Create a webpage that displays Teams Basic Info

* Create a command-line application that accepts user input
* The user is prompted to enter information on team members
  + a webpage will be generated that displays the team's basic info
    - Email addresses
    - GitHub profiles
* Generate an HTML file that displays a formatted team roster based on user input
* Functionality within HTML
  + Email addresses (clickable)
    - When clicked, default email program opens and populates the TO field of the email with the address
  + GitHub username (clickable)
    - When clicked, GitHub profile opens in a new tab
* When the command-line application is started
  + User prompted to enter team manager’s information:
    - Name
    - employee ID
    - Email address
    - Office phone number
  + Present user with a menu with the option to add:
    - **Engineer** 
      * Then prompt to enter
        + engineer’s name
        + ID number
        + Email Address
        + GitHub username (back to menu after username entered)
    - **Intern**
      * Then prompt to enter
        + Intern’s name
        + ID number
        + Email Address
        + School (back to menu after school name entered)

* + - **Add another team member**
* When finished building team
  + Exit the application
  + HTML is generated

**Grading Requirements**

This Challenge is graded based on the following criteria:

**Deliverables: 15%**

* A sample HTML file generated using the application must be submitted.
* Your GitHub repository containing your application code.

**Walkthrough Video: 32%**

* A walkthrough video that demonstrates the functionality of the Team Profile Generator and passing tests must be submitted, and a link to the video should be included in your README file.
* The walkthrough video must show all four tests passing from the command line.
* The walkthrough video must demonstrate how a user would invoke the application from the command line.
* The walkthrough video must demonstrate how a user would enter responses to all the prompts in the application.
* The walkthrough video must demonstrate a generated HTML file that matches the user input.

**Technical Acceptance Criteria: 40%**

* Satisfies all the preceding acceptance criteria plus the following:
  + Uses the [Inquirer package (Links to an external site.)](https://www.npmjs.com/package/inquirer).
  + Uses the [Jest package (Links to an external site.)](https://www.npmjs.com/package/jest) for a suite of unit tests.
* The application must have these classes: Employee, Manager, Engineer, and Intern.

**Repository Quality: 13%**

* Repository has a unique name.
* Repository follows best practices for file structure and naming conventions.
* Repository follows best practices for class/id naming conventions, indentation, quality comments, etc.
* Repository contains multiple descriptive commit messages.
* Repository contains a high-quality README with description and a link to a walkthrough video.