Overall Plans and Considerations for Web Based Tool

Things to consider overall:

* UI/UX design
* Back end
  + Scalable
  + Integrate existing Python code
* Customization Options
  + (see below)

Customization Options:

* Type of clinical note
  + “Initial consultation, follow-up, prostate cancer, lung cancer, on-treatment visit, treatment summary, and more”
* Patient Demographic
  + “Age, Sex, Race, Diagnosis, Specific Treatment”
  + \*Options can be left blank and will be randomized if so\*
* Variability Control
  + Slider or “Low, Medium, High” preset do determine how variable the rephrasing is from the original created template
  + \*Allows the choice of creating a formal or informal document\*
* Disease Site
  + “Prostate, Lung”
  + Users could select a primary disease site and choose from a list of predefined cancer types or other relevant conditions.
* Clinical Sections
  + “History of present illness (HPI), Vitals, social history, medical list, physical exam, imaging results, treatment plan, etc.”
  + \*Users should be able to add or remove sections as needed\*
* Date and Time Parameters
  + “Consultation dates, biopsy dates, etc.”
  + Input or randomization
* Option for chronological notes of treatment period
  + Offer the option for generating chronological synthetic notes over a patient’s entire treatment period (e.g., from diagnosis to follow-up).
  + This could involve a timeline view where users can set the number of notes and how spread out they are over time.

Some Approach Options:

* Frontend
  + Use a framework like React, Vue.js, or Angular for the UI.
  + Maybe include preview features that show what the template may look like while changing data fields.
* Backend
  + Frameworks like Flask or FastAPI are lightweight ways to integrate Python script (Since the note generation tool is made with Python.).
  + Backend would need endpoints to handle requests for generating synthetic notes based on the user inputs. Store or generate templates in a database or as files and allow the backend to randomly populate placeholders with values or use the values provided by users.
* LLM Integration
  + If users want to apply rephrasing, the backend could call an LLM like GPT-4 or another pre-trained model to rephrase the generated note sections.
  + Likely it will be handled by August.
* Testing and Feedback
  + Once an iteration is made, we will need to develop tests to ensure the effectiveness of the site.
  + Could get the help of Shashank on this.