

# ✨ MiracleDex Project Report

**Student Name:** Martina Licheri

**Course:** CSCI 270 – Web/Mobile App Development

**Project Title:** MiracleDex

**Submission Date:** Monday, June 9th

**GitHub Repo:** <https://github.com/martinalicheri/Project1-CSCI270>

**Live Site (GitHub Pages):** <https://martinalicheri.github.io/>

## ◆ Part 1 — HTML/CSS Layout and Responsive Design

### ✓ Requirements Addressed

- Landing page created with heading and container
- Used Flexbox or Grid layout
- Mobile-friendly design with media queries
- Elegant, reverent styling with a Google Font
- External CSS file used

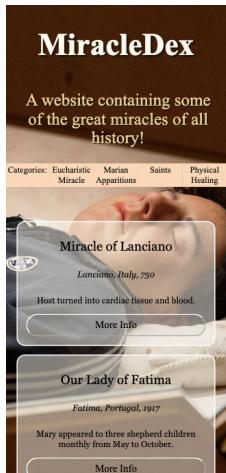
### 📸 Screenshots

Paste 1–2 screenshots showing your site on desktop and mobile.

**Computer:**



**Phone:**



## 🔍 Code Snippets

Include relevant HTML and CSS snippets here (layout structure, responsive styling, etc.).

### HTML:

```
<!-- This is to make the categories' menu -->
<div class="grid">
  <div class="categories">Categories:</div>
  <div class="categories">Eucharistic Miracle</div>
  <div class="categories">Marian Apparitions</div>
  <div class="categories">Saints</div>
  <div class="categories">Physical Healing</div>
</div>

<!-- This is where I put the fetched data -->
<div id="cards"></div>
```

### CSS:

```
/* This is how I adjusted the grid */
```

```
#cards{
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-column-gap: 1rem;
  grid-row-gap: 1rem;
  justify-self: center;
  padding-top: 4rem;
  width: 90%;
  justify-content: center ;
}
```

```
/* And this is the media query */
```

```
@media (max-width: 600px) {
  #cards{
    grid-template-columns: 1fr;
  }
  .grid{
    padding-top: 0;
  }
  #page{
    background-position: 45%;
  }
  .title-subtitle{
    padding-bottom: 1vw;
  }
}
```

### ✍ Reflection

- What challenges did you face in designing a responsive layout?

One challenge I faced was in adjusting the background picture. If it was a smaller screen, then the only visible part of the picture was his arm. Centering the image wouldn't do it either. So I had to work it out and figure out how to adjust the picture depending on the screen size.

- What did you learn about structuring HTML/CSS for real-world use?

It has much more details than I expected. Perhaps there's higher functions and styling options I am yet unaware of, but at the same time, looking at the code now (once I've finished), I realize it is fairly little code to have such a web!

It gave me more practices with grids and dynamic elements. I realized that CSS can do more than I thought. Yes, it styles, but the style is very important!

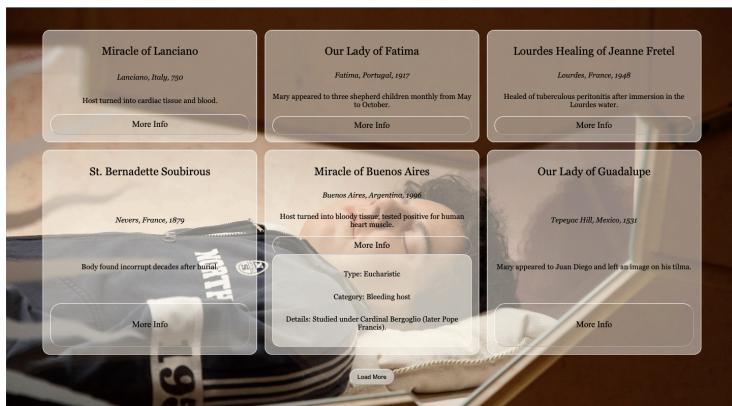
## ◆ Part 2 — JavaScript + DOM + JSON Integration

### ✓ Requirements Addressed

- Fetched data from:  
<https://gist.githubusercontent.com/trevortomesh/7bbf97b2fb96639ebf1a254b6a7a70/raw/miracles.json>
- Rendered miracle title, location, year, summary
- Used `fetch()` and `async/await`
- Implemented “Load More” or pagination
- Added modal or expandable section with full miracle details

## 📸 Screenshots

Show your miracle cards and one expanded view or modal.



