

PUBLIC VIDEO

Instructions

We have a lot of time in our final exam period, so we expect you to show around 8 mins of a video (+/- 30seconds) that can let everyone know what you did and how well your project went. For bragging rights and sharing lessons learned, produce a video no longer than 8.5 mins and no shorter than 6.5 mins that covers the following :

- Team Name and Number
- Brief mention on video or with a picture of each member and their role(s)
- The product reveal and sizzle demo - show off the product as a user would experience. Think of it like what you might do at an event to launch a product and it has demos of features, things you are proud of, etc. (**2-3 mins**)
- A message to your fellow SWEs in the class (**the balance of the time 5+ mins**) which goes over your tech, pipeline, repo, project experience (pros/cons), future plans, things you wished you did, etc. Every team member should have a moment in that video to shine about what they did.

You are free to make your video humorous and style it any way you like as long as you believe it won't offend anyone. Have fun with this as we are going to vote for the ones we like the best in the final exam and the top three videos will get some bonus points (and of course bragging rights). A second video will also be due, and you can find details in its assignment; that video does not have to be polished and will be more informational and only viewed by the TAs and Prof.

Script

Team Intro (1-2 second)

Only show the team name and logo on screen, people will infer that we are working on pomotimer from our team name.

Product Reveal and Sizzle Demo (whatever time we have left (3 minutes)):

Scene 1 **NO LONGER THAN A MINUTE**: Student in room

Be on phone, look at watch.

Talking: "Ah man, it's finals week already? I have so many things to do and I don't know where to start! (**pan** to a whiteboard of some sort with a billion things on it, **pan back**, eye roll) Oh well, I'll just go on my phone then."

Transition: Spongebob *6 hours later* title

Student is still on their bed looking at a meme, **pan** to a clock. **Pan** back to student “*Gasp. Oh man, I need to get work done!


Student goes on the computer, opens up Google, looks up “How can I be productive?” while saying it out loud too, it would be funny if the person misspelled stuff/had funny things in their recent search history like “What is Java Script?”, league of legend builds, “is water wet?” and “How can I be like Deepak?”

Press enter, they get sent straight to our website (might need to cut and do a transition here), shocked/confused face

Scene 2: We have max 2 minutes to do this Screen sharing the website, **ENABLE SOUND**
“Oh wow, what is this?”

1. Tour, where we have the new user read each thing aloud (kind of fast paced, 2-3 min is not a lot of time
2. They look at FAQ to understand the app (maybe mention 1 or two things from there)
3. They look at the previous version of the app, say “wow, a lot has changed huh”
4. They play around with each setting, set custom timer lengths, let's plan out which colors to pick so that we ensure it looks good, show some of the fun sounds that we made/added, do dark mode, then switch back to light,
5. They make their tasks
 - a. Make 112 project video - 1 pomodoro
 - b. Work on 141L project - 2 pomodoros
 - c. Fold laundry/clean room - 1 pomodoro
 - d. Study for final - try 12 pomodoros, then move it down to 5
6. Move around and delete fold laundry/clean room task “**say eh I can do that later**”
7. have user accidentally click delete all, then press cancel “**say ooh dang glad I didn't accidentally delete everything**”
8. Run through sessions, but only show the last however few seconds of each pomodoro session (or a time lapse of the timer) “**maybe put some montage music over this**”
 - a. Showcase the device not required feature and distraction feature with the user going on their phone or something for 2 sessions
9. Show them with the day finished, have them look through stats page “**nice, I didn't get too distracted today**”
10. Show a relatively-happy student, have them walk outside, **touch grass**
11. Narrator: “And that was the first time this student has been outside in over a week”
12. The end?

A message to your fellow SWEs (at most 5.5 minutes)

 TeamXV Finals Message

Team member introductions (1-2 minutes, 12 seconds MAX each)

Each member: “Hi, my name is XXX, my role on the team was XXX, and...”. Here we can each take a moment to show a slide and talk over what we were most proud of working on in the project. (JUST AUDIO)

Matthew: Hi, my name is Matthew and I’m one of the team leads for Team XV. Since this is my first time leading, I was very happy to see the project through to completion. Besides managing stuff, some things I’m proud of on the dev side were improving the task list, reworking the settings popup, and creating the session indicators.

Yizhou:

Allen:

Nick K: Hi, my name is Nick, my role on the team was development and documentation, and some of the work that I did was handle Github repo organization, the pipelines, and polishing features and bug finding. Because this project was my old 110 project and I was the repo owner this time around, I spent a lot of time learning about Github, as well as learning about testing using Jest. I am incredibly proud to see how far this project has come, but beyond this class, I want to keep improving my technical skills, as I think this is the area where I can improve the most.

Zhuoran: My name is Zhuoran, my role in team XV was developer. I mainly participated in works of checklist attributes, minimization of distractions during focus sessions and redesigned the logos and icons.

Pablo: Hi, my name is Pablo, and my role on the team was developer. I was helpful in providing feedback on design and feature decisions, and notes taking for much of the group's meetings. I also worked on the editing and voice over of both the middle of the quarter video and the private final video.

Edgar: Hi, my name is Edgar, and my role in team XV was developer. Some of the work that I’m most proud of is working on variable timer lengths and adding new custom work and break sounds, some of which I created from scratch through ableton live. Aside from this, I also worked on styling, UI polishing and fixing numerous bugs.

George: Hey, I'm George. I was the planner and one of the developers on the team. My favourite contribution was adding color customisation capabilities to our app and improving the UI.

Joseph: Hi, my name is Joseph, my role here in Team XV was developer. I was involved in mostly code polishing, but my most proud work would be the initial team logo design, the prompt when user is about to delete all tasks and also the implementation of browser notifications. I also added a few animations for the timer, current task, and task list itself when user enters focus mode.

Westley: Hi, my name is Westley, I'm a developer in Team XV. I mostly help with code testing. Most of the time involves fixing and updating failed tests from other's new commits. Although most of my time is spent googling on stackoverflow only to realize the issue is caused by completely different results, I'm still glad that I managed to locate and fix the issues. I also help decide on some element locations that improve the UI/UX of the timer.

(Remaining 5 slides should each take at most 36 seconds each)

Tech:

- Vanilla JS
- Materialize
- Jest

CI pipeline:

- No major changes to the pipeline
- Token issues (Thanks deepak!)
- Was planning on adding HTML/CSS validator in CI but scratched/ not implemented in the end

CD pipeline:

- Github pages
- Github automatically deploys the pages after testing for public repository (happy accident?)
-

Now for our pipeline, the developer first have to create a new branch, where the name have to follow our guideline in github wiki, after making new features and push their commits. Upon creating a new pull request, github actions will run to run through tests and linting check. In addition, the code have to also get approval from 2 peers in order to merge onto main. Once all of the above is approved and successfully merged, Github pages will automatically deploy the site with new features!

-

Repo:

README for how to get started

Separate documentation and code base repos

Wiki for how to contribute to the code base

ESDoc for documentations

Testing:

Exceed global threshold in jest? Just change the threshold limit! (not really) LOL

Project Experience (pros and cons):

Future Plans

"Finally, here are some of the things that we wish we did or that we plan to do in the future. First of all, we feel that the mobile experience could be further improved upon, especially the UI. In

terms of accessibility, we feel that our color scheme is not very friendly for colorblind users, and we inherited a case of div-itis from the previous team, so our HTML is not as semantic as it could be. In the future, fixing up the HTML, refactoring code, and having better documentation would help improve the developer experience. Better statistics to track daily habits of the user was also something we wished to add, as well as just continuing work on existing issues that we didn't finish this quarter."

PRIVATE VIDEO

Instructions

To communicate information both to the graders and to a person who might take over your project at a future date you should make a video (shorter than 15 minutes, and longer than 6 minutes) to help us understand your project, pipeline, and team effort. This video does not have to be fancy or polished, but it might be wise to at least have a nice title and conclusion section for presentation consistency.

