Miro link:
https://miro.com/welcomeonboard/QHAWlQVE5N54HhJWPKvl8ObXjSc7BlinhpcyQ9Eg1eOttmmQ0Y3w5BzqHnSqHYcf

Tech:
HTML, CSS, JavaScript (no frameworks)

Tools:
Github (Issues, Storyboarding, Version control)

Process: Agile

Testing: Unit tests,

Goal: Make a web-based pomodoro timer with a satisfying and functional interface with a responsive server and backend.

Pomo Philosophy:

1. The Pomodoro timer's main function is to alleviate the anxiety of having a unwieldily large goal at hand
2. As such, in order for it to function properly it must be respected.
    a. The timer bell must either cease or start work, it can't be blown off for a few extra minutes nor can it be implied early if the task doesn't take the full session
    b. Work done in a work session must be genuine, if the session is broken for any reason then that Pomodoro is lost
3. The Pomodoro technique/Pomodoro apps should be: **Simple to learn, Simple to apply, Nearly transparent in application**

DO:

1. Automate the manual aspects of the Pomo technique
2. Simplify meta-analysis
3. Have a clean UI with the timer in focus

DON'TS:

- Pause/play button (different from start/stop)
- Cluttered UI (be overloaded with features)
- Take effort to learn/understand
- App shouldn't be overloaded with features/ cluttered UI
- App shouldn't take effort to learn/understand

Features (Five Main Stages)

1. Planning
    a. When: At the start of the day

    b. Why: To decide on the day's activities
    c. In this stage you create your To-Do list for the day
        i. This is critical in giving your work sessions direction and holding you accountable for doing actual work
    d. Where do you start when planning?
        i. First, you'll need to to identify how many Pomos you would like to complete today, this will give you a task budget
        ii. Then you'll create a simple table with three columns: Task Name, Estimated # of Pomos, and Actual # of Pomos
            1. The comparison of your estimated number of pomos to your actual number of pomos will aid in your meta-analysis later on
2. Tracking
    a. Throughout your day you will be tracking 3 metrics
    b. Sessions Completed: Only record complete, full work sessions that were achieved
    c. Distracting Thoughts: Tally anytime an outside thought or urge that wasn't related to the task at hand
    d. External Distractions: Tally anytime a phone rang, someone outside the group tried to talk to you, or any other external source distracted you
3. Recording
    a. When: At the end of the day
    b. Why: To compile an archive of daily observations
    c. All of the tally marks and data manually collected earlier is of no use if it is not compiled into a larger document or spreadsheet
4. Processing
    a. When: At the end of the day
    b. Why: To transform raw data into information
    c. While the data is now in a format that is easier to manipulate, it is still just raw data at the end of the day
    d. The data needs to be manipulated, processed, and transformed into something that insight can be drawn from
5. Visualizing
    a. When: At the end of the day
    b. Why: To present the information in a format that facilitates understanding and clarifies paths to improvement
    c. The final step is where the real meta-analysis comes into play
        i. Now that the data is ready, visualizations need to be created so that humans can easily draw insight from them
    d. Analyzing your habits, when you work best, when you don't work best, and under what conditions are some of the things that will help you optimize your work flow to the max
    e. Theme/Graphics
6. Account management

App features:

1. Automatic recording / logging of activity
2. Muting of notifications / some way to preventing distractions
3. Skeuomorphic design when setting the timer (to mimic the physicality and ceremony of a real timer)
4. A way to visualize recorded / logged activity
5. A way to interact with the application without inherently opening it up
6. Tips for users
    a. If it takes more than five to seven Pomodoros, break it down into multiple tasks
    b. Similarly, if it takes less than one Pomodoro, combine it with some other smaller tasks
    c. Before creating a To-Do list, jot everything you need to do down in an unorganized fashion, this will be your “Activity Sheet”. From there, create your To-Do list by sorting your Activity Sheet by importance

Example sites

- Pomodoro Timer Online - Pomofocus
- TomatoTimer (tomato-timer.com)
- Pomodoro Timer (tomatotimers.com)
- Pomodoro Method Style Time Management Tool & Timer | Marinara by 352 (marinaratimer.com)
- The Pomodoro Tracker — A useful tool for time management based on Pomodoro Technique. (pomodoro-tracker.com)

More info on Pomodoro Technique

- What is the Pomodoro Timer? - the Muse
- Pomodoro Technique - Wikipedia
- The Pomodoro Technique - ToDoist
- Productivity101: An introduction to the Pomodoro Technique
- How to Pomodoro your way To-long lasting Productivity

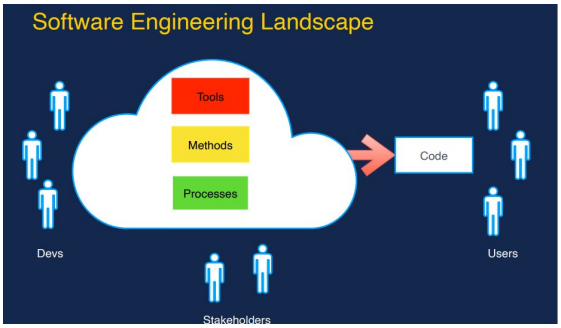
Agile:

1. 2 or three weekly meetings per week
2. Standup notes
    a. Accomplishments from last time
    b. Condensing what to say
3. Backlogs
    a. Breaking changes
    b. Changes happens next cycle
4. Sprint
    a. ~Two weeks changes of software
    b. Just like versions
    c. We can do one week

Prof Powell's Agile "Manifesto":

1. Customer Satisfaction by rapid delivery of software
2. Welcoming changing requirements even late in development
3. Working software is delivered frequently (weeks not months)
4. Working software is the principal measure of progress
5. Sustainable development, able to maintain a constant pace
6. Close, daily cooperation between business people and developers
7. Face-to-face conversation is the best form of communication (co-location)
8. Projects are built around motivated individuals, who should be trusted
9. Continuous attention to technical excellence and good design
10. Simplicity - the art of maximizing the amount of work not done - is essential
11. Self-organizing teams
12. Regular adaptation to changing circumstances
13. Individuals and interactions over process and tools
14. Working software over comprehensive documentation
15. Customer collaboration over contract negotiation
16. Responding to change over following a plan

Process Models

1. General Software Lifecycle Activities (May be discrete)
    a. Requirements
    b. Design
    c. Development
    d. Validation
    e. Deployment
    f. Evolution
2. "Umbrella activities" likely continuous
    a. Project Management
    b. Configuration Management
    c. Quality Control
    d. Process Improvement
3. Software Engineering Landscape
    a. 

Rough timeline

1. Week 1 - 2
    a. Brainstorming
        i. During the Meeting assign/think about people's roles, what GitHub Issues to create, how we might integrate our server/backend, figma/miro, wireframe UI design
        ii. To Do: Everyone should solo research for brainstorming sessions.

Bare Features

1. Checklist/ Task List
    a. If a single cycle takes more than 4 cycles, split up into more tasks
    b. Label Systems (tags, colors, categories)
2. 25 Minute
    a. If you finish your task early, "overstudy"
3. Short Break
4. Long Time
5. Logging/Analysis
6. Account Management
7. Settings Tab:
    a. How long the Pomodoro timer is
    b. How long the short/long takes
    c. For stop button, add an "are you sure?" button
    d. Limiting Distractions
        i. Remove seconds counter/show minutes to remove distractions
        ii. Locked all other features besides the stop button