Assignment : 1 -/>

Create ui following .

UI will have two parts left part will have div

The right part will have a button (Change Pic) .

on clicking the change pic button , the image should change .



Assignment : 2 -/>

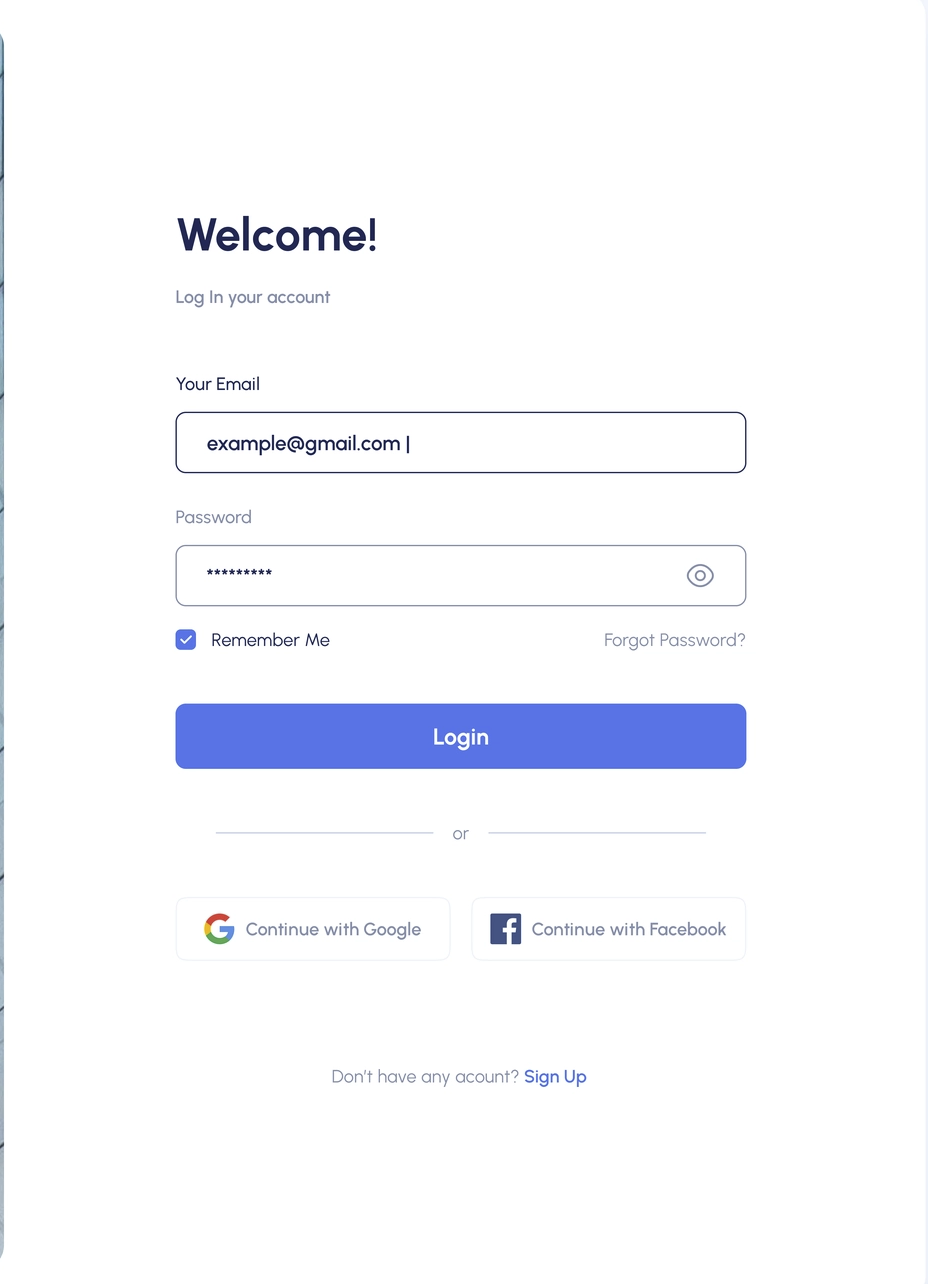
create a text area with a character limit of 100 .

