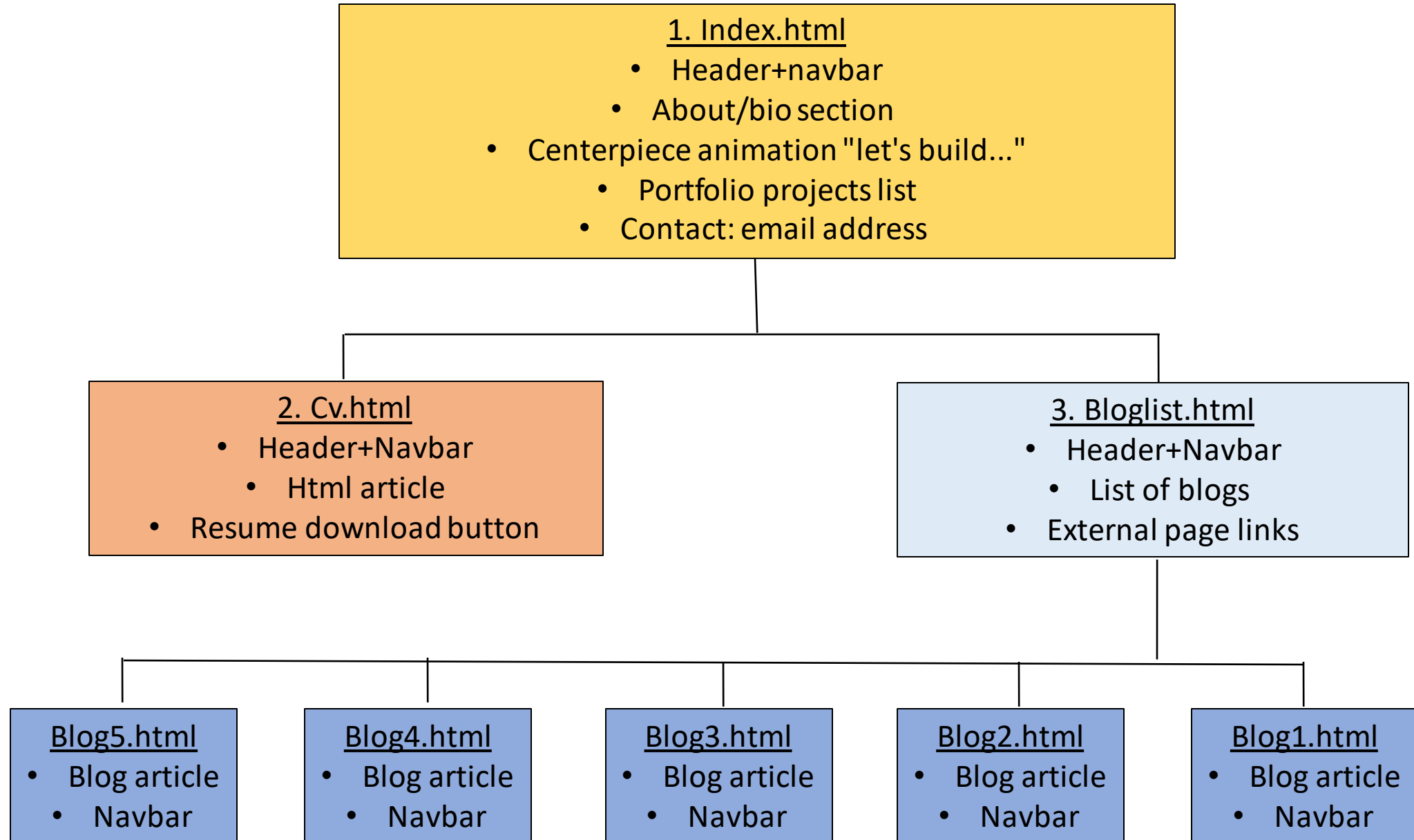
