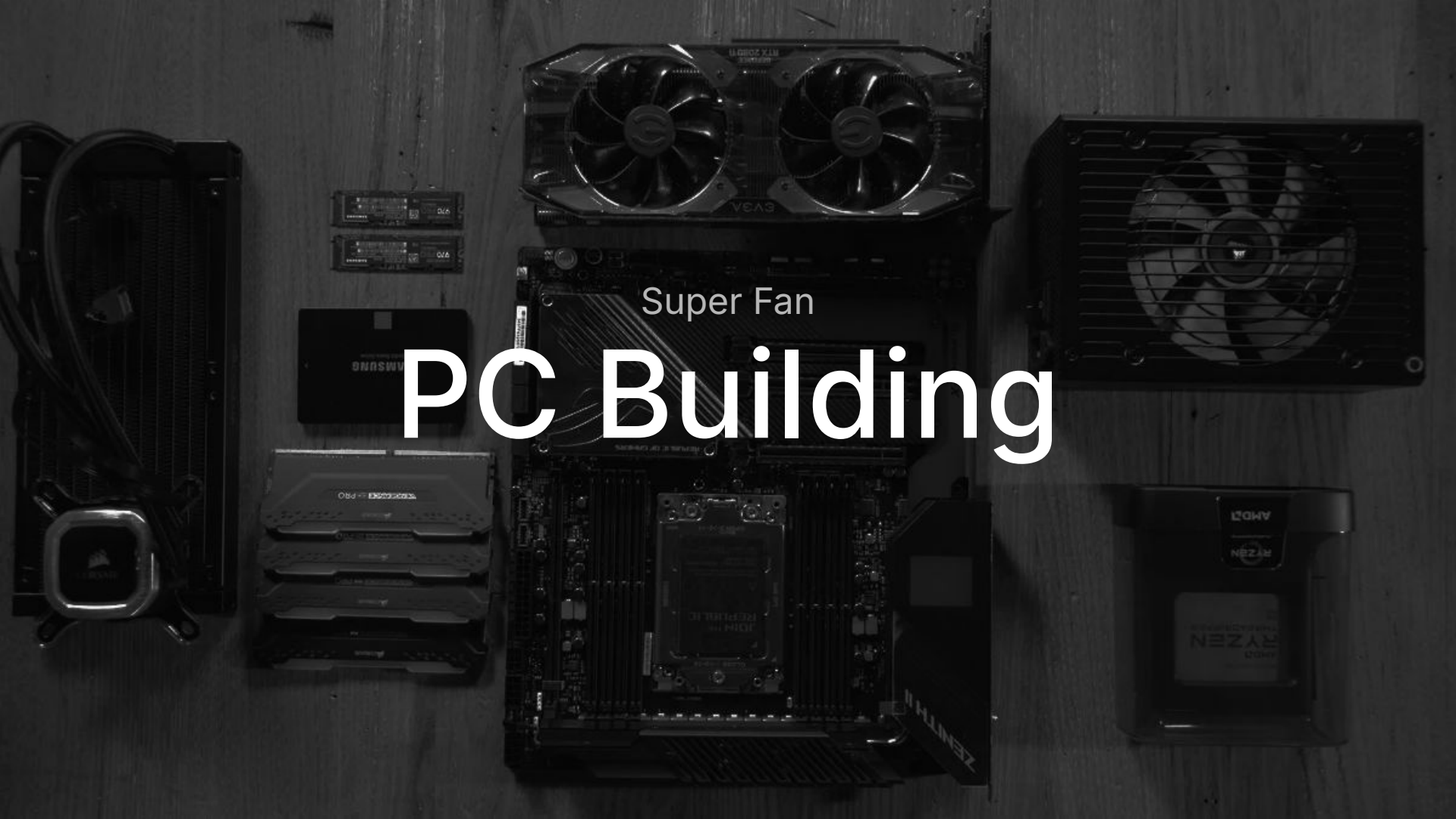
Computer Science POL

Justin Tran


Super Fan
PC Building

What is this?

```
function loadPrompt(part) {
  AvailableBudget += PartList.prices[part];
  PartList.prices[part] = 0;
  PromptTitle.textContent = part.charAt(0).toUpperCase() + part.slice(1);
  for (let individualpart in PartDatabase[part]) {
    let Part = PartDatabase[part][individualpart];

    var NewPart = partTemplate.cloneNode(true);
    NewPart.setAttribute("class", "individual-part primary");
    PromptPartContainer.appendChild(NewPart);

    NewPart.querySelector(".part-info h1").textContent = Part.name;
    NewPart.querySelector(".part-info p").textContent = Part.desc;
    NewPart.querySelector(".part-image").src = Part.path;
    NewPart.querySelector(".part-select").textContent = "$" + Part.price;

    const priceBtn = NewPart.querySelector(".part-select");
    priceBtn.addEventListener("click", () => {
      selectPart(part, Part);
    });

    // Loop limits
    for (let limitId in Part.limits) {
      var limit = Part.limits[limitId];
      var CategoryLimit = NewPart.querySelector(
        ".part-individual-category"
      ).cloneNode(true);
      CategoryLimit.querySelector("h1").textContent = limit.category;
      CategoryLimit.querySelector("p").textContent = limit.limit;
      CategoryLimit.classList.remove("hidden");
      NewPart.querySelector(".part-s-categories").appendChild(CategoryLimit);

      var MobileCategory = NewPart.querySelector(
        ".part-mobile-individual-category"
      ).cloneNode(true);

      MobileCategory.querySelector("h1").textContent = limit.category;
      MobileCategory.querySelector("p").textContent = limit.limit;
      MobileCategory.classList.remove("hidden");

      NewPart.querySelector(".part-row2 .part-s-categories").appendChild(
        MobileCategory
      );

      checkCompatibility(limit, NewPart);
    }

    if (Part.price > AvailableBudget) {
      NewPart.querySelector(".incompatible-overlay").classList.remove("hidden");
      NewPart.querySelector(".incompatible-overlay p").textContent =
        "Out of Budget";
      NewPart.querySelector(".part-select").disabled = true;
    }
  }
  PromptOverlay.classList.add("show");
  PromptContent.classList.remove("hidden");
  ResultsContent.classList.add("hidden");
  attachButtonSounds();
}
```

```
function loadPrompt(part) {
  AvailableBudget += PartList.prices[part];
  PromptTitle.textContent = part.charAt(0).toUpperCase() + part.slice(1);
  for (let individualpart in PartDatabase[part]) {
    let Part = PartDatabase[part][individualpart];

    var NewPart = partTemplate.cloneNode(true);
    NewPart.setAttribute("class", "individual-part primary");
    PromptPartContainer.appendChild(NewPart);

    NewPart.querySelector(".part-info h1").textContent = Part.name;
    NewPart.querySelector(".part-info p").textContent = Part.desc;
    NewPart.querySelector(".part-image").src = Part.path;
    NewPart.querySelector(".part-select").textContent = "$" + Part.price;

    const priceBtn = NewPart.querySelector(".part-select");
    priceBtn.addEventListener("click", () => {
      selectPart(part, Part);
    });

    // Loop limits
    for (let limitId in Part.limits) {
      var limit = Part.limits[limitId];
      var CategoryLimit = NewPart.querySelector(
        ".part-individual-category"
      ).cloneNode(true);
      CategoryLimit.querySelector("h1").textContent = limit.category;
      CategoryLimit.querySelector("p").textContent = limit.limit;
      CategoryLimit.classList.remove("hidden");
      NewPart.querySelector(".part-s-categories").appendChild(CategoryLimit);

      var MobileCategory = NewPart.querySelector(
        ".part-mobile-individual-category"
      ).cloneNode(true);

      MobileCategory.querySelector("h1").textContent = limit.category;
      MobileCategory.querySelector("p").textContent = limit.limit;
      MobileCategory.classList.remove("hidden");

      NewPart.querySelector(".part-row2 .part-s-categories").appendChild(
        MobileCategory
      );

      checkCompatibility(limit, NewPart);
    }

    if (Part.price > AvailableBudget) {
      NewPart.querySelector(".incompatible-overlay").classList.remove("hidden");
      NewPart.querySelector(".incompatible-overlay p").textContent =
        "Out of Budget";
      NewPart.querySelector(".part-select").disabled = true;
    }
  }
  PromptOverlay.classList.add("show");
  PromptContent.classList.remove("hidden");
  ResultsContent.classList.add("hidden");
  attachButtonSounds();
}
```

```
AvailableBudget += PartList.prices[part];
PromptTitle.textContent = part.charAt(0).toUpperCase() + part.slice(1);
```

Returns back the money if a part was selected and changes the title of the prompt

