

# TechUnicorn Backend Engineer Assessment

Please refer to the following links for references if needed:

- RESTful Api
  - <https://www.youtube.com/watch?v=7YcW25PHnAA>
- JWT Token
  - <https://www.youtube.com/watch?v=K6pwjJ5h0Gg>
  - <https://youtu.be/7Q17ubqLfaM?t=41>
- Query Params vs Route Param
  - <https://www.youtube.com/watch?v=tG7x9Ty1ocg>
- Creating a web server (Note: you are not limited to these technologies and you may pick whichever technology you prefer and more comfortable with)
  - Javascript/TypeScript: Express + Nodejs → <https://www.youtube.com/watch?v=-MTSQjw5DrM>
  - Javascript/TypeScript: NestJs → <https://www.thisdot.co/blog/introduction-to-restful-apis-with-nestjs>
  - Java: Spring Framework → <https://www.youtube.com/watch?v=MWLe1tqPmUo>
  - C#: ASP.Net Core → <https://www.youtube.com/watch?v=sWJayOop4k8>
  - Python: FastAPI → <https://www.youtube.com/watch?v=-ykeT6kk4bk>
  - Python: Django → <https://www.youtube.com/watch?v=TmsD8QExZ84>
  - GoLang: Gin-Gonic → <https://www.youtube.com/watch?v=RkmvVFZJJvs>
  - GoLang: Fiber → <https://www.youtube.com/watch?v=lq2qT0fRhAA>
  - Ruby : Rails → <https://www.youtube.com/watch?v=3S9fyfmCf1A>
  - Elixir: Phoenix → <https://www.youtube.com/watch?v=fqbcngTBR7E>
  - Kotlin: Spring → <https://www.youtube.com/watch?v=coM6wWKau5E>
  - Swift: Vapor → [https://www.youtube.com/watch?v=G\\_4YfwJHcKI](https://www.youtube.com/watch?v=G_4YfwJHcKI)

These Links are there for reference, however you are not tied to using these technologies only, You may chose any technology of your choosing as long as the tasks below are met

You may use **ANY** Language that you feel comfortable with develop the web server. You must integrate this solution with **ANY** database of your choosing and are not tied to any choice or schema.

The Deadline to submit the assignment is **7 Days**

Note: You will be quizzed on the choices you make and the reason behind the technologies you used

# Problem Statement

A clinic would like a Web Application to schedule meetings between doctors and their patients. Your Task as a backend engineer is to develop the RESTful APIs that the Frontend Team will use in their web app. The actors interacting with this application are:

- Doctor
- Clinic Admins
- Patients

Patients and doctors are able to see the doctors' available schedule using the following endpoint `/doctors/:doctorId/slots`

⇒ note: when a **doctor** uses that endpoint they are able to see the patient name, However when a patient uses it the patient name is not sent in the response.

Patients are able to book an appointment with a doctor.

Minimum Duration: 15 mins, Max Duration: 2 Hours

Doctors are expected to have a maximum of 12 different patients Or 8 hours maximum total appointments per day

patients are able to see all the available doctors in the clinic

clinic administrators help in managing doctors and are able to see our list of doctors with the most appointments on the day as well as a list of doctors who have total appointments exceeding six hours and the ability to cancel appointments on behalf of the doctor.

## Task List:

- ☐ Setup Authentication for the accounts
  - ☐ Each account must be authenticated to be able to use the app
  - ☐ Each account must be able to register
- ☐ Patients and Doctors and Clinic Admins are Able to see any given doctors schedule.
  - ☐ Patient info must be hidden to Patient users but shown to doctors and Clinic Admins
- ☐ Patients must be able to book an appointment with a doctor
  - ☐ a doctor can have a maximum of 12 appointments
  - ☐ a doctor can have a maximum total appointments time of 8 hours in a day
- ☐ Doctors must be able to see the appointment details
  - ☐ the patient who booked the appointment is also able to see the appointment details

- ☐ Patients must be able a history of their previous appointments
  - ☐ Doctors must also be able to see the patient's appointment history
- ☐ Doctors and Clinic admins can cancel appointments
- ☐ Clinic Admins must able to view the the availability of **ALL** doctors in the clinic
  - ☐ Patients are able to see the list of all available doctors
- ☐ Clinic Admins must be able to see which doctors have the most appointments in a given day
  - ☐ (Bonus) viewing a sorted list of doctors who are
- ☐ Clinic Admins must be able to view which doctors have total appointments in a day exceeding 6 hours.

## Bouns Points:

- Develop the Postman Test APIs testing each api endpoint
- Allowing/restricting access to APIs based on provided JWT Token
- Write unit tests for your functions
- Properly document your code
- Implement GraphQL
- Stating how will the application would be deployed
- Sequence Diagrams and swim-lane diagrams of your application
- Any other way show your backend skills

## Endpoints

Endpoint	Requirement	Response	Access(Bonus)
/register	allow all users to register	*	All
/login	allow users to login	*	All
/doctors	View list of doctors	*	All
/doctors/id	View Doctor information.	*	All
/doctors/:doctorId/slots	View Doctors available slots	*	**
*	Book an appointment with a doctor	*	Patient only
*	cancel appointment	*	**
*	View availability of all Doctors	*	**
*	view appointment details	*	**
*	View patient appointment history	*	**

Endpoint	Requirement	Response	Access(Bonus)
*	view doctors with the most appointments in a given day	*	**
*	view doctors who have 6+ hours total appointments in a day	*	**

(\*) Up to Your Choosing, However will be asked about this in your technical interview

(\*\*) Bouns, Will be assessed on if implemented

Should you have any questions, concerns or face any issues please contact us.

**Please Submit a GitHub link of your assignment.**