**SI 539 Waiver Part 1: CSS component**

# Speciﬁcations:

The instructions can be found here. You must use your umich google account to access it:

* **You may not collaborate with others**. Students who give or receive assistance will be reported to the Dean.

## Instructions

1. If you haven’t already, download the html file, starter css folder, and the images folder, which can be found on this assignment’s Canvas page You may *NOT* modify the html in any way other than to comment out the link to your css or js file. Any other changes will result in a 0 for the exam.
2. You will use the same starter code for both the CSS portion and the js portion.
3. Write your own css to achieve the styles below.
   1. Code mobile first. ***There will be a 10 point deduction for every use of the term max-width***.
   2. Do ***NOT*** add any properties that you do not need.
4. You will write five media queries in this portion of the exam.
5. *We highly recommend watching the Example video on Canvas before beginning the Waiver exercise.*

NOTE : Before we start,

Unzip the Folder. **SI539Waiver-2022-main**

1. **Wavier.html** – Open in Chrome Browser.

2. CSS Folder has **waiver-starter.CSS (Original Code – Base Code)**

3.**waiver.css- Code Base Modified**

4.Images Folder will have all Images – **Provided by Company**

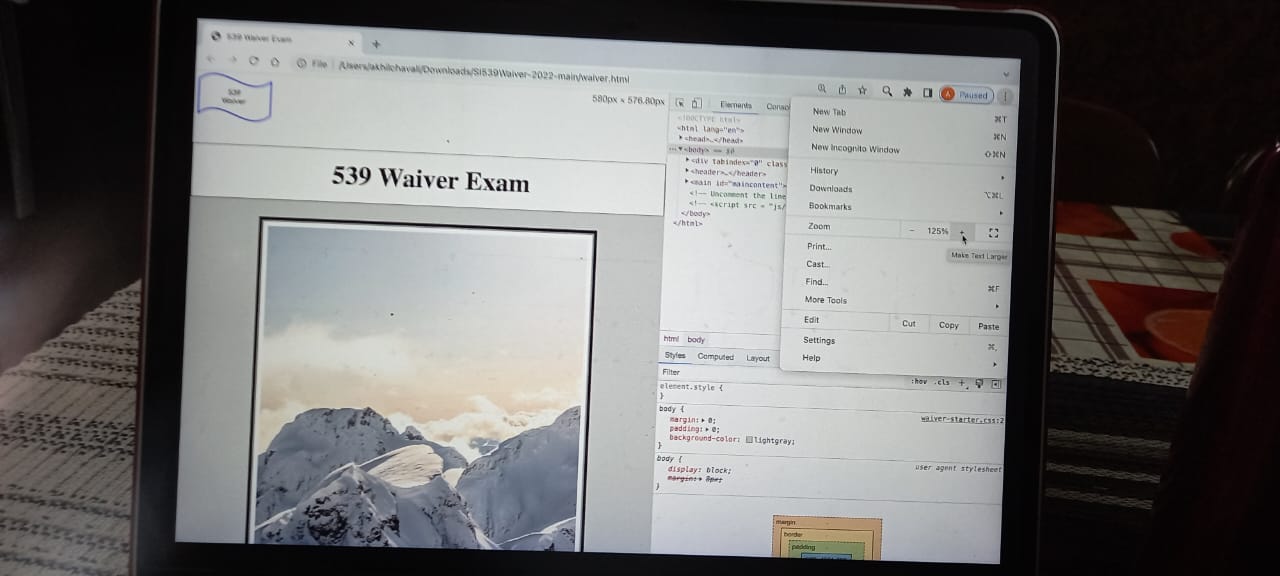
**Solutioning:**

## Problem 1: Mobile First View

In the mobile view:

1. The container class should use grid and:
   1. Take up 70% of the screen but have a maximum width of 375px.
   2. Be centered on the screen.
   3. Have a single column that takes up the entire width of the grid. (Use fr or %)

Open in Chrome the Wavier.HTML File, Go to Developer Tools and Zoom (+ OR -) if Size of the Browser is less than 600 Pixels or in Mobile view, Single Column should be Shown. Only one Image will displayed and Start Scrolling Down you can see more Single Images. You can see the Screen Resolution of less than 600 Pixels on the Top Corner in the below image.

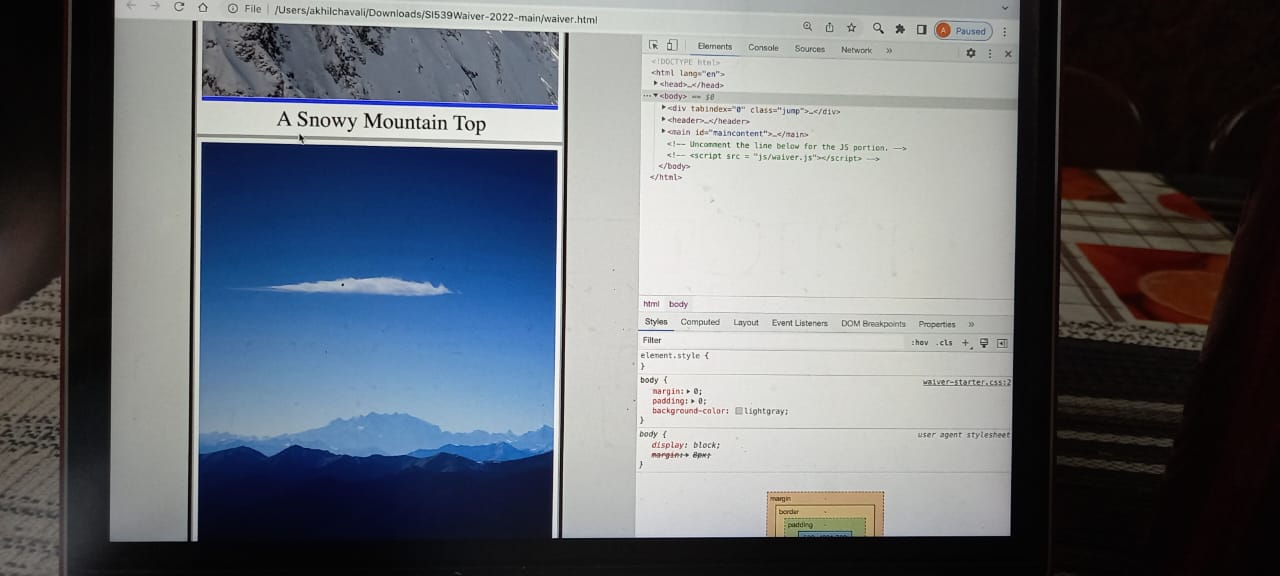


1. All descendants of the grid element should
   1. Shrink/grow with the browser and
   2. Fit within the grid columns ***without*** using overflow hidden.

(Did Not Write Overflow Hidden as specified in Point B.) If we use, Overflow Hidden, if the Container size is reduced less than the image size, the image will be shown outside of the Container and will be removed or Rectified.

1. Every figure caption should be:
   1. Centered
   2. The first letter of each word in the caption should be uppercase.

The First Letter of Each Word is In Uppercase and the Image is also centered.



Code 1 : Is Written in Wavier.CSS open in Notepad to Check.

