

You must create a gray paper that explains the HTML, CSS, and JS that you did. It should outline at least 30 different points, showing the code and explaining what it does.

Home Page - Sukhnandan Kaur


1. For the “Home” page, when I was implementing transitions to the second section of the page which is the section that contains an image of a lady and a list giving a breakdown of how our tool works, the content of that section would disappear. I was using the Animation on Scroll library and realized I had only linked the CSS link to the library in the head tag. To fix this issue, I implemented the JavaScript source link and initialized the library.

```
<!-- How Our Application Works Section -->
<div class="how_container">
  <div data-aos="fade-right" data-aos-duration="1000" class="how_moto_content">
    <h2>Manage complex tasks easily</h2>
    <p>Thriving for efficiency</p>
    <a href="index.html"></a>
  </div>
  <div data-aos="fade-left" data-aos-duration="1000" class="how_list">
    <ol>
      <li>
        <span>01 <br />Create a to-do list </span>
        <p>Get started by simply clicking on "add a new task" on the very right hand side and start adding all the tasks you have to complete. </p>
      </li>
      <li>
        <span>02 <br />View your tasks</span>
        <p>After adding your tasks, they will appear on the "view of upcoming task" window or you can click on "quick view of tasks" to view them. </p>
      </li>
      <li>
        <span>03 <br />Mark them done</span>
        <p>Once completing a task or tasks you can click on the task in the "view of upcoming task" window to remove it. View your past tasks in " </p>
      </li>
    </ol>
  </div>
</div>
<script src="https://unpkg.com/aos@2.3.1/dist/aos.js"></script>
<script>
  AOS.init();
</script>


<script src="https://unpkg.com/aos@2.3.1/dist/aos.js"></script>
<script>
  AOS.init();
</script>
```

2. Another one of the issues I ran into was when I was trying to incorporate the incrementing counter from the Udemy course in the home page. The CSS for the counter would not reflect or update to the home page right away even though I was using a live server. I tried playing around with the position and directions to see if my counter can be displayed the way it was shown in the course video, however nothing was working. After I researched and tried everything I could, I decided to restart VSCode and the CSS for the counter ended up working and was positioned how I had wished for.

Helping people like you get the job done


302678
Tasks Completed


250
Partnered Companies


4329
Satisfied Customers

```
.counter_section {
  display: flex;
  align-items: center;
  justify-content: center;
  height: 70vh;
  overflow: hidden;
  margin: 0;
}

.counter_container {
  display: flex;
  flex-direction: column;
  justify-content: center;
  text-align: center;
  margin: 10px 120px;
}

.counter_container .counter {
  font-size: 55px;
  margin-top: 10px;
  color: #F24726;
}

.counter_container span {
  font-size: 20px;
  margin-top: 15px;
  color: #F36C51;
}
```

3. After completing the home page an issue that was brought up was that the position of the logo in the footer kept on changing when one of my teammates opened the page on their computer. The issue may have been caused since the CSS of the logo in the footer did not reflect the same on different screen sizes. To fix this To make sure the content of the footer was centered, especially the position of the logo I set the “width:100%” and “display:flex.”

```
.footer {  
  margin-top: auto;  
  width:100%;  
  height:8px;  
  border-bottom:0.2px solid #rgb(220,220,220);  
}
```

```
.footer_logo {  
  display: flex;  
  margin-left:20px;  
  align-items: center;  
  height:50px;  
}
```

4. Another issue I ran into was when positioning the main content of the home page, a scroll on the bottom of the page would be created. Essentially, for the user to view the full content of the page they would have to scroll the page horizontally (to the right). This issue was occurring because the width of the content was exceeding the max-width of the page. Since this is not user-friendly, I changed the “max-width:1720px;” and created a “display:grid” to fix it.

F O C U S

HOME ABOUT US FAQ LOG IN GET STARTED

Manange Your Tasks Efficently.

Plan, organize, and never miss a deadline with our powerful task management tool.

