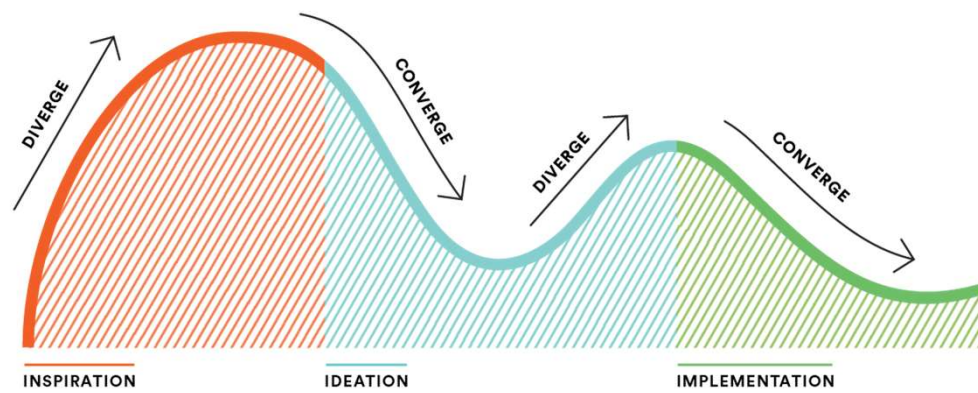


# IN4MATX 133: User Interface Software

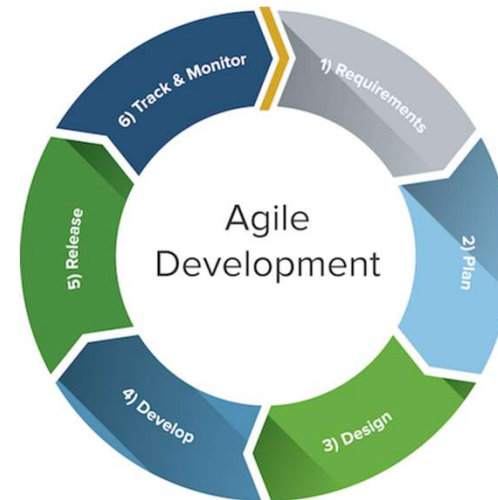
Lecture 1:  
Introduction & History

I am thrilled that you have decided to take this class and that I get it to teach you!

# Product design process

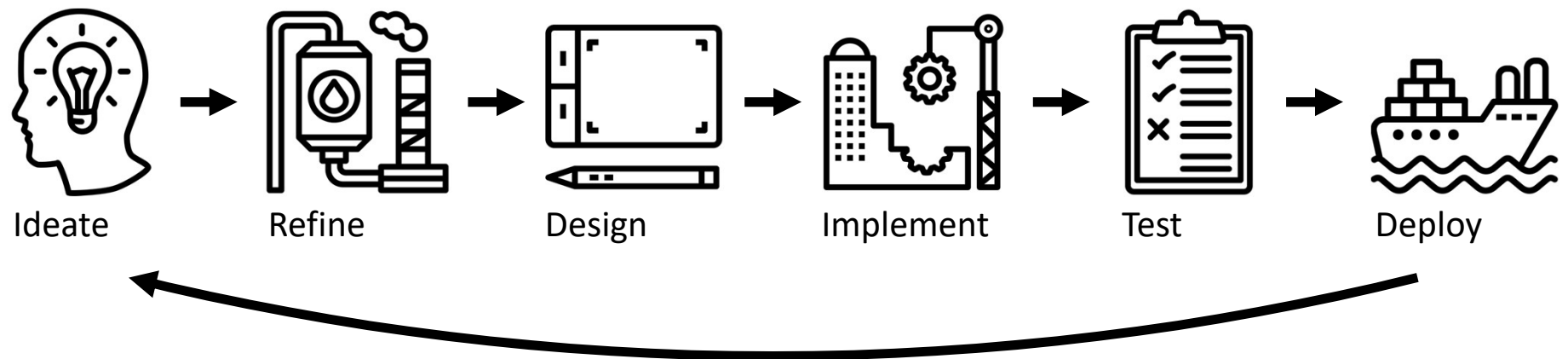


Human-Centered Design, IDEO



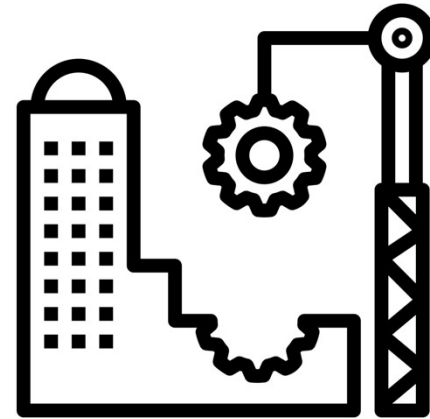
Agile Development, Agile Manifesto

# Product design process, simplified



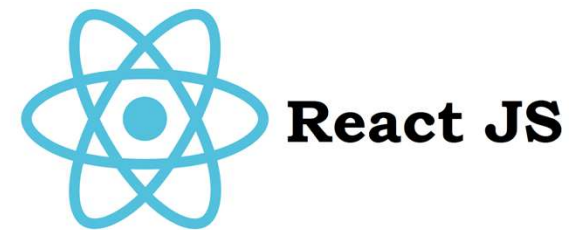
# User interface implementation

- Has the power to turn ideas into reality
- Often dictates design decisions and timelines, for better or for worse
- Either you will be implementing, or you will need to communicate with your colleagues who are



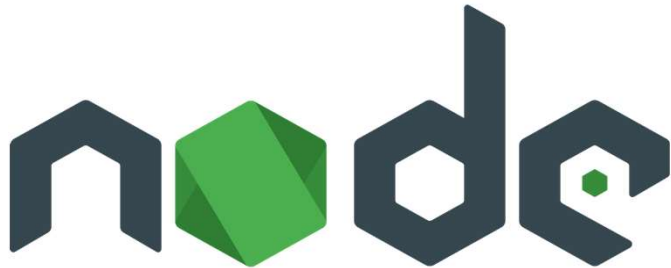
# What is interface implementation today?

Often HTML, CSS, and JavaScript



Bootstrap

METEOR



There are lot of languages  
and development frameworks.

Why do most people use web tools?

# Today's goals

**By the end of today, you should be able to...**

- Describe how society got to today's ubiquitous computing
- Hypothesize why web technology has become the de-facto tool for interface development
- Identify your course staff
- Summarize this course's goals and know how to find policies
- Describe upcoming course tasks



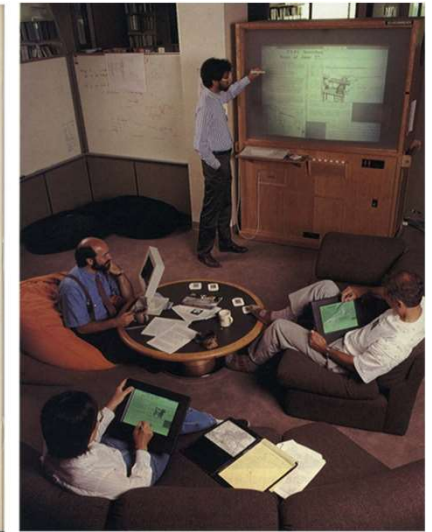
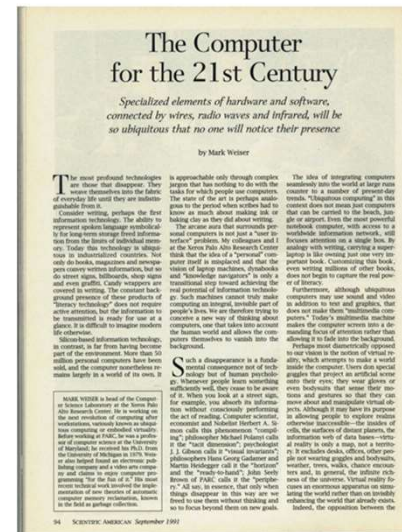
# Today's goals

## **By the end of today, you should be able to...**

- Describe how society got to today's ubiquitous computing
- Hypothesize why web technology has become the de-facto tool for interface development
- Identify your course staff
- Summarize this course's goals and know how to find policies
- Describe upcoming course tasks

# The Computer for the 21st Century

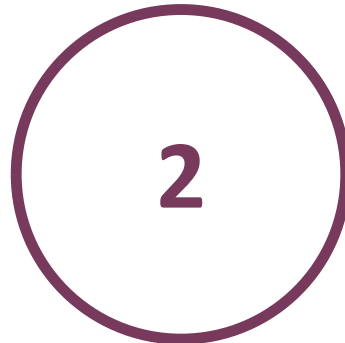
- By Mark Weiser, Chief Scientist, Xerox Parc
- Published in Scientific American, 1991
- Coined “Ubiquitous Computing”
  - Reflective and speculative
- <https://dl.acm.org/citation.cfm?id=329126>



# Three waves of computing



Mainframe  
computing



Personal  
computing



Ubiquitous  
computing

# Three waves of computing



Mainframe  
computing



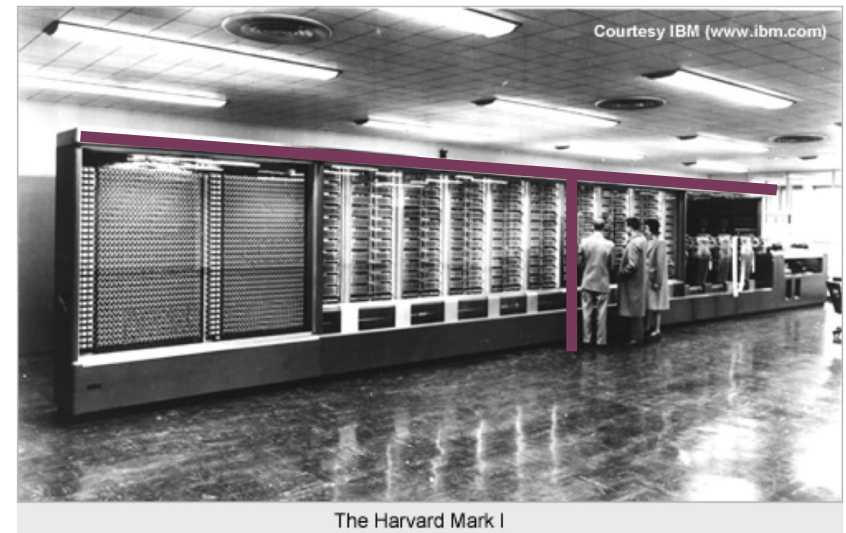
Personal  
computing



Ubiquitous  
computing

# First wave: mainframe computing

- Harvard Mark I
- Large (55 feet wide, 8 feet high, 5 tons)
- Expensive (enclosure alone was \$50,000 in 1945!)
- Used to calculate implosion during the Manhattan Project

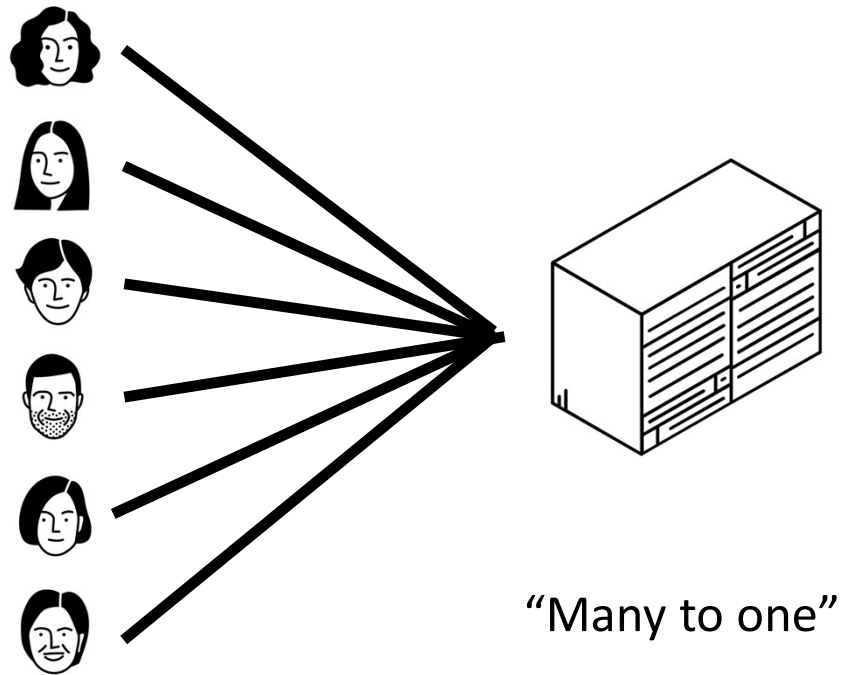


# First wave: mainframe computing

- Batch processing
  - Write your program on punch cards
  - Wait your turn for the computer
  - Run program, hope it works
  - If it doesn't, you'll have to fix it and wait for your next turn
  - Efficient use of resources, but poor interactivity

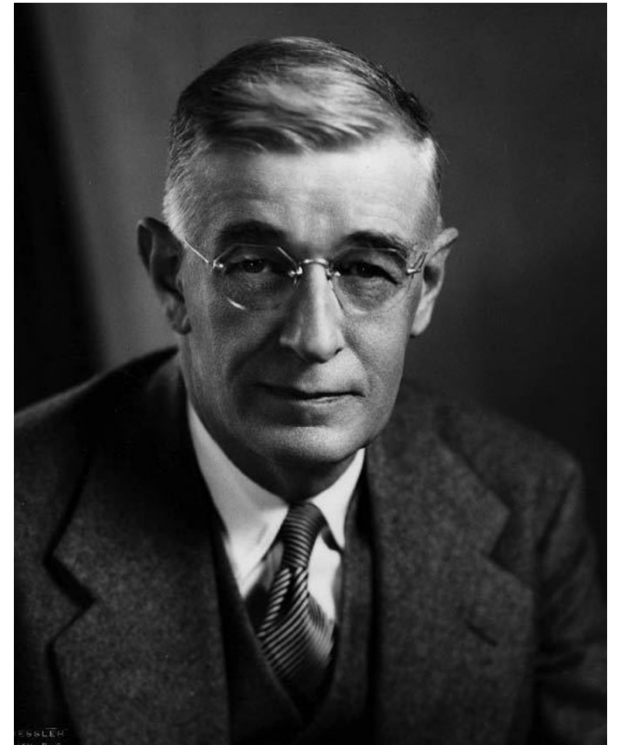


# First wave: mainframe computing



# Vannever Bush

- Faculty at MIT
- Oversaw National Defense Research Committee, which led the Manhattan Project
- Post-war, helped define mission of the National Science Foundation
  - Federal government funds universities
  - Universities do basic scientific research
  - Research helps economy and defense

