PostPilot

Software Engineer

Take-Home Project

Overview
Details
- Step 1: Review the Shape Doc
- Step 2: Break Down and Track The Work
- Step 3: Work the Project!
- Step 4: Write an Update and Submit Your Work

Overview

At PostPilot, we're looking for engineers that enjoy taking on broad responsibilities, solving problems, and working asynchronously. Obviously, we're looking for people that are (or on their way to becoming!) excellent software engineers - but more than that, we're looking for people who can work effectively within our environment!

We place a heavy emphasis on well-defined objectives and team autonomy, with accountability to outcomes over output. We embrace a "manager of one" philosophy, and expect our engineers to be exceptional communicators and be able to independently own the planning and delivery of software projects from concept to completion. In exchange for strong accountability and ownership, we give our team wide latitude and independence to solve problems.

The purpose of this take-home project is two-fold: we want to assess you as a candidate to understand better how effective you can be in this environment, but just as importantly, we want you to assess *us* as an employer!

Finally, the need to prioritize and live under time constraints are a ruthless constant at a growth-phase startup like PostPilot. You will need to be mindful of time-boxing your work, and this project will be no exception! **We want you to try to limit the time you spend on this effort to around 2, no more than 4 hours**.

Details

At PostPilot, we follow a plan/build process inspired by Basecamp's "ShapeUp". In a nutshell, our engineers work from project plans we call Shape Docs, in which we outline the work to be done, as much applicable context and information as we can provide. (Notably, they are deliberately intended to be somewhat high-level, providing the "what" and "why", but leaving as much of the "how" as possible up to the implementing engineer and broader team.)

Our engineers are expected to take these documents and work independently (often with other members of the team) to break down the work into tasks and to-dos in Basecamp, complete them, regularly update the project hill chart with status updates, frequently integrate and ship updates to production (usually behind a feature flag), and ultimately release the new capability to our users.

In this take-home, we're going to do a "mini project" to simulate this cycle. Some details aren't important right now: for example, it's OK if you've never used Basecamp and aren't familiar with the concept of a hill chart. But others are: can you write clean code? Document your process and progress clearly? And so on.

To complete this project, we'd like you to do the following:

1. Open and review the shape document.
2. Using a project management tool of your choice, break down and track the work to be done.
3. Implement as much of the project as you can given the timebox provided above, updating and documenting your progress in the project management tool.
4. Send us your completed project along with a short written summary of what you were able to complete.

Step 1: Review the Shape Doc

We generally use Google Docs for shaping - because defining new work is such a collaborative process, we've found the collaboration tools in Google Docs to be a great way to asynchronously define new work.

For this project, you'll find the shape document outlining the work to be completed here: [Shape Doc] Engineering Take-Home Project We've attempted to structure the document similar to a "real" shape doc we might use for product development.

To get started, pull up the doc linked above, read through, and familiarize yourself with the task at hand. (And don't forget that at PostPilot, you'll work with a team around you - so don't be afraid to reach out with any questions as you go!)

Step 2: Break Down and Track The Work

You can use whatever software tool you prefer for this step - if you're familiar with Basecamp and want to sign up and use a free account for this, great! But Trello, Jira, Pivotal Tracker, etc. are all OK.

After reviewing the shaping document, think through how you'd like to tackle the project - what tasks will you work on first? What items need discovery? Are there ways you can sequence the work to reduce project risk, and ensure you can ship a minimally-viable feature at the end of our timebox?

Do an initial pass and track all the major contours of the work as you think you'll approach it. Write up each task clearly, and provide relevant details inside each task as you see fit.

The goal here: we want you to demonstrate how you tactically think through the execution of a project, and clearly and independently communicate the work to be done in a tracking tool!

Step 3: Work the Project!

Dive in and begin coding!

As you work, update your project tracker, and note updates and changes as you go. In real life, the tasks we expect to do at the outset of projects often change dramatically as we get into the weeds, so don't be afraid to update and update and refactor your work tasks in the tool as you learn more in the course of implementation.