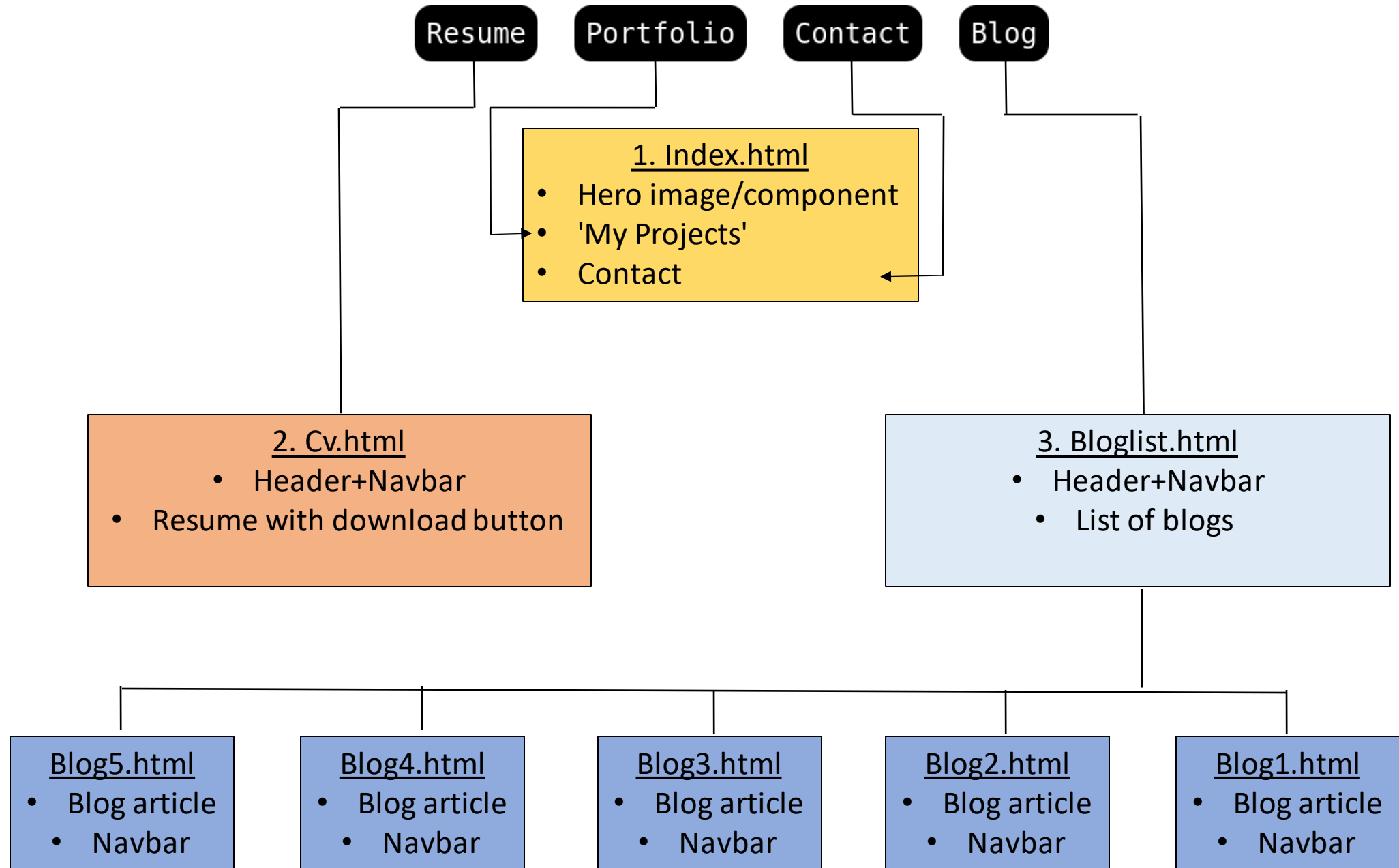