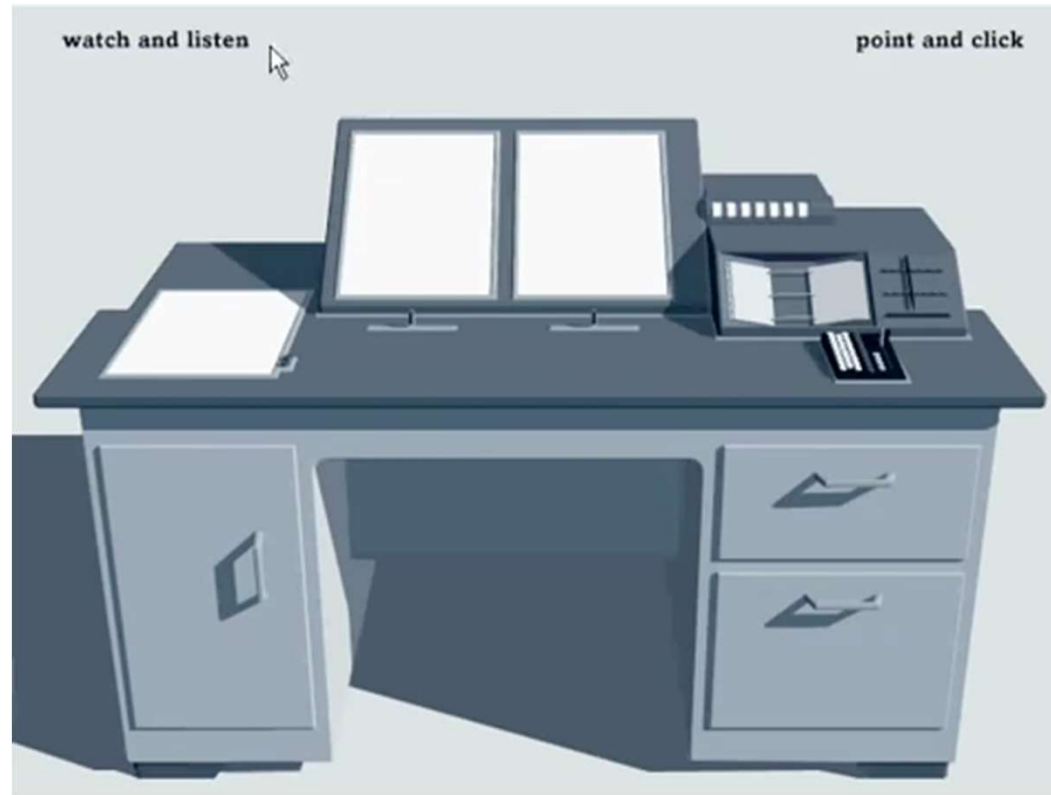




# As We May Think

- Published in Atlantic Monthly, 1945
- <http://www.theatlantic.com/magazine/print/1945/07/as-we-may-think/3881/>
- In part, set out to define a post-war scientific research agenda
  - Speculative, not reflective

# Memex (1945 speculative design)



<https://www.youtube.com/watch?v=c539cK58ees>

(video from 1995 animation presented at SIGIR, not from 1945)

Did you recognize any interactions that are commonly used today?

# Memex (1945 speculative design)

- Linking information across devices and sources
  - Hypertext, the foundation of the web
- Pen-based annotation of primary sources

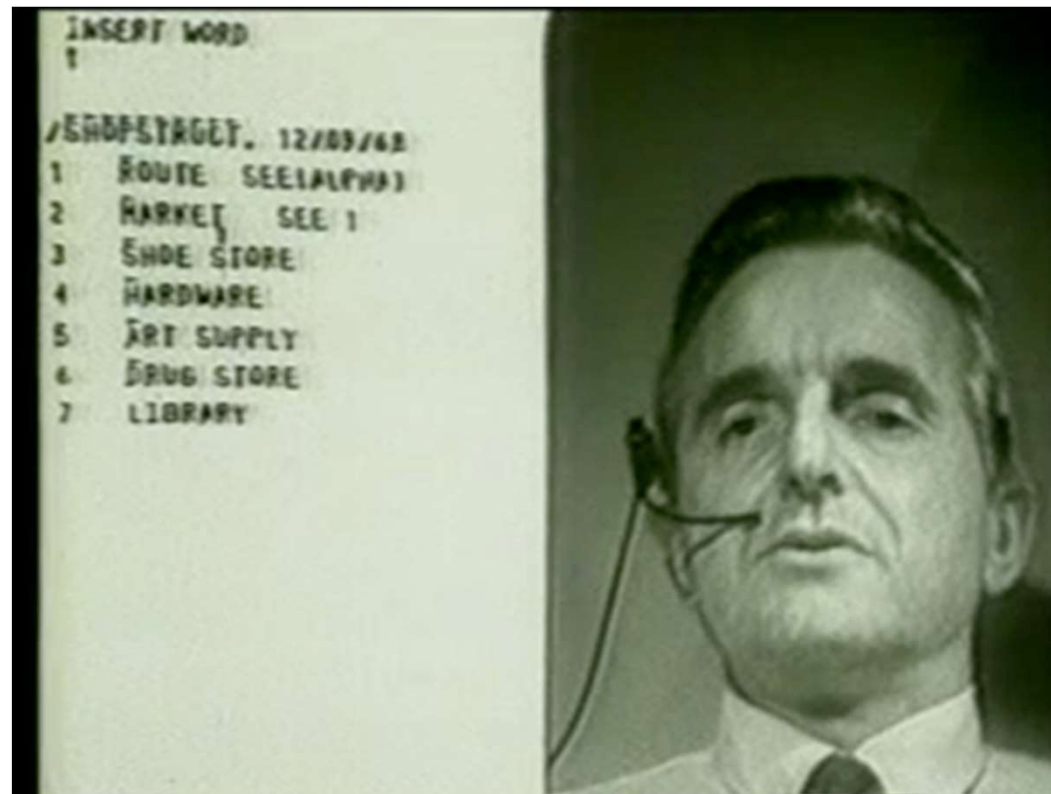


# Command-Line interfaces (1960's)

- Originally used in a terminal connected to a mainframe
  - Was eventually integrated into personal computing (in Unix, etc.)
- A person could change execution based on output
- Enabled real-time debugging



# Doug Engelbart's NLS (1968)

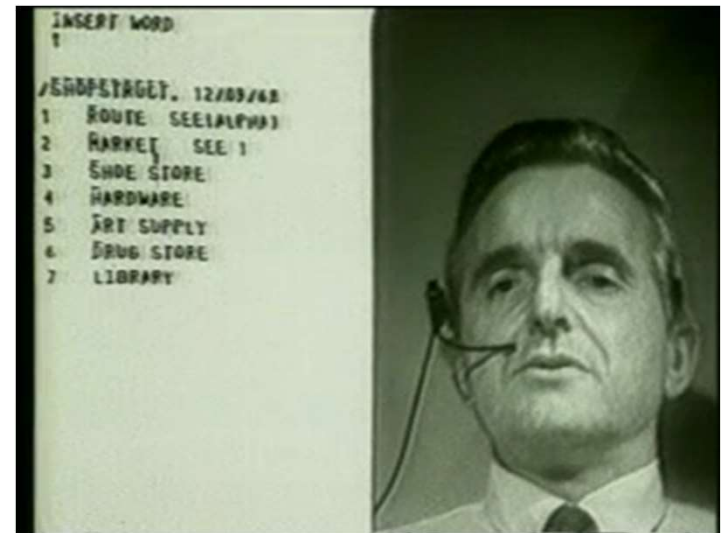


<http://www.dougelbart.org/firsts/1968-demo-interactive.html>

Did you recognize any interactions that are commonly used today?