## 🔍 Code Snippets

Include the function you used to fetch the data, render the cards, and handle interaction.

```
async function loadMiracles() {  
  
  // fetch data  
  const response = await fetch("https://gist.githubusercontent.com/trevortomesh/7bbf97b2fbae96639ebf1a254b6a7a70/raw/miracles.json");  
  if (!response.ok) {  
    throw new Error("HTTP error! Status: " + response.status);  
  }  
  const data = await response.json();  
  const dataCards = document.getElementById("cards");  
  
  // render cards  
  data.forEach(card => {  
    const div = document.createElement("div");  
    const h1 = document.createElement("h1");  
    const h2 = document.createElement("h2");  
    const p = document.createElement("p");  
    const button = document.createElement("button");  
    h1.textContent = `${card.title}`;  
    h2.textContent = `${card.location}, ${card.year}`;  
    p.textContent = `${card.summary}`;  
    button.textContent = 'More Info';  
    div.appendChild(h1);  
    div.appendChild(h2);  
    div.appendChild(p);  
    div.appendChild(button);  
    // (...)  
  
    // Show or hide modal information - handle interaction  
    button.onclick = () => {  
      if (div_info.style.display === "none") {  
        div_info.style.display = "grid";  
      } else {  
        div_info.style.display = "none";  
      }  
    }  
  })  
  
  // handle interaction  
  const loadMoreButton = document.getElementById("load-more");  
  loadMoreButton.addEventListener("click", async () => {  
    cardAmount += 6;  
    if (cardAmount > totalCards) {  
      loadMoreButton.style.display = "none"; // hide button when no more cards to load  
    }  
    for (; count <= cardAmount; count++) {  
      let div = document.getElementById(`${count}`);  
      div.style.display = "grid";  
    }  
  })  
}  
  
loadMiracles();
```

## ✍️ Reflection

- What did you learn about asynchronous JavaScript? What debugging techniques did you use or discover?

It helps you run other things while you are executing a certain task/function.

I noticed it in that the web didn't wait for the background image to load to display the initial 6 cards.

---

## ◆ Part 3 — GitHub Repository and Documentation

### ✓ Requirements Addressed

- GitHub repo created and pushed
- GitHub Pages deployed
- README.md contains project description, instructions, and screenshots

### 📎 Links

- **GitHub Repo:** <https://github.com/martinalicheri/Project1-CSCI270>
- **Live GitHub Pages Site:** <https://martinalicheri.github.io/>

### ✍ Reflection

- How did using GitHub affect your development process? What new Git or GitHub skills did you gain?

In all truth, I had barely used GitHub before; I only did the bare minimum for the assignment.

Now that I actually had to make a usable repo and a page using GitHub, I found it very confusing at first, and had to struggle through it, but it started making more sense. In a way, then, I think I gained all the beginner's skills, because I went from knowing nothing to being able to make and use a repo for a personal project!

---

## 🧠 Final Reflection

- *Imagine you're explaining this project to a friend who doesn't code.  
What does your app do? What are you most proud of? What was the hardest part to get working?*

This website dynamically displays cards with information regarding certain miracles of the faith.

What I am most proud of is the first thing one sees when they open the app: the title and background image. It took me quite a while to be able to figure out the position and size!

The hardest part for me was the “Load More” button. I did not know how to make them appear dynamically when all the cards had been fetched already. However, it was very satisfying when I thought of hiding them until the button was clicked!

- *Reflect on both the technical and the spiritual aspects of building a project about Catholic miracles.*

This project, although it seemed very simple, took me a lot of time. I had to wrestle through a lot of concepts in making it, even concepts that I thought I had understood. I have to say, making the web about something religious gave me strength to keep going.

I think this relates a lot, too, to the spiritual life. One might think that he understands how things go, but when the hour to put them into practice comes, then the truth shines. We can't do anything on our own. Thankfully, we were able to access help from the internet and previous lessons. I could not have done it without that external help. Because even if the main idea was there, sometimes a small detail might prevent the whole website from working.

---

Assignment created by Dr. Trevor Tomesh with assistance from ChatGPT.

All source miracle data provided from public JSON:

<https://gist.github.com/trevortomesh/7bbf97b2fbac96639ebf1a254b6a7a70>