

Capstone Two: Step Two -

Write a Project Proposal You're about to tackle step two of your final capstone project! To review the other steps of the project, refer to this document. Project Proposal (2-4 Hours) Step Two is all about fleshing out your project idea. For this step, please write a proposal based on the project idea you agreed on with your mentor. This proposal should be a 1-2 page document that answers the following questions:

1. What tech stack will you use for your final project? We recommend that you use React and Node for this project, however if you are extremely interested in becoming a Python developer you are welcome to use Python/Flask for this project.

React + Node

2. Is the front-end UI or the back-end going to be the focus of your project? Or are you going to make an evenly focused full-stack application?

I want it to be well-rounded, so full-stack, maybe with a little more focus on back-end than front-end, but I would still like it to have a decent UI as well.

3. Will this be a website? A mobile app? Something else?

Website

4. What goal will your project be designed to achieve?

To help healthcare professionals manage patients - it will be a hospital/office management website

5. What kind of users will visit your app? In other words, what is the demographic of your users?

Doctors, patients, and admin - anyone working or going to a hypothetical hospital who may need access to this info

6. What data do you plan on using? How are you planning on collecting your data? You may have not picked your actual API yet, which is fine, just outline what kind of data you would like it to contain. You are welcome to create your own API and populate it with data. If you are using a Python/Flask stack are required to create your own API.

Would need to be probably simulated patient data or would just input a few manually - in a real setting, doctors would just sign in and so would patients

7. In brief, outline your approach to creating your project (knowing that you may not know everything in advance and that these details might change later). Answer questions like the ones below, but feel free to add more information: a. What does your database schema look like? Patient Information Table - Stores patient demographics, contact information, and medical history.

Doctors Table - Contains doctor profiles, and specializations

Patient notes Table - notes from their last visit

Staff Table - this would be the admins who add doctors to the site .b. What kinds of issues might you run into with your API? This is especially important if you are creating your own API, web scraping produces notoriously messy data.

Protecting sensitive data from unauthorized access.

Making sure patients cannot access the admin login

c. Is there any sensitive information you need to secure? Things such as patient info is important to keep safe, as this is personal medical information that will be stored. Also would be important to keep doctor information secure so that they cannot be impersonated, and any login info as well. d. What functionality will your app include? Signup, profiles, and notes from last

visit - it would be a huge app, so appointments and pharmacy might be later, but for now I would like to keep it simple and add on when I have more time. e. What will the user flow look like? [Registration/Login - Secure login process for different user types \(patients, doctors, staff\).](#) [Interfaces for each user type to access relevant functionalities.](#) [Medical Records - access to medical records for patients and authorized personnel.](#) f. What features make your site more than a CRUD app? What are your stretch goals? [Possible stretch goals or add-ons for later integration: Adding pharmacy or appointments](#) Please create a GitHub repository for this Capstone Project and label it accordingly. All code and further documentation associated with this project should be added to this repository. Once you've added your proposal, please submit a link to your repository for your mentor to review. You may have to iterate on this submission several times before it is approved. Once your mentor has approved your proposal, please share your GitHub repository URL on Slack and ask for feedback.

[Notes: Make sure to add doctor timeslot - restrict characters to less than 100 - billing and insurance least priority - 1. User 2. MVP for product? Add features later ERP enterprise result planning - too large, so focus on main parts - CSS is possibility, but tedious - can use bootstrap as well - react bootstrap table - copy in and customize - or Tailwind CSS - for NODE use express - don't combine node and express like flask - make two different repositories - try to order priority of features - UI mockup -](#)