# Doug Engelbart's NLS (1968)

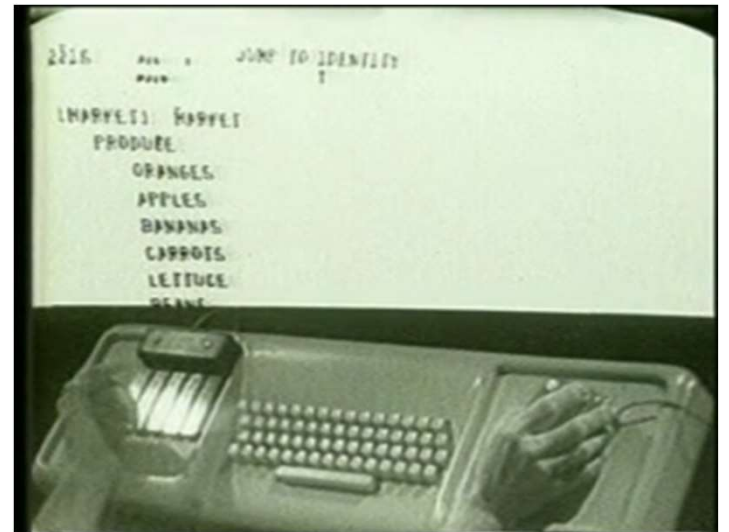
- First working hypertext system
- Invention of the mouse
- Simple graphics  
(earlier systems had this,  
but used in a full system here)





# Doug Engelbart's NLS (1968)

- It introduced other ideas as well
  - A chording keyboard
  - Remote collaboration
- Some people thought he “faked it”
- Others thought it was irrelevant because “the terminal can do the same”
- Engelbart Won Turing Award in 1997



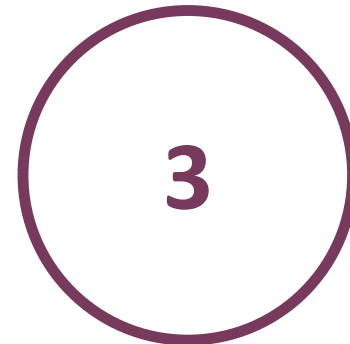
# Three waves of computing



Mainframe  
computing



Personal  
computing



Ubiquitous  
computing

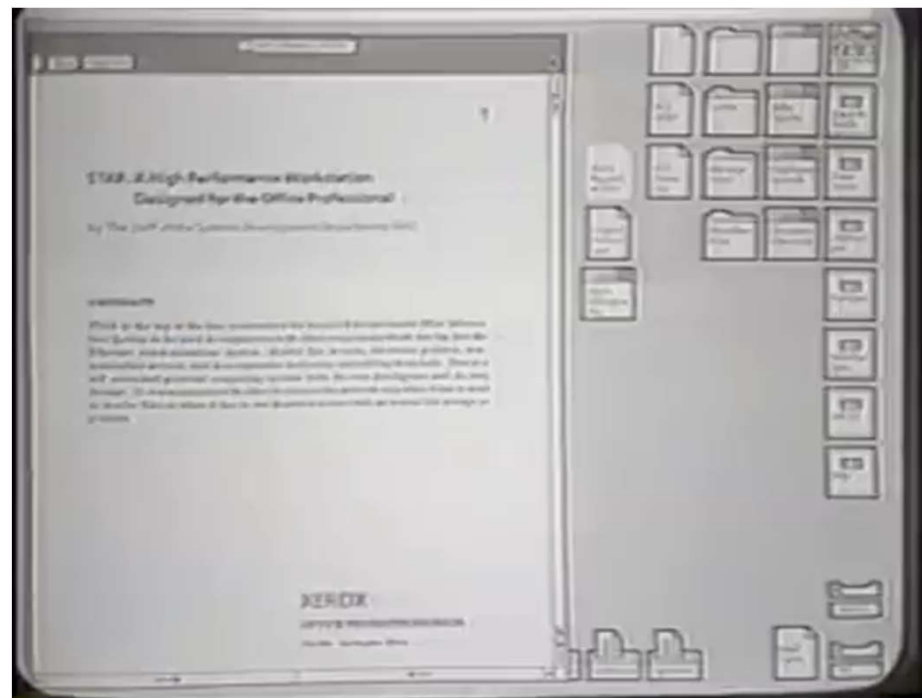
# Second wave: personal computing

- First introduced by Xerox
- Xerox Alto, 1973
  - Mouse
  - Chording keyboard
- Xerox Star, 1981
- Xerox models were commercially unsuccessful
  - Still expensive, too few applications



# Second wave: personal computing

## Xerox Star (1981)



<https://www.youtube.com/watch?v=ODZBL80JPqw>

Did you recognize any interactions that are commonly used today?

# Second wave: personal computing

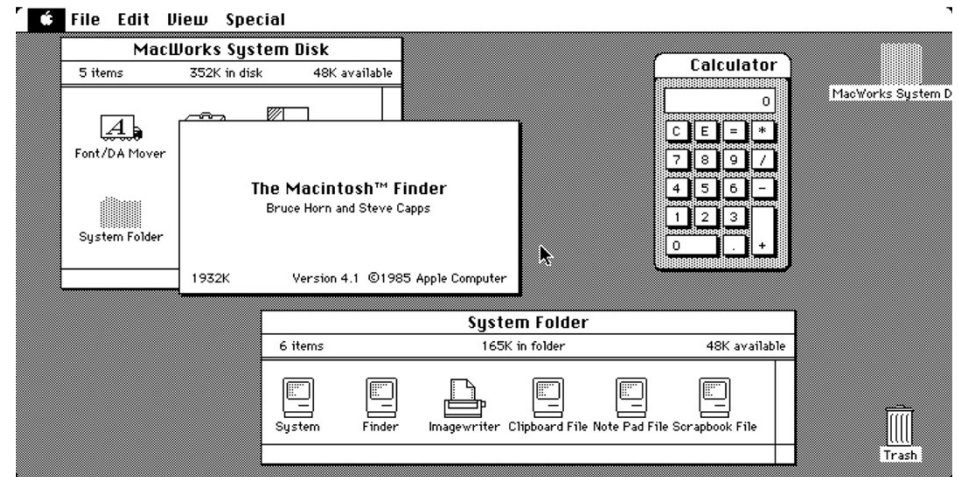
## Xerox Star (1981)

- Software running in windows
- Desktop with icons for navigating between files and programs
- Super slow!