Your private video should follow this format

- Team Name and Number
- Each team member should be present and in their own words state what they did on the project, what they are proud of and what they feel they could improve upon or address beyond the class (assume around ~30seconds per member)
 - Similar to our team intro, send this to google drive.
- A video discussion about how to access your repo, a tour of the repo organization and a demonstration of making a small change to the repo and the completion of the build process. You may need to write a more detailed discussion of building or onboarding the product than the video supports. If so, create a markdown file called "onboard.md" and put it in your documentation which goes into more details.
- A more involved retrospective about how your Agile practices were conducted and what challenges and victories the team faced. Honesty is the best course of action here as we are more likely to be lenient if we understand you are aware of what you didn't do well on the project and practices.
- Conclude your video with a discussion of what things a team taking on your project should do next. Discuss the features not built yet, improvements to the build pipeline or anything else you would have gotten to given another

quarter worth of work.

Do not make this video in a marketing or humor sense, be informational and candid. Remember this video is for grading and will not be shown to your fellow Spring 22 students.

Script

Show team logo for a couple seconds, transition to individual introductions.

Individual portion:

(For each member, show your face in zoom with the PomoXV background, and discuss what you've done. This can be very close to what you said in the public video)

Just show introductions for each member.

Repo portion:

After individual introductions, indicate that the next section will be a repository tour with a title card or something of the sort.

Show repository home page

https://github.com/nickkro25/cse112_team15

Welcome to our repository! The repo is public so it can easily be accessed by anybody. The code for our app is organized in a straightforward fashion - we have our JavaScript files in the JS folder, our CSS files in the CSS folder, our images and audio files in the assets folder, and our unit tests in the tests folder.

Show readme

Scrolling down to the readme, a link to the publicly deployed version of the app is included, and basic instructions are given on how to get the app set up and running on your local machine.

Show commands being entered

To get started, once we have NPM and Node.js installed, we just clone the repo and use npm to get any other dependencies. We can then run the `npm start` command and navigate to the link indicated by the console output to check out the app locally.

Show "How to Contribute" wiki

Now if we want to start making changes, we can check out the wiki's "How to Contribute" section for more details. To put it simply, you first create a separate branch, title it in this format:

```
(feature|bugfix|meeting_notes|admin)/DESCRIPTION_OF_BRANCH
```

Show actual demo of this

And get to work. You can use `npm run lint` and `npm run test` at any time to check if your work will pass our linter and unit tests. Finally, you can pull and merge from main to update your local files and fix any conflicts that might pop up. You can then push your branch to remote and make a pull request, listing any issues your work might close, what changes you've made, and any lint changes. Once your work passes the automatic verification process, it will then need to be approved by two team members before it can be merged with main. Once everything is cleared and merged, the branch can be safely closed.

By the way, code documentation can be found in the dev-docs branch, while logistical documentation such as ADRS, meeting notes, and other such information can be found in a separate repo linked in the readme.

Retrospective portion:

NOTES

- How were Agile practices conducted?
 - Didn't really do formal sprints, but we did review our work at the end of each week and assigned tasks for the next week.
 - Standups using Slack bot
 - Always maintained a working version of the app with CI and CD working
 - Welcoming change
 - Self-organizing (people worked on what they wanted to work on)
- Challenges
 - No Designer
 - Felt directionless, very uncertain about UI and goals in mind, didn't have specific plans
 - Development didn't actively start until later weeks
 - Midterm got in the way of the first checkpoint, pushed us back by an entire week
 - Having to jump on an existing code base
 - Had to give up on gamify portion of the application
 - Spent less time on polishing, more time on new features
 - CI/CD broken with token issues
 - Mobile support was challenging (IOS)
 - Did not test on different OS such as IOS until last week
 - Other classes/obligations getting in the way of making progress
 - Lack of gender diversity on the team, limited our perspectives (yes we blame our luck draw)