This is a loop

```
for (let individualpart in PartDatabase[part]) {
  let Part = PartDatabase[part][individualpart];

  var NewPart = partTemplate.cloneNode(true);
  NewPart.setAttribute("class", "individual-part primary");
  PromptPartContainer.appendChild(NewPart);

  NewPart.querySelector(".part-info h1").textContent = Part.name;
  NewPart.querySelector(".part-info p").textContent = Part.desc;
  NewPart.querySelector(".part-image").src = Part.path;
  NewPart.querySelector(".part-select").textContent = "$" + Part.price;

  const priceBtn = NewPart.querySelector(".part-select");
  priceBtn.addEventListener("click", () => {
    selectPart(part, Part);
  });

  // Loop limits
  for (let limitId in Part.limits) {
    var limit = Part.limits[limitId];
    var CategoryLimit = NewPart.querySelector(
      ".part-individual-category"
    ).cloneNode(true);
    CategoryLimit.querySelector("h1").textContent = limit.category;
    CategoryLimit.querySelector("p").textContent = limit.limit;
    CategoryLimit.classList.remove("hidden");
    NewPart.querySelector(".part-s-categories").appendChild(CategoryLimit);

    var MobileCategory = NewPart.querySelector(
      ".part-mobile-individual-category"
    ).cloneNode(true);

    MobileCategory.querySelector("h1").textContent = limit.category;
    MobileCategory.querySelector("p").textContent = limit.limit;
    MobileCategory.classList.remove("hidden");

    NewPart.querySelector(".part-row2 .part-s-categories").appendChild(
      MobileCategory
    );

    checkCompatibility(limit, NewPart);
  }

  if (Part.price > AvailableBudget) {
    NewPart.querySelector(".incompatible-overlay").classList.remove("hidden");
    NewPart.querySelector(".incompatible-overlay p").textContent =
      "Out of Budget";
    NewPart.querySelector(".part-select").disabled = true;
  }
}
```

This selects the current individual part being looped through in the part database

This creates a clone of the template and attaches it to the prompt

Sets the information for each part

```
"1": {
  "name": "Montech XR",
  "desc": "Mid-Tower, Steel Frame, Tempered Glass Side Panels",
  "price": 79.99,
  "path": "images/parts/cases/Montech_XR.png",
  "limits": {
```

Detects when the buy button is pressed and runs a function

```
for (let individualpart in PartDatabase[part]) {
  let Part = PartDatabase[part][individualpart];

  var NewPart = partTemplate.cloneNode(true);
  NewPart.setAttribute("class", "individual-part primary");
  PromptPartContainer.appendChild(NewPart);

  NewPart.querySelector(".part-info h1").textContent = Part.name;
  NewPart.querySelector(".part-info p").textContent = Part.desc;
  NewPart.querySelector(".part-image").src = Part.path;
  NewPart.querySelector(".part-select").textContent = "$" + Part.price;

  const priceBtn = NewPart.querySelector(".part-select");
  priceBtn.addEventListener("click", () => {
    selectPart(part, Part);
  });

  // Loop limits
  for (let limitId in Part.limits) {
    var limit = Part.limits[limitId];
    var CategoryLimit = NewPart.querySelector(
      ".part-individual-category"
    ).cloneNode(true);
    CategoryLimit.querySelector("h1").textContent = limit.category;
    CategoryLimit.querySelector("p").textContent = limit.limit;
    CategoryLimit.classList.remove("hidden");
    NewPart.querySelector(".part-s-categories").appendChild(CategoryLimit);

    var MobileCategory = NewPart.querySelector(
      ".part-mobile-individual-category"
    ).cloneNode(true);

    MobileCategory.querySelector("h1").textContent = limit.category;
    MobileCategory.querySelector("p").textContent = limit.limit;
    MobileCategory.classList.remove("hidden");

    NewPart.querySelector(".part-row2 .part-s-categories").appendChild(
      MobileCategory
    );

    checkCompatibility(limit, NewPart);
  }

  if (Part.price > AvailableBudget) {
    NewPart.querySelector(".incompatible-overlay").classList.remove("hidden");
    NewPart.querySelector(".incompatible-overlay p").textContent =
      "Out of Budget";
    NewPart.querySelector(".part-select").disabled = true;
  }
}
```

Not all parts are compatible with each other!

Looping through limits this time

Creates clone of limit

Specifications

Motherboard Size	Socket
ATX	AM5

Sets category and limit

```
"limits": {
  "1": {
    "category": "Case Size",
    "limit": "ATX"
  },
  "2": {
    "category": "GPU Size Limit",
    "limit": "420mm"
  }
}
```

Makes it visible and adds it to the category div

Runs a compatibility function which checks if part is within the build parameters

This is essentially the same process except it is for the mobile and tablet layout

Conditional checks if part goes over budget

```
if (Part.price > AvailableBudget) {
  NewPart.querySelector(".incompatible-overlay").classList.remove("hidden");
  NewPart.querySelector(".incompatible-overlay p").textContent =
    "Out of Budget";
  NewPart.querySelector(".part-select").disabled = true;
}
```

If it is, it will make the incompatible layer appear on top

Change the text inside the text to express the reason why

Disable the button so it can't be selected

```
PromptOverlay.classList.add("show");
PromptContent.classList.remove("hidden");
ResultsContent.classList.add("hidden");
attachButtonSounds();
```

Makes the prompt overlay show up on the screen

Makes it so the prompt content itself is not hiding

Adds sounds to the buttons after they have been cloned

Hide the results content incase a user wants to make changes after seeing results of their build

Digital Portfolio

Justin Tran

Home | Computer Science | About Me

SuperFan Interactive Experience

The Challenge

Create an interactive, intuitive, and responsive front-end website that centered around a topic I am super passionate about.