GET STARTED

VIEW DEMO

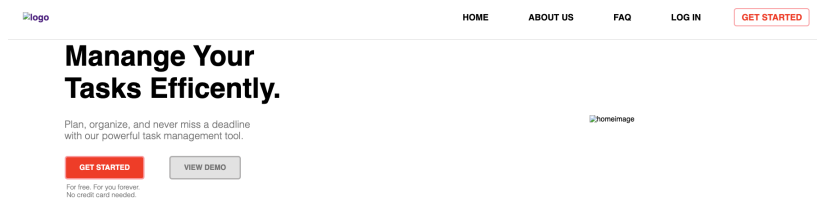
For free. For you forever.
No credit card needed.



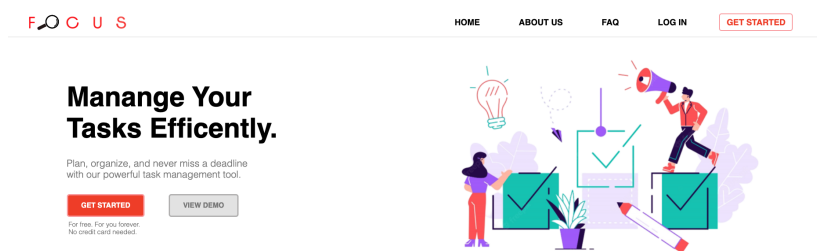
```
/* Hero Section */  
.hero_container {  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
  align-items: center;  
  justify-self: center;  
  margin: 0 auto;  
  margin-bottom: 20vh;  
  z-index: 1;  
  width: 100%;  
  max-width: 1720px;  
  padding-right: 50px;  
  padding-left: 50px;  
}  
.hero_content {  
  width: 100%;  
}  
.hero_content h1 {  
  font-size: 62px;  
}  
.hero_content h2 {  
  font-size: 22px;  
  font-weight: lighter;  
  color: #gray;  
  margin-top: 2rem;  
}
```

5. There were times when I tried to open up the home page in my default browser the image used in the page would not appear. Therefore, to fix this I opened the images from my folder in Chrome and copied the path of the image to the image source. However, since this is not good when it comes to expanding our website, I updated Chrome and changed the path for the images.

Before



After:



6. Something cool that I implemented in the homepage is when the user hovers over the “get started” and “view demo” under the heading it changes to a gradient color. To do this I include instructions in the CSS for what happens when the user hovers off or on the buttons and uses “linear-gradient” to create the effect.

```
/* Start Button */
.startbtn{
  background-color: #F24726;
  padding: 14px 38px;
  border: 3px solid #fba5a8;
  border-radius: 6px;
  margin-top: 2rem;
  position: relative;
  transition: all 0.35s;
}

.startbtn a{
  position: relative;
  z-index: 2;
  color: #ffff;
  font-weight: bold;
  font-size: 16px;
  text-decoration: none;
}

.startbtn:after {
  position: absolute;
  content: '';
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background-image: linear-gradient(to top, #d669e7 5%, #fba5a8 100%);
  transition: all 0.35s;
}

.startbtn a: hover{
  color: rgb(252, 255, 254);
}

.startbtn: hover: after {
  width: 100%;
}

/* Demo Button */
.demobtn{
  background-color: #E6E6E6;
  border: 3px solid #bab9b9;
  padding: 14px 38px;
  margin-top: 2rem;
  position: relative;
  left: 58px;
  transition: all 0.35s;
}

.demobtn a{
  position: relative;
  z-index: 2;
  color: #808080;
  font-weight: bold;
  font-size: 16px;
  text-decoration: none;
}

.demobtn:after {
  position: absolute;
  content: '';
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background-image: linear-gradient(to top, #a7a4be 0%, #141918 120%);
  transition: all 0.35s;
}

.demobtn a: hover{
  color: white;
}

.demobtn: hover: after {
  width: 100%;
}
```

About Us Page -

7. One bug I faced is with the flipping cards not actually flipping over. When I hovered over a card the front would flip but then the spacing below would flip over. This I fixed by setting the back card position to absolute so that the flip would happen on the same location.

```
.card-back {  
  text-align: center;  
  transform: rotateY(180deg);  
  position: absolute;  
  top: 0; right: 0; bottom: 0; left: 0;  
}
```

