
- ❖ Delete node which have a greater value on right side
- ❖ Rearrange a given Linked List in place
- ❖ Merge sort for Linked List [***]
- ❖ Find the middle element of a linked list
- ❖ Quick sort for linked list [***]
- ❖ Segregate even and odd elements in a linked list

- ❖ Merge K sorted linked list
- ❖ Find the first non-repeating character from a stream of characters
- ❖ Multiply 2 no. represented by Linked list
- ❖ Subtract two numbers represented as linked lists
- ❖ Remove duplicate in a sorted Linked list
- ❖ Rotate Doubly Linked List by N nodes.
- ❖ Sort a "k" sorted doubly linked list [***]
- ❖ Flattening a linked list
- ❖ Rotate a doubly linked list in group of given size [***]

Binary Trees

- ❖ Level order traversal
- ❖ Reverse level order traversal
- ❖ Height of tree
- ❖ Invert Binary Tree
- ❖ Mirror of a tree
- ❖ Subtree of another Tree
- ❖ Left view of binary tree
- ❖ ZigZag Tree Traversal
- ❖ Leaf at same level
- ❖ Diameter of a tree
- ❖ inorder traversal of a tree both using recursion and iteration
- ❖ Top view of the tree
- ❖ Check if a tree is balanced or not
- ❖ Diagonal Traversal of a binary tree
- ❖ Convert binary tree into doubly linked list
- ❖ Find minimum swaps required to convert a binary tree into BST

- ❖ Find distance between 2 nodes in a binary tree
- ❖ Find all duplicates subtree in a binary tree [***]
- ❖ Construct binary tree from string with brackets
- ❖ Kth ancestor of node in a binary tree
- ❖ Tree Isomorphism Problem
- ❖ Check if binary tree is sum tree or not
- ❖ Create a mirror tree from the given binary tree
- ❖ Check if given graph is tree or not [***]
- ❖ Find largest subtree sum in a tree

Binary Search Trees

- ❖ Find a value in a BST
- ❖ Delete of a node in a BST
- ❖ Minimum element in BST
- ❖ Find min and max value in a BST
- ❖ Kth smallest element in a BST
- ❖ Validate Binary search Tree
- ❖ Preorder and Postorder
- ❖ Check it a tree is a BST or not
- ❖ Find LCA of 2 nodes in a BST
- ❖ Convert binary tree in BST
- ❖ Merge two BST [***]
- ❖ Find Kth largest element in a BST
- ❖ Largest BST in a binary tree [***]
- ❖ Flatten BST to sorted list
- ❖ Populate Inorder Successor for all nodes
- ❖ Check preorder is Valid or Not
- ❖ Check whether BST contain Dead End
- ❖ Find Kth smallest Element
- ❖ Replace every element
- ❖ Construct BST from given preorder traversal

Backtracking

- ❖ Remove Invalid Parentheses
- ❖ Backtracking set 2 in a Maze
- ❖ Combinational Sum
- ❖ Printing all solutions in N-Queen Problem
- ❖ Word Break problem using backtracking
- ❖ M coloring problem
- ❖ Crossword-puzzle
- ❖ Longest Possible route in a matrix with Hurdles
- ❖ Solve the Soduko
- ❖ Knight Tour
- ❖ Tug of war
- ❖ Print all palindrome partitions of a string
- ❖ Find shortest safe route in path with landmines
- ❖ Partition of set into K subset with equal sum
- ❖ Find maximum number possible by doing at most K swaps
- ❖ Backtracking set-7 hamiltonian cycle
- ❖ Arithmetic expressions
- ❖ Find paths from corner cell to middle cell in maze

Stack & Queue

- ❖ Implement stack from scratch
- ❖ Implement Queue from scratch
- ❖ Implement Stack using queues
- ❖ Evaluation of Postfix Expression
- ❖ Implement two stacks in an array
- ❖ Find the middle element of a stack
- ❖ Reverse a String using Stack
- ❖ Reverse First K elements of queue
- ❖ Infix to Postfix
- ❖ Design Stack with Middle Operation
- ❖ Arithmetic Expression evaluation
- ❖ Reverse a stack using recursion
- ❖ Sort a stack using recursion
- ❖ Find if an expression has duplicate parenthesis or not
- ❖ Stack permutations check if an array is stack permutation of other
- ❖ Implement "n" queue in an array

- ❖ Implement a Circular queue
- ❖ The celebrity problem
- ❖ Iterative tower of hanoi
- ❖ Lru cache implementation
- ❖ Find a tour that visits all stations
- ❖ Largest rectangular Area in Histogram
- ❖ Next smaller element
- ❖ Rotten-oranges
- ❖ Distance of nearest Cell

Graphs & Trie

- ❖ Create a Graph and print it
- ❖ Clone graph
- ❖ Implement BFS Algorithm
- ❖ Search in a Maze
- ❖ Steps by Knight
- ❖ Detect Cycle in Directed Graph using BFS/DFS Algo
- ❖ Number of operation to make network connected
- ❖ Dijkstra's shortest path algorithm
- ❖ Making Wired Connections
- ❖ Implement Topological Sort
- ❖ Find the no. of Islands
- ❖ Implement Kruksal's Algo
- ❖ Implement Prim's Algo
- ❖ Travelling salesman problem
- ❖ Oliver and the Game
- ❖ Floyd Warshall

