URL Shortener

In this mini project you will create your own url shortener. Users should be able to input a link and an *alias*. Then, when someone visits YOUR-APP.COM/*alias*, they should be redirected to the input link. You will need to establish communication between backend and frontend in order to store these aliases and URLs in a database.

Task 1. Initialize a new Next.js project
1. Follow the process described in discussion or lecture slides. You may use Tailwind CSS but are not required to.

Task 2. Setup MongoDB
1. Follow the process described in discussion or lecture slides. You should have an **.env.local** file with something similar to the following.

   ```
   MONGO_URI="your-connection-string-here"
   ```
2. Make sure you enable network access from any IP address so that your deployed version can access the database.
3. Install the mongodb package using the command **npm install mongodb**

Task 3. Figure it out!
1. Here is an example of what your finished project may look like: https://cs391-url-shortener.vercel.app/. However, yours may look and function slightly differently. At the end of this assignment you will find the functionality requirements you must meet.
2. You may want to look at the "next/navigation" package to figure out how to redirect users to the appropriate website. Client-side redirection can be achieved using the **useRouter()** hook and server-side redirection can be achieved using the **redirect()** function. Both of these are acceptable methods; the demo uses the server-side **redirect()** function.

Functionality Requirements:
1. Your database secrets must not be exposed to the client or on GitHub
   a. When deploying, ensure you upload the relevant environment variables (e.g. mongo uri)
2. A user must not be able to submit an invalid URL. URL validation must occur on the backend but may also occur on the frontend (this may be somewhat challenging to figure out). There should be an error message displayed to the user if the alias they chose is already taken.
3. Repeated aliases must not be allowed. There should be an error message displayed to the user if the alias they chose is already taken.
4. Users must be redirected to the URL associated with a given alias when visiting YOUR-APP.COM/*alias*. They should not need to click anything.

       a. You may also use something like [YOUR-APP.com/r/*alias*](YOUR-APP.com/r/alias) if it makes it easier to account for conflicting namespaces

5. Users must be able to copy and share their shortened URLs. In other words, the shortened URL must be displayed somewhere on the page after it has been successfully created

Additional Requirements:
- In addition to these functionality requirements, your project must also adhere to programming best practices (e.g. DRY, separation of concerns) and be styled appropriately.
- The **only** external packages (other than **mongodb**) you may use are those used for UI/styling, such as MUI and styled-components.
- You may **NOT** use the Next.js pages router; you should not have a directory called "**pages**" or use a **getServerSideProps()** function.

Please check that the deployed/submitted version of your project works as intended. Make sure that you upload your Mongo URI to Vercel so that the deployed version of your application can access your database.