

My Mom's Life: An Animated Timeline

*Responsive at mobile, tablet, and desktop screen sizes

Part 1

Purpose: In order to capture and pay tribute to my mom's unique lived experiences growing up as the daughter of a colonel in the Indian Army, I implemented an interactive timeline that maps the trajectory of my mom's especially fascinating life through a linear narrative structure.

Information: At each major timepoint, I display images from my mom's life at that time as well as a brief, succinct description of my mom's experiences, common activities, and relationships there. Because of her associatively intimate position to the Indian government, there are a number of timepoints that specifically reference political upheaval and unrest in India and how those external circumstances directly affected her life. There are other spaces where I speak to my mom's personal challenges and changing goals and aspirations as she navigated frequently moving schools, family dynamics, college education, and marriage. Finally, I hope that I effectively convey admiration for my mom's strength, courage, and fervor for life.

Interesting & Engaging: It was my intention to leverage animations in such a way as to centralize my mom's story and highlight her life without unnecessary visual distraction. While the most interesting component of the website is certainly the narrative, I have tried to incorporate seamless transitions between timepoints to create visual intrigue, set the scene, and draw the user into the fold.

Target Audience: I initially planned this website to be a birthday gift for my mom. Through conversations with her during the development process, however, I have come to appreciate just how incredibly captivating her life has been thus far. Her rich experiences afford valuable lessons such that anyone can read about her life and take something meaningful away.

Part 2

- **Interaction Type:** Primarily implemented JQuery and CSS animations
- **Action:** Scroll down (slowly) to the top of the succeeding image, pause for 1 second
 - The animations occur automatically and are calculated based on mouse position.
 - Animations include automatically changing blur filter, background image, scroll-snap, viewport height adjustment, and text slide-in.

Part 3

- **Tool:** JQuery

- **Why?:** JQuery vastly simplifies JavaScript code, and makes it especially convenient to modify HTML elements dynamically in fewer lines of code. Moreover, because I wanted to create a vertical timeline, I needed to constantly keep track of mouse position, which JQuery enables quite easily.
- **How?:** I leveraged jQuery to implement a set of functions that calculate the precise mouse position and update the DOM accordingly on this metric. Within these functions, I referenced and applied different CSS classes to modify content and prompt vertical scroll animations.
- **What does it add?:** JQuery supports both the vertical timeline layout and efficient access, retrieval, and adjustment of CSS classes, allowing for seamless transitions between timepoints. The animations implemented as a product of the interaction between CSS and JQuery add visual interest and intrigue to the timeline, prompting users to continue scrolling.


Part 4

I significantly altered the design and scope of the website. Initially, I had planned to incorporate a hover effect over each image, such that the image would change to one of a map of the location at which my mom was during that timepoint. I implemented this functionality and subsequently realized that hover effects are especially inconvenient for accessibility purposes and responsive design. Accordingly, I scrapped this component from the website altogether. Moreover, as I began deploying the website, I modified a number of the smaller micro-interactions and animations that I had conceived of. For instance, I implemented a parallax effect, a blur filter, and a text slide-in that I had not previously committed to.

Part 5

I faced major challenges in integrating vanilla JavaScript functionality (e.g., cloning template tags) with the jQuery functionality that was enabling the HTML element updates. I generally also had a tough time with the learning curve for CSS animations and layering those atop a flexbox layout.

Appendix


 **WAVE**
web accessibility evaluation tool


powered by
[WebAIM](#)


Address: <https://anikavaish01.github.io/pui-hw/timeliir>


Styles: OFF ☐ ON ☒


Summary


 Summary


 Details


 Reference


 Order


 Structure


 Contrast


 0
Errors

 0
Contrast Errors

 0
Alerts


 15
Features

 16
Structural Elements

 0
ARIA

[View details >](#)

Congratulations! No errors were detected! Manual testing is still necessary to ensure compliance and optimal accessibility.


 **WAVE**
web accessibility evaluation tool


powered by
[WebAIM](#)


Address: <https://anikavaish01.github.io/pui-hw/timeliir>


Styles: OFF ☐ ON ☒


Details


 Summary


 Details

 Reference

 Order

 Structure

 Contrast

☒  15 Features

☒ 14 X Alternative text