/\* ------------1st Point Code---------------MObile view------------------------------------------------------------- \*/

Container Size / Postion /

.container{

width: 70%;

position: relative;

top:0;

bottom: 0;

left: 0;

right: 0;

margin: auto;

}

Container Image : Shrink or Expand

.container img{

height: 90%;

width: 100%;

}

Container Image : First Letter Capitol

.container figure figcaption{

text-transform: capitalize;

text-align: center;

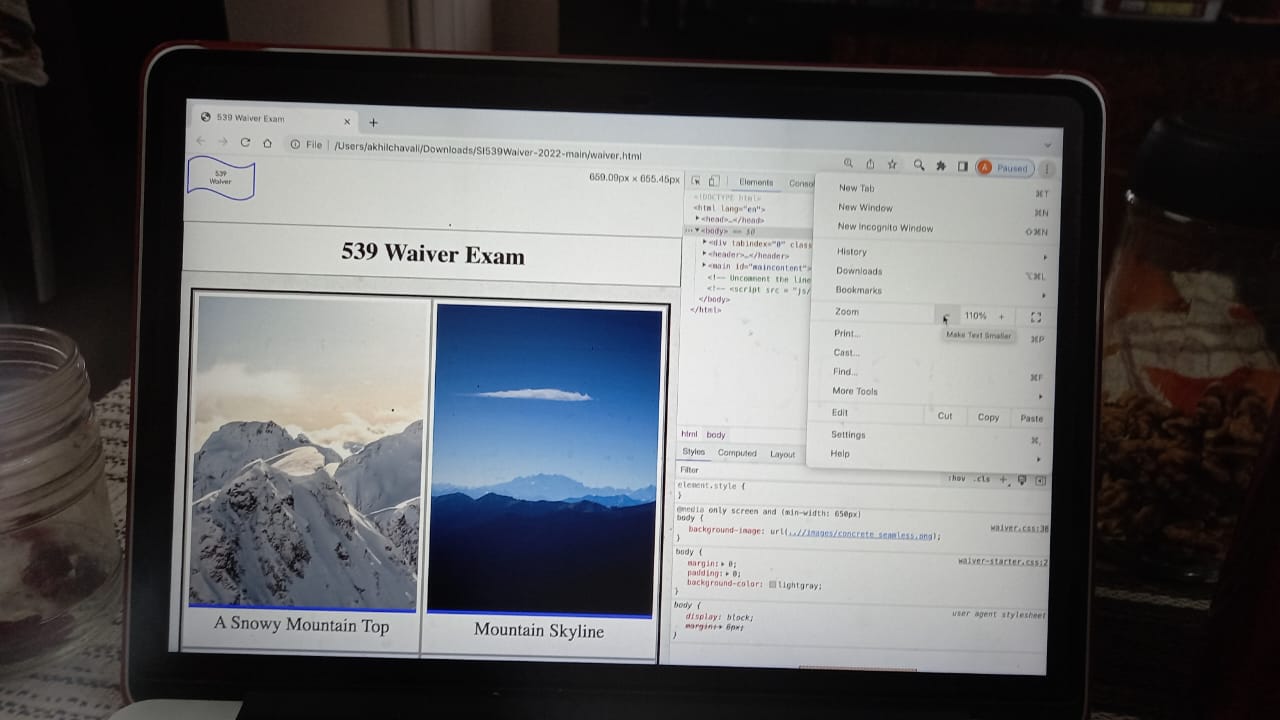
}

===============

## Problem 2: 650px and up (do not use max-width)

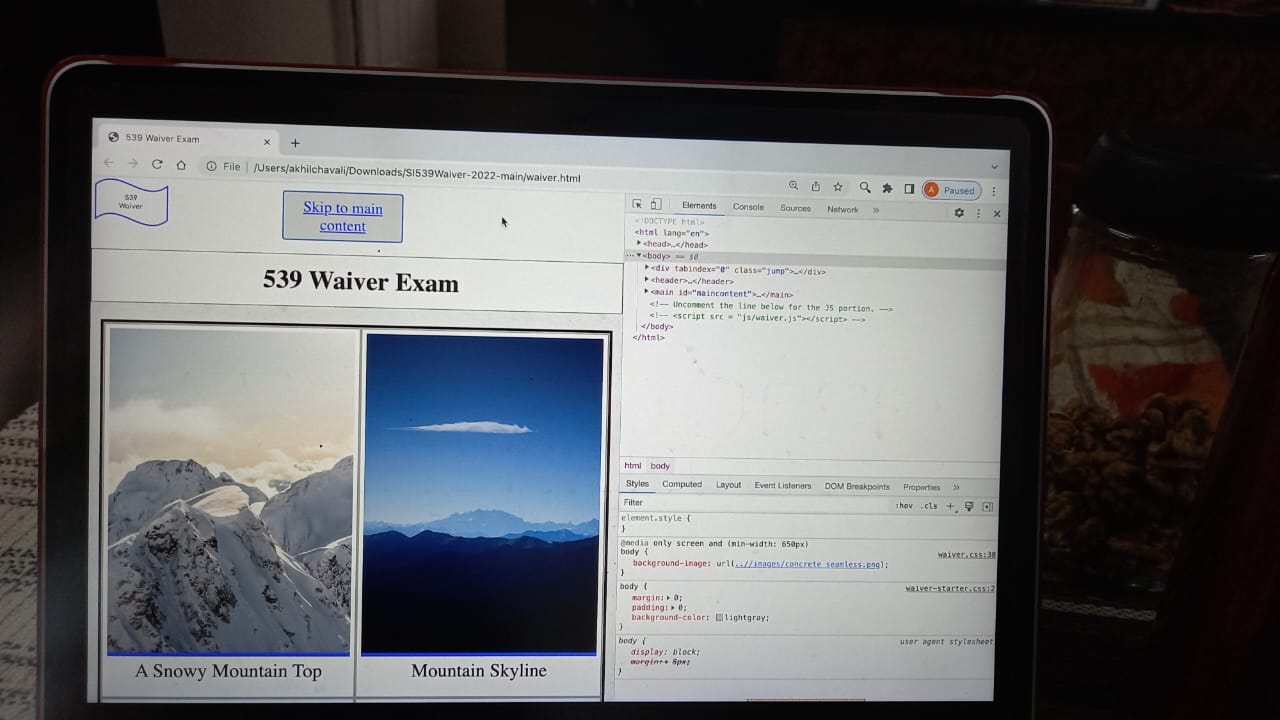
In screen sizes over 650px:

Fits 2 Images - More than 650 to 750

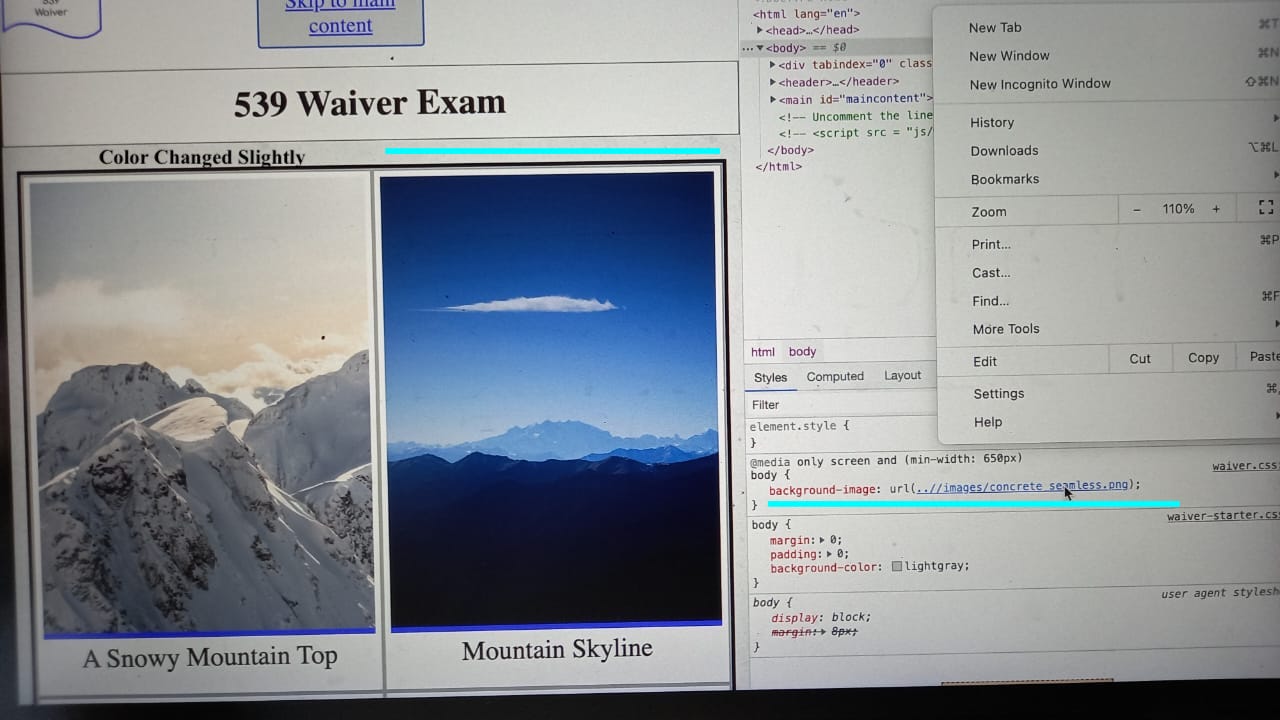


1. “Skip to main content” should appear on the screen the first time we hit the tab key.

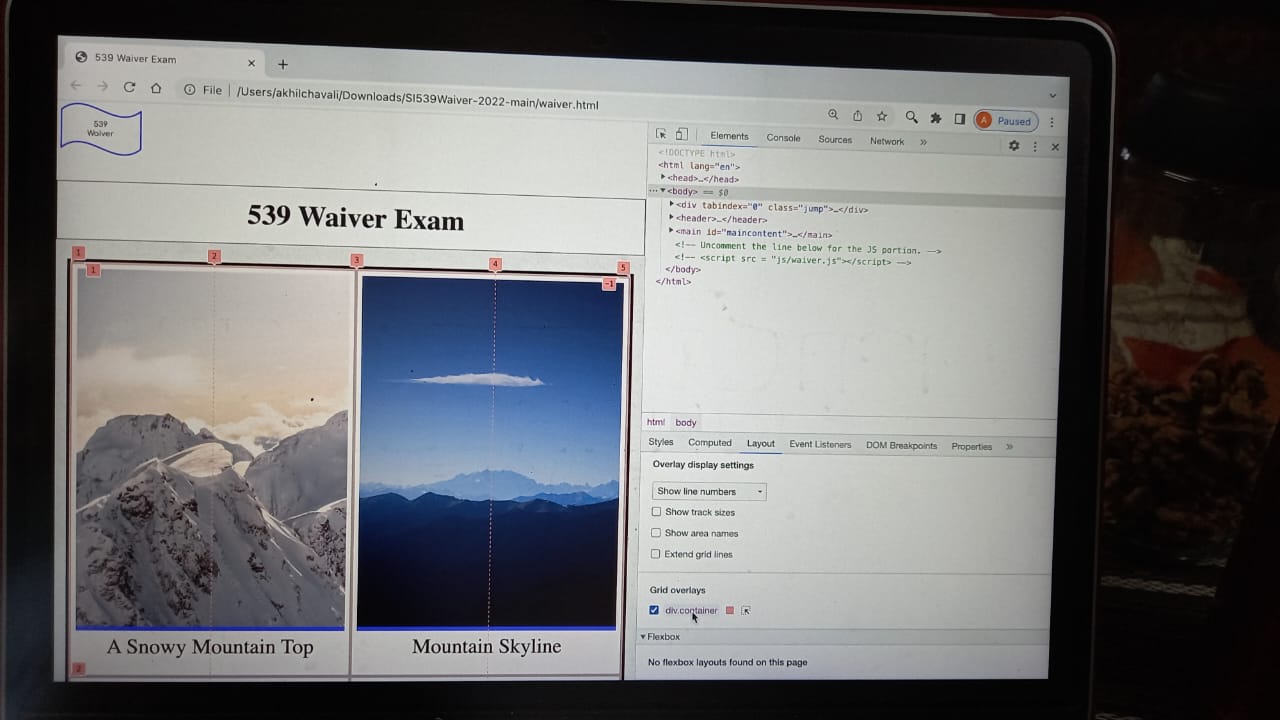
When Tab Clicked, Skip to Main Content will be displayed



1. The container class should:
   1. Take up 95% and no longer use a maximum width. (In the video it may reference 90%, but it should be 95%.)
   2. Use the image ***concrete\_seamless*** (from the images folder) as a background image.



* 1. Have three equal columns that each cover 25% of the grid width.
  2. The children should be spaced out to match our example – equal spacing around each side of the children



1. All children of the grid element should
   1. Cover ***two*** columns
   2. The children in rows 1, 3, 5, etc should start in the 1st column
   3. The children in all other rows start in the 2nd column.

ODD Children were placed in the First Column and Even Children are placed in the Second Column as per the Requrement. HTML File has the Order.

1. If we add more children the pattern should continue

**CODE 2 : is Written in Wavier.css**

/\*------------------------------ problem - 2 width upto 650px ---------------------------------------------------------------- \*/

@media only screen and (min-width: 650px) {

.container {

display: grid;

grid-template-columns: repeat(4,25%);

Comment : Divided in to 25 % each for each Children

}

.container {

width: 95%;

}

body{

background-image: url("..//images/concrete\_seamless.png");

Comment : Body Image Color Changed

}

figure:nth-child(odd){

grid-column: 1/3;

Arrangement of Children in Odd in Column 1

}

figure:nth-child(even){

grid-column: 3/5;

Arrangement of Children in Even in Column 2

}

}

Code to SKIP Main Content by Using TAB

/\* ------------------------------------------Skip to main Content--------------------------------------------- \*/

.jump {

left:-999px;

position:absolute;

top:auto;

width:1px;

height:1px;

overflow:hidden;

z-index:-999;

}

.jump:focus, .jump:active {

left: auto;

top: auto;

width: 20%;

height: auto;

overflow:auto;

margin: 10px 35%;

padding:1% 1%;

text-align:center;

font-size:1.2em;

z-index:980;

background-color:rgb(223, 223, 223) ;

}

.jump a{

color: blue;