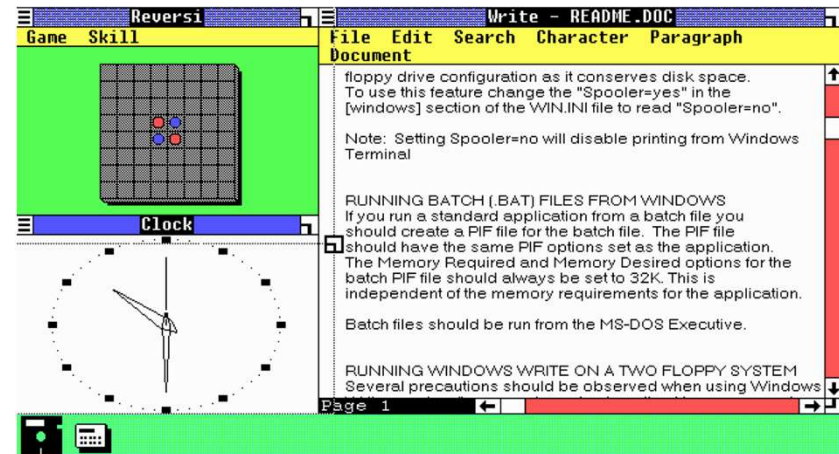
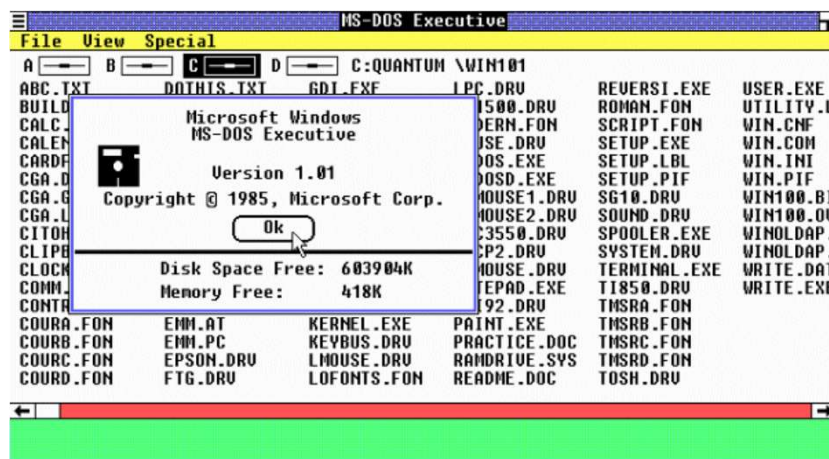
# Second wave: personal computing

## Macintosh (1984)



# Second wave: personal computing

## Windows 1.0 (1985)

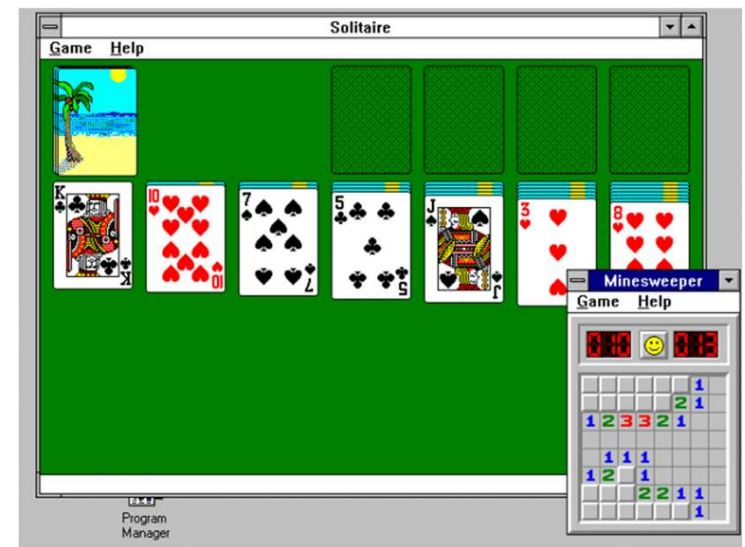




# Second wave: personal computing

## Windows 3.0 & 3.1 (1990 & 1992)

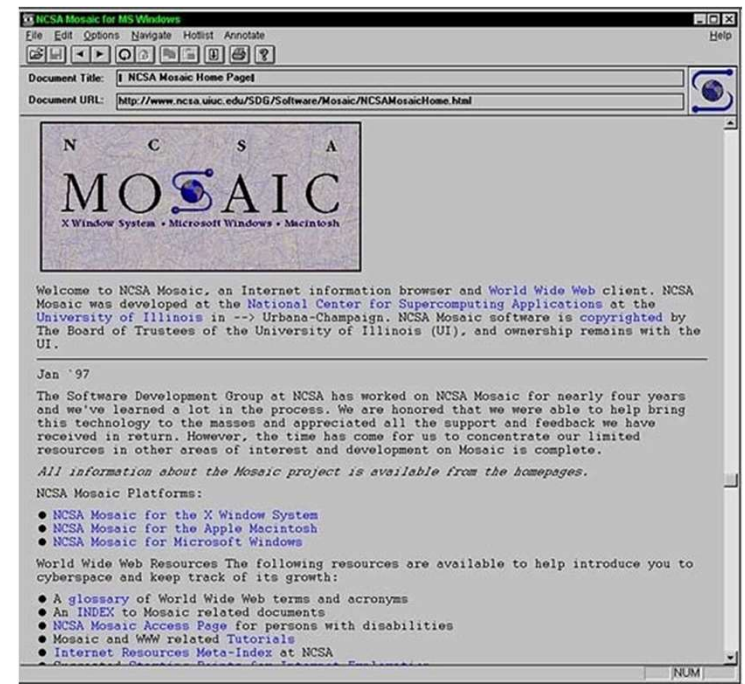
- Windowing became primary
- Added games: Solitaire, Minesweeper, and FreeCell!
- These were a trick to teach mouse skills



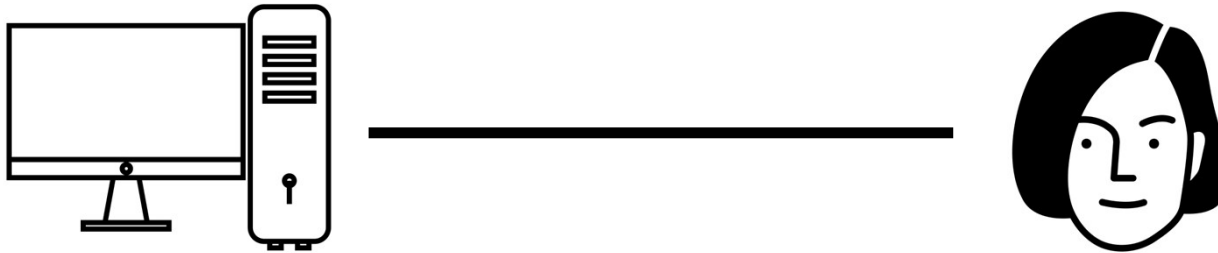
# Second wave: personal computing

## Mosaic Web Browser (1993)

- Originally for Unix systems, later ported to Mac and Windows
- “First” graphical web browser
- Microsoft IE came in 1995
- Apple didn’t make a browser until Safari in 2003