- ❖ Construct a trie from scratch
- ❖ Find shortest unique prefix for every word in a given list
- ❖ Word Break problem
- ❖ Given a sequence of words , print all anagrams together
- ❖ Implement a phone directory
- ❖ Print unique rows in a given boolean matrix
- ❖ Snakes and Ladders
- ❖ Journey to Moon
- ❖ Find bridge in a graph
- ❖ M coloring problem

Dynamic Programming

- ❖ Coin Change Problem
- ❖ Climbing Stairs
- ❖ BBT counter
- ❖ Maximum difference of zeros and ones in binary string
- ❖ Longest Repeating subsequence
- ❖ Knapsack Problem
- ❖ Edit Distance
- ❖ Permutation coefficient problem
- ❖ Subset Sum Problem
- ❖ Problem for nth catalan number
- ❖ Friends Pairing Problem
- ❖ Space Optimized solution of LCS
- ❖ Maximize the cut segment
- ❖ House Robber
- ❖ Assembly Line Scheduling Problem
- ❖ Painting the Fenceproblem
- ❖ Max length chain
- ❖ Egg dropping puzzle
- ❖ Longest common Substring
- ❖ Word Wrap

Greedy

- ❖ Activity selection problem greedy algo
- ❖ Job Sequencing problem
- ❖ Water Connection Problem
- ❖ Fractional Knapsack problem
- ❖ Minimum sum absolute difference pairs two arrays
- ❖ Find maximum height pyramid from the given array of objects
- ❖ Find maximum equal sum of every three stacks
- ❖ Greedy algorithm egyptian fraction
- ❖ Maximum length chain of pairs
- ❖ Find smallest number with given number of digits and digits sum
- ❖ paper-cut minimum number squares
- ❖ Check if it is possible to survive on Island
- ❖ Minimum cost to cut a board into squares
- ❖ Minimum platform problem
- ❖ Buy maximum stocks if stocks can be bought on i-th day
- ❖ Lexicographically smallest array-k consecutive swaps

- ❖ Problems-CHOCOLA
- ❖ Program for shortest Job First or SJF CPU scheduling
- ❖ Find minimum time to finish all jobs with given constraints
- ❖ Maximize the sum of $\text{arr}[i]*i$
- ❖ Maximum product subset of an array
- ❖ Find maximum meetings in one room
- ❖ Maximize array sum after K negations

Bit Manipulation

- ❖ Count set bits in an integer
- ❖ Find the two non-repeating element in an array of repeating element
- ❖ Find position of the only set bit
- ❖ Problem to find whether a no is power of two
- ❖ Count number of bits to be flipped to convert A to B
- ❖ Copy set Bits in a range
- ❖ Power set
- ❖ Calculate square of a number without using *, / and pow()
- ❖ Divide two integers without using multiplication division and mod operator

Conclusion

Finally now you know everything you need to know about Full-Stack Development and UI Design . The main motive of e-book was to give you clear and simple roadmap so that you can start your journey easily.

Now a days in market people are talking about there is no big scope in Front-End Development and Back-End Development. and hiring Only Full-Stack Developer. It's not like that Any field...literally Any field will never going to end. it is just about knowledge. if you have knowledge and skills then you will be hired.

Don't think about one of your friend doing good in coding and you are doing practice but not getting results and that's why you are comparing you to other person.

Keep in mind Do not compare yourself to Anyone, Instead compare yourself with yourself. remember you are creating something from nothing.

If you are Disciplined and you are following any path proper. results will be definitely come.

If you want to crack big company , it might be challenging but with proper roadmap or strategy and preparation you can crack. try to contact with people who are in that company in which you want to place. keep interacting with them on social media like Linkedin. Plus make your DSA strong . For Front-End Basic is necessary but For Back-End need Strong Knowledge of DSA

I will recommend you to Choose C language to do in DSA. Solve only one problem per day. and maintain this consistency.

You need to learn time and space complexity , recursion etc then stack , queues , linked list Etc. future learn about algorithms like searching after that you can learn maps , trees , heap then most important graph and dynamic programming etc.

you have to give it some time , it will take time this will not happen in 15 to 20 days. only you have keep in mind do not break consistency until unless you have exam.

At start you feel it is too hard but don't be afraid if it taking time just do practice. results will automatically comes to you.

I have given 240+ DSA questions. daily you have to choose problem one by one and start doing practice

Do Projects after Learning Each thing.

I hope folks This roadmap will help you. give your genuine feedback.



That's it for now Folks !

Connect with me on social media.



Instagram



Linkedin

**"If You Have A Hard Burning Desire
And Patience, Then You Will Achieve
Anything In Your Life. No Matter How
Hard It Get's , Never Give Up"**

All the best Folks , Stay Healthy

Om Ganesh Patil



THE END