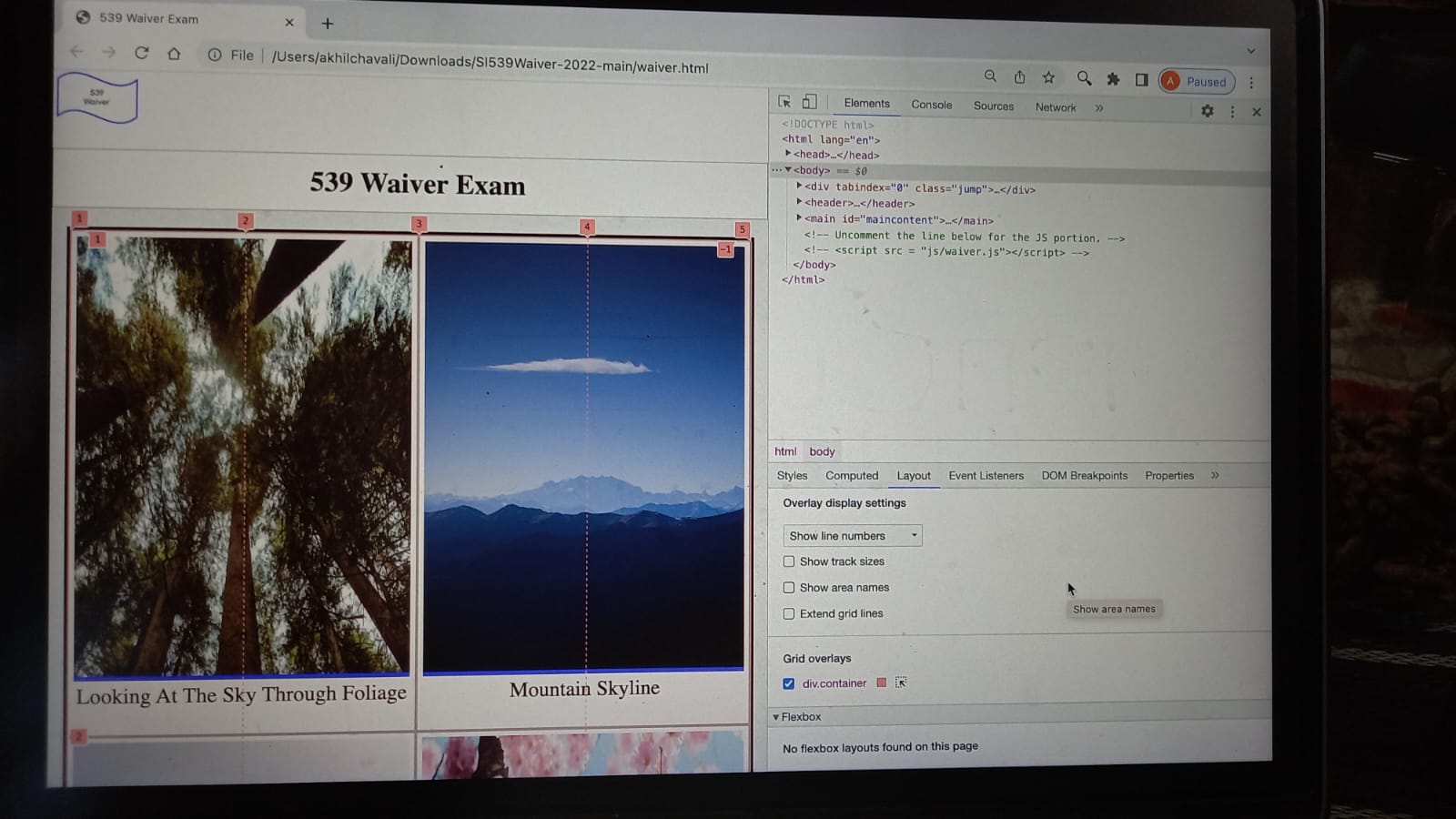
}

## Problem 3: 750px and up (do not use max-width)

In screen sizes over 750px:

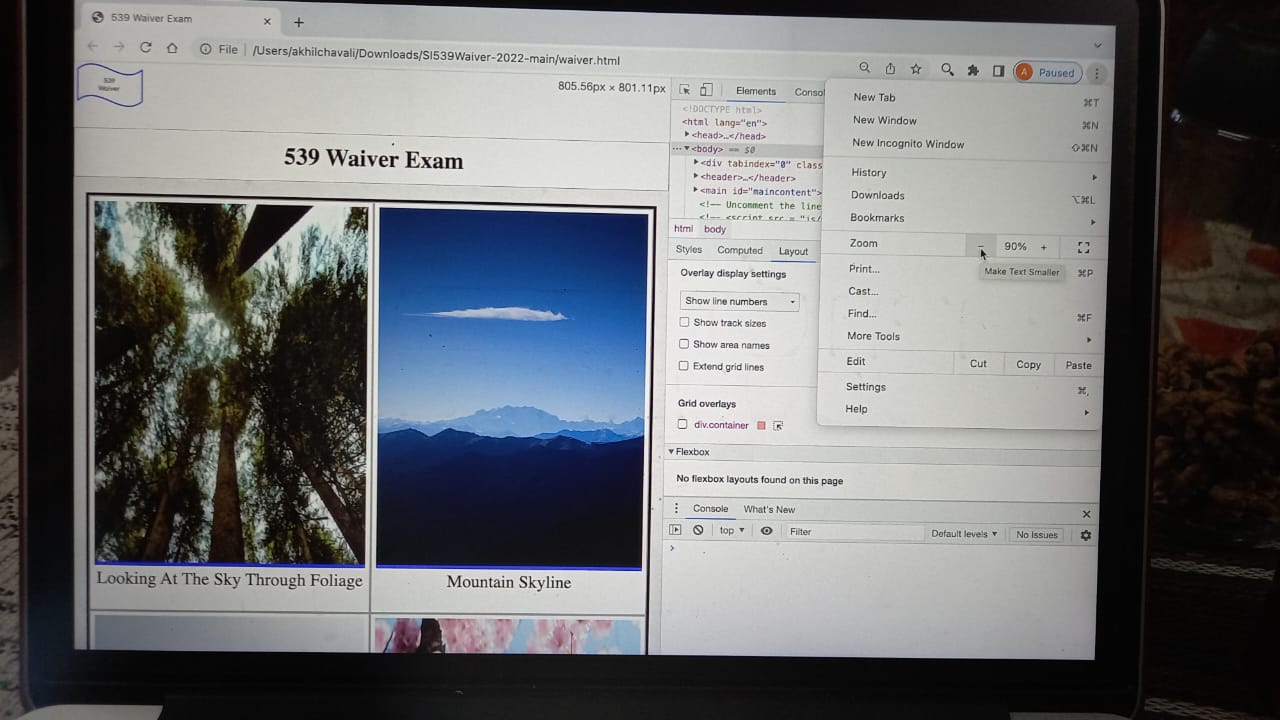
1. The container class should:
   1. Have four equal columns that are each 20% of the width of the grid.
   2. Lay out the children elements as far to the left and right of the grid as possible.

Each Grid is set to 20% of Size, The Image is set to Edges without any Gap set to Max Size



1. All children of the grid element should
   1. Match the layout so that each child element covers two columns
   2. The last child always starts in the 2nd column.
   3. Do ***NOT*** add any gap properties or properties that do not modify this view

Comment : The Last Image displayed in First Columns. Refer to Point 2 all points (A,B,C)



Code base 3 : in Wavier.CSS

/\*------------------------------ problem - 3 width upto 750px ---------------------------------------------------------------- \*/

@media only screen and (min-width: 750px) {

.container {

display: grid;

grid-template-columns: repeat(4,25%);

grid-auto-flow: dense;

Comment : 25% Size Grid of 4 Columns (Adjusted to 20% to accommodate Edges)

}

.container {

width: 95%;

}

figure:nth-child(odd){

grid-column: 1/3;

grid-auto-rows: auto;

}

Each Child covers 2 Columns

figure:nth-child(even){

grid-column: 3/5;

grid-row: inherit;

Last Child will start in the 2nd Column

}

figure:last-child {

order: -1;