# Assignment 2

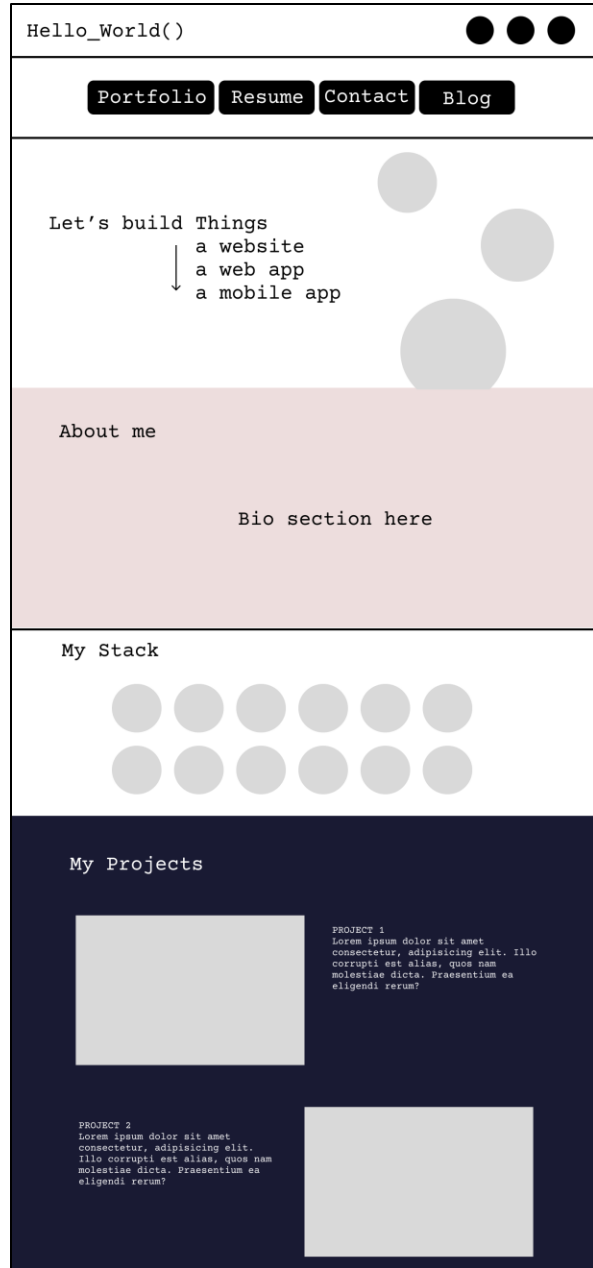
Portfolio website using HTML5/CSS3

## Sitemap and components used



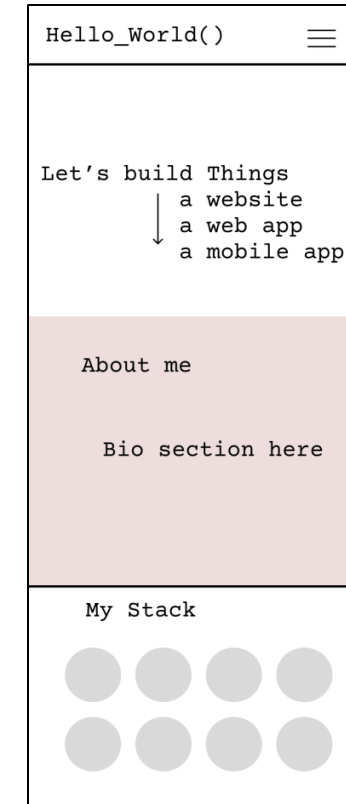


# Wireframes – Responsive index.html



600-1000px (tablets-  
desktop)

Essential info (Bio/skills/  
projects/contact) were all  
put on a single page as time  
spent on portfolio sites is  
short... Also, I felt it  
(hopefully) showcases an  
ability to build 'long'  
responsive websites.