Roles

Graphic Designer

User Interface Developer

User Experience Designer

Tools Used

JavaScript / CSS / HTML

CodeSandbox / GitHub Pages

Google Slides / Google Sheets

Brainstorm & Evaluate

Before making a decision about my topic I spent some time generating ideas and considering how they would meet the project requirements. After quickly weighing the pros/cons of each option I made a decision.

Idea 1: Apple Products

Theme

Idea 2: PC Building

Theme

Idea 3: KPop Demon Hunters

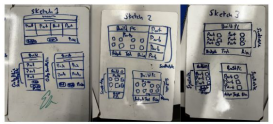
Theme

Evaluate + Choose Your Idea PC Building

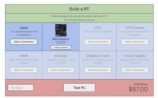
Digital Portfolio

Responsive Design & Digital Mockup

After making multiple sketches considering possible layouts I consulted with my teacher and we chose a design. Before coding I wanted to make sure everything looked professional so I used Google Slides to make a mockup.



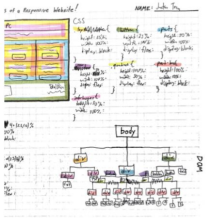
Home Page Sketch

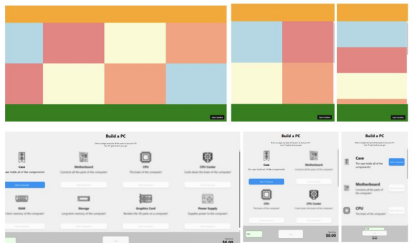


Home Page Sketch

Plan & Incrementally Build

Before coding I sat down and carefully planned out my site considering the structure of the page and the relative measurements + display properties. After I had my plan I built-up my layout step-by-step.

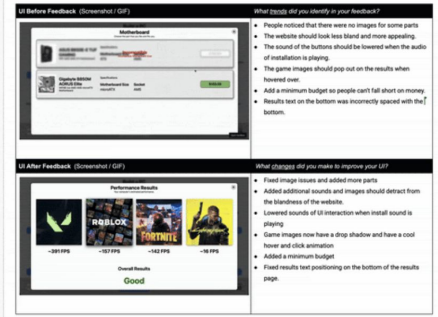




User Experience (UX) Testing & User Interface (UI) Iteration

Once my first iteration was complete I conducted user testing with peers from my class. After hearing from 5 users I identified trends in my feedback and made improvements to my site to enhance the page's appearance, functionality, and intuitiveness.

	User Interface: *How could the styling of elements be changed to make a more visually appealing, clear, and space efficient layout?* (ex: sizes, colors, fonts, alignment, etc.)	**User Experience:** *What elements of the user experience could be improved to make the page more intuitive and more fun/satisfying?* (ex: sounds, hover effects, press effects, etc.)
User 1 Name: *Romeo Pearson*	• Romeo says that there are no images for some parts. He says that it would be better if they worked. • He also says that the text at the top is pretty small and that the budget textbox is pretty small so it should be better. • Romeo wants to see a completed computer and a snippet of the game preview.	• He says that it feels very interactive. • When the part sounds like it's being installed, he says that the audio of the buttons should be turned down so you can hear the parts being installed. • He also wants the games on the results page to be clickable.
User 2 Name: *Nathaniel Castillo*	• Nate wants some better graphics so that it doesn't look so bland because of the white background. • Make the text have some color and be less bland. • The website right now looks a bit bland so consider adding more elements.	• Nate wants more information on the results page for the computer. • He also says that there should be an animation on the total price when you buy something. • Make it so the budget has a minimum so that people don't fall short on money when buying.
User 3 Name: *Julien Villery*	• Julien saw that some images are missing and should be added. • He wants the total price to be moved over a bit so you can see the price of the build. • The bottom text should be moved up a bit.	• When you hover the image, the image should pop out on the results page.
User 4 Name: *Desmond Sparks*	• Some of the images didn't work like for some of the parts. • The bottom part of the results text should be a bit higher since it's a bit close to the bottom.	• Desmond didn't really know that he needed to put in a budget before he started.
User 5 Name: *Anthony Trejo*	• He likes that everything is well sized but said that the pictures need to be worked on. • He said that the background on the parts should be removed. • Make the website less bland by adding more color into it.	• The budget should be more clear and there should be like a minimum so people don't run out of money midway through.



GitHub Repository

Here is a link to my final project!

GitHub

Reflection

During this project I was able to practice my responsive design skills and also extended my knowledge of JavaScript. If I had more time I would explore the power of JavaScript to make the page more interactive and conduct a second round of UX Testing to improve my work.

Contact:
justin_tran@student.davincischools.org

Thank you

Questions?