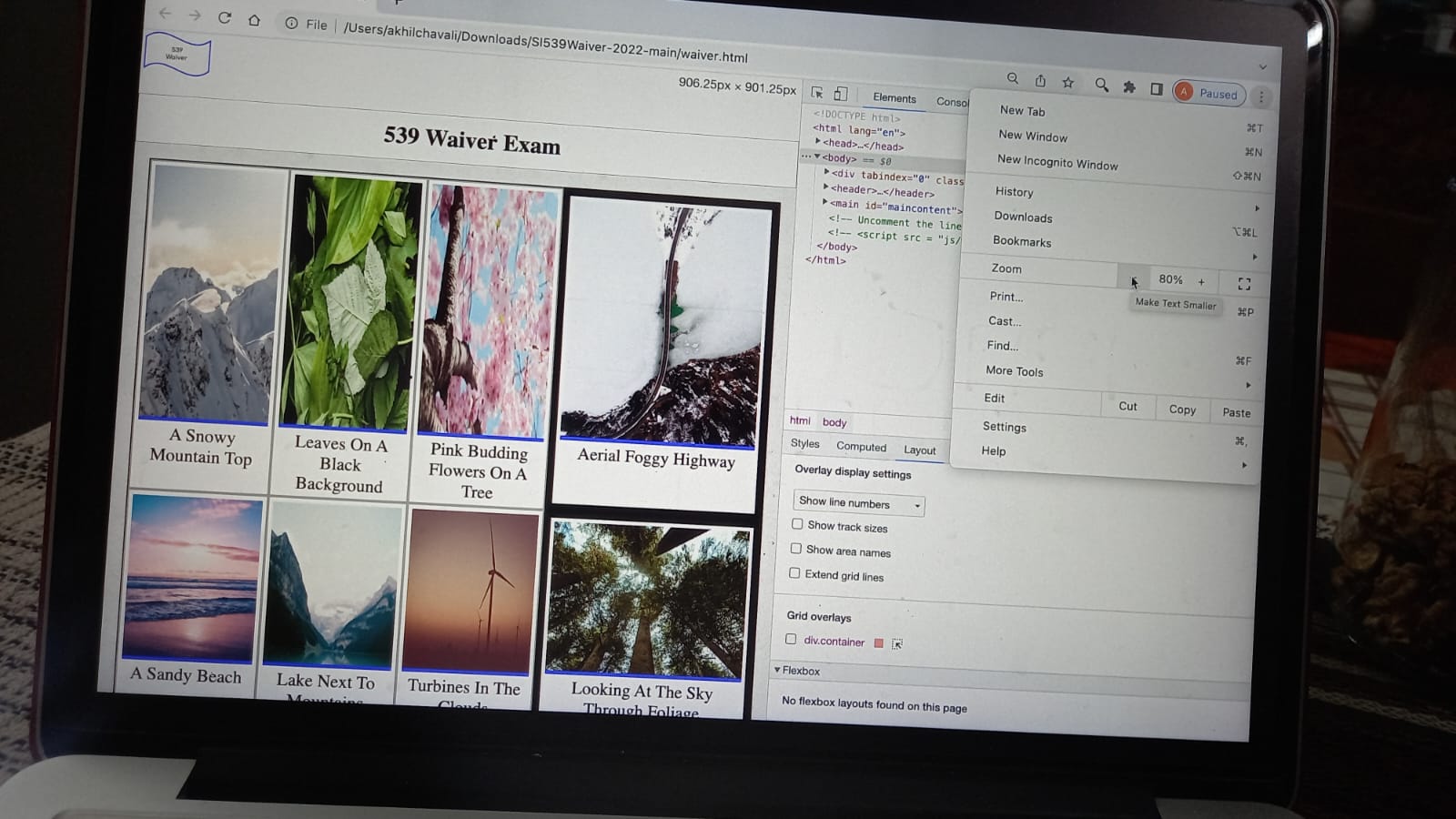
}

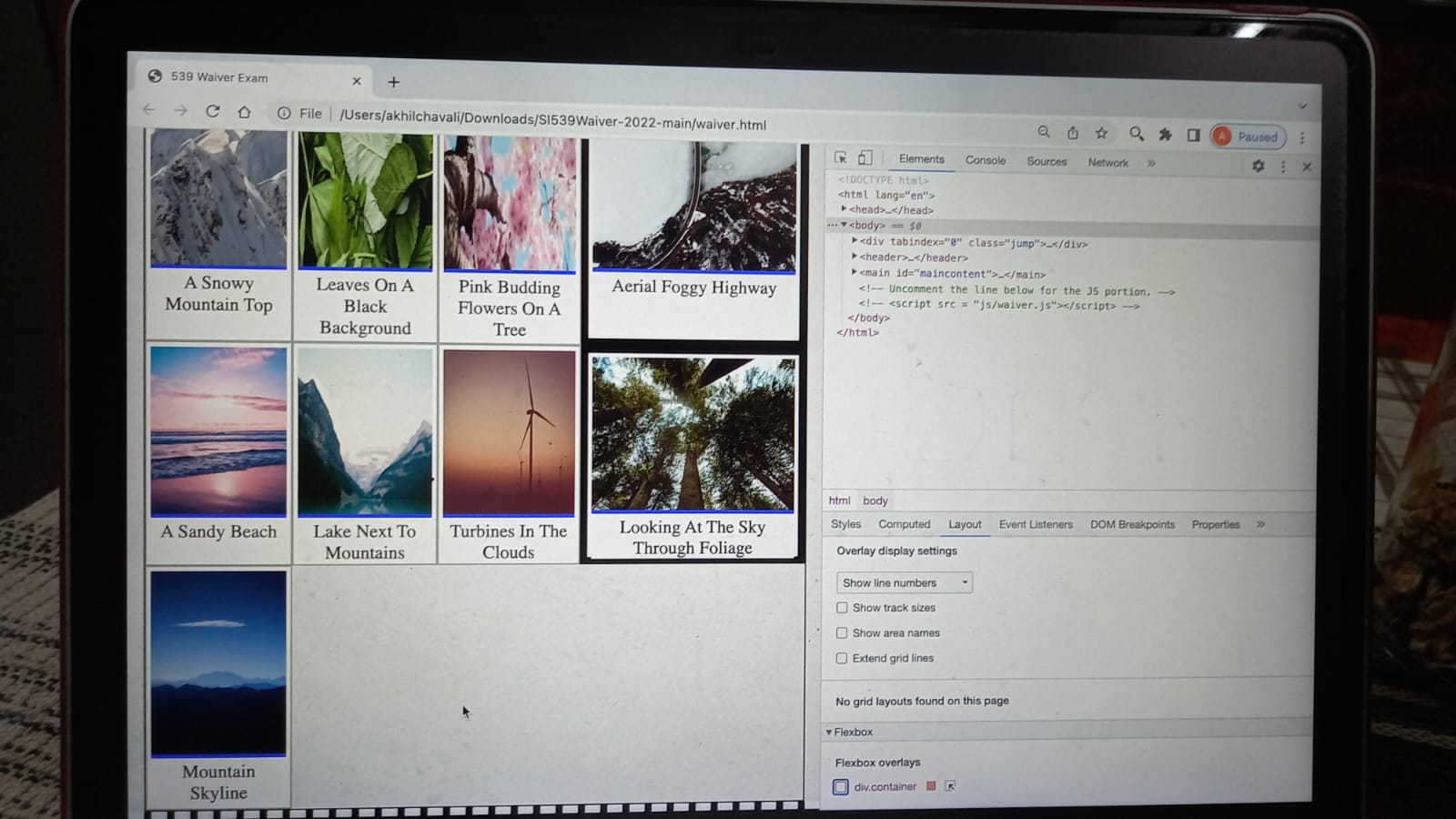
}

## Problem 4: 900px and up (do not use max-width)

In screen sizes over 900px:

Each Grid is set to 20% of Size, The Image is set to Edges without any Gap set to Max Size





All the below Points 1 , 2 and 3 Were Taken Care

1. The figure elements should have a width of 20% except for every 4th element that should have a width of 40% and a 10px border. (Remember - we may add more children elements.)
2. The 2nd figure (the one with the caption “mountain skyline” should be the last image displayed.
3. The container class should now use flex and lay out the children images as shown.

