Free full stack development: 30DaysCoding.com

Courses: https://courses.30dayscoding.com/s/store

Full stack dev topics

1. HTML, CSS
    a. Simple projects
2. Javascript 1
3. Javascript 2
    a. JS projects
4. React 1
5. React 2
    a. React projects
6. Next JS 12
7. Next JS 13
    a. Projects
    b. Templates
8. Node, express
9. DB
    a. MongoDB
    b. Firebase, supabase
10. Big projects
    a. Working on ideas
    b. MVP launches
    c. Freelance projects
    d. Industry level

Prerequisites

React: https://nextjs.org/docs/getting-started/react-essentials
Walkthrough: https://nextjs.org/learn/foundations/about-nextjs
Chat gpt prompts: https://chat.openai.com/share/8c5038bf-29e4-4d4d-b95f-7c08f0018400
https://chat.openai.com/share/931781cc-32fc-481c-b43c-3df5b82e579c

https://bit.ly/30dc-pe

Learn React

https://react.dev/
https://react.dev/learn/installation
https://legacy.reactjs.org/docs/introducing-jsx.html
https://www.geeksforgeeks.org/why-are-fragments-better-than-container-divs/
https://react.dev/learn/your-first-component
https://react.dev/learn/passing-props-to-a-component
https://legacy.reactjs.org/docs/hooks-effect.html

Next JS main topics

1. File structure
2. Next vs react
3. Simple component
4. Fetch API
5. <Link>
6. Image
7. Get static props
8. getserversideProps
9. /api folder
10. [slug].js
11. Graph ql
12. Firebase, supabase
13. Deployment - Vercel

Next js docs

https://nextjs.org/docs/getting-started/react-essentials
https://nextjs.org/docs/pages/building-your-application/routing/pages-and-layouts
https://nextjs.org/docs/pages/building-your-application/routing/dynamic-routes
https://nextjs.org/docs/pages/building-your-application/routing/linking-and-navigating
Extra: https://nextjs.org/docs/app/api-reference/functions

Auth

Next auth - https://next-auth.js.org/getting-started/example
Next auth - https://next-auth-example.vercel.app/

Firebase -
https://levelup.gitconnected.com/auth-with-firebase-for-your-next-js-react-app-in-2023-a30b65b5b2b1?gi=a46038b788e9

Context, reducer, action, Redux

https://react.dev/learn/extracting-state-logic-into-a-reducer
https://vercel.com/guides/react-context-state-management-nextjs
https://react.dev/learn/scaling-up-with-reducer-and-context
https://redux.js.org/tutorials/fundamentals/part-3-state-actions-reducers
https://www.freecodecamp.org/news/what-is-redux-store-actions-reducers-explained/

Database

Firebase:
https://kenanbek.medium.com/using-firebase-and-firestore-in-a-next-js-application-a-step-by-step-tutorial-5a9f22df18cf
Firebase: https://firebase.google.com/docs/firestore
https://supabase.com/docs/guides/getting-started/tutorials/with-nextjs
https://supabase.com/docs/guides/getting-started/quickstarts/nextjs
https://vercel.com/guides/using-databases-with-vercel
https://vercel.com/guides/integrating-next-js-and-contentful-for-your-headless-cms

Graph ql

https://www.airplane.dev/blog/graphql-vs-rest-apis-a-complete-guide
https://github.com/aagam29/graphql-clients/tree/main
https://hygraph.com/blog/nextjs-graphql
https://github.com/aagam29/graphql-clients/blob/main/pages/fetch.jsx
https://www.airplane.dev/blog/how-to-build-a-graphql-api-with-next-js
https://github.com/tomanagle/NextJS-NestJS-GraphQL-Starter

ORM - Prisma

https://vercel.com/guides/nextjs-prisma-postgres
https://www.prisma.io/nextjs
https://www.prisma.io/docs/guides/other/troubleshooting-orm/help-articles/nextjs-prisma-client-dev-practices

Next JS 13

next 13 - https://vercel.com/templates/next.js/prisma-postgres-auth-starter
https://vercel.com/templates/next.js/nextjs-portfolio-pageview-counter
https://vercel.com/templates/next.js/taxonomy
https://vercel.com/templates/next.js/strapi-nextjs-corporate-starter

Deploying

Firebase: https://firebase.google.com/docs/hosting/frameworks/nextjs
Vercel: https://vercel.com

Complete apps + tutorials

https://www.freecodecamp.org/news/nextjs-firebase-tutorial-build-an-evernote-clone/
https://www.freecodecamp.org/news/create-full-stack-app-with-nextjs13-and-firebase/
https://blog.jarrodwatts.com/the-ultimate-guide-to-firebase-with-nextjs
https://egghead.io/lessons/react-initialize-a-next-js-9-project-with-typescript

Github repositories

https://github.com/singh1aryan/firebase-everything
https://github.com/singh1aryan/nextjs-prisma-sql
https://github.com/singh1aryan/pinecone-vercel-ai
https://github.com/singh1aryan/NextJS-NestJS-GraphQL-Starter
https://github.com/singh1aryan/api-saas-project
https://github.com/singh1aryan/subscription-starter
https://github.com/singh1aryan/taxonomy

10 Full stack projects

Idea: Online Food Ordering System

- Tech stack: React, Node.js (backend), MongoDB (database), Stripe API (payment)
- Uniqueness: Integration of real-time order tracking, user reviews, and personalized recommendations.
- How to build:
  - Create a responsive frontend using React for user interactions.
  - Develop a backend using Node.js and MongoDB to manage menus, orders, and user profiles.

  - Integrate Stripe API for secure online payments.
  - Implement a real-time order tracking system using WebSockets.
- How to market:
  - Partner with local restaurants for initial listings.
  - Offer promotional discounts and referral rewards.
  - Use social media and targeted online advertising.
- Examples: UberEats, DoorDash

Idea: Task Management Application

- Tech stack: React, Express.js (backend), PostgreSQL (database), JWT for authentication
- Uniqueness: Advanced task scheduling with recurring tasks, task prioritization, and collaboration features.
- How to build:
  - Create a React frontend with user authentication.
  - Build a RESTful API using Express.js and PostgreSQL to manage tasks and user data.
  - Implement JWT-based authentication for user accounts.
  - Add features like recurring tasks, task categories, and shared task lists.
- How to market:
  - Offer a free tier with basic features.
  - Focus on productivity communities and forums.
  - Implement referral programs.
- Examples: Todoist, Trello

Idea: E-Learning Platform

- Tech stack: React, Django (backend), PostgreSQL (database), WebRTC for live video streaming
- Uniqueness: Live virtual classrooms, interactive quizzes, and personalized learning paths.
- How to build:
  - Develop a frontend with React to display courses, quizzes, and user profiles.
  - Create a backend using Django and PostgreSQL to manage course content and user data.
  - Implement WebRTC for live video streaming and virtual classrooms.
  - Incorporate a recommendation engine for personalized learning paths.
- How to market:
  - Partner with educators and institutions for course creation.
  - Offer a free trial period.
  - Leverage social media and content marketing.
- Examples: Coursera, Udemy

Idea: Fitness Tracking App

- Tech stack: React Native (for mobile app), Firebase (backend), Google Fit API (fitness data)
- Uniqueness: Integration with wearables for real-time fitness data tracking, workout plans, and social fitness challenges.
- How to build:
    - Develop a mobile app using React Native for cross-platform compatibility.
    - Use Firebase for user authentication, data storage, and real-time syncing.
    - Integrate Google Fit API for access to fitness data.
    - Implement social features like challenges and workout sharing.
- How to market:
    - Collaborate with fitness influencers for promotion.
    - Gamify fitness challenges with rewards.
    - Utilize app store optimization techniques.
- Examples: MyFitnessPal, Strava