## Second wave: personal computing



“One to one”

# Three waves of computing



Mainframe  
computing



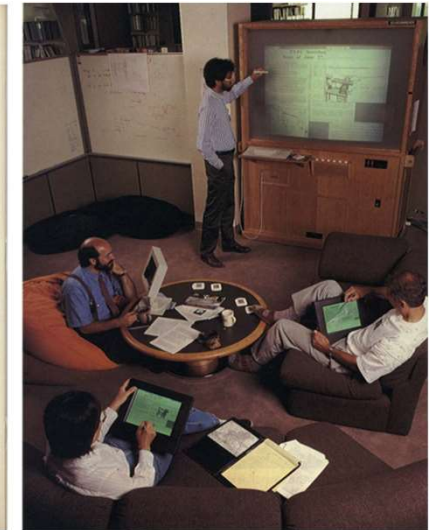
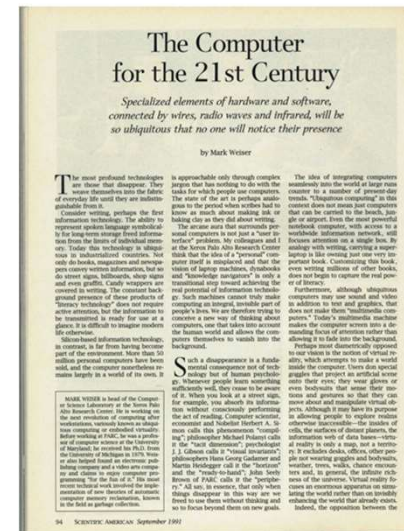
Personal  
computing



Ubiquitous  
computing

# Third wave: ubiquitous computing

- Weiser speculated people would interact with three types of computers
  - Tabs: inch-scale devices, like post-its
  - Pads: foot-scale devices, like paper
  - Boards: yard-scale devices, like whiteboards
- Speculated devices would have shared ownership



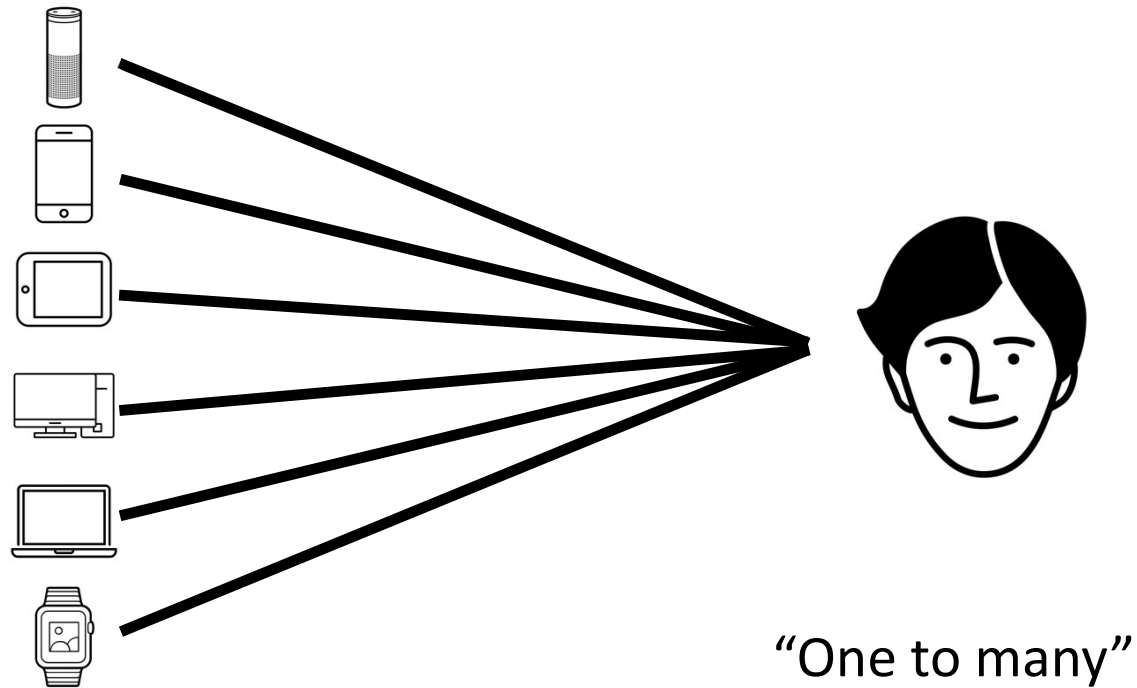
# Third wave: ubiquitous computing



# Third wave: ubiquitous computing

- Lines up with what we use today, for the most part
  - Tabs = phones and watches
  - Pads = tablets and laptops
  - Boards = interactive projectors? smart TVs? augmented reality?
- Still a strong sense of device ownership

# Third wave: ubiquitous computing

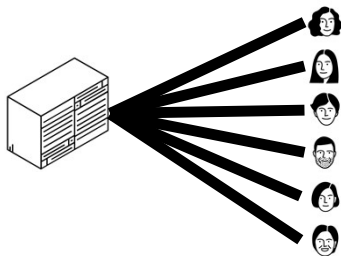




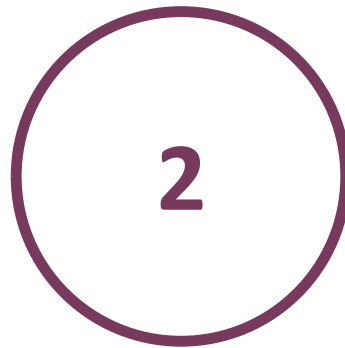
# Three waves of computing



Mainframe  
computing



“Many to one”



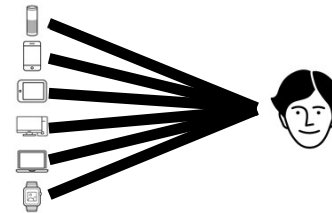
Personal  
computing



“One to one”



