**Documentation of MERN stack**

***Project: Society Management System***

The MERN stack is a popular stack of technologies for building a modern single-page application. In this tutorial, you will learn the MERN stack by building an exercise tracker.

The MERN stack consists of the following technologies:

**MongoDB**: A document-based open source database.

**Express**: A web application framework for Node.js.

**React**: A JavaScript front-end library for building user interfaces.

**Node.js**: JavaScript run-time environment that executes JavaScript code outside of a browser (such as a server).

**Initial set-up:**

Time to code! In this we’ll be building up the code a little bit at a time,

but you can see the boiler code plate here: <https://github.com/gautrohit/SocietyMangSys>

-Verify you have Node.js installed on your system by typing the following on the command line:

$ node -v

-If node software not installed in your system, so here’s is the path for download: <https://nodejs.org/en/download/>

-Next, we’ll create the initial React project by using create-react-app. The npx command allows us to run create-react-app without installing it first. Run this command:

This command uses to install the react very first time in the system:

$ npm create-react-app -g

This command used to create the react project:

$ npx create-react-app SocietyMangSys

Change into the newly created folder:

$ cd SocietyMangSys

Start the development web server by running the following command:

$ npm start

This starts the development server for the front end of the app. But before we work more on the front end, we’ll create the back end and connect it to MongoDB.

**Back end:**

Inside the root folder (“SocietyMangSys”), create a new folder and change into the folder by running the following commands in the terminal:

$ mkdir backend

$ cd backend

We’ll create a package.json file inside the folder by running:

$ npm init -y

Now we can install a few dependencies:

$ npm install express cors mongoose

So, what are these packages?

**CORS:**

Cross-origin resource sharing (CORS) allows AJAX requests to skip the Same-origin policy and access resources from remote hosts. The cors package provides an Express middleware that can enable CORS with different options.

**MongoDB:**

And we already discussed mongoose. It makes interacting with MongoDB through Node.js simpler.