Idea: Event Management Platform

- Tech stack: React, Ruby on Rails (backend), PostgreSQL (database), Google Maps API (location)
- Uniqueness: Event discovery, ticketing, and seamless event registration.
- How to build:
    - Create a React frontend with event listings, search, and user profiles.
    - Develop a backend using Ruby on Rails to manage events, users, and tickets.
    - Utilize Google Maps API for location-based event searching.
    - Implement secure payment processing for ticket sales.
- How to market:
    - Collaborate with event organizers to list their events.
    - Offer early bird discounts and group booking incentives.
    - Use targeted social media advertising.
- Examples: Eventbrite, Meetup

Idea: Personal Finance Dashboard

- Tech stack: React, Node.js (backend), PostgreSQL (database), Plaid API (financial data)
- Uniqueness: Comprehensive financial tracking, budgeting, and investment analysis.
- How to build:
    - Create a React dashboard for visualizing income, expenses, and investment portfolios.
    - Develop a Node.js backend to securely fetch and store financial data using the Plaid API.

    - Implement budgeting features, expense categorization, and investment performance analysis.
- How to market:
    - Emphasize data security and privacy.
    - Provide educational content on financial management.
    - Use social media advertising targeting financially conscious individuals.
- Examples: Mint, Personal Capital

Idea: Social Media Scheduler

- Tech stack: React, Django (backend), PostgreSQL (database), OAuth for social media integration
- Uniqueness: Centralized content scheduling for multiple social media platforms with analytics.
- How to build:
    - Create a React frontend for user-friendly content scheduling and analytics.
    - Develop a Django backend for user account management and social media API integrations.
    - Integrate OAuth for secure social media account linking.
    - Provide insights and analytics on post engagement.
- How to market:
    - Offer a free plan with limited features.
    - Target social media marketers, influencers, and businesses.
    - Leverage content marketing and partnerships with social media experts.
- Examples: Buffer, Hootsuite

Idea: Language Learning App

- Tech stack: React Native (for mobile app), Firebase (backend), Speech Recognition API
- Uniqueness: Real-time language pronunciation assessment and interactive lessons.
- How to build:
    - Develop a mobile app using React Native for language lessons and exercises.
    - Use Firebase for user authentication, data storage, and progress tracking.
    - Implement speech recognition to evaluate pronunciation.
    - Offer interactive lessons, quizzes, and a community forum.
- How to market:
    - Partner with language schools and teachers.
    - Provide a free trial with limited content.
    - Focus on language learning communities online.
- Examples: Duolingo, Rosetta Stone

Idea: Smart Home Control Panel

- Tech stack: React, Raspberry Pi (for IoT), MQTT (for communication)
- Uniqueness: Centralized control of smart home devices with customizable automation.
- How to build:
  - Create a React web interface for managing IoT devices and automation rules.
  - Set up Raspberry Pi as a local server for device communication.
  - Use MQTT for real-time communication between devices and the control panel.
  - Allow users to create custom automation routines.
- How to market:
  - Offer compatibility with popular smart home devices.
  - Promote energy-saving automation features.
  - Target tech-savvy homeowners and IoT enthusiasts.
- Examples: Home Assistant, OpenHAB

Idea: Online Collaborative Code Editor

- Tech stack: React, Node.js (backend), WebSocket for real-time collaboration
- Uniqueness: Real-time code collaboration with features like code sharing and debugging.
- How to build:
  - Create a React-based code editor with syntax highlighting and collaborative features.
  - Develop a Node.js backend for code storage and user management.
  - Implement WebSocket for real-time code syncing and chat.
  - Add code debugging and version control features.
- How to market:
  - Offer a free plan for open-source contributors.
  - Target developer communities and coding bootcamps.
  - Collaborate with coding tutorial creators for integration.
- Examples: CodePen, Repl.it

Complete live classes: 30dayscoding.com