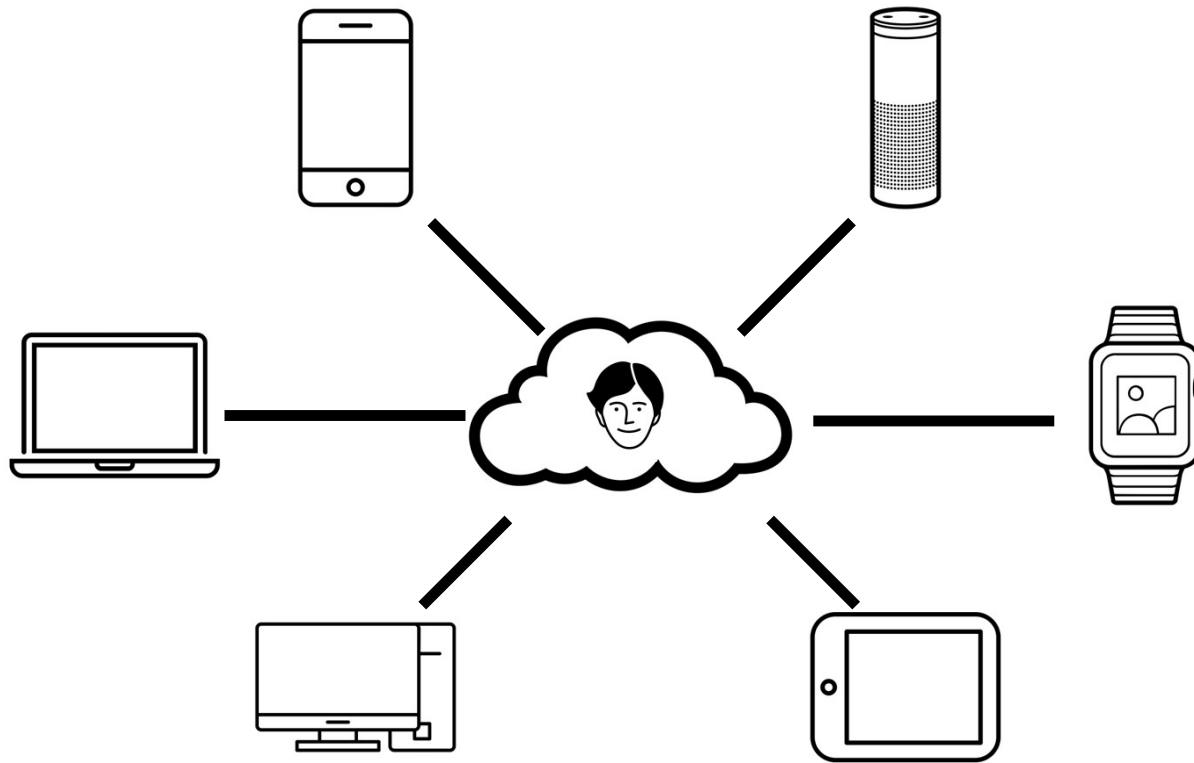
Ubiquitous  
computing



“One to many”

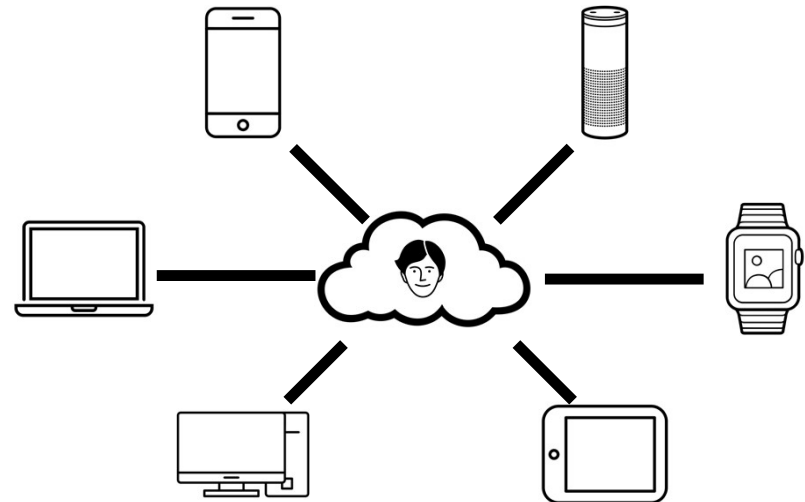
Why are web tools now the standard for interface development?

One to many, synced over the cloud



# One to many, synced over the cloud

- Use HTTP requests to send data to the cloud and receive data from it
  - JavaScript provided early tools to do this
- Render that data with HTML
- Style it with CSS



Ubiquitous computing is, in large part, why web tools are the current standard for interface development

# Web tools as the standard

- Nearly every platform needs to communicate with a cloud system
- Most need a web browser so people can access sites
- Shared programming language and development environment enables efficient work
- Developers can write once, deploy to many platforms
  - Hopefully customize style and functionality to the device
- Other reasons?

# Today's goals

**By the end of today, you should be able to...**

- Describe how society got to today's ubiquitous computing
- Hypothesize why web technology has become the de-facto tool for interface development
- Identify your course staff
- Summarize this course's goals and policies
- Describe upcoming course tasks

# Course Overview

- Course staff introductions
- Administravia
- Topics covered
- A0 (due Monday)



# Who we are

**Professor Mark S Baldwin (he/his/him)**

- Ph.D. Informatics,  
University of California, Irvine 2020
- Master of Human-Computer Interaction,  
Carnegie Mellon University 2013

**Radhit Dedania TA**

**Jensine Raihan TA**

**Devi Addanki TA**

**Xinru Tang Reader**

# The syllabus

- Explains due dates/times, assignment policies, exam goals, etc.
- It probably answers your question
  - Please check it before you ask us

# Staying in touch

- Web: <http://inf133.markbaldw.in/>
- Email us: See Syllabus
- Zulip: See Syllabus
  - Information will go out to Zulip first!
- For the most part, Canvas will only be used for submission and grades.
- Think about your questions, if it is a personal matter, then reach out via email. Otherwise use Zulip #general/#random streams!

# Staying in touch

- Office hours: on calendar
  - Instructor – I will be on Zoom every Tuesday during our scheduled class time. One on one, by appointment
  - TA's by appointment
- Submission: clone starter code from GitHub Classroom
  - Zip up finished assignment, submit on Canvas
  - Unsure about Git or GitHub? Go to discussion on Monday!
- YuJa: All weekly lectures will be uploaded here every Monday and linked in weekly notes page

# Communication best practices

- Zulip is best for assignment clarification and assistance
  - Please use the public channels to allow your peers to help
  - Direct messages to me are ok for now. But unless you ask a question during scheduled class hours, don't expect an immediate response.
  - Keep TA requests in public channels.
- Email is best for personal communication (absences, concerns, regrades)

# Course goals

## By the end of this course, you should be able to...

- Build webpages in HTML, CSS, and JavaScript which are sensitive to screen size and a person's abilities
- Build mobile apps in TypeScript frameworks, following design best practices
- Leverage external web APIs (databases, information sources) and device resources (photos, sensors) to lower development burden and enable new capabilities
- For a given design, choose appropriate devices to support and development frameworks to use

# Assignments

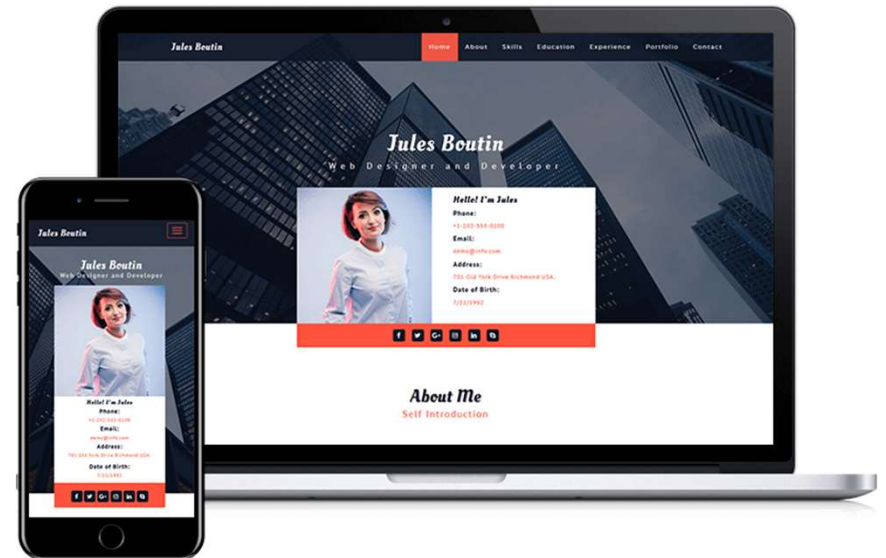
- A1: Personal web portfolio
- A2: Programming on the web
- A3: Web frameworks
- A4: Mobile development
- A5: Alternative interaction



