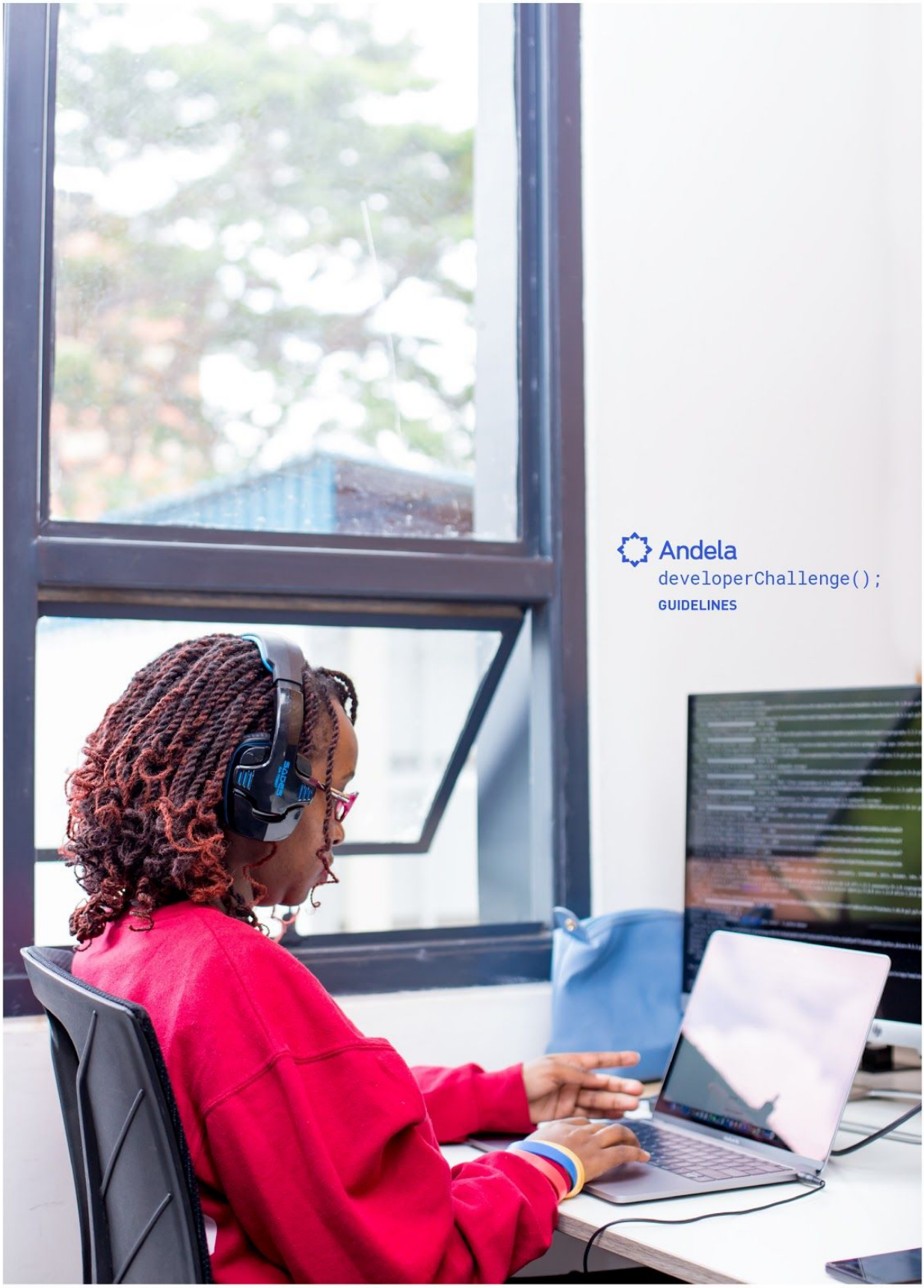
Andela



Andela

developerChallenge();

GUIDELINES

Andela

Andela Developer Challenge

Build A Product: **Store Manager**

Andela

BUILD A PRODUCT: Store Manager

Project Overview

Store Manager is a web application that helps store owners manage sales and product inventory records. This application is meant for use in a single store.

Required Features

1. Store attendant can search and add products to buyer's cart.
2. Store attendant can see his/her sale records but can't modify them.
3. App should show available products, quantity and price.
4. Store owner can see sales and can filter by attendants.
5. Store owner can add, modify and delete products.

Optional Features

1. Store owner can give admin rights to a store attendant.
2. Products should have categories.
3. Store attendants should be able to add products to specific categories.

NB

1. This application should help store owners avoid selling products that have run out of stock.
2. The store owner can be referred to as the admin of the application.

Andela

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**

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

   ***Tip**: find how to create a Github Repository here.*

Important Dates:

Project Challenge	Date of Release	Due Date
Challenge 1	**06/10/2018**	**12/10/2018**
Challenge 2	**13/10/2018**	**19/10/2018**
Challenge 3	**22/10/2018**	**26/10/2018**
Challenge 4	**29/10/2018**	**02/11/2018**

Andela

Challenge 1 - Create UI Templates

Timelines

- **Expected Length to Complete: 1 week**
- **Due Date: 12/10/2018**

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 templates when you are done working on them*
- *Do not use any CSS frameworks e.g Bootstrap, Materialize, sass/scss.*
- *Do not download or use an already built website template.*

Guidelines

1. **On Pivotal Tracker, create user stories to setup the User Interface elements:**
    a. User login page.
    b. A page/pages where a **store attendant** can do the following:
        i. Add products to shopping cart, and create sale record.
        ii. View sale records created by the individual store attendant, which cannot be edited.
    c. A page/pages where a **store attendant/admin** can do the following:
        i. View all available products.
        ii. View an individual product details such as the quantity in inventory and also the minimum inventory quantity allowed at any time.
    d. A page/pages where an **admin** can do the following:
        i. Create a new sale attendant user account.
        ii. Create, modify and delete a product.

iii. View sale records by all store attendants.

Optional:

e. A page for a **store attendant's profile** which, at minimum displays:
   i. The total number of sale records created by the individual sale attendant.
   ii. The total number of products the individual sale attendant has sold.
   iii. The total worth of goods sold by the individual sale attendant.

f. A page where an **admin** can assign a product to a specific category.

2. **On Pivotal Tracker, create stories to capture any other tasks not captured above. A task 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.**

__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. • Here 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, Commit Message and Pull Request (PR) standards.
Front-End Development	Using HTML and CSS to create user interfaces.	• See this tutorial • See this tutorial also

Andela

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.