Abstract:
This document describes the test task that is related to the clinics site api and administration control system.

Task description:
The main goal is to create an admin panel (with or without UI) that allows the user to:

- Authenticate himself in the system (no registration);
- Create a clinic;
- Create a doctor;
- Create a service that doctor can provide;
- Get a list of clinics populated with services provided in each clinic.

Business logic:
We have five clinics and at those clinics we have doctors. One doctor can work in multiple clinics. Each of these doctors can provide one or more services, for example, traumatology, therapy, vertebrology, etc. As an administrator, I want to manage doctors, clinics, and services. I want to see lists of clinics, doctors, and services and add new entities to those lists.
Clinics list item includes name of clinic and list of services provided by doctors who work in that clinic. Doctors list item includes doctor's name and lists of his/her clinics and services. Services list item includes its name. Clinics list can be filtered by one or multiple categories and doctors.

For example:
We have “Kyiv Vertebrology clinic”, “Kyiv General clinic” and “Kyiv Dental Clinic”. Doctor John Doe works in Vertebrology clinic and General clinic on different days and he is the only vertebrologist in all our clinics. User wants to see all clinics that can provide vertebrology service, so he will see only “Kyiv Vertebrology Clinic” and “Kyiv General Clinic” in the list. The same list will be displayed if the user wants to see all clinics where John Doe works.

Task conditions:
In this task, the developer must use the actual version of NodeJs and at least one database (SQL or noSQL does not matter). The development of the user interface for the application is optional for developer. The developer can choose any libraries that he will use for the task, but please consider not using ORM tools or query builders.

Expected output is the following:

- Git repository with an application code and launch instructions in README;
- README should also contain a list of API endpoints with brief descriptions;
- Database dump in any format applicable for restore (e.g. zipped folder with .json/.bson files in case of mongodb).