Assignment : 3 -/>

create from as follows .

for signUp / register , we will be showing 4 fields -/> (firstName , lastName ,email and password)

for signIn / login -/> (email and password fields only)



Instructions :

1. Keep the js and css file separate for each component .

2. Follow the design provided .

3. If design is not proper then only you guys can create design as per your thinking.

4. Don’t write all code in a single file and make components .

5. Create button components separately , pass css and data in that to design it .

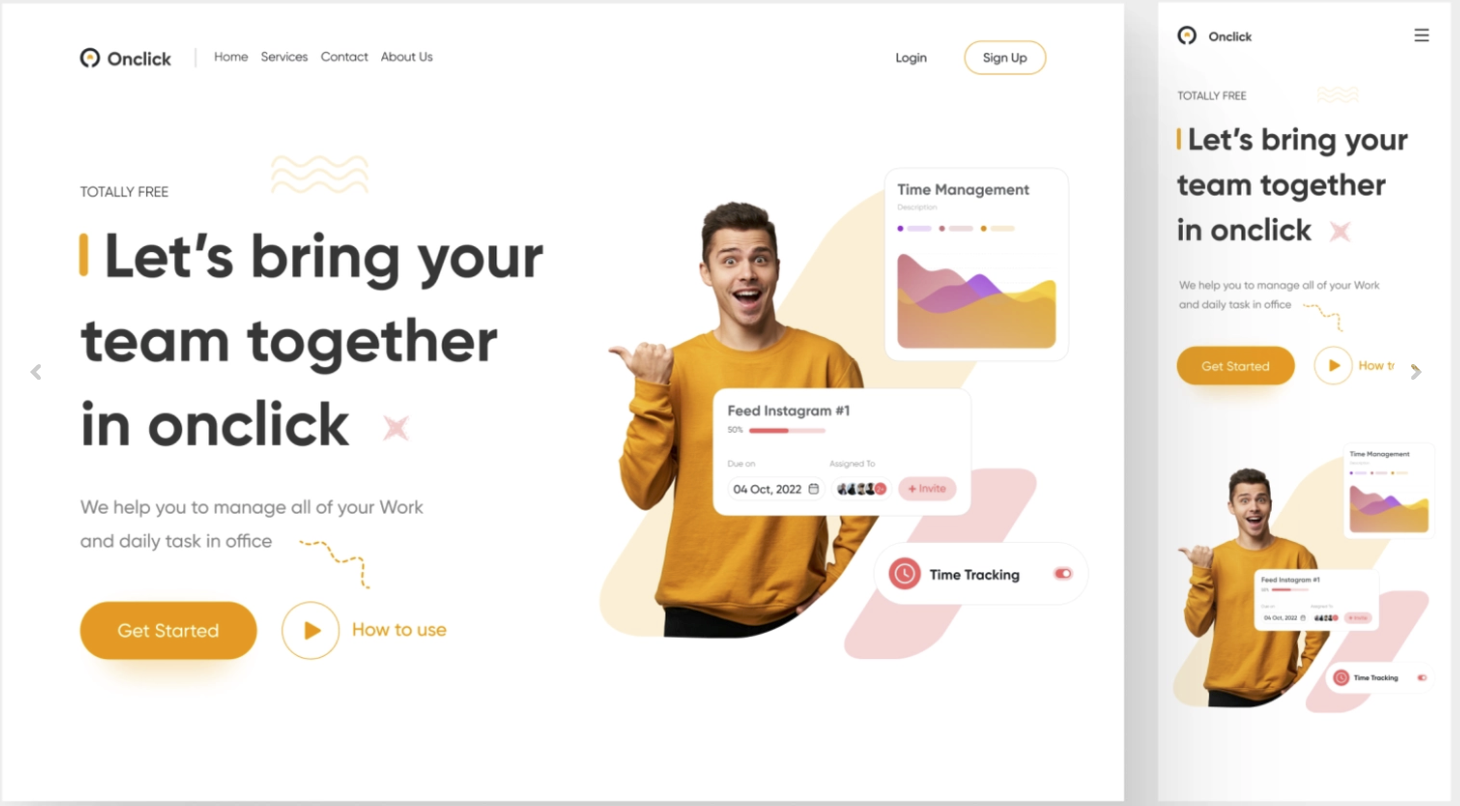
6. Ignore icons for now . If you can explore on your own you guys can use react-icons package for icons or we will discuss this in the next session .

7. https://www.npmjs.com/package/react-icons (react icons)

