



 **Andela**
developerChallenge();
GUIDELINES

© Andela Confidential



Andela Developer Challenge

Build A Product: **Questioner**



BUILD A PRODUCT: Questioner

Project Overview

Crowd-source questions for a meetup. **Questioner** helps the meetup organizer prioritize questions to be answered. Other users can vote on asked questions and they bubble to the top or bottom of the log.

Project Timelines

- **Total Duration:** **4 weeks**
- **Final Due Date:**

Required Features

1. Admin can create meetups.
2. Users can create an account and log in.
3. Users can post questions to a specific meetup.
4. Users can **upvote** or **downvote** a question.
5. Questions are sorted based on the number of upvotes a question has, which helps the meetup organizer(s) to prioritize questions most users are interested in.
6. Users can post comments to a specific question.

Optional Features

1. Admin can add images to a meetup record.
2. Admin can add tags to a meetup record.
3. User can reset password.



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	02/01/2019	07/01/2019
Challenge 2	07/01/2019	11/01/2019
Challenge 3	14/01/2019	18/01/2019
Challenge 4	21/01/2019	25/01/2019



Challenge 1 - Create UI Templates

Challenge Summary

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

Timelines

- **Duration:** **5 days**
- **Due Date:** **7th January 2019**

NB:

- *You are not implementing the core functionality yet, you are only building the User Interface (UI) elements, pages, and views!*
- *You are to create a pull request for each feature in the challenge and then merge into your develop branch.*
- *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 (UI) elements:**
 - a. User sign-up page.
 - b. User sign-in page.
 - c. A page/pages where a **user** can do the following:
 - i. Post a question to a specific meetup.
 - ii. **Upvote** and **Downvote** a question.
 - iii. Add a comment to a question.
 - d. A page/pages for a user's profile which, at minimum displays:
 - i. The number of **questions** the user has posted.
 - ii. The number of **questions** the user has commented on.
 - iii. **Top questions feed** for upcoming meetup(s) that the user is scheduled to attend.
 - e. A page/pages where an **Admin** can do the following:
 - i. Create a meetup.
 - ii. Delete a meetup.



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 learned and be able to demonstrate the following skills.

Skill	Description	Helpful Links
Project management	Using a project management tool (Pivotal Tracker) to manage your progress while working on tasks.	<ul style="list-style-type: none"> To get started with Pivotal Tracker, use Pivotal Tracker quick start. Here is a sample template for creating Pivotal Tracker user stories.
Version control with GIT	Using GIT to manage and track changes in your project.	<ul style="list-style-type: none"> Use the recommended Git Workflow, Commit Message and Pull Request (PR) standards.
Front-End Development	Using HTML and CSS to create user interfaces.	<ul style="list-style-type: none"> See this tutorial See this tutorial also
UI/UX	Creating good UI interface and user experience	<ul style="list-style-type: none"> See rules for good UI design here See this article for More guide For color palettes, see this link



Self / Peer Assessment Guidelines

Use this as general guidelines to assess the 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 to recommended GIT workflow and uses badges.
Front-End Development	Fails to develop the specified web pages using HTML/CSS/JavaScript or uses an already built out website template, or output fails to observe valid HTML/CSS/Javascript syntax or structure.	Successfully develops web pages 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.
UI/UX	The page is unresponsive, elements are not proportional, the color scheme is not complementary and uses alerts to display user feedback	The page is responsive (at least across mobile, tablet and desktops), the color scheme is complementary, and uses properly designed dialog boxes to give the user feedback	User interface is well thought out, resulting in a memorable user experience. UI is functional with captivating aesthetics.