CODE 4 :

/\*------------------------------ problem - 4 width upto 900px ---------------------------------------------------------------- \*/

@media (min-width: 900px) {

.container {

display: flex;

flex-direction: row;

flex-wrap: wrap;

border: 2px solid black;

**Used FLEX – Depending up on the Image the Children will be fit accordingly.**

}

.container {

width: 95%;

}

.container img{

height: 80%;

width: 100%;

}

figure:nth-child(even)

{

order: 0;

}

figure:nth-child(odd)

{

order: 0;

}

figure:nth-child(5n),figure:nth-child(9n)

{

flex-grow: 1;

border: 10px solid black;

4th Image in Each Row having Border 10Pixels

}

figure:nth-child(2){

order: 1;

Second Image will be Displayed at Last

}

.container figure{

width: 20%;

}

}

## Problem 5: 1050px and up (do not use max-width)

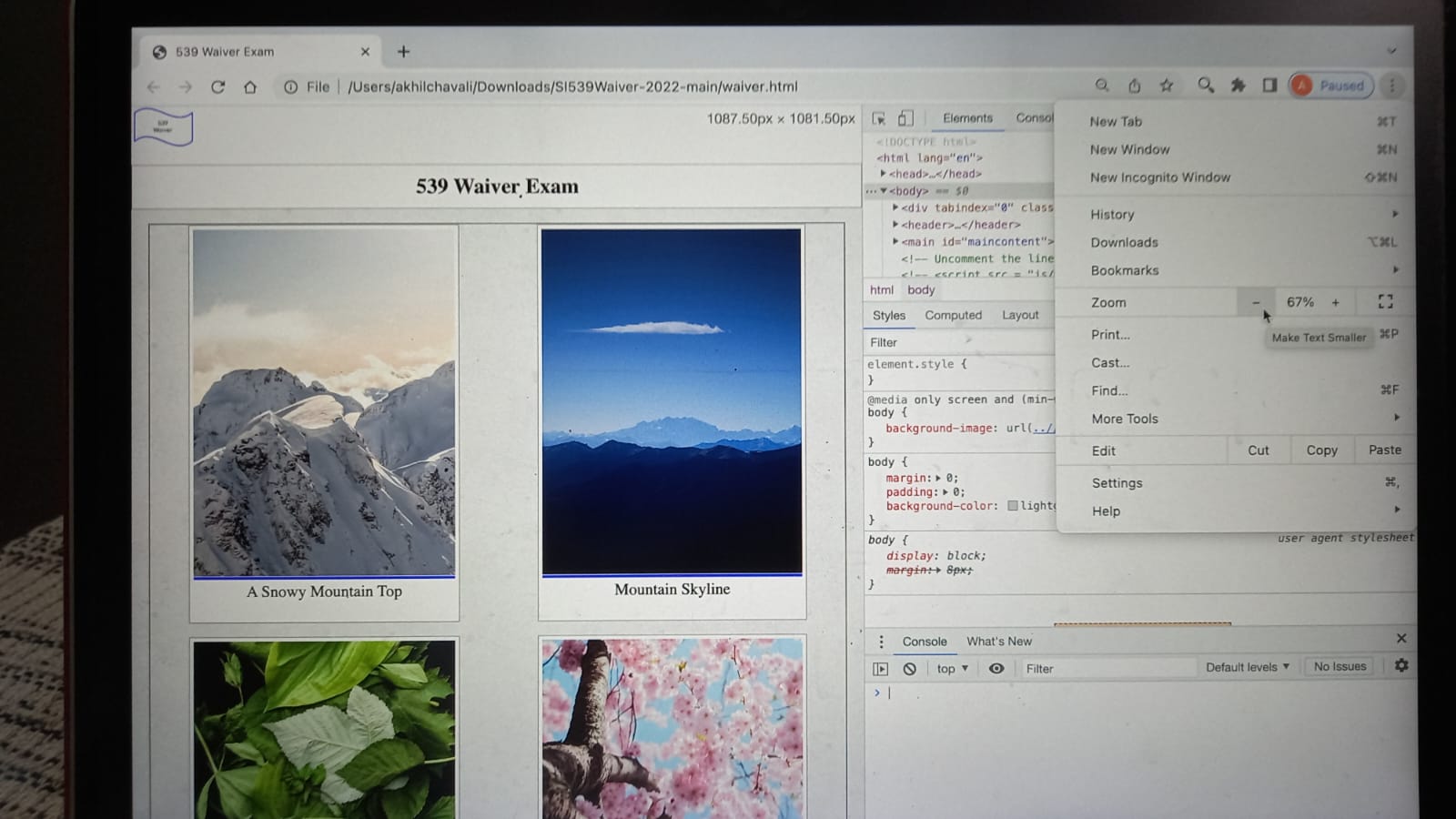
In screen sizes over 1050px:

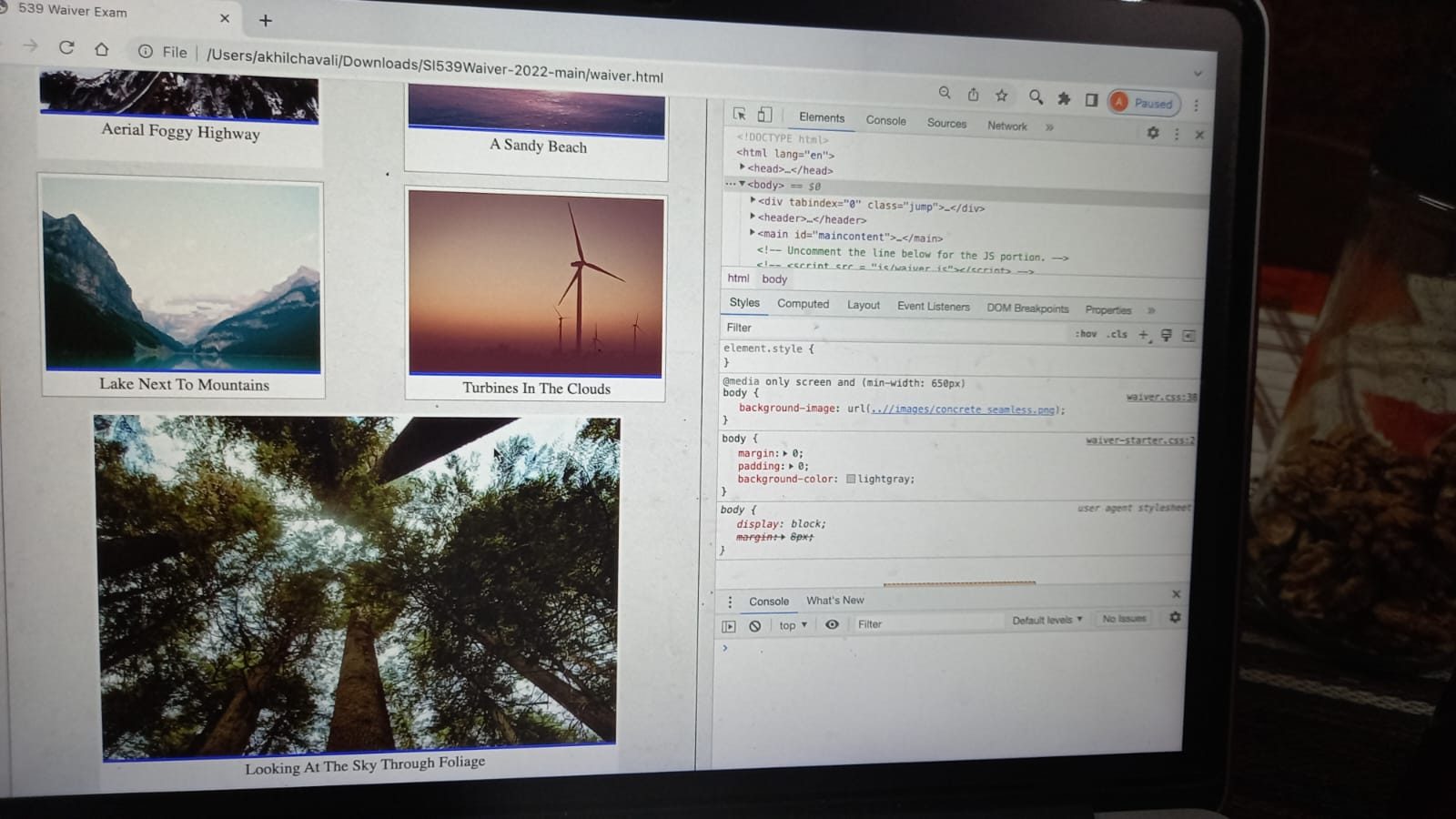
1. **ALL** figure elements should have a width of 75% and a margin of 20px on the bottom.
2. The 2nd figure should appear back in the proper order.
3. The container class should use the grid property again and:
   1. Have two equal columns that together cover the entire width of the grid.
   2. Center the children horizontally
4. All children of the grid element should
   1. Match the layout so that each child element covers one column
   2. The last child covers two columns.
   3. *Note: If we add more children, this pattern should continue.*