**Media Query and css assignment - 20th March**

1. Use media query
2. Use module level css

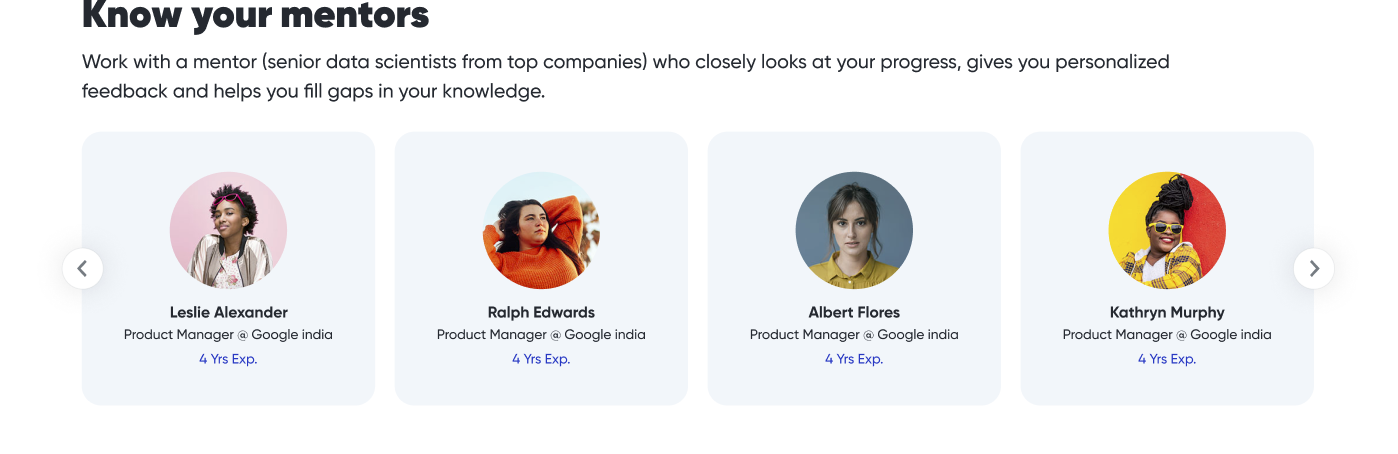
Figma link : <https://dribbble.com/shots/17349283-Onclick-Task-Management>



1. Create this design as shown in figma .
2. No need to create this design as it is as we have not gone so deep till now .
3. Make sure the header will be as it is as shown in figma in desktop and mobile both the view .
4. For header , In mobile view (menu) icon will be visible in header , on click that icon we will show our header options (home,services,contact , aboutUs and login , signUp … etc)
5. For section below the header no need to create as it is , just make sure left side will have some text and button and right side will have any image and manage this section in mobile view as shown in figma

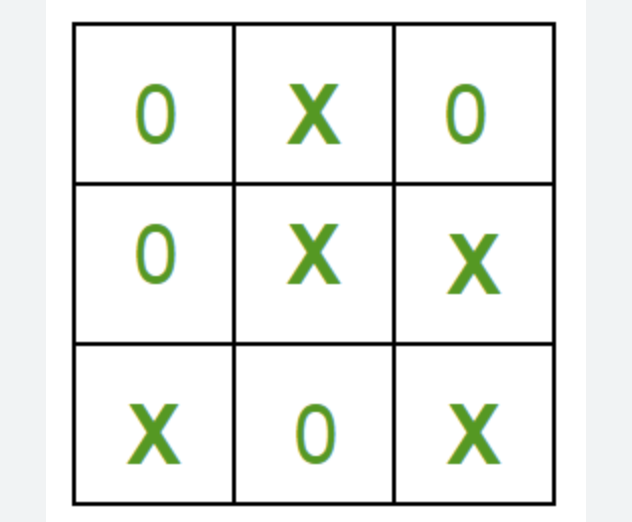
**Assignment - 21st March 2023**

Horizontal scroller



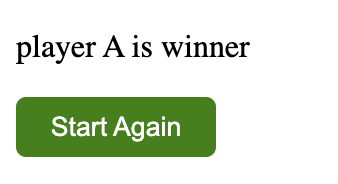
1. Ignore arrows for now
2. Create components as shown above .

