User Management

So for the user management system we will need to be able to view the following things

* Sign up information
  + Recent Signups
  + All sign ups
  + Location sign ups
* User information
  + Edit user info
  + Edit profile info
  + Delete user
  + View user info
  + View user profile
  + Reset password
  + Change email
  + Create a new user
* View all logs
  + Graphs, carts of log information
  + All written logs are accessible from the user management panel

So the user management panel will need access to the sql server which stores the user information and it will need to have the ability to edit everything in the user account.

All of the pages within the user management system must be only accessible by the authorized users. These accounts will have to be created by the web admins within the database.

The webpage should also not be really visible to the outside world, you will need to hide it in a way so that you need to look for it

The front end of the user management system will be coded in HTML bootstrap then we will code the application layer with c#

The log information should be vizulationed in graphs and charts so that it is easy to read them

All actions taken by the user in the admin panel should be logged so that the other admin can see what changes had been made to the admin parts of the site to verify that there is no malicious activity going on

Logging

There is a framework built for c# to handle logging, not sure if we can use it though it's called nlog

Throughout the application all the logs should have a consistent pattern so setting this pattern is really important

Logs should be separated into levels

* Non Error Events(used in development but not in production)
  + Debug
  + Info
  + Trace
* Production
  + Warn
  + Error
  + Fatal

By separating the logs you have a better way to find logs that actually pertain to critical services

For technical application you usually put the logg at the end of a try catch, so that you can confirm that an error did occurs that you were planning on and log that information

Try{

}catch{

log.create();

}

Once you have all of the log logging on error you need to figure out how to view them, we will be using the user management system I assume to visualize all of the logs

We can use this logging 3rd party maybe since it is built into c#: <https://docs.microsoft.com/en-us/aspnet/core/fundamentals/logging/?view=aspnetcore-5.0>

For each log this system add an entry into a json file, and you can create the parameters for the adding and add whatever data you want to.

To use this logging system we will have to first create the logger object model. We define the ILogger \_logger then this creates the \_logger object. When you want to log something in the system you will have to then do \_logger.logInformation(“What you want to log”).

With logging the system we are using to do the loggin for us will keep track of the pertinent information like ip it came from and time it happened so we only have to focus on actually logging the necessary messages

We need to log out development logs which we will have to take out during production so we have, so we should have all of these with the debug condition, so they on run when we are debugging