- Fixing tests and adding new tests (uncovered lines)
- Varying skills in web dev (lots of stack overflow and mozilla :)
- Victories
 - Practiced web dev skills
 - Learned about working in an established code base
 - Created a safe team environment (okay to make mistakes or take time off for unforeseen issues) :)
 - Grew as a team, cohesiveness increased throughout
 - Good accountability, dependable teammates
 - Deep dive into UI/UX design choices
 - Learn to work collaboratively with teammates
 - Showed restraint in terms of new features, focused on the quality of the app first
 - Decided features as a team with voting
 - Whole team got invested and wanted the product to succeed

Future team:

- Improved statistics
- Testing can be more robust
- Account system
- Ugly color theme transition upon refresh
- Better mobile UI
- Restructure CSS
- Add CSS/HTML validator to the pipeline
- Nav header switches colors slower
- Better accessibility
- More consistent UI (highlighting, colors, modals such as the delete all modal)
- Receive user feedback
- Work on better documentation
- Improve DX (divitus, rigidity)

SCRIPT

Transition to title card indicating “Retrospective”

Reflecting back on this quarter, our team, while not perfect, did try to adhere to as many Agile practices as we could.

Show meeting note example

For example, while we didn’t have formal sprints layed out, we did review our work at the end of each week and assign tasks for the next week as a sort of informal retrospective.

Show standups

We also held daily standups using a Slack bot to keep track of what members were working on. Early on, our members rarely filled out the standups, but the team got better about filling it out as we got deeper into development.

Our team also did a pretty good job upholding Agile principles. For example,

- We stayed open to changes even late in the project (i.e session indicators being added last week of development)
- We managed to maintain an online and working version of the app at all times thanks to our CI/CD pipeline
- We made changes frequently and continuously (small issues worked on and merged to main)
- Our team was relatively self-organizing (people picked issues they wanted to work on, collaborated with others who had the same interest).

However, there were definitely a lot of challenges that we had to deal with along the way:

- Lack of direction early on - very uncertain about UI, didn't have any specific visions in mind. To make matters worse, our designer dropped the class, so we were forced to make up for it by trying to design things as a team (imagine trying to come to a consensus quickly and efficiently with 10 members)
- Development started later than expected - slow to plan things out, first week of development dropped to work on midterm. Slow start to development also forced us to give up some features we were excited about implementing, such as improved statistics and gamification.
- Despite a promising initial scan, actual work turned out to be harder than expected due to some spaghetti in the code base (task list was a pain to work with, CSS files were kind of a mess, app straight up didn't work on iOS)
- Other classes and obligations often put team members out of commission
- Wide range of CSE110 backgrounds also made it tougher for some team members, being forced to play catch up with the web dev stack

Still, despite these challenges, our team had a lot of victories:

- Many members were happy to have gotten practice with web dev skills and UI / UX design
- Got experience working in an established code base, which is good practice for future career
- Entire team agreed that we cultivated a safe team environment. Major changes to the project were decided on as a team. Members felt okay to make mistakes or take time off for unforeseen issues, took accountability, and found each other dependable.
- Growth as a team - our cohesiveness and collaboration noticeably increased as the quarter went on
- Stayed true to our principles and vision by showing restraint in terms of new features and focusing on the quality of the app first
- Managed to improve the quality of the app to a level we're happy with.

Of course, there are still changes that we didn't get around to that future teams could work on:

- Improved statistics and an account system were features we felt would be useful to the user but didn't have time to implement. The statistics could help the user track their habits, and the account system could let users share the same statistics and task list across multiple devices.
- There are still existing issues we wanted to address:
 - The mobile UI is usable but could still be more friendly.
 - The developer experience is not great; there's band-aid code lying around, the HTML is full of div-itis, the CSS steps on its own toes, and the color theme code is frankly just disgusting.
 - There's an annoying color theme transition that always happens upon refresh.
 - Certain parts of the UI are still inconsistent, such as the delete all modal.
 - We wanted to make the app more accessible, especially in terms of color contrast and assistive device support.
 - Analytics would also have been good to add, to get better feedback of how users are interacting with the app.
 - Code documentation is okay, but not great.
 - And of course, testing could always be more robust.

Overall, though, despite not getting as close to 1.0 as we would have liked, our team is proud of the work we've accomplished.