911 notebook

Techstack:

- Front end
  - React + TypeScript
  - Tailwind + shadsc/ui
  - supabase/websockets
  - Supabase password: sjaBKCzE3MONOb1k

- Backend
  - Python or nodejs
  - postgreSQL
    - Store reports as a JSONB
- Real-time sync
  - Supabase
- Speech to text
  - Openai or deepgram(AWS)
- LLM assistant
  - OPENAI api
- AWS Service Integration
  - Login Authentication
  - Gen AI
  - ID processing

https://catalog.us-east-1.prod.workshops.aws/event/dashboard/en-US

Access: 64d7-16b3f1-6a

Roles:

System admin: creates organizations

Dispatcher: starts the police workers' chart
Police-worker: an underling of the police chief who will fill incident reports
Police Chief: the supervisor of the police worker who will review all of their data

Triage nurse: starts the doctor's chart based on the paramedics' info.

ER-doctor: a worker in the ER who handles patient care based off the paramedics' immediate information, meaning they will have access to the paramedics' charts in real time

ER-paramedic: a worker who informs the doctor of the situation and condition of a patient before the ambulance arrives

ER-attending: oversees the entire ER patients and doctors

Product:

We are developing a synchronized application for creating, editing, and completing medical and first responders' medical paperwork using text-to-speech technology and LLM chatbots to expedite the process.

life - cycle:

The system admin creates an org for either police or medical ->
The org is then populated with users by admin ->
user creates a file ->
user edits the file either by using a text-to-speech button or by using their keyboard ->
AI scans and looks for discrepancies ->
if found, prompts the user to add additional information to the file->
file is then saved and able to be accessed or edited again->
The head account of the org will then review the file->
The file will then be locked and only be able to be accessed for viewing.

Epic:

- Create a database that stores input from the users in JSON files that can be displayed on preset incident reports.

- Create the incident report and chart report JSON schema and the ability to display that to the consumer
    - This includes writing and developing most of the front-end for the text editor

- create organizations in which these operations happen within, but are separate
- Create a lifecycle of a document from creation to filled out to filed/locked after the care is complete.

- develop websockets for real-time data transfer

-Have a sign into an account that has access to all other accounts in which they have ownership (police chief over the police-worker), and the information that is stored within them

-Have an account system where each account has access to specific information, such as previous records, which they can edit, create new, and delete from their profile.

- create a text editor to fill out certain regions of data that are needed inside a report JSON, such as date, time,victimLname,victimFname, etc

-Use a text-to-speech framework to read in microphone input and parse the data into a form that JSON can handle

-Use the OPENAI api to prompt the user on information that is missing from the JSON using a chatbot.

-Use the OPENAI api to review JSON for hallucinations.

-export as PDF

Stories: