

1. A description of your minimum viable project - what is it that you absolutely need (and can) deliver by ~April of 2024

A centralized, configurable, form-filling website that allows organizations to create forms and clients to fill in forms from these organizations. The forms will be automatically filled in with relevant information for the client's profile.

2. Detailed descriptions and diagrams of your system, sub-systems/components, and expected user interactions
  - Website homepage with admin and client logins, and option for signing up.
  - Signup page that gathers basic information that will later be used for auto-filling forms.
  - Client dashboard that displays available forms and allows client to edit autofill information.
  - After clicking a form, the form will be auto-filled with the client's known information. The client will then be prompted to fill in any additional information that could not be auto-filled.
  - For each form, there will be buttons to save, cancel, and submit the form.
  - Admin center that can display all the forms filled out and export the data into a formatted file.
  - Database backend that stores all data from the forms and clients autofill data.
  - Database backend for storing login credentials.
3. A rough start of mapping out specific tasks (and dependencies between tasks) that need to be completed in order for you to implement the systems and subsystems you have listed in 2
  - a. Create an interface to create a form
  - b. Create an interface to edit a created form (reuse create?), dependent on A
  - c. Create an interface to fill a form
  - d. Create an interface to view submitted forms
  - e. Create a system to submit a form
  - f. Create dashboard interface
  - g. Design a database to store all form and dashboard information
  - h. Create system to retrieve dashboard information on form update, dependent on B, F, G
  - i. Create system to auto-submit form on dashboard update, dependent on E, F, G
  - j. Create a user account system, differentiating between organizational admin and organizational client accounts
  - k. Design a secure database for credentials to link with J, including permissions and privileges
  - l. Link account privileges to form and dashboard actions, dependent on almost everything
  - m. Create a system to export submitted forms as a readable file, dependent on D
  - n. Create a system to store a non-ready form as a draft, dependent on C, G
  - o. Create a system for organizations to manage clients, depending on J, L

4. You do not need to assign team members to those tasks yet, but you should be thinking about the constraints, dependencies, and requirements (resources?) needed to complete a task
  - a. Need to determine how to host, or whether a localhost-based system would be sufficient for the project
  - b. Need to determine what web server to use, and whether to use a separate client technology (e.g. React client with Apache server)
  - c. Need to determine what database framework to use