Create tic-tac-toe- game

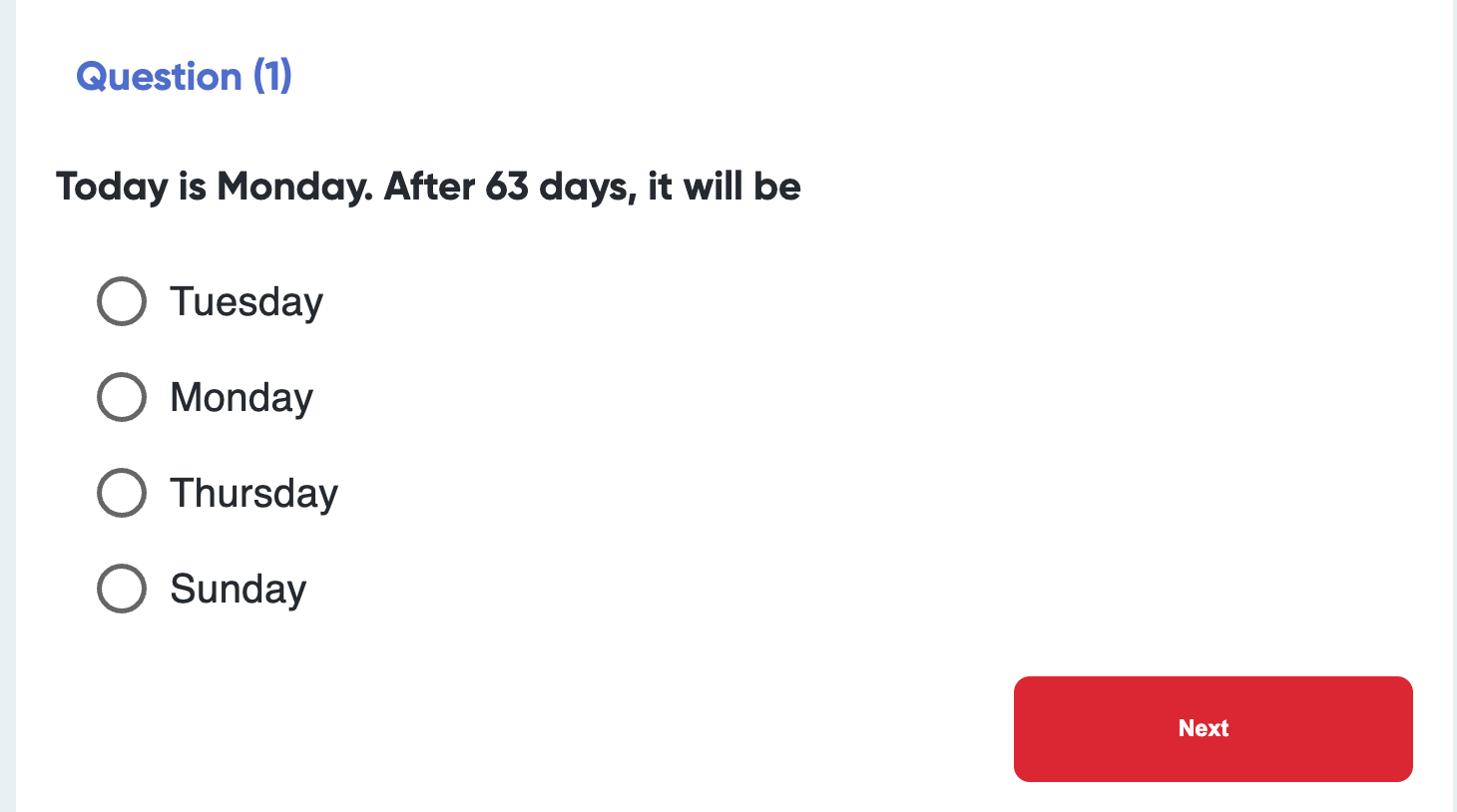


There will be two player .

If any winner is won show the winner name on screen and start button to start it again



On click start again restart the game



1. Create the test of 10 questions
2. Each question has 4 option
3. Save the selected option using useRef .
4. On click next we will show the next question .
5. When there is a 10th question button text will change from ‘next’ -/> ‘Submit’ .
6. On submitting last question we will , we will display the result
7. Each question will have 2 marks and passing marks will be 12 out of 20 .
8. On passing the test show greeting .
9. On failing the test show button to restart the test .

Const questionsData = [

{

Question : ‘’,

Options : [1,2,3,4],

Correct answer : ‘’

}

]

Keep data format as above

3) Create guess lucky number game .

Follow the ui .



If the number is less than the lucky number , alert -/> you guessed a smaller number .

If the number is bigger than the lucky number , alert -/> you guessed a bigger number .

If guessed right -/> show message ’congratulations you guessed the right number in <count> attempts ’

4) Create a navbar with four options (home, about, login, register) .

Add routing on that ,

On clicking the options it will redirect us to the respective page .

Every page will contain hedging like ‘Welcome to home page’’

**RECOIL**

1. **Change color button**

****

On clicking the button color will change from blue to red and red to blue .

1. Create 2 page (home and tasks page)
2. Home page will have this ui (fig-1)

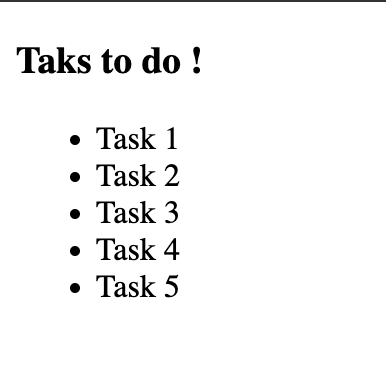


From here you can add the tasks

Save this tasks in recoil atom named ‘taskListAtom’ ,

Check task button will take us to the tasks page .

1. Tasks page will have this ui



Where you will see all the added tasks , adding tasks at the home page should get updated in this list using recoil .