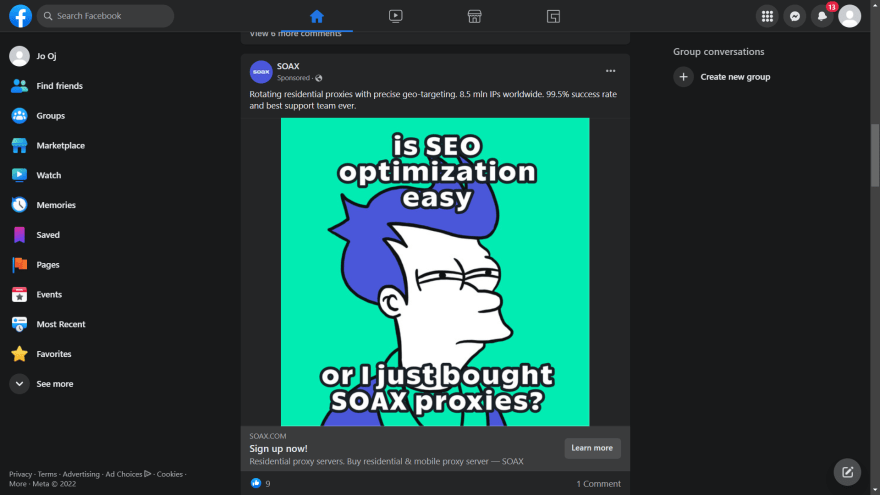
**Social Media App**

**Examples:** Instagram, Twitter, Facebook  
[](https://res.cloudinary.com/practicaldev/image/fetch/s--9W8DgziD--/c_limit,f_auto,fl_progressive,q_auto,w_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/ahrsfvc7mhq230l5f5m1.png)

Social media applications like Instagram, Facebook, Twitter are no doubt become unseperable part of our lives, we can chat, read latest breaking news, share videos, be-friend with strangers and more thanks to these applications. You can build them by yourself to improve your coding skills.

API is built with GraphQL or with REST API and data is stored in MongoDB or Postgres. Users are required to authenticate with Google or with email and authorization is implemented with JWT. You create animations with react libraries such react spring or motion or animation library of your choice. Images and videos are uploaded to Cloudinary API so you don't bug the server huge dataset

You use heroku to host your Backend code and netlify to host your React code

**Front-End:**

* React/NextJs
* Redux/Redux Toolkits
* Apollo-Client/REST API
* react spring/motion
* socket.io-client
* Optional: Unit Tests, E2E, Integration Tests with React Testing library, Cypress

**Back-End:**

* Nodejs
* Authentication with Passportjs
* Authorization with JWT
* MongoDB with Mongoose, Postgres with Prisma or MySQL
* Cloudinary API for image and video uploads
* Apollo-Server/REST API
* socket.io
* Optional: Unit Test and E2E with Jest

**Deploying:**

* Front-End on Netlify
* Back-End: on Heroku

The libraries and frameworks that you have listed are all commonly used in the MERN stack for building web applications. Here is a brief overview of how each of these libraries and frameworks could be used in a social site for school teachers and students:

* React/Next.js: React is a JavaScript library for building user interfaces, and Next.js is a framework that provides server-side rendering and other features for building React apps. Together, these libraries and frameworks could be used to build the frontend of the social site, including the user interface, interactions, and components.
* Redux/Redux Toolkit: Redux is a state management library for JavaScript applications, and Redux Toolkit is a library that provides a recommended set of packages and tools for using Redux. These libraries could be used to manage the state of the social site, such as the data for teacher and student accounts, posts, and messages.
* Apollo-Client/REST API: Apollo-Client is a library for using GraphQL in React apps, and REST API is a common architectural style for building web APIs. These libraries and architectures could be used to define and implement the API endpoints that the React frontend uses to communicate with the Node.js backend.
* react-spring/motion: react-spring is a library for animating components in React apps, and motion is a library for building components with the same API as react-spring, but with the added ability to animate other HTML elements. These libraries could be used to add animations and transitions to the user interface of the social site, to make it more visually appealing and engaging.
* socket.io-client: socket.io is a library for implementing real-time communication in web applications, and the client-side version of the library, socket.io-client, could be used in the React frontend to enable real-time communication with the Node.js backend. For example, socket.io-client could be used to implement instant notifications when a teacher creates a new post or a student sends a message or file.
* Node.js: Node.js is a JavaScript runtime environment that allows you to run JavaScript on the server-side. It could be used to build the backend of the social site, including the server-side logic and APIs that the React frontend uses to communicate with the database and other services.
* Passport.js: Passport.js is a library for implementing authentication in a Node.js backend. It could be used to manage teacher and student accounts and verify the identity of users who access the social site.
* JWT: JWT (JSON Web Tokens) is a standard for encoding and transmitting data in web applications. It could be used in combination with Passport.js to implement secure authentication and authorization in the social site.
* MongoDB/Mongoose: MongoDB is a popular NoSQL database, and Mongoose is an object



Building a SaaS (Software as a Service) website for a social media platform for schools using Node.js and React.js can be a challenging but rewarding project. Here are some steps you might consider taking to get started:

1. Define your target audience and objectives. Who will be using your platform? What specific needs will it address for students, teachers, and other school staff?
2. Research and analyze the competition. Look at other social media platforms for schools and consider what features they offer and how you can differentiate your platform.
3. Design a wireframe or prototype of your platform. This will help you visualize how the platform will look and function, and it will also be helpful for getting feedback and refining your design.
4. Set up your development environment and start building the platform using Node.js and React.js. You may want to consider using a framework such as Express.js or Next.js to help speed up development.
5. Test and debug your platform. Make sure to test all of the features and functionality thoroughly before launching.
6. Launch and promote your platform. Once you are confident in the stability and usability of your platform, it's time to make it available to your target audience and start promoting it to potential users.

Good luck with your project!