<600px (mobile)



# Wireframes - CV.html

- Resume was written directly onto page as html
- Given time constrains decision was made not to write paragraphs as individual sections which will help for future styling changes and will be fixed in the future
- CV is easily downloadable however with proper styling

Hello\_World()



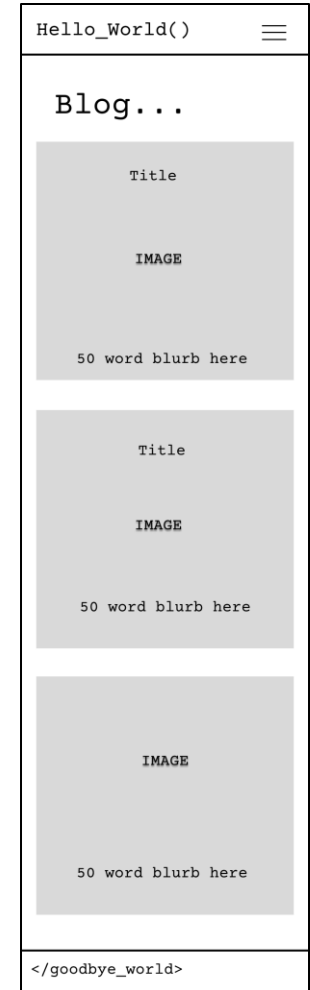
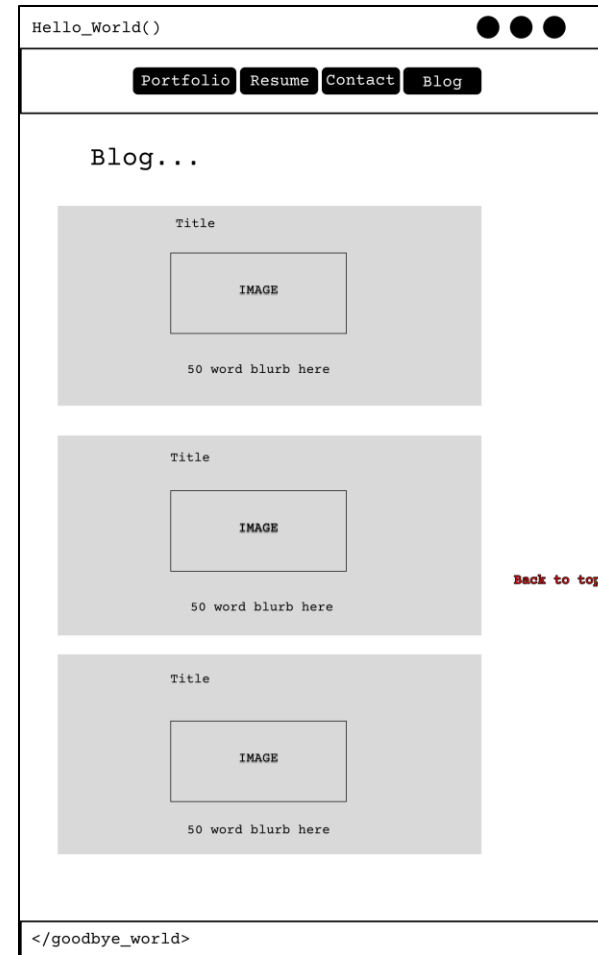
Portfolio Resume Contact Blog

