

Telehealth app:

Description: This app will allow doctors to have online consultations with patients and keeps records of patient medicine prescriptions. Will focus on building functionalities for doctors.

Features:

- 1) online consultations in the form of a forum (exactly like the post-class exercise on comments)
- 2) sign up and login for doctors and users.
- 3) patient can schedule/cancel a date to consult a doctor and view past consultation records. (Many-to-many: users-to-doctors)
- 4) doctor can accept, start, cancel and end a consultation and view past consultations.
- 5) forum-like chatbox for patient and doctor to interaction during consultation.
- 6) doctor can prescribe medicine for the patient (not sure how to prescribe more than 1 medicine without reloading the webpage due to the large variety of medicines out there so checkboxes might not be suitable. Might have to reload the page each time a new medicine is added to the prescription).

Routes:

signup/login:

/login: (GET) to render login form and 2 signup buttons (GET) (/signup?=patient and /signup?identity=doctor)

/login: (POST) ask for email and password to login

/signup?identity=patient: (GET) to render signup form.

/signup: (POST) ask for name, email, password, allergies isdoctor=false to create a patient.

/signup?identity=doctor: (GET) to render signup form.

/signup: (POST) ask for name, email, password, clinics to create a doctor

/logout: (DELETE) Log a user out. Get rid of their cookie.

doctors:

/doctor-dashboard/:id: (GET) to show the dashboard: ongoing consultations (/consultation/:id/edit and /consultations?status=ongoing), consultations to accept(/consultation/:id/edit and /consultations?status=requests), upcoming consultations(/consultation/:id/edit and /consultations?status=accepted), history btn (/consultations?status=ended), profile btn(/profile/:id)

/consultation/:id/edit: (GET) to render the consultation details (consider showing different ejs files based on status?): 1. accept consult btn(PUT) (only show accept button when status is req). 2. Start consult btn (PUT) (only show start btn when status is accepted. Also create a chat entry in database so can access it during consultation.). 3. Description, diagnosis, etc form (PUT) (only show form when status is ongoing). 4. Chat btn (/chat/:id) (only show this btn when status is ongoing).

/consultation/:id: (PUT from /consultation/:id/edit) to update details of the consult.

/consultations?status=requests: (GET) to render list of consult requests.

/consultations?status=accepted: (GET) to render list of accepted consults.

/consultations?status=ongoing: (GET) to render list of ongoing consults.

/consultations?status=ended: (GET) to render list of past consults.

patients:

/patient-dashboard: (GET) to show dashboard. 1. New consult btn (/consultation/new/clinics). 2. Upcoming consultations (/consultations?status=accepted). 3. history btn (/consultations?status=ended). 4. Profile btn (/profile/:id).

/consultation/new/clinics (GET) to show a list of clinics. 1. Select clinic btn (/consultation/new/clinics/:id)

/consultation/new/clinics/:id (GET) to show a list of doctors. 1. Create appt btn (/consultation/new/doctor/:id)

/consultation/new/doctor/:id (GET) to render a form to create a consultation. Form must contain doctor's name and id. 1. Submit btn (POST) (/consultation): to send a post request to create the consultation.

/consultation: (POST) to receive post request to create consultation. Redirect to (/consultation/:id)

Both:

/consultation/:id (GET) render details of consultation. 1. Cancel consult btn (DELETE) (Do app.delete(/consultation/:id). Only show this btn to patient and consultation status of accepted)

/chat/:id: (GET) render a list of messages for the consultation chat. 1. Form to post a message (/chat/:id/message) (hide this form when consult status is ended).

/chat/:id/message: (POST) post a message to the consultation chat.

/profile/:id: (GET) render doctor/patient info (either ejs file for doctor or patient) based on the id found in the database. 1. Edit profile btn (/profile/:id/edit). 2. Logout btn (DELETE) (/logout).

/profile/:id/edit: (GET) render form for doctor/patient info. 1. Update btn(/profile/:id/edit) to put updated info.

/profile/:id/edit: (PUT) update the user info.