Next.js Syllabus (Basic to Advanced)

Next.js Syllabus (Basic to Advanced)

Module 0: Introduction to Next.js and React.js Limitations

1. What is Next.js and why use it?

2. Differences between Next.js and Create React App (CRA)

3. Problems and limitations of React.js for production applications

4. Benefits of server-side rendering and static site generation

5. SEO challenges in React.js and how Next.js solves them

6. Improved performance and caching in Next.js

7. Real-world use cases and success stories of Next.js

8. How Next.js simplifies routing and data fetching

9. Setting up your Next.js development environment

Module 1: Getting Started with Next.js

1. Introduction to Next.js and its benefits

2. Setting up a Next.js project

3. Project structure and file organization

4. Pages and routes in Next.js

5. Creating your first Next.js page

6. Linking between pages with next/link

7. Styling in Next.js (CSS, Sass, CSS Modules, Tailwind CSS)

8. Adding global styles and custom fonts

9. Next.js vs Create React App - Key differences

Module 2: Routing and Navigation

1. Dynamic routes and route parameters
2. Nested routing and route groups
3. Catch-all routes and optional catch-all routes
4. Pre-rendering in Next.js (Static Generation vs Server-Side Rendering)
5. Client-side navigation and next/router
6. Custom 404 pages and error handling
7. Page layout patterns and persistent layouts
8. Route-based data fetching strategies
9. Middleware for route protection

Module 3: Data Fetching in Next.js

1. getStaticProps - Static Site Generation (SSG)
2. getServerSideProps - Server-Side Rendering (SSR)
3. getStaticPaths for dynamic routes
4. Incremental Static Regeneration (ISR)
5. Client-side data fetching with SWR or React Query
6. Data fetching best practices and caching strategies
7. Real-time data with WebSockets and SSE
8. Error handling and fallbacks for data fetching

Module 4: API Routes and Backend Development

1. Creating API routes in Next.js
2. Handling API requests and responses
3. Building REST APIs with Next.js
4. Connecting with databases (MongoDB, PostgreSQL, Prisma)
5. Authentication and authorization
6. Middleware in Next.js 13+

7. Implementing WebSockets for real-time data

8. Secure API design and best practices

Module 5: Advanced Next.js Features

1. Middleware for authentication and authorization

2. Image optimization with next/image

3. SEO optimization and meta tags

4. Internationalization (i18n) in Next.js

5. Using Environment Variables

6. Custom Document and App components

7. Customizing the Webpack and Babel configurations

8. Using next/head for dynamic page titles and meta tags

9. Implementing complex navigation patterns (breadcrumbs, multi-step forms)

Module 6: Performance Optimization

1. Optimizing build and deployment

2. Caching strategies for faster load times

3. Lazy loading components and images

4. Code splitting and tree shaking

5. Analyzing and reducing bundle size

6. Using next/image and next/script for improved performance

7. Implementing Service Workers and PWA features

8. Using Content Delivery Networks (CDNs) for static assets

9. Real-time performance monitoring and logging

Module 7: Next.js with TypeScript

1. Setting up Next.js with TypeScript

2. Type-safe pages, API routes, and components

3. Using TypeScript with getStaticProps and getServerSideProps

4. Handling complex types and interfaces

5. TypeScript with React hooks and context

6. Advanced TypeScript patterns in Next.js

7. Testing TypeScript components and API routes

8. Using TypeScript for Next.js middleware and plugins

Module 8: Deployment and Production

1. Deploying to Vercel, AWS, and other platforms

2. Environment setup for production

3. Configuring custom domains and SSL

4. Using GitHub Actions for CI/CD

5. Advanced build optimizations and caching strategies

6. Security best practices for production

7. Managing secrets and environment variables

8. Real-time logging and error tracking

9. Cost optimization for Next.js applications

Module 9: Mastering Next.js Plugins and Integrations

1. Using NextAuth.js for authentication

2. Integrating Stripe for payments

3. Connecting Next.js with Firebase

4. Using headless CMS like Sanity, Contentful, or Strapi

5. Implementing notifications and real-time updates

6. Building serverless functions with Next.js

7. Integrating third-party APIs and SDKs

8. Building custom Next.js plugins and modules