Resume  
<article/>

</goodbye\_world>

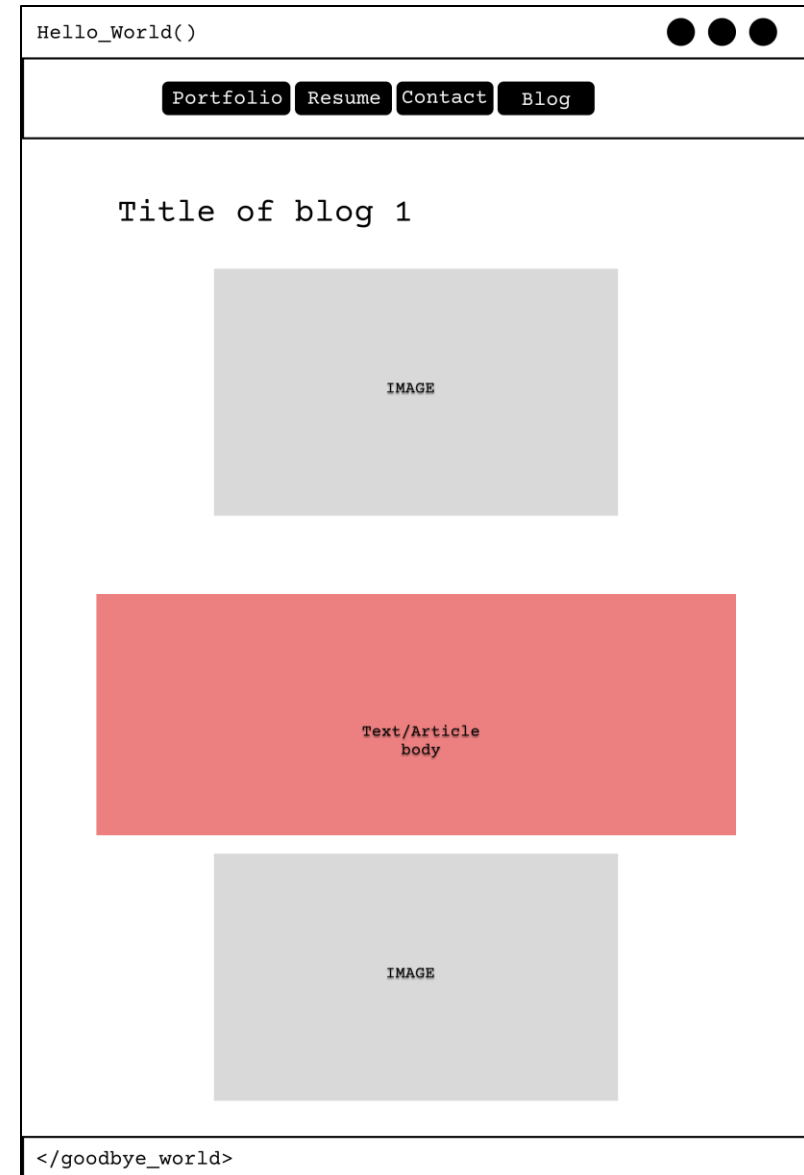
# Wireframes – bloglist.html

- Design for the blog index was minimal
- My initial idea was to lay blog thumbnails out in a grid however the long-list option seemed better particularly for the mobile screen scroll.
- A 'back to top' button included in case the list of blogs gets too extensive