In the same media query:

1. Rotate every other child element (except the last one) by 15 degrees or -15 degrees to match the screenshot.
2. Ensure that the figures don’t exceed the grid container.

ALL ABOVE Points are Covered





**CODE BASE Point 5**

/\*------------------------------ problem - 5 width upto 1050px ---------------------------------------------------------------- \*/

@media only screen and (min-width:1050px)

{

.container{

width: 95%;

display: grid;

grid-template-columns: repeat(auto-fit,50%);

justify-items: center;

}

.container figure{

width: 75%; (Each Image width 75%)

margin-bottom: 20px;

(Each Image having Margin of 20 Pixel at bottom)

}

.container img {

height: 90%;

}

figure:nth-child(5n),figure:nth-child(9n)

{

border: 0px solid white;

}

figure:nth-child(even)

{

order: 0;

grid-column: 2;

}

figure:nth-child(odd)

{

order: 0;

grid-column: 1;

}

figure:last-child{

grid-column: span 2; (**Spanned 2 Columns last Image)**

}

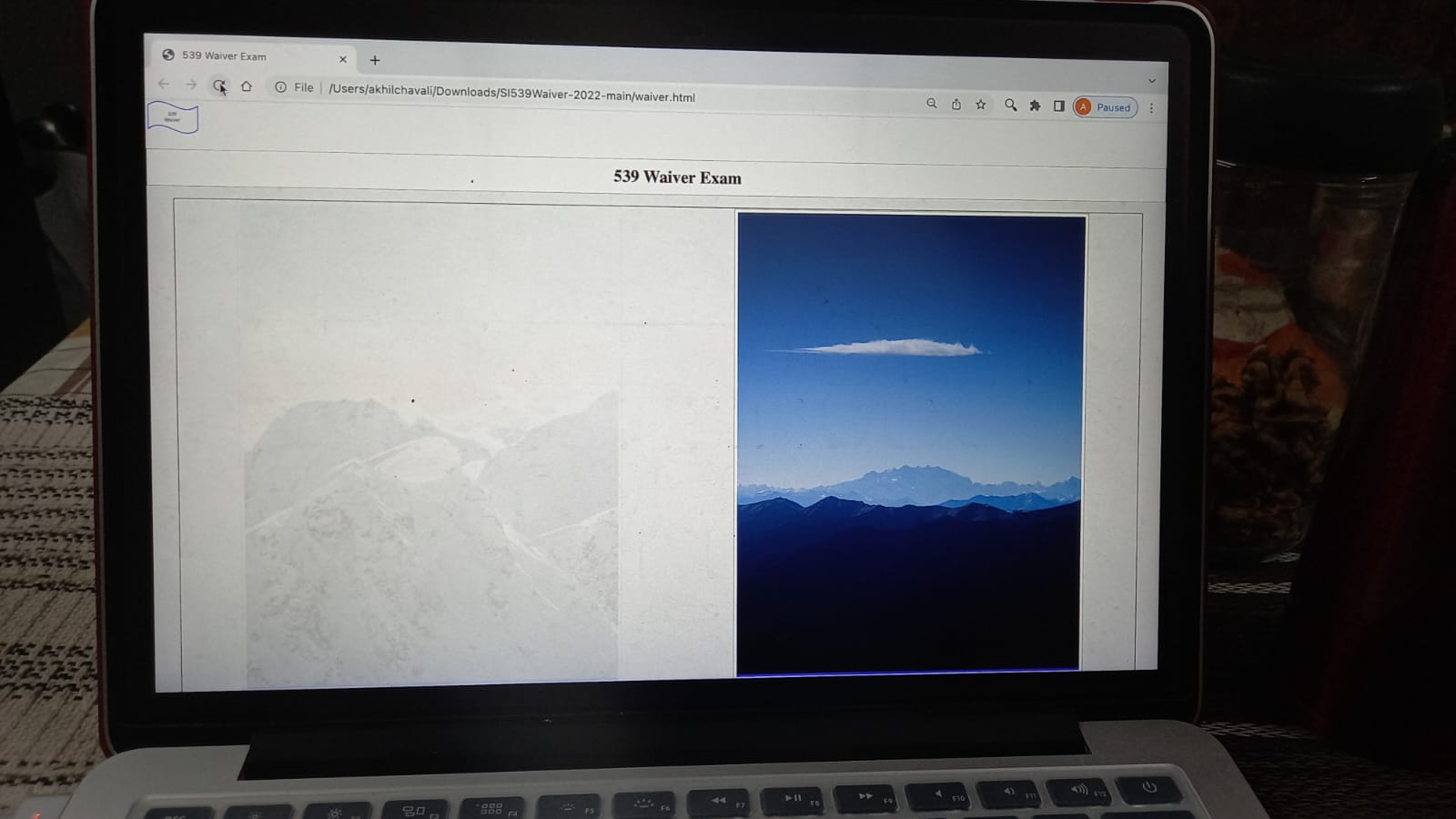
}

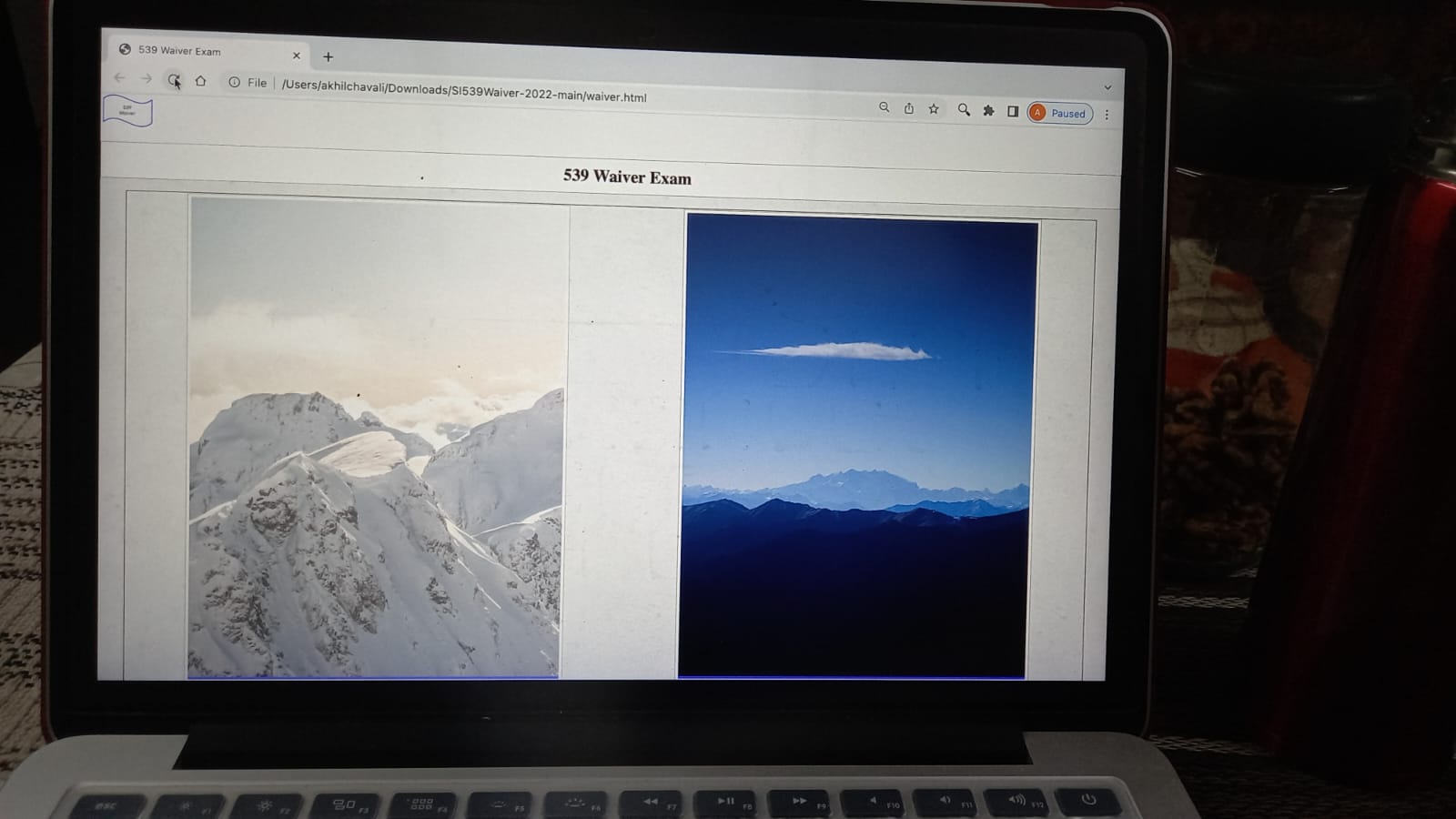
## Problem 6: Animation and prefers Reduced motion

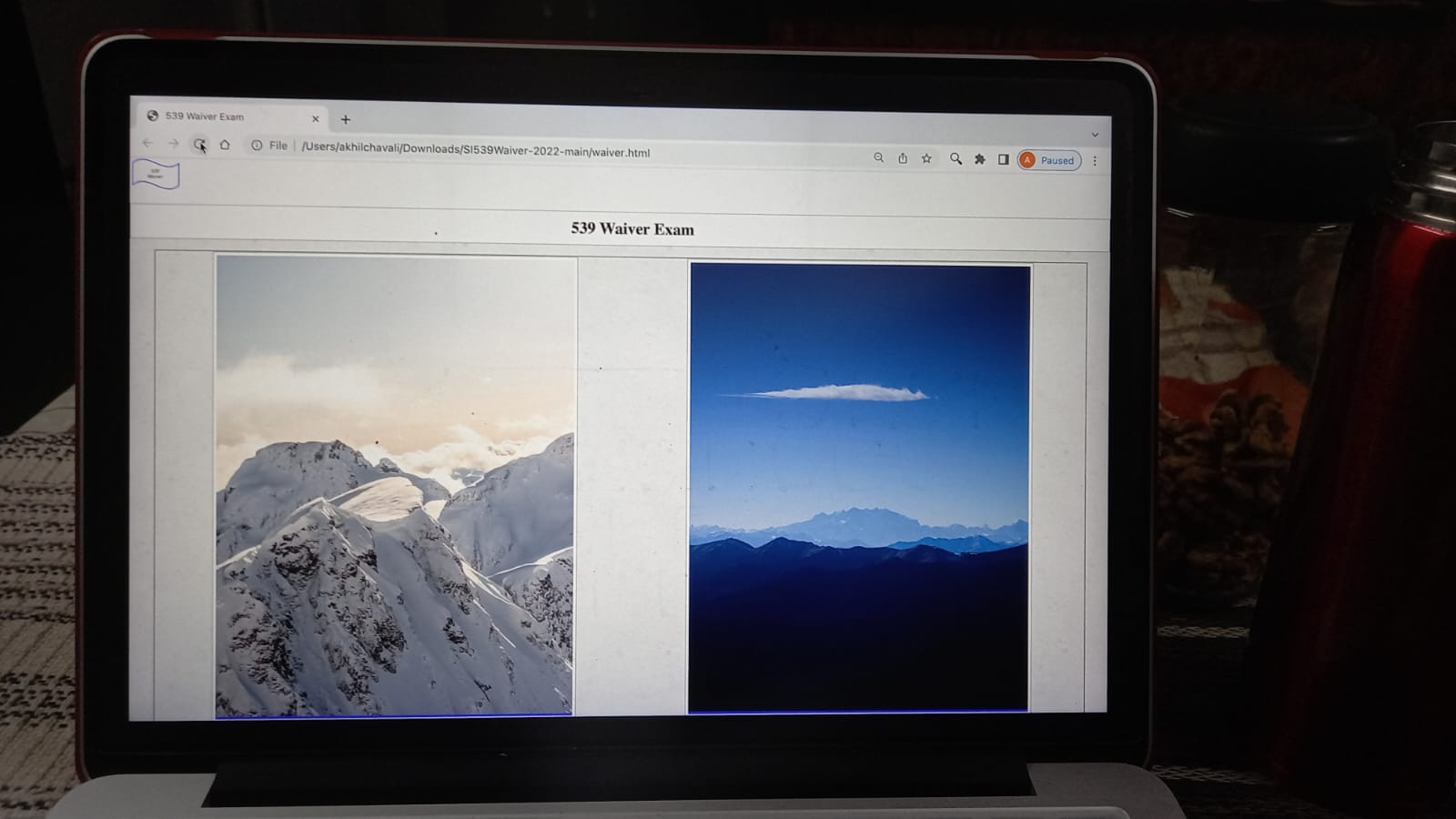
1. Use keyframes to alter the opacity of the first image from complete transparency to full opacity over the course of 10 seconds.
