**Installing the building app components**

React Native is an open-source framework for building mobile applications using JavaScript and React. It was developed by Facebook and was first released in 2015. React Native allows developers to create mobile apps for multiple platforms, including iOS and Android, using a single codebase. Here are some key points about

**React Native:**

JavaScript and React: React Native uses JavaScript, a popular programming language, and the React library, which is used for building user interfaces. This means that developers can leverage their existing web development skills to build mobile applications.

Cross-Platform Development: One of the primary advantages of React Native is that it enables cross-platform development. You can write code once and use it for both iOS and Android, with minimal platform-specific adjustments.

Native Components: React Native provides a set of components that are native to the platform, allowing developers to create a truly native look and feel for their applications. These components can be easily integrated with the JavaScript code.

Performance: React Native aims to deliver high performance and responsiveness, thanks to its use of native components and the ability to access native APIs directly.

Hot Reloading: Developers can make changes to the code and see the results in real-time without restarting the app. This feature, known as "hot reloading," speeds up the development process.

Large Ecosystem: React Native has a large and active community, which has led to the development of many third-party libraries, tools, and plugins. This ecosystem can help you extend the functionality of your mobile app.

Cost-Efficient: Building apps with React Native can be cost-efficient since you can share a significant portion of your codebase between iOS and Android.

Popular Apps: Many popular apps, such as Facebook, Instagram, Airbnb, and Uber Eats, use React Native for their mobile a

**Installing dependencies**

The goal is to build a simple real-time chat application with sentiment analysis.

Required software includes NextJS, Pusher, Sentiment, and React.

Start by creating a new directory and installing necessary dependencies.

Use the command npm init -y to initialize a package and install app dependencies.

Follow the steps to create a Next.js app named "realtime-chat-app."

Set up environment variables by creating a Pusher application on your dashboard and adding the credentials to a .env file.

Create a next.config.js file to configure Webpack for handling environment variables.

This configuration allows the React components to access environment variables via the process.env object.

This activity sets the foundation for building a real-time chat application with sentiment analysis.

**Setting up the Server**

The goal is to set up a server using Next.js, wrap an Express application server, add necessary middlewares, and configure Pusher using environmental variables.

You need to create a server.js file in the root directory of your application.

In the server.js file, you include code for handling Express, Pusher, Sentiment, and other dependencies.

Ensure your Pusher credentials are correctly set in the .env file.

The code sets up a server, uses CORS, and listens on a specified port.

This activity sets up the foundation for integrating the real-time chat application with Pusher and sentiment analysis.

**Modifying npm scripts**

In this section, you are tasked with modifying the scripts in the package.json file to set up commands for development and production.

The modified scripts in package.json should look like the following:

json

Copy code

"scripts": {

"dev": "node server.js",

"build": "next build",

"start": "NODE\_ENV=production node server.js"

}

The "dev" script is set to run the "node server.js" command, which starts the development server.

The "build" script is used for building the Next.js application.

The "start" script is for starting the application in production mode and sets the NODE\_ENV to "production" before running "node server.js."

These scripts are essential for developing and deploying the real-time chat application.