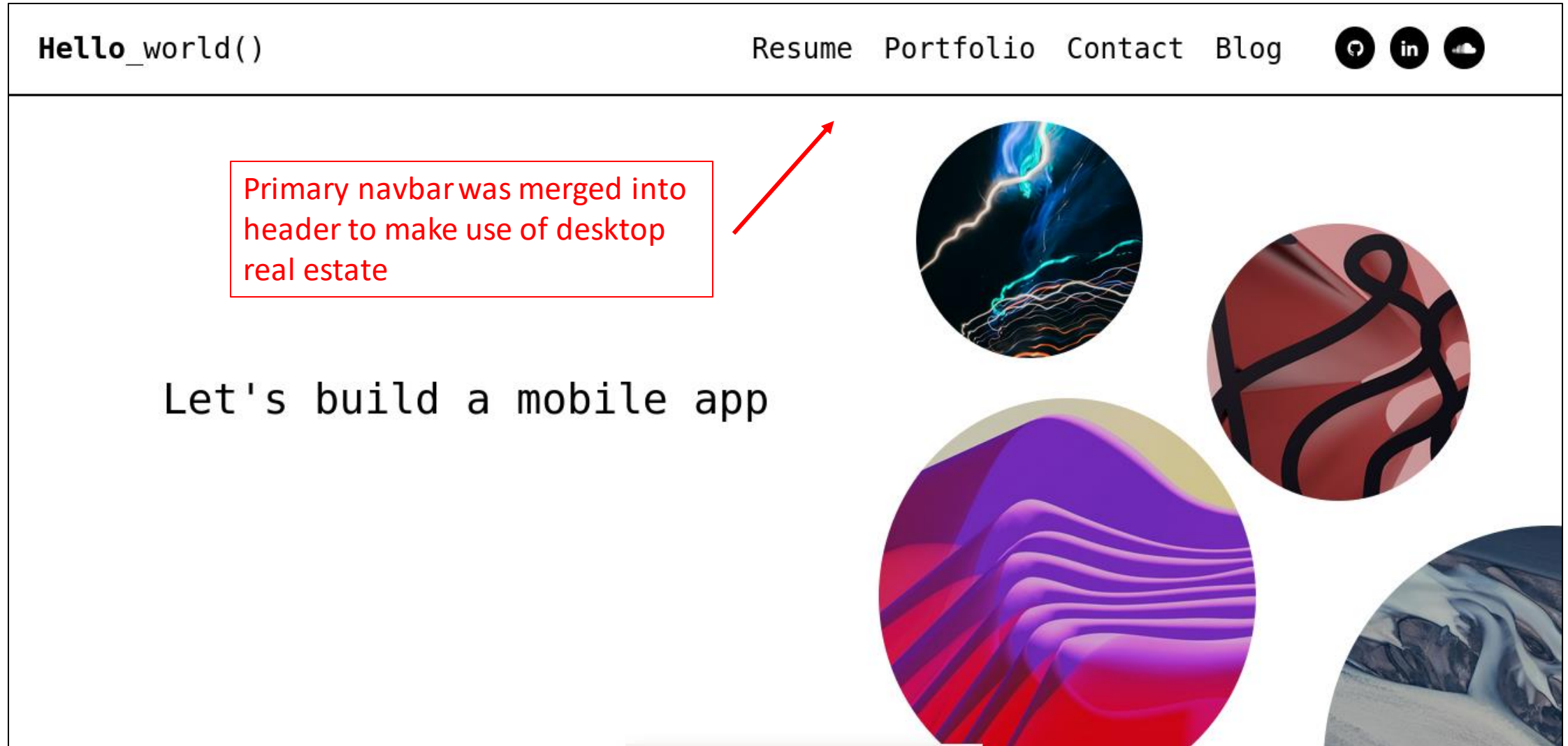


# Wireframes – blog1.html (2, 3, 4, 5)

- Blog entry followed a news article layout
- All 5 pages have same placeholder layout however will likely change depending on the blog post



# Changes to navbar layout for desktop screens





# How layout boosts UX/engagement

To me personal websites are like paper resumes: the less pages the better

- I decided to put my Bio, Skills, projects and contact info on the index.html as I thought scrolling (especially on mobile) is quicker and easier for the user to navigate these crucial bits of information.
- Another reason to put these elements on index.html is that it (hopefully) showcases an ability to build 'long' responsive sites.



# How style enforces a (relevant) theme

Monospaced font was used throughout to reflect code/programming-style text

- drawbacks: singular fonts for both text and headings can seem boring
- Hero image not dominating nor 'limiting', stylistically

Hello\_World() ≡

Let's build Things  
↓  
a website  
a web app  
a mobile app

About me

Bio section here

My Stack



Hello\_world()



Resume

Portfolio

Contact

Blog

About me

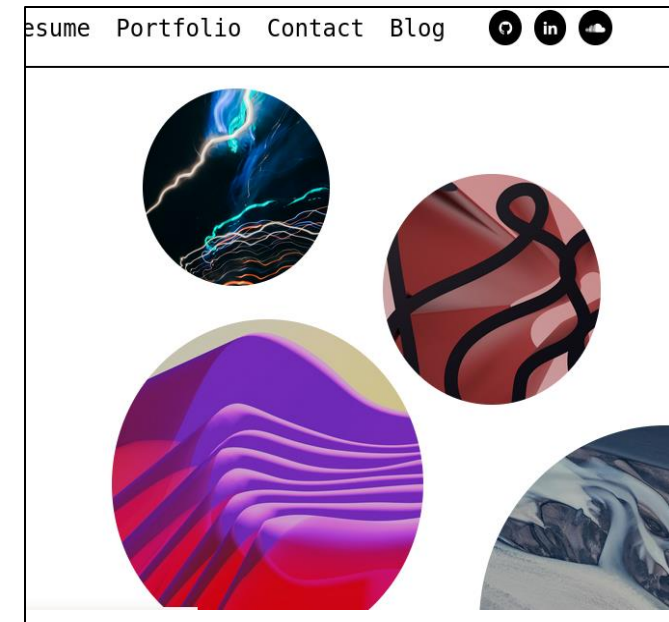
Hi. My name is Sean Gyuris.

