Show Tech 3 Ways

Overview.

This project is a website based on some key papers in the theory of computation, internet history and the ever-expanding technology revolution, most recently including AI. As the tech expands, there are new interfaces between tech products and humans. We want to show and enjoy three types. We want to use graphics and text to explore.

Research.

There are a couple of main ideas to explore, so there will be research along side the software engineering. Librarian Lauren Movlai will assist with the research. There are huge advances in sizes of databases and statistics to tell us what is out there.

One is idea to watch is publications. In the age of Large Language Models (LLMs), we can anticipate that published information can be copyright or possibly unclear.

A second idea is to watch jobs --- there are a lot of jobs in the world and now many involve using tech, to some extent. One recent approach is that you are likely to find the job you design yourself, using networking that leads you to humans who want you to do that job for them.

A third idea is to think of the expansion of tech as a cone that keeps widening. Some like to say that the tech revolution developed from the mathematical work of Dedekind in the 1890s. The UMB CS Department belonged to the initial internet which was for sharing academic papers. By about 2006, the very much larger internet could still have all its member sites drawn on one piece of paper. Now we have the the internet of things with many mac addresses on many IPs and many router-enabled wifi nets serving printers, laptops, servers, phones and appliances, as well you and me at home, at work and out at Starbucks etc. and all exploring about various aspects of AI.

Within tech_cone, more theoretically, we look at the infinity of the number of computer programs (that could EVER be written) as compared to the (bigger) infinity of the number of functions that could ever be described, which lines up with the full range of human endeavors. We draw on the work of Cantor and Dedekind, and the definitions and proofs in The Theory of Formal Languages with
Applications by Dan A. Simovici and Richard L. Tenney (World Scientific, 1999).

SO we put these pieces together and we (hopefully graphically and by example) create a fun site to visit and explore. For this project, the three areas are: 1) images of sites that attract many visitors, 2) Dedekind's demonstration of numbers as successors of each other, key to writing a program and 3) explaining how a website works by showing how it is put together and how it operates.

The website is for fun. You build a free tier. You can also provide logins for a more advanced tier.

© 2026 DeBlois, UMB. All rights reserved.

Main Functional Requirements.

As a minimum:
1. Create a website called ShowTech in the https://www.cs.umb.edu/cs410/f26/longprojects area.
2. Show examples for each of the three areas.
3. Allow users to click on an area to advance the display.
4. Use google authenticator to offer a login or stay on as a guest.
5. Include a clicker to demo how you and your team use the site for fun.

Area1: Images of World Sites with Lots of Visitors By Year.

1. Show Images of World Sites with Lots of Visitors by Year.
2. Offer 3 Choices to start: Maybe the pyramids in Egypt, the Eiffel Tower in Paris, France or Trinity Church in Boston (over 70,000 visitors per year).
3. Expand this.

Area2: Dedekind invents the Key Definition for Writing Programs

1. Show Dedekind's definition of the infinity of numbers as successors to the previous one.
2. Offer 3 quotations from the papers of Dedekind, Cantor and others in the 1890s.
3. Expand this.

Area3: Show Minnie Front-End and Back-End

1. Show a photo of the Minnie Setup explained in class Sep 29th.
2. Offer 3 or more additional drawings or comments or photographs of whiteboard or screen. Focus on what surprised you about your explorations. What did you learn about full-stack tech and how it works and how can you show it to someone else? Use the terms requirements, tests and software (that runs and does something that is an error or what was intended) including front-end and back-end if needed. Choose 3 areas from areas listed below (done in class) to start.
3. Expand this.

Areas:

a. Hardware diagram(s) for the setup to use in class (a headless server, so no keyboard, mouse or screen):

Client Laptops (students)
(contain radio receivers for wifi)

Desktop Server (Minnie) running OS Debian (about 13.7)

Ethernet Cable to connect router to server

Router Mikrotik routerOS (about 7.21.3)

PowerSwitch with Surge Protector and Power from Grid

© 2026 DeBlois, UMB. All rights reserved.

b. Connect your laptop to forever410 instead of eudoroam

c. Do tests using IP to access the router and server

d. Login to Minnie, same username, last four of id is password

e. Edit some back-end in views.py (avoid error by making a small edit to start)

f. Use browser to render

g. In class, you accessed minnie using your assigned account and password. You edited a text message, part of the Django BackEnd code. We might get Internal Server error because there was a syntax errors. Tom Mullaly will let us keep minnie in the CS server room on the 95 Network.

© 2026 DeBlois, UMB. All rights reserved.