2. Write a media query to remove this animation when the user has reduced motion as a preference.

**Reminder, do NOT change the html file at all.**

**Refresh HTML by Coming ut of Developer Tool, The First Image will be Displayed from Nothing to Full Image in a slow Animated Manner..**

****

****

****

**Code 6 Base Code :**

/\*------------------------------ problem - 6 Tranperent Animation ---------------------------------------------------------------- \*/

figure:nth-child(1)

{

animation:animatedColor 10s ease;

}

@keyframes animatedColor{

(Code Keyframes is used for Animation from Transperent to Opaque)

0%{

background-color: rgba(224, 224, 224, 0);

opacity: 0.1;

}

100%{

background-color:white;

}

}

@media (prefers-reduced-motion) {

(If user wants to Disable Animation it will be removed)

figure:nth-child(1) {

animation-name: dissolve;

}

}

# Starter Code

The starter code can be found at: <https://github.com/UMSIComplexWebDesign/SI539Waiver-2022>

[(Links to an external site.)](https://github.com/UMSIComplexWebDesign/SI539Waiver-2022)

# Submission Requirements:

* 1. Host the site at the server of your choice and submit a link to a working demo. We want to see the **deployed page**, not the code.
  2. We will not accept ﬁle uploads for grading, or download code from the Internet.