8. Another issue I found was the back card not spanning the same size as the front card when the page size is changed. Meaning it wasn't responsive like the front side. This was fixed by having the backcard width and height to be 100% instead of hardcoded the pixel size.

```
.card-flip>div {  
  border-radius: 10px;  
  backface-visibility: hidden;  
  transition: transform 300ms;  
  transition-timing-function: linear;  
  width: 100%;  
  height: 100%;  
}
```

9. The getting started page has the overview of the team's mission and then a card for each individual member with a short blurb about us. With a button leading the user to the linkedin page. I used bootstrap to create the container for the rows and columns for the card layout

```
<section>  
  <div class="container">  
    <div class="row">  
      <div class="col">  
        <div class="card-flip">  
          <div class="card-front raj">  
            <h3 class="Names">Rajveer</h3>  
              
          </div>  
          <div class="card-back raj-back">  
            <p class="message">  
              Hi my name is Raj. I am currently a Computer Science major in my Senior year at Queens  
              College hoping to one day be a Software Engineer. I also like working out and playing video  
              games in my free time. <br>  
            </p>  
            <a href="https://www.linkedin.com/in/rajveer-singh-1b27a21bb/" class="button">  
              <button class="ripple">Linked In</button>  
            </a>  
          </div>  
        </div>  
      </div>  
    </div>  
  </div>
```

10. I used a button animation from the Udemy to create a pulse effect when clicking a button on the back of the card. Clicking the button will do an animation and then take the user to the github page of the member of the team.

```
const buttons = document.querySelectorAll('.ripple')
buttons.forEach(button => {
  button.addEventListener('click', function(e){
    const x = e.clientX
    const y = e.clientY
    console.log(x,y)

    const buttonTop = e.target.offsetTop
    const buttonLeft = e.target.offsetLeft

    const xInside = x - buttonLeft
    const yInside = y - buttonTop

    const circle = document.createElement('span')
    circle.classList.add('circle')
    circle.style.top = yInside + 'px'
    circle.style.left = xInside + 'px'

    this.appendChild(circle)

    setTimeout(() => circle.remove(), 500)
  })
})
```

11. Another issue I had was implementing JavaScript from the Udemy. The button animation was not appearing when clicking and this was because of the way code worked. It took the position of the clicked location and subtracted it from the location of the button to display an animation. However this would not always lead to a small number if it was a singular button. So to combat this I made the effect of the button much stronger so that no matter what the animation will always come through.

```
@keyframes scale {
  to {
    transform: translate(-50%, -50%) scale(50);
    opacity: 0;
  }
}
```

12. Another error I encountered is that the text would leak outside of the card when the page is zoomed in too much. Meaning it was not dynamic. However using vh would not fit and I wanted the sizing of the text to be dynamic with the page. To counter this I set the px size to the desired size and had overflow to auto. So if the user zooms in too much it would create a scrollbar to view the full paragraph behind the card.

```
div .message {
  height: 70%;
  margin: 10% 10% 0% 10%;
  font-size: 20px;
  overflow: auto;
}
```

FAQ Page

13. When creating the cells for each FAQ item, the initial plan was a div to contain the entire FAQ items. Then, a div for the question and answer and one for each question and answer separately as well as for the button. Through this, my code was getting a bit messy but then I was able to realize that tags could have classes; I didn't need to overuse divs which made the CSS a lot easier to implement and understand. This clean structure of the FAQ items as well as the collapse icon for the button "fas fa-chevron-down" was thanks to one of the Udemy 50 Projects Day 12 FAQ Collapse.

```
<div class="container">

  <div class="question-container">
    <h3 class="question-text">
      What is Focus?
    </h3>
    <br>
    <p class="answer-text">
      Focus is a life style brand whose "focus" is to keep anyone from students to the average working person organized.
      Everyone has things they need to get done but not everyone is organized in doing so. Focus offers a medium for us to take control of your day.
    </p>
    <br>
    <button class="faq-toggle">
      <i class="fas fa-chevron-down"></i>
      <i class="fas fa-times"></i>
    </button>
  </div>
```