I am a junior full-stack developer who [will] currently builds with HTML/CSS, ReactJS and Flask.

My past projects [will] feature a range of functionalities not limited to: profile creation, payments integration (Stripe), paywalls and media players.

I [will] take pride in writing clean code using an agile approach both collaboratively or solo.

My Stack



# How components and animations boost engagement & showcase unique UX/UI

- Text reel on hero image:
  - Demonstrates an ability to build a animation type featured on many modern websites

Let's build Things  
↓  
a website  
a web app  
a mobile app

- Blinking cursor
  - Demonstrates an ability to build a unique animation styling (to attract potential future clients) – also was used to liken these blogs as 'journal' entries

**Blog**  
**Blog |**  
**Blog**

# Roadblocks & work-throughs

- **Relative linking WITHIN the page:** it seems Mozilla does not recognise internal links without anchor tags or id="" descriptors. So to get cross-browser linking functionality, I used id="", and added "a" tags within divs that had internal relative links.
- **displaying .svg images in Chrome was problematic:** SOME of my svg. Images [screenshots] set within an 'img' tag failed to display in Chrome upon deployment so I was forced to either use png or an .svg as a div background image. A potential work-around that I decided not to use can be found here: <https://stackoverflow.com/questions/10737166/chrome-not-rendering-svg-referenced-via-img-element>

# Roadblocks & work-throughs

## •The Hamburger Nav dropdown using only CSS:

To build a drop-down menu using only vanilla CSS I:

- 1.used the 'input' element with a 'checkbox' attribute to make the 'toggle button'.
- 2.Since checkbox has a 'checked' pseudo element, I simply made a @media query that stated: "if checked, display a dropdown menu (<mobileNav/>). If unchecked, then mobileNav{ display:none}.
- 3.All that needed to be done to substitute the checkbox with a hamburger menu icon which I did by putting a html 'Label' over the checkbox (<label for="checkbox" class="hamburgerIcon" >),then set the box's visibility to none.

## HTML

```
<input type="checkbox" id="hamburger-nav" class="hamburger-nav">
```

```
<mobileNav>
  <ul class="Navlist">
    <li><a href="cv.html">Resume</a></li>
    <li><a href="Index.html#Portfolio">Portfolio</a></li>
    <li><a href="Index.html#contact">Contact</a></li>
    <li><a href="bloglist.html">Blog</a></li>
    <icons class="hamburgerSocials">
      <a class="fa fa-github" href="https://github.com/seanlb02" target= "_blank"></a>
      <a class="fa fa-linkedin" href="https://www.linkedin.com/in/sean-gyuris-72b5a1127/"></a>
      <a class="fa fa-soundcloud" href="https://soundcloud.com/cult-trip-records"target=
    </icons>
  </ul>
</mobileNav>
```

```
<label for="hamburger-nav" class="hamburgerIcon">
  <i class="fa fa-solid fa-bars fa-lg"></i>
</label>
```

## CSS

```
.hamburger-nav{
  display:none;
}
.hamburger-nav:checked ~ mobileNav {
  display:block;
  margin-left: 4%;
  margin-right: 5%;
}
@media screen and (min-width: 600px){
  .hamburger-nav, .hamburgerIcon {
    display: none;
  }
}
```

# Thank you

REPO: <https://github.com/seanlb02/CA-portfolio>

Live demo: <https://thriving-treacle-8b6b78.netlify.app/>