Creativity in motion Developer Challenge

Build A Product: **Maintenance Tracker**

### **BUILD A PRODUCT: Maintenance Tracker**

#### Project Overview

Maintenance Tracker App is an application that provides users with the ability to reach out to operations or repairs department regarding repair or maintenance requests and monitor the status of their request.

#### Required Features

1. Users can create an account and log in.
2. The users should be able to make maintenance or repairs request.
3. An admin should be able to approve/reject a repair/maintenance request.
4. The admin should be able to mark request as resolved once it is done.
5. The admin should be able to view all maintenance/repairs requests on the application
6. The admin should be able to filter requests
7. The user can view all his/her requests

### **Preparation Guidelines**

These are the steps you ought to take to get ready to start building the project

Steps

1. Create a **Pivotal Tracker Board**

1. Create a **Github Repository, add a README, and clone it to your computer**

***Tip****: find how to create a Github Repository* [*here*](https://www.google.com.ng/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=0ahUKEwiYjuDNnNnVAhXlJ8AKHSacB9gQFggsMAI&url=https%3A%2F%2Fhelp.github.com%2Farticles%2Fcreate-a-repo%2F&usg=AFQjCNEkq3HBxdtnBJ1YKqccJVhFEv1Iuw)*.*

### **Challenge 1 - Create UI Templates**

#### Timelines

* **Expected Length to Complete: 1 week**
* **Due Date:**  7th June at 11:00 AM

#### Challenge Summary

You are required to create UI templates with **HTML**, **CSS** and **Javascript**.

***NB:***

* *You are not implementing the core functionality yet, you are only building the User Interface elements, pages and views!*
* *You are to create a pull request to elicit review and feedback for the UI template when you are done working on them*
* *Do not use any css frameworks e.g bootstrap, materialize .*
* *Do not download or use an already built website template.*

#### Guidelines

1. **On Pivotal Tracker, create user stories to setup the User Interface elements:**
   1. User signup and signin pages.
   2. A page where an admin can see all the requests on the application.
   3. A page/pages where an admin can do the following:
      1. See the details of a request
      2. Approve or disapprove a request
      3. Resolve a request
   4. A page/pages where user can do the following:
      1. See all his/her requests
      2. create requests
2. **On Pivotal Tracker create stories to capture any other tasks not captured above. The tasks can be feature, bug or chore for this challenge.**
3. **On a feature branch, create a directory called UI in your local git repo and build out all the necessary pages specified above and UI elements that will allow the application function into the UI directory**
4. **Host your UI templates on** [**GitHub Pages**](https://pages.github.com/)**.**

***Tip:*** *It is recommended that you create a* ***gh-pages*** *branch off the branch containing your UI template. When following the GitHub Pages guide, select* ***"Project site"*** ***>>*** ***"Start from scratch"****. Remember to choose the* ***gh-pages*** *branch as the* ***source*** *when configuring Repository Settings.*

**Target skills**

After completing this challenge, you should have learnt and be able to demonstrate the following skills.

|  |  |  |
| --- | --- | --- |
| **Skill** | **Description** | **Helpful Links** |
| **Project management** | Using project management tool(pivotal tracker) to manage your progress while working on tasks. | * To get started with Pivotal Tracker, use [Pivotal Tracker quick start](https://www.pivotaltracker.com/help/articles/quick_start/). * [Here](https://docs.google.com/document/d/1ADcZ54o1s2aBtZ0dZ7fs_liFew1zTg10JqZxgRuNqm4/edit) is an sample template for creating Pivotal Tracker user stories. |
| **Version control with GIT** | Using GIT to manage and track changes in your project. | * Use the recommended [Git Workflow](https://www.atlassian.com/git/tutorials/comparing-workflows" \l "gitflow-workflow), [Commit Message](http://chris.beams.io/posts/git-commit/) and [Pull Request (PR)](https://guides.github.com/activities/hello-world/" \l "pr) standards. |
| **Front-End Development** | Using html and css to create user interfaces. | * See this [tutorial](https://www.youtube.com/watch?v=Rf_DjL_dbug) * See this [tutorial](https://www.youtube.com/watch?v=kbLfWKGVsMQ) also |

#### Self / Peer Assessment Guidelines

Use this as general guidelines to assess quality of your work. Peers, mentors, and  facilitators should use this to give **feedback** on areas that should be improved on.

|  |  |  |  |
| --- | --- | --- | --- |
| **Criterion** | **Does not Meet Expectation** | **Meets Expectations** | **Exceed Expectations** |
| **Project management** | Fails to break down modules into smaller, manageable tasks. Cannot tell the difference between chores, bugs and features | Breaks down each module into smaller tasks and classifies them. Constantly updates the tool with progress or lack of it | Accurately, assigns points to the tasks. Informs stakeholders of project progress/blockers in a timely manner |
| **Version Control with Git** | Does not utilize branching but commits to master branch directly instead. | Utilizes branching, pull-requests, and merges to the develop branch. Use of recommended commit messages. | Adheres recommended GIT workflow and uses badges. |
| **Front-End Development** | Fails to develop specified HTML/CSS web pages or uses an already built out website template, or output fails to observe valid HTML document structure | Successfully develops HTML/CSS webpages while observing standards such as doctype declaration, proper document structure, no inline CSS in HTML elements, and HTML document has consistent markup | Writes modular css that can be reused through markup selectors such as class, id. Understands the concepts and can confidently rearrange divs on request. |