# A1

## Responsive Portfolio in HTML and CSS

- Learning goal: develop familiarity with HTML and CSS, which form the foundation of all web design
- Apply *responsive* design, or adapt to screen size and orientation

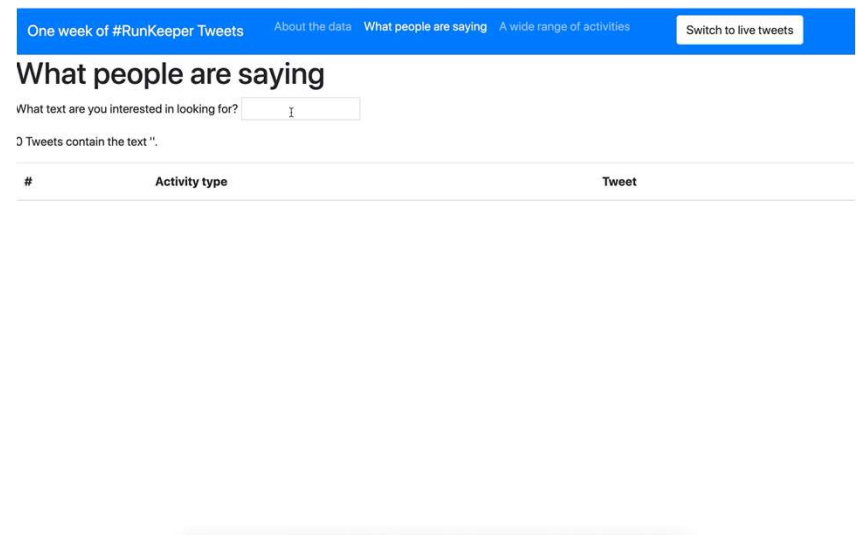




# A2

## Runkeeper Tweet Report in JavaScript and TypeScript

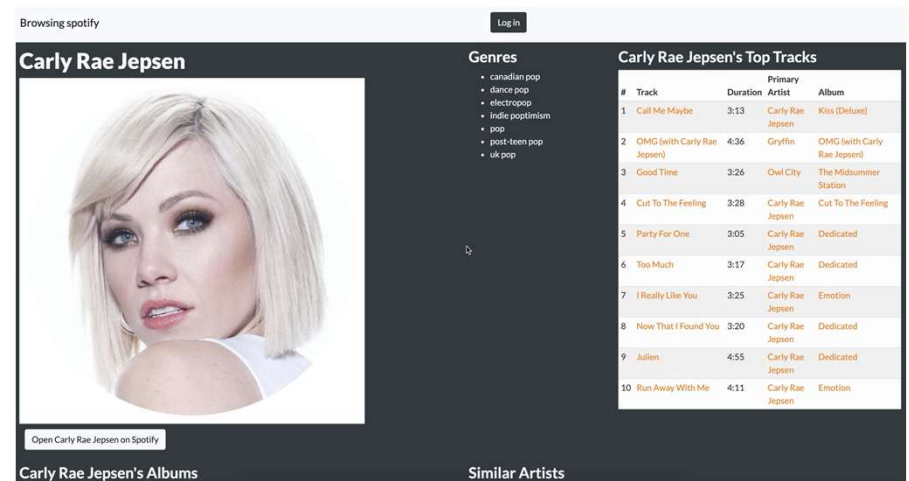
- Learning goal: become comfortable with JavaScript, a widely-used development language on the web
- Will learn to use JavaScript libraries for visualization and interaction



# A3

## Spotify Browser in Angular

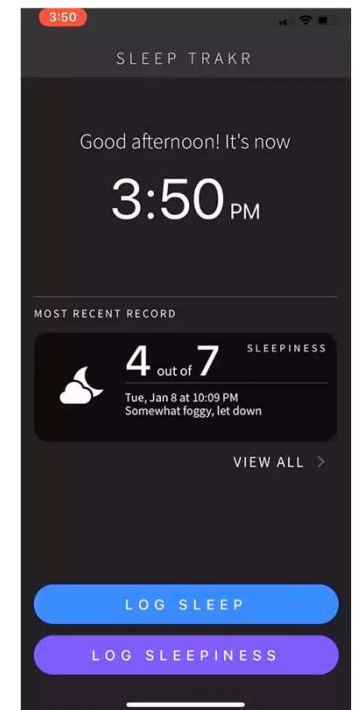
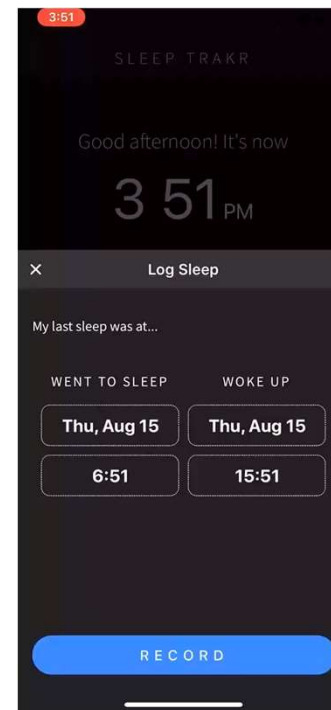
- Learning goal: develop skills in web frameworks which separate interface from data and interaction (Model-View-Controller)
- Will make an interactive browser of Spotify's library



# A4

## Sleep Tracker Mobile App

- Learning goal: learn to leverage UI components in a mobile framework and align with principles of good mobile design
- Will implement an app to log daily sleep



# A5

## Alternative Interaction

- Learning goal: Apply principles of user interface design to build an alternative mode of interaction
- Will implement with a web app of your choice
- Optional partner

# No Exams!

- A5 will serve as your final exam
- Weekly Socrative quizzes (we'll start next week, make sure you have Internet connection in class)

# Participation

- Socrative quizzes. Must complete all quizzes. Remote option available.
- Answer each other's questions on Zulip!
  - You can also get participation extra credit this way
  - You can respond faster than we can
  - Often times, you've experienced the same pitfalls

# Discussion sections

- Are optional
- Are a great opportunity to get more depth on topics discussed in lecture
- Opportunity to engage live with TA's

# Reflection

- This is an applied course with a lot of programming.
  - About half of the class will cover implementation techniques
  - The other half is theoretical concepts which inform design and development
- We'll teach principles and languages at a high level, but you'll need to pick up the specifics of APIs, packages, etc. on your own.
  - We're happy to help, but we haven't used every aspect of every API



# Reflection

- We have high expectations
  - We want you to make cool things
- But we also care and will listen
  - Let us know how things are going, ask questions
- Be “all in”
  - If you’re not ready to commit, please drop now
  - Someone else will be happy to take your spot
- The more active you are the better the experience will be for everyone

# Today's goals

**By the end of today, you should be able to...**

- Describe how society got to today's ubiquitous computing
- Hypothesize why web technology has become the de-facto tool for interface development
- Identify your course staff
- Summarize this course's goals and know how to find policies
- Describe upcoming course tasks