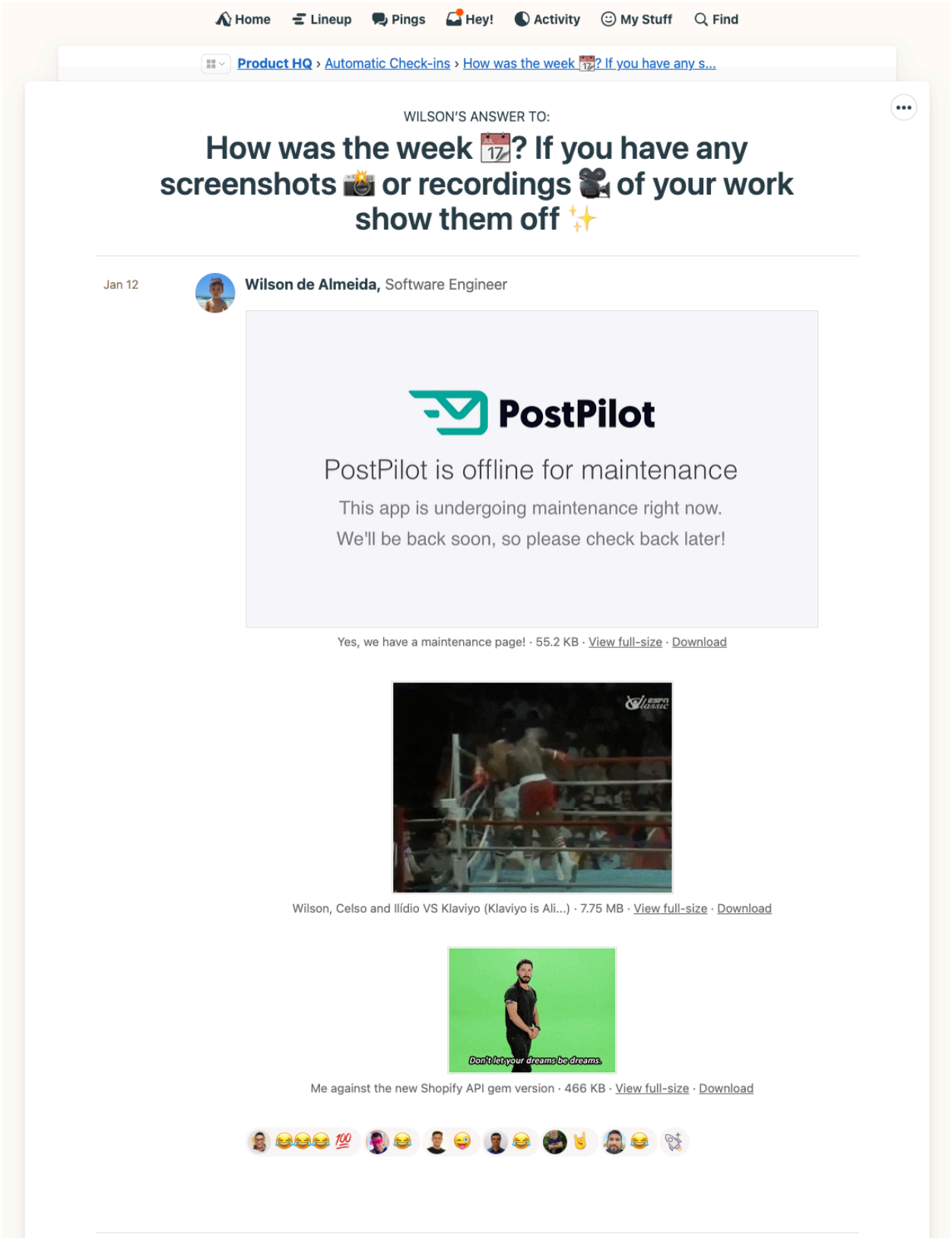
The goal is to hit the end of the time-box with a functional implementation, but you might run out of time before we get there. (Something that also happens a lot in real life!) That's ok, too - but hopefully you'll have been able to get as close as possible.

Step 4: Write an Update and Submit Your Work

At the end of every week, our engineering team has a practice of writing up a short update of their efforts for the week, and sharing it with the broader team.

Since we work asynchronously, it's been a great way to keep track of what everyone is doing, as well as celebrate what we've been able to accomplish. As part of this, we love to include screenshots or recordings, funny memes, and anything else to help us stay connected.

For an example, here's a couple past updates from folks on our team:



As the final step of your take-home assignment, we'd love for you to write up a short update for us summarizing your work on the project. Again, use whatever tool you'd like to complete this (email is fine!), and send it back to us, including your completed implementation for review.

From here, we'll take a look, and reach back out to follow up as soon as we can.

Good luck!

♥ *PostPilot engineering team*