**Code Style Document**

**https://chakra-ui.com/docs/components/overview**

**https://chakra-ui.com/docs/styled-system/features/style-props?scroll=true**

You can pass any css attributes to a component instead of writing it in a css file.

Try to get familiar with Chakra UI and React hooks, useState and useEffect are the most common.

All code should be uniform to make it more readable and familiar for developers working in it. This decreases the chances of mistakes when working on something someone else has written.

All names should be descriptive and clear, with the exception of things like the i in for loops . camelcase should be used for variables. PascalCase should be used for classes. All interfaces should follow the convention to use the Props suffix, i.e. for the class Object, the interface will be named objectProps

The convention is to use the Props suffix. For Widget it would be WidgetProps.

```
interface boatProps {
length: number
}
const Boat = (props: boatProps) => {
}
```

Use const function_name = () => {} instead of function function_name(){}

If using Chakra ui all css numbers like width, height, padding should have curly braces and no quotes all strings should have quotes and no curly braces.

ex) padding={8} and backgroundColor=”white”.

Use indentation properly to make it readable. Good help is prettier and google/youtube format on save.

Variable that are either constant or should not be changed by anyone should follow the CONSTANT_CASE convention. I.e. const METER_UNIT = { ‘meter’: ‘m’} should never change it’s value.

Use const and let, never use var.

All http calls must be checked for errors.

All images, documents and such in the res folder.

Use types at all times when possible.

If no task exist for what you need to do to complete a userstory, create one so that your progress can be tracked

Create a branch for each task

Name the branch descriptively, using under_score

Comment the changes that have been made to each commit

Try to use js array functions like map() instead of for and while loops.
Only data structure allowed is a BST.
Try using DOM as much as possible instead of React to prove yourself.

Try keeping the components to a minimal size and split them up if a component is getting too large.

**GIT**

In the terminal, type the following commands inside your project folder (/janet-jackson):

```
git checkout main                          , byter branch till main
git pull                                   , hämta de senaste uppdateringarna
git checkout -b namnet_på_din_branch      (((((using _ for instead of space))))))), byt branch
till din
//JOBBA
git add .                                  , lägger till alla filer
git commit -m “meddelande”                 ,  meddelande
git checkout main                          , byt branch till main
git pull                                   , hämta senaste uppdateringarna
git checkout namnet_på_din_branch          , byt branch till din
git rebase main                            , rebasa main
git push                                   , pusha dina ändringar
```

gå in på git välj någon att göra en review till höger och klicka gröna knappen create pull request.

Personen som reviewat trycker uppe på en grön knapp review och skriv en kommentar sedan mergea eller skicka tbx feedback sedan ifall ni mergar tryck sedan delete bransch ifall ni har mergat in de.

Efter personen har tagit bort din branch på git kan du deleta den lokalt med:

```
git branch -D namnet_på_din_branch
```

Write comments explaining what the functions do.