14. Another one of the issues I came across was actually resizing each of the input boxes so that they end up being the same size. The solution was selecting `input[type=type]` in the CSS whereas I was trying the id with not much success. From this little thing, I was able to align and make sure all of the input fields were in a uniform format.

```
130 input[type=text],
131 select,
132 textarea {
133   width: 100%;
134   padding: 12px;
135   border: 1px solid #70, 68, 68;
136   border-radius: 4px;
137   resize: vertical;
138 }
139
140 input[type=email],
141 select,
142 textarea {
143   width: 100%;
144   padding: 12px;
145   border: 1px solid #70, 68, 68;
146   border-radius: 4px;
147   resize: vertical;
148 }
149
150 input[type=submit] {
151   background-color: #f44336;
152   color: white;
153   padding: 12px 20px;
154   border: none;
155   border-radius: 4px;
156   cursor: pointer;
157   float: right;
158 }
```

Full Name

Email

Country

Question

Submit

15. One of the biggest challenges I came across was getting both of these forms to look uniform next to each other. The first thing I tried was using float:left which honestly did some resizing that I wasn't able to change up and overall it didn't work. The next thing I tried was using flexbox but I quickly saw that flexbox didn't give the desired outcome I was looking for. Finally, I used a grid-layout along with the grid-template-columns and was able to perfectly put these forms next to each other with the correct gap.

```

main-container {
  display: grid;
  grid-template-columns: 1fr 1fr;
  grid-gap: 20px;
  width: auto;
  padding-left: 7%;
}

right-container {
  border-radius: 5px;
  padding: 10px;
  width: 85%;
}

left-container {
  border-radius: 5px;
  padding: 20px;
  width: 85%;
}

```

Questions

Full Name

Your name.

Email

Your Email.

Country

Select Country

Question

Write something.

Submit

Feedback

Full Name

Your name.

Email

Your Email.

Country

Select Country

Feedback

Write something.

Submit

16. Getting the FAQ item set up for the button to display and clear the answer was something I didn't know how to do. From W3schools I was able to learn about the .active selector. The CSS below essentially hides the X when the button was not clicked as well as hides the downwards arrow when the button is clicked. Furthermore, when the button is clicked it gets a gray background color. As well as the shadow appears to give it some emphasis that it is its own card.

What is Focus?

Focus is a life style brand whose "focus" is to keep anyone from students to the average working person organized. Everyone has things they need to get done but not everyone is organized in doing so. Focus offers a medium for us to take control of your day.

Does Focus Have Plans To Change The Free Tier?

```

.question-container.active {
  background-color: #fff;
  box-shadow: 0 3px 6px rgba(0, 0, 0, 0.1), 0 3px 6px rgba(0, 0, 0, 0.1);
}

.question-container.active::before {
  color: #3498db;
  top: -10px;
  left: -30px;
  transform: rotateY(180deg);
}

109 .question-container.active .faq-toggle .fa-times {
110   color: #fff;
111   display: block;
112 }
113
114 .question-container.active .faq-toggle .fa-chevron-down {
115   display: none;
116 }
117
118 .question-container.active .faq-toggle {
119   background-color: #9fa4a8;
120 }

```


17. The javascript for actually doing the action of toggling the buttons was something that I learned from the 50 Projects in 50 Days course 12, FAQ Collapse. The idea of this js code is to find all tags with the class of .faq-toggle. Then for every one of these tags aka buttons, we have a function that will activate the active class from the classList of each of these buttons. When the active class is toggled, then what we see in the CSS from the previous point is now activated.

```
JS faq_js.js > ...
1  const toggle_buttons = document.querySelectorAll('.faq-toggle')
2
3  toggle_buttons.forEach(toggle => {
4    toggle_buttons.addEventListener('click', () => {
5      toggle_buttons.parentNode.classList.toggle('active')
6    })
7  })
```

18. While I was able to fix the uniformity in size of each of these input cells, getting them all to line up was a challenge in itself. Using the :after tag to edit it after the fact, each input is in a div with a class of "row" that contains the label name and the input box. By setting the display to the table it gave me the structure that I was really looking for.

```
108 <div class="main-container">
109
110   <div class="left-form">
111     <h2 style="padding-left: 2%; font-size: 32px;">Questions</h2>
112     <div class="left-container">
113       <form>
114         <div class="row">
115           <div class="name">
116             <label for="fname">Full Name</label>
117           </div>
118           <div class="field">
119             <input type="text" id="fname" name="firstname" placeholder="Your name..">
120           </div>
121         </div>
```

```
182 .row:after {
183   content: "";
184   display: table;
185   clear: both;
186 }
```

Full Name

Email

Country

Question

Login Page

19. When styling everything some parts of the Login Page were changing even though i didn't want them changed, such as the text color was changing to a color i didn't want. To fix this I created divs for each main section of the page so everything can be more organized and so I can easily style different parts of the page.

```
<body>
  <nav class="nav_bar" id="navbar">
    <div class="nav_container">
      <a href="index.html"></a>
      <ul class="nav_list">
        </ul>
      </div>
    </nav>

    <div class="container">
      <h1>Sign In</h1>

      <form action="index.html">
        <div class="form-control">
          <input type="text" name="email/user" required>
          <label for="email/user">Username/Email</label>
        </div>

        <div class="form-control">
          <input type="password" name="pass" required>
          <label for="pass">Password</label>
        </div>
        <script src="js/wave.js"></script>

        <div class="bottom">
          <span class="psw"> <a href="#">Forgot password?</a></span>

          <span class="checkbox">
            <input type="checkbox" id="checked" name="checked" value="remember" checked="checked">
            <label for="checked">Remember me</label><br>
          </span>
        </div>

        <button type="submit">Login</button>
      </form>
    </div>
```

20. A problem I faced while working on the Login Page, was positioning the input type for both the email and password. It only positioned itself to the right of the text, however setting the display as a grid allowed the input to go below the label.

```
input[type=text], input[type=password] {
  padding: 12px 20px;
  border: 1px solid #ccc;
  box-sizing: border-box;
  align-self: flex-start;
  display: grid;
  align-items: center;
  justify-items: center;
}
```

Username/Email

Password

21. Centering the whole thing was a problem because if i used the text align to center the label and input and everything else the starting point would be at the center so it would look way off, instead i centered most of my page based on the margin, I set mostly everything with a margin that made it looked centered..

```
h1{
  font-size: 40px;
  text-align: center;
  color: #EC4427;
}
```

```
button {
  background-color: #EC4427;
  color: white;
  padding: 15px 20px;
  margin-left: 32%;
  margin-top: 2%;
  border: none;
  cursor: pointer;
  width: 36%;
  border-radius: 10px;
}
```

```
.form-control {
  position: relative;
  margin: 20px 0 40px;
}
```

22. There was a problem with the Remember Me part of the Login Page,if I changed the size of the window page the remember me would move and end up misaligned. To fix this, a container was created which would go around everything except the logo picture. Everything would then be set to the margin distance from the left edge of the container.

```
.container {
  margin: 25px auto;
  padding: 20px 0px;
  width: 50%;
  box-shadow: 10px 10px 8px 10px #EC4427;
  box-shadow: 0 10px 20px rgba(0,0,0,0.19), 0 6px 6px rgba(0,0,0,0.23);
  padding: 20px 40px;
}
```

```
.checkbox {
  float: right;
  margin-right: 5%;
}
```

23. When using Udemy, I implemented a wave animation to change the Username and Password up. The js portion was easy to add, however it greatly depended on the placement of the works. This was easily fixed by making the word placement within the line of the answer (since in forms it is close to each other).

```
.form-control input {
  background-color: transparent;
  border: 0;
  border-bottom: 2px solid #000;
  display: block;
  width: 100%;
  padding: 15px 0;
  font-size: 18px;
}
```

```
input[type=text], input[type=password] {
  padding: 12px 20px;
  border: 1px solid #ccc;
  box-sizing: border-box;
  align-self: flex-start;
  display: grid;
  align-items: center;
  justify-items: center;
}
```

24. The container that holds the Login information blended with the background which did not allow you to see the container. I implemented a box shadow which fixed this problem and also made the site more visually appealing.

```
.container {
  margin: 25px auto;
  padding: 20px 0px;
  width: 50%;
  box-shadow: 0 10px 20px rgba(0,0,0,0.19), 0 6px 6px rgba(0,0,0,0.23);
  padding: 20px 40px;
}
```

Get Started Page

25. For the “Get Started” page, one thing I implemented was the buttons to which wires everything together. I originally had issues making them place nicely on the side, but this was easily fixed using widths with percentages using CSS.

```
<div id="buttons">
  <table>
    <tr>
      <td><button type="button" id="todo" onclick=modal(id)>Quick View of Tasks</button></td>
      <td><button type="button" id="allTask" onclick=modal(id)>View all Tasks</button></td>
      <td><button type="button" id="newTask" onclick=modal(id)>Add New Task</button></td>
    </tr>
  </table>
</div>
```

```
table {
  float: right;
  width: 50%;
  margin-right: 5%;
}

button {
  width: 100%;
  border: solid 1px red;
  border-radius: 10px;
  color: red;
  padding: 15px;
  background-color: transparent;
}

tr td {
  padding: 0px 10% 5% 0px;
}
```

26. Within each button I implemented my own modal that shows a form of display box, this allows us to add list elements and look at each list element. I had issues setting up the close button for the other buttons which was easily fixed by duplicating and making many variables per each button.

```
<div id="todoModal" class="modal">
  <div class="modal-content">
    <span class="todoClose">&times;</span>
    <h2><strong>To Do List</strong></h2>
    <ul class="todoList" id="todoList">
    </ul>
  </div>
</div>

<div id="allTaskModal" class="modal">
  <div class="modal-content">
    <span class="allTaskClose">&times;</span>
    <h2><strong>View all Tasks</strong></h2>
    <ul class="taskList" id="taskList" reversed></ul>
  </div>
</div>

<div id="newTaskModal" class="modal">
  <div class="modal-content">
    <span class="newTaskClose">&times;</span>
    <h2><strong>Add a New Task</strong></h2>
    <form id="form">
      <input type="text" class="input" id="add" placeholder="Enter your To-Do" autocomplete="off">
    </form>
  </div>
</div>
```

```
.todoClose,
.newTaskClose,
.allTaskClose{
  color: #aaa;
  float: right;
  font-size: 28px;
  font-weight: bold;
}

.todoClose:hover,
.newTaskClose:hover,
.allTaskClose:hover,
.todoClose:focus,
.newTaskClose:focus,
.allTaskClose:focus{
  color: black;
  text-decoration: none;
  cursor: pointer;
}
```

27. I used Udeemy to make the To-Do list work for my page, however the Udeemy only allows for printing of one list; which was easily fixed by adding many variables to take in the value entered in the “Add Task” button and then appending it to each list.

```
if(todoText){
  const todoListEl = document.createElement('li')
  const viewAllTaskEl = document.createElement('li')
  const quickViewEl = document.createElement('li')

  todoListEl.innerText = todoText
  viewAllTaskEl.innerText = todoText
  quickViewEl.innerText = todoText
}
```

```
todoModalUL.appendChild(todoListEl)
allTaskModal.appendChild(viewAllTaskEl)
quickViewModal.appendChild(quickViewEl)
```

28. Using the same Udemy video, I was able to mark elements in my list as done. One issue, however, was maintaining each list to be marked as done; which was fixed by noticing when I click an element in the Upcoming Task it disappears, which then also crosses the element out on the Quick View and View all Task list.

```
quickViewEl.addEventListener('click', () => {
  todoListEl.classList.toggle('completed')
  updateLS()
})

quickViewEl.addEventListener('click', () => {
  viewAllTaskEl.classList.toggle('completed')
  updateLS()
})

quickViewEl.addEventListener('click', (e) => {
  e.preventDefault()

  quickViewEl.remove()
  updateLS()
})
```

29. In the same Udemy video, I was also able to implement a local storage mechanism. Originally, after refreshing the page everything that was marked as completed was unmarked. However, I quickly realized that I forgot to update the local storage after something is marked as done.

```
if (todo && todo.completed) {
  todoListEl.classList.add('completed')
  viewAllTaskEl.classList.add('completed')
  quickViewEl.classList.add('completed')
  quickViewEl.remove()
  todoListEl.remove()
}

input.value = ''

updateLS()
```

30. One element of the “Get Started” page was making the list of “View all Task” show descendingly, the purpose of this is to make everything that was added first (so should be completed first) be towards the bottom of the list. The way I implemented this was making the flex direction to be column reverse.

```
.taskList {
  display: flex;
